# *MSPM0 G-Series 80MHz Microcontrollers*

## *Technical Reference Manual*

Literature Number: SLAU846E
JUNE 2023 – REVISED JULY 2026

# Table of Contents


**Read This First** ..... 13
About This Manual ..... 13
Notational Conventions ..... 13
Glossary ..... 13
Support Resources ..... 13
Trademarks ..... 13
**1 Architecture** ..... 14
1.1 Architecture Overview ..... 15
1.2 Bus Organization ..... 15
1.3 Platform Memory Map ..... 17
1.3.1 Code Region ..... 17
1.3.2 SRAM Region ..... 17
1.3.3 Peripheral Region ..... 20
1.3.4 Subsystem Region ..... 20
1.3.5 System PPB Region ..... 20
1.4 Boot Configuration ..... 20
1.4.1 Configuration Memory (NONMAIN) ..... 20
1.4.2 Boot Configuration Routine (BCR) ..... 22
1.4.3 Bootstrap Loader (BSL) ..... 28
1.4.4 NONMAIN Layout Types ..... 30
1.4.5 NONMAIN_TYPEA Registers ..... 31
1.4.6 NONMAIN_TYPEE Registers ..... 60
1.4.7 NONMAIN_TYPEF Registers ..... 101
1.5 Factory Constants ..... 140
1.5.1 FACTORYREGION Layout Types ..... 140
1.5.2 FACTORYREGION_TYPEA Registers ..... 142
1.5.3 FACTORYREGION_TYPEE Registers ..... 161
1.5.4 FACTORYREGION_TYPEF Registers ..... 181
1.5.5 FACTORYREGION_TYPEG Registers ..... 209
**2 PMCU** ..... 228
2.1 PMCU Overview ..... 229
2.1.1 Power Domains ..... 230
2.1.2 Operating Modes ..... 230
2.2 Power Management (PMU) ..... 234
2.2.1 Power Supply ..... 234
2.2.2 Core Regulator ..... 234
2.2.3 Supply Supervisors ..... 234
2.2.4 Bandgap Reference ..... 236
2.2.5 VBOOST for Analog Muxes ..... 236
2.2.6 Peripheral Enable ..... 239
2.3 Clock Module (CKM) ..... 240
2.3.1 Oscillators ..... 240
2.3.2 Clocks ..... 252
2.3.3 Clock Tree ..... 261
2.3.4 Clock Monitors ..... 263
2.3.5 Frequency Clock Counter (FCC) ..... 266
2.4 System Controller (SYSCTL) ..... 269
2.4.1 Resets and Device Initialization ..... 269
2.4.2 Operating Mode Selection ..... 277





Copyright © 2026 Texas Instruments Incorporated

2.4.3 Asynchronous Fast Clock Requests....279
2.4.4 SRAM Write Protection....281
2.4.5 Flash Wait States....281
2.4.6 Flash Bank Address Swap....282
2.4.7 Shutdown Mode Handling (if present)....282
2.4.8 Configuration Lockout....282
2.4.9 System Status....283
2.4.10 Error Handling....283
2.4.11 SYSCTL Events....285
2.5 SYSCTL_G351x_G151x_G321x_G320x_G121x_G120x_G122x_G322x Registers....287
2.6 SYSCTL_G351x_G151x Registers....347
2.7 SYSCTL_G511x_G5187 Registers....407
2.8 Quick Start Reference....466
2.8.1 Default Device Configuration....466
2.8.2 Leveraging MFCLK....467
2.8.3 Optimizing Power Consumption in STOP Mode....467
2.8.4 Optimizing Power Consumption in STANDBY Mode....467
2.8.5 Increasing MCLK Precision....467
2.8.6 Configuring MCLK for Maximum Speed....468
2.8.7 High Speed Clock (SYSPLL, HFCLK) Handling in Low-Power Modes....468
2.8.8 Optimizing for Lowest Wakeup Latency....468
2.8.9 Optimizing for Lowest Peak Current in RUN/SLEEP Mode....469
**3 CPU**....470
3.1 Overview....471
3.2 Arm Cortex-M0+ CPU....471
3.2.1 CPU Register File....472
3.2.2 Stack Behavior....475
3.2.3 Execution Modes and Privilege Levels....475
3.2.4 Address Space and Supported Data Sizes....475
3.3 Interrupts and Exceptions....476
3.3.1 Peripheral Interrupts (IRQs)....477
3.3.2 Interrupt and Exception Table....480
3.3.3 Processor Lockup Scenario....483
3.4 CPU Peripherals....483
3.4.1 System Control Block (SCB)....483
3.4.2 System Tick Timer (SysTick)....484
3.4.3 Memory Protection Unit (MPU)....485
3.5 Read-Only Memory (ROM)....486
3.6 CPUSS Registers....487
3.7 WUC Registers....502
**4 SECURITY**....504
4.1 Overview....504
4.1.1 Secure Boot....505
4.1.2 Customer Secure Code (CSC)....505
4.2 Boot and Startup Sequence....505
4.2.1 CSC Programming Overview....506
4.3 Secure Key Storage....507
4.4 Flash Memory Protection....508
4.4.1 Bank Swapping....508
4.4.2 Write Protection....508
4.4.3 Read-Execute Protection....509
4.4.4 IP Protection....509
4.4.5 Data Bank Protection....510
4.4.6 Hardware Monotonic Counter....510
4.5 SRAM Protection....510
4.6 SECURITY Registers....512
**5 Direct Memory Access (DMA)**....523
5.1 DMA Overview....524
5.2 DMA Operation....525
5.2.1 Addressing Modes....525
5.2.2 Channel Types....527



Copyright © 2026 Texas Instruments Incorporated

5.2.3 Transfer Modes....528
5.2.4 Extended Modes....530
5.2.5 Initiating DMA Transfers....532
5.2.6 Stopping DMA Transfers....532
5.2.7 Channel Priorities....532
5.2.8 Burst Block Mode....533
5.2.9 Using DMA with System Interrupts....533
5.2.10 DMA Controller Interrupts....533
5.2.11 DMA Trigger Event Status....533
5.2.12 DMA Operating Mode Support....534
5.2.13 DMA Address and Data Errors....534
5.2.14 Interrupt and Event Support....535
5.3 DMA Registers....536
**6 NVM (Flash)**....584
6.1 NVM Overview....585
6.1.1 Key Features....585
6.1.2 System Components....585
6.1.3 Terminology....585
6.2 Flash Memory Bank Organization....586
6.2.1 Banks....586
6.2.2 Flash Memory Regions....586
6.2.3 Addressing....586
6.2.4 Memory Organization Examples....587
6.3 Flash Controller....588
6.3.1 Overview of Flash Controller Commands....589
6.3.2 NOOP Command....589
6.3.3 PROGRAM Command....589
6.3.4 ERASE Command....594
6.3.5 READVERIFY Command....595
6.3.6 BLANKVERIFY Command....596
6.3.7 Command Diagnostics....597
6.3.8 Overriding the System Address With a Bank ID, Region ID, and Bank Address....597
6.3.9 FLASHCTL Events....598
6.4 Write Protection....598
6.4.1 Write Protection Resolution....598
6.4.2 Static Write Protection....598
6.4.3 Dynamic Write Protection....599
6.5 Read Interface....600
6.5.1 Bank Address Swapping....600
6.5.2 ECC Error Handling....600
6.6 FLASHCTL Registers....601
**7 Events**....640
7.1 Events Overview....641
7.1.1 Event Publisher....641
7.1.2 Event Subscriber....641
7.1.3 Event Fabric Routing....641
7.1.4 Event Routing Map....643
7.1.5 Event Propagation Latency....644
7.2 Events Operation....645
7.2.1 CPU Interrupt....645
7.2.2 DMA Trigger....645
7.2.3 Peripheral to Peripheral Event....646
7.2.4 Extended Module Description Register....647
7.2.5 Using Event Registers....647
**8 IOMUX**....650
8.1 IOMUX Overview....651
8.1.1 IO Types and Analog Sharing....651
8.2 IOMUX Operation....654
8.2.1 Peripheral Function (PF) Assignment....654
8.2.2 Logic High to Hi-Z Conversion....654
8.2.3 Logic Inversion....655



Copyright © 2026 Texas Instruments Incorporated

8.2.4 SHUTDOWN Mode Wakeup Logic....655
8.2.5 Pullup/Pulldown Resistors....656
8.2.6 Drive Strength Control....656
8.2.7 Hysteresis and Logic Level Control....656
8.3 IOMUX Registers....658
**9 General-Purpose Input/Output (GPIO)**....661
9.1 GPIO Overview....662
9.2 GPIO Operation....662
9.2.1 GPIO Ports....663
9.2.2 GPIO Read/Write Interface....663
9.2.3 GPIO Input Glitch Filtering and Synchronization....663
9.2.4 GPIO Fast Wake....664
9.2.5 GPIO DMA Interface....665
9.2.6 Event Publishers and Subscribers....665
9.3 GPIO Registers....666
**10 MATHACL**....774
10.1 Overview....774
10.2 Data Format....774
10.2.1 Unsigned 32-bit integers....774
10.2.2 Signed 32-bit integers....775
10.2.3 Unsigned 32-bit numbers....775
10.2.4 Signed 32-bit numbers....775
10.3 Basic Operation....776
10.4 Configuration Details with Examples....776
10.4.1 Sine and Cosine (SINCOS)....776
10.4.2 Arc Tangent (ATAN2)....777
10.4.3 Square Root (SQRT)....779
10.4.4 Division (DIV)....780
10.4.5 Multiplication....782
10.4.6 Multiply-Accumulate (MAC)....786
10.4.7 Square Accumulate (SAC)....788
10.5 MATHACL Registers....790
**11 TinyEngine™ NPU**....802
11.1 Overview....802
11.2 NPU Operation....802
11.3 NPU Toolchain....804
**12 AES**....806
12.1 AES Overview....807
12.1.1 AES Performance....807
12.2 AES Operation....808
12.2.1 AES Register Access Rules....809
12.2.2 Loading the Key....810
12.2.3 Loading Data....810
12.2.4 Reading Data....810
12.2.5 Triggering an Encryption or Decryption....810
12.2.6 Single Block Operations....810
12.2.7 Block Cipher Mode Operations....813
12.2.8 AES Events....826
12.3 AES Registers....829
**13 AESADV**....870
13.1 AES Overview....871
13.1.1 AESADV Performance....872
13.2 AESADV Operation....872
13.2.1 Loading the Key....872
13.2.2 Writing Input Data....872
13.2.3 Reading Output Data....872
13.2.4 Operation Descriptions....873
13.2.5 AES Events....891
13.3 AESADV Registers....893
**14 CRC**....963
14.1 CRC Overview....964



Copyright © 2026 Texas Instruments Incorporated

14.1.1 CRC16-CCITT ... 964
14.1.2 CRC32-ISO3309 ... 964
14.2 CRC Operation ... 964
14.2.1 CRC Generator Implementation ... 965
14.2.2 Configuration ... 965
14.3 CRCP0 Registers ... 967
**15 Keystore** ... 980
15.1 Overview ... 981
15.2 Detailed Description ... 981
15.3 KEYSTORECTL Registers ... 982
**16 TRNG** ... 989
16.1 TRNG Overview ... 990
16.2 TRNG Operation ... 990
16.2.1 TRNG Generation Data Path ... 990
16.2.2 Clock Configuration and Output Rate ... 991
16.2.3 Behavior in Low Power Modes ... 991
16.2.4 Health Tests ... 991
16.2.5 Configuration ... 993
16.3 TRNG Registers ... 997
**17 Temperature Sensor** ... 1014
**18 ADC** ... 1016
18.1 ADC Overview ... 1017
18.2 ADC Operation ... 1018
18.2.1 ADC Core ... 1019
18.2.2 Voltage Reference Options ... 1019
18.2.3 Generic Resolution Modes ... 1020
18.2.4 Hardware Averaging ... 1020
18.2.5 ADC Clocking ... 1021
18.2.6 Common ADC Use Cases ... 1021
18.2.7 Power Down Behavior ... 1023
18.2.8 Sampling Trigger Sources and Sampling Modes ... 1023
18.2.9 Sampling Period ... 1025
18.2.10 Conversion Modes ... 1026
18.2.11 Data Format ... 1027
18.2.12 Advanced Features ... 1027
18.2.13 Status Register ... 1032
18.2.14 ADC Events ... 1032
18.3 ADC12 Registers ... 1036
**19 COMP** ... 1113
19.1 Comparator Overview ... 1114
19.2 Comparator Operation ... 1119
19.2.1 Comparator Configuration ... 1119
19.2.2 Comparator Channels Selection ... 1119
19.2.3 Comparator Output ... 1121
19.2.4 Output Filter ... 1121
19.2.5 Sampled Output Mode ... 1121
19.2.6 Blanking Mode ... 1122
19.2.7 Reference Voltage Generator ... 1123
19.2.8 Window Comparator Mode ... 1125
19.2.9 Comparator Hysteresis ... 1126
19.2.10 Input SHORT Switch ... 1127
19.2.11 Interrupt and Events Support ... 1128
19.3 COMP0 Registers ... 1131
**20 DAC** ... 1160
20.1 DAC Introduction ... 1161
20.2 DAC Operation ... 1162
20.2.1 DAC Core ... 1162
20.2.2 DAC Output ... 1162
20.2.3 DAC Voltage Reference ... 1162
20.2.4 DAC Output Buffers ... 1162
20.2.5 DAC Data Formats ... 1163



Copyright © 2026 Texas Instruments Incorporated

20.2.6 Sample Time Generator....1163
20.2.7 DAC FIFO Structure....1164
20.2.8 DAC Operation With DMA Controller....1164
20.2.9 DAC Operation With CPU....1166
20.2.10 Data Register Format....1166
20.2.11 DAC Output Amplifier Offset Calibration....1167
20.2.12 Interrupt and Event Support....1168
20.3 DAC12 Registers....1171
**21 OPA**....1201
21.1 OPA Overview....1202
21.2 OPA Operation....1203
21.2.1 Analog Core....1203
21.2.2 Power Up Behavior....1204
21.2.3 Inputs....1204
21.2.4 Output....1204
21.2.5 Clock Requirements....1204
21.2.6 Chopping....1205
21.2.7 OPA Amplifier Modes....1205
21.2.8 OPA Configuration Selection....1211
21.2.9 Burnout Current Source....1212
21.3 OA Registers....1213
**22 GPAMP**....1223
22.1 GPAMP Overview....1224
22.2 GPAMP Operation....1224
22.2.1 Analog Core....1224
22.2.2 Power Up Behavior....1225
22.2.3 Inputs....1225
22.2.4 Output....1225
22.2.5 GPAMP Amplifier Modes....1225
22.2.6 Chopping....1227
22.3 GPAMP Registers....1227
**23 VREF**....1228
23.1 VREF Overview....1229
23.2 VREF Operation....1230
23.2.1 Internal Reference Generation....1231
23.2.2 External Reference Input....1231
23.2.3 Analog Peripheral Interface....1232
23.2.4 Sample and Hold Mode....1232
23.3 VREF Registers....1233
**24 UART**....1242
24.1 UART Overview....1243
24.1.1 Purpose of the Peripheral....1243
24.1.2 Features....1243
24.1.3 Functional Block Diagram....1244
24.2 UART Operation....1245
24.2.1 Clock Control....1245
24.2.2 Signal Descriptions....1245
24.2.3 General Architecture and Protocol....1245
24.2.4 Low Power Operation....1260
24.2.5 Reset Considerations....1261
24.2.6 Initialization....1261
24.2.7 Interrupt and Events Support....1261
24.2.8 Emulation Modes....1264
24.3 UART Registers....1265
**25 $I^2C$**....1322
25.1 $I^2C$ Overview....1323
25.1.1 Purpose of the Peripheral....1323
25.1.2 Features....1323
25.1.3 Functional Block Diagram....1324
25.1.4 Environment and External Connections....1324
25.2 $I^2C$ Operation....1325



Copyright © 2026 Texas Instruments Incorporated

25.2.1 Clock Control....1325
25.2.2 Signal Descriptions....1326
25.2.3 General Architecture....1326
25.2.4 Protocol Descriptions....1335
25.2.5 Reset Considerations....1346
25.2.6 Initialization....1347
25.2.7 Interrupt and Events Support....1347
25.2.8 Emulation Modes....1349
25.3 I2C Registers....1350
**26 SPI**....1414
26.1 SPI Overview....1415
26.1.1 Purpose of the Peripheral....1415
26.1.2 Features....1415
26.1.3 Functional Block Diagram....1416
26.1.4 External Connections and Signal Descriptions....1416
26.2 SPI Operation....1418
26.2.1 Clock Control....1418
26.2.2 General Architecture....1418
26.2.3 Protocol Descriptions....1423
26.2.4 Reset Considerations....1428
26.2.5 Initialization....1428
26.2.6 Interrupt and Events Support....1428
26.2.7 Emulation Modes....1430
26.3 SPI Registers....1431
**27 UNICOMM**....1474
27.1 Overview....1475
27.1.1 Block Diagram....1477
27.2 Unicomm Architecture....1477
27.2.1 Serial Peripheral Group (SPG) Configurations....1477
27.2.2 Enables & Resets....1478
27.3 High-Level Initialization....1479
27.4 UNICOMM/SPGSS Registers....1479
27.4.1 UNICOMM Registers....1479
27.4.2 SPG Registers....1485
**28 UNICOMM-UART**....1495
28.1 UNICOMM-UART Overview....1496
28.1.1 Purpose of the Peripheral....1496
28.1.2 Features....1496
28.1.3 Functional Block Diagram....1497
28.2 UART Operation....1497
28.2.1 Clock Control....1497
28.2.2 General Architecture and Protocol....1498
28.2.3 Additional Protocol and Feature Support....1507
**29 UNICOMMUART Registers**....1523
**30 UNICOMM-I2C**....1570
30.1 UNICOMM-I2C Overview....1571
30.1.1 Purpose of the Peripheral....1571
30.1.2 Features....1571
30.1.3 Functional Block Diagram....1572
30.1.4 Environment and External Connections....1573
30.2 UNICOMM Common Infrastructure....1574
30.3 Peripheral Functional Description....1574
30.3.1 Clock Control....1574
30.3.2 Signal Descriptions....1575
30.3.3 General Architecture....1575
30.3.4 Protocol Descriptions & Initialization....1591
30.3.5 Reset Considerations....1603
30.3.6 Initialization....1604
30.3.7 Interrupt and Events Support....1604
30.3.8 Emulation Modes....1607
30.4 UNICOMM I2C Registers....1608



Copyright © 2026 Texas Instruments Incorporated

30.4.1 UNICOMMI2CC Registers....1609
30.4.2 UNICOMMI2CT Registers....1651
**31 UNICOMM-SPI**....1693
31.1 UNICOMM-SPI Overview....1694
31.1.1 Purpose of the Peripheral....1694
31.1.2 Features....1694
31.1.3 Functional Block Diagram....1695
31.1.4 External Connections and Signal Descriptions....1695
31.2 SPI Operation....1697
31.2.1 Clock Frequency Support....1697
31.2.2 General Architecture....1698
31.2.3 FIFO Operation....1700
31.2.4 Suspend communication....1702
31.2.5 Internal Loopback Operation....1702
31.2.6 Repeat Transfer mode....1703
31.2.7 Receive Timeout....1703
31.2.8 Line Timeout....1704
31.2.9 Protocol Descriptions....1704
31.2.10 Status Flags....1709
31.2.11 Module configuration....1709
31.2.12 Reset Considerations....1709
31.2.13 Initialization....1709
31.2.14 Interrupt and Events Support....1710
31.2.15 Emulation Modes....1712
31.3 UNICOMMSPI Registers....1714
**32 CAN-FD**....1748
32.1 MCAN Overview....1749
32.1.1 MCAN Features....1750
32.2 MCAN Environment....1750
32.3 CAN Network Basics....1751
32.4 MCAN Functional Description....1753
32.4.1 Clock Setup....1754
32.4.2 Module Clocking Requirements....1755
32.4.3 Interrupt Requests....1755
32.4.4 Operating Modes....1757
32.4.5 Software Initialization....1759
32.4.6 Transmitter Delay Compensation....1760
32.4.7 Restricted Operation Mode....1762
32.4.8 Bus Monitoring Mode....1763
32.4.9 Disabled Automatic Retransmission (DAR) Mode....1764
32.4.10 Clock Stop Mode....1764
32.4.11 Test Modes....1770
32.4.12 Timestamp Generation....1771
32.4.13 Timeout Counter....1772
32.4.14 Safety....1773
32.4.15 Tx Handling....1775
32.4.16 Rx Handling....1785
32.4.17 Rx FIFOs....1789
32.4.18 Dedicated Rx Buffers....1791
32.4.19 Message RAM....1791
32.5 MCAN Integration....1802
32.6 Interrupt and Event Support....1803
32.6.1 CPU Interrupt Event Publisher (CPU_INT)....1803
32.7 MCAN Registers....1805
**33 USB**....1935
33.1 USB Overview....1936
33.1.1 Features....1936
33.1.2 USB Related Collateral....1936
33.1.3 Block Diagram....1936
33.1.4 VBus Recommendations....1937
33.2 Functional Description....1939



Copyright © 2026 Texas Instruments Incorporated

33.2.1 Operation as a Device ... 1939
33.2.2 Operation as a Host ... 1944
33.2.3 DMA Operation ... 1948
33.3 Initialization and Configuration ... 1948
33.3.1 Pin Configuration ... 1948
33.3.2 Endpoint Configuration ... 1948
33.4 Interrupt and Event Support ... 1948
33.5 USB Registers ... 1950
**34 I2S/TDM** ... 2050
34.1 I2S/TDM Introduction ... 2051
34.1.1 I2S/TDM features ... 2051
34.2 I2S/TDM Operation ... 2051
34.2.1 Functional Block Diagram ... 2051
34.2.2 Modes of Operation ... 2051
34.2.3 Clock and Timing Control ... 2052
34.2.4 Frame Synchronization ... 2053
34.2.5 Slot Mapping and Configuration ... 2055
34.2.6 Serial Frame Format Examples ... 2058
34.2.7 Initialization ... 2062
34.2.8 Disabling I2S ... 2063
34.2.9 Interrupts and Events Support ... 2063
34.2.10 Emulation Modes ... 2064
34.3 I2S Registers ... 2065
**35 Timers (TIMx)** ... 2103
35.1 TIMx Overview ... 2104
35.1.1 TIMG Overview ... 2104
35.1.2 TIMA Overview ... 2105
35.1.3 TIMx Instance Configuration ... 2107
35.2 TIMx Operation ... 2107
35.2.1 Timer Counter ... 2108
35.2.2 Counting Mode Control ... 2110
35.2.3 Capture/Compare Module ... 2116
35.2.4 Shadow Load and Shadow Compare ... 2130
35.2.5 Output Generator ... 2133
35.2.6 Fault Handler (TIMA only) ... 2143
35.2.7 Synchronization With Cross Trigger ... 2149
35.2.8 Low Power Operation ... 2152
35.2.9 Interrupt and Event Support ... 2152
35.2.10 Debug Handler (TIMA Only) ... 2155
35.3 TIMx Registers ... 2157
**36 TIMB** ... 2251
36.1 TIMB Overview ... 2252
36.1.1 Features ... 2252
36.1.2 TIMB Block Diagram ... 2252
36.2 TIMB Operation ... 2253
36.2.1 Counter Block Operation ... 2253
36.3 TIMB Example Applications ... 2258
36.3.1 Periodic Interrupt Generation ... 2258
36.3.2 Counter Chaining ... 2259
36.3.3 Event Counting ... 2260
36.3.4 Event Duration Measurement ... 2261
36.3.5 Event Sequence Checking ... 2262
36.3.6 PWM Generation ... 2262
36.4 TIMB Registers ... 2264
**37 Low Frequency Subsystem (LFSS_B)** ... 2298
37.1 Overview ... 2299
37.2 Clock System ... 2299
37.3 LFSS Reset ... 2300
37.4 Real Time Counter (RTC_x) ... 2300
37.5 Independent Watchdog Timer (IWDT) ... 2301
37.6 Lock Function of RTC and IWDT ... 2301



Copyright © 2026 Texas Instruments Incorporated

37.7 LFSS Registers ... 2302
**38 RTC** ... 2423
38.1 Overview ... 2424
38.1.1 RTC Instances ... 2424
38.2 Basic Operation ... 2425
38.3 Configuration ... 2427
38.3.1 Clocking ... 2427
38.3.2 Reading and Writing to RTC Peripheral Registers ... 2427
38.3.3 Binary vs. BCD ... 2428
38.3.4 Leap Year Handling ... 2428
38.3.5 Calendar Alarm Configuration ... 2428
38.3.6 Interval Alarm Configuration ... 2429
38.3.7 Periodic Alarm Configuration ... 2430
38.3.8 Calibration ... 2430
38.3.9 RTC Prescaler Extension ... 2433
38.3.10 RTC Timestamp Capture ... 2434
38.3.11 RTC Events ... 2436
38.4 RTC Registers ... 2438
**39 IWDT** ... 2505
39.1 IWDT Overview ... 2506
39.2 IWDT Clock Configuration ... 2506
39.3 IWDT Period Selection ... 2506
39.4 Debug Behavior of the IWDT ... 2507
39.5 IWDT Registers ... 2507
**40 Window Watchdog Timer (WWDT)** ... 2508
40.1 WWDT Overview ... 2509
40.1.1 Watchdog Mode ... 2509
40.1.2 Interval Timer Mode ... 2509
40.2 WWDT Operation ... 2510
40.2.1 Mode Selection ... 2510
40.2.2 Clock Configuration ... 2510
40.2.3 Low-Power Mode Behavior ... 2512
40.2.4 Debug Behavior ... 2512
40.2.5 WWDT Events ... 2512
40.3 WWDT Registers ... 2514
**41 Debug** ... 2532
41.1 DEBUGSS Overview ... 2533
41.1.1 Debug Interconnect ... 2533
41.1.2 Physical Interface ... 2534
41.1.3 Debug Access Ports ... 2535
41.2 DEBUGSS Operation ... 2536
41.2.1 Debug Features ... 2536
41.2.2 Behavior in Low Power Modes ... 2537
41.2.3 Restricting Debug Access ... 2538
41.2.4 Mailbox (DSSM) ... 2540
41.3 DEBUGSS Registers ... 2544
**42 Revision History** ... 2560



Copyright © 2026 Texas Instruments Incorporated

# *Read This First*

## About This Manual

This manual describes the modules and peripherals of the MSPM0G family of devices. Each description presents the module or peripheral in a general sense. Not all features and functions of all modules or peripherals are present on all devices. In addition, modules or peripherals can differ in their exact implementation on different devices. Pin functions, internal signal connections, and operational parameters differ from device to device. See the device-specific data sheet for these details.

## Notational Conventions

This document uses the following conventions.

- Hexadecimal numbers can be shown with the suffix h or the prefix 0x. For example, the following number is 40 hexadecimal (decimal 64): 40h or 0x40.
- Registers in this document are shown in figures and described in tables.
  - Each register figure shows a rectangle divided into fields that represent the fields of the register. Each field is labeled with its bit name, its beginning and ending bit numbers above, and its read/write properties with default reset value below. A legend explains the notation used for the properties.
  - Reserved bits in a register figure can have one of multiple meanings:
    - Not implemented on the device
    - Reserved for future device expansion
    - Reserved for TI testing
    - Reserved configurations of the device that are not supported
  - Writing nondefault values to the Reserved bits could cause unexpected behavior and should be avoided.

## Glossary

TI Glossary — This glossary lists and explains terms, acronyms, and definitions.

## Support Resources

TI E2E™ support forums are an engineer's go-to source for fast, verified answers and design help — straight from the experts. Search existing answers or ask your own question to get the quick design help you need.

Linked content is provided "AS IS" by the respective contributors. They do not constitute TI specifications and do not necessarily reflect TI's views; see TI's Terms of Use.

## Trademarks

TI E2E™ and EnergyTrace™ are trademarks of Texas Instruments.
USB Specification Revision 2.0™ is a trademark of Compaq Computer Corp.
Arm® and Cortex® are registered trademarks of Arm Limited.
All trademarks are the property of their respective owners.



Copyright © 2026 Texas Instruments Incorporated

Chapter 1

# Architecture

The device architecture includes the bus organization, the platform memory map, and the boot configuration.


**1.1 Architecture Overview** ..... 15
**1.2 Bus Organization** ..... 15
**1.3 Platform Memory Map** ..... 17
**1.4 Boot Configuration** ..... 20
**1.5 Factory Constants** ..... 140





Copyright © 2026 Texas Instruments Incorporated

## 1.1 Architecture Overview

MSPM0 G-series MCUs (MSPM0Gxx) combine 32-bit compute performance together with precision analog to enable a wide variety of sensing, interface, control, and housekeeping applications. The device architecture supports both high-performance and low-power applications through a flexible, easy-to-configure power management and clocking system.

MSPM0 G-series devices also offer enhanced robustness with ECC-protected flash memory, parity-protected SRAM, available dual window-watchdog timers, and support for 125°C ambient temperature and AEC-Q100 Grade 1 qualification.

This chapter introduces the device architecture, including an overview of the power domains and bus organization, the platform memory map, and the device boot configuration.

## 1.2 Bus Organization

There are three main power domains on MSPM0Gxx devices:

- PD1 (power domain 1) which contains the CPU subsystem, memory interfaces, and high-speed peripherals
- PD0 (power domain 0) which contains the low-speed low-power peripherals
- The supply voltage (VDD) which powers IOs, analog modules, and limited logic directly from the supply

The PD1 domain supports higher clock speeds for performance, andis disabled in certain operating modes to minimize power consumption. The PD0 domain supports ultra-low-power performance and is always enabled in operating modes in which the core regulator is operating.

There are four main data buses on MSPM0Gxx devices:

- The AHB bus matrix, which interfaces the CPU to the device memory systems (ROM, SRAM, and flash memory) and the peripheral buses
- The PD1 (power domain 1) CPU-only peripheral bus, clocked from MCLK
- The PD1 (power domain 1) peripheral bus, clocked from MCLK
- The PD0 (power domain 0) peripheral bus, clocked from ULPCLK

The CPU and the DMA controller are the only two bus controllers in the device. Arbitration between the CPU and the DMA for shared peripherals happens between the CPU-only PD1 peripheral bus and the CPU/DMA PD1 peripheral bus. The DMA does not have access to peripherals on the CPU-only PD1 peripheral bus or the CPU bus matrix (the green components in the bus diagram). As such, the CPU can access peripherals on the CPU-only PD1 peripheral bus at the same time that the DMA is processing a transaction on the PD1 or PD0 bus.

Likewise, the CPU can access SRAM or flash memory through the AHB bus matrix at the same time that the DMA is processing a transaction, so long as the DMA is not accessing the same memory that the CPU is attempting to access. Arbitration between the CPU and the DMA for memory systems (SRAM or flash memory) happens at the memory interface itself. All arbitration between the CPU and DMA is done on a round-robin basis.



Copyright © 2026 Texas Instruments Incorporated

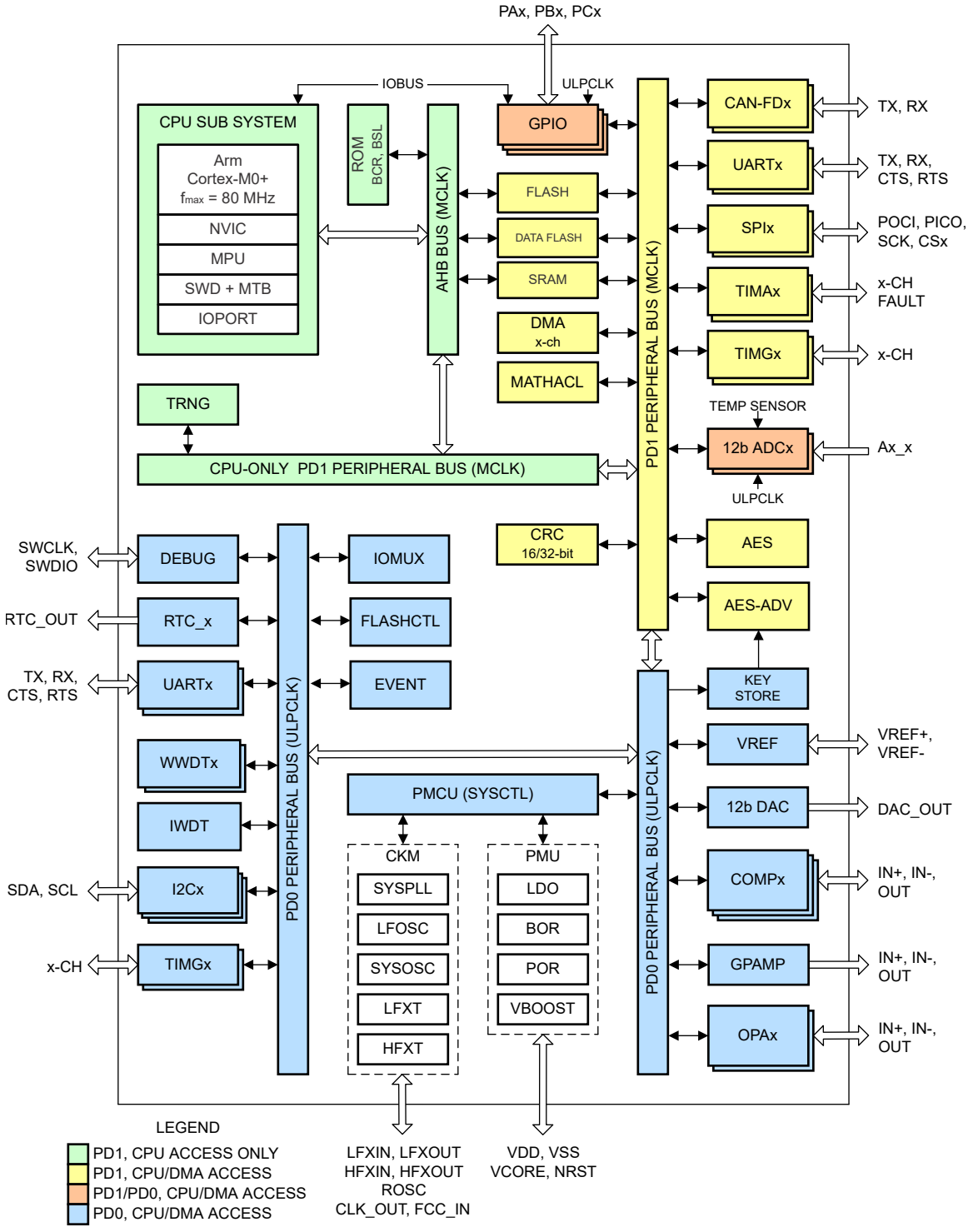



**Figure 1-1. MSPM0Gxx Bus Organization**



Copyright © 2026 Texas Instruments Incorporated

**Note**

This is a generic diagram of the typical peripherals on an MSPM0Gxx device and their respective bus locations. Not all devices have all peripheral options shown here. To determine the peripherals which are available on a given device, see the device-specific data sheet.

The GPIO and ADC peripherals (the orange components in the bus diagram) have special capabilities to enable both fast register access from the CPU and operation in low power operating modes.

- GPIO peripherals interface to the system through two mechanisms: the PD1 peripheral bus, and the Arm® Cortex®-M0+ single-cycle high-speed IO bus.
  - Accesses from the CPU to any GPIO registers are transacted through the single-cycle IO bus for best performance, enabling fast toggling of IO under software control.
  - The GPIO DOUT registers (data out) are also available on the PD1 peripheral bus, primarily so that the DMA can be used to load values to the GPIO DOUT registers.
  - While the bus interfaces to the GPIO peripherals are in the PD1 power domain (for best read/write performance), the GPIO logic itself is in the PD0 power domain so that it is available in all operating modes in which the core regulator is active.
- ADC peripherals interface to the system through the PD1 peripheral bus but contain functional logic in the PD0 power domain.
  - ADC peripheral register accesses are processed through the PD1 peripheral bus (for best read/write performance)
  - The the ADC conversion logic is in the PD0 power domain to enable running timer-triggered ADC conversions without CPU interaction in certain low-power modes when PD1 is disabled.

## 1.3 Platform Memory Map

All MSPM0Gxx devices share a common platform memory map. Peripherals are assigned a fixed address space and have the same address space on all devices within the family. The memory map is compliant with the standard Arm Cortex-M memory regions.

**Table 1-1. Top Level Memory Map**

| Memory Region | Start Address | End Address | Description |
|---|---|---|---|
| Code | 0x0000.0000 | 0x1FFF.FFFF | Flash memory and ROM |
| SRAM | 0x2000.0000 | 0x3FFF.FFFF | SRAM |
| Peripheral | 0x4000.0000 | 0x5FFF.FFFF | Global peripheral memory-mapped registers and global nonexecutable data memory |
| Subsystem | 0x6000.0000 | 0x7FFF.FFFF | Local CPU subsystem memory-mapped registers |
| System PPB | 0xE000.0000 | 0xE00F.FFFF | Arm private peripheral bus |

### *1.3.1 Code Region*

The code region contains the flash memory used to store executable code and data. Accesses to the flash memory from the CPU through the code region are processed through the AHB bus matrix to the flash read interface directly. The flash memory is aliased to two address spaces in the code region: one returning ECC-corrected data, the other providing uncorrected (raw) data. See Section 6.2.3.1 for the detailed flash memory map.

The code region also contains the read-only memory (ROM) used for the TI device boot code and the bootstrap loader. The ROM is only available during the initial device boot process.

### *1.3.2 SRAM Region*

The SRAM region contains the system memory (SRAM). The SRAM supports zero wait state access at the maximum MCLK frequency (80MHz). Accesses to the SRAM from the CPU are processed through the AHB bus matrix to the SRAM interface directly. See the device-specific data sheet for the amount of SRAM present on a given device.



Copyright © 2026 Texas Instruments Incorporated

Certain devices optionally support parity or parity and ECC checking of the SRAM. Refer to the device-specific data sheet to determine if a device supports ECC or parity checked SRAM. For information on how parity and ECC errors are handled by the device, see Section 2.4.10.

### Parity Checking

In the case of parity checking (if available), 1 parity bit is provided per 8 data bits. Parity checking is capable of detecting a single bit error in the corresponding 8 bits of data (SED). Writing data to a parity checked address updates the corresponding parity bits based on the new data. Reading data from a parity checked address checks the read data against the corresponding parity bits. Upon a read, if the data does not match the corresponding parity bits, a parity error is generated. For information on how parity errors are handled by the device, see Section 2.4.10

### ECC Checking

In the case of ECC checking (if available), 8 ECC bits are provided per 64 data bits. ECC is capable of correcting single bit errors (SEC) and detecting dual bit errors (DED) in the corresponding 64 data bits. Writing data to an ECC checked address updates the corresponding ECC code based on the new data. Reading data from an ECC checked address checks the read data against the corresponding ECC code. If a single bit error is found, it is corrected automatically and a correctable ECC error is generated. If a dual bit error is found, an uncorrectable ECC error is generated.

### Aliased Subregions

The physical SRAM on a device is aliased into multiple address subregions in the overall SRAM region, as shown in the following table. The default, parity checked, and unchecked address subregions all map to the same physical SRAM memory. The difference between each aliased subregion is in the type of integrity checks which are applied to the access. For example, writing data to address 0x2000.0000 (the default subregion) will cause the same data to appear at address 0x2020.0000 (the unchecked subregion).

The default subregion (0x2000.0000) is available on all MSPM0 devices, and when used, provides the highest level of integrity checking available on the device. The parity checked subregion (0x2010.0000) is available on devices which support ECC or parity checking, and accesses are always processed with parity checking. The unchecked subregion (0x2020.0000) is available on all devices, and when using this subregion, no integrity checks are performed. The parity/ECC code subregion (0x2030.0000) is available on devices with ECC or parity checking, and it returns the parity or ECC code which corresponds to the address being read.

**Note**

To improve robustness, there is no mechanism provided to disable parity checking of the parity-checked SRAM address space. To operate without parity checking, link the application against the unchecked SRAM address space.

**Table 1-2. SRAM Region Memory Map**

| Subregion | Start | End (Maximum) | Description |
|---|---|---|---|
| Default | 0x2000.0000 | 0x200F.FFFF | The highest available integrity check on the device is always applied to accesses in this subregion:<br>• If the device supports ECC, this subregion is ECC checked.<br>• If the device only supports parity (no ECC), this subregion is parity checked and accesses are equivalent to accesses to the parity checked subregion.<br>• If the device does not support ECC or parity, no checks are applied to accesses in this subregion and the region is equivalent to the unchecked subregion. |
| Parity checked | 0x2010.0000 | 0x201F.FFFF | If the device supports parity, accesses to this subregion are parity checked. |
| Unchecked | 0x2020.0000 | 0x202F.FFFF | No ECC or parity checks are applied to accesses in this subregion. |



Copyright © 2026 Texas Instruments Incorporated

**Table 1-2. SRAM Region Memory Map (continued)**

| Subregion | Start | End (Maximum) | Description |
|---|---|---|---|
| Parity/ECC code | 0x2030.0000 | 0x203F.FFFF | If the device supports parity or ECC, the parity or ECC codes may be directly accessed through this subregion:<br>• If the device supports ECC, access to any address within a 64-bit boundary returns the 8 bits corresponding to the ECC code (if the application is linked against the default region) or the parity bits (one per byte) if the application is linked against the parity checked region.<br>• If the device supports parity only, access to any address within a 32-bit boundary returns the 4 parity bits (one per byte) if the application is linked against the default region or parity checked region.<br>• If the device does not support ECC or parity, access to this region always returns zeros. |

**Note**

In some devices, the SRAM is organized into two or more banks. Not all the banks may have have ECC and/or parity. Typically, the first bank will include ECC and/or parity while additional banks may or may not include ECC/parity. In such cases, the size of each subregion will depend on the device-specific SRAM configuration. For example, if a device features 128KB of SRAM with the first 64KB having ECC/parity protection and the second 64KB without ECC/parity, then the size of ECC protected subregion, the parity checked subregion and the codes region is each 64KB while the unchecked region will be 128KB in size. In all cases, the unchecked subregion is contiguous and of the full size, allowing applications to link code and data freely in this subregion.

**Note**

When ECC checking is used on devices which support ECC, writes to the SRAM require two cycles to complete. Read accesses only require a single cycle and do not incur any additional performance penalty.

On devices supporting parity or ECC, it is possible for application software to partition the usage of the physical SRAM into arbitrary zones which are intended to be ECC checked, parity checked, or unchecked. For example, if a device has 32KB of total SRAM memory, and it supports ECC and parity checking, it is possible to configure the application software to link against two subregions, one being ECC checked and the other being parity checked.

**Note**

It is not recommended to mix and match ECC checked, parity checked, and unchecked accesses to the same memory locations, as this can result in unintended parity/ECC errors. For example, if an SRAM memory location is written to through the parity checked subregion on a device which supports ECC, and then later that location is read through the default (ECC checked) subregion, an ECC error may be generated because the data was written with parity stored to the ECC/parity code memory, as opposed to the correct ECC code being stored.

**Note**

The SRAM contents may be random at power-up or when existing SHUTDOWN mode. A read of an SRAM location through an ECC/parity checked region which has not yet been written to may result in a ECC/parity error being detected, causing a hard fault in the processor. Ensure that all SRAM locations are initialized first before being read.



Copyright © 2026 Texas Instruments Incorporated

If the application is using only the unchecked memory region, then the Parity/ECC code region can be used for additional application SRAM memory.

### 1.3.3 Peripheral Region

The peripheral region contains the memory-mapped peripherals on the three peripheral buses. The flash memory is also aliased in the peripheral region.

**Table 1-3. Peripheral Region Memory Map**

| Type | Start | End | Description |
|---|---|---|---|
| Peripherals | 0x4000.0000 | 0x40FF.FFFF | Memory-mapped registers of peripherals on the peripheral buses |
| Aliased flash memory | 0x4100.0000 | 0x41FF.FFFF | See Section 6.2.3.1 |

### 1.3.4 Subsystem Region

The subsystem region contains memory-mapped registers which are specific to the CPU subsystem and do not need to be accessed globally. See the CPU subsystem chapter for the memory-mapped registers in the subsystem region.

### 1.3.5 System PPB Region

The system private peripheral bus (PPB) region contains memory-mapped registers on the Arm private peripheral bus. These registers are tightly coupled to the CPU and are the interface for peripherals such as the memory protection unit (MPU), SysTick timer, and CPU power management and reset functions.

## 1.4 Boot Configuration

After a BOOTRST, the device executes the start-up boot routines to configure the device for operation before starting the main application. Boot routines are executed from read-only memory (ROM) before the main application is started. There are two boot routines: the boot configuration routine (BCR), and the bootstrap loader (BSL). The boot configuration routine sets up the device security policies, configures the device for operation, and optionally starts the BSL if it presents. The BSL, if started by the BCR, can be used to program or verify the device memory (flash and SRAM) through the use of a standard serial interface (UART or I2C).

After the start-up routines have successfully completed execution, the CPU is reset and the application is started by unconditionally fetching the stack pointer (SP) and reset vector from 0x0000.0000 and 0x0000.0004 of the flash memory. To enable secure boot, this single point of entry into the application code is enforced by the boot sequence. It is not possible to boot into a different memory location.

### 1.4.1 Configuration Memory (NONMAIN)

The NONMAIN is a dedicated region of flash memory which stores the configuration data used by the BCR and BSL to boot the device. The region is not used for any other purpose. The BCR and BSL both have configuration policies which can be left at their default values (as is typical during development and evaluation), or modified for specific purposes (as is typical during production programming) by altering the values programmed into the NONMAIN flash region.

The BCR and BSLconfiguration data structures are both contained within a single flash sector in the NONMAIN flash memory region. To change any parameter in the boot configuration, it is necessary to erase the entire NONMAIN sector and re-program both the BCR and BSLconfiguration structures with the desired settings.

The configuration data in the NONMAIN flash region is not affected by a mass erase command, but it is erased and re-programmed to factory defaults by a factory reset command sent to the BCR via the debug sub system mailbox (DSSM) over SWD.

The NONMAIN flash is also erased by a factory reset command sent to the BSL using the UART or I2C BSL interface. However, unlike the DSSM factory reset, the BSL factory reset does not program TI factory defaults to the NONMAIN memory following the erase. As such, it is the responsibility of the host which is connected to



Copyright © 2026 Texas Instruments Incorporated

the MSPM0 target (via the BSL interface) to re-program the NONMAIN memory with a valid configuration before terminating the BSL session.

**Note**

If a factory reset command is executed through the BSL, and a valid NONMAIN configuration is not programmed back into the device before the BSL session is terminated, the device will assume a maximally restrictive state upon the next reset cycle, and it will not be possible to access the device via SWD or the BSL. Always ensure that a valid NONMAIN configuration is programmed back when using the BSL factory reset command.

The address ranges for the NONMAIN data structures are given in the *NONMAIN Registers* section. A detailed breakdown of the NONMAIN region is provided at the end of this section.

#### 1.4.1.1 CRC-Backed Configuration Data

The BCR configuration data and BSL configuration data structures in the NONMAIN memory each include a CRC value corresponding to the CRC digest of the respective structure. During the device boot process, the BCR will compute the CRC digest of the data structures and compare it with the stored CRC values before the data contained within the structures is trusted for use.

CRC caluculation:

- Polynomial should be as per the chosen standard (CRC32-ISO3309 or CRC16-CCITT). Please refer to related datasheet.
- Input reflected
- Output reflected
- Initial value should be 0xFFFFFFFF
- Final XOR value should be 0x0

**BCR Configuration CRC Fail Handling**

In the event that the BCR configuration data (which contains the SWD policies, BSL enable/disable policy, and flash memory protection and integrity check policies) fails its CRC check during boot, a catastrophic boot error results and the following limitations are imposed:

- The error cause will be logged in the CFG-AP as a boot diagnostic
- The BSL will not be invoked, even if it was configured to be enabled
- The user application is not started
- No application debug access is enabled
- A pending SWD factory reset command, if enabled or enabled with password, is honored
- A pending TI failure analysis flow entry, if enabled, is honored
- The boot process will re-attempt up to 3 times
  - If the 2nd or 3rd attempt pass, the device boots normally
  - If the 3rd attempt does not pass, no further boot attempts are made until the next BOR or POR

The benefit of the this CRC check is that any bit flips in configuration data, such as the static write protection configuration (which is a pillar of secure boot), may be detected with high confidence during the boot process. The fail handling procedure explicitly prevents the BSL and user application from running, and the only supported options (SWD factory reset and TI FA) are protected by 16-bit pattern-match fields.

**BSL Configuration CRC Fail Handling**

If the BSL configuration data (which contains the BSL password and BSL policies) fails the CRC check during BSL invocation, a catastrophic boot error results and the following limitations are imposed:

- The error cause is logged in the CFG-AP as a boot diagnostic
- The BSL is not invoked, even if it was configured to be enabled
- The user application is not started
- No application debug access is enabled



Copyright © 2026 Texas Instruments Incorporated

- The boot process re-attempts up to 3 times
  - If the 2nd or 3rd attempt pass, the device boots normally
  - If the 3rd attempt does not pass, no further boot attempts are made until the next BOR or POR

The benefit of this CRC check s that any bit flips in the BSL configuration data may be detected with high confidence during the invoke process. The failure handling procedure prevents the BSL from starting with invalid data which could lead to a loss of security.

**TI Factory Trim Data CRC Fail Handling**

In addition to the user-specified configuration data, if the TI factory trim fails its CRC check during boot, a catastrophic boot error will also result with the following limitations:

- The error cause will be logged in the CFG-AP as a boot diagnostic
- The BSL will not be invoked, even if it was configured to be enabled
- The user application is not started
- No application debug access is enabled
- A pending TI failure analysis flow entry, if enabled, is honored
- The boot process will re-attempt up to 3 times
  - If the 2nd or 3rd attempt pass, the device boots normally
  - If the 3rd attempt does not pass, no further boot attempts are made until the next BOR or POR

#### 1.4.1.2 16-bit Pattern Match for Critical Fields

Critical policies in the BCR configuration memory, such as the SWD security policies, are implemented as 16-bit pattern-match fields in the NONMAIN memory, with the following characteristics:

- An exact pattern match is required to enable lower security states
- Any value in the 16-bit field not matching the exact defined patterns results in a maximally secure state for the respective parameter

This behavior prevents single bit flips from causing the device to enter a lower security state than that which was originally specified.

### *1.4.2 Boot Configuration Routine (BCR)*

The boot configuration routine is the first firmware to run on the device after a BOOTRST. The BCR manages the following at boot time:

- Configuring the debug interface security policy
- Optionally executing a mass erase
- Optionally executing a factory reset
- Configuring the flash memory static write protection policy
- Optionally verifying the integrity of some or all of the application firmware (with a 32-bit CRC)
- Optionally starting the bootstrap loader (BSL)

#### 1.4.2.1 Serial Wire Debug Related Policies

The serial wire debug related policies configure the functionality which is available through the device's physical debug interface (SWD). By default, MSPM0 devices come from TI in an unrestricted state. This state allows for easy production programming, evaluation, and development. However, this unrestricted state is not recommended for mass production, as it leaves a large attack surface present. To accommodate a variety of needs while keeping the configuration process simple, MSPM0 devices support three generic security levels: no restrictions (Level 0), custom restrictions (Level 1), and fully restricted (Level 2). Table 1-4 shows the three generic security levels, from least restrictive to most restrictive.

There are 4 main uses of the SWD interface for which protection needs to be considered:

- Application debug access, which includes:
  - Full access to the processor, memory map, and peripherals through the AHB-AP
  - Access to the device EnergyTrace+ state information through the ET-AP
  - Access to the device power state controls for debug through the PWR-AP



Copyright © 2026 Texas Instruments Incorporated

- Mass erase access, which includes:
  - Ability to send a command through SWD to erase the MAIN memory region while leaving the NONMAIN device configuration memory intact
- Factory reset access, which includes:
  - Ability to send a command through SWD to erase the MAIN memory region and reset the NONMAIN device configuration memory to TI factory defaults (Level 0)
- TI failure analysis access, which includes:
  - Ability for TI to initiate a failure analysis return flow through SWD (note that the TI FA flow always forces a factory reset before FA access is given to TI; this ensures that TI does not have any mechanism to read proprietary customer information stored in the device flash memory when a failure analysis flow is initiated)

**Table 1-4. Generic Security Levels**

| Level | Scenario | SW-DP Policy | App Debug Policy | Mass Erase Policy | Factory Reset Policy | TI FA Policy |
|---|---|---|---|---|---|---|
| **0** | No restrictions | EN | EN | ENEN, | EN | EN |
| **1** | Custom restrictions | EN | EN, EN with PW, DIS | EN, EN with PWDIS | EN, EN with PW, DIS | EN, DIS |
| **2** | Fully restricted | DIS | Don't care (access not possible with SW-DP disabled)(1) | | | |

(1) When the SW-DP policy is **SW-DP disabled**, the mass erase and factory reset policies are a don't care from the point of view of the SWD interface. However, if the bootstrap loader (BSL) is enabled, the mass erase and factory reset policies do impact what functionality is available through the BSL. See the BSL security section for details on securing the BSL.

#### *1.4.2.1.1 SWD Security Level 0*

SWD security level 0 is the least restrictive SWD security state. This is the default state of a new device from TI, and it is also the state of a device following a successful factory reset. There are no restrictions on application debug access, mass erase, factory reset, for failure analysis in this state.

**When to Use This State**

Level 0 is well suited for prototyping and development, as it allows programming of the device memory and debug of the processor and peripherals.

**When to Not Use this State**

Level 0 should not be used in mass production. An attacker would have full freedom to read the contents of the device memory, manipulate the execution of the device, and possibly change the flash memory contents (depending on the flash memory write protection scheme).

#### *1.4.2.1.2 SWD Security Level 1*

SWD security level 1 allows for a customized security configuration. The physical debug port (SW-DP) is left enabled, and each function (application debug, mass erase command, factory reset command, and TI failure analysis) may be individually enabled, disabled, or (in some cases) enabled through password authentication, providing considerable flexibility to tailor the device behavior to specific use-cases.

**When to Use This State**

Level 1 is well suited for restricted prototyping/development scenarios and for mass production scenarios where the desire is to retain certain SWD functions (such as factory reset and TI failure analysis) while disabling other functions (such as application debug). Common examples of Level 1 customized configurations are given in Table 1-5.



Copyright © 2026 Texas Instruments Incorporated

**Table 1-5. Examples of Level 1 Configurations**

| Level 1 Scenario | Configuration | | | |
|---|---|---|---|---|
| | App Debug | Mass Erase | Factory Reset | TI FA |
| This scenario restricts debug access with a user-specified password, but it leaves the factory reset and TI failure analysis available. This configuration allows field debug (with password), and it also allows the device to be brought back to the default "Level 0" state through factory reset. | EN with PW | DIS | EN | EN |
| This scenario does not allow debug. It does allow factory reset, but only with a user-specified password. This provides a way to open up a device in the field by clearing the MAIN memory contents and bringing the device back to a "Level 0" state if the password is known. Importantly, even if the factory reset password were compromised, it would not be possible for an attacker to read proprietary information in the MAIN flash memory. | DIS | DIS | EN with PW | EN |
| This scenario does not allow debug and it does not allow TI failure analysis. This prevents TI from performing a factory reset and further FA activities on the device, unless the user executes a factory reset with their user-specified password before returning the devices to TI for FA. | DIS | DIS | EN with PW | DIS |

**Note**

Level 1 is the recommended configuration for most standard production use-cases. For applications which do not require secure boot, TI recommends using Level 1 in production with factory reset left enabled (with password) and TI failure analysis left enabled. In such a configuration, the device may be recovered to a less restrictive state after provisioning either by the user (with password) or by TI (through the failure analysis return flow). In use-cases requiring maximum secure boot assurance, a more restrictive Level 1 or Level 2 may be used for production, with the trade-off that devices may not be recoverable to a less restrictive state once provisioned.

**When to Not Use this State**

Level 1 should not be used during prototyping if complete access to the device is desired; in such a case, Level 0 should be used instead.

Level 1 should also not be used in a mass production scenario where a maximally restrictive state is desired and no SWD functions are to be enabled; in such a case, Level 2 should be used instead as it directly disables the complete SWD physical interface and minimizes the possibility of misconfiguration.

**Note**

If a device is configured with application debug and factory reset disabled, the only way for a user to restore debug access to the device is if the user application code provides a mechanism to change the NONMAIN configuration to a less restrictive state. If the NONMAIN is locked through static write protection then the state is not reversible and there is no way for a user to re-gain debug access.

#### *1.4.2.1.3 SWD Security Level 2*

SWD security level 2 configures the device in a maximally restrictive state. The physical debug port (SW-DP) is completely disabled, and all of the SWD-accessible functions (application debug, mass erase, factory reset, and TI failure analysis) are not accessible through SWD, regardless of their individual configuration.

When level 2 is selected (SW-DP disabled), the application debug configuration and TI failure analysis configuration fields are don't care fields which do not impact the device configuration.

If the BSL is disabled, then the mass erase and factory reset configuration fields are also don't care fields. However, if the BSL is enabled, then the mass erase and factory reset configuration fields are still used by the BSL to authorize mass erase or factory reset commands originating from the BSL interface.

**When to Use This State**



Copyright © 2026 Texas Instruments Incorporated

Use Level 2 only for mass production when no further access to any SWD functions is required and a maximally secure state is desired for the device.

**When to Not Use this State**

Do not use Level 2 in the following cases:

- Future application debug or reprogramming through SWD is required
- So that TI can perform failure analysis on the device
- To remove proprietary information from the flash memory by sending a mass erase or factory reset command through SWD

**Note**

After a device is configured for level 2 (SW-DP disabled), further access to the device through SWD **is not possible**. The only way to bring a device back to a level 0 or level 1 state with SWD access restored is if the BSL and factory reset are both enabled (allowing a BSL factory reset command to be sent), or a mechanism in the user application code is included which can change the NONMAIN configuration to a less restrictive state. In either scenario, if the NONMAIN is locked through static write protection then the level 2 state is not reversible and there is no way to re-gain SWD access.

#### 1.4.2.2 SWD Mass Erase andFactory Reset Commands

The BCR provides mass erase and factory reset functionality through commands sent to the device over SWD from a debug probe using the debug subsystem mailbox (DSSM). These commands are not available in SWD security level 2, but they are optionally available in security level 0 and 1. When the device is not configured for SWD security level 2, the mass erase andfactory reset commandscan be individually configured to be enabled, enabled with a unique 128-bit password, or disabled. By default, both commands are enabled.

The SWD mass erase andfactory reset DSSM commandssuperseded any static write protection policies. For example, if SWD factory reset is configured to be enabled or enabledwith password, the BCR configuration data can be reset even if it is statically write protected.

**SWD Mass Erase**

A SWD mass erase is an erase of the MAIN flash regions only, which typically includes the user application. The BCR and BSL policies stored in the NONMAIN flash region are not affected by a mass erase. A mass erase is useful for erasing all application code and data while leaving the device configuration itself intact.

To set the mass erase command mode and password, configure the BOOTCFG3.MASSERASECMDACCESS field and the PWDMASSERASE password fields in the NONMAIN memory.

**SWD Factory Reset**

A SWD factory reset is an erase of the MAIN flash regions followed by a reset of the NONMAIN flash region to default values. Such an erase is useful for completely resetting the BCR and BSL device boot policies while also erasing the application code and data.

To set the factory reset command modeand password, configure the BOOTCFG3.FACTORYRESETCMDACCESS fieldand the PWDFACTORYRESET password fieldsin the NONMAIN memory.

#### 1.4.2.3 Flash Memory Protection and Integrity Related Policies

The flash memory protection and integrity policies specify which sectors of flash memory are locked from modification, as well as which sectors are to be checked for integrity during the boot process before the user application is started.



Copyright © 2026 Texas Instruments Incorporated

#### 1.4.2.3.1 Locking the Application (MAIN) Flash Memory

MSPM0 MCUs implement a static write protection scheme to lock out user defined sectors in the MAIN flash region from any program or erase operations at runtime. The desired static write protection scheme is configured as a part of the boot security policies in the NONMAIN flash region.

**Purpose**

Static write protection enables placement of a fixed, user-defined, application in the flash memory that has the following characteristics:

- Once programmed and locked, the application is not modifiable by the application code or ROM bootloader
- If placed at the beginning of the flash memory, the application is the first code that executes when the ROM boot configuration routine transfers execution to the user application

MSPM0 static write protection supports both characteristics, which must be satisfied to implement a secure boot image manager.

**Capabilities**

Any sector that is configured in the NONMAIN to be write-locked is functionally immutable when the boot configuration routine transfers execution to either the bootstrap loader or the user application code in MAIN flash. Any attempt to program or erase a statically protected sector by the application code or the bootstrap loader results in a hardware flash operation error, and the sector is not modified.

While static write protection prevents any modification by application code or the bootloader, a mass erase or factory reset command sent through the SWD interface is honored. If this behavior is not desired, the mass erase or factory reset SWD commands can be protected with unique passwords or disabled(see the SWD policies). To completely remove any means of modifying statically write protected MAIN flash sectors, the mass erase and factory reset commands (or the SW-DP) must be disabled, and the NONMAIN boot configuration memory must also be statically write protected to prevent application code from changing the underling write protection scheme by modifying the contents NONMAIN region. This is discussed in the following section.

#### 1.4.2.3.2 Locking the Configuration (NONMAIN) Flash Memory

MSPM0 MCUs implement a static write protection scheme to lock out the NONMAIN flash region from any program/erase operations at runtime. The write protection scheme is configured as a part of the boot security policies in the NONMAIN flash region.

**Purpose**

By default, the NONMAIN configuration memory (which contains the user-specified boot security policies and bootstrap loader policies) is not write protected. This enables the NONMAIN to be erased by the user during provisioning and re-programmed with the user-specified policies to use in mass production.

In many cases, it is desirable for the configuration memory to be locked after it has been provisioned. Locking the configuration memory has the benefit of preventing any unauthorized modification of the security policies, bootstrap loader policies, and static write protection policies by either the bootstrap loader or the application code. In most applications, devices in mass production do not require modification of the configuration memory, even when the device firmware is updated.

**Capabilities**

When configured to be protected, the entire NONMAIN region will be write-locked and will be functionally immutable when the boot configuration routine transfers execution to either the bootstrap loader or the user application code in MAIN flash. Any attempt to program or erase the NONMAIN by the application code or the bootstrap loader will result in a hardware flash operation error, and the sector will not be modified.

While static write protection prevents any modification by application code or the boot loader, a factory reset command sent through the SWD interface would still be honored. If this behavior is not desired, the factory reset SWD command may be protected with a unique password or disabled altogether (see the SWD policies). To



Copyright © 2026 Texas Instruments Incorporated

completely remove any means of modifying the NONMAIN configuration memory, the factory reset command and TI FA (or the SW-DP) must be disabled.

**Note**

When the NONMAIN is statically write protected, and the factory reset command and TI FA (or the SW-DP) are disabled, the NONMAIN is equivalent to immutable read-only memory, and it is no longer possible to change the device configuration by any means. Further, if any MAIN memory region sectors are configured with static protection, these sectors also cannot be modified by any means and may be considered as immutable.

#### *1.4.2.3.3 Static Write Protection NONMAIN Fields*

Write protection may be enabled on a per-sector basis for the first 32 sectors of the MAIN flash memory. For the remaining sectors of flash memory, if present, write protection may be enabled per 8 sectors. To set a static write protection policy, configure the FLASHSWP0 and FLASHSWP1 fields in the NONMAIN memory.

**Note**

Mass erase and factory reset commands Factory reset commandsent to the BCR via the debug sub system mailbox (DSSM) will override the specified static write protection policy. If this behavior is not desired, configure the mass erase and factory reset commandsto be enabled with password or disabled. Note that mass erase and factory reset commands sent to the BSL will respect the specified static write protection policy (the BSL has the same permissions as application code).

### 1.4.2.4 Application CRC Verification

The BCR supports executing a complete CRC32 integrity check of the application code and data contained in the MAIN flash regions during the boot process before starting the user application. This is useful to ensure the integrity of some or all of the application code and data before executing it.

To enable the CRC32 integrity check at boot, the following information must be programmed into the BCR configuration in NONMAIN flash memory:

- 32-bit starting address of the CRC check (APPCRCSTART.ADDRESS field)
- The length of the application (specified in bytes) for which the CRC check is applied (APPCRCLEN.LENGTH field)
- The precalculated 32-bit CRC value to test against (APPCRC.DIGEST field)
- The enabled key value for the CRC check (BOOTCFG3.APPCRCMODE field)

In the event that an application CRC check fails at boot, the application in MAIN flash is not started. If the boot strap loader is enabled, it is entered. If the BSL is not enabled, then the boot fails.

### 1.4.2.5 Fast Boot

The execution time of the BCR can be reduced by enabling fast boot mode. Fast boot mode, when enabled, speeds up the boot process using the following methods:

- Limiting the BSL entry methods. When fast boot mode is enabled, only the SYSCTL register invoke method and the DSSM invoke method can be used to enter the bootloader. The other BSL invoke conditions are not tested (for example, pin based invoke).
- Bypassing the application CRC check (even if the application CRC check is enabled).

To enable fast boot mode, set the fast boot mode to enabled in the BOOTCFG2.FASTBOOTMODE field in the NONMAIN flash memory.

### 1.4.2.6 Bootstrap Loader (BSL) Enable/Disable Policy

The bootstrap loader (BSL) provides a means to program and verify the device memory through a standard serial interface (UART or I2C), as opposed to the serial wire debug interface. The BSL has its own configuration policy, but the BCR determines if the BSL is enabled to be invoked, or if it is to be disabled (noninvokable).



Copyright © 2026 Texas Instruments Incorporated

Since the BSL presents an additional attack surface, if it is not used in an application it may be disabled in the user-specified boot security policies. If the BSL is used in an application, then the BSL security settings (including the BSL access password) are managed in the BSL configuration policy.

#### *1.4.2.6.1 BSL Enable*

The bootstrap loader (BSL) can be enabled or disabled by setting the bootloader mode to enabled or disabled in the BOOTCFG2.BSLMODE field in the NONMAIN flash memory. When the BSL is disabled, the bootloader cannot be entered through any invocation mechanism.

### *1.4.3 Bootstrap Loader (BSL)*

The bootstrap loader (BSL) provides a method to program and/or verify the device memory through a standard UART or I2C serial interface. Key features of the BSL which are accessible through the serial interface include:

- Programming and erase of flash memory
- Ability to return a firmware version number through a pointer to the main flash
- Ability to specify a hardware invoke GPIO
- Ability to enable code/data read-out (disabled by default)
- Ability to return a 32-bit CRC of a code/data region (1KB minimum region size) to verify programming
- Access is always protected with a 256-bit password
- Configurable security alert handling for resisting brute force attacks
- Support for MAIN flash plug-ins to enable additional interfaces beyond UART and I2C

For a complete description of the BSL features, see the BSL user's guide.

The BSL can be completely disabled if desired by properly configuring the BSL mode in the BCR configuration in the NONMAIN flash. See the BSL Enable section for details on enabling or disabling the BSL.

#### 1.4.3.1 GPIO Invoke

The bootloader supports hardware invoking after a BOOTRST through the use of a GPIO. The BSL configuration in the NONMAIN flash memory contains the pad, pin, and polarity definition for the GPIO invocation. Devices come configured from TI for a specific GPIO and polarity, but software can change this default by modifying the GPIO pin configuration in the BSL configuration in NONMAIN flash memory.

To specify the polarity of the BSL_invoke pin, configure the BSLCONFIG0.BSLIVK_LVL field in the NONMAIN memory.

To specify the device pin to be used for BSL_invoke, configure the following fields in the NONMAIN memory:

- Store the IOMUX PINCMx index into the BSLCONFIG0.BSLIVK_PAD_NUM field
- Store the GPIO port (A or B) into the BSLCONFIG0.BSLIVK_GPIOPORT field
- Store the GPIO pin (0-31) into the BSLCONFIG0.BSLIVK_GPIOPIN field

See the device specific data sheet to determine the default BSL invoke GPIO.

#### 1.4.3.2 Bootstrap Loader (BSL) Security Policies

The BSL security policies are interpreted by the boot loader when it is invoked, and include the following parameters:

- BSL access password, described in Section 1.4.3.2.1
- BSL read-out policy, described in Section 1.4.3.2.2
- BSL security alert policy (tamper detection), described in Section 1.4.3.2.3

##### *1.4.3.2.1 BSL Access Password*

Access to the BSL is always protected by a 256-bit user-specified password. There is no option to disable the password. The password must be provided to the BSL after invocation for access to most BSL functions to be granted. When the password is not provided, the only BSL commands allowed are *Get Identity* and *Start Application*.



Copyright © 2026 Texas Instruments Incorporated

If a wrong password is provided to the BSL, the BSL halts for 2 seconds, after which an additional attempt can be made to send the correct password. After three failed password attempts, the security alert function is activated (see Section 1.4.3.2.3).

#### *1.4.3.2.2 BSL Read-out Policy*

The BSL optionally supports read-out of the device memory for debug and/or diagnostic purposes (after access to the BSL has been granted with a correct password match). By default, this capability is disabled for security to prevent extraction of sensitive code and/or data from the device. When the BSL read-out policy is disabled, the only information which may be provided to a host through the BSL interface is a CRC32 digest of a memory segment with a minimum segment length of 1 kilobyte. If direct read-out of the device memory is desired, it may be enabled in the BSL configuration.

#### *1.4.3.2.3 BSL Security Alert Policy*

The BSL provides an alert mechanism for taking action when tampering is suspected. Specifically, if an incorrect password is passed to the BSL 3 times during one BSL session, the security alert is activated and the BSL may respond in one of three different ways based on the specified security alert policy:

1. Issue a factory reset (erasing the MAIN flash and resetting the NONMAIN flash regions)
2. Disable the BSL (leaves the MAIN flash intact but re-configures the NONMAIN to block BSL access)
3. Ignore (do not modify the configuration and allow password attempts to continue)

**Note**

Options 1 and 2 require that the NONMAIN flash region not be statically write protected (see Section 1.4.2.3.2).

When option 1 is selected, any MAIN memory region which is configured to be statically write protected (see Section 1.4.2.3.1) will not be erased during the factory reset.

### 1.4.3.3 Application Version

The BSL supports returning an application version number through the BSL serial interface. This allows the BSL host to interrogate the firmware version without being able to read the firmware. The version field is 32 bits in length. To link the application version command to a version number programmed in the main flash, program the address of the version number in the BSLAPPVER.ADDRESS field in NONMAIN flash memory. The version data is only returned if the address specified in BSLAPPVER.ADDRESS corresponds to a valid flash memory address.

### 1.4.3.4 BSL Triggered Mass Erase and Factory Reset

It is possible to send a mass erase or factory reset command to the BSL. The commands work in a similar way as the SWD mass erase and factory reset commands, with several key exceptions.

#### BSL Mass Erase

A mass erase command sent to the BSL will erase the MAIN flash memory. Any MAIN flash memory sectors which are configured to be statically write protected (via the FLASHSWP0 and FLASHSWP1 fields in the NONMAIN configuration memory) will not be erased. The NONMAIN device configuration memory is never erased by a mass erase.

#### BSL Factory Reset

A factory reset command sent to the BSL will first perform a BSL mass erase to erase the main flash memory (excluding any sectors which are statically write protected). Then, it will additionally erase the NONMAIN device configuration memory.

A BSL factory reset command is only accepted if the following are true:



Copyright © 2026 Texas Instruments Incorporated

1. The NONMAIN memory itself is not currently configured to be statically write protected (BOOTCFG4.NONMAINSWP field in the NONMAIN is set to not protected)
2. The factory reset command is not configured to be disabled (BOOTCFG3.FACTORYRESETCMDACCESS field in the NONMAIN is not set to disabled)

The BSL host must program valid configuration data back into the NONMAIN (via BSL commands) before terminating the BSL session, or the device may enter an unrecoverable state.

**Note**

If the NONMAIN is left unprogrammed after a BSL factory reset, the device will assume a maximally restrictive state on the next reset cycle, any application code in MAIN flash will not be started, and it will not be possible to access the device via any means. To prevent lockout, always ensure that valid configuration data is programmed into NONMAIN following a BSL factory reset.

### *1.4.4 NONMAIN Layout Types*

The below table shows which NONMAIN layout type to use with which devices. The MSPM0-SDK includes a configurator tool to help the user configure NONMAIN contents.

**Table 1-6. NONMAIN Layout Types**

| NONMAIN Layout Type | Overview | Supported Devices |
|---|---|---|
| Type A | • No CSC support<br>• Passwords stored in plaintext format<br>• App integrity check has CRC32 support only | • MSPM0G110x<br>• MSPM0G150x<br>• MSPM0G310x<br>• MSPM0G350x |
| Type E | • Includes CSC support<br>• Passwords stored in SHA256 hashed format<br>• App integrity check has CRC32 or SHA256 hash option<br>• UART default baudrate configuration<br>• Disable NRST in BSL | • MSPM0Gx22x<br>• MSPM0Gx21x<br>• MSPM0G511x<br>• MSPM0G5187 |
| Type F | • Includes CSC support<br>• Passwords stored in SHA256 hashed format<br>• App integrity check has CRC32 or SHA256 hash option<br>• UART default baudrate configuration | • MSPM0G151x<br>• MSPM0G351x<br>• MSPM0G352x |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.5 NONMAIN_TYPEA Registers

Table 1-7 lists the memory-mapped registers for the NONMAIN_TYPEA registers. All register offset addresses not listed in Table 1-7 should be considered as reserved locations and the register contents should not be modified.

**Table 1-7. NONMAIN_TYPEA Registers**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 41C00000h | BCRCONFIGID | Configuration ID of BCR Structure | | Go |
| 41C00004h | BOOTCFG0 | Serial wire debug (SWD) lock policy. | | Go |
| 41C00008h | BOOTCFG1 | TI Failure Analysis (FA) mode and BSL invoke pin policies. | | Go |
| 41C0000Ch + formula | PWDDEBUGLOCK[y] | Debug Lock Password.<br>This password is used when the BOOTCFG0.DEBUGACCESS field is configured as Enable with password. | | Go |
| 41C00010h | BOOTCFG2 | Fast boot mode policy and BSL mode policy | | Go |
| 41C00020h | BOOTCFG3 | Mass erase and factory reset mode policies.<br>These policies affect SWD initiated and BSL initiated mass erase and factory reset commands.<br>If the SW-DP is disabled (SWDP_MODE is disabled), SWD initiated commands are not allowed as the SW-DP is fully disabled.<br>If the BSL is disabled (BSLMODE is disabled), these settings are a a don't care for BSL initiated commands as the BSL is not enabled to be invoked. | | Go |
| 41C00024h + formula | PWDMASSERASE[y] | SWD mass erase command password (must be provided via DSSM to authenticate a mass erase command) | | Go |
| 41C00034h + formula | PWDFACTORYRESET[y] | SWD factory reset command password (must be provided via DSSM to authenticate a factory reset command) | | Go |
| 41C00044h | FLASHSWP0 | Static write protection policy for the first 32kB of flash memory.<br>When protected, sectors will not be available for program or erase by either the bootloader or application code | | Go |
| 41C00048h | FLASHSWP1 | Static write protection policy for additional sectors of flash memory.<br>When protected, sectors will not be available for program or erase by either the bootloader or application code. | | Go |
| 41C0004Ch | BOOTCFG4 | Nonmain protection and Application CRC check policies | | Go |
| 41C00050h | APPCRCSTART | Start address of the application CRC check (must be an address in a MAIN flash region). | | Go |
| 41C00054h | APPCRCLENGTH | Length of the application area to include in the application CRC check (in bytes), starting from APPCRCSTART. | | Go |
| 41C00058h | APPCRC | Expected application CRC check digest (CRC-32) to test against during boot. | | Go |



Copyright © 2026 Texas Instruments Incorporated

**Table 1-7. NONMAIN_TYPEA Registers (continued)**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 41C0005 Ch | BOOTCRC | CRC digest of the BCR configuration portion of the NONMAIN memory. | | Go |
| 41C0010 0h | BSLCONFIGID | Configuration ID of BSL Structure | | Go |
| 41C0010 4h | BSLPINCFG0 | BSL UART PIN Configuration. The reset value of these fields differ for each device, based on the pins used. | | Go |
| 41C0010 8h | BSLPINCFG1 | BSL I2C PIN Configuration. The reset value of these fields differ for each device, based on the pins used. | | Go |
| 41C0010 Ch | BSLCONFIG0 | BSL invoke pin configuration and memory read-out policy. | | Go |
| 41C0011 0h + formula | PWDBSL[y] | BSL access password | | Go |
| 41C0013 0h | BSLPLUGINCFG | Defines the presence and type of a BSL plug-in in MAIN flash memory. | | Go |
| 41C0013 4h + formula | BSLPLUGINHOOK[y] | Function pointers for plug-in init, receive, transmit, and de-init functions. | | Go |
| 41C0014 4h | BSLCONFIG1 | | | Go |
| 41C0014 8h | SBLADDRESS | Address of an alternate BSL. | | Go |
| 41C0014 Ch | BSLAPPVER | Address of the application version word. | | Go |
| 41C0015 0h | BSLCONFIG2 | Configures the BSL Alert Configuration and I2C Slave address of BSL. | | Go |
| 41C0015 4h | BSLCRC | CRC digest of the BSL_CONFIG portion of the NONMAIN memory. | | Go |

Complex bit access types are encoded to fit into small table cells. Table 1-8 shows the codes that are used for access types in this section.

**Table 1-8. NONMAIN_TYPEA Access Type Codes**

| Access Type | Code | Description |
|---|---|---|
| Read Type | | |
| R | R | Read |
| Write Type | | |
| W | W | Write |
| Reset or Default Value | | |
| *-n* | | Value after reset or the default value |
| Register Array Variables | | |
| i,j,k,l,m,n | | When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula. |
| y | | When this variable is used in a register name, an offset, or an address it refers to the value of a register array. |



Copyright © 2026 Texas Instruments Incorporated

#### 1.4.5.1 BCRCONFIGID (Offset = 41C00000h) [Reset = 00000000h]

BCRCONFIGID is shown in Figure 1-2 and described in Table 1-9.

Return to the Summary Table.

Configuration ID of BCR Structure

**Figure 1-2. BCRCONFIGID**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CONFIG | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-9. BCRCONFIGID Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | CONFIG | R/W | 0h | Configuration ID of the BCR Structure |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.5.2 BOOTCFG0 (Offset = 41C00004h) [Reset = AABBAABBh]

BOOTCFG0 is shown in Figure 1-3 and described in Table 1-10.

Return to the Summary Table.

Serial wire debug (SWD) lock policy.

**Figure 1-3. BOOTCFG0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SWDP_MODE | | | | | | | | | | | | | | | | DEBUGACCESS | | | | | | | | | | | | | | | |
| R/W-AABBh | | | | | | | | | | | | | | | | W-AABBh | | | | | | | | | | | | | | | |

**Table 1-10. BOOTCFG0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | SWDP_MODE | R/W | AABBh | The serial wire debug port (SW-DP) access policy. This policy sets whether any communication is allowed with the device via the SWD pins (to any DAP). When disabled, no SWD communication is possible regardless of the configuration of the DEBUGACCESS field.<br>AABBh = The SW-DP is enabled and device access is set by the additional policies in NONMAIN<br>FFFFh = The SW-DP is fully disabled and no device access is possible via the SW-DP (Any value other than 0xAABB) |
| 15-0 | DEBUGACCESS | W | AABBh | The debug access policy for accessing the AHB-AP, ET-AP, and PWR-AP debug access ports. Note that if SWDP_MODE is set to DISABLED, the value of this field is ignored and the debug port will remain fully locked<br>AABBh = Access to AHB-AP, ET-AP, and PWR-AP via SWD is enabled<br>CCDDh = Access to AHB-AP, ET-AP, and PWR-AP via SWD is only enabled when the correct password is provided via the DSSM before BCR execution<br>FFFFh = Access to AHB-AP, ET-AP, and PWR-AP via SWD is disabled (Any value other than 0xCCDD or 0xAABB) |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.5.3 BOOTCFG1 (Offset = 41C00008h) [Reset = AABBAABBh]

BOOTCFG1 is shown in Figure 1-4 and described in Table 1-11.

Return to the Summary Table.

TI Failure Analysis (FA) mode and BSL invoke pin policies.

**Figure 1-4. BOOTCFG1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BSL_PIN_INVOKE | | | | | | | | | | | | | | | | TI_FA_MODE | | | | | | | | | | | | | | | |
| R/W-AABBh | | | | | | | | | | | | | | | | R/W-AABBh | | | | | | | | | | | | | | | |

**Table 1-11. BOOTCFG1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | BSL_PIN_INVOKE | R/W | AABBh | Boot strap loader (BSL) pin invoke method enable/disable policy<br>AABBh = The BSL_INVOKE pin is checked during boot<br>FFFFh = The BSL_INVOKE pin is not checked during boot (Any value other than 0xAABB) |
| 15-0 | TI_FA_MODE | R/W | AABBh | Sets the TI failure analysis enable/disable policy. If enabled, a re-test request through DSSM is allowed, else it is not allowed. Note that if SWDP_MODE is set to disabled, this field is ignored and failure analysis is not possible.<br>AABBh = TI failure analysis is allowed<br>FFFFh = TI failure analysis is not allowed (Any value other than 0xAABB) |



Copyright © 2026 Texas Instruments Incorporated

#### 1.4.5.4 PWDDEBUGLOCK[y] (Offset = 41C0000Ch + formula) [Reset = FFFFFFFFh]

PWDDEBUGLOCK[y] is shown in Figure 1-5 and described in Table 1-12.

Return to the Summary Table.

Debug Lock Password. This password is used when the BOOTCFG0.DEBUGACCESS field is configured as Enable with password.

Offset = 41C0000Ch + (y * 4h); where y = 0h to 3h

**Figure 1-5. PWDDEBUGLOCK[y]**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PASSWORD | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-FFFFFFFFh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-12. PWDDEBUGLOCK[y] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | PASSWORD | R/W | FFFFFFFFh | Debug Lock Password |



Copyright © 2026 Texas Instruments Incorporated

#### 1.4.5.5 BOOTCFG2 (Offset = 41C0001Ch) [Reset = AABBFFFFh]

BOOTCFG2 is shown in Figure 1-6 and described in Table 1-13.

Return to the Summary Table.

Fast boot mode policy and BSL mode policy

**Figure 1-6. BOOTCFG2**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BSLMODE | | | | | | | | | | | | | | | | FASTBOOTMODE | | | | | | | | | | | | | | | |
| R/W-AABBh | | | | | | | | | | | | | | | | R/W-FFFFh | | | | | | | | | | | | | | | |

**Table 1-13. BOOTCFG2 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | BSLMODE | R/W | AABBh | BSLMODE configures the boot strap loader enable/disable policy<br>AABBh = The BSL is enabled<br>FFFFh = The BSL is disabled (Any value other than 0xAABB) |
| 15-0 | FASTBOOTMODE | R/W | FFFFh | FASTBOOTMODE configures the fast boot mode enable/disable policy<br>AABBh = Fast boot mode is enabled<br>FFFFh = Fast boot mode is disabled. (Any value other than 0xAABB) |



Copyright © 2026 Texas Instruments Incorporated

#### 1.4.5.6 BOOTCFG3 (Offset = 41C00020h) [Reset = AABBAABBh]

BOOTCFG3 is shown in Figure 1-7 and described in Table 1-14.

Return to the Summary Table.

Mass erase and factory reset mode policies. These policies affect SWD initiated and BSL initiated mass erase and factory reset commands. If the SW-DP is disabled (SWDP_MODE is disabled), SWD initiated commands are not allowed as the SW-DP is fully disabled. If the BSL is disabled (BSLMODE is disabled), these settings are a a don't care for BSL initiated commands as the BSL is not enabled to be invoked.

**Figure 1-7. BOOTCFG3**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FACTORYRESETCMDACCESS | | | | | | | | | | | | | | | |
| R/W-AABBh | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MASSERASECMDACCESS | | | | | | | | | | | | | | | |
| R/W-AABBh | | | | | | | | | | | | | | | |

**Table 1-14. BOOTCFG3 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | FACTORYRESETCMDACCESS | R/W | AABBh | The factory reset command policy<br>AABBh = The factory reset command is allowed<br>CCDDh = The factory reset command is allowed only when the matching password is provided via the DSSM<br>FFFFh = The factory reset command is not allowed (Any value other than 0xAABB or 0xCCDD) |
| 15-0 | MASSERASECMDACCESS | R/W | AABBh | The mass erase command policy<br>AABBh = The mass erase command is allowed<br>CCDDh = The mass erase command is allowed only when the matching password is provided via the DSSM<br>FFFFh = The mass erase command is not allowed (Any value other than 0xAABB or 0xCCDD) |



Copyright © 2026 Texas Instruments Incorporated

#### 1.4.5.7 PWDMASSERASE[y] (Offset = 41C00024h + formula) [Reset = FFFFFFFFh]

PWDMASSERASE[y] is shown in Figure 1-8 and described in Table 1-15.

Return to the Summary Table.

SWD mass erase command password (must be provided via DSSM to authenticate a mass erase command)

Offset = 41C00024h + (y * 4h); where y = 0h to 3h

**Figure 1-8. PWDMASSERASE[y]**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PASSWORD | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-FFFFFFFFh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-15. PWDMASSERASE[y] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | PASSWORD | R/W | FFFFFFFFh | SHA2-256 Bit digest of Mass Erase Password |



Copyright © 2026 Texas Instruments Incorporated

#### 1.4.5.8 PWDFACTORYRESET[y] (Offset = 41C00034h + formula) [Reset = FFFFFFFFh]

PWDFACTORYRESET[y] is shown in Figure 1-9 and described in Table 1-16.

Return to the Summary Table.

SWD factory reset command password (must be provided via DSSM to authenticate a factory reset command)

Offset = 41C00034h + (y * 4h); where y = 0h to 3h

**Figure 1-9. PWDFACTORYRESET[y]**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PASSWORD | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-FFFFFFFFh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-16. PWDFACTORYRESET[y] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | PASSWORD | R/W | FFFFFFFFh | SHA2-256 Bit digest of Factory Reset Password |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.5.9 FLASHSWP0 (Offset = 41C00044h) [Reset = FFFFFFFFh]

FLASHSWP0 is shown in Figure 1-10 and described in Table 1-17.

Return to the Summary Table.

Static write protection policy for the first 32kB of flash memory. When protected, sectors will not be available for program or erase by either the bootloader or application code

**Figure 1-10. FLASHSWP0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-FFFFFFFFh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-17. FLASHSWP0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | R/W | FFFFFFFFh | 1 bit per sector<br>0h = A value of 0 indicates the write protection is applied to the corresponding sector<br>1h = A value of 1 indicates the sector is un-protected |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.5.10 FLASHSWP1 (Offset = 41C00048h) [Reset = FFFFFFFFh]

FLASHSWP1 is shown in Figure 1-11 and described in Table 1-18.

Return to the Summary Table.

Static write protection policy for additional sectors of flash memory. When protected, sectors will not be available for program or erase by either the bootloader or application code.

**Figure 1-11. FLASHSWP1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-FFFFFFFFh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-18. FLASHSWP1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | R/W | FFFFFFFFh | 1 bit per 8 sectors. This is used to protect Flash memory from (32 to 256 kb)<br>0h = A value of 0 indicates the write protection is applied to the corresponding sectors<br>1h = A value of 1 indicates the sector is un-protected |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.5.11 BOOTCFG4 (Offset = 41C0004Ch) [Reset = FFFFFFFFh]

BOOTCFG4 is shown in Figure 1-12 and described in Table 1-19.

Return to the Summary Table.

Nonmain protection and Application CRC check policies

**Figure 1-12. BOOTCFG4**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| APPCRCMODE | | | | | | | | | | | | | | | | NONMAINSWP | | | | | | | | | | | | | | | |
| R/W-FFFFh | | | | | | | | | | | | | | | | R/W-FFFFh | | | | | | | | | | | | | | | |

**Table 1-19. BOOTCFG4 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | APPCRCMODE | R/W | FFFFh | APPCRCMODE enables or disables the boot time CRC check of a segment of MAIN flash memory.<br>AABBh = The boot time MAIN flash CRC check is enabled. If the boot time CRC check passes, the application code in MAIN flash is started unless the reset vector or stack pointer are blank (unprogrammed). In the event of a failing CRC check, the application code in MAIN flash will not be started and the boot process fails.<br>FFFFh = The boot time MAIN flash CRC check is disabled. The application code in MAIN flash is always started unless the reset vector or stack pointer are blank (unprogrammed). |
| 15-0 | NONMAINSWP | R/W | FFFFh | Static write protection policy for entire NONMAIN device configuration memory. Only Bit 0 is used.<br>0h = Protected, does not allow program/erase of the NONMAIN by all means other than a SWD-initiated factory reset<br>1h = Unprotected, allows program/erase of the NONMAIN by normal means. |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.5.12 APPCRCSTART (Offset = 41C00050h) [Reset = FFFFFFFFh]

APPCRCSTART is shown in Figure 1-13 and described in Table 1-20.

Return to the Summary Table.

Start address of the application CRC check (must be an address in a MAIN flash region).

**Figure 1-13. APPCRCSTART**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADDRESS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-FFFFFFFFh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-20. APPCRCSTART Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | ADDRESS | R/W | FFFFFFFFh | Application CRC check start address |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.5.13 APPCRCLENGTH (Offset = 41C00054h) [Reset = FFFFFFFFh]

APPCRCLENGTH is shown in Figure 1-14 and described in Table 1-21.

Return to the Summary Table.

Length of the application area to include in the application CRC check (in bytes), starting from APPCRCSTART.

**Figure 1-14. APPCRCLENGTH**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LENGTH | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-FFFFFFFFh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-21. APPCRCLENGTH Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | LENGTH | R/W | FFFFFFFFh | Application CRC check source data length |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.5.14 APPCRC (Offset = 41C00058h) [Reset = FFFFFFFFh]

APPCRC is shown in Figure 1-15 and described in Table 1-22.

Return to the Summary Table.

Expected application CRC check digest (CRC-32) to test against during boot.

**Figure 1-15. APPCRC**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DIGEST | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-FFFFFFFFh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-22. APPCRC Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DIGEST | R/W | FFFFFFFFh | Application CRC check digest |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.5.15 BOOTCRC (Offset = 41C0005Ch) [Reset = 00000000h]

BOOTCRC is shown in Figure 1-16 and described in Table 1-23.

Return to the Summary Table.

CRC digest of the BCR configuration portion of the NONMAIN memory.

**Figure 1-16. BOOTCRC**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DIGEST | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-23. BOOTCRC Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DIGEST | R/W | 0h | BCR configuration data CRC digest. 32 bit CRC Digest, if the device supports CRC32-ISO3309. Otherwise 16 bit CRC digest, using polynomial CRC16-CCITT. Configuration to be used for CRC calulation: 1. Polynomial should be as per the chosen standard, 2. Input reflected, 3. Output reflected, 4. Initial value should be 0xFFFFFFFF, 5. Final XOR value should be 0x0. |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.5.16 BSLCONFIGID (Offset = 41C00100h) [Reset = 00000000h]

BSLCONFIGID is shown in Figure 1-17 and described in Table 1-24.

Return to the Summary Table.

Configuration ID of BSL Structure

**Figure 1-17. BSLCONFIGID**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CONFIG | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-24. BSLCONFIGID Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | CONFIG | R/W | 0h | Configuration ID of BSL Structure |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.5.17 BSLPINCFG0 (Offset = 41C00104h) [Reset = 00000000h]

BSLPINCFG0 is shown in Figure 1-18 and described in Table 1-25.

Return to the Summary Table.

BSL UART PIN Configuration. The reset value of these fields differ for each device, based on the pins used.

**Figure 1-18. BSLPINCFG0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| UARTTX_MUX_SEL | | | | | | | | UARTTX_PAD_NUM | | | | | | | |
| R/W-0h | | | | | | | | R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| UARTRX_MUX_SEL | | | | | | | | UARTRX_PAD_NUM | | | | | | | |
| R/W-0h | | | | | | | | R/W-0h | | | | | | | |

**Table 1-25. BSLPINCFG0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | UARTTX_MUX_SEL | R/W | 0h | UART TX IOMUX PINCM mux selection. |
| 23-16 | UARTTX_PAD_NUM | R/W | 0h | UART TX IOMUX PINCM register |
| 15-8 | UARTRX_MUX_SEL | R/W | 0h | UART RX IOMUX PINCM mux selection |
| 7-0 | UARTRX_PAD_NUM | R/W | 0h | UART RX IOMUX PINCM register |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.5.18 BSLPINCFG1 (Offset = 41C00108h) [Reset = 00000000h]

BSLPINCFG1 is shown in Figure 1-19 and described in Table 1-26.

Return to the Summary Table.

BSL I2C PIN Configuration. The reset value of these fields differ for each device, based on the pins used.

**Figure 1-19. BSLPINCFG1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| I2CSCL_MUX_SEL | | | | | | | | I2CSCL_PAD_NUM | | | | | | | |
| R/W-0h | | | | | | | | R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| I2CSDA_MUX_SEL | | | | | | | | I2CSDA_PAD_NUM | | | | | | | |
| R/W-0h | | | | | | | | R/W-0h | | | | | | | |

**Table 1-26. BSLPINCFG1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | I2CSCL_MUX_SEL | R/W | 0h | I2C SCL IOMUX PINCM mux selection |
| 23-16 | I2CSCL_PAD_NUM | R/W | 0h | I2C SCL IOMUX PINCM register |
| 15-8 | I2CSDA_MUX_SEL | R/W | 0h | I2C SDA IOMUX PINCM mux selection |
| 7-0 | I2CSDA_PAD_NUM | R/W | 0h | I2C SDA IOMUX PINCM register |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.5.19 BSLCONFIG0 (Offset = 41C0010Ch) [Reset = 0000XX00h]

BSLCONFIG0 is shown in Figure 1-20 and described in Table 1-27.

Return to the Summary Table.

BSL invoke pin configuration and memory read-out policy.

**Figure 1-20. BSLCONFIG0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| READOUTEN | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| READOUTEN | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | BSLIVK_GPIOPORT | BSLIVK_GPIOPIN | | | | |
| R-0h | | R/W-Xh | R/W-Xh | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BSLIVK_LVL | RESERVED | BSLIVK_PAD_NUM | | | | | |
| R/W-0h | R-0h | R/W-0h | | | | | |

**Table 1-27. BSLCONFIG0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | READOUTEN | R/W | FFFFh | Sets the memory read-out policy for the BSL interface<br>AABBh = Memory contents can be read via the BSL interface<br>FFFFh = Memory read-out is not possible via the BSL interface (Any value other than 0xAABB) |
| 15-14 | RESERVED | R | 0h | |
| 13 | BSLIVK_GPIOPORT | R/W | x | The BSL_invoke GPIO port index corresponding to the pad used for BSL_invoke<br>0h = The BSL_invoke pin is on GPIO port A<br>1h = The BSL_invoke pin is on GPIO port B |
| 12-8 | BSLIVK_GPIOPIN | R/W | x | The BSL_invoke GPIO pin index corresponding to the pad used for BSL_invoke. Valid values are 0 to 31. |
| 7 | BSLIVK_LVL | R/W | 1h | The BSL_invoke input logic level which shall invoke the BSL<br>0h = Pin state LOW will invoke BSL<br>1h = Pin state HIGH will invoke BSL |
| 6 | RESERVED | R | 0h | |
| 5-0 | BSLIVK_PAD_NUM | R/W | 0h | The IOMUX PINCM register corresponding to the pad to be used for BSL_invoke |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.5.20 PWDBSL[y] (Offset = 41C00110h + formula) [Reset = FFFFFFFFh]

PWDBSL[y] is shown in Figure 1-21 and described in Table 1-28.

Return to the Summary Table.

BSL access password

Offset = 41C00110h + (y * 4h); where y = 0h to 7h

**Figure 1-21. PWDBSL[y]**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PASSWORD | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-FFFFFFFFh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-28. PWDBSL[y] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | PASSWORD | R/W | FFFFFFFFh | BSL access Password of size 256 bit |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.5.21 BSLPLUGINCFG (Offset = 41C00130h) [Reset = FFFFFFFFh]

BSLPLUGINCFG is shown in Figure 1-22 and described in Table 1-29.

Return to the Summary Table.

Defines the presence and type of a BSL plug-in in MAIN flash memory.

**Figure 1-22. BSLPLUGINCFG**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SRAMUSED | | | | | | | | FLASHPLUGINEXISTS | | | | | | | |
| R/W-FFh | | | | | | | | R/W-FFh | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PLUGINTYPE | | | | | | | | | | | | | | | |
| R/W-FFFFh | | | | | | | | | | | | | | | |

**Table 1-29. BSLPLUGINCFG Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | SRAMUSED | R/W | FFh | SRAM consumed by Flash plugin, from 0x00 to 0xFF. |
| 23-16 | FLASHPLUGINEXISTS | R/W | FFh | The field tells if Flash Plugin exists are not<br>BBh = Flash Plugin exists<br>FFh = Only ROM plugin will be used |
| 15-0 | PLUGINTYPE | R/W | FFFFh | The Plugin type specifies which interface plugin is added in Flash. When the same interface is supported by ROM Bootloader, then ROM interface will be overridden with the Flash interface. If the same interface as in ROM BSL should be added as an additional interface without overriding ROM interface, a new value that is not part of this list should be chosen.<br>1000h = Plug-in is for UART interface. ROM UART interface will be overriden with this Flash plug-in.<br>2000h = Plug-in is for I2C interface. ROM I2C interface will be overriden with this Flash plug-in.<br>3000h = Plug-is for a new interface. This interface should be added on top of interfaces supported by ROM BSL. It can be any value other than 0x1000, 0x2000 and 0xFFFF.<br>FFFFh = For all other values. Any other interfaces with valid hooks will be added to Plugin list. |



Copyright © 2026 Texas Instruments Incorporated

#### 1.4.5.22 BSLPLUGINHOOK[y] (Offset = 41C00134h + formula) [Reset = FFFFFFFFh]

BSLPLUGINHOOK[y] is shown in Figure 1-23 and described in Table 1-30.

Return to the Summary Table.

Function pointers for plug-in init, receive, transmit, and de-init functions.

Offset = 41C00134h + (y * 4h); where y = 0h to 3h

**Figure 1-23. BSLPLUGINHOOK[y]**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BSLPLUGININIT | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-FFFFFFFFh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-30. BSLPLUGINHOOK[y] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 127-96 | BSLPLUGINDEINIT | R/W | FFFFFFFFh | Function pointer for the Deinit API |
| 95-64 | BSLPLUGINSEND | R/W | FFFFFFFFh | Function pointer for the Send API |
| 63-32 | BSLPLUGINRECEIVE | R/W | FFFFFFFFh | Function pointer for the Receive API |
| 31-0 | BSLPLUGININIT | R/W | FFFFFFFFh | Function pointer for the Init API |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.5.23 BSLCONFIG1 (Offset = 41C00144h) [Reset = FFFFFFFFh]

BSLCONFIG1 is shown in Figure 1-24 and described in Table 1-31.

Return to the Summary Table.

**Figure 1-24. BSLCONFIG1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | ALTBSLCONFIG | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | R/W-FFFFh | | | | | | | | | | | | | | | |

**Table 1-31. BSLCONFIG1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | FFFFh | |
| 15-0 | ALTBSLCONFIG | R/W | FFFFh | Controls the invocation of alternate BSL in main flash region.<br>AABBh = Uses the alternate BSL<br>FFFFh = Does not use alternate BSL. Uses ROM BSL. |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.5.24 SBLADDRESS (Offset = 41C00148h) [Reset = FFFFFFFFh]

SBLADDRESS is shown in Figure 1-25 and described in Table 1-32.

Return to the Summary Table.

Address of an alternate BSL.

**Figure 1-25. SBLADDRESS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADDRESS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-FFFFFFFFh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-32. SBLADDRESS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | ADDRESS | R/W | FFFFFFFFh | Address of the alternate BSL, if present. |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.5.25 BSLAPPVER (Offset = 41C0014Ch) [Reset = FFFFFFFFh]

BSLAPPVER is shown in Figure 1-26 and described in Table 1-33.

Return to the Summary Table.

Address of the application version word.

**Figure 1-26. BSLAPPVER**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADDRESS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-FFFFFFFFh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-33. BSLAPPVER Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | ADDRESS | R/W | FFFFFFFFh | Address of the application version word (must be a valid flash address to be returned). If the given flash address is not programmed, then 0h will be returned. |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.5.26 BSLCONFIG2 (Offset = 41C00150h) [Reset = 0048FFFFh]

BSLCONFIG2 is shown in Figure 1-27 and described in Table 1-34.

Return to the Summary Table.

Configures the BSL Alert Configuration and I2C Slave address of BSL.

**Figure 1-27. BSLCONFIG2**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| I2CTARGETADDR | | | | | | | | | | | | | | | | ALERTACTION | | | | | | | | | | | | | | | |
| R/W-48h | | | | | | | | | | | | | | | | R/W-FFFFh | | | | | | | | | | | | | | | |

**Table 1-34. BSLCONFIG2 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | I2CTARGETADDR | R/W | 0048h | I2C target address to be used for the ROM BSL I2C communication. |
| 15-0 | ALERTACTION | R/W | FFFFh | Action to take upon a security alert condition.<br>AABBh = Trigger a factory reset. Note that if sectors in MAIN or NONMAIN flash are write protected they will not be affected by the BSL factory reset<br>CCDDh = Re-configure the NONMAIN region to disable the BSL. This is not supported if the NONMAIN region is configured to be write protected<br>000FFFFFh = Ignore the security alert condition (Any value other than 0xAABB) |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.5.27 BSLCRC (Offset = 41C00154h) [Reset = XXXXXXXXh]

BSLCRC is shown in Figure 1-28 and described in Table 1-35.

Return to the Summary Table.

CRC digest of the BSL_CONFIG portion of the NONMAIN memory.

**Figure 1-28. BSLCRC**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DIGEST | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-35. BSLCRC Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DIGEST | R/W | X | BSL configuration data CRC digest. 32 bit CRC Digest, if the device supports CRC32-ISO3309. Otherwise 16 bit CRC digest, using polynomial CRC16-CCITT. Configuration to be used for CRC calulation: 1. Polynomial should be as per the chosen standard, 2. Input reflected, 3. Output reflected, 4. Initial value should be 0xFFFFFFFF, 5. Final XOR value should be 0x0.<br>xxxxxxxx |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.6 NONMAIN_TYPEE Registers

Table 1-36 lists the memory-mapped registers for the NONMAIN_TYPEE registers. All register offset addresses not listed in Table 1-36 should be considered as reserved locations and the register contents should not be modified.

**Table 1-36. NONMAIN_TYPEE Registers**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 41C00000h | BCRCONFIGID | Configuration ID of BCR Structure | | Go |
| 41C00004h | BOOTCFG0 | Serial wire debug (SWD) lock policy. | | Go |
| 41C00008h | BOOTCFG1 | TI Failure Analysis (FA) mode and BSL invoke pin policies | | Go |
| 41C0000Ch | FLASHSWP0 | Static write protection policy for the first 32kB of flash memory.<br>When protected, sectors will not be available for program or erase by either the bootloader or application code | | Go |
| 41C00010h | FLASHSWP1 | Static write protection policy for additional sectors of flash memory.<br>When protected, sectors will not be available for program or erase by either the bootloader or application code | | Go |
| 41C00014h | BOOTCFG4 | Configures Static write protection policy for Configuration flash memory(NONMAIN) and Debug Hold Policy | | Go |
| 41C00018h | BOOTCFG5 | Configures the existence of Customer Secure Code(CSC) and Flash Bank Swap Policy | | Go |
| 41C0001Ch | BOOTCFG2 | Fast boot mode policy and BSL mode policy | | Go |
| 41C00020h | BOOTCFG3 | Mass erase and factory reset mode policies.<br>These policies affect SWD initiated and BSL initiated mass erase and factory reset commands.<br>If the SW-DP is disabled (SWDP_MODE is disabled), SWD initiated commands are not allowed as the SW-DP is fully disabled.<br>If the BSL is disabled (BSLMODE is disabled), these settings are a a don't care for BSL initiated commands as the BSL is not enabled to be invoked. | | Go |
| 41C00024h + formula | PWDMASSERASE[y] | SHA2-256 Digest of the SWD mass erase command password (must be provided via DSSM to authenticate a mass erase command) | | Go |
| 41C00044h + formula | PWDFACTORYRESET[y] | SHA2-256 digest of the SWD factory reset command password (must be provided via DSSM to authenticate a factory reset command) | | Go |
| 41C00064h + formula | PWDDEBUGLOCK[y] | SHA2-256 Digest of the Debug Lock Password.<br>This password is used when the BOOTCFG0.DEBUGACCESS field is configured as Enable with password. | | Go |
| 41C00084h | BOOTCFG6 | Application digest check policy | | Go |



Copyright © 2026 Texas Instruments Incorporated

**Table 1-36. NONMAIN_TYPEE Registers (continued)**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 41C00088h | APPDIGESTSTART | Start address of the application CRC/SHA Digest check (must be an address in a MAIN flash region). | | Go |
| 41C0008Ch | APPDIGESTLENGTH | Length of the application to be include in the application CRC/SHA Digest check (in bytes), starting from APPDIGESTSTART | | Go |
| 41C00090h + formula | APPDIGEST[y] | Expected application CRC32/SHA2-256 digest to test against during boot. Note: If CRC check is enabled only the first 32bit word of this DIGEST is used. | | Go |
| 41C000B0h | FLASHSWP2 | Static write protection policy for additional sectors of flash memory. When protected, sectors will not be available for program or erase by either the bootloader or application code | | Go |
| 41C000B4h | BOOTCRC | CRC digest of the BCR configuration portion of the NONMAIN memory. | | Go |
| 41C00100h | BSLCONFIGID | Configuration ID of BSL Structure | | Go |
| 41C00104h | BSLPINCFG0 | BSL UART PIN Configuration. The reset value of these fields differ for each device, based on the pins used. | | Go |
| 41C00108h | BSLPINCFG1 | BSL I2C PIN Configuration. The reset value of these fields differ for each device, based on the pins used. | | Go |
| 41C0010Ch | BSLCONFIG0 | BSL invoke pin configuration and memory read-out policy. | | Go |
| 41C00110h | PWDBSL0 | SHA2-256 Digest of BSL password. | | Go |
| 41C00114h | PWDBSL1 | SHA2-256 Digest of BSL password. | | Go |
| 41C00118h | PWDBSL2 | SHA2-256 Digest of BSL password. | | Go |
| 41C0011Ch | PWDBSL3 | SHA2-256 Digest of BSL password. | | Go |
| 41C00120h | PWDBSL4 | SHA2-256 Digest of BSL password. | | Go |
| 41C00124h | PWDBSL5 | SHA2-256 Digest of BSL password. | | Go |
| 41C00128h | PWDBSL6 | SHA2-256 Digest of BSL password. | | Go |
| 41C0012Ch | PWDBSL7 | SHA2-256 Digest of BSL password. | | Go |
| 41C00130h | BSLPLUGINCFG | Defines the presence and type of a BSL plug-in in MAIN flash memory. | | Go |
| 41C00134h + formula | BSLPLUGINHOOK[y] | Function pointers for plug-in init, receive, transmit, and de-init functions. | | Go |
| 41C00144h | BSLCONFIG1 | | | Go |
| 41C00148h | SBLADDRESS | Address of an alternate BSL. | | Go |
| 41C0014Ch | BSLAPPVER | Address of the application version word. | | Go |



Copyright © 2026 Texas Instruments Incorporated

**Table 1-36. NONMAIN_TYPEE Registers (continued)**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 41C00150h | BSLCONFIG2 | Configures the BSL Alert Configuration and I2C Slave address of BSL. | | Go |
| 41C00154h | BSLCONFIG3 | Disable NRST policy | | Go |
| 41C00158h | RESERVED | | | |
| 41C0015Ch | BSLCRC | CRC digest of the BSL_CONFIG portion of the NONMAIN memory. | | Go |

Complex bit access types are encoded to fit into small table cells. Table 1-37 shows the codes that are used for access types in this section.

**Table 1-37. NONMAIN_TYPEE Access Type Codes**

| Access Type | Code | Description |
|---|---|---|
| Read Type | | |
| R | R | Read |
| Write Type | | |
| W | W | Write |
| Reset or Default Value | | |
| -*n* | | Value after reset or the default value |
| Register Array Variables | | |
| i,j,k,l,m,n | | When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula. |
| y | | When this variable is used in a register name, an offset, or an address it refers to the value of a register array. |



Copyright © 2026 Texas Instruments Incorporated

#### 1.4.6.1 BCRCONFIGID (Offset = 41C00000h) [Reset = 00000000h]

BCRCONFIGID is shown in Figure 1-29 and described in Table 1-38.

Return to the Summary Table.

Configuration ID of BCR Structure

**Figure 1-29. BCRCONFIGID**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CONFIG | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-38. BCRCONFIGID Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | CONFIG | R/W | 0h | Configuration ID of the BCR Structure |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.6.2 BOOTCFG0 (Offset = 41C00004h) [Reset = AABBAABBh]

BOOTCFG0 is shown in Figure 1-30 and described in Table 1-39.

Return to the Summary Table.

Serial wire debug (SWD) lock policy.

**Figure 1-30. BOOTCFG0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SWDP_MODE | | | | | | | | | | | | | | | | DEBUGACCESS | | | | | | | | | | | | | | | |
| R/W-AABBh | | | | | | | | | | | | | | | | W-AABBh | | | | | | | | | | | | | | | |

**Table 1-39. BOOTCFG0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | SWDP_MODE | R/W | AABBh | The serial wire debug port (SW-DP) access policy. This policy sets whether any communication is allowed with the device via the SWD pins (to any DAP). When disabled, no SWD communication is possible regardless of the configuration of the DEBUGACCESS field.<br>AABBh = The SW-DP is enabled and device access is set by the additional policies in NONMAIN<br>FFFFh = The SW-DP is fully disabled and no device access is possible via the SW-DP (Any value other than 0xAABB) |
| 15-0 | DEBUGACCESS | W | AABBh | The debug access policy for accessing the AHB-AP, ET-AP, and PWR-AP debug access ports. Note that if SWDP_MODE is set to DISABLED, the value of this field is ignored and the debug port will remain fully locked<br>AABBh = Access to AHB-AP, ET-AP, and PWR-AP via SWD is enabled<br>CCDDh = Access to AHB-AP, ET-AP, and PWR-AP via SWD is only enabled when the correct password is provided via the DSSM before BCR execution<br>FFFFh = Access to AHB-AP, ET-AP, and PWR-AP via SWD is disabled (Any value other than 0xCCDD or 0xAABB) |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.6.3 BOOTCFG1 (Offset = 41C00008h) [Reset = AABBAABBh]

BOOTCFG1 is shown in Figure 1-31 and described in Table 1-40.

Return to the Summary Table.

TI Failure Analysis (FA) mode and BSL invoke pin policies

**Figure 1-31. BOOTCFG1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BSL_PIN_INVOKE | | | | | | | | | | | | | | | | TI_FA_MODE | | | | | | | | | | | | | | | |
| R/W-AABBh | | | | | | | | | | | | | | | | R/W-AABBh | | | | | | | | | | | | | | | |

**Table 1-40. BOOTCFG1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | BSL_PIN_INVOKE | R/W | AABBh | Boot strap loader (BSL) pin invoke method enable/disable policy<br>AABBh = The BSL_INVOKE pin is checked during boot<br>FFFFh = The BSL_INVOKE pin is not checked during boot (Any value other than 0xAABB) |
| 15-0 | TI_FA_MODE | R/W | AABBh | Sets the TI failure analysis enable/disable policy. If enabled, a re-test request through DSSM is allowed, else it is not allowed. Note that if SWDP_MODE is set to disabled, this field is ignored and failure analysis is not possible.<br>AABBh = TI failure analysis is allowed<br>FFFFh = TI failure analysis is not allowed (Any value other than 0xAABB) |



Copyright © 2026 Texas Instruments Incorporated

#### 1.4.6.4 FLASHSWP0 (Offset = 41C0000Ch) [Reset = FFFFFFFFh]

FLASHSWP0 is shown in Figure 1-32 and described in Table 1-41.

Return to the Summary Table.

Static write protection policy for the first 32kB of flash memory. When protected, sectors will not be available for program or erase by either the bootloader or application code

**Figure 1-32. FLASHSWP0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-FFFFFFFFh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-41. FLASHSWP0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | R/W | FFFFFFFFh | 1 bit per sector<br>0h = A value of 0 indicates the write protection is applied to the corresponding sector<br>1h = A value of 1 indicates the sector is un-protected |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.6.5 FLASHSWP1 (Offset = 41C00010h) [Reset = FFFFFFFFh]

FLASHSWP1 is shown in Figure 1-33 and described in Table 1-42.

Return to the Summary Table.

Static write protection policy for additional sectors of flash memory. When protected, sectors will not be available for program or erase by either the bootloader or application code

**Figure 1-33. FLASHSWP1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-FFFFFFFFh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-42. FLASHSWP1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | R/W | FFFFFFFFh | 1 bit per 8 sectors. This is used to protect Flash memory from (0 to 256 kb)<br>0h = A value of 0 indicates the write protection is applied to the corresponding sectors<br>1h = A value of 1 indicates the sector is un-protected |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.6.6 BOOTCFG4 (Offset = 41C00014h) [Reset = AABBAABBh]

BOOTCFG4 is shown in Figure 1-34 and described in Table 1-43.

Return to the Summary Table.

Configures Static write protection policy for Configuration flash memory(NONMAIN) and Debug Hold Policy

**Figure 1-34. BOOTCFG4**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DEBUGHOLD | | | | | | | | | | | | | | | | NONMAINSWP | | | | | | | | | | | | | | | |
| R/W-AABBh | | | | | | | | | | | | | | | | R/W-AABBh | | | | | | | | | | | | | | | |

**Table 1-43. BOOTCFG4 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | DEBUGHOLD | R/W | AABBh | Controls the release of debug access until INITDONE is issued by Customer Secure Code(CSC). This configuration is applicable only when DEBUGACCESS and CSCEXISTS is enabled.<br>AABBh = Debug Hold is disabled. Debug access should be available during CSC execution.<br>FFFFh = Debug Hold is enabled. Debug access will not be available during CSC execution. Access will be available for application debug after CSC INITDONE. (Any value other than 0xAABB) |
| 15-0 | NONMAINSWP | R/W | AABBh | Static write protection policy for entire NONMAIN device configuration memory.<br>AABBh = Protection is disabled. Allows program/erase of the NONMAIN by normal means.<br>FFFFh = Protection is enabled. Does not allow program/erase of the NONMAIN by all means other than a SWD-initiated factory reset (Any value other than 0xAABB) |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.6.7 BOOTCFG5 (Offset = 41C00018h) [Reset = AABBAABBh]

BOOTCFG5 is shown in Figure 1-35 and described in Table 1-44.

Return to the Summary Table.

Configures the existence of Customer Secure Code(CSC) and Flash Bank Swap Policy

**Figure 1-35. BOOTCFG5**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FLASHBANKSWAPPOLICY | | | | | | | | | | | | | | | |
| R/W-AABBh | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CSCEXISTS | | | | | | | | | | | | | | | |
| R/W-AABBh | | | | | | | | | | | | | | | |

**Table 1-44. BOOTCFG5 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | FLASHBANKSWAPPOLICY | R/W | AABBh | Controls the Flash Bank Swap Policy. This configuration is valid only, if CSCEXISTS is enabled.<br>AABBh = Flash Bank swap policy is disabled. In CSC Banks cannot be swapped. Complete Main flash region has Read, Write, Execute access.<br>FFFFh = Flash Bank swap policy is enabled. In CSC Banks can be swapped. One bank wil have Read, Execute access and other Bank will have Read/Write access, as per the CSC configuration. |
| 15-0 | CSCEXISTS | R/W | AABBh | Controls the existence of Customer Secure Code(CSC).<br>AABBh = CSC does not exist. INITDONE will be already SET.<br>FFFFh = CSC Exists. INITDONE will be SET by the CSC. |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.6.8 BOOTCFG2 (Offset = 41C0001Ch) [Reset = AABBFFFFh]

BOOTCFG2 is shown in Figure 1-36 and described in Table 1-45.

Return to the Summary Table.

Fast boot mode policy and BSL mode policy

**Figure 1-36. BOOTCFG2**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BSLMODE | | | | | | | | | | | | | | | | FASTBOOTMODE | | | | | | | | | | | | | | | |
| R/W-AABBh | | | | | | | | | | | | | | | | R/W-FFFFh | | | | | | | | | | | | | | | |

**Table 1-45. BOOTCFG2 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | BSLMODE | R/W | AABBh | BSLMODE configures the boot strap loader enable/disable policy<br>AABBh = The BSL is enabled<br>FFFFh = The BSL is disabled (Any value other than 0xAABB) |
| 15-0 | FASTBOOTMODE | R/W | FFFFh | FASTBOOTMODE configures the fast boot mode enable/disable policy<br>AABBh = Fast boot mode is enabled<br>FFFFh = Fast boot mode is disabled. (Any value other than 0xAABB) |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.6.9 BOOTCFG3 (Offset = 41C00020h) [Reset = AABBAABBh]

BOOTCFG3 is shown in Figure 1-37 and described in Table 1-46.

Return to the Summary Table.

Mass erase and factory reset mode policies. These policies affect SWD initiated and BSL initiated mass erase and factory reset commands. If the SW-DP is disabled (SWDP_MODE is disabled), SWD initiated commands are not allowed as the SW-DP is fully disabled. If the BSL is disabled (BSLMODE is disabled), these settings are a a don't care for BSL initiated commands as the BSL is not enabled to be invoked.

**Figure 1-37. BOOTCFG3**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FACTORYRESETCMDACCESS | | | | | | | | | | | | | | | |
| R/W-AABBh | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MASSERASECMDACCESS | | | | | | | | | | | | | | | |
| R/W-AABBh | | | | | | | | | | | | | | | |

**Table 1-46. BOOTCFG3 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | FACTORYRESETCMDACCESS | R/W | AABBh | The factory reset command policy<br>AABBh = The factory reset command is allowed<br>CCDDh = The factory reset command is allowed only when the matching password is provided via the DSSM<br>FFFFh = The factory reset command is not allowed (Any value other than 0xAABB or 0xCCDD) |
| 15-0 | MASSERASECMDACCESS | R/W | AABBh | The mass erase command policy<br>AABBh = The mass erase command is allowed<br>CCDDh = The mass erase command is allowed only when the matching password is provided via the DSSM<br>FFFFh = The mass erase command is not allowed (Any value other than 0xAABB or 0xCCDD) |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.6.10 PWDMASSERASE[y] (Offset = 41C00024h + formula) [Reset = FFFFFFFFh]

PWDMASSERASE[y] is shown in Figure 1-38 and described in Table 1-47.

Return to the Summary Table.

SHA2-256 Digest of the SWD mass erase command password (must be provided via DSSM to authenticate a mass erase command)

Offset = 41C00024h + (y * 4h); where y = 0h to 7h

**Figure 1-38. PWDMASSERASE[y]**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DIGEST | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-FFFFFFFFh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-47. PWDMASSERASE[y] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DIGEST | R/W | FFFFFFFFh | SHA2-256 Bit digest of Mass Erase Password of size 128 bit |



Copyright © 2026 Texas Instruments Incorporated

#### 1.4.6.11 PWDFACTORYRESET[y] (Offset = 41C00044h + formula) [Reset = FFFFFFFFh]

PWDFACTORYRESET[y] is shown in Figure 1-39 and described in Table 1-48.

Return to the Summary Table.

SHA2-256 digest of the SWD factory reset command password (must be provided via DSSM to authenticate a factory reset command)

Offset = 41C00044h + (y * 4h); where y = 0h to 7h

**Figure 1-39. PWDFACTORYRESET[y]**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DIGEST | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-FFFFFFFFh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-48. PWDFACTORYRESET[y] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DIGEST | R/W | FFFFFFFFh | SHA2-256 Bit digest of Factory Reset Password of size 128 bit |



Copyright © 2026 Texas Instruments Incorporated

#### 1.4.6.12 PWDDEBUGLOCK[y] (Offset = 41C00064h + formula) [Reset = FFFFFFFFh]

PWDDEBUGLOCK[y] is shown in Figure 1-40 and described in Table 1-49.

Return to the Summary Table.

SHA2-256 Digest of the Debug Lock Password. This password is used when the BOOTCFG0.DEBUGACCESS field is configured as Enable with password.

Offset = 41C00064h + (y * 4h); where y = 0h to 7h

**Figure 1-40. PWDDEBUGLOCK[y]**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DIGEST | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-FFFFFFFFh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-49. PWDDEBUGLOCK[y] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DIGEST | R/W | FFFFFFFFh | SHA2-256 Bit digest of Debug Lock Password of size 128 bit |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.6.13 BOOTCFG6 (Offset = 41C00084h) [Reset = FFFFFFFFh]

BOOTCFG6 is shown in Figure 1-41 and described in Table 1-50.

Return to the Summary Table.

Application digest check policy

**Figure 1-41. BOOTCFG6**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| APPDIGESTMODE | | | | | | | | | | | | | | | | RESERVED | | | | | | | | | | | | | | | |
| R/W-FFFFh | | | | | | | | | | | | | | | | R-0h | | | | | | | | | | | | | | | |

**Table 1-50. BOOTCFG6 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | APPDIGESTMODE | R/W | FFFFh | APPDIGESTMODE configures the boot time CRC/SHA Digest check of a segment of MAIN flash memory.<br>AABBh = The boot time MAIN flash CRC check is enabled. If the boot time CRC check passes, the application code in MAIN flash is started unless the reset vector or stack pointer are blank (unprogrammed). In the event of a failing CRC check, the application code in MAIN flash will not be started and the boot process fails.<br>CCDDh = The boot time MAIN flash SHA2-256 Hash check is enabled. If the boot time Hash check passes, the application code in MAIN flash is started unless the reset vector or stack pointer are blank (unprogrammed). In the event of a failing hash check, the application code in MAIN flash will not be started and the boot process fails.<br>FFFFh = The boot time MAIN flash Digest check is disabled. The application code in MAIN flash is always started unless the reset vector or stack pointer are blank(unprogrammed ) |
| 15-0 | RESERVED | R | FFFFh | |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.6.14 APPDIGESTSTART (Offset = 41C00088h) [Reset = FFFFFFFFh]

APPDIGESTSTART is shown in Figure 1-42 and described in Table 1-51.

Return to the Summary Table.

Start address of the application CRC/SHA Digest check (must be an address in a MAIN flash region).

**Figure 1-42. APPDIGESTSTART**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADDRESS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-FFFFFFFFh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-51. APPDIGESTSTART Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | ADDRESS | R/W | FFFFFFFFh | Application CRC/SHA Digest check start address |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.6.15 APPDIGESTLENGTH (Offset = 41C0008Ch) [Reset = FFFFFFFFh]

APPDIGESTLENGTH is shown in Figure 1-43 and described in Table 1-52.

Return to the Summary Table.

Length of the application to be include in the application CRC/SHA Digest check (in bytes), starting from APPDIGESTSTART

**Figure 1-43. APPDIGESTLENGTH**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LENGTH | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-FFFFFFFFh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-52. APPDIGESTLENGTH Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | LENGTH | R/W | FFFFFFFFh | Application CRC/SHA Digest check source data length |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.6.16 APPDIGEST[y] (Offset = 41C00090h + formula) [Reset = FFFFFFFFh]

APPDIGEST[y] is shown in Figure 1-44 and described in Table 1-53.

Return to the Summary Table.

Expected application CRC32/SHA2-256 digest to test against during boot.
Note: If CRC check is enabled only the first 32bit word of this DIGEST is used.

Offset = 41C00090h + (y * 4h); where y = 0h to 7h

**Figure 1-44. APPDIGEST[y]**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DIGEST | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-FFFFFFFFh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-53. APPDIGEST[y] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DIGEST | R/W | FFFFFFFFh | CRC-32 bit/or SHA-256 bit Digest of the application in Main Flash |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.6.17 FLASHSWP2 (Offset = 41C000B0h) [Reset = FFFFFFFFh]

FLASHSWP2 is shown in Figure 1-45 and described in Table 1-54.

Return to the Summary Table.

Static write protection policy for additional sectors of flash memory. When protected, sectors will not be available for program or erase by either the bootloader or application code

**Figure 1-45. FLASHSWP2**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-FFFFFFFFh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-54. FLASHSWP2 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | R/W | FFFFFFFFh | 1 bit per 8 sectors. This is used to protect Flash memory from (256 to 512 kb). It is not applicable for devices that have Flash memory less than 256KB.<br>0h = A value of 0 indicates the write protection is applied to the corresponding sectors<br>1h = A value of 1 indicates the sector is un-protected |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.6.18 BOOTCRC (Offset = 41C000B4h) [Reset = 00000000h]

BOOTCRC is shown in Figure 1-46 and described in Table 1-55.

Return to the Summary Table.

CRC digest of the BCR configuration portion of the NONMAIN memory.

**Figure 1-46. BOOTCRC**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DIGEST | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-55. BOOTCRC Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DIGEST | R/W | 0h | BCR configuration data CRC digest. 32 bit CRC Digest, if the device supports CRC32-ISO3309. Otherwise 16 bit CRC digest, using polynomial CRC16-CCITT. Configuration to be used for CRC calulation: 1. Polynomial should be as per the chosen standard, 2. Input reflected, 3. Output reflected, 4. Initial value should be 0xFFFFFFFF, 5. Final XOR value should be 0x0. |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.6.19 BSLCONFIGID (Offset = 41C00100h) [Reset = 00000000h]

BSLCONFIGID is shown in Figure 1-47 and described in Table 1-56.

Return to the Summary Table.

Configuration ID of BSL Structure

**Figure 1-47. BSLCONFIGID**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CONFIG | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-56. BSLCONFIGID Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | CONFIG | R/W | 0h | Configuration ID of BSL Structure |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.6.20 BSLPINCFG0 (Offset = 41C00104h) [Reset = 00000000h]

BSLPINCFG0 is shown in Figure 1-48 and described in Table 1-57.

Return to the Summary Table.

BSL UART PIN Configuration. The reset value of these fields differ for each device, based on the pins used.

**Figure 1-48. BSLPINCFG0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| UARTTX_MUX_SEL | | | | | | | | UARTTX_PAD_NUM | | | | | | | |
| R/W-0h | | | | | | | | R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| UARTRX_MUX_SEL | | | | | | | | UARTRX_PAD_NUM | | | | | | | |
| R/W-0h | | | | | | | | R/W-0h | | | | | | | |

**Table 1-57. BSLPINCFG0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | UARTTX_MUX_SEL | R/W | 0h | UART TX IOMUX PINCM mux selection. |
| 23-16 | UARTTX_PAD_NUM | R/W | 0h | UART TX IOMUX PINCM register |
| 15-8 | UARTRX_MUX_SEL | R/W | 0h | UART RX IOMUX PINCM mux selection |
| 7-0 | UARTRX_PAD_NUM | R/W | 0h | UART RX IOMUX PINCM register |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.6.21 BSLPINCFG1 (Offset = 41C00108h) [Reset = 00000000h]

BSLPINCFG1 is shown in Figure 1-49 and described in Table 1-58.

Return to the Summary Table.

BSL I2C PIN Configuration. The reset value of these fields differ for each device, based on the pins used.

**Figure 1-49. BSLPINCFG1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| I2CSCL_MUX_SEL | | | | | | | | I2CSCL_PAD_NUM | | | | | | | |
| R/W-0h | | | | | | | | R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| I2CSDA_MUX_SEL | | | | | | | | I2CSDA_PAD_NUM | | | | | | | |
| R/W-0h | | | | | | | | R/W-0h | | | | | | | |

**Table 1-58. BSLPINCFG1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | I2CSCL_MUX_SEL | R/W | 0h | I2C SCL IOMUX PINCM mux selection |
| 23-16 | I2CSCL_PAD_NUM | R/W | 0h | I2C SCL IOMUX PINCM register |
| 15-8 | I2CSDA_MUX_SEL | R/W | 0h | I2C SDA IOMUX PINCM mux selection |
| 7-0 | I2CSDA_PAD_NUM | R/W | 0h | I2C SDA IOMUX PINCM register |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.6.22 BSLCONFIG0 (Offset = 41C0010Ch) [Reset = 0000XX00h]

BSLCONFIG0 is shown in Figure 1-50 and described in Table 1-59.

Return to the Summary Table.

BSL invoke pin configuration and memory read-out policy.

**Figure 1-50. BSLCONFIG0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| READOUTEN | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| READOUTEN | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | BSLIVK_GPIOPORT | BSLIVK_GPIOPIN | | | | |
| R-0h | | R/W-Xh | R/W-Xh | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BSLIVK_LVL | RESERVED | BSLIVK_PAD_NUM | | | | | |
| R/W-0h | R-0h | R/W-0h | | | | | |

**Table 1-59. BSLCONFIG0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | READOUTEN | R/W | FFFFh | Sets the memory read-out policy for the BSL interface<br>AABBh = Memory contents can be read via the BSL interface<br>FFFFh = Memory read-out is not possible via the BSL interface (Any value other than 0xAABB) |
| 15-14 | RESERVED | R | 0h | |
| 13 | BSLIVK_GPIOPORT | R/W | x | The BSL_invoke GPIO port index corresponding to the pad used for BSL_invoke<br>0h = The BSL_invoke pin is on GPIO port A<br>1h = The BSL_invoke pin is on GPIO port B |
| 12-8 | BSLIVK_GPIOPIN | R/W | x | The BSL_invoke GPIO pin index corresponding to the pad used for BSL_invoke. Valid values are 0 to 31. |
| 7 | BSLIVK_LVL | R/W | 1h | The BSL_invoke input logic level which shall invoke the BSL<br>0h = Pin state LOW will invoke BSL<br>1h = Pin state HIGH will invoke BSL |
| 6 | RESERVED | R | 0h | |
| 5-0 | BSLIVK_PAD_NUM | R/W | 0h | The IOMUX PINCM register corresponding to the pad to be used for BSL_invoke |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.6.23 PWDBSL0 (Offset = 41C00110h) [Reset = 761396AFh]

PWDBSL0 is shown in Figure 1-51 and described in Table 1-60.

Return to the Summary Table.

SHA2-256 Digest of BSL password.

**Figure 1-51. PWDBSL0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DIGEST | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-761396AFh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-60. PWDBSL0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DIGEST | R/W | 761396AFh | SHA2-256 Bit Digest of BSL access Password of size 256 bit. The hash of the default password (all 1's of 256 bit password) is stored here by default when devices are shipped from factory. |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.6.24 PWDBSL1 (Offset = 41C00114h) [Reset = 5F63720Fh]

PWDBSL1 is shown in Figure 1-52 and described in Table 1-61.

Return to the Summary Table.

SHA2-256 Digest of BSL password.

**Figure 1-52. PWDBSL1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DIGEST | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-5F63720Fh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-61. PWDBSL1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DIGEST | R/W | 5F63720Fh | SHA2-256 Bit Digest of BSL access Password of size 256 bit. The hash of the default password (all 1's of 256 bit password) is stored here by default when devices are shipped from factory. |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.6.25 PWDBSL2 (Offset = 41C00118h) [Reset = 5A4AB4BDh]

PWDBSL2 is shown in Figure 1-53 and described in Table 1-62.

Return to the Summary Table.

SHA2-256 Digest of BSL password.

**Figure 1-53. PWDBSL2**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DIGEST | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-5A4AB4BDh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-62. PWDBSL2 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DIGEST | R/W | 5A4AB4BDh | SHA2-256 Bit Digest of BSL access Password of size 256 bit. The hash of the default password (all 1's of 256 bit password) is stored here by default when devices are shipped from factory. |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.6.26 PWDBSL3 (Offset = 41C0011Ch) [Reset = 9FC3630Ah]

PWDBSL3 is shown in Figure 1-54 and described in Table 1-63.

Return to the Summary Table.

SHA2-256 Digest of BSL password.

**Figure 1-54. PWDBSL3**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DIGEST | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-9FC3630Ah | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-63. PWDBSL3 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DIGEST | R/W | 9FC3630Ah | SHA2-256 Bit Digest of BSL access Password of size 256 bit. The hash of the default password (all 1's of 256 bit password) is stored here by default when devices are shipped from factory. |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.6.27 PWDBSL4 (Offset = 41C00120h) [Reset = F930AF12h]

PWDBSL4 is shown in Figure 1-55 and described in Table 1-64.

Return to the Summary Table.

SHA2-256 Digest of BSL password.

**Figure 1-55. PWDBSL4**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DIGEST | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-F930AF12h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-64. PWDBSL4 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DIGEST | R/W | F930AF12h | SHA2-256 Bit Digest of BSL access Password of size 256 bit. The hash of the default password (all 1's of 256 bit password) is stored here by default when devices are shipped from factory. |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.6.28 PWDBSL5 (Offset = 41C00124h) [Reset = 00000000h]

PWDBSL5 is shown in Figure 1-56 and described in Table 1-65.

Return to the Summary Table.

SHA2-256 Digest of BSL password.

**Figure 1-56. PWDBSL5**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DIGEST | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-65. PWDBSL5 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DIGEST | R/W | 5CEEA650h | SHA2-256 Bit Digest of BSL access Password of size 256 bit. The hash of the default password (all 1's of 256 bit password) is stored here by default when devices are shipped from factory. |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.6.29 PWDBSL6 (Offset = 41C00128h) [Reset = 88E11B97h]

PWDBSL6 is shown in Figure 1-57 and described in Table 1-66.

Return to the Summary Table.

SHA2-256 Digest of BSL password.

**Figure 1-57. PWDBSL6**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DIGEST | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-88E11B97h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-66. PWDBSL6 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DIGEST | R/W | 88E11B97h | SHA2-256 Bit Digest of BSL access Password of size 256 bit. The hash of the default password (all 1's of 256 bit password) is stored here by default when devices are shipped from factory. |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.6.30 PWDBSL7 (Offset = 41C0012Ch) [Reset = 51409CE8h]

PWDBSL7 is shown in Figure 1-58 and described in Table 1-67.

Return to the Summary Table.

SHA2-256 Digest of BSL password.

**Figure 1-58. PWDBSL7**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DIGEST | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-51409CE8h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-67. PWDBSL7 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DIGEST | R/W | 51409CE8h | SHA2-256 Bit Digest of BSL access Password of size 256 bit. The hash of the default password (all 1's of 256 bit password) is stored here by default when devices are shipped from factory. |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.6.31 BSLPLUGINCFG (Offset = 41C00130h) [Reset = FFFFFFFFh]

BSLPLUGINCFG is shown in Figure 1-59 and described in Table 1-68.

Return to the Summary Table.

Defines the presence and type of a BSL plug-in in MAIN flash memory.

**Figure 1-59. BSLPLUGINCFG**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SRAMUSED | | | | | | | | FLASHPLUGINEXISTS | | | | | | | |
| R/W-FFh | | | | | | | | R/W-FFh | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PLUGINTYPE | | | | | | | | | | | | | | | |
| R/W-FFFFh | | | | | | | | | | | | | | | |

**Table 1-68. BSLPLUGINCFG Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | SRAMUSED | R/W | FFh | SRAM consumed by Flash plugin, from 0x00 to 0xFF. |
| 23-16 | FLASHPLUGINEXISTS | R/W | FFh | The field tells if Flash Plugin exists are not<br>BBh = Flash Plugin exists<br>FFh = Only ROM plugin will be used |
| 15-0 | PLUGINTYPE | R/W | FFFFh | The Plugin type specifies which interface plugin is added in Flash. When the same interface is supported by ROM Bootloader, then ROM interface will be overridden with the Flash interface. If the same interface as in ROM BSL should be added as an additional interface without overriding ROM interface, a new value that is not part of this list should be chosen.<br>1000h = Plug-in is for UART interface. ROM UART interface will be overriden with this Flash plug-in.<br>2000h = Plug-in is for I2C interface. ROM I2C interface will be overriden with this Flash plug-in.<br>3000h = Plug-is for a new interface. This interface should be added on top of interfaces supported by ROM BSL. It can be any value other than 0x1000, 0x2000 and 0xFFFF.<br>FFFFh = For all other values. Any other interfaces with valid hooks will be added to Plugin list. |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.6.32 BSLPLUGINHOOK[y] (Offset = 41C00134h + formula) [Reset = FFFFFFFFh]

BSLPLUGINHOOK[y] is shown in Figure 1-60 and described in Table 1-69.

Return to the Summary Table.

Function pointers for plug-in init, receive, transmit, and de-init functions.

Offset = 41C00134h + (y * 4h); where y = 0h to 3h

**Figure 1-60. BSLPLUGINHOOK[y]**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BSLPLUGININIT | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-FFFFFFFFh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-69. BSLPLUGINHOOK[y] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 127-96 | BSLPLUGINDEINIT | R/W | FFFFFFFFh | Function pointer for the Deinit API |
| 95-64 | BSLPLUGINSEND | R/W | FFFFFFFFh | Function pointer for the Send API |
| 63-32 | BSLPLUGINRECEIVE | R/W | FFFFFFFFh | Function pointer for the Receive API |
| 31-0 | BSLPLUGININIT | R/W | FFFFFFFFh | Function pointer for the Init API |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.6.33 BSLCONFIG1 (Offset = 41C00144h) [Reset = FFFFFFFFh]

BSLCONFIG1 is shown in Figure 1-61 and described in Table 1-70.

Return to the Summary Table.

**Figure 1-61. BSLCONFIG1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ALTBSLCONFIG | | | | | | | | | | | | | | | | UART_DEFBAUDRATE | | | | | | | | | | | | | | | |
| R/W-FFFFh | | | | | | | | | | | | | | | | R/W-FFFFh | | | | | | | | | | | | | | | |

**Table 1-70. BSLCONFIG1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | ALTBSLCONFIG | R/W | FFFFh | Controls the invocation of alternate BSL in main flash region.<br>AABBh = Uses the alternate BSL<br>FFFFh = Does not use alternate BSL. Uses ROM BSL. |
| 15-0 | UART_DEFBAUDRATE | R/W | FFFFh | Configures the Default Baud rate for the ROM BSL UART interface. Numeric value corresponding to the intended baud rate should be used from the list below. Baud rates are given in units of 'bits per second (bps)'<br>1h = 4800<br>2h = 9600<br>3h = 19200<br>4h = 38400<br>5h = 57600<br>6h = 115200<br>7h = 1000000<br>8h = 2000000 |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.6.34 SBLADDRESS (Offset = 41C00148h) [Reset = FFFFFFFFh]

SBLADDRESS is shown in Figure 1-62 and described in Table 1-71.

Return to the Summary Table.

Address of an alternate BSL.

**Figure 1-62. SBLADDRESS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADDRESS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-FFFFFFFFh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-71. SBLADDRESS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | ADDRESS | R/W | FFFFFFFFh | Address of the alternate BSL, if present. |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.6.35 BSLAPPVER (Offset = 41C0014Ch) [Reset = FFFFFFFFh]

BSLAPPVER is shown in Figure 1-63 and described in Table 1-72.

Return to the Summary Table.

Address of the application version word.

**Figure 1-63. BSLAPPVER**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADDRESS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-FFFFFFFFh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-72. BSLAPPVER Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | ADDRESS | R/W | FFFFFFFFh | Address of the application version word (must be a valid flash address to be returned). If the given flash address is not programmed, then 0h will be returned. |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.6.36 BSLCONFIG2 (Offset = 41C00150h) [Reset = 0048FFFFh]

BSLCONFIG2 is shown in Figure 1-64 and described in Table 1-73.

Return to the Summary Table.

Configures the BSL Alert Configuration and I2C Slave address of BSL.

**Figure 1-64. BSLCONFIG2**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| I2CTARGETADDR | | | | | | | | | | | | | | | | ALERTACTION | | | | | | | | | | | | | | | |
| R/W-48h | | | | | | | | | | | | | | | | R/W-FFFFh | | | | | | | | | | | | | | | |

**Table 1-73. BSLCONFIG2 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | I2CTARGETADDR | R/W | 48h | I2C target address to be used for the ROM BSL I2C communication. |
| 15-0 | ALERTACTION | R/W | FFFFh | Action to take upon a security alert condition.<br>AABBh = Trigger a factory reset. Note that if sectors in MAIN or NONMAIN flash are write protected they will not be affected by the BSL factory reset<br>CCDDh = Re-configure the NONMAIN region to disable the BSL. This is not supported if the NONMAIN region is configured to be write protected<br>000FFFFFh = Ignore the security alert condition (Any value other than 0xAABB) |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.6.37 BSLCONFIG3 (Offset = 41C00154h) [Reset = FFFFFFFFh]

BSLCONFIG3 is shown in Figure 1-65 and described in Table 1-74.

Return to the Summary Table.

Disable NRST policy

**Figure 1-65. BSLCONFIG3**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | DISABLENRST | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | R/W-FFFFh | | | | | | | | | | | | | | | |

**Table 1-74. BSLCONFIG3 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | FFFFh | |
| 15-0 | DISABLENRST | R/W | FFFFh | Disables NRST Pin during the Bootloader execution if configured. Pin will be disabled when device enters the BSL, and gets re-enabled after POR.<br>AABBh = Disables NRST Pin<br>FFFFh = Keeps the NRST pin enabled |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.6.38 BSLCRC (Offset = 41C0015Ch) [Reset = XXXXXXXXh]

BSLCRC is shown in Figure 1-66 and described in Table 1-75.

Return to the Summary Table.

CRC digest of the BSL_CONFIG portion of the NONMAIN memory.

**Figure 1-66. BSLCRC**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DIGEST | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-75. BSLCRC Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DIGEST | R/W | X | BSL configuration data CRC digest. 32 bit CRC Digest, if the device supports CRC32-ISO3309. Otherwise 16 bit CRC digest, using polynomial CRC16-CCITT. Configuration to be used for CRC calulation: 1. Polynomial should be as per the chosen standard, 2. Input reflected, 3. Output reflected, 4. Initial value should be 0xFFFFFFFF, 5. Final XOR value should be 0x0.<br>xxxxxxxx |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.7 NONMAIN_TYPEF Registers

lists the memory-mapped registers for the NONMAIN_TYPEF registers. All register offset addresses not listed in should be considered as reserved locations and the register contents should not be modified.

**Table 1-76. NONMAIN_TYPEF Registers**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 41C00000h | BCRCONFIGID | Configuration ID of BCR Structure | | Go |
| 41C00004h | BOOTCFG0 | Serial wire debug (SWD) lock policy. | | Go |
| 41C00008h | BOOTCFG1 | TI Failure Analysis (FA) mode and BSL invoke pin policies | | Go |
| 41C0000Ch | FLASHSWP0 | Static write protection policy for the first 32kB of flash memory.<br>When protected, sectors will not be available for program or erase by either the bootloader or application code | | Go |
| 41C00010h | FLASHSWP1 | Static write protection policy for additional sectors of flash memory.<br>When protected, sectors will not be available for program or erase by either the bootloader or application code | | Go |
| 41C00014h | BOOTCFG4 | Configures Static write protection policy for Configuration flash memory(NONMAIN) and Debug Hold Policy | | Go |
| 41C00018h | BOOTCFG5 | Configures the existence of Customer Secure Code(CSC) and Flash Bank Swap Policy | | Go |
| 41C0001Ch | BOOTCFG2 | Fast boot mode policy and BSL mode policy | | Go |
| 41C00020h | BOOTCFG3 | Mass erase and factory reset mode policies.<br>These policies affect SWD initiated and BSL initiated mass erase and factory reset commands.<br>If the SW-DP is disabled (SWDP_MODE is disabled), SWD initiated commands are not allowed as the SW-DP is fully disabled.<br>If the BSL is disabled (BSLMODE is disabled), these settings are a a don't care for BSL initiated commands as the BSL is not enabled to be invoked. | | Go |
| 41C00024h + formula | PWDMASSERASE[y] | SHA2-256 Digest of the SWD mass erase command password (must be provided via DSSM to authenticate a mass erase command) | | Go |
| 41C00044h + formula | PWDFACTORYRESET[y] | SHA2-256 digest of the SWD factory reset command password (must be provided via DSSM to authenticate a factory reset command) | | Go |
| 41C00064h + formula | PWDDEBUGLOCK[y] | SHA2-256 Digest of the Debug Lock Password.<br>This password is used when the BOOTCFG0.DEBUGACCESS field is configured as Enable with password. | | Go |
| 41C00084h | BOOTCFG6 | Application digest check policy | | Go |



Copyright © 2026 Texas Instruments Incorporated

**Table 1-76. NONMAIN_TYPEF Registers (continued)**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 41C00088h | APPDIGESTSTART | Start address of the application CRC/SHA Digest check (must be an address in a MAIN flash region). | | Go |
| 41C0008Ch | APPDIGESTLENGTH | Length of the application to be include in the application CRC/SHA Digest check (in bytes), starting from APPDIGESTSTART | | Go |
| 41C00090h + formula | APPDIGEST[y] | Expected application CRC32/SHA2-256 digest to test against during boot. Note: If CRC check is enabled only the first 32bit word of this DIGEST is used. | | Go |
| 41C000B0h | FLASHSWP2 | Static write protection policy for additional sectors of flash memory. When protected, sectors will not be available for program or erase by either the bootloader or application code | | Go |
| 41C000B4h | BOOTCRC | CRC digest of the BCR configuration portion of the NONMAIN memory. | | Go |
| 41C00100h | BSLCONFIGID | Configuration ID of BSL Structure | | Go |
| 41C00104h | BSLPINCFG0 | BSL UART PIN Configuration. The reset value of these fields differ for each device, based on the pins used. | | Go |
| 41C00108h | BSLPINCFG1 | BSL I2C PIN Configuration. The reset value of these fields differ for each device, based on the pins used. | | Go |
| 41C0010Ch | BSLCONFIG0 | BSL invoke pin configuration and memory read-out policy. | | Go |
| 41C00110h | PWDBSL0 | SHA2-256 Digest of BSL password. | | Go |
| 41C00114h | PWDBSL1 | SHA2-256 Digest of BSL password. | | Go |
| 41C00118h | PWDBSL2 | SHA2-256 Digest of BSL password. | | Go |
| 41C0011Ch | PWDBSL3 | SHA2-256 Digest of BSL password. | | Go |
| 41C00120h | PWDBSL4 | SHA2-256 Digest of BSL password. | | Go |
| 41C00124h | PWDBSL5 | SHA2-256 Digest of BSL password. | | Go |
| 41C00128h | PWDBSL6 | SHA2-256 Digest of BSL password. | | Go |
| 41C0012Ch | PWDBSL7 | SHA2-256 Digest of BSL password. | | Go |
| 41C00130h | BSLPLUGINCFG | Defines the presence and type of a BSL plug-in in MAIN flash memory. | | Go |
| 41C00134h + formula | BSLPLUGINHOOK[y] | Function pointers for plug-in init, receive, transmit, and de-init functions. | | Go |
| 41C00144h | BSLCONFIG1 | | | Go |
| 41C00148h | SBLADDRESS | Address of an alternate BSL. | | Go |
| 41C0014Ch | BSLAPPVER | Address of the application version word. | | Go |



Copyright © 2026 Texas Instruments Incorporated

**Table 1-76. NONMAIN_TYPEF Registers (continued)**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 41C00150h | BSLCONFIG2 | Configures the BSL Alert Configuration and I2C Slave address of BSL. | | Go |
| 41C00154h | BSLCRC | CRC digest of the BSL_CONFIG portion of the NONMAIN memory. | | Go |

Complex bit access types are encoded to fit into small table cells. shows the codes that are used for access types in this section.

**Table 1-77. NONMAIN_TYPEF Access Type Codes**

| Access Type | Code | Description |
|---|---|---|
| Read Type | | |
| R | R | Read |
| Write Type | | |
| W | W | Write |
| Reset or Default Value | | |
| -*n* | | Value after reset or the default value |
| Register Array Variables | | |
| i,j,k,l,m,n | | When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula. |
| y | | When this variable is used in a register name, an offset, or an address it refers to the value of a register array. |



Copyright © 2026 Texas Instruments Incorporated

#### 1.4.7.1 BCRCONFIGID (Offset = 41C00000h) [Reset = 00000000h]

BCRCONFIGID is shown in and described in .

Return to the Summary Table.

Configuration ID of BCR Structure

**Figure 1-67. BCRCONFIGID**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CONFIG | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-78. BCRCONFIGID Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | CONFIG | R/W | 0h | Configuration ID of the BCR Structure |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.7.2 BOOTCFG0 (Offset = 41C00004h) [Reset = AABBAABBh]

BOOTCFG0 is shown in and described in .

Return to the Summary Table.

Serial wire debug (SWD) lock policy.

**Figure 1-68. BOOTCFG0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SWDP_MODE | | | | | | | | | | | | | | | | DEBUGACCESS | | | | | | | | | | | | | | | |
| R/W-AABBh | | | | | | | | | | | | | | | | W-AABBh | | | | | | | | | | | | | | | |

**Table 1-79. BOOTCFG0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | SWDP_MODE | R/W | AABBh | The serial wire debug port (SW-DP) access policy. This policy sets whether any communication is allowed with the device via the SWD pins (to any DAP). When disabled, no SWD communication is possible regardless of the configuration of the DEBUGACCESS field.<br>AABBh = The SW-DP is enabled and device access is set by the additional policies in NONMAIN<br>FFFFh = The SW-DP is fully disabled and no device access is possible via the SW-DP (Any value other than 0xAABB) |
| 15-0 | DEBUGACCESS | W | AABBh | The debug access policy for accessing the AHB-AP, ET-AP, and PWR-AP debug access ports. Note that if SWDP_MODE is set to DISABLED, the value of this field is ignored and the debug port will remain fully locked<br>AABBh = Access to AHB-AP, ET-AP, and PWR-AP via SWD is enabled<br>CCDDh = Access to AHB-AP, ET-AP, and PWR-AP via SWD is only enabled when the correct password is provided via the DSSM before BCR execution<br>FFFFh = Access to AHB-AP, ET-AP, and PWR-AP via SWD is disabled (Any value other than 0xCCDD or 0xAABB) |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.7.3 BOOTCFG1 (Offset = 41C00008h) [Reset = AABBAABBh]

BOOTCFG1 is shown in and described in .

Return to the Summary Table.

TI Failure Analysis (FA) mode and BSL invoke pin policies

**Figure 1-69. BOOTCFG1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BSL_PIN_INVOKE | | | | | | | | | | | | | | | | TI_FA_MODE | | | | | | | | | | | | | | | |
| R/W-AABBh | | | | | | | | | | | | | | | | R/W-AABBh | | | | | | | | | | | | | | | |

**Table 1-80. BOOTCFG1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | BSL_PIN_INVOKE | R/W | AABBh | Boot strap loader (BSL) pin invoke method enable/disable policy<br>AABBh = The BSL_INVOKE pin is checked during boot<br>FFFFh = The BSL_INVOKE pin is not checked during boot (Any value other than 0xAABB) |
| 15-0 | TI_FA_MODE | R/W | AABBh | Sets the TI failure analysis enable/disable policy. If enabled, a re-test request through DSSM is allowed, else it is not allowed. Note that if SWDP_MODE is set to disabled, this field is ignored and failure analysis is not possible.<br>AABBh = TI failure analysis is allowed<br>FFFFh = TI failure analysis is not allowed (Any value other than 0xAABB) |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.7.4 FLASHSWP0 (Offset = 41C0000Ch) [Reset = FFFFFFFFh]

FLASHSWP0 is shown in and described in .

Return to the Summary Table.

Static write protection policy for the first 32kB of flash memory. When protected, sectors will not be available for program or erase by either the bootloader or application code

**Figure 1-70. FLASHSWP0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-FFFFFFFFh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-81. FLASHSWP0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | R/W | FFFFFFFFh | 1 bit per sector<br>0h = A value of 0 indicates the write protection is applied to the corresponding sector<br>1h = A value of 1 indicates the sector is un-protected |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.7.5 FLASHSWP1 (Offset = 41C00010h) [Reset = FFFFFFFFh]

FLASHSWP1 is shown in and described in .

Return to the Summary Table.

Static write protection policy for additional sectors of flash memory. When protected, sectors will not be available for program or erase by either the bootloader or application code

**Figure 1-71. FLASHSWP1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-FFFFFFFFh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-82. FLASHSWP1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | R/W | FFFFFFFFh | 1 bit per 8 sectors. This is used to protect Flash memory from (0 to 256 kb)<br>0h = A value of 0 indicates the write protection is applied to the corresponding sectors<br>1h = A value of 1 indicates the sector is un-protected |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.7.6 BOOTCFG4 (Offset = 41C00014h) [Reset = AABBAABBh]

BOOTCFG4 is shown in and described in .

Return to the Summary Table.

Configures Static write protection policy for Configuration flash memory(NONMAIN) and Debug Hold Policy

**Figure 1-72. BOOTCFG4**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DEBUGHOLD | | | | | | | | | | | | | | | | NONMAINSWP | | | | | | | | | | | | | | | |
| R/W-AABBh | | | | | | | | | | | | | | | | R/W-AABBh | | | | | | | | | | | | | | | |

**Table 1-83. BOOTCFG4 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | DEBUGHOLD | R/W | AABBh | Controls the release of debug access until INITDONE is issued by Customer Secure Code(CSC). This configuration is applicable only when DEBUGACCESS and CSCEXISTS is enabled.<br>AABBh = Debug Hold is disabled. Debug access should be available during CSC execution.<br>FFFFh = Debug Hold is enabled. Debug access will not be available during CSC execution. Access will be available for application debug after CSC INITDONE. (Any value other than 0xAABB) |
| 15-0 | NONMAINSWP | R/W | AABBh | Static write protection policy for entire NONMAIN device configuration memory.<br>AABBh = Protection is disabled. Allows program/erase of the NONMAIN by normal means.<br>FFFFh = Protection is enabled. Does not allow program/erase of the NONMAIN by all means other than a SWD-initiated factory reset (Any value other than 0xAABB) |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.7.7 BOOTCFG5 (Offset = 41C00018h) [Reset = AABBAABBh]

BOOTCFG5 is shown in and described in .

Return to the Summary Table.

Configures the existence of Customer Secure Code(CSC) and Flash Bank Swap Policy

**Figure 1-73. BOOTCFG5**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FLASHBANKSWAPPOLICY | | | | | | | | | | | | | | | |
| R/W-AABBh | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CSCEXISTS | | | | | | | | | | | | | | | |
| R/W-AABBh | | | | | | | | | | | | | | | |

**Table 1-84. BOOTCFG5 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | FLASHBANKSWAPPOLICY | R/W | AABBh | Controls the Flash Bank Swap Policy. This configuration is valid only, if CSCEXISTS is enabled.<br>AABBh = Flash Bank swap policy is disabled. In CSC Banks cannot be swapped. Complete Main flash region has Read, Write, Execute access.<br>FFFFh = Flash Bank swap policy is enabled. In CSC Banks can be swapped. One bank wil have Read, Execute access and other Bank will have Read/Write access, as per the CSC configuration. |
| 15-0 | CSCEXISTS | R/W | AABBh | Controls the existence of Customer Secure Code(CSC).<br>AABBh = CSC does not exist. INITDONE will be already SET.<br>FFFFh = CSC Exists. INITDONE will be SET by the CSC. |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.7.8 BOOTCFG2 (Offset = 41C0001Ch) [Reset = AABBFFFFh]

BOOTCFG2 is shown in and described in .

Return to the Summary Table.

Fast boot mode policy and BSL mode policy

**Figure 1-74. BOOTCFG2**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BSLMODE | | | | | | | | | | | | | | | | FASTBOOTMODE | | | | | | | | | | | | | | | |
| R/W-AABBh | | | | | | | | | | | | | | | | R/W-FFFFh | | | | | | | | | | | | | | | |

**Table 1-85. BOOTCFG2 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | BSLMODE | R/W | AABBh | BSLMODE configures the boot strap loader enable/disable policy<br>AABBh = The BSL is enabled<br>FFFFh = The BSL is disabled (Any value other than 0xAABB) |
| 15-0 | FASTBOOTMODE | R/W | FFFFh | FASTBOOTMODE configures the fast boot mode enable/disable policy<br>AABBh = Fast boot mode is enabled<br>FFFFh = Fast boot mode is disabled. (Any value other than 0xAABB) |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.7.9 BOOTCFG3 (Offset = 41C00020h) [Reset = AABBAABBh]

BOOTCFG3 is shown in and described in .

Return to the Summary Table.

Mass erase and factory reset mode policies. These policies affect SWD initiated and BSL initiated mass erase and factory reset commands. If the SW-DP is disabled (SWDP_MODE is disabled), SWD initiated commands are not allowed as the SW-DP is fully disabled. If the BSL is disabled (BSLMODE is disabled), these settings are a a don't care for BSL initiated commands as the BSL is not enabled to be invoked.

**Figure 1-75. BOOTCFG3**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FACTORYRESETCMDACCESS | | | | | | | | | | | | | | | |
| R/W-AABBh | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MASSERASECMDACCESS | | | | | | | | | | | | | | | |
| R/W-AABBh | | | | | | | | | | | | | | | |

**Table 1-86. BOOTCFG3 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | FACTORYRESETCMDACCESS | R/W | AABBh | The factory reset command policy<br>AABBh = The factory reset command is allowed<br>CCDDh = The factory reset command is allowed only when the matching password is provided via the DSSM<br>FFFFh = The factory reset command is not allowed (Any value other than 0xAABB or 0xCCDD) |
| 15-0 | MASSERASECMDACCESS | R/W | AABBh | The mass erase command policy<br>AABBh = The mass erase command is allowed<br>CCDDh = The mass erase command is allowed only when the matching password is provided via the DSSM<br>FFFFh = The mass erase command is not allowed (Any value other than 0xAABB or 0xCCDD) |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.7.10 PWDMASSERASE[y] (Offset = 41C00024h + formula) [Reset = FFFFFFFFh]

PWDMASSERASE[y] is shown in and described in .

Return to the Summary Table.

SHA2-256 Digest of the SWD mass erase command password (must be provided via DSSM to authenticate a mass erase command)

Offset = 41C00024h + (y * 4h); where y = 0h to 7h

**Figure 1-76. PWDMASSERASE[y]**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DIGEST | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-FFFFFFFFh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-87. PWDMASSERASE[y] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DIGEST | R/W | FFFFFFFFh | SHA2-256 Bit digest of Mass Erase Password of size 128 bit |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.7.11 PWDFACTORYRESET[y] (Offset = 41C00044h + formula) [Reset = FFFFFFFFh]

PWDFACTORYRESET[y] is shown in and described in .

Return to the Summary Table.

SHA2-256 digest of the SWD factory reset command password (must be provided via DSSM to authenticate a factory reset command)

Offset = 41C00044h + (y * 4h); where y = 0h to 7h

**Figure 1-77. PWDFACTORYRESET[y]**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DIGEST | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-FFFFFFFFh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-88. PWDFACTORYRESET[y] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DIGEST | R/W | FFFFFFFFh | SHA2-256 Bit digest of Factory Reset Password of size 128 bit |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.7.12 PWDDEBUGLOCK[y] (Offset = 41C00064h + formula) [Reset = FFFFFFFFh]

PWDDEBUGLOCK[y] is shown in and described in .

Return to the Summary Table.

SHA2-256 Digest of the Debug Lock Password. This password is used when the BOOTCFG0.DEBUGACCESS field is configured as Enable with password.

Offset = 41C00064h + (y * 4h); where y = 0h to 7h

**Figure 1-78. PWDDEBUGLOCK[y]**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DIGEST | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-FFFFFFFFh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-89. PWDDEBUGLOCK[y] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DIGEST | R/W | FFFFFFFFh | SHA2-256 Bit digest of Debug Lock Password of size 128 bit |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.7.13 BOOTCFG6 (Offset = 41C00084h) [Reset = FFFFFFFFh]

BOOTCFG6 is shown in and described in .

Return to the Summary Table.

Application digest check policy

**Figure 1-79. BOOTCFG6**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| APPDIGESTMODE | | | | | | | | | | | | | | | | RESERVED | | | | | | | | | | | | | | | |
| R/W-FFFFh | | | | | | | | | | | | | | | | R-0h | | | | | | | | | | | | | | | |

**Table 1-90. BOOTCFG6 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | APPDIGESTMODE | R/W | FFFFh | APPDIGESTMODE configures the boot time CRC/SHA Digest check of a segment of MAIN flash memory.<br>AABBh = The boot time MAIN flash CRC check is enabled. If the boot time CRC check passes, the application code in MAIN flash is started unless the reset vector or stack pointer are blank (unprogrammed). In the event of a failing CRC check, the application code in MAIN flash will not be started and the boot process fails.<br>CCDDh = The boot time MAIN flash SHA2-256 Hash check is enabled. If the boot time Hash check passes, the application code in MAIN flash is started unless the reset vector or stack pointer are blank (unprogrammed). In the event of a failing hash check, the application code in MAIN flash will not be started and the boot process fails.<br>FFFFh = The boot time MAIN flash Digest check is disabled. The application code in MAIN flash is always started unless the reset vector or stack pointer are blank(unprogrammed ) |
| 15-0 | RESERVED | R | FFFFh | |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.7.14 APPDIGESTSTART (Offset = 41C00088h) [Reset = FFFFFFFFh]

APPDIGESTSTART is shown in and described in .

Return to the Summary Table.

Start address of the application CRC/SHA Digest check (must be an address in a MAIN flash region).

**Figure 1-80. APPDIGESTSTART**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADDRESS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-FFFFFFFFh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-91. APPDIGESTSTART Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | ADDRESS | R/W | FFFFFFFFh | Application CRC/SHA Digest check start address |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.7.15 APPDIGESTLENGTH (Offset = 41C0008Ch) [Reset = FFFFFFFFh]

APPDIGESTLENGTH is shown in and described in .

Return to the Summary Table.

Length of the application to be include in the application CRC/SHA Digest check (in bytes), starting from APPDIGESTSTART

**Figure 1-81. APPDIGESTLENGTH**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LENGTH | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-FFFFFFFFh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-92. APPDIGESTLENGTH Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | LENGTH | R/W | FFFFFFFFh | Application CRC/SHA Digest check source data length |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.7.16 APPDIGEST[y] (Offset = 41C00090h + formula) [Reset = FFFFFFFFh]

APPDIGEST[y] is shown in and described in .

Return to the Summary Table.

Expected application CRC32/SHA2-256 digest to test against during boot.
Note: If CRC check is enabled only the first 32bit word of this DIGEST is used.

Offset = 41C00090h + (y * 4h); where y = 0h to 7h

**Figure 1-82. APPDIGEST[y]**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DIGEST | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-FFFFFFFFh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-93. APPDIGEST[y] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DIGEST | R/W | FFFFFFFFh | CRC-32 bit/or SHA-256 bit Digest of the application in Main Flash |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.7.17 FLASHSWP2 (Offset = 41C000B0h) [Reset = FFFFFFFFh]

FLASHSWP2 is shown in and described in .

Return to the Summary Table.

Static write protection policy for additional sectors of flash memory. When protected, sectors will not be available for program or erase by either the bootloader or application code

**Figure 1-83. FLASHSWP2**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-FFFFFFFFh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-94. FLASHSWP2 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | R/W | FFFFFFFFh | 1 bit per 8 sectors. This is used to protect Flash memory from (256 to 512 kb). It is not applicable for devices that have Flash memory less than 256KB.<br>0h = A value of 0 indicates the write protection is applied to the corresponding sectors<br>1h = A value of 1 indicates the sector is un-protected |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.7.18 BOOTCRC (Offset = 41C000B4h) [Reset = 00000000h]

BOOTCRC is shown in and described in .

Return to the Summary Table.

CRC digest of the BCR configuration portion of the NONMAIN memory.

**Figure 1-84. BOOTCRC**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DIGEST | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-95. BOOTCRC Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DIGEST | R/W | 0h | BCR configuration data CRC digest. 32 bit CRC Digest, if the device supports CRC32-ISO3309. Otherwise 16 bit CRC digest, using polynomial CRC16-CCITT. Configuration to be used for CRC calulation: 1. Polynomial should be as per the chosen standard, 2. Input reflected, 3. Output reflected, 4. Initial value should be 0xFFFFFFFF, 5. Final XOR value should be 0x0. |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.7.19 BSLCONFIGID (Offset = 41C00100h) [Reset = 00000000h]

BSLCONFIGID is shown in and described in .

Return to the Summary Table.

Configuration ID of BSL Structure

**Figure 1-85. BSLCONFIGID**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CONFIG | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-96. BSLCONFIGID Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | CONFIG | R/W | 0h | Configuration ID of BSL Structure |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.7.20 BSLPINCFG0 (Offset = 41C00104h) [Reset = 00000000h]

BSLPINCFG0 is shown in and described in .

Return to the Summary Table.

BSL UART PIN Configuration. The reset value of these fields differ for each device, based on the pins used.

**Figure 1-86. BSLPINCFG0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| UARTTX_MUX_SEL | | | | | | | | UARTTX_PAD_NUM | | | | | | | |
| R/W-0h | | | | | | | | R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| UARTRX_MUX_SEL | | | | | | | | UARTRX_PAD_NUM | | | | | | | |
| R/W-0h | | | | | | | | R/W-0h | | | | | | | |

**Table 1-97. BSLPINCFG0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | UARTTX_MUX_SEL | R/W | 0h | UART TX IOMUX PINCM mux selection. |
| 23-16 | UARTTX_PAD_NUM | R/W | 0h | UART TX IOMUX PINCM register |
| 15-8 | UARTRX_MUX_SEL | R/W | 0h | UART RX IOMUX PINCM mux selection |
| 7-0 | UARTRX_PAD_NUM | R/W | 0h | UART RX IOMUX PINCM register |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.7.21 BSLPINCFG1 (Offset = 41C00108h) [Reset = 00000000h]

BSLPINCFG1 is shown in and described in .

Return to the Summary Table.

BSL I2C PIN Configuration. The reset value of these fields differ for each device, based on the pins used.

**Figure 1-87. BSLPINCFG1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| I2CSCL_MUX_SEL | | | | | | | | I2CSCL_PAD_NUM | | | | | | | |
| R/W-0h | | | | | | | | R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| I2CSDA_MUX_SEL | | | | | | | | I2CSDA_PAD_NUM | | | | | | | |
| R/W-0h | | | | | | | | R/W-0h | | | | | | | |

**Table 1-98. BSLPINCFG1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | I2CSCL_MUX_SEL | R/W | 0h | I2C SCL IOMUX PINCM mux selection |
| 23-16 | I2CSCL_PAD_NUM | R/W | 0h | I2C SCL IOMUX PINCM register |
| 15-8 | I2CSDA_MUX_SEL | R/W | 0h | I2C SDA IOMUX PINCM mux selection |
| 7-0 | I2CSDA_PAD_NUM | R/W | 0h | I2C SDA IOMUX PINCM register |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.7.22 BSLCONFIG0 (Offset = 41C0010Ch) [Reset = 0000XX00h]

BSLCONFIG0 is shown in and described in .

Return to the Summary Table.

BSL invoke pin configuration and memory read-out policy.

**Figure 1-88. BSLCONFIG0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| READOUTEN | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| READOUTEN | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | BSLIVK_GPIOPORT | BSLIVK_GPIOPIN | | | | |
| R-0h | | R/W-Xh | R/W-Xh | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BSLIVK_LVL | RESERVED | BSLIVK_PAD_NUM | | | | | |
| R/W-0h | R-0h | R/W-0h | | | | | |

**Table 1-99. BSLCONFIG0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | READOUTEN | R/W | FFFFh | Sets the memory read-out policy for the BSL interface<br>AABBh = Memory contents can be read via the BSL interface<br>FFFFh = Memory read-out is not possible via the BSL interface (Any value other than 0xAABB) |
| 15-14 | RESERVED | R | 0h | |
| 13 | BSLIVK_GPIOPORT | R/W | x | The BSL_invoke GPIO port index corresponding to the pad used for BSL_invoke<br>0h = The BSL_invoke pin is on GPIO port A<br>1h = The BSL_invoke pin is on GPIO port B |
| 12-8 | BSLIVK_GPIOPIN | R/W | x | The BSL_invoke GPIO pin index corresponding to the pad used for BSL_invoke. Valid values are 0 to 31. |
| 7 | BSLIVK_LVL | R/W | 1h | The BSL_invoke input logic level which shall invoke the BSL<br>0h = Pin state LOW will invoke BSL<br>1h = Pin state HIGH will invoke BSL |
| 6 | RESERVED | R | 0h | |
| 5-0 | BSLIVK_PAD_NUM | R/W | 0h | The IOMUX PINCM register corresponding to the pad to be used for BSL_invoke |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.7.23 PWDBSL0 (Offset = 41C00110h) [Reset = 761396AFh]

PWDBSL0 is shown in and described in .

Return to the Summary Table.

SHA2-256 Digest of BSL password.

**Figure 1-89. PWDBSL0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DIGEST | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-761396AFh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-100. PWDBSL0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DIGEST | R/W | 761396AFh | SHA2-256 Bit Digest of BSL access Password of size 256 bit. The hash of the default password (all 1's of 256 bit password) is stored here by default when devices are shipped from factory. |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.7.24 PWDBSL1 (Offset = 41C00114h) [Reset = 5F63720Fh]

PWDBSL1 is shown in and described in .

Return to the Summary Table.

SHA2-256 Digest of BSL password.

**Figure 1-90. PWDBSL1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DIGEST | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-5F63720Fh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-101. PWDBSL1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DIGEST | R/W | 5F63720Fh | SHA2-256 Bit Digest of BSL access Password of size 256 bit. The hash of the default password (all 1's of 256 bit password) is stored here by default when devices are shipped from factory. |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.7.25 PWDBSL2 (Offset = 41C00118h) [Reset = 5A4AB4BDh]

PWDBSL2 is shown in and described in .

Return to the Summary Table.

SHA2-256 Digest of BSL password.

**Figure 1-91. PWDBSL2**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DIGEST | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-5A4AB4BDh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-102. PWDBSL2 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DIGEST | R/W | 5A4AB4BDh | SHA2-256 Bit Digest of BSL access Password of size 256 bit. The hash of the default password (all 1's of 256 bit password) is stored here by default when devices are shipped from factory. |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.7.26 PWDBSL3 (Offset = 41C0011Ch) [Reset = 9FC3630Ah]

PWDBSL3 is shown in and described in .

Return to the Summary Table.

SHA2-256 Digest of BSL password.

**Figure 1-92. PWDBSL3**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DIGEST | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-9FC3630Ah | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-103. PWDBSL3 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DIGEST | R/W | 9FC3630Ah | SHA2-256 Bit Digest of BSL access Password of size 256 bit. The hash of the default password (all 1's of 256 bit password) is stored here by default when devices are shipped from factory. |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.7.27 PWDBSL4 (Offset = 41C00120h) [Reset = F930AF12h]

PWDBSL4 is shown in and described in .

Return to the Summary Table.

SHA2-256 Digest of BSL password.

**Figure 1-93. PWDBSL4**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DIGEST | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-F930AF12h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-104. PWDBSL4 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DIGEST | R/W | F930AF12h | SHA2-256 Bit Digest of BSL access Password of size 256 bit. The hash of the default password (all 1's of 256 bit password) is stored here by default when devices are shipped from factory. |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.7.28 PWDBSL5 (Offset = 41C00124h) [Reset = 00000000h]

PWDBSL5 is shown in and described in .

Return to the Summary Table.

SHA2-256 Digest of BSL password.

**Figure 1-94. PWDBSL5**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DIGEST | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-105. PWDBSL5 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DIGEST | R/W | 5CEEA650h | SHA2-256 Bit Digest of BSL access Password of size 256 bit. The hash of the default password (all 1's of 256 bit password) is stored here by default when devices are shipped from factory. |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.7.29 PWDBSL6 (Offset = 41C00128h) [Reset = 88E11B97h]

PWDBSL6 is shown in and described in .

Return to the Summary Table.

SHA2-256 Digest of BSL password.

**Figure 1-95. PWDBSL6**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DIGEST | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-88E11B97h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-106. PWDBSL6 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DIGEST | R/W | 88E11B97h | SHA2-256 Bit Digest of BSL access Password of size 256 bit. The hash of the default password (all 1's of 256 bit password) is stored here by default when devices are shipped from factory. |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.7.30 PWDBSL7 (Offset = 41C0012Ch) [Reset = 51409CE8h]

PWDBSL7 is shown in and described in .

Return to the Summary Table.

SHA2-256 Digest of BSL password.

**Figure 1-96. PWDBSL7**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DIGEST | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-51409CE8h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-107. PWDBSL7 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DIGEST | R/W | 51409CE8h | SHA2-256 Bit Digest of BSL access Password of size 256 bit. The hash of the default password (all 1's of 256 bit password) is stored here by default when devices are shipped from factory. |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.7.31 BSLPLUGINCFG (Offset = 41C00130h) [Reset = FFFFFFFFh]

BSLPLUGINCFG is shown in and described in .

Return to the Summary Table.

Defines the presence and type of a BSL plug-in in MAIN flash memory.

**Figure 1-97. BSLPLUGINCFG**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SRAMUSED | | | | | | | | FLASHPLUGINEXISTS | | | | | | | |
| R/W-FFh | | | | | | | | R/W-FFh | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PLUGINTYPE | | | | | | | | | | | | | | | |
| R/W-FFFFh | | | | | | | | | | | | | | | |

**Table 1-108. BSLPLUGINCFG Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | SRAMUSED | R/W | FFh | SRAM consumed by Flash plugin, from 0x00 to 0xFF. |
| 23-16 | FLASHPLUGINEXISTS | R/W | FFh | The field tells if Flash Plugin exists are not<br>BBh = Flash Plugin exists<br>FFh = Only ROM plugin will be used |
| 15-0 | PLUGINTYPE | R/W | FFFFh | The Plugin type specifies which interface plugin is added in Flash. When the same interface is supported by ROM Bootloader, then ROM interface will be overridden with the Flash interface. If the same interface as in ROM BSL should be added as an additional interface without overriding ROM interface, a new value that is not part of this list should be chosen.<br>1000h = Plug-in is for UART interface. ROM UART interface will be overriden with this Flash plug-in.<br>2000h = Plug-in is for I2C interface. ROM I2C interface will be overriden with this Flash plug-in.<br>3000h = Plug-is for a new interface. This interface should be added on top of interfaces supported by ROM BSL. It can be any value other than 0x1000, 0x2000 and 0xFFFF.<br>FFFFh = For all other values. Any other interfaces with valid hooks will be added to Plugin list. |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.7.32 BSLPLUGINHOOK[y] (Offset = 41C00134h + formula) [Reset = FFFFFFFFh]

BSLPLUGINHOOK[y] is shown in and described in .

Return to the Summary Table.

Function pointers for plug-in init, receive, transmit, and de-init functions.

Offset = 41C00134h + (y * 4h); where y = 0h to 3h

**Figure 1-98. BSLPLUGINHOOK[y]**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BSLPLUGININIT | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-FFFFFFFFh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-109. BSLPLUGINHOOK[y] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 127-96 | BSLPLUGINDEINIT | R/W | FFFFFFFFh | Function pointer for the Deinit API |
| 95-64 | BSLPLUGINSEND | R/W | FFFFFFFFh | Function pointer for the Send API |
| 63-32 | BSLPLUGINRECEIVE | R/W | FFFFFFFFh | Function pointer for the Receive API |
| 31-0 | BSLPLUGININIT | R/W | FFFFFFFFh | Function pointer for the Init API |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.7.33 BSLCONFIG1 (Offset = 41C00144h) [Reset = FFFFFFFFh]

BSLCONFIG1 is shown in and described in .

Return to the Summary Table.

**Figure 1-99. BSLCONFIG1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ALTBSLCONFIG | | | | | | | | | | | | | | | | UART_DEFBAUDRATE | | | | | | | | | | | | | | | |
| R/W-FFFFh | | | | | | | | | | | | | | | | R/W-FFFFh | | | | | | | | | | | | | | | |

**Table 1-110. BSLCONFIG1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | ALTBSLCONFIG | R/W | FFFFh | Controls the invocation of alternate BSL in main flash region.<br>AABBh = Uses the alternate BSL<br>FFFFh = Does not use alternate BSL. Uses ROM BSL. |
| 15-0 | UART_DEFBAUDRATE | R/W | FFFFh | Configures the Default Baud rate for the ROM BSL UART interface. Numeric value corresponding to the intended baud rate should be used from the list below. Baud rates are given in units of 'bits per second (bps)'<br>1h = 4800<br>2h = 9600<br>3h = 19200<br>4h = 38400<br>5h = 57600<br>6h = 115200<br>7h = 1000000<br>8h = 2000000 |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.7.34 SBLADDRESS (Offset = 41C00148h) [Reset = FFFFFFFFh]

SBLADDRESS is shown in and described in .

Return to the Summary Table.

Address of an alternate BSL.

**Figure 1-100. SBLADDRESS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADDRESS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-FFFFFFFFh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-111. SBLADDRESS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | ADDRESS | R/W | FFFFFFFFh | Address of the alternate BSL, if present. |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.7.35 BSLAPPVER (Offset = 41C0014Ch) [Reset = FFFFFFFFh]

BSLAPPVER is shown in and described in .

Return to the Summary Table.

Address of the application version word.

**Figure 1-101. BSLAPPVER**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADDRESS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-FFFFFFFFh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-112. BSLAPPVER Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | ADDRESS | R/W | FFFFFFFFh | Address of the application version word (must be a valid flash address to be returned). If the given flash address is not programmed, then 0h will be returned. |



Copyright © 2026 Texas Instruments Incorporated

### 1.4.7.36 BSLCONFIG2 (Offset = 41C00150h) [Reset = 0048FFFFh]

BSLCONFIG2 is shown in and described in .

Return to the Summary Table.

Configures the BSL Alert Configuration and I2C Slave address of BSL.

**Figure 1-102. BSLCONFIG2**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| I2CTARGETADDR | | | | | | | | | | | | | | | | ALERTACTION | | | | | | | | | | | | | | | |
| R/W-48h | | | | | | | | | | | | | | | | R/W-FFFFh | | | | | | | | | | | | | | | |

**Table 1-113. BSLCONFIG2 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | I2CTARGETADDR | R/W | 48h | I2C target address to be used for the ROM BSL I2C communication. |
| 15-0 | ALERTACTION | R/W | FFFFh | Action to take upon a security alert condition.<br>AABBh = Trigger a factory reset. Note that if sectors in MAIN or NONMAIN flash are write protected they will not be affected by the BSL factory reset<br>CCDDh = Re-configure the NONMAIN region to disable the BSL. This is not supported if the NONMAIN region is configured to be write protected<br>000FFFFFh = Ignore the security alert condition (Any value other than 0xAABB) |



Copyright © 2026 Texas Instruments Incorporated

#### 1.4.7.37 BSLCRC (Offset = 41C00154h) [Reset = XXXXXXXXh]

BSLCRC is shown in and described in .

Return to the Summary Table.

CRC digest of the BSL_CONFIG portion of the NONMAIN memory.

**Figure 1-103. BSLCRC**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DIGEST | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-114. BSLCRC Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DIGEST | R/W | X | BSL configuration data CRC digest. 32 bit CRC Digest, if the device supports CRC32-ISO3309. Otherwise 16 bit CRC digest, using polynomial CRC16-CCITT. Configuration to be used for CRC calulation: 1. Polynomial should be as per the chosen standard, 2. Input reflected, 3. Output reflected, 4. Initial value should be 0xFFFFFFFF, 5. Final XOR value should be 0x0.<br>xxxxxxxx |

## 1.5 Factory Constants

All devices include a memory-mapped FACTORY region which provides read-only data describing the capabilities of a device as well as any factory-provided trim information for use by application software.

Key data provided in the FACTORY memory region includes:

- The device unique 96-bit identity
- The default BSL pins
- The total MAIN region flash memory size (in KB)
- The total DATA region flash memory size (in KB), if present
- The flash bank count
- The total SRAM memory size (in KB)
- The temperature sensor calibration value
- The SYSPLL startup parameters

### *1.5.1 FACTORYREGION Layout Types*

The below table shows which FACTORYREGION layout type to use with which devices. The MSPM0-SDK includes a configurator tool to help the user configure FACTORYREGION contents.

**Table 1-115. FACTORYREGION Layout Types**

| NONMAIN Layout Type | Supported Devices |
|---|---|
| Type A | • MSPM0G110x<br>• MSPM0G150x<br>• MSPM0G310x<br>• MSPM0G350x |
| Type E | • MSPM0G120x<br>• MSPM0G121x<br>• MSPM0G122x<br>• MSPM0G320x<br>• MSPM0G321x<br>• MSPM0G322x |
| Type F | • MSPM0G511x<br>• MSPM0G5187 |



Copyright © 2026 Texas Instruments Incorporated

**Table 1-115. FACTORYREGION Layout Types (continued)**

| NONMAIN Layout Type | Supported Devices |
|---|---|
| Type G | • MSPM0G151x<br>• MSPM0G351x<br>• MSPM0G352x |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.2 FACTORYREGION_TYPEA Registers

Table 1-116 lists the memory-mapped registers for the FACTORYREGION_TYPEA registers. All register offset addresses not listed in Table 1-116 should be considered as reserved locations and the register contents should not be modified.

**Table 1-116. FACTORYREGION_TYPEA Registers**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 41C40000h | TRACEID | Defined by TI, during ATE, based on wafer | | Go |
| 41C40004h | DEVICEID | Device identifier | | Go |
| 41C40008h | USERID | Device variant identifier | | Go |
| 41C4000Ch | BSLPIN_UART | BSL UART Pin Configuration | | Go |
| 41C40010h | BSLPIN_I2C | BSL I2C Pin Configuration | | Go |
| 41C40014h | BSLPIN_INVOKE | BSL Pin Invocation Configuration | | Go |
| 41C40018h | SRAMFLASH | | | Go |
| 41C4001Ch | PLLSTARTUP0_4_8MHZ | | | Go |
| 41C40020h | PLLSTARTUP1_4_8MHZ | System PLL Paramater 1 MMR --- Data from Flash Table Lookup | | Go |
| 41C40024h | PLLSTARTUP0_8_16MHZ | | | Go |
| 41C40028h | PLLSTARTUP1_8_16MHZ | System PLL Paramater 1 MMR --- Data from Flash Table Lookup | | Go |
| 41C4002Ch | PLLSTARTUP0_16_32MHZ | | | Go |
| 41C40030h | PLLSTARTUP1_16_32MHZ | System PLL Paramater 1 MMR --- Data from Flash Table Lookup | | Go |
| 41C40034h | PLLSTARTUP0_32_48MHZ | | | Go |
| 41C40038h | PLLSTARTUP1_32_48MHZ | System PLL Paramater 1 MMR --- Data from Flash Table Lookup | | Go |
| 41C4003Ch | TEMP_SENSE0 | Temperature sensor room temperature calibration code.<br>This is ADC conversion results of temperature sensor output voltage.<br>Included in BOOTCRC calculation. | | Go |
| 41C40040h | RESERVED0 | | | |
| 41C40044h | RESERVED1 | | | |
| 41C40048h | RESERVED2 | | | |
| 41C4004Ch | RESERVED3 | | | |
| 41C40050h | RESERVED4 | | | |
| 41C40054h | RESERVED5 | | | |
| 41C40058h | RESERVED6 | | | |



Copyright © 2026 Texas Instruments Incorporated

**Table 1-116. FACTORYREGION_TYPEA Registers (continued)**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 41C4005Ch | RESERVED7 | | | |
| 41C40060h | RESERVED8 | | | |
| 41C40064h | RESERVED9 | | | |
| 41C40068h | RESERVED10 | | | |
| 41C4006Ch | RESERVED11 | | | |
| 41C40070h | RESERVED12 | | | |
| 41C40074h | RESERVED13 | | | |
| 41C40078h | RESERVED14 | | | |
| 41C4007Ch | BOOTCRC | BOOTCRC records the 32-bit CRC of all locations in OPEN including reserved locations. | | |

Complex bit access types are encoded to fit into small table cells. Table 1-117 shows the codes that are used for access types in this section.

**Table 1-117. FACTORYREGION_TYPEA Access Type Codes**

| Access Type | Code | Description |
|---|---|---|
| Read Type | | |
| R | R | Read |
| Reset or Default Value | | |
| *-n* | | Value after reset or the default value |



Copyright © 2026 Texas Instruments Incorporated

#### 1.5.2.1 TRACEID (Offset = 41C40000h) [Reset = XXXXXXXXh]

TRACEID is shown in Figure 1-104 and described in Table 1-118.

Return to the Summary Table.

unique per part shipped, done per established TI process

**Figure 1-104. TRACEID**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-XXXXXXXXh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-118. TRACEID Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | R | x | |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.2.2 DEVICEID (Offset = 41C40004h) [Reset = XXXXXXXXh]

DEVICEID is shown in Figure 1-105 and described in Table 1-119.

Return to the Summary Table.

Device identifier (die revision specific). Refer to device factory constants section of the datasheet

**Figure 1-105. DEVICEID**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| VERSION | | | | PARTNUM | | | |
| R-Xh | | | | R-XXXXh | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| PARTNUM | | | | | | | |
| R-XXXXh | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| PARTNUM | | | | MANUFACTURER | | | |
| R-XXXXh | | | | R-XXh | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MANUFACTURER | | | | | | | ALWAYS_1 |
| R-XXh | | | | | | | R-Xh |

**Table 1-119. DEVICEID Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-28 | VERSION | R | x | Revision of the device. This field should change each time that the logic or mask set of the device is revised. |
| 27-12 | PARTNUM | R | x | Part number of the device. |
| 11-1 | MANUFACTURER | R | x | TI's JEDEC bank and company code, which is: 00000010111b |
| 0 | ALWAYS_1 | R | x | This is always 1 |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.2.3 USERID (Offset = 41C40008h) [Reset = XXXXXXXXh]

USERID is shown in Figure 1-106 and described in Table 1-120.

Return to the Summary Table.

Defines the device variant feature-set. Refer to device factory constants section of the datasheet

**Figure 1-106. USERID**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| START | MAJORREV | | | MINORREV | | | |
| R-X | R-X | | | R-X | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| VARIANT | | | | | | | |
| R-X | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| PART | | | | | | | |
| R-X | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PART | | | | | | | |
| R-X | | | | | | | |

**Table 1-120. USERID Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | START | R | 1h | |
| 30-28 | MAJORREV | R | X | Monotonic increasing value indicating a new revision of the SKU significant enough that users of the device may have to revise PCB or or software design<br>x |
| 27-24 | MINORREV | R | X | Monotonic increasing value indicating a new revision of the SKU that preserves compatibility with lesser minorrev values. New capability may be introduced such that lesser minorrev numbers may not be compatible with greater if the new capability is used.<br>x |
| 23-16 | VARIANT | R | X | Bit pattern uniquely identifying a variant of a part. This is used to indicate memory or package variations of the same part number. This number shall be selected at random among the remaining numbers for the relevant combination of IDCODE.device and USERCODE.part such that the order of creation cannot be inferred by the number. The variant number does not encode specifics of the variant directly.<br>x |
| 15-0 | PART | R | X | Bit pattern that uniquely identifying a part. This is used to identify the specific part based on the die identified in DEVICEID.device. This number shall be selected at random among the remaining numbers for DEVICEID.device such that the order of creation cannot be inferred by the number. This value does not encode the part number directly.<br>x |



Copyright © 2026 Texas Instruments Incorporated

#### 1.5.2.4 BSLPIN_UART (Offset = 41C4000Ch) [Reset = XXXXXXXXh]

BSLPIN_UART is shown in Figure 1-107 and described in Table 1-121.

Return to the Summary Table.

BSL UART Pin Configuration

**Figure 1-107. BSLPIN_UART**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| UART_TXD_PF | | | | | | | | UART_TXD_PAD | | | | | | | |
| R-XXh | | | | | | | | R-XXh | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| UART_RXD_PF | | | | | | | | UART_RXD_PAD | | | | | | | |
| R-XXh | | | | | | | | R-XXh | | | | | | | |

**Table 1-121. BSLPIN_UART Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | UART_TXD_PF | R | x | UART TXD Pin Function Selection Value |
| 23-16 | UART_TXD_PAD | R | x | UART TXD Pin used by BSL |
| 15-8 | UART_RXD_PF | R | x | UART RXD Pin Function Selection Value |
| 7-0 | UART_RXD_PAD | R | x | UART RXD Pad used by BSL |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.2.5 BSLPIN_I2C (Offset = 41C40010h) [Reset = XXXXXXXXh]

BSLPIN_I2C is shown in Figure 1-108 and described in Table 1-122.

Return to the Summary Table.

BSL I2C Pin Configuration

**Figure 1-108. BSLPIN_I2C**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| I2C_SCL_PF | | | | | | | | I2C_SCL_PAD | | | | | | | |
| R-XXh | | | | | | | | R-XXh | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| I2C_SDA_PF | | | | | | | | I2C_SDA_PAD | | | | | | | |
| R-XXh | | | | | | | | R-XXh | | | | | | | |

**Table 1-122. BSLPIN_I2C Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | I2C_SCL_PF | R | x | I2C SCL Pin Function Selection Value |
| 23-16 | I2C_SCL_PAD | R | x | I2C SCL Pin used by BSL |
| 15-8 | I2C_SDA_PF | R | x | I2C SDA Pin Function Selection Value |
| 7-0 | I2C_SDA_PAD | R | x | I2C SDA Pin used by BSL |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.2.6 BSLPIN_INVOKE (Offset = 41C40014h) [Reset = 0000XXXXh]

BSLPIN_INVOKE is shown in Figure 1-109 and described in Table 1-123.

Return to the Summary Table.

BSL Pin Invocation Configuration

**Figure 1-109. BSLPIN_INVOKE**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | GPIO_REG_SEL | | GPIO_PIN_SEL | | | | |
| R-0h | R-Xh | | R-Xh | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GPIO_LEVEL | BSL_PAD | | | | | | |
| R-Xh | R-Xh | | | | | | |

**Table 1-123. BSLPIN_INVOKE Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-15 | RESERVED | R | 0h | |
| 14-13 | GPIO_REG_SEL | R | x | GPIO Module Selection |
| 12-8 | GPIO_PIN_SEL | R | x | GPIO Pin Number in GPIO Module |
| 7 | GPIO_LEVEL | R | x | GPIO Level Configuration for BSL Invocation |
| 6-0 | BSL_PAD | R | x | BSL Invocation Pin Number |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.2.7 SRAMFLASH (Offset = 41C40018h) [Reset = XXXXXXXXh]

SRAMFLASH is shown in Figure 1-110 and described in Table 1-124.

Return to the Summary Table.

**Figure 1-110. SRAMFLASH**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DATAFLASH_SZ | | | | | | SRAM_SZ | |
| R-Xh | | | | | | R-XXh | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| SRAM_SZ | | | | | | | |
| R-XXh | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | MAINNUMBANKS | | MAINFLASH_SZ | | | |
| R-0h | | R-Xh | | R-XXXh | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MAINFLASH_SZ | | | | | | | |
| R-XXXh | | | | | | | |

**Table 1-124. SRAMFLASH Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | DATAFLASH_SZ | R | x | The encoding of the field is that the value of the field is an integer to be interpreted as number of KBs.<br>For eg: if the value of the field id 4, then it is 4KB, if the value is 32, then 32KB, and so on. |
| 25-16 | SRAM_SZ | R | x | The encoding of the field is that the value of the field is an integer to be interpreted as number of KBs. For eg: if the value of the field id 4, then it is 4KB, if the value is 32, then 32KB, and so on. |
| 15-14 | RESERVED | R | 0h | |
| 13-12 | MAINNUMBANKS | R | x | Value of 0 means there is a single flash bank; value of 1 means there are two flash banks. Etc.<br>0h = 0<br>1h = 1<br>2h = 2<br>3h = 3 |
| 11-0 | MAINFLASH_SZ | R | x | The encoding of the field is that the value of the field is an integer to be interpreted as number of KBs. For eg: if the value of the field id 4, then it is 4KB, if the value is 32, then 32KB, and so on. |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.2.8 PLLSTARTUP0_4_8MHZ (Offset = 41C4001Ch) [Reset = XXXXXXXXh]

PLLSTARTUP0_4_8MHZ is shown in Figure 1-111 and described in Table 1-125.

Return to the Summary Table.

**Figure 1-111. PLLSTARTUP0_4_8MHZ**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CAPBOVERRIDE | RESERVED | | | CAPBVAL | | | |
| R-Xh | R-0h | | | R-Xh | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | CPCURRENT | | | | | |
| R-0h | | R-Xh | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | STARTTIMELP | | | | | |
| R-0h | | R-Xh | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | STARTTIME | | | | | |
| R-0h | | R-Xh | | | | | |

**Table 1-125. PLLSTARTUP0_4_8MHZ Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | CAPBOVERRIDE | R | x | Override Enable For Cap B<br>0h = 0<br>1h = 1 |
| 30-29 | RESERVED | R | 0h | |
| 28-24 | CAPBVAL | R | x | Override Value for Cap B |
| 23-22 | RESERVED | R | 0h | |
| 21-16 | CPCURRENT | R | x | Charge Pump Current |
| 15-14 | RESERVED | R | 0h | |
| 13-8 | STARTTIMELP | R | x | Startup time from Low Power Exit to Locked Clock in resolution of 1usec |
| 7-6 | RESERVED | R | 0h | |
| 5-0 | STARTTIME | R | x | Startup time from Enable to Locked Clock in resolution of 1usec |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.2.9 PLLSTARTUP1_4_8MHZ (Offset = 41C40020h) [Reset = XX0XXXXXh]

PLLSTARTUP1_4_8MHZ is shown in Figure 1-112 and described in Table 1-126.

Return to the Summary Table.

**Figure 1-112. PLLSTARTUP1_4_8MHZ**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LPFRESC | | | | | | | | RESERVED | | | | | | LPFRESA | |
| R-XXh | | | | | | | | R-0h | | | | | | R-XXh | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LPFRESA | | | | | | | | RESERVED | | | LPFCAPA | | | | |
| R-XXh | | | | | | | | R-0h | | | R-Xh | | | | |

**Table 1-126. PLLSTARTUP1_4_8MHZ Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | LPFRESC | R | x | Loop Filter Res C |
| 23-18 | RESERVED | R | 0h | |
| 17-8 | LPFRESA | R | x | Loop Filter Res A |
| 7-5 | RESERVED | R | 0h | |
| 4-0 | LPFCAPA | R | x | Loop Filter Cap A |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.2.10 PLLSTARTUP0_8_16MHZ (Offset = 41C40024h) [Reset = XXXXXXXXh]

PLLSTARTUP0_8_16MHZ is shown in Figure 1-113 and described in Table 1-127.

Return to the Summary Table.

**Figure 1-113. PLLSTARTUP0_8_16MHZ**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CAPBOVERRIDE | RESERVED | | | CAPBVAL | | | |
| R-Xh | R-0h | | | R-Xh | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | CPCURRENT | | | | | |
| R-0h | | R-Xh | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | STARTTIMELP | | | | | |
| R-0h | | R-Xh | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | STARTTIME | | | | | |
| R-0h | | R-Xh | | | | | |

**Table 1-127. PLLSTARTUP0_8_16MHZ Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | CAPBOVERRIDE | R | x | Override Enable For Cap B<br>0h = 0<br>1h = 1 |
| 30-29 | RESERVED | R | 0h | |
| 28-24 | CAPBVAL | R | x | Override Value for Cap B |
| 23-22 | RESERVED | R | 0h | |
| 21-16 | CPCURRENT | R | x | Charge Pump Current |
| 15-14 | RESERVED | R | 0h | |
| 13-8 | STARTTIMELP | R | x | Startup time from Low Power Exit to Locked Clock in resolution of 1usec |
| 7-6 | RESERVED | R | 0h | |
| 5-0 | STARTTIME | R | x | Startup time from Enable to Locked Clock in resolution of 1usec |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.2.11 PLLSTARTUP1_8_16MHZ (Offset = 41C40028h) [Reset = XX0XXXXXh]

PLLSTARTUP1_8_16MHZ is shown in Figure 1-114 and described in Table 1-128.

Return to the Summary Table.

**Figure 1-114. PLLSTARTUP1_8_16MHZ**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LPFRESC | | | | | | | | RESERVED | | | | | | LPFRESA | |
| R-XXh | | | | | | | | R-0h | | | | | | R-XXh | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LPFRESA | | | | | | | | RESERVED | | | LPFCAPA | | | | |
| R-XXh | | | | | | | | R-0h | | | R-Xh | | | | |

**Table 1-128. PLLSTARTUP1_8_16MHZ Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | LPFRESC | R | x | Loop Filter Res C |
| 23-18 | RESERVED | R | 0h | |
| 17-8 | LPFRESA | R | x | Loop Filter Res A |
| 7-5 | RESERVED | R | 0h | |
| 4-0 | LPFCAPA | R | x | Loop Filter Cap A |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.2.12 PLLSTARTUP0_16_32MHZ (Offset = 41C4002Ch) [Reset = XXXXXXXXh]

PLLSTARTUP0_16_32MHZ is shown in Figure 1-115 and described in Table 1-129.

Return to the Summary Table.

**Figure 1-115. PLLSTARTUP0_16_32MHZ**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CAPBOVERRIDE | RESERVED | | | CAPBVAL | | | |
| R-Xh | R-0h | | | R-Xh | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | CPCURRENT | | | | | |
| R-0h | | R-Xh | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | STARTTIMELP | | | | | |
| R-0h | | R-Xh | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | STARTTIME | | | | | |
| R-0h | | R-Xh | | | | | |

**Table 1-129. PLLSTARTUP0_16_32MHZ Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | CAPBOVERRIDE | R | x | Override Enable For Cap B<br>0h = 0<br>1h = 1 |
| 30-29 | RESERVED | R | 0h | |
| 28-24 | CAPBVAL | R | x | Override Value for Cap B |
| 23-22 | RESERVED | R | 0h | |
| 21-16 | CPCURRENT | R | x | Charge Pump Current |
| 15-14 | RESERVED | R | 0h | |
| 13-8 | STARTTIMELP | R | x | Startup time from Low Power Exit to Locked Clock in resolution of 1usec |
| 7-6 | RESERVED | R | 0h | |
| 5-0 | STARTTIME | R | x | Startup time from Enable to Locked Clock in resolution of 1usec |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.2.13 PLLSTARTUP1_16_32MHZ (Offset = 41C40030h) [Reset = XX0XXXXXh]

PLLSTARTUP1_16_32MHZ is shown in Figure 1-116 and described in Table 1-130.

Return to the Summary Table.

**Figure 1-116. PLLSTARTUP1_16_32MHZ**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LPFRESC | | | | | | | | RESERVED | | | | | | LPFRESA | |
| R-XXh | | | | | | | | R-0h | | | | | | R-XXh | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LPFRESA | | | | | | | | RESERVED | | | LPFCAPA | | | | |
| R-XXh | | | | | | | | R-0h | | | R-Xh | | | | |

**Table 1-130. PLLSTARTUP1_16_32MHZ Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | LPFRESC | R | x | Loop Filter Res C |
| 23-18 | RESERVED | R | 0h | |
| 17-8 | LPFRESA | R | x | Loop Filter Res A |
| 7-5 | RESERVED | R | 0h | |
| 4-0 | LPFCAPA | R | x | Loop Filter Cap A |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.2.14 PLLSTARTUP0_32_48MHZ (Offset = 41C40034h) [Reset = XXXXXXXXh]

PLLSTARTUP0_32_48MHZ is shown in Figure 1-117 and described in Table 1-131.

Return to the Summary Table.

**Figure 1-117. PLLSTARTUP0_32_48MHZ**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CAPBOVERRIDE | RESERVED | | | CAPBVAL | | | |
| R-Xh | R-0h | | | R-Xh | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | CPCURRENT | | | | | |
| R-0h | | R-Xh | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | STARTTIMELP | | | | | |
| R-0h | | R-Xh | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | STARTTIME | | | | | |
| R-0h | | R-Xh | | | | | |

**Table 1-131. PLLSTARTUP0_32_48MHZ Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | CAPBOVERRIDE | R | x | Override Enable For Cap B<br>0h = 0<br>1h = 1 |
| 30-29 | RESERVED | R | 0h | |
| 28-24 | CAPBVAL | R | x | Override Value for Cap B |
| 23-22 | RESERVED | R | 0h | |
| 21-16 | CPCURRENT | R | x | Charge Pump Current |
| 15-14 | RESERVED | R | 0h | |
| 13-8 | STARTTIMELP | R | x | Startup time from Low Power Exit to Locked Clock in resolution of 1usec |
| 7-6 | RESERVED | R | 0h | |
| 5-0 | STARTTIME | R | x | Startup time from Enable to Locked Clock in resolution of 1usec |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.2.15 PLLSTARTUP1_32_48MHZ (Offset = 41C40038h) [Reset = XX0XXXXXh]

PLLSTARTUP1_32_48MHZ is shown in Figure 1-118 and described in Table 1-132.

Return to the Summary Table.

**Figure 1-118. PLLSTARTUP1_32_48MHZ**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LPFRESC | | | | | | | | RESERVED | | | | | | LPFRESA | |
| R-XXh | | | | | | | | R-0h | | | | | | R-XXh | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LPFRESA | | | | | | | | RESERVED | | | LPFCAPA | | | | |
| R-XXh | | | | | | | | R-0h | | | R-Xh | | | | |

**Table 1-132. PLLSTARTUP1_32_48MHZ Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | LPFRESC | R | x | Loop Filter Res C |
| 23-18 | RESERVED | R | 0h | |
| 17-8 | LPFRESA | R | x | Loop Filter Res A |
| 7-5 | RESERVED | R | 0h | |
| 4-0 | LPFCAPA | R | x | Loop Filter Cap A |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.2.16 TEMP_SENSE0 (Offset = 41C4003Ch) [Reset = XXXXXXXXh]

TEMP_SENSE0 is shown in Figure 1-119 and described in Table 1-133.

Return to the Summary Table.

Temperature sensor room temperature calibration code. This is ADC conversion results of temperature sensor output voltage. Included in BOOTCRC calculation.

**Figure 1-119. TEMP_SENSE0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-XXXXXXXXh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-133. TEMP_SENSE0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | R | x | |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.2.17 BOOTCRC (Offset = 41C4007Ch) [Reset = XXXXXXXXh]

BOOTCRC is shown in Figure 1-120 and described in Table 1-134.

Return to the Summary Table.

BOOTCRC records the 32-bit CRC of all locations in OPEN including reserved locations.

**Figure 1-120. BOOTCRC**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-XXXXXXXXh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-134. BOOTCRC Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | R | x | |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.3 FACTORYREGION_TYPEE Registers

Table 1-135 lists the memory-mapped registers for the FACTORYREGION_TYPEE registers. All register offset addresses not listed in Table 1-135 should be considered as reserved locations and the register contents should not be modified.

**Table 1-135. FACTORYREGION_TYPEE Registers**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 41C4000 0h | TRACEID | Defined by TI, during ATE, based on wafer | | Go |
| 41C4000 4h | DEVICEID | Device identifier | | Go |
| 41C4000 8h | USERID | Device variant identifier | | Go |
| 41C4000 Ch | BSLPIN_UART | BSL UART Pin Configuration | | Go |
| 41C4001 0h | BSLPIN_I2C | BSL I2C Pin Configuration | | Go |
| 41C4001 4h | BSLPIN_INVOKE | BSL Pin Invocation Configuration | | Go |
| 41C4001 8h | SRAMFLASH | | | Go |
| 41C4001 Ch | PLLSTARTUP0_4_8MHZ | | | Go |
| 41C4002 0h | PLLSTARTUP1_4_8MHZ | System PLL Paramater 1 MMR --- Data from Flash Table Lookup | | Go |
| 41C4002 4h | PLLSTARTUP0_8_16MHZ | | | Go |
| 41C4002 8h | PLLSTARTUP1_8_16MHZ | System PLL Paramater 1 MMR --- Data from Flash Table Lookup | | Go |
| 41C4002 Ch | PLLSTARTUP0_16_32MHZ | | | Go |
| 41C4003 0h | PLLSTARTUP1_16_32MHZ | System PLL Paramater 1 MMR --- Data from Flash Table Lookup | | Go |
| 41C4003 4h | PLLSTARTUP0_32_48MHZ | | | Go |
| 41C4003 8h | PLLSTARTUP1_32_48MHZ | System PLL Paramater 1 MMR --- Data from Flash Table Lookup | | Go |
| 41C4003 Ch | TEMP_SENSE0 | Temperature sensor room temperature calibration code.<br>This is ADC conversion results of temperature sensor output voltage.<br>Included in BOOTCRC calculation. | | Go |
| 41C4004 0h | RESERVED0 | | | |
| 41C4004 4h | RESERVED1 | | | |
| 41C4004 8h | RESERVED2 | | | |
| 41C4004 Ch | RESERVED3 | | | |
| 41C4005 0h | RESERVED4 | | | |
| 41C4005 4h | RESERVED5 | | | |
| 41C4005 8h | RESERVED6 | | | |



Copyright © 2026 Texas Instruments Incorporated

**Table 1-135. FACTORYREGION_TYPEE Registers (continued)**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 41C4005Ch | RESERVED7 | | | |
| 41C40060h | RESERVED8 | | | |
| 41C40064h | RESERVED9 | | | |
| 41C40068h | RESERVED10 | | | |
| 41C4006Ch | RESERVED11 | | | |
| 41C40070h | RESERVED12 | | | |
| 41C40074h | RESERVED13 | | | |
| 41C40078h | TEMP_SENSE_0KELVIN | | | Go |
| 41C4007Ch | BOOTCRC | BOOTCRC records the 32-bit CRC of all locations in OPEN including reserved locations. | | |

Complex bit access types are encoded to fit into small table cells. Table 1-136 shows the codes that are used for access types in this section.

**Table 1-136. FACTORYREGION_TYPEE Access Type Codes**

| Access Type | Code | Description |
|---|---|---|
| Read Type | | |
| R | R | Read |
| Reset or Default Value | | |
| *-n* | | Value after reset or the default value |



Copyright © 2026 Texas Instruments Incorporated

#### 1.5.3.1 TRACEID (Offset = 41C40000h) [Reset = XXXXXXXXh]

TRACEID is shown in Figure 1-121 and described in Table 1-137.

Return to the Summary Table.

unique per part shipped, done per established TI process

**Figure 1-121. TRACEID**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-XXXXXXXXh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-137. TRACEID Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | R | x | |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.3.2 DEVICEID (Offset = 41C40004h) [Reset = XXXXXXXXh]

DEVICEID is shown in Figure 1-122 and described in Table 1-138.

Return to the Summary Table.

Device identifier (die revision specific). Refer to device factory constants section of the datasheet

**Figure 1-122. DEVICEID**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| VERSION | | | | PARTNUM | | | |
| R-Xh | | | | R-XXXXh | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| PARTNUM | | | | | | | |
| R-XXXXh | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| PARTNUM | | | | MANUFACTURER | | | |
| R-XXXXh | | | | R-XXh | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MANUFACTURER | | | | | | | ALWAYS_1 |
| R-XXh | | | | | | | R-Xh |

**Table 1-138. DEVICEID Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-28 | VERSION | R | x | Revision of the device. This field should change each time that the logic or mask set of the device is revised. |
| 27-12 | PARTNUM | R | x | Part number of the device. |
| 11-1 | MANUFACTURER | R | x | TI's JEDEC bank and company code, which is: 00000010111b |
| 0 | ALWAYS_1 | R | x | This is always 1 |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.3.3 USERID (Offset = 41C40008h) [Reset = XXXXXXXXh]

USERID is shown in Figure 1-123 and described in Table 1-139.

Return to the Summary Table.

Defines the device variant feature-set. Refer to device factory constants section of the datasheet

**Figure 1-123. USERID**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| START | MAJORREV | | | MINORREV | | | |
| R-X | R-X | | | R-X | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| VARIANT | | | | | | | |
| R-X | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| PART | | | | | | | |
| R-X | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PART | | | | | | | |
| R-X | | | | | | | |

**Table 1-139. USERID Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | START | R | 1h | |
| 30-28 | MAJORREV | R | X | Monotonic increasing value indicating a new revision of the SKU significant enough that users of the device may have to revise PCB or or software design<br>x |
| 27-24 | MINORREV | R | X | Monotonic increasing value indicating a new revision of the SKU that preserves compatibility with lesser minorrev values. New capability may be introduced such that lesser minorrev numbers may not be compatible with greater if the new capability is used.<br>x |
| 23-16 | VARIANT | R | X | Bit pattern uniquely identifying a variant of a part. This is used to indicate memory or package variations of the same part number. This number shall be selected at random among the remaining numbers for the relevant combination of IDCODE.device and USERCODE.part such that the order of creation cannot be inferred by the number. The variant number does not encode specifics of the variant directly.<br>x |
| 15-0 | PART | R | X | Bit pattern that uniquely identifying a part. This is used to identify the specific part based on the die identified in DEVICEID.device. This number shall be selected at random among the remaining numbers for DEVICEID.device such that the order of creation cannot be inferred by the number. This value does not encode the part number directly.<br>x |



Copyright © 2026 Texas Instruments Incorporated

#### 1.5.3.4 BSLPIN_UART (Offset = 41C4000Ch) [Reset = XXXXXXXXh]

BSLPIN_UART is shown in Figure 1-124 and described in Table 1-140.

Return to the Summary Table.

BSL UART Pin Configuration

**Figure 1-124. BSLPIN_UART**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| UART_TXD_PF | | | | | | | | UART_TXD_PAD | | | | | | | |
| R-XXh | | | | | | | | R-XXh | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| UART_RXD_PF | | | | | | | | UART_RXD_PAD | | | | | | | |
| R-XXh | | | | | | | | R-XXh | | | | | | | |

**Table 1-140. BSLPIN_UART Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | UART_TXD_PF | R | x | UART TXD Pin Function Selection Value |
| 23-16 | UART_TXD_PAD | R | x | UART TXD Pin used by BSL |
| 15-8 | UART_RXD_PF | R | x | UART RXD Pin Function Selection Value |
| 7-0 | UART_RXD_PAD | R | x | UART RXD Pad used by BSL |



Copyright © 2026 Texas Instruments Incorporated

#### 1.5.3.5 BSLPIN_I2C (Offset = 41C40010h) [Reset = XXXXXXXXh]

BSLPIN_I2C is shown in Figure 1-125 and described in Table 1-141.

Return to the Summary Table.

BSL I2C Pin Configuration

**Figure 1-125. BSLPIN_I2C**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| I2C_SCL_PF | | | | | | | | I2C_SCL_PAD | | | | | | | |
| R-XXh | | | | | | | | R-XXh | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| I2C_SDA_PF | | | | | | | | I2C_SDA_PAD | | | | | | | |
| R-XXh | | | | | | | | R-XXh | | | | | | | |

**Table 1-141. BSLPIN_I2C Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | I2C_SCL_PF | R | x | I2C SCL Pin Function Selection Value |
| 23-16 | I2C_SCL_PAD | R | x | I2C SCL Pin used by BSL |
| 15-8 | I2C_SDA_PF | R | x | I2C SDA Pin Function Selection Value |
| 7-0 | I2C_SDA_PAD | R | x | I2C SDA Pin used by BSL |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.3.6 BSLPIN_INVOKE (Offset = 41C40014h) [Reset = 0000XXXXh]

BSLPIN_INVOKE is shown in Figure 1-126 and described in Table 1-142.

Return to the Summary Table.

BSL Pin Invocation Configuration

**Figure 1-126. BSLPIN_INVOKE**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | GPIO_REG_SEL | | GPIO_PIN_SEL | | | | |
| R-0h | R-Xh | | R-Xh | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GPIO_LEVEL | BSL_PAD | | | | | | |
| R-Xh | R-Xh | | | | | | |

**Table 1-142. BSLPIN_INVOKE Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-15 | RESERVED | R | 0h | |
| 14-13 | GPIO_REG_SEL | R | x | GPIO Module Selection |
| 12-8 | GPIO_PIN_SEL | R | x | GPIO Pin Number in GPIO Module |
| 7 | GPIO_LEVEL | R | x | GPIO Level Configuration for BSL Invocation |
| 6-0 | BSL_PAD | R | x | BSL Invocation Pin Number |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.3.7 SRAMFLASH (Offset = 41C40018h) [Reset = XXXXXXXXh]

SRAMFLASH is shown in Figure 1-127 and described in Table 1-143.

Return to the Summary Table.

**Figure 1-127. SRAMFLASH**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DATAFLASH_SZ | | | | | | SRAM_SZ | |
| R-Xh | | | | | | R-XXh | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| SRAM_SZ | | | | | | | |
| R-XXh | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | MAINNUMBANKS | | MAINFLASH_SZ | | | |
| R-0h | | R-Xh | | R-XXXh | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MAINFLASH_SZ | | | | | | | |
| R-XXXh | | | | | | | |

**Table 1-143. SRAMFLASH Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | DATAFLASH_SZ | R | x | The encoding of the field is that the value of the field is an integer to be interpreted as number of KBs.<br>For eg: if the value of the field id 4, then it is 4KB, if the value is 32, then 32KB, and so on. |
| 25-16 | SRAM_SZ | R | x | The encoding of the field is that the value of the field is an integer to be interpreted as number of KBs. For eg: if the value of the field id 4, then it is 4KB, if the value is 32, then 32KB, and so on. |
| 15-14 | RESERVED | R | 0h | |
| 13-12 | MAINNUMBANKS | R | x | Value of 0 means there is a single flash bank; value of 1 means there are two flash banks. Etc.<br>0h = 0<br>1h = 1<br>2h = 2<br>3h = 3 |
| 11-0 | MAINFLASH_SZ | R | x | The encoding of the field is that the value of the field is an integer to be interpreted as number of KBs. For eg: if the value of the field id 4, then it is 4KB, if the value is 32, then 32KB, and so on. |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.3.8 PLLSTARTUP0_4_8MHZ (Offset = 41C4001Ch) [Reset = XXXXXXXXh]

PLLSTARTUP0_4_8MHZ is shown in Figure 1-128 and described in Table 1-144.

Return to the Summary Table.

**Figure 1-128. PLLSTARTUP0_4_8MHZ**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CAPBOVERRIDE | RESERVED | | CAPBVAL | | | | |
| R-Xh | R-0h | | R-Xh | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | CPCURRENT | | | | | |
| R-0h | | R-Xh | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | STARTTIMELP | | | | | |
| R-0h | | R-Xh | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | STARTTIME | | | | | |
| R-0h | | R-Xh | | | | | |

**Table 1-144. PLLSTARTUP0_4_8MHZ Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | CAPBOVERRIDE | R | x | Override Enable For Cap B<br>0h = 0<br>1h = 1 |
| 30-29 | RESERVED | R | 0h | |
| 28-24 | CAPBVAL | R | x | Override Value for Cap B |
| 23-22 | RESERVED | R | 0h | |
| 21-16 | CPCURRENT | R | x | Charge Pump Current |
| 15-14 | RESERVED | R | 0h | |
| 13-8 | STARTTIMELP | R | x | Startup time from Low Power Exit to Locked Clock in resolution of 1usec |
| 7-6 | RESERVED | R | 0h | |
| 5-0 | STARTTIME | R | x | Startup time from Enable to Locked Clock in resolution of 1usec |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.3.9 PLLSTARTUP1_4_8MHZ (Offset = 41C40020h) [Reset = XX0XXXXXh]

PLLSTARTUP1_4_8MHZ is shown in Figure 1-129 and described in Table 1-145.

Return to the Summary Table.

**Figure 1-129. PLLSTARTUP1_4_8MHZ**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LPFRESC | | | | | | | | RESERVED | | | | | | LPFRESA | |
| R-XXh | | | | | | | | R-0h | | | | | | R-XXh | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LPFRESA | | | | | | | | RESERVED | | | LPFCAPA | | | | |
| R-XXh | | | | | | | | R-0h | | | R-Xh | | | | |

**Table 1-145. PLLSTARTUP1_4_8MHZ Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | LPFRESC | R | x | Loop Filter Res C |
| 23-18 | RESERVED | R | 0h | |
| 17-8 | LPFRESA | R | x | Loop Filter Res A |
| 7-5 | RESERVED | R | 0h | |
| 4-0 | LPFCAPA | R | x | Loop Filter Cap A |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.3.10 PLLSTARTUP0_8_16MHZ (Offset = 41C40024h) [Reset = XXXXXXXXh]

PLLSTARTUP0_8_16MHZ is shown in Figure 1-130 and described in Table 1-146.

Return to the Summary Table.

**Figure 1-130. PLLSTARTUP0_8_16MHZ**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CAPBOVERRIDE | RESERVED | | | CAPBVAL | | | |
| R-Xh | R-0h | | | R-Xh | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | CPCURRENT | | | | | |
| R-0h | | R-Xh | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | STARTTIMELP | | | | | |
| R-0h | | R-Xh | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | STARTTIME | | | | | |
| R-0h | | R-Xh | | | | | |

**Table 1-146. PLLSTARTUP0_8_16MHZ Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | CAPBOVERRIDE | R | x | Override Enable For Cap B<br>0h = 0<br>1h = 1 |
| 30-29 | RESERVED | R | 0h | |
| 28-24 | CAPBVAL | R | x | Override Value for Cap B |
| 23-22 | RESERVED | R | 0h | |
| 21-16 | CPCURRENT | R | x | Charge Pump Current |
| 15-14 | RESERVED | R | 0h | |
| 13-8 | STARTTIMELP | R | x | Startup time from Low Power Exit to Locked Clock in resolution of 1usec |
| 7-6 | RESERVED | R | 0h | |
| 5-0 | STARTTIME | R | x | Startup time from Enable to Locked Clock in resolution of 1usec |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.3.11 PLLSTARTUP1_8_16MHZ (Offset = 41C40028h) [Reset = XX0XXXXXh]

PLLSTARTUP1_8_16MHZ is shown in Figure 1-131 and described in Table 1-147.

Return to the Summary Table.

**Figure 1-131. PLLSTARTUP1_8_16MHZ**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LPFRESC | | | | | | | | RESERVED | | | | | | LPFRESA | |
| R-XXh | | | | | | | | R-0h | | | | | | R-XXh | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LPFRESA | | | | | | | | RESERVED | | | LPFCAPA | | | | |
| R-XXh | | | | | | | | R-0h | | | R-Xh | | | | |

**Table 1-147. PLLSTARTUP1_8_16MHZ Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | LPFRESC | R | x | Loop Filter Res C |
| 23-18 | RESERVED | R | 0h | |
| 17-8 | LPFRESA | R | x | Loop Filter Res A |
| 7-5 | RESERVED | R | 0h | |
| 4-0 | LPFCAPA | R | x | Loop Filter Cap A |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.3.12 PLLSTARTUP0_16_32MHZ (Offset = 41C4002Ch) [Reset = XXXXXXXXh]

PLLSTARTUP0_16_32MHZ is shown in Figure 1-132 and described in Table 1-148.

Return to the Summary Table.

**Figure 1-132. PLLSTARTUP0_16_32MHZ**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CAPBOVERRIDE | RESERVED | | | CAPBVAL | | | |
| R-Xh | R-0h | | | R-Xh | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | CPCURRENT | | | | | |
| R-0h | | R-Xh | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | STARTTIMELP | | | | | |
| R-0h | | R-Xh | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | STARTTIME | | | | | |
| R-0h | | R-Xh | | | | | |

**Table 1-148. PLLSTARTUP0_16_32MHZ Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | CAPBOVERRIDE | R | x | Override Enable For Cap B<br>0h = 0<br>1h = 1 |
| 30-29 | RESERVED | R | 0h | |
| 28-24 | CAPBVAL | R | x | Override Value for Cap B |
| 23-22 | RESERVED | R | 0h | |
| 21-16 | CPCURRENT | R | x | Charge Pump Current |
| 15-14 | RESERVED | R | 0h | |
| 13-8 | STARTTIMELP | R | x | Startup time from Low Power Exit to Locked Clock in resolution of 1usec |
| 7-6 | RESERVED | R | 0h | |
| 5-0 | STARTTIME | R | x | Startup time from Enable to Locked Clock in resolution of 1usec |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.3.13 PLLSTARTUP1_16_32MHZ (Offset = 41C40030h) [Reset = XX0XXXXXh]

PLLSTARTUP1_16_32MHZ is shown in Figure 1-133 and described in Table 1-149.

Return to the Summary Table.

**Figure 1-133. PLLSTARTUP1_16_32MHZ**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LPFRESC | | | | | | | | RESERVED | | | | | | LPFRESA | |
| R-XXh | | | | | | | | R-0h | | | | | | R-XXh | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LPFRESA | | | | | | | | RESERVED | | | LPFCAPA | | | | |
| R-XXh | | | | | | | | R-0h | | | R-Xh | | | | |

**Table 1-149. PLLSTARTUP1_16_32MHZ Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | LPFRESC | R | x | Loop Filter Res C |
| 23-18 | RESERVED | R | 0h | |
| 17-8 | LPFRESA | R | x | Loop Filter Res A |
| 7-5 | RESERVED | R | 0h | |
| 4-0 | LPFCAPA | R | x | Loop Filter Cap A |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.3.14 PLLSTARTUP0_32_48MHZ (Offset = 41C40034h) [Reset = XXXXXXXXh]

PLLSTARTUP0_32_48MHZ is shown in Figure 1-134 and described in Table 1-150.

Return to the Summary Table.

**Figure 1-134. PLLSTARTUP0_32_48MHZ**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CAPBOVERRIDE | RESERVED | | | CAPBVAL | | | |
| R-Xh | R-0h | | | R-Xh | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | CPCURRENT | | | | | |
| R-0h | | R-Xh | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | STARTTIMELP | | | | | |
| R-0h | | R-Xh | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | STARTTIME | | | | | |
| R-0h | | R-Xh | | | | | |

**Table 1-150. PLLSTARTUP0_32_48MHZ Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | CAPBOVERRIDE | R | x | Override Enable For Cap B<br>0h = 0<br>1h = 1 |
| 30-29 | RESERVED | R | 0h | |
| 28-24 | CAPBVAL | R | x | Override Value for Cap B |
| 23-22 | RESERVED | R | 0h | |
| 21-16 | CPCURRENT | R | x | Charge Pump Current |
| 15-14 | RESERVED | R | 0h | |
| 13-8 | STARTTIMELP | R | x | Startup time from Low Power Exit to Locked Clock in resolution of 1usec |
| 7-6 | RESERVED | R | 0h | |
| 5-0 | STARTTIME | R | x | Startup time from Enable to Locked Clock in resolution of 1usec |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.3.15 PLLSTARTUP1_32_48MHZ (Offset = 41C40038h) [Reset = XX0XXXXXh]

PLLSTARTUP1_32_48MHZ is shown in Figure 1-135 and described in Table 1-151.

Return to the Summary Table.

**Figure 1-135. PLLSTARTUP1_32_48MHZ**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LPFRESC | | | | | | | | RESERVED | | | | | | LPFRESA | |
| R-XXh | | | | | | | | R-0h | | | | | | R-XXh | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LPFRESA | | | | | | | | RESERVED | | | LPFCAPA | | | | |
| R-XXh | | | | | | | | R-0h | | | R-Xh | | | | |

**Table 1-151. PLLSTARTUP1_32_48MHZ Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | LPFRESC | R | x | Loop Filter Res C |
| 23-18 | RESERVED | R | 0h | |
| 17-8 | LPFRESA | R | x | Loop Filter Res A |
| 7-5 | RESERVED | R | 0h | |
| 4-0 | LPFCAPA | R | x | Loop Filter Cap A |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.3.16 TEMP_SENSE0 (Offset = 41C4003Ch) [Reset = XXXXXXXXh]

TEMP_SENSE0 is shown in Figure 1-136 and described in Table 1-152.

Return to the Summary Table.

Temperature sensor room temperature calibration code. This is ADC conversion results of temperature sensor output voltage. Included in BOOTCRC calculation.

**Figure 1-136. TEMP_SENSE0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-XXXXXXXXh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-152. TEMP_SENSE0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | R | x | |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.3.17 TEMP_SENSE_0KELVIN (Offset = 41C40078h) [Reset = XXXXXXXXh]

TEMP_SENSE_0KELVIN is shown in Figure 1-137 and described in Table 1-153.

Return to the Summary Table.

**Figure 1-137. TEMP_SENSE_0KELVIN**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-XXXXXXXXh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-153. TEMP_SENSE_0KELVIN Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | R | x | |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.3.18 BOOTCRC (Offset = 41C4007Ch) [Reset = XXXXXXXXh]

BOOTCRC is shown in Figure 1-138 and described in Table 1-154.

Return to the Summary Table.

BOOTCRC records the 32-bit CRC of all locations in OPEN including reserved locations.

**Figure 1-138. BOOTCRC**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-XXXXXXXXh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-154. BOOTCRC Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | R | x | |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.4 FACTORYREGION_TYPEF Registers

Table 1-155 lists the memory-mapped registers for the FACTORYREGION_TYPEF registers. All register offset addresses not listed in Table 1-155 should be considered as reserved locations and the register contents should not be modified.

**Table 1-155. FACTORYREGION_TYPEF Registers**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 41C4000 0h | TRACEID | Defined by TI, during ATE, based on wafer | | Go |
| 41C4000 4h | DEVICEID | Device identifier | | Go |
| 41C4000 8h | USERID | Device variant identifier | | Go |
| 41C4000 Ch | BSLPIN_UART | BSL UART Pin Configuration | | Go |
| 41C4001 0h | BSLPIN_I2C | BSL I2C Pin Configuration | | Go |
| 41C4001 4h | BSLPIN_INVOKE | BSL Pin Invocation Configuration | | Go |
| 41C4001 8h | SRAMFLASH | | | Go |
| 41C4001 Ch | PLLSTARTUP0_4_8MHZ | | | Go |
| 41C4002 0h | PLLSTARTUP1_4_8MHZ | System PLL Paramater 1 MMR --- Data from Flash Table Lookup | | Go |
| 41C4002 4h | PLLSTARTUP0_8_16MHZ | | | Go |
| 41C4002 8h | PLLSTARTUP1_8_16MHZ | System PLL Paramater 1 MMR --- Data from Flash Table Lookup | | Go |
| 41C4002 Ch | PLLSTARTUP0_16_32MHZ | | | Go |
| 41C4003 0h | PLLSTARTUP1_16_32MHZ | System PLL Paramater 1 MMR --- Data from Flash Table Lookup | | Go |
| 41C4003 4h | PLLSTARTUP0_32_48MHZ | | | Go |
| 41C4003 8h | PLLSTARTUP1_32_48MHZ | System PLL Paramater 1 MMR --- Data from Flash Table Lookup | | Go |
| 41C4003 Ch | TEMP_SENSE0 | Temperature sensor room temperature calibration code.<br>This is ADC conversion results of temperature sensor output voltage.<br>Included in BOOTCRC calculation. | | Go |
| 41C4004 0h | SEED0 | Random seed | | Go |
| 41C4004 4h | SEED1 | | | Go |
| 41C4004 8h | SEED2 | | | Go |
| 41C4004 Ch | SEED3 | | | Go |
| 41C4005 0h | SEED4 | | | Go |
| 41C4005 4h | SEED5 | | | Go |
| 41C4005 8h | SEED6 | | | Go |



Copyright © 2026 Texas Instruments Incorporated

**Table 1-155. FACTORYREGION_TYPEF Registers (continued)**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 41C4005 Ch | SEED7 | | | Go |
| 41C4006 0h | RESERVED0 | | | |
| 41C4006 4h | RESERVED1 | | | |
| 41C4006 8h | RESERVED2 | | | |
| 41C4006 Ch | RESERVED3 | | | |
| 41C4007 0h | RESERVED4 | | | |
| 41C4007 4h | RESERVED5 | | | |
| 41C4007 8h | TEMP_SENSE_0KELVIN | | | Go |
| 41C4007 Ch | BOOTCRC | BOOTCRC records the 32-bit CRC of all locations in OPEN including reserved locations. | | |

Complex bit access types are encoded to fit into small table cells. Table 1-156 shows the codes that are used for access types in this section.

**Table 1-156. FACTORYREGION_TYPEF Access Type Codes**

| Access Type | Code | Description |
|---|---|---|
| Read Type | | |
| R | R | Read |
| Reset or Default Value | | |
| *-n* | | Value after reset or the default value |



Copyright © 2026 Texas Instruments Incorporated

#### 1.5.4.1 TRACEID (Offset = 41C40000h) [Reset = XXXXXXXXh]

TRACEID is shown in Figure 1-139 and described in Table 1-157.

Return to the Summary Table.

unique per part shipped, done per established TI process

**Figure 1-139. TRACEID**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-XXXXXXXXh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-157. TRACEID Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | R | x | |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.4.2 DEVICEID (Offset = 41C40004h) [Reset = XXXXXXXXh]

DEVICEID is shown in Figure 1-140 and described in Table 1-158.

Return to the Summary Table.

Device identifier (die revision specific). Refer to device factory constants section of the datasheet

**Figure 1-140. DEVICEID**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| VERSION | | | | PARTNUM | | | |
| R-Xh | | | | R-XXXXh | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| PARTNUM | | | | | | | |
| R-XXXXh | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| PARTNUM | | | | MANUFACTURER | | | |
| R-XXXXh | | | | R-XXh | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MANUFACTURER | | | | | | | ALWAYS_1 |
| R-XXh | | | | | | | R-Xh |

**Table 1-158. DEVICEID Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-28 | VERSION | R | x | Revision of the device. This field should change each time that the logic or mask set of the device is revised. |
| 27-12 | PARTNUM | R | x | Part number of the device. |
| 11-1 | MANUFACTURER | R | x | TI's JEDEC bank and company code, which is: 00000010111b |
| 0 | ALWAYS_1 | R | x | This is always 1 |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.4.3 USERID (Offset = 41C40008h) [Reset = XXXXXXXXh]

USERID is shown in Figure 1-141 and described in Table 1-159.

Return to the Summary Table.

Defines the device variant feature-set. Refer to device factory constants section of the datasheet

**Figure 1-141. USERID**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| START | MAJORREV | | | MINORREV | | | |
| R-X | R-X | | | R-X | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| VARIANT | | | | | | | |
| R-X | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| PART | | | | | | | |
| R-X | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PART | | | | | | | |
| R-X | | | | | | | |

**Table 1-159. USERID Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | START | R | 1h | |
| 30-28 | MAJORREV | R | X | Monotonic increasing value indicating a new revision of the SKU significant enough that users of the device may have to revise PCB or or software design<br>x |
| 27-24 | MINORREV | R | X | Monotonic increasing value indicating a new revision of the SKU that preserves compatibility with lesser minorrev values. New capability may be introduced such that lesser minorrev numbers may not be compatible with greater if the new capability is used.<br>x |
| 23-16 | VARIANT | R | X | Bit pattern uniquely identifying a variant of a part. This is used to indicate memory or package variations of the same part number. This number shall be selected at random among the remaining numbers for the relevant combination of IDCODE.device and USERCODE.part such that the order of creation cannot be inferred by the number. The variant number does not encode specifics of the variant directly.<br>x |
| 15-0 | PART | R | X | Bit pattern that uniquely identifying a part. This is used to identify the specific part based on the die identified in DEVICEID.device. This number shall be selected at random among the remaining numbers for DEVICEID.device such that the order of creation cannot be inferred by the number. This value does not encode the part number directly.<br>x |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.4.4 BSLPIN_UART (Offset = 41C4000Ch) [Reset = XXXXXXXXh]

BSLPIN_UART is shown in Figure 1-142 and described in Table 1-160.

Return to the Summary Table.

BSL UART Pin Configuration

**Figure 1-142. BSLPIN_UART**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| UART_TXD_PF | | | | | | | | UART_TXD_PAD | | | | | | | |
| R-XXh | | | | | | | | R-XXh | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| UART_RXD_PF | | | | | | | | UART_RXD_PAD | | | | | | | |
| R-XXh | | | | | | | | R-XXh | | | | | | | |

**Table 1-160. BSLPIN_UART Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | UART_TXD_PF | R | x | UART TXD Pin Function Selection Value |
| 23-16 | UART_TXD_PAD | R | x | UART TXD Pin used by BSL |
| 15-8 | UART_RXD_PF | R | x | UART RXD Pin Function Selection Value |
| 7-0 | UART_RXD_PAD | R | x | UART RXD Pad used by BSL |



Copyright © 2026 Texas Instruments Incorporated

#### 1.5.4.5 BSLPIN_I2C (Offset = 41C40010h) [Reset = XXXXXXXXh]

BSLPIN_I2C is shown in Figure 1-143 and described in Table 1-161.

Return to the Summary Table.

BSL I2C Pin Configuration

**Figure 1-143. BSLPIN_I2C**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| I2C_SCL_PF | | | | | | | | I2C_SCL_PAD | | | | | | | |
| R-XXh | | | | | | | | R-XXh | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| I2C_SDA_PF | | | | | | | | I2C_SDA_PAD | | | | | | | |
| R-XXh | | | | | | | | R-XXh | | | | | | | |

**Table 1-161. BSLPIN_I2C Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | I2C_SCL_PF | R | x | I2C SCL Pin Function Selection Value |
| 23-16 | I2C_SCL_PAD | R | x | I2C SCL Pin used by BSL |
| 15-8 | I2C_SDA_PF | R | x | I2C SDA Pin Function Selection Value |
| 7-0 | I2C_SDA_PAD | R | x | I2C SDA Pin used by BSL |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.4.6 BSLPIN_INVOKE (Offset = 41C40014h) [Reset = 0000XXXXh]

BSLPIN_INVOKE is shown in Figure 1-144 and described in Table 1-162.

Return to the Summary Table.

BSL Pin Invocation Configuration

**Figure 1-144. BSLPIN_INVOKE**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | GPIO_REG_SEL | | GPIO_PIN_SEL | | | | |
| R-0h | R-Xh | | R-Xh | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GPIO_LEVEL | BSL_PAD | | | | | | |
| R-Xh | R-Xh | | | | | | |

**Table 1-162. BSLPIN_INVOKE Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-15 | RESERVED | R | 0h | |
| 14-13 | GPIO_REG_SEL | R | x | GPIO Module Selection |
| 12-8 | GPIO_PIN_SEL | R | x | GPIO Pin Number in GPIO Module |
| 7 | GPIO_LEVEL | R | x | GPIO Level Configuration for BSL Invocation |
| 6-0 | BSL_PAD | R | x | BSL Invocation Pin Number |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.4.7 SRAMFLASH (Offset = 41C40018h) [Reset = XXXXXXXXh]

SRAMFLASH is shown in Figure 1-145 and described in Table 1-163.

Return to the Summary Table.

**Figure 1-145. SRAMFLASH**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DATAFLASH_SZ | | | | | | SRAM_SZ | |
| R-Xh | | | | | | R-XXh | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| SRAM_SZ | | | | | | | |
| R-XXh | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | MAINNUMBANKS | | MAINFLASH_SZ | | | |
| R-0h | | R-Xh | | R-XXXh | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MAINFLASH_SZ | | | | | | | |
| R-XXXh | | | | | | | |

**Table 1-163. SRAMFLASH Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | DATAFLASH_SZ | R | x | The encoding of the field is that the value of the field is an integer to be interpreted as number of KBs.<br>For eg: if the value of the field id 4, then it is 4KB, if the value is 32, then 32KB, and so on. |
| 25-16 | SRAM_SZ | R | x | The encoding of the field is that the value of the field is an integer to be interpreted as number of KBs. For eg: if the value of the field id 4, then it is 4KB, if the value is 32, then 32KB, and so on. |
| 15-14 | RESERVED | R | 0h | |
| 13-12 | MAINNUMBANKS | R | x | Value of 0 means there is a single flash bank; value of 1 means there are two flash banks. Etc.<br>0h = 0<br>1h = 1<br>2h = 2<br>3h = 3 |
| 11-0 | MAINFLASH_SZ | R | x | The encoding of the field is that the value of the field is an integer to be interpreted as number of KBs. For eg: if the value of the field id 4, then it is 4KB, if the value is 32, then 32KB, and so on. |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.4.8 PLLSTARTUP0_4_8MHZ (Offset = 41C4001Ch) [Reset = XXXXXXXXh]

PLLSTARTUP0_4_8MHZ is shown in Figure 1-146 and described in Table 1-164.

Return to the Summary Table.

**Figure 1-146. PLLSTARTUP0_4_8MHZ**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CAPBOVERRIDE | RESERVED | | CAPBVAL | | | | |
| R-Xh | R-0h | | R-Xh | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | CPCURRENT | | | | | |
| R-0h | | R-Xh | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | STARTTIMELP | | | | | |
| R-0h | | R-Xh | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | STARTTIME | | | | | |
| R-0h | | R-Xh | | | | | |

**Table 1-164. PLLSTARTUP0_4_8MHZ Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | CAPBOVERRIDE | R | x | Override Enable For Cap B<br>0h = 0<br>1h = 1 |
| 30-29 | RESERVED | R | 0h | |
| 28-24 | CAPBVAL | R | x | Override Value for Cap B |
| 23-22 | RESERVED | R | 0h | |
| 21-16 | CPCURRENT | R | x | Charge Pump Current |
| 15-14 | RESERVED | R | 0h | |
| 13-8 | STARTTIMELP | R | x | Startup time from Low Power Exit to Locked Clock in resolution of 1usec |
| 7-6 | RESERVED | R | 0h | |
| 5-0 | STARTTIME | R | x | Startup time from Enable to Locked Clock in resolution of 1usec |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.4.9 PLLSTARTUP1_4_8MHZ (Offset = 41C40020h) [Reset = XX0XXXXXh]

PLLSTARTUP1_4_8MHZ is shown in Figure 1-147 and described in Table 1-165.

Return to the Summary Table.

**Figure 1-147. PLLSTARTUP1_4_8MHZ**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LPFRESC | | | | | | | | RESERVED | | | | | | LPFRESA | |
| R-XXh | | | | | | | | R-0h | | | | | | R-XXh | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LPFRESA | | | | | | | | RESERVED | | | LPFCAPA | | | | |
| R-XXh | | | | | | | | R-0h | | | R-Xh | | | | |

**Table 1-165. PLLSTARTUP1_4_8MHZ Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | LPFRESC | R | x | Loop Filter Res C |
| 23-18 | RESERVED | R | 0h | |
| 17-8 | LPFRESA | R | x | Loop Filter Res A |
| 7-5 | RESERVED | R | 0h | |
| 4-0 | LPFCAPA | R | x | Loop Filter Cap A |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.4.10 PLLSTARTUP0_8_16MHZ (Offset = 41C40024h) [Reset = XXXXXXXXh]

PLLSTARTUP0_8_16MHZ is shown in Figure 1-148 and described in Table 1-166.

Return to the Summary Table.

**Figure 1-148. PLLSTARTUP0_8_16MHZ**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CAPBOVERRIDE | RESERVED | | | CAPBVAL | | | |
| R-Xh | R-0h | | | R-Xh | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | CPCURRENT | | | | | |
| R-0h | | R-Xh | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | STARTTIMELP | | | | | |
| R-0h | | R-Xh | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | STARTTIME | | | | | |
| R-0h | | R-Xh | | | | | |

**Table 1-166. PLLSTARTUP0_8_16MHZ Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | CAPBOVERRIDE | R | x | Override Enable For Cap B<br>0h = 0<br>1h = 1 |
| 30-29 | RESERVED | R | 0h | |
| 28-24 | CAPBVAL | R | x | Override Value for Cap B |
| 23-22 | RESERVED | R | 0h | |
| 21-16 | CPCURRENT | R | x | Charge Pump Current |
| 15-14 | RESERVED | R | 0h | |
| 13-8 | STARTTIMELP | R | x | Startup time from Low Power Exit to Locked Clock in resolution of 1usec |
| 7-6 | RESERVED | R | 0h | |
| 5-0 | STARTTIME | R | x | Startup time from Enable to Locked Clock in resolution of 1usec |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.4.11 PLLSTARTUP1_8_16MHZ (Offset = 41C40028h) [Reset = XX0XXXXXh]

PLLSTARTUP1_8_16MHZ is shown in Figure 1-149 and described in Table 1-167.

Return to the Summary Table.

**Figure 1-149. PLLSTARTUP1_8_16MHZ**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LPFRESC | | | | | | | | RESERVED | | | | | | LPFRESA | |
| R-XXh | | | | | | | | R-0h | | | | | | R-XXh | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LPFRESA | | | | | | | | RESERVED | | | LPFCAPA | | | | |
| R-XXh | | | | | | | | R-0h | | | R-Xh | | | | |

**Table 1-167. PLLSTARTUP1_8_16MHZ Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | LPFRESC | R | x | Loop Filter Res C |
| 23-18 | RESERVED | R | 0h | |
| 17-8 | LPFRESA | R | x | Loop Filter Res A |
| 7-5 | RESERVED | R | 0h | |
| 4-0 | LPFCAPA | R | x | Loop Filter Cap A |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.4.12 PLLSTARTUP0_16_32MHZ (Offset = 41C4002Ch) [Reset = XXXXXXXXh]

PLLSTARTUP0_16_32MHZ is shown in Figure 1-150 and described in Table 1-168.

Return to the Summary Table.

**Figure 1-150. PLLSTARTUP0_16_32MHZ**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CAPBOVERRIDE | RESERVED | | | CAPBVAL | | | |
| R-Xh | R-0h | | | R-Xh | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | CPCURRENT | | | | | |
| R-0h | | R-Xh | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | STARTTIMELP | | | | | |
| R-0h | | R-Xh | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | STARTTIME | | | | | |
| R-0h | | R-Xh | | | | | |

**Table 1-168. PLLSTARTUP0_16_32MHZ Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | CAPBOVERRIDE | R | x | Override Enable For Cap B<br>0h = 0<br>1h = 1 |
| 30-29 | RESERVED | R | 0h | |
| 28-24 | CAPBVAL | R | x | Override Value for Cap B |
| 23-22 | RESERVED | R | 0h | |
| 21-16 | CPCURRENT | R | x | Charge Pump Current |
| 15-14 | RESERVED | R | 0h | |
| 13-8 | STARTTIMELP | R | x | Startup time from Low Power Exit to Locked Clock in resolution of 1usec |
| 7-6 | RESERVED | R | 0h | |
| 5-0 | STARTTIME | R | x | Startup time from Enable to Locked Clock in resolution of 1usec |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.4.13 PLLSTARTUP1_16_32MHZ (Offset = 41C40030h) [Reset = XX0XXXXXh]

PLLSTARTUP1_16_32MHZ is shown in Figure 1-151 and described in Table 1-169.

Return to the Summary Table.

**Figure 1-151. PLLSTARTUP1_16_32MHZ**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LPFRESC | | | | | | | | RESERVED | | | | | | LPFRESA | |
| R-XXh | | | | | | | | R-0h | | | | | | R-XXh | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LPFRESA | | | | | | | | RESERVED | | | LPFCAPA | | | | |
| R-XXh | | | | | | | | R-0h | | | R-Xh | | | | |

**Table 1-169. PLLSTARTUP1_16_32MHZ Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | LPFRESC | R | x | Loop Filter Res C |
| 23-18 | RESERVED | R | 0h | |
| 17-8 | LPFRESA | R | x | Loop Filter Res A |
| 7-5 | RESERVED | R | 0h | |
| 4-0 | LPFCAPA | R | x | Loop Filter Cap A |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.4.14 PLLSTARTUP0_32_48MHZ (Offset = 41C40034h) [Reset = XXXXXXXXh]

PLLSTARTUP0_32_48MHZ is shown in Figure 1-152 and described in Table 1-170.

Return to the Summary Table.

**Figure 1-152. PLLSTARTUP0_32_48MHZ**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CAPBOVERRIDE | RESERVED | | | CAPBVAL | | | |
| R-Xh | R-0h | | | R-Xh | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | CPCURRENT | | | | | |
| R-0h | | R-Xh | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | STARTTIMELP | | | | | |
| R-0h | | R-Xh | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | STARTTIME | | | | | |
| R-0h | | R-Xh | | | | | |

**Table 1-170. PLLSTARTUP0_32_48MHZ Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | CAPBOVERRIDE | R | x | Override Enable For Cap B<br>0h = 0<br>1h = 1 |
| 30-29 | RESERVED | R | 0h | |
| 28-24 | CAPBVAL | R | x | Override Value for Cap B |
| 23-22 | RESERVED | R | 0h | |
| 21-16 | CPCURRENT | R | x | Charge Pump Current |
| 15-14 | RESERVED | R | 0h | |
| 13-8 | STARTTIMELP | R | x | Startup time from Low Power Exit to Locked Clock in resolution of 1usec |
| 7-6 | RESERVED | R | 0h | |
| 5-0 | STARTTIME | R | x | Startup time from Enable to Locked Clock in resolution of 1usec |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.4.15 PLLSTARTUP1_32_48MHZ (Offset = 41C40038h) [Reset = XX0XXXXXh]

PLLSTARTUP1_32_48MHZ is shown in Figure 1-153 and described in Table 1-171.

Return to the Summary Table.

**Figure 1-153. PLLSTARTUP1_32_48MHZ**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LPFRESC | | | | | | | | RESERVED | | | | | | LPFRESA | |
| R-XXh | | | | | | | | R-0h | | | | | | R-XXh | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LPFRESA | | | | | | | | RESERVED | | | LPFCAPA | | | | |
| R-XXh | | | | | | | | R-0h | | | R-Xh | | | | |

**Table 1-171. PLLSTARTUP1_32_48MHZ Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | LPFRESC | R | x | Loop Filter Res C |
| 23-18 | RESERVED | R | 0h | |
| 17-8 | LPFRESA | R | x | Loop Filter Res A |
| 7-5 | RESERVED | R | 0h | |
| 4-0 | LPFCAPA | R | x | Loop Filter Cap A |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.4.16 TEMP_SENSE0 (Offset = 41C4003Ch) [Reset = XXXXXXXXh]

TEMP_SENSE0 is shown in Figure 1-154 and described in Table 1-172.

Return to the Summary Table.

Temperature sensor room temperature calibration code. This is ADC conversion results of temperature sensor output voltage. Included in BOOTCRC calculation.

**Figure 1-154. TEMP_SENSE0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-XXXXXXXXh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-172. TEMP_SENSE0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | R | x | |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.4.17 SEED0 (Offset = 41C40040h) [Reset = 00000000h]

SEED0 is shown in Figure 1-155 and described in Table 1-173.

Return to the Summary Table.

Random seed

**Figure 1-155. SEED0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-173. SEED0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | RESERVED | R | 0h | Reserved |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.4.18 SEED1 (Offset = 41C40044h) [Reset = 00000000h]

SEED1 is shown in Figure 1-156 and described in Table 1-174.

Return to the Summary Table.

**Figure 1-156. SEED1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-174. SEED1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | RESERVED | R | 0h | Reserved |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.4.19 SEED2 (Offset = 41C40048h) [Reset = 00000000h]

SEED2 is shown in Figure 1-157 and described in Table 1-175.

Return to the Summary Table.

**Figure 1-157. SEED2**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-175. SEED2 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | RESERVED | R | 0h | Reserved |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.4.20 SEED3 (Offset = 41C4004Ch) [Reset = 00000000h]

SEED3 is shown in Figure 1-158 and described in Table 1-176.

Return to the Summary Table.

**Figure 1-158. SEED3**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-176. SEED3 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | RESERVED | R | 0h | Reserved |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.4.21 SEED4 (Offset = 41C40050h) [Reset = 00000000h]

SEED4 is shown in Figure 1-159 and described in Table 1-177.

Return to the Summary Table.

**Figure 1-159. SEED4**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-177. SEED4 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | RESERVED | R | 0h | Reserved |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.4.22 SEED5 (Offset = 41C40054h) [Reset = 00000000h]

SEED5 is shown in Figure 1-160 and described in Table 1-178.

Return to the Summary Table.

**Figure 1-160. SEED5**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-178. SEED5 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | RESERVED | R | 0h | Reserved |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.4.23 SEED6 (Offset = 41C40058h) [Reset = 00000000h]

SEED6 is shown in Figure 1-161 and described in Table 1-179.

Return to the Summary Table.

**Figure 1-161. SEED6**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-179. SEED6 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | RESERVED | R | 0h | Reserved |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.4.24 SEED7 (Offset = 41C4005Ch) [Reset = 00000000h]

SEED7 is shown in Figure 1-162 and described in Table 1-180.

Return to the Summary Table.

**Figure 1-162. SEED7**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-180. SEED7 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | RESERVED | R | 0h | Reserved |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.4.25 TEMP_SENSE_0KELVIN (Offset = 41C40078h) [Reset = 00000000h]

TEMP_SENSE_0KELVIN is shown in Figure 1-163 and described in Table 1-181.

Return to the Summary Table.

**Figure 1-163. TEMP_SENSE_0KELVIN**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-181. TEMP_SENSE_0KELVIN Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | RESERVED | R | 0h | Reserved |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.4.26 BOOTCRC (Offset = 41C4007Ch) [Reset = XXXXXXXXh]

BOOTCRC is shown in Figure 1-164 and described in Table 1-182.

Return to the Summary Table.

BOOTCRC records the 32-bit CRC of all locations in OPEN including reserved locations.

**Figure 1-164. BOOTCRC**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-XXXXXXXXh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-182. BOOTCRC Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | R | x | |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.5 FACTORYREGION_TYPEG Registers

Table 1-183 lists the memory-mapped registers for the FACTORYREGION_TYPEG registers. All register offset addresses not listed in Table 1-183 should be considered as reserved locations and the register contents should not be modified.

**Table 1-183. FACTORYREGION_TYPEG Registers**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 41C4000 0h | TRACEID | Defined by TI, during ATE, based on wafer | | Go |
| 41C4000 4h | DEVICEID | Device identifier | | Go |
| 41C4000 8h | USERID | Device variant identifier | | Go |
| 41C4000 Ch | BSLPIN_UART | BSL UART Pin Configuration | | Go |
| 41C4001 0h | BSLPIN_I2C | BSL I2C Pin Configuration | | Go |
| 41C4001 4h | BSLPIN_INVOKE | BSL Pin Invocation Configuration | | Go |
| 41C4001 8h | SRAMFLASH | | | Go |
| 41C4001 Ch | PLLSTARTUP0_4_8MHZ | | | Go |
| 41C4002 0h | PLLSTARTUP1_4_8MHZ | System PLL Paramater 1 MMR --- Data from Flash Table Lookup | | Go |
| 41C4002 4h | PLLSTARTUP0_8_16MHZ | | | Go |
| 41C4002 8h | PLLSTARTUP1_8_16MHZ | System PLL Paramater 1 MMR --- Data from Flash Table Lookup | | Go |
| 41C4002 Ch | PLLSTARTUP0_16_32MHZ | | | Go |
| 41C4003 0h | PLLSTARTUP1_16_32MHZ | System PLL Paramater 1 MMR --- Data from Flash Table Lookup | | Go |
| 41C4003 4h | PLLSTARTUP0_32_48MHZ | | | Go |
| 41C4003 8h | PLLSTARTUP1_32_48MHZ | System PLL Paramater 1 MMR --- Data from Flash Table Lookup | | Go |
| 41C4003 Ch | TEMP_SENSE0 | Temperature sensor room temperature calibration code.<br>This is ADC conversion results of temperature sensor output voltage.<br>Included in BOOTCRC calculation. | | Go |
| 41C4004 0h | BOOTCRC | BOOTCRC records the 32-bit CRC of all locations in OPEN including reserved locations. | | |
| 41C4004 4h | RESERVED0 | | | |
| 41C4004 8h | RESERVED1 | | | |
| 41C4004 Ch | RESERVED2 | | | |
| 41C4005 0h | RESERVED3 | | | |
| 41C4005 4h | RESERVED4 | | | |



Copyright © 2026 Texas Instruments Incorporated

**Table 1-183. FACTORYREGION_TYPEG Registers (continued)**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 41C4005 8h | RESERVED5 | | | |
| 41C4005 Ch | RESERVED6 | | | |
| 41C4006 0h | RESERVED7 | | | |
| 41C4006 4h | RESERVED8 | | | |
| 41C4006 8h | RESERVED9 | | | |
| 41C4006 Ch | RESERVED10 | | | |
| 41C4007 0h | RESERVED11 | | | |
| 41C4007 4h | RESERVED12 | | | |
| 41C4007 8h | RESERVED13 | | | |
| 41C4007 Ch | RESERVED14 | | | |

Complex bit access types are encoded to fit into small table cells. Table 1-184 shows the codes that are used for access types in this section.

**Table 1-184. FACTORYREGION_TYPEG Access Type Codes**

| Access Type | Code | Description |
|---|---|---|
| Read Type | | |
| R | R | Read |
| Reset or Default Value | | |
| *-n* | | Value after reset or the default value |



Copyright © 2026 Texas Instruments Incorporated

#### 1.5.5.1 TRACEID (Offset = 41C40000h) [Reset = XXXXXXXXh]

TRACEID is shown in Figure 1-165 and described in Table 1-185.

Return to the Summary Table.

unique per part shipped, done per established TI process

**Figure 1-165. TRACEID**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-XXXXXXXXh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-185. TRACEID Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | R | x | |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.5.2 DEVICEID (Offset = 41C40004h) [Reset = XXXXXXXXh]

DEVICEID is shown in Figure 1-166 and described in Table 1-186.

Return to the Summary Table.

Device identifier (die revision specific). Refer to device factory constants section of the datasheet

**Figure 1-166. DEVICEID**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| VERSION | | | | PARTNUM | | | |
| R-Xh | | | | R-XXXXh | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| PARTNUM | | | | | | | |
| R-XXXXh | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| PARTNUM | | | | MANUFACTURER | | | |
| R-XXXXh | | | | R-XXh | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MANUFACTURER | | | | | | | ALWAYS_1 |
| R-XXh | | | | | | | R-Xh |

**Table 1-186. DEVICEID Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-28 | VERSION | R | x | Revision of the device. This field should change each time that the logic or mask set of the device is revised. |
| 27-12 | PARTNUM | R | x | Part number of the device. |
| 11-1 | MANUFACTURER | R | x | TI's JEDEC bank and company code, which is: 00000010111b |
| 0 | ALWAYS_1 | R | x | This is always 1 |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.5.3 USERID (Offset = 41C40008h) [Reset = XXXXXXXXh]

USERID is shown in Figure 1-167 and described in Table 1-187.

Return to the Summary Table.

Defines the device variant feature-set. Refer to device factory constants section of the datasheet

**Figure 1-167. USERID**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| START | MAJORREV | | | MINORREV | | | |
| R-X | R-X | | | R-X | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| VARIANT | | | | | | | |
| R-X | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| PART | | | | | | | |
| R-X | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PART | | | | | | | |
| R-X | | | | | | | |

**Table 1-187. USERID Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | START | R | 1h | |
| 30-28 | MAJORREV | R | X | Monotonic increasing value indicating a new revision of the SKU significant enough that users of the device may have to revise PCB or or software design<br>x |
| 27-24 | MINORREV | R | X | Monotonic increasing value indicating a new revision of the SKU that preserves compatibility with lesser minorrev values. New capability may be introduced such that lesser minorrev numbers may not be compatible with greater if the new capability is used.<br>x |
| 23-16 | VARIANT | R | X | Bit pattern uniquely identifying a variant of a part. This is used to indicate memory or package variations of the same part number. This number shall be selected at random among the remaining numbers for the relevant combination of IDCODE.device and USERCODE.part such that the order of creation cannot be inferred by the number. The variant number does not encode specifics of the variant directly.<br>x |
| 15-0 | PART | R | X | Bit pattern that uniquely identifying a part. This is used to identify the specific part based on the die identified in DEVICEID.device. This number shall be selected at random among the remaining numbers for DEVICEID.device such that the order of creation cannot be inferred by the number. This value does not encode the part number directly.<br>x |



Copyright © 2026 Texas Instruments Incorporated

#### 1.5.5.4 BSLPIN_UART (Offset = 41C4000Ch) [Reset = XXXXXXXXh]

BSLPIN_UART is shown in Figure 1-168 and described in Table 1-188.

Return to the Summary Table.

BSL UART Pin Configuration

**Figure 1-168. BSLPIN_UART**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| UART_TXD_PF | | | | | | | | UART_TXD_PAD | | | | | | | |
| R-XXh | | | | | | | | R-XXh | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| UART_RXD_PF | | | | | | | | UART_RXD_PAD | | | | | | | |
| R-XXh | | | | | | | | R-XXh | | | | | | | |

**Table 1-188. BSLPIN_UART Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | UART_TXD_PF | R | x | UART TXD Pin Function Selection Value |
| 23-16 | UART_TXD_PAD | R | x | UART TXD Pin used by BSL |
| 15-8 | UART_RXD_PF | R | x | UART RXD Pin Function Selection Value |
| 7-0 | UART_RXD_PAD | R | x | UART RXD Pad used by BSL |



Copyright © 2026 Texas Instruments Incorporated

#### 1.5.5.5 BSLPIN_I2C (Offset = 41C40010h) [Reset = XXXXXXXXh]

BSLPIN_I2C is shown in Figure 1-169 and described in Table 1-189.

Return to the Summary Table.

BSL I2C Pin Configuration

**Figure 1-169. BSLPIN_I2C**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| I2C_SCL_PF | | | | | | | | I2C_SCL_PAD | | | | | | | |
| R-XXh | | | | | | | | R-XXh | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| I2C_SDA_PF | | | | | | | | I2C_SDA_PAD | | | | | | | |
| R-XXh | | | | | | | | R-XXh | | | | | | | |

**Table 1-189. BSLPIN_I2C Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | I2C_SCL_PF | R | x | I2C SCL Pin Function Selection Value |
| 23-16 | I2C_SCL_PAD | R | x | I2C SCL Pin used by BSL |
| 15-8 | I2C_SDA_PF | R | x | I2C SDA Pin Function Selection Value |
| 7-0 | I2C_SDA_PAD | R | x | I2C SDA Pin used by BSL |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.5.6 BSLPIN_INVOKE (Offset = 41C40014h) [Reset = 0000XXXXh]

BSLPIN_INVOKE is shown in Figure 1-170 and described in Table 1-190.

Return to the Summary Table.

BSL Pin Invocation Configuration

**Figure 1-170. BSLPIN_INVOKE**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | GPIO_REG_SEL | | GPIO_PIN_SEL | | | | |
| R-0h | R-Xh | | R-Xh | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GPIO_LEVEL | BSL_PAD | | | | | | |
| R-Xh | R-Xh | | | | | | |

**Table 1-190. BSLPIN_INVOKE Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-15 | RESERVED | R | 0h | |
| 14-13 | GPIO_REG_SEL | R | x | GPIO Module Selection |
| 12-8 | GPIO_PIN_SEL | R | x | GPIO Pin Number in GPIO Module |
| 7 | GPIO_LEVEL | R | x | GPIO Level Configuration for BSL Invocation |
| 6-0 | BSL_PAD | R | x | BSL Invocation Pin Number |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.5.7 SRAMFLASH (Offset = 41C40018h) [Reset = XXXXXXXXh]

SRAMFLASH is shown in Figure 1-171 and described in Table 1-191.

Return to the Summary Table.

**Figure 1-171. SRAMFLASH**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DATAFLASH_SZ | | | | | | SRAM_SZ | |
| R-Xh | | | | | | R-XXh | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| SRAM_SZ | | | | | | | |
| R-XXh | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | MAINNUMBANKS | | MAINFLASH_SZ | | | |
| R-0h | | R-Xh | | R-XXXh | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MAINFLASH_SZ | | | | | | | |
| R-XXXh | | | | | | | |

**Table 1-191. SRAMFLASH Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | DATAFLASH_SZ | R | x | The encoding of the field is that the value of the field is an integer to be interpreted as number of KBs.<br>For eg: if the value of the field id 4, then it is 4KB, if the value is 32, then 32KB, and so on. |
| 25-16 | SRAM_SZ | R | x | The encoding of the field is that the value of the field is an integer to be interpreted as number of KBs. For eg: if the value of the field id 4, then it is 4KB, if the value is 32, then 32KB, and so on. |
| 15-14 | RESERVED | R | 0h | |
| 13-12 | MAINNUMBANKS | R | x | Value of 0 means there is a single flash bank; value of 1 means there are two flash banks. Etc.<br>0h = 0<br>1h = 1<br>2h = 2<br>3h = 3 |
| 11-0 | MAINFLASH_SZ | R | x | The encoding of the field is that the value of the field is an integer to be interpreted as number of KBs. For eg: if the value of the field id 4, then it is 4KB, if the value is 32, then 32KB, and so on. |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.5.8 PLLSTARTUP0_4_8MHZ (Offset = 41C4001Ch) [Reset = XXXXXXXXh]

PLLSTARTUP0_4_8MHZ is shown in Figure 1-172 and described in Table 1-192.

Return to the Summary Table.

**Figure 1-172. PLLSTARTUP0_4_8MHZ**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CAPBOVERRIDE | RESERVED | | CAPBVAL | | | | |
| R-Xh | R-0h | | R-Xh | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | CPCURRENT | | | | | |
| R-0h | | R-Xh | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | STARTTIMELP | | | | | |
| R-0h | | R-Xh | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | STARTTIME | | | | | |
| R-0h | | R-Xh | | | | | |

**Table 1-192. PLLSTARTUP0_4_8MHZ Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | CAPBOVERRIDE | R | x | Override Enable For Cap B<br>0h = 0<br>1h = 1 |
| 30-29 | RESERVED | R | 0h | |
| 28-24 | CAPBVAL | R | x | Override Value for Cap B |
| 23-22 | RESERVED | R | 0h | |
| 21-16 | CPCURRENT | R | x | Charge Pump Current |
| 15-14 | RESERVED | R | 0h | |
| 13-8 | STARTTIMELP | R | x | Startup time from Low Power Exit to Locked Clock in resolution of 1usec |
| 7-6 | RESERVED | R | 0h | |
| 5-0 | STARTTIME | R | x | Startup time from Enable to Locked Clock in resolution of 1usec |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.5.9 PLLSTARTUP1_4_8MHZ (Offset = 41C40020h) [Reset = XX0XXXXXh]

PLLSTARTUP1_4_8MHZ is shown in Figure 1-173 and described in Table 1-193.

Return to the Summary Table.

**Figure 1-173. PLLSTARTUP1_4_8MHZ**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LPFRESC | | | | | | | | RESERVED | | | | | | LPFRESA | |
| R-XXh | | | | | | | | R-0h | | | | | | R-XXh | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LPFRESA | | | | | | | | RESERVED | | | LPFCAPA | | | | |
| R-XXh | | | | | | | | R-0h | | | R-Xh | | | | |

**Table 1-193. PLLSTARTUP1_4_8MHZ Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | LPFRESC | R | x | Loop Filter Res C |
| 23-18 | RESERVED | R | 0h | |
| 17-8 | LPFRESA | R | x | Loop Filter Res A |
| 7-5 | RESERVED | R | 0h | |
| 4-0 | LPFCAPA | R | x | Loop Filter Cap A |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.5.10 PLLSTARTUP0_8_16MHZ (Offset = 41C40024h) [Reset = XXXXXXXXh]

PLLSTARTUP0_8_16MHZ is shown in Figure 1-174 and described in Table 1-194.

Return to the Summary Table.

**Figure 1-174. PLLSTARTUP0_8_16MHZ**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CAPBOVERRIDE | RESERVED | | | CAPBVAL | | | |
| R-Xh | R-0h | | | R-Xh | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | CPCURRENT | | | | | |
| R-0h | | R-Xh | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | STARTTIMELP | | | | | |
| R-0h | | R-Xh | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | STARTTIME | | | | | |
| R-0h | | R-Xh | | | | | |

**Table 1-194. PLLSTARTUP0_8_16MHZ Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | CAPBOVERRIDE | R | x | Override Enable For Cap B<br>0h = 0<br>1h = 1 |
| 30-29 | RESERVED | R | 0h | |
| 28-24 | CAPBVAL | R | x | Override Value for Cap B |
| 23-22 | RESERVED | R | 0h | |
| 21-16 | CPCURRENT | R | x | Charge Pump Current |
| 15-14 | RESERVED | R | 0h | |
| 13-8 | STARTTIMELP | R | x | Startup time from Low Power Exit to Locked Clock in resolution of 1usec |
| 7-6 | RESERVED | R | 0h | |
| 5-0 | STARTTIME | R | x | Startup time from Enable to Locked Clock in resolution of 1usec |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.5.11 PLLSTARTUP1_8_16MHZ (Offset = 41C40028h) [Reset = XX0XXXXXh]

PLLSTARTUP1_8_16MHZ is shown in Figure 1-175 and described in Table 1-195.

Return to the Summary Table.

**Figure 1-175. PLLSTARTUP1_8_16MHZ**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LPFRESC | | | | | | | | RESERVED | | | | | | LPFRESA | |
| R-XXh | | | | | | | | R-0h | | | | | | R-XXh | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LPFRESA | | | | | | | | RESERVED | | | LPFCAPA | | | | |
| R-XXh | | | | | | | | R-0h | | | R-Xh | | | | |

**Table 1-195. PLLSTARTUP1_8_16MHZ Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | LPFRESC | R | x | Loop Filter Res C |
| 23-18 | RESERVED | R | 0h | |
| 17-8 | LPFRESA | R | x | Loop Filter Res A |
| 7-5 | RESERVED | R | 0h | |
| 4-0 | LPFCAPA | R | x | Loop Filter Cap A |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.5.12 PLLSTARTUP0_16_32MHZ (Offset = 41C4002Ch) [Reset = XXXXXXXXh]

PLLSTARTUP0_16_32MHZ is shown in Figure 1-176 and described in Table 1-196.

Return to the Summary Table.

**Figure 1-176. PLLSTARTUP0_16_32MHZ**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CAPBOVERRIDE | RESERVED | | | CAPBVAL | | | |
| R-Xh | R-0h | | | R-Xh | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | CPCURRENT | | | | | |
| R-0h | | R-Xh | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | STARTTIMELP | | | | | |
| R-0h | | R-Xh | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | STARTTIME | | | | | |
| R-0h | | R-Xh | | | | | |

**Table 1-196. PLLSTARTUP0_16_32MHZ Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | CAPBOVERRIDE | R | x | Override Enable For Cap B<br>0h = 0<br>1h = 1 |
| 30-29 | RESERVED | R | 0h | |
| 28-24 | CAPBVAL | R | x | Override Value for Cap B |
| 23-22 | RESERVED | R | 0h | |
| 21-16 | CPCURRENT | R | x | Charge Pump Current |
| 15-14 | RESERVED | R | 0h | |
| 13-8 | STARTTIMELP | R | x | Startup time from Low Power Exit to Locked Clock in resolution of 1usec |
| 7-6 | RESERVED | R | 0h | |
| 5-0 | STARTTIME | R | x | Startup time from Enable to Locked Clock in resolution of 1usec |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.5.13 PLLSTARTUP1_16_32MHZ (Offset = 41C40030h) [Reset = XX0XXXXXh]

PLLSTARTUP1_16_32MHZ is shown in Figure 1-177 and described in Table 1-197.

Return to the Summary Table.

**Figure 1-177. PLLSTARTUP1_16_32MHZ**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LPFRESC | | | | | | | | RESERVED | | | | | | LPFRESA | |
| R-XXh | | | | | | | | R-0h | | | | | | R-XXh | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LPFRESA | | | | | | | | RESERVED | | | LPFCAPA | | | | |
| R-XXh | | | | | | | | R-0h | | | R-Xh | | | | |

**Table 1-197. PLLSTARTUP1_16_32MHZ Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | LPFRESC | R | x | Loop Filter Res C |
| 23-18 | RESERVED | R | 0h | |
| 17-8 | LPFRESA | R | x | Loop Filter Res A |
| 7-5 | RESERVED | R | 0h | |
| 4-0 | LPFCAPA | R | x | Loop Filter Cap A |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.5.14 PLLSTARTUP0_32_48MHZ (Offset = 41C40034h) [Reset = XXXXXXXXh]

PLLSTARTUP0_32_48MHZ is shown in Figure 1-178 and described in Table 1-198.

Return to the Summary Table.

**Figure 1-178. PLLSTARTUP0_32_48MHZ**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CAPBOVERRIDE | RESERVED | | | CAPBVAL | | | |
| R-Xh | R-0h | | | R-Xh | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | CPCURRENT | | | | | |
| R-0h | | R-Xh | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | STARTTIMELP | | | | | |
| R-0h | | R-Xh | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | STARTTIME | | | | | |
| R-0h | | R-Xh | | | | | |

**Table 1-198. PLLSTARTUP0_32_48MHZ Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | CAPBOVERRIDE | R | x | Override Enable For Cap B<br>0h = 0<br>1h = 1 |
| 30-29 | RESERVED | R | 0h | |
| 28-24 | CAPBVAL | R | x | Override Value for Cap B |
| 23-22 | RESERVED | R | 0h | |
| 21-16 | CPCURRENT | R | x | Charge Pump Current |
| 15-14 | RESERVED | R | 0h | |
| 13-8 | STARTTIMELP | R | x | Startup time from Low Power Exit to Locked Clock in resolution of 1usec |
| 7-6 | RESERVED | R | 0h | |
| 5-0 | STARTTIME | R | x | Startup time from Enable to Locked Clock in resolution of 1usec |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.5.15 PLLSTARTUP1_32_48MHZ (Offset = 41C40038h) [Reset = XX0XXXXXh]

PLLSTARTUP1_32_48MHZ is shown in Figure 1-179 and described in Table 1-199.

Return to the Summary Table.

**Figure 1-179. PLLSTARTUP1_32_48MHZ**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LPFRESC | | | | | | | | RESERVED | | | | | | LPFRESA | |
| R-XXh | | | | | | | | R-0h | | | | | | R-XXh | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LPFRESA | | | | | | | | RESERVED | | | LPFCAPA | | | | |
| R-XXh | | | | | | | | R-0h | | | R-Xh | | | | |

**Table 1-199. PLLSTARTUP1_32_48MHZ Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | LPFRESC | R | x | Loop Filter Res C |
| 23-18 | RESERVED | R | 0h | |
| 17-8 | LPFRESA | R | x | Loop Filter Res A |
| 7-5 | RESERVED | R | 0h | |
| 4-0 | LPFCAPA | R | x | Loop Filter Cap A |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.5.16 TEMP_SENSE0 (Offset = 41C4003Ch) [Reset = XXXXXXXXh]

TEMP_SENSE0 is shown in Figure 1-180 and described in Table 1-200.

Return to the Summary Table.

Temperature sensor room temperature calibration code. This is ADC conversion results of temperature sensor output voltage. Included in BOOTCRC calculation.

**Figure 1-180. TEMP_SENSE0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-XXXXXXXXh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-200. TEMP_SENSE0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | R | x | |



Copyright © 2026 Texas Instruments Incorporated

### 1.5.5.17 BOOTCRC (Offset = 41C40040h) [Reset = XXXXXXXXh]

BOOTCRC is shown in Figure 1-181 and described in Table 1-201.

Return to the Summary Table.

BOOTCRC records the 32-bit CRC of all locations in OPEN including reserved locations.

**Figure 1-181. BOOTCRC**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-XXXXXXXXh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 1-201. BOOTCRC Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | R | x | |



Copyright © 2026 Texas Instruments Incorporated

Chapter 2

# PMCU

The power management and clock unit (PMCU) is a unified system module which provides all power management, clock configuration, and reset control functionality for the device. All power management unit (PMU) and clock module (CKM) policies for device operation are configured through memory-mapped registers in the system controller (SYSCTL).


**2.1 PMCU Overview** ..... 229
**2.2 Power Management (PMU)** ..... 234
**2.3 Clock Module (CKM)** ..... 240
**2.4 System Controller (SYSCTL)** ..... 269
**2.5 SYSCTL_G351x_G151x_G321x_G320x_G121x_G120x_G122x_G322x Registers** ..... 287
**2.6 SYSCTL_G351x_G151x Registers** ..... 347
**2.7 SYSCTL_G511x_G5187 Registers** ..... 407
**2.8 Quick Start Reference** ..... 466





Copyright © 2026 Texas Instruments Incorporated

## 2.1 PMCU Overview

The power management and clock unit (PMCU) provides all power, clocking, reset, and system control services for the device. The PMCU contains three submodules to provide this functionality: the power management unit (**PMU**), the clock module (**CKM**), and the system controller (**SYSCTL**).

The PMU is an analog submodule that generates the internal regulated supplies for the device and supervises the condition of the external supply. The PMU also contains voltage and current reference circuits used by the on-chip regulators and analog peripherals.

The CKM is an analog submodule that provides clock sources (internal and external oscillators) and presents these clock sources to SYSCTL. SYSCTL distributes these clock sources to the CPU, buses, and peripherals on the device.

The SYSCTL is a digital submodule that provides the control logic for all functions in the PMCU. In addition, SYSCTL contains the memory-mapped registers used by software to configure power management and clocks, assess the status of the device, and control resets. SYSCTL also provides general-purpose memory that is retained in SHUTDOWN mode and can be used to store status information in SHUTDOWN mode when SRAM and register contents are lost.

Figure 2-1 shows the interfaces between the PMCU and the device supplies, clocks, and signals. Configuration of the PMCU by software is always done through memory-mapped registers in the SYSCTL submodule.

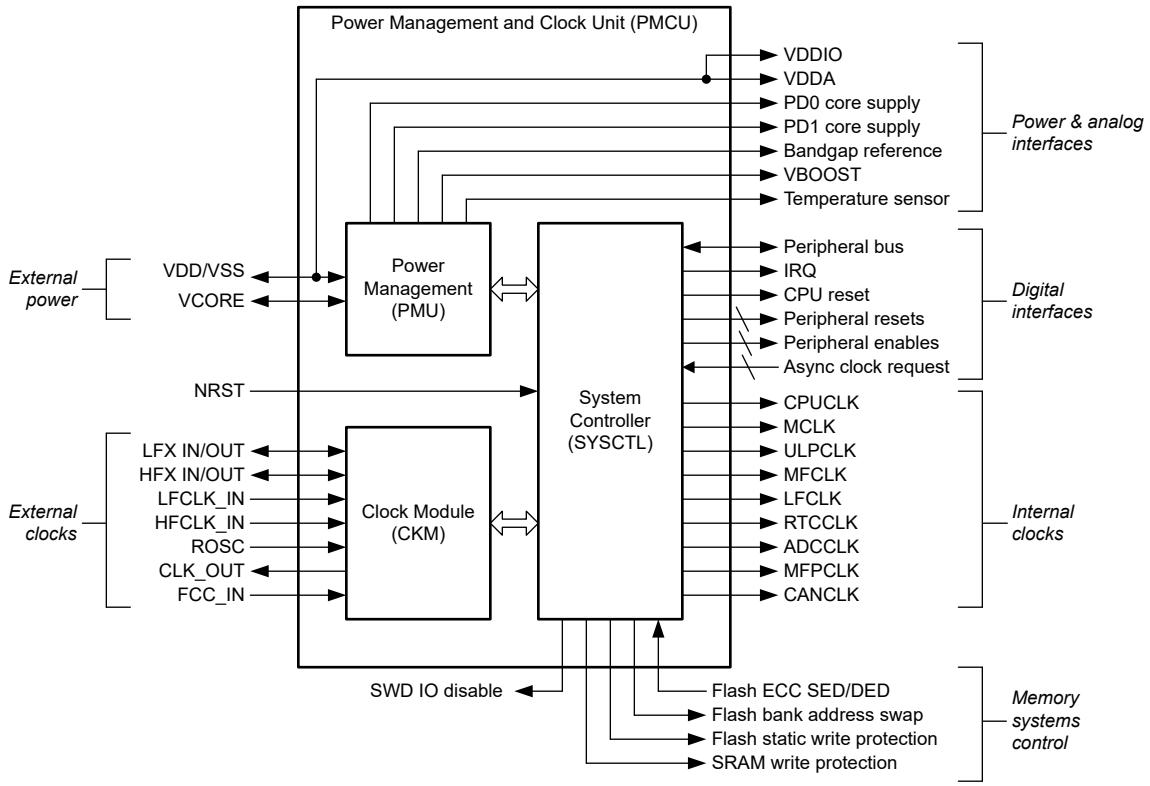



**Figure 2-1. PMCU Top-Level Diagram**

**Note**

Not all devices have all of the PMCU features shown in Figure 2-1. For example, not all devices have LFXT (low-frequency crystal oscillator), HFXT (high-frequency crystal oscillator) , and CANCLK. See the device-specific data sheet to understand the features present on a given device.

### Using this Guide



Copyright © 2026 Texas Instruments Incorporated

The PMU, CKM, and SYSCTL sections of this chapter describe the functionality provided by each submodule in detail.

The quick start section describes overall system level operation of the PMCU and how to configure the PMCU for different application scenarios.

### 2.1.1 Power Domains

Two core power domains are provided on the device: PD1 and PD0. PD1 is always powered in RUN and SLEEP modes, but is disabled in all other modes. PD0 is always powered in RUN, SLEEP, STOP and STANDBY modes. PD1 and PD0 are both disabled in SHUTDOWN mode.

- The PD1 domain includes the CPU subsystem, the SRAM memory, PD1 peripherals, and the PD1 peripheral bus, which runs from MCLK (including the DMA). While PD1 is disabled in STOP and STANDBY mode, the CPU registers, SRAM, and peripheral configuration registers are maintained in retention such that they are available to resume operation immediately when STOP or STANDBY modes are exited.
- The PD0 domain includes the PD0 peripherals and PD0 bus segment which runs from ULPCLK in RUN and SLEEP mode, 4MHz in STOP mode, and 32kHz in STANDBY mode. The PD0 domain is powered in all modes except SHUTDOWN and can be thought of as an "always-on" domain.

The device-specific data sheet describes which peripherals on a device are in PD1 and which are in PD0.

The device also has a primary external supply (VDD) domain that provides power to the IO and analog peripherals.

The device also has a primary external supply (VDD) domain that provides power to the IO and analog peripherals.

### 2.1.2 Operating Modes

Various operating modes (power modes) are provided to allow for optimization of the device power consumption based on application requirements. In order of decreasing power, the modes are: RUN, SLEEP, STOP, STANDBY and SHUTDOWN. The following figure shows the interaction between the modes.

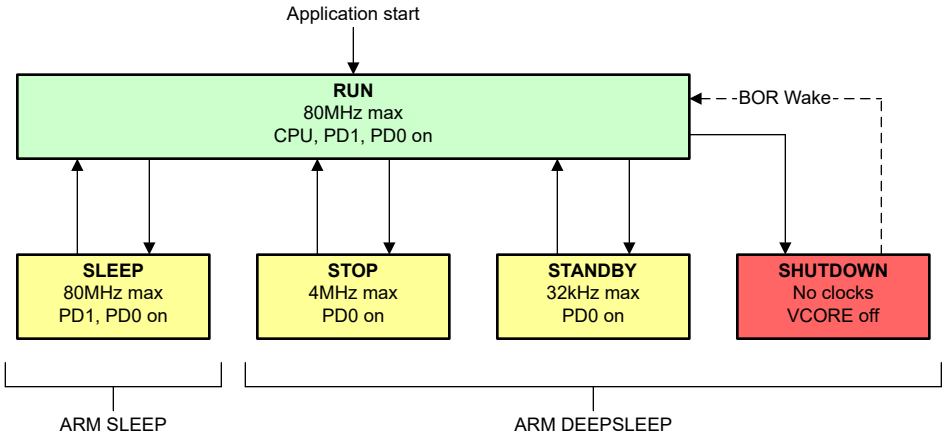



**Figure 2-2. MSPM0Gxx Operating Modes**

Section 2.1.2.6 indicates what functions are available in each operating mode of the device. See the operating mode selection section for information on how to configure the device for a particular operating mode.

#### Operating Mode Concept

MSPM0 MCUs implement a policy-based power and clock management scheme. Policies can be configured through application software for how the clocking is to be managed in each operating mode to obtain the best balance of power and performance for a given application.



Copyright © 2026 Texas Instruments Incorporated

After the operating policy for each mode is configured, application software can enter and exit the various operating modes through simple register commands, and SYSCTL automatically manages all the necessary PMU states, oscillator and clock enable and disables, and the SYSOSC frequency according to the software-defined policies and software-selected mode.

A variety of hardware-triggered low-power mode suspension mechanisms also exist to enable on-demand access to a fast clock when requested by supported peripherals, as well as functions such as DMA and ADC triggering from low-power modes.

The policy-driven operating mode scheme together with the asynchronous low-power mode suspension mechanisms enable application software to select the operating mode and corresponding policy that provide the lowest possible power consumption for background activities, with transient foreground activities either bringing up the DMA, bringing up a fast clock, or bringing the device to RUN (in the case of an IRQ) for burst handling.

#### 2.1.2.1 RUN Mode

In RUN mode, the CPU is active executing code and any peripheral can be enabled.

There are three RUN mode policy options: RUN0, RUN1, and RUN2.

- **RUN0**: The MCLK and the CPUCLK run from a fast clock source (SYSOSC, HFCLK, or SYSPLL).
- **RUN1**: The MCLK and the CPUCLK run from LFCLK (at 32kHz) to reduce active power, but SYSOSC is left enabled to service analog modules such as an ADC.
- **RUN2**: The MCLK and the CPUCLK run from LFCLK (at 32kHz), and SYSOSC is completely disabled to save power. This is the lowest power state with the CPU running.

**Note**

SYSPLL is not automatically disabled, and needs to be manually disabled through the HSCLKEN.SYSPLLEN field within the SYSCTL registers in order to reduce power consumption

#### 2.1.2.2 SLEEP Mode

In SLEEP mode, the CPU is disabled (clock gated) but otherwise the device configuration is the same as RUN. There are three SLEEP mode policy options: SLEEP0, SLEEP1, and SLEEP2. The SLEEPx policy is determined by the current RUNx policy when SLEEP mode is entered.

- **SLEEP0**: Identical to RUN0, with the CPU disabled.
- **SLEEP1**: Identical to RUN1, with the CPU disabled.
- **SLEEP2**: Identical to RUN2, with the CPU disabled.

**Note**

SYSPLL is not automatically disabled, and needs to be manually disabled through the HSCLKEN.SYSPLLEN field within the SYSCTL registers in order to reduce power consumption

#### 2.1.2.3 STOP Mode

In STOP mode, the CPU, SRAM, and PD1 peripherals are disabled and in retention (if applicable). PD0 peripherals are available with a max ULPCLK frequency of 4MHz. SYSOSC can run at higher frequencies to support operation of analog peripheral (such as an ADC), but ULPCLK will be automatically limited to the 4MHz SYSOSC output by SYSCTL. For details on peripheral availability, please refer to the "supported functionality by operating mode" table within the device-specific datasheet.

**Note**

SYSPLL is not automatically disabled, and needs to be manually disabled through the HSCLKEN.SYSPLLEN field within the SYSCTL registers in order to reduce power consumption

DMA is available to be triggered. A DMA trigger wakes the PD1 power domain to make the SRAM and DMA available for processing the DMA transfer, and the DMA transfer is processed at the current MCLK rate. After the transfer completes, the SRAM is returned to retention and PD1 is disabled automatically.



Copyright © 2026 Texas Instruments Incorporated

STOP mode is the lowest power mode that supports the operation of analog peripherals.

The policy options for STOP mode are: STOP0, STOP1, and STOP2.

- **STOP0**: The SYSOSC is left running at the current frequency when entering STOP mode. ULPCLK is always limited to 4MHz automatically by hardware, but SYSOSC is not disturbed to support consistent operation of analog peripherals.
  - NOTE: If STOP0 is entered from RUN1 (SYSOSC enabled but MCLK sourced from LFCLK), SYSOSC remains enabled as in RUN1, and ULPCLK remains at 32kHz as in RUN1.
  - NOTE: If STOP0 is entered from RUN2 (SYSOSC was disabled and MCLK was sourced from LFCLK), SYSOSC remains disabled as in RUN2, and ULPCLK remains at 32kHz as in RUN2.
- **STOP1**: The SYSOSC is gear shifted from its current frequency to 4MHz for the lowest power consumption in STOP mode with SYSOSC running. SYSOSC and ULPCLK both run at 4MHz.
- **STOP2**: The SYSOSC is disabled and the ULPCLK is sourced from LFCLK at 32kHz. This is the lowest power state in STOP mode.

#### 2.1.2.4 STANDBY Mode

In STANDBY mode, the CPU, SRAM, and PD1 peripherals are disabled and in retention. PD0 peripherals, with the exception of the ADC, DAC, and OPA are available with a maximum ULPCLK frequency of 32kHz. . High-speed oscillators such as (SYSPLL, HFXT, HFCLK_IN) and SYSOSC are disabled. Wakeup sources includes digital peripherals like $I^2C$, UART start detection, PD0 timers elapsing, RTC wakeup, GPIO toggle.

**Note**

SYSPLL is not automatically disabled, and needs to be manually disabled through the HSCLKEN.SYSPLLEN field within the SYSCTL registers in order to reduce power consumption

STANDBY mode supports asynchronous fast clock request to switch to STOP mode and back to STANDBY. In the case of an event generating an interrupt, the device continues to RUN mode.

DMA is available to be triggered. A DMA trigger wakes the PD1 power domain to make the SRAM and DMA available for processing the DMA transfer, and the DMA transfer is processed at the current MCLK. After the transfer completes, the SRAM is returned to retention and PD1 is disabled automatically.

Analog peripherals (such as an ADC, 12-bit DAC, OPA, and high-speed COMP) are not supported in STANDBY mode. For details on peripheral availability, please refer to the "supported functionality by operating mode" table within the device-specific datasheet.

There are 2 policy options for STANDBY mode: STANDBY0 and STANDBY1.

- **STANDBY0**: All PD0 peripherals receive the ULPCLK and LFCLK, while the RTC receives the RTCCLK.
- **STANDBY1**: Only a few general purpose timers receive ULPCLK or LFCLK, see the device-specific data sheet to determine them. If the device contains an RTC module, it continues to receive RTCCLK. A general timer interrupt, RTC interrupt, or ADC trigger in STANDBY1 always triggers an asynchronous fast clock request to wake the system. Other PD0 peripherals (such as UART, I2C, GPIO, and low-power COMP) can also wake the system upon an external event through an asynchronous fast clock request, but they are not actively clocked in STANDBY1.

#### 2.1.2.5 SHUTDOWN Mode

In SHUTDOWN mode, no clocks are available. The core regulator is completely disabled and all SRAM and register contents are lost, with the exception of the general-purpose memory in SYSCTL that can be used to store state information. The BOR and bandgap circuit are disabled.

Specific IO configuration settings are retained in the SHUTDOWN operating mode.

- Drive strength
- Pullup enable
- Pulldown enable



Copyright © 2026 Texas Instruments Incorporated

- Input enable
- Output enable
- Output state

The device can wake through a wake-up capable IO, a debug connection, or NRST.

SHUTDOWN mode has the lowest current consumption of any operating mode. Exiting SHUTDOWN mode triggers a BOR.

#### 2.1.2.6 Supported Functionality by Operating Mode

Supported functionality in each operating mode is given in *Supported Functionality by Operating Mode* table in the detailed description section of the device-specific data sheet.

#### 2.1.2.7 Suspended Low-Power Mode Operation

Some peripherals can be configured to temporarily suspend STOP or STANDBY mode operation to handle a temporary activity or process an event. There are two ways in which STOP or STANDBY mode can be suspended:

- An asynchronous fast clock request
- A DMA trigger

**Suspended STOP or STANDBY for an Asynchronous Fast Clock Request**

An asynchronous fast clock request temporarily suspends any active low-power mode and runs the MCLK and ULPCLK tree at 32MHz, sourced from SYSOSC. Asynchronous fast clock requests are also functional in RUN and SLEEP mode if MCLK is sourced from either LFCLK at 32kHz or SYSOSC at a frequency lower than 32MHz.

This functionality enables use cases such as:

- General purpose IOs (GPIO) for glitch filtering
- On-demand UART, $I^2C$, or SPI communication
- Timer-triggered ADC sampling from STANDBY mode

**Suspended STOP or STANDBY for a DMA Trigger**

If a DMA trigger is asserted in STOP or STANDBY mode, the low-power mode is temporarily suspended and the PD1 power domain (including the SRAM and flash memory) is enabled to process the DMA request. Unlike the asynchronous fast clock request, DMA transfers do not change the clock tree configuration. A DMA request in STOP or STANDBY mode is processed at the current MCLK rate.

**Note**

If a temporary suspension from STOP or STANDBY mode results in an interrupt, then the device will transition to and remain in RUN mode operation to process the interrupt.



Copyright © 2026 Texas Instruments Incorporated

## 2.2 Power Management (PMU)

The Power Management Unit (PMU) generates the regulated core supplies for the device and provides supervision of the external supply. It also contains a bandgap voltage reference used by the PMU and other analog peripherals. Refer to the device specific datasheet for the supported PMU features.

Key PMU features include:

- Support for the device across the device operating supply range
- **Core LDO:** Low-dropout linear voltage regulator to generate the internal core logic supply, with multiple operating modes for reducing device current in low-power modes (managed automatically by SYSCTL)
- Power-on reset (POR) for VDD supply
- Brownout reset (BOR) for VDD supply
- Bandgap voltage reference supporting the BOR, core regulator, and analog peripherals
- Analog mux VBOOST unit for increasing analog mux performance

### *2.2.1 Power Supply*

Power is supplied to the device through the VDD and VSS connections. A decoupling capacitor ($C_{VDD}$) must be placed across all VDD and VSS supply pairs. See the device-specific data sheet for the correct value and tolerance of $C_{VDD}$. Some devices with a higher pincount my feature multiple VDD/VSS power pairs.

VDD is used directly to provide the IO supply (VDDIO) and the analog supply (VDDA). VDDIO and VDDA are internally connected to VDD so that additional power supply pins are not required.

Some devices may feature an independent power supply for device-specific special features such as VBAT (for the low-frequency sub system) and VUSB (for the USB PHY). Refer to the device-specific datasheet for details on the power source for associated IO pins and features.

### *2.2.2 Core Regulator*

The PMU uses an on-chip, configurable, low-dropout linear voltage regulator to generate a 1.35V supply rail to power the device core. The core regulator output ($V_{CORE}$) supplies power to the core logic, which includes the CPU, digital peripherals and the device memory. The core regulator requires an external capacitor ($C_{VCORE}$) which is connected between the device VCORE pin and VSS (ground). See the device specific data sheet for the correct value and tolerance of $C_{VCORE}$.

The core regulator is active in all power modes except for SHUTDOWN. In all other power modes (RUN, SLEEP, STOP, and STANDBY) the drive strength of the regulator is configured automatically to support the max load current of each mode. This reduces the quiescent current of the regulator when using low-power modes, improving low power performance. SYSCTL automatically configures the regulator for best power consumption based on the power mode which is currently active.

### *2.2.3 Supply Supervisors*

The PMU provides two supply supervisor circuits:

- A power-on reset (POR) circuit to indicate that the external supply has reached sufficient voltage to start the on-chip bandgap reference and BOR circuit
- A user-programmable brownout reset (BOR) circuit which verifies that the external supply is maintained at the specified voltage to support correct operation of the device

#### 2.2.3.1 Power-on Reset (POR)

The power-on reset (POR) circuit assures a proper start-up of the device. During cold power-up, the device is held in a POR state until VDD passes the POR+ threshold. When VDD has passed POR+, the POR state is released and the bandgap reference and BOR monitor circuit are started. If VDD drops below the POR- level, then a POR- violation is asserted and the device is again held in a POR reset state.

The POR circuit does not indicate that VDD has reached a level high enough to support correct operation of the device. Rather, it is the first step in the boot process and is used to determine if the supply voltage is sufficient to



Copyright © 2026 Texas Instruments Incorporated

power up the bandgap reference and BOR circuit, which are then used to determine if the supply has reached a level sufficient to for the device to run correctly.

The POR circuit is active in all power modes including SHUTDOWN and cannot be disabled.

#### 2.2.3.2 Brownout Reset (BOR)

The brownout reset (BOR) supervisor monitors the external supply (VDD) and asserts or deasserts a BOR violation to SYSCTL. The primary responsibility of the BOR circuit is to make sure that the external supply is maintained high enough to enable correct operation of internal circuits, including the core regulator. The BOR threshold reference is derived from the internal bandgap circuit. The threshold is programmable and is always higher than the POR threshold. During cold start, after VDD passes the POR+ threshold, the bandgap reference and BOR circuit are started. The device is then held in a BOR state until VDD passes the BOR0+ threshold. When VDD passes BOR0+, the BOR supervisor releases the device to continue the boot process, and the PMU is started.

There are up to four selectable BOR threshold levels (BOR0-BOR3). During startup, the BOR threshold is always BOR0 (the lowest value) to make sure that the device starts at the specified VDD minimum. After boot, software can optionally reconfigure the BOR circuit to use a different (higher) threshold level (BOR1-BOR3).

**Note**

Some devices feature only one BOR threshold level (BOR0).

When the BOR threshold is BOR0, a BOR0- violation always generates a BOR- violation signal to SYSCTL, generating a BOR level reset. When the BOR threshold is reconfigured to BOR1, BOR2, or BOR3, the BOR circuit generates a SYSCTL interrupt rather than asserting the BOR- violation. This can be used to give the application an indication that the supply has dropped below a certain level without causing a reset.

To change the BOR level from the default (BOR0), first select the desired value in the LEVEL field of the BORTHRESHOLD register in SYSCTL. Then, activate the threshold set in the LEVEL field by setting the GO bit in the BORCLRCMD register. The change can be validated by testing the BORCURTHRESHOLD field in the SYSSTATUS register, which returns a value corresponding to the currently active BOR threshold. The BOR threshold change takes approximately 15µs to complete, during this time, the BOR circuit is blind to changes in the supply.

If the BOR is in interrupt mode (a threshold level of BOR1-BOR3), and the supply drops below the corresponding BORx- level, then an interrupt is generated and the BOR circuit automatically switches the BOR threshold level to BOR0 to make sure that a BOR- violation is asserted if VDD drops below BOR0-. Application software can set the BOR level back to the level specified in the LEVEL field of the BORTHRESHOLD register by setting the GO bit again in the BORCLRCMD register.

The BOR supervisor is active in RUN, SLEEP, STOP, and STANDBY modes but is disabled automatically in SHUTDOWN mode.

#### 2.2.3.3 POR and BOR Behavior During Supply Changes

When the supply voltage (VDD) drops below POR-, the entire device state is cleared. Small variations in VDD that do not pass below the BOR0- threshold do not cause a BOR- violation, and the device continues to run. Behavior for BORx thresholds other than BOR0 (for example, BOR1-BOR3) is the same as is shown for BOR0, except that the BOR circuit is configured to generate an interrupt rather than immediately triggering a BOR reset. If a BOR1-BOR3 interrupt is generated, the BOR level is automatically switched to BOR0.



Copyright © 2026 Texas Instruments Incorporated

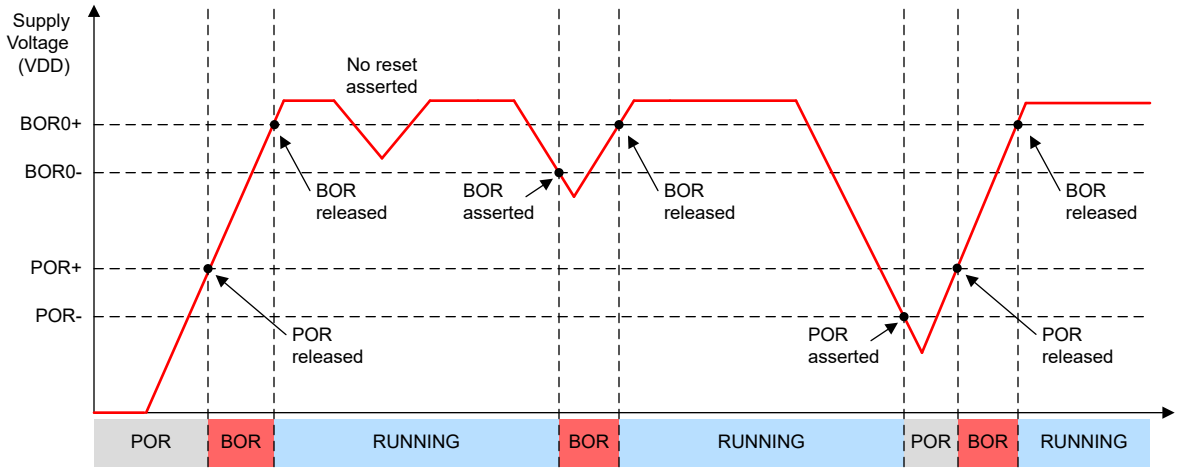



**Figure 2-3. POR/BOR vs. Supply Voltage (VDD)**

### *2.2.4 Bandgap Reference*

The PMU provides a voltage reference that is stable across temperature and VDD supply voltage and is used by the device for internal functions, including:

- Deriving the brownout reset circuit thresholds
- Setting the output voltage for the core regulator
- Deriving the on-chip VREF levels for on-chip analog peripherals

The bandgap reference is enabled in RUN, SLEEP, STOP modes. This reference operates in a sampled mode in STANDBY to reduce power consumption and is disabled in SHUTDOWN mode. SYSCTL manages the bandgap state automatically; no user configuration is required.

### *2.2.5 VBOOST for Analog Muxes*

The VBOOST circuit in the PMU generates an internal VBOOST supply that is used by the analog muxes in specific modules. The VBOOST circuit enables consistent analog mux performance across the external supply voltage (VDD) range.

**Modules supported by VBOOST**

- COMP
- OPA
- GPAMP
- SYSOSC (for FCL-ON external resistor mode operation)
- HFXT

**Note**

Refer to the data sheet for device-specific features

**Enabling and Disabling VBOOST**

SYSCTL automatically manages the enable request for the VBOOST circuit based on the following parameters:

- The COMP, OPA, and GPAMP peripheral PWREN settings
- The ANACPUMPCFG control bits in the GENCLKCFG register in SYSCTL (please refer to the table below)
- The enabling of HFXT as a HFCLK clock source



Copyright © 2026 Texas Instruments Incorporated

**Note**

If SYSOSC is configured with FCL ON and EXT-R, then VBOOST is kept active through low power modes to provide fast exit out of LPM. This is hardware managed, and not dependent on the ANACPUMPCFG setting

VBOOST is disabled by default following a SYSRST. Application software is not required to enable the VBOOST circuit prior to using associated peripherals. When the peripheral is enabled by application software, SYSCTL also enables the VBOOST circuit to support the peripheral.

The VBOOST circuit has a startup time requirement to transition from a disabled state to an enabled state. If a peripheral is enabled when VBOOST is disabled, then the corresponding peripheral ready status is not asserted until both the peripheral and the VBOOST circuit are ready. If the startup time of the peripheral is less than the VBOOST startup time, the peripheral startup time is then equivalent to the VBOOST startup time.

Alternatively, application software can force the VBOOST circuit to be enabled at all times (with ANACPUMPCFG=0x2) or while in RUN or SLEEP mode (with ANACPUMPCFG=0x1) so that there is no additional startup delay when enabling a peripheral supported by VBOOST. Table 2-1 gives the behavior of the ANACPUMPCFG control and corresponding application use cases.

**Table 2-1. Forcing VBOOST Enable With ANACPUMPCFG**

| VBOOST Control (ANACPUMPCFG) | | Behavior | |
|---|---|---|---|
| **VAL** | **MODE** | **VBOOST Enable** | **Application Use Case** |
| 0x0 | ONDEMAND | VBOOST is automatically enabled by SYSCTL only when a supported peripheral is enabled. | This setting provides the lowest power consumption in all modes when fast startup of the peripheral is not critical. |
| 0x1 | ONACTIVE | VBOOST is forced to be enabled when the device is in RUN or SLEEP mode. VBOOST is also kept enabled in STOP or STANDBY mode if a supported peripheral is enabled. | This setting provides low power consumption by disabling the VBOOST automatically in STOP and STANDBY modes when no peripherals requiring VBOOST are enabled. VBOOST is automatically re-enabled upon exit to RUN mode to provide fast startup of the supported peripherals in the event that application software enables a said peripheral in RUN mode. |
| 0x2 | ONALWAYS | VBOOST is forced to be enabled in all operating modes except SHUTDOWN. | This setting makes sure that the peripherals never incur additional startup latency due to VBOOST startup in applications where fast startup is critical. |

The VBOOST enable, VBOOST clock selection, and VBOOST clock error logic is shown in Figure 2-4.

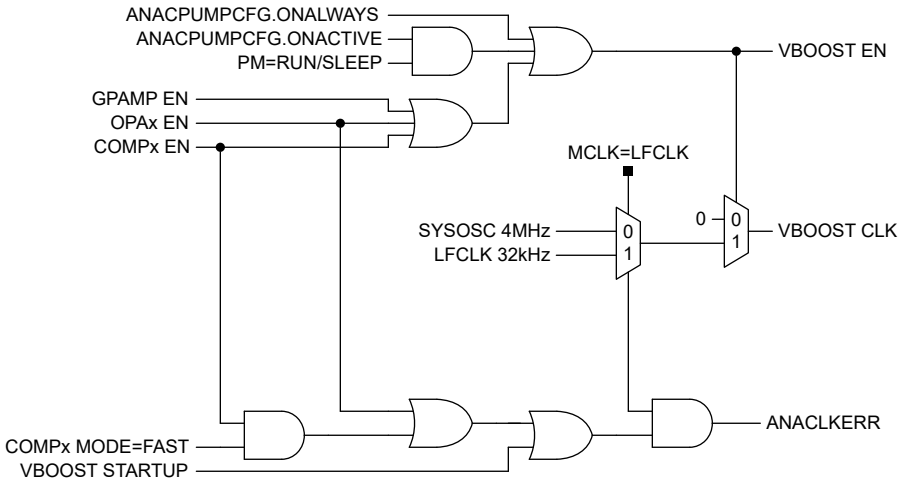



**Figure 2-4. VBOOST Request Logic**



Copyright © 2026 Texas Instruments Incorporated

### VBOOST Clock

The VBOOST circuit requires a functional clock to operate. VBOOST is clocked by either the SYSOSC (4MHz output) or the LFCLK (32kHz) based on the currently active MCLK and ULPCLK tree source. The VBOOST clock is selected by automatically by SYSCTL according to the logic shown in Figure 2-4 and described here:

- When the MCLK and ULPCLK are sourced from LFCLK (32kHz), the VBOOST is also sourced from LFCLK.
- In all other scenarios, VBOOST is sourced from the SYSOSC 4MHz output.

Certain VBOOST operating conditions require that the VBOOST clock be 4MHz (sourced from SYSOSC) and not 32kHz (sourced from LFCLK) for well-designed startup time. Such conditions include:

- OPA operation or fast mode COMP operation
- VBOOST is starting up (transitioning from disabled to enabled)

The application software must make sure that MCLK and ULPCLK are not sourced from LFCLK when either of these conditions are present. SYSCTL does not change the current system clock configuration for a VBOOST request. If the MCLK and ULPCLK tree is sourced from LFCLK in one of these scenarios, SYSCTL asserts an ANACLKERR status in the CLKSTATUS register in SYSCTL to indicate to application software that there is a mismatch between the VBOOST clock requirement and the current MCLK and ULPCLK configuration.

The complete clock requirements for VBOOST are given in Table 2-2.

**Table 2-2. VBOOST Clock Requirements**

| VBOOST Request | | VBOOST Circuit Clock Requirement | MCLK and ULPCLK Source Requirement(1) | Supported Operating Modes or Policies |
|---|---|---|---|---|
| No active request (VBOOST is disabled) | | N/A | Don't care | Don't care |
| VBOOST is starting up from a disabled state | | 4MHz | Not LFCLK | RUN0, SLEEP0, STOP0, STOP1 |
| OPA enabled | | 4MHz | Not LFCLK | RUN0, SLEEP0, STOP0, STOP1 |
| COMP enabled | Fast mode (FAST) | 4MHz | Not LFCLK | RUN0 |
| | Ultra-low-power mode (ULP)(2) | 4MHz or 32kHz | Don't care | RUN0, RUN1, RUN2, SLEEP0, SLEEP1, SLEEP2, STOP0, STOP1, STOP2, STANDBY0 |
| GPAMP enabled(2) | | 4MHz or 32kHz | Don't care | RUN0, RUN1, RUN2, SLEEP0, SLEEP1, SLEEP2, STOP0, STOP1, STOP2, STANDBY0 |
| VBOOST running with no peripheral request active (due to ANACPUMPCFG!=0x0)(2) | | 4MHz or 32kHz | Don't care | RUN0, RUN1, RUN2, SLEEP0, SLEEP1, SLEEP2, STOP0, STOP1, STOP2, STANDBY0 |
| HFXT | | 4MHz or 32kHz | Don't care | RUN0, SLEEP0, STOP0, STOP1 |
| SYSOSC (with FCL ON and EXT-R) | | 4MHz or 32kHz | Don't care | RUN0, SLEEP0, STOP0, STOP1 |

(1) Application software must make sure that the clock system is configured as stated in the previous table to support proper VBOOST operation for each scenario. SYSCTL **does not change the system clock configuration when VBOOST is requested**. If a VBOOST request requires the VBOOST clock to be 4MHz and MCLK is sourced from LFCLK, SYSCTL asserts an ANACLKERR to indicate to the application that the current system clock configuration is not valid to support the current VBOOST request.

(2) If no OPA is enabled and no COMP is enabled and configured in FAST mode, the VBOOST **does not require a 4MHz clock** and VBOOST can run from LFCLK to reduce power consumption. However, if the VBOOST was previously disabled (no request present) and it is requested by a peripheral becoming enabled or by application software (ANACPUMPCFG!=0x0), **a 4MHz VBOOST clock is required for VBOOST startup**. After VBOOST startup is complete, the VBOOST clock can be 32kHz sourced from LFCLK if no OPA is enabled and no COMP is enabled in FAST mode. Changing the MCLK source to LFCLK during VBOOST startup can lead to incorrect operation of the VBOOST and inconsistent analog peripheral performance. When the VBOOST circuit has started and is ready, the ANACPUMPGOOD bit is set in the SYSSTATUS register in SYSCTL.



Copyright © 2026 Texas Instruments Incorporated

### 2.2.6 Peripheral Enable

All peripherals on a device, with the exception of infrastructure peripherals such as SYSCTL itself and the IOMUX, require two user actions to enable the peripheral to be ready for operation:

- Peripheral Write Enable
  - Provided by the peripheral write enable control register (PWREN) with a KEY and ENABLE field.
  - Before any other peripheral registers are configured by software, the peripheral itself must be enabled by writing the ENABLE bit together with the appropriate KEY value to the peripheral's PWREN register
  - When the PWREN.ENABLE bit is cleared, the peripheral's registers are not accessible for read/write operations.
- Peripheral Function Enable
  - Peripherals may have a functional enable bit in the corresponding register map to provide access to power or for operation

**Note**

After setting the ENABLE | KEY bits in the PWREN register to enable a peripheral, wait at least 4 ULPCLK clock cycles before accessing the rest of the peripheral's memory-mapped registers. The 4 cycles allow for the bus isolation signals at the peripheral's bus interface to update.

#### 2.2.6.1 Automatic Peripheral Disable in Low Power Modes

Peripherals in power domain 1 (PD1) will be forced to a disabled state by SYSCTL upon entry into a STOP or STANDBY low-power mode. As such, these peripherals will not be available for use in STOP or STANDBY.

Most PD1 peripherals will retain their configuration settings after being automatically disabled, such that re-configuration is not required upon exit from STOP or STANDBY mode. See the peripheral-specific chapter in this guide for details on which peripheral registers are retained through STOP and STANDBY mode for PD1 peripherals.

If a PD1 peripheral was multiplexed to an IO pin (through the IOMUX) in an output configuration, the last valid output state (logic 0 or logic 1) from the peripheral to the IO is latched upon entry to STOP or STANDBY mode. This prevents external circuits from being disturbed by SYSCTL disabling a peripheral during low-power operation. Upon exit from STOP or STANDBY mode, the IO is again connected to the peripheral as the peripheral becomes re-enabled.



Copyright © 2026 Texas Instruments Incorporated

## 2.3 Clock Module (CKM)

The clock module contains the internal and external oscillators, the clock monitors, and the clock selection and control logic. A frequency clock counter is also provided for checking and/or calibrating the frequency of high-speed clocks against either the LFXT/LFCLK_IN or a reference period/pulse provided on an IO pin (FCC_IN).

### *2.3.1 Oscillators*

Several internal and external oscillators are provided for generating low to high frequency clocks for use by the system. The CKM contains all the oscillators in the device and uses them to generate the system clocks. See the device specific data sheet for whether external oscillators are available.

**Internal Oscillators**

- LFOSC: low frequency oscillator (32kHz typical frequency)
- SYSOSC: system oscillator (up to 32MHz frequency)
- *SYSPLL*: system PLL (phase-locked loop) with programmable frequency (optional)
- *USBFLL*: USB FLL (frequency-locked loop) for crystal-less USB operation (optional)

**External Oscillators**

- LFXT: low frequency, low power crystal oscillator (32kHz typical frequency)
- HFXT: high frequency crystal oscillator (up to 48MHz typical frequency)

In addition to the oscillators, low frequency (LFCLK_IN) and high frequency (HFCLK_IN) digital clock inputs are provided to support cases where the LFXT or HFXT is not used and a direct digital clock is supplied to the device.

#### 2.3.1.1 Internal Low-Frequency Oscillator (LFOSC)

The low-frequency oscillator (LFOSC) is an on-chip low power oscillator which is factory trimmed to a frequency of 32.768kHz. LFOSC provides a stable low frequency clock source in cases where the low frequency crystal oscillator (LFXT) is either not present or not used.

LFOSC can provide higher accuracy when used over a reduced temperature range. See the device-specific data sheet for details.

The LFOSC is active by default after a BOOTRST, sourcing the LFCLK. The LFOSC startup monitor sets the LFOSCGOOD bit in the CLKSTATUS register when LFOSC is ready.

If user software selects either LFXT or LFCLK_IN as the LFCLK source, the LFOSC is disabled until the next BOOTRST.

#### 2.3.1.2 Internal System Oscillator (SYSOSC)

The system oscillator (SYSOSC) is an on-chip, accurate high frequency clock source (up to 32MHz). The SYSOSC provides a flexible high-speed clock source to the system in cases where the high frequency crystal oscillator (HFXT) is either not present or not used, or where fast wake-up from a low-power mode is required.

Key features of the SYSOSC include:

- High accuracy when using optional frequency correction loop (FCL) and reference resistor
  - The frequency correction loop may support correction via an external resistor (ROSC) or an internal resistor, depending on the device capabilities. Refer to the device-specific data sheet to determine if a device supports the FCL with an internal or external resistor, or both
- Fast start-up time from a low power state
- Capable of switching from base frequency to low frequency, or low frequency to base frequency, in one clock cycle with no functional interruption (gear shift)
  - Phase-aligned transition to minimize disturbance to peripherals
  - Fast settling to specified accuracy
  - SYSCTL can initiate seamless gear shift frequency switch in STOP mode to reduce SYSOSC current



Copyright © 2026 Texas Instruments Incorporated

  - Refer to the device-specific data sheet to determine if multiple frequency settings for SYSOSC are supported
- The SYSOSC clock source provides a user trimmable frequency start at 16MHz or 24MHz.
  - In-system trimming is supported by the usage of the FCC (frequency clock counter)
- A secondary output with a constant 4MHz frequency for use by MFCLK
  - When $f_{SYSOSC}$ = 4MHz, the MFCLK/MFPCLK output is derived from SYSOSC directly.
  - When $f_{SYSOSC}$ is not 4MHz, the MFCLK/MFPCLK output is derived by a digital divider sourcing from SYSOSC with the appropriate ratio
  - SYSCTL manages the digital divider on this output to ensure a constant 4MHz output regardless of the user selected SYSOSC frequency.
  - Peripherals using this output see a continuous 4MHz functional clock in RUN, SLEEP, and STOP modes, with reduced current in STOP mode when gear shift is enabled due to SYSOSC running natively at 4MHz.

The SYSOSC is active at the device's base frequency by default after a brownout reset, sourcing MCLK.

#### *2.3.1.2.1 SYSOSC Gear Shift*

The SYSOSC can shift from base frequency (e.g. 32MHz) to low frequency (4MHz), or low frequency back to base frequency, in a single SYSOSC clock cycle with the frequency change being phase aligned to minimize jitter. This feature is particularly well suited for low power, low cost microcontroller applications because it allows a single oscillator to be used to generate two frequencies:

- A high-speed (e.g. 32MHz) clock (for running the CPU during compute), which is only enabled when needed
- A lower-speed 4MHz clock (for running peripherals such as timers, UART interfaces, and $I^2C$ interfaces) which is continuously running

When a high-speed clock is needed (for example, in RUN and SLEEP modes), SYSOSC can run at its base frequency to provide high performance. In such a case, SYSOSC can be used directly to source MCLK so that MCLK can be used by the CPU, DMA, and peripherals clocked by the bus. For peripherals that run continuously but do not require the high clock frequency, the middle frequency system clock (MFCLK) is provided at 4MHz by taking the base frequency of SYSOSC and dividing down to achieve 4MHz.

When the fast clock is not needed (for example, if the CPU has completed processing data), then in STOP mode SYSOSC can be gear shifted down to 4MHz in one cycle, and MFCLK switches to being clocked directly by SYSOSC- keeping a continuously running 4MHz clock, but with a reduction in SYSOSC current and no need for a separate oscillator to provide the 4MHz clock. When the CPU is again needed, SYSOSC can be switched back from 4MHz to the base frequency. This transition also occurs in one SYSOSC clock cycle. During the gear shift up, MFCLK switches back to keep the 4MHz constant frequency (with suitable digital division). The timing transition between base frequency and low frequency during a gearshift down request is shown in Figure 2-5.

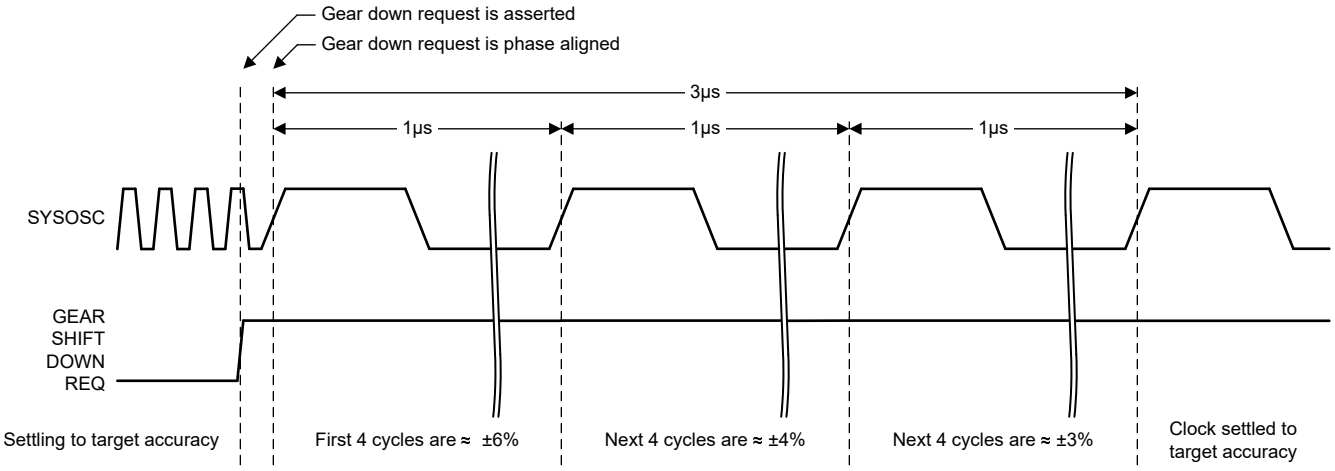



**Figure 2-5. SYSOSC Gearshift Down Timing**



Copyright © 2026 Texas Instruments Incorporated

The SYSOSC gearshift mode is tightly integrated with the power management scheme and is controlled by SYSCTL. Gearshift mode can be used when switching between RUN mode and STOP mode (gear down) and STOP mode and RUN mode (gear up). To use gearshift in STOP mode to reduce SYSOSC current, set the USE4MHZSTOP bit in the SYSOSCCFG register. Note that if SYSOSC is configured for 4MHz operation in RUN and SLEEP mode as well as STOP mode (no gear shift), then the USE4MHZSTOP bit does not need to be set to keep the SYSOSC at 4MHz when entering STOP mode.

**Note**

After issuing the command to reconfigure SYSOSC to 4MHz, the user can poll the SYSOSCFREQ bit of the SYSCTL CLKSTATUS register to confirm the operating frequency.

#### *2.3.1.2.2 SYSOSC Frequency*

SYSOSC comes factory trimmed for operation at either the base frequency or 4MHz. The default frequency of operation is the base frequency. It is possible to run SYSOSC continuously at 4MHz for reduced current operation if 4MHz provides suitable performance for the application. To change the SYSOSC frequency to 4MHz, set the FREQ field in the SYSOSCCFG register to 4M.

**Note**

The FREQ field in the SYSOSCCFG register must only be changed when SYSOSC is the MCLK source. Do not change the SYSOSC FREQ configuration when MCLK is sourced from LFCLK or from HSCLK.

SYSOSC also supports user-trimmed operation at either 24MHz or 16MHz. No other frequencies are allowed. To run SYSOSC at a user trimmed frequency, first ensure that SYSOSC is running at the base frequency. Then, configure the SYSOSCTRIMUSER register for the target frequency. Finally, select USER in the FREQ field of the SYSOSCCFG register. For trimming, see the trim procedure.

**Note**

When running at a user frequency (16MHz or 24MHz), changing the SYSOSC frequency selection to low frequency (4MHz) requires a transition through the base frequency before SYSOSC switches to low frequency. The same is true when switching from low frequency to a user frequency. This change is managed entirely in hardware. Software can directly select the low frequency when running at a user frequency (and the reverse) but the hardware transitions through the base frequency when making the frequency change. If MFCLK (4MHz) is enabled, it will be maintained at 4MHz during the transition.

#### *2.3.1.2.3 SYSOSC Frequency Correction Loop*

The SYSOSC frequency accuracy can be improved through the use of the SYSOSC frequency correction loop (FCL) feature. The FCL circuit uses either an internal resistor or external resistor (populated between the ROSC pin and VSS), to stabilize the SYSOSC frequency by providing a precise reference current for the SYSOSC. The overall frequency accuracy which is achievable depends on the operating temperature range together with the tolerance and temperature drift of the selected reference resistor.

**Note**

Not all devices support operation with an internal resistor and an external resistor for FCL. Some devices support external mode, some devices support internal mode, and some devices support both modes. See the device-specific data sheet to determine the features of a given device.

**Note**

The power consumption of SYSOSC is marginally higher with the FCL enabled due to the reference current that flows through ROSC. Settling time from startup to target accuracy can also be longer. See the device-specific data sheet for startup times.



Copyright © 2026 Texas Instruments Incorporated

#### 2.3.1.2.3.1 SYSOSC FCL in External Resistor Mode (ROSC)

This section describes the procedure for selecting the external ROSC resistor (including computing the achievable SYSOSC accuracy) as well as the procedure for enabling the mode through the device SYSCTL registers.

**SYSOSC Overall Frequency Accuracy with TI Recommended ROSC Resistor**

The device-specific data sheet includes specifications for overall SYSOSC frequency accuracy when the TI-recommended resistor is used for ROSC. The TI-recommended resistor enables a variety of applications, including UART communication. If the overall accuracy values in the device-specific data sheet with the TI-recommended resistor specifications meets the application and cost requirements, then there is no need to compute the accuracy for an alternate resistor or alternate temperature ranges.

However, if a higher accuracy or reduced cost resistor is needed, the designer can use a more precise or less precise resistor for ROSC to balance cost with frequency accuracy for a given application scenario.

**SYSOSC Overall Frequency Accuracy with Designer Selected Resistor**

Assuming an ideal (zero error, zero temperature drift) reference resistor, the FCL mode improves the SYSOSC circuit accuracy to the level specified in the $f_{SYSOSC}$:*"SYSOSC frequency accuracy when frequency correction loop (FCL) is enabled and an ideal ROSC resistor is assumed"* section of the SYSOSC specifications table in the device-specific data sheet. The overall SYSOSC frequency accuracy in FCL mode which is achievable for a given external resistor and temperature range is determined by combining the following error sources to determine the total frequency error:

1. The ROSC resistor tolerance (at 25°C); this comes from the resistor specifications
2. The ROSC resistor temperature coefficient of resistance (TCR); this comes from the resistor specifications
3. The minimum and maximum expected operating temperatures for the system being designed
4. The SYSOSC frequency accuracy with an ideal resistor for the desired temperature range (taken from the MSPM0 device-specific data sheet)

The following table gives an example of how to compute the accuracy for different conditions, given the following typical application scenario:

- A target operating temperature range of -40 to 85°C.
- ROSC resistor with a ±0.1% tolerance and ±25 ppm/°C TCR selected

**Table 2-3. Computing Overall SYSOSC FCL Frequency Accuracy based on External Resistor (ROSC) Specifications**

| Step | Description | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|---|
| 1 | Define the operating temp range of the application. | -40 | 25 | 85 | ℃ |
| 2 | Select the target $f_{SYSOSC}$ frequency (4 or 32MHz) | | 32.00 | | MHz |
| 3 | Look up the corresponding $f_{SYSOSC}$ ideal resistor accuracy from the device-specific data sheet based on the temperature range in Step 1. | -0.80 | | 0.93 | % |
| 4 | Compute the min and max $f_{SYSOSC}$ with an ideal resistor based on the selected frequency in Step 2 and the accuracy range in Step 3. | 31.744 | | 32.298 | MHz |
| 5 | Select an ROSC resistor with acceptable tolerance (the TI-recommended ±0.1% tolerance used in this example) | 99.9 | 100 | 100.1 | kΩ |
| 6 | Note the temperature coefficient of resistance (TCR) of the ROSC resistor. | | ±25 | | ppm/℃ |
| 7 | Compute the max temperature variation from 25°C (nominal) based on the temperature range defined in Step 1. | | \| 25 - (-40) \| = 65 | | ℃ |
| 8 | Compute the max ROSC drift over the max temperature variation (from Step 7) and the TCR (from Step 6). | -0.16 | | 0.16 | % |
| 9 | Compute the min and max ROSC resistance including tolerance (from Step 5) and temperature drift (from Step 8). | 99.74 | | 100.26 | kΩ |
| 10 | Compute the overall ROSC resistor accuracy (in percent), using the overall ROSC resistance range from Step 9 vs. the nominal resistor value. | -0.26 | | 0.26 | % |



Copyright © 2026 Texas Instruments Incorporated

**Table 2-3. Computing Overall SYSOSC FCL Frequency Accuracy based on External Resistor (ROSC) Specifications (continued)**

| Step | Description | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|---|
| 11 | Compute the variation in the min and max $f_{SYSOSC}$ by degrading the min/max $f_{SYSOSC}$ computed in Step 4 by the ROSC accuracy computed in Step 10. | 31.66 | | 32.38 | MHz |
| 12 | If desired, convert the raw min/max $f_{SYSOSC}$ computed in Step 11 into a %-based accuracy format relative to the target $f_{SYSOSC}$ from Step 2. | -1.06 | | 1.20 | % |

### Enabling FCL with an External Resistor (ROSC)

To increase the SYSOC accuracy with FCL, follow this procedure:

1. Verify that all digital functions on the ROSC pin are disabled in IOMUX (this is the default state after power up).
2. Verify that a ROSC reference resistor that meets the application accuracy requirements is placed between the ROSC pin and the device ground (VSS).
3. Enable FCL mode by setting the SETUSEFCL bit in the SYSOSCFCLCTL register.
   a. If the device supports both internal resistor and external resistor FCL modes, also set the SETUSEEXRES bit in the SYSOSCFCLCTL register in addition to the SETUSEFCL bit.
4. When the FCL mode is enabled, software cannot disable the mode. A BOOTRST is required before a change to the FCL mode.

#### *2.3.1.2.3.2 SYSOSC FCL in Internal Resistor Mode*

This section describes the procedure for selecting the internal resistor mode through the device SYSCTL registers. Enabling FCL in internal resistor mode can be beneficial to the user's application which require improved SYSOSC performance while not adding cost to the system for an external resistor.

### Enabling FCL with Internal Resistor

To increase the SYSOSC accuracy with FCL, follow this procedure:

1. Enable FCL mode by setting the SETUSEFCL bit in the SYSOSCFCLCTL register.
   a. If the device supports both internal resistor and external resistor FCL modes, do not set the SETUSEEXRES bit in the SYSOSCFCLCTL register when setting the SETUSEFCL bit.
2. When the FCL mode is enabled, software cannot disable the mode. A BOOTRST is required before a change to the FCL mode.

#### *2.3.1.2.4 SYSOSC User Trim Procedure*

The SYSOSC can be configured at 16MHz or 24MHz as an alternative to the base frequency. To utilize these configurations, the trim values need to be applied. Trimming is accomplished by manipulating the fields in the SYSOSCTRIMUSER to achieve the target frequency. To trim the SYSOSC, the SYSOSC clock can be brought out externally to a pin using the CLK_OUT unit so that it can be measured.

Use the following procedure to derive the user trim values. The trimmed values can be programmed into the flash memory of the device to be recalled later. Each device must be trimmed individually for accuracy. SYSOSC can also be trimmed internally by using the frequency clock counter (FCC).

### Trim procedure using CLK_OUT with frequency correction loop (FCL) disabled (No ROSC resistor)

1. Enable the CLK_OUT unit and select SYSOSC as the clock source, following the instructions in the CLK_OUT section
2. Set the SYSOSC frequency to BASE to start (ensure that the FREQ field in the SYSOSCCFG register is set to 0h for BASE), and leave the FCL mode disabled
3. Program an initial tuning for the target frequency in the SYSOSCTRIMUSER register:



Copyright © 2026 Texas Instruments Incorporated

   a. Set the RESCOARSE trim field in the SYSOSCTRIMUSER register to mid-range
   b. Set the RESFINE trim field in the SYSOSCTRIMUSER register to mid-range
4. Switch SYSOSC to the user-trimmed frequency by selecting USER in the FREQ field of the SYSOSCCFG register
5. Measure the SYSOSC frequency at the CLK_OUT pin
6. Switch SYSOSC back to BASE frequency by selecting BASE in the FREQ field of the SYSOSCCFG register. Writing to SYSOSCTRIMUSER requires SYSOSC to be operating at BASE frequency.
7. Adjust the SYSOSCTRIMUSER parameters to close on the target frequency:
   a. Set the target frequency (16 or 24MHz) in the FREQ field of the SYSOSCTRIMUSER register
   b. Adjust trim parameters to achieve 16 or 24 MHZ based on the selection made in step 7a
      i. Increasing the CAP trim field in the SYSOSCTRIMUSER register will reduce the SYSOSC range. For 24MHz, select CAP=0, and for 16MHz select CAP=1
      ii. Increasing the RESCOARSE trim field in SYSOSCTRIMUSER will reduce the frequency in large steps of approximately 1MHz
      iii. Increasing the RESFINE trim field in SYSOSCTRIMUSER will decrease the frequency in small steps of approximately 100kHz
8. Repeat steps 4 through 7 until the desired accuracy has been achieved
9. Save the resulting SYSOSCTRIMUSER value for recall

**Trim procedure using CLK_OUT with frequency correction loop (FCL) enabled (ROSC resistor present)**

1. First, complete the FCL disabled trim procedure (defined above) to obtain starting values for the CAP, RESCOARSE, and RESFINE trim fields in the SYSOSCTRIMUSER register
2. Increase the RESCOARSE value by 02h from the value obtained in Step 1
3. With CLK_OUT still enabled (SYSOSC as the source for CLK_OUT), ensure that the FREQ field in the SYSOSCCFG register is set to 0h for BASE, and enable the FCL mode by setting the SETUSEFCL bit in the SYSOSCFCLCTL register
4. Set the RDIV trim field in the SYSOSCUSERTRIM register to mid-range
5. Switch SYSOSC to the user-trimmed frequency by selecting USER in the FREQ field of the SYSOSCCFG register
6. Measure the SYSOSC frequency at the CLK_OUT pin
7. Switch SYSOSC back to BASE frequency by selecting BASE in the FREQ field of the SYSOSCCFG register
8. Adjust trim parameters to close on the target frequency:
   a. Increasing the RDIV trim field in the SYSOSCTRIMUSER register will increase the frequency in small steps of approximately 50kHz
   b. Continue increasing or decreasing RDIV if the frequency is in saturation (not appearing to adjust)
9. Repeat steps 5-8 until the desired accuracy has been achieved
10. Save the resulting SYSOSCTRIMUSER value for recall

**Note**

SYSOSCTRIMUSER parameters only take effect when switching to the USER frequency. Once USER frequency is selected, changes to SYSOSCTRIMUSER have no effect. To refresh the USER values, first switch to BASE frequency, then update SYSOSCTRIMUSER, and then switch SYSOSC back to BASE frequency.

#### *2.3.1.2.5 Disabling SYSOSC*

SYSOSC can be disabled in STOP mode by setting the DISABLESTOP bit in the SYSOSCCFG register. Doing so forces the MCLK to use LFCLK in STOP mode. This provides the lowest possible power consumption in STOP mode, as the system runs at 32kHz and SYSOSC consumes no current. When exiting STOP mode to RUN mode, SYSCTL will automatically re-enable SYSOSC and switch MCLK back to SYSOSC.



Copyright © 2026 Texas Instruments Incorporated

SYSOSC can be disabled manually by setting the DISABLE bit in the SYSOSCCFG register. When SYSOSCCFG.DISABLE is set, the system will run from LFCLK in all power modes.

**Note**

The DISABLE and DISABLESTOP bits in SYSOSCCFG are mutually exclusive, and must not be set at the same time.

It is not possible to disable SYSOSC when using a different high speed clock to source MCLK (such as HSCLK). This is because SYSOSC is used by SYSCTL logic when MCLK is sourced from HSCLK.

SYSOSC is always disabled automatically in STANDBY and SHUTDOWN modes.

#### 2.3.1.3 System Phase-Locked Loop (SYSPLL)

The system phase locked loop (SYSPLL) takes an input reference clock SYSPLLREF and scales the input frequency to produce user-specified high frequency clocks (SYSPLLCLK0, SYSPLLCLK1, and SYSPLLCLK2X) for use by the device. Specifically, the SYSPLL clock outputs can be used as sources to MCLK, CANCLK and USBCLK. Figure 2-6 shows the block diagram of the SYSPLL.

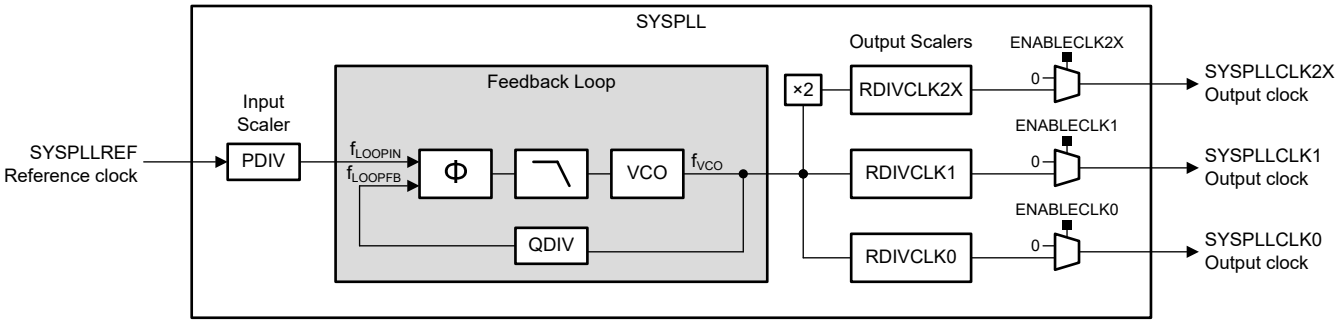



**Figure 2-6. SYSPLL Diagram**

**Note**

SYSOSC must not be disabled when the SYSPLL is enabled for use.

##### *2.3.1.3.1 Configuring SYSPLL Output Frequencies*

The SYSPLL accepts an input reference clock from 4-48MHz. The available reference clocks include SYSOSC and HFCLK. The predivider PDIV scales the selected input reference clock ahead of the PLL feedback loop. PDIV can be selected as /1, /2, /4, or /8 by programming 0x0 to 0x3, respectively, into the PDIV field in the SYSPLLCFG1 register.

The effective divider can be calculated from the PDIV register setting as shown in Equation 1.

$$SYSPLLREF_{DIV} = 2^{PDIV} \quad (1)$$

The PLL feedback loop sets the voltage controlled oscillator (VCO) output equal to the divided input reference clock $f_{LOOPIN}$ multiplied by the QDIV feedback divider. The QDIV divider is an integer divider with a valid range of /2 to /127. The desired QDIV divider is selected by programming 0x01 to 0x7E for /2 to /127, respectively, into the QDIV field of the SYSPLLCFG1 register. SYSPLLCFG1.QDIV=0x00 is an invalid configuration. The effective feedback divider can be calculated from the QDIV register setting as shown in Equation 2.

$$SYSPLLFB_{DIV} = QDIV + 1 \quad (2)$$

The output frequency of the VCO $f_{VCO}$ is given in Equation 3.

$$f_{VCO} = f_{SYSPLLREF} \times SYSPLLFB_{DIV} / SYSPLLREF_{DIV} \quad (3)$$



Copyright © 2026 Texas Instruments Incorporated

The VCO output sources three separate SYSPLL outputs (SYSPLLCLK0, SYSPLLCLK1, and SYSPLLCLK2X). Each output has its own divider unit to enable generation of up to 3 different output frequencies for use by different modules in the device. The third output (SYSPLLCLK2X) also contains a frequency doubler before the divider unit to provide a wider range of output frequencies and lower power consumption.

For SYSPLLCLK2X, the output divider can be set from /1 to /16 in steps of 1. To set the SYSPLLCLK2X output divider, program 0x0-0xF for /1 up to /16, respectively, into the RDIVCLK2X field in the SYSPLLCFG0 register. Equation 4 shows how to compute the effective SYSPLLCLK2X divider based on a given RDIVCLK2X register setting.

$$SYSPLLCLK2X_{DIV} = RDIVCLK2X + 1 \quad (4)$$

For SYSPLLCLK1 and SYSPLLCLK0, the output dividers can be set from /2 to /32 in steps of 2. To set the SYSPLLCLK0 or SYSPLLCLK1 output divider, program 0x0-0xF for /2 to /32, respectively, into the corresponding RDIVCLKx field in the SYSPLLCFG0 register. Equation 5 shows how to compute the effective SYSPLLCLK0 divider based on a given RDIVCLK0 setting, and Equation 6 shows how to compute the effective SYSPLLCLK1 divider based on a given RDIVCLK1 setting.

$$SYSPLLCLK0_{DIV} = 2 \times (RDIVCLK0+1) \quad (5)$$

$$SYSPLLCLK1_{DIV} = 2 \times (RDIVCLK1+1) \quad (6)$$

The SYSPLL output clock frequencies are thus set by the combination of $f_{VCO}$ and the respective dividers:

$$f_{SYSPLLCLK0} = f_{VCO} / SYSPLLCLK0_{DIV} \quad (7)$$

$$f_{SYSPLLCLK1} = f_{VCO} / SYSPLLCLK1_{DIV} \quad (8)$$

$$f_{SYSPLLCLK2X} = 2 \times f_{VCO} / SYSPLLCLK2X_{DIV} \quad (9)$$

### Enabling and Disabling the SYSPLL

After configuration, enable the SYSPLL by setting the SYSPLLEN bit in the HSCLKEN register. Before enabling the SYSPLL, make sure that the SYSPLL is in a disabled state by verifying that the SYSPLLOFF bit in the CLKSTATUS register is set. After the SYSPLL is enabled, application software must not disable it until the SYSPLLGOOD bit or SYSPLLOFF bit is set in the CLKSTATUS register, indicating that the SYSPLL transitioned to a stable active state or a stable dead state. When the SYSPLL is enabled, the SYSPLL reference clock selection must not be changed.

**Note**

SYSOSC must be enabled and running at base frequency when the SYSPLL is enabled, even if HFCLK is used as the SYSPLL reference clock.

### SYSPLL Usage Example (for CAN protocol)

To illustrate the above relationships, take as an example the following requirements:

- The internal system oscillator (SYSOSC) is used as the SYSPLL reference (32MHz)
- The SYSPLL must be configured with an output frequency of 80MHz to source MCLK and 40MHz to source CANCLK

To achieve this, the VCO can be configured for 80MHz through the use of PDIV and QDIV. Then, SYSPLLCLK1 can feed CANCLK with an output divider of /2, and SYSPLLCLK2X can feed MCLK with an output divider of /2.

The steps below describe how to configure the CKM to use SYSPLL in this way:

1. Verify that the SYSPLL is disabled (SYSPLLOFF is set in CLKSTATUS)



Copyright © 2026 Texas Instruments Incorporated

2. Make sure that SYSOSC is running at base frequency (32MHz); this is a requirement for SYSPLL operation even if HFCLK is used as the SYSPLL reference clock instead of SYSOSC
3. Set SYSOSC as the SYSPLL reference (make sure that the SYSPLLREF bit in the SYSPLLCFG0 register is cleared; this is the default state after reset)
4. Select a predivider PDIV to /2 (set SYSPLLCFG1.PDIV to 0x01), setting $f_{LOOPIN}$ to 16MHz (32 divided by 2)
5. Load the PLL parameters into SYSPLLPARAM0 and SYSPLLPARAM1 to support $f_{LOOPIN}$ of 16MHz
6. Set the feedback divider QDIV to 5 (set SYSPLLCFG1.QDIV to 4), giving $f_{VCO}$ =80MHz (16MHz multiplied by 5)
7. Set the SYSPLL output dividers for SYSPLLCLK1 and SYSPLLCLK2X to /2 (set SYSPLLCFG0.RDIVCLK1 to 0x0 and SYSPLLCFG0.RDIVCLK2X to 0x1) to get 40MHz and 80MHz at SYSPLLCLK1 and SYSPLLCLK2X, respectively
8. Enable SYSPLLCLK1 and SYSPLLCLK2X outputs by setting the ENABLECLK1 and ENABLECLK2X bits in the SYSPLLCFG0 register
9. Select SYSPLLCLK2X as the PLL output to the HSCLK mux by setting MCLK2XVCO in the SYSPLLCFG0 register
10. With SYSOSC enabled and running at base frequency (32MHz, this is the default state out of reset), enable the SYSPLL by setting SYSPLLEN in the HSCLKEN register
11. Wait for the SYSPLLGOOD indication by testing SYSPLLGOOD in the CLKSTATUS register
12. Select the SYSPLL as the HSCLK source by ensuring that the HSCLKSEL bit is cleared in the HSCLKCFG register (this is the default state)
13. Select the high-speed clock (HSCLK) as the source for MCLK by setting the USEHSCLK bit in the MCLKCFG register. This will switch MCLK from SYSOSC to HSCLK. MCLK is now running from SYSPLLCLK2X at 80MHz.
14. To configure the CANCLK, set the CANCLKSRC bit in the GENCLKCFG register.

The SYSPLL divider values used in this example are summarized below for reference.

**Table 2-4. SYSPLL Divider Example Settings**

| Parameter | Register | Bit Field | Bit Field Value | Actual Divider |
|---|---|---|---|---|
| Input reference clock divider | SYSPLLCFG1 | PDIV | 0x1 | /2 |
| VCO feedback loop divider | SYSPLLCFG1 | QDIV | 0x4 | /5 |
| Output clock 1 divider | SYSPLLCFG0 | RDIVCLK1 | 0x0 | /2 |
| Output clock 2X divider | SYSPLLCFG0 | RDIVCLK2X | 0x1 | /2 |

### Tuning Guidelines

In cases where there are multiple combinations of PDIV, QDIV, and RDIVCLKx that provide the desired output frequencies, consider these tuning guidelines to determine the best possible values for an application:

- Lower VCO frequencies ($f_{VCO}$ ) result in lower power consumption. Refer to the device data sheet for the allowable range of $f_{VCO}$ .
- Higher feedback loop input frequencies ($f_{LOOPIN}$ ) have faster startup. For example, if an 80MHz output frequency is desired with $f_{VCO}$ =40MHz, $f_{VCO}$ = 40MHz can be derived from a SYSPLLREF clock 32MHz by setting PDIV to /8 and QDIV to /10. However, this gives slower startup because $f_{LOOPIN}$ is <8MHz. The same result can be achieved by setting PDIV to /4 and QDIV to /5, but because $f_{LOOPIN}$ is 8MHz in this case (32MHz divided by 4), the SYSPLL parameters for the higher input range can be used which give faster startup.

#### *2.3.1.3.2 Loading SYSPLL Lookup Parameters*

Several tuning parameters must be configured in the SYSPLLPARAM0 and SYSPLLPARAM1 registers before using the SYSPLL. The values are determined by PLL feedback loop input clock frequency ($f_{LOOPIN}$).



Copyright © 2026 Texas Instruments Incorporated

SYSPLL supports four $f_{LOOPIN}$ frequency ranges, given in SYSPLL Parameter Lookup. Each frequency range has a 64-bit lookup value in the FACTORY flash memory region that must be copied from flash into the SYSPLLPARAM0 and SYSPLLPARAM1 registers in SYSCTL before enabling the SYSPLL.

**Table 2-5. SYSPLL Parameter Lookup**

| $f_{LOOPIN}$ | Lookup Address (PARAM0) | Lookup Address (PARAM1) | SDK Symbol |
|---|---|---|---|
| 32MHz ≤ FREQ ≤ 48MHz | 0x41C4.0034 | 0x41C4.0038 | DL_SYSCTL_SYSPLL_INPUT_FREQ_32_48_MHZ |
| 16MHz ≤ FREQ < 32MHz | 0x41C4.002C | 0x41C4.0030 | DL_SYSCTL_SYSPLL_INPUT_FREQ_16_32_MHZ |
| 8MHz ≤ FREQ < 16MHz | 0x41C4.0024 | 0x41C4.0028 | DL_SYSCTL_SYSPLL_INPUT_FREQ_8_16_MHZ |
| 4MHz ≤ FREQ < 8MHz | 0x41C4.001C | 0x41C4.0020 | DL_SYSCTL_SYSPLL_INPUT_FREQ_4_8_MHZ |

#### *2.3.1.3.3 SYSPLL Startup Time*

The PLL startup time depends on the PLL feedback loop input frequency $f_{LOOPIN}$ and if the PLL is starting after previously running (for example, exiting a low-power mode) or starting for the first time after device boot.

Table 2-6 lists an example of how the PLL startup times can be affected by the loop input frequency $f_{LOOPIN}$. Note that $f_{LOOPIN}$ is determined by the SYSPLLREF and user-configured PDIV and QDIV.

**Note**

These values are used as a relative example only. For specific device performance, please refer to the product datasheet.

**Table 2-6. SYSPLL Startup Times (example)**

| $f_{LOOPIN}$ | Condition | SYSPLL Startup Time (µs) | SYSPLL Startup Time on Exit From Low-Power Mode (µs) |
|---|---|---|---|
| 32MHz ≤ FREQ | SYSPLLREF=32MHz, PDIV=0, QDIV=4 | 13 | 11 |
| 16MHz ≤ FREQ < 32MHz | SYSPLLREF=32MHz, PDIV=1, QDIV=9 | 16 | 14 |
| 8MHz ≤ FREQ < 16MHz | SYSPLLREF=32MHz, PDIV=2, QDIV=19 | 18 | 16 |
| 4MHz ≤ FREQ < 8MHz | SYSPLLREF=32MHz, PDIV=3, QDIV=39 | 26 | 24 |

### 2.3.1.4 Low Frequency Crystal Oscillator (LFXT)

The low-frequency crystal oscillator (LFXT) is an ultra-low-power crystal oscillator that supports a standard 32.768kHz watch crystal.

To use the LFXT, place a watch crystal between the LFXIN and LFXOUT pins. Place loading capacitors on both pins to circuit ground (VSS). Size the crystal load capacitors according to the specifications of the crystal being used. A variety of crystal types are supported through a programmable drive mechanism that allows for balancing drive current with the required drive strength for a particular crystal.

The LFXT pins LFXIN and LFXOUT are shared with digital IO functions. To use the LFXT, first configure the IOMUX to support LFXT functionality on the LFXIN and LFXOUT pins. Configure IOMUX to disable any digital IO functionality on the LFXIN and LFXOUT pins. The LFXT defaults to the highest drive strength (LFCLKCFG.XT1DRIVE == 0x03 in SYSCTL), which is recommended for fast reliable start-up of the oscillator. If a crystal with ultra-low capacitance (<3pF) is used, power consumption of the LFXT can be lowered further by setting the LOWCAP bit LFCLKCFG (LOWCAP is cleared by default for compatibility with most crystals).



Copyright © 2026 Texas Instruments Incorporated

After completing configuration, start the LFXT by setting the STARTLFXT bit in the LFXTCTL register in SYSCTL. When the oscillator has started successfully, the LFXT startup monitor sets the LFXTGOOD bit in the CLKSTATUS register in SYSCTL. Crystal drive strength can then be reduced to conserve power, if desired.

To switch the LFCLK tree to use LFXT as the 32kHz clock source rather than LFOSC, set the SETUSELFXT bit in the LFXTCTL register, and SYSCTL permanently switches the LFCLK source to LFXT until a BOR reset or a fatal clock fault occurs.

After the LFXT is enabled, the internal LFOSC is disabled to conserve power. LFOSC cannot be re-enabled and LFCLK thus cannot be switched back to LFOSC other than by a software-triggered POR reset.

**Note**

To source MCLK from LFCLK with LFCLK sourced from the LFXT, first configure LFCLK to use LFXT and then configure MCLK to use LFCLK. Do not switch MCLK to LFCLK while LFCLK runs from LFOSC, and then later switch LFCLK to LFXT.

**Note**

In the event that LFXT is selected for LFCLK at startup but the crystal fails to start properly, it is necessary to execute a POR reset to re-enable the internal LFOSC and run LFCLK from LFOSC or else a 32kHz LFCLK will not be available for the system.

**Note**

In a scenario where LFXT fails to start, application software may leverage the flash memory to store a flag indicating that the crystal did not start properly, and then execute the POR reset. The flash memory contents can be read following a POR reset and thus in the next startup sequence the application code can decide to keep LFCLK sourced from LFOSC versus the external crystal (in the event that there is a hardware issue with the crystal preventing it from starting).

#### 2.3.1.5 LFCLK_IN (Digital Clock)

It is possible to bypass the LFXT circuit and bring in a 32.768kHz typical frequency digital clock into the device to use as the LFCLK source instead of LFOSC or LFXT. To configure LFCLK to use a digital clock input instead of LFXT or LFOSC, first configure the IOMUX to enable the LFCLK_IN function on the appropriate pin. When IOMUX is configured correctly and the external clock source is outputting a 32kHz clock to LFCLK_IN, set the SETUSEEXLF bit in the EXLFCTL register in SYSCTL.

LFCLK_IN is compatible with digital square wave CMOS clock inputs and should have a typical duty cycle of 50%.

When the system is initialized, LFOSC is the primary clock source for LFCLK. Before switching LFCLK to use LFCLK_IN, it is possible to check for a valid clock signal on LFCLK_IN by enabling the LFCLK monitor through setting the MONITOR bit in LFCLKCFG register.

After LFCLK_IN is selected as the LFCLK source, it is not possible to change back to LFOSC or LFXT without going through a BOR reset.

**Note**

If MCLK is to be sourced from LFCLK with LFCLK sourced from LFCLK_IN, first configure LFCLK to use LFCLK_IN and then configure MCLK to use LFCLK. Do not switch MCLK to LFCLK with LFCLK running from LFOSC, and then later switch LFCLK to LFCLK_IN.

**Note**

LFCLK_IN and LFXT are mutually exclusive and must not be enabled at the same time. Do not set the SETUSEEXLF bit in the EXLFCTL register if the SETUSELFXT bit or the STARTLFXT bit is set in the LFXTCTL register.



Copyright © 2026 Texas Instruments Incorporated

#### 2.3.1.6 High Frequency Crystal Oscillator (HFXT)

The high frequency crystal oscillator (HFXT) can be used with standard crystals and resonators to generate a stable high-speed reference clock for the system. The HFXT can be used to clock the primary device clock tree (MCLK) directly or it can be used as a precision reference to the on-chip PLL where higher frequencies can be generated. In addition, the HFXT can be provided directly to the ADC as the sampling clock source or for other peripherals such as the CAN-FD functional clock source. This allows for a functional clock that is asynchronous from the main system clock (MCLK).

To use the HFXT, a crystal or resonator must be populated between the HFXIN and HFXOUT pins. Loading capacitors must be placed on both pins to circuit ground (VSS). The crystal load capacitors must be sized according to the specifications of the crystal being used. The IOMUX must be configured to enable HFXT functionality on the HFXIN and HFXOUT pins. Configure IOMUX to disable any digital IO functionality on the HFXIN and HFXOUT pins. The HFXT frequency range must be set by configuring the HFXTRSEL bits in the HFCLKCLKCFG register in SYSCTL.

A programmable HFXT startup time is provided with 64µs resolution. Program an appropriate startup time based on the desired crystal or resonator specifications into the HFXTTIME field in the HFCLKCLKCFG register in SYSCTL before starting the HFXT.

Once configured properly, the HFXT is started by setting the HFXTEN bit in the HSCLKEN register in SYSCTL. When the oscillator has started successfully, the HFCLK startup monitor will assert the HFCLKGOOD bit in the CLKSTATUS register in SYSCTL.

**Note**

SYSOSC must be enabled at base frequency when the HFXT is enabled.

After setting HFXTEN to enable the HFXT, application software must verify that either an HFCLKGOOD indication or an HFCLKOFF (off/dead) indication in the CLKSTATUS register was asserted by hardware before attempting to disable the HFXT by clearing HFXTEN. When disabling the HFXT by clearing HFXTEN, the HFXT must not be re-enabled again until the HFCLKOFF bit in the CLKSTATUS register is set by hardware.

To use HFXT as the PLL reference after receiving an HFCLKGOOD status, set the SYSPLLREF bit in the SYSPLLCFG0 register in SYSCTL. If HFXT is selected as a reference for the SYSPLL and the SYSPLL is enabled, then the SYSPLL must be disabled and the SYSPLLOFF bit in the CLKSTATUS register must be set before the HFXT can be disabled.

To use the HFXT directly as the MCLK source after receiving an HFCLKGOOD status, first set the HSCLKSEL bit in the HSCLKCFG register to select HFCLK as the high-speed clock source (rather than the system PLL output). Then, set the USEHSCLK bit in the MCLKCFG register to select the high-speed clock source as the MCLK source. Once USEHSCLK is set, HSCLKCFG must not change and the HFXT must not be disabled until the MCLK source is switched back to SYSOSC by clearing USEHSCLK and verifying that the HSCLKMUX bit in CLKSTATUS is cleared by hardware.

#### 2.3.1.7 HFCLK_IN (Digital clock)

It is possible to bypass the HFXT circuit and bring in an external digital clock signal into the device to use as the HFCLK source instead of HFXT. To configure HFCLK to use a digital clock input, first configure the IOMUX to enable the HFCLK_IN function on the appropriate pin. When IOMUX is configured correctly and the clock source is outputting a clock to HFCLK_IN, set the USEEXTHFCLK bit in the HSCLKEN register in SYSCTL.

**Note**

SYSOSC must be enabled at base frequency when the HFCLK_IN is enabled.

The HFCLK_IN can be used as the reference for SYSPLL. The SYSPLL must be off before changing the configuration of the SYSPLL reference clock source. The user can verify that SYSPLL is off by checking the SYSPLLOFF status bit in the CLKSTATUS register. The user must then select HFCLK_IN as the HFCLK source (by setting the USEEXTHFCLK bit in the HSCLKEN register) before setting the SYSPLLREF bit in the



Copyright © 2026 Texas Instruments Incorporated

SYSPLLCFG0 register in SYSCTL. Finally, the SYSPLL can be enabled by using the SYSPLLEN bit in the HSCLKEN register.

To source MCLK from HFCLK_IN after selecting HFCLK_IN as the HFCLK source, first set the HSCLKSEL bit in the HSCLKCFG register to select HFCLK as the high-speed clock source. Then, set the USEHSCLK bit in the MCLKCFG register to select the high-speed clock source as the MCLK source. Once USEHSCLK is set, HSCLKCFG must not change and the HFCLK_IN must not be disabled until the MCLK source is switched back to SYSOSC by clearing USEHSCLK and verifying that the HSCLKMUX bit in CLKSTATUS was cleared by hardware.

HFCLK_IN is compatible with digital square wave CMOS clock inputs and should have a typical duty cycle of 50%.

**Note**

HFCLK_IN and HFXT are mutually exclusive and must not be enabled at the same time. Do not set the USEEXTHFCLK bit if the HFXTEN bit is also set in HSCLKEN.

**Note**

When HFCLK_IN is used, USEMFTICK in the MCLKCFG register is not supported. MFTICK can only used with SYSOSC.

### *2.3.2 Clocks*

The CKM takes oscillator outputs and generates a variety of functional clocks for use by the device. Refer to the device specific datasheet to determine the supported functional clocks.

**Clocks**

- System Clocks
  - MCLK: Main system clock for PD1 peripherals and PD1 bus
  - CPUCLK: CPU clock, derived from MCLK is available only in RUN mode
  - ULPCLK: Main system clock for PD0 peripherals and PD0 bus, derived from MCLK
  - MFCLK: Fixed 4MHz clock, synchronized to MCLK/ULPCLK
  - MFPCLK: Fixed 4MHz clock, utilized for the DAC
  - LFCLK: Fixed 32kHz clock, synchronized to MCLK/ULPCLK
  - HFCLK: High frequency external clock
  - HSCLK: High speed configurable clock for use by MCLK, sourced from SYSPLL or HFCLK
- Peripheral Specific Clocks
  - ADCCLK: ADC sampling period clock
  - CANCLK: CAN-FD module functional clock
  - RTCCLK: Fixed 32kHz clock direct to RTC
- External Clocks
  - CLK_OUT: External clock output with divider for pushing out a clock to external circuits

All clocks are disabled in SHUTDOWN mode.

In addition to the configurable clocks listed above, several direct clock connections are made to analog peripherals (see the *Direct Clock Connections for Infrastructure* section).

#### 2.3.2.1 MCLK (Main Clock) Tree

The MCLK is the main system clock and the root point of synchronization for all synchronized clocks (MCLK, CPUCLK, ULPCLK, MFCLK, and LFCLK). It is typically the highest speed clock in the system and supports operation up to 80MHz across the full temperature range of the device. The MCLK tree is the root source for the CPUCLK (in RUN mode), the PD1 high speed peripheral bus clock (in RUN and SLEEP modes), and the ULPCLK low power bus clock (in RUN, SLEEP, STOP, and STANDBY modes). In addition, the 4MHz MFCLK and 32kHz LFCLK outputs are synchronized to MCLK.



Copyright © 2026 Texas Instruments Incorporated

The MCLK output to PD1 peripherals is enabled in RUN and SLEEP modes, and disabled in all other power modes. While the MCLK output to PD1 is disabled in STOP and STANDBY modes, the MCLK tree is still running to source ULPCLK and to provide synchronization for MFCLK and LFCLK.

The MCLK source is selected with a glitch free clock mux and can be changed dynamically at runtime by user software. It can also be changed automatically by hardware when entering STOP and STANDBY modes or during an asynchronous fast clock request.

The available sources for MCLK include:

- **HSCLK** (high-speed clock) at up to 80MHz which can be sourced by:
  - **SYSPLLCLK0** or **SYSPLLCLK2X** (used to reach the max MCLK speed of 80MHz)
  - **HFCLK** for applications where the main clock needs to be as accurate as possible
- **SYSOSC** at 4, 16, 24, or 32MHz (the default clock source after reset)
- **SYSOSC** at 4MHz with **MDIV** divider (for applications where the peripheral bus and CPU run between 250kHz and 4MHz)
- **LFCLK** at 32kHz for applications where the entire system, including the CPU, runs at 32kHz with low peak operating current

**NOTE:** Set **SYSPLLCLK0** or **SYSPLLCLK2X** as the source of **HSCLK** before setting **HSCLK** as the clock source of **MCLK**, otherwise the device can be in an unpredictable state.

### Using MCLK in RUN and SLEEP Mode

After boot, MCLK is sourced from SYSOSC by default. The decision of which oscillator to use to source MCLK is important because MCLK sets both the CPUCLK frequency and the bus clock frequency for PD1 peripherals. As a result, the accuracy and the clock speed of the oscillator selected for MCLK must be appropriate not only for the operation of the CPU but also for the operation of the PD1 peripherals that use the bus clock as their functional clock.

The clock source and frequency selection decisions made for MCLK also affect ULPCLK in RUN and SLEEP modes. See the ULPCLK section for more information on how MCLK and ULPCLK are related in RUN and SLEEP mode.

### Using MCLK in STOP and STANDBY Mode

In STOP and STANDBY modes, the MCLK output to PD1 peripherals is disabled, but the ULPCLK, which is the bus clock for PD0 peripherals, is still active in STOP and optionally active in STANDBY. See the ULPCLK section for more information on how the MCLK source and ULPCLK are related in STOP and STANDBY mode.

### MCLK Source Selection

Application software can change the MCLK source from SYSOSC to the SYSPLL or to the HFCLK (which is either HFXT or HFCLK_IN) by configuring the MCLKCFG.USEHSCLK and HSCLKCFG.HSCLKSEL register bits appropriately in SYSCTL. It is also possible to select LFCLK as the MCLK source in all modes by setting MCLKCFG.USELFCLK, giving the low peak current consumption with the CPU and PD1 peripherals operational. The following table gives the proper register bit configurations for selecting different clocks for MCLK in RUN and SLEEP modes.

**Table 2-7. MSPM0Gxx MCLK Source Selection in RUN and SLEEP Mode**

| Desired Source | MCLKCFG.USEHSCLK | MCLKCFG.USELFCLK | HSCLKCFG.HSCLKSEL | SYSPLLCFG0.MCLK2XVCO |
|---|---|---|---|---|
| SYSOSC | 0 | 0 | Don't care | Don't care |
| SYSPLLCLK0 | 1 | 0 | 0 | 0 |
| SYSPLLCLK2X | 1 | 0 | 0 | 1 |
| HFCLK | 1 | 0 | 1 | Don't care |
| LFCLK | 0 | 1 | Don't care | Don't care |



Copyright © 2026 Texas Instruments Incorporated

To switch MCLK from SYSOSC to LFCLK in RUN mode:

1. If any high speed oscillators (SYSPLL,HFXT, HFCLK_IN) are enabled, disable them before proceeding
2. Verify that MCLK is sourced from SYSOSC (CLKSTATUS.CURMCLKSEL is cleared)
3. If SYSOSC is not running at base frequency, and SYSOSC is to be left enabled when switching MCLK to LFCLK, set SYSOSC to base frequency before proceeding
4. Set MCLKCFG.USELFCLK to switch MCLK to LFCLK and leave SYSOSC enabled, or set SYSOSCCFG.DISABLE to switch MCLK to LFCLK and disable SYSOSC

To switch MCLK from LFCLK to SYSOSC in RUN mode:

1. Verify that MCLK is sourced from LFCLK (CLKSTATUS.CURMCLKSEL is set)
2. Clear MCLKCFG.USELFCLK or SYSOSCCFG.DISABLE, whichever was set to switch MCLK to LFCLK

To switch MCLK from SYSOSC to HSCLK:

1. Verify that MCLK is sourced from SYSOSC (CLKSTATUS.HSCLKMUX is cleared).
2. Enable the desired high speed sources (SYSPLL, HFXT, HFCLK_IN) according to their respective requirements.
3. Select the desired HSCLK source through the HSCLKCFG.HSCLKSEL control
4. Verify that CLKSTATUS.HSCLKGOOD is set, indicating that the selected HSCLK source is valid.
5. Set MCLKCFG.USEHSCLK to switch MCLK to HSCLK.

To switch MCLK from HSCLK to SYSOSC:

1. Verify that MCLK is sourced from HSCLK (CLKSTATUS.HSCLKMUX is set).
2. Clear MCLKCFG.USEHSCLK to switch MCLK to SYSOSC.
3. Wait for CLKSTATUS.HSCLKMUX to clear, indicating that MCLK is now sourced from SYSOSC.
4. If desired, disable any high-speed clock sources (SYSPLL orHFXT)

As shown in Table 2-7, the USELFCLK and USEHSCLK bits in MCLKCFG are mutually exclusive and must not be set at the same time. To switch MCLK from LFCLK to HSCLK, or HSCLK to LFCLK, MCLK is first switched to SYSOSC before switching to the final clock source. The procedures below describe how to switch MCLK from HSCLK to LFCLK or from LFCLK to HSCLK.

To switch MCLK from HSCLK to LFCLK:

1. Verify that MCLK is sourced from HSCLK (CLKSTATUS.HSCLKMUX is set).
2. Clear MCLKCFG.USEHSCLK to switch MCLK to SYSOSC
3. Wait for CLKSTATUS.HSCLKMUX to clear, indicating that MCLK is now sourced from SYSOSC
4. Disable any high speed sources (SYSPLL,HFXT, HFCLK_IN) which were enabled
5. Wait for CLKSTATUS.HSCLKSOFF to be asserted, indicating that high-speed clocks are off
6. Set MCLKCFG.USELFCLK to switch MCLK to LFCLK while leaving SYSOSC enabled, or set SYSOSCCFG.DISABLE to switch MCLK to LFCLK and disable SYSOSC

To switch MCLK from LFCLK to HSCLK:

1. Verify that MCLK is sourced from LFCLK (CLKSTATUS.CURMCLKSEL is set)
2. Clear SYSOSCCFG.DISABLE or MCLKCFG.USELFCLK, based on which was set previously
3. Wait for CLKSTATUS.CURMCLKSEL to clear, indicating that MCLK is now sourced from SYSOSC
4. Enable the desired high speed oscillators (SYSPLL,HFXT, HFCLK_IN) according to their respective requirements
5. Select the desired HSCLK source through the HSCLKCFG.HSCLKSEL control
6. Verify that CLKSTATUS.HSCLKGOOD is set, indicating that the selected HSCLK source is valid
7. Set MCLKCFG.USEHSCLK to switch MCLK to HSCLK



Copyright © 2026 Texas Instruments Incorporated

**Note**

When MCLK is actively sourced by HSCLK (the HSCLKMUX bit in the CLKSTATUS register is set), the HSCLK source selection must not be changed (the HSCLKSEL bit in the HSCLKCFG register and the MCLK2XVCO bit in the SYSPLLCFG0 register must not be changed). To change the HSCLK source, first switch MCLK to SYSOSC using the procedure given above, re-configure the HSCLK source, and then switch MCLK back to HSCLK.

### MCLK Divider (MDIV)

An MCLK source divider (MDIV) is provided to enable MCLK operation in between the lowest SYSOSC frequency (4MHz) and the LFCLK frequency (32kHz). MDIV is for applications with a limited peak current but that still require a higher clock speed than 32kHz. MDIV supports dividing the 4MHz SYSOSC frequency by up to 16, enabling the additional MCLK frequency options given in Table 2-8. For example, a 500kHz MCLK frequency can be obtained by setting SYSOSC to 4MHz and setting MDIV to 7 (divide-by-8).

Table 2-8 shows the MCLK frequency which is realized with /2, /4, /8, and /16 MDIV configurations, but MDIV can be set to any integer divider between /2 and /16 (MDIV register values of 0x1 through 0xF, respectively, with 0x0 disabling the MDIV).

**Table 2-8. Typical MCLK Configurations for Operation Between 4MHz and 32kHz**

| MCLK Source | MDIV | MCLK Frequency |
|---|---|---|
| SYSOSC (4MHz) | 0 (Disabled) | 4MHz |
| | 1 (/2) | 2MHz |
| | 3 (/4) | 1MHz |
| | 7 (/8) | 500kHz |
| | 15 (/16) | 250kHz |

To use MDIV to operate MCLK at an intermediate frequency below 4MHz, follow the steps below:

1. Disable asynchronous fast clock requests (make sure that the BLOCKASYNCALL bit is set in the in SYSOSCCFG register)
2. Make sure that SYSOSC is selected as the MCLK source
3. Set the SYSOSC frequency to 4MHz (set the FREQ field in the SYSOSCCFG register to 0x01)
4. Delay for 10 MCLK cycles
5. Set the desired divider value in the MDIV field in the MCLKCFG register (from 1 to 15, corresponding to /2 to /16, respectively)

Several rules apply when using MDIV to reduce the MCLK frequency:

- MDIV must not be used when high speed oscillators are used to source MCLK (SYSPLL, HFCLK); if SYSPLL or HFCLK are selected to source MCLK, MDIV must be disabled (MCLKCFG.MDIV=0x00)
- The SYSOSC frequency must be kept at 4MHz when MDIV is enabled
- Asynchronous requests to change the SYSOSC frequency must remain blocked

To disable MDIV:

1. Disable MDIV (set the MDIV field in the MCLKCFG register to 0x0)
2. Wait for 16 MCLK cycles before changing the SYSOSC frequency from 4MHz to another frequency

**Note**

MDIV does not apply to LFCLK, as MDIV is not in the LFCLK path when LFCLK is selected as the MCLK source. If LFCLK is selected for MCLK, MDIV must be disabled.



Copyright © 2026 Texas Instruments Incorporated

#### 2.3.2.2 CPUCLK (Processor Clock)

The processor clock (CPU clock) is always derived directly from MCLK and is active in RUN mode at the MCLK frequency. In all other power modes, CPUCLK is disabled.

#### 2.3.2.3 ULPCLK (Low-Power Clock)

The ULPCLK is the bus clock for peripherals in the PD0 power domain. It supports operation up to 40MHz and is derived directly from the MCLK tree through a clock divider (UDIV) which is enabled only when MCLK is sourced from a high-speed clock (SYSPLL, HFXT, or HFCLK_IN). The ULPCLK frequency is dependent on the MCLK configuration, the selected power mode, and the selected UDIV divider.

### ULPCLK Behavior in RUN and SLEEP Modes

The PD0 power domain has a frequency limit of 40MHz in RUN and SLEEP modes. As such, ULPCLK must be maintained to be ≤40MHz at all times. When MCLK is configured to run from SYSOSC or LFCLK, SYSCTL disables UDIV automatically and $f_{ULPCLK}$=$f_{MCLK}$ as these clock sources are always ≤32MHz.

However, when MCLK is configured to run from a high-speed clock (SYSPLL, HFXT, or HFCLK_IN), hardware cannot ensure that ULPCLK is ≤40MHz because SYSCTL does not have knowledge of the MCLK frequency. In this case it is the responsibility of the application software to ensure that ULPCLK is ≤40MHz in RUN and SLEEP modes by configuring UDIV appropriately. By default, the ULPCLK divider is set to divide-by-2 (MCLKCFG.UDIV==0x1) which provides safe operation up to the max MCLK frequency (80MHz). If MCLK is configured to be ≤40MHz, the ULPCLK speed is set equal to the MCLK speed by changing MCLKCFG.UDIV from 0x01 to 0x0 (divide-by-1). When UDIV==0x0, MCLK==ULPCLK and accesses to PD0 peripherals do not incur additional latency.

### ULPCLK Behavior in STOP and STANDBY Modes

SYSCTL automatically disables UDIV in STOP and STANDBY modes.

- In STOP mode, the MCLK tree (and by extension, the ULPCLK) can run from SYSOSC with a 4MHz rate (if SYSOSCCFG.DISABLESTOP=0x0) or from LFCLK at 32kHz (if SYSOSCCFG.DISABLESTOP=0x1). When SYSOSC is used (SYSOSCCFG.DISABLESTOP=0x0), SYSCTL ensures that ULPCLK is always 4MHz even if SYSOSC is running at a higher frequency (due to user configuration or due to an asynchronous request from a peripheral).
- In STANDBY mode, the MCLK tree (and by extension, the ULPCLK) either run from LFCLK (STANDBY0) or are disabled (STANDBY1) to conserve power. In STANDBY1, only a few timer peripherals receive ULPCLK.

**Table 2-9. MSPM0Gxx ULPCLK by Operating Mode**

| Selected Power Mode | Configuration | Register Settings | ULPCLK Frequency |
|---|---|---|---|
| RUN or SLEEP (40MHz maximum) | MCLK source is SYSOSC (RUN0, SLEEP0) | MCLKCFG.USEHSCLK=0x0, MCLKCFG.USELFCLK=0x0 | ULPCLK is sourced from MCLK according to the MCLK configuration with $f_{ULPCLK}$ = $f_{MCLK}$ |
| | MCLK source is HSCLK (SYSPLL, HFXT, or HFCLK_IN) (RUN0, SLEEP0) | MCLKCFG.USEHSCLK=0x1, MCLKCFG.USELFCLK=0x0 | ULPCLK is sourced from MCLK according to the MCLK configuration with $f_{ULPCLK}$ = $f_{MCLK}$/ UDIV |
| | MCLK source is LFCLK (RUN1/2, SLEEP1/2) | MCLKCFG.USELFCLK=0x1 or SYSOSCCFG.DISABLE=0x1 | ULPCLK is sourced from LFCLK with $f_{ULPCLK}$ = $f_{LFCLK}$ = 32kHz |
| STOP (4MHz maximum) | STOP with SYSOSC enabled (STOP0/1) | SYSOSCCFG.DISABLESTOP = 0x0 | ULPCLK is sourced from SYSOSC with $f_{ULPCLK}$ = 4MHz |
| | STOP with SYSOSC disabled (STOP2) | SYSOSCCFG.DISABLESTOP = 0x1 | ULPCLK is sourced from LFCLK with $f_{ULPCLK}$ = $f_{LFCLK}$ = 32kHz |
| STANDBY (32kHz maximum) | STANDBY with ULPCLK and LFCLK enabled (STANDBY0) | MCLKCFG.STOPCLKSTBY= 0x0 | ULPCLK is sourced from LFCLK with $f_{ULPCLK}$ = $f_{LFCLK}$ = 32kHz |
| | STANDBY with ULPCLK and LFCLK disabled (STANDBY1) | MCLKCFG.STOPCLKSTBY= 0x1 | ULPCLK is disabled to all peripherals except TIMG0 and TIMG1, which receive $f_{ULPCLK}$ = $f_{LFCLK}$ = 32kHz |



Copyright © 2026 Texas Instruments Incorporated

**Table 2-9. MSPM0Gxx ULPCLK by Operating Mode (continued)**

| Selected Power Mode | Configuration | Register Settings | ULPCLK Frequency |
|---|---|---|---|
| SHUTDOWN (Off) | - | - | ULPCLK is off |

#### 2.3.2.4 MFCLK (Middle Frequency Clock)

The MFCLK provides a continuous 4MHz clock to a variety of peripherals on the device. The MFCLK 4MHz rate is always derived from the SYSOSC. As the SYSOSC frequency is not fixed (it can be configured for 32, 24, 16, or 4MHz), SYSCTL automatically applies a divider to SYSOSC to keep MFCLK at a constant 4MHz rate regardless of the current SYSOSC frequency. MFCLK can be used by peripherals such as timers and serial interfaces that require a constant clock source in RUN, SLEEP, and STOP power modes.

After a SYSRST, MFCLK is initially disabled. MFCLK can be enabled in software by setting USEMFTICK in the MCLKCFG register in SYSCTL. MFCLK is active in RUN, SLEEP, and STOP power modes only, and SYSOSC must be enabled for MFCLK to operate.

All MFCLK edges are synchronized to the main system clocks (MCLK and ULPCLK), meaning that the registers of peripherals clocked by MFCLK can be read or written to at any time without any special handling.

Peripherals can select MFCLK as their functional clock source through their respective CLKSEL mux. Not all peripherals support running from MFCLK.

**Using MFCLK in STOP Mode**

When using MFCLK in STOP mode, SYSOSC can be configured to automatically switch to 4MHz (low frequency) when entering STOP mode and automatically switch back to the previously selected frequency when exiting STOP mode to RUN mode (gear shift mode). As MFCLK is a 4MHz clock source, running SYSOSC at 4MHz in STOP mode reduces power consumption when in STOP mode. To configure SYSOSC for gear shift mode, see Section 2.3.1.2.1 .

**Requirements for Using MFCLK**

1. When using MFCLK, the MDIV (MCLK divider) must be disabled (set to /1). Disable MDIV by setting MDIV in the MCLKCFG register to 0x0. SYSCTL hardware does not allow MFCLK to run when MCLKCFG.MDIV != 0.
2. When using MFCLK, if MCLK is sourced from a source other than SYSOSC then the MCLK frequency must be ≥32MHz for correct operation of MFCLK.
3. When using MFCLK, if MCLK is to be sourced from the high-speed clock (HSCLK), application software must enable MFCLK by setting the USEMFTICK bit before switching the MCLK source from SYSOSC to HSCLK. Do not change the state of USEMFTICK when MCLK is sourced from HSCLK.
4. When using MFCLK, do not enable MFCLK when MCLK is sourced from the low-frequency clock (LFCLK). Application software must enable MFCLK by setting the USEMFTICK bit before switching the MCLK source from SYSOSC to LFCLK. Software can switch MCLK to LFCLK after USEMFTICK is set. In this case, MFCLK halts when MCLK is sourced from LFCLK, and it resumes when MCLK is switched back to SYSOSC.
5. When MFCLK is enabled by setting USEMFTICK in the MCLKCFG register, it is considered by the hardware as a static policy. Do not clear USEMFTICK.

When MFCLK is configured to be enabled, it is only active when SYSOSC is active and MCLK is not sourced from LFCLK. When MCLK is sourced from LFCLK, MFCLK is stopped by hardware automatically. Note that if the device is in STANDBY, MCLK is always sourced from LFCLK and MFCLK is always disabled by hardware.

Asynchronous fast clock requests, if configured, temporarily enable SYSOSC to handle specific peripheral events and activity. If MFCLK is configured to be enabled (USEMFTICK is set), then MFTICK runs when a peripheral asserts an asynchronous fast clock request.



Copyright © 2026 Texas Instruments Incorporated

#### 2.3.2.5 MFPCLK (Middle Frequency Precision Clock)

MFPCLK provides a continuous 4MHz clock to the external clock output (CLK_OUT) mux and 12-bit DAC modulesin RUN, SLEEP, or STOP mode. Unlike MFCLK, which also produces a continuous 4MHz clock for use by most peripherals, MFPCLK is asynchronous to MCLK/ULPCLK and it can be sourced from either the SYSOSC or from HFCLK (HFXT or HFCLK_IN) to provide higher precision to the DACs, improving DAC performance.

The 12-bit DAC modules do not have a clock selection mux. The 12-bit DAC clock source is selected by configuring MFPCLK. To select HFCLK (HFXT or HFCLK_IN) to source the MFPCLK, set the MFPCLKSRC in the GENCLKCFG register in SYSCTL.

To enable the MFPCLK, set the MFPCLKEN bit in the GENCLKEN register in SYSCTL.

**Note**

HFCLK is disabled in STOP mode by SYSCTL automatically when entering STOP mode. If the 12-bit DACs are to be used in STOP mode, then SYSOSC must be the clock source for MFPCLK.

#### 2.3.2.6 LFCLK (Low-Frequency Clock)

LFCLK provides a continuous 32kHz clock to a variety of peripherals on the device. After a BOOTRST, LFCLK is initiallysourced by the internal 32kHz oscillator (LFOSC). After boot, LFCLK can be switched by software to either the low-frequency crystal oscillator (LFXT) or the low-frequency digital clock input (LFCLK_IN). See the respective oscillator section for instructions on switching the LFCLK source. When the LFCLK source is changed, the change is locked and LFOSC is disabled to save power. It is not possible to again select LFOSC as the LFCLK source without executing a BOOTRST.

**Note**

If LFCLK is to be selected as the MCLK source, and LFXT or LFCLK_IN are to be used as the LFCLK source, configure the LFXT or LFCLK_IN as the LFCLK source before selecting LFCLK as the source for MCLK.

LFCLK is active in RUN, SLEEP, STOP, and STANDBY power modes. It is possible to disable both ULPCLK and LFCLK together to most peripherals in STANDBY mode to achieve the lowest possible STANDBY mode power consumption (STANDBY1). To do so, set the STOPCLKSTBY bit in the MCLKCFG register in SYSCTL before entering STANDBY. In this state, the RTC anda few times are the only clocked peripherals.

LFCLK is a synchronized clock. All LFCLK edges are synchronized to the main system clocks (MCLK and ULPCLK), meaning that the registers of peripherals clocked by LFCLK can be read or written to at any time without any special handling. The RTC is the only exception to this rule, as the RTC receives RTCCLK which is the asynchronous LFCLK.



Copyright © 2026 Texas Instruments Incorporated

**Note**

When MCLK/ULPCLK are not sourced by LFCLK (for example, when they are sourced by SYSOSC) there is a 5 ULPCLK cycle synchronization delay between the low frequency clock source's clock edge and the corresponding LFCLK edge as seen by peripherals running from LFCLK. When the MCLK/ULPCLK frequency is constant, this delay is constant and it does not add jitter to LFCLK. If the MCLK/ULPCLK frequency changes, the synchronization delay changes proportionally and this results in a small single-cycle LFCLK jitter at the MCLK/ULPCLK frequency transition point. This jitter changes the duty cycle of one LFCLK period, but there is no accumulation of error (there is never a change in the number of LFCLK periods, ensuring an accurate LFCLK time base for peripherals). A common case would be MCLK/ULPCLK sourced from SYSOSC at 32MHz, with SYSOSC occasionally switching to 4MHz to reduce power consumption. When SYSOSC (and, by definition, MCLK/ULPCLK) switch from 32MHz to 4MHz, the synchronization delay increases from 0.16µs to 1.25µs (an increase of 1.1µs). Thus, during the SYSOSC frequency transition, one LFCLK period will appear 1.1µs (3.6%) longer than normal. When SYSOSC switches back from 4MHz to 32MHz, one LFCLK period will appear 1.1µs shorter than normal, and the phase shift is recovered and error is not accumulated in the long term.

Peripherals can select LFCLK as their functional clock source through their respective CLKSEL mux. Not all peripherals support running from LFCLK. It is possible to run the main clock (MCLK) from LFCK, in which case the entire device runs at the LFCLK rate (32kHz).

#### 2.3.2.7 HFCLK (High-Frequency External Clock)

The high-frequency external clock (HFCLK) is the output of the high-frequency external clock selection mux, and can be selected as either the high-frequency oscillator (HFXT) output or the high-frequency digital clock input (HFCLK_IN).

The HFCLK can be used as the source for the following clocks:

- HSCLK source (to source MCLK)
- SYSPLL reference clock
- CANCLK source
- ADCCLK source
- MFPCLK source (with optional divider)

The HFCLK is only available in RUN and SLEEP modes. HFCLK is automatically disabled by SYSCTL in all other modes, along with the HFXT itself (if enabled).

#### 2.3.2.8 HSCLK (High Speed Clock)

The high-speed clock (HSCLK) is the output of the high-speed clock selection mux, and can be selected to source MCLK by setting the USEHSCLK bit in the MCLKCFG register. The HSCLK is only a selection option for MCLK; it does not source any other functions. HSCLK can be configured to be sourced from the SYSPLL (SYSPLLCLK0 or SYSPLLCLK2X) or the HFCLK (HFXT or HFCLK_IN). By default, the HSCLK is sourced from the SYSPLL.To change the HSCLK source to HFCLK, set the HSCLKSEL bit in the HSCLKCFG register.

The HSCLK is only available in RUN and SLEEP modes. It is automatically disabled by SYSCTL in all other modes, along with theSYSPLL and HFXT (if enabled).

SYSCTL will not switch MCLK to HSCLK, even if requested by software, if the HSCLK status check indicates that selected HSCLK source was not started correctly.

#### 2.3.2.9 ADCCLK (ADC Sample Period Clock)

ADCCLK is used by the ADC module to set the ADC sampling period. The ADCCLK for a given ADC is provided from the CKM to the ADC, but the ADCCLK clock selection is done within each ADC peripheral's configuration registers. See the ADC chapter for information on configuring the ADCCLK. ADCCLK can be selected as ULPCLK, SYSOSC, or HFCLK (HFXT or HFCLK_IN).



Copyright © 2026 Texas Instruments Incorporated

**Note**

On devices which contain more than one ADC, a unique ADCCLK is mapped to each ADC, making it possible for ADCs to have different sampling clocks.

#### 2.3.2.10 CANCLK (CAN-FD Functional Clock)

The CANCLK is a functional clock provided directly to the CAN-FD module from either the HFCLK (HFXT or HFCLK_IN) or the SYSPLL (SYSPLLCLK1). This clock is provided as a functional clock to the CAN-FD module asynchronous from the main clock (MCLK) for the highest possible accuracy. The CANCLK source is selected in SYSCTL by configuring the CANCLKSRC bit in the GENCLKCFG register. Additional CAN-FD clock configuration is provided within the CAN-FD peripheral itself (review the *CAN-FD* chapter for more detail).

#### 2.3.2.11 RTCCLK (RTC Clock)

RTCCLK is the sole clock source for the real-time clock peripheral. Because of the critical time-keeping nature of the RTC, the RTC is provided LFCLK directly before LFCLK is synchronized to the main clock (MCLK). This also enables the RTC to continue counting even when the system clock tree is completely gated in STANDBY mode to reduce power consumption. No special configuration is required for RTCCLK. Because the RTC logic is clocked asynchronously from the system main clock, special rules apply when reading certain RTC registers. See the RTC chapter to understand the rules for handling RTC registers.

#### 2.3.2.12 External Clock Output (CLK_OUT)

A clock output unit is provided for sending digital clock signals from the device to external circuits or to the frequency clock counter. This feature is useful for clocking external circuitry such as an external ADC that does not have a clock source. The clock output unit has a flexible set of sources to select and includes a programmable divider.

Available clock sources for CLK_OUT:

- SYSPLLCLK1
- HFCLK
- SYSOSC
- ULPCLK
- MFPCLK
- LFCLK

The selected clock source can be divided by 1 (no divide), 2, 4, 6, 8, 10, 12, 14, or 16 before being output to the pin or to the frequency clock counter.

To use the clock output unit:

1. Configure IOMUX to select the CLK_OUT function on the device pin with CLK_OUT.
2. Select the desired clock source in the EXCLKSRC field of the GENCLKCFG register.
3. Set the desired clock divider, if necessary, in the EXCLKDIVVAL field of the GENCLKCFG register, and enable the divider by setting the EXCLKDIVEN bit. This must be done while EXCLKEN=0 (before CLK_OUT is enabled)
4. Enable the external clock output by setting the EXCLKEN bit in the GENCLKEN register.

**Note**

When the CLK_OUT source is selected as ULPCLK or MFPCLK, the clock divider must be enabled (GENCLKCFG.EXCLKDIVEN must be set).

**Note**

When clearing the EXCLKEN bit to disable CLK_OUT, allow the clock source to run for 10 clock cycles to stabilize the EXCLKSRC mux.



Copyright © 2026 Texas Instruments Incorporated

**Note**

When disabling a clock source which is selected for CLK_OUT, it is recommended to disable the CLK_OUT function before disabling the clock source if it is important that CLK_OUT be logic low (0) when the clock source is disabled. If CLK_OUT is left enabled and the source for CLK_OUT is disabled, it is possible that CLK_OUT may stop in a logic high (1) state.

**Note**

When the CLK_OUT source is selected as SYSPLLCLK1, the SYSPLLCLK1 output must be ≤48MHz. Further speed restrictions can exist depending on the IO capabilities of a specific device and pin; see the Digital IO specifications in the device-specific data sheet for details on supported IO speeds.

#### 2.3.2.13 Direct Clock Connections for Infrastructure

Several direct clock connections are made in the device to support specific analog functionality:

- SYSOSC to ADCs
- SYSOSC to OPAs, if OPAs are present on a device
- SYSOSC and LFCLK to PMU analog mux VBOOST circuit

**Direct Connections to ADCs**

In addition to receiving ADCCLK to set the sampling window, the ADC modules also receive the direct output of SYSOSC. The SYSOSC direct output to the ADCs is used by the charge pump logic in the ADC modules. SYSOSC can be configured at any frequency to support this function. The ADC supports requesting SYSOSC automatically before a conversion, so there is not a requirement for application software to ensure that SYSOSC is running before triggering an ADC conversion.

**Direct Connections to OPAs**

The SYSOSC direct clock connection to the OPAs, if present, enable all OPA operating modes to be used even when the peripheral bus clock (ULPCLK) is running at a lower frequency than what is required by the OPA. Certain OPA modes require that SYSOSC be running at 32MHz. See the OPA section for clocking details. Configuration of SYSOSC to a frequency which supports the selected OPA mode is the responsibility of the application software. If the OPA is enabled and configured for a mode which requires SYSOSC to be running at a certain frequency, and software has not configured SYSOSC to run at this frequency, an OPAMPCLKERR will be asserted in the CLKSTATUS register in SYSCTL and the OPA can not function properly.

**Direct Connections to the PMU Analog Mux VBOOST Circuit**

Both the SYSOSC and the LFCLK are connected directly to the analog mux VBOOST circuit in the PMU, which is used to boost the performance of the analog muxes used by the COMPs, OPAs, and GPAMP. The VBOOST circuit is enabled automatically when a peripheral requires it for correct operation. It is the responsibility of the application software to ensure that the SYSOSC or LFCLK are enabled to support correct operation of the VBOOST circuit. If the VBOOST circuit requires SYSOSC or LFCLK, and the clock is not available, an ANACLKERR is asserted in the CLKSTATUS register in SYSCTL. SYSCTL can also be configured to raise a SYSCTL interrupt when ANACLKERR is asserted, to alert the application that a COMP, OPA, or the GPAMP can not be functioning properly as expected.

### *2.3.3 Clock Tree*

The following figure shows the top level clocking tree for MSPM0Gxx family devices. This diagram shows the mapping between oscillators (sources) and clocks (destinations), as well as the SYSCTL register bit fields for the selection muxes. Note that not all devices have all clock system features shown in this figure.



Copyright © 2026 Texas Instruments Incorporated

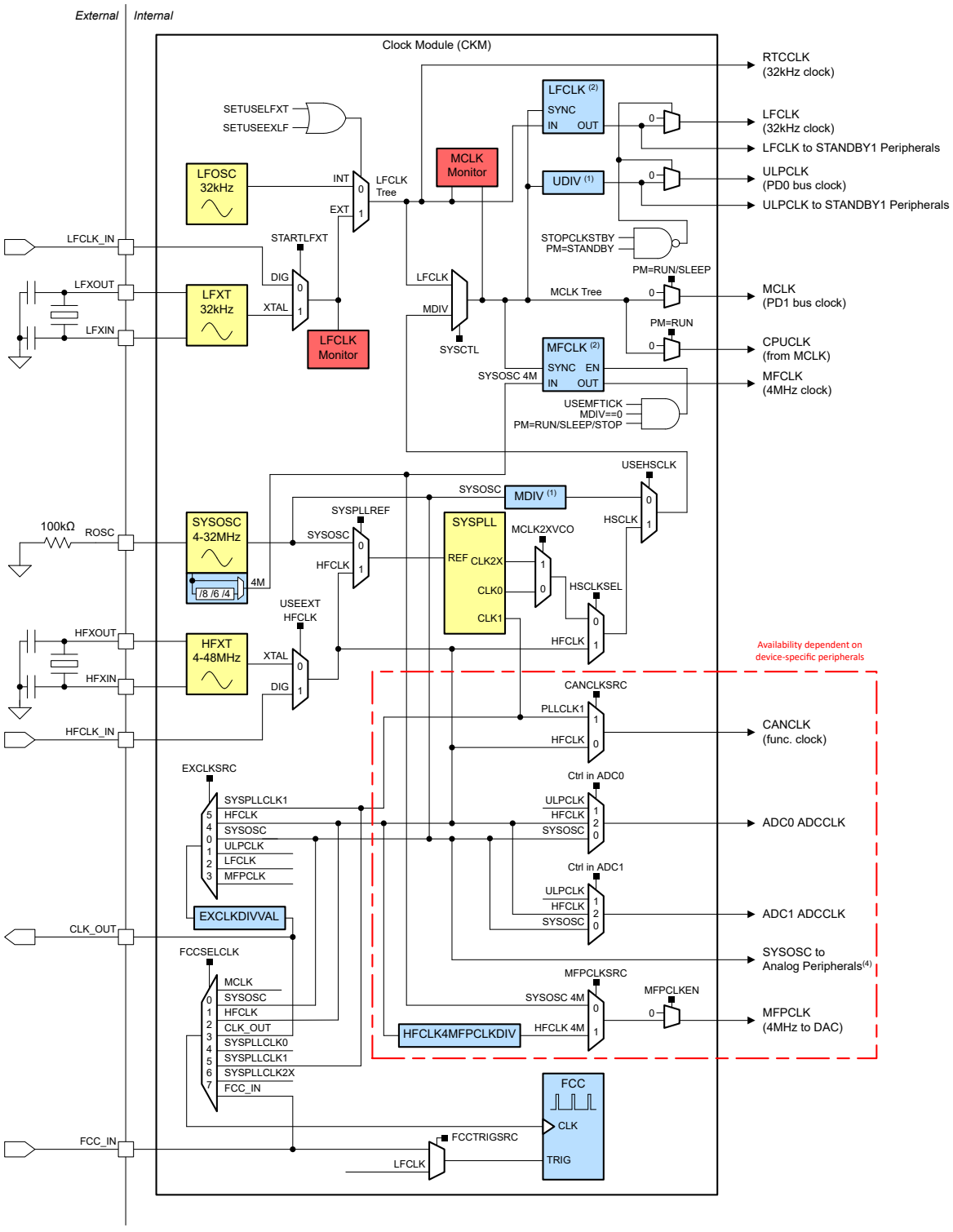



**Figure 2-7. MSPM0Gxx Top Level Clocking Tree**



Copyright © 2026 Texas Instruments Incorporated

1. MDIV and UDIV are clock dividers that support specific use-cases. Review the MCLK and ULPCLK sections, respectively, for specific design considerations regarding MDIV and UDIV.
2. LFCLK and MFCLK are fixed-frequency 32kHz and 4MHz clocks, respectively, that can be selected by certain peripherals for ensuring a constant clock rate even when MCLK or ULPCLK changes source or rate. LFCLK and MFCLK are always synchronized to each other and to MCLK and ULPCLK.
3. Some peripherals (such as select general purpose timers) receive an ungated LFCLK and ULPCLK, enabling them to continue operating even in STANDBY1 when STOPCLKSTBY is asserted to gate the LFCLK and ULPCLK to all other peripherals to save additional power in STANDY mode.
4. SYSOSC is provided to the OPAs for use directly. The OPAs do not request SYSOSC automatically. It is the responsibility of application software to ensure that SYSOSC is enable and running at the necessary frequency to support correct OPA operation before enabling the OPA.

#### 2.3.3.1 Peripheral Clock Source Selection

Most peripherals on the device contain an input clock selection mux which is used to select, and optionally divide down, the functional clock for the peripheral. Figure 2-8 shows the superset peripheral clock selection mux and optional clock divider. Note that not every peripheral has every clock source shown in Figure 2-8. For example, accelerators such as CRC, AES, and DMA run off of the bus clock. There is no option to select MFCLK or LFCLK for these peripherals. To determine the available clock sources for a peripheral, see the chapter for the specific peripheral and review the clock input selections.

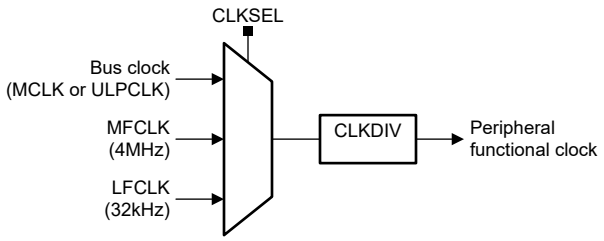



**Figure 2-8. Peripheral Clock Selection Mux and Divider**

### Exceptions

There are also several peripherals that have a unique clock selection scheme and do not use the standard peripheral clock mux shown above. Typically this is due to a requirement for a peripheral to have a clock source that is asynchronous to the rest of the system. Cases where this occurs include:

- The real-time clock (RTC), where RTCCLK is used directly and there is not a clock selection option
- An analog-to-digital converter (ADC), where ADCCLK has a special selection mux to take asynchronous clock sources directly (the ADC sampling clock is selected with a special selection in the ADC control registers)
- The CAN-FD controller, where the CANCLK can be the a PLL output or the HFCLK directly (the CAN-FD functional clock source is selected in SYSCTL)
- The DAC, where MFPCLK is used directly and there is not a clock selection option

### *2.3.4 Clock Monitors*

Several hardware clock monitors are provided to ensure that the CKM is functioning properly. Clock faults are processed through SYSCTL and result in either a brownout reset (in the event of a fatal fault) or a SYSCTL interrupt.

#### 2.3.4.1 LFCLK Monitor

A low-power analog continuously operating clock monitor is provided to ensure that LFCLK is running when it is not sourced internally (for example, when LFCLK is sourced from LFXT or LFCLK_IN and not from LFOSC). The LFCLK monitor is only intended to check for clock stuck faults. It is not intended to be used to verify that the frequency of LFCLK is within a specific tolerance.



Copyright © 2026 Texas Instruments Incorporated

### Enabling the LFCLK Monitor

The LFCLK monitor is disabled at startup after a BOR reset or a fatal clock fault reset.

If the STARTLFXT bit in the LFXTCTL register is set, the LFCLK monitor monitors the LFXT. If the STARTLFXT bit is left cleared, the LFCLK monitor monitors the LFCLK_IN.

Before enabling the LFCLK monitor, make sure that the source clock has started and is operating at 32kHz. If STARTLFXT is set and the monitor is checking the LFXT, wait for the LFXTGOOD indication before enabling the LFCLK monitor. If STARTLFXT is cleared and the monitor is checking the LFCLK_IN digital clock input, make sure that the external clock signal is active before enabling the LFCLK monitor. If a valid clock is not present when starting the monitor, the monitor will assert a fault.

The LFCLK continuous monitor can be enabled by setting the MONITOR bit in LFCLKCFG register.

**Note**

The LFCLK monitor requires 100µs to become active after being enabled. The LFCLK monitor can be enabled to monitor LFXT or LFCLK_IN before switching the LFCLK source from the internal LFOSC.

### LFCLK Monitor Fault Handling

If an LFCLK stuck fault is detected, the system responds in one of two ways depending on the system clock configuration:

- If LFCLK is the source for MCLK (for example, if the entire system is running at 32kHz derived from LFCLK), then a LFCLK fault is considered fatal and an elevated BOOTRST is asserted. Following this elevated scenario, LFCLK will be sourced from LFOSC and the LFCLK monitor is disabled.
- If LFCLK is not the source for MCLK at the time of the failure or after the failure, an LFCLK fault asserts a nonmaskable interrupt (NMI) to the processor so that the application can immediately handle the fault and take any necessary action.

**Note**

A typical BOOTRST does not reset the LFCLK source selection and LFCLK monitor configuration. Only this elevated BOOTRST caused by a fatal clock fault will transition the LFCLK to select LFOSC.

**Note**

In all BOOTRST scenarios, MCLK will be sourced from SYSOSC at BASE frequency.

### Falling Back to LFOSC After LFXT Failure

In the event that an LFXT or LFCLK_IN failure is detected and the failure is due to a system level / PCB level issue that is preventing the external clock source from being able to run reliably, it is possible to fall back to the internal 32kHz LFOSC rather then continuously trying to use the LFXT or LFCLK_IN. In this case, the following procedure is recommended:

1. Configure the device with LFXT or LFCLK_IN sourcing LFCLK and with the LFCLK monitor enabled, per the procedure given above in **Enabling the LFCLK Monitor**.
2. In the event of an LFCLK monitor fault detection:
   a. If MCLK is sourced from LFCLK, the failure will generate a BOOTRST automatically.
   b. If MCLK is not sourced from LFCLK, an NMI will be generated. In the NMI handling, application software can switch MCLK to LFCLK when the LFCLK fault is detected. This action will trigger the hardware to generate a BOOTRST automatically. Switching the MCLK source to LFCLK may be done by changing the operating mode from RUN0 to RUN1/RUN2 or STOP2/STANDBY.
3. Following the BOOTRST, MCLK will be sourced from SYSOSC at BASE frequency, and LFCLK will be sourced from LFOSC. Application software can check the cause of the reset was a fatal clock fault and decide what action to take.



Copyright © 2026 Texas Instruments Incorporated

a. Application software may attempt again to use LFXT/LFCLK_IN. If the LFCLK fails again, application software can return to step 2 above to again switch LFCLK back to LFOSC.
b. To track the number of LFCLK failures, application software can store diagnostic information in the shutdown memory of the device. The SHUTDNSTOREx memory locations are retained through a BOOTRST. As such, it is possible for the application software to track how many LFCLK failures have occurred, and if that number exceeded some threshold, application software can elect to remain with LFOSC as the LFCLK source rather then again attempting to use the LFXT/LFCLK_IN.

#### 2.3.4.2 MCLK Monitor

A digital clock monitor can be used with MCLK. The MCLK monitor asserts an MCLK fault if there is no MCLK activity for a period of 1-12 LFCLK cycles. An MCLK fault is always considered fatal to the system and generates a BOOTRST.

The MCLK monitor can be enabled after the LFCLK is configured and running. To enable the MCLK monitor, set the MCLKDEADCHK bit in the MCLKCFG register in SYSCTL. When enabled, the MCLK monitor runs in all operating modes except for STANDBY1 and SHUTDOWN.

#### 2.3.4.3 Startup Monitors

Clock startup monitors are provided for application software to check that the LFOSC, LFXT/LFCLK_IN, HFXT/HFCLK_IN, SYSPLL, and HSCLK sources are alive before they areselected by software to be used to source a clock in the system. When a clock source has started successfully and is ready, a GOOD indication is given in the CLKSTATUS register in SYSCTL and an interrupt is generated. The startup monitors only provide a status indication when a related clock system configuration change is made. When an initial GOOD indication is given, the clock is not continuously monitored by the startup monitor. Continuous monitoring is provided for *LFCLK* and *MCLK*.

##### *2.3.4.3.1 LFOSC Startup Monitor*

The LFOSC is started automatically after a BOOTRST. Please refer to the device-specific data sheet for the LFOSC startup time. A startup monitor is provided to indicate to the application software when LFOSC startup has completed, at which time the LFCLK is available for use by peripherals. When LFSOSC startup has completed, the LFOSC startup monitor asserts the LFOSCGOOD bit in the CLKSTATUS register in SYSCTL and the LFOSCGOOD interrupt is asserted to alert the application.

##### *2.3.4.3.2 LFXT Startup Monitor*

The LFXT takes time to start after being enabled. A startup monitor is provided to indicate to the application software that the LFXT has successfully started, at which point the LFXT can be selected as the LFCLK source. Once the LFXT startup has completed, the LFXT startup monitor will assert the LFXTGOOD bit in the CLKSTATUS register in SYSCTL and the LFXTGOOD interrupt will be asserted to alert the application. See the device-specific data sheet for the LFXT startup time.

##### *2.3.4.3.3 HFCLK Startup Monitor*

The HFXT takes time to start after being enabled. A startup monitor is provided to indicate to the application software if the HFXT has successfully started, at which point the HFCLK can be selected to source a variety of system functions. The HFCLK startup monitor also supports checking the HFCLK_IN digital clock input for a clock stuck fault.

To enable the HFCLK startup monitor, clear the HFCLKFLTCHK bit in the HFCLKCLKCFG register in SYSCTL (the default state is disabled).

When HFXT is started or the HFCLK_IN is selected as the HFCLK source, the HFCLKGOOD and HFCLKOFF bits in the CLKSTATUS register in SYSCTL are cleared.

In the case of HFXT being used, after the specified HFXT startup time has expired the HFXT status is tested. If the HFXT started successfully, the HFXT startup monitor asserts the HFCLKGOOD bit in the CLKSTATUS



Copyright © 2026 Texas Instruments Incorporated

register and the HFCLKGOOD interrupt is also asserted. If the HFXT did not start within the specified startup time, the HFCLKOFF bit is set, indicating that the HFXT was dead at startup.

In the case of HFCLK_IN being used, after HFCLK_IN is selected a clock stuck check is performed. If the clock is alive, the HFCLKGOOD bit is set in the CLKSTATUS register and the HFCLKGOOD interrupt is also asserted. If the HFCLK_IN signal was stuck, the HFCLKOFF bit in CLKSTATUS register is set, indicating that the HFCLK_IN is dead.

If desired, checking of the HFCLK by the HFCLK startup monitor can be left disabled by keeping the HFCLKFLTCHK bit set in the HFCLKCLKCFG register in SYSCTL.

**Note**

The HFCLK must be in a stable state before attempting to enter STOP or STANDBY low power modes. Before entering STOP or STANDBY, make sure that either the HFCLKGOOD bit or the HFCLKOFF bit is set.

#### *2.3.4.3.4 SYSPLL Startup Monitor*

The SYSPLL takes time to start and settle after being enabled. A startup monitor is provided to indicate to the application software if the SYSPLL has successfully started, at which point the clock outputs from the SYSPLL can be selected to source a variety of system functions.

When the SYSPLL is started, the SYSPLLGOOD and SYSPLLOFF bits in the CLKSTATUS register in SYSCTL are cleared. After the startup/settling time has expired, the SYSPLL status is tested. If the SYSPLL started successfully, the SYSPLL startup monitor will assert the SYSPLLGOOD bit in the CLKSTATUS register and the SYSPLLGOOD interrupt will also be asserted. If the SYSPLL did not start within the specified time, the SYSPLLOFF bit will be set, indicating that the SYSPLL was dead at startup.

**Note**

The SYSPLL must be in a stable state before attempting to enter STOP or STANDBY low power modes. Before entering STOP or STANDBY, make sure that either the SYSPLLGOOD bit or the SYSPLLOFF bit is set.

#### *2.3.4.3.5 HSCLK Status*

The HSCLK is sourced from the HFCLK or the SYSPLL. The CLKSTATUS register in SYSCTL provides HSCLKGOOD and HSCLKDEAD indications, which indicate if the selected HSCLK source started successfully with a GOOD status or failed with a DEAD status, respectively. SYSCTL will not switch MCLK over to HSCLK, even if requested, if HSCLKGOOD is not set.

In addition, the HSCLKSOFF bit is provided in CLKSTATUS to indicate if all HSCLK sources (SYSPLL,HFCLK) are either off (disabled) or started with a DEAD status.

### *2.3.5 Frequency Clock Counter (FCC)*

The frequency clock counter (FCC) enables flexible in-system testing and calibration of a variety of oscillators and clocks on the device. The FCC counts the number of clock periods seen on the selected source clock within a known fixed trigger period (derived from a secondary reference source) to provide an estimation of the frequency of the source clock.

Application software can use the FCC to measure the frequency of the following source oscillators and clocks (selected through the FCCSELCLK field in the GENCLKCFG register):

- MCLK
- SYSOSC
- HFCLK
- CLK_OUT
- SYSPLL (any of the two SYSPLL outputs)
- The external FCC input (FCC_IN)



Copyright © 2026 Texas Instruments Incorporated

The reference clock used to set the trigger time over which pulses of the source clock are counted is configurable (through the FCCTRIGSRC field in the GENCLKCFG register), and can be driven by:

- The external FCC input (FCC_IN)
- LFCLK
- The output of the mux of LFOSC, LFXT, or LFCLK_IN (32kHz)

The trigger time period can be set in one of two ways (through the FCCLVLTRIG field in the GENCLKCFG register):

- Level triggered (one rising edge to one falling edge of the reference clock input). Please note that LFCLK_IN cannot be used as a trigger clock source if using level triggering.
- Rising-edge to rising-edge triggered, for a defined number of clock periods of the reference clock (selectable from 1 to 32 through the FCCTRIGCNT field in the GENCLKCFG register)

When the trigger source is selected as the external FCC input in level-triggered mode, a user-specified counting period can be set by applying a logic high pulse on the FCC_IN pin of the desired trigger length.

When the trigger source is selected as the LFXT, using rising-edge to rising-edge triggering causes the FCC to capture the number of source clock pulses which occurred within 1 to 32 32.768kHz clock periods of the LFXT (30.5µs).

The FCC counter is 22 bits and supports counting from 0 up to $2^{22} - 1 = 4{,}194{,}303$.

While the external FCC input (FCC_IN function) can be used as either the FCC clock source or the FCC trigger input, the signal cannot be used for both functions during the same FCC capture. FCC_IN must be configured as either the FCC clock source or the FCC trigger.

#### 2.3.5.1 Using the FCC

**Rising-Edge to Rising-Edge Triggered Mode with FCC_IN Trigger**

The following steps describe how to use the FCC to count the number of source clock pulses within the trigger period set by the reference clock, with the FCC_IN pin being selected as the reference clock and the SYSOSC being selected as the source clock. This example would be useful for calibrating the SYSOSC frequency with respect to an accurate clock source provided to the FCC_IN pin externally.

1. Set the source clock to SYSOSC by configuring the FCCSELCLK field in the GENCLKCFG register.
2. Set the reference clock to FCC_IN by clearing the FCCTRIGSRC bit in the GENCLKCFG register.
3. Select rising-edge to rising-edge triggering by clearing the FCCLVLTRIG bit in the GENCLKCFG register.
4. Select the desired number of reference clock periods to count the source clock over in the FCCTRIGCNT field in the GENCLKCFG register.
5. Ensure that SYSOSC is enabled at the desired frequency, and that the external clock source connected to FCC_IN is constantly running before continuing.
6. Write the GO bit and KEY field to the FCCCMD register to start the FCC capture on the next trigger clock period.
7. Poll the FCCDONE status bit in the CLKSTATUS register. When the capture completes, FCCDONE will be set by hardware. FCCDONE is read-only and is automatically cleared by hardware when a new capture is started.
8. Extract the resulting count from the 22-bit DATA field in the FCC register.

**Note**

For an accurate count of FCC cycles, it is best to wait at least 6 pulses of FCC before setting FCCTRIGCNT=0x0



Copyright © 2026 Texas Instruments Incorporated

### Rising-Edge to Rising-Edge Triggered Mode with LFXT Trigger

The following steps describe how to use the FCC to count the number of source clock pulses within the reference clock period, with the LFXT being selected as the reference clock and the SYSOSC being selected as the source clock. This example would be useful for calibrating the SYSOSC frequency with respect to an accurate 32.768kHz watch crystal.

1. Set the source clock to SYSOSC by configuring the FCCSELCLK field in the GENCLKCFG register.
2. Set the reference clock to LFXT by setting the FCCTRIGSRC bit in the GENCLKCFG register.
3. Select rising-edge to rising-edge triggering by clearing the FCCLVLTRIG bit in the GENCLKCFG register.
4. Select the desired number of reference clock periods to count the source clock over in the FCCTRIGCNT field in the GENCLKCFG register.
5. Ensure that SYSOSC is enabled at the desired frequency, and that the LFXT is running correctly before continuing.
6. Write the GO bit and KEY field to the FCCCMD register to start the FCC capture on the next trigger clock period.
7. Poll the FCCDONE status bit in the CLKSTATUS register. When the capture completes, FCCDONE will be set by hardware. FCCDONE is read-only and is automatically cleared by hardware when a new capture is started.
8. Extract the resulting count from the 22-bit DATA field in the FCC register. If SYSOSC was running at 32MHz and FCCTRIGCNT was set to '0' (one reference clock period), the result should be approximately 976 cycles counted within the single 32.768kHz period.
   a. To calibrate SYSOSC for 24MHz operation, the SYSOSC user trim must be adjusted until approximately 732 cycles are counted.
   b. To calibrate SYSOSC for 16MHz operation, the SYSOSC user trim must be adjusted until approximately 488 cycles are counted.

   In general, increasing the FCCTRIGCNT value increases the accuracy of the measurement, at the expense of longer measurement time.

### Level Triggered Mode with FCC_IN Trigger and HFCLK_IN Clock

The following steps describe how to use the FCC to count the number of source clock pulses within one external reference pulse window, with HFCLK_IN being selected as the source clock. This example would be useful for measuring the frequency of an external clock source with respect to a fixed pulse width driven by an external signal.

1. Set the source clock to HFCLK by configuring the FCCSELCLK field in the GENCLKCFG register.
2. Set the trigger clock to the FCC_IN pin function by clearing the FCCTRIGSRC bit in the GENCLKCFG register.
3. Set level triggering by setting the FCCLVLTRIG bit in the GENCLKCFG register.
4. Ensure that IOMUX is configured for FCC_IN, that HFCLK is configured for HFCLK_IN, and that an external clock is sourcing HFCLK_IN.
5. Write the GO bit and KEY field to the FCCCMD register to start the FCC capture when FCC_IN goes logic high. Note that if FCC_IN is already logic high when GO is asserted, counting starts immediately. When using level mode, FCC_IN should be low when GO is set, and the trigger pulse should be sent to FCC_IN after GO is set.
6. Poll the FCCDONE status bit in the CLKSTATUS register. When the capture completes, FCCDONE will be set by hardware. FCCDONE is read-only and is automatically cleared by hardware when a new capture is started.
7. Extract the resulting count from the 22-bit DATA field in the FCC register.

#### 2.3.5.2 FCC Frequency Computation and Accuracy

The frequency of the source clock can be computed after capture if the trigger time is known. The frequency is computed by dividing the number of source clock cycles captured by the trigger time. For example, if the trigger



Copyright © 2026 Texas Instruments Incorporated

source was a 32.768kHz clock, the trigger mode was rising-edge to rising-edge, and the period count was 1, then the trigger time is one 32.768kHz clock period (30.5µs). If the captured source count were to come back as 122, the frequency of the source clock is computed as 122 divided by 30.5µs, giving a source clock frequency of approximately 3.99MHz.

$$f_{source} = FCC.DATA / ((GENCLKCFG.FCCTRIGCNT+1) / f_{ref}) \quad (10)$$

The FCC is accuracy is dependent on the trigger clock accuracy as well as the total number of clock cycles captured. The FCC intrinsic error is ≤2 source clock cycles per capture due to synchronization of the trigger to the source clock. Therefore, the impact of these two clock cycles is reduced as more cycles are counted (as the trigger time is increased and/or the source clock frequency is increased). Approximate intrinsic error of the FCC for various source clock frequencies captured against one 32.678kHz period (FCCTRIGCNT=0) and 32 clock periods (FCCTRIGCNT=31) are given in Table 2-10.

**Table 2-10. FCC Error**

| Use Case (Source Clock Frequency) | FCC Trigger Time | FCC Count Result | FCC Count Uncertainty | Approximate FCC Intrinsic Uncertainty Error |
|---|---|---|---|---|
| 4MHz source clock | 30.5µs | 122 | 2 cycles | 1.6% |
| | 976.6µs | 3906 | | 0.05% |
| 16MHz source clock | 30.5µs | 488 | | 0.4% |
| | 976.6µs | 15625 | | 0.01% |
| 24MHz source clock | 30.5µs | 732 | | 0.27% |
| | 976.6µs | 23437 | | 0.01% |
| 32MHz source clock | 30.5µs | 976 | | 0.20% |
| | 976.6µs | 31250 | | 0.01% |
| 80MHz source clock | 30.5µs | 2441 | | 0.08% |
| | 976.6µs | 78125 | | 0.003% |
| 200MHz source clock | 30.5µs | 6100 | | 0.03% |
| | 976.6µs | 195200 | | 0.001% |

**Note**

When using the FCC_IN signal, it is recommended to have a fast slew rate of 10ns or less on the FCC_IN pin to minimize measurement uncertainty.

## 2.4 System Controller (SYSCTL)

The system controller (SYSCTL) contains all control logic for managing the configuration and state of the PMU and CKM analog circuitry. SYSCTL also provides reset management, NRST and SWD pin mux control, flash bank swap control, and the ECC error handling for flash and SRAM.

All power, clock, and reset configuration is done through the SYSCTL memory-mapped register interface.

### *2.4.1 Resets and Device Initialization*

The SYSCTL manages device reset levels and device initialization.

#### 2.4.1.1 Reset Levels

The device has five reset levels:

1. Power-on reset (POR)
2. Brownout reset (BOR)
3. Boot reset (BOOTRST)



Copyright © 2026 Texas Instruments Incorporated

4. System reset (SYSRST)
5. CPU reset (CPURST)

The relationships between reset levels are given in Figure 2-9.

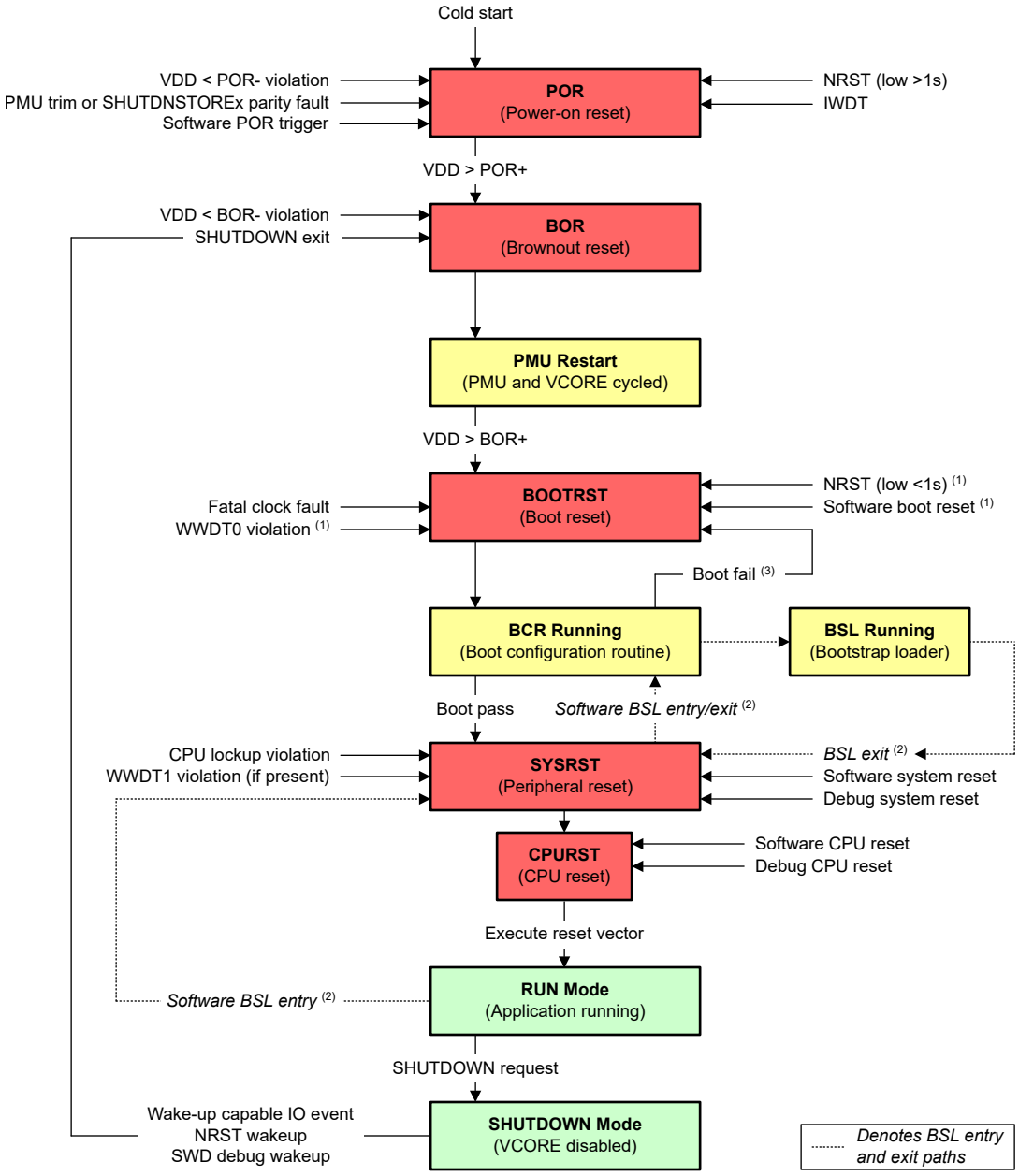



**Figure 2-9. Device Reset Levels**

(1) An NRST (low <1 s), software boot reset, or WWDT0 violation triggered BOOTRST runs the boot configuration routine. This does not reset the RTC, LFXT, LFCLK, LFCLK_IN, and IOMUX configuration of any IO pins used by LFXT or LFCLK_IN. This allows the RTC to keep time through an external reset trigger.

(2) A software-triggered bootstrap loader (BSL) entry command first triggers a SYSRST, after which the boot configuration routine (BCR) is run to authenticate the BSL entry before starting the BSL. After the BSL execution concludes, a SYSRST is generated and the BCR executes again. When the BCR completes, a final SYSRST is issued and the application is started. This entire process does nto reset the RTC, LFXT/LFCLK/LFCLK_IN,



Copyright © 2026 Texas Instruments Incorporated

the IOMUX configuration of any IO pins used by LFXT or LFCLK_IN, and SYSOSC FCL enable configurations, as only a SYSRST reset level is asserted throughout the process. This allows the RTC to keep time through an external reset trigger.

(3) If a boot fail occurs during execution of the boot configuration routine, a BOOTRST can be generated by SYSCTL to attempt the boot process again from the BOOTRST level. See Section 2.4.1.8.

**Note**

SLEEP, STOP, and STANDBY operating modes are not shown in this diagram. These modes originate from and return to RUN mode unless an exception occurs which causes a reset level to be asserted or a mode to be suspended.

#### *2.4.1.1.1 Power-on Reset (POR) Reset Level*

A power-on reset (POR) is a complete device reset.

The following conditions generate a POR:

- The device powers up (cold start)
- A *POR-supply monitor violation*(VDD drops below the POR supply monitor negative-going threshold)
- A parity fault on PMU trim data or the *shutdown memory*
- Software triggers a POR through SYSCTL *(RESETLEVEL 0x03)*
- The NRST pin is held low for more than one second when in NRST mode
- Independent Watchdog Timer (IWDT)

A POR always resets the *shutdown memory*, re-enables the NRST/SWD pin functions (if disabled), and triggers a BOR.

#### *2.4.1.1.2 Brownout Reset (BOR) Reset Level*

A brownout reset (BOR) resets the device power management unit (PMU). All regulated core logic powered from VCORE are power cycled.

The following conditions generate a BOR:

- A *POR*
- A *BOR0- Supply Monitor Violation* (VDD drops below the BOR0- supply monitor lower threshold voltage level)
- An exit from SHUTDOWN mode (through a wakeup-capable IO, NRST, or SWD) if SHUTDOWN mode is active

The following are not reset by a BOR:

- The shutdown memory (SHUTDNSTOREx) if SHUTDOWN mode is present in the device
- The NRST state, if disabled by software
- The SWD state, if disabled by software
- The latched IO pin state, when SHUTDOWN mode is active, and the cause of the BOR is an exit from SHUTDOWN mode (see *SHUTDOWN mode handling*)

A BOR always triggers a *BOOTRST* when VDD > BOR0+ (brownout supply monitor upper threshold voltage level).

**Note**

Not all the devices support SHUTDOWN mode, check the device specific data sheet to determine if it's present or not.

#### *2.4.1.1.3 Boot Reset (BOOTRST) Reset Level*

A boot reset (BOOTRST) triggers execution of the device boot configuration routine and resets the majority of the core logic, system memory (SRAM), and SYSOSC FCL mode (if enabled).



Copyright © 2026 Texas Instruments Incorporated

The BOOTRST does not reset the low-power clocking configuration (RTC, LFCLK, LFXT, LFCLK_IN, and related IOMUX configuration for these features) in the system, unless the BOOTRST was due to a fatal clock fault.

As a result, if LFCLK is configured to run from the LFXT or LFCLK_IN, an NRST pin BOOTRST condition or a software triggered BOOTRST condition does not reset LFCLK to the default internal oscillator (LFOSC). LFCLK continues to run from LFXT or LFCLK_IN and cannot be reconfigured. This lets the RTC maintain a time base through an external reset.

The following conditions generate a BOOTRST:

- A BOR
- A fatal clock failure (see LFCLK Monitor and MCLK Monitor)
- A WWDT0 violation
- Software triggers a BOOTRST through SYSCTL *(RESETLEVEL 0x01)*
- The NRST pin is held low for longer than the minimum reset pulse time but less than one second when in NRST mode
- A BOOTRST followed by a boot failure (re-attempt of a failed boot sequence)

The following are not reset by a BOOTRST:

- The shutdown memory (SHUTDNSTOREx) if SHUTDOWN mode is present
- The NRST disable state, if disabled by software
- The SWD disable state, if disabled by software
- The RTC, LFCLK, LFXT, LFCLK_IN, and related IOMUX configuration for these features (if the cause of the BOOTRST was not a fatal clock fault)

Following a BOOTRST, a SYSRST is always triggered upon successful completion of the boot configuration routine. If the boot configuration routine fails to complete successfully, a BOOTRST is again generated and the boot process is attempted again from the BOOTRST point. The boot process attempts to complete successfully up to 3 times, after which the device state locks until a BOR or POR reset occurs (see Section 2.4.1.8).

#### *2.4.1.1.4 System Reset (SYSRST) Reset Level*

A system reset clears the state of the CPU and all the peripherals, with the exceptions listed below.

The following conditions generate a SYSRST:

- A BOOTRST followed by a boot pass
- Software triggers a SYSRST with BSL entry through SYSCTL *(RESETLEVEL 0x02)*
- A bootstrap loader (BSL) exit (if present), which is always followed by execution of the boot configuration routine (BCR)
- A CPU lockup violation
- A WWDT1 violation (if present)
- Software triggers a SYSRST through SYSCTL *(RESETLEVEL 0x00)*
- The debug subsystem triggers a system reset

The following are not reset by a SYSRST:

- The shutdown memory (SHUTDNSTOREx) (if present)
- The NRST state, if disabled by software
- The SWD state, if disabled by software
- The RTC, LFCLK, and LFXT/LFCLK_IN configuration, including the IOMUX settings for any IO pins configured to be used by LFXT/LFCLK_IN
- The SYSOSC frequency correction loop (FCL), if enabled by software

In most cases, the device is in RUN mode after a SYSRST, and the CPU executes the reset vector and begins execution of the application software.

There are exceptions to this:

- If the SYSRST was triggered with a BSL entry request, the BCR runs followed by the BSL.



Copyright © 2026 Texas Instruments Incorporated

- If the SYSRST was triggered due to a BSL exit request, the BCR runs, followed by an additional SYSRST, after which the application software starts.

#### *2.4.1.1.5 CPU-only Reset (CPURST) Reset Level*

A CPU-only reset clears the state of the CPU logic only. Peripheral states are not affected by a CPU reset. A CPU reset is only generated by software through the CPU AIRCR local register or by the debug subsystem.

### 2.4.1.2 Initial Conditions After Power-Up

After a POR, when the boot process completes and the CPU starts the application, the initial device conditions are as follows:

- The NRST pin is configured in NRST mode
- Serial wire debug (SWD) IO are configured in SWD mode (SWDIO is pulled high, and SWCLK is pulled low)
- All other configurable I/O pins are high impedance (Hi-Z)
- Peripheral modules are reset as described in the respective chapters of this manual
- The device is in RUN mode
- MCLK is sourced from the internal SYSOSC at base frequency
- LFCLK is sourced from the internal LFOSC (note that LFOSC requires time to start up before LFCLK can be used)
- MFCLK is disabled
- MFPCLK is disabled
- SYSPLL is disabled
- HFXT is disabled
- Peripherals are disabled
- Any flash sectors configured to be write protected at boot are write protected

### 2.4.1.3 NRST Pin

After a cold start, the NRST pin is configured in NRST mode. The NRST pin must be high for the device to boot successfully. There is no internal pullup resistor on NRST. External circuitry (either a pullup resistor to VDD or a reset control circuit) must actively pull NRST high for the device to start. After the device is started, a low pulse on NRST <1 second in duration triggers a BOOTRST. If a low pulse on NRST is held for >1 second, a POR is triggered.

Some low pin count devices support reconfiguring the NRST pin to be a GPIO pin. See the pin configuration of the device-specific data sheet to see if GPIO functionality is shared with NRST. Application software can disable the NRST functionality of the NRST pin, allowing GPIO functionality to be enabled. To disable NRST, set the DISABLE bit in the EXRSTPIN register along with the KEY. Then configure IOMUX for the desired functionality.

After the NRST pin function is disabled, only after a POR reset can the NRST pin function be re-enabled.

**Note**

When the NRST pin is shared with the I2C open-drain pin, the user's system must be designed such that the device is powered up and in I2C mode before any transactions occur on the I2C bus. The device can be inadvertently reset due to a low signal on the shared reset or I2C SDA line before this point.

To prevent this, pullups on the NRST pin that are shared with an I2C open-drain IO must be selected to meet the I2C pullup requirements for minimum and maximum values.

### 2.4.1.4 SWD Pins

There are two serial wire debug (SWD) pins present on all devices: SWDIO and SWCLK

After a cold start, the SWD pins are configured in SWD mode to allow a debug connection to be established.

- SWDIO (serial wire data input/output) pin is configured in the IOMUX and the pull-up is enabled
- SWCLK (serial wire clock) pin is configured in the IOMUX and the pull-down is enabled



Copyright © 2026 Texas Instruments Incorporated

It is possible to re-configure the SWD pins as general purpose IO (GPIO) in software to enable use of these pins in an application when debug support is no longer required. To disable SWD functionality, set the DISABLE bit in the SWDCFG register in SYSCTL along with the KEY. Then configure IOMUX for the desired functionality.

Once the SWD pin functions are disabled, they can only be re-enabled by triggering a POR.

#### 2.4.1.5 Generating Resets in Software

Software can generate a software POR, a software BOOTRST, a software SYSRST with bootstrap loader (BSL) entry, or a software SYSRST by issuing the appropriate command to SYSCTL. To issue a reset, first select the desired reset level in the RESETLEVEL register in SYSCTL. Then set the GO bit in the RESETCMD register along with the KEY value.

**Table 2-11. Software Generated SYSCTL Reset Commands**

| LEVEL | Action |
|---|---|
| 0x0 | Software SYSRST |
| 0x1 | Software BOOTRST |
| 0x2 | Software SYSRST with BSL entry |
| 0x3 | Software POR |

A CPU-only reset (CPURST) which does not reset the peripherals can also be triggered in software within the device CPU by setting the SYSRESETREQ bit in the AIRCR local CPU register. See the CPU Sub System chapter for more information.

### Starting the BSL From Software (if present)

The software-triggered BSL entry *(RESETLEVEL 0x02)* is a special case of the SYSRST which provides a mechanism for the application software to start the ROM bootstrap loader (BSL). It is not possible to jump to the bootloader code directly during normal software execution in RUN mode. When application software commands a software-triggered BSL entry *(RESETLEVEL 0x02)*, a SYSRST is generated first, followed by execution of the boot configuration routine (for authentication), after which the BSL is started (if the device security policy has the BSL configured to be enabled).

Once the BSL has completed execution, a second SYSRST is issued and the BCR is executed. When the BCR completes, a final SYSRST is asserted to return control of the system back to the application software.

If the BSL was disabled by the user configuration, and a software-triggered BSL entry is invoked, the device will issue a standard SYSRST instead and return control of the system back to the application software.

Any system configuration which is not reset by a SYSRST is maintained through this entire process. As such, the RTC can continue to run without disruption during execution of a software-triggered BSL entry and exit.

#### 2.4.1.6 Reset Cause

After a device reset occurs, the lowest level reset cause which occurred during reset processing is captured in hardware so that application software can interrogate the reason for the reset and take any appropriate action when starting the application. The lowest level reset cause is encoded into a 5-bit field in the reset cause register in SYSCTL. The contents of the reset cause register are always cleared upon a read, and return zero after being read if no reset has occurred after the read. The reset cause encodings are given in Table 2-12.



Copyright © 2026 Texas Instruments Incorporated

**Table 2-12. Reset Cause Encoding**

| Reset | | | | Device Modules Reset | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reset Level | Cause ID | | Reset Cause | NRST/SWD Disables | SHUTDN STOREx | Core Regulator | Debug Subsystem | RTC, LFXT, LFCLK State | SRAM | BCR Execution | IOMUX | EVENT, DMA, FLASHCTL, SYSCTL | Peripherals | CPU |
| | 0x00 | 0 | No reset since last read | | | | | | | | | | | |
| POR | 0x01 | 1 | VDD < POR- violation | R | R | R | R | R | R | R | R | R | R | R |
| | | | PMU trim parity fault | | | | | | | | | | | |
| | | | SHUTDNSTOREx parity fault (if present) | | | | | | | | | | | |
| | 0x02 | 2 | NRST pin reset (>1s) | R | R | R | R | R | R | R | R | R | R | R |
| | | | IWDT | R | R | R | R | R | R | R | R | R | R | R |
| | 0x03 | 3 | Software-triggered POR | R | R | R | R | R | R | R | R | R | R | R |
| BOR | 0x04 | 4 | VDD < BOR- violation | | | R | R | R | R | R | R | R | R | R |
| | 0x05 | 5 | Wake from SHUTDOWN (if present) | | | R | R | R | R | R | R[1] | R | R | R |
| | 0x06 | 6 | Reserved | | | | | | | | | | | |
| | 0x07 | 7 | Reserved | | | | | | | | | | | |
| BOOTRST | 0x08 | 8 | Reserved | | | | | | | | | | | |
| | 0x09 | 9 | Fatal clock fault | | | | | R | R | R | R | R | R | R |
| | 0x0A | 10 | Reserved | | | | | | | | | | | |
| | 0x0B | 11 | Reserved | | | | | | | | | | | |
| | 0x0C | 12 | NRST pin reset (<1 s) | | | | | R | R | R[2] | R | R | R | |
| | 0x0D | 13 | Software-triggered BOOTRST | | | | | | R | R | R[2] | R | R | R |
| | 0x0E | 14 | WWDT0 violation | | | | | | R | R | R[2] | R | R | R |
| | 0x0F | 15 | Reserved | | | | | | | | | | | |
| SYSRST | 0x10 | 16 | BSL exit (if present) | | | | | | | R | R[2] | R[3] | R | R |
| | 0x11 | 17 | BSL entry (if present) | | | | | | | R | R[2] | R[3] | R | R |
| | 0x12 | 18 | Reserved | | | | | | | | | | | |
| | 0x13 | 19 | WWDT1 violation | | | | | | | | R[2] | R[3] | R | R |
| | 0x14 | 20 | Reserved | | | | | | | | | | | |
| | 0x15 | 21 | CPU lockup violation | | | | | | | | R[2] | R[3] | R | R |
| | 0x16 | 22 | Reserved | | | | | | | | | | | |
| | 0x17 | 23 | Reserved | | | | | | | | | | | |
| | 0x18 | 24 | Reserved | | | | | | | | | | | |
| | 0x19 | 25 | Reserved | | | | | | | | | | | |
| | 0x1A | 26 | Debug-triggered SYSRST | | | | | | | | R[2] | R[3] | R | R |
| | 0x1B | 27 | Software-triggered SYSRST | | | | | | | | R[2] | R[3] | R | R |



Copyright © 2026 Texas Instruments Incorporated

**Table 2-12. Reset Cause Encoding (continued)**

| Reset | | | | Device Modules Reset | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reset Level | Cause ID | | Reset Cause | NRST/SWD Disables | SHUTDN STOREx | Core Regulator | Debug Subsystem | RTC, LFXT, LFCLK State | SRAM | BCR Execution | IOMUX | EVENT, DMA, FLASHCTL, SYSCTL | Peripherals | CPU |
| CPURST | 0x1C | 28 | Debug-triggered CPURST | | | | | | | | | | | R |
| | 0x1D | 29 | Software-triggered CPURST | | | | | | | | | | | R |
| | 0x1E | 30 | Reserved | | | | | | | | | | | |
| | 0x1F | 31 | Reserved | | | | | | | | | | | |

(1) In the case of a SHUTDOWN mode exit, the IOMUX registers are always reset but the IOs retain the last state from the point of entry into SHUTDOWN until the user clears the RELEASE bit in the SHDNIOREL register in SYSCTL. This enables application software to be able to reconfigure IOMUX and any corresponding peripherals before releasing the IO after a SHUTDOWN exit. See shutdown mode handling and IOMUX wake.
(2) IOMUX is reset, but if LFXT or LFCLK_IN are enabled then the IOMUX settings for these pads does not reset. The RTC, LFXT, LFCLK_IN, and LFCLK continue to operate without interruption.
(3) The SYSOSC frequency correction loop (FCL) is not reset (if enabled by software)

If two reset causes occur simultaneously, the lowest cause reset ID value is prioritized and reported. For example, if a WWDT0 violation (cause 0x0E) occurs at the same time that a VDD < BOR- violation (cause 0x04) occurs, the reported reset cause is a BOR- violation (cause 0x04), as this is a lower level reset which clears additional aspects of the device state.

The reset cause encoding enables simple software handling during application startup. The reset cause value can be read by application software and tested to be within a certain value range to determine if the following occurred:

- **RESETCAUSE==0x00:** No reset since last read
- **RESETCAUSE<0x04:** The NRST/SWD disable state was reset and can need to be reconfigured
- **RESETCAUSE<0x04:** The SHUTDNSTOREx memory (if present) was reset and can need to be reconfigured
- **RESETCAUSE<0x08:** The regulated VCORE domain was power cycled
- **RESETCAUSE<0x0C:** The RTC and low frequency clock configuration were reset and can need to be reconfigured
- **RESETCAUSE<0x10:** The SRAM is reinitialized and the contents were lost
- **RESETCAUSE<0x1C:** The peripherals were reset and can need to be reconfigured

The following example shows how the reset cause can be tested to take specific actions when starting an application after a reset:

```
// Read reset cause into SRAM variable
uint8_t cause = RESETCAUSE;

// Handle device re-configuration based on reset cause level
if (cause!=0)
{
    if (cause<0x04)
    {
        // NRST/SWD disable state was lost
        // SHUTDNSTOREx memory state was lost
        // PMU/VCORE domain state was lost
        // RTC/LFXT/LFCLK state was lost
```



Copyright © 2026 Texas Instruments Incorporated

```
    }
    if (cause<0x0C)
    {
        // RTC/LFXT/LFCLK state was lost
    }
    if (cause<0x1C)
    {
        // The peripherals were reset
    }
}
```

#### 2.4.1.7 Peripheral Reset Control

Each peripheral on a device contains a reset control register (RSTCTL) and a status register (STAT).

The STAT register is a read-only register which contains a RESETSTKY bit, indicating if the peripheral was reset. This bit can be read by application software to determine if a peripheral was reset and needs to be re-configured. The RESETSTKY bit is cleared by writing the RESETSTKYCLR bit together with the KEY value to the RSTCTL register.

Application software can also force a reset of the peripheral by writing the RESETASSERT bit together with the KEY value to the RSTCTL register. This action resets the peripheral to the default state and sets the RESETSTKY bit in the STAT register.

**Note**

The RSTCTL register does not reset the FPUB and FSUB registers for a given peripheral. Use SYSRST or directly modify FPUB and FSUB directly of the peripheral to reset the publisher and subscriber event registers.

#### 2.4.1.8 Boot Fail Handling

If a boot fails during execution of the boot configuration routine (BCR), SYSCTL asserts a BOOTRST to attempt another boot. A boot fail can be caused by the following:

- Boot configuration data integrity error
- Device trim integrity error
- BCR timeout (BCR takes significantly longer than expected to complete for any other reason)

Up to three attempts to successfully boot the device are made by hardware. If the first, second, or third boot attempt is successful, the application starts normally. If the third attempt fails, then the boot process fails, no further boot attempts are made, and the application software is not started.

The purpose of the additional boot attempts is to allow the device to boot correctly if a transient (temporary) error was the cause of the boot fail. If three boot attempts are not successful, a steady-state error condition is likely present and the application is not started to prevent unexpected operation.

**Note**

If a device is locked due to three failed attempts to boot, and a BOR- violation occurs, a BOR and BOOTRST are still generated (by definition) and a single boot attempt is made. Under the same conditions, if power is completely removed from the device (triggering a POR- violation), then the device again attempts to boot up to three times.

### *2.4.2 Operating Mode Selection*

The device operating mode is configured through the use of the following:

1. Policy bits in the SYSOSCCFG and MCLKCFG registers in SYSCTL (to control the behavior of SYSOSC in RUN, SLEEP, and STOP modes)
2. Policy bits in the PMODECFG register in SYSCTL (to set the deep sleep level of STOP, STANDBY or SHUTDOWN (if present))
3. SLEEPDEEP policy bit in the SCR local CPU register (to select whether a WFI instruction triggers SLEEP mode or STOP/STANDBY/SHUTDOWN mode (if present))



Copyright © 2026 Texas Instruments Incorporated

4. Use of the Arm WFI (wait for interrupt) CPU instruction (to enter the configured SLEEP/STOP/STANDBY/ SHUTDOWN state (if present))

Before entering an operating mode where the CPU is disabled, make sure that the appropriate peripheral that can wake the CPU from sleep has been configured to generate a CPU interrupt on the desired event.

For a detailed description of the behavior of each operating mode, see the operating modes section.

### Policy Bit Configuration

Table 2-13 defines how to configure the relevant policy bits for each operating mode. All values are indicated in binary format. A dash (-) indicates that the particular policy bit is a don't care for the specified operating mode.

**Table 2-13. Operating Mode Policy Bit Configuration**

| Operating Mode Policy Control | | RUN | | | SLEEP(2) | | | STOP | | | STANDBY | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Register | Bit | RUN0 | RUN1 | RUN2 | SLEEP0 | SLEEP1 | SLEEP2 | STOP0 | STOP2 (if present) | STOP2 | STANDBY0 | STANDBY1 | SHUTDOWN (if present) |
| SYSOSCCFG | DISABLE(1) | 0 | 0 | 1 | 0 | 0 | 1 | - | - | 1(3) | - | - | - |
| | USE4MHZSTOP | - | - | - | - | - | - | 0 | 1 | 0 | - | - | - |
| | DISABLESTOP | - | - | - | - | - | - | 0 | 0 | 1(3) | - | - | - |
| MCLKCFG | USELFCLK(1) | 0 | 1 | - | 0 | 1 | - | 0 | 0 | - | - | - | - |
| | STOPCLKSTBY | - | - | - | - | - | - | - | - | - | 0 | 1 | - |
| PMODECFG | DSLEEP | - | - | - | - | - | - | 00 | 00 | 00 | 01 | 01 | 10 |
| SCR | SLEEPDEEP | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |

(1) The SYSOSCCFG.DISABLE and MCLKCFG.USELFCLK policy bits take effect immediately after being configured, as these bits affect the RUN mode behavior. Other policy bits only take effect when the CPU is put into deep sleep.
(2) SLEEP mode behavior is always identical to RUN mode, except with the CPUCLK disabled. As such, the SLEEP behavior is determined by the configuration of RUN mode.
(3) The STOP2 policy for STOP mode can be configured by setting the DISABLESTOP bit or DISABLE bit in the SYSOSCCFG register before entering DEEPSLEEP. When DISABLESTOP is set and DISABLE is cleared, SYSOSC is only disabled when DEEPSLEEP is requested. SYSOSC continues to run in RUN and SLEEP modes. When DISABLE is set, DISABLESTOP becomes a don't care, and SYSOSC is disabled immediately and is kept disabled in STOP mode.

### Entering SLEEP Mode

Entering **SLEEP** mode disables the CPU, but otherwise maintains the same configuration as **RUN**. To enter **SLEEP** mode:

1. Configure the CPU for SLEEP by clearing the SLEEPDEEP bit in the local SCR register.
2. Enter sleep mode by executing the WFI (wait for interrupt) CPU instruction.

### Entering STOP or STANDBY Modes

To enter **STOP** or **STANDBY** mode:

1. Configure the PMODECFG register in SYSCTL to 0b00 (STOP) or 0b01 (STANDBY).
2. Configure the CPU for DEEP SLEEP by setting the SLEEPDEEP bit in the local SCR register.
3. Enter sleep mode by executing a WFI (wait for interrupt) CPU instruction.



Copyright © 2026 Texas Instruments Incorporated

**Entering SHUTDOWN Mode (if present)**

To enter **SHUTDOWN** mode:

1. Configure the PMODECFG register in SYSCTL to 0b10 (SHUTDOWN).
2. Configure the CPU for DEEP SLEEP by setting the SLEEPDEEP bit in the local SCR register.
3. Enter sleep mode by executing a WFI (wait for interrupt) CPU instruction.

### *2.4.3 Asynchronous Fast Clock Requests*

Peripherals are configured to asynchronously assert a hardware request to the SYSCTL for a fast clock source if the device is operating in lower power states (such as RUN2, SLEEP1/2, STOP0/1/2, or STANDBY0/1).

This mechanism is for applications where the MCLK/ULPCLK tree is normally sourced from either LFCLK (at 32kHz) or SYSOSC, but a faster clock is temporarily needed to quickly handle a peripheral event (for example, a timer IRQ or GPIO IRQ) or peripheral activity (such as serial communication or an ADC conversion).

Asynchronous fast clock requests are also useful for scenarios where the device is running in STANDBY1 mode. In STANDBY1 (when STOPCLKSTBY is set), the ULPCLK and LFCLK are disabled to all peripherals except for a few TIMGx, leaving the supported TIMGx and RTC as the only clocked peripherals. To wake up the device from this state where the PD0 bus clock (ULPCLK) is disabled, a TIMGx or RTC interrupt request forces an asynchronous fast clock request to wake the device to RUN mode. Other peripherals can also wake the device from this state if support for detecting an asynchronous event (for example GPIO, comparator, and serial interfaces) is configured.

Asynchronous fast clock requests temporarily provide peripherals with bus clock ( MCLK/ULPCLK), sourced from the SYSOSC, for the duration of the request. MFCLK, if enabled for use, is also enabled during the asynchronous request.

**Asynchronous Fast Clock Behavior**

When configured, SYSCTL responds to a peripheral fast clock request in the following way:

1. If the device is currently in a STOP or STANDBY mode, the low power state is temporarily suspended to support running the PD0 bus clock (ULPCLK) at the SYSOSC base frequency.
2. If disabled, SYSOSC is forced to be enabled; if SYSOSC is already running but at a different frequency than base frequency, SYSOSC is forced to base frequency.
3. The MCLK/ULPCLK tree is forced to be sourced from SYSOSC at the base frequency; if the device is in RUN mode then the CPUCLK is also switched to the SYSOSC rate (the CPUCLK is always derived from MCLK)
4. If the MFCLK is configured to be used, MFCLK is activated

After the configuration above is applied, values are held for the duration of time that the asynchronous request remains asserted plus approximately 1µs after the request is removed. The system then returns to the configuration which existed before the fast clock request, provided the CPU did not change the configuration during the request.

Asynchronous fast clock requests are ignored and results in no effect on the device configuration if any of the following are true:

- MCLK is already sourced from SYSOSC at base frequency
- MCLK is sourced from HSCLK (HFCLK or SYSPLL)
- Asynchronous fast clock requests are globally blocked by setting the BLOCKASYNCALL bit in the SYSOSCCFG register in SYSCTL

**Peripheral Support**

An asynchronous event can occur from various sources listed below. The purpose, request source, and configuration requirements for these peripherals are given in Table 2-14.



Copyright © 2026 Texas Instruments Incorporated

- Timers such as RTC, TIMGx, and TIMBx
- GPIO
- Comparator
- Serial interfaces such as UART, I2C, SPI
- ADC

**Table 2-14. Peripheral Support for Asynchronous Fast Clock Requests**

| Peripheral | Purpose | Request Source | Configuration |
|---|---|---|---|
| RTC | Fast CPU wake from RTC event | RTC IRQ to CPU | The RTC can be configured to generate an asynchronous fast clock request in response to an RTC IRQ in any operating mode by clearing the BLOCKASYNC bit in the RTC CLKCFG register; this provides the lowest latency RTC event response. In STANDBY1 mode, an RTC IRQ event always generates an asynchronous fast clock request; this is needed to wake the device as the main ULPCLK is disabled to reduce power consumption. |
| TIMGx | Fast CPU wake from TIMGx event | TIMGx, IRQ to CPU | An IRQ event from TIMGx generates an asynchronous fast clock request when the corresponding IMASK interrupt is set in the TIMG registers. This is needed to wake the device as the ULPCLK is disabled to reduce power consumption. This request is also supported in STANDBY1 mode for specific TIMGx peripherals. |
| TIMBx | Fast CPU wake from TIMBx event | TIMBx, IRQ to CPU | An IRQ event from TIMBx generates an asynchronous fast clock request when the corresponding IMASK interrupt is set in the TIMB registers. This is needed to wake the device as the ULPCLK is disabled to reduce power consumption. This request is also supported in STANDBY1 mode for specific TIMBx peripherals. |
| GPIO | Fast CPU wake from GPIO event | GPIO activity | The GPIO generates an asynchronous fast clock request through the GPIO configuration registers. This is for applications where GPIO wake from lower power modes is desired, as the fast clock request results in the GPIO digital glitch filters running at SYSOSC base frequency. In addition to configuring the GPIO registers to request the fast clock, the BLOCKASYNCALL bit must be cleared in the SYSOSCCFG register to allow the request to propagate. |
| Comparator | Fast wake from a comparator event | Comparator event | Comparator output toggle generates an asynchronous fast clock request when in ULP mode to provide the lowest latency comparator event response. The BLOCKASYNC bit in the CLKCFG register of the respective comparator must be disabled. |
| SPI | Temporarily use fast clock for bit clock generation | SPI activity | SPI activity generates an asynchronous fast clock request when the BLOCKASYNC bit is cleared in the CLKCFG register of the respective SPI peripheral. |
| I2C | Temporarily use fast clock for bit clock generation | I2C activity | I2C activity generates an asynchronous fast clock request when the BLOCKASYNC bit is cleared in the CLKCFG register of the respective I2C peripheral. |
| UART | Temporarily use a fast clock for baud rate generation | UART activity | UART activity generates an asynchronous fast clock request when the BLOCKASYNC bit is cleared in the CLKCFG register of the respective UART peripheral. |
| ADC | Temporarily run the SYSOSC to support timer-triggered ADC operation from a low-power mode | ADC | If an ADC conversion is triggered when SYSOSC is disabled, an asynchronous fast clock request is generated to enable the SYSOSC (SYSOSC is required for correct ADC operation). |

### Fast CPU Event Handling

In addition to the peripheral event and activity fast clock request triggers, the SYSCTL can be configured to generate an asynchronous fast clock request upon any IRQ request to the CPU. This provides the lowest latency interrupt handling when the system is running at the LFCLK rate (32kHz), as the IRQ request propagates through the wake-up logic at the SYSOSC rate vs. the LFCLK rate (32kHz). When the FASTCPUEVENT bit



Copyright © 2026 Texas Instruments Incorporated

is set in the SYSOSCCFG register in SYSCTL, any interrupt request to the CPU also generates a fast clock request.

### Asynchronous Fast Clock Request Logic

The logic for asserting a fast clock request is given in the following figure.

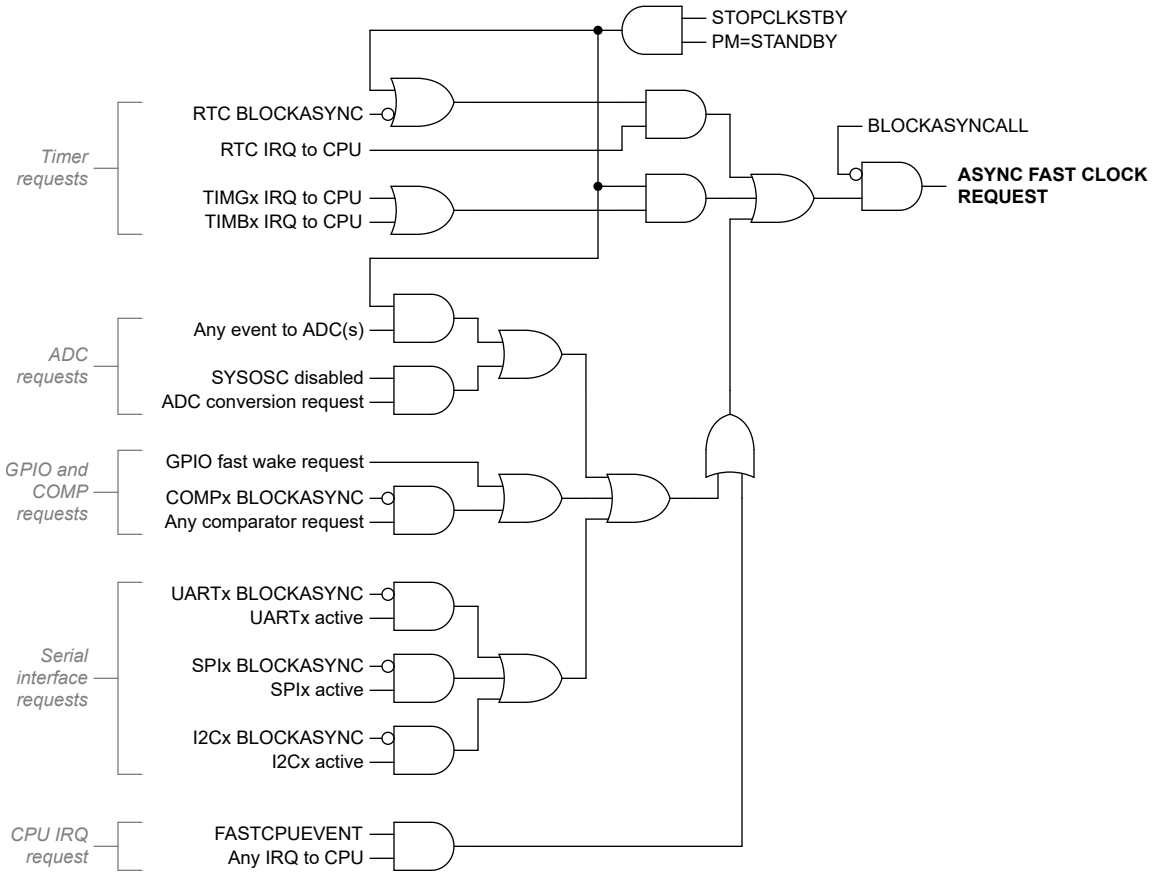



**Figure 2-10. Asynchronous Fast Clock Request Logic**

**Note**

Check the device specific data sheet to determine which peripherals are present and supported.

### *2.4.4 SRAM Write Protection*

SRAM Write Protection Control for this device is implemented in the Global Security Controller (GSC). Please refer to the related chapter for more information.

### *2.4.5 Flash Wait States*

Flash wait states are managed automatically by SYSCTL when MCLK is running from SYSOSC or LFCLK. If MCLK will not be switched to SYSPLL, HFXT, or HFCLK_IN, no wait state configuration is required. Application software is not required to configure any wait states for proper operation of the device.

If MCLK is to be configured to run from one of the high-speed clock sources such as SYSPLL, HFXT, or HFCLK_IN, the flash wait state configuration in MCLKCFG.FLASHWAIT is applied. By default, MCLKCFG.FLASHWAIT is set to 0x2 (2 wait states) which supports operation at the maximum MCLK frequency of 80MHz. If MCLK is configured to run from a high-speed clock but the MCLK frequency allows operation with fewer than 2 wait states, then MCLKCFG.FLASHWAIT can be reduced.



Copyright © 2026 Texas Instruments Incorporated

Refer to the *Recommended Operation Conditions* section of the device specific data sheet to determine the max clock frequency supported with 0 or 1 wait state.

### *2.4.6 Flash Bank Address Swap*

Devices with multiple flash banks provide a mechanism for swapping the address space of the upper banks with the address space of the lower banks to enable firmware updates, where the firmware image has no knowledge of the containing memory bank, to be able to execute properly on the hardware.

To swap the addresses of the upper flash banks with the addresses of the lower flash banks, set the USEUPPER bit in the FLBANKSWAP register while also writing the KEY value.

Special care must be taken when swapping the flash bank address space. See the nonvolatile memory system chapter for additional considerations when using flash bank address swapping.

### *2.4.7 Shutdown Mode Handling (if present)*

When the device is configured to enter SHUTDOWN mode, the core regulator is powered down and the device register contents and SRAM contents are lost. An exit from SHUTDOWN mode generates a BOR level reset. Two mechanisms are provided to preserve the device state when entering SHUTDOWN mode: IO latching and a small shutdown memory.

#### Shutdown IO State

The digital IO pin states (output low/high, pullup/pulldown, Hi-Z, drive configuration) are latched and retained upon entry to SHUTDOWN. After exiting SHUTDOWN mode, the IOs are held in the previous state until released by application software setting the RELEASE bit in the SHDNIOREL register along with the matching KEY value. When exiting SHUTDOWN mode, application software must first re-configure the IO to their proper state, then release the IO. To determine at startup if the cause of a reset was an exit from SHUTDOWN mode, application software must read the RSTCAUSE register in SYSCTL.

**Note**

When exiting SHUTDOWN, **the IOs specific to serial wire debug (SWD) pins are released from their latched state regardless of the RELEASE bits being set or not**. This is to ensure that the user can enable a debugger connection.

**Note**

When exiting SHUTDOWN, the bootstrap loader (BSL) invoke pin must be held at a logic low level to prevent unintended entry into the BSL during exit from SHUTDOWN. An entry to the BSL during SHUTDOWN exit prevents the application code from starting, the BSL interfaces are not be available, and a SWD connection is possible as the IO states remain latched through SHUTDOWN exit until application software releases the IOs.

#### Shutdown Memory

To enable saving of application state information before entering SHUTDOWN mode, 4 bytes of shutdown memory are provided in SYSCTL. These memory locations are retained in SHUTDOWN mode and are readable by the application after exiting SHUTDOWN. To save data to the SHUTDOWN memory, write to the SHUTDNSTORE0-SHUTDNSTORE3 registers in SYSCTL.

### *2.4.8 Configuration Lockout*

Configuration registers in SYSCTL can be locked out from writes to add a layer of robustness against unintended changes to the PMCU at runtime. To lock out the configuration registers from writes, set the ACTIVE bit in the WRITELOCK register in SYSCTL.

All SYSCTL registers are protected by the WRITELOCK functionality except for those listed below:

- WRITELOCK



Copyright © 2026 Texas Instruments Incorporated

- PMODECFG
- FCC, FCCCMD
- FLBANKSWAP (if present)
- RSTCAUSE (read-to-clear), RESETLEVEL, RESETCMD
- BORTHRESHOLD, BORCLRCMD
- SHDNIOREL (if present)
- PMUOPAMP
- SHUTDNSTOREx (if present)

In addition to the overall SYSCTL configuration write lock feature, many SYSCTL registers also require a KEY value to be written in conjunction with the desire configuration data for the write to take effect.

### *2.4.9 System Status*

The status of various aspects of the PMCU can be polled by software by reading the CLKSTATUS and SYSSTATUS registers in SYSCTL.

#### Checking Clock Status (CLKSTATUS)

The CLKSTATUS register in SYSCTL is a read-only register which indicates the current configuration and status of the clock module. Key status information provided in CLKSTATUS includes:

- The current SYSOSC frequency
- The current HSCLK selection
- The current LFCLK selection
- The current MCLK selection
- The HFCLK and SYSPLL (if present) status
- The LFXT status
- The LFOSC status
- Indications that the HSCLK, HFCLK, and SYSPLL (if present) are disabled
- Error indications if a peripheral requested a clock and the clock cannot be generated

This status information is useful to validate that a requested clock change has completed successfully, or to check the true SYSOSC frequency in applications where SYSOSC can have asynchronous activation or frequency requests issued by peripherals.

#### Checking System Status (SYSSTATUS)

The SYSSTATUS register in SYSCTL is a read-only register which indicates flash ECC errors (SED and DED) along with other peripheral-specific status information. ECC error bits in SYSTATUS are sticky (they remain set when an ECC error occurs even if future reads do not have errors). These bits can be reset (cleared) by setting the ALLECC bit in the SYSSTATUSCLR register along with the KEY value.

The SYSSTATUS register in SYSCTL is a read-only register which indicates the peripheral-specific status information.

### *2.4.10 Error Handling*

MSPM0 devices include several diagnostic mechanism to detect errors at runtime. Table 2-15 lists error sources and their corresponding handling mechanism.

**Note**

Not all MSPM0 devices support all diagnostic features. For example, some devices do not have ECC/parity on memories and some devices do not have dual watchdog timers. Always refer to the device-specific data sheet to understand which diagnostic features are available for a given device. In the PMCU registers section, register maps are also provided for each MCU subfamily detailing the specific registers available for a given device.



Copyright © 2026 Texas Instruments Incorporated

**Table 2-15. Error Sources and Handling Mechanisms**

| Error Source | Error | Handling Mechanism |
|---|---|---|
| Flash (if device has ECC) | Non-correctable ECC error (if device has ECC) | • For a CPU or DMA request, a FLASHDED nonmaskable interrupt is generated to the processor or a SYSRST is generated depending on configuration of the FLASHECCRSTDIS bit<br>• The FLASHDED sticky bit is set in the SYSSTATUS register in SYSCTL |
| | Correctable ECC error (if device has ECC) | • A FLASHSEC interrupt is also generated in SYSCTL<br>• The FLASHSEC sticky bit is set in the SYSSTATUS register in SYSCTL |
| SRAM | Non-correctable ECC error (if device has ECC) | • An SRAMDED nonmaskable interrupt is generated to the processor |
| | Correctable ECC error (if device has ECC) | • A SYSCTL SRAMSED interrupt is generated to the processor |
| | Parity error (if device has parity) | • Nonmaskable interrupt is generated to the processor if the request was from the CPU<br>• DMA data error interrupt is generated if the request was from the DMA |
| | Address error on CPU access | • A hard fault is generated in the CPU |
| | Address error on DMA access | • A DMA address error interrupt is generated in the DMA controller |
| | ECC error on CAN SRAM (if device has CAN-FD) | • An interrupt is generated in the CAN-FD peripheral |
| SHUTDNSTOREx Memory (if present) | Parity error | • A POR is generated |
| CKM | MCLK failure | • A BOOTRST is generated |
| | LFCLK failure (if present) | • A BOOTRST is generated if LFCLK is sourcing MCLK<br>• An LFCLKFAIL nonmaskable interrupt is generated in the SYSCTL NMI registers. |
| CPUSS (if device has MPU) | Memory protection unit violation | • A hard fault is generated in the CPU |
| WWDT | WWDT0 violation | • A BOOTRST is generated or a nonmaskable interrupt is generated in the SYSCTL NMI registers depending on configuration of the WWDTLP0RSTDIS bit |
| | WWDT1 violation (if present) | • A BOOTRST is generated or a nonmaskable interrupt is generated in the SYSCTL NMI registers depending on configuration of the WWDTLP1RSTDIS bit |
| PMU | Trim parity error | • A POR is generated |
| | POR0- supply error | • A POR is generated |
| | BOR0- supply error | • A BOR is generated |
| | BOR1/2/3- supply error (if present) | • A BORLVL nonmaskable interrupt is generated in the SYSCTL NMI registers |
| CPUSS | Memory protection unit violation (if present) | • A hard fault is generated in the CPU |



Copyright © 2026 Texas Instruments Incorporated

### Configurable NMI Triggers

Error sources can be configured to trigger either a nonmaskable interrupt or a different handling mechanism. The SYSTEMCFG register in SYSCTL may be used to specify the desired error handling mechanism. For example, the WWDT0 may be configured to generate either a BOOTRST or an NMI, with BOOTRST being the default case. Refer to the SYSTEMCFG register for the relevant device subfamily for the available error handling options.

## *2.4.11 SYSCTL Events*

The SYSCTL module contains two event publishers and no event subscribers. One event publisher manages SYSCTL interrupt requests (IRQs) to the CPU subsystem. The second publisher manages nonmaskable interrupts to the CPU subsystem for critical diagnostics.

The SYSCTL events are summarized in SYSCTL Events.

**Table 2-16. SYSCTL Events**

| Event | Type | Source | Destination | Route | Configuration | Functionality |
|---|---|---|---|---|---|---|
| CPU interrupt | Publisher | SYSCTL | CPU Subsystem | Static route | SYSCTL interrupt registers | Fixed interrupt route from SYSCTL to CPU |
| CPU nonmaskable interrupt (NMI) | Publisher | SYSCTL | CPU Subsystem | Static route | NMI interrupt registers | Fixed interrupt route from SYSCTL to CPU |

### 2.4.11.1 CPU Interrupt Event (CPU_INT)

The SYSCTL module provides several interrupt sources which can be configured to source a CPU interrupt event. In order of decreasing interrupt priority, the CPU interrupt events from the SYSCTL are given in Table 2-17.

**Table 2-17. SYSCTL CPU Interrupt Event Sources**

| Index (IIDX) | Name | Description |
|---|---|---|
| 0 | NONE | No interrupt pending. |
| 1 | LFOSCGOOD | Indicates when LFOSC is ready during startup, as LFOSC takes ≈1ms to start. |
| 2 | ANACLKERR | Indicates that an analog function was enabled and expecting a SYSOSC to be operation at a certain frequency, but SYSOSC was either not available or not operating at the required frequency. |
| 3 | FLASHSEC | Indicates that a flash memory single-bit correctable error was detected |
| 4 | SRAMSEC | Indicates that a SRAM single-bit correctable error was detected |
| 5 | LFXTGOOD | Indicates when the low frequency external clock (either the LFXT oscillator or LFCLK_IN digital clock) are ready. This indication is useful when starting the clock system and waiting for LFXT or LFCK_IN to be ready before switching the LFCLK source from LFOSC to an external source. |
| 6 | HFCLKGOOD | Indicates when the high frequency external clock (either the HFXT oscillator or HFCLK_IN digital clock) are ready. This indication is useful when starting the clock system and waiting for HFCLK to be ready before switching the MCLK source to HFCLK or before starting the SYSPLL with HFCLK as the SYSPLL reference. |
| 7 | SYSPLLGOOD | Indicates when the SYSPLL is ready and available for use. This indication is useful when starting the clock system and waiting for SYSPLL to be ready before switching the MCLK source to SYSPLL. |



Copyright © 2026 Texas Instruments Incorporated

**Table 2-17. SYSCTL CPU Interrupt Event Sources (continued)**

| Index (IIDX) | Name | Description |
|---|---|---|
| 8 | HSCLKGOOD | Indicates when the HSCLK (sourced by either HFCLK or a SYSPLL output) is ready. This indication is useful when waking up from STOP or STANDBY mode when HSCLK is configured as the MCLK source in RUN/SLEEP mode. When waking from STOP or STANDBY, MCLK will run from SYSOSC until the HSCLK is available, at which time SYSCTL automatically switches the MCLK source to the selected HSCLK source. This interrupt communicates that after wake-up from STOP/STANDBY, MCLK has switched over to the desired HSCLK source and it is now possible to use timing-sensitive peripherals sourced by MCLK. |

The CPU interrupt event configuration is managed with the SYSCTL IIDX, IMASK, RIS, MIS, ISET, and ICLR event management registers. See Section 7.2.5 for guidance on configuring these registers for CPU interrupts.

#### 2.4.11.2 Nonmaskable Interrupt Event (NMI)

The SYSCTL module provides x interrupt sources which can be configured to source a CPU interrupt event. In order of decreasing interrupt priority, the CPU interrupt events from the SYSCTL are given in Table 2-17.

**Table 2-18. Nonmaskable Interrupt Event Table**

| Index (IIDX) | Name | Description |
|---|---|---|
| 0 | NONE | No NMI pending. |
| 1 | BORLVL | Indicates that VDD has dropped below the specified VBOR- threshold. |
| 2 | WWDT0 | A WWDT0 violation occurred. |
| 3 | WWDT1 | A WWDT1 violation occurred. |
| 4 | LFCLKFAIL | Indicates that the LFXT or LFCLK_IN clock source is dead. This indication is useful for handling LFCLK errors when LFCLK is not sourcing MCLK but is sourcing a peripheral (for example, the RTC) |
| 5 | FLASHDED | Indicates that a flash memory double-bit uncorrectable error was detected. |
| 6 | SRAMDED | Indicates that an SRAM double-bit uncorrectable error or parity error was detected. |

The CPU nonmaskable interrupt event configuration is managed with the NMIIIDX, NMIRIS, NMIISET, and NMIICLR event management registers. See Section 7.2.5 for guidance on configuring the interrupt management registers.



Copyright © 2026 Texas Instruments Incorporated

## 2.5 SYSCTL_G351x_G151x_G321x_G320x_G121x_G120x_G122x_G322x Registers

Table 2-19 lists the memory-mapped registers for the SYSCTL_G351x_G151x_G321x_G320x_G121x_G120x_G122x_G322x registers. All register offset addresses not listed in Table 2-19 should be considered as reserved locations and the register contents should not be modified.

**Table 2-19. SYSCTL_G351X_G151X_G321X_G320X_G121X_G120X_G122X_G322X Registers**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 1020h | IIDX | SYSCTL interrupt index | | Go |
| 1028h | IMASK | SYSCTL interrupt mask | | Go |
| 1030h | RIS | SYSCTL raw interrupt status | | Go |
| 1038h | MIS | SYSCTL masked interrupt status | | Go |
| 1040h | ISET | SYSCTL interrupt set | | Go |
| 1048h | ICLR | SYSCTL interrupt clear | | Go |
| 1050h | NMIIIDX | NMI interrupt index | | Go |
| 1060h | NMIRIS | NMI raw interrupt status | | Go |
| 1070h | NMIISET | NMI interrupt set | | Go |
| 1078h | NMIICLR | NMI interrupt clear | | Go |
| 1100h | SYSOSCCFG | SYSOSC configuration | | Go |
| 1104h | MCLKCFG | Main clock (MCLK) configuration | | Go |
| 1108h | HSCLKEN | High-speed clock (HSCLK) source enable/disable | | Go |
| 110Ch | HSCLKCFG | High-speed clock (HSCLK) source selection | | Go |
| 1110h | HFCLKCLKCFG | High-frequency clock (HFCLK) configuration | | Go |
| 1114h | LFCLKCFG | Low frequency crystal oscillator (LFXT) configuration | | Go |
| 1120h | SYSPLLCFG0 | SYSPLL reference and output configuration | | Go |
| 1124h | SYSPLLCFG1 | SYSPLL reference and feedback divider | | Go |
| 1128h | SYSPLLPARAM0 | SYSPLL PARAM0 (load from FACTORY region) | | Go |
| 112Ch | SYSPLLPARAM1 | SYSPLL PARAM1 (load from FACTORY region) | | Go |
| 1138h | GENCLKCFG | General clock configuration | | Go |
| 113Ch | GENCLKEN | General clock enable control | | Go |
| 1140h | PMODECFG | Power mode configuration | | Go |
| 1150h | FCC | Frequency clock counter (FCC) count | | Go |
| 1170h | SYSOSCTRIMUSER | SYSOSC user-specified trim | | Go |
| 1178h | SRAMBOUNDARY | SRAM Write Boundary | | Go |
| 117Ch | SRAMBOUNDARYHIGH | SRAM Write Boundary High | | Go |
| 1180h | SYSTEMCFG | System configuration | | Go |
| 1184h | SRAMCFG | System SRAM configuration | | Go |
| 1200h | WRITELOCK | SYSCTL register write lockout | | Go |
| 1204h | CLKSTATUS | Clock module (CKM) status | | Go |
| 1208h | SYSSTATUS | System status information | | Go |
| 120Ch | DEDERRADDR | Memory DED Address | | Go |
| 1220h | RSTCAUSE | Reset cause | | Go |



Copyright © 2026 Texas Instruments Incorporated

**Table 2-19. SYSCTL_G351X_G151X_G321X_G320X_G121X_G120X_G122X_G322X Registers (continued)**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 1300h | RESETLEVEL | Reset level for application-triggered reset command | | Go |
| 1304h | RESETCMD | Execute an application-triggered reset command | | Go |
| 1308h | BORTHRESHOLD | BOR threshold selection | | Go |
| 130Ch | BORCLRCMD | Set the BOR threshold | | Go |
| 1310h | SYSOSCFCLCTL | SYSOSC frequency correction loop (FCL) ROSC enable | | Go |
| 1314h | LFXTCTL | LFXT and LFCLK control | | Go |
| 1318h | EXLFCTL | LFCLK_IN and LFCLK control | | Go |
| 131Ch | SHDNIOREL | SHUTDOWN IO release control | | Go |
| 1320h | EXRSTPIN | Disable the reset function of the NRST pin | | Go |
| 1324h | SYSSTATUSCLR | Clear sticky bits of SYSSTATUS | | Go |
| 1328h | SWDCFG | Disable the SWD function on the SWD pins | | Go |
| 132Ch | FCCCMD | Frequency clock counter start capture | | Go |
| 1400h | SHUTDNSTORE0 | Shutdown storage memory (byte 0) | | Go |
| 1404h | SHUTDNSTORE1 | Shutdown storage memory (byte 1) | | Go |
| 1408h | SHUTDNSTORE2 | Shutdown storage memory (byte 2) | | Go |
| 140Ch | SHUTDNSTORE3 | Shutdown storage memory (byte 3) | | Go |
| 2040h | IDREG | ID Register | | Go |

Complex bit access types are encoded to fit into small table cells. Table 2-20 shows the codes that are used for access types in this section.

**Table 2-20. SYSCTL_G351x_G151x_G321x_G320x_G121x_G120x_G122x_G322x Access Type Codes**

| Access Type | Code | Description |
|---|---|---|
| Read Type | | |
| R | R | Read |
| RC | R<br>C | Read<br>to Clear |
| Write Type | | |
| W | W | Write |
| W1C | W<br>1C | Write<br>1 to clear |
| W1S | W<br>1S | Write<br>1 to set |
| Reset or Default Value | | |
| *-n* | | Value after reset or the default value |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.1 IIDX (Offset = 1020h) [Reset = 00000000h]

IIDX is shown in Figure 2-11 and described in Table 2-21.

Return to the Summary Table.

SYSCTL interrupt index

**Figure 2-11. IIDX**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | STAT | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | |

**Table 2-21. IIDX Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3-0 | STAT | R | 0h | The SYSCTL interrupt index (IIDX) register generates a value corresponding to the highest priority pending interrupt source. This value may be used as an address offset for fast, deterministic handling in the interrupt service routine. A read of the IIDX register will clear the corresponding interrupt status in the RIS and MIS registers.<br>0h = No interrupt pending<br>1h = LFOSCGOOD interrupt pending<br>2h = 2<br>3h = 3<br>4h = 4<br>5h = 5<br>6h = 6<br>7h = 7<br>8h = 8 |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.2 IMASK (Offset = 1028h) [Reset = 00000000h]

IMASK is shown in Figure 2-12 and described in Table 2-22.

Return to the Summary Table.

SYSCTL interrupt mask

**Figure 2-12. IMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| HSCLKGOOD | SYSPLLGOOD | HFCLKGOOD | LFXTGOOD | SRAMSEC | FLASHSEC | ANACLKERR | LFOSCGOOD |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

**Table 2-22. IMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7 | HSCLKGOOD | R/W | 0h | HSCLK GOOD<br>0h = 0<br>1h = 1 |
| 6 | SYSPLLGOOD | R/W | 0h | SYSPLL GOOD<br>0h = 0<br>1h = 1 |
| 5 | HFCLKGOOD | R/W | 0h | HFCLK GOOD<br>0h = 0<br>1h = 1 |
| 4 | LFXTGOOD | R/W | 0h | LFXT GOOD<br>0h = 0<br>1h = 1 |
| 3 | SRAMSEC | R/W | 0h | SRAM Single Error Correct<br>0h = 0<br>1h = 1 |
| 2 | FLASHSEC | R/W | 0h | Flash Single Error Correct<br>0h = 0<br>1h = 1 |
| 1 | ANACLKERR | R/W | 0h | Analog Clocking Consistency Error<br>0h = 0<br>1h = 1 |
| 0 | LFOSCGOOD | R/W | 0h | Enable or disable the LFOSCGOOD interrupt. LFOSCGOOD indicates that the LFOSC has started successfully.<br>0h = Interrupt disabled<br>1h = Interrupt enabled |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.3 RIS (Offset = 1030h) [Reset = 00000000h]

RIS is shown in Figure 2-13 and described in Table 2-23.

Return to the Summary Table.

SYSCTL raw interrupt status

**Figure 2-13. RIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| HSCLKGOOD | SYSPLLGOOD | HFCLKGOOD | LFXTGOOD | SRAMSEC | FLASHSEC | ANACLKERR | LFOSCGOOD |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 2-23. RIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7 | HSCLKGOOD | R | 0h | HSCLK GOOD<br>0h = 0<br>1h = 1 |
| 6 | SYSPLLGOOD | R | 0h | SYSPLL GOOD<br>0h = 0<br>1h = 1 |
| 5 | HFCLKGOOD | R | 0h | HFCLK GOOD<br>0h = 0<br>1h = 1 |
| 4 | LFXTGOOD | R | 0h | LFXT GOOD<br>0h = 0<br>1h = 1 |
| 3 | SRAMSEC | R | 0h | SRAM Single Error Correct<br>0h = 0<br>1h = 1 |
| 2 | FLASHSEC | R | 0h | Flash Single Error Correct<br>0h = 0<br>1h = 1 |
| 1 | ANACLKERR | R | 0h | Analog Clocking Consistency Error<br>0h = 0<br>1h = 1 |
| 0 | LFOSCGOOD | R | 0h | Raw status of the LFOSCGOOD interrupt.<br>0h = No interrupt pending<br>1h = Interrupt pending |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.4 MIS (Offset = 1038h) [Reset = 00000000h]

MIS is shown in Figure 2-14 and described in Table 2-24.

Return to the Summary Table.

SYSCTL masked interrupt status

**Figure 2-14. MIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| HSCLKGOOD | SYSPLLGOOD | HFCLKGOOD | LFXTGOOD | SRAMSEC | FLASHSEC | ANACLKERR | LFOSCGOOD |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 2-24. MIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7 | HSCLKGOOD | R | 0h | HSCLK GOOD<br>0h = 0<br>1h = 1 |
| 6 | SYSPLLGOOD | R | 0h | SYSPLL GOOD<br>0h = 0<br>1h = 1 |
| 5 | HFCLKGOOD | R | 0h | HFCLK GOOD<br>0h = 0<br>1h = 1 |
| 4 | LFXTGOOD | R | 0h | LFXT GOOD<br>0h = 0<br>1h = 1 |
| 3 | SRAMSEC | R | 0h | SRAM Single Error Correct<br>0h = 0<br>1h = 1 |
| 2 | FLASHSEC | R | 0h | Flash Single Error Correct<br>0h = 0<br>1h = 1 |
| 1 | ANACLKERR | R | 0h | Analog Clocking Consistency Error<br>0h = 0<br>1h = 1 |
| 0 | LFOSCGOOD | R | 0h | Masked status of the LFOSCGOOD interrupt.<br>0h = No interrupt pending<br>1h = Interrupt pending |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.5 ISET (Offset = 1040h) [Reset = 00000000h]

ISET is shown in Figure 2-15 and described in Table 2-25.

Return to the Summary Table.

SYSCTL interrupt set

**Figure 2-15. ISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| HSCLKGOOD | SYSPLLGOOD | HFCLKGOOD | LFXTGOOD | SRAMSEC | FLASHSEC | ANACLKERR | LFOSCGOOD |
| W1S-0h | W1S-0h | W1S-0h | W1S-0h | W1S-0h | W1S-0h | W1S-0h | W1S-0h |

**Table 2-25. ISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7 | HSCLKGOOD | W1S | 0h | HSCLK GOOD<br>0h = 0<br>1h = 1 |
| 6 | SYSPLLGOOD | W1S | 0h | SYSPLL GOOD<br>0h = 0<br>1h = 1 |
| 5 | HFCLKGOOD | W1S | 0h | HFCLK GOOD<br>0h = 0<br>1h = 1 |
| 4 | LFXTGOOD | W1S | 0h | LFXT GOOD<br>0h = 0<br>1h = 1 |
| 3 | SRAMSEC | W1S | 0h | SRAM Single Error Correct<br>0h = 0<br>1h = 1 |
| 2 | FLASHSEC | W1S | 0h | Flash Single Error Correct<br>0h = 0<br>1h = 1 |
| 1 | ANACLKERR | W1S | 0h | Analog Clocking Consistency Error<br>0h = 0<br>1h = 1 |
| 0 | LFOSCGOOD | W1S | 0h | Set the LFOSCGOOD interrupt.<br>0h = Writing 0h hs no effect<br>1h = Set interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.6 ICLR (Offset = 1048h) [Reset = 00000000h]

ICLR is shown in Figure 2-16 and described in Table 2-26.

Return to the Summary Table.

SYSCTL interrupt clear

**Figure 2-16. ICLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| HSCLKGOOD | SYSPLLGOOD | HFCLKGOOD | LFXTGOOD | SRAMSEC | FLASHSEC | ANACLKERR | LFOSCGOOD |
| W1C-0h | W1C-0h | W1C-0h | W1C-0h | W1C-0h | W1C-0h | W1C-0h | W1C-0h |

**Table 2-26. ICLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7 | HSCLKGOOD | W1C | 0h | HSCLK GOOD<br>0h = 0<br>1h = 1 |
| 6 | SYSPLLGOOD | W1C | 0h | SYSPLL GOOD<br>0h = 0<br>1h = 1 |
| 5 | HFCLKGOOD | W1C | 0h | HFCLK GOOD<br>0h = 0<br>1h = 1 |
| 4 | LFXTGOOD | W1C | 0h | LFXT GOOD<br>0h = 0<br>1h = 1 |
| 3 | SRAMSEC | W1C | 0h | SRAM Single Error Correct<br>0h = 0<br>1h = 1 |
| 2 | FLASHSEC | W1C | 0h | Flash Single Error Correct<br>0h = 0<br>1h = 1 |
| 1 | ANACLKERR | W1C | 0h | Analog Clocking Consistency Error<br>0h = 0<br>1h = 1 |
| 0 | LFOSCGOOD | W1C | 0h | Clear the LFOSCGOOD interrupt.<br>0h = Writing 0h has no effect<br>1h = Clear interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.7 NMIIIDX (Offset = 1050h) [Reset = 00000000h]

NMIIIDX is shown in Figure 2-17 and described in Table 2-27.

Return to the Summary Table.

NMI interrupt index

**Figure 2-17. NMIIIDX**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | STAT | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | |

**Table 2-27. NMIIIDX Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3-0 | STAT | R | 0h | The NMI interrupt index (NMIIIDX) register generates a value corresponding to the highest priority pending NMI source. This value may be used as an address offset for fast, deterministic handling in the NMI service routine. A read of the NMIIIDX register will clear the corresponding interrupt status in the NMIRIS register.<br>0h = No NMI pending<br>1h = BOR Threshold NMI pending<br>2h = 2<br>3h = 3<br>4h = 4<br>5h = 5<br>6h = 6 |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.8 NMIRIS (Offset = 1060h) [Reset = 00000000h]

NMIRIS is shown in Figure 2-18 and described in Table 2-28.

Return to the Summary Table.

NMI raw interrupt status

**Figure 2-18. NMIRIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | SRAMDED | FLASHDED | LFCLKFAIL | WWDT1 | WWDT0 | BORLVL |
| R-0h | | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 2-28. NMIRIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | RESERVED | R | 0h | |
| 5 | SRAMDED | R | 0h | SRAM Double Error Detect<br>0h = 0<br>1h = 1 |
| 4 | FLASHDED | R | 0h | Flash Double Error Detect<br>0h = 0<br>1h = 1 |
| 3 | LFCLKFAIL | R | 0h | LFXT-EXLF Monitor Fail<br>0h = 0<br>1h = 1 |
| 2 | WWDT1 | R | 0h | Watch Dog 0 Fault<br>0h = 0<br>1h = 1 |
| 1 | WWDT0 | R | 0h | Watch Dog 0 Fault<br>0h = 0<br>1h = 1 |
| 0 | BORLVL | R | 0h | Raw status of the BORLVL NMI<br>0h = No interrupt pending<br>1h = Interrupt pending |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.9 NMIISET (Offset = 1070h) [Reset = 00000000h]

NMIISET is shown in Figure 2-19 and described in Table 2-29.

Return to the Summary Table.

NMI interrupt set

**Figure 2-19. NMIISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | SRAMDED | FLASHDED | LFCLKFAIL | WWDT1 | WWDT0 | BORLVL |
| R-0h | | W1S-0h | W1S-0h | W1S-0h | W1S-0h | W1S-0h | W1S-0h |

**Table 2-29. NMIISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | RESERVED | R | 0h | |
| 5 | SRAMDED | W1S | 0h | SRAM Double Error Detect<br>0h = 0<br>1h = 1 |
| 4 | FLASHDED | W1S | 0h | Flash Double Error Detect<br>0h = 0<br>1h = 1 |
| 3 | LFCLKFAIL | W1S | 0h | LFXT-EXLF Monitor Fail<br>0h = 0<br>1h = 1 |
| 2 | WWDT1 | W1S | 0h | Watch Dog 0 Fault<br>0h = 0<br>1h = 1 |
| 1 | WWDT0 | W1S | 0h | Watch Dog 0 Fault<br>0h = 0<br>1h = 1 |
| 0 | BORLVL | W1S | 0h | Set the BORLVL NMI<br>0h = Writing 0h hs no effect<br>1h = Set interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.10 NMIICLR (Offset = 1078h) [Reset = 00000000h]

NMIICLR is shown in Figure 2-20 and described in Table 2-30.

Return to the Summary Table.

NMI interrupt clear

**Figure 2-20. NMIICLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | SRAMDED | FLASHDED | LFCLKFAIL | WWDT1 | WWDT0 | BORLVL |
| R-0h | | W1C-0h | W1C-0h | W1C-0h | W1C-0h | W1C-0h | W1C-0h |

**Table 2-30. NMIICLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | RESERVED | R | 0h | |
| 5 | SRAMDED | W1C | 0h | SRAM Double Error Detect<br>0h = 0<br>1h = 1 |
| 4 | FLASHDED | W1C | 0h | Flash Double Error Detect<br>0h = 0<br>1h = 1 |
| 3 | LFCLKFAIL | W1C | 0h | LFXT-EXLF Monitor Fail<br>0h = 0<br>1h = 1 |
| 2 | WWDT1 | W1C | 0h | Watch Dog 0 Fault<br>0h = 0<br>1h = 1 |
| 1 | WWDT0 | W1C | 0h | Watch Dog 0 Fault<br>0h = 0<br>1h = 1 |
| 0 | BORLVL | W1C | 0h | Clr the BORLVL NMI<br>0h = Writing 0h hs no effect<br>1h = Clear interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.11 SYSOSCCFG (Offset = 1100h) [Reset = 0002XXXXh]

SYSOSCCFG is shown in Figure 2-21 and described in Table 2-31.

Return to the Summary Table.

SYSOSC configuration

**Figure 2-21. SYSOSCCFG**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | FASTCPUEVENT | BLOCKASYNCALL |
| R-0h | | | | | | R/W-1h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | DISABLE | DISABLESTOP | USE4MHZSTOP |
| R-0h | | | | | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | FREQ | |
| R-0h | | | | | | R/W-0h | |

**Table 2-31. SYSOSCCFG Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-18 | RESERVED | R | 0h | |
| 17 | FASTCPUEVENT | R/W | 1h | FASTCPUEVENT may be used to assert a fast clock request when an interrupt is asserted to the CPU, reducing interrupt latency.<br>0h = An interrupt to the CPU will not assert a fast clock request<br>1h = An interrupt to the CPU will assert a fast clock request |
| 16 | BLOCKASYNCALL | R/W | 0h | BLOCKASYNCALL may be used to mask block all asynchronous fast clock requests, preventing hardware from dynamically changing the active clock configuration when operating in a given mode.<br>0h = Asynchronous fast clock requests are controlled by the requesting peripheral<br>1h = All asynchronous fast clock requests are blocked |
| 15-11 | RESERVED | R | 0h | |
| 10 | DISABLE | R/W | 0h | DISABLE sets the SYSOSC enable/disable policy. SYSOSC may be powered off in RUN, SLEEP, and STOP modes to reduce power consumption. When SYSOSC is disabled, MCLK and ULPCLK are sourced from LFCLK.<br>0h = Do not disable SYSOSC<br>1h = Disable SYSOSC immediately and source MCLK and ULPCLK from LFCLK |
| 9 | DISABLESTOP | R/W | 0h | DISABLESTOP sets the SYSOSC stop mode enable/disable policy. When operating in STOP mode, the SYSOSC may be automatically disabled. When set, ULPCLK will run from LFCLK in STOP mode and SYSOSC will be disabled to reduce power consumption.<br>0h = Do not disable SYSOSC in STOP mode<br>1h = Disable SYSOSC in STOP mode and source ULPCLK from LFCLK |
| 8 | USE4MHZSTOP | R/W | 0h | USE4MHZSTOP sets the SYSOSC stop mode frequency policy. When entering STOP mode, the SYSOSC frequency may be automatically switched to 4MHz to reduce SYSOSC power consumption.<br>0h = Do not gear shift the SYSOSC to 4MHz in STOP mode<br>1h = Gear shift SYSOSC to 4MHz in STOP mode |



Copyright © 2026 Texas Instruments Incorporated

**Table 2-31. SYSOSCCFG Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7-2 | RESERVED | R | 0h | |
| 1-0 | FREQ | R/W | 0h | Target operating frequency for the system oscillator (SYSOSC)<br>0h = Base frequency (32MHz)<br>1h = Low frequency (4MHz)<br>2h = User-trimmed frequency (16 or 24 MHz) |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.12 MCLKCFG (Offset = 1104h) [Reset = 000XX2X0h]

MCLKCFG is shown in Figure 2-22 and described in Table 2-32.

Return to the Summary Table.

Main clock (MCLK) configuration

**Figure 2-22. MCLKCFG**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | MCLKDEADCHK | STOPCLKSTBY | USELFCLK | RESERVED | | | USEHSCLK |
| R-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | | | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | USEMFTICK | FLASHWAIT | | | |
| R-0h | | | R/W-0h | R/W-2h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | UDIV | | MDIV | | | |
| R-0h | | R/W-1h | | R/W-0h | | | |

**Table 2-32. MCLKCFG Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-23 | RESERVED | R | 0h | |
| 22 | MCLKDEADCHK | R/W | 0h | MCLKDEADCHK enables or disables the continuous MCLK dead check monitor. LFCLK must be running before MCLKDEADCHK is enabled.<br>0h = The MCLK dead check monitor is disabled<br>1h = The MCLK dead check monitor is enabled |
| 21 | STOPCLKSTBY | R/W | 0h | STOPCLKSTBY sets the STANDBY mode policy (STANDBY0 or STANDBY1). When set, ULPCLK and LFCLK are disabled to all peripherals in STANDBY mode, with the exception of TIMG0 and TIMG1 which continue to run. Wake-up is only possible via an asynchronous fast clock request.<br>0h = ULPCLK/LFCLK runs to all PD0 peripherals in STANDBY mode<br>1h = ULPCLK/LFCLK is disabled to all peripherals in STANDBY mode except TIMG0 and TIMG1 |
| 20 | USELFCLK | R/W | 0h | USELFCLK sets the MCLK source policy. Set USELFCLK to use LFCLK as the MCLK source. Note that setting USELFCLK does not disable SYSOSC, and SYSOSC remains available for direct use by analog modules.<br>0h = MCLK will not use the low frequency clock (LFCLK)<br>1h = MCLK will use the low frequency clock (LFCLK) |
| 19-17 | RESERVED | R | 0h | |
| 16 | USEHSCLK | R/W | 0h | USEHSCLK, together with USELFCLK, sets the MCLK source policy. Set USEHSCLK to use HSCLK (HFCLK or SYSPLL) as the MCLK source in RUN and SLEEP modes.<br>0h = MCLK will not use the high speed clock (HSCLK)<br>1h = MCLK will use the high speed clock (HSCLK) in RUN and SLEEP mode |
| 15-13 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

**Table 2-32. MCLKCFG Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 12 | USEMFTICK | R/W | 0h | USEMFTICK specifies whether the 4MHz constant-rate clock (MFCLK) to peripherals is enabled or disabled. When enabled, MDIV must be disabled (set to 0h=/1).<br>0h = The 4MHz rate MFCLK to peripherals is disabled.<br>1h = The 4MHz rate MFCLK to peripherals is enabled. |
| 11-8 | FLASHWAIT | R/W | 2h | FLASHWAIT specifies the number of flash wait states when MCLK is sourced from HSCLK. FLASHWAIT has no effect when MCLK is sourced from SYSOSC or LFCLK.<br>0h = No flash wait states are applied<br>1h = One flash wait state is applied<br>2h = 2 flash wait states are applied |
| 7-6 | RESERVED | R | 0h | |
| 5-4 | UDIV | R/W | 1h | UDIV specifies the ULPCLK divider when MCLK is sourced from HSCLK. UDIV has no effect when MCLK is sourced from SYSOSC or LFCLK.<br>0h = ULPCLK is not divided and is equal to MCLK<br>1h = ULPCLK is MCLK/2 (divided-by-2) |
| 3-0 | MDIV | R/W | 0h | MDIV may be used to divide the MCLK frequency when MCLK is sourced from SYSOSC. MDIV=0h corresponds to /1 (no divider). MDIV=1h corresponds to /2 (divide-by-2). MDIV=Fh corresponds to /16 (divide-by-16). MDIV may be set between /1 and /16 on an integer basis. |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.13 HSCLKEN (Offset = 1108h) [Reset = 0000XXXXh]

HSCLKEN is shown in Figure 2-23 and described in Table 2-33.

Return to the Summary Table.

High-speed clock (HSCLK) source enable/disable

**Figure 2-23. HSCLKEN**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | USEEXTHFCLK |
| R-0h | | | | | | | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | SYSPLLEN |
| R-0h | | | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | HFXTEN |
| R-0h | | | | | | | R/W-0h |

**Table 2-33. HSCLKEN Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-17 | RESERVED | R | 0h | |
| 16 | USEEXTHFCLK | R/W | 0h | USEEXTHFCLK selects the HFCLK_IN digital clock input to be the source for HFCLK. When disabled, HFXT is the HFCLK source and HFXTEN may be set. Do not set HFXTEN and USEEXTHFCLK simultaneously.<br>0h = Use HFXT as the HFCLK source<br>1h = Use the HFCLK_IN digital clock input as the HFCLK source |
| 15-9 | RESERVED | R | 0h | |
| 8 | SYSPLLEN | R/W | 0h | SYSPLLEN enables or disables the system phase-lock loop (SYSPLL).<br>0h = Disable the SYSPLL<br>1h = Enable the SYSPLL |
| 7-1 | RESERVED | R | 0h | |
| 0 | HFXTEN | R/W | 0h | HFXTEN enables or disables the high frequency crystal oscillator (HFXT).<br>0h = Disable the HFXT<br>1h = Enable the HFXT |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.14 HSCLKCFG (Offset = 110Ch) [Reset = 00000000h]

HSCLKCFG is shown in Figure 2-24 and described in Table 2-34.

Return to the Summary Table.

High-speed clock (HSCLK) source selection

**Figure 2-24. HSCLKCFG**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | HSCLKSEL |
| R-0h | | | | | | | R/W-0h |

**Table 2-34. HSCLKCFG Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | HSCLKSEL | R/W | 0h | HSCLKSEL selects the HSCLK source (SYSPLL or HFCLK).<br>0h = HSCLK is sourced from the SYSPLL<br>1h = HSCLK is sourced from the HFCLK |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.15 HFCLKCLKCFG (Offset = 1110h) [Reset = 1XXXXX00h]

HFCLKCLKCFG is shown in Figure 2-25 and described in Table 2-35.

Return to the Summary Table.

High-frequency clock (HFCLK) configuration

**Figure 2-25. HFCLKCLKCFG**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | HFCLKFLTCHK | RESERVED | | | |
| R-0h | | | R/W-1h | R-0h | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | HFXTRSEL | | RESERVED | | | |
| R-0h | | R/W-0h | | R-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| HFXTTIME | | | | | | | |
| R/W-0h | | | | | | | |

**Table 2-35. HFCLKCLKCFG Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | RESERVED | R | 0h | |
| 28 | HFCLKFLTCHK | R/W | 1h | HFCLKFLTCHK enables or disables the HFCLK startup monitor.<br>0h = HFCLK startup is not checked<br>1h = HFCLK startup is checked |
| 27-14 | RESERVED | R | 0h | |
| 13-12 | HFXTRSEL | R/W | 0h | HFXT Range Select<br>0h = 4MHz <= HFXT frequency <= 8MHz<br>1h = 8MHz < HFXT frequency <= 16MHz<br>2h = 16MHz < HFXT frequency <= 32MHz<br>3h = 32MHz < HFXT frequency <= 48MHz |
| 11-8 | RESERVED | R | 0h | |
| 7-0 | HFXTTIME | R/W | 0h | HFXTTIME specifies the HFXT startup time in 64us resolution. If the HFCLK startup monitor is enabled (HFCLKFLTCHK), HFXT will be checked after this time expires.<br>0h = Minimum startup time (approximatly zero)<br>FFh = Maximum startup time (approximatly 16.32ms) |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.16 LFCLKCFG (Offset = 1114h) [Reset = 000000XXh]

LFCLKCFG is shown in Figure 2-26 and described in Table 2-36.

Return to the Summary Table.

Low frequency crystal oscillator (LFXT) configuration

**Figure 2-26. LFCLKCFG**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | LOWCAP |
| R-0h | | | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | MONITOR | RESERVED | | XT1DRIVE | |
| R-0h | | | R/W-0h | R-0h | | R/W-3h | |

**Table 2-36. LFCLKCFG Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-9 | RESERVED | R | 0h | |
| 8 | LOWCAP | R/W | 0h | LOWCAP controls the low-power LFXT mode. When the LFXT load capacitance is less than 3pf, LOWCAP may be set for reduced power consumption.<br>0h = LFXT low capacitance mode is disabled<br>1h = LFXT low capacitance mode is enabled |
| 7-5 | RESERVED | R | 0h | |
| 4 | MONITOR | R/W | 0h | MONITOR enables or disables the LFCLK monitor, which continuously checks LFXT or LFCLK_IN for a clock stuck fault.<br>0h = Clock monitor is disabled<br>1h = Clock monitor is enabled |
| 3-2 | RESERVED | R | 0h | |
| 1-0 | XT1DRIVE | R/W | 3h | XT1DRIVE selects the low frequency crystal oscillator (LFXT) drive strength.<br>0h = Lowest drive and current<br>1h = Lower drive and current<br>2h = Higher drive and current<br>3h = Highest drive and current |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.17 SYSPLLCFG0 (Offset = 1120h) [Reset = 000000XXh]

SYSPLLCFG0 is shown in Figure 2-27 and described in Table 2-37.

Return to the Summary Table.

SYSPLL reference and output configuration

**Figure 2-27. SYSPLLCFG0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | RDIVCLK2X | | | |
| R-0h | | | | R/W-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RDIVCLK1 | | | | RDIVCLK0 | | | |
| R/W-0h | | | | R/W-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | ENABLECLK2X | ENABLECLK1 | ENABLECLK0 | RESERVED | | MCLK2XVCO | SYSPLLREF |
| R-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | | R/W-0h | R/W-0h |

**Table 2-37. SYSPLLCFG0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | RESERVED | R | 0h | |
| 19-16 | RDIVCLK2X | R/W | 0h | RDIVCLK2X sets the final divider for the SYSPLLCLK2X output.<br>0h = SYSPLLCLK1 is divided by 1<br>1h = SYSPLLCLK1 is divided by 2<br>2h = SYSPLLCLK1 is divided by 3<br>3h = SYSPLLCLK1 is divided by 4<br>4h = SYSPLLCLK1 is divided by 5<br>5h = SYSPLLCLK1 is divided by 6<br>6h = SYSPLLCLK1 is divided by 7<br>7h = SYSPLLCLK1 is divided by 8<br>8h = SYSPLLCLK1 is divided by 9<br>9h = SYSPLLCLK1 is divided by 10<br>Ah = SYSPLLCLK1 is divided by 11<br>Bh = SYSPLLCLK1 is divided by 12<br>Ch = SYSPLLCLK1 is divided by 13<br>Dh = SYSPLLCLK1 is divided by 14<br>Eh = SYSPLLCLK1 is divided by 15<br>Fh = SYSPLLCLK1 is divided by 16 |
| 15-12 | RDIVCLK1 | R/W | 0h | RDIVCLK1 sets the final divider for the SYSPLLCLK1 output.<br>0h = SYSPLLCLK1 is divided by 2<br>1h = SYSPLLCLK1 is divided by 4<br>2h = SYSPLLCLK1 is divided by 6<br>3h = SYSPLLCLK1 is divided by 8<br>4h = SYSPLLCLK1 is divided by 10<br>5h = SYSPLLCLK1 is divided by 12<br>6h = SYSPLLCLK1 is divided by 14<br>7h = SYSPLLCLK1 is divided by 16<br>8h = SYSPLLCLK1 is divided by 18<br>9h = SYSPLLCLK1 is divided by 20<br>Ah = SYSPLLCLK1 is divided by 22<br>Bh = SYSPLLCLK1 is divided by 24<br>Ch = SYSPLLCLK1 is divided by 26<br>Dh = SYSPLLCLK1 is divided by 28<br>Eh = SYSPLLCLK1 is divided by 30<br>Fh = SYSPLLCLK1 is divided by 32 |



Copyright © 2026 Texas Instruments Incorporated

**Table 2-37. SYSPLLCFG0 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 11-8 | RDIVCLK0 | R/W | 0h | RDIVCLK0 sets the final divider for the SYSPLLCLK0 output.<br>0h = SYSPLLCLK0 is divided by 2<br>1h = SYSPLLCLK0 is divided by 4<br>2h = SYSPLLCLK0 is divided by 6<br>3h = SYSPLLCLK0 is divided by 8<br>4h = SYSPLLCLK0 is divided by 10<br>5h = SYSPLLCLK0 is divided by 12<br>6h = SYSPLLCLK0 is divided by 14<br>7h = SYSPLLCLK0 is divided by 16<br>8h = SYSPLLCLK0 is divided by 18<br>9h = SYSPLLCLK0 is divided by 20<br>Ah = SYSPLLCLK0 is divided by 22<br>Bh = SYSPLLCLK0 is divided by 24<br>Ch = SYSPLLCLK0 is divided by 26<br>Dh = SYSPLLCLK0 is divided by 28<br>Eh = SYSPLLCLK0 is divided by 30<br>Fh = SYSPLLCLK0 is divided by 32 |
| 7 | RESERVED | R | 0h | |
| 6 | ENABLECLK2X | R/W | 0h | ENABLECLK2X enables or disables the SYSPLLCLK2X output.<br>0h = SYSPLLCLK2X is disabled<br>1h = SYSPLLCLK2X is enabled |
| 5 | ENABLECLK1 | R/W | 0h | ENABLECLK1 enables or disables the SYSPLLCLK1 output.<br>0h = SYSPLLCLK1 is disabled<br>1h = SYSPLLCLK1 is enabled |
| 4 | ENABLECLK0 | R/W | 0h | ENABLECLK0 enables or disables the SYSPLLCLK0 output.<br>0h = SYSPLLCLK0 is disabled<br>1h = SYSPLLCLK0 is enabled |
| 3-2 | RESERVED | R | 0h | |
| 1 | MCLK2XVCO | R/W | 0h | MCLK2XVCO selects the SYSPLL output which is sent to the HSCLK mux for use by MCLK.<br>0h = The SYSPLLCLK0 output is sent to the HSCLK mux<br>1h = The SYSPLLCLK2X output is sent to the HSCLK mux |
| 0 | SYSPLLREF | R/W | 0h | SYSPLLREF selects the system PLL (SYSPLL) reference clock source.<br>0h = SYSPLL reference is SYSOSC<br>1h = SYSPLL reference is HFCLK |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.18 SYSPLLCFG1 (Offset = 1124h) [Reset = 000000XXh]

SYSPLLCFG1 is shown in Figure 2-28 and described in Table 2-38.

Return to the Summary Table.

SYSPLL reference and feedback divider

**Figure 2-28. SYSPLLCFG1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | QDIV | | | | | | | RESERVED | | | | | | PDIV | |
| R-0h | R/W-0h | | | | | | | R-0h | | | | | | R/W-0h | |

**Table 2-38. SYSPLLCFG1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-15 | RESERVED | R | 0h | |
| 14-8 | QDIV | R/W | 0h | QDIV selects the SYSPLL feedback path divider.<br>0h = Divide-by-one is not a valid QDIV option<br>1h = Feedback path is divided by 2<br>7Eh = Feedback path is divided by 127 (0x7E) |
| 7-2 | RESERVED | R | 0h | |
| 1-0 | PDIV | R/W | 0h | PDIV selects the SYSPLL reference clock prescale divider.<br>0h = SYSPLLREF is not divided<br>1h = SYSPLLREF is divided by 2<br>2h = SYSPLLREF is divided by 4<br>3h = SYSPLLREF is divided by 8 |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.19 SYSPLLPARAM0 (Offset = 1128h) [Reset = X0X0X0X0h]

SYSPLLPARAM0 is shown in Figure 2-29 and described in Table 2-39.

Return to the Summary Table.

SYSPLL PARAM0 (load from FACTORY region)

**Figure 2-29. SYSPLLPARAM0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CAPBOVERRIDE | RESERVED | | CAPBVAL | | | | |
| R/W-0h | R-0h | | R/W-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | CPCURRENT | | | | | |
| R-0h | | R/W-0h | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | STARTTIMELP | | | | | |
| R-0h | | R/W-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | STARTTIME | | | | | |
| R-0h | | R/W-0h | | | | | |

**Table 2-39. SYSPLLPARAM0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | CAPBOVERRIDE | R/W | 0h | CAPBOVERRIDE controls the override for Cap B<br>0h = Cap B override disabled<br>1h = Cap B override enabled |
| 30-29 | RESERVED | R | 0h | |
| 28-24 | CAPBVAL | R/W | 0h | Override value for Cap B |
| 23-22 | RESERVED | R | 0h | |
| 21-16 | CPCURRENT | R/W | 0h | Charge pump current |
| 15-14 | RESERVED | R | 0h | |
| 13-8 | STARTTIMELP | R/W | 0h | Startup time from low power mode exit to locked clock, in 1us resolution |
| 7-6 | RESERVED | R | 0h | |
| 5-0 | STARTTIME | R/W | 0h | Startup time from enable to locked clock, in 1us resolution |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.20 SYSPLLPARAM1 (Offset = 112Ch) [Reset = 00XX00X0h]

SYSPLLPARAM1 is shown in Figure 2-30 and described in Table 2-40.

Return to the Summary Table.

SYSPLL PARAM1 (load from FACTORY region)

**Figure 2-30. SYSPLLPARAM1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LPFRESC | | | | | | | | RESERVED | | | | | | LPFRESA | |
| R/W-0h | | | | | | | | R-0h | | | | | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LPFRESA | | | | | | | | RESERVED | | | LPFCAPA | | | | |
| R/W-0h | | | | | | | | R-0h | | | R/W-0h | | | | |

**Table 2-40. SYSPLLPARAM1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | LPFRESC | R/W | 0h | Loop filter Res C |
| 23-18 | RESERVED | R | 0h | |
| 17-8 | LPFRESA | R/W | 0h | Loop filter Res A |
| 7-5 | RESERVED | R | 0h | |
| 4-0 | LPFCAPA | R/W | 0h | Loop filter Cap A |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.21 GENCLKCFG (Offset = 1138h) [Reset = 00000X0Xh]

GENCLKCFG is shown in Figure 2-31 and described in Table 2-41.

Return to the Summary Table.

General clock configuration

**Figure 2-31. GENCLKCFG**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | FCCTRIGCNT | | | | |
| R-0h | | | R/W-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| ANACPUMPCFG | | FCCLVLTRIG | FCCTRIGSRC | FCCSELCLK | | | |
| R/W-0h | | R/W-0h | R/W-0h | R/W-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| HFCLK4MFPCLKDIV | | | | RESERVED | | MFPCLKSRC | CANCLKSRC |
| R/W-0h | | | | R-0h | | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| EXCLKDIVEN | EXCLKDIVVAL | | | RESERVED | EXCLKSRC | | |
| R/W-0h | R/W-0h | | | R-0h | R/W-0h | | |

**Table 2-41. GENCLKCFG Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | RESERVED | R | 0h | |
| 28-24 | FCCTRIGCNT | R/W | 0h | FCCTRIGCNT specifies the number of trigger clock periods in the trigger window. FCCTRIGCNT=0h (one trigger clock period) up to 1Fh (32 trigger clock periods) may be specified. |
| 23-22 | ANACPUMPCFG | R/W | 0h | ANACPUMPCFG selects the analog mux charge pump (VBOOST) enable method.<br>0h = VBOOST is enabled on request from a COMP, GPAMP, or OPA<br>1h = VBOOST is enabled when the device is in RUN or SLEEP mode, or when a COMP/GPAMP/OPA is enabled<br>2h = VBOOST is always enabled |
| 21 | FCCLVLTRIG | R/W | 0h | FCCLVLTRIG selects the frequency clock counter (FCC) trigger mode.<br>0h = Rising edge to rising edge triggered<br>1h = Level triggered |
| 20 | FCCTRIGSRC | R/W | 0h | FCCTRIGSRC selects the frequency clock counter (FCC) trigger source.<br>0h = FCC trigger is the external pin<br>1h = FCC trigger is the LFCLK |
| 19-16 | FCCSELCLK | R/W | 0h | FCCSELCLK selectes the frequency clock counter (FCC) clock source.<br>0h = FCC clock is MCLK<br>1h = FCC clock is SYSOSC<br>2h = FCC clock is HFCLK<br>3h = FCC clock is the CLK_OUT selection<br>4h = FCC clock is SYSPLLCLK0<br>5h = FCC clock is SYSPLLCLK1<br>6h = FCC clock is SYSPLLCLK2X<br>7h = FCC clock is the FCCIN external input |



Copyright © 2026 Texas Instruments Incorporated

**Table 2-41. GENCLKCFG Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15-12 | HFCLK4MFPCLKDIV | R/W | 0h | HFCLK4MFPCLKDIV selects the divider applied to HFCLK when HFCLK is used as the MFPCLK source. Integer dividers from /1 to /16 may be selected.<br>0h = HFCLK is not divided before being used for MFPCLK<br>1h = HFCLK is divided by 2 before being used for MFPCLK<br>2h = HFCLK is divided by 3 before being used for MFPCLK<br>3h = HFCLK is divided by 4 before being used for MFPCLK<br>4h = HFCLK is divided by 5 before being used for MFPCLK<br>5h = HFCLK is divided by 6 before being used for MFPCLK<br>6h = HFCLK is divided by 7 before being used for MFPCLK<br>7h = HFCLK is divided by 8 before being used for MFPCLK<br>8h = HFCLK is divided by 9 before being used for MFPCLK<br>9h = HFCLK is divided by 10 before being used for MFPCLK<br>Ah = HFCLK is divided by 11 before being used for MFPCLK<br>Bh = HFCLK is divided by 12 before being used for MFPCLK<br>Ch = HFCLK is divided by 13 before being used for MFPCLK<br>Dh = HFCLK is divided by 14 before being used for MFPCLK<br>Eh = HFCLK is divided by 15 before being used for MFPCLK<br>Fh = HFCLK is divided by 16 before being used for MFPCLK |
| 11-10 | RESERVED | R | 0h | |
| 9 | MFPCLKSRC | R/W | 0h | MFPCLKSRC selects the MFPCLK (middle frequency precision clock) source.<br>0h = MFPCLK is sourced from SYSOSC<br>1h = MFPCLK is sourced from HFCLK |
| 8 | CANCLKSRC | R/W | 0h | CANCLKSRC selects the CANCLK source.<br>0h = CANCLK source is HFCLK<br>1h = CANCLK source is SYSPLLCLK1 |
| 7 | EXCLKDIVEN | R/W | 0h | EXCLKDIVEN enables or disables the divider function of the CLK_OUT external clock output block.<br>0h = CLock divider is disabled (passthrough, EXCLKDIVVAL is not applied)<br>1h = Clock divider is enabled (EXCLKDIVVAL is applied) |
| 6-4 | EXCLKDIVVAL | R/W | 0h | EXCLKDIVVAL selects the divider value for the divider in the CLK_OUT external clock output block.<br>0h = CLK_OUT source is divided by 2<br>1h = CLK_OUT source is divided by 4<br>2h = CLK_OUT source is divided by 6<br>3h = CLK_OUT source is divided by 8<br>4h = CLK_OUT source is divided by 10<br>5h = CLK_OUT source is divided by 12<br>6h = CLK_OUT source is divided by 14<br>7h = CLK_OUT source is divided by 16 |
| 3 | RESERVED | R | 0h | |
| 2-0 | EXCLKSRC | R/W | 0h | EXCLKSRC selects the source for the CLK_OUT external clock output block. ULPCLK and MFPCLK require the CLK_OUT divider (EXCLKDIVEN) to be enabled<br>0h = CLK_OUT is SYSOSC<br>1h = CLK_OUT is ULPCLK (EXCLKDIVEN must be enabled)<br>2h = CLK_OUT is LFCLK<br>3h = CLK_OUT is MFPCLK (EXCLKDIVEN must be enabled)<br>4h = CLK_OUT is HFCLK<br>5h = CLK_OUT is SYSPLLCLK1 (SYSPLLCLK1 must be <=48MHz) |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.22 GENCLKEN (Offset = 113Ch) [Reset = 0000000Xh]

GENCLKEN is shown in Figure 2-32 and described in Table 2-42.

Return to the Summary Table.

General clock enable control

**Figure 2-32. GENCLKEN**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | MFPCLKEN | RESERVED | | | EXCLKEN |
| R-0h | | | R/W-0h | R-0h | | | R/W-0h |

**Table 2-42. GENCLKEN Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-5 | RESERVED | R | 0h | |
| 4 | MFPCLKEN | R/W | 0h | MFPCLKEN enables the middle frequency precision clock (MFPCLK).<br>0h = MFPCLK is disabled<br>1h = MFPCLK is enabled |
| 3-1 | RESERVED | R | 0h | |
| 0 | EXCLKEN | R/W | 0h | EXCLKEN enables the CLK_OUT external clock output block.<br>0h = CLK_OUT block is disabled<br>1h = CLK_OUT block is enabled |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.23 PMODECFG (Offset = 1140h) [Reset = 00000000h]

PMODECFG is shown in Figure 2-33 and described in Table 2-43.

Return to the Summary Table.

Power mode configuration

**Figure 2-33. PMODECFG**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | | | DSLEEP | |
| R-0h | | | | | | | | | | | | | | R/W-0h | |

**Table 2-43. PMODECFG Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | |
| 1-0 | DSLEEP | R/W | 0h | DSLEEP selects the operating mode to enter upon a DEEPSLEEP request from the CPU.<br>0h = STOP mode is entered<br>1h = STANDBY mode is entered<br>2h = SHUTDOWN mode is entered<br>3h = Reserved |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.24 FCC (Offset = 1150h) [Reset = 00000000h]

FCC is shown in Figure 2-34 and described in Table 2-44.

Return to the Summary Table.

Frequency clock counter (FCC) count

**Figure 2-34. FCC**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | DATA | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | R-0h | | | | | | | | | | | | | | | | | | | | | |

**Table 2-44. FCC Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-22 | RESERVED | R | 0h | |
| 21-0 | DATA | R | 0h | Frequency clock counter (FCC) count value. |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.25 SYSOSCTRIMUSER (Offset = 1170h) [Reset = 0000X0XXh]

SYSOSCTRIMUSER is shown in Figure 2-35 and described in Table 2-45.

Return to the Summary Table.

SYSOSC user-specified trim

**Figure 2-35. SYSOSCTRIMUSER**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | RDIV | | | | |
| R-0h | | | R/W-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RDIV | | | | RESFINE | | | |
| R/W-0h | | | | R/W-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | RESCOARSE | | | | | |
| R-0h | | R/W-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | CAP | | | RESERVED | | FREQ | |
| R-0h | R/W-0h | | | R-0h | | R/W-0h | |

**Table 2-45. SYSOSCTRIMUSER Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | RESERVED | R | 0h | |
| 28-20 | RDIV | R/W | 0h | RDIV specifies the frequency correction loop (FCL) resistor trim. This value changes with the target frequency. |
| 19-16 | RESFINE | R/W | 0h | RESFINE specifies the resister fine trim. This value changes with the target frequency. |
| 15-14 | RESERVED | R | 0h | |
| 13-8 | RESCOARSE | R/W | 0h | RESCOARSE specifies the resister coarse trim. This value changes with the target frequency. |
| 7 | RESERVED | R | 0h | |
| 6-4 | CAP | R/W | 0h | CAP specifies the SYSOSC capacitor trim. This value changes with the target frequency. |
| 3-2 | RESERVED | R | 0h | |
| 1-0 | FREQ | R/W | 0h | FREQ specifies the target user-trimmed frequency for SYSOSC.<br>0h = Reserved<br>1h = 16MHz user frequency<br>2h = 24MHz user frequency<br>3h = Reserved |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.26 SRAMBOUNDARY (Offset = 1178h) [Reset = 000000XXh]

SRAMBOUNDARY is shown in Figure 2-36 and described in Table 2-46.

Return to the Summary Table.

SRAM Write Boundary

**Figure 2-36. SRAMBOUNDARY**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | ADDR | | | | | | | | | | | | | | | RESERVED | | | | |
| R-0h | | | | | | | | | | | | R/W-0h | | | | | | | | | | | | | | | R-0h | | | | |

**Table 2-46. SRAMBOUNDARY Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | RESERVED | R | 0h | |
| 19-5 | ADDR | R/W | 0h | SRAM boundary configuration. The value configured into this acts such that: SRAM accesses to addresses less than or equal value will be RW only. SRAM accesses to addresses greater than value will be RX only. Value of 0 is not valid (system will have no stack). If set to 0, the system acts as if the entire SRAM is RWX. Any non-zero value can be configured, including a value = SRAM size. |
| 4-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.27 SRAMBOUNDARYHIGH (Offset = 117Ch) [Reset = 000000XXh]

SRAMBOUNDARYHIGH is shown in Figure 2-37 and described in Table 2-47.

Return to the Summary Table.

SRAM Write Boundary High

**Figure 2-37. SRAMBOUNDARYHIGH**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | ADDR | | | | | | | | | | | | | | | RESERVED | | | | |
| R-0h | | | | | | | | | | | | R/W-0h | | | | | | | | | | | | | | | R-0h | | | | |

**Table 2-47. SRAMBOUNDARYHIGH Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | RESERVED | R | 0h | |
| 19-5 | ADDR | R/W | 0h | SRAM boundary high configuration. The value configured into this acts such that: SRAM accesses to greater than value will be RW only. SRAM accesses to addresses less than or equal to value will be RX until meeting SRAMBOUNDARY MMR Value. Value of 0 is default. If value set to less than or equal to SRAMBOUNDARY MMR, the system will only use the SRAMBOUNDARY MMR as criteria. |
| 4-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.28 SYSTEMCFG (Offset = 1180h) [Reset = 00XXXXXXh]

SYSTEMCFG is shown in Figure 2-38 and described in Table 2-48.

Return to the Summary Table.

System configuration

**Figure 2-38. SYSTEMCFG**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | FLASHECCRSTDIS | WWDTLP1RSTDIS | WWDTLP0RSTDIS |
| R-0h | | | | | R/W-1h | R/W-1h | R/W-0h |

**Table 2-48. SYSTEMCFG Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | The key value of 1Bh (27) must be written to KEY together with contents to be updated. Reads as 0<br>1Bh = Issue write |
| 23-3 | RESERVED | R | 0h | |
| 2 | FLASHECCRSTDIS | R/W | 1h | FLASHECCRSTDIS specifies whether a flash ECC double error detect (DED) will trigger a SYSRST or an NMI.<br>0h = Flash ECC DED will trigger a SYSRST<br>1h = Flash ECC DED will trigger a NMI |
| 1 | WWDTLP1RSTDIS | R/W | 1h | WWDTLP1RSTDIS specifies whether a WWDT Error Event will trigger a SYSRST or an NMI.<br>0h = WWDTLP1 Error Event will trigger a SYSRST<br>1h = WWDTLP1 Error Event will trigger an NMI |
| 0 | WWDTLP0RSTDIS | R/W | 0h | WWDTLP0RSTDIS specifies whether a WWDT Error Event will trigger a BOOTRST or an NMI.<br>0h = WWDTLP0 Error Event will trigger a BOOTRST<br>1h = WWDTLP0 Error Event will trigger an NMI |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.29 SRAMCFG (Offset = 1184h) [Reset = 00XXXXXXh]

SRAMCFG is shown in Figure 2-39 and described in Table 2-49.

Return to the Summary Table.

System SRAM configuration

**Figure 2-39. SRAMCFG**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | BANKSTOP1 | RESERVED |
| R-0h | | | | | | R/W-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | BANKOFF1 | RESERVED |
| R-0h | | | | | | R/W-0h | R-0h |

**Table 2-49. SRAMCFG Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | The key value of B5h (181) must be written to KEY together with contents to be updated. Reads as 0<br>B5h = Issue write |
| 23-10 | RESERVED | R | 0h | |
| 9 | BANKSTOP1 | R/W | 0h | SRAM BANK1 power level for STOP mode<br>0h = SRAM BANK1 power OFF for STOP mode<br>1h = SRAM BANK1 power RETAIN for STOP mode |
| 8-2 | RESERVED | R | 0h | |
| 1 | BANKOFF1 | R/W | 0h | SRAM BANK1 power level for RUN mode<br>0h = SRAM BANK1 power ON for RUN mode<br>1h = SRAM BANK1 power OFF for RUN mode |
| 0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.30 WRITELOCK (Offset = 1200h) [Reset = 00000000h]

WRITELOCK is shown in Figure 2-40 and described in Table 2-50.

Return to the Summary Table.

SYSCTL register write lockout

**Figure 2-40. WRITELOCK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | ACTIVE |
| R-0h | | | | | | | R/W-0h |

**Table 2-50. WRITELOCK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | ACTIVE | R/W | 0h | ACTIVE controls whether critical SYSCTL registers are write protected or not.<br>0h = Allow writes to lockable registers<br>1h = Disallow writes to lockable registers |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.31 CLKSTATUS (Offset = 1204h) [Reset = XXXXX0XXh]

CLKSTATUS is shown in Figure 2-41 and described in Table 2-51.

Return to the Summary Table.

Clock module (CKM) status

**Figure 2-41. CLKSTATUS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| ANACLKERR | RESERVED | SYSPLLBLKUPD | HFCLKBLKUPD | RESERVED | | FCCDONE | FCLMODE |
| R-0h | R-0h | R-0h | R-0h | R-0h | | R-0h | R-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| LFCLKFAIL | RESERVED | HSCLKGOOD | HSCLKDEAD | RESERVED | | CURMCLKSEL | CURHSCLKSEL |
| R-0h | R-0h | R-0h | R-0h | R-0h | | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | SYSPLLOFF | HFCLKOFF | HSCLKSOFF | LFOSCGOOD | LFXTGOOD | SYSPLLGOOD | HFCLKGOOD |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LFCLKMUX | | RESERVED | HSCLKMUX | RESERVED | | SYSOSCFREQ | |
| R-0h | | R-0h | R-0h | R-0h | | R-0h | |

**Table 2-51. CLKSTATUS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | ANACLKERR | R | 0h | ANACLKERR is set when the device clock configuration does not support an enabled analog peripheral mode and the analog peripheral may not be functioning as expected.<br>0h = No analog clock errors detected<br>1h = Analog clock error detected |
| 30 | RESERVED | R | 0h | |
| 29 | SYSPLLBLKUPD | R | 0h | SYSPLLBLKUPD indicates when writes to SYSPLLCFG0/1 and SYSPLLPARAM0/1 are blocked.<br>0h = writes to SYSPLLCFG0/1 and SYSPLLPARAM0/1 are allowed<br>1h = writes to SYSPLLCFG0/1 and SYSPLLPARAM0/1 are blocked |
| 28 | HFCLKBLKUPD | R | 0h | HFCLKBLKUPD indicates when writes to the HFCLKCLKCFG register are blocked.<br>0h = Writes to HFCLKCLKCFG are allowed<br>1h = Writes to HFCLKCLKCFG are blocked |
| 27-26 | RESERVED | R | 0h | |
| 25 | FCCDONE | R | 0h | FCCDONE indicates when a frequency clock counter capture is complete.<br>0h = FCC capture is not done<br>1h = FCC capture is done |
| 24 | FCLMODE | R | 0h | FCLMODE indicates if the SYSOSC frequency correction loop (FCL) is enabled.<br>0h = SYSOSC FCL is disabled<br>1h = SYSOSC FCL is enabled |
| 23 | LFCLKFAIL | R | 0h | LFCLKFAIL indicates when the continous LFCLK monitor detects a LFXT or LFCLK_IN clock stuck failure.<br>0h = No LFCLK fault detected<br>1h = LFCLK stuck fault detected |
| 22 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

**Table 2-51. CLKSTATUS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 21 | HSCLKGOOD | R | 0h | HSCLKGOOD is set by hardware if the selected clock source for HSCLK started successfully.<br>0h = The HSCLK source did not start correctly<br>1h = The HSCLK source started correctly |
| 20 | HSCLKDEAD | R | 0h | HSCLKDEAD is set by hardware if the selected source for HSCLK was started but did not start successfully.<br>0h = The HSCLK source was not started or started correctly<br>1h = The HSCLK source did not start correctly |
| 19-18 | RESERVED | R | 0h | |
| 17 | CURMCLKSEL | R | 0h | CURMCLKSEL indicates if MCLK is currently sourced from LFCLK.<br>0h = MCLK is not sourced from LFCLK<br>1h = MCLK is sourced from LFCLK |
| 16 | CURHSCLKSEL | R | 0h | CURHSCLKSEL indicates the current clock source for HSCLK.<br>0h = HSCLK is currently sourced from the SYSPLL<br>1h = HSCLK is currently sourced from the HFCLK |
| 15 | RESERVED | R | 0h | |
| 14 | SYSPLLOFF | R | 0h | SYSPLLOFF indicates if the SYSPLL is disabled or was dead at startup. When the SYSPLL is started, SYSPLLOFF is cleared by hardware. Following startup of the SYSPLL, if the SYSPLL startup monitor determines that the SYSPLL was not started correctly, SYSPLLOFF is set.<br>0h = SYSPLL started correctly and is enabled<br>1h = SYSPLL is disabled or was dead startup |
| 13 | HFCLKOFF | R | 0h | HFCLKOFF indicates if the HFCLK is disabled or was dead at startup. When the HFCLK is started, HFCLKOFF is cleared by hardware. Following startup of the HFCLK, if the HFCLK startup monitor determines that the HFCLK was not started correctly, HFCLKOFF is set.<br>0h = HFCLK started correctly and is enabled<br>1h = HFCLK is disabled or was dead at startup |
| 12 | HSCLKSOFF | R | 0h | HSCLKSOFF is set when the high speed clock sources (SYSPLL, HFCLK) are disabled or dead. It is the logical AND of HFCLKOFF and SYSPLLOFF.<br>0h = SYSPLL, HFCLK, or both were started correctly and remain enabled<br>1h = SYSPLL and HFCLK are both either off or dead |
| 11 | LFOSCGOOD | R | 0h | LFOSCGOOD indicates when the LFOSC startup has completed and the LFOSC is ready for use.<br>0h = LFOSC is not ready<br>1h = LFOSC is ready |
| 10 | LFXTGOOD | R | 0h | LFXTGOOD indicates if the LFXT started correctly. When the LFXT is started, LFXTGOOD is cleared by hardware. After the startup settling time has expired, the LFXT status is tested. If the LFXT started successfully the LFXTGOOD bit is set, else it is left cleared.<br>0h = LFXT did not start correctly<br>1h = LFXT started correctly |
| 9 | SYSPLLGOOD | R | 0h | SYSPLLGOOD indicates if the SYSPLL started correctly. When the SYSPLL is started, SYSPLLGOOD is cleared by hardware. After the startup settling time has expired, the SYSPLL status is tested. If the SYSPLL started successfully the SYSPLLGOOD bit is set, else it is left cleared.<br>0h = SYSPLL did not start correctly<br>1h = SYSPLL started correctly |
| 8 | HFCLKGOOD | R | 0h | HFCLKGOOD indicates that the HFCLK started correctly. When the HFXT is started or HFCLK_IN is selected as the HFCLK source, this bit will be set by hardware if a valid HFCLK is detected, and cleared if HFCLK is not operating within the expected range.<br>0h = HFCLK did not start correctly<br>1h = HFCLK started correctly |



Copyright © 2026 Texas Instruments Incorporated

**Table 2-51. CLKSTATUS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7-6 | LFCLKMUX | R | 0h | LFCLKMUX indicates if LFCLK is sourced from the internal LFOSC, the low frequency crystal (LFXT), or the LFCLK_IN digital clock input.<br>0h = LFCLK is sourced from the internal LFOSC<br>1h = LFCLK is sourced from the LFXT (crystal)<br>2h = LFCLK is sourced from LFCLK_IN (external digital clock input) |
| 5 | RESERVED | R | 0h | |
| 4 | HSCLKMUX | R | 0h | HSCLKMUX indicates if MCLK is currently sourced from the high-speed clock (HSCLK).<br>0h = MCLK is not sourced from HSCLK<br>1h = MCLK is sourced from HSCLK |
| 3-2 | RESERVED | R | 0h | |
| 1-0 | SYSOSCFREQ | R | 0h | SYSOSCFREQ indicates the current SYSOSC operating frequency.<br>0h = SYSOSC is at base frequency (32MHz)<br>1h = SYSOSC is at low frequency (4MHz)<br>2h = SYSOSC is at the user-trimmed frequency (16 or 24MHz)<br>3h = Reserved |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.32 SYSSTATUS (Offset = 1208h) [Reset = XXXXXXX0h]

SYSSTATUS is shown in Figure 2-42 and described in Table 2-52.

Return to the Summary Table.

System status information

**Figure 2-42. SYSSTATUS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REBOOTATTEMPTS | | RESERVED | | | | | |
| R-0h | | R-0h | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | SRAMBANK1READY | RESERVED |
| R-0h | | | | | | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | SHDNIOLOCK | SWDCFGDIS | EXTRSTPINDIS | RESERVED | | MCAN1READY | MCAN0READY |
| R-0h | R-0h | R-0h | R-0h | R-0h | | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | PMUIREFGOOD | ANACPUMPGOOD | BORLVL | BORCURTHRESHOLD | | FLASHSEC | FLASHDED |
| R-0h | R-0h | R-0h | R-0h | R-0h | | R-0h | R-0h |

**Table 2-52. SYSSTATUS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | REBOOTATTEMPTS | R | 0h | REBOOTATTEMPTS indicates the number of boot attempts taken before the user application starts. |
| 29-18 | RESERVED | R | 0h | |
| 17 | SRAMBANK1READY | R | 0h | SRAM BANK1 READY STATE<br>0h = SRAM BANK1 is NOT READY for access<br>1h = SRAM BANK1 is READY for access |
| 16-15 | RESERVED | R | 0h | |
| 14 | SHDNIOLOCK | R | 0h | SHDNIOLOCK indicates when IO is locked due to SHUTDOWN<br>0h = IO IS NOT Locked due to SHUTDOWN<br>1h = IO IS Locked due to SHUTDOWN |
| 13 | SWDCFGDIS | R | 0h | SWDCFGDIS indicates when user has disabled the use of SWD Port<br>0h = SWD Port Enabled<br>1h = SWD Port Disabled |
| 12 | EXTRSTPINDIS | R | 0h | EXTRSTPINDIS indicates when user has disabled the use of external reset pin<br>0h = External Reset Pin Enabled<br>1h = External Reset Pin Disabled |
| 11-10 | RESERVED | R | 0h | |
| 9 | MCAN1READY | R | 0h | MCAN1READY indicates when the MCAN1 peripheral is ready.<br>0h = MCAN1 is not ready<br>1h = MCAN1 is ready |
| 8 | MCAN0READY | R | 0h | MCAN0READY indicates when the MCAN0 peripheral is ready.<br>0h = MCAN0 is not ready<br>1h = MCAN0 is ready |
| 7 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

**Table 2-52. SYSSTATUS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 6 | PMUIREFGOOD | R | 0h | PMUIREFGOOD is set by hardware when the PMU current reference is ready.<br>0h = IREF is not ready<br>1h = IREF is ready |
| 5 | ANACPUMPGOOD | R | 0h | ANACPUMPGOOD is set by hardware when the VBOOST analog mux charge pump is ready.<br>0h = VBOOST is not ready<br>1h = VBOOST is ready |
| 4 | BORLVL | R | 0h | BORLVL indicates if a BOR event occured and the BOR threshold was switched to BOR0 by hardware.<br>0h = No BOR violation occured<br>1h = A BOR violation occured and the BOR threshold was switched to BOR0 |
| 3-2 | BORCURTHRESHOLD | R | 0h | BORCURTHRESHOLD indicates the active brown-out reset supply monitor configuration.<br>0h = Default minimum threshold; a BOR0- violation triggers a BOR<br>1h = A BOR1- violation generates a BORLVL interrupt<br>2h = A BOR2- violation generates a BORLVL interrupt<br>3h = A BOR3- violation generates a BORLVL interrupt |
| 1 | FLASHSEC | R | 0h | FLASHSEC indicates if a flash ECC single bit error was detected and corrected (SEC).<br>0h = No flash ECC single bit error detected<br>1h = Flash ECC single bit error was detected and corrected |
| 0 | FLASHDED | R | 0h | FLASHDED indicates if a flash ECC double bit error was detected (DED).<br>0h = No flash ECC double bit error detected<br>1h = Flash ECC double bit error detected |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.33 DEDERRADDR (Offset = 120Ch) [Reset = 00000000h]

DEDERRADDR is shown in Figure 2-43 and described in Table 2-53.

Return to the Summary Table.

Memory DED Address

**Figure 2-43. DEDERRADDR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADDR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 2-53. DEDERRADDR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | ADDR | R | 0h | Address of MEMORY DED Error. |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.34 RSTCAUSE (Offset = 1220h) [Reset = 00000000h]

RSTCAUSE is shown in Figure 2-44 and described in Table 2-54.

Return to the Summary Table.

Reset cause

**Figure 2-44. RSTCAUSE**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | ID | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | RC-0h | | | | |

**Table 2-54. RSTCAUSE Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-5 | RESERVED | R | 0h | |
| 4-0 | ID | RC | 0h | ID is a read-to-clear field which indicates the lowest level reset cause since the last read.<br>0h = No reset since last read<br>1h = POR- violation, SHUTDNSTOREx or PMU trim parity fault<br>2h = NRST triggered POR (>1s hold)<br>3h = Software triggered POR<br>4h = BOR0- violation<br>5h = SHUTDOWN mode exit<br>8h = Non-PMU trim parity fault<br>9h = Fatal clock failure<br>Ch = NRST triggered BOOTRST (<1s hold)<br>Dh = Software triggered BOOTRST<br>Eh = WWDT0 violation<br>10h = BSL exit<br>11h = BSL entry<br>13h = WWDT1 violation<br>14h = Flash uncorrectable ECC error<br>15h = CPULOCK violation<br>1Ah = Debug triggered SYSRST<br>1Bh = Software triggered SYSRST<br>1Ch = Debug triggered CPURST<br>1Dh = Software triggered CPURST |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.35 RESETLEVEL (Offset = 1300h) [Reset = 00000000h]

RESETLEVEL is shown in Figure 2-45 and described in Table 2-55.

Return to the Summary Table.

Reset level for application-triggered reset command

**Figure 2-45. RESETLEVEL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | | LEVEL | | |
| R-0h | | | | | | | | | | | | | R/W-0h | | |

**Table 2-55. RESETLEVEL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | RESERVED | R | 0h | |
| 2-0 | LEVEL | R/W | 0h | LEVEL is used to specify the type of reset to be issued when RESETCMD is set to generate a software triggered reset.<br>0h = Issue a SYSRST (CPU plus peripherals only)<br>1h = Issue a BOOTRST (CPU, peripherals, and boot configuration routine)<br>2h = Issue a SYSRST and enter the boot strap loader (BSL)<br>3h = Issue a power-on reset (POR)<br>4h = Issue a SYSRST and exit the boot strap loader (BSL) |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.36 RESETCMD (Offset = 1304h) [Reset = 00XXXXXXh]

RESETCMD is shown in Figure 2-46 and described in Table 2-56.

Return to the Summary Table.

Execute an application-triggered reset command

**Figure 2-46. RESETCMD**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| KEY | | | | | | | | RESERVED | | | | | | | |
| W-0h | | | | | | | | R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | | | | GO |
| R-0h | | | | | | | | | | | | | | | W-0h |

**Table 2-56. RESETCMD Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | The key value of E4h (228) must be written to KEY together with GO to trigger the reset.<br>E4h = Issue reset |
| 23-1 | RESERVED | R | 0h | |
| 0 | GO | W | 0h | Execute the reset specified in RESETLEVEL.LEVEL. Must be written together with the KEY.<br>1h = Issue reset |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.37 BORTHRESHOLD (Offset = 1308h) [Reset = 00000000h]

BORTHRESHOLD is shown in Figure 2-47 and described in Table 2-57.

Return to the Summary Table.

BOR threshold selection

**Figure 2-47. BORTHRESHOLD**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | | | LEVEL | |
| R-0h | | | | | | | | | | | | | | R/W-0h | |

**Table 2-57. BORTHRESHOLD Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | |
| 1-0 | LEVEL | R/W | 0h | LEVEL specifies the desired BOR threshold and BOR mode.<br>0h = Default minimum threshold; a BOR0- violation triggers a BOR<br>1h = A BOR1- violation generates a BORLVL interrupt<br>2h = A BOR2- violation generates a BORLVL interrupt<br>3h = A BOR3- violation generates a BORLVL interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.38 BORCLRCMD (Offset = 130Ch) [Reset = 00XXXXXXh]

BORCLRCMD is shown in Figure 2-48 and described in Table 2-58.

Return to the Summary Table.

Set the BOR threshold

**Figure 2-48. BORCLRCMD**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| KEY | | | | | | | | RESERVED | | | | | | | |
| W-0h | | | | | | | | R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | | | | GO |
| R-0h | | | | | | | | | | | | | | | W-0h |

**Table 2-58. BORCLRCMD Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | The key value of C7h (199) must be written to KEY together with GO to trigger the clear and BOR threshold change.<br>C7h = Issue clear |
| 23-1 | RESERVED | R | 0h | |
| 0 | GO | W | 0h | GO clears any prior BOR violation status indications and attempts to change the active BOR mode to that specified in the LEVEL field of the BORTHRESHOLD register.<br>1h = Issue clear |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.39 SYSOSCFCLCTL (Offset = 1310h) [Reset = 00XXXXXXh]

SYSOSCFCLCTL is shown in Figure 2-49 and described in Table 2-59.

Return to the Summary Table.

SYSOSC frequency correction loop (FCL) ROSC enable

**Figure 2-49. SYSOSCFCLCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | SETUSEFCL |
| R-0h | | | | | | | W-0h |

**Table 2-59. SYSOSCFCLCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | The key value of 2Ah (42) must be written to KEY together with SETUSEFCL to enable the FCL.<br>2Ah = Issue Command |
| 23-1 | RESERVED | R | 0h | |
| 0 | SETUSEFCL | W | 0h | Set SETUSEFCL to enable the frequency correction loop in SYSOSC. Once enabled, this state is locked until the next BOOTRST.<br>1h = Enable the SYSOSC FCL |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.40 LFXTCTL (Offset = 1314h) [Reset = 00XXXXXXh]

LFXTCTL is shown in Figure 2-50 and described in Table 2-60.

Return to the Summary Table.

LFXT and LFCLK control

**Figure 2-50. LFXTCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | SETUSELFXT | STARTLFXT |
| R-0h | | | | | | W-0h | W-0h |

**Table 2-60. LFXTCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | The key value of 91h (145) must be written to KEY together with either STARTLFXT or SETUSELFXT to set the corresponding bit.<br>91h = Issue command |
| 23-2 | RESERVED | R | 0h | |
| 1 | SETUSELFXT | W | 0h | Set SETUSELFXT to switch LFCLK to LFXT. Once set, SETUSELFXT remains set until the next BOOTRST.<br>0h = 0<br>1h = Use LFXT as the LFCLK source |
| 0 | STARTLFXT | W | 0h | Set STARTLFXT to start the low frequency crystal oscillator (LFXT). Once set, STARTLFXT remains set until the next BOOTRST.<br>0h = LFXT not started<br>1h = Start LFXT |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.41 EXLFCTL (Offset = 1318h) [Reset = 00XXXXXXh]

EXLFCTL is shown in Figure 2-51 and described in Table 2-61.

Return to the Summary Table.

LFCLK_IN and LFCLK control

**Figure 2-51. EXLFCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | SETUSEEXLF |
| R-0h | | | | | | | W-0h |

**Table 2-61. EXLFCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | The key value of 36h (54) must be written to KEY together with SETUSEEXLF to set SETUSEEXLF.<br>36h = Issue command |
| 23-1 | RESERVED | R | 0h | |
| 0 | SETUSEEXLF | W | 0h | Set SETUSEEXLF to switch LFCLK to the LFCLK_IN digital clock input. Once set, SETUSEEXLF remains set until the next BOOTRST.<br>1h = Use LFCLK_IN as the LFCLK source |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.42 SHDNIOREL (Offset = 131Ch) [Reset = 00XXXXXXh]

SHDNIOREL is shown in Figure 2-52 and described in Table 2-62.

Return to the Summary Table.

SHUTDOWN IO release control

**Figure 2-52. SHDNIOREL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | RELEASE |
| R-0h | | | | | | | W-0h |

**Table 2-62. SHDNIOREL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | The key value 91h must be written to KEY together with RELEASE to set RELEASE.<br>91h = Issue command |
| 23-1 | RESERVED | R | 0h | |
| 0 | RELEASE | W | 0h | Set RELEASE to release the IO after a SHUTDOWN mode exit.<br>1h = Release IO |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.43 EXRSTPIN (Offset = 1320h) [Reset = 00XXXXXXh]

EXRSTPIN is shown in Figure 2-53 and described in Table 2-63.

Return to the Summary Table.

Disable the reset function of the NRST pin

**Figure 2-53. EXRSTPIN**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | DISABLE |
| R-0h | | | | | | | W-0h |

**Table 2-63. EXRSTPIN Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | The key value 1Eh must be written together with DISABLE to disable the reset function.<br>1Eh = Issue command |
| 23-1 | RESERVED | R | 0h | |
| 0 | DISABLE | W | 0h | Set DISABLE to disable the reset function of the NRST pin. Once set, this configuration is locked until the next POR.<br>0h = Reset function of NRST pin is enabled<br>1h = Reset function of NRST pin is disabled |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.44 SYSSTATUSCLR (Offset = 1324h) [Reset = 00XXXXXXh]

SYSSTATUSCLR is shown in Figure 2-54 and described in Table 2-64.

Return to the Summary Table.

Clear sticky bits of SYSSTATUS

**Figure 2-54. SYSSTATUSCLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | ALLECC |
| R-0h | | | | | | | W-0h |

**Table 2-64. SYSSTATUSCLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | The key value CEh (206) must be written to KEY together with ALLECC to clear the ECC state.<br>CEh = Issue command |
| 23-1 | RESERVED | R | 0h | |
| 0 | ALLECC | W | 0h | Set ALLECC to clear all ECC related SYSSTATUS indicators.<br>1h = Clear ECC error state |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.45 SWDCFG (Offset = 1328h) [Reset = 00XXXXXXh]

SWDCFG is shown in Figure 2-55 and described in Table 2-65.

Return to the Summary Table.

Disable the SWD function on the SWD pins

**Figure 2-55. SWDCFG**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | DISABLE |
| R-0h | | | | | | | W-0h |

**Table 2-65. SWDCFG Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | The key value 62h (98) must be written to KEY together with DISBALE to disable the SWD functions.<br>62h = Issue command |
| 23-1 | RESERVED | R | 0h | |
| 0 | DISABLE | W | 0h | Set DISABLE to disable the SWD function on SWD pins, allowing the SWD pins to be used as GPIO.<br>1h = Disable SWD function on SWD pins |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.46 FCCCMD (Offset = 132Ch) [Reset = 00XXXXXXh]

FCCCMD is shown in Figure 2-56 and described in Table 2-66.

Return to the Summary Table.

Frequency clock counter start capture

**Figure 2-56. FCCCMD**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| KEY | | | | | | | | RESERVED | | | | | | | |
| W-0h | | | | | | | | R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | | | | GO |
| R-0h | | | | | | | | | | | | | | | W-0h |

**Table 2-66. FCCCMD Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | The key value 0Eh (14) must be written with GO to start a capture.<br>0Eh = Issue command |
| 23-1 | RESERVED | R | 0h | |
| 0 | GO | W | 0h | Set GO to start a capture with the frequency clock counter (FCC).<br>1h = 1 |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.47 SHUTDNSTORE0 (Offset = 1400h) [Reset = 00000000h]

SHUTDNSTORE0 is shown in Figure 2-57 and described in Table 2-67.

Return to the Summary Table.

Shutdown storage memory (byte 0)

**Figure 2-57. SHUTDNSTORE0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DATA | | | | | | | |
| R/W-0h | | | | | | | |

**Table 2-67. SHUTDNSTORE0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-10 | RESERVED | R | 0h | |
| 7-0 | DATA | R/W | 0h | Shutdown storage byte 0 |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.48 SHUTDNSTORE1 (Offset = 1404h) [Reset = 00000000h]

SHUTDNSTORE1 is shown in Figure 2-58 and described in Table 2-68.

Return to the Summary Table.

Shutdown storage memory (byte 1)

**Figure 2-58. SHUTDNSTORE1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DATA | | | | | | | |
| R/W-0h | | | | | | | |

**Table 2-68. SHUTDNSTORE1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-10 | RESERVED | R | 0h | |
| 7-0 | DATA | R/W | 0h | Shutdown storage byte 1 |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.49 SHUTDNSTORE2 (Offset = 1408h) [Reset = 00000000h]

SHUTDNSTORE2 is shown in Figure 2-59 and described in Table 2-69.

Return to the Summary Table.

Shutdown storage memory (byte 2)

**Figure 2-59. SHUTDNSTORE2**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DATA | | | | | | | |
| R/W-0h | | | | | | | |

**Table 2-69. SHUTDNSTORE2 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-10 | RESERVED | R | 0h | |
| 7-0 | DATA | R/W | 0h | Shutdown storage byte 2 |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.50 SHUTDNSTORE3 (Offset = 140Ch) [Reset = 00000000h]

SHUTDNSTORE3 is shown in Figure 2-60 and described in Table 2-70.

Return to the Summary Table.

Shutdown storage memory (byte 3)

**Figure 2-60. SHUTDNSTORE3**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DATA | | | | | | | |
| R/W-0h | | | | | | | |

**Table 2-70. SHUTDNSTORE3 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-10 | RESERVED | R | 0h | |
| 7-0 | DATA | R/W | 0h | Shutdown storage byte 3 |



Copyright © 2026 Texas Instruments Incorporated

### 2.5.51 IDREG (Offset = 2040h) [Reset = 00000000h]

IDREG is shown in Figure 2-61 and described in Table 2-71.

Return to the Summary Table.

JTAG USERCODE

**Figure 2-61. IDREG**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 2-71. IDREG Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | R | 0h | ID Register |



Copyright © 2026 Texas Instruments Incorporated

## 2.6 SYSCTL_G351x_G151x Registers

Table 2-72 lists the memory-mapped registers for the SYSCTL_G351x_G151x registers. All register offset addresses not listed in Table 2-72 should be considered as reserved locations and the register contents should not be modified.

**Table 2-72. SYSCTL_G351X_G151X Registers**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 1020h | IIDX | SYSCTL interrupt index | | Go |
| 1028h | IMASK | SYSCTL interrupt mask | | Go |
| 1030h | RIS | SYSCTL raw interrupt status | | Go |
| 1038h | MIS | SYSCTL masked interrupt status | | Go |
| 1040h | ISET | SYSCTL interrupt set | | Go |
| 1048h | ICLR | SYSCTL interrupt clear | | Go |
| 1050h | NMIIIDX | NMI interrupt index | | Go |
| 1060h | NMIRIS | NMI raw interrupt status | | Go |
| 1070h | NMIISET | NMI interrupt set | | Go |
| 1078h | NMIICLR | NMI interrupt clear | | Go |
| 1100h | SYSOSCCFG | SYSOSC configuration | | Go |
| 1104h | MCLKCFG | Main clock (MCLK) configuration | | Go |
| 1108h | HSCLKEN | High-speed clock (HSCLK) source enable/disable | | Go |
| 110Ch | HSCLKCFG | High-speed clock (HSCLK) source selection | | Go |
| 1110h | HFCLKCLKCFG | High-frequency clock (HFCLK) configuration | | Go |
| 1114h | LFCLKCFG | Low frequency crystal oscillator (LFXT) configuration | | Go |
| 1120h | SYSPLLCFG0 | SYSPLL reference and output configuration | | Go |
| 1124h | SYSPLLCFG1 | SYSPLL reference and feedback divider | | Go |
| 1128h | SYSPLLPARAM0 | SYSPLL PARAM0 (load from FACTORY region) | | Go |
| 112Ch | SYSPLLPARAM1 | SYSPLL PARAM1 (load from FACTORY region) | | Go |
| 1138h | GENCLKCFG | General clock configuration | | Go |
| 113Ch | GENCLKEN | General clock enable control | | Go |
| 1140h | PMODECFG | Power mode configuration | | Go |
| 1150h | FCC | Frequency clock counter (FCC) count | | Go |
| 1170h | SYSOSCTRIMUSER | SYSOSC user-specified trim | | Go |
| 1178h | SRAMBOUNDARY | SRAM Write Boundary | | Go |
| 117Ch | SRAMBOUNDARYHIGH | SRAM Write Boundary High | | Go |
| 1180h | SYSTEMCFG | System configuration | | Go |
| 1184h | SRAMCFG | System SRAM configuration | | Go |
| 1200h | WRITELOCK | SYSCTL register write lockout | | Go |
| 1204h | CLKSTATUS | Clock module (CKM) status | | Go |
| 1208h | SYSSTATUS | System status information | | Go |
| 120Ch | DEDERRADDR | Memory DED Address | | Go |
| 1220h | RSTCAUSE | Reset cause | | Go |
| 1300h | RESETLEVEL | Reset level for application-triggered reset command | | Go |



Copyright © 2026 Texas Instruments Incorporated

**Table 2-72. SYSCTL_G351X_G151X Registers (continued)**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 1304h | RESETCMD | Execute an application-triggered reset command | | Go |
| 1308h | BORTHRESHOLD | BOR threshold selection | | Go |
| 130Ch | BORCLRCMD | Set the BOR threshold | | Go |
| 1310h | SYSOSCFCLCTL | SYSOSC frequency correction loop (FCL) ROSC enable | | Go |
| 1314h | LFXTCTL | LFXT and LFCLK control | | Go |
| 1318h | EXLFCTL | LFCLK_IN and LFCLK control | | Go |
| 131Ch | SHDNIOREL | SHUTDOWN IO release control | | Go |
| 1320h | EXRSTPIN | Disable the reset function of the NRST pin | | Go |
| 1324h | SYSSTATUSCLR | Clear sticky bits of SYSSTATUS | | Go |
| 1328h | SWDCFG | Disable the SWD function on the SWD pins | | Go |
| 132Ch | FCCCMD | Frequency clock counter start capture | | Go |
| 1400h | SHUTDNSTORE0 | Shutdown storage memory (byte 0) | | Go |
| 1404h | SHUTDNSTORE1 | Shutdown storage memory (byte 1) | | Go |
| 1408h | SHUTDNSTORE2 | Shutdown storage memory (byte 2) | | Go |
| 140Ch | SHUTDNSTORE3 | Shutdown storage memory (byte 3) | | Go |
| 2040h | IDREG | ID Register | | Go |

Complex bit access types are encoded to fit into small table cells. Table 2-73 shows the codes that are used for access types in this section.

**Table 2-73. SYSCTL_G351x_G151x Access Type Codes**

| Access Type | Code | Description |
|---|---|---|
| Read Type | | |
| R | R | Read |
| RC | R<br>C | Read<br>to Clear |
| Write Type | | |
| W | W | Write |
| W1C | W<br>1C | Write<br>1 to clear |
| W1S | W<br>1S | Write<br>1 to set |
| Reset or Default Value | | |
| *-n* | | Value after reset or the default value |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.1 IIDX (Offset = 1020h) [Reset = 00000000h]

IIDX is shown in Figure 2-62 and described in Table 2-74.

Return to the Summary Table.

SYSCTL interrupt index

**Figure 2-62. IIDX**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | STAT | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | |

**Table 2-74. IIDX Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3-0 | STAT | R | 0h | The SYSCTL interrupt index (IIDX) register generates a value corresponding to the highest priority pending interrupt source. This value may be used as an address offset for fast, deterministic handling in the interrupt service routine. A read of the IIDX register will clear the corresponding interrupt status in the RIS and MIS registers.<br>0h = No interrupt pending<br>1h = LFOSCGOOD interrupt pending<br>2h = 2<br>3h = 3<br>4h = 4<br>5h = 5<br>6h = 6<br>7h = 7<br>8h = 8 |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.2 IMASK (Offset = 1028h) [Reset = 00000000h]

IMASK is shown in Figure 2-63 and described in Table 2-75.

Return to the Summary Table.

SYSCTL interrupt mask

**Figure 2-63. IMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| HSCLKGOOD | SYSPLLGOOD | HFCLKGOOD | LFXTGOOD | SRAMSEC | FLASHSEC | ANACLKERR | LFOSCGOOD |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

**Table 2-75. IMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7 | HSCLKGOOD | R/W | 0h | HSCLK GOOD<br>0h = 0<br>1h = 1 |
| 6 | SYSPLLGOOD | R/W | 0h | SYSPLL GOOD<br>0h = 0<br>1h = 1 |
| 5 | HFCLKGOOD | R/W | 0h | HFCLK GOOD<br>0h = 0<br>1h = 1 |
| 4 | LFXTGOOD | R/W | 0h | LFXT GOOD<br>0h = 0<br>1h = 1 |
| 3 | SRAMSEC | R/W | 0h | SRAM Single Error Correct<br>0h = 0<br>1h = 1 |
| 2 | FLASHSEC | R/W | 0h | Flash Single Error Correct<br>0h = 0<br>1h = 1 |
| 1 | ANACLKERR | R/W | 0h | Analog Clocking Consistency Error<br>0h = 0<br>1h = 1 |
| 0 | LFOSCGOOD | R/W | 0h | Enable or disable the LFOSCGOOD interrupt. LFOSCGOOD indicates that the LFOSC has started successfully.<br>0h = Interrupt disabled<br>1h = Interrupt enabled |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.3 RIS (Offset = 1030h) [Reset = 00000000h]

RIS is shown in Figure 2-64 and described in Table 2-76.

Return to the Summary Table.

SYSCTL raw interrupt status

**Figure 2-64. RIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| HSCLKGOOD | SYSPLLGOOD | HFCLKGOOD | LFXTGOOD | SRAMSEC | FLASHSEC | ANACLKERR | LFOSCGOOD |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 2-76. RIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7 | HSCLKGOOD | R | 0h | HSCLK GOOD<br>0h = 0<br>1h = 1 |
| 6 | SYSPLLGOOD | R | 0h | SYSPLL GOOD<br>0h = 0<br>1h = 1 |
| 5 | HFCLKGOOD | R | 0h | HFCLK GOOD<br>0h = 0<br>1h = 1 |
| 4 | LFXTGOOD | R | 0h | LFXT GOOD<br>0h = 0<br>1h = 1 |
| 3 | SRAMSEC | R | 0h | SRAM Single Error Correct<br>0h = 0<br>1h = 1 |
| 2 | FLASHSEC | R | 0h | Flash Single Error Correct<br>0h = 0<br>1h = 1 |
| 1 | ANACLKERR | R | 0h | Analog Clocking Consistency Error<br>0h = 0<br>1h = 1 |
| 0 | LFOSCGOOD | R | 0h | Raw status of the LFOSCGOOD interrupt.<br>0h = No interrupt pending<br>1h = Interrupt pending |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.4 MIS (Offset = 1038h) [Reset = 00000000h]

MIS is shown in Figure 2-65 and described in Table 2-77.

Return to the Summary Table.

SYSCTL masked interrupt status

**Figure 2-65. MIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| HSCLKGOOD | SYSPLLGOOD | HFCLKGOOD | LFXTGOOD | SRAMSEC | FLASHSEC | ANACLKERR | LFOSCGOOD |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 2-77. MIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7 | HSCLKGOOD | R | 0h | HSCLK GOOD<br>0h = 0<br>1h = 1 |
| 6 | SYSPLLGOOD | R | 0h | SYSPLL GOOD<br>0h = 0<br>1h = 1 |
| 5 | HFCLKGOOD | R | 0h | HFCLK GOOD<br>0h = 0<br>1h = 1 |
| 4 | LFXTGOOD | R | 0h | LFXT GOOD<br>0h = 0<br>1h = 1 |
| 3 | SRAMSEC | R | 0h | SRAM Single Error Correct<br>0h = 0<br>1h = 1 |
| 2 | FLASHSEC | R | 0h | Flash Single Error Correct<br>0h = 0<br>1h = 1 |
| 1 | ANACLKERR | R | 0h | Analog Clocking Consistency Error<br>0h = 0<br>1h = 1 |
| 0 | LFOSCGOOD | R | 0h | Masked status of the LFOSCGOOD interrupt.<br>0h = No interrupt pending<br>1h = Interrupt pending |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.5 ISET (Offset = 1040h) [Reset = 00000000h]

ISET is shown in Figure 2-66 and described in Table 2-78.

Return to the Summary Table.

SYSCTL interrupt set

**Figure 2-66. ISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| HSCLKGOOD | SYSPLLGOOD | HFCLKGOOD | LFXTGOOD | SRAMSEC | FLASHSEC | ANACLKERR | LFOSCGOOD |
| W1S-0h | W1S-0h | W1S-0h | W1S-0h | W1S-0h | W1S-0h | W1S-0h | W1S-0h |

**Table 2-78. ISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7 | HSCLKGOOD | W1S | 0h | HSCLK GOOD<br>0h = 0<br>1h = 1 |
| 6 | SYSPLLGOOD | W1S | 0h | SYSPLL GOOD<br>0h = 0<br>1h = 1 |
| 5 | HFCLKGOOD | W1S | 0h | HFCLK GOOD<br>0h = 0<br>1h = 1 |
| 4 | LFXTGOOD | W1S | 0h | LFXT GOOD<br>0h = 0<br>1h = 1 |
| 3 | SRAMSEC | W1S | 0h | SRAM Single Error Correct<br>0h = 0<br>1h = 1 |
| 2 | FLASHSEC | W1S | 0h | Flash Single Error Correct<br>0h = 0<br>1h = 1 |
| 1 | ANACLKERR | W1S | 0h | Analog Clocking Consistency Error<br>0h = 0<br>1h = 1 |
| 0 | LFOSCGOOD | W1S | 0h | Set the LFOSCGOOD interrupt.<br>0h = Writing 0h hs no effect<br>1h = Set interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.6 ICLR (Offset = 1048h) [Reset = 00000000h]

ICLR is shown in Figure 2-67 and described in Table 2-79.

Return to the Summary Table.

SYSCTL interrupt clear

**Figure 2-67. ICLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| HSCLKGOOD | SYSPLLGOOD | HFCLKGOOD | LFXTGOOD | SRAMSEC | FLASHSEC | ANACLKERR | LFOSCGOOD |
| W1C-0h | W1C-0h | W1C-0h | W1C-0h | W1C-0h | W1C-0h | W1C-0h | W1C-0h |

**Table 2-79. ICLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7 | HSCLKGOOD | W1C | 0h | HSCLK GOOD<br>0h = 0<br>1h = 1 |
| 6 | SYSPLLGOOD | W1C | 0h | SYSPLL GOOD<br>0h = 0<br>1h = 1 |
| 5 | HFCLKGOOD | W1C | 0h | HFCLK GOOD<br>0h = 0<br>1h = 1 |
| 4 | LFXTGOOD | W1C | 0h | LFXT GOOD<br>0h = 0<br>1h = 1 |
| 3 | SRAMSEC | W1C | 0h | SRAM Single Error Correct<br>0h = 0<br>1h = 1 |
| 2 | FLASHSEC | W1C | 0h | Flash Single Error Correct<br>0h = 0<br>1h = 1 |
| 1 | ANACLKERR | W1C | 0h | Analog Clocking Consistency Error<br>0h = 0<br>1h = 1 |
| 0 | LFOSCGOOD | W1C | 0h | Clear the LFOSCGOOD interrupt.<br>0h = Writing 0h has no effect<br>1h = Clear interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.7 NMIIIDX (Offset = 1050h) [Reset = 00000000h]

NMIIIDX is shown in Figure 2-68 and described in Table 2-80.

Return to the Summary Table.

NMI interrupt index

**Figure 2-68. NMIIIDX**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | STAT | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | |

**Table 2-80. NMIIIDX Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3-0 | STAT | R | 0h | The NMI interrupt index (NMIIIDX) register generates a value corresponding to the highest priority pending NMI source. This value may be used as an address offset for fast, deterministic handling in the NMI service routine. A read of the NMIIIDX register will clear the corresponding interrupt status in the NMIRIS register.<br>0h = No NMI pending<br>1h = BOR Threshold NMI pending<br>2h = 2<br>3h = 3<br>4h = 4<br>5h = 5<br>6h = 6 |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.8 NMIRIS (Offset = 1060h) [Reset = 00000000h]

NMIRIS is shown in Figure 2-69 and described in Table 2-81.

Return to the Summary Table.

NMI raw interrupt status

**Figure 2-69. NMIRIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | SRAMDED | FLASHDED | LFCLKFAIL | WWDT1 | WWDT0 | BORLVL |
| R-0h | | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 2-81. NMIRIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | RESERVED | R | 0h | |
| 5 | SRAMDED | R | 0h | SRAM Double Error Detect<br>0h = 0<br>1h = 1 |
| 4 | FLASHDED | R | 0h | Flash Double Error Detect<br>0h = 0<br>1h = 1 |
| 3 | LFCLKFAIL | R | 0h | LFXT-EXLF Monitor Fail<br>0h = 0<br>1h = 1 |
| 2 | WWDT1 | R | 0h | Watch Dog 0 Fault<br>0h = 0<br>1h = 1 |
| 1 | WWDT0 | R | 0h | Watch Dog 0 Fault<br>0h = 0<br>1h = 1 |
| 0 | BORLVL | R | 0h | Raw status of the BORLVL NMI<br>0h = No interrupt pending<br>1h = Interrupt pending |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.9 NMIISET (Offset = 1070h) [Reset = 00000000h]

NMIISET is shown in Figure 2-70 and described in Table 2-82.

Return to the Summary Table.

NMI interrupt set

**Figure 2-70. NMIISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | SRAMDED | FLASHDED | LFCLKFAIL | WWDT1 | WWDT0 | BORLVL |
| R-0h | | W1S-0h | W1S-0h | W1S-0h | W1S-0h | W1S-0h | W1S-0h |

**Table 2-82. NMIISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | RESERVED | R | 0h | |
| 5 | SRAMDED | W1S | 0h | SRAM Double Error Detect<br>0h = 0<br>1h = 1 |
| 4 | FLASHDED | W1S | 0h | Flash Double Error Detect<br>0h = 0<br>1h = 1 |
| 3 | LFCLKFAIL | W1S | 0h | LFXT-EXLF Monitor Fail<br>0h = 0<br>1h = 1 |
| 2 | WWDT1 | W1S | 0h | Watch Dog 0 Fault<br>0h = 0<br>1h = 1 |
| 1 | WWDT0 | W1S | 0h | Watch Dog 0 Fault<br>0h = 0<br>1h = 1 |
| 0 | BORLVL | W1S | 0h | Set the BORLVL NMI<br>0h = Writing 0h hs no effect<br>1h = Set interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.10 NMIICLR (Offset = 1078h) [Reset = 00000000h]

NMIICLR is shown in Figure 2-71 and described in Table 2-83.

Return to the Summary Table.

NMI interrupt clear

**Figure 2-71. NMIICLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | SRAMDED | FLASHDED | LFCLKFAIL | WWDT1 | WWDT0 | BORLVL |
| R-0h | | W1C-0h | W1C-0h | W1C-0h | W1C-0h | W1C-0h | W1C-0h |

**Table 2-83. NMIICLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | RESERVED | R | 0h | |
| 5 | SRAMDED | W1C | 0h | SRAM Double Error Detect<br>0h = 0<br>1h = 1 |
| 4 | FLASHDED | W1C | 0h | Flash Double Error Detect<br>0h = 0<br>1h = 1 |
| 3 | LFCLKFAIL | W1C | 0h | LFXT-EXLF Monitor Fail<br>0h = 0<br>1h = 1 |
| 2 | WWDT1 | W1C | 0h | Watch Dog 0 Fault<br>0h = 0<br>1h = 1 |
| 1 | WWDT0 | W1C | 0h | Watch Dog 0 Fault<br>0h = 0<br>1h = 1 |
| 0 | BORLVL | W1C | 0h | Clr the BORLVL NMI<br>0h = Writing 0h hs no effect<br>1h = Clear interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.11 SYSOSCCFG (Offset = 1100h) [Reset = 0002XXXXh]

SYSOSCCFG is shown in Figure 2-72 and described in Table 2-84.

Return to the Summary Table.

SYSOSC configuration

**Figure 2-72. SYSOSCCFG**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | FASTCPUEVENT | BLOCKASYNCALL |
| R-0h | | | | | | R/W-1h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | DISABLE | DISABLESTOP | USE4MHZSTOP |
| R-0h | | | | | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | FREQ | |
| R-0h | | | | | | R/W-0h | |

**Table 2-84. SYSOSCCFG Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-18 | RESERVED | R | 0h | |
| 17 | FASTCPUEVENT | R/W | 1h | FASTCPUEVENT may be used to assert a fast clock request when an interrupt is asserted to the CPU, reducing interrupt latency.<br>0h = An interrupt to the CPU will not assert a fast clock request<br>1h = An interrupt to the CPU will assert a fast clock request |
| 16 | BLOCKASYNCALL | R/W | 0h | BLOCKASYNCALL may be used to mask block all asynchronous fast clock requests, preventing hardware from dynamically changing the active clock configuration when operating in a given mode.<br>0h = Asynchronous fast clock requests are controlled by the requesting peripheral<br>1h = All asynchronous fast clock requests are blocked |
| 15-11 | RESERVED | R | 0h | |
| 10 | DISABLE | R/W | 0h | DISABLE sets the SYSOSC enable/disable policy. SYSOSC may be powered off in RUN, SLEEP, and STOP modes to reduce power consumption. When SYSOSC is disabled, MCLK and ULPCLK are sourced from LFCLK.<br>0h = Do not disable SYSOSC<br>1h = Disable SYSOSC immediately and source MCLK and ULPCLK from LFCLK |
| 9 | DISABLESTOP | R/W | 0h | DISABLESTOP sets the SYSOSC stop mode enable/disable policy. When operating in STOP mode, the SYSOSC may be automatically disabled. When set, ULPCLK will run from LFCLK in STOP mode and SYSOSC will be disabled to reduce power consumption.<br>0h = Do not disable SYSOSC in STOP mode<br>1h = Disable SYSOSC in STOP mode and source ULPCLK from LFCLK |
| 8 | USE4MHZSTOP | R/W | 0h | USE4MHZSTOP sets the SYSOSC stop mode frequency policy. When entering STOP mode, the SYSOSC frequency may be automatically switched to 4MHz to reduce SYSOSC power consumption.<br>0h = Do not gear shift the SYSOSC to 4MHz in STOP mode<br>1h = Gear shift SYSOSC to 4MHz in STOP mode |



Copyright © 2026 Texas Instruments Incorporated

**Table 2-84. SYSOSCCFG Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7-2 | RESERVED | R | 0h | |
| 1-0 | FREQ | R/W | 0h | Target operating frequency for the system oscillator (SYSOSC)<br>0h = Base frequency (32MHz)<br>1h = Low frequency (4MHz)<br>2h = User-trimmed frequency (16 or 24 MHz) |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.12 MCLKCFG (Offset = 1104h) [Reset = 000XX2X0h]

MCLKCFG is shown in Figure 2-73 and described in Table 2-85.

Return to the Summary Table.

Main clock (MCLK) configuration

**Figure 2-73. MCLKCFG**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | MCLKDEADCHK | STOPCLKSTBY | USELFCLK | RESERVED | | | USEHSCLK |
| R-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | | | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | USEMFTICK | FLASHWAIT | | | |
| R-0h | | | R/W-0h | R/W-2h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | UDIV | | MDIV | | | |
| R-0h | | R/W-1h | | R/W-0h | | | |

**Table 2-85. MCLKCFG Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-23 | RESERVED | R | 0h | |
| 22 | MCLKDEADCHK | R/W | 0h | MCLKDEADCHK enables or disables the continuous MCLK dead check monitor. LFCLK must be running before MCLKDEADCHK is enabled.<br>0h = The MCLK dead check monitor is disabled<br>1h = The MCLK dead check monitor is enabled |
| 21 | STOPCLKSTBY | R/W | 0h | STOPCLKSTBY sets the STANDBY mode policy (STANDBY0 or STANDBY1). When set, ULPCLK and LFCLK are disabled to all peripherals in STANDBY mode, with the exception of TIMG0 and TIMG1 which continue to run. Wake-up is only possible via an asynchronous fast clock request.<br>0h = ULPCLK/LFCLK runs to all PD0 peripherals in STANDBY mode<br>1h = ULPCLK/LFCLK is disabled to all peripherals in STANDBY mode except TIMG0 and TIMG1 |
| 20 | USELFCLK | R/W | 0h | USELFCLK sets the MCLK source policy. Set USELFCLK to use LFCLK as the MCLK source. Note that setting USELFCLK does not disable SYSOSC, and SYSOSC remains available for direct use by analog modules.<br>0h = MCLK will not use the low frequency clock (LFCLK)<br>1h = MCLK will use the low frequency clock (LFCLK) |
| 19-17 | RESERVED | R | 0h | |
| 16 | USEHSCLK | R/W | 0h | USEHSCLK, together with USELFCLK, sets the MCLK source policy. Set USEHSCLK to use HSCLK (HFCLK or SYSPLL) as the MCLK source in RUN and SLEEP modes.<br>0h = MCLK will not use the high speed clock (HSCLK)<br>1h = MCLK will use the high speed clock (HSCLK) in RUN and SLEEP mode |
| 15-13 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

**Table 2-85. MCLKCFG Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 12 | USEMFTICK | R/W | 0h | USEMFTICK specifies whether the 4MHz constant-rate clock (MFCLK) to peripherals is enabled or disabled. When enabled, MDIV must be disabled (set to 0h=/1).<br>0h = The 4MHz rate MFCLK to peripherals is enabled<br>1h = The 4MHz rate MFCLK to peripherals is enabled. |
| 11-8 | FLASHWAIT | R/W | 2h | FLASHWAIT specifies the number of flash wait states when MCLK is sourced from HSCLK. FLASHWAIT has no effect when MCLK is sourced from SYSOSC or LFCLK.<br>0h = No flash wait states are applied<br>1h = One flash wait state is applied<br>2h = 2 flash wait states are applied |
| 7-6 | RESERVED | R | 0h | |
| 5-4 | UDIV | R/W | 1h | UDIV specifies the ULPCLK divider when MCLK is sourced from HSCLK. UDIV has no effect when MCLK is sourced from SYSOSC or LFCLK.<br>0h = ULPCLK is not divided and is equal to MCLK<br>1h = ULPCLK is MCLK/2 (divided-by-2) |
| 3-0 | MDIV | R/W | 0h | MDIV may be used to divide the MCLK frequency when MCLK is sourced from SYSOSC. MDIV=0h corresponds to /1 (no divider). MDIV=1h corresponds to /2 (divide-by-2). MDIV=Fh corresponds to /16 (divide-by-16). MDIV may be set between /1 and /16 on an integer basis. |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.13 HSCLKEN (Offset = 1108h) [Reset = 0000XXXXh]

HSCLKEN is shown in Figure 2-74 and described in Table 2-86.

Return to the Summary Table.

High-speed clock (HSCLK) source enable/disable

**Figure 2-74. HSCLKEN**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | USEEXTHFCLK |
| R-0h | | | | | | | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | SYSPLLEN |
| R-0h | | | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | HFXTEN |
| R-0h | | | | | | | R/W-0h |

**Table 2-86. HSCLKEN Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-17 | RESERVED | R | 0h | |
| 16 | USEEXTHFCLK | R/W | 0h | USEEXTHFCLK selects the HFCLK_IN digital clock input to be the source for HFCLK. When disabled, HFXT is the HFCLK source and HFXTEN may be set. Do not set HFXTEN and USEEXTHFCLK simultaneously.<br>0h = Use HFXT as the HFCLK source<br>1h = Use the HFCLK_IN digital clock input as the HFCLK source |
| 15-9 | RESERVED | R | 0h | |
| 8 | SYSPLLEN | R/W | 0h | SYSPLLEN enables or disables the system phase-lock loop (SYSPLL).<br>0h = Disable the SYSPLL<br>1h = Enable the SYSPLL |
| 7-1 | RESERVED | R | 0h | |
| 0 | HFXTEN | R/W | 0h | HFXTEN enables or disables the high frequency crystal oscillator (HFXT).<br>0h = Disable the HFXT<br>1h = Enable the HFXT |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.14 HSCLKCFG (Offset = 110Ch) [Reset = 00000000h]

HSCLKCFG is shown in Figure 2-75 and described in Table 2-87.

Return to the Summary Table.

High-speed clock (HSCLK) source selection

**Figure 2-75. HSCLKCFG**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | HSCLKSEL |
| R-0h | | | | | | | R/W-0h |

**Table 2-87. HSCLKCFG Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | HSCLKSEL | R/W | 0h | HSCLKSEL selects the HSCLK source (SYSPLL or HFCLK).<br>0h = HSCLK is sourced from the SYSPLL<br>1h = HSCLK is sourced from the HFCLK |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.15 HFCLKCLKCFG (Offset = 1110h) [Reset = 1XXXXX00h]

HFCLKCLKCFG is shown in Figure 2-76 and described in Table 2-88.

Return to the Summary Table.

High-frequency clock (HFCLK) configuration

**Figure 2-76. HFCLKCLKCFG**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | HFCLKFLTCHK | RESERVED | | | |
| R-0h | | | R/W-1h | R-0h | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | HFXTRSEL | | RESERVED | | | |
| R-0h | | R/W-0h | | R-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| HFXTTIME | | | | | | | |
| R/W-0h | | | | | | | |

**Table 2-88. HFCLKCLKCFG Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | RESERVED | R | 0h | |
| 28 | HFCLKFLTCHK | R/W | 1h | HFCLKFLTCHK enables or disables the HFCLK startup monitor.<br>0h = HFCLK startup is not checked<br>1h = HFCLK startup is checked |
| 27-14 | RESERVED | R | 0h | |
| 13-12 | HFXTRSEL | R/W | 0h | HFXT Range Select<br>0h = 4MHz <= HFXT frequency <= 8MHz<br>1h = 8MHz < HFXT frequency <= 16MHz<br>2h = 16MHz < HFXT frequency <= 32MHz<br>3h = 32MHz < HFXT frequency <= 48MHz |
| 11-8 | RESERVED | R | 0h | |
| 7-0 | HFXTTIME | R/W | 0h | HFXTTIME specifies the HFXT startup time in 64us resolution. If the HFCLK startup monitor is enabled (HFCLKFLTCHK), HFXT will be checked after this time expires.<br>0h = Minimum startup time (approximatly zero)<br>FFh = Maximum startup time (approximatly 16.32ms) |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.16 LFCLKCFG (Offset = 1114h) [Reset = 000000XXh]

LFCLKCFG is shown in Figure 2-77 and described in Table 2-89.

Return to the Summary Table.

Low frequency crystal oscillator (LFXT) configuration

**Figure 2-77. LFCLKCFG**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | LOWCAP |
| R-0h | | | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | MONITOR | RESERVED | | XT1DRIVE | |
| R-0h | | | R/W-0h | R-0h | | R/W-3h | |

**Table 2-89. LFCLKCFG Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-9 | RESERVED | R | 0h | |
| 8 | LOWCAP | R/W | 0h | LOWCAP controls the low-power LFXT mode. When the LFXT load capacitance is less than 3pf, LOWCAP may be set for reduced power consumption.<br>0h = LFXT low capacitance mode is disabled<br>1h = LFXT low capacitance mode is enabled |
| 7-5 | RESERVED | R | 0h | |
| 4 | MONITOR | R/W | 0h | MONITOR enables or disables the LFCLK monitor, which continuously checks LFXT or LFCLK_IN for a clock stuck fault.<br>0h = Clock monitor is disabled<br>1h = Clock monitor is enabled |
| 3-2 | RESERVED | R | 0h | |
| 1-0 | XT1DRIVE | R/W | 3h | XT1DRIVE selects the low frequency crystal oscillator (LFXT) drive strength.<br>0h = Lowest drive and current<br>1h = Lower drive and current<br>2h = Higher drive and current<br>3h = Highest drive and current |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.17 SYSPLLCFG0 (Offset = 1120h) [Reset = 000000XXh]

SYSPLLCFG0 is shown in Figure 2-78 and described in Table 2-90.

Return to the Summary Table.

SYSPLL reference and output configuration

**Figure 2-78. SYSPLLCFG0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | RDIVCLK2X | | | |
| R-0h | | | | R/W-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RDIVCLK1 | | | | RDIVCLK0 | | | |
| R/W-0h | | | | R/W-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | ENABLECLK2X | ENABLECLK1 | ENABLECLK0 | RESERVED | | MCLK2XVCO | SYSPLLREF |
| R-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | | R/W-0h | R/W-0h |

**Table 2-90. SYSPLLCFG0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | RESERVED | R | 0h | |
| 19-16 | RDIVCLK2X | R/W | 0h | RDIVCLK2X sets the final divider for the SYSPLLCLK2X output.<br>0h = SYSPLLCLK1 is divided by 1<br>1h = SYSPLLCLK1 is divided by 2<br>2h = SYSPLLCLK1 is divided by 3<br>3h = SYSPLLCLK1 is divided by 4<br>4h = SYSPLLCLK1 is divided by 5<br>5h = SYSPLLCLK1 is divided by 6<br>6h = SYSPLLCLK1 is divided by 7<br>7h = SYSPLLCLK1 is divided by 8<br>8h = SYSPLLCLK1 is divided by 9<br>9h = SYSPLLCLK1 is divided by 10<br>Ah = SYSPLLCLK1 is divided by 11<br>Bh = SYSPLLCLK1 is divided by 12<br>Ch = SYSPLLCLK1 is divided by 13<br>Dh = SYSPLLCLK1 is divided by 14<br>Eh = SYSPLLCLK1 is divided by 15<br>Fh = SYSPLLCLK1 is divided by 16 |
| 15-12 | RDIVCLK1 | R/W | 0h | RDIVCLK1 sets the final divider for the SYSPLLCLK1 output.<br>0h = SYSPLLCLK1 is divided by 2<br>1h = SYSPLLCLK1 is divided by 4<br>2h = SYSPLLCLK1 is divided by 6<br>3h = SYSPLLCLK1 is divided by 8<br>4h = SYSPLLCLK1 is divided by 10<br>5h = SYSPLLCLK1 is divided by 12<br>6h = SYSPLLCLK1 is divided by 14<br>7h = SYSPLLCLK1 is divided by 16<br>8h = SYSPLLCLK1 is divided by 18<br>9h = SYSPLLCLK1 is divided by 20<br>Ah = SYSPLLCLK1 is divided by 22<br>Bh = SYSPLLCLK1 is divided by 24<br>Ch = SYSPLLCLK1 is divided by 26<br>Dh = SYSPLLCLK1 is divided by 28<br>Eh = SYSPLLCLK1 is divided by 30<br>Fh = SYSPLLCLK1 is divided by 32 |



Copyright © 2026 Texas Instruments Incorporated

**Table 2-90. SYSPLLCFG0 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 11-8 | RDIVCLK0 | R/W | 0h | RDIVCLK0 sets the final divider for the SYSPLLCLK0 output.<br>0h = SYSPLLCLK0 is divided by 2<br>1h = SYSPLLCLK0 is divided by 4<br>2h = SYSPLLCLK0 is divided by 6<br>3h = SYSPLLCLK0 is divided by 8<br>4h = SYSPLLCLK0 is divided by 10<br>5h = SYSPLLCLK0 is divided by 12<br>6h = SYSPLLCLK0 is divided by 14<br>7h = SYSPLLCLK0 is divided by 16<br>8h = SYSPLLCLK0 is divided by 18<br>9h = SYSPLLCLK0 is divided by 20<br>Ah = SYSPLLCLK0 is divided by 22<br>Bh = SYSPLLCLK0 is divided by 24<br>Ch = SYSPLLCLK0 is divided by 26<br>Dh = SYSPLLCLK0 is divided by 28<br>Eh = SYSPLLCLK0 is divided by 30<br>Fh = SYSPLLCLK0 is divided by 32 |
| 7 | RESERVED | R | 0h | |
| 6 | ENABLECLK2X | R/W | 0h | ENABLECLK2X enables or disables the SYSPLLCLK2X output.<br>0h = SYSPLLCLK2X is disabled<br>1h = SYSPLLCLK2X is enabled |
| 5 | ENABLECLK1 | R/W | 0h | ENABLECLK1 enables or disables the SYSPLLCLK1 output.<br>0h = SYSPLLCLK1 is disabled<br>1h = SYSPLLCLK1 is enabled |
| 4 | ENABLECLK0 | R/W | 0h | ENABLECLK0 enables or disables the SYSPLLCLK0 output.<br>0h = SYSPLLCLK0 is disabled<br>1h = SYSPLLCLK0 is enabled |
| 3-2 | RESERVED | R | 0h | |
| 1 | MCLK2XVCO | R/W | 0h | MCLK2XVCO selects the SYSPLL output which is sent to the HSCLK mux for use by MCLK.<br>0h = The SYSPLLCLK0 output is sent to the HSCLK mux<br>1h = The SYSPLLCLK2X output is sent to the HSCLK mux |
| 0 | SYSPLLREF | R/W | 0h | SYSPLLREF selects the system PLL (SYSPLL) reference clock source.<br>0h = SYSPLL reference is SYSOSC<br>1h = SYSPLL reference is HFCLK |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.18 SYSPLLCFG1 (Offset = 1124h) [Reset = 000000XXh]

SYSPLLCFG1 is shown in Figure 2-79 and described in Table 2-91.

Return to the Summary Table.

SYSPLL reference and feedback divider

**Figure 2-79. SYSPLLCFG1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | QDIV | | | | | | | RESERVED | | | | | | PDIV | |
| R-0h | R/W-0h | | | | | | | R-0h | | | | | | R/W-0h | |

**Table 2-91. SYSPLLCFG1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-15 | RESERVED | R | 0h | |
| 14-8 | QDIV | R/W | 0h | QDIV selects the SYSPLL feedback path divider.<br>0h = Divide-by-one is not a valid QDIV option<br>1h = Feedback path is divided by 2<br>7Eh = Feedback path is divided by 127 (0x7E) |
| 7-2 | RESERVED | R | 0h | |
| 1-0 | PDIV | R/W | 0h | PDIV selects the SYSPLL reference clock prescale divider.<br>0h = SYSPLLREF is not divided<br>1h = SYSPLLREF is divided by 2<br>2h = SYSPLLREF is divided by 4<br>3h = SYSPLLREF is divided by 8 |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.19 SYSPLLPARAM0 (Offset = 1128h) [Reset = X0X0X0X0h]

SYSPLLPARAM0 is shown in Figure 2-80 and described in Table 2-92.

Return to the Summary Table.

SYSPLL PARAM0 (load from FACTORY region)

**Figure 2-80. SYSPLLPARAM0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CAPBOVERRIDE | RESERVED | | | CAPBVAL | | | |
| R/W-0h | R-0h | | | R/W-0h | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | CPCURRENT | | | | | |
| R-0h | | R/W-0h | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | STARTTIMELP | | | | | |
| R-0h | | R/W-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | STARTTIME | | | | | |
| R-0h | | R/W-0h | | | | | |

**Table 2-92. SYSPLLPARAM0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | CAPBOVERRIDE | R/W | 0h | CAPBOVERRIDE controls the override for Cap B<br>0h = Cap B override disabled<br>1h = Cap B override enabled |
| 30-29 | RESERVED | R | 0h | |
| 28-24 | CAPBVAL | R/W | 0h | Override value for Cap B |
| 23-22 | RESERVED | R | 0h | |
| 21-16 | CPCURRENT | R/W | 0h | Charge pump current |
| 15-14 | RESERVED | R | 0h | |
| 13-8 | STARTTIMELP | R/W | 0h | Startup time from low power mode exit to locked clock, in 1us resolution |
| 7-6 | RESERVED | R | 0h | |
| 5-0 | STARTTIME | R/W | 0h | Startup time from enable to locked clock, in 1us resolution |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.20 SYSPLLPARAM1 (Offset = 112Ch) [Reset = 00XX00X0h]

SYSPLLPARAM1 is shown in Figure 2-81 and described in Table 2-93.

Return to the Summary Table.

SYSPLL PARAM1 (load from FACTORY region)

**Figure 2-81. SYSPLLPARAM1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LPFRESC | | | | | | | | RESERVED | | | | | | LPFRESA | |
| R/W-0h | | | | | | | | R-0h | | | | | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LPFRESA | | | | | | | | RESERVED | | | LPFCAPA | | | | |
| R/W-0h | | | | | | | | R-0h | | | R/W-0h | | | | |

**Table 2-93. SYSPLLPARAM1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | LPFRESC | R/W | 0h | Loop filter Res C |
| 23-18 | RESERVED | R | 0h | |
| 17-8 | LPFRESA | R/W | 0h | Loop filter Res A |
| 7-5 | RESERVED | R | 0h | |
| 4-0 | LPFCAPA | R/W | 0h | Loop filter Cap A |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.21 GENCLKCFG (Offset = 1138h) [Reset = 00000X0Xh]

GENCLKCFG is shown in Figure 2-82 and described in Table 2-94.

Return to the Summary Table.

General clock configuration

**Figure 2-82. GENCLKCFG**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | FCCTRIGCNT | | | | |
| R-0h | | | R/W-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| ANACPUMPCFG | | FCCLVLTRIG | FCCTRIGSRC | FCCSELCLK | | | |
| R/W-0h | | R/W-0h | R/W-0h | R/W-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| HFCLK4MFPCLKDIV | | | | RESERVED | | MFPCLKSRC | CANCLKSRC |
| R/W-0h | | | | R-0h | | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| EXCLKDIVEN | EXCLKDIVVAL | | | RESERVED | EXCLKSRC | | |
| R/W-0h | R/W-0h | | | R-0h | R/W-0h | | |

**Table 2-94. GENCLKCFG Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | RESERVED | R | 0h | |
| 28-24 | FCCTRIGCNT | R/W | 0h | FCCTRIGCNT specifies the number of trigger clock periods in the trigger window. FCCTRIGCNT=0h (one trigger clock period) up to 1Fh (32 trigger clock periods) may be specified. |
| 23-22 | ANACPUMPCFG | R/W | 0h | ANACPUMPCFG selects the analog mux charge pump (VBOOST) enable method.<br>0h = VBOOST is enabled on request from a COMP, GPAMP, or OPA<br>1h = VBOOST is enabled when the device is in RUN or SLEEP mode, or when a COMP/GPAMP/OPA is enabled<br>2h = VBOOST is always enabled |
| 21 | FCCLVLTRIG | R/W | 0h | FCCLVLTRIG selects the frequency clock counter (FCC) trigger mode.<br>0h = Rising edge to rising edge triggered<br>1h = Level triggered |
| 20 | FCCTRIGSRC | R/W | 0h | FCCTRIGSRC selects the frequency clock counter (FCC) trigger source.<br>0h = FCC trigger is the external pin<br>1h = FCC trigger is the LFCLK |
| 19-16 | FCCSELCLK | R/W | 0h | FCCSELCLK selectes the frequency clock counter (FCC) clock source.<br>0h = FCC clock is MCLK<br>1h = FCC clock is SYSOSC<br>2h = FCC clock is HFCLK<br>3h = FCC clock is the CLK_OUT selection<br>4h = FCC clock is SYSPLLCLK0<br>5h = FCC clock is SYSPLLCLK1<br>6h = FCC clock is SYSPLLCLK2X<br>7h = FCC clock is the FCCIN external input |



Copyright © 2026 Texas Instruments Incorporated

**Table 2-94. GENCLKCFG Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15-12 | HFCLK4MFPCLKDIV | R/W | 0h | HFCLK4MFPCLKDIV selects the divider applied to HFCLK when HFCLK is used as the MFPCLK source. Integer dividers from /1 to /16 may be selected.<br>0h = HFCLK is not divided before being used for MFPCLK<br>1h = HFCLK is divided by 2 before being used for MFPCLK<br>2h = HFCLK is divided by 3 before being used for MFPCLK<br>3h = HFCLK is divided by 4 before being used for MFPCLK<br>4h = HFCLK is divided by 5 before being used for MFPCLK<br>5h = HFCLK is divided by 6 before being used for MFPCLK<br>6h = HFCLK is divided by 7 before being used for MFPCLK<br>7h = HFCLK is divided by 8 before being used for MFPCLK<br>8h = HFCLK is divided by 9 before being used for MFPCLK<br>9h = HFCLK is divided by 10 before being used for MFPCLK<br>Ah = HFCLK is divided by 11 before being used for MFPCLK<br>Bh = HFCLK is divided by 12 before being used for MFPCLK<br>Ch = HFCLK is divided by 13 before being used for MFPCLK<br>Dh = HFCLK is divided by 14 before being used for MFPCLK<br>Eh = HFCLK is divided by 15 before being used for MFPCLK<br>Fh = HFCLK is divided by 16 before being used for MFPCLK |
| 11-10 | RESERVED | R | 0h | |
| 9 | MFPCLKSRC | R/W | 0h | MFPCLKSRC selects the MFPCLK (middle frequency precision clock) source.<br>0h = MFPCLK is sourced from SYSOSC<br>1h = MFPCLK is sourced from HFCLK |
| 8 | CANCLKSRC | R/W | 0h | CANCLKSRC selects the CANCLK source.<br>0h = CANCLK source is HFCLK<br>1h = CANCLK source is SYSPLLCLK1 |
| 7 | EXCLKDIVEN | R/W | 0h | EXCLKDIVEN enables or disables the divider function of the CLK_OUT external clock output block.<br>0h = CLock divider is disabled (passthrough, EXCLKDIVVAL is not applied)<br>1h = Clock divider is enabled (EXCLKDIVVAL is applied) |
| 6-4 | EXCLKDIVVAL | R/W | 0h | EXCLKDIVVAL selects the divider value for the divider in the CLK_OUT external clock output block.<br>0h = CLK_OUT source is divided by 2<br>1h = CLK_OUT source is divided by 4<br>2h = CLK_OUT source is divided by 6<br>3h = CLK_OUT source is divided by 8<br>4h = CLK_OUT source is divided by 10<br>5h = CLK_OUT source is divided by 12<br>6h = CLK_OUT source is divided by 14<br>7h = CLK_OUT source is divided by 16 |
| 3 | RESERVED | R | 0h | |
| 2-0 | EXCLKSRC | R/W | 0h | EXCLKSRC selects the source for the CLK_OUT external clock output block. ULPCLK and MFPCLK require the CLK_OUT divider (EXCLKDIVEN) to be enabled<br>0h = CLK_OUT is SYSOSC<br>1h = CLK_OUT is ULPCLK (EXCLKDIVEN must be enabled)<br>2h = CLK_OUT is LFCLK<br>3h = CLK_OUT is MFPCLK (EXCLKDIVEN must be enabled)<br>4h = CLK_OUT is HFCLK<br>5h = CLK_OUT is SYSPLLCLK1 (SYSPLLCLK1 must be <=48MHz) |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.22 GENCLKEN (Offset = 113Ch) [Reset = 0000000Xh]

GENCLKEN is shown in Figure 2-83 and described in Table 2-95.

Return to the Summary Table.

General clock enable control

**Figure 2-83. GENCLKEN**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | MFPCLKEN | RESERVED | | | EXCLKEN |
| R-0h | | | R/W-0h | R-0h | | | R/W-0h |

**Table 2-95. GENCLKEN Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-5 | RESERVED | R | 0h | |
| 4 | MFPCLKEN | R/W | 0h | MFPCLKEN enables the middle frequency precision clock (MFPCLK).<br>0h = MFPCLK is disabled<br>1h = MFPCLK is enabled |
| 3-1 | RESERVED | R | 0h | |
| 0 | EXCLKEN | R/W | 0h | EXCLKEN enables the CLK_OUT external clock output block.<br>0h = CLK_OUT block is disabled<br>1h = CLK_OUT block is enabled |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.23 PMODECFG (Offset = 1140h) [Reset = 00000000h]

PMODECFG is shown in Figure 2-84 and described in Table 2-96.

Return to the Summary Table.

Power mode configuration

**Figure 2-84. PMODECFG**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | | | DSLEEP | |
| R-0h | | | | | | | | | | | | | | R/W-0h | |

**Table 2-96. PMODECFG Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | |
| 1-0 | DSLEEP | R/W | 0h | DSLEEP selects the operating mode to enter upon a DEEPSLEEP request from the CPU.<br>0h = STOP mode is entered<br>1h = STANDBY mode is entered<br>2h = SHUTDOWN mode is entered<br>3h = Reserved |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.24 FCC (Offset = 1150h) [Reset = 00000000h]

FCC is shown in Figure 2-85 and described in Table 2-97.

Return to the Summary Table.

Frequency clock counter (FCC) count

**Figure 2-85. FCC**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | DATA | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | R-0h | | | | | | | | | | | | | | | | | | | | | |

**Table 2-97. FCC Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-22 | RESERVED | R | 0h | |
| 21-0 | DATA | R | 0h | Frequency clock counter (FCC) count value. |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.25 SYSOSCTRIMUSER (Offset = 1170h) [Reset = 0000X0XXh]

SYSOSCTRIMUSER is shown in Figure 2-86 and described in Table 2-98.

Return to the Summary Table.

SYSOSC user-specified trim

**Figure 2-86. SYSOSCTRIMUSER**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | RDIV | | | | |
| R-0h | | | R/W-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RDIV | | | | RESFINE | | | |
| R/W-0h | | | | R/W-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | RESCOARSE | | | | | |
| R-0h | | R/W-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | CAP | | | RESERVED | | FREQ | |
| R-0h | R/W-0h | | | R-0h | | R/W-0h | |

**Table 2-98. SYSOSCTRIMUSER Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | RESERVED | R | 0h | |
| 28-20 | RDIV | R/W | 0h | RDIV specifies the frequency correction loop (FCL) resistor trim. This value changes with the target frequency. |
| 19-16 | RESFINE | R/W | 0h | RESFINE specifies the resister fine trim. This value changes with the target frequency. |
| 15-14 | RESERVED | R | 0h | |
| 13-8 | RESCOARSE | R/W | 0h | RESCOARSE specifies the resister coarse trim. This value changes with the target frequency. |
| 7 | RESERVED | R | 0h | |
| 6-4 | CAP | R/W | 0h | CAP specifies the SYSOSC capacitor trim. This value changes with the target frequency. |
| 3-2 | RESERVED | R | 0h | |
| 1-0 | FREQ | R/W | 0h | FREQ specifies the target user-trimmed frequency for SYSOSC.<br>0h = Reserved<br>1h = 16MHz user frequency<br>2h = 24MHz user frequency<br>3h = Reserved |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.26 SRAMBOUNDARY (Offset = 1178h) [Reset = 000000XXh]

SRAMBOUNDARY is shown in Figure 2-87 and described in Table 2-99.

Return to the Summary Table.

SRAM Write Boundary

**Figure 2-87. SRAMBOUNDARY**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | ADDR | | | | | | | | | | | | | | | RESERVED | | | | |
| R-0h | | | | | | | | | | | | R/W-0h | | | | | | | | | | | | | | | R-0h | | | | |

**Table 2-99. SRAMBOUNDARY Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | RESERVED | R | 0h | |
| 19-5 | ADDR | R/W | 0h | SRAM boundary configuration. The value configured into this acts such that: SRAM accesses to addresses less than or equal value will be RW only. SRAM accesses to addresses greater than value will be RX only. Value of 0 is not valid (system will have no stack). If set to 0, the system acts as if the entire SRAM is RWX. Any non-zero value can be configured, including a value = SRAM size. |
| 4-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.27 SRAMBOUNDARYHIGH (Offset = 117Ch) [Reset = 000000XXh]

SRAMBOUNDARYHIGH is shown in Figure 2-88 and described in Table 2-100.

Return to the Summary Table.

SRAM Write Boundary High

**Figure 2-88. SRAMBOUNDARYHIGH**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | ADDR | | | | | | | | | | | | | | | RESERVED | | | | |
| R-0h | | | | | | | | | | | | R/W-0h | | | | | | | | | | | | | | | R-0h | | | | |

**Table 2-100. SRAMBOUNDARYHIGH Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | RESERVED | R | 0h | |
| 19-5 | ADDR | R/W | 0h | SRAM boundary high configuration. The value configured into this acts such that: SRAM accesses to greater than value will be RW only. SRAM accesses to addresses less than or equal to value will be RX until meeting SRAMBOUNDARY MMR Value. Value of 0 is default. If value set to less than or equal to SRAMBOUNDARY MMR, the system will only use the SRAMBOUNDARY MMR as criteria. |
| 4-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.28 SYSTEMCFG (Offset = 1180h) [Reset = 00XXXXXXh]

SYSTEMCFG is shown in Figure 2-89 and described in Table 2-101.

Return to the Summary Table.

System configuration

**Figure 2-89. SYSTEMCFG**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | FLASHECCRSTDIS | WWDTLP1RSTDIS | WWDTLP0RSTDIS |
| R-0h | | | | | R/W-1h | R/W-1h | R/W-0h |

**Table 2-101. SYSTEMCFG Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | The key value of 1Bh (27) must be written to KEY together with contents to be updated. Reads as 0<br>1Bh = Issue write |
| 23-3 | RESERVED | R | 0h | |
| 2 | FLASHECCRSTDIS | R/W | 1h | FLASHECCRSTDIS specifies whether a flash ECC double error detect (DED) will trigger a SYSRST or an NMI.<br>0h = Flash ECC DED will trigger a SYSRST<br>1h = Flash ECC DED will trigger a NMI |
| 1 | WWDTLP1RSTDIS | R/W | 1h | WWDTLP1RSTDIS specifies whether a WWDT Error Event will trigger a SYSRST or an NMI.<br>0h = WWDTLP1 Error Event will trigger a SYSRST<br>1h = WWDTLP1 Error Event will trigger an NMI |
| 0 | WWDTLP0RSTDIS | R/W | 0h | WWDTLP0RSTDIS specifies whether a WWDT Error Event will trigger a BOOTRST or an NMI.<br>0h = WWDTLP0 Error Event will trigger a BOOTRST<br>1h = WWDTLP0 Error Event will trigger an NMI |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.29 SRAMCFG (Offset = 1184h) [Reset = 00XXXXXXh]

SRAMCFG is shown in Figure 2-90 and described in Table 2-102.

Return to the Summary Table.

System SRAM configuration

**Figure 2-90. SRAMCFG**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | BANKSTOP1 | RESERVED |
| R-0h | | | | | | R/W-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | BANKOFF1 | RESERVED |
| R-0h | | | | | | R/W-0h | R-0h |

**Table 2-102. SRAMCFG Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | The key value of B5h (181) must be written to KEY together with contents to be updated. Reads as 0<br>B5h = Issue write |
| 23-10 | RESERVED | R | 0h | |
| 9 | BANKSTOP1 | R/W | 0h | SRAM BANK1 power level for STOP mode<br>0h = SRAM BANK1 power OFF for STOP mode<br>1h = SRAM BANK1 power RETAIN for STOP mode |
| 8-2 | RESERVED | R | 0h | |
| 1 | BANKOFF1 | R/W | 0h | SRAM BANK1 power level for RUN mode<br>0h = SRAM BANK1 power ON for RUN mode<br>1h = SRAM BANK1 power OFF for RUN mode |
| 0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.30 WRITELOCK (Offset = 1200h) [Reset = 00000000h]

WRITELOCK is shown in Figure 2-91 and described in Table 2-103.

Return to the Summary Table.

SYSCTL register write lockout

**Figure 2-91. WRITELOCK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | ACTIVE |
| R-0h | | | | | | | R/W-0h |

**Table 2-103. WRITELOCK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | ACTIVE | R/W | 0h | ACTIVE controls whether critical SYSCTL registers are write protected or not.<br>0h = Allow writes to lockable registers<br>1h = Disallow writes to lockable registers |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.31 CLKSTATUS (Offset = 1204h) [Reset = XXXXX0XXh]

CLKSTATUS is shown in Figure 2-92 and described in Table 2-104.

Return to the Summary Table.

Clock module (CKM) status

**Figure 2-92. CLKSTATUS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| ANACLKERR | RESERVED | SYSPLLBLKUPD | HFCLKBLKUPD | RESERVED | | FCCDONE | FCLMODE |
| R-0h | R-0h | R-0h | R-0h | R-0h | | R-0h | R-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| LFCLKFAIL | RESERVED | HSCLKGOOD | HSCLKDEAD | RESERVED | | CURMCLKSEL | CURHSCLKSEL |
| R-0h | R-0h | R-0h | R-0h | R-0h | | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | SYSPLLOFF | HFCLKOFF | HSCLKSOFF | LFOSCGOOD | LFXTGOOD | SYSPLLGOOD | HFCLKGOOD |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LFCLKMUX | | RESERVED | HSCLKMUX | RESERVED | | SYSOSCFREQ | |
| R-0h | | R-0h | R-0h | R-0h | | R-0h | |

**Table 2-104. CLKSTATUS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | ANACLKERR | R | 0h | ANACLKERR is set when the device clock configuration does not support an enabled analog peripheral mode and the analog peripheral may not be functioning as expected.<br>0h = No analog clock errors detected<br>1h = Analog clock error detected |
| 30 | RESERVED | R | 0h | |
| 29 | SYSPLLBLKUPD | R | 0h | SYSPLLBLKUPD indicates when writes to SYSPLLCFG0/1 and SYSPLLPARAM0/1 are blocked.<br>0h = writes to SYSPLLCFG0/1 and SYSPLLPARAM0/1 are allowed<br>1h = writes to SYSPLLCFG0/1 and SYSPLLPARAM0/1 are blocked |
| 28 | HFCLKBLKUPD | R | 0h | HFCLKBLKUPD indicates when writes to the HFCLKCLKCFG register are blocked.<br>0h = Writes to HFCLKCLKCFG are allowed<br>1h = Writes to HFCLKCLKCFG are blocked |
| 27-26 | RESERVED | R | 0h | |
| 25 | FCCDONE | R | 0h | FCCDONE indicates when a frequency clock counter capture is complete.<br>0h = FCC capture is not done<br>1h = FCC capture is done |
| 24 | FCLMODE | R | 0h | FCLMODE indicates if the SYSOSC frequency correction loop (FCL) is enabled.<br>0h = SYSOSC FCL is disabled<br>1h = SYSOSC FCL is enabled |
| 23 | LFCLKFAIL | R | 0h | LFCLKFAIL indicates when the continous LFCLK monitor detects a LFXT or LFCLK_IN clock stuck failure.<br>0h = No LFCLK fault detected<br>1h = LFCLK stuck fault detected |
| 22 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

**Table 2-104. CLKSTATUS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 21 | HSCLKGOOD | R | 0h | HSCLKGOOD is set by hardware if the selected clock source for HSCLK started successfully.<br>0h = The HSCLK source did not start correctly<br>1h = The HSCLK source started correctly |
| 20 | HSCLKDEAD | R | 0h | HSCLKDEAD is set by hardware if the selected source for HSCLK was started but did not start successfully.<br>0h = The HSCLK source was not started or started correctly<br>1h = The HSCLK source did not start correctly |
| 19-18 | RESERVED | R | 0h | |
| 17 | CURMCLKSEL | R | 0h | CURMCLKSEL indicates if MCLK is currently sourced from LFCLK.<br>0h = MCLK is not sourced from LFCLK<br>1h = MCLK is sourced from LFCLK |
| 16 | CURHSCLKSEL | R | 0h | CURHSCLKSEL indicates the current clock source for HSCLK.<br>0h = HSCLK is currently sourced from the SYSPLL<br>1h = HSCLK is currently sourced from the HFCLK |
| 15 | RESERVED | R | 0h | |
| 14 | SYSPLLOFF | R | 0h | SYSPLLOFF indicates if the SYSPLL is disabled or was dead at startup. When the SYSPLL is started, SYSPLLOFF is cleared by hardware. Following startup of the SYSPLL, if the SYSPLL startup monitor determines that the SYSPLL was not started correctly, SYSPLLOFF is set.<br>0h = SYSPLL started correctly and is enabled<br>1h = SYSPLL is disabled or was dead startup |
| 13 | HFCLKOFF | R | 0h | HFCLKOFF indicates if the HFCLK is disabled or was dead at startup. When the HFCLK is started, HFCLKOFF is cleared by hardware. Following startup of the HFCLK, if the HFCLK startup monitor determines that the HFCLK was not started correctly, HFCLKOFF is set.<br>0h = HFCLK started correctly and is enabled<br>1h = HFCLK is disabled or was dead at startup |
| 12 | HSCLKSOFF | R | 0h | HSCLKSOFF is set when the high speed clock sources (SYSPLL, HFCLK) are disabled or dead. It is the logical AND of HFCLKOFF and SYSPLLOFF.<br>0h = SYSPLL, HFCLK, or both were started correctly and remain enabled<br>1h = SYSPLL and HFCLK are both either off or dead |
| 11 | LFOSCGOOD | R | 0h | LFOSCGOOD indicates when the LFOSC startup has completed and the LFOSC is ready for use.<br>0h = LFOSC is not ready<br>1h = LFOSC is ready |
| 10 | LFXTGOOD | R | 0h | LFXTGOOD indicates if the LFXT started correctly. When the LFXT is started, LFXTGOOD is cleared by hardware. After the startup settling time has expired, the LFXT status is tested. If the LFXT started successfully the LFXTGOOD bit is set, else it is left cleared.<br>0h = LFXT did not start correctly<br>1h = LFXT started correctly |
| 9 | SYSPLLGOOD | R | 0h | SYSPLLGOOD indicates if the SYSPLL started correctly. When the SYSPLL is started, SYSPLLGOOD is cleared by hardware. After the startup settling time has expired, the SYSPLL status is tested. If the SYSPLL started successfully the SYSPLLGOOD bit is set, else it is left cleared.<br>0h = SYSPLL did not start correctly<br>1h = SYSPLL started correctly |
| 8 | HFCLKGOOD | R | 0h | HFCLKGOOD indicates that the HFCLK started correctly. When the HFXT is started or HFCLK_IN is selected as the HFCLK source, this bit will be set by hardware if a valid HFCLK is detected, and cleared if HFCLK is not operating within the expected range.<br>0h = HFCLK did not start correctly<br>1h = HFCLK started correctly |



Copyright © 2026 Texas Instruments Incorporated

**Table 2-104. CLKSTATUS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7-6 | LFCLKMUX | R | 0h | LFCLKMUX indicates if LFCLK is sourced from the internal LFOSC, the low frequency crystal (LFXT), or the LFCLK_IN digital clock input.<br>0h = LFCLK is sourced from the internal LFOSC<br>1h = LFCLK is sourced from the LFXT (crystal)<br>2h = LFCLK is sourced from LFCLK_IN (external digital clock input) |
| 5 | RESERVED | R | 0h | |
| 4 | HSCLKMUX | R | 0h | HSCLKMUX indicates if MCLK is currently sourced from the high-speed clock (HSCLK).<br>0h = MCLK is not sourced from HSCLK<br>1h = MCLK is sourced from HSCLK |
| 3-2 | RESERVED | R | 0h | |
| 1-0 | SYSOSCFREQ | R | 0h | SYSOSCFREQ indicates the current SYSOSC operating frequency.<br>0h = SYSOSC is at base frequency (32MHz)<br>1h = SYSOSC is at low frequency (4MHz)<br>2h = SYSOSC is at the user-trimmed frequency (16 or 24MHz)<br>3h = Reserved |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.32 SYSSTATUS (Offset = 1208h) [Reset = XXXXXXX0h]

SYSSTATUS is shown in Figure 2-93 and described in Table 2-105.

Return to the Summary Table.

System status information

**Figure 2-93. SYSSTATUS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REBOOTATTEMPTS | | RESERVED | | | | | |
| R-0h | | R-0h | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | SRAMBANK1READY | RESERVED |
| R-0h | | | | | | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | SHDNIOLOCK | SWDCFGDIS | EXTRSTPINDIS | RESERVED | | MCAN1READY | MCAN0READY |
| R-0h | R-0h | R-0h | R-0h | R-0h | | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | PMUIREFGOOD | ANACPUMPGOOD | BORLVL | BORCURTHRESHOLD | | FLASHSEC | FLASHDED |
| R-0h | R-0h | R-0h | R-0h | R-0h | | R-0h | R-0h |

**Table 2-105. SYSSTATUS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | REBOOTATTEMPTS | R | 0h | REBOOTATTEMPTS indicates the number of boot attempts taken before the user application starts. |
| 29-18 | RESERVED | R | 0h | |
| 17 | SRAMBANK1READY | R | 0h | SRAM BANK1 READY STATE<br>0h = SRAM BANK1 is NOT READY for access<br>1h = SRAM BANK1 is READY for access |
| 16-15 | RESERVED | R | 0h | |
| 14 | SHDNIOLOCK | R | 0h | SHDNIOLOCK indicates when IO is locked due to SHUTDOWN<br>0h = IO IS NOT Locked due to SHUTDOWN<br>1h = IO IS Locked due to SHUTDOWN |
| 13 | SWDCFGDIS | R | 0h | SWDCFGDIS indicates when user has disabled the use of SWD Port<br>0h = SWD Port Enabled<br>1h = SWD Port Disabled |
| 12 | EXTRSTPINDIS | R | 0h | EXTRSTPINDIS indicates when user has disabled the use of external reset pin<br>0h = External Reset Pin Enabled<br>1h = External Reset Pin Disabled |
| 11-10 | RESERVED | R | 0h | |
| 9 | MCAN1READY | R | 0h | MCAN1READY indicates when the MCAN1 peripheral is ready.<br>0h = MCAN1 is not ready<br>1h = MCAN1 is ready |
| 8 | MCAN0READY | R | 0h | MCAN0READY indicates when the MCAN0 peripheral is ready.<br>0h = MCAN0 is not ready<br>1h = MCAN0 is ready |
| 7 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

**Table 2-105. SYSSTATUS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 6 | PMUIREFGOOD | R | 0h | PMUIREFGOOD is set by hardware when the PMU current reference is ready.<br>0h = IREF is not ready<br>1h = IREF is ready |
| 5 | ANACPUMPGOOD | R | 0h | ANACPUMPGOOD is set by hardware when the VBOOST analog mux charge pump is ready.<br>0h = VBOOST is not ready<br>1h = VBOOST is ready |
| 4 | BORLVL | R | 0h | BORLVL indicates if a BOR event occured and the BOR threshold was switched to BOR0 by hardware.<br>0h = No BOR violation occured<br>1h = A BOR violation occured and the BOR threshold was switched to BOR0 |
| 3-2 | BORCURTHRESHOLD | R | 0h | BORCURTHRESHOLD indicates the active brown-out reset supply monitor configuration.<br>0h = Default minimum threshold; a BOR0- violation triggers a BOR<br>1h = A BOR1- violation generates a BORLVL interrupt<br>2h = A BOR2- violation generates a BORLVL interrupt<br>3h = A BOR3- violation generates a BORLVL interrupt |
| 1 | FLASHSEC | R | 0h | FLASHSEC indicates if a flash ECC single bit error was detected and corrected (SEC).<br>0h = No flash ECC single bit error detected<br>1h = Flash ECC single bit error was detected and corrected |
| 0 | FLASHDED | R | 0h | FLASHDED indicates if a flash ECC double bit error was detected (DED).<br>0h = No flash ECC double bit error detected<br>1h = Flash ECC double bit error detected |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.33 DEDERRADDR (Offset = 120Ch) [Reset = 00000000h]

DEDERRADDR is shown in Figure 2-94 and described in Table 2-106.

Return to the Summary Table.

Memory DED Address

**Figure 2-94. DEDERRADDR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADDR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 2-106. DEDERRADDR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | ADDR | R | 0h | Address of MEMORY DED Error. |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.34 RSTCAUSE (Offset = 1220h) [Reset = 00000000h]

RSTCAUSE is shown in Figure 2-95 and described in Table 2-107.

Return to the Summary Table.

Reset cause

**Figure 2-95. RSTCAUSE**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | ID | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | RC-0h | | | | |

**Table 2-107. RSTCAUSE Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-5 | RESERVED | R | 0h | |
| 4-0 | ID | RC | 0h | ID is a read-to-clear field which indicates the lowest level reset cause since the last read.<br>0h = No reset since last read<br>1h = POR- violation, SHUTDNSTOREx or PMU trim parity fault<br>2h = NRST triggered POR (>1s hold)<br>3h = Software triggered POR<br>4h = BOR0- violation<br>5h = SHUTDOWN mode exit<br>8h = Non-PMU trim parity fault<br>9h = Fatal clock failure<br>Ch = NRST triggered BOOTRST (<1s hold)<br>Dh = Software triggered BOOTRST<br>Eh = WWDT0 violation<br>10h = BSL exit<br>11h = BSL entry<br>13h = WWDT1 violation<br>14h = Flash uncorrectable ECC error<br>15h = CPULOCK violation<br>1Ah = Debug triggered SYSRST<br>1Bh = Software triggered SYSRST<br>1Ch = Debug triggered CPURST<br>1Dh = Software triggered CPURST |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.35 RESETLEVEL (Offset = 1300h) [Reset = 00000000h]

RESETLEVEL is shown in Figure 2-96 and described in Table 2-108.

Return to the Summary Table.

Reset level for application-triggered reset command

**Figure 2-96. RESETLEVEL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | | LEVEL | | |
| R-0h | | | | | | | | | | | | | R/W-0h | | |

**Table 2-108. RESETLEVEL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | RESERVED | R | 0h | |
| 2-0 | LEVEL | R/W | 0h | LEVEL is used to specify the type of reset to be issued when RESETCMD is set to generate a software triggered reset.<br>0h = Issue a SYSRST (CPU plus peripherals only)<br>1h = Issue a BOOTRST (CPU, peripherals, and boot configuration routine)<br>2h = Issue a SYSRST and enter the boot strap loader (BSL)<br>3h = Issue a power-on reset (POR)<br>4h = Issue a SYSRST and exit the boot strap loader (BSL) |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.36 RESETCMD (Offset = 1304h) [Reset = 00XXXXXXh]

RESETCMD is shown in Figure 2-97 and described in Table 2-109.

Return to the Summary Table.

Execute an application-triggered reset command

**Figure 2-97. RESETCMD**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| KEY | | | | | | | | RESERVED | | | | | | | |
| W-0h | | | | | | | | R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | | | | GO |
| R-0h | | | | | | | | | | | | | | | W-0h |

**Table 2-109. RESETCMD Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | The key value of E4h (228) must be written to KEY together with GO to trigger the reset.<br>E4h = Issue reset |
| 23-1 | RESERVED | R | 0h | |
| 0 | GO | W | 0h | Execute the reset specified in RESETLEVEL.LEVEL. Must be written together with the KEY.<br>1h = Issue reset |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.37 BORTHRESHOLD (Offset = 1308h) [Reset = 00000000h]

BORTHRESHOLD is shown in Figure 2-98 and described in Table 2-110.

Return to the Summary Table.

BOR threshold selection

**Figure 2-98. BORTHRESHOLD**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | | | LEVEL | |
| R-0h | | | | | | | | | | | | | | R/W-0h | |

**Table 2-110. BORTHRESHOLD Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | |
| 1-0 | LEVEL | R/W | 0h | LEVEL specifies the desired BOR threshold and BOR mode.<br>0h = Default minimum threshold; a BOR0- violation triggers a BOR<br>1h = A BOR1- violation generates a BORLVL interrupt<br>2h = A BOR2- violation generates a BORLVL interrupt<br>3h = A BOR3- violation generates a BORLVL interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.38 BORCLRCMD (Offset = 130Ch) [Reset = 00XXXXXXh]

BORCLRCMD is shown in Figure 2-99 and described in Table 2-111.

Return to the Summary Table.

Set the BOR threshold

**Figure 2-99. BORCLRCMD**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| KEY | | | | | | | | RESERVED | | | | | | | |
| W-0h | | | | | | | | R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | | | | GO |
| R-0h | | | | | | | | | | | | | | | W-0h |

**Table 2-111. BORCLRCMD Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | The key value of C7h (199) must be written to KEY together with GO to trigger the clear and BOR threshold change.<br>C7h = Issue clear |
| 23-1 | RESERVED | R | 0h | |
| 0 | GO | W | 0h | GO clears any prior BOR violation status indications and attempts to change the active BOR mode to that specified in the LEVEL field of the BORTHRESHOLD register.<br>1h = Issue clear |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.39 SYSOSCFCLCTL (Offset = 1310h) [Reset = 00XXXXXXh]

SYSOSCFCLCTL is shown in Figure 2-100 and described in Table 2-112.

Return to the Summary Table.

SYSOSC frequency correction loop (FCL) ROSC enable

**Figure 2-100. SYSOSCFCLCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | SETUSEFCL |
| R-0h | | | | | | | W-0h |

**Table 2-112. SYSOSCFCLCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | The key value of 2Ah (42) must be written to KEY together with SETUSEFCL to enable the FCL.<br>2Ah = Issue Command |
| 23-1 | RESERVED | R | 0h | |
| 0 | SETUSEFCL | W | 0h | Set SETUSEFCL to enable the frequency correction loop in SYSOSC. Once enabled, this state is locked until the next BOOTRST.<br>1h = Enable the SYSOSC FCL |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.40 LFXTCTL (Offset = 1314h) [Reset = 00XXXXXXh]

LFXTCTL is shown in Figure 2-101 and described in Table 2-113.

Return to the Summary Table.

LFXT and LFCLK control

**Figure 2-101. LFXTCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | SETUSELFXT | STARTLFXT |
| R-0h | | | | | | W-0h | W-0h |

**Table 2-113. LFXTCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | The key value of 91h (145) must be written to KEY together with either STARTLFXT or SETUSELFXT to set the corresponding bit.<br>91h = Issue command |
| 23-2 | RESERVED | R | 0h | |
| 1 | SETUSELFXT | W | 0h | Set SETUSELFXT to switch LFCLK to LFXT. Once set, SETUSELFXT remains set until the next BOOTRST.<br>0h = 0<br>1h = Use LFXT as the LFCLK source |
| 0 | STARTLFXT | W | 0h | Set STARTLFXT to start the low frequency crystal oscillator (LFXT). Once set, STARTLFXT remains set until the next BOOTRST.<br>0h = LFXT not started<br>1h = Start LFXT |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.41 EXLFCTL (Offset = 1318h) [Reset = 00XXXXXXh]

EXLFCTL is shown in Figure 2-102 and described in Table 2-114.

Return to the Summary Table.

LFCLK_IN and LFCLK control

**Figure 2-102. EXLFCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | SETUSEEXLF |
| R-0h | | | | | | | W-0h |

**Table 2-114. EXLFCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | The key value of 36h (54) must be written to KEY together with SETUSEEXLF to set SETUSEEXLF.<br>36h = Issue command |
| 23-1 | RESERVED | R | 0h | |
| 0 | SETUSEEXLF | W | 0h | Set SETUSEEXLF to switch LFCLK to the LFCLK_IN digital clock input. Once set, SETUSEEXLF remains set until the next BOOTRST.<br>1h = Use LFCLK_IN as the LFCLK source |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.42 SHDNIOREL (Offset = 131Ch) [Reset = 00XXXXXXh]

SHDNIOREL is shown in Figure 2-103 and described in Table 2-115.

Return to the Summary Table.

SHUTDOWN IO release control

**Figure 2-103. SHDNIOREL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | RELEASE |
| R-0h | | | | | | | W-0h |

**Table 2-115. SHDNIOREL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | The key value 91h must be written to KEY together with RELEASE to set RELEASE.<br>91h = Issue command |
| 23-1 | RESERVED | R | 0h | |
| 0 | RELEASE | W | 0h | Set RELEASE to release the IO after a SHUTDOWN mode exit.<br>1h = Release IO |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.43 EXRSTPIN (Offset = 1320h) [Reset = 00XXXXXXh]

EXRSTPIN is shown in Figure 2-104 and described in Table 2-116.

Return to the Summary Table.

Disable the reset function of the NRST pin

**Figure 2-104. EXRSTPIN**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | DISABLE |
| R-0h | | | | | | | W-0h |

**Table 2-116. EXRSTPIN Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | The key value 1Eh must be written together with DISABLE to disable the reset function.<br>1Eh = Issue command |
| 23-1 | RESERVED | R | 0h | |
| 0 | DISABLE | W | 0h | Set DISABLE to disable the reset function of the NRST pin. Once set, this configuration is locked until the next POR.<br>0h = Reset function of NRST pin is enabled<br>1h = Reset function of NRST pin is disabled |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.44 SYSSTATUSCLR (Offset = 1324h) [Reset = 00XXXXXXh]

SYSSTATUSCLR is shown in Figure 2-105 and described in Table 2-117.

Return to the Summary Table.

Clear sticky bits of SYSSTATUS

**Figure 2-105. SYSSTATUSCLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | ALLECC |
| R-0h | | | | | | | W-0h |

**Table 2-117. SYSSTATUSCLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | The key value CEh (206) must be written to KEY together with ALLECC to clear the ECC state.<br>CEh = Issue command |
| 23-1 | RESERVED | R | 0h | |
| 0 | ALLECC | W | 0h | Set ALLECC to clear all ECC related SYSSTATUS indicators.<br>1h = Clear ECC error state |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.45 SWDCFG (Offset = 1328h) [Reset = 00XXXXXXh]

SWDCFG is shown in Figure 2-106 and described in Table 2-118.

Return to the Summary Table.

Disable the SWD function on the SWD pins

**Figure 2-106. SWDCFG**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | DISABLE |
| R-0h | | | | | | | W-0h |

**Table 2-118. SWDCFG Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | The key value 62h (98) must be written to KEY together with DISBALE to disable the SWD functions.<br>62h = Issue command |
| 23-1 | RESERVED | R | 0h | |
| 0 | DISABLE | W | 0h | Set DISABLE to disable the SWD function on SWD pins, allowing the SWD pins to be used as GPIO.<br>1h = Disable SWD function on SWD pins |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.46 FCCCMD (Offset = 132Ch) [Reset = 00XXXXXXh]

FCCCMD is shown in Figure 2-107 and described in Table 2-119.

Return to the Summary Table.

Frequency clock counter start capture

**Figure 2-107. FCCCMD**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| KEY | | | | | | | | RESERVED | | | | | | | |
| W-0h | | | | | | | | R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | | | | GO |
| R-0h | | | | | | | | | | | | | | | W-0h |

**Table 2-119. FCCCMD Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | The key value 0Eh (14) must be written with GO to start a capture.<br>0Eh = Issue command |
| 23-1 | RESERVED | R | 0h | |
| 0 | GO | W | 0h | Set GO to start a capture with the frequency clock counter (FCC).<br>1h = 1 |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.47 SHUTDNSTORE0 (Offset = 1400h) [Reset = 00000000h]

SHUTDNSTORE0 is shown in Figure 2-108 and described in Table 2-120.

Return to the Summary Table.

Shutdown storage memory (byte 0)

**Figure 2-108. SHUTDNSTORE0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DATA | | | | | | | |
| R/W-0h | | | | | | | |

**Table 2-120. SHUTDNSTORE0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-10 | RESERVED | R | 0h | |
| 7-0 | DATA | R/W | 0h | Shutdown storage byte 0 |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.48 SHUTDNSTORE1 (Offset = 1404h) [Reset = 00000000h]

SHUTDNSTORE1 is shown in Figure 2-109 and described in Table 2-121.

Return to the Summary Table.

Shutdown storage memory (byte 1)

**Figure 2-109. SHUTDNSTORE1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DATA | | | | | | | |
| R/W-0h | | | | | | | |

**Table 2-121. SHUTDNSTORE1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-10 | RESERVED | R | 0h | |
| 7-0 | DATA | R/W | 0h | Shutdown storage byte 1 |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.49 SHUTDNSTORE2 (Offset = 1408h) [Reset = 00000000h]

SHUTDNSTORE2 is shown in Figure 2-110 and described in Table 2-122.

Return to the Summary Table.

Shutdown storage memory (byte 2)

**Figure 2-110. SHUTDNSTORE2**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DATA | | | | | | | |
| R/W-0h | | | | | | | |

**Table 2-122. SHUTDNSTORE2 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-10 | RESERVED | R | 0h | |
| 7-0 | DATA | R/W | 0h | Shutdown storage byte 2 |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.50 SHUTDNSTORE3 (Offset = 140Ch) [Reset = 00000000h]

SHUTDNSTORE3 is shown in Figure 2-111 and described in Table 2-123.

Return to the Summary Table.

Shutdown storage memory (byte 3)

**Figure 2-111. SHUTDNSTORE3**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DATA | | | | | | | |
| R/W-0h | | | | | | | |

**Table 2-123. SHUTDNSTORE3 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-10 | RESERVED | R | 0h | |
| 7-0 | DATA | R/W | 0h | Shutdown storage byte 3 |



Copyright © 2026 Texas Instruments Incorporated

### 2.6.51 IDREG (Offset = 2040h) [Reset = 00000000h]

IDREG is shown in Figure 2-112 and described in Table 2-124.

Return to the Summary Table.

JTAG USERCODE

**Figure 2-112. IDREG**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 2-124. IDREG Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | R | 0h | ID Register |



Copyright © 2026 Texas Instruments Incorporated

## 2.7 SYSCTL_G511x_G5187 Registers

Table 2-125 lists the memory-mapped registers for the SYSCTL_G511x_G5187 registers. All register offset addresses not listed in Table 2-125 should be considered as reserved locations and the register contents should not be modified.

**Table 2-125. SYSCTL_G511X_G5187 Registers**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 1020h | IIDX | SYSCTL interrupt index | | Go |
| 1028h | IMASK | SYSCTL interrupt mask | | Go |
| 1030h | RIS | SYSCTL raw interrupt status | | Go |
| 1038h | MIS | SYSCTL masked interrupt status | | Go |
| 1040h | ISET | SYSCTL interrupt set | | Go |
| 1048h | ICLR | SYSCTL interrupt clear | | Go |
| 1050h | NMIIIDX | NMI interrupt index | | Go |
| 1060h | NMIRIS | NMI raw interrupt status | | Go |
| 1070h | NMIISET | NMI interrupt set | | Go |
| 1078h | NMIICLR | NMI interrupt clear | | Go |
| 1100h | SYSOSCCFG | SYSOSC configuration | | Go |
| 1104h | MCLKCFG | Main clock (MCLK) configuration | | Go |
| 1108h | HSCLKEN | High-speed clock (HSCLK) source enable/disable | | Go |
| 110Ch | HSCLKCFG | High-speed clock (HSCLK) source selection | | Go |
| 1110h | HFCLKCLKCFG | High-frequency clock (HFCLK) configuration | | Go |
| 1114h | LFCLKCFG | Low frequency crystal oscillator (LFXT) configuration | | Go |
| 1120h | SYSPLLCFG0 | SYSPLL reference and output configuration | | Go |
| 1124h | SYSPLLCFG1 | SYSPLL reference and feedback divider | | Go |
| 1128h | SYSPLLPARAM0 | SYSPLL PARAM0 (load from FACTORY region) | | Go |
| 112Ch | SYSPLLPARAM1 | SYSPLL PARAM1 (load from FACTORY region) | | Go |
| 1138h | GENCLKCFG | General clock configuration | | Go |
| 113Ch | GENCLKEN | General clock enable control | | Go |
| 1140h | PMODECFG | Power mode configuration | | Go |
| 1150h | FCC | Frequency clock counter (FCC) count | | Go |
| 1170h | SYSOSCTRIMUSER | SYSOSC user-specified trim | | Go |
| 1178h | SRAMBOUNDARY | SRAM Write Boundary | | Go |
| 1180h | SYSTEMCFG | System configuration | | Go |
| 1200h | WRITELOCK | SYSCTL register write lockout | | Go |
| 1204h | CLKSTATUS | Clock module (CKM) status | | Go |
| 1208h | SYSSTATUS | System status information | | Go |
| 120Ch | DEDERRADDR | Memory DED Address | | Go |
| 1220h | RSTCAUSE | Reset cause | | Go |
| 1300h | RESETLEVEL | Reset level for application-triggered reset command | | Go |
| 1304h | RESETCMD | Execute an application-triggered reset command | | Go |



Copyright © 2026 Texas Instruments Incorporated

**Table 2-125. SYSCTL_G511X_G5187 Registers (continued)**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 1308h | BORTHRESHOLD | BOR threshold selection | | Go |
| 130Ch | BORCLRCMD | Set the BOR threshold | | Go |
| 1310h | SYSOSCFCLCTL | SYSOSC frequency correction loop (FCL) ROSC enable | | Go |
| 1314h | LFXTCTL | LFXT and LFCLK control | | Go |
| 1318h | EXLFCTL | LFCLK_IN and LFCLK control | | Go |
| 131Ch | SHDNIOREL | SHUTDOWN IO release control | | Go |
| 1320h | EXRSTPIN | Disable the reset function of the NRST pin | | Go |
| 1324h | SYSSTATUSCLR | Clear sticky bits of SYSSTATUS | | Go |
| 1328h | SWDCFG | Disable the SWD function on the SWD pins | | Go |
| 132Ch | FCCCMD | Frequency clock counter start capture | | Go |
| 1340h | USBFLLCTL | USB FLL Control Register | | Go |
| 1344h | USBFLLADJ | USB FLL Manual Adjustment Register | | Go |
| 1348h | USBFLLSTAT | USB FLL Status Register | | Go |
| 1400h | SHUTDNSTORE0 | Shutdown storage memory (byte 0) | | Go |
| 1404h | SHUTDNSTORE1 | Shutdown storage memory (byte 1) | | Go |
| 1408h | SHUTDNSTORE2 | Shutdown storage memory (byte 2) | | Go |
| 140Ch | SHUTDNSTORE3 | Shutdown storage memory (byte 3) | | Go |

Complex bit access types are encoded to fit into small table cells. Table 2-126 shows the codes that are used for access types in this section.

**Table 2-126. SYSCTL_G511x_G5187 Access Type Codes**

| Access Type | Code | Description |
|---|---|---|
| Read Type | | |
| R | R | Read |
| RC | R<br>C | Read<br>to Clear |
| Write Type | | |
| W | W | Write |
| W1C | W<br>1C | Write<br>1 to clear |
| W1S | W<br>1S | Write<br>1 to set |
| Reset or Default Value | | |
| *-n* | | Value after reset or the default value |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.1 IIDX (Offset = 1020h) [Reset = 00000000h]

IIDX is shown in Figure 2-113 and described in Table 2-127.

Return to the Summary Table.

SYSCTL interrupt index

**Figure 2-113. IIDX**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | STAT | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | |

**Table 2-127. IIDX Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3-0 | STAT | R | 0h | The SYSCTL interrupt index (IIDX) register generates a value corresponding to the highest priority pending interrupt source. This value may be used as an address offset for fast, deterministic handling in the interrupt service routine. A read of the IIDX register will clear the corresponding interrupt status in the RIS and MIS registers.<br>0h = No interrupt pending<br>1h = LFOSCGOOD interrupt pending<br>2h = 2<br>3h = 3<br>4h = 4<br>5h = 5<br>6h = 6<br>7h = 7<br>8h = 8<br>9h = 9 |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.2 IMASK (Offset = 1028h) [Reset = 00000000h]

IMASK is shown in Figure 2-114 and described in Table 2-128.

Return to the Summary Table.

SYSCTL interrupt mask

**Figure 2-114. IMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | USBFLLERR |
| R-0h | | | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| HSCLKGOOD | SYSPLLGOOD | HFCLKGOOD | LFXTGOOD | SRAMSEC | FLASHSEC | ANACLKERR | LFOSCGOOD |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

**Table 2-128. IMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-9 | RESERVED | R | 0h | |
| 8 | USBFLLERR | R/W | 0h | USB FLL Reference Clock Error<br>0h = 0<br>1h = 1 |
| 7 | HSCLKGOOD | R/W | 0h | HSCLK GOOD<br>0h = 0<br>1h = 1 |
| 6 | SYSPLLGOOD | R/W | 0h | SYSPLL GOOD<br>0h = 0<br>1h = 1 |
| 5 | HFCLKGOOD | R/W | 0h | HFCLK GOOD<br>0h = 0<br>1h = 1 |
| 4 | LFXTGOOD | R/W | 0h | LFXT GOOD<br>0h = 0<br>1h = 1 |
| 3 | SRAMSEC | R/W | 0h | SRAM Single Error Correct<br>0h = 0<br>1h = 1 |
| 2 | FLASHSEC | R/W | 0h | Flash Single Error Correct<br>0h = 0<br>1h = 1 |
| 1 | ANACLKERR | R/W | 0h | Analog Clocking Consistency Error<br>0h = 0<br>1h = 1 |
| 0 | LFOSCGOOD | R/W | 0h | Enable or disable the LFOSCGOOD interrupt. LFOSCGOOD indicates that the LFOSC has started successfully.<br>0h = Interrupt disabled<br>1h = Interrupt enabled |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.3 RIS (Offset = 1030h) [Reset = 00000000h]

RIS is shown in Figure 2-115 and described in Table 2-129.

Return to the Summary Table.

SYSCTL raw interrupt status

**Figure 2-115. RIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | USBFLLERR |
| R-0h | | | | | | | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| HSCLKGOOD | SYSPLLGOOD | HFCLKGOOD | LFXTGOOD | SRAMSEC | FLASHSEC | ANACLKERR | LFOSCGOOD |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 2-129. RIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-9 | RESERVED | R | 0h | |
| 8 | USBFLLERR | R | 0h | USB FLL Reference Clock Error<br>0h = 0<br>1h = 1 |
| 7 | HSCLKGOOD | R | 0h | HSCLK GOOD<br>0h = 0<br>1h = 1 |
| 6 | SYSPLLGOOD | R | 0h | SYSPLL GOOD<br>0h = 0<br>1h = 1 |
| 5 | HFCLKGOOD | R | 0h | HFCLK GOOD<br>0h = 0<br>1h = 1 |
| 4 | LFXTGOOD | R | 0h | LFXT GOOD<br>0h = 0<br>1h = 1 |
| 3 | SRAMSEC | R | 0h | SRAM Single Error Correct<br>0h = 0<br>1h = 1 |
| 2 | FLASHSEC | R | 0h | Flash Single Error Correct<br>0h = 0<br>1h = 1 |
| 1 | ANACLKERR | R | 0h | Analog Clocking Consistency Error<br>0h = 0<br>1h = 1 |
| 0 | LFOSCGOOD | R | 0h | Raw status of the LFOSCGOOD interrupt.<br>0h = No interrupt pending<br>1h = Interrupt pending |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.4 MIS (Offset = 1038h) [Reset = 00000000h]

MIS is shown in Figure 2-116 and described in Table 2-130.

Return to the Summary Table.

SYSCTL masked interrupt status

**Figure 2-116. MIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | USBFLLERR |
| R-0h | | | | | | | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| HSCLKGOOD | SYSPLLGOOD | HFCLKGOOD | LFXTGOOD | SRAMSEC | FLASHSEC | ANACLKERR | LFOSCGOOD |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 2-130. MIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-9 | RESERVED | R | 0h | |
| 8 | USBFLLERR | R | 0h | USB FLL Reference Clock Error<br>0h = 0<br>1h = 1 |
| 7 | HSCLKGOOD | R | 0h | HSCLK GOOD<br>0h = 0<br>1h = 1 |
| 6 | SYSPLLGOOD | R | 0h | SYSPLL GOOD<br>0h = 0<br>1h = 1 |
| 5 | HFCLKGOOD | R | 0h | HFCLK GOOD<br>0h = 0<br>1h = 1 |
| 4 | LFXTGOOD | R | 0h | LFXT GOOD<br>0h = 0<br>1h = 1 |
| 3 | SRAMSEC | R | 0h | SRAM Single Error Correct<br>0h = 0<br>1h = 1 |
| 2 | FLASHSEC | R | 0h | Flash Single Error Correct<br>0h = 0<br>1h = 1 |
| 1 | ANACLKERR | R | 0h | Analog Clocking Consistency Error<br>0h = 0<br>1h = 1 |
| 0 | LFOSCGOOD | R | 0h | Masked status of the LFOSCGOOD interrupt.<br>0h = No interrupt pending<br>1h = Interrupt pending |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.5 ISET (Offset = 1040h) [Reset = 00000000h]

ISET is shown in Figure 2-117 and described in Table 2-131.

Return to the Summary Table.

SYSCTL interrupt set

**Figure 2-117. ISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | USBFLLERR |
| R-0h | | | | | | | W1S-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| HSCLKGOOD | SYSPLLGOOD | HFCLKGOOD | LFXTGOOD | SRAMSEC | FLASHSEC | ANACLKERR | LFOSCGOOD |
| W1S-0h | W1S-0h | W1S-0h | W1S-0h | W1S-0h | W1S-0h | W1S-0h | W1S-0h |

**Table 2-131. ISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-9 | RESERVED | R | 0h | |
| 8 | USBFLLERR | W1S | 0h | USB FLL Reference Clock Error<br>0h = 0<br>1h = 1 |
| 7 | HSCLKGOOD | W1S | 0h | HSCLK GOOD<br>0h = 0<br>1h = 1 |
| 6 | SYSPLLGOOD | W1S | 0h | SYSPLL GOOD<br>0h = 0<br>1h = 1 |
| 5 | HFCLKGOOD | W1S | 0h | HFCLK GOOD<br>0h = 0<br>1h = 1 |
| 4 | LFXTGOOD | W1S | 0h | LFXT GOOD<br>0h = 0<br>1h = 1 |
| 3 | SRAMSEC | W1S | 0h | SRAM Single Error Correct<br>0h = 0<br>1h = 1 |
| 2 | FLASHSEC | W1S | 0h | Flash Single Error Correct<br>0h = 0<br>1h = 1 |
| 1 | ANACLKERR | W1S | 0h | Analog Clocking Consistency Error<br>0h = 0<br>1h = 1 |
| 0 | LFOSCGOOD | W1S | 0h | Set the LFOSCGOOD interrupt.<br>0h = Writing 0h hs no effect<br>1h = Set interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.6 ICLR (Offset = 1048h) [Reset = 00000000h]

ICLR is shown in Figure 2-118 and described in Table 2-132.

Return to the Summary Table.

SYSCTL interrupt clear

**Figure 2-118. ICLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | USBFLLERR |
| R-0h | | | | | | | W1C-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| HSCLKGOOD | SYSPLLGOOD | HFCLKGOOD | LFXTGOOD | SRAMSEC | FLASHSEC | ANACLKERR | LFOSCGOOD |
| W1C-0h | W1C-0h | W1C-0h | W1C-0h | W1C-0h | W1C-0h | W1C-0h | W1C-0h |

**Table 2-132. ICLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-9 | RESERVED | R | 0h | |
| 8 | USBFLLERR | W1C | 0h | USB FLL Reference Clock Error<br>0h = 0<br>1h = 1 |
| 7 | HSCLKGOOD | W1C | 0h | HSCLK GOOD<br>0h = 0<br>1h = 1 |
| 6 | SYSPLLGOOD | W1C | 0h | SYSPLL GOOD<br>0h = 0<br>1h = 1 |
| 5 | HFCLKGOOD | W1C | 0h | HFCLK GOOD<br>0h = 0<br>1h = 1 |
| 4 | LFXTGOOD | W1C | 0h | LFXT GOOD<br>0h = 0<br>1h = 1 |
| 3 | SRAMSEC | W1C | 0h | SRAM Single Error Correct<br>0h = 0<br>1h = 1 |
| 2 | FLASHSEC | W1C | 0h | Flash Single Error Correct<br>0h = 0<br>1h = 1 |
| 1 | ANACLKERR | W1C | 0h | Analog Clocking Consistency Error<br>0h = 0<br>1h = 1 |
| 0 | LFOSCGOOD | W1C | 0h | Clear the LFOSCGOOD interrupt.<br>0h = Writing 0h has no effect<br>1h = Clear interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.7 NMIIIDX (Offset = 1050h) [Reset = 00000000h]

NMIIIDX is shown in Figure 2-119 and described in Table 2-133.

Return to the Summary Table.

NMI interrupt index

**Figure 2-119. NMIIIDX**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | STAT | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | |

**Table 2-133. NMIIIDX Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3-0 | STAT | R | 0h | The NMI interrupt index (NMIIIDX) register generates a value corresponding to the highest priority pending NMI source. This value may be used as an address offset for fast, deterministic handling in the NMI service routine. A read of the NMIIIDX register will clear the corresponding interrupt status in the NMIRIS register.<br>0h = No NMI pending<br>1h = BOR Threshold NMI pending<br>2h = 2<br>3h = 3<br>4h = 4<br>5h = 5<br>6h = 6 |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.8 NMIRIS (Offset = 1060h) [Reset = 00000000h]

NMIRIS is shown in Figure 2-120 and described in Table 2-134.

Return to the Summary Table.

NMI raw interrupt status

**Figure 2-120. NMIRIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | SRAMDED | FLASHDED | LFCLKFAIL | WWDT1 | WWDT0 | BORLVL |
| R-0h | | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 2-134. NMIRIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | RESERVED | R | 0h | |
| 5 | SRAMDED | R | 0h | SRAM Double Error Detect<br>0h = 0<br>1h = 1 |
| 4 | FLASHDED | R | 0h | Flash Double Error Detect<br>0h = 0<br>1h = 1 |
| 3 | LFCLKFAIL | R | 0h | LFXT-EXLF Monitor Fail<br>0h = 0<br>1h = 1 |
| 2 | WWDT1 | R | 0h | Watch Dog 0 Fault<br>0h = 0<br>1h = 1 |
| 1 | WWDT0 | R | 0h | Watch Dog 0 Fault<br>0h = 0<br>1h = 1 |
| 0 | BORLVL | R | 0h | Raw status of the BORLVL NMI<br>0h = No interrupt pending<br>1h = Interrupt pending |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.9 NMIISET (Offset = 1070h) [Reset = 00000000h]

NMIISET is shown in Figure 2-121 and described in Table 2-135.

Return to the Summary Table.

NMI interrupt set

**Figure 2-121. NMIISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | SRAMDED | FLASHDED | LFCLKFAIL | WWDT1 | WWDT0 | BORLVL |
| R-0h | | W1S-0h | W1S-0h | W1S-0h | W1S-0h | W1S-0h | W1S-0h |

**Table 2-135. NMIISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | RESERVED | R | 0h | |
| 5 | SRAMDED | W1S | 0h | SRAM Double Error Detect<br>0h = 0<br>1h = 1 |
| 4 | FLASHDED | W1S | 0h | Flash Double Error Detect<br>0h = 0<br>1h = 1 |
| 3 | LFCLKFAIL | W1S | 0h | LFXT-EXLF Monitor Fail<br>0h = 0<br>1h = 1 |
| 2 | WWDT1 | W1S | 0h | Watch Dog 0 Fault<br>0h = 0<br>1h = 1 |
| 1 | WWDT0 | W1S | 0h | Watch Dog 0 Fault<br>0h = 0<br>1h = 1 |
| 0 | BORLVL | W1S | 0h | Set the BORLVL NMI<br>0h = Writing 0h hs no effect<br>1h = Set interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.10 NMIICLR (Offset = 1078h) [Reset = 00000000h]

NMIICLR is shown in Figure 2-122 and described in Table 2-136.

Return to the Summary Table.

NMI interrupt clear

**Figure 2-122. NMIICLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | SRAMDED | FLASHDED | LFCLKFAIL | WWDT1 | WWDT0 | BORLVL |
| R-0h | | W1C-0h | W1C-0h | W1C-0h | W1C-0h | W1C-0h | W1C-0h |

**Table 2-136. NMIICLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | RESERVED | R | 0h | |
| 5 | SRAMDED | W1C | 0h | SRAM Double Error Detect<br>0h = 0<br>1h = 1 |
| 4 | FLASHDED | W1C | 0h | Flash Double Error Detect<br>0h = 0<br>1h = 1 |
| 3 | LFCLKFAIL | W1C | 0h | LFXT-EXLF Monitor Fail<br>0h = 0<br>1h = 1 |
| 2 | WWDT1 | W1C | 0h | Watch Dog 0 Fault<br>0h = 0<br>1h = 1 |
| 1 | WWDT0 | W1C | 0h | Watch Dog 0 Fault<br>0h = 0<br>1h = 1 |
| 0 | BORLVL | W1C | 0h | Clr the BORLVL NMI<br>0h = Writing 0h hs no effect<br>1h = Clear interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.11 SYSOSCCFG (Offset = 1100h) [Reset = 0002XXXXh]

SYSOSCCFG is shown in Figure 2-123 and described in Table 2-137.

Return to the Summary Table.

SYSOSC configuration

**Figure 2-123. SYSOSCCFG**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | FASTCPUEVENT | BLOCKASYNCALL |
| R-0h | | | | | | R/W-1h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | DISABLE | DISABLESTOP | USE4MHZSTOP |
| R-0h | | | | | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | FREQ | |
| R-0h | | | | | | R/W-0h | |

**Table 2-137. SYSOSCCFG Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-18 | RESERVED | R | 0h | |
| 17 | FASTCPUEVENT | R/W | 1h | FASTCPUEVENT may be used to assert a fast clock request when an interrupt is asserted to the CPU, reducing interrupt latency.<br>0h = An interrupt to the CPU will not assert a fast clock request<br>1h = An interrupt to the CPU will assert a fast clock request |
| 16 | BLOCKASYNCALL | R/W | 0h | BLOCKASYNCALL may be used to mask block all asynchronous fast clock requests, preventing hardware from dynamically changing the active clock configuration when operating in a given mode.<br>0h = Asynchronous fast clock requests are controlled by the requesting peripheral<br>1h = All asynchronous fast clock requests are blocked |
| 15-11 | RESERVED | R | 0h | |
| 10 | DISABLE | R/W | 0h | DISABLE sets the SYSOSC enable/disable policy. SYSOSC may be powered off in RUN, SLEEP, and STOP modes to reduce power consumption. When SYSOSC is disabled, MCLK and ULPCLK are sourced from LFCLK.<br>0h = Do not disable SYSOSC<br>1h = Disable SYSOSC immediately and source MCLK and ULPCLK from LFCLK |
| 9 | DISABLESTOP | R/W | 0h | DISABLESTOP sets the SYSOSC stop mode enable/disable policy. When operating in STOP mode, the SYSOSC may be automatically disabled. When set, ULPCLK will run from LFCLK in STOP mode and SYSOSC will be disabled to reduce power consumption.<br>0h = Do not disable SYSOSC in STOP mode<br>1h = Disable SYSOSC in STOP mode and source ULPCLK from LFCLK |
| 8 | USE4MHZSTOP | R/W | 0h | USE4MHZSTOP sets the SYSOSC stop mode frequency policy. When entering STOP mode, the SYSOSC frequency may be automatically switched to 4MHz to reduce SYSOSC power consumption.<br>0h = Do not gear shift the SYSOSC to 4MHz in STOP mode<br>1h = Gear shift SYSOSC to 4MHz in STOP mode |



Copyright © 2026 Texas Instruments Incorporated

**Table 2-137. SYSOSCCFG Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7-2 | RESERVED | R | 0h | |
| 1-0 | FREQ | R/W | 0h | Target operating frequency for the system oscillator (SYSOSC)<br>0h = Base frequency (32MHz)<br>1h = Low frequency (4MHz)<br>2h = User-trimmed frequency (16 or 24 MHz) |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.12 MCLKCFG (Offset = 1104h) [Reset = 000XX2X0h]

MCLKCFG is shown in Figure 2-124 and described in Table 2-138.

Return to the Summary Table.

Main clock (MCLK) configuration

**Figure 2-124. MCLKCFG**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | MCLKDEADCHK | STOPCLKSTBY | USELFCLK | RESERVED | | | USEHSCLK |
| R-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | | | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | USEMFTICK | FLASHWAIT | | | |
| R-0h | | | R/W-0h | R/W-2h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | UDIV | | MDIV | | | |
| R-0h | | R/W-1h | | R/W-0h | | | |

**Table 2-138. MCLKCFG Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-23 | RESERVED | R | 0h | |
| 22 | MCLKDEADCHK | R/W | 0h | MCLKDEADCHK enables or disables the continuous MCLK dead check monitor. LFCLK must be running before MCLKDEADCHK is enabled.<br>0h = The MCLK dead check monitor is disabled<br>1h = The MCLK dead check monitor is enabled |
| 21 | STOPCLKSTBY | R/W | 0h | STOPCLKSTBY sets the STANDBY mode policy (STANDBY0 or STANDBY1). When set, ULPCLK and LFCLK are disabled to all peripherals in STANDBY mode, with the exception of TIMG0 and TIMG1 which continue to run. Wake-up is only possible via an asynchronous fast clock request.<br>0h = ULPCLK/LFCLK runs to all PD0 peripherals in STANDBY mode<br>1h = ULPCLK/LFCLK is disabled to all peripherals in STANDBY mode except TIMG0 and TIMG1 |
| 20 | USELFCLK | R/W | 0h | USELFCLK sets the MCLK source policy. Set USELFCLK to use LFCLK as the MCLK source. Note that setting USELFCLK does not disable SYSOSC, and SYSOSC remains available for direct use by analog modules.<br>0h = MCLK will not use the low frequency clock (LFCLK)<br>1h = MCLK will use the low frequency clock (LFCLK) |
| 19-17 | RESERVED | R | 0h | |
| 16 | USEHSCLK | R/W | 0h | USEHSCLK, together with USELFCLK, sets the MCLK source policy. Set USEHSCLK to use HSCLK (HFCLK or SYSPLL) as the MCLK source in RUN and SLEEP modes.<br>0h = MCLK will not use the high speed clock (HSCLK)<br>1h = MCLK will use the high speed clock (HSCLK) in RUN and SLEEP mode |
| 15-13 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

**Table 2-138. MCLKCFG Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 12 | USEMFTICK | R/W | 0h | USEMFTICK specifies whether the 4MHz constant-rate clock (MFCLK) to peripherals is enabled or disabled. When enabled, MDIV must be disabled (set to 0h=/1).<br>0h = The 4MHz rate MFCLK to peripherals is enabled<br>1h = The 4MHz rate MFCLK to peripherals is enabled. |
| 11-8 | FLASHWAIT | R/W | 2h | FLASHWAIT specifies the number of flash wait states when MCLK is sourced from HSCLK. FLASHWAIT has no effect when MCLK is sourced from SYSOSC or LFCLK.<br>0h = No flash wait states are applied<br>1h = One flash wait state is applied<br>2h = 2 flash wait states are applied |
| 7-6 | RESERVED | R | 0h | |
| 5-4 | UDIV | R/W | 1h | UDIV specifies the ULPCLK divider when MCLK is sourced from HSCLK. UDIV has no effect when MCLK is sourced from SYSOSC or LFCLK.<br>0h = ULPCLK is not divided and is equal to MCLK<br>1h = ULPCLK is MCLK/2 (divided-by-2) |
| 3-0 | MDIV | R/W | 0h | MDIV may be used to divide the MCLK frequency when MCLK is sourced from SYSOSC. MDIV=0h corresponds to /1 (no divider). MDIV=1h corresponds to /2 (divide-by-2). MDIV=Fh corresponds to /16 (divide-by-16). MDIV may be set between /1 and /16 on an integer basis. |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.13 HSCLKEN (Offset = 1108h) [Reset = 0000XXXXh]

HSCLKEN is shown in Figure 2-125 and described in Table 2-139.

Return to the Summary Table.

High-speed clock (HSCLK) source enable/disable

**Figure 2-125. HSCLKEN**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | USEEXTHFCLK |
| R-0h | | | | | | | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | SYSPLLEN |
| R-0h | | | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | HFXTEN |
| R-0h | | | | | | | R/W-0h |

**Table 2-139. HSCLKEN Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-17 | RESERVED | R | 0h | |
| 16 | USEEXTHFCLK | R/W | 0h | USEEXTHFCLK selects the HFCLK_IN digital clock input to be the source for HFCLK. When disabled, HFXT is the HFCLK source and HFXTEN may be set. Do not set HFXTEN and USEEXTHFCLK simultaneously.<br>0h = Use HFXT as the HFCLK source<br>1h = Use the HFCLK_IN digital clock input as the HFCLK source |
| 15-9 | RESERVED | R | 0h | |
| 8 | SYSPLLEN | R/W | 0h | SYSPLLEN enables or disables the system phase-lock loop (SYSPLL).<br>0h = Disable the SYSPLL<br>1h = Enable the SYSPLL |
| 7-1 | RESERVED | R | 0h | |
| 0 | HFXTEN | R/W | 0h | HFXTEN enables or disables the high frequency crystal oscillator (HFXT).<br>0h = Disable the HFXT<br>1h = Enable the HFXT |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.14 HSCLKCFG (Offset = 110Ch) [Reset = 00000000h]

HSCLKCFG is shown in Figure 2-126 and described in Table 2-140.

Return to the Summary Table.

High-speed clock (HSCLK) source selection

**Figure 2-126. HSCLKCFG**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | USBFLL4HSCLK | HSCLKSEL |
| R-0h | | | | | | R/W-0h | R/W-0h |

**Table 2-140. HSCLKCFG Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | |
| 1 | USBFLL4HSCLK | R/W | 0h | USBFLL4HSCLK selects the USBFLL as High Speed clock source<br>0h = HSCLK is sourced from the HSCLKCFG.HSCLKSEL<br>1h = HSCLK is sourced from the USBFLL |
| 0 | HSCLKSEL | R/W | 0h | HSCLKSEL selects the HSCLK source (SYSPLL or HFCLK).<br>0h = HSCLK is sourced from the SYSPLL<br>1h = HSCLK is sourced from the HFCLK |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.15 HFCLKCLKCFG (Offset = 1110h) [Reset = 1XXXXX00h]

HFCLKCLKCFG is shown in Figure 2-127 and described in Table 2-141.

Return to the Summary Table.

High-frequency clock (HFCLK) configuration

**Figure 2-127. HFCLKCLKCFG**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | HFCLKFLTCHK | RESERVED | | | |
| R-0h | | | R/W-1h | R-0h | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | HFXTRSEL | | RESERVED | | | |
| R-0h | | R/W-0h | | R-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| HFXTTIME | | | | | | | |
| R/W-0h | | | | | | | |

**Table 2-141. HFCLKCLKCFG Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | RESERVED | R | 0h | |
| 28 | HFCLKFLTCHK | R/W | 1h | HFCLKFLTCHK enables or disables the HFCLK startup monitor.<br>0h = HFCLK startup is not checked<br>1h = HFCLK startup is checked |
| 27-14 | RESERVED | R | 0h | |
| 13-12 | HFXTRSEL | R/W | 0h | HFXT Range Select<br>0h = 4MHz <= HFXT frequency <= 8MHz<br>1h = 8MHz < HFXT frequency <= 16MHz<br>2h = 16MHz < HFXT frequency <= 32MHz<br>3h = 32MHz < HFXT frequency <= 48MHz |
| 11-8 | RESERVED | R | 0h | |
| 7-0 | HFXTTIME | R/W | 0h | HFXTTIME specifies the HFXT startup time in 64us resolution. If the HFCLK startup monitor is enabled (HFCLKFLTCHK), HFXT will be checked after this time expires.<br>0h = Minimum startup time (approximatly zero)<br>FFh = Maximum startup time (approximatly 16.32ms) |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.16 LFCLKCFG (Offset = 1114h) [Reset = 000000XXh]

LFCLKCFG is shown in Figure 2-128 and described in Table 2-142.

Return to the Summary Table.

Low frequency crystal oscillator (LFXT) configuration

**Figure 2-128. LFCLKCFG**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | LOWCAP |
| R-0h | | | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | MONITOR | RESERVED | | XT1DRIVE | |
| R-0h | | | R/W-0h | R-0h | | R/W-3h | |

**Table 2-142. LFCLKCFG Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-9 | RESERVED | R | 0h | |
| 8 | LOWCAP | R/W | 0h | LOWCAP controls the low-power LFXT mode. When the LFXT load capacitance is less than 3pf, LOWCAP may be set for reduced power consumption.<br>0h = LFXT low capacitance mode is disabled<br>1h = LFXT low capacitance mode is enabled |
| 7-5 | RESERVED | R | 0h | |
| 4 | MONITOR | R/W | 0h | MONITOR enables or disables the LFCLK monitor, which continuously checks LFXT or LFCLK_IN for a clock stuck fault.<br>0h = Clock monitor is disabled<br>1h = Clock monitor is enabled |
| 3-2 | RESERVED | R | 0h | |
| 1-0 | XT1DRIVE | R/W | 3h | XT1DRIVE selects the low frequency crystal oscillator (LFXT) drive strength.<br>0h = Lowest drive and current<br>1h = Lower drive and current<br>2h = Higher drive and current<br>3h = Highest drive and current |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.17 SYSPLLCFG0 (Offset = 1120h) [Reset = 000000XXh]

SYSPLLCFG0 is shown in Figure 2-129 and described in Table 2-143.

Return to the Summary Table.

SYSPLL reference and output configuration

**Figure 2-129. SYSPLLCFG0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | RDIVCLK2X | | | |
| R-0h | | | | R/W-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RDIVCLK1 | | | | RDIVCLK0 | | | |
| R/W-0h | | | | R/W-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | ENABLECLK2X | ENABLECLK1 | ENABLECLK0 | RESERVED | | MCLK2XVCO | SYSPLLREF |
| R-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | | R/W-0h | R/W-0h |

**Table 2-143. SYSPLLCFG0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | RESERVED | R | 0h | |
| 19-16 | RDIVCLK2X | R/W | 0h | RDIVCLK2X sets the final divider for the SYSPLLCLK2X output.<br>0h = SYSPLLCLK1 is divided by 1<br>1h = SYSPLLCLK1 is divided by 2<br>2h = SYSPLLCLK1 is divided by 3<br>3h = SYSPLLCLK1 is divided by 4<br>4h = SYSPLLCLK1 is divided by 5<br>5h = SYSPLLCLK1 is divided by 6<br>6h = SYSPLLCLK1 is divided by 7<br>7h = SYSPLLCLK1 is divided by 8<br>8h = SYSPLLCLK1 is divided by 9<br>9h = SYSPLLCLK1 is divided by 10<br>Ah = SYSPLLCLK1 is divided by 11<br>Bh = SYSPLLCLK1 is divided by 12<br>Ch = SYSPLLCLK1 is divided by 13<br>Dh = SYSPLLCLK1 is divided by 14<br>Eh = SYSPLLCLK1 is divided by 15<br>Fh = SYSPLLCLK1 is divided by 16 |
| 15-12 | RDIVCLK1 | R/W | 0h | RDIVCLK1 sets the final divider for the SYSPLLCLK1 output.<br>0h = SYSPLLCLK1 is divided by 2<br>1h = SYSPLLCLK1 is divided by 4<br>2h = SYSPLLCLK1 is divided by 6<br>3h = SYSPLLCLK1 is divided by 8<br>4h = SYSPLLCLK1 is divided by 10<br>5h = SYSPLLCLK1 is divided by 12<br>6h = SYSPLLCLK1 is divided by 14<br>7h = SYSPLLCLK1 is divided by 16<br>8h = SYSPLLCLK1 is divided by 18<br>9h = SYSPLLCLK1 is divided by 20<br>Ah = SYSPLLCLK1 is divided by 22<br>Bh = SYSPLLCLK1 is divided by 24<br>Ch = SYSPLLCLK1 is divided by 26<br>Dh = SYSPLLCLK1 is divided by 28<br>Eh = SYSPLLCLK1 is divided by 30<br>Fh = SYSPLLCLK1 is divided by 32 |



Copyright © 2026 Texas Instruments Incorporated

**Table 2-143. SYSPLLCFG0 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 11-8 | RDIVCLK0 | R/W | 0h | RDIVCLK0 sets the final divider for the SYSPLLCLK0 output.<br>0h = SYSPLLCLK0 is divided by 2<br>1h = SYSPLLCLK0 is divided by 4<br>2h = SYSPLLCLK0 is divided by 6<br>3h = SYSPLLCLK0 is divided by 8<br>4h = SYSPLLCLK0 is divided by 10<br>5h = SYSPLLCLK0 is divided by 12<br>6h = SYSPLLCLK0 is divided by 14<br>7h = SYSPLLCLK0 is divided by 16<br>8h = SYSPLLCLK0 is divided by 18<br>9h = SYSPLLCLK0 is divided by 20<br>Ah = SYSPLLCLK0 is divided by 22<br>Bh = SYSPLLCLK0 is divided by 24<br>Ch = SYSPLLCLK0 is divided by 26<br>Dh = SYSPLLCLK0 is divided by 28<br>Eh = SYSPLLCLK0 is divided by 30<br>Fh = SYSPLLCLK0 is divided by 32 |
| 7 | RESERVED | R | 0h | |
| 6 | ENABLECLK2X | R/W | 0h | ENABLECLK2X enables or disables the SYSPLLCLK2X output.<br>0h = SYSPLLCLK2X is disabled<br>1h = SYSPLLCLK2X is enabled |
| 5 | ENABLECLK1 | R/W | 0h | ENABLECLK1 enables or disables the SYSPLLCLK1 output.<br>0h = SYSPLLCLK1 is disabled<br>1h = SYSPLLCLK1 is enabled |
| 4 | ENABLECLK0 | R/W | 0h | ENABLECLK0 enables or disables the SYSPLLCLK0 output.<br>0h = SYSPLLCLK0 is disabled<br>1h = SYSPLLCLK0 is enabled |
| 3-2 | RESERVED | R | 0h | |
| 1 | MCLK2XVCO | R/W | 0h | MCLK2XVCO selects the SYSPLL output which is sent to the HSCLK mux for use by MCLK.<br>0h = The SYSPLLCLK0 output is sent to the HSCLK mux<br>1h = The SYSPLLCLK2X output is sent to the HSCLK mux |
| 0 | SYSPLLREF | R/W | 0h | SYSPLLREF selects the system PLL (SYSPLL) reference clock source.<br>0h = SYSPLL reference is SYSOSC<br>1h = SYSPLL reference is HFCLK |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.18 SYSPLLCFG1 (Offset = 1124h) [Reset = 000000XXh]

SYSPLLCFG1 is shown in Figure 2-130 and described in Table 2-144.

Return to the Summary Table.

SYSPLL reference and feedback divider

**Figure 2-130. SYSPLLCFG1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | QDIV | | | | | | | RESERVED | | | | | | PDIV | |
| R-0h | R/W-0h | | | | | | | R-0h | | | | | | R/W-0h | |

**Table 2-144. SYSPLLCFG1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-15 | RESERVED | R | 0h | |
| 14-8 | QDIV | R/W | 0h | QDIV selects the SYSPLL feedback path divider.<br>0h = Divide-by-one is not a valid QDIV option<br>1h = Feedback path is divided by 2<br>7Eh = Feedback path is divided by 127 (0x7E) |
| 7-2 | RESERVED | R | 0h | |
| 1-0 | PDIV | R/W | 0h | PDIV selects the SYSPLL reference clock prescale divider.<br>0h = SYSPLLREF is not divided<br>1h = SYSPLLREF is divided by 2<br>2h = SYSPLLREF is divided by 4<br>3h = SYSPLLREF is divided by 8 |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.19 SYSPLLPARAM0 (Offset = 1128h) [Reset = X0X0X0X0h]

SYSPLLPARAM0 is shown in Figure 2-131 and described in Table 2-145.

Return to the Summary Table.

SYSPLL PARAM0 (load from FACTORY region)

**Figure 2-131. SYSPLLPARAM0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CAPBOVERRIDE | RESERVED | | CAPBVAL | | | | |
| R/W-0h | R-0h | | R/W-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | CPCURRENT | | | | | |
| R-0h | | R/W-0h | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | STARTTIMELP | | | | | |
| R-0h | | R/W-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | STARTTIME | | | | | |
| R-0h | | R/W-0h | | | | | |

**Table 2-145. SYSPLLPARAM0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | CAPBOVERRIDE | R/W | 0h | CAPBOVERRIDE controls the override for Cap B<br>0h = Cap B override disabled<br>1h = Cap B override enabled |
| 30-29 | RESERVED | R | 0h | |
| 28-24 | CAPBVAL | R/W | 0h | Override value for Cap B |
| 23-22 | RESERVED | R | 0h | |
| 21-16 | CPCURRENT | R/W | 0h | Charge pump current |
| 15-14 | RESERVED | R | 0h | |
| 13-8 | STARTTIMELP | R/W | 0h | Startup time from low power mode exit to locked clock, in 1us resolution |
| 7-6 | RESERVED | R | 0h | |
| 5-0 | STARTTIME | R/W | 0h | Startup time from enable to locked clock, in 1us resolution |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.20 SYSPLLPARAM1 (Offset = 112Ch) [Reset = 00XX00X0h]

SYSPLLPARAM1 is shown in Figure 2-132 and described in Table 2-146.

Return to the Summary Table.

SYSPLL PARAM1 (load from FACTORY region)

**Figure 2-132. SYSPLLPARAM1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LPFRESC | | | | | | | | RESERVED | | | | | | LPFRESA | |
| R/W-0h | | | | | | | | R-0h | | | | | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LPFRESA | | | | | | | | RESERVED | | | LPFCAPA | | | | |
| R/W-0h | | | | | | | | R-0h | | | R/W-0h | | | | |

**Table 2-146. SYSPLLPARAM1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | LPFRESC | R/W | 0h | Loop filter Res C |
| 23-18 | RESERVED | R | 0h | |
| 17-8 | LPFRESA | R/W | 0h | Loop filter Res A |
| 7-5 | RESERVED | R | 0h | |
| 4-0 | LPFCAPA | R/W | 0h | Loop filter Cap A |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.21 GENCLKCFG (Offset = 1138h) [Reset = 00000X0Xh]

GENCLKCFG is shown in Figure 2-133 and described in Table 2-147.

Return to the Summary Table.

General clock configuration

**Figure 2-133. GENCLKCFG**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | FCCLFCLKSRC | FCCTRIGCNT | | | | |
| R-0h | | R/W-0h | R/W-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| ANACPUMPCFG | | FCCLVLTRIG | FCCTRIGSRC | FCCSELCLK | | | |
| R/W-0h | | R/W-0h | R/W-0h | R/W-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| HFCLK4MFPCLKDIV | | | | RESERVED | USBCLKSRC | MFPCLKSRC | RESERVED |
| R/W-0h | | | | R-0h | R/W-0h | R/W-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| EXCLKDIVEN | EXCLKDIVVAL | | | RESERVED | EXCLKSRC | | |
| R/W-0h | R/W-0h | | | R-0h | R/W-0h | | |

**Table 2-147. GENCLKCFG Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | RESERVED | R | 0h | |
| 29 | FCCLFCLKSRC | R/W | 0h | FCCLFCLKSRC selects between SYSTEM LFCLK and EXTERNAL SOURCED LFCLK. |
| 28-24 | FCCTRIGCNT | R/W | 0h | FCCTRIGCNT specifies the number of trigger clock periods in the trigger window. FCCTRIGCNT=0h (one trigger clock period) up to 1Fh (32 trigger clock periods) may be specified. |
| 23-22 | ANACPUMPCFG | R/W | 0h | ANACPUMPCFG selects the analog mux charge pump (VBOOST) enable method.<br>0h = VBOOST is enabled on request from a COMP, GPAMP, or OPA<br>1h = VBOOST is enabled when the device is in RUN or SLEEP mode, or when a COMP/GPAMP/OPA is enabled<br>2h = VBOOST is always enabled |
| 21 | FCCLVLTRIG | R/W | 0h | FCCLVLTRIG selects the frequency clock counter (FCC) trigger mode.<br>0h = Rising edge to rising edge triggered<br>1h = Level triggered |
| 20 | FCCTRIGSRC | R/W | 0h | FCCTRIGSRC selects the frequency clock counter (FCC) trigger source.<br>0h = FCC trigger is the external pin<br>1h = FCC trigger is the LFCLK |
| 19-16 | FCCSELCLK | R/W | 0h | FCCSELCLK selectes the frequency clock counter (FCC) clock source.<br>0h = FCC clock is MCLK<br>1h = FCC clock is SYSOSC<br>2h = FCC clock is HFCLK<br>3h = FCC clock is the CLK_OUT selection<br>4h = FCC clock is SYSPLLCLK0<br>5h = FCC clock is SYSPLLCLK1<br>6h = FCC clock is SYSPLLCLK2X<br>7h = FCC clock is the FCCIN external input<br>8h = FCC clock is USBFLL |



Copyright © 2026 Texas Instruments Incorporated

**Table 2-147. GENCLKCFG Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15-12 | HFCLK4MFPCLKDIV | R/W | 0h | HFCLK4MFPCLKDIV selects the divider applied to HFCLK when HFCLK is used as the MFPCLK source. Integer dividers from /1 to /16 may be selected.<br>0h = HFCLK is not divided before being used for MFPCLK<br>1h = HFCLK is divided by 2 before being used for MFPCLK<br>2h = HFCLK is divided by 3 before being used for MFPCLK<br>3h = HFCLK is divided by 4 before being used for MFPCLK<br>4h = HFCLK is divided by 5 before being used for MFPCLK<br>5h = HFCLK is divided by 6 before being used for MFPCLK<br>6h = HFCLK is divided by 7 before being used for MFPCLK<br>7h = HFCLK is divided by 8 before being used for MFPCLK<br>8h = HFCLK is divided by 9 before being used for MFPCLK<br>9h = HFCLK is divided by 10 before being used for MFPCLK<br>Ah = HFCLK is divided by 11 before being used for MFPCLK<br>Bh = HFCLK is divided by 12 before being used for MFPCLK<br>Ch = HFCLK is divided by 13 before being used for MFPCLK<br>Dh = HFCLK is divided by 14 before being used for MFPCLK<br>Eh = HFCLK is divided by 15 before being used for MFPCLK<br>Fh = HFCLK is divided by 16 before being used for MFPCLK |
| 11 | RESERVED | R | 0h | |
| 10 | USBCLKSRC | R/W | 0h | USBCLKSRC selects the USBCLK source.<br>0h = USBCLK source is USBFLL<br>1h = USBCLK source is SYSPLLCLK1 |
| 9 | MFPCLKSRC | R/W | 0h | MFPCLKSRC selects the MFPCLK (middle frequency precision clock) source.<br>0h = MFPCLK is sourced from SYSOSC<br>1h = MFPCLK is sourced from HFCLK |
| 8 | RESERVED | R | 0h | |
| 7 | EXCLKDIVEN | R/W | 0h | EXCLKDIVEN enables or disables the divider function of the CLK_OUT external clock output block.<br>0h = CLock divider is disabled (passthrough, EXCLKDIVVAL is not applied)<br>1h = Clock divider is enabled (EXCLKDIVVAL is applied) |
| 6-4 | EXCLKDIVVAL | R/W | 0h | EXCLKDIVVAL selects the divider value for the divider in the CLK_OUT external clock output block.<br>0h = CLK_OUT source is divided by 2<br>1h = CLK_OUT source is divided by 4<br>2h = CLK_OUT source is divided by 6<br>3h = CLK_OUT source is divided by 8<br>4h = CLK_OUT source is divided by 10<br>5h = CLK_OUT source is divided by 12<br>6h = CLK_OUT source is divided by 14<br>7h = CLK_OUT source is divided by 16 |
| 3 | RESERVED | R | 0h | |
| 2-0 | EXCLKSRC | R/W | 0h | EXCLKSRC selects the source for the CLK_OUT external clock output block. ULPCLK and MFPCLK require the CLK_OUT divider (EXCLKDIVEN) to be enabled<br>0h = CLK_OUT is SYSOSC<br>1h = CLK_OUT is ULPCLK (EXCLKDIVEN must be enabled)<br>2h = CLK_OUT is LFCLK<br>3h = CLK_OUT is MFPCLK (EXCLKDIVEN must be enabled)<br>4h = CLK_OUT is HFCLK<br>5h = CLK_OUT is SYSPLLCLK1 (SYSPLLCLK1 must be <=48MHz)<br>6h = CLK_OUT is USBFLL (At 60Mhz must divide for external output) |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.22 GENCLKEN (Offset = 113Ch) [Reset = 0000000Xh]

GENCLKEN is shown in Figure 2-134 and described in Table 2-148.

Return to the Summary Table.

General clock enable control

**Figure 2-134. GENCLKEN**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | MFPCLKEN | RESERVED | | | EXCLKEN |
| R-0h | | | R/W-0h | R-0h | | | R/W-0h |

**Table 2-148. GENCLKEN Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-5 | RESERVED | R | 0h | |
| 4 | MFPCLKEN | R/W | 0h | MFPCLKEN enables the middle frequency precision clock (MFPCLK).<br>0h = MFPCLK is disabled<br>1h = MFPCLK is enabled |
| 3-1 | RESERVED | R | 0h | |
| 0 | EXCLKEN | R/W | 0h | EXCLKEN enables the CLK_OUT external clock output block.<br>0h = CLK_OUT block is disabled<br>1h = CLK_OUT block is enabled |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.23 PMODECFG (Offset = 1140h) [Reset = 00000000h]

PMODECFG is shown in Figure 2-135 and described in Table 2-149.

Return to the Summary Table.

Power mode configuration

**Figure 2-135. PMODECFG**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | | | DSLEEP | |
| R-0h | | | | | | | | | | | | | | R/W-0h | |

**Table 2-149. PMODECFG Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | |
| 1-0 | DSLEEP | R/W | 0h | DSLEEP selects the operating mode to enter upon a DEEPSLEEP request from the CPU.<br>0h = STOP mode is entered<br>1h = STANDBY mode is entered<br>2h = SHUTDOWN mode is entered<br>3h = Reserved |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.24 FCC (Offset = 1150h) [Reset = 00000000h]

FCC is shown in Figure 2-136 and described in Table 2-150.

Return to the Summary Table.

Frequency clock counter (FCC) count

**Figure 2-136. FCC**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | DATA | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | R-0h | | | | | | | | | | | | | | | | | | | | | |

**Table 2-150. FCC Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-22 | RESERVED | R | 0h | |
| 21-0 | DATA | R | 0h | Frequency clock counter (FCC) count value. |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.25 SYSOSCTRIMUSER (Offset = 1170h) [Reset = 0000X0XXh]

SYSOSCTRIMUSER is shown in Figure 2-137 and described in Table 2-151.

Return to the Summary Table.

SYSOSC user-specified trim

**Figure 2-137. SYSOSCTRIMUSER**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | RDIV | | | | |
| R-0h | | | R/W-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RDIV | | | | RESFINE | | | |
| R/W-0h | | | | R/W-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | RESCOARSE | | | | | |
| R-0h | | R/W-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | CAP | | | RESERVED | | FREQ | |
| R-0h | R/W-0h | | | R-0h | | R/W-0h | |

**Table 2-151. SYSOSCTRIMUSER Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | RESERVED | R | 0h | |
| 28-20 | RDIV | R/W | 0h | RDIV specifies the frequency correction loop (FCL) resistor trim. This value changes with the target frequency. |
| 19-16 | RESFINE | R/W | 0h | RESFINE specifies the resister fine trim. This value changes with the target frequency. |
| 15-14 | RESERVED | R | 0h | |
| 13-8 | RESCOARSE | R/W | 0h | RESCOARSE specifies the resister coarse trim. This value changes with the target frequency. |
| 7 | RESERVED | R | 0h | |
| 6-4 | CAP | R/W | 0h | CAP specifies the SYSOSC capacitor trim. This value changes with the target frequency. |
| 3-2 | RESERVED | R | 0h | |
| 1-0 | FREQ | R/W | 0h | FREQ specifies the target user-trimmed frequency for SYSOSC.<br>0h = Reserved<br>1h = 16MHz user frequency<br>2h = 24MHz user frequency<br>3h = Reserved |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.26 SRAMBOUNDARY (Offset = 1178h) [Reset = 000000XXh]

SRAMBOUNDARY is shown in Figure 2-138 and described in Table 2-152.

Return to the Summary Table.

SRAM Write Boundary

**Figure 2-138. SRAMBOUNDARY**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | ADDR | | | | | | | | | | | | | | | RESERVED | | | | |
| R-0h | | | | | | | | | | | | R/W-0h | | | | | | | | | | | | | | | R-0h | | | | |

**Table 2-152. SRAMBOUNDARY Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | RESERVED | R | 0h | |
| 19-5 | ADDR | R/W | 0h | SRAM boundary configuration. The value configured into this acts such that: SRAM accesses to addresses less than or equal value will be RW only. SRAM accesses to addresses greater than value will be RX only. Value of 0 is not valid (system will have no stack). If set to 0, the system acts as if the entire SRAM is RWX. Any non-zero value can be configured, including a value = SRAM size. |
| 4-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.27 SYSTEMCFG (Offset = 1180h) [Reset = 00XXXXXXh]

SYSTEMCFG is shown in Figure 2-139 and described in Table 2-153.

Return to the Summary Table.

System configuration

**Figure 2-139. SYSTEMCFG**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | FLASHECCRSTDIS | WWDTLP1RSTDIS | WWDTLP0RSTDIS |
| R-0h | | | | | R/W-1h | R/W-1h | R/W-0h |

**Table 2-153. SYSTEMCFG Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | The key value of 1Bh (27) must be written to KEY together with contents to be updated. Reads as 0<br>1Bh = Issue write |
| 23-3 | RESERVED | R | 0h | |
| 2 | FLASHECCRSTDIS | R/W | 1h | FLASHECCRSTDIS specifies whether a flash ECC double error detect (DED) will trigger a SYSRST or an NMI.<br>0h = Flash ECC DED will trigger a SYSRST<br>1h = Flash ECC DED will trigger a NMI |
| 1 | WWDTLP1RSTDIS | R/W | 1h | WWDTLP1RSTDIS specifies whether a WWDT Error Event will trigger a SYSRST or an NMI.<br>0h = WWDTLP1 Error Event will trigger a SYSRST<br>1h = WWDTLP1 Error Event will trigger an NMI |
| 0 | WWDTLP0RSTDIS | R/W | 0h | WWDTLP0RSTDIS specifies whether a WWDT Error Event will trigger a BOOTRST or an NMI.<br>0h = WWDTLP0 Error Event will trigger a BOOTRST<br>1h = WWDTLP0 Error Event will trigger an NMI |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.28 WRITELOCK (Offset = 1200h) [Reset = 00000000h]

WRITELOCK is shown in Figure 2-140 and described in Table 2-154.

Return to the Summary Table.

SYSCTL register write lockout

**Figure 2-140. WRITELOCK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | ACTIVE |
| R-0h | | | | | | | R/W-0h |

**Table 2-154. WRITELOCK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | ACTIVE | R/W | 0h | ACTIVE controls whether critical SYSCTL registers are write protected or not.<br>0h = Allow writes to lockable registers<br>1h = Disallow writes to lockable registers |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.29 CLKSTATUS (Offset = 1204h) [Reset = XXXXX0XXh]

CLKSTATUS is shown in Figure 2-141 and described in Table 2-155.

Return to the Summary Table.

Clock module (CKM) status

**Figure 2-141. CLKSTATUS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| ANACLKERR | RESERVED | SYSPLLBLKUPD | HFCLKBLKUPD | RESERVED | | FCCDONE | FCLMODE |
| R-0h | R-0h | R-0h | R-0h | R-0h | | R-0h | R-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| LFCLKFAIL | RESERVED | HSCLKGOOD | HSCLKDEAD | RESERVED | | CURMCLKSEL | CURHSCLKSEL |
| R-0h | R-0h | R-0h | R-0h | R-0h | | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | SYSPLLOFF | HFCLKOFF | HSCLKSOFF | LFOSCGOOD | LFXTGOOD | SYSPLLGOOD | HFCLKGOOD |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LFCLKMUX | | RESERVED | HSCLKMUX | RESERVED | | SYSOSCFREQ | |
| R-0h | | R-0h | R-0h | R-0h | | R-0h | |

**Table 2-155. CLKSTATUS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | ANACLKERR | R | 0h | ANACLKERR is set when the device clock configuration does not support an enabled analog peripheral mode and the analog peripheral may not be functioning as expected.<br>0h = No analog clock errors detected<br>1h = Analog clock error detected |
| 30 | RESERVED | R | 0h | |
| 29 | SYSPLLBLKUPD | R | 0h | SYSPLLBLKUPD indicates when writes to SYSPLLCFG0/1 and SYSPLLPARAM0/1 are blocked.<br>0h = writes to SYSPLLCFG0/1 and SYSPLLPARAM0/1 are allowed<br>1h = writes to SYSPLLCFG0/1 and SYSPLLPARAM0/1 are blocked |
| 28 | HFCLKBLKUPD | R | 0h | HFCLKBLKUPD indicates when writes to the HFCLKCLKCFG register are blocked.<br>0h = Writes to HFCLKCLKCFG are allowed<br>1h = Writes to HFCLKCLKCFG are blocked |
| 27-26 | RESERVED | R | 0h | |
| 25 | FCCDONE | R | 0h | FCCDONE indicates when a frequency clock counter capture is complete.<br>0h = FCC capture is not done<br>1h = FCC capture is done |
| 24 | FCLMODE | R | 0h | FCLMODE indicates if the SYSOSC frequency correction loop (FCL) is enabled.<br>0h = SYSOSC FCL is disabled<br>1h = SYSOSC FCL is enabled |
| 23 | LFCLKFAIL | R | 0h | LFCLKFAIL indicates when the continous LFCLK monitor detects a LFXT or LFCLK_IN clock stuck failure.<br>0h = No LFCLK fault detected<br>1h = LFCLK stuck fault detected |
| 22 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

**Table 2-155. CLKSTATUS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 21 | HSCLKGOOD | R | 0h | HSCLKGOOD is set by hardware if the selected clock source for HSCLK started successfully.<br>0h = The HSCLK source did not start correctly<br>1h = The HSCLK source started correctly |
| 20 | HSCLKDEAD | R | 0h | HSCLKDEAD is set by hardware if the selected source for HSCLK was started but did not start successfully.<br>0h = The HSCLK source was not started or started correctly<br>1h = The HSCLK source did not start correctly |
| 19-18 | RESERVED | R | 0h | |
| 17 | CURMCLKSEL | R | 0h | CURMCLKSEL indicates if MCLK is currently sourced from LFCLK.<br>0h = MCLK is not sourced from LFCLK<br>1h = MCLK is sourced from LFCLK |
| 16 | CURHSCLKSEL | R | 0h | CURHSCLKSEL indicates the current clock source for HSCLK.<br>0h = HSCLK is currently sourced from the SYSPLL<br>1h = HSCLK is currently sourced from the HFCLK |
| 15 | RESERVED | R | 0h | |
| 14 | SYSPLLOFF | R | 0h | SYSPLLOFF indicates if the SYSPLL is disabled or was dead at startup. When the SYSPLL is started, SYSPLLOFF is cleared by hardware. Following startup of the SYSPLL, if the SYSPLL startup monitor determines that the SYSPLL was not started correctly, SYSPLLOFF is set.<br>0h = SYSPLL started correctly and is enabled<br>1h = SYSPLL is disabled or was dead startup |
| 13 | HFCLKOFF | R | 0h | HFCLKOFF indicates if the HFCLK is disabled or was dead at startup. When the HFCLK is started, HFCLKOFF is cleared by hardware. Following startup of the HFCLK, if the HFCLK startup monitor determines that the HFCLK was not started correctly, HFCLKOFF is set.<br>0h = HFCLK started correctly and is enabled<br>1h = HFCLK is disabled or was dead at startup |
| 12 | HSCLKSOFF | R | 0h | HSCLKSOFF is set when the high speed clock sources (SYSPLL, HFCLK) are disabled or dead. It is the logical AND of HFCLKOFF and SYSPLLOFF.<br>0h = SYSPLL, HFCLK, or both were started correctly and remain enabled<br>1h = SYSPLL and HFCLK are both either off or dead |
| 11 | LFOSCGOOD | R | 0h | LFOSCGOOD indicates when the LFOSC startup has completed and the LFOSC is ready for use.<br>0h = LFOSC is not ready<br>1h = LFOSC is ready |
| 10 | LFXTGOOD | R | 0h | LFXTGOOD indicates if the LFXT started correctly. When the LFXT is started, LFXTGOOD is cleared by hardware. After the startup settling time has expired, the LFXT status is tested. If the LFXT started successfully the LFXTGOOD bit is set, else it is left cleared.<br>0h = LFXT did not start correctly<br>1h = LFXT started correctly |
| 9 | SYSPLLGOOD | R | 0h | SYSPLLGOOD indicates if the SYSPLL started correctly. When the SYSPLL is started, SYSPLLGOOD is cleared by hardware. After the startup settling time has expired, the SYSPLL status is tested. If the SYSPLL started successfully the SYSPLLGOOD bit is set, else it is left cleared.<br>0h = SYSPLL did not start correctly<br>1h = SYSPLL started correctly |
| 8 | HFCLKGOOD | R | 0h | HFCLKGOOD indicates that the HFCLK started correctly. When the HFXT is started or HFCLK_IN is selected as the HFCLK source, this bit will be set by hardware if a valid HFCLK is detected, and cleared if HFCLK is not operating within the expected range.<br>0h = HFCLK did not start correctly<br>1h = HFCLK started correctly |



Copyright © 2026 Texas Instruments Incorporated

**Table 2-155. CLKSTATUS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7-6 | LFCLKMUX | R | 0h | LFCLKMUX indicates if LFCLK is sourced from the internal LFOSC, the low frequency crystal (LFXT), or the LFCLK_IN digital clock input.<br>0h = LFCLK is sourced from the internal LFOSC<br>1h = LFCLK is sourced from the LFXT (crystal)<br>2h = LFCLK is sourced from LFCLK_IN (external digital clock input) |
| 5 | RESERVED | R | 0h | |
| 4 | HSCLKMUX | R | 0h | HSCLKMUX indicates if MCLK is currently sourced from the high-speed clock (HSCLK).<br>0h = MCLK is not sourced from HSCLK<br>1h = MCLK is sourced from HSCLK |
| 3-2 | RESERVED | R | 0h | |
| 1-0 | SYSOSCFREQ | R | 0h | SYSOSCFREQ indicates the current SYSOSC operating frequency.<br>0h = SYSOSC is at base frequency (32MHz)<br>1h = SYSOSC is at low frequency (4MHz)<br>2h = SYSOSC is at the user-trimmed frequency (16 or 24MHz)<br>3h = Reserved |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.30 SYSSTATUS (Offset = 1208h) [Reset = XXXX0XX0h]

SYSSTATUS is shown in Figure 2-142 and described in Table 2-156.

Return to the Summary Table.

System status information

**Figure 2-142. SYSSTATUS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REBOOTATTEMPTS | | RESERVED | | | | USBFS0READY | TINIEREADY |
| R-0h | | R-0h | | | | R-0h | R-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | SHDNIOLOCK | SWDCFGDIS | EXTRSTPINDIS | RESERVED | | | |
| R-0h | R-0h | R-0h | R-0h | R-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | PMUIREFGOOD | ANACPUMPGOOD | BORLVL | BORCURTHRESHOLD | | FLASHSEC | FLASHDED |
| R-0h | R-0h | R-0h | R-0h | R-0h | | R-0h | R-0h |

**Table 2-156. SYSSTATUS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | REBOOTATTEMPTS | R | 0h | REBOOTATTEMPTS indicates the number of boot attempts taken before the user application starts. |
| 29-26 | RESERVED | R | 0h | |
| 25 | USBFS0READY | R | 0h | USBFS0READY indicates when the USBFS0 peripheral is ready.<br>0h = USBFS0 is not ready<br>1h = USBFS0 is ready |
| 24 | TINIEREADY | R | 0h | TINIEREADY indicates when the TINIE peripheral is ready.<br>0h = TINIE is not ready<br>1h = TINIE is ready |
| 23-15 | RESERVED | R | 0h | |
| 14 | SHDNIOLOCK | R | 0h | SHDNIOLOCK indicates when IO is locked due to SHUTDOWN<br>0h = IO IS NOT Locked due to SHUTDOWN<br>1h = IO IS Locked due to SHUTDOWN |
| 13 | SWDCFGDIS | R | 0h | SWDCFGDIS indicates when user has disabled the use of SWD Port<br>0h = SWD Port Enabled<br>1h = SWD Port Disabled |
| 12 | EXTRSTPINDIS | R | 0h | EXTRSTPINDIS indicates when user has disabled the use of external reset pin<br>0h = External Reset Pin Enabled<br>1h = External Reset Pin Disabled |
| 11-7 | RESERVED | R | 0h | |
| 6 | PMUIREFGOOD | R | 0h | PMUIREFGOOD is set by hardware when the PMU current reference is ready.<br>0h = IREF is not ready<br>1h = IREF is ready |
| 5 | ANACPUMPGOOD | R | 0h | ANACPUMPGOOD is set by hardware when the VBOOST analog mux charge pump is ready.<br>0h = VBOOST is not ready<br>1h = VBOOST is ready |



Copyright © 2026 Texas Instruments Incorporated

**Table 2-156. SYSSTATUS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 4 | BORLVL | R | 0h | BORLVL indicates if a BOR event occured and the BOR threshold was switched to BOR0 by hardware.<br>0h = No BOR violation occured<br>1h = A BOR violation occured and the BOR threshold was switched to BOR0 |
| 3-2 | BORCURTHRESHOLD | R | 0h | BORCURTHRESHOLD indicates the active brown-out reset supply monitor configuration.<br>0h = Default minimum threshold; a BOR0- violation triggers a BOR<br>1h = A BOR1- violation generates a BORLVL interrupt<br>2h = A BOR2- violation generates a BORLVL interrupt<br>3h = A BOR3- violation generates a BORLVL interrupt |
| 1 | FLASHSEC | R | 0h | FLASHSEC indicates if a flash ECC single bit error was detected and corrected (SEC).<br>0h = No flash ECC single bit error detected<br>1h = Flash ECC single bit error was detected and corrected |
| 0 | FLASHDED | R | 0h | FLASHDED indicates if a flash ECC double bit error was detected (DED).<br>0h = No flash ECC double bit error detected<br>1h = Flash ECC double bit error detected |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.31 DEDERRADDR (Offset = 120Ch) [Reset = 00000000h]

DEDERRADDR is shown in Figure 2-143 and described in Table 2-157.

Return to the Summary Table.

Memory DED Address

**Figure 2-143. DEDERRADDR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADDR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 2-157. DEDERRADDR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | ADDR | R | 0h | Address of MEMORY DED Error. |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.32 RSTCAUSE (Offset = 1220h) [Reset = 00000000h]

RSTCAUSE is shown in Figure 2-144 and described in Table 2-158.

Return to the Summary Table.

Reset cause

**Figure 2-144. RSTCAUSE**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | ID | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | RC-0h | | | | |

**Table 2-158. RSTCAUSE Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-5 | RESERVED | R | 0h | |
| 4-0 | ID | RC | 0h | ID is a read-to-clear field which indicates the lowest level reset cause since the last read.<br>0h = No reset since last read<br>1h = POR- violation, SHUTDNSTOREx or PMU trim parity fault<br>2h = NRST triggered POR (>1s hold)<br>3h = Software triggered POR<br>4h = BOR0- violation<br>5h = SHUTDOWN mode exit<br>8h = Non-PMU trim parity fault<br>9h = Fatal clock failure<br>Ch = NRST triggered BOOTRST (<1s hold)<br>Dh = Software triggered BOOTRST<br>Eh = WWDT0 violation<br>10h = BSL exit<br>11h = BSL entry<br>13h = WWDT1 violation<br>14h = Flash uncorrectable ECC error<br>15h = CPULOCK violation<br>1Ah = Debug triggered SYSRST<br>1Bh = Software triggered SYSRST<br>1Ch = Debug triggered CPURST<br>1Dh = Software triggered CPURST |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.33 RESETLEVEL (Offset = 1300h) [Reset = 00000000h]

RESETLEVEL is shown in Figure 2-145 and described in Table 2-159.

Return to the Summary Table.

Reset level for application-triggered reset command

**Figure 2-145. RESETLEVEL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | | LEVEL | | |
| R-0h | | | | | | | | | | | | | R/W-0h | | |

**Table 2-159. RESETLEVEL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | RESERVED | R | 0h | |
| 2-0 | LEVEL | R/W | 0h | LEVEL is used to specify the type of reset to be issued when RESETCMD is set to generate a software triggered reset.<br>0h = Issue a SYSRST (CPU plus peripherals only)<br>1h = Issue a BOOTRST (CPU, peripherals, and boot configuration routine)<br>2h = Issue a SYSRST and enter the boot strap loader (BSL)<br>3h = Issue a power-on reset (POR)<br>4h = Issue a SYSRST and exit the boot strap loader (BSL) |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.34 RESETCMD (Offset = 1304h) [Reset = 00XXXXXXh]

RESETCMD is shown in Figure 2-146 and described in Table 2-160.

Return to the Summary Table.

Execute an application-triggered reset command

**Figure 2-146. RESETCMD**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| KEY | | | | | | | | RESERVED | | | | | | | |
| W-0h | | | | | | | | R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | | | | GO |
| R-0h | | | | | | | | | | | | | | | W-0h |

**Table 2-160. RESETCMD Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | The key value of E4h (228) must be written to KEY together with GO to trigger the reset.<br>E4h = Issue reset |
| 23-1 | RESERVED | R | 0h | |
| 0 | GO | W | 0h | Execute the reset specified in RESETLEVEL.LEVEL. Must be written together with the KEY.<br>1h = Issue reset |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.35 BORTHRESHOLD (Offset = 1308h) [Reset = 00000000h]

BORTHRESHOLD is shown in Figure 2-147 and described in Table 2-161.

Return to the Summary Table.

BOR threshold selection

**Figure 2-147. BORTHRESHOLD**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | | | LEVEL | |
| R-0h | | | | | | | | | | | | | | R/W-0h | |

**Table 2-161. BORTHRESHOLD Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | |
| 1-0 | LEVEL | R/W | 0h | LEVEL specifies the desired BOR threshold and BOR mode.<br>0h = Default minimum threshold; a BOR0- violation triggers a BOR<br>1h = A BOR1- violation generates a BORLVL interrupt<br>2h = A BOR2- violation generates a BORLVL interrupt<br>3h = A BOR3- violation generates a BORLVL interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.36 BORCLRCMD (Offset = 130Ch) [Reset = 00XXXXXXh]

BORCLRCMD is shown in Figure 2-148 and described in Table 2-162.

Return to the Summary Table.

Set the BOR threshold

**Figure 2-148. BORCLRCMD**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| KEY | | | | | | | | RESERVED | | | | | | | |
| W-0h | | | | | | | | R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | | | | GO |
| R-0h | | | | | | | | | | | | | | | W-0h |

**Table 2-162. BORCLRCMD Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | The key value of C7h (199) must be written to KEY together with GO to trigger the clear and BOR threshold change.<br>C7h = Issue clear |
| 23-1 | RESERVED | R | 0h | |
| 0 | GO | W | 0h | GO clears any prior BOR violation status indications and attempts to change the active BOR mode to that specified in the LEVEL field of the BORTHRESHOLD register.<br>1h = Issue clear |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.37 SYSOSCFCLCTL (Offset = 1310h) [Reset = 00XXXXXXh]

SYSOSCFCLCTL is shown in Figure 2-149 and described in Table 2-163.

Return to the Summary Table.

SYSOSC frequency correction loop (FCL) ROSC enable

**Figure 2-149. SYSOSCFCLCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | SETUSEEXRES | SETUSEFCL |
| R-0h | | | | | | W-0h | W-0h |

**Table 2-163. SYSOSCFCLCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | The key value of 2Ah (42) must be written to KEY together with SETUSEFCL to enable the FCL.<br>2Ah = Issue Command |
| 23-2 | RESERVED | R | 0h | |
| 1 | SETUSEEXRES | W | 0h | Set SETUSEEXRES to specify that an external resistor will be used for the FCL. An appropriate resistor must be populated on the ROSC pin. This state is locked until the next BOOTRST.<br>1h = Enable the SYSOSC external Resistor |
| 0 | SETUSEFCL | W | 0h | Set SETUSEFCL to enable the frequency correction loop in SYSOSC. Once enabled, this state is locked until the next BOOTRST.<br>1h = Enable the SYSOSC FCL |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.38 LFXTCTL (Offset = 1314h) [Reset = 00XXXXXXh]

LFXTCTL is shown in Figure 2-150 and described in Table 2-164.

Return to the Summary Table.

LFXT and LFCLK control

**Figure 2-150. LFXTCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | SETUSELFXT | STARTLFXT |
| R-0h | | | | | | W-0h | W-0h |

**Table 2-164. LFXTCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | The key value of 91h (145) must be written to KEY together with either STARTLFXT or SETUSELFXT to set the corresponding bit.<br>91h = Issue command |
| 23-2 | RESERVED | R | 0h | |
| 1 | SETUSELFXT | W | 0h | Set SETUSELFXT to switch LFCLK to LFXT. Once set, SETUSELFXT remains set until the next BOOTRST.<br>0h = 0<br>1h = Use LFXT as the LFCLK source |
| 0 | STARTLFXT | W | 0h | Set STARTLFXT to start the low frequency crystal oscillator (LFXT). Once set, STARTLFXT remains set until the next BOOTRST.<br>0h = LFXT not started<br>1h = Start LFXT |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.39 EXLFCTL (Offset = 1318h) [Reset = 00XXXXXXh]

EXLFCTL is shown in Figure 2-151 and described in Table 2-165.

Return to the Summary Table.

LFCLK_IN and LFCLK control

**Figure 2-151. EXLFCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | SETUSEEXLF |
| R-0h | | | | | | | W-0h |

**Table 2-165. EXLFCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | The key value of 36h (54) must be written to KEY together with SETUSEEXLF to set SETUSEEXLF.<br>36h = Issue command |
| 23-1 | RESERVED | R | 0h | |
| 0 | SETUSEEXLF | W | 0h | Set SETUSEEXLF to switch LFCLK to the LFCLK_IN digital clock input. Once set, SETUSEEXLF remains set until the next BOOTRST.<br>1h = Use LFCLK_IN as the LFCLK source |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.40 SHDNIOREL (Offset = 131Ch) [Reset = 00XXXXXXh]

SHDNIOREL is shown in Figure 2-152 and described in Table 2-166.

Return to the Summary Table.

SHUTDOWN IO release control

**Figure 2-152. SHDNIOREL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | RELEASE |
| R-0h | | | | | | | W-0h |

**Table 2-166. SHDNIOREL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | The key value 91h must be written to KEY together with RELEASE to set RELEASE.<br>91h = Issue command |
| 23-1 | RESERVED | R | 0h | |
| 0 | RELEASE | W | 0h | Set RELEASE to release the IO after a SHUTDOWN mode exit.<br>1h = Release IO |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.41 EXRSTPIN (Offset = 1320h) [Reset = 00XXXXXXh]

EXRSTPIN is shown in Figure 2-153 and described in Table 2-167.

Return to the Summary Table.

Disable the reset function of the NRST pin

**Figure 2-153. EXRSTPIN**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | DISABLE |
| R-0h | | | | | | | W-0h |

**Table 2-167. EXRSTPIN Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | The key value 1Eh must be written together with DISABLE to disable the reset function.<br>1Eh = Issue command |
| 23-1 | RESERVED | R | 0h | |
| 0 | DISABLE | W | 0h | Set DISABLE to disable the reset function of the NRST pin. Once set, this configuration is locked until the next POR.<br>0h = Reset function of NRST pin is enabled<br>1h = Reset function of NRST pin is disabled |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.42 SYSSTATUSCLR (Offset = 1324h) [Reset = 00XXXXXXh]

SYSSTATUSCLR is shown in Figure 2-154 and described in Table 2-168.

Return to the Summary Table.

Clear sticky bits of SYSSTATUS

**Figure 2-154. SYSSTATUSCLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | ALLECC |
| R-0h | | | | | | | W-0h |

**Table 2-168. SYSSTATUSCLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | The key value CEh (206) must be written to KEY together with ALLECC to clear the ECC state.<br>CEh = Issue command |
| 23-1 | RESERVED | R | 0h | |
| 0 | ALLECC | W | 0h | Set ALLECC to clear all ECC related SYSSTATUS indicators.<br>1h = Clear ECC error state |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.43 SWDCFG (Offset = 1328h) [Reset = 00XXXXXXh]

SWDCFG is shown in Figure 2-155 and described in Table 2-169.

Return to the Summary Table.

Disable the SWD function on the SWD pins

**Figure 2-155. SWDCFG**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | DISABLE |
| R-0h | | | | | | | W-0h |

**Table 2-169. SWDCFG Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | The key value 62h (98) must be written to KEY together with DISBALE to disable the SWD functions.<br>62h = Issue command |
| 23-1 | RESERVED | R | 0h | |
| 0 | DISABLE | W | 0h | Set DISABLE to disable the SWD function on SWD pins, allowing the SWD pins to be used as GPIO.<br>1h = Disable SWD function on SWD pins |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.44 FCCCMD (Offset = 132Ch) [Reset = 00XXXXXXh]

FCCCMD is shown in Figure 2-156 and described in Table 2-170.

Return to the Summary Table.

Frequency clock counter start capture

**Figure 2-156. FCCCMD**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| KEY | | | | | | | | RESERVED | | | | | | | |
| W-0h | | | | | | | | R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | | | | GO |
| R-0h | | | | | | | | | | | | | | | W-0h |

**Table 2-170. FCCCMD Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | The key value 0Eh (14) must be written with GO to start a capture.<br>0Eh = Issue command |
| 23-1 | RESERVED | R | 0h | |
| 0 | GO | W | 0h | Set GO to start a capture with the frequency clock counter (FCC).<br>1h = 1 |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.45 USBFLLCTL (Offset = 1340h) [Reset = 000000XXh]

USBFLLCTL is shown in Figure 2-157 and described in Table 2-171.

Return to the Summary Table.

USB FLL Control Register

**Figure 2-157. USBFLLCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | REFSEL |
| R-0h | | | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | FLLEN | CLKEN |
| R-0h | | | | | | R/W-0h | R/W-0h |

**Table 2-171. USBFLLCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-9 | RESERVED | R | 0h | |
| 8 | REFSEL | R/W | 0h | Reference Clock Selection for Frequency Lock Loop<br>1h = Use LFCLK as Reference Clock |
| 7-2 | RESERVED | R | 0h | |
| 1 | FLLEN | R/W | 0h | USB Oscillator Frequeny Lock Loop Enable<br>0h = USB CLK Frequeny Lock Loop is disabled<br>1h = USB CLK Frequeny Lock Loop is enabled |
| 0 | CLKEN | R/W | 0h | USB Oscillator Enable<br>0h = USB CLK is disabled<br>1h = USB CLK is enabled |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.46 USBFLLADJ (Offset = 1344h) [Reset = 000000X0h]

USBFLLADJ is shown in Figure 2-158 and described in Table 2-172.

Return to the Summary Table.

USB FLL Manual Adjustment Register

**Figure 2-158. USBFLLADJ**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | FRQMOD | | |
| R-0h | | | | | R/W-0h | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | FRQADJ | | | | | |
| R-0h | | R/W-0h | | | | | |

**Table 2-172. USBFLLADJ Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-11 | RESERVED | R | 0h | |
| 10-8 | FRQMOD | R/W | 0h | Frequency Lock Loop Modulator Value |
| 7-6 | RESERVED | R | 0h | |
| 5-0 | FRQADJ | R/W | 0h | Frequency Lock Loop Adjust |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.47 USBFLLSTAT (Offset = 1348h) [Reset = 00X0XXXXh]

USBFLLSTAT is shown in Figure 2-159 and described in Table 2-173.

Return to the Summary Table.

USB FLL Status Register

**Figure 2-159. USBFLLSTAT**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | FRQMOD | |
| R-0h | | | | | | R-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | FRQADJ | | | | | |
| R-0h | | R-0h | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | REFRDY |
| R-0h | | | | | | | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | FLLLOCK | CLKRDY |
| R-0h | | | | | | R-0h | R-0h |

**Table 2-173. USBFLLSTAT Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | RESERVED | R | 0h | |
| 26-24 | FRQMOD | R | 0h | Current Frequency Lock Loop Modulator Value |
| 23-22 | RESERVED | R | 0h | |
| 21-16 | FRQADJ | R | 0h | Current Frequency Lock Loop Adjust |
| 15-9 | RESERVED | R | 0h | |
| 8 | REFRDY | R | 0h | Reference Clock for Frequency Lock Loop ready indication<br>0h = USB FLL Reference clock is not ready<br>1h = USB FLL Reference clock is ready |
| 7-2 | RESERVED | R | 0h | |
| 1 | FLLLOCK | R | 0h | USB Oscillator Frequeny Lock Loop Lock Indication<br>0h = USB CLK Frequeny Lock Loop is not locked<br>1h = USB CLK Frequeny Lock Loop is locked |
| 0 | CLKRDY | R | 0h | USB Oscillator Ready Indication<br>0h = USB CLK is not ready<br>1h = USB CLK is ready |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.48 SHUTDNSTORE0 (Offset = 1400h) [Reset = 00000000h]

SHUTDNSTORE0 is shown in Figure 2-160 and described in Table 2-174.

Return to the Summary Table.

Shutdown storage memory (byte 0)

**Figure 2-160. SHUTDNSTORE0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DATA | | | | | | | |
| R/W-0h | | | | | | | |

**Table 2-174. SHUTDNSTORE0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-10 | RESERVED | R | 0h | |
| 7-0 | DATA | R/W | 0h | Shutdown storage byte 0 |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.49 SHUTDNSTORE1 (Offset = 1404h) [Reset = 00000000h]

SHUTDNSTORE1 is shown in Figure 2-161 and described in Table 2-175.

Return to the Summary Table.

Shutdown storage memory (byte 1)

**Figure 2-161. SHUTDNSTORE1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DATA | | | | | | | |
| R/W-0h | | | | | | | |

**Table 2-175. SHUTDNSTORE1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-10 | RESERVED | R | 0h | |
| 7-0 | DATA | R/W | 0h | Shutdown storage byte 1 |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.50 SHUTDNSTORE2 (Offset = 1408h) [Reset = 00000000h]

SHUTDNSTORE2 is shown in Figure 2-162 and described in Table 2-176.

Return to the Summary Table.

Shutdown storage memory (byte 2)

**Figure 2-162. SHUTDNSTORE2**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DATA | | | | | | | |
| R/W-0h | | | | | | | |

**Table 2-176. SHUTDNSTORE2 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-10 | RESERVED | R | 0h | |
| 7-0 | DATA | R/W | 0h | Shutdown storage byte 2 |



Copyright © 2026 Texas Instruments Incorporated

### 2.7.51 SHUTDNSTORE3 (Offset = 140Ch) [Reset = 00000000h]

SHUTDNSTORE3 is shown in Figure 2-163 and described in Table 2-177.

Return to the Summary Table.

Shutdown storage memory (byte 3)

**Figure 2-163. SHUTDNSTORE3**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DATA | | | | | | | |
| R/W-0h | | | | | | | |

**Table 2-177. SHUTDNSTORE3 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-10 | RESERVED | R | 0h | |
| 7-0 | DATA | R/W | 0h | Shutdown storage byte 3 |

## 2.8 Quick Start Reference

The PMCU is designed to provide a simple, easy-to-use power management, clocking, and reset management functionality. This section describes the basic operating principles of the PMCU as well as tips and tricks for taking the default configuration out of reset and optimizing it for particular applications.

### *2.8.1 Default Device Configuration*

The default operating configuration of the device provides basic functionality which can be suitable for many applications without modification.

MSPM0Gxx devices power up and release reset for execution of application code when the external supply (VDD and VSS) reaches 1.62V. When the application code is released for execution, the device is in RUN mode with MCLK, which is sourced from the internal SYSOSC at 32MHz. The CPUCLK and ULPCLK are also 32MHz , derived from MCLK. LFCLK starts automatically, sourced from the internal LFOSC. In RUN mode with the default configuration, all peripherals are available to be enabled. Peripherals such as the DMA, CRC, and AES run directly from MCLK at the MCLK rate. Other peripherals, such as timers and serial interfaces, can run from the bus clock at 32MHz or from the low-frequency 32kHz clock (LFCLK) based on their peripheral clock selection.

Power consumption can be reduced by entering SLEEP, STOP, STANDBY, or SHUTDOWN mode. By default, these modes behave in the following way:

- In SLEEP mode, the CPUCLK is disabled but all peripherals, including the DMA, continue to run as configured at either 32MHz or 32kHz. This power mode is designed for scenarios where PD1 peripherals are used and having the lowest possible wakeup latency is more important than having the lowest power consumption.



Copyright © 2026 Texas Instruments Incorporated

- In STOP mode, (STOP0 by default), the MCLK source to PD1 peripherals is disabled and PD1 peripherals are disabled and in retention (unavailable for use). The default STOP mode is designed for scenarios where power optimization is important but the ADC, OPA, DAC, or a clock faster than 32kHz is needed.
  - SYSOSC by default will continue to run at 32MHz the max operating frequency, but the MCLK tree will run at 4MHz (SYSOSC/8) and peripherals in PD0 which are still active will see the bus clock (ULPCLK) change from 32MHz to 4MHz.
  - PD0 peripherals configured to run from LFCLK continue to run at 32kHz.
  - All OPA modes can be used
  - The ADC will always see SYSOSC at 32MHz for sampling
- In STANDBY mode (STANDBY0 by default), the MCLK tree runs from LFCLK at 32kHz and SYSOSC is disabled. PD0 peripherals running from the bus clock change to 32kHz. PD0 peripherals running from LFCLK continue to run at 32kHz with no change.

### 2.8.2 Leveraging MFCLK

When running with the default PMCU configuration, timers and serial interfaces can select either the bus clock (MCLK/ULPCLK) or the LFCLK as their clock source. LFCLK is always 32kHz in RUN, SLEEP, STOP, and STANDBY, but MCLK/ULPCLK changes to 4MHz in STOP and to 32kHz in STANDBY, meaning that peripherals running from the bus clock see the source clock frequency change when transitioning power modes.

MFCLK, by contrast, works like LFCLK in that it provides a constant frequency clock source for peripherals across RUN, SLEEP, and STOP modes. MFCLK provides a constant 4MHz as an alternative to LFCLK which runs at 32kHz. The 4MHz time base for MFCLK is always derived from SYSOSC. Peripherals, specifically PD0 peripherals that can be used in STOP mode, can select MFCLK as their clock source instead of ULPCLK. MFCLK is maintained at 4MHz in RUN, SLEEP and STOP for peripherals like UART, I2C, and low-power timers that need a consistent clock but require a clock source greater than 32kHz.

For information on using MFCLK, see the MFCLK section.

### 2.8.3 Optimizing Power Consumption in STOP Mode

The STOP mode provides considerable flexibility for tailoring the device to an application's specific power and performance requirements. There are several options available for reducing power consumption in STOP mode:

- Enable SYSOSC gear shift mode in STOP to reduce SYSOSC current. In STOP mode, the MCLK tree (and ULPCLK) are always limited to 4MHz maximum.By default, SYSOSC runs in STOP mode at 32MHz (base frequency) with a divide-by-8 to meet the 4MHz max frequency limit in STOP mode. Alternatively, SYSOSC can be gear shifted to run natively at 4MHz in STOP mode (STOP1), saving SYSOSC current. See SYSOSC gear shift mode.
- If a 32kHz clock is sufficient to run the needed peripherals, it is possible to run in STOP mode with MCLK sourced from LFCLK at 32kHz. SYSOSC can be disabled to conserve power. To disable SYSOSC in STOP mode and run from LFCLK (STOP2), see disabling SYSOSC and operating mode selection.

### 2.8.4 Optimizing Power Consumption in STANDBY Mode

In STANDBY mode, if only RTC, TIMG0, TIMG1,or asynchronous fast wake from GPIO, comparator (low-power mode), or a serial interface is desired, the lowest possible power consumption can be achieved by configuring the ULPCLK and LFCLK to be disabled when entering STANDBY, leaving only the RTC, TIMG0, TIMG1, (STANDBY1). See the LFCLK section. In this state, RTC, TIMG0, TIMG1,or asynchronous activity/event will trigger an asynchronous fast clock request to wake the system.

### 2.8.5 Increasing MCLK Precision

If an application requires high clocking accuracy for high-frequency peripherals, the best accuracy is achieved by using an external high-frequency crystal with the HFXT and sourcing the MCLK tree directly from HFCLK. By sourcing the MCLK directly from HFCLK, the bus clock to all peripherals in PD1 and PD0 will be the HFCLK.

Crystal frequencies up to 48MHz are supported by the HFXT, but because PD0 peripherals using ULPCLK are limited to 40MHz, 40MHz is the highest crystal frequency supported to run the PD1 and PD0 peripherals at the



Copyright © 2026 Texas Instruments Incorporated

same frequency. If a 48MHz crystal is used, PD1 peripherals and the CPUCLK can run at 48MHz directly from the crystal, but PD0 peripherals must run at MCLK/2 (24MHz).

If precision clocking is needed for the CAN-FD controller or the ADC (for sampling window generation), and high CPU performance (high MCLK) is also required, it is also possible to source the CAN-FD controller (CANCLK) and the ADC sampling clock (ADCCLK) from HFCLK directly, asynchronous to MCLK, while running MCLK at maximum frequency using the PLL for best compute performance.

### 2.8.6 Configuring MCLK for Maximum Speed

The best CPU compute performance is obtained by using the SYSPLL to generate an 80MHz clock from either the SYSOSC reference or HFXT. To configure the SYSPLL to generate an 80MHz output sourcing MCLK, see the SYSPLL configuration section.

Running MCLK at 80MHz also provides the best possible timer resolution for TIMA and TIMG peripherals in the PD1 domain (12.5ns).

### 2.8.7 High Speed Clock (SYSPLL, HFCLK) Handling in Low-Power Modes

The SYSPLL and HFCLK (HFXT, HFCLK_IN) high speed clock sources are not supported in the STOP and STANDBY operating modes. When a high-speed clock source ( SYSPLL , HFCLK) is enabled, entering either the STOP mode or STANDBY mode will cause SYSCTL to automatically disable the SYSPLLand/or HFCLK before entering STOP or STANDBY mode. Upon exit from STOP or STANDBY mode to RUN mode, SYSCTL will automatically re-enable the SYSPLL and/or HFCLK if they were previously enabled before entering the low-power mode.

Before entering STOP or STANDBY for the first time after enabling the SYSPLL or HFCLK, and after waking up from STOP or STANDBY mode, application software must wait to enter STOP or STANDBY mode until any previously enabled high speed clock sources have completed startup.

Application software must check the following before entering STOP or STANDBY mode:

- If the **SYSPLL was enabled**, then application software must wait for the SYSPLL to complete startup. When the SYSPLL startup is complete, the SYSPLLGOOD bit will be set in the CLKSTATUS register in SYSCTL. If the SYSPLL fails to start, the SYSPLLOFF bit will be set instead. The SYSPLLOFF bit indicates that the SYSPLL was dead at startup or was not previously enabled. Ensure that either SYSPLLGOOD or SYSPLLOFF is set before attempting to enter STOP or STANDBY.
- If the **HFCLK was enabled**, then application software must wait for the HFCLK to complete startup. When the HFCLK startup is complete, the HFCLKGOOD bit will be set in the CLKSTATUS register in SYSCTL. If the HFCLK fails to start, the HFCLKOFF bit will be set instead. The HFCLKOFF bit indicates that the HFCLK was dead at startup or was not previously enabled. Ensure that either HFCLKGOOD or HFCLKOFF is set before attempting to enter STOP or STANDBY.

In the event that the MCLK was configured to be sourced from HSCLK before entry to STOP or STANDBY, upon exit from STOP or STANDBY the MCLK will be sourced from SYSOSC initially and the CPU will be released to begin executing code at the SYSOSC frequency. SYSCTL will automatically restore the MCLK configuration to the previously selected high-speed clock when the high-speed clock has started and is ready for use. When MCLK switches back to the high-speed clock, SYSCTL will generate an HSCLK GOOD interrupt to alert the application that MCLK is again running from the high-speed clock.

### 2.8.8 Optimizing for Lowest Wakeup Latency

To ensure the lowest possible wakeup latency from STOP or STANDBY mode to RUN mode, set MCLK to SYSOSC with SYSOSC running at base frequency (32MHz) before entering STOP or STANDBY. SYSOSC always starts at base frequency and latency is minimized if SYSOSC does not need to change to an alternate frequency.



Copyright © 2026 Texas Instruments Incorporated

### 2.8.9 Optimizing for Lowest Peak Current in RUN/SLEEP Mode

In applications which are peak current limited, there are two options for reducing active current in RUN and SLEEP modes:

- If 32kHz provides sufficient performance, run MCLK from LFCLK. MCLK can be selected to run from LFCLK with SYSOSC disabled. If no fast handling of events is needed, SYSOSC asynchronous requests can be disabled to ensure that the device always runs from LFCLK. This provides the lowest possible current with the CPU still running (RUN2). See the MCLK, SYSOSC, and Operating Mode Selection sections.
- If 32kHz does not provide sufficient performance, MCLK can be selected to run from SYSOSC with SYSOSC set to low frequency (4MHz). With MCLK running from SYSOSC, the MDIV divider for MCLK can be applied to reduce current consumption. For example, MCLK can be configured to run as low as 250kHz by setting MDIV to /16 when sourced from SYSOSC running at 4MHz. See the MCLK section.



Copyright © 2026 Texas Instruments Incorporated

Chapter 3

# CPU

The CPU subsystem (MCPUSS) includes the Arm Cortex-M0+ processor, the interrupt logic, and the prefetch and cache logic.


**3.1 Overview** ..... 471
**3.2 Arm Cortex-M0+ CPU** ..... 471
**3.3 Interrupts and Exceptions** ..... 476
**3.4 CPU Peripherals** ..... 483
**3.5 Read-Only Memory (ROM)** ..... 486
**3.6 CPUSS Registers** ..... 487
**3.7 WUC Registers** ..... 502





Copyright © 2026 Texas Instruments Incorporated

## 3.1 Overview

The MSP CPU subsystem (MCPUSS) contains the central processing unit (CPU) along with associated supporting logic and read-only memory (ROM). The functional blocks that comprise the MCPUSS include:

- The Arm Cortex-M0+ 32-bit CPU and internal peripherals
- The CPU bus splitter and router
- The interrupt management logic and DEEPSLEEP entry and exit logic
- The nonvolatile memory system prefetch and cache logic
- The CPU debug interface to the debug subsystem (DEBUGSS)
- The processor trace memory
- The read-only memory (ROM) used for the BCR and BSL

The top level architecture of the MCPUSS is shown in Figure 3-1 .

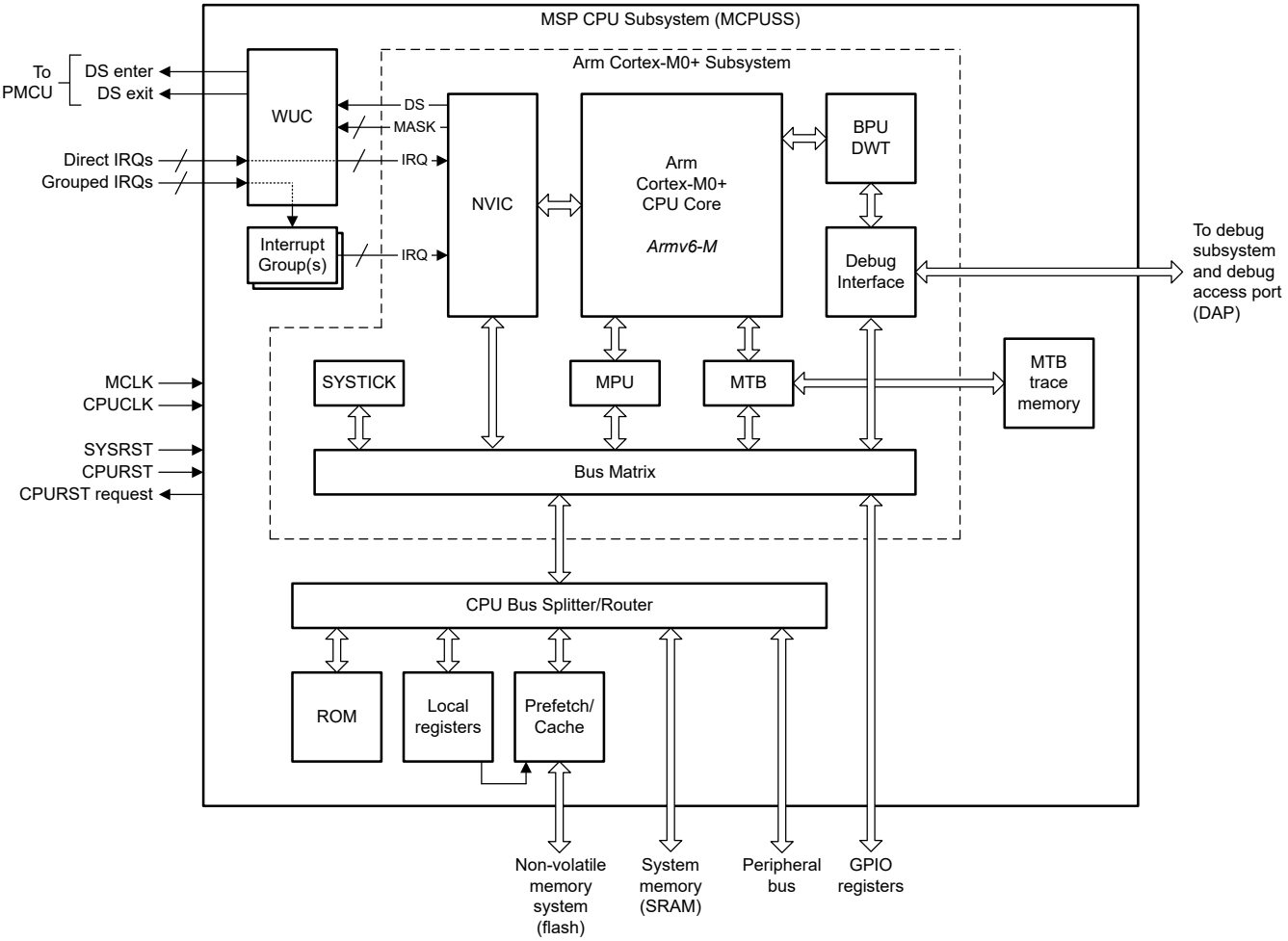



**Figure 3-1. MSPM0Gxx MCPUSS Top Level Diagram**

## 3.2 Arm Cortex-M0+ CPU

The MCPUSS contains an energy-efficient Arm Cortex-M0+ CPU implementing the Armv6-M instruction set architecture (ISA) with support for CPU clock speeds up to 80MHz. The Cortex-M0+ is a Von Neumann style 32-bit processor with a 2-stage ultra-low power pipeline and a single-cycle access port to the GPIO registers for efficient GPIO manipulation.

The Cortex-M0+ implementation on MSPM0Gxx devices has the following features:

- Up to 80MHz execution frequency



Copyright © 2026 Texas Instruments Incorporated

- Little-endian (least significant byte at lowest byte address location)
- Support for 32-bit word instruction fetches
- Single cycle 32×32 multiply instruction
- Tightly integrated memory protection unit (MPU)
- User and privileged execution modes
- Integrated 24-bit system tick timer (SYSTICK)
- Four hardware breakpoints and two hardware watchpoints for debug
- Micro trace buffer
- Reset-all-registers support
- Vector table offset support

The Cortex-M0+ architecture enables excellent code density, deterministic interrupt handling, and upwards compatibility with other processor architectures in the Arm Cortex-M family.

A general overview of the Arm Cortex-M0+ is given in this section to provide a basic understanding of the features of the processor. For detailed information on developing with the Arm Cortex-M0+ processor, refer to the *Arm Cortex-M0+ Devices Generic User's Guide*.

### *3.2.1 CPU Register File*

The Arm Cortex-M0+ processor instructions operate on registers in the CPU register file. The processor contains a register file consisting of 16 standard registers and 3 special registers as shown in Figure 3-2.



Copyright © 2026 Texas Instruments Incorporated

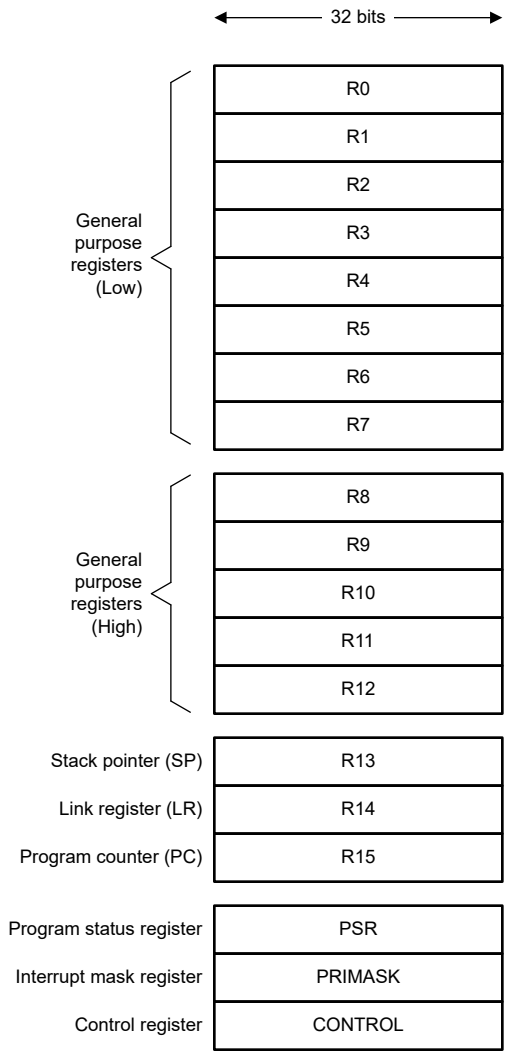



**Figure 3-2. CPU Registers**

### General Purpose Registers (R0-R12)

The processor provides 13 general purpose registers, R0-R12, for operating on data. Registers R0 to R7 (low registers) are accessible by all instructions which specify a general purpose register. Registers R8 to R12 (high registers) are not accessible by 16-bit instructions but are accessible by any 32-bit instructions which specify a general purpose register.

### Stack Pointer Register (R13)

The stack pointer is contained in R13, and can contain the main stack pointer (MSP) or the process stack pointer (PSP) . When the processor is running in handler mode, the main stack pointer (MSP) is always used. When the processor is running in thread mode, the MSP or the process stack pointer (PSP) can be used, depending on the configuration of the SPSEL bit in the CONTROL register.

After a CPURST, the processor automatically and unconditionally fetches the default stack pointer from the first address of main flash (0x0000.0000) as the main stack pointer (MSP).



Copyright © 2026 Texas Instruments Incorporated

### Link Register (R14)

R14 serves as the link register and contains the return value of function calls as well as exceptions. The link register must be set before being used as it is not reset to any known value. It is accessible in privileged and unprivileged mode.

### Program Counter Register (R15)

The program counter register (R15) contains the address of the next instruction to be executed. The PC is accessible in privileged and unprivileged mode.

After a CPURST, the processor automatically and unconditionally fetches the default PC from the second word of main flash (0x0000.0004).

### Special Registers

Special registers include the program status register (PSR), the interrupt mask register (PRIMASK), and the control register (CONTROL). Special registers are typically accessed by using the CPS, MRS, and MSR system instructions.

- **Program Status Register (PSR):** The PSR is a combination of the application status (APSR), interrupt status (IPSR), and execution status (EPSR) registers. Application software can access the PSR with MRS and MSR instructions, accessing either the complete PSR or a combination of one or more registers, with some restrictions. The PSR registers can be accessed with MRS and MSR instructions using the mnemonics given in Table 3-1.
  - The application status register (APSR) contains the N, Z, C, and V flags which are used by the processor to evaluate conditional branch instructions. These bits are located in BIT31, BIT30, BIT29, and BIT28 of the PSR, respectively.
  - The interrupt status register (IPSR) reports the current exception number for a currently executing exception when in handler mode. In thread mode it reads as zero. The processor ignores writes to this register. The exception number field is presented in from BIT0 to BIT5 of the PSR.
  - The execution status register (EPSR) contains the T bit (BIT24), which defines whether the processor is in Thumb state. This bit cannot be read or written by software, but it is used by the processor.
- **Interrupt Mask Register (PRIMASK):** BIT0 of the PRIMASK register (PM) can be used to mask all interrupts to the processor which have configurable priority (see Section 3.3). This can be thought of as a global peripheral interrupt mask control. The processor ignores unprivileged writes to PRIMASK. Clearing PM to 0 enables interrupts. Setting PM to 1 disables interrupts. The CPS instruction can be used to change the PM bit value in the PRIMASK register.
- **Control Register (CONTROL):** The control register can be used to define whether code executing in thread mode is privileged or unprivileged by clearing or setting the nPRIV bit (BIT0), respectively. It can also be used to select the stack pointer used in R13 as either the main stack pointer (MSP) or process stack pointer (PSP) by clearing or setting the SPSEL bit (BIT1), respectively. A CPURST clears the CONTROL register to zero. The processor ignores unprivileged writes to the CONTROL register. The SPSEL stack pointer selection bit is updated by the processor automatically when entering and returning from exceptions. Note that software must implement an ISB barrier instruction after writing to CONTROL to ensure that any changes take effect before the next application instruction is executed by the processor.

**Table 3-1. Program Status Register (PSR) Access Mnemonics**

| Mnemonic | Subregisters Included |
|---|---|
| APSR | APSR |
| IPSR | IPSR |
| EPSR | EPSR |
| IAPSR | IPSR and APSR |
| EAPSR | EPSR and APSR |
| XPSR | APSR, IPSR, EPSR |



Copyright © 2026 Texas Instruments Incorporated

**Table 3-1. Program Status Register (PSR) Access Mnemonics (continued)**

| Mnemonic | Subregisters Included |
|---|---|
| IEPSR | IPSR and EPSR |

### 3.2.2 Stack Behavior

The Arm Cortex-M0+ processor implements a full descending stack protocol. The stack pointer register (SP) always indicates the location of the last stacked data. When new data is added to the call stack, the SP value is decremented and the data is written to the location indicated by the SP after being decremented.

The Arm Cortex-M0+ supports managing two independent stacks with two pointers: the main stack (MSP) and the process stack (PSP).

### 3.2.3 Execution Modes and Privilege Levels

The processor supports two primary modes of execution:

- **Thread mode** (for executing application software)
- **Handler mode** (for handling processor exceptions and peripheral interrupts)

By default, the processor is in thread mode out of reset. If an exception is issued to the processor, the processor will handle the exception in handler mode and return to thread mode after handler execution is complete. Code running in thread mode can be configured as being privileged or unprivileged, based on the configuration of the processor's internal CONTROL register. Code running in handler mode always executes as privileged.

In general, code which executes as privileged has complete control of the processor configuration, including control of the MPU, SysTick, NVIC, and SCB. Only privileged code can change the privilege level for code running in thread mode.

Code that is executing in thread mode in an unprivileged state cannot access the previously mentioned resources ( MPU, SysTick, NVIC, SCB)and can be subject to MPU restrictions if the MPU is configured and enabled.

### 3.2.4 Address Space and Supported Data Sizes

MSPM0 devices implement a flat memory map with a 32-bit byte-addressable address space. Byte addresses are unsigned numbers ranging from zero to $2^{32}$-1.

#### Address Space

The processor sees the address space as containing $2^{30}$ 32-bit words, with each word being word-aligned (4-byte aligned). Pointers are always 32 bits, and stack operations (for example, push , pop) increment the stack pointer by 4 addresses (4 bytes). Address calculations by the processor wrap around if they overflow or underflow the 32-bit memory space.

Instruction fetches by the processor are always 16-bit half-word aligned.

Data reads by the processor must be naturally aligned (for example, words must be word aligned, half words must be half-word aligned, etc.).

#### Supported Data Sizes

The processor supports 8-bit byte, 16-bit half-word, and 32-bit word data sizes. Signed and unsigned data is supported, and signed data is stored in CPU registers in 32-bit two's complement format. The Armv6-M instruction set does not provide native instructions supporting operations on 64-bit double-word data.

Load operations from memory to a CPU register can be signed or unsigned when the data size is less than 32 bits. When loading unsigned half-word or byte data to a CPU register, the value is zero-extended to 32 bits automatically. When loading signed half-word or byte data to a CPU register, the value is sign-extended to 32 bits automatically.



Copyright © 2026 Texas Instruments Incorporated

Stores from CPU registers to memory are sign agnostic.

All instruction and data accesses use little endian byte order.

## 3.3 Interrupts and Exceptions

Peripheral interrupt exceptions and system exceptions temporarily pause the processor's normal execution flow so that the processor can be used to handle an event.

The following can cause interruption of normal execution flow:

- A CPURST
- A nonmaskable interrupt (NMI, software trigger or hardware error signal from SYSCTL)
- A fault exception in the system (HardFault)
- Execution of a supervisor call instruction (SVCall)
- Setting of a pending supervisor service request (PendSV)
- A SysTick exception
- An enabled peripheral interrupt (IRQ)
- A breakpoint instruction (for debug)

### Exception State

Each exception source to the processor will be in one of the below states at any given point in time:

- **Inactive** (not active, not pending)
- **Pending** (waiting to be serviced by the processor)
- **Active** (actively being serviced by the processor)
- **Active and pending** (actively being serviced by the processor when the same source generates another exception)

### Exception Prioritization, Entry, and Exit

Exceptions are prioritized by the processor together with the nested vectored interrupt controller (NVIC). Each exception has either a fixed priority (reset, NMI,hard fault) or a configurable priority (SVCall, PendSV, SysTick,peripheral IRQs). Exceptions with configurable priority can be disabled by application software running in privileged mode. Exceptions with fixed priority cannot be disabled.

The processor exception model supports preemption, tail-chaining, and late-arrival features to boost exception handling performance:

- In the **preemption** case, if an exception of higher priority is pending when an exception of lower priority is executing, the higher priority exception will preempt the ongoing handler servicing the lower priority exception.
- In the **tail-chaining** case, if a valid-for-entry exception is pending at the time of completion of an exception handler, then the application context is not restored from the stack and control is given immediately to the pending exception.
- In the **late-arriving** case, if a higher priority exception occurs during entry to a lower priority exception, the higher priority exception will be serviced first after the processor state is saved to the stack. Once the higher priority exception handling is complete, the lower priority exception (which is still pending) is serviced based on the tail-chaining procedure.

An exception entry is issued if and when all of the following are true:

- An exception is in a **pending** state
- The priority of the pending exception is higher than the limit set by the exception mask register (PRIMASK)
- The processor is currently in either thread mode (not servicing an exception) or the newly pending exception is higher priority than the exception which is currently being serviced (resulting in a preemption)



Copyright © 2026 Texas Instruments Incorporated

Processor exceptions are vectored. When an exception occurs, the current processor state is pushed onto the stack which was active at the time of the event, and execution is vectored to the entry point address in the vector table corresponding to the exception which is to be processed.

If the exception is tail-chained to a previous handler which has completed, then there is no need to push any state to the stack and the interrupt service routine can be vectored to immediately. Likewise, if the exception is higher priority than a previous exception which started entry but did not complete entry, then there is no need to save the context again (late arrival).

Upon completion of an exception handler, if there is no exception pending which needs to be handled then the processor will pop the state from the stack and restore the processor to the previous state which it was in when the exception occurred.

### *3.3.1 Peripheral Interrupts (IRQs)*

Peripheral interrupt functionality is managed by several components on the device:

- The nested vectored interrupt controller (NVIC)
- One or more interrupt groups (INT_GROUP)
- The wake-up controller (WUC)

MSPM0 devices include an Arm nested vectored interrupt controller (NVIC) with the Cortex-M0+ CPU for managing peripheral interrupts. The NVIC operation is tightly integrated with the processor and supports up to 32 native peripheral interrupt sources.

In addition to the NVIC, interrupt grouping modules can be present on a device to enable interfacing of more than 32 peripheral interrupts to the NVIC. High-priority interrupt sources that might require preemption capability are mapped directly to the NVIC and behave like normal NVIC interrupts. Lower priority interrupt sources which do not commonly require preemption capability are mapped to an interrupt grouping module, the output of which is then mapped to the NVIC as a native NVIC interrupt source. This routing arrangement is shown in Figure 3-3.

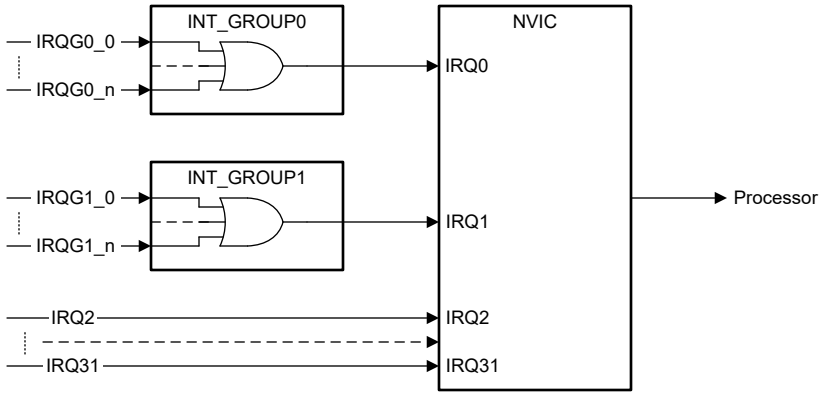



**Figure 3-3. Peripheral Interrupt Hierarchy**

The wake-up controller (WUC) determines if the PD1 power domain (containing the processor) needs to be powered up to service a peripheral interrupt if the PD1 domain is powered down in STOP or STANDBY mode.

#### 3.3.1.1 Nested Vectored Interrupt Controller (NVIC)

The nested vectored interrupt controller (NVIC) is an industry-standard Arm component which interfaces peripheral interrupts (which are external to the processor) into the CPU. The NVIC supports connection of up to 32 native peripheral interrupt sources.

The NVIC is configured through memory-mapped registers in the system private peripheral bus (PPB) region. See Table 3-2 for the list of NVIC registers. The software development kit (SDK) provided with the devices supports the standard Arm Cortex Microcontroller Software Interface Standard (CMSIS) register access



Copyright © 2026 Texas Instruments Incorporated

definitions for the NVIC. Application software must use 32-bit aligned, word-size transactions when accessing any NVIC register.

In addition to interfacing peripheral interrupts to the processor, the NVIC also supports programmable priority for each interrupt.

### Enabling and Disabling Interrupts

Peripheral interrupt enables can be read, set, and cleared through the interrupt set-enable (ISER) and interrupt clear-enable (ICER) registers in the NVIC. The 32 interrupts are mapped to the ISER and ICER registers with interrupt zero in the BIT0 position (LSB) and interrupt 31 in the BIT31 position (MSB) of each register. To enable an interrupt, set the corresponding enable bit in the ISER register. Writing a '0' to ISER has no effect. It is possible to read the ISER register to determine which interrupts are enabled. Upon a read, a '1' indicates that an interrupt is enabled; a '0' indicates disabled. To disable an interrupt, set the corresponding enable bit in the ICER register. Writing a '0' to ICER has no effect.

**Note**

In addition to enabling a peripheral interrupt at the NVIC, it is generally necessary to also configure the interrupt configuration of the corresponding peripheral as well. Most peripherals have multiple interrupt sources, which are merged together in the peripheral to source a single NVIC interrupt. Masking of individual peripheral interrupts is done within the peripheral's interrupt management registers.

In the event that an interrupt is disabled in the NVIC, if the interrupt is asserted by the corresponding peripheral then the NVIC interrupt will go to a pending state but the processor is not interrupted. If an interrupt is disabled when in an active state (when a handler is running) it will remain active until the exception handler returns or a reset occurs, but no further activations will happen.

**Note**

If a peripheral asserts an interrupt to the NVIC, but that peripheral's interrupt in the NVIC is disabled, the device may remain in a higher power mode than expected as the wake-up controller (WUC) is holding an event for the processor. To prevent this situation, ensure peripheral interrupts are masked at the peripheral directly, versus only masking interrupts at the NVIC.

### Setting and Clearing Pending Interrupt Status

Pending interrupt status can be read, set, and cleared through the interrupt set-pending (ISPR) and interrupt clear-pending (ICPR) registers in the NVIC. The 32 interrupts are mapped to the ISPR and ICPR registers with interrupt zero in the BIT0 position (LSB) and interrupt 31 in the BIT31 position (MSB) of each register. To read if an interrupt is pending, read either the ISPR or the ICPR. Upon a read, a '1' indicates that an interrupt is pending; a '0' indicates not pending. To set an interrupt to a pending state through software, set the corresponding bit in the ISPR register. Writing a '0' to ISPR has no effect. To clear an interrupt pending state, set the corresponding bit in the ICPR register. Writing a '0' to ICPR has no effect. Note that if a peripheral interrupt condition is still present, the pending state will be set again by hardware even if it is cleared.

### Setting Interrupt Priority

Interrupts on the NVIC have programmable priority. There are four priority levels possible. Priority is set by programming the eight IPRx registers in the NVIC. Each priority field is 8 bits in length, and the priority for 4 interrupts is configured per 32-bit register. The Arm Cortex-M0+ only implements the most significant 2 bits of each 8-bit priority field (giving the 4 priority levels). Lower priority values have higher priority. System exceptions (reset, NMI, hard fault) have fixed priorities of -3, -2, and -1, respectively. As such, these exceptions always have higher priority than peripheral interrupts. Peripheral interrupt priorities are programmable as 0, 64, 128, or 192, with 0 being highest priority and 192 being lowest priority.



Copyright © 2026 Texas Instruments Incorporated

If the processor is currently handling an exception, it can only be preempted by a higher priority exception. In the event that there are multiple exceptions in a pending state which all have the same priority level assigned, the exception with the lowest exception number is taken first.

**Note**

Application software must not change the priority of an interrupt while the corresponding interrupt is either active (being handled) or enabled. Doing so can result in unpredictable behavior.

**Table 3-2. Arm Cortex-M0+ NVIC Registers**

| Address | Register | CMSIS | Description |
|---|---|---|---|
| 0xE000.E100 | NVIC_ISER | NVIC->ISER[0] | Interrupt set-enable register |
| 0xE000.E180 | NVIC_ICER | NVIC->ICER[0] | Interrupt clear-enable register |
| 0xE000.E200 | NVIC_ISPR | NVIC->ISPR[0] | Interrupt set-pending register |
| 0xE000.E280 | NVIC_ICPR | NVIC->ICPR[0] | Interrupt clear-pending register |
| 0xE000.E400 | NVIC_IPR0 | NVIC->IP[0] | Interrupt priority register (0-3) |
| 0xE000.E404 | NVIC_IPR1 | NVIC->IP[1] | Interrupt priority register (4-7) |
| 0xE000.E408 | NVIC_IPR2 | NVIC->IP[2] | Interrupt priority register (8-11) |
| 0xE000.E40C | NVIC_IPR3 | NVIC->IP[3] | Interrupt priority register (12-15) |
| 0xE000.E410 | NVIC_IPR4 | NVIC->IP[4] | Interrupt priority register (16-19) |
| 0xE000.E414 | NVIC_IPR5 | NVIC->IP[5] | Interrupt priority register (20-23) |
| 0xE000.E418 | NVIC_IPR6 | NVIC->IP[6] | Interrupt priority register (24-27) |
| 0xE000.E41C | NVIC_IPR7 | NVIC->IP[7] | Interrupt priority register (28-31) |

#### 3.3.1.2 Interrupt Groups

To support mapping of more than 32 peripheral interrupt sources to the NVIC, certain MSPM0 devices include interrupt grouping logic (INT_GROUP) in the MCPUSS to combine several interrupts together to source one native NVIC interrupt.

The INT_GROUP interrupt grouping uses the MSPM0 event management register structure with the key difference being that all peripheral interrupt sources to the interrupt group are always unmasked (always enabled) such that no additional enable configuration is needed beyond the peripheral interrupt configuration and the NVIC configuration. The IMASK register itself is read-only and hardwired to enable all sources to the interrupt group. Figure 3-4 shows the INT_GROUP structure.

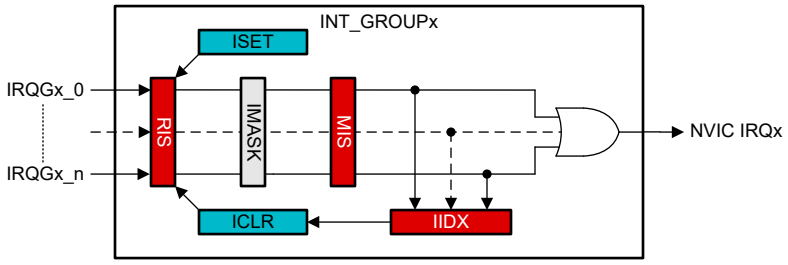



**Figure 3-4. Interrupt Group (INT_GROUP)**

Because no masking control is required, application software only needs to interface with the interrupt index (IIDX) register to efficiently handle a peripheral interrupt which is registered to the NVIC through an INT_GROUP.

Application software can read the IIDX register in the INT_GROUP to determine and clear the highest priority pending peripheral interrupt in the group. A read to IIDX will return an index corresponding to the highest priority peripheral which set an interrupt. The read action will also simultaneously clear the RIS and MIS bits



Copyright © 2026 Texas Instruments Incorporated

corresponding to the highest priority interrupt whose index was returned by the read. The value read from the IIDX register can then be used in a case statement, as shown below.

```
void GROUP_HANDLER(void)
{
  switch(IIDX)
  {
    case 0:         // no IRQ pending
      break;
    case 1:         // IRQ[0]
      do_peripheral_1_ISR();
      break;
    case 2:         // IRQ[1]
      do_peripheral_2_ISR();
      break;
    default:        // out of range
      illegal();
  }
}
```

**Usage**

Because peripheral interrupts that are grouped together into an INT_GROUP source a single NVIC interrupt, it is not possible to have one peripheral interrupt in a group preempt the execution of an active handler for the interrupt group. For example, take the scenario where the WWDT0 interrupt request line and the PMCU interrupt request line are connected to INT_GROUP0, and INT_GROUP0 sources NVIC peripheral interrupt 0. If the WWDT0 interrupt is asserted, INT_GROUP0 will assert an interrupt request to the NVIC. If no higher priority interrupt is active, the NVIC will vector the processor to the INT_GROUP0 handler. Application software can then read the INT_GROUP0 IIDX register to determine that it was the WWDT0 that triggered the INT_GROUP0 interrupt on the NVIC, and software can jump to the WWDT0 handler function.

If the PMCU asserts its interrupt line while the processor is still in handler mode servicing the WWDT0 function (which is really a part of the INT_GROUP0 handler), the PMCU interrupt can not preempt the WWDT0 handler. When the WWDT0 handler completes, The INT_GROUP0 handler will return. At that time, the processor will see that the INT_GROUP0 request was asserted again (this time, due to the PMCU), and it will tail-chain a second entry to the interrupt handler. This time, application software will read the IIDX and determine that the PMCU was the cause of the INT_GROUP0 interrupt being asserted to the NVIC.

If there are two or more peripheral interrupts pending to a single interrupt group, software can set the priority with which the interrupts are handled by first reading the RIS or MIS register to test which peripheral interrupts are asserted, followed by executing the software-determined priority. Alternatively, if the interrupt index (IIDX) register is used, the interrupt group hardware will return the highest priority index based on the index order.

#### 3.3.1.3 Wake Up Controller (WUC)

The wake up controller (WUC) is responsible for monitoring for assertion of interrupts when the processor is powered down in the STOP or STANDBY operating mode. In these modes, the entire PD1 power domain is power gated, and as such, the processor and NVIC are not available to check for interrupts. The WUC retains a copy of which peripheral interrupt sources to the NVIC were enabled when the processor entered STOP or STANDBY mode. In the event that an enabled interrupt is issued, the WUC will handshake with the PMCU to bring the device out of STOP or STANDBY mode so that the CPU can service the interrupt. The WUC will capture the interrupt state and present it to the NVIC and processor when the processor is brought up, such that the processor will see the interrupt even if the raw interrupt status of the peripheral is removed before the processor finishes powering up to service the interrupt.

The WUC requires no configuration by application software upon entry to or exit from low-power modes, and operation is transparent to application software.

### *3.3.2 Interrupt and Exception Table*

The Arm Cortex-M0+ interrupt vector table is 48 words long (192 bytes). The complete platform interrupt and exception table with vector table addresses is given in Table 3-3.



Copyright © 2026 Texas Instruments Incorporated

See the device-specific data sheet for a complete list of which interrupts a particular device supports.

**Table 3-3. MSPM0 Platform Processor Interrupt and Exception Table**

| Exception Number | NVIC Number[1] | Priority Group | Exception or Interrupt | Vector Table Address | Vector Description |
|---|---|---|---|---|---|
| - | - | - | - | 0x0000.0000 | Stack pointer |
| 1 | - | -3 | Reset | 0x0000.0004 | Reset vector |
| 2 | - | -2 | NMI | 0x0000.0008 | NMI handler |
| 3 | - | -1 | Hard fault | 0x0000.000C | Hard fault handler |
| 4 | - | - | Reserved | 0x0000.0010 | - |
| 5 | - | - | Reserved | 0x0000.0014 | - |
| 6 | - | - | Reserved | 0x0000.0018 | - |
| 7 | - | - | Reserved | 0x0000.001C | - |
| 8 | - | - | Reserved | 0x0000.0020 | - |
| 9 | - | - | Reserved | 0x0000.0024 | - |
| 10 | - | - | Reserved | 0x0000.0028 | - |
| 11 | - | Selectable | SVCall | 0x0000.002C | Supervisor call handler |
| 12 | - | - | Reserved | 0x0000.0030 | - |
| 13 | - | - | Reserved | 0x0000.0034 | - |
| 14 | - | Selectable | PendSV | 0x0000.0038 | Pended supervisor handler |
| 15 | - | Selectable | SysTick | 0x0000.003C | SysTick handler |
| 16 | 0 | Selectable | Device Interrupt 0 | 0x0000.0040 | Handler for Device Interrupt 0 |
| 17 | 1 | Selectable | Device Interrupt 1 | 0x0000.0044 | Handler for Device Interrupt 1 |
| 18 | 2 | Selectable | Device Interrupt 2 | 0x0000.0048 | Handler for Device Interrupt 2 |
| 19 | 3 | Selectable | Device Interrupt 3 | 0x0000.004C | Handler for Device Interrupt 3 |
| 20 | 4 | Selectable | Device Interrupt 4 | 0x0000.0050 | Handler for Device Interrupt 4 |
| 21 | 5 | Selectable | Device Interrupt 5 | 0x0000.0054 | Handler for Device Interrupt 5 |
| 22 | 6 | Selectable | Device Interrupt 6 | 0x0000.0058 | Handler for Device Interrupt 6 |
| 23 | 7 | Selectable | Device Interrupt 7 | 0x0000.005C | Handler for Device Interrupt 7 |
| 24 | 8 | Selectable | Device Interrupt 8 | 0x0000.0060 | Handler for Device Interrupt 8 |
| 25 | 9 | Selectable | Device Interrupt 9 | 0x0000.0064 | Handler for Device Interrupt 9 |
| 26 | 10 | Selectable | Device Interrupt 10 | 0x0000.0068 | Handler for Device Interrupt 10 |
| 27 | 11 | Selectable | Device Interrupt 11 | 0x0000.006C | Handler for Device Interrupt 11 |
| 28 | 12 | Selectable | Device Interrupt 12 | 0x0000.0070 | Handler for Device Interrupt 12 |
| 29 | 13 | Selectable | Device Interrupt 13 | 0x0000.0074 | Handler for Device Interrupt 13 |
| 30 | 14 | Selectable | Device Interrupt 14 | 0x0000.0078 | Handler for Device Interrupt 14 |
| 31 | 15 | Selectable | Device Interrupt 15 | 0x0000.007C | Handler for Device Interrupt 15 |
| 32 | 16 | Selectable | Device Interrupt 16 | 0x0000.0080 | Handler for Device Interrupt 16 |
| 33 | 17 | Selectable | Device Interrupt 17 | 0x0000.0084 | Handler for Device Interrupt 17 |
| 34 | 18 | Selectable | Device Interrupt 18 | 0x0000.0088 | Handler for Device Interrupt 18 |
| 35 | 19 | Selectable | Device Interrupt 19 | 0x0000.008C | Handler for Device Interrupt 19 |
| 36 | 20 | Selectable | Device Interrupt 20 | 0x0000.0090 | Handler for Device Interrupt 20 |
| 37 | 21 | Selectable | Device Interrupt 21 | 0x0000.0094 | Handler for Device Interrupt 21 |
| 38 | 22 | Selectable | Device Interrupt 22 | 0x0000.0098 | Handler for Device Interrupt 22 |
| 39 | 23 | Selectable | Device Interrupt 23 | 0x0000.009C | Handler for Device Interrupt 23 |
| 40 | 24 | Selectable | Device Interrupt 24 | 0x0000.00A0 | Handler for Device Interrupt 24 |
| 41 | 25 | Selectable | Device Interrupt 25 | 0x0000.00A4 | Handler for Device Interrupt 25 |
| 42 | 26 | Selectable | Device Interrupt 26 | 0x0000.00A8 | Handler for Device Interrupt 26 |



Copyright © 2026 Texas Instruments Incorporated

**Table 3-3. MSPM0 Platform Processor Interrupt and Exception Table (continued)**

| Exception Number | NVIC Number(1) | Priority Group | Exception or Interrupt | Vector Table Address | Vector Description |
|---|---|---|---|---|---|
| 43 | 27 | Selectable | Device Interrupt 27 | 0x0000.00AC | Handler for Device Interrupt 27 |
| 44 | 28 | Selectable | Device Interrupt 28 | 0x0000.00B0 | Handler for Device Interrupt 28 |
| 45 | 29 | Selectable | Device Interrupt 29 | 0x0000.00B4 | Handler for Device Interrupt 29 |
| 46 | 30 | Selectable | Device Interrupt 30 | 0x0000.00B8 | Handler for Device Interrupt 30 |
| 47 | 31 | Selectable | Device Interrupt 31 | 0x0000.00BC | Handler for Device Interrupt 31 |

(1) The NVIC number also indicates the relative interrupt priority if multiple NVIC interrupts have the same group priority. However, an interrupt does not preempt an active handler for another interrupt with the same group priority, even if the interrupt has a higher (numerically lower) NVIC position. For preemption to occur, the new interrupt must be configured to a higher priority group (numerically lower).

### Non-Maskable Interrupt (NMI)

The CPU implements a nonmaskable interrupt, which handles critical interrupts which must be serviced immediately by the processor. The NMI interrupt sources are managed by SYSCTL. See the corresponding NMI information in the SYSCTL section of the PMCU chapter.

### INT_GROUP0 Peripheral Interrupt Group

The INT_GROUP0 peripheral interrupt group sources an interrupt to NVIC0 (exception 16) if any peripheral in the group has a pending interrupt. The peripheral interrupts mapped to INT_GROUP0 are given in Table 3-4.

**Note**

Not all of the devices support Interrupt Group or NMI (non-maskable interrupt). Refer to the device-specific data sheet to see which devices support them.

**Table 3-4. INT_GROUP0 Interrupts**

| Priority | IIDX Index | Interrupt | Description |
|---|---|---|---|
| 0 | 1 | WWDT0 | WWDT0 interrupt handler |
| 1 | 2 | WWDT1 | WWDT1 interrupt handler |
| 2 | 3 | DEBUGSS | Debug subsystem interrupt handler |
| 3 | 4 | FLASHCTL | Flash controller interrupt handler |
| 4 | 5 | WUC FSUB0 | Generic event subscriber 0 interrupt handler |
| 5 | 6 | WUC FSUB1 | Generic event subscriber 1 interrupt handler |
| 6 | 7 | PMCU (SYSCTL) | PMCU (system controller) interrupt handler |
| 7 | 8 | Reserved | - |

### INT_GROUP1 Peripheral Interrupt Group

The INT_GROUP1 peripheral interrupt group sources an interrupt to NVIC1 (exception 17) if any peripheral in the group has a pending interrupt. The peripheral interrupts mapped to INT_GROUP1 are given in Table 3-5.

**Note**

Not all of the devices support Interrupt Group or NMI (non-maskable interrupt). Refer to the device-specific data sheet to see which devices support them.

**Table 3-5. INT_GROUP1 Interrupts**

| Priority | IIDX Index | Interrupt | Description |
|---|---|---|---|
| 0 | 1 | GPIO0 | GPIO0 interrupt handler |
| 1 | 2 | GPIO1 | GPIO1 interrupt handler |



Copyright © 2026 Texas Instruments Incorporated

**Table 3-5. INT_GROUP1 Interrupts (continued)**

| Priority | IIDX Index | Interrupt | Description |
|---|---|---|---|
| 2 | 3 | COMP0 | COMP0 interrupt handler |
| 3 | 4 | COMP1 | COMP1 interrupt handler |
| 4 | 5 | COMP2 | COMP2 interrupt handler |
| 5 | 6 | TRNG | TRNG interrupt handler |
| 6 | 7 | Reserved | - |
| 7 | 8 | Reserved | - |

### *3.3.3 Processor Lockup Scenario*

There are several exception conditions which can cause the processor to enter a lockup state. On MSPM0 devices, a processor lockup is considered a fatal fault which always triggers a SYSRST to clear the lockup condition and restart the system.

A lockup state is entered by the processor if an SVC (supervisor call) or fault condition occurs while the processor is handling an exception with priority of -1 or higher (numerically lower). Such a fault is considered by the processor to be unexpected under normal operating conditions.

The following examples are conditions that can trigger a lockup state in the processor:

- The processor cannot fetch the stack pointer or reset vector at reset
- The processor cannot fetch the NMI vector
- The processor cannot fetch the hard fault vector
- A memory fault occurs when the processor is already handling an exception with priority of -1 or -2 (hard fault or NMI)
- A supervisor call (SVC) occurs when the processor is already handling an exception with priority of -1 or -2 (hard fault or NMI)
- A usage fault or undefined instruction is fetched when the processor is already handling an exception with priority of -1 or -2 (hard fault or NMI)
- A BKPT instruction is executed when the processor is already handling an exception with priority of -1 or -2 (hard fault or NMI)

## 3.4 CPU Peripherals

The Arm Cortex-M0+ includes tightly coupled peripherals for system timing and memory protection.

### *3.4.1 System Control Block (SCB)*

The system control block (SCB) provides system implementation information and system control functionality, as well as configuration, control, and reporting of processor exceptions.

The SCB is configured through memory-mapped registers in the system private peripheral bus (PPB) region. See Table 3-6 for the list of SCB registers. The software development kit (SDK) provided with the devices supports the standard Arm Cortex Microcontroller Software Interface Standard (CMSIS) register access definitions for the SCB. Application software must use 32-bit aligned, word-size transactions when accessing any SCB register.

**Table 3-6. Arm Cortex-M0+ System Control Block Registers**

| Address | Register | CMSIS | Description |
|---|---|---|---|
| 0xE000.ED00 | CPUID | SCB->CPUID | Read-only register indicating the CPU type and revision |
| 0xE000.ED04 | ICSR | SCB->ICSR | Provides specific interrupt controls and state |
| 0xE000.ED08 | VTOR | SCB->VTOR | Used to specify the vector table offset from 0x0000.0000 |
| 0xE000.ED0C | AIRCR | SCB->AIRCR | Used to issue a CPU reset request (SYSRESETREQ) |
| 0xE000.ED10 | SCR | SCB->SCR | System control register, used to control low-power mode behavior |
| 0xE000.ED14 | CCR | SCB->CCR | Read-only register indicating behavior of the processor |
| 0xE000.ED1C | SHPR2 | SCB->SHP[2] | Used to configure the priority of the SVCall system handler |



Copyright © 2026 Texas Instruments Incorporated

**Table 3-6. Arm Cortex-M0+ System Control Block Registers (continued)**

| Address | Register | CMSIS | Description |
|---|---|---|---|
| 0xE000.ED20 | SHPR3 | SCB->SHP[3] | Used to configure the priority of the SysTickand PendSV system handlers |

For detailed information on the system control block register configuration, see the SCB section of the Arm Cortex-M0+ devices generic user guide.

### *3.4.2 System Tick Timer (SysTick)*

The system tick timer (SysTick) is a tightly-coupled timer clocked by MCLK, which can be used for system time keeping. The SysTick is available in RUN and SLEEP modes but is not available for use in STOP, STANDBY, or SHUTDOWN modes.

The SysTick is configured through memory-mapped registers in the system private peripheral bus (PPB) region. See Table 3-7 for the list of SysTick configuration registers. The software development kit (SDK) provided with the devices supports the standard Arm Cortex Microcontroller Software Interface Standard (CMSIS) register access definitions for the SysTick.

The SysTick timer can be used in several different ways, including:

- As an RTOS timer which fires at a programmable rate (for example, 100Hz) and invokes a SysTick routine
- A high-speed alarm timer
- A simple counter used to measure time to completion and time used in an application
- A timeout counter to check that a routine has not timed out within a specified period

The SysTick timer is a simple 24-bit down counter, which counts down from its reload value to zero. Upon reaching zero, SysTick will re-load the value programmed into the reload value register (SYST_RVR) on the next clock cycle, and then again begin counting down to zero.

A SysTick event is generated when the SysTick counter reaches zero, and which point the COUNTFLAG status flag will be set. Reading the SYST_CSR register will clear the COUNTFLAG status bit. Writing to the current value register (SYST_CVR) clears the register and the COUNTFLAG status but does not generate an interrupt to the CPU. Reading SYST_CVR returns the counter value at time of access.

**Table 3-7. SysTick Registers**

| Address | Register | CMSIS | Description |
|---|---|---|---|
| 0xE000.E010 | SYST_CSR | SysTick->CTRL | Control and status register, used to enable/disable SysTick and its related exception, and to check the COUNTFLAG status |
| 0xE000.E014 | SYST_RVR | SysTick->LOAD | Reload register used to program the counter reload value to set the SysTick interval in MCLK cycles |
| 0xE000.E018 | SYST_CVR | SysTick->VAL | Returns the current value of the SysTick counter; a write clears the counter to zero and clears COUNTFLAG in SYST_CSR |
| 0xE000.E01C | SYST_CALIB | SysTick->CALIB | Not implemented |

Application software must only use 32-bit word-aligned word accesses to the SysTick registers. To initialize the SysTick, follow the steps below:

1. Program the desired reload value (example: to generate a flag every 1000 MCLK cycles, program 999) to SYST_RVR
2. Clear the current value by writing to the SYST_CVR register
3. Program the SYST_CSR register to enable SysTick

**Note**

The SysTick counter does not decrement when the CPU is halted for debug.



Copyright © 2026 Texas Instruments Incorporated

### 3.4.3 Memory Protection Unit (MPU)

The memory protection unit (MPU) can be used to check all memory accesses made by the processor against a set of access permission policies which can be defined by the programmer. When used together with the privileged/unprivileged execution modes of the Arm Cortex-M0+, the MPU supports limiting access to certain memory locations to privileged code only. If unprivileged code accesses a nonrestricted region, execution continues as if the MPU was not present. However, if unprivileged code issues an access to a restricted region, a hard fault is generated in the processor. It is also possible to restrict access to both privileged and unprivileged code (no access possible through the processor).

The MPU is configured through memory-mapped registers in the system private peripheral bus (PPB) region. See Table 3-8 for the list of MPU configuration registers. The software development kit (SDK) provided with the devices supports the standard Arm Cortex Microcontroller Software Interface Standard (CMSIS) register access definitions for the MPU.

Because the MPU checks all memory accesses from the processor (including accesses to flash, SRAM, and peripherals), it is well suited for improving reliability and robustness in threaded applications involving an RTOS. The MPU can be used to restrict the memory access of individual threads, including establishing stack boundaries and limiting access to specific peripherals. Key state data used by the RTOS can be protected from modification by unprivileged threads.

**Note**

The MPU only monitors access to the memory map which originate from the processor. The MPU does not restrict the access of the DMA controller.

**Note**

Protecting flash memory regions with the MPU does not protect the flash memory from being read or modified by the flash controller. This is because the flash controller can take control of the flash memory banks directly for read verify, program, and erase operations. Dedicated write protection functions are provided within the flash controller, and these mechanisms are for use when write protecting the flash memory. Software can configure the MPU to restrict access to the flash controller registers, thus preventing the CPU from configuring or initiating a flash command directly.

The MPU provides a mechanism to partition the device memory map into 8 regions (numbered 0-7) plus a default background region. Each region can be configured with access permissions and memory attributes, and regions can be configured to overlap if desired. In the case of overlapping regions, a memory access to a location existing in multiple regions is subjected to the attributes of the region with the highest number.

When the MPU is not enabled (ENABLE bit is cleared in the MPU_CTRL register), the device uses the default memory map and the CPU has access to the memory map as if the MPU was not present.

When the MPU is enabled for use, access to the vector table and the system control space are always permitted, but access to any other location depends on the following:

- The region configurations
- Whether the access was privileged or unprivileged
- The privileged software default memory map access control configuration (PRIVDEFENA)

Because the Arm Cortex-M0+ is a single-bus CPU architecture, there is no delineation between an instruction fetch and a data access by the MPU. Instructions and data are treated the same.

**Note**

If the MPU is enabled (ENABLE bit is set), then PRIVDEFENA must be set to give privileged software access to the memory map, or at least one memory region must be configured and enabled for the CPU to proceed without a hard fault. If PRIVDEFENA is cleared, no regions are defined and enabled, and the MPU is enabled, then the entire memory map is restricted, and any access generates a fault.



Copyright © 2026 Texas Instruments Incorporated

**Table 3-8. MPU Registers**

| Address | Register | CMSIS | Description |
|---|---|---|---|
| 0xE000.ED90 | MPU_TYPE | MPU->TYPE | Type register indicating that the MPU is present. |
| 0xE000.ED94 | MPU_CTRL | MPU->CTRL | MPU control register for enabling and configuring the MPU. |
| 0xE000.ED98 | MPU_RNR | MPU->RNR | Region select register for using MPU_RBAR and MPU_RASR. |
| 0xE000.ED9C | MPU_RBAR | MPU->RBAR | Region base address configuration register. |
| 0xE000.EDA0 | MPU_RASR | MPU->RASR | Region size and memory attributes register. |

For detailed information on the MPU register configuration, see the MPU section of the Arm Cortex-M0+ Devices Generic User Guide.

## 3.5 Read-Only Memory (ROM)

The MCPUSS contains a read-only memory which contains the executable code for the boot configuration routine (BCR) and bootstrap loader (BSL). The ROM is active after a BOOTRST, or after a SYSRST with BSL entry/exit. The ROM is disabled automatically when the application is started and is not accessible by application software.



Copyright © 2026 Texas Instruments Incorporated

## 3.6 CPUSS Registers

Table 3-9 lists the memory-mapped registers for the CPUSS registers. All register offset addresses not listed in Table 3-9 should be considered as reserved locations and the register contents should not be modified.

**Table 3-9. CPUSS Registers**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 10E0h | EVT_MODE | Event Mode | | Go |
| 10FCh | DESC | Module Description | | Go |
| 1100h | IIDX | Interrupt index | CPU_INT_GROUP0 | Go |
| 1108h | IMASK | Interrupt mask | CPU_INT_GROUP0 | Go |
| 1110h | RIS | Raw interrupt status | CPU_INT_GROUP0 | Go |
| 1118h | MIS | Masked interrupt status | CPU_INT_GROUP0 | Go |
| 1120h | ISET | Interrupt set | CPU_INT_GROUP0 | Go |
| 1128h | ICLR | Interrupt clear | CPU_INT_GROUP0 | Go |
| 1130h | IIDX | Interrupt index | CPU_INT_GROUP1 | Go |
| 1138h | IMASK | Interrupt mask | CPU_INT_GROUP1 | Go |
| 1140h | RIS | Raw interrupt status | CPU_INT_GROUP1 | Go |
| 1148h | MIS | Masked interrupt status | CPU_INT_GROUP1 | Go |
| 1150h | ISET | Interrupt set | CPU_INT_GROUP1 | Go |
| 1158h | ICLR | Interrupt clear | CPU_INT_GROUP1 | Go |
| 1300h | CTL | Prefetch/Cache control | | Go |

Complex bit access types are encoded to fit into small table cells. Table 3-10 shows the codes that are used for access types in this section.

**Table 3-10. CPUSS Access Type Codes**

| Access Type | Code | Description |
|---|---|---|
| Read Type | | |
| R | R | Read |
| Write Type | | |
| W | W | Write |
| Reset or Default Value | | |
| *-n* | | Value after reset or the default value |



Copyright © 2026 Texas Instruments Incorporated

### 3.6.1 EVT_MODE (Offset = 10E0h) [Reset = 00000000h]

EVT_MODE is shown in Figure 3-5 and described in Table 3-11.

Return to the Summary Table.

Event mode register. It is used to select whether each line is disabled, in software mode (software clears the RIS) or in hardware mode (hardware clears the RIS)

**Figure 3-5. EVT_MODE**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | INT_CFG | |
| R-0h | | | | | | R-0h | |

**Table 3-11. EVT_MODE Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | |
| 1-0 | INT_CFG | R | 1h | Event line mode select<br>0h = The interrupt or event line is disabled.<br>1h = Event handled by software. Software must clear the associated RIS flag.<br>2h = Event handled by hardware. The hardware (another module) clears automatically the associated RIS flag. |



Copyright © 2026 Texas Instruments Incorporated

### 3.6.2 DESC (Offset = 10FCh) [Reset = 00000000h]

DESC is shown in Figure 3-6 and described in Table 3-12.

Return to the Summary Table.

This register identifies the peripheral and its exact version.

**Figure 3-6. DESC**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MODULEID | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| FEATUREVER | | | | RESERVED | | | | MAJREV | | | | MINREV | | | |
| R-0h | | | | R-0h | | | | R-0h | | | | R-0h | | | |

**Table 3-12. DESC Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | MODULEID | R | 2711h | Module identification contains a unique peripheral identification number. The assignments are maintained in a central database for all of the platform modules to ensure uniqueness. |
| 15-12 | FEATUREVER | R | 0h | Feature Set for the module *instance* |
| 11-8 | RESERVED | R | 0h | |
| 7-4 | MAJREV | R | 0h | Major rev of the IP |
| 3-0 | MINREV | R | 0h | Minor rev of the IP |



Copyright © 2026 Texas Instruments Incorporated

### 3.6.3 IIDX (Offset = 1100h) [Reset = 00000000h]

IIDX is shown in Figure 3-7 and described in Table 3-13.

Return to the Summary Table.

Interrupt index register. This read-only register provides the interrupt index of the pending interrupt with the highest priority. It also indicates if no interrupt is pending. The priority order is fixed: lower index equals higher priority. Alternatively to the use of IIDX, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred.
On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flags in the RIS and MIS are cleared as well. After a read from the CPU (not from the debug interface), the register must be updated with the next highest priority interrupt or indicate that no interrupt is pending. Only interrupts which are selected via IMASK are indicated.

**Figure 3-7. IIDX**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | STAT | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | |

**Table 3-13. IIDX Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | STAT | R | 0h | Interrupt index status<br>0h = No pending interrupt<br>1h = Interrupt 0<br>2h = Interrupt 1<br>3h = Interrupt 2<br>4h = Interrupt 3<br>5h = Interrupt 4<br>6h = Interrupt 5<br>7h = Interrupt 6<br>8h = Interrupt 7 |



Copyright © 2026 Texas Instruments Incorporated

### 3.6.4 IMASK (Offset = 1108h) [Reset = 000000FFh]

IMASK is shown in Figure 3-8 and described in Table 3-14.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS."

**Figure 3-8. IMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | INT | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R/W-FFh | | | | | | | |

**Table 3-14. IMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | INT | R/W | FFh | Masks the corresponding interrupt<br>1h = Interrupt 0<br>2h = Interrupt 1<br>4h = Interrupt 2<br>8h = Interrupt 3<br>10h = Interrupt 4<br>20h = Interrupt 5<br>40h = Interrupt 6<br>80h = Interrupt 7 |



Copyright © 2026 Texas Instruments Incorporated

### 3.6.5 RIS (Offset = 1110h) [Reset = 00000000h]

RIS is shown in Figure 3-9 and described in Table 3-15.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Figure 3-9. RIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | INT | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | |

**Table 3-15. RIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | INT | R | 0h | Raw interrupt status for INT<br>1h = Interrupt 0<br>2h = Interrupt 1<br>4h = Interrupt 2<br>8h = Interrupt 3<br>10h = Interrupt 4<br>20h = Interrupt 5<br>40h = Interrupt 6<br>80h = Interrupt 7 |



Copyright © 2026 Texas Instruments Incorporated

### 3.6.6 MIS (Offset = 1118h) [Reset = 00000000h]

MIS is shown in Figure 3-10 and described in Table 3-16.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 3-10. MIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | INT | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | |

**Table 3-16. MIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | INT | R | 0h | Masked interrupt status for INT0<br>1h = Interrupt 0<br>2h = Interrupt 1<br>4h = Interrupt 2<br>8h = Interrupt 3<br>10h = Interrupt 4<br>20h = Interrupt 5<br>40h = Interrupt 6<br>80h = Interrupt 7 |



Copyright © 2026 Texas Instruments Incorporated

### 3.6.7 ISET (Offset = 1120h) [Reset = 00000000h]

ISET is shown in Figure 3-11 and described in Table 3-17.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 3-11. ISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | INT | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | W-0h | | | | | | | |

**Table 3-17. ISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | INT | W | 0h | Sets INT in RIS register<br>1h = Interrupt 0<br>2h = Interrupt 1<br>4h = Interrupt 2<br>8h = Interrupt 3<br>10h = Interrupt 4<br>20h = Interrupt 5<br>40h = Interrupt 6<br>80h = Interrupt 7 |



Copyright © 2026 Texas Instruments Incorporated

### 3.6.8 ICLR (Offset = 1128h) [Reset = 00000000h]

ICLR is shown in Figure 3-12 and described in Table 3-18.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

**Figure 3-12. ICLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | INT | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | W-0h | | | | | | | |

**Table 3-18. ICLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | INT | W | 0h | Clears INT in RIS register<br>1h = Interrupt 0<br>2h = Interrupt 1<br>4h = Interrupt 2<br>8h = Interrupt 3<br>10h = Interrupt 4<br>20h = Interrupt 5<br>40h = Interrupt 6<br>80h = Interrupt 7 |



Copyright © 2026 Texas Instruments Incorporated

### 3.6.9 IIDX (Offset = 1130h) [Reset = 00000000h]

IIDX is shown in Figure 3-13 and described in Table 3-19.

Return to the Summary Table.

Interrupt index register. This read-only register provides the interrupt index of the pending interrupt with the highest priority. It also indicates if no interrupt is pending. The priority order is fixed: lower index equals higher priority. Alternatively to the use of IIDX, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred.
On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flags in the RIS and MIS are cleared as well. After a read from the CPU (not from the debug interface), the register must be updated with the next highest priority interrupt or indicate that no interrupt is pending. Only interrupts which are selected via IMASK are indicated.

**Figure 3-13. IIDX**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | STAT | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | |

**Table 3-19. IIDX Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | STAT | R | 0h | Interrupt index status<br>0h = No pending interrupt<br>1h = Interrupt 0<br>2h = Interrupt 1<br>3h = Interrupt 2<br>4h = Interrupt 3<br>5h = Interrupt 4<br>6h = Interrupt 5<br>7h = Interrupt 6<br>8h = Interrupt 7 |



Copyright © 2026 Texas Instruments Incorporated

### 3.6.10 IMASK (Offset = 1138h) [Reset = 000000FFh]

IMASK is shown in Figure 3-14 and described in Table 3-20.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS."

**Figure 3-14. IMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | INT | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R/W-FFh | | | | | | | |

**Table 3-20. IMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | INT | R/W | FFh | Masks the corresponding interrupt<br>1h = Interrupt 0<br>2h = Interrupt 1<br>4h = Interrupt 2<br>8h = Interrupt 3<br>10h = Interrupt 4<br>20h = Interrupt 5<br>40h = Interrupt 6<br>80h = Interrupt 7 |



Copyright © 2026 Texas Instruments Incorporated

### 3.6.11 RIS (Offset = 1140h) [Reset = 00000000h]

RIS is shown in Figure 3-15 and described in Table 3-21.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Figure 3-15. RIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | INT | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | |

**Table 3-21. RIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | INT | R | 0h | Raw interrupt status for INT<br>1h = Interrupt 0<br>2h = Interrupt 1<br>4h = Interrupt 2<br>8h = Interrupt 3<br>10h = Interrupt 4<br>20h = Interrupt 5<br>40h = Interrupt 6<br>80h = Interrupt 7 |



Copyright © 2026 Texas Instruments Incorporated

### 3.6.12 MIS (Offset = 1148h) [Reset = 00000000h]

MIS is shown in Figure 3-16 and described in Table 3-22.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 3-16. MIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | INT | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | |

**Table 3-22. MIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | INT | R | 0h | Masked interrupt status for INT0<br>1h = Interrupt 0<br>2h = Interrupt 1<br>4h = Interrupt 2<br>8h = Interrupt 3<br>10h = Interrupt 4<br>20h = Interrupt 5<br>40h = Interrupt 6<br>80h = Interrupt 7 |



Copyright © 2026 Texas Instruments Incorporated

### 3.6.13 ISET (Offset = 1150h) [Reset = 00000000h]

ISET is shown in Figure 3-17 and described in Table 3-23.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 3-17. ISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | INT | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | W-0h | | | | | | | |

**Table 3-23. ISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | INT | W | 0h | Sets INT in RIS register<br>1h = Interrupt 0<br>2h = Interrupt 1<br>4h = Interrupt 2<br>8h = Interrupt 3<br>10h = Interrupt 4<br>20h = Interrupt 5<br>40h = Interrupt 6<br>80h = Interrupt 7 |



Copyright © 2026 Texas Instruments Incorporated

### 3.6.14 ICLR (Offset = 1158h) [Reset = 00000000h]

ICLR is shown in Figure 3-18 and described in Table 3-24.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

**Figure 3-18. ICLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | INT | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | W-0h | | | | | | | |

**Table 3-24. ICLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | INT | W | 0h | Clears INT in RIS register<br>1h = Interrupt 0<br>2h = Interrupt 1<br>4h = Interrupt 2<br>8h = Interrupt 3<br>10h = Interrupt 4<br>20h = Interrupt 5<br>40h = Interrupt 6<br>80h = Interrupt 7 |



Copyright © 2026 Texas Instruments Incorporated

### 3.6.15 CTL (Offset = 1300h) [Reset = 00000007h]

CTL is shown in Figure 3-19 and described in Table 3-25.

Return to the Summary Table.

Flash prefetch and cache control register.

**Figure 3-19. CTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | LITEN | ICACHE | PREFETCH |
| R-0h | | | | | R/W-1h | R/W-1h | R/W-1h |

**Table 3-25. CTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | RESERVED | R | 0h | |
| 2 | LITEN | R/W | 1h | Literal caching and prefetch enable.<br>This bit is a subset of ICACHE/PREFETCH bit i.e. literal caching or literal prefetching will only happen if ICACHE or PREFETCH bits have been set respectively<br>When enabled, the cache and prefetcher structures inside CPUSS will cache and prefetch literals<br>When disabled, the cache and prefetcher structures inside CPUSS will not cache and prefetch literals<br>0h = Literal caching disabled<br>1h = Literal caching enabled |
| 1 | ICACHE | R/W | 1h | Used to enable/disable Instruction caching on flash access.<br>0h = Disable instruction caching.<br>1h = Enable instruction caching. |
| 0 | PREFETCH | R/W | 1h | Used to enable/disable instruction prefetch to Flash.<br>0h = Disable instruction prefetch.<br>1h = Enable instruction prefetch. |

## 3.7 WUC Registers

Table 3-26 lists the memory-mapped registers for the WUC registers. All register offset addresses not listed in Table 3-26 should be considered as reserved locations and the register contents should not be modified.

**Table 3-26. WUC Registers**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 400h | FSUB_0 | Subscriber Port 0 | FSUB_0 Register (Offset = 400h) [Reset = 00h] |
| 404h | FSUB_1 | Subscriber Port 1 | FSUB_1 Register (Offset = 404h) [Reset = 00h] |



Copyright © 2026 Texas Instruments Incorporated

### 3.7.1 FSUB_0 Register (Offset = 400h) [Reset = 00h]

FSUB_0 is shown in Figure 3-20 and described in Table 3-27.

Return to the Summary Table.

Subscriber port

**Figure 3-20. FSUB_0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | CHANID | | | |
| R/W-0h | | | | | | | | | | | | R/W-0h | | | |

**Table 3-27. FSUB_0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R/W | 0h | |
| 3-0 | CHANID | R/W | 0h | 0 = disconnected.<br>1-15 = connected to channel ID = CHANID.<br>0h = A value of 0 specifies that the event is not connected<br>Fh = Consult your device data sheet as the actual allowed maximum can be less than 15. |

### 3.7.2 FSUB_1 Register (Offset = 404h) [Reset = 00h]

FSUB_1 is shown in Figure 3-21 and described in Table 3-28.

Return to the Summary Table.

Subscriber port

**Figure 3-21. FSUB_1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | CHANID | | | |
| R/W-0h | | | | | | | | | | | | R/W-0h | | | |

**Table 3-28. FSUB_1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R/W | 0h | |
| 3-0 | CHANID | R/W | 0h | 0 = disconnected.<br>1-15 = connected to channel ID = CHANID.<br>0h = A value of 0 specifies that the event is not connected<br>Fh = Consult your device data sheet as the actual allowed maximum can be less than15. |



Copyright © 2026 Texas Instruments Incorporated

Chapter 4
# SECURITY

## 4.1 Overview

Some MSPM0 devices offer several extensive cybersecurity features as part of the device architecture, including:

- Debug security
- Device identify
- Crypto acceleration
- True random number generation
- Flash write-erase protection
- Flash read-execute protection
- Flash IP protection
- SRAM write-execute mutual exclusion
- Secure boot
- Secure firmware update
- Secure key storage
- Customer secure code

The security architecture decouples mechanisms (controls offered by MSPM0 devices) from policies that are application-defined. This allows for a wide variety of security requirements to be supported.

The overall security configuration offered by MSPM0 devices is implemented in two steps:

1. Secure Boot **-** TI boot code executes on a BOOTRST and configures the requested security policies. Once boot code finishes execution, the system undergoes a SYSRST followed by the start of Step 2.
2. Customer Secure Code (CSC) - CSC executes from flash memory to further configure security elements. Note that CSC is customer-owned secure software. This two-step secure boot concept enables application-specific policies to be implemented and managed by CSC while standard security enablers are implemented by boot code. When Step 2 has completed execution, the system undergoes a second SYSRST followed by the start of the main application.

In this security model, both TI boot code and CSC are considered *trusted.* The main application is considered *untrusted.* When execution jumps to the main application, all security elements are expected to have been configured, including restrictions on debug, memory accesses, key management, and so on.

**Note**

As MSPM0 Security architecture provides a clean separation between policies and mechanisms, it is possible to build applications where no CSC is implemented and the main application execution begins right after BOOTRST.

**Note**

Check the device-specific data sheet to determine if the security architecture features are available.



Copyright © 2026 Texas Instruments Incorporated

#### 4.1.1 Secure Boot

In Step 1, boot ROM code configures a set of security elements based on contents of the configuration memory (NONMAIN). This includes:

- **Debug Security**: Debug access can be allowed unconditionally, disallowed unconditionally or allowed after a successful password verification
- **Mass Erase**: Mass erase operation can be allowed unconditionally, disallowed unconditionally or allowed after a successful password verification
- **Factory Reset**: Factory reset operation can be allowed unconditionally, disallowed unconditionally or allowed after a successful password verification
- **Write-protection of MAIN flash sectors**: MAIN bank contents can be write protected at 1x sector or 8x sector granularity
- **Write-protection of configuration memory (NONMAIN)**: Because boot configuration memory (NONMAIN) holds key security configuration settings, this configuration memory can be write protected to avoid inadvertent or malicious downgrading of security.
- **Presence/Absence of Customer Secure Code (CSC)**: This attribute controls whether a second level of security and trusted flash-based code is provisioned or not.
- **Bank Swap Policy**: In dual/quad-bank devices, this attribute indicates if the device operates the flash banks in a swap mode or not. If swap mode, then by hardware default, the upper bank (or pair) is enforced read-write access privileges and no execute privilege. The lower bank (or pair) is enforced read-execute access privileges and no write/erase privilege.
- **Delay Debug:** Security applications can disable debug access during the execution of the CSC. This attribute controls whether debug access is delayed until the end of CSC execution or not.

#### 4.1.2 Customer Secure Code (CSC)

In this step, customer owned secure code executes and implements additional security capabilities:

- Bank swap decision
- Secure firmware update
- Secure key storage
- Flash read-execute protection
- Flash IP protection
- SRAM write-execute mutual exclusion

CSC specifies which bank holds the more recent authenticated application image. If that bank is physical bank 0 (same bank as where CSC is executing from), then bank 1 is read-write only and does not have execute privilege. If the correct application image is determined to be on physical bank 1, then the CSC must issue a bank-swap request.

Besides image authentication, the CSC sets up additional security configurations that are described later in this document. The CSC indicates the end of CSC execution by writing to the SYSCTL.SECCFG.INITDONE register with a PASS value (0x1) along with a KEY value of 0x9d. Successfully writing to the INITDONE register results in a second SYSRST operation during which the bank-swap takes effect, as well as any additional security configurations.

### 4.2 Boot and Startup Sequence

Figure 4-1 illustrates the boot and startup sequence in security enabled applications. At BOOTRST, TI boot-code execution commences. After successful boot, boot-code issues BOOTDONE. At this point, SYSCTL issues a SYSRST to the device to trigger execution from flash memory. Depending on the boot configuration record, this leads either to the start of the main application (if CSC does not exist in this configuration) or to the start of the CSC (if CSC is configured).

CSC is responsible for determining execution bank, memory region protections, secure key initialization into the keystore, etc. When the customer secure code issues INITDONE (by writing to SYSCTL.SECCFG.INITDONE MMR), then SYSCTL issues a second SYSRST. The device again starts execution from 0x0 mapped to flash, and the CSC executes a second time. This time, the CSC will find that INITDONE has already been issued



Copyright © 2026 Texas Instruments Incorporated

previously (this is determined by reading the SYSCTL.SECCFG.SECSTATUS.INITDONE bit) and it directly calls the main application.

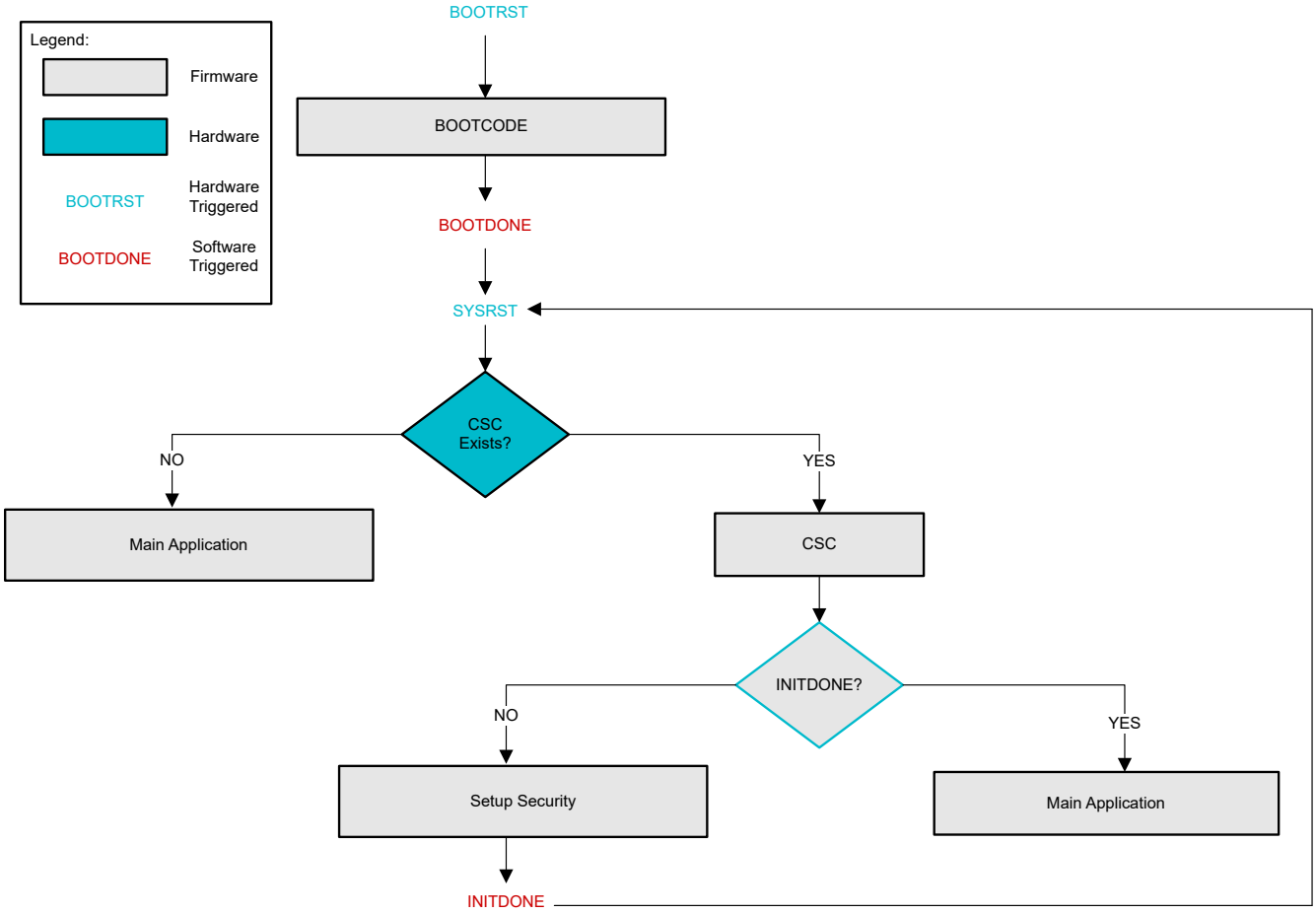



**Figure 4-1. Secure Boot and Start Sequence**

**Note**

To keep the boot flow compatible with nonsecurity enabled devices, the default settings of boot configuration are set to "CSC does not exist" state.

The secure execution flow is the path where CSC_EXISTS = YES. In this case, it may be observed that after BOOTRST, two SYSRSTs will be issued before the main application is launched. After first SYSRST, the customer startup code gets to execute. It configures security and issues INITDONE. At this point, the security configuration is locked and enforced. A second SYSRST is issued at this point, restarting startup code execution. At the second SYSRST, since INITDONE is YES, the main application is launched.

### *4.2.1 CSC Programming Overview*

The CSC is expected to be a separate image from the main application. This allows a complete isolation of the CSC from the application as opposed to a scheme where the CSC functionality is embedded into the application as a single image.

At the first SYSRST following BOOTDONE, CSC will be invoked via the reset handler at 0x0. The pseudo-code below provides an overview of the CSC programming model.

```
void resetHandler(void)
{
```



Copyright © 2026 Texas Instruments Incorporated

```
_asm("b __c_init00"); // this is the call to secure startup, which would issue INITDONE
}
```

The CSC will check if INITDONE has already been issued, and accordingly decide to perform application image authentication and security configuration before launching the authenticated image.

```
bool init_done = (*(volatile long *)(SYSCTL_SECCFG_SECTSTAT)) & 0x1;
If (! Init_done)
{
setupKeystorage(); // AES symmetric keys
entry_point = findImageEntryPoint ();
stack_ptr = findImageStackPtr();
setBankSwap(0 or 1); // depending on which bank the image is in

// setup SRAM boundary
copyFromFlashToSRAM();
setupSRAMBoundary();
lockSRAMBoundary();
setupFlashFirewalls();
INITDONE = 1; // *(volatile long *) (SYSCTL_SECCFG_INITDONE) = 1 | (0x9D << 24);
// This triggers a HW-initiated SYSRST
}

// we will come here if INITDONE = 1 after a SYSRST
launchApp(); // using entry_point and stack pointer base from image metadata
```

As outlined, the CSC would perform a set of security configuration actions that include:

- Transferring AES keys from flash memory to the secure key storage
- Identifying the correct application image to run and which bank it is located in. This step requires the startup code to use some pre-defined metadata scheme for where the app may be loaded, how to find the app version, how to authenticate the app, what its entry point is and where the app's vector table is located.
- Set up bank swap information
- Set up SRAM boundary to separate code region from RW region
- Provision required read/write/IP protect flash firewalls
- Mark INITDONE

Note that the SRAM boundary set up is optional and needed only if the application intends to run any code (especially interrupt handlers) out of SRAM. This is motivated by a requirement to keep communications ISRs running while the main flash memory could be busy with integrity checks.

The final step would trigger a second SYSRST and the reset handler gets invoked a second time. Since init_done is now found to be set, CSC will simply invoke the main application at this point, using the entry_point that the CSC obtained from the application image.

**Note**

Because the CSC enforces additional security policies over and above boot-code, and therefore an attack on CSC execution can disrupt intended security of the application, CSC development should follow the same best practices as TI boot-code. This includes analyzing and hardening CSC against glitching attacks. Also, because the CSC is customer-owned and the time it takes before calling INITDONE is variable, it is recommended that the CSC use a watchdog to ensure that INITDONE is called in a timely manner.

## 4.3 Secure Key Storage

Keys (private keys in asymmetric schemes such as RSA and symmetric schemes such as AES) need to be protected to ensure confidentiality. Only trusted code should be provisioned with a mechanism to securely deposit keys from flash memory to a location that only crypto engines can access. This is the concept of the MSPM0 secure key storage: secure storage that only the CSC can configure keys into. Subsequently, the main application can configure the crypto engine to use one of the stored keys but can never access any stored keys.



Copyright © 2026 Texas Instruments Incorporated

The key-store is housed as part of the key-store controller IP. Depending on the device, a certain number of key slots are provisioned. These slots can be used to hold some number of 256-bit and 128-bit keys. Trusted software (CSC) can write to the key-store to deposit keys securely. Writing to the key-store is automatically disabled when INITDONE is issued. Key-store contents are reset at BOOTRST and the store is unlocked for writing. Key-store contents are unaffected by SYSRST and lower order resets.

A key that is required for a crypto accelerator service (such as AES encryption) is transferred to the crypto engine when the application software configures the crypto service. The configuration is performed in the key-store controller and is of the form “transfer the 128-bit key in slot 3 to the AES engine”. This transfer is performed securely and is not visible to the application code, DMA or the debugger.

## 4.4 Flash Memory Protection

In order to meet diverse security requirements, various types of flash memory protection mechanisms are provided.

### *4.4.1 Bank Swapping*

In dual-bank/quad-bank devices, based on which bank (or pair) is executable, that bank (pair) gets read-execute privileges and loses write/erase privileges. The other bank (or pair) is readable, and writeable but not executable. This mechanism enforces the policy that any firmware update can only be saved in the writeable bank in the current session but can never be executed.

Upon a subsequent BOOTRST, the CSC runs and needs to authenticate the update and decide if the updated image should be made executable. If yes, then the bank holding the updated image can be configured to become executable (this is done by configuring the SYSCTL.SECCFG.FLBANKSWP.USEUPPER to 1 along with the correct KEY value of 0x58). When the CSC issues INITDONE, this bank-swap will take effect, swapping the execution/write privileges of the upper and lower banks as well as re-mapping the upper bank (or pair) to the lower half of the flash memory address space and re-mapping the upper bank (or pair) to the upper half of the flash memory address space.

Creating swappable configurations is a two-step process.

1. Boot configuration has to indicate that the flash memory will be used in a swappable configuration. Boot-code will read this configuration and write to the SYSCTL.SECCFG.FLBANKSWPPOLICY with the appropriate KEY (0xCA). Note that by hardware default, the swappable configuration is ENABLED.
2. CSC will determine which bank will be the executable bank. If physical bank 0 is going to be the execution bank, then no further action is needed. However, if physical bank 1 is going to be the execution bank, then a bank swap has to be issued by configuring the SYSCTL.SECCFG.FLBANKSWP.USEUPPER to 1 along with the correct KEY value of 0x58.

At any time, status of the bank swap policy and the currently executable bank can be obtained by reading the SYSCTL.SECCFG.SECSTATUS register. Fields FLBANKSWPPOLICY and FLBANKSWP provide this status.

### *4.4.2 Write Protection*

There are three levels of write-protection that are provided:

1. Write-protection that is enforced by TI boot-code
2. Write-protection that is enforced by CSC to further protect data that it is allowed to update but that should not be modified by the application
3. Write-protection in the context of bank swap (covered in the Bank Swapping section)

Write-protection enforced by TI boot-code can protect sectors of both the MAIN bank and the boot configuration data held in NONMAIN. The mechanisms are described in the Boot Configuration chapter. In particular, this capability is useful in cases where the CSC needs to be treated as an extension of the root-of-trust and made immutable.



Copyright © 2026 Texas Instruments Incorporated

The CSC can further extend the write protection to additional MAIN bank sectors. On every boot, the CSC can perform write/erase operations on sectors holding secure data/metadata and then lock these sectors from any modifications when the application executes.

Configuring write protection of the MAIN bank is done by setting sector-specific bits to 1 in the SYSCTL.SECCFG.FWEPROTMAIN register. The sectors that are already write-protected by boot-code continue to be write-protected. Note that only the first 32KB of flash memory can be configured with additional write protections by the CSC.

**Note**

In bank-swappable configuration, the protections are automatically mirrored to both banks.

### *4.4.3 Read-Execute Protection*

A region of flash memory can be configured for read-execute protection - read and instruction fetch accesses to this region will return an error. CPU, DMA and debugger accesses are all treated the same way. This is configured by writing to the SYSCTL.SECCFG.FRXPROTMAINSTART and SYSCTL.SECCFG.FRXPROTMAINEND registers with the start and end addresses of the range to be protected respectively. Both addresses are set up for writing at 64B granularity. To enable this protection, the FLRXPROT bit must be ENABLED (1) by writing to SYSCTL.SECCFG.FWENABLE register along with the correct KEY value (0x76).

At any time, status of the RX protection can be obtained by reading the SYSCTL.SECCFG.SECSTATUS register. The FLRXPROT field provide this status.

This mechanism is useful in scenarios where it may be required to prevent re-execution of the CSC. The protected address range of the CSC essentially runs only once at BOOTRST and can not subsequently be re-executed. Note that caution needs to be exercised in configuring the flash memory range that is configured for this protection. Address space that holds the interrupt vectors or the portion of CSC that needs to be able to run a second time after BOOTRST should not be execute-protected.

**Note**

In bank-swappable configuration, the protection is automatically mirrored to both banks.

### *4.4.4 IP Protection*

A region of flash memory can be configured for read protection - read accesses to this region will return an error while instruction fetch accesses are allowed. CPU, DMA and debugger accesses are all treated the same way. This is configured by writing to the SYSCTL.SECCFG.FIPPROTMAINSTART and SYSCTL.SECCFG.FIPPROTMAINEND registers with the start and end addresses of the range to be protected respectively. Both addresses are set up for writing at 64B granularity. To enable this protection, the FLIPPROT bit must be ENABLED (1) by writing to SYSCTL.SECCFG.FWENABLE register along with the correct KEY value (0x76).

At any time, status of the IP protection can be obtained by reading the SYSCTL.SECCFG.SECSTATUS register. The FLIPPROT field provide this status.

This mechanism is useful in scenarios where it may be required to prevent code read-out of third-party vendor-supplied software IP. Note that code that is meant to be IP-protected must be compiled such that there are no embedded data accesses (literal fetches) to this region. Since data accesses return access errors, such code must be compiled with a flag such as -mexecute-only (TI Clang Compiler flag) to ensure no literal constant accesses are emitted.

**Note**

In bank-swappable configuration, the protection is automatically mirrored to both banks.



Copyright © 2026 Texas Instruments Incorporated

#### 4.4.5 Data Bank Protection

A region of flash DATA bank can be configured for read-write protection- either reads or writes or both types of accesses can be blocked. CPU, DMA and debugger accesses are all treated the same way. This is configured by writing to the SYSCTL.SECCFG.FRWPROTDATA register. Only the first 4KB of the DATA bank can be protected at a sector (1KB) granularity. Each sector can be:

- Read protected
- Write protected
- Both
- Neither

This mechanism allows CSC to hold secret or sensitive data in the DATA bank that it alone can read/modify during start-up.

**Note**

Not all devices support a DATA bank. Refer to the device-specific data sheet to determine if the device supports a DATA bank.

#### 4.4.6 Hardware Monotonic Counter

Some devices provision a second NONMAIN sector (1KB) that is accessed with the following restrictions:

- It can never be erased (factory reset, mass erase operations included)
- It can be written only until CSC calls INITDONE, and subsequently it can only be read.

This protection serves to implement hardware-based monotone counter feature. Because the sector can only be programmed, it effectively works as a nonvolatile up-counter (or down-counter depending on how the counter value is interpreted by software). The CSC can maintain revision or roll-back protection information in this sector, assuring that a version lower than the counter value cannot be activated.

**Note**

Refer to the device-specific data sheet to determine if the device supports a hardware monotonic counter.

### 4.5 SRAM Protection

Buffer overflows are a common source of exploits wherein, for example, a corrupt return address can cause execution to jump to malicious code. In order to mitigate such exploits, an SRAM code protection feature is available, wherein the SRAM can be partitioned into two regions:

- Region 1: Read-Write (RW)
- Region 2: Read-Execute (RX)

This is set up by configuring the SYSCTL.SOCLOCK.SRAMBOUNDARY register with an address A such that:

- **Addresses >= A** will be permitted for **read-execute** and not for writes
- **Addresses < A** will be permitted for **read-write** and not for execution (instruction fetch)

**Note**

Configuring A equal to the total SRAM capacity implies that the entire SRAM is for RW only and no code execution from SRAM is enabled.

**Note**

A = 0 is treated as entire SRAM being RWX. This is the reset state of the SRAM.



Copyright © 2026 Texas Instruments Incorporated

In security-enabled devices, the SRAMBOUNDARY configuration can be locked by the CSC by configuring the SYSCTL.SECCFG.FWENABLE.SRAMBOUNDARYLOCK bit. Setting the LOCK ensures that the untrusted application can not change the SRAM protection. In devices without the security add-on, the SRAMBOUNDARY functionality is still available but the LOCK functionality is not present. This enables applications to use this feature to catch memory access problems.



Copyright © 2026 Texas Instruments Incorporated

## 4.6 SECURITY Registers

Table 4-1 lists the memory-mapped registers for the SECURITY registers. All register offset addresses not listed in Table 4-1 should be considered as reserved locations and the register contents should not be modified.

**Table 4-1. SECURITY Registers**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 3000h | FWEPROTMAIN | 1 Sector Write-Erase per bit starting at address 0x0 of flash | | Go |
| 3018h | FRXPROTMAINSTART | Flash RX Protection Start Address | | Go |
| 301Ch | FRXPROTMAINEND | Flash RX Protection End Address | | Go |
| 3020h | FIPPROTMAINSTART | Flash IP Protection Start Address | | Go |
| 3024h | FIPPROTMAINEND | Flash IP Protection End Address | | Go |
| 3038h | FLBANKSWPPOLICY | Flash Bank Swap Policy | | Go |
| 303Ch | FLBANKSWP | Flash MAIN bank address swap | | Go |
| 3044h | FWENABLE | Security Firewall Enable Register | | Go |
| 3048h | SECSTATUS | Security Configuration status | | Go |
| 3060h | INITDONE | INITCODE PASS | | Go |

Complex bit access types are encoded to fit into small table cells. Table 4-2 shows the codes that are used for access types in this section.

**Table 4-2. SECURITY Access Type Codes**

| Access Type | Code | Description |
|---|---|---|
| Read Type | | |
| R | R | Read |
| Write Type | | |
| W | W | Write |
| Reset or Default Value | | |
| *-n* | | Value after reset or the default value |



Copyright © 2026 Texas Instruments Incorporated

### 4.6.1 FWEPROTMAIN (Offset = 3000h) [Reset = 00000000h]

FWEPROTMAIN is shown in Figure 4-2 and described in Table 4-3.

Return to the Summary Table.

1 Sector Write-Erase per bit starting at address 0x0 of flash

**Figure 4-2. FWEPROTMAIN**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 4-3. FWEPROTMAIN Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | R/W | 0h | 1 Sector Write Erase protection 1: prohibits write-erase, 0: allows |



Copyright © 2026 Texas Instruments Incorporated

### 4.6.2 FRXPROTMAINSTART (Offset = 3018h) [Reset = 000000XXh]

FRXPROTMAINSTART is shown in Figure 4-3 and described in Table 4-4.

Return to the Summary Table.

Flash RX Protection Start Address

**Figure 4-3. FRXPROTMAINSTART**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | ADDR | | | | | | | | | | | | | | | | RESERVED | | | | | |
| R-0h | | | | | | | | | | R/W-0h | | | | | | | | | | | | | | | | R-0h | | | | | |

**Table 4-4. FRXPROTMAINSTART Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-22 | RESERVED | R | 0h | |
| 21-6 | ADDR | R/W | 0h | Flash RX Protection Start Address 64B granularity |
| 5-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 4.6.3 FRXPROTMAINEND (Offset = 301Ch) [Reset = 000000XXh]

FRXPROTMAINEND is shown in Figure 4-4 and described in Table 4-5.

Return to the Summary Table.

Flash RX Protection End Address

**Figure 4-4. FRXPROTMAINEND**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | ADDR | | | | | | | | | | | | | | | | RESERVED | | | | | |
| R-0h | | | | | | | | | | R/W-0h | | | | | | | | | | | | | | | | R-0h | | | | | |

**Table 4-5. FRXPROTMAINEND Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-22 | RESERVED | R | 0h | |
| 21-6 | ADDR | R/W | 0h | Flash RX Protection End Address 64B granularity |
| 5-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 4.6.4 FIPPROTMAINSTART (Offset = 3020h) [Reset = 000000XXh]

FIPPROTMAINSTART is shown in Figure 4-5 and described in Table 4-6.

Return to the Summary Table.

Flash IP Protection Start Address

**Figure 4-5. FIPPROTMAINSTART**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | ADDR | | | | | | | | | | | | | | | | RESERVED | | | | | |
| R-0h | | | | | | | | | | R/W-0h | | | | | | | | | | | | | | | | R-0h | | | | | |

**Table 4-6. FIPPROTMAINSTART Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-22 | RESERVED | R | 0h | |
| 21-6 | ADDR | R/W | 0h | Flash IP Protection Start Address 64B granularity |
| 5-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 4.6.5 FIPPROTMAINEND (Offset = 3024h) [Reset = 000000XXh]

FIPPROTMAINEND is shown in Figure 4-6 and described in Table 4-7.

Return to the Summary Table.

Flash IP Protection End Address

**Figure 4-6. FIPPROTMAINEND**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | ADDR | | | | | | | | | | | | | | | | RESERVED | | | | | |
| R-0h | | | | | | | | | | R/W-0h | | | | | | | | | | | | | | | | R-0h | | | | | |

**Table 4-7. FIPPROTMAINEND Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-22 | RESERVED | R | 0h | |
| 21-6 | ADDR | R/W | 0h | Flash IP Protection End Address 64B granularity |
| 5-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 4.6.6 FLBANKSWPPOLICY (Offset = 3038h) [Reset = 00XXXXXXh]

FLBANKSWPPOLICY is shown in Figure 4-7 and described in Table 4-8.

Return to the Summary Table.

Flash Bank Swap Policy

**Figure 4-7. FLBANKSWPPOLICY**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | DISABLE |
| R-0h | | | | | | | W-0h |

**Table 4-8. FLBANKSWPPOLICY Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | Must have KEY==0xCA(202) for write<br>CAh = Write Key |
| 23-1 | RESERVED | R | 0h | |
| 0 | DISABLE | W | 0h | 1: Disables Policy To Allow Flash Bank Swapping<br>1h = Disallow Bank Swap |



Copyright © 2026 Texas Instruments Incorporated

### 4.6.7 FLBANKSWP (Offset = 303Ch) [Reset = 00XXXXXXh]

FLBANKSWP is shown in Figure 4-8 and described in Table 4-9.

Return to the Summary Table.

Flash MAIN bank address swap

**Figure 4-8. FLBANKSWP**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | USEUPPER |
| R-0h | | | | | | | W-0h |

**Table 4-9. FLBANKSWP Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | The key value of 58h (88) must be written with USEUPPER to change the bank swap configuration.<br>58h = Issue write |
| 23-1 | RESERVED | R | 0h | |
| 0 | USEUPPER | W | 0h | 1: Use Upper Bank as Logical 0<br>0h = Normal (default) memory map addressing scheme<br>1h = Flash upper region address space swapped with lower region |



Copyright © 2026 Texas Instruments Incorporated

### 4.6.8 FWENABLE (Offset = 3044h) [Reset = 00XXXXXXh]

FWENABLE is shown in Figure 4-9 and described in Table 4-10.

Return to the Summary Table.

Security Firewall Enable Register

**Figure 4-9. FWENABLE**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | SRAMBOUNDARYLOCK |
| R-0h | | | | | | | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | FLIPPROT | RESERVED | FLRXPROT | RESERVED | | | |
| R-0h | W-0h | R-0h | W-0h | R-0h | | | |

**Table 4-10. FWENABLE Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | Must have KEY==0x76(118) for write<br>76h = Write Key |
| 23-9 | RESERVED | R | 0h | |
| 8 | SRAMBOUNDARYLOCK | W | 0h | 1: Blocks Writes from Changing SRAMBOUNDARY MMR<br>1h = SRAMBOUNDARY MMR Locked |
| 7 | RESERVED | R | 0h | |
| 6 | FLIPPROT | W | 0h | 1: Flash Read IP ProtectionActive<br>1h = Turn On Flash IP Protection |
| 5 | RESERVED | R | 0h | |
| 4 | FLRXPROT | W | 0h | 1: Flash Read Execute Protection Active<br>1h = Turn On Flash Read-eXecute Protection |
| 3-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 4.6.9 SECSTATUS (Offset = 3048h) [Reset = 00000XXXh]

SECSTATUS is shown in Figure 4-10 and described in Table 4-11.

Return to the Summary Table.

Security Configuration status

**Figure 4-10. SECSTATUS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | FLBANKSWP | RESERVED | FLBANKSWPPOLICY | RESERVED | SRAMBOUNDARYLOCK |
| R-0h | | | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | FLIPPROT | RESERVED | FLRXPROT | RESERVED | CSCEXISTS | RESERVED | INITDONE |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 4-11. SECSTATUS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-13 | RESERVED | R | 0h | |
| 12 | FLBANKSWP | R | 0h | 1: Upper and Lower Banks have been swapped |
| 11 | RESERVED | R | 0h | |
| 10 | FLBANKSWPPOLICY | R | 0h | 1: Upper and Lower Banks allowed to be swapped<br>0h = 0<br>1h = 1 |
| 9 | RESERVED | R | 0h | |
| 8 | SRAMBOUNDARYLOCK | R | 0h | 1: SRAM Boundary MMR Locked<br>0h = 0<br>1h = 1 |
| 7 | RESERVED | R | 0h | |
| 6 | FLIPPROT | R | 0h | 1: Flash IP Protection Active<br>0h = 0<br>1h = 1 |
| 5 | RESERVED | R | 0h | |
| 4 | FLRXPROT | R | 0h | 1: Flash Read Execute Protection Active<br>0h = 0<br>1h = 1 |
| 3 | RESERVED | R | 0h | |
| 2 | CSCEXISTS | R | 0h | 1: CSC Exists in the system<br>0h = System does not have a CSC<br>1h = System does have a CSC |
| 1 | RESERVED | R | 0h | |
| 0 | INITDONE | R | 0h | 1: CSC has been completed<br>0h = INIT is not yet done<br>1h = INIT is done |



Copyright © 2026 Texas Instruments Incorporated

### 4.6.10 INITDONE (Offset = 3060h) [Reset = 00XXXXXXh]

INITDONE is shown in Figure 4-11 and described in Table 4-12.

Return to the Summary Table.

INITCODE PASS

**Figure 4-11. INITDONE**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | PASS |
| R-0h | | | | | | | W-0h |

**Table 4-12. INITDONE Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | Must have KEY==0x9D(157) for write<br>9Dh = Issue Reset |
| 23-1 | RESERVED | R | 0h | |
| 0 | PASS | W | 0h | INITCODE writes 1 for PASS, left unwritten a timeout will occur if not blocked<br>1h = INITCODE PASS |



Copyright © 2026 Texas Instruments Incorporated

Chapter 5

# Direct Memory Access (DMA)

The direct memory access (DMA) controller module transfers data from one address to another, without CPU intervention. This chapter describes the operation of the DMA controller.


**5.1 DMA Overview** ..... 524
**5.2 DMA Operation** ..... 525
**5.3 DMA Registers** ..... 536





Copyright © 2026 Texas Instruments Incorporated

## 5.1 DMA Overview

The DMA controller transfers data from a source address to a destination address without CPU intervention. For example, the DMA controller can be used to move data from ADC conversion memory to SRAM.

Devices can have up to sixteen DMA channels available. Therefore, depending on the number of DMA channels available, some features described in this chapter are not applicable to all devices. Please refer to the device-specific data sheet for the actual channel count of the DMA.

Using the DMA controller can increase the throughput of peripheral modules. It can also reduce system power consumption by allowing the CPU to remain in a low-power mode, without having to awaken to move data to or from a peripheral.

DMA controller features include:

- Up to sixteen independent transfer channels
- Configurable DMA channel priorities
- Byte (8-bit), half-word (16-bit), word (32-bit), long-word (64-bit), and long-long-word (128-bit) or mixed byte and word transfer capability
- Transfer counter block size supports up to 64k transfers of any data type
- Configurable DMA transfer trigger selection
- Seven flexible addressing modes
- Single or block transfer modes

The DMA controller block diagram is shown in Figure 5-1.



Copyright © 2026 Texas Instruments Incorporated

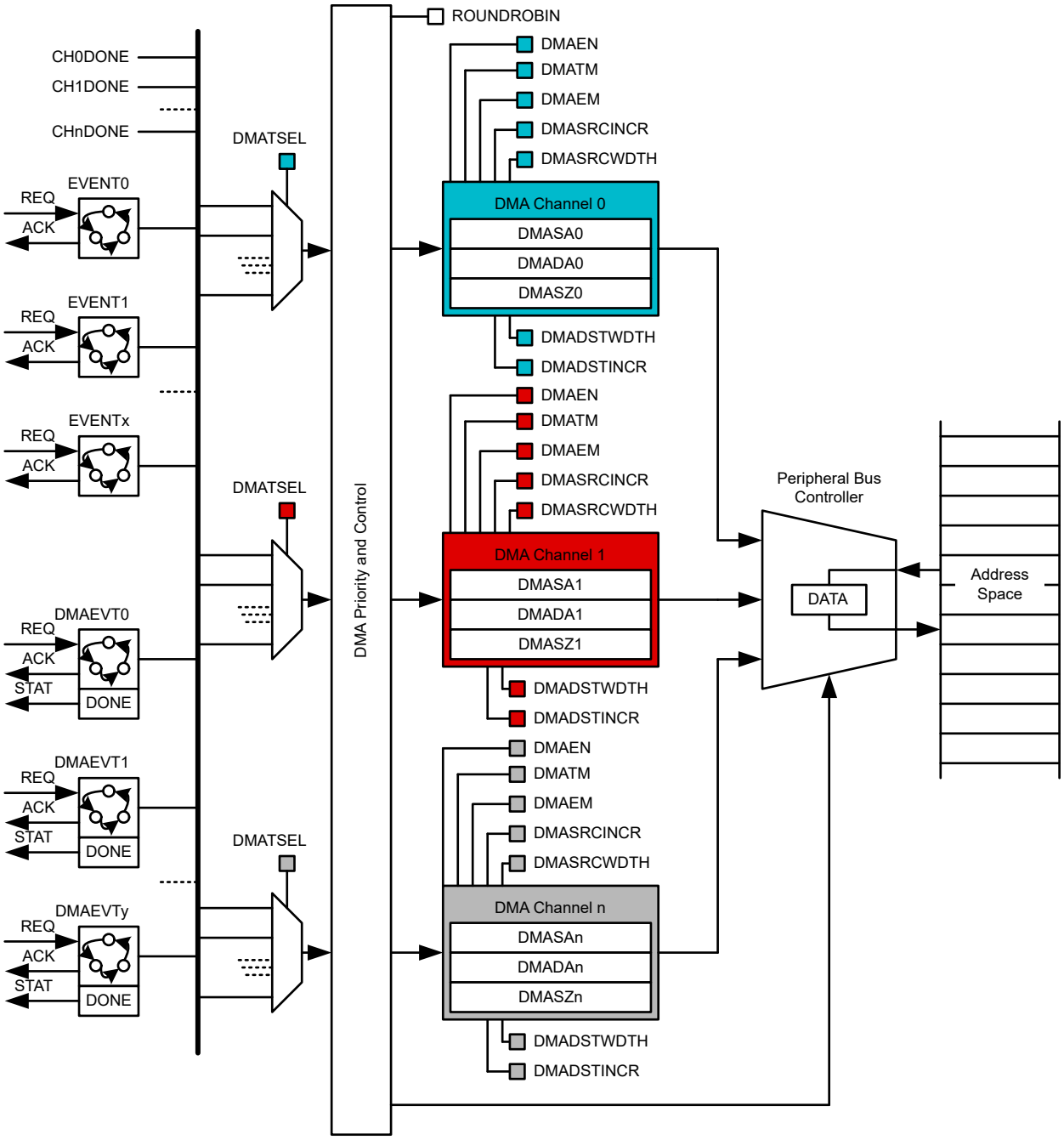



**Figure 5-1. DMA Block Diagram**

## 5.2 DMA Operation

The DMA controller is configured with user software. The setup and operation of the DMA is discussed in the following sections.

### *5.2.1 Addressing Modes*

The DMA controller has seven addressing modes. The addressing mode for each DMA channel is independently configurable. For example, channel 0 can transfer between two fixed addresses, while channel 1 transfers between two blocks of addresses. The basic addressing modes are shown in basic address modes

The addressing modes are:



Copyright © 2026 Texas Instruments Incorporated

1. Fixed address to fixed address
2. Fixed address to block of addresses
3. Block of addresses to fixed address
4. Block of addresses to block of addresses
5. Fill data to block of addresses
6. Data table to specific address
7. Gather data from address table to fixed address or block of addresses

Addressing modes 1-4 shown above are the basic adressing modes are simply configured with the DMASRCINCR and DMADSTINCR control bits and are available in all channel types. The DMASRCINCR bits select if the source address is incremented, decremented, or unchanged after each transfer. The DMADSTINCR bits select if the destination address is incremented, decremented, or unchanged after each transfer.

Addressing modes 5 -7 shown above are also configured with the DMASRCINCR and DMADSTINCR control bits along with the help of additional parameters such as DMAEM for leveraging the extended modes of the DMA. Refer to Section 5.2.4.2, Section 5.2.4.3 or [Gather Mode] for more details on how to properly configure and use the DMA in Fill-Mode, Table-Mode and Gather-Mode.

Transfers can be byte to byte, half-word to half-word, word to word, long-word to long-word, long-long-word to long-long-word or any combination of the five. When transferring a wider bit width source to a shorter bit width destination, only the lower bits of the destination data transfers. When transferring a shorter bith width source to a wider bit width destination, the upper bytes of the destination data is cleared when the transfer occurs. There is no packing or unpacking support by combining several source byte transfers to one single destination word or the reverse.



Copyright © 2026 Texas Instruments Incorporated

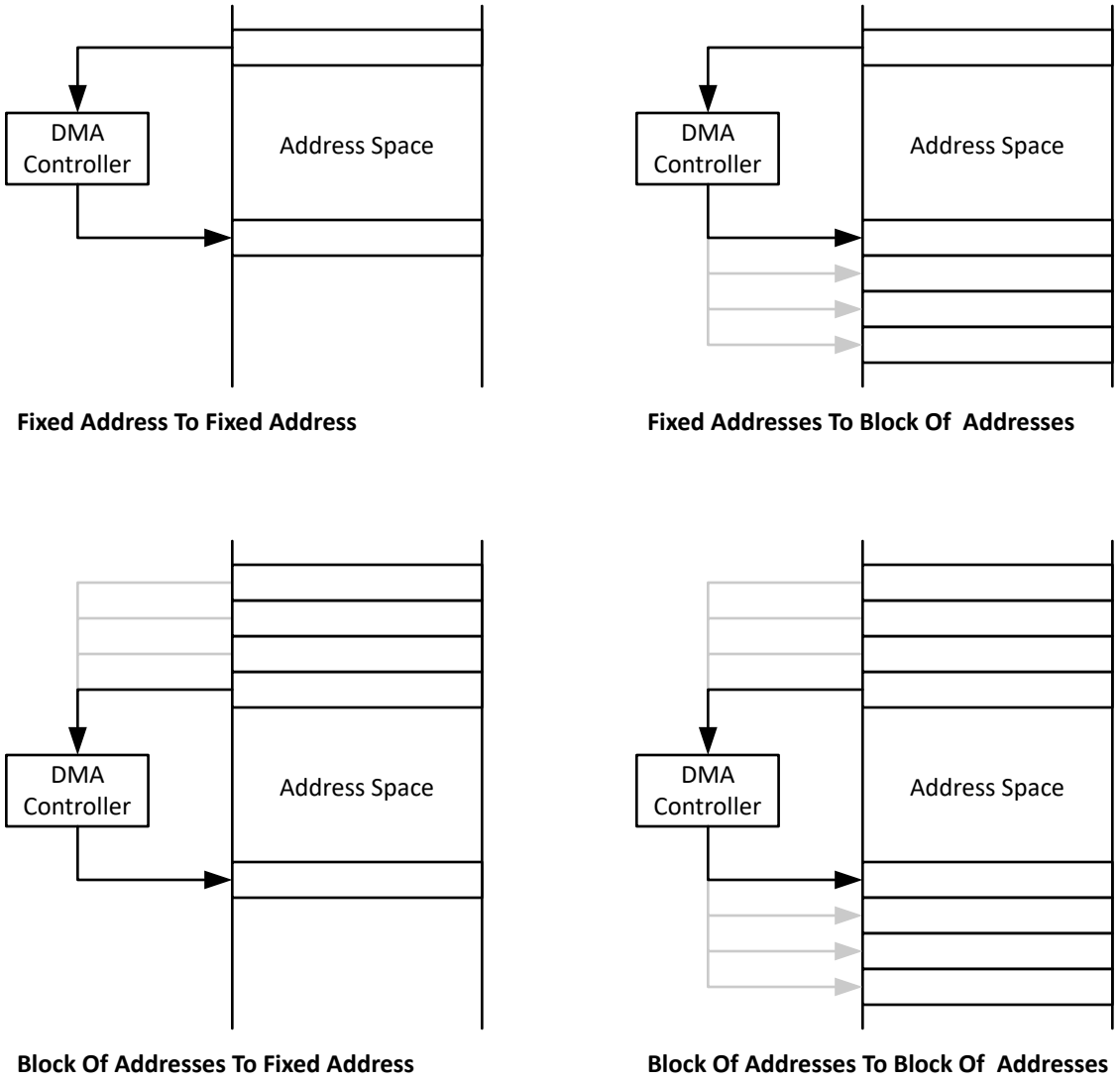



**Figure 5-2. Basic DMA Addressing Modes**

### *5.2.2 Channel Types*

There are two types of DMA: DMA_A and DMA_B. Each DMA supports basic (BASIC) and full-featured (FULL) channels. BASIC channels support only single or block transfers, and FULL channels support repeated single and repeated block transfers with additional features such as early interrupt request generation and extended table, fill and gather modes.

The highest priority DMA channels (starting with DMA0) are FULL channels, and the remaining priority channels are BASIC channels.

**Note**

See the device-specific data sheet to determine the type of DMA and the corresponding FULL and BASIC channels supported

Feature comparison table shows the features supported in the available basic and full-feature DMA channel types.



Copyright © 2026 Texas Instruments Incorporated

**Table 5-1. Feature Comparison of DMA_A and DMA_B Channels**

| DMA Feature | DMA_A | | DMA_B | |
|---|---|---|---|---|
| | Full-Feature | Basic Channel | Full-Feature | Basic Channel |
| Repeated mode | – | – | ✓ | – |
| Table & fill mode | ✓ | – | ✓ | – |
| Gather mode | – | – | ✓ | – |
| Early IRQ notification | ✓ | – | ✓ | – |
| Auto enable | – | – | ✓ | ✓ |
| Long long (128-bit) transfer | – | – | ✓ | ✓ |
| Stride mode | ✓ | ✓ | ✓ | ✓ |
| Cascading channel support | – | – | ✓ | ✓ |

### *5.2.3 Transfer Modes*

The DMA controller has four transfer modes selected by the DMATM bits as listed in DMA tranfer mode table. Each channel is individually configurable for its transfer mode. For example, channel 0 can be configured in repeated block transfer mode, while channel 1 is configured for block transfer mode, and channel 2 operates in single transfer mode. The transfer mode (single, block, repeat) is configured independently from the addressing mode (decrement, fixed, increment, stride). Any addressing mode can be used with any transfer mode.

Five types of data can be transferred selectable by the DMADSTWDTH and DMASRCWDTH control bits. The source and destination locations can be either byte (8-bit), half-word (16-bit), word (32-bit), long-word (64-bit), or long-long-word (128-bit) data. It is also possible to transfer byte to byte, half-word to half-word, word to word, long-word to long-word, long-long-word to long-long-word, or any combination.

Additionally, all transfers modes support a stride mode where the DMA source and destination can be incremented to a higher value to support re-organization of data.

**Table 5-2. DMA Transfer Modes**

| DMATM | Transfer Mode | Description | Channel Type |
|---|---|---|---|
| 0 | Single transfer | Each transfer requires a trigger. DMAEN is automatically cleared when DMASZx transfers have been made. | Basic |
| 1 | Block transfer | A complete block is transferred with one trigger. DMAEN is automatically cleared at the end of the block transfer. | Basic |
| 2 | Repeated single transfer | Each transfer requires a trigger. The DAMSA, DMADA and DMASZ registers are reloaded to the original value when the DMASZ counted down to zero. DMAEN remains enabled. | Full-feature |
| 3 | Repeated block transfer | A complete block is transferred with one trigger and continous transfering. The DAMSA, DMADA and DMASZ registers are reloaded to the original value when the DMASZ counted down to zero. DMAEN remains enabled. | Full-feature |

#### 5.2.3.1 Single Transfer

In single transfer mode (DMATM = 0), each byte, half-word, word, long-word, or long-long-word transfer requires a separate trigger. Single transfer mode is available in basic and full-feature DMA channels.

The DMASZ register defines the number of transfers to be made. The DMADSTINCR and DMASRCINCR bits select if the destination address and the source address are incremented or decremented after each transfer. If DMASZ = 0, no transfers occur.

The DMASA, DMADA, and DMASZ registers are incremented or decremented after each transfer. The DMADSTWDTH will indicate whether the destination address will increment or decrement by 1, 2, 4, 8, or 16 with each transfer cycle. The same is true for the DMASRCWDTH and the source address respectively. When the DMASZ register decrements to zero, the corresponding RIS flag is set.



Copyright © 2026 Texas Instruments Incorporated

The DMAEN bit is cleared automatically when DMASZ decrements to zero and must be set again for another transfer to occur.

#### 5.2.3.2 Block Transfer

In block transfer mode (DMATM = 1), a transfer of a complete block of data occurs after one trigger.

The DMASZ register defines the size of the block, and the DMADSTINCR and DMASRCINCR bits select if the destination address and the source address are incremented or decremented after each transfer of the block. If DMASZ = 0, no transfers occur.

The DMADSTWDTH will indicate whether the destination address will increment or decrement by 1, 2, 4, 8 or 16 with each transfer cycle. The same is true for the DMASRCWDTH and the source address respectively. The DMASZ register is decremented after each transfer and shows the number of transfers remaining.

The DMAEN bit is cleared after the completion of the block transfer and must be set again before another block transfer can be triggered. After a block transfer has started, another trigger signal that occurs during the block transfer is ignored.

#### 5.2.3.3 Repeated Single Transfer

In repeated single transfer mode (DMATM = 2), the DMA controller remains enabled with DMAEN = 1, and a transfer occurs every time a trigger occurs. Repeated single transfer modes are available in full-featured DMA channels only.

The DMASA, DMADA, and DMASZ registers are copied into temporary registers. The values of DMASA and DMADA are incremented or decremented after each transfer. The DMASZ register is decremented after each register. The DMADSTWDTH will indicate whether the destination address will increment or decrement by 1, 2, 4, 8 or 16 with each transfer cycle. The same is true for the DMASRCWDTH and the source address respectively. When the DMASZ register decrements to zero, it is reloaded from its temporary register and the corresponding RIS flag is set.

**Note**

When using repeated single transfer mode, the DMA does not support pausing and continuing a transfer by disabling a channel (to pause) and then re-enabling the channel (to continue).

#### 5.2.3.4 Repeated Block Transfer

In repeated block transfer mode (DMATM = 3), the DMAEN bit remains set after completion of the block transfer. The next trigger after the completion of a repeated block transfer starts another block transfer. Repeated block transfer modes are available in full-featured DMA channels only.

The DMASA, DMADA, and DMASZ registers are copied into temporary registers. The temporary values of DMASA and DMADA are incremented or decremented after each transfer in the block. The DMADSTWDTH will indicate whether the destination address will increment or decrement by 1, 2, 4, 8 or 16 with each transfer cycle. The same is true for the DMASRCWDTH and the source address respectively. The DMASZ register is decremented after each transfer of the block and shows the number of transfers remaining in the block. When the DMASZ register decrements to zero, it is reloaded from its temporary register and the corresponding RIS flag is set.

#### 5.2.3.5 Stride Mode

All transfer modes support a “stride” mode where the DMA source and destination can be incremented to a higher value (rather than +1) after a transfer. This is helpful for re-organizing the order of data between the source and destination.

To support incremental strides, set the DMADSTINCR and/or DMASRCINCR to STRIDE_n, where n is the number of destination and/or source increments. The real increments are based in terms of the definitions DMADSTWDTH and/or DMASRCWDTH, respectively. For example, if external ADC data is transmitted to the MCU as a six-word SPI frame, DMADSTINCR can be set to STRIDE_6 during a block transfer so that the destination address is incremented by 6 and the data is organized to make processing easier.



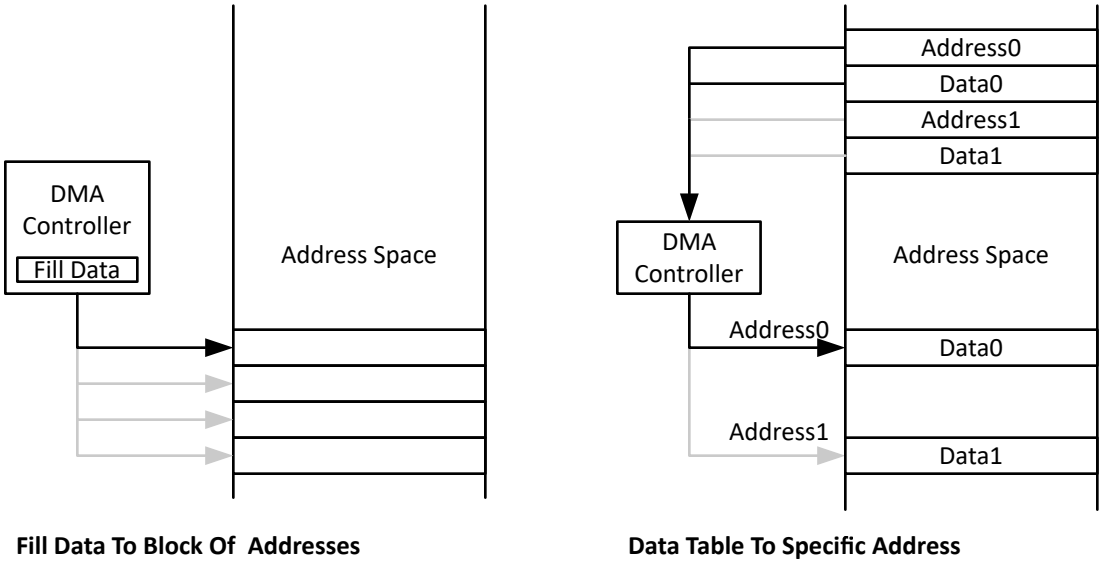
Copyright © 2026 Texas Instruments Incorporated

### 5.2.4 Extended Modes

In FULL channels, the DMA controller has two extended modes selected by the DMAEM bits as listed in DMA extended mode table. An graphical overview of the extended modes can be seen inDMA extended mode figuer. Each channel is individually configurable for the extended mode. For example, channel 0 can be configured in table mode while channel 1 is configured in fill mode.

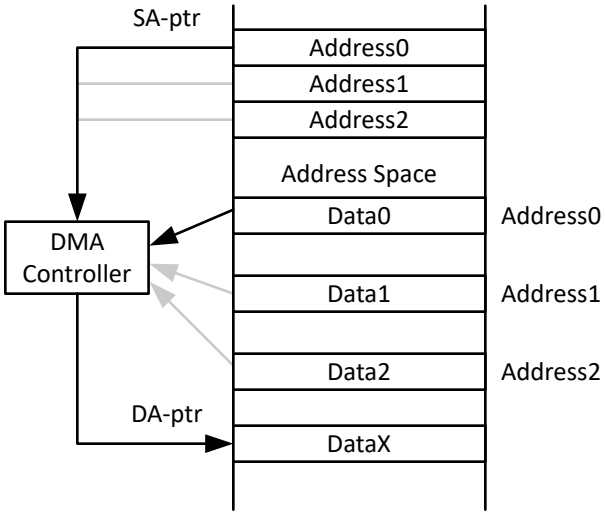
**Table 5-3. DMA Extended Modes**

| DMAEM | Extended Mode | Description |
|---|---|---|
| 0 | Normal mode | Operation is defined by DMATM |
| 1 | Gather mode | Used to read data from an address table and copy to configurable address |
| 2 | Fill mode | Used to fill predefined data patterns into memory |
| 3 | Table mode | Used to help configure a table of peripheral control registers |




**Fill Data To Block Of Addresses**

**Data Table To Specific Address**




**Gather Table To Specific Address**

**Figure 5-3. DMA Extended Modes Diagram**



Copyright © 2026 Texas Instruments Incorporated

#### 5.2.4.1 Gather Mode

In table mode (DMAEM = 1), the DMA controller points to a address table in memory to gather the data at these addresses. The DMA fills the data at these addresses at the destination address. This enables the programmer to move data from non-contiguous source locations into a contiguous destination buffer.

#### 5.2.4.2 Fill Mode

In fill mode (DMAEM = 2), the DMA controller takes a predefined FILL pattern and writes the pattern to a user defined segment of memory. The DMATM bits are ignored and the automatic transfer mode used is "block transfer".

The DMASAx register is used as the FILL pattern data. The DMASRCINCR bit field is used to indicate whether the FILL pattern data should be constant or incremented/decremented with every write cycle. This feature allows for filling a memory block with a sequential pattern (for example. 0, 1, 2, 3, …). The DMASRCWDTH bit field indicates the magnitude of increment of the FILL mode data. Refer to Table 5-4 for how to use DMASRCWDTH in fill mode.

**Table 5-4. DMASRCWDTH in Fill Mode**

| DMASRCWDTH | FILL Mode Data Increment Value |
|---|---|
| 0 | ±1 |
| 1 | ±2 |
| 2 | ±4 |
| 3 | ±8 |

The destination registers and bit fields DMADAx, DMADSTINCR, and DMADSTWDTH all behave as expected and influence where and how in memory the FILL pattern is written.

#### 5.2.4.3 Table Mode

In table mode (DMAEM = 11b), the DMA controller executes 2 reads from the source and one write to a determined destination. This feature can be leveraged to interpret a table of addresses and data and uses the DMA to efficiently program that data to their associated addresses without CPU intervention. Table mode allows you to parse through a table of addresses and data to configure peripheral memory mapped registers in a single block transfer.

The DMASRCWDTH bit field should be set to "3" (64-bit mode) and the DMADSTWDTH bit field should be set to "2" (32-bit mode). The DMASRCINCR bit field can be set to 10b to decrement the source address or 11b to increment the source address after transfers. DMASZ is set to represent the number of entries in the table and DMATM should be set to "01" for block transfer mode. DMADAx and DMADSTINCR are ignored in table mode and can be treated as "don’t care" values.

The DMASAx register needs to be programmed with the start address of the table, which needs to be aligned to 64-bit data (that is, DMASAx[2:0] = "000"). The address stored in the table needs to be on the lower word of a 64-bit data (ADDR[2:0] = "000" while the data needs to be on the upper word of a 64-bit data (ADDR[2:0] = "100"). Table 5-5 is an example of a table in memory compatible with the DMA table mode.

For applications requiring 128 bit transfers, a variable or array needs to be defined within the software program to enforce a 16-byte alignment for the transfers as shown below.

```
__attribute__((aligned(16))) attribute
```

For 32 and 64 bit transfers, a variable or array needs to be defined with uint32_t and uint64_t respectively.

**Table 5-5. Example of an Incremental Table Compatible with DMA Table Mode**

| Table Address | Table Data |
|---|---|
| 0x0000 | Address 0 |
| 0x0004 | Data 0 |
| 0x0008 | Address 1 |



Copyright © 2026 Texas Instruments Incorporated

**Table 5-5. Example of an Incremental Table Compatible with DMA Table Mode (continued)**

| Table Address | Table Data |
|---|---|
| 0x000C | Data 1 |

### 5.2.5 Initiating DMA Transfers

Each DMA channel is independently configured for its trigger source with DMATSEL. The DMATSEL bits should be modified only when the DMACTLx.DMAEN bit is 0; otherwise, unpredictable DMA triggers can occur.

See the device-specific data sheet for the list of triggers available, along with their respective DMATSEL values.

When selecting the trigger, the trigger must not have already occurred, or the transfer does not take place.

DMA channels can be internally triggered upon the completion of activity on another channel to support cascading. Completion of activity occurs when a DMA channel's DMASZ counter reaches zero. This is beneficial for applications where data can be retrieved, transferred, and/or error-checked without an interrupt or event configuration.

Set the DMATINT bit to internally trigger the next DMA channel based on the DMATSEL trigger source. Once the DMATSEL trigger occurs, the next DMA channel begins to automatically execute.

For example, if UART data is received and transmitted to SRAM through DMA channel 0 and DMATSEL is set to UART RX, then DMA channel 1 can be internally triggered when the UART is finished receiving data. If DMA channel 1 is configured to transmit the data from SRAM to CRC, then the DMA transfer will trigger once the UART data is received. In this case, the DMA channels are cascaded from Channel 0 to Channel 1.

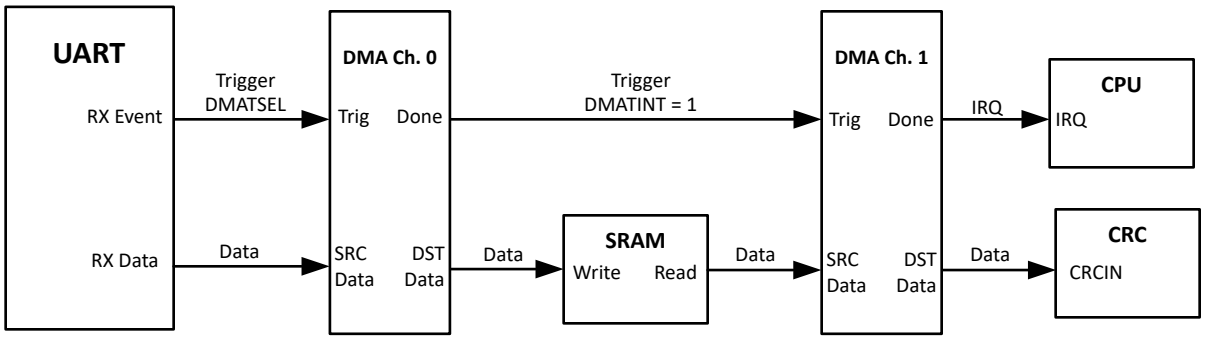



**Figure 5-4. DMA Cascading Channels**

### 5.2.6 Stopping DMA Transfers

A DMA block transfer in progress can be stopped by clearing the DMAEN bit. The DMA will stop after the completion of the ongoing transfer cycle and all the channel registers will stay in the current state. The block transfer can be continued as originally configured after setting the DMAEN bit again and resending the trigger. Please note that a trigger is necessary for halted transfer to resume.

**Note**

A single transfer in progress cannot be interrupted.

### 5.2.7 Channel Priorities

The default DMA channel priorities are DMA0 through DMA15. If two or three triggers happen simultaneously or are pending, the channel with the highest priority completes its transfer (single or block transfer) first, then the second priority channel, then the third priority channel. Transfers in progress are not halted if a higher-priority channel is triggered. The higher-priority channel waits until the transfer in progress completes before starting.

The DMA channel priorities are configurable with the ROUNDROBIN bit. When the ROUNDROBIN bit is set, the channel that completes a transfer becomes the lowest priority. The order of the priority of the channels always



Copyright © 2026 Texas Instruments Incorporated

stays the same, DMA0-DMA1-DMA2, for example, for three channels. When the ROUNDROBIN bit is cleared, the channel priority returns to the default priority.

**Table 5-6. Round-Robin DMA Priority Example**

| Current DMA Priority | Transfer Occurs | New DMA Priority |
|---|---|---|
| DMA0 - DMA1 - DMA2 | DMA1 | DMA2 - DMA0 - DMA1 |
| DMA2 - DMA0 - DMA1 | DMA2 | DMA0 - DMA1 - DMA2 |
| DMA2 - DMA0 - DMA1 | DMA0 | DMA1 - DMA2 - DMA0 |

### *5.2.8 Burst Block Mode*

The DMA module supports a burst block mode for suspending an active channel after a configurable number of transfers to service other pending channels. The burst block size is configurable by setting DMAPRIO.BURSTSZ to 8, 16, 32, or an infinite number of transfers. If a higher priority channel is pending after the burst block, the DMA will execute the higher priority channel and resume on the suspended channel once the higher priority channel is complete. If no other channel is pending, the priority logic assigns the control back to the block transfer for the next burst.

### *5.2.9 Using DMA with System Interrupts*

System interrupt service routines are interrupted by DMA transfers. If an interrupt service routine or other routine must execute with no interruptions, the DMA controller should be disabled before executing the routine.

### *5.2.10 DMA Controller Interrupts*

Each DMA channel has its own RIS flag. Each RIS flag is set in any mode when the corresponding DMASZx register counts to zero. If the corresponding MASK and RIS bits are set, an interrupt request is generated.

All RIS flags are prioritized, with DMA0 being the highest, and combined to source a single interrupt vector. The highest-priority enabled interrupt generates a number in the IIDX register. This number can be evaluated or added to the program counter (PC) to automatically enter the appropriate software routine.

Any access, read or write, of the IIDX register automatically resets the highest pending interrupt flag. If another interrupt flag is set, another interrupt is immediately generated after servicing the initial interrupt. For example, assume that DMA0 has the highest priority. If the DMA0-RIS and DMA2-RIS flags are set when the interrupt service routine accesses the IIDX register, DMA0-RIS is reset automatically. After the interrupt service routine is executed, the DMA2-RIS generates another interrupt.

### *5.2.11 DMA Trigger Event Status*

The DMA controller supports dedicated DMA events. See Section 7.2.2 for details on the DMA event trigger protocol. The idea is that the DMA can inform the event triggering peripheral about the status of the assigned DMA channel. This will allow the triggering peripheral to issue an interrupt itself after the completion of a repeated transfer, instead of the DMA issuing an interrupt event. The advantage is, that the DMA interrupt service routine does not need to keep track of the assigned function of the channel. As a result, the DMA triggering peripheral interrupt service routine will deal with the completion of the DMA transfer.

**Note**

When a peripheral triggers a DMA transfer it is important that the peripheral remains enabled until the DMA transfer completes. Otherwise, the DMA request can hang, preventing the device from fully entering a low-power mode.

The status will reflect the value of the DMASZx register. If the last DMA transfer resulted in a size decrement to zero, the DMA will return the status of 1, indicating the end of the transfer. Otherwise the status will be 0.

Additionally, the DMA module can generate an early interrupt request to the CPU to indicate that a transfer will complete within a configurable number of transfers (1, 2, 4, 8, 32, 64, half-DMASZ).



Copyright © 2026 Texas Instruments Incorporated

An early IRQ event is enabled by setting DMAPREIRQ to the desired number of transfers. When the DMA has reached the number of transfers, the corresponding DMA channel's PREIRQ interrupt is set.

Early DMA interrupt generation is useful to:

- Reduce the interrupt latency in timing-critical applications where it would be beneficial to let the DMA preemptively generate the IRQ before the DMA transfer is complete
- Serve as a "progress notification" when scheduling other tasks for the CPU to complete
- Transfer weights of a neural network layer and notify the CPU to complete software configuration writes to the IP
- Implement a ping-pong buffer (by setting DMAPREIRQ to half)

**Note**

This feature is available on repeat-capable channels only.

### *5.2.12 DMA Operating Mode Support*

The DMA supports triggered transfers in RUN mode, as well as in the SLEEP, STOP, and STANDBY low-power modes. Refer to the following sections for more details.

#### 5.2.12.1 Transfer in RUN Mode

In RUN mode the system is fully operational. The CPU and all other peripherals and resources are available, therefore there is no restriction on the DMA functionality in RUN mode.

#### 5.2.12.2 Transfer in SLEEP Mode

In SLEEP mode only the CPU is halted. All other peripherals and resources are available as when in RUN mode, therefore there is no restriction on the DMA functionality in SLEEP mode. All peripherals that can trigger a DMA transfer in RUN mode will also be able to trigger a DMA transfer in SLEEP mode.

#### 5.2.12.3 Transfer in STOP Mode

In STOP mode the CPU is halted and the ULPCLK is limited to 4MHz operation. PD1 peripherals are disabled and in retention mode when applicable. Only PD0 peripherals are functional and therefore only PD0 peripherals are able to trigger a DMA transfer in STOP mode. The DMA itself is located in PD1 and is therefore in retention during STOP mode. The event manager will detect a DMA trigger event and request the PMU to enter a "suspended STOP" state. For more info on this state refer to *Suspended Low-Power Mode Operation*.While STOP mode is suspended, the DMA is fully functional and will work on the pending DMA trigger request. Once the DMA transfer is complete, the DMA will acknowledge the pending trigger event and the event subsystem removes the power mode request from the PMU. If the PMU has no other pending requests, the SoC will transition back into normal STOP mode.

#### 5.2.12.4 Transfers in STANDBY Mode

In STANDBY mode the CPU is halted and the ULPCLK is limited to 32kHz operation. PD1 peripherals are disabled and in retention mode when applicable. Only PD0 peripherals are functional and therefore only PD0 peripherals are able to trigger a DMA transfer in STANDBY mode. The DMA itself is located in PD1 and therefore in retention during STANDBY mode.The event manager will detect a DMA trigger event and request the PMU to enter a "suspended STANDBY" state. For more info on this state refer to *Suspended Low-Power Mode Operation*.While STANDBY mode is suspended, the DMA is fully functional and will work on the pending DMA trigger request. Once the DMA transfer is complete, the DMA will acknowledge the pending trigger event and the event subsystem removes the power mode request from the PMU. If the PMU has no other pending requests, the SoC will transition back into normal STANDBY mode.

### *5.2.13 DMA Address and Data Errors*

The DMA itself has the ability to flag address or data errors. Source or destination address errors can come from accessing a protected or nonexisting memory range. If an address error occurs, the interrupt index IIDX[j].STAT



Copyright © 2026 Texas Instruments Incorporated

flags a DMA address error (11h). Address error interrupts can be masked, set, and cleared using the ADDERR bit.

There are some portions of memory where a DMA address error or a CPU hard fault error will not be observed upon access. This includes:

- The address between the end of flash to the start of ROM
- 4KB aperture of each IP starting from the base address of that IP

**Note**

The DMA itself does not perform range checking. If the DMA transfer occurs over a protected memory range, the destination data will report zeros (0h) for each byte of the DMA transaction that overlaps the protected or nonexisting memory range.

Data errors can occur in SRAM or flash if it has an ECC or parity error. If a data error occurs, the interrupt index IIDX[j].STAT flags a DMA data error (12h). Data error interrupts can be masked, set, and cleared using the DATERR bit.

### *5.2.14 Interrupt and Event Support*

The DMA module contains one event publishers and two event subscribers.

- One event publisher (CPU_INT) manages DMA interrupt requests (IRQs) to the CPU subsystem through a static event route.
- The second and third event (GEN_EVENT) are used to setup the generic event publishers and subscribers through Generic route.

DMA events are summarized in Table 5-7.

**Table 5-7. DMA Events**

| Event | Type | Source | Destination | Route | Configuration | Functionality |
|---|---|---|---|---|---|---|
| CPU interrupt | Publisher | DMA | CPU Subsystem | Static route | CPU_INT registers | Fixed interrupt route from TIMx to CPU |
| Generic publisher event | Publisher | DMA | Other peripherals | Generic route | GEN_EVENT and FPUB_1 registers | Configurable interrupt route from DMA to other peripherals |
| Generic subscriber event | Subscriber | Other peripherals | DMA | Generic route | FSUB_0 | Configurable interrupt route from other peripherals to DMA |
| Generic subscriber event | Subscriber | Other peripherals | DMA | Generic route | FSUB_1 | Configurable interrupt route from other peripherals to DMA |



Copyright © 2026 Texas Instruments Incorporated

## 5.3 DMA Registers

Table 5-8 lists the memory-mapped registers for the DMA registers. All register offset addresses not listed in Table 5-8 should be considered as reserved locations and the register contents should not be modified.

**Table 5-8. DMA Registers**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 400h | FSUB_0 | Subscriber Port 0 | | Go |
| 404h | FSUB_1 | Subscriber Port 1 | | Go |
| 444h | FPUB_1 | Publisher Port 0 | | Go |
| 1018h | PDBGCTL | Peripheral Debug Control | | Go |
| 1020h | IIDX | Interrupt index | CPU_INT | Go |
| 1028h | IMASK | Interrupt mask | CPU_INT | Go |
| 1030h | RIS | Raw interrupt status | CPU_INT | Go |
| 1038h | MIS | Masked interrupt status | CPU_INT | Go |
| 1040h | ISET | Interrupt set | CPU_INT | Go |
| 1048h | ICLR | Interrupt clear | CPU_INT | Go |
| 1050h | IIDX | Interrupt index | GEN_EVENT | Go |
| 1058h | IMASK | Interrupt mask | GEN_EVENT | Go |
| 1060h | RIS | Raw interrupt status | GEN_EVENT | Go |
| 1068h | MIS | Masked interrupt status | GEN_EVENT | Go |
| 1070h | ISET | Interrupt set | GEN_EVENT | Go |
| 1078h | ICLR | Interrupt clear | GEN_EVENT | Go |
| 10E0h | EVT_MODE | Event Mode | | Go |
| 10FCh | DESC | Module Description | | Go |
| 1100h | DMAPRIO | DMA Channel Priority Control | | Go |
| 1110h + formula | DMATCTL[j] | DMA Trigger Select | | Go |
| 1200h + formula | DMACTL[j] | DMA Channel Control | | Go |
| 1204h + formula | DMASA[j] | DMA Channel Source Address | | Go |
| 1208h + formula | DMADA[j] | DMA Channel Destination Address | | Go |
| 120Ch + formula | DMASZ[j] | DMA Channel Size | | Go |

Complex bit access types are encoded to fit into small table cells. Table 5-9 shows the codes that are used for access types in this section.

**Table 5-9. DMA Access Type Codes**

| Access Type | Code | Description |
|---|---|---|
| Read Type | | |
| R | R | Read |
| Write Type | | |
| W | W | Write |
| Reset or Default Value | | |
| *-n* | | Value after reset or the default value |
| Register Array Variables | | |



Copyright © 2026 Texas Instruments Incorporated

**Table 5-9. DMA Access Type Codes (continued)**

| Access Type | Code | Description |
|---|---|---|
| i,j,k,l,m,n | | When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula. |
| y | | When this variable is used in a register name, an offset, or an address it refers to the value of a register array. |



Copyright © 2026 Texas Instruments Incorporated

### 5.3.1 FSUB_0 (Offset = 400h) [Reset = 00000000h]

FSUB_0 is shown in Figure 5-5 and described in Table 5-10.

Return to the Summary Table.

Subscriber port

**Figure 5-5. FSUB_0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | CHANID | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R/W-0h | | | | | | | |

**Table 5-10. FSUB_0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | CHANID | R/W | 0h | 0 = disconnected.<br>1-255 = connected to channelID = CHANID.<br>0h = A value of 0 specifies that the event is not connected<br>FFh = Consult your device datasheet as the actual allowed maximum may be less than 255. |



Copyright © 2026 Texas Instruments Incorporated

### 5.3.2 FSUB_1 (Offset = 404h) [Reset = 00000000h]

FSUB_1 is shown in Figure 5-6 and described in Table 5-11.

Return to the Summary Table.

Subscriber port

**Figure 5-6. FSUB_1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | CHANID | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R/W-0h | | | | | | | |

**Table 5-11. FSUB_1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | CHANID | R/W | 0h | 0 = disconnected.<br>1-255 = connected to channelID = CHANID.<br>0h = A value of 0 specifies that the event is not connected<br>FFh = Consult your device datasheet as the actual allowed maximum may be less than 255. |



Copyright © 2026 Texas Instruments Incorporated

### 5.3.3 FPUB_1 (Offset = 444h) [Reset = 00000000h]

FPUB_1 is shown in Figure 5-7 and described in Table 5-12.

Return to the Summary Table.

Publisher port

**Figure 5-7. FPUB_1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | CHANID | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R/W-0h | | | | | | | |

**Table 5-12. FPUB_1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | CHANID | R/W | 0h | 0 = disconnected.<br>1-255 = connected to channelID = CHANID.<br>0h = A value of 0 specifies that the event is not connected<br>FFh = Consult your device datasheet as the actual allowed maximum may be less than 255. |



Copyright © 2026 Texas Instruments Incorporated

### 5.3.4 PDBGCTL (Offset = 1018h) [Reset = 00000000h]

PDBGCTL is shown in Figure 5-8 and described in Table 5-13.

Return to the Summary Table.

This register can be used by the software developer to control the behavior of the peripheral relative to the 'Core Halted' input

**Figure 5-8. PDBGCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | SOFT | FREE |
| R-0h | | | | | | R/W-0h | R/W-0h |

**Table 5-13. PDBGCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | |
| 1 | SOFT | R/W | 1h | Soft halt boundary control. This function is only available, if FREE is set to 'STOP'<br>0h = The peripheral will halt immediately, even if the resultant state will result in corruption if the system is restarted<br>1h = The peripheral blocks the debug freeze until it has reached a boundary where it can resume without corruption |
| 0 | FREE | R/W | 1h | Free run control<br>0h = The peripheral freezes functionality while the Core Halted input is asserted and resumes when it is deasserted.<br>1h = The peripheral ignores the state of the Core Halted input |



Copyright © 2026 Texas Instruments Incorporated

### 5.3.5 IIDX (Offset = 1020h) [Reset = 00000000h]

IIDX is shown in Figure 5-9 and described in Table 5-14.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. Value 0x00 means no event pending. Interrupt 1 is the highest priority, IIDX next highest, 4, 8, . . . IIDX^31 is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred.
On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in RIS [RIS] and MIS [MIS] are cleared as well. After a read from the CPU (not from the debug interface), the register is updated with the next highest priority interrupt, if none are pending, then it should display 0x0.
Note: The number of DMACH is device dependent. Please consult the datasheet of the specific device to map which channel number is implemented.

**Figure 5-9. IIDX**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | STAT | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | |

**Table 5-14. IIDX Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | STAT | R | 0h | Interrupt index status<br>00h = No bit is set means there is no pending interrupt request<br>01h = DMA Channel 0 size counter reached zero (DMASZ=0).<br>02h = DMA Channel 1 size counter reached zero (DMASZ=0).<br>03h = DMA Channel 2 size counter reached zero (DMASZ=0).<br>04h = DMA Channel 3 size counter reached zero (DMASZ=0).<br>05h = DMA Channel 4 size counter reached zero (DMASZ=0).<br>06h = DMA Channel 5 size counter reached zero (DMASZ=0).<br>07h = DMA Channel 6 size counter reached zero (DMASZ=0).<br>08h = DMA Channel 7 size counter reached zero (DMASZ=0).<br>09h = DMA Channel 8 size counter reached zero (DMASZ=0).<br>0Ah = DMA Channel 9 size counter reached zero (DMASZ=0).<br>0Bh = DMA Channel 10 size counter reached zero (DMASZ=0).<br>0Ch = DMA Channel 11 size counter reached zero (DMASZ=0).<br>0Dh = DMA Channel 12 size counter reached zero (DMASZ=0).<br>0Eh = DMA Channel 13 size counter reached zero (DMASZ=0).<br>0Fh = DMA Channel 14 size counter reached zero (DMASZ=0).<br>10h = DMA Channel 15 size counter reached zero (DMASZ=0).<br>11h = PRE-IRQ event for DMA Channel 0.<br>12h = PRE-IRQ event for DMA Channel 1.<br>13h = PRE-IRQ event for DMA Channel 2.<br>14h = PRE-IRQ event for DMA Channel 3.<br>15h = PRE-IRQ event for DMA Channel 4.<br>16h = PRE-IRQ event for DMA Channel 5.<br>17h = PRE-IRQ event for DMA Channel 6.<br>18h = PRE-IRQ event for DMA Channel 7.<br>19h = DMA address error, SRC address not reachable.<br>1Ah = DMA data error, SRC data might be corrupted (PAR or ECC error). |



Copyright © 2026 Texas Instruments Incorporated

### 5.3.6 IMASK (Offset = 1028h) [Reset = 00000000h]

IMASK is shown in Figure 5-10 and described in Table 5-15.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then the corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX [IIDX], as well as MIS [MIS].
Note: The number of DMACH is device dependent. Please consult the datasheet of the specific device to map which channel number is implemented.

**Figure 5-10. IMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | DATAERR | ADDRERR |
| R-0h | | | | | | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| PREIRQCH7 | PREIRQCH6 | PREIRQCH5 | PREIRQCH4 | PREIRQCH3 | PREIRQCH2 | PREIRQCH1 | PREIRQCH0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DMACH15 | DMACH14 | DMACH13 | DMACH12 | DMACH11 | DMACH10 | DMACH9 | DMACH8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DMACH7 | DMACH6 | DMACH5 | DMACH4 | DMACH3 | DMACH2 | DMACH1 | DMACH0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

**Table 5-15. IMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | RESERVED | R | 0h | |
| 25 | DATAERR | R/W | 0h | DMA data error, SRC data might be corrupted (PAR or ECC error).<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 24 | ADDRERR | R/W | 0h | DMA address error, SRC address not reachable.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 23 | PREIRQCH7 | R/W | 0h | Pre-IRQ for Channel 7. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 22 | PREIRQCH6 | R/W | 0h | Pre-IRQ for Channel 6. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 21 | PREIRQCH5 | R/W | 0h | Pre-IRQ for Channel 5. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 20 | PREIRQCH4 | R/W | 0h | Pre-IRQ for Channel 4. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 19 | PREIRQCH3 | R/W | 0h | Pre-IRQ for Channel 3. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 18 | PREIRQCH2 | R/W | 0h | Pre-IRQ for Channel 2. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 17 | PREIRQCH1 | R/W | 0h | Pre-IRQ for Channel 1. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |



Copyright © 2026 Texas Instruments Incorporated

**Table 5-15. IMASK Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 16 | PREIRQCH0 | R/W | 0h | Pre-IRQ for Channel 0. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 15 | DMACH15 | R/W | 0h | DMA Channel 15 interrupt signal. Size counter reached zero (DMASZ=0).<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 14 | DMACH14 | R/W | 0h | DMA Channel 14 interrupt signal. Size counter reached zero (DMASZ=0).<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 13 | DMACH13 | R/W | 0h | DMA Channel 13 interrupt signal. Size counter reached zero (DMASZ=0).<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 12 | DMACH12 | R/W | 0h | DMA Channel 12 interrupt signal. Size counter reached zero (DMASZ=0).<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 11 | DMACH11 | R/W | 0h | DMA Channel 11 interrupt signal. Size counter reached zero (DMASZ=0).<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 10 | DMACH10 | R/W | 0h | DMA Channel 10 interrupt signal. Size counter reached zero (DMASZ=0).<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 9 | DMACH9 | R/W | 0h | DMA Channel 9 interrupt signal. Size counter reached zero (DMASZ=0).<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 8 | DMACH8 | R/W | 0h | DMA Channel 8 interrupt signal. Size counter reached zero (DMASZ=0).<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 7 | DMACH7 | R/W | 0h | DMA Channel 7 interrupt signal. Size counter reached zero (DMASZ=0).<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 6 | DMACH6 | R/W | 0h | DMA Channel 6 interrupt signal. Size counter reached zero (DMASZ=0).<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 5 | DMACH5 | R/W | 0h | DMA Channel 5 interrupt signal. Size counter reached zero (DMASZ=0).<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 4 | DMACH4 | R/W | 0h | DMA Channel 4 interrupt signal. Size counter reached zero (DMASZ=0).<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 3 | DMACH3 | R/W | 0h | DMA Channel 3 interrupt signal. Size counter reached zero (DMASZ=0).<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 2 | DMACH2 | R/W | 0h | DMA Channel 2 interrupt signal. Size counter reached zero (DMASZ=0).<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |



Copyright © 2026 Texas Instruments Incorporated

**Table 5-15. IMASK Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 1 | DMACH1 | R/W | 0h | DMA Channel 1 interrupt signal. Size counter reached zero (DMASZ=0).<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 0 | DMACH0 | R/W | 0h | DMA Channel 0 interrupt signal. Size counter reached zero (DMASZ=0).<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |



Copyright © 2026 Texas Instruments Incorporated

### 5.3.7 RIS (Offset = 1030h) [Reset = 00000000h]

RIS is shown in Figure 5-11 and described in Table 5-16.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR [ICLR] register bit even if the corresponding IMASK [IMASK] bit is not enabled.
Note: The number of DMACH is device dependent. Please consult the datasheet of the specific device to map which channel number is implemented.

**Figure 5-11. RIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | DATAERR | ADDRERR |
| R-0h | | | | | | R-0h | R-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| PREIRQCH7 | PREIRQCH6 | PREIRQCH5 | PREIRQCH4 | PREIRQCH3 | PREIRQCH2 | PREIRQCH1 | PREIRQCH0 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DMACH15 | DMACH14 | DMACH13 | DMACH12 | DMACH11 | DMACH10 | DMACH9 | DMACH8 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DMACH7 | DMACH6 | DMACH5 | DMACH4 | DMACH3 | DMACH2 | DMACH1 | DMACH0 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 5-16. RIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | RESERVED | R | 0h | |
| 25 | DATAERR | R | 0h | DMA data error, SRC data might be corrupted (PAR or ECC error).<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 24 | ADDRERR | R | 0h | DMA address error, SRC address not reachable.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 23 | PREIRQCH7 | R | 0h | Pre-IRQ for Channel 7. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 22 | PREIRQCH6 | R | 0h | Pre-IRQ for Channel 6. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 21 | PREIRQCH5 | R | 0h | Pre-IRQ for Channel 5. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 20 | PREIRQCH4 | R | 0h | Pre-IRQ for Channel 4. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 19 | PREIRQCH3 | R | 0h | Pre-IRQ for Channel 3. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 18 | PREIRQCH2 | R | 0h | Pre-IRQ for Channel 2. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |



Copyright © 2026 Texas Instruments Incorporated

**Table 5-16. RIS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 17 | PREIRQCH1 | R | 0h | Pre-IRQ for Channel 1. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 16 | PREIRQCH0 | R | 0h | Pre-IRQ for Channel 0. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 15 | DMACH15 | R | 0h | DMA Channel 15 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 14 | DMACH14 | R | 0h | DMA Channel 14 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 13 | DMACH13 | R | 0h | DMA Channel 13 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 12 | DMACH12 | R | 0h | DMA Channel 12 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 11 | DMACH11 | R | 0h | DMA Channel 11 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 10 | DMACH10 | R | 0h | DMA Channel 10 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 9 | DMACH9 | R | 0h | DMA Channel 9 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 8 | DMACH8 | R | 0h | DMA Channel 8 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 7 | DMACH7 | R | 0h | DMA Channel 7 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 6 | DMACH6 | R | 0h | DMA Channel 6 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 5 | DMACH5 | R | 0h | DMA Channel 5 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 4 | DMACH4 | R | 0h | DMA Channel 4 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 3 | DMACH3 | R | 0h | DMA Channel 3 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

**Table 5-16. RIS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | DMACH2 | R | 0h | DMA Channel 2 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 1 | DMACH1 | R | 0h | DMA Channel 1 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 0 | DMACH0 | R | 0h | DMA Channel 0 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

### 5.3.8 MIS (Offset = 1038h) [Reset = 00000000h]

MIS is shown in Figure 5-12 and described in Table 5-17.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK [IMASK] and RIS [RIS] registers.
Note: The number of DMACH is device dependent. Please consult the datasheet of the specific device to map which channel number is implemented.

**Figure 5-12. MIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | DATAERR | ADDRERR |
| R-0h | | | | | | R-0h | R-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| PREIRQCH7 | PREIRQCH6 | PREIRQCH5 | PREIRQCH4 | PREIRQCH3 | PREIRQCH2 | PREIRQCH1 | PREIRQCH0 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DMACH15 | DMACH14 | DMACH13 | DMACH12 | DMACH11 | DMACH10 | DMACH9 | DMACH8 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DMACH7 | DMACH6 | DMACH5 | DMACH4 | DMACH3 | DMACH2 | DMACH1 | DMACH0 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 5-17. MIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | RESERVED | R | 0h | |
| 25 | DATAERR | R | 0h | DMA data error, SRC data might be corrupted (PAR or ECC error).<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 24 | ADDRERR | R | 0h | DMA address error, SRC address not reachable.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 23 | PREIRQCH7 | R | 0h | Pre-IRQ for Channel 7. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 22 | PREIRQCH6 | R | 0h | Pre-IRQ for Channel 6. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 21 | PREIRQCH5 | R | 0h | Pre-IRQ for Channel 5. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 20 | PREIRQCH4 | R | 0h | Pre-IRQ for Channel 4. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 19 | PREIRQCH3 | R | 0h | Pre-IRQ for Channel 3. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 18 | PREIRQCH2 | R | 0h | Pre-IRQ for Channel 2. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 17 | PREIRQCH1 | R | 0h | Pre-IRQ for Channel 1. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |



Copyright © 2026 Texas Instruments Incorporated

**Table 5-17. MIS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 16 | PREIRQCH0 | R | 0h | Pre-IRQ for Channel 0. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 15 | DMACH15 | R | 0h | DMA Channel 15 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 14 | DMACH14 | R | 0h | DMA Channel 14 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 13 | DMACH13 | R | 0h | DMA Channel 13 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 12 | DMACH12 | R | 0h | DMA Channel 12 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 11 | DMACH11 | R | 0h | DMA Channel 11 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 10 | DMACH10 | R | 0h | DMA Channel 10 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 9 | DMACH9 | R | 0h | DMA Channel 9 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 8 | DMACH8 | R | 0h | DMA Channel 8 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 7 | DMACH7 | R | 0h | DMA Channel 7 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 6 | DMACH6 | R | 0h | DMA Channel 6 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 5 | DMACH5 | R | 0h | DMA Channel 5 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 4 | DMACH4 | R | 0h | DMA Channel 4 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 3 | DMACH3 | R | 0h | DMA Channel 3 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 2 | DMACH2 | R | 0h | DMA Channel 2 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

**Table 5-17. MIS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 1 | DMACH1 | R | 0h | DMA Channel 1 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 0 | DMACH0 | R | 0h | DMA Channel 0 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

### 5.3.9 ISET (Offset = 1040h) [Reset = 00000000h]

ISET is shown in Figure 5-13 and described in Table 5-18.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS [RIS] bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS [MIS] bit is also set.
Note: The number of DMACH is device dependent. Please consult the datasheet of the specific device to map which channel number is implemented.

**Figure 5-13. ISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | DATAERR | ADDRERR |
| R-0h | | | | | | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| PREIRQCH7 | PREIRQCH6 | PREIRQCH5 | PREIRQCH4 | PREIRQCH3 | PREIRQCH2 | PREIRQCH1 | PREIRQCH0 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DMACH15 | DMACH14 | DMACH13 | DMACH12 | DMACH11 | DMACH10 | DMACH9 | DMACH8 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DMACH7 | DMACH6 | DMACH5 | DMACH4 | DMACH3 | DMACH2 | DMACH1 | DMACH0 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

**Table 5-18. ISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | RESERVED | R | 0h | |
| 25 | DATAERR | W | 0h | DMA data error, SRC data might be corrupted (PAR or ECC error).<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 24 | ADDRERR | W | 0h | DMA address error, SRC address not reachable.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 23 | PREIRQCH7 | W | 0h | Pre-IRQ for Channel 7. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 22 | PREIRQCH6 | W | 0h | Pre-IRQ for Channel 6. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 21 | PREIRQCH5 | W | 0h | Pre-IRQ for Channel 5. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 20 | PREIRQCH4 | W | 0h | Pre-IRQ for Channel 4. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 19 | PREIRQCH3 | W | 0h | Pre-IRQ for Channel 3. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 18 | PREIRQCH2 | W | 0h | Pre-IRQ for Channel 2. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |



Copyright © 2026 Texas Instruments Incorporated

**Table 5-18. ISET Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 17 | PREIRQCH1 | W | 0h | Pre-IRQ for Channel 1. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 16 | PREIRQCH0 | W | 0h | Pre-IRQ for Channel 0. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 15 | DMACH15 | W | 0h | DMA Channel 0 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 14 | DMACH14 | W | 0h | DMA Channel 14 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 13 | DMACH13 | W | 0h | DMA Channel 13 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 12 | DMACH12 | W | 0h | DMA Channel 12 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 11 | DMACH11 | W | 0h | DMA Channel 11 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 10 | DMACH10 | W | 0h | DMA Channel 0 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 9 | DMACH9 | W | 0h | DMA Channel 9 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 8 | DMACH8 | W | 0h | DMA Channel 8 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 7 | DMACH7 | W | 0h | DMA Channel 7 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 6 | DMACH6 | W | 0h | DMA Channel 6 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 5 | DMACH5 | W | 0h | DMA Channel 5 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 4 | DMACH4 | W | 0h | DMA Channel 4 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 3 | DMACH3 | W | 0h | DMA Channel 3 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Set interrupt |



Copyright © 2026 Texas Instruments Incorporated

**Table 5-18. ISET Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | DMACH2 | W | 0h | DMA Channel 2 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 1 | DMACH1 | W | 0h | DMA Channel 1 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 0 | DMACH0 | W | 0h | DMA Channel 0 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Set interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 5.3.10 ICLR (Offset = 1048h) [Reset = 00000000h]

ICLR is shown in Figure 5-14 and described in Table 5-19.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.
Note: The number of DMACH is device dependent. Please consult the datasheet of the specific device to map which channel number is implemented.

**Figure 5-14. ICLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | DATAERR | ADDRERR |
| R-0h | | | | | | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| PREIRQCH7 | PREIRQCH6 | PREIRQCH5 | PREIRQCH4 | PREIRQCH3 | PREIRQCH2 | PREIRQCH1 | PREIRQCH0 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DMACH15 | DMACH14 | DMACH13 | DMACH12 | DMACH11 | DMACH10 | DMACH9 | DMACH8 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DMACH7 | DMACH6 | DMACH5 | DMACH4 | DMACH3 | DMACH2 | DMACH1 | DMACH0 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

**Table 5-19. ICLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | RESERVED | R | 0h | |
| 25 | DATAERR | W | 0h | DMA data error, SRC data might be corrupted (PAR or ECC error).<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 24 | ADDRERR | W | 0h | DMA address error, SRC address not reachable.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 23 | PREIRQCH7 | W | 0h | Pre-IRQ for Channel 7. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 22 | PREIRQCH6 | W | 0h | Pre-IRQ for Channel 6. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 21 | PREIRQCH5 | W | 0h | Pre-IRQ for Channel 5. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 20 | PREIRQCH4 | W | 0h | Pre-IRQ for Channel 4. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 19 | PREIRQCH3 | W | 0h | Pre-IRQ for Channel 3. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 18 | PREIRQCH2 | W | 0h | Pre-IRQ for Channel 2. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 17 | PREIRQCH1 | W | 0h | Pre-IRQ for Channel 1. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |



Copyright © 2026 Texas Instruments Incorporated

**Table 5-19. ICLR Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 16 | PREIRQCH0 | W | 0h | Pre-IRQ for Channel 0. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 15 | DMACH15 | W | 0h | DMA Channel 15 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 14 | DMACH14 | W | 0h | DMA Channel 14 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 13 | DMACH13 | W | 0h | DMA Channel 13 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 12 | DMACH12 | W | 0h | DMA Channel 12 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 11 | DMACH11 | W | 0h | DMA Channel 11 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 10 | DMACH10 | W | 0h | DMA Channel 10 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 9 | DMACH9 | W | 0h | DMA Channel 9 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 8 | DMACH8 | W | 0h | DMA Channel 8 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 7 | DMACH7 | W | 0h | DMA Channel 7 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 6 | DMACH6 | W | 0h | DMA Channel 6 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 5 | DMACH5 | W | 0h | DMA Channel 5 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 4 | DMACH4 | W | 0h | DMA Channel 4 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 3 | DMACH3 | W | 0h | DMA Channel 3 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 2 | DMACH2 | W | 0h | DMA Channel 2 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |



Copyright © 2026 Texas Instruments Incorporated

**Table 5-19. ICLR Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 1 | DMACH1 | W | 0h | DMA Channel 1 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 0 | DMACH0 | W | 0h | DMA Channel 0 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 5.3.11 IIDX (Offset = 1050h) [Reset = 00000000h]

IIDX is shown in Figure 5-15 and described in Table 5-20.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. Value 0x00 means no event pending. Interrupt 1 is the highest priority, IIDX next highest, 4, 8, . . . IIDX^31 is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred.
On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in RIS [RIS] and MIS [MIS] are cleared as well. After a read from the CPU (not from the debug interface), the register is updated with the next highest priority interrupt, if none are pending, then it should display 0x0.
Note: The number of DMACH is device dependent. Please consult the datasheet of the specific device to map which channel number is implemented.

**Figure 5-15. IIDX**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | STAT | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | |

**Table 5-20. IIDX Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | STAT | R | 0h | Interrupt index status<br>00h = No bit is set means there is no pending interrupt request<br>01h = DMA Channel 0 size counter reached zero (DMASZ=0).<br>02h = DMA Channel 1 size counter reached zero (DMASZ=0).<br>03h = DMA Channel 2 size counter reached zero (DMASZ=0).<br>04h = DMA Channel 3 size counter reached zero (DMASZ=0).<br>05h = DMA Channel 4 size counter reached zero (DMASZ=0).<br>06h = DMA Channel 5 size counter reached zero (DMASZ=0).<br>07h = DMA Channel 6 size counter reached zero (DMASZ=0).<br>08h = DMA Channel 7 size counter reached zero (DMASZ=0).<br>09h = DMA Channel 8 size counter reached zero (DMASZ=0).<br>0Ah = DMA Channel 9 size counter reached zero (DMASZ=0).<br>0Bh = DMA Channel 10 size counter reached zero (DMASZ=0).<br>0Ch = DMA Channel 11 size counter reached zero (DMASZ=0).<br>0Dh = DMA Channel 12 size counter reached zero (DMASZ=0).<br>0Eh = DMA Channel 13 size counter reached zero (DMASZ=0).<br>0Fh = DMA Channel 14 size counter reached zero (DMASZ=0).<br>10h = DMA Channel 15 size counter reached zero (DMASZ=0).<br>11h = PRE-IRQ event for DMA Channel 0.<br>12h = PRE-IRQ event for DMA Channel 1.<br>13h = PRE-IRQ event for DMA Channel 2.<br>14h = PRE-IRQ event for DMA Channel 3.<br>15h = PRE-IRQ event for DMA Channel 4.<br>16h = PRE-IRQ event for DMA Channel 5.<br>17h = PRE-IRQ event for DMA Channel 6.<br>18h = PRE-IRQ event for DMA Channel 7.<br>19h = DMA address error, SRC address not reachable.<br>1Ah = DMA data error, SRC data might be corrupted (PAR or ECC error). |



Copyright © 2026 Texas Instruments Incorporated

### 5.3.12 IMASK (Offset = 1058h) [Reset = 00000000h]

IMASK is shown in Figure 5-16 and described in Table 5-21.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then the corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX [IIDX], as well as MIS [MIS].
Note: The number of DMACH is device dependent. Please consult the datasheet of the specific device to map which channel number is implemented.

**Figure 5-16. IMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | DATAERR | ADDRERR |
| R-0h | | | | | | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| PREIRQCH7 | PREIRQCH6 | PREIRQCH5 | PREIRQCH4 | PREIRQCH3 | PREIRQCH2 | PREIRQCH1 | PREIRQCH0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DMACH15 | DMACH14 | DMACH13 | DMACH12 | DMACH11 | DMACH10 | DMACH9 | DMACH8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DMACH7 | DMACH6 | DMACH5 | DMACH4 | DMACH3 | DMACH2 | DMACH1 | DMACH0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

**Table 5-21. IMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | RESERVED | R | 0h | |
| 25 | DATAERR | R/W | 0h | DMA data error, SRC data might be corrupted (PAR or ECC error).<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 24 | ADDRERR | R/W | 0h | DMA address error, SRC address not reachable.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 23 | PREIRQCH7 | R/W | 0h | Pre-IRQ for Channel 7. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 22 | PREIRQCH6 | R/W | 0h | Pre-IRQ for Channel 6. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 21 | PREIRQCH5 | R/W | 0h | Pre-IRQ for Channel 5. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 20 | PREIRQCH4 | R/W | 0h | Pre-IRQ for Channel 4. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 19 | PREIRQCH3 | R/W | 0h | Pre-IRQ for Channel 3. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 18 | PREIRQCH2 | R/W | 0h | Pre-IRQ for Channel 2. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 17 | PREIRQCH1 | R/W | 0h | Pre-IRQ for Channel 1. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |



Copyright © 2026 Texas Instruments Incorporated

**Table 5-21. IMASK Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 16 | PREIRQCH0 | R/W | 0h | Pre-IRQ for Channel 0. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 15 | DMACH15 | R/W | 0h | DMA Channel 15 interrupt signal. Size counter reached zero (DMASZ=0).<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 14 | DMACH14 | R/W | 0h | DMA Channel 14 interrupt signal. Size counter reached zero (DMASZ=0).<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 13 | DMACH13 | R/W | 0h | DMA Channel 13 interrupt signal. Size counter reached zero (DMASZ=0).<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 12 | DMACH12 | R/W | 0h | DMA Channel 12 interrupt signal. Size counter reached zero (DMASZ=0).<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 11 | DMACH11 | R/W | 0h | DMA Channel 11 interrupt signal. Size counter reached zero (DMASZ=0).<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 10 | DMACH10 | R/W | 0h | DMA Channel 10 interrupt signal. Size counter reached zero (DMASZ=0).<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 9 | DMACH9 | R/W | 0h | DMA Channel 9 interrupt signal. Size counter reached zero (DMASZ=0).<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 8 | DMACH8 | R/W | 0h | DMA Channel 8 interrupt signal. Size counter reached zero (DMASZ=0).<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 7 | DMACH7 | R/W | 0h | DMA Channel 7 interrupt signal. Size counter reached zero (DMASZ=0).<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 6 | DMACH6 | R/W | 0h | DMA Channel 6 interrupt signal. Size counter reached zero (DMASZ=0).<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 5 | DMACH5 | R/W | 0h | DMA Channel 5 interrupt signal. Size counter reached zero (DMASZ=0).<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 4 | DMACH4 | R/W | 0h | DMA Channel 4 interrupt signal. Size counter reached zero (DMASZ=0).<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 3 | DMACH3 | R/W | 0h | DMA Channel 3 interrupt signal. Size counter reached zero (DMASZ=0).<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 2 | DMACH2 | R/W | 0h | DMA Channel 2 interrupt signal. Size counter reached zero (DMASZ=0).<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |



Copyright © 2026 Texas Instruments Incorporated

**Table 5-21. IMASK Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 1 | DMACH1 | R/W | 0h | DMA Channel 1 interrupt signal. Size counter reached zero (DMASZ=0).<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 0 | DMACH0 | R/W | 0h | DMA Channel 0 interrupt signal. Size counter reached zero (DMASZ=0).<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |



Copyright © 2026 Texas Instruments Incorporated

### 5.3.13 RIS (Offset = 1060h) [Reset = 00000000h]

RIS is shown in Figure 5-17 and described in Table 5-22.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR [ICLR] register bit even if the corresponding IMASK [IMASK] bit is not enabled.
Note: The number of DMACH is device dependent. Please consult the datasheet of the specific device to map which channel number is implemented.

**Figure 5-17. RIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | DATAERR | ADDRERR |
| R-0h | | | | | | R-0h | R-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| PREIRQCH7 | PREIRQCH6 | PREIRQCH5 | PREIRQCH4 | PREIRQCH3 | PREIRQCH2 | PREIRQCH1 | PREIRQCH0 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DMACH15 | DMACH14 | DMACH13 | DMACH12 | DMACH11 | DMACH10 | DMACH9 | DMACH8 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DMACH7 | DMACH6 | DMACH5 | DMACH4 | DMACH3 | DMACH2 | DMACH1 | DMACH0 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 5-22. RIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | RESERVED | R | 0h | |
| 25 | DATAERR | R | 0h | DMA data error, SRC data might be corrupted (PAR or ECC error).<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 24 | ADDRERR | R | 0h | DMA address error, SRC address not reachable.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 23 | PREIRQCH7 | R | 0h | Pre-IRQ for Channel 7. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 22 | PREIRQCH6 | R | 0h | Pre-IRQ for Channel 6. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 21 | PREIRQCH5 | R | 0h | Pre-IRQ for Channel 5. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 20 | PREIRQCH4 | R | 0h | Pre-IRQ for Channel 4. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 19 | PREIRQCH3 | R | 0h | Pre-IRQ for Channel 3. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 18 | PREIRQCH2 | R | 0h | Pre-IRQ for Channel 2. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |



Copyright © 2026 Texas Instruments Incorporated

**Table 5-22. RIS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 17 | PREIRQCH1 | R | 0h | Pre-IRQ for Channel 1. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 16 | PREIRQCH0 | R | 0h | Pre-IRQ for Channel 0. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 15 | DMACH15 | R | 0h | DMA Channel 15 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 14 | DMACH14 | R | 0h | DMA Channel 14 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 13 | DMACH13 | R | 0h | DMA Channel 13 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 12 | DMACH12 | R | 0h | DMA Channel 12 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 11 | DMACH11 | R | 0h | DMA Channel 11 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 10 | DMACH10 | R | 0h | DMA Channel 10 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 9 | DMACH9 | R | 0h | DMA Channel 9 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 8 | DMACH8 | R | 0h | DMA Channel 8 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 7 | DMACH7 | R | 0h | DMA Channel 7 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 6 | DMACH6 | R | 0h | DMA Channel 6 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 5 | DMACH5 | R | 0h | DMA Channel 5 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 4 | DMACH4 | R | 0h | DMA Channel 4 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 3 | DMACH3 | R | 0h | DMA Channel 3 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

**Table 5-22. RIS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | DMACH2 | R | 0h | DMA Channel 2 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 1 | DMACH1 | R | 0h | DMA Channel 1 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 0 | DMACH0 | R | 0h | DMA Channel 0 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

### 5.3.14 MIS (Offset = 1068h) [Reset = 00000000h]

MIS is shown in Figure 5-18 and described in Table 5-23.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK [IMASK] and RIS [RIS] registers.
Note: The number of DMACH is device dependent. Please consult the datasheet of the specific device to map which channel number is implemented.

**Figure 5-18. MIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | DATAERR | ADDRERR |
| R-0h | | | | | | R-0h | R-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| PREIRQCH7 | PREIRQCH6 | PREIRQCH5 | PREIRQCH4 | PREIRQCH3 | PREIRQCH2 | PREIRQCH1 | PREIRQCH0 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DMACH15 | DMACH14 | DMACH13 | DMACH12 | DMACH11 | DMACH10 | DMACH9 | DMACH8 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DMACH7 | DMACH6 | DMACH5 | DMACH4 | DMACH3 | DMACH2 | DMACH1 | DMACH0 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 5-23. MIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | RESERVED | R | 0h | |
| 25 | DATAERR | R | 0h | DMA data error, SRC data might be corrupted (PAR or ECC error).<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 24 | ADDRERR | R | 0h | DMA address error, SRC address not reachable.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 23 | PREIRQCH7 | R | 0h | Pre-IRQ for Channel 7. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 22 | PREIRQCH6 | R | 0h | Pre-IRQ for Channel 6. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 21 | PREIRQCH5 | R | 0h | Pre-IRQ for Channel 5. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 20 | PREIRQCH4 | R | 0h | Pre-IRQ for Channel 4. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 19 | PREIRQCH3 | R | 0h | Pre-IRQ for Channel 3. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 18 | PREIRQCH2 | R | 0h | Pre-IRQ for Channel 2. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 17 | PREIRQCH1 | R | 0h | Pre-IRQ for Channel 1. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |



Copyright © 2026 Texas Instruments Incorporated

**Table 5-23. MIS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 16 | PREIRQCH0 | R | 0h | Pre-IRQ for Channel 0. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 15 | DMACH15 | R | 0h | DMA Channel 15 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 14 | DMACH14 | R | 0h | DMA Channel 14 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 13 | DMACH13 | R | 0h | DMA Channel 13 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 12 | DMACH12 | R | 0h | DMA Channel 12 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 11 | DMACH11 | R | 0h | DMA Channel 11 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 10 | DMACH10 | R | 0h | DMA Channel 10 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 9 | DMACH9 | R | 0h | DMA Channel 9 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 8 | DMACH8 | R | 0h | DMA Channel 8 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 7 | DMACH7 | R | 0h | DMA Channel 7 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 6 | DMACH6 | R | 0h | DMA Channel 6 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 5 | DMACH5 | R | 0h | DMA Channel 5 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 4 | DMACH4 | R | 0h | DMA Channel 4 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 3 | DMACH3 | R | 0h | DMA Channel 3 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 2 | DMACH2 | R | 0h | DMA Channel 2 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

**Table 5-23. MIS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 1 | DMACH1 | R | 0h | DMA Channel 1 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 0 | DMACH0 | R | 0h | DMA Channel 0 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

### 5.3.15 ISET (Offset = 1070h) [Reset = 00000000h]

ISET is shown in Figure 5-19 and described in Table 5-24.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS [RIS] bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS [MIS] bit is also set.
Note: The number of DMACH is device dependent. Please consult the datasheet of the specific device to map which channel number is implemented.

**Figure 5-19. ISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | DATAERR | ADDRERR |
| R-0h | | | | | | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| PREIRQCH7 | PREIRQCH6 | PREIRQCH5 | PREIRQCH4 | PREIRQCH3 | PREIRQCH2 | PREIRQCH1 | PREIRQCH0 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DMACH15 | DMACH14 | DMACH13 | DMACH12 | DMACH11 | DMACH10 | DMACH9 | DMACH8 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DMACH7 | DMACH6 | DMACH5 | DMACH4 | DMACH3 | DMACH2 | DMACH1 | DMACH0 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

**Table 5-24. ISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | RESERVED | R | 0h | |
| 25 | DATAERR | W | 0h | DMA data error, SRC data might be corrupted (PAR or ECC error).<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 24 | ADDRERR | W | 0h | DMA address error, SRC address not reachable.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 23 | PREIRQCH7 | W | 0h | Pre-IRQ for Channel 7. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 22 | PREIRQCH6 | W | 0h | Pre-IRQ for Channel 6. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 21 | PREIRQCH5 | W | 0h | Pre-IRQ for Channel 5. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 20 | PREIRQCH4 | W | 0h | Pre-IRQ for Channel 4. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 19 | PREIRQCH3 | W | 0h | Pre-IRQ for Channel 3. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 18 | PREIRQCH2 | W | 0h | Pre-IRQ for Channel 2. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |



Copyright © 2026 Texas Instruments Incorporated

**Table 5-24. ISET Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 17 | PREIRQCH1 | W | 0h | Pre-IRQ for Channel 1. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 16 | PREIRQCH0 | W | 0h | Pre-IRQ for Channel 0. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 15 | DMACH15 | W | 0h | DMA Channel 0 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 14 | DMACH14 | W | 0h | DMA Channel 14 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 13 | DMACH13 | W | 0h | DMA Channel 13 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 12 | DMACH12 | W | 0h | DMA Channel 12 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 11 | DMACH11 | W | 0h | DMA Channel 11 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 10 | DMACH10 | W | 0h | DMA Channel 0 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 9 | DMACH9 | W | 0h | DMA Channel 9 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 8 | DMACH8 | W | 0h | DMA Channel 8 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 7 | DMACH7 | W | 0h | DMA Channel 7 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 6 | DMACH6 | W | 0h | DMA Channel 6 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 5 | DMACH5 | W | 0h | DMA Channel 5 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 4 | DMACH4 | W | 0h | DMA Channel 4 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 3 | DMACH3 | W | 0h | DMA Channel 3 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Set interrupt |



Copyright © 2026 Texas Instruments Incorporated

**Table 5-24. ISET Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | DMACH2 | W | 0h | DMA Channel 2 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 1 | DMACH1 | W | 0h | DMA Channel 1 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 0 | DMACH0 | W | 0h | DMA Channel 0 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Set interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 5.3.16 ICLR (Offset = 1078h) [Reset = 00000000h]

ICLR is shown in Figure 5-20 and described in Table 5-25.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.
Note: The number of DMACH is device dependent. Please consult the datasheet of the specific device to map which channel number is implemented.

**Figure 5-20. ICLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | DATAERR | ADDRERR |
| R-0h | | | | | | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| PREIRQCH7 | PREIRQCH6 | PREIRQCH5 | PREIRQCH4 | PREIRQCH3 | PREIRQCH2 | PREIRQCH1 | PREIRQCH0 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DMACH15 | DMACH14 | DMACH13 | DMACH12 | DMACH11 | DMACH10 | DMACH9 | DMACH8 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DMACH7 | DMACH6 | DMACH5 | DMACH4 | DMACH3 | DMACH2 | DMACH1 | DMACH0 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

**Table 5-25. ICLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | RESERVED | R | 0h | |
| 25 | DATAERR | W | 0h | DMA data error, SRC data might be corrupted (PAR or ECC error).<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 24 | ADDRERR | W | 0h | DMA address error, SRC address not reachable.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 23 | PREIRQCH7 | W | 0h | Pre-IRQ for Channel 7. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 22 | PREIRQCH6 | W | 0h | Pre-IRQ for Channel 6. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 21 | PREIRQCH5 | W | 0h | Pre-IRQ for Channel 5. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 20 | PREIRQCH4 | W | 0h | Pre-IRQ for Channel 4. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 19 | PREIRQCH3 | W | 0h | Pre-IRQ for Channel 3. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 18 | PREIRQCH2 | W | 0h | Pre-IRQ for Channel 2. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 17 | PREIRQCH1 | W | 0h | Pre-IRQ for Channel 1. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |



Copyright © 2026 Texas Instruments Incorporated

**Table 5-25. ICLR Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 16 | PREIRQCH0 | W | 0h | Pre-IRQ for Channel 0. Size counter reached Pre-IRQ threshold.<br>0h = Clear interrupt mask bit<br>1h = Set interrupt mask bit |
| 15 | DMACH15 | W | 0h | DMA Channel 15 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 14 | DMACH14 | W | 0h | DMA Channel 14 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 13 | DMACH13 | W | 0h | DMA Channel 13 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 12 | DMACH12 | W | 0h | DMA Channel 12 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 11 | DMACH11 | W | 0h | DMA Channel 11 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 10 | DMACH10 | W | 0h | DMA Channel 10 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 9 | DMACH9 | W | 0h | DMA Channel 9 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 8 | DMACH8 | W | 0h | DMA Channel 8 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 7 | DMACH7 | W | 0h | DMA Channel 7 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 6 | DMACH6 | W | 0h | DMA Channel 6 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 5 | DMACH5 | W | 0h | DMA Channel 5 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 4 | DMACH4 | W | 0h | DMA Channel 4 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 3 | DMACH3 | W | 0h | DMA Channel 3 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 2 | DMACH2 | W | 0h | DMA Channel 2 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |



Copyright © 2026 Texas Instruments Incorporated

**Table 5-25. ICLR Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 1 | DMACH1 | W | 0h | DMA Channel 1 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 0 | DMACH0 | W | 0h | DMA Channel 0 interrupt signals that size counter reached zero (DMASZ=0).<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 5.3.17 EVT_MODE (Offset = 10E0h) [Reset = 00000000h]

EVT_MODE is shown in Figure 5-21 and described in Table 5-26.

Return to the Summary Table.

Event mode register. It is used to select whether each line is disabled, in software mode (software clears the RIS) or in hardware mode (hardware clears the RIS)

**Figure 5-21. EVT_MODE**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | EVT1_CFG | | INT0_CFG | |
| R-0h | | | | R-0h | | R-0h | |

**Table 5-26. EVT_MODE Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3-2 | EVT1_CFG | R | 2h | Event line mode select for event corresponding to generic event GEN_EVENT<br>0h = The interrupt or event line is disabled.<br>1h = The interrupt or event line is in software mode. Software must clear the RIS.<br>2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag. |
| 1-0 | INT0_CFG | R | 1h | Event line mode select for event corresponding to interrupt event CPU_INT<br>0h = The interrupt or event line is disabled.<br>1h = The interrupt or event line is in software mode. Software must clear the RIS.<br>2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag. |



Copyright © 2026 Texas Instruments Incorporated

### 5.3.18 DESC (Offset = 10FCh) [Reset = 00000000h]

DESC is shown in Figure 5-22 and described in Table 5-27.

Return to the Summary Table.

This register identifies the peripheral and its exact version.

**Figure 5-22. DESC**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MODULEID | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| FEATUREVER | | | | RESERVED | | | | MAJREV | | | | MINREV | | | |
| R-0h | | | | R-0h | | | | R-0h | | | | R-0h | | | |

**Table 5-27. DESC Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | MODULEID | R | 2511h | Module identification contains a unique peripheral identification number. The assignments are maintained in a central database for all of the platform modules to ensure uniqueness.<br>0h = Smallest value<br>FFFFh = Highest possible value |
| 15-12 | FEATUREVER | R | Fh | Feature Set for the DMA: number of DMA channel minus one (e.g. 0->1ch, 2->3ch, 15->16ch).<br>0h = Smallest value (1 channel)<br>Fh = Highest value (16 channel) |
| 11-8 | RESERVED | R | 0h | |
| 7-4 | MAJREV | R | 0h | Major rev of the IP<br>0h = Smallest value<br>Fh = Highest possible value |
| 3-0 | MINREV | R | 0h | Minor rev of the IP<br>0h = Smallest value<br>Fh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 5.3.19 DMAPRIO (Offset = 1100h) [Reset = 00000000h]

DMAPRIO is shown in Figure 5-23 and described in Table 5-28.

Return to the Summary Table.

DMA Channel Priority Control

**Figure 5-23. DMAPRIO**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | BURSTSZ | |
| R-0h | | | | | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | ROUNDROBIN |
| R-0h | | | | | | | R/W-0h |

**Table 5-28. DMAPRIO Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-18 | RESERVED | R | 0h | |
| 17-16 | BURSTSZ | R/W | 0h | Define the burst size of a block transfer, before the priority is re-evaluated<br>0h = There is no burst size, the whole block transfer is completed on one transfer without interruption<br>1h = The burst size is 8, after 9 transfers the block transfer is interrupted and the priority is reevaluated<br>2h = The burst size is 16, after 17 transfers the block transfer is interrupted and the priority is reevaluated<br>3h = The burst size is 32, after 33 transfers the block transfer is interrupted and the priority is reevaluated |
| 15-1 | RESERVED | R | 0h | |
| 0 | ROUNDROBIN | R/W | 0h | Round robin. This bit enables the round-robin DMA channel priorities.<br>0h = Roundrobin priority disabled, DMA channel priority is fixed: DMA0-DMA1-DMA2-...-DMA16<br>1h = Roundrobin priority enabled, DMA channel priority changes with each transfer |



Copyright © 2026 Texas Instruments Incorporated

### 5.3.20 DMATCTL[j] (Offset = 1110h + formula) [Reset = 00000000h]

DMATCTL[j] is shown in Figure 5-24 and described in Table 5-29.

Return to the Summary Table.

DMA Trigger Control

Offset = 1110h + (j * 4h); where j = 0h to Fh

**Figure 5-24. DMATCTL[j]**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DMATINT | RESERVED | DMATSEL | | | | | |
| R/W-0h | R-0h | R/W-0h | | | | | |

**Table 5-29. DMATCTL[j] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7 | DMATINT | R/W | 0h | DMA Trigger by Internal Channel<br>0h = DMATSEL will define external trigger select as transfer trigger.<br>1h = DMATSEL will define internal channel as transfer trigger select.<br>0-> Channel0-done, 1-> Channel1-done, ... |
| 6 | RESERVED | R | 0h | |
| 5-0 | DMATSEL | R/W | 0h | DMA Trigger Select<br>Note: Reference the datasheet of the device to see the specific trigger mapping.<br>00h = Software trigger request<br>3Fh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 5.3.21 DMACTL[j] (Offset = 1200h + formula) [Reset = 00000000h]

DMACTL[j] is shown in Figure 5-25 and described in Table 5-30.

Return to the Summary Table.

DMA Channel Control

Offset = 1200h + (j * 10h); where j = 0h to Fh

**Figure 5-25. DMACTL[j]**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | DMATM | | RESERVED | | DMAEM | |
| R-0h | | R/W-0h | | R-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DMADSTINCR | | | | DMASRCINCR | | | |
| R/W-0h | | | | R/W-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | DMADSTWDTH | | | RESERVED | DMASRCWDTH | | |
| R-0h | R/W-0h | | | R-0h | R/W-0h | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | DMAPREIRQ | | | DMAAUTOEN | | DMAEN | DMAREQ |
| R-0h | R/W-0h | | | R/W-0h | | R/W-0h | R/W-0h |

**Table 5-30. DMACTL[j] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | RESERVED | R | 0h | |
| 29-28 | DMATM | R/W | 0h | DMA transfer mode register<br>Note: The repeat-single (2h) and repeat-block (3h) transfer are only available in a FULL-channel configuration. Please consult the datasheet of the specific device to map which channel number has FULL or BASIC capability. In a BASIC channel configuration only the values for single (0h) and block (1h) transfer can be set.<br>0h = Single transfer. Each transfers requires a new trigger. When the DMASZ counts down to zero an event can be generated and the DMAEN is cleared.<br>1h = Block transfer. Each trigger transfers the complete block defined in DMASZ. After the transfer is complete an event can be generated and the DMAEN is cleared.<br>2h = Repeated single transfer. Each transfers requires a new trigger. When the DMASZ counts down to zero an event can be generated. After the last transfer the DMASA, DMADA, DAMSZ registers are restored to its initial value and the DMAEN stays enabled.<br>3h = Repeated block transfer. Each trigger transfers the complete block defined in DMASZ. After the last transfer the DMASA, DMADA, DAMSZ registers are restored to its initial value and the DMAEN stays enabled. |
| 27-26 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

**Table 5-30. DMACTL[j] Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 25-24 | DMAEM | R/W | 0h | DMA extended mode<br>Note: The extended transfer modes are only available in a FULL-channel configuration. Please consult the datasheet of the specific device to map which channel number has FULL or BASIC capability. In a BASIC channel configuration this register is a read-only register and reads 0x0.<br>0h = Normal mode is related to transfers from SRC to DST<br>1h = Gather mode will read a data from an address table located at SA, and the data is transfered to the DA. Note: This feature is not present in all devices. Consult the device datasheet.<br>2h = Fill mode will copy the SA register content as data to DA<br>3h = Table mode will read an address and data value from SA and write the data to address |
| 23-20 | DMADSTINCR | R/W | 0h | DMA destination increment. This bit selects automatic incrementing or decrementing of the destination address DMADA for each transfer. The amount of change to the DMADA is based on the definitin in the DMADSTWDTH. For example an increment of 1 (+1) on a WORD transfer will increment the DMADA by 4. Note: Stride options are not present in all devices. Consult the device datasheet.<br>0h = Address is unchanged (+0)<br>2h = Decremented by 1 (-1 * DMADSTWDTH)<br>3h = Incremented by 1 (+1 * DMADSTWDTH)<br>8h = Stride size 2 (+2 * DMADSTWDTH)<br>9h = Stride size 3 (+3 * DMADSTWDTH)<br>Ah = Stride size 4 (+4 * DMADSTWDTH)<br>Bh = Stride size 5 (+5 * DMADSTWDTH)<br>Ch = Stride size 6 (+6 * DMADSTWDTH)<br>Dh = Stride size 7 (+7 * DMADSTWDTH)<br>Eh = Stride size 8 (+8 * DMADSTWDTH)<br>Fh = Stride size 9 (+9 * DMADSTWDTH) |
| 19-16 | DMASRCINCR | R/W | 0h | DMA source increment. This bit selects automatic incrementing or decrementing of the source address DMASA for each transfer. The amount of change to the DMASA is based on the definitin in the DMASRCWDTH. For example an increment of 1 (+1) on a WORD transfer will increment the DMASA by 4. Note: Stride options are not present in all devices. Consult the device datasheet.<br>0h = Address is unchanged (+0)<br>2h = Decremented by 1 (-1 * DMASRCWDTH)<br>3h = Incremented by 1 (+1 * DMASRCWDTH)<br>8h = Stride size 2 (+2 * DMASRCWDTH)<br>9h = Stride size 3 (+3 * DMASRCWDTH)<br>Ah = Stride size 4 (+4 * DMASRCWDTH)<br>Bh = Stride size 5 (+5 * DMASRCWDTH)<br>Ch = Stride size 6 (+6 * DMASRCWDTH)<br>Dh = Stride size 7 (+7 * DMASRCWDTH)<br>Eh = Stride size 8 (+8 * DMASRCWDTH)<br>Fh = Stride size 9 (+9 * DMASRCWDTH) |
| 15 | RESERVED | R | 0h | |
| 14-12 | DMADSTWDTH | R/W | 0h | DMA destination width. This bit selects the destination as a byte, half word, word, long word or long-long word.<br>0h = Destination data width is BYTE (8-bit)<br>1h = Destination data width is HALF-WORD (16-bit)<br>2h = Destination data width is WORD (32-bit)<br>3h = Destination data width is LONG-WORD (64-bit)<br>4h = Destination data width is LONG-LONG-WORD (128-bit). Note: This feature is not present in all devices. Consult the device datasheet. |
| 11 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

**Table 5-30. DMACTL[j] Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 10-8 | DMASRCWDTH | R/W | 0h | DMA source width. This bit selects the source data width as a byte, half word, word, long word or long-long word.<br>0h = Source data width is BYTE (8-bit)<br>1h = Source data width is HALF-WORD (16-bit)<br>2h = Source data width is WORD (32-bit)<br>3h = Source data width is LONG-WORD (64-bit)<br>4h = Source data width is LONG-LONG-WORD (128-bit). Note: This feature is not present in all devices. Consult the device datasheet. |
| 7 | RESERVED | R | 0h | |
| 6-4 | DMAPREIRQ | R/W | 0h | Enable an early IRQ event. This can help software to react quicker to and DMA done event or allows some additional configuration before the channel is complete.<br>Note: This register is only available in a FULL-channel configuration. Please consult the datasheet of the specific device to map which channel number has FULL or BASIC capability. In a BASIC configuration this register is a read only value and always reads as 0x0.<br>0h = Pre-IRQ event disabled.<br>1h = Issure Pre-IRQ event when DMASZ=1<br>2h = Issure Pre-IRQ event when DMASZ=2<br>3h = Issure Pre-IRQ event when DMASZ=4<br>4h = Issure Pre-IRQ event when DMASZ=8<br>5h = Issure Pre-IRQ event when DMASZ=32<br>6h = Issure Pre-IRQ event when DMASZ=64<br>7h = Issure Pre-IRQ event when DMASZ reached the half size point of the original transfer size |
| 3-2 | DMAAUTOEN | R/W | 0h | Automatic DMA channel enable on DMASA, DMADA, DMASZ register write.<br>If channel is configured as SW trigger (DMATCTL=0), the AUTOEN will set the DMAEN and DMAREQ.<br>If channel is configured as HW trigger (DMACTL!=0), the AUTOEN will only set the DMAEN. Note: This feature is not present in all devices. Consult the device specific datasheet.<br>0h = No automatic DMA enable<br>1h = Automatic DMA enable on DMASA register write.<br>2h = Automatic DMA enable on DMADA register write.<br>3h = Automatic DMA enable on DMASZ register write. |
| 1 | DMAEN | R/W | 0h | DMA enable<br>0h = DMA channel disabled<br>1h = DMA channel enabled |
| 0 | DMAREQ | R/W | 0h | DMA request. Software-controlled DMA start. DMAREQ is reset automatically.<br>0h = Default read value<br>1h = DMA transfer request (start DMA) |



Copyright © 2026 Texas Instruments Incorporated

### 5.3.22 DMASA[j] (Offset = 1204h + formula) [Reset = 00000000h]

DMASA[j] is shown in Figure 5-26 and described in Table 5-31.

Return to the Summary Table.

DMA Channel Source Address

Offset = 1204h + (j * 10h); where j = 0h to Fh

**Figure 5-26. DMASA[j]**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADDR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 5-31. DMASA[j] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | ADDR | R/W | 0h | DMA Channel Source Address<br>0h = Smallest value<br>FFFFFFFFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 5.3.23 DMADA[j] (Offset = 1208h + formula) [Reset = 00000000h]

DMADA[j] is shown in Figure 5-27 and described in Table 5-32.

Return to the Summary Table.

DMA Channel Destination Address

Offset = 1208h + (j * 10h); where j = 0h to Fh

**Figure 5-27. DMADA[j]**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADDR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 5-32. DMADA[j] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | ADDR | R/W | 0h | DMA Channel Destination Address<br>0h = Smallest value<br>FFFFFFFFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 5.3.24 DMASZ[j] (Offset = 120Ch + formula) [Reset = 00000000h]

DMASZ[j] is shown in Figure 5-28 and described in Table 5-33.

Return to the Summary Table.

DMA Channel Size

Offset = 120Ch + (j * 10h); where j = 0h to Fh

**Figure 5-28. DMASZ[j]**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | SIZE | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | R/W-0h | | | | | | | | | | | | | | | |

**Table 5-33. DMASZ[j] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15-0 | SIZE | R/W | 0h | DMA Channel Size in number of transfers<br>0h = Smallest value<br>FFFFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

Chapter 6

# NVM (Flash)

The nonvolatile memory (NVM) system provides nonvolatile flash memory for storing executable code and data.


**6.1 NVM Overview** ..... 585
**6.2 Flash Memory Bank Organization** ..... 586
**6.3 Flash Controller** ..... 588
**6.4 Write Protection** ..... 598
**6.5 Read Interface** ..... 600
**6.6 FLASHCTL Registers** ..... 601





Copyright © 2026 Texas Instruments Incorporated

## 6.1 NVM Overview

The nonvolatile memory system provides in-system programmable flash memory for storing executable code and data. This chapter describes the entire functionality provided by the nonvolatile memory system.

### 6.1.1 Key Features

Key features of the nonvolatile memory system include:

- In-circuit program and erase supported across the entire supply voltage range
- Internal programming voltage generation
- 64-bit flash word size (72-bit when optional ECC is present)
- Static write protection (latched at boot and held until BOR or POR)
- Dynamic write protection (configurable at runtime)
- Sector (1KB) and bank (up to 256KB) erase
- Automatic hardware preverification to extend flash bank longevity
- Automatic hardware post-verification of program/erase
- Optional ECC protection (SECDED)
- Optional bank address swap mode (for seamless dual-image firmware updates)
- Optional program register cache for time efficient programming (2, 4, or 8 flash words)

**Note**

To determine if a device has any of the optional features described above, review the nonvolatile memory system detailed description in the corresponding device data sheet.

### 6.1.2 System Components

The nonvolatile memory system consists of three components (listed below):

- One or more flash memory banks (for storing code and data)
- The flash controller (for managing all program/erase operations on the flash banks)
- The read interface (for interfacing the flash banks to the CPU and peripheral bus)

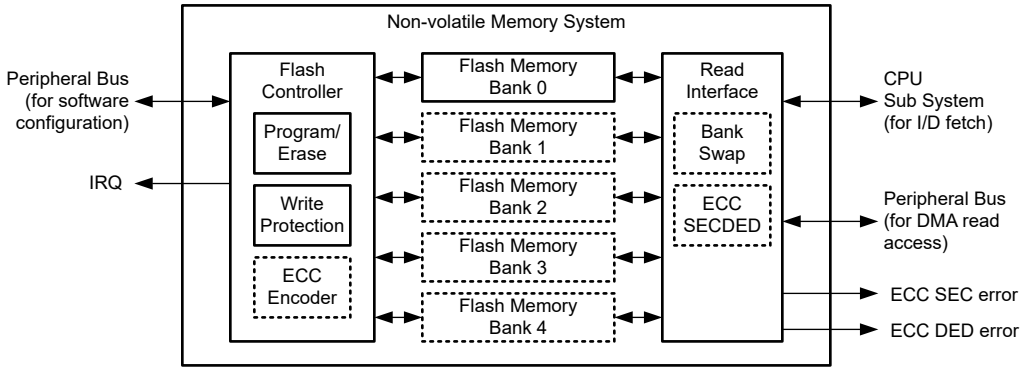



**Figure 6-1. Non-volatile Memory System Components**

### 6.1.3 Terminology

Key flash bank terms are defined in this section to be used as a reference for the rest of this chapter.

**Table 6-1. NVM System Terminology**

| Term | Definition | Size |
|---|---|---|
| Flash word | Basic data size for program and read operations on the flash memory (also the read bus width to the system) | 64 data bits (72 bits with ECC) |
| Word line | Group of flash words within a sector, with maximum program operation limit before sector erase | 16 flash words (128 data bytes, optionally 16 ECC bytes) |
| Sector | Group of word lines that are erased together (minimum erase resolution of the flash memory) | 8 word lines (1024 data bytes, optionally 128 ECC bytes) |



Copyright © 2026 Texas Instruments Incorporated

**Table 6-1. NVM System Terminology (continued)**

| Term | Definition | Size |
|---|---|---|
| Bank | Group of sectors that can be mass erased in one operation. Only one read, program, erase, or verify operation can run concurrently on a given bank. | Variable |
| Region | Logical assignment of a region of flash memory from a bank. | Variable |

## 6.2 Flash Memory Bank Organization

The flash memory is used for storing application code and data, the device boot configuration, and parameters which are preprogrammed by TI from the factory. The flash memory is organized into one or more banks, and the memory in each bank is further mapped into one or more logical memory regions and assigned system address space for use by the application.

### *6.2.1 Banks*

The nonvolatile memory system provides support for up to 5 flash memory banks (enumerated as BANK0 through BANK4). The number of flash banks present is device dependent. To determine the bank scheme of a particular device, review the detailed description section of the specific device data sheet. Most devices implement a single flash bank (BANK0).

On devices with a single flash bank, an ongoing program/erase operation will stall all read requests to the flash memory until the operation has completed and the flash controller has released control of the bank. On devices with more than one flash bank, a program/erase operation on a bank will also stall read requests issued to the bank which is executing the program/erase operation, but it will not stall read requests issued to any other bank. As such, the presence of multiple banks enables application cases such as:

- Dual-image firmware updates (an application can execute code out of one flash bank while a second image is programmed to a second symmetrical flash bank without stalling the application execution)
- EEPROM emulation (an application can execute code out of one flash bank while a second flash bank is used for writing data without stalling the application execution)

### *6.2.2 Flash Memory Regions*

The memory within each bank is mapped to one or more logical regions based upon the functions that the memory in each bank supports. There are four regions: FACTORY, NONMAIN (Configuration NVM), MAIN (Flash Memory) , and DATA.

**Table 6-2. Flash Memory Regions**

| Flash Memory Region | Region Contents | Executable | Used by | Programmed by |
|---|---|---|---|---|
| FACTORY | Device ID and other parameters | No | Application | TI only (not modifiable) |
| NONMAIN (Configuration NVM) | Device boot configuration (BCR and BSL) | No | Boot ROM | TI, User |
| MAIN (Flash Memory) | Application code and data | Yes | Application | User |
| DATA | Data, or EEPROM emulation | No | Application | User |

Devices with one bank implement the FACTORY, NONMAIN, and MAIN regions on BANK0 (the only bank present), and the data region is not available. Devices with multiple banks also implement FACTORY, NONMAIN, and MAIN regions on BANK0, but include additional banks (BANK1 through BANK4) that can implement MAIN or DATA regions.

For a detailed description of the contents of the read-only FACTORY region, see Section 1.5.

### *6.2.3 Addressing*

The flash memory regions are assigned to address space in the system memory map.



Copyright © 2026 Texas Instruments Incorporated

The NONMAIN, DATA, and FACTORY regions are assigned to the peripheral address space (0x4000.0000) as they do not contain any executable code. The CPU should not fetch executable instructions from this region.

The MAIN region is assigned to both the code address space (0x0000.0000) and the peripheral address space (0x4000.0000). Instruction and data fetches are recommended to always be done through the code address space as this gives the best performance. The CPU should not fetch executable instructions from this region.

**ECC**

On devices which have error correction code (ECC) support, the ECC codes for all memory regions are also assigned to address space and it is possible for software to read the ECC codes as data for diagnostic purposes. It is also possible to read the contents of any of the memory regions without ECC correction applied.

#### 6.2.3.1 Flash Memory Map

The following table lists the system address space assignments. These assignments are consistent for all devices.

**Table 6-3. Flash Region Memory Map**

| Region | Read Type | ECC Behavior | Base Address |
|---|---|---|---|
| NONMAIN | Data read | Corrected | 0x41C0.0000 |
| | | Uncorrected | 0x41C1.0000 |
| | | ECC code | 0x41C2.0000 |
| MAIN | Instruction fetch or data read | Corrected | 0x0000.0000 |
| | | Uncorrected | 0x0040.0000 |
| | Data read | ECC code | 0x4180.0000 |
| DATA | Data read | Corrected | 0x41D0.0000 |
| | | Uncorrected | 0x41E0.0000 |
| | | ECC code | 0x41F0.0000 |
| FACTORY | Data read | Corrected | 0x41C4.0000 |
| | | Uncorrected | 0x41C5.0000 |
| | | ECC code | 0x41C6.0000 |

NONMAIN, DATA, and FACTORY data reads are processed through the peripheral bus and peripheral address space only. MAIN regions can be accessed through either the CPU bus matrix or through the peripheral bus, depending on whether code address space or peripheral address space is used. The code address space is recommended for CPU accesses (instruction fetches or data reads), as these accesses do not cross the peripheral bus and thus do not compete with the DMA for control of the peripheral bus. See the bus architecture section for a detailed description of the bus interconnect.

On devices that have ECC, an access to an ECC code address returns the 8-bit ECC value for the entire 64-bit flash word that was accessed. On devices that do not have ECC, accesses to the corrected and uncorrected ECC address spaces with the same offset read the same, and ECC code addresses read as 0x0.

### *6.2.4 Memory Organization Examples*

Figure 6-2 is an example of a single bank configuration with a 64KB MAIN region. NONMAIN and FACTORY regions are also included in the single bank (BANK0) with the MAIN region. Most devices with a main region ≤128KB in size implement a single-bank configuration.



Copyright © 2026 Texas Instruments Incorporated

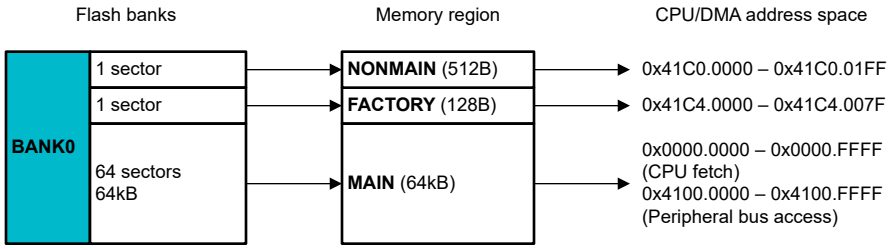



**Figure 6-2. Memory Organization Example - Single Bank Configuration**

Figure 6-3 is an example of a three bank configuration with a 512KB MAIN region split across BANK0 and BANK1, with a 16KB DATA region provided in BANK2. Like the single bank example, NONMAIN and FACTORY regions are included in BANK0. This example supports EEPROM emulation in the DATA region without stalling fetches to MAIN, and it also supports dual-image applications where BANK0 main can be written to without stalling fetches to BANK1 main (and the reverse). Most devices with a main region ≥256KB in size implement a multibank configuration.

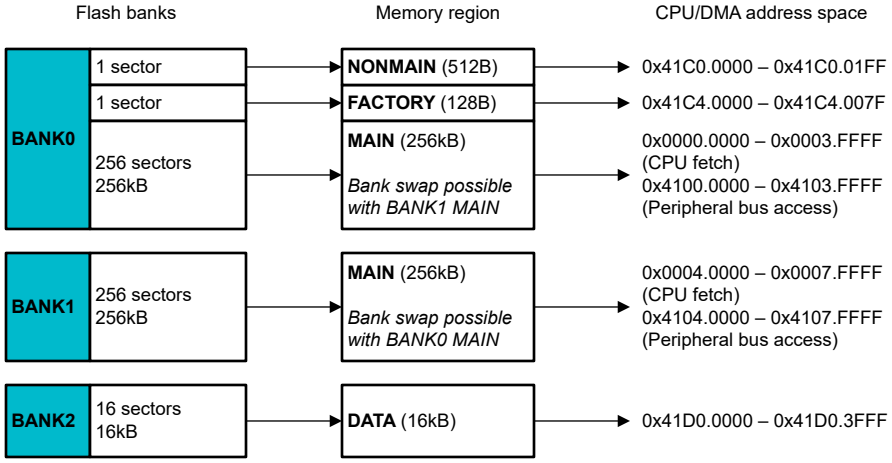



**Figure 6-3. Memory Organization Example - Multiple Bank Configuration**

## 6.3 Flash Controller

The flash controller manages all program, erase, and verification operations performed on the nonvolatile memory system. It contains memory-mapped registers in the peripheral region of the device memory map which must be configured by software to perform operations on the flash memory.

TI provides software abstraction for the flash controller as a part of the DriverLib layer of the software development kit (SDK). It is recommended to use the DriverLib abstraction layer when operating on the flash memory with software, but it is not mandatory to do so. To use the DriverLib software abstraction layer to perform operations on the flash memory, review the software development kit (SDK) documentation provided separately from this document. To directly operate on the flash memory with using low level register accesses to the flash controller, review the remainder of this section in detail.

**Note**

The FLASHCTL registers may not always be configured to default values after a reset. This may occur if the boot configuration routine (BCR) or boot strap loader (BSL) perform an operation on the flash memory during boot. When configuring the FLASHCTL registers for an operation, ensure that all registers which are relevant for the operation are correctly configured.



Copyright © 2026 Texas Instruments Incorporated

### 6.3.1 Overview of Flash Controller Commands

Operations on the flash memory are executed by configuring the CMDTYPE and CMDCTL registers for the desired command, along with any other registers which must be configured for a particular command, and writing 0x01 to the CMDEXEC register to initiate the command.

When 0x01 is set in CMDEXEC, the commanded operation begins executing. While an operation is executing, most configuration registers are blocked for writes until the operation completes. Some registers (for example, mask registers) can change state under hardware control while the operation runs to completion. The flash controller indicates completion of the commanded operation by setting the CMDDONE bit in the STATCMD register. The flash controller also sources an interrupt vector to the CPU subsystem to indicate a “DONE” status when an operation has completed.

The software sequence of setting the CMDEXEC bit and waiting for the CMDDONE response must be executed from either the device SRAM or from a different flash bank from the bank that is being operated on, as the flash controller will take control of the flash bank undergoing the operation. Reads to the flash bank that is being operated on while the flash controller is executing the command are not predictable.

The flash controller provides five basic commands for operating on the flash memory, specified in the COMMAND field of the CMDTYPE register. These commands are described in Table 6-4.

**Table 6-4. Flash Controller Commands**

| Command | Description |
|---|---|
| NOOP | No operation (default setting). |
| PROGRAM | Selects a program operation on the flash memory. |
| ERASE | Selects an erase operation on the flash memory. |
| READVERIFY | Selects a standalone read verify operation. |
| BLANKVERIFY | Selects a standalone blank verify operation. |

**Note**

It is recommended to call clear status command (CMDTYPE = 0x5) before call any flash operation.

### 6.3.2 NOOP Command

When not using the flash controller, it is best to set the COMMAND field to NOOP to prevent any unintentional operations on the flash memory in the event that the VAL bit in CMDEXEC is unintentionally set. Executing a NOOP command has no effect on the flash memory.

### 6.3.3 PROGRAM Command

The program command is used to write (program) the flash memory. Specifically, the purpose of a PROGRAM operation is to configure the flash bits in one or more flash words from the nondeterministic erased state to the deterministic programmed state. Once a byte is programmed using the PROGRAM command, the byte can not be re-programmed unless the sector is erased using the ERASE command.

All devices support single flash word programming of 64 data bits (plus 8 ECC bits on devices with ECC) at a time, with control to limit the scope of a program operation to specific bytes within a 64-bit flash word.

Some devices additionally have support for a multi-word programming mode where 2, 4, or 8 flash words can be written with a single commanded operation. Multi-word programming, when available, significantly speeds up programming when multiple words need to be programmed (for example, during production programming or firmware updates). See the device-specific data sheet to determine if multi-word programming is supported, and if so, how many flash word buffers are provided.

#### 6.3.3.1 Program Bit Masking Behavior

The flash controller provides a program verification mechanism to extend the lifetime of the flash bank. During program operations, the CMDDATAx registers are used by the flash controller as a programming bit mask to



Copyright © 2026 Texas Instruments Incorporated

indicate which specific bits in the flash word require program pulses. As a result, data which is loaded into the CMDDATAx registers before starting the program operation will be lost from the CMDDATAx registers during and/or upon completion of the program operation. If the same data is to be programmed again, the CMDDATAx registers must be re-loaded by software with the correct data values to be programmed.

#### 6.3.3.2 Programming Less Than One Flash Word

In general, the simplest way to program the flash memory is one flash word at a time (64 bits plus 8 ECC bits if ECC is present). It is possible to program the flash memory with 32-, 16-, or 8-bit (byte) resolution, but special care must be taken when doing so to ensure the following:

1. On devices with ECC, ECC bits must be handled properly to prevent inadvertent ECC errors.
2. The number of program operations applied to a given word line must be monitored to ensure that the maximum word line program limit before erase is not violated.
3. Once a byte has been programmed, the sector containing the byte must be erased before attempting to re-program the same byte.

To program less than one flash word, the CMDBYTEN register must be configured to indicate which bytes in the flash word are to be programmed before starting the program operation. Each bit in CMDBYTEN corresponds to a byte in the flash word to be programmed, including the ECC bits.

**Handling ECC**

On devices with ECC, programming 64 bits of data at a time ensures that the 8 ECC bits which correspond to the 64-bit data word are also programmed both correctly and at the same time. Doing so prevents ECC errors from occurring if the memory locations are read by the CPU or DMA after programming.

If use of ECC is planned and partial programming is required, one approach is to mask (not program) the ECC bits until all 64 bits of a flash word are programmed, at which time the ECC bits can also be programmed. Programming of ECC bits is masked by clearing BIT8 (0x100) in the CMDBYTEN register. This prevents a situation where the entire sector must be erased each time a program operation is done to re-program ECC bits to match the new 64-bit data. However, in this case, a read to a partially programmed word where the ECC bits are not yet written would result in an ECC error. To avoid an ECC error, the software must either wait until the full 64-bit flash word and the 8 ECC bits are written, or read the data from the uncorrected address space.

**Maximum Program Operations per Word Line Before Erase**

The device data sheet specifies a maximum limit on the number of program operations per word line before erasure of the sector containing the word line is required. Exceeding this maximum can result in data corruption within the word line.

If 16-bit or greater program operations are performed, and no 16-bit location is programmed more than once before a sector is erased, **the maximum limit will never be reached and thus does not need to be considered**.

If 8-bit (byte) program operations are performed, the maximum program limit per word line must be considered and not exceeded. Program operations performed on ECC locations, if done independently from other program operations, count towards the number of writes before an erase is required.

***Excuting a Program Less Than One Flash Word***

To program the flash memory with 16-bits at address 0x1000:

1. Program 16 bit at address 0x1000 (ECC is not programmed at this step):
   a. CMDATAT0 is written with DATA1, CMDDATA1 is not used.
   b. CMDBYTEEN = 0x003.
2. Program 16 bit at address 0x1002 (ECC is not programmed at this step):
   a. CMDATAT0 is written with DATA2, CMDDATA1 is not used.



Copyright © 2026 Texas Instruments Incorporated

b. CMDBYTEEN = 0x00C.

3. Program 16 bit at address 0x1004 (ECC is not programmed at this step):
   a. CMDATAT1 is written with DATA3, CMDDATA0 is not used.
   b. CMDBYTEEN = 0x030.
4. Program 16 bit at address 0x1006 (ECC is programmed at this step):
   a. CMDATAT0 is written with DATA1 and DATA2.
   b. CMDATAT1 is written with DATA3 and DATA4.
   c. CMDBYTEEN = 0x1FF.

#### 6.3.3.3 Target Data Alignment (Devices with Single Flash Word Programming Only)

For devices which only support single word programming, only the CMDDATA0, CMDDATA1, and CMDECC0 registers are used to load data to be programmed to the flash memory. CMDDATA0 is always loaded with BIT31-BIT0 of the target data, and CMDDATA1 is always loaded with BIT63-BIT32 of the target data. ECC data, if specified directly and not computed automatically, is loaded into BIT7-BIT0 of CMDECC0. No other CMDDATAx or CMDECCx registers are used, and CMDDATAINDEX is not used. If fewer than 64 data bits are being programmed, see the special handling requirements section above for programming less than one flash word.

Single-word program operations must be flash word (64-bit) aligned. This means that the target system address specified in CMDADDR must be aligned to a 0b000 boundary (for example, the 3 LSBs in CMDADDR must be zero).

#### 6.3.3.4 Target Data Alignment (Devices With Multiword Programming)

For devices that support 2-, 4-, or 8-word programming, there are two options for loading data to be programmed: direct mode or indexed mode. The programmer must select the mode that is most suitable to the application requirements.

Additional alignment rules apply when loading data into the CMDDATAx and CMDECCx registers on devices that support multiword programming, even if the multiword programming feature is not used and only single word programming is not used.

- 1-word program operations must have CMDADDR (the target system address) aligned to a 0b000 boundary (for example, the 3 LSBs in CMDADDR must be zero).
- 2-word program operations must have CMDADDR (the target system address) aligned to a 0b0000 boundary (for example, the 4 LSBs in CMDADDR must be zero).
- 4-word program operations must have CMDADDR (the target system address) aligned to a 0b0.0000 boundary (for example, the 5 LSBs in CMDADDR must be zero).
- 8-word program operations must have CMDADDR (the target system address) aligned to a 0b00.0000 boundary (for example, the 6 LSBs in CMDADDR must be zero).

**Direct Data Load**

To configure a program operation with direct data loading, the target data is loaded into the appropriate CMDDATAx registers based on the number of flash words supported by the device, the target address alignment, and the target data size. For example, if a 4-word program operation is to be initiated on a device supporting 4-word programming, the CMDDATA0-CMDDATA7 registers would be populated with the target data. If ECC is being specified directly (rather than automatically calculated by the flash controller) then the appropriate CMDECCx registers also needs to be populated with the ECC values for each data word being programmed.

**Indexed Data Load**

Rather than buffering data into all the CMDDATAx registers individually, it is possible to use only the CMDDATA0-CMDDATA1 registers in combination with an index register (CMDDATAINDEX) to indicate the flash word offset of the data being loaded. In this way, the index can be adjusted for each word loaded, and each



Copyright © 2026 Texas Instruments Incorporated

target data word can be written to the same 64-bit space (CMDDATA0-CMDDATA1). When an index is applied, the loaded data will be mapped by the hardware into the appropriate CMDDATAx register. For example, if a 4-word program operation is to be initiated on a device supporting 4-word programming, the CMDDATA1:0 registers are loaded 4 times with the target data, with CMDDATAINDEX being incremented by one before each new word is loaded into CMDDATA1:0.

### Alignment Rules

The alignment rules for each device configuration (2, 4, or 8 words) is given in the tables below with guidance on how target data must be placed with the CMDDATAx, CMDECCx, and CMDDATAINDEX registers for 1, 2, 4, or 8 word programming operations.

**Table 6-5. Data Load Alignment for Devices Supporting Programming of 2 Flash Words**

| Direct Load Registers | Indexed Load Index | 1 Word Aligned to 0b000 | 2 Words Aligned to 0b0000 | 4 Words Aligned to 0b0.0000 | 8 Words Aligned to 0b00.0000 |
|---|---|---|---|---|---|
| CMDDATA1:0 / CMDECC0 | CMDINDEX = 0 | Target data word 0 if the target address ends in 0b0000 | Target data word 0 | Not supported | Not supported |
| CMDDATA3:2 / CMDECC1 | CMDINDEX = 1 | Target data word 0 if the target address ends in 0b1000 | Target data word 1 | | |

**Table 6-6. Data Load Alignment for Devices Supporting Programming of 4 Flash Words**

| Direct Load Registers | Indexed Load Index | 1 Word Aligned to 0b000 | 2 Words Aligned to 0b0000 | 4 Words Aligned to 0b0.0000 | 8 Words Aligned to 0b00.0000 |
|---|---|---|---|---|---|
| CMDDATA1:0 / CMDECC0 | CMDINDEX= 0 | Target data word 0 if the target address ends in 0b0.0000 | Target data word 0 if the target address ends in 0b0.0000 | Target data word 0 | Not supported |
| CMDDATA3:2 / CMDECC1 | CMDINDEX= 1 | Target data word 0 if the target address ends in 0b0.1000 | Target data word 1 if the target address ends in 0b0.0000 | Target data word 1 | |
| CMDDATA5:4 / CMDECC2 | CMDINDEX= 2 | Target data word 0 if the target address ends in 0b1.0000 | Target data word 0 if the target address ends in 0b1.0000 | Target data word 2 | |
| CMDDATA7:6 / CMDECC3 | CMDINDEX= 3 | Target data word 0 if the target address ends in 0b1.1000 | Target data word 1 if the target address ends in 0b1.0000 | Target data word 3 | |

**Table 6-7. Data Load Alignment for Devices Supporting Programming of 8 Flash Words**

| Direct Load Registers | Indexed Load Index | 1 Word Aligned to 0b000 | 2 Words Aligned to 0b0000 | 4 Words Aligned to 0b0.0000 | 8 Words Aligned to 0b00.0000 |
|---|---|---|---|---|---|
| CMDDATA1:0 / CMDECC0 | CMDINDEX = 0 | Target data word 0 if the target address ends in 0b00.0000 | Target data word 0 if the target address ends in 0b00.0000 | Target data word 0 if the target address ends in 0b00.0000 | Target data word 0 |
| CMDDATA3:2 / CMDECC1 | CMDINDEX = 1 | Target data word 0 if the target address ends in 0b00.1000 | Target data word 1 if the target address ends in 0b00.0000 | Target data word 1 if the target address ends in 0b00.0000 | Target data word 1 |
| CMDDATA5:4 / CMDECC2 | CMDINDEX = 2 | Target data word 0 if the target address ends in 0b01.0000 | Target data word 0 if the target address ends in 0b01.0000 | Target data word 2 if the target address ends in 0b00.0000 | Target data word 2 |
| CMDDATA7:6 / CMDECC3 | CMDINDEX = 3 | Target data word 0 if the target address ends in 0b01.1000 | Target data word 1 if the target address ends in 0b01.0000 | Target data word 3 if the target address ends in 0b00.0000 | Target data word 3 |
| CMDDATA9:8 / CMDECC4 | CMDINDEX = 4 | Target data word 0 if the target address ends in 0b10.0000 | Target data word 0 if the target address ends in 0b10.0000 | Target data word 0 if the target address ends in 0b10.0000 | Target data word 4 |



Copyright © 2026 Texas Instruments Incorporated

**Table 6-7. Data Load Alignment for Devices Supporting Programming of 8 Flash Words (continued)**

| Direct Load Registers | Indexed Load Index | 1 Word Aligned to 0b000 | 2 Words Aligned to 0b0000 | 4 Words Aligned to 0b0.0000 | 8 Words Aligned to 0b00.0000 |
|---|---|---|---|---|---|
| CMDDATA11:10 / CMDECC5 | CMDINDEX = 5 | Target data word 0 if the target address ends in 0b10.1000 | Target data word 1 if the target address ends in 0b10.0000 | Target data word 1 if the target address ends in 0b10.0000 | Target data word 5 |
| CMDDATA13:12 / CMDECC6 | CMDINDEX = 6 | Target data word 0 if the target address ends in 0b11.0000 | Target data word 0 if the target address ends in 0b11.0000 | Target data word 2 if the target address ends in 0b10.0000 | Target data word 6 |
| CMDDATA15:14 / CMDECC7 | CMDINDEX = 7 | Target data word 0 if the target address ends in 0b11.1000 | Target data word 1 if the target address ends in 0b11.0000 | Target data word 3 if the target address ends in 0b10.0000 | Target data word 7 |

#### 6.3.3.5 Executing a PROGRAM Operation

To program the flash memory:

1. Select the command in the CMDTYPE register:
   a. Set the COMMAND field in the CMDTYPE register to PROGRAM.
   b. Set the SIZE field in the CMDTYPE register to the desired size (1, 2, 4, or 8 flash words). If a device does not support multi-word programming, select ONEWORD. If a device supports multi-word programming, and multi-word programming is desired, select the desired size which is less than or equal to the max size supported by the target device. The hardware will not check for invalid configuration of the SIZE field; software must ensure that the selection option is supported by the device. Note that SECTOR and BANK sizes are not valid sizes for PROGRAM operations. These sizes only apply to erase operations.
2. Configure the program command in the CMDCTL register:
   a. On devices with ECC, the flash controller by default will generate the needed ECC bits from the data during the PROGRAM operation. Optionally, software can override the hardware ECC code generation and manually provide the ECC code to be programmed by setting the ECCGENOVR bit in CMDCTL register.
3. Select the target programming address in the CMDADDR and CMDBYTEN register:
   a. Load the target system address into the CMDADDR register to indicate the base address from which programming will start. The target address must be a flash word address (64-bit aligned). The flash controller will translate the system address into the applicable flash region, bank ID, and bank address. If desired, after the operation completes the flash region, bank ID, and bank address can be read from the STATADDR register. In a multi-word program, STATADDR will indicate the bank ID and the final address which was programmed.
   b. If subword programming (programming of less than the full 64 or 72 bit flash word) is desired, configure the CMDBYTEN register to set the bytes within the addressed flash word which are to be programmed. Each bit in CMDBYTEN corresponds to a byte in the addressed flash word to be programmed, including the ECC byte. For example, programming of the ECC code can be masked by clearing bit 8 in CMDBYTEN while programming the data bytes of the flash word. Note that there is a maximum number of program operations allowed per word line before a sector erase must be applied (see the device specific data sheet for the maximum).
4. Load the data to program into the CMDDATAx registers:
   a. For a single flash word programming operation (64 or 72 bits depending on the presence of ECC), load the data into the CMDDATAx registers consistent with the alignment requirements (for devices which only support single-word programming, CMDDATA0 and CMDDATA1 are always used regardless of the target address).
   b. For multi-word programming (if available and selected), load data into the CMDDATAx registers consistent with the alignment rules and the size of the multi-word program operation specified in step 1.



Copyright © 2026 Texas Instruments Incorporated

c. If ECCGENOVR in the CMDCTL register was set above (disabling hardware ECC code generation), then write the ECC data in the CMDDATAECC0 register (for single word programming) and optionally additional CMDDATAECCx registers as applicable for multi-word programming.
d. Note that the CMDDATA registers are used as bit masking registers by the flash controller during the program operation; after the operation completes, the values written to these registers will have been overwritten by the flash controller.

5. Ensure the write protection scheme is configured to allow writes to the target addresses (see the write protection section of this guide for additional information on configuring write protection).
6. Execute the program operation by writing 0x1 to the CMDEXEC register.
7. Monitor for completion of the program operation:
   a. The STATCMD register can be polled to determine the status of the program operation. The CMDINPROGRESS bit will be set by hardware as soon as the command is initiated. The CMDDONE bit will be set when the operation terminates.
   b. When CMDDONE is set, the CMDPASS bit will be reset or set at the same time to indicate whether the operation completed successfully or failed. If a program was attempted on a protected region, the FAILWEPROT or FAILILLADDR bit is asserted depend on dynamic or static write protection. If the program operation cannot be completed successfully within the maximum program pulse count limit FAILVERIFY will be asserted. See the device-specific data sheet for maximum program times.
8. After completion of a program operation, the flash controller will configure several settings:
   a. All dynamic write protection registers are set to a protected state (to protect against inadvertent programming)
   b. All data registers are set to 1.
   c. All program byte enables are cleared to 0.
9. Following programming of the flash memory, it is possible that there may be stale data in the processor's cache and prefetch logic. Before reading locations which were programmed, it is recommended to first flush the cache in the CPU subsystem.

### *6.3.4 ERASE Command*

The erase command is used to erase individual sectors of flash memory (for MAIN, NONMAIN, or DATA regions) or a complete bank of flash memory (for MAIN regions only). From this erased state, bits can later be programmed to a ‘0’ state or a '1' state as desired using the PROGRAM command. It is not possible to erase with a resolution lower than one sector (1KB) with sector alignment. For devices with multiple banks, a bank erase must be executed on all banks to erase the entire MAIN region on the device.

#### 6.3.4.1 Erase Sector Masking Behavior

The flash controller provides an erase verification mechanism to extend the lifetime of the flash bank. The CMDWEPROTx registers are used as an erase mask and are manipulated by the flash controller during the execution of the erase operation. At the end of all erase operations, the CMDWEPROTx registers are set to a fully protected state to prevent against inadvertent programming and must be re-configured before attempting another program or erase operation.

#### 6.3.4.2 Executing an ERASE Operation

To erase a sector or bank of the flash memory, follow these steps:

1. Select the command in the CMDTYPE register:
   a. Set the COMMAND field in the CMDTYPE register to ERASE.
   b. Set the SIZE field in the CMDTYPE register to SECTOR or BANK. Sizes other than SECTOR or BANK (for example, ONEWORD) are not supported by the ERASE command. It is the responsibility of software to check the configuration before issuing an ERASE command.
2. Select the target erase address in the CMDADDR register:



Copyright © 2026 Texas Instruments Incorporated

a. Store the target system address into the CMDADDR register to indicate the base address of the sector or bank to be erased. When performing a bank erase with write protection enabled (such that only unprotected sectors are erased), ensure that the address written to CMDADDR is in an unprotected sector. The flash controller will translate the system address into the applicable flash region, bank ID, and bank address. If desired, after the operation completes the flash region, bank ID, and bank address can be read from the STATADDR register.

3. Ensure the write protection scheme is configured to allow writes to the target sectors (see the write protection section of this guide for additional information on configuring write protection).
4. Execute the erase operation by writing 0x1 to the CMDEXEC register.
5. Monitor for completion of the erase operation:
   a. The STATCMD register can be polled to determine the status of the erase operation. The CMDINPROGRESS bit will be set by hardware as soon as the command is initiated. The CMDDONE bit will be set when the operation terminates. When CMDDONE is set, the CMDPASS bit will be reset or set at the same time to indicate whether the operation completed successfully or failed.
   b. If an erase was attempted on a protected region, the FAILWEPROT bit is asserted.
   c. If the erase operation cannot be completed successfully within the maximum erase pulse count limit, FAILVERIFY will be asserted.
6. After completion of the erase operation, the flash controller will configure several settings to protect against inadvertent programming:
   a. All dynamic write protection registers are set to a protected state.

### *6.3.5 READVERIFY Command*

The read verify command can be used to read a flash location and compare the data to data which is preloaded into the CMDDATA registers of the flash controller. The command can be applied to a single flash word, multiple flash words (if the device supports multi-word programming), an entire sector, or an entire bank. When performing a read verify on an entire sector or bank, the data in CMDDATAx will be re-used.

#### 6.3.5.1 Executing a READVERIFY Operation

To execute a read verify command, follow these steps:

1. Select the command in the CMDTYPE register:
   a. Set the COMMAND field in the CMDTYPE register to READVERIFY.
   b. Set the SIZE field in the CMDTYPE register to the desired size.
2. Configure the read verify command in the CMDCTL register:
   a. If the desire is to manually provide ECC bits along with the data, set the ECCGENOVR bit in the CMDCTL register. If ECCGENOVR is cleared, the flash controller will generate ECC bits for comparison based on the provided compare data.
3. Select the target address to verify on the CMDADDR register:
   a. Load the target system address into the CMDADDR register to indicate the base address to be verified. The flash controller will translate the system address into the applicable flash region, bank ID, and bank address. If desired, after the operation completes the flash region, bank ID, and bank address can be read from the STATADDR register.
4. Load the data to verify into the CMDDATAx registers:
   a. For single word verification, write the data to be verified to the CMDDATA0 and CMDDATA1 registers. For multi-word verification, if available on the target device, write the data to be verified to the appropriate CMDDATAx registers beyond CMDDATA0 and CMDDATA1.
5. Configure the byte enable settings in the CMDBYTEN register:



Copyright © 2026 Texas Instruments Incorporated

a. Any CMDBYTEN bit set to logic “0” will mask the associated data byte from being compared during the execution of the READVERIFY command. This can be used to verify data less than one flash word (less than 64 bits).

6. Execute the read verify operation by writing 0x1 to the CMDEXEC register.
7. Monitor for completion of the read verify operation:
   a. The STATCMD register can be polled to determine the status of the erase operation. The CMDINPROGRESS bit will be set by hardware as soon as the command is initiated. The CMDDONE bit will be set when the operation terminates. When CMDDONE is set, the CMDPASS bit will be reset or set at the same time to indicate whether the read verification passed or failed.
   b. The FAILVERIFY bit in STATCMD will be set if any data read from the flash did not match the expected data loaded in CMDDATAx.
8. After completion of the read verify operation, the flash controller will configure several settings:
   a. All dynamic write protection registers are set to a protected state (to protect against inadvertent programming).
   b. All data registers are set to ‘1’s.
   c. All program byte enables are cleared to ‘0’s.

### *6.3.6 BLANKVERIFY Command*

The blank verify (BLANKVERIFY) command can be used by application software to verify that a flash word is blank. A blank flash word is defined as a flash word which has been successfully erased with the ERASE command and not yet programmed away from that nonerased state with the PROGRAM command.

After erase, a flash word is not in a deterministic state until it is programmed using the PROGRAM command. This means that application software can not expect that erased bits will read back as '1's' after an erase. A memory location must be successfully programmed using the PROGRAM command before a read of that memory location can be considered deterministic and used by application software.

Because it is not possible to determine if a flash word is in an erased state by simply reading the location directly (as an erased location will return nondeterministic data when read), the BLANKVERIFY command can be used to test if a flash word is in a blank state, indicating it has not yet been programmed away from an erased state.

The BLANKVERIFY command can only be applied to a single flash word at a time.

**Note**

After erasure, if programming flash with 0xFFFFFFFF, performing a blank verify at this location will also pass.

#### 6.3.6.1 Executing a BLANKVERIFY Operation

To execute a blank verify operation, follow these steps:

1. Select the command in the CMDTYPE register:
   a. Set the COMMAND field in the CMDTYPE register to BLANKVERIFY.
   b. Set the SIZE field in the CMDTYPE register to one word.
2. Select the target address to verify in the CMDADDR register:
   a. Store the target system address into the CMDADDR register to indicate the base address to be verified. The flash controller will translate the system address into the applicable flash region, bank ID, and bank address. If desired, after the operation completes the flash region, bank ID, and bank address can be read from the STATADDR register. All 8 data bytes in the specified flash word, and the corresponding ECC byte, will be checked by BLANKVERIFY.
3. Execute the blank verify operation by writing 0x1 to the CMDEXEC register.
4. Monitor for completion of the blank verify operation:



Copyright © 2026 Texas Instruments Incorporated

a. The STATCMD register can be polled to determine the status of the erase operation. The CMDINPROGRESS bit will be set by hardware as soon as the command is initiated. The CMDDONE bit will be set when the operation terminates. When CMDDONE is set, the CMDPASS bit will be reset or set at the same time to indicate whether the blank verification passed or failed.
b. The FAILVERIFY bit in STATCMD will be set if the flash location is not erased.

5. After completion of the blank verify operation, the flash controller will configure several settings:
   a. All dynamic write protection registers are set to a protected state ( to protect against inadvertent programming)
   b. All data registers are set to ‘1’s.
   c. All program byte enables are cleared to ‘0’s.

### *6.3.7 Command Diagnostics*

The flash controller updates several software-readable registers to communicate information about an initiated operation.

#### 6.3.7.1 Command Status

The STATCMD register is a read-only register which provides diagnostic information about an operation which is been initiated or completed. The CMDINPROGRESS bit indicates that an operation is currently ongoing. The CMDDONE bit indicates that an operation has completed. These bits can be polled by software to determine the state of the flash controller during operations.

#### 6.3.7.2 Address Translation

The STATADDR register is a read-only register which can be read to determine the current bank ID, region ID, and bank address which the flash controller is pointing to. These values can increment during execution of certain commands, in which case the value present after the completion of a command indicates the last address touched by the flash controller.

#### 6.3.7.3 Pulse Counts

The STATPCNT register is a read-only register which can be read to determine the current pulse count applied during a program or erase operation.

### *6.3.8 Overriding the System Address With a Bank ID, Region ID, and Bank Address*

Normally, flash controller commands are targeted to a specific flash location by loading a system memory map address into the CMDADDR register. This is the recommended way to specify the target address for a PROGRAM, ERASE, READVERIFY, or BLANKVERIFY command. In this mode of operation, the flash controller will automatically translate the system address into the corresponding bank ID, region ID, and bank address which are used to execute the command on the flash memory. Application software does not need to specify these items individually; only the system address is needed.

However, in some circumstances it can be desirable to directly specify the target flash bank, region, and address in the specified bank/region. For example, if the desire is to erase a complete bank when doing a mass erase operation on the device, application software actually does not need to have any knowledge of the system address of the bank to be erased- it only needs to specify the bank ID and region ID to the flash controller to erase the bank.

To use the flash controller to execute a command in address override mode, set the ADDRXLATEOVR bit in the CMDCTL register, and specify the bank ID, region, and bank address before executing the command. To return to system addressed mode, clear ADDRXLATEOVR. ADDRXLATEOVR is cleared by default (supporting system address operation).



Copyright © 2026 Texas Instruments Incorporated

**Example Case - Bank Erase with ADDRXLATEOVR**

To erase the MAIN region of BANK0 by specifying the bank ID and region instead of the system address, follow the steps in Section 6.3.4.2, but replace step 2 with the alternate steps given below:

1. Set the ADDRXLATEOVR bit in CMDCTL to enable address translation override mode
2. Specify BANK0 by setting the BANKSEL field to 0x1 in the CMDCTL register
3. Specify the MAIN region by setting the REGIONSEL field to 0x1 in the CMDCTL register
4. Set the CMDADDR register to 0x0000.0000

### *6.3.9 FLASHCTL Events*

The flash controller contains one event publisher and no event subscribers. One event publisher (CPU_INT) manages FLASHCTL interrupt requests (IRQs) to the CPU subsystem through a static event route.

Table 6-8 summarizes the FLASHCTL events.

**Table 6-8. FLASHCTL Events**

| Event | Type | Source | Destination | Route | Configuration | Functionality |
|---|---|---|---|---|---|---|
| CPU Interrupt Event | Publisher | FLASHCTL | CPU Subsystem | Static route | CPU_INT registers | Fixed interrupt route from FLASHCTL to CPU |

#### 6.3.9.1 CPU Interrupt Event Publisher

The FLASHCTL module provides one interrupt source which can be configured to source a CPU interrupt event. The FLASHCTL interrupt conditions are given in Table 6-9.

**Table 6-9. FLASHCTL CPU Interrupt Conditions**

| Index (IIDX) | Name | Description |
|---|---|---|
| 0 | DONE | Indicating that the FLASHCTL operation has completed |

The CPU interrupt event configuration is managed with the CPU_INT event management registers. See Section 7.2.5 for guidance on configuring these registers.

## 6.4 Write Protection

The flash controller provides two write protection mechanisms (one static, one dynamic) which are applied in parallel (logical OR) to protect user-specified sectors during any attempted program or erase operation. If a program or erase operation is issued to a write protected flash sector, the operation will terminate with a FAILWEPROT error reported in the STATCMD register.

### *6.4.1 Write Protection Resolution*

The write protection resolution for both the static and dynamic write protection mechanisms is dependent on the flash bank and memory region which is being protected.

**Table 6-10. Write Protection Resolution by Region**

| Flash Bank | Memory Region | Write Protection Resolution |
|---|---|---|
| 0 | NONMAIN | 512B (entire region) |
| | MAIN | 1KB (1 sector) for first 32KB (32 sectors)<br>8KB (8 sectors) for remaining sectors |
| 1-4 | MAIN | 8KB (8 sectors) |
| | DATA | 1KB (1 sector) |

### *6.4.2 Static Write Protection*

The static write protection scheme is configured and latched during device boot by the immutable ROM boot code before the main application in flash can execute. After a flash sector is configured to be write protected through the static write protection mechanism, software cannot remove the protection at runtime without a



Copyright © 2026 Texas Instruments Incorporated

reboot. Thus, sectors protected by static write protection can be thought of as immutable after boot. This type of protection is useful for protecting a custom bootloader in a dual-image application, or for extending the secure root of trust from the ROM boot code into a portion of the main flash region to enable a secure boot manager with locked public keys.

The static write protection scheme is configured by programming the appropriate bits in the NONMAIN flash region, which is read by the boot code before the main application starts. The NONMAIN flash sector can be statically write protected, resulting in a system in which the static write protection scheme is fully permanent and cannot be modified. If the NONMAIN sector is configured to be statically protected, along with any other flash sectors, all statically protected flash sectors can be functionally viewed as read only memory which cannot be updated by any means.

Static write protection can be configured for all sectors of flash memory across all banks present on a device. Program or erase static write protected area will cause illegal address failure, and the FAILILLADDR bit in STATCMD will be set. Instructions for configuring the static write protection, along with the rest of the NONMAIN region, are given in Section 1.4.

### *6.4.3 Dynamic Write Protection*

The dynamic write protection scheme is intended to be configured at runtime by software. This scheme provides a simple way for application software to specify sectors to protect from modification by any program or erase operations that are issued with the flash controller. Unlike the static write protection mechanism, the dynamic mechanism is not lockable and thus does not provide any level of data security.

There are two primary uses for dynamic write protection. First, it provides an extra level of robustness against unintentional program or erase of specified sectors in applications that involve in-system programming for either firmware updates or EEPROM emulation. Second, it provides a way to simplify a situation where a bank erase is desired but a small number of sectors should not be erased when a bank erase command is issued. One example would be an application running on a single-bank device in which most of the MAIN region sectors are used to store the executable image, but a few sectors are used to store device-specific data that should not be erased during a firmware update. In that case, the sectors containing the device-specific data can be protected with dynamic write protection, and a bank erase command can be issued to erase all other sectors. This has the benefit that the majority of the MAIN region can be erased with a single command (a bank erase) rather than individual sector-by-sector commands, which would have a longer overall erase time and use more energy.

The dynamic write protection scheme is configured by setting up the CMDWEPROTx registers in the flash controller. The CMDWEPROTx registers cover one flash bank at a time. This means that these registers must be configured with knowledge of which flash bank an attempted program or erase operation will be applied to. Note that the CMDWEPROTx registers are reset to a protected state at the end of all program and erase operations. These registers must be re-configured by software before a new operation is initiated. Program or erase dynamic write protected area will cause write/erase protction failure, and the FAILWEPROT bit in STATCMD will be set.

#### 6.4.3.1 Configuring Protection for the MAIN Region

The CMDWEPROTA register is used to configure the dynamic write protection for the first 32 sectors (32KB) of the MAIN region in BANK0. Each bit corresponds to one sector of the MAIN region, starting from the beginning of the MAIN region. CMDWEPROTA is only applicable to operations on the lower 32 sectors of the MAIN region of BANK0 (sectors 0-31). It is not used during program/erase operations applied to other sectors.

The CMDWEPROTB register is also used to configure the dynamic write protection for the MAIN region. There are two modes in which CMDWEPROTB is applied, depending on whether the program/erase operation is being applied to BANK0 or BANK1-4. In the case of a program/erase operation on BANK0, CMDWEPROTB protects sectors 32-255 in 8-sector increments (1 bit per 8KB), starting from BIT4. BIT0-BIT3 in CMDWEPROTB are ignored. The lower 32 sectors (sectors 0-31 of BANK0) are protected by CMDWEPROTA. In the case of a program/erase operation on BANK1-4, CMDWEPROTB protects sectors 0-255 in 8-sector increments (1 bit per 8 sectors), starting from BIT0 in CMDWEPROTB.

The CMDWEPROTC register configures the dynamic write protection for the MAIN region. For BANK0-4, CMDWEPROTC protects sectors 256-511 in 8-sector increments (1 bit per 8 sectors).



Copyright © 2026 Texas Instruments Incorporated

#### 6.4.3.2 Configuring Protection for the NONMAIN Region

The CMDWEPROTNM register protects the NONMAIN region from program and erase. One protection bit is provided per sector. Devices only have one NONMAIN sector and it will be in BANK0.

## 6.5 Read Interface

The read interface provides the read path to the CPU subsystem (for instruction/data fetch), the read path to the peripheral bus (for use by the DMA controller or CPU), main bank address swapping, and detection and reporting of ECC SEC or DED errors.

### *6.5.1 Bank Address Swapping*

Devices that contain more than one bank support swapping of the MAIN regions of the banks within the address space. This mechanism enables two versions of application firmware to be programmed into the device without the firmware needing to know which physical bank it exists in.

Table 6-11 gives an example of the mapping before and after a bank swap is requested for a device with a 512KB main flash split across 2 banks (256KB each).

**Table 6-11. Bank Address Swap Translation**

| Bank and Region | Address Space Before Swap | Address Space After Swap |
|---|---|---|
| BANK0 MAIN | 0x0000.0000 – 0x0003.FFFF | 0x0004.0000 – 0x0007.FFFF |
| BANK1 MAIN | 0x0004.0000 – 0x0007.FFFF | 0x0000.0000 – 0x0003.FFFF |

After a device reset, the MAIN region of the lower bank is always mapped to the lowest address space. The application software is responsible for determining if a bank address swap is to be applied. The bank address swap control is contained within the SYSCTL module; see the SYSCTL chapter for register and bit definitions. During an address swap, the application software must meet the following constraints:

1. The software must disable interrupts before issuing the bank swap command.
2. The software must poll the bank swap status after issuing the bank swap command before proceeding with execution.
3. If the swap command and poll status routines are executing from flash, they must exist at the exact same location in both banks such that execution resumes from where it left off after the bank swap. This restriction does not apply if the bank swap and status polling is done from SRAM and not from flash memory.

### *6.5.2 ECC Error Handling*

The read interface corrects single bit errors in a 64-bit flash word (SEC) and detects dual-bit errors in a 64-bit flash word (DED). ECC checks are ignored for an “all-1’s” scenario or an “all-0’s” scenario.

#### 6.5.2.1 Single bit (correctable) errors

Single bit errors are corrected automatically before the requested data is returned to either the CPU subsystem or the peripheral bus (DMA). Errors occurring as a result of a debug access are not reported.

#### 6.5.2.2 Dual bit (uncorrectable) errors

Dual bit errors are detectable but not correctable and are indicated to SYSCTL. SYSCTL can be configured to generate either a nonmaskable interrupt (NMI) or a reset, as uncorrectable errors can be fatal. Errors occurring as a result of a debug access are not reported.



Copyright © 2026 Texas Instruments Incorporated

## 6.6 FLASHCTL Registers

Table 6-12 lists the memory-mapped registers for the FLASHCTL registers. All register offset addresses not listed in Table 6-12 should be considered as reserved locations and the register contents should not be modified.

**Table 6-12. FLASHCTL Registers**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 1020h | IIDX | Interrupt Index Register | Section 6.6.1 |
| 1028h | IMASK | Interrupt Mask Register | Section 6.6.2 |
| 1030h | RIS | Raw Interrupt Status Register | Section 6.6.3 |
| 1038h | MIS | Masked Interrupt Status Register | Section 6.6.4 |
| 1040h | ISET | Interrupt Set Register | Section 6.6.5 |
| 1048h | ICLR | Interrupt Clear Register | Section 6.6.6 |
| 10E0h | EVT_MODE | Event Mode | Section 6.6.7 |
| 10FCh | DESC | Hardware Version Description Register | Section 6.6.8 |
| 1100h | CMDEXEC | Command Execute Register | Section 6.6.9 |
| 1104h | CMDTYPE | Command Type Register | Section 6.6.10 |
| 1108h | CMDCTL | Command Control Register | Section 6.6.11 |
| 1120h | CMDADDR | Command Address Register | Section 6.6.12 |
| 1124h | CMDBYTEN | Command Program Byte Enable Register | Section 6.6.13 |
| 1130h | CMDDATA0 | Command Data Register 0 | Section 6.6.14 |
| 1134h | CMDDATA1 | Command Data Register 1 | Section 6.6.15 |
| 11B0h | CMDDATAECC0 | Command Data Register ECC 0 | Section 6.6.16 |
| 11D0h | CMDWEPROTA | Command Write Erase Protect A Register | Section 6.6.17 |
| 11D4h | CMDWEPROTB | Command Write Erase Protect B Register | Section 6.6.18 |
| 1210h | CMDWEPROTNM | Command Write Erase Protect Non-Main Register | Section 6.6.19 |
| 13B0h | CFGCMD | Command Configuration Register | Section 6.6.20 |
| 13B4h | CFGPCNT | Pulse Counter Configuration Register | Section 6.6.21 |
| 13D0h | STATCMD | Command Status Register | Section 6.6.22 |
| 13D8h | STATPCNT | Pulse Count Status Register | Section 6.6.23 |
| 13DCh | STATMODE | Mode Status Register | Section 6.6.24 |
| 13F0h | GBLINFO0 | Global Information Register 0 | Section 6.6.25 |
| 13F4h | GBLINFO1 | Global Information Register 1 | Section 6.6.26 |
| 13F8h | GBLINFO2 | Global Information Register 2 | Section 6.6.27 |
| 1400h | BANK0INFO0 | Bank Information Register 0 for Bank 0 | Section 6.6.28 |
| 1404h | BANK0INFO1 | Bank Information Register 1 for Bank 0 | Section 6.6.29 |
| 1410h | BANK1INFO0 | Bank Information Register 0 for Bank 1 | Section 6.6.30 |
| 1414h | BANK1INFO1 | Bank Information Register 1 for Bank 1 | Section 6.6.31 |
| 1420h | BANK2INFO0 | Bank Information Register 0 for Bank 2 | Section 6.6.32 |
| 1424h | BANK2INFO1 | Bank Information Register 1 for Bank 2 | Section 6.6.33 |

Complex bit access types are encoded to fit into small table cells. Table 6-13 shows the codes that are used for access types in this section.

**Table 6-13. FLASHCTL Access Type Codes**

| Access Type | Code | Description |
|---|---|---|
| Read Type | | |
| R | R | Read |
| Write Type | | |



Copyright © 2026 Texas Instruments Incorporated

**Table 6-13. FLASHCTL Access Type Codes (continued)**

| Access Type | Code | Description |
|---|---|---|
| W | W | Write |
| Reset or Default Value | | |
| *-n* | | Value after reset or the default value |



Copyright © 2026 Texas Instruments Incorporated

### 6.6.1 IIDX Register (Offset = 1020h) [Reset = 00000000h]

IIDX is shown in Figure 6-4 and described in Table 6-14.

Return to the Summary Table.

Interrupt Index Register:
The IIDX register provides the highest priority enabled interrupt index.
PSD compliant register.
Note that it is not recommended to use this register if the system clock is running at a slower clock frequency than the flash wrapper clock. If this is the case, then reading this register may fail to update the RIS register correctly. The MIS register should be read directly, and a write to ICLR should be used to clear interrupts when this clock relationship is present.

**Figure 6-4. IIDX Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | STAT |
| R-0h | | | | | | | R-0h |

**Table 6-14. IIDX Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | Reserved |
| 0 | STAT | R | 0h | Indicates which interrupt has fired. 0x0 means no event pending. The priority order is fixed. On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flags in the RIS and MIS are cleared as well. After a read from the CPU (not from the debug interface), the register must be updated with the next highest priority interrupt.<br>0h = No Interrupt Pending<br>1h = DONE Interrupt Pending |



Copyright © 2026 Texas Instruments Incorporated

### 6.6.2 IMASK Register (Offset = 1028h) [Reset = 00000000h]

IMASK is shown in Figure 6-5 and described in Table 6-15.

Return to the Summary Table.

Interrupt Mask Register: The IMASK register holds the current interrupt mask settings. Masked interrupts are read in the MIS register. PSD compliant register.

**Figure 6-5. IMASK Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | DONE |
| R-0h | | | | | | | R/W-0h |

**Table 6-15. IMASK Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | Reserved |
| 0 | DONE | R/W | 0h | Interrupt mask for DONE: 0: Interrupt is disabled in MIS register 1: Interrupt is enabled in MIS register<br>0h = Interrupt is masked out<br>1h = Interrupt will request an interrupt service routine and corresponding bit in MIS will be set |



Copyright © 2026 Texas Instruments Incorporated

### 6.6.3 RIS Register (Offset = 1030h) [Reset = 00000000h]

RIS is shown in Figure 6-6 and described in Table 6-16.

Return to the Summary Table.

Raw Interrupt Status Register: The RIS register reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing a 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled. A flag can be set by software by writing a 1 to the ISET register. Reading the IIDX register will also clear the corresponding bit in RIS. PSD compliant register.

**Figure 6-6. RIS Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | DONE |
| R-0h | | | | | | | R-0h |

**Table 6-16. RIS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | Reserved |
| 0 | DONE | R | 0h | Flash wrapper operation completed.<br>This interrupt bit is set by firmware or the corresponding bit in the ISET register.<br>It is cleared by the corresponding bit in in the ICLR register or reading the IIDX register when this interrupt is the highest priority.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

### 6.6.4 MIS Register (Offset = 1038h) [Reset = 00000000h]

MIS is shown in Figure 6-7 and described in Table 6-17.

Return to the Summary Table.

Masked Interrupt Status Register: The MIS register is a bit-wise AND of the contents of the IMASK and RIS registers. This is kept mainly for ARM compatibility, and has limited use since the highest priority interrupt index is returned via the IIDX register. PSD compliant register.

**Figure 6-7. MIS Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | DONE |
| R-0h | | | | | | | R-0h |

**Table 6-17. MIS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | Reserved |
| 0 | DONE | R | 0h | Flash wrapper operation completed.<br>This masked interrupt bit reflects the bitwise AND of the corresponding RIS and IMASK bits.<br>0h = Masked interrupt did not occur<br>1h = Masked interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

### 6.6.5 ISET Register (Offset = 1040h) [Reset = 00000000h]

ISET is shown in Figure 6-8 and described in Table 6-18.

Return to the Summary Table.

Interrupt Set Register: The ISET register allows software to write a 1 to set corresponding interrupt. Safety: This meets a safety requirement to allow software diagnostics to trigger interrupts. PSD compliant register.

**Figure 6-8. ISET Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | DONE |
| R-0h | | | | | | | W-0h |

**Table 6-18. ISET Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | Reserved |
| 0 | DONE | W | 0h | 0: No effect 1: Set the DONE interrupt in the RIS register<br>0h = Writing a 0 has no effect<br>1h = Set RIS bit |



Copyright © 2026 Texas Instruments Incorporated

### 6.6.6 ICLR Register (Offset = 1048h) [Reset = 00000000h]

ICLR is shown in Figure 6-9 and described in Table 6-19.

Return to the Summary Table.

Interrupt Clear Register. The ICLR register allows allows software to write a 1 to clear corresponding interrupt. PSD compliant register.

**Figure 6-9. ICLR Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | DONE |
| R-0h | | | | | | | W-0h |

**Table 6-19. ICLR Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | Reserved |
| 0 | DONE | W | 0h | 0: No effect 1: Clear the DONE interrupt in the RIS register<br>0h = Writing a 0 has no effect<br>1h = Clear RIS bit |



Copyright © 2026 Texas Instruments Incorporated

### 6.6.7 EVT_MODE Register (Offset = 10E0h) [Reset = 00000000h]

EVT_MODE is shown in Figure 6-10 and described in Table 6-20.

Return to the Summary Table.

Event mode register. It is used to select whether each line is disabled, in software mode (software clears the RIS) or in hardware mode (hardware clears the RIS).

**Figure 6-10. EVT_MODE Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | INT0_CFG | |
| R-0h | | | | | | R-0h | |

**Table 6-20. EVT_MODE Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | Reserved |
| 1-0 | INT0_CFG | R | 1h | Event line mode select for peripheral event<br>0h = The interrupt or event line is disabled.<br>1h = The interrupt or event line is in software mode. Software must clear the RIS.<br>2h = The interrupt or event line is in hardware mode. Hardware should clear the RIS. |



Copyright © 2026 Texas Instruments Incorporated

### 6.6.8 DESC Register (Offset = 10FCh) [Reset = 00000000h]

DESC is shown in Figure 6-11 and described in Table 6-21.

Return to the Summary Table.

Hardware Version Description Register:
This register identifies the flash wrapper hardware version and feature set used.

**Figure 6-11. DESC Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MODULEID | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| FEATUREVER | | | | INSTNUM | | | | MAJREV | | | | MINREV | | | |
| R-0h | | | | R-0h | | | | R-0h | | | | R-0h | | | |

**Table 6-21. DESC Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | MODULEID | R | 0h | Module ID<br>0h = Smallest value<br>FFFFh = Highest possible value |
| 15-12 | FEATUREVER | R | 0h | Feature set<br>0h = Minimum Value<br>Fh = Maximum Value |
| 11-8 | INSTNUM | R | 0h | Instance number<br>0h = Smallest value<br>Fh = Highest possible value |
| 7-4 | MAJREV | R | 0h | Major Revision<br>0h = Smallest value<br>Fh = Highest possible value |
| 3-0 | MINREV | R | 0h | Minor Revision<br>0h = Smallest value<br>Fh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 6.6.9 CMDEXEC Register (Offset = 1100h) [Reset = 00000000h]

CMDEXEC is shown in Figure 6-12 and described in Table 6-22.

Return to the Summary Table.

Command Execute Register:
Initiates execution of the command specified in the CMDTYPE register.
This register is blocked for writes after being written to 1 and prior to
STATCMD.DONE being set by the flash wrapper hardware.
flash wrapper hardware clears this register after the processing of the command
has completed.

**Figure 6-12. CMDEXEC Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | | | | VAL |
| R-0h | | | | | | | | | | | | | | | R/W-0h |

**Table 6-22. CMDEXEC Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | Reserved |
| 0 | VAL | R/W | 0h | Command Execute value Initiates execution of the command specified in the CMDTYPE register.<br>0h = Command will not execute or is not executing in flash wrapper<br>1h = Command will execute or is executing in flash wrapper |



Copyright © 2026 Texas Instruments Incorporated

### 6.6.10 CMDTYPE Register (Offset = 1104h) [Reset = 00000000h]

CMDTYPE is shown in Figure 6-13 and described in Table 6-23.

Return to the Summary Table.

Command Type Register
This register specifies the type of command to be executed by the flash wrapper hardware.
This register is blocked for writes after CMDEXEC is written to a 1 and prior to STATCMD.DONE being set by the hardware to indicate that command execution has completed.

**Figure 6-13. CMDTYPE Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | SIZE | | | RESERVED | COMMAND | | |
| R-0h | R/W-0h | | | R-0h | R/W-0h | | |

**Table 6-23. CMDTYPE Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-7 | RESERVED | R | 0h | Reserved |
| 6-4 | SIZE | R/W | 0h | Command size<br>0h = Operate on 1 flash word<br>4h = Operate on a flash sector<br>5h = Operate on an entire flash bank |
| 3 | RESERVED | R | 0h | Reserved |
| 2-0 | COMMAND | R/W | 0h | Command type<br>0h = No Operation<br>1h = Program<br>2h = Erase<br>3h = Read Verify - Perform a standalone read verify operation.<br>5h = Clear Status - Clear status bits in FW_SMSTAT only.<br>6h = Blank Verify - Check whether a flash word is in the erased state. This command may only be used with CMDTYPE.SIZE = ONEWORD |



Copyright © 2026 Texas Instruments Incorporated

### 6.6.11 CMDCTL Register (Offset = 1108h) [Reset = 00000000h]

CMDCTL is shown in Figure 6-14 and described in Table 6-24.

Return to the Summary Table.

Command Control Register This register configures specific capabilities of the state machine for related to the execution of a command. This register is blocked for writes after CMDEXEC is written to a 1 and prior to STATCMD.DONE being set by the hardware to indicate that command execution has completed.

**Figure 6-14. CMDCTL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | DATAVEREN | RESERVED | | | ECCGENOVR | ADDRXLATEOVR |
| R-0h | | R/W-0h | R-0h | | | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | RESERVED | REGIONSEL | | | | RESERVED |
| R-0h | | R-0h | R/W-0h | | | | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | BANKSEL | MODESEL | | | |
| R-0h | | | R/W-0h | R/W-0h | | | |

**Table 6-24. CMDCTL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-22 | RESERVED | R | 0h | Reserved |
| 21 | DATAVEREN | R/W | 0h | Enable invalid data verify. This checks for 0->1 transitions in the memory when a program operation is initiated. If such a transition is found, the program will fail with an error without doing any programming.<br>0h = Disable<br>1h = Enable |
| 20-18 | RESERVED | R | 0h | |
| 17 | ECCGENOVR | R/W | 0h | Override hardware generation of ECC data for program. Use data written to CMDDATAECC*.<br>0h = Do not override<br>1h = Override |
| 16 | ADDRXLATEOVR | R/W | 0h | Override hardware address translation of address in CMDADDR from a system address to a bank address and bank ID. Use data written to CMDADDR directly as the bank address. Use the value written to CMDCTL.BANKSEL directly as the bank ID. Use the value written to CMDCTL.REGIONSEL directly as the region ID.<br>0h = Do not override<br>1h = Override |
| 15-14 | RESERVED | R | 0h | |
| 13 | RESERVED | R | 0h | Reserved |
| 12-9 | REGIONSEL | R/W | 0h | Bank Region A specific region ID can be written to this field to indicate to which region an operation should be applied if CMDCTL.ADDRXLATEOVR is set.<br>1h = Main Region<br>2h = Non-Main Region<br>4h = Trim Region<br>8h = Engr Region |
| 8-5 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

**Table 6-24. CMDCTL Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 4 | BANKSEL | R/W | 0h | Bank Select A specific Bank ID can be written to this field to indicate to which bank an operation should be applied if CMDCTL.ADDRXLATEOVR is set.<br>1h = Bank 0<br>2h = Bank 1<br>4h = Bank 2<br>8h = Bank 3<br>10h = Bank 4 |
| 3-0 | MODESEL | R/W | 0h | Mode This field is only used for the Mode Change command type. Otherwise, bank and pump modes are set automaticlly via the NW hardware.<br>0h = Read Mode<br>2h = Read Margin 0 Mode<br>4h = Read Margin 1 Mode<br>6h = Read Margin 0B Mode<br>7h = Read Margin 1B Mode<br>9h = Program Verify Mode |



Copyright © 2026 Texas Instruments Incorporated

### 6.6.12 CMDADDR Register (Offset = 1120h) [Reset = 00000000h]

CMDADDR is shown in Figure 6-15 and described in Table 6-25.

Return to the Summary Table.

Command Address Register:
This register forms the target address of a command. The use cases are as follows:
1) For single-word program, this address indicates the flash bank word to be programmed.
2) For multi-word program, this address indicates the first flash bank address for the program. The address will be incremented for further words.
3) For sector erase, this address indicates the sector to be erased.
4) For bank erase, the address indicates the bank to be erased.
5) For read verify, the address indications follow program/erase listed above.
Note the address written to this register will be submitted for translation to the flash wrapper address translation interface, and the translated address will be used to access the bank. However, if the CMDCTL.ADDRXLATEOVR bit is set, then the address written to this register will be used directly as the bank address.
This register is blocked for writes after a 1 is written to the CMDEXEC register and prior to STATCMD.DONE being set by the flash wrapper hardware.

**Figure 6-15. CMDADDR Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| VAL | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 6-25. CMDADDR Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | VAL | R/W | 0h | Address value<br>0h = Minimum value of [VAL]<br>FFFFFFFFh = Maximum value of [VAL] |



Copyright © 2026 Texas Instruments Incorporated

### 6.6.13 CMDBYTEN Register (Offset = 1124h) [Reset = 00000000h]

CMDBYTEN is shown in Figure 6-16 and described in Table 6-26.

Return to the Summary Table.

Command Program Byte Enable Register:
This register forms a per-byte enable for programming data. For data bytes to be programmed, a 1 must be written to the corresponding bit in this register. Normally, all bits are written to 1, allowing program of full flash words. However, leaving some bits 0 allows programming of 8-bit, 16-bit, 32-bit or 64-bit portions of a flash word.
In addtion, the read verify command will ignore data bytes read from the flash in its comparison if the corresponding CMDBYTEN bit is 0.
ECC data bytes are protected by the 1-2 MSB bits in this register, depending on the presence of ECC and the flash word data width.
This register is blocked for writes after a 1 is written to the CMDEXEC register and prior to STATCMD.DONE being set by the flash wrapper hardware.
This register is written to all 0 after the completion of all flash wrapper commands.

**Figure 6-16. CMDBYTEN Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | RESERVED | | | | | | | | | | VAL | | | | | | | |
| R-0h | | | | | | | | | | | | | | R-0h | | | | | | | | | | R/W-0h | | | | | | | |

**Table 6-26. CMDBYTEN Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-18 | RESERVED | R | 0h | Reserved |
| 17-8 | RESERVED | R | 0h | |
| 7-0 | VAL | R/W | 0h | Command Byte Enable value. A 1-bit per flash word byte value is placed in this register.<br>0h = Minimum value of [VAL]<br>0003FFFFh = Maximum value of [VAL] |



Copyright © 2026 Texas Instruments Incorporated

### 6.6.14 CMDDATA0 Register (Offset = 1130h) [Reset = 00000000h]

CMDDATA0 is shown in Figure 6-17 and described in Table 6-27.

Return to the Summary Table.

Command Data Register 0
This register forms the data for a command.
For DATAWIDTH == 128: This register represents bits 31:0 of flash word data register 0.
For DATAWIDTH == 64: This register represents bits 31:0 of flash word data register 0.
This register is blocked for writes after a 1 is written to the CMDEXEC
register and prior to STATCMD.DONE being set by the flash wrapper
hardware.
This register is used to aggregate masking for bits that do not
require additional program pulses during program operations, and will be
written to all 1 after the completion of all flash wrapper commands.
Use cases for the CMDDATA* registers are as follows:
1) Program - These registers contain the data to be programmed.
2) Erase - These registers are not used.
3) Read Verify - These registers contain data to be verified.

**Figure 6-17. CMDDATA0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| VAL | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 6-27. CMDDATA0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | VAL | R/W | FFFFFFFFh | A 32-bit data value is placed in this field.<br>0h = Minimum value of [VAL]<br>FFFFFFFFh = Maximum value of [VAL] |



Copyright © 2026 Texas Instruments Incorporated

### 6.6.15 CMDDATA1 Register (Offset = 1134h) [Reset = 00000000h]

CMDDATA1 is shown in Figure 6-18 and described in Table 6-28.

Return to the Summary Table.

Command Data Register 1
This register forms the data for a command.
For DATAWIDTH == 128: This register represents bits 63:32 of flash word data register 0.
For DATAWIDTH == 64: This register represents bits 63:32 of flash word data register 0.
This register is blocked for writes after a 1 is written to the CMDEXEC
register and prior to CMDSTAT.DONE being set by the flash wrapper
hardware.
This register is used to aggregate masking for bits that do not
require additional program pulses during program operations, and will be
written to all 1 after the completion of all flash wrapper commands.
Use cases for the CMDDATA* registers are as follows:
1) Program - These registers contain the data to be programmed.
2) Erase - These registers are not used.
3) Read Verify - These registers contain data to be verified.

**Figure 6-18. CMDDATA1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| VAL | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 6-28. CMDDATA1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | VAL | R/W | FFFFFFFFh | A 32-bit data value is placed in this field.<br>0h = Minimum value of [VAL]<br>FFFFFFFFh = Maximum value of [VAL] |



Copyright © 2026 Texas Instruments Incorporated

### 6.6.16 CMDDATAECC0 Register (Offset = 11B0h) [Reset = 00000000h]

CMDDATAECC0 is shown in Figure 6-19 and described in Table 6-29.

Return to the Summary Table.

Command Data Register 0
This register forms the ECC portion of the data for a command. This ECC data in this register covers flash data register 0.
The hardware ECC generation in flash wrapper can be overridden and ECC data developed elsewhere can be used. ECC data is placed in this register.
This register is blocked for writes after a 1 is written to the CMDEXEC register and prior to STATCMD.DONE being set by the flash wrapper hardware.
Use cases for the CMDDATA* registers are as follows:
1) Program - These registers contain the data to be programmed.
2) Erase - These registers are not used.
3) Read Verify - These registers contain data to be verified.
This register is used to aggregate masking for bits that do not require additional program pulses during program operations, and will be written to all 1 after the completion of all flash wrapper commands.

**Figure 6-19. CMDDATAECC0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | VAL1 | | | | | | | | VAL0 | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | R/W-0h | | | | | | | | R/W-0h | | | | | | | |

**Table 6-29. CMDDATAECC0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | Reserved |
| 15-8 | VAL1 | R/W | FFh | ECC data for bits 127:64 of the data is placed here.<br>0h = Minimum value<br>FFh = Maximum value |
| 7-0 | VAL0 | R/W | FFh | ECC data for bits 63:0 of the data is placed here.<br>0h = Minimum value<br>FFh = Maximum value |



Copyright © 2026 Texas Instruments Incorporated

### 6.6.17 CMDWEPROTA Register (Offset = 11D0h) [Reset = 00000000h]

CMDWEPROTA is shown in Figure 6-20 and described in Table 6-30.

Return to the Summary Table.

Command WriteErase Protect A Register
NOTE: This register is not implemented in some products.
Please consult the product-specific datasheet.
This register allows the first 32 sectors of the main region to be protected from program or erase, with 1 bit protecting each sector. If the main region size is smaller than 32 sectors, then this register provides protection for the whole region.
This register is blocked for writes after a 1 is written to the CMDEXEC register and prior to STATCMD.DONE being set by the flash wrapper hardware.
In addition, this register is used to aggregate masking for sectors that do not require additional erase pulses during bank erase operations, and will be written to all 1 after the completion of all flash wrapper commands.

**Figure 6-20. CMDWEPROTA Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| VAL | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 6-30. CMDWEPROTA Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | VAL | R/W | FFFFFFFFh | Each bit protects 1 sector. bit [0]: When 1, sector 0 of the flash memory will be protected from program and erase. bit [1]: When 1, sector 1 of the flash memory will be protected from program and erase. : : bit [31]: When 1, sector 31 of the flash memory will be protected from program and erase.<br>0h = Minimum value of [VAL]<br>FFFFFFFFh = Maximum value of [VAL] |



Copyright © 2026 Texas Instruments Incorporated

### 6.6.18 CMDWEPROTB Register (Offset = 11D4h) [Reset = 00000000h]

CMDWEPROTB is shown in Figure 6-21 and described in Table 6-31.

Return to the Summary Table.

Command WriteErase Protect B Register
This register allows main region sectors to be protected from program and erase. Each bit corresponds to a group of 8 sectors.
There are 3 cases for how these protect bits are applied:
1. Single-bank system:
In the case where only a single flash bank is present,
the first 32 sectors are protected via the CMDWEPROTA register. Thus, the protection give by the bits in CMDWEPROTB begin with sector 32.
2. Multi-bank system, Bank 0:
When multiple flash banks are present, the first
32 sectors of bank 0 are protected via the CMDWEPROTA register. Thus, only bits 4 and above of CMDWEPROTB would be applicable to bank 0. The protection of bit 4 and above would begin at sector 32. Bits 3:0
of WEPROTB are ignored for bank 0.
3. Multi-bank system, Banks 1-N:
For banks other than bank 0 in a multi-bank system, CMDWEPROTA has no effect, so the bits in CMDWEPROTB will protect these banks starting from sector 0.
This register is blocked for writes after a 1 is written to the CMDEXEC register and prior to STATCMD.DONE being set by the flash wrapper hardware.
In addition, this register is used to aggregate masking for sectors that do not require additional erase pulses during bank erase operations, and will be written to all 1 after the completion of all flash wrapper commands.

**Figure 6-21. CMDWEPROTB Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| VAL | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 6-31. CMDWEPROTB Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | VAL | R/W | FFFFFFFFh | Each bit protects a group of 8 sectors. When a bit is 1, the associated 8 sectors in the flash will be protected from program and erase. A maximum of 256 sectors can be protected with this register.<br>0h = Minimum value of [VAL]<br>FFFFFFFFh = Maximum value of [VAL] |



Copyright © 2026 Texas Instruments Incorporated

### 6.6.19 CMDWEPROTNM Register (Offset = 1210h) [Reset = 00000000h]

CMDWEPROTNM is shown in Figure 6-22 and described in Table 6-32.

Return to the Summary Table.

Command WriteErase Protect Non-Main
Register
This register allows non-main region region sectors to be protected from program and erase. Each bit corresponds to 1 sector.
This register is blocked for writes after a 1 is written to the CMDEXEC register and prior to STATCMD.DONE being set by the flash wrapper hardware.
In addition, this register is used to aggregate masking for sectors that do not require additional erase pulses during bank erase operations, and will be written to all 1 after the completion of all flash wrapper commands.

**Figure 6-22. CMDWEPROTNM Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| VAL | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 6-32. CMDWEPROTNM Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | VAL | R/W | FFFFFFFFh | Each bit protects 1 sector. bit [0]: When 1, sector 0 of the non-main region will be protected from program and erase. bit [1]: When 1, sector 1 of the non-main region will be protected from program and erase. : : bit [31]: When 1, sector 31 of the non-main will be protected from program and erase.<br>0h = Minimum value of [VAL]<br>FFFFFFFFh = Maximum value of [VAL] |



Copyright © 2026 Texas Instruments Incorporated

### 6.6.20 CFGCMD Register (Offset = 13B0h) [Reset = 00000000h]

CFGCMD is shown in Figure 6-23 and described in Table 6-33.

Return to the Summary Table.

Command Configuration Register This register configures specific capabilities of the state machine for related to the execution of a command. This register is blocked for writes after CMDEXEC is written to a 1 and prior to STATCMD.DONE being set by the hardware to indicate that command execution has completed.

**Figure 6-23. CFGCMD Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | HOLDCLKSTREN | CTRLCLKSTREN | RDCLKSTREN | WAITSTATE | | | |
| R-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | | | |

**Table 6-33. CFGCMD Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-7 | RESERVED | R | 0h | Reserved |
| 6 | HOLDCLKSTREN | R/W | 1h | Enable pulse stretching for the clocking of the hold latches for inputs to the flash bank. This effectively divides the flash controller internal clock in order to create a 50/50 duty cycle clock for hold latching.<br>0h = Disable<br>1h = Enable |
| 5 | CTRLCLKSTREN | R/W | 1h | Enable pulse stretching when generating a control clock to the flash bank from the<br>flash wrapper. This effectively divides the control clock driven to the bank in order<br>to avoid minimum pulse width requirements at the bank.<br>0h = Disable<br>1h = Enable |
| 4 | RDCLKSTREN | R/W | 1h | Enable pulse stretching when generating a read clock to the flash bank from the<br>flash wrapper. This effectively divides the read clock driven to the bank in order<br>to avoid minimum pulse width requirements at the bank.<br>0h = Disable<br>1h = Enable |
| 3-0 | WAITSTATE | R/W | 2h | Wait State setting for program verify, erase verify and read verify<br>0h = Minimum value<br>Fh = Maximum value |



Copyright © 2026 Texas Instruments Incorporated

### 6.6.21 CFGPCNT Register (Offset = 13B4h) [Reset = 00000000h]

CFGPCNT is shown in Figure 6-24 and described in Table 6-34.

Return to the Summary Table.

Pulse Counter Configuration Register
This register allows further configuration of maximum pulse counts for program and erase operations.
This register is blocked for writes after a 1 is written to the CMDEXEC register and prior to STATCMD.DONE being set by the flash wrapper hardware.

**Figure 6-24. CFGPCNT Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MAXERSPCNTVAL | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MAXERSPCNTVAL | | | | RESERVED | | | MAXERSPCNTOVR |
| R/W-0h | | | | R-0h | | | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | MAXPCNTVAL | | | |
| R-0h | | | | R/W-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MAXPCNTVAL | | | | RESERVED | | | MAXPCNTOVR |
| R/W-0h | | | | R-0h | | | R/W-0h |

**Table 6-34. CFGPCNT Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | MAXERSPCNTVAL | R/W | 0h | Override maximum pulse count for erase with this value.<br>If MAXERSPCNTOVR = 0, then this field is ignored. If MAXERSPCNTOVR = 1, then this value will be used to override the max pulse count for erase.<br>0h = Minimum value<br>FFFh = Maximum value |
| 19-17 | RESERVED | R | 0h | Reserved |
| 16 | MAXERSPCNTOVR | R/W | 0h | Override hard-wired maximum pulse count for erase. If set, then the value in MAXERSPCNTVAL will be used as the max pulse count for erase operations. By default, this bit is 0, and a hard-wired max pulse count is used.<br>0h = Use hard-wired (default) value for maximum pulse count<br>1h = Use value from MAXERSPCNTVAL field as maximum erase pulse count |
| 15-12 | RESERVED | R | 0h | Reserved |
| 11-4 | MAXPCNTVAL | R/W | 0h | Override maximum pulse counter with this value. If MAXPCNTOVR = 0, then this field is ignored. If MAXPCNTOVR = 1 and MAXERSPCNTOVR = 0, then this value will be used to override the max pulse count for both program and erase. Full max value will be {4'h0, MAXPCNTVAL} . If MAXPCNTOVR = 1 and MAXERSPCNTOVR = 1, then this value will be used to override the max pulse count for program only. Full max value will be {4'h0, MAXPCNTVAL}.<br>0h = Minimum value<br>FFh = Maximum value |
| 3-1 | RESERVED | R | 0h | Reserved |



Copyright © 2026 Texas Instruments Incorporated

**Table 6-34. CFGPCNT Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 0 | MAXPCNTOVR | R/W | 0h | Override hard-wired maximum pulse count. If MAXERSPCNTOVR is not set, then setting this value alone will override the max pulse count for both program and erase. If MAXERSPCNTOVR is set, then this bit will only control the max pulse count setting for program. By default, this bit is 0, and a hard-wired max pulse count is used.<br>0h = Use hard-wired (default) value for maximum pulse count<br>1h = Use value from MAXPCNTVAL field as maximum puse count |



Copyright © 2026 Texas Instruments Incorporated

### 6.6.22 STATCMD Register (Offset = 13D0h) [Reset = 00000000h]

STATCMD is shown in Figure 6-25 and described in Table 6-35.

Return to the Summary Table.

Command Status Register This register contains status regarding completion and errors of command execution.

**Figure 6-25. STATCMD Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | FAILMISC | RESERVED | | | FAILINVDATA |
| R-0h | | | R-0h | R-0h | | | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| FAILMODE | FAILILLADDR | FAILVERIFY | FAILWEPROT | RESERVED | CMDINPROGRESS | CMDPASS | CMDDONE |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 6-35. STATCMD Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-13 | RESERVED | R | 0h | Reserved |
| 12 | FAILMISC | R | 0h | Command failed due to error other than write/erase protect violation or verify error. This is an extra bit in case a new failure mechanism is added which requires a status bit.<br>0h = No Fail<br>1h = Fail |
| 11-9 | RESERVED | R | 0h | Reserved |
| 8 | FAILINVDATA | R | 0h | Program command failed because an attempt was made to program a stored 0 value to a 1.<br>0h = No Fail<br>1h = Fail |
| 7 | FAILMODE | R | 0h | Command failed because a bank has been set to a mode other than READ. Program and Erase commands cannot be initiated unless all banks are in READ mode.<br>0h = No Fail<br>1h = Fail |
| 6 | FAILILLADDR | R | 0h | Command failed due to the use of an illegal address<br>0h = No Fail<br>1h = Fail |
| 5 | FAILVERIFY | R | 0h | Command failed due to verify error<br>0h = No Fail<br>1h = Fail |
| 4 | FAILWEPROT | R | 0h | Command failed due to Write/Erase Protect Sector Violation<br>0h = No Fail<br>1h = Fail |
| 3 | RESERVED | R | 0h | Reserved |
| 2 | CMDINPROGRESS | R | 0h | Command In Progress<br>0h = Complete<br>1h = In Progress |



Copyright © 2026 Texas Instruments Incorporated

**Table 6-35. STATCMD Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 1 | CMDPASS | R | 0h | Command Pass - valid when CMD_DONE field is 1<br>0h = Fail<br>1h = Pass |
| 0 | CMDDONE | R | 0h | Command Done<br>0h = Not Done<br>1h = Done |



Copyright © 2026 Texas Instruments Incorporated

### 6.6.23 STATPCNT Register (Offset = 13D8h) [Reset = 00000000h]

STATPCNT is shown in Figure 6-26 and described in Table 6-36.

Return to the Summary Table.

Current Pulse Count Register: Read only register giving read access to the state machine current pulse count value for program/erase operations.

**Figure 6-26. STATPCNT Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | PULSECNT | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | | | | | |

**Table 6-36. STATPCNT Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-12 | RESERVED | R | 0h | Reserved |
| 11-0 | PULSECNT | R | 0h | Current Pulse Counter Value<br>0h = Minimum value<br>FFFh = Maximum value |



Copyright © 2026 Texas Instruments Incorporated

### 6.6.24 STATMODE Register (Offset = 13DCh) [Reset = 00000000h]

STATMODE is shown in Figure 6-27 and described in Table 6-37.

Return to the Summary Table.

Mode Status Register Indicates any banks which not in READ mode, and it indicates the mode which the bank(s) are in.

**Figure 6-27. STATMODE Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | BANK1TRDY | BANK2TRDY |
| R-0h | | | | | | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | BANKMODE | | | |
| R-0h | | | | R-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | RESERVED | | | | BANKNOTINRD |
| R-0h | | | R-0h | | | | R-0h |

**Table 6-37. STATMODE Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-18 | RESERVED | R | 0h | Reserved |
| 17 | BANK1TRDY | R | 0h | Bank 1T Ready. Bank(s) are ready for 1T access. This is accomplished when the bank and pump have been trimmed.<br>0h = Not ready<br>1h = Ready |
| 16 | BANK2TRDY | R | 0h | Bank 2T Ready. Bank(s) are ready for 2T access. This is accomplished when the pump has fully driven power rails to the bank(s).<br>0h = Not ready<br>1h = Ready |
| 15-12 | RESERVED | R | 0h | Reserved |
| 11-8 | BANKMODE | R | 0h | Indicates mode of bank(s) that are not in READ mode<br>0h = Read Mode<br>2h = Read Margin 0 Mode<br>4h = Read Margin 1 Mode<br>6h = Read Margin 0B Mode<br>7h = Read Margin 1B Mode<br>9h = Program Verify Mode<br>Ah = Program Single Word<br>Bh = Erase Verify Mode<br>Ch = Erase Sector<br>Eh = Program Multiple Word<br>Fh = Erase Bank |
| 7-5 | RESERVED | R | 0h | Reserved |
| 4-1 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

**Table 6-37. STATMODE Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 0 | BANKNOTINRD | R | 0h | Bank not in read mode. Indicates which banks are not in READ mode. There is 1 bit per bank.<br>1h = Bank 0<br>2h = Bank 1<br>4h = Bank 2<br>8h = Bank 3<br>10h = Bank 4 |



Copyright © 2026 Texas Instruments Incorporated

### 6.6.25 GBLINFO0 Register (Offset = 13F0h) [Reset = 00000000h]

GBLINFO0 is shown in Figure 6-28 and described in Table 6-38.

Return to the Summary Table.

Global Info 0 Register Read only register detailing information about sector size and number of banks present.

**Figure 6-28. GBLINFO0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | NUMBANKS | | |
| R-0h | | | | | | | | | | | | | R-0h | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SECTORSIZE | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |

**Table 6-38. GBLINFO0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-19 | RESERVED | R | 0h | Reserved |
| 18-16 | NUMBANKS | R | 0h | Number of banks instantiated Minimum: 1 Maximum: 5<br>1h = Minimum value<br>5h = Maximum value |
| 15-0 | SECTORSIZE | R | 800h | Sector size in bytes<br>400h = Sector size is ONEKB<br>800h = Sector size is TWOKB |



Copyright © 2026 Texas Instruments Incorporated

### 6.6.26 GBLINFO1 Register (Offset = 13F4h) [Reset = 00000000h]

GBLINFO1 is shown in Figure 6-29 and described in Table 6-39.

Return to the Summary Table.

Global Info 1 Register Read only register detailing information about data, ecc and redundant data widths in bits.

**Figure 6-29. GBLINFO1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | REDWIDTH | | |
| R-0h | | | | | | | | | | | | | R-0h | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | ECCWIDTH | | | | | DATAWIDTH | | | | | | | |
| R-0h | | | R-0h | | | | | R-0h | | | | | | | |

**Table 6-39. GBLINFO1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-19 | RESERVED | R | 0h | Reserved |
| 18-16 | REDWIDTH | R | 4h | Redundant data width in bits<br>0h = Redundant data width is 0. Redundancy/Repair not present.<br>2h = Redundant data width is 2 bits<br>4h = Redundant data width is 4 bits |
| 15-13 | RESERVED | R | 0h | Reserved |
| 12-8 | ECCWIDTH | R | 10h | ECC data width in bits<br>0h = ECC data width is 0. ECC not used.<br>8h = ECC data width is 8 bits<br>10h = ECC data width is 16 bits |
| 7-0 | DATAWIDTH | R | 80h | Data width in bits<br>40h = Data width is 64 bits<br>80h = Data width is 128 bits |



Copyright © 2026 Texas Instruments Incorporated

### 6.6.27 GBLINFO2 Register (Offset = 13F8h) [Reset = 00000000h]

GBLINFO2 is shown in Figure 6-30 and described in Table 6-40.

Return to the Summary Table.

Global Info 2 Register Read only register detailing information about the number of data registers present.

**Figure 6-30. GBLINFO2 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | DATAREGISTERS | | | |
| R-0h | | | | R-0h | | | |

**Table 6-40. GBLINFO2 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | Reserved |
| 3-0 | DATAREGISTERS | R | 1h | Number of data registers present.<br>1h = Minimum value of DATAREGISTERS<br>8h = Maximum value of DATAREGISTERS |



Copyright © 2026 Texas Instruments Incorporated

### 6.6.28 BANK0INFO0 Register (Offset = 1400h) [Reset = 00000000h]

BANK0INFO0 is shown in Figure 6-31 and described in Table 6-41.

Return to the Summary Table.

Bank Info 0 Register for bank 0. Read only register detailing information about Main region size in the bank.

**Figure 6-31. BANK0INFO0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | MAINSIZE | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | | | | | |

**Table 6-41. BANK0INFO0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-12 | RESERVED | R | 0h | Reserved |
| 11-0 | MAINSIZE | R | 0h | Main region size in sectors Minimum: 0x8 (8) Maximum: 0x200 (512)<br>8h = Minimum value of [MAINSIZE]<br>200h = Maximum value of [MAINSIZE] |



Copyright © 2026 Texas Instruments Incorporated

### 6.6.29 BANK0INFO1 Register (Offset = 1404h) [Reset = 00000000h]

BANK0INFO1 is shown in Figure 6-32 and described in Table 6-42.

Return to the Summary Table.

Bank Info1 Register for bank 0. Read only register detailing information about Non-Main, Trim, and Engr region sizes in the bank.

**Figure 6-32. BANK0INFO1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | RESERVED | | | | | | | |
| R-0h | | | | | | | | R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | NONMAINSIZE | | | | | | | |
| R-0h | | | | | | | | R-0h | | | | | | | |

**Table 6-42. BANK0INFO1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | RESERVED | R | 0h | Reserved |
| 23-8 | RESERVED | R | 0h | |
| 7-0 | NONMAINSIZE | R | 20h | Non-main region size in sectors Minimum: 0x0 (0) Maximum: 0x10 (16)<br>0h = Minimum value of [NONMAINSIZE]<br>20h = Maximum value of [NONMAINSIZE] |



Copyright © 2026 Texas Instruments Incorporated

### 6.6.30 BANK1INFO0 Register (Offset = 1410h) [Reset = 00000000h]

BANK1INFO0 is shown in Figure 6-33 and described in Table 6-43.

Return to the Summary Table.

Bank Info 0 Register for bank 1. Read only register detailing information about Main region size in the bank.

**Figure 6-33. BANK1INFO0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | MAINSIZE | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | | | | | |

**Table 6-43. BANK1INFO0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-12 | RESERVED | R | 0h | Reserved |
| 11-0 | MAINSIZE | R | 200h | Main region size in sectors Minimum: 0x8 (8) Maximum: 0x200 (512)<br>8h = Minimum value of [MAINSIZE]<br>200h = Maximum value of [MAINSIZE] |



Copyright © 2026 Texas Instruments Incorporated

### 6.6.31 BANK1INFO1 Register (Offset = 1414h) [Reset = 00000000h]

BANK1INFO1 is shown in Figure 6-34 and described in Table 6-44.

Return to the Summary Table.

Bank Info1 Register for bank 1. Read only register detailing information about Non-Main, Trim, and Engr region sizes in the bank.

**Figure 6-34. BANK1INFO1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | RESERVED | | | | | | | |
| R-0h | | | | | | | | R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | NONMAINSIZE | | | | | | | |
| R-0h | | | | | | | | R-0h | | | | | | | |

**Table 6-44. BANK1INFO1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | RESERVED | R | 0h | Reserved |
| 23-8 | RESERVED | R | 0h | |
| 7-0 | NONMAINSIZE | R | 20h | Non-main region size in sectors<br>0h = Minimum value of [NONMAINSIZE]<br>20h = Maximum value of [NONMAINSIZE] |



Copyright © 2026 Texas Instruments Incorporated

### 6.6.32 BANK2INFO0 Register (Offset = 1420h) [Reset = 00000000h]

BANK2INFO0 is shown in Figure 6-35 and described in Table 6-45.

Return to the Summary Table.

Bank Info 0 Register for bank 2. Read only register detailing information about Main region size in the bank.

**Figure 6-35. BANK2INFO0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | MAINSIZE | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | | | | | |

**Table 6-45. BANK2INFO0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-12 | RESERVED | R | 0h | Reserved |
| 11-0 | MAINSIZE | R | 200h | Main region size in sectors Minimum: 0x8 (8) Maximum: 0x200 (512)<br>8h = Minimum value of [MAINSIZE]<br>200h = Maximum value of [MAINSIZE] |



Copyright © 2026 Texas Instruments Incorporated

### 6.6.33 BANK2INFO1 Register (Offset = 1424h) [Reset = 00000000h]

BANK2INFO1 is shown in Figure 6-36 and described in Table 6-46.

Return to the Summary Table.

Bank Info1 Register for bank 2. Read only register detailing information about Non-Main, Trim, and Engr region sizes in the bank.

**Figure 6-36. BANK2INFO1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | RESERVED | | | | | | | |
| R-0h | | | | | | | | R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | NONMAINSIZE | | | | | | | |
| R-0h | | | | | | | | R-0h | | | | | | | |

**Table 6-46. BANK2INFO1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | RESERVED | R | 0h | Reserved |
| 23-8 | RESERVED | R | 0h | |
| 7-0 | NONMAINSIZE | R | 20h | Non-main region size in sectors<br>0h = Minimum value of [NONMAINSIZE]<br>20h = Maximum value of [NONMAINSIZE] |



Copyright © 2026 Texas Instruments Incorporated

# Chapter 7
# Events

The event manager provides the peripheral-to-peripheral, peripheral-to-DMA, and peripheral-to-CPU (IRQ) event connections.

**7.1 Events Overview**.....641
**7.2 Events Operation**.....645



Copyright © 2026 Texas Instruments Incorporated

## 7.1 Events Overview

The event manager transfers digital events from one entity (for example, a peripheral) to another (for example, a second peripheral, the DMA, or the CPU). The event manager implements event transfer through a defined set of event publishers (generators) and subscribers (receivers) which are interconnected through an event fabric containing a combination of fixed (static) and programmable routes.

Events which are transferred by the event manager include:

- Peripheral event transferred to the CPU as an interrupt request (IRQ)
  - Example: RTC interrupt is sent to the CPU
- Peripheral event transferred to the DMA as a DMA trigger
  - Example: UART data receive trigger to DMA to request a DMA transfer
- Peripheral event transferred to another peripheral to directly trigger an action in hardware
  - Example: TIMx timer peripheral publishes a periodic event to the ADC subscriber port, and the ADC uses the event to trigger start-of-sampling

In addition to providing the event transfer logic, the event manager also interfaces with the power management and clock unit (PMCU) if an event requires the power and/or clock configuration of the device to change to handle the event properly. For example, if a peripheral asserts an event that targets the DMA, and the device is in a STOP or STANDBY operating mode (DMA is disabled), the event manager will handshake with the PMCU to suspend the low power operating mode state temporarily and enable the DMA such that the DMA transfer can be processed.

The event manager configuration is device dependent, as different devices support different peripherals. See the device-specific data sheet for information on the device-specific event implementation.

### *7.1.1 Event Publisher*

An event publisher is the source of an event which is propagated on the event fabric. Peripherals contain event publishers for publishing CPU interrupts, DMA triggers and generic events to the event fabric through the publishing port FPUB_x. Publisher behavior is configured with standardized event management registers.

### *7.1.2 Event Subscriber*

Event subscribers are included within the processor, the DMA, and certain peripherals (see Section 7.1.4). Event subscribers subscribe the events through the subscribing port FSUB_x. Event subscribers enable modules to be able to subscribe to, and take a predefined action upon, events which are published to the event fabric by an event publisher.

### *7.1.3 Event Fabric Routing*

There are three different types of routes through the event fabric which are used to connect a publisher to a subscriber: CPU interrupt events, DMA trigger events, and generic events.

#### 7.1.3.1 CPU Interrupt Event Route (CPU_INT)

A CPU interrupt event route is a fixed, point-to-point connection between one event publisher (inside a peripheral module) and one event subscriber (the CPU subsystem) used to propagate CPU interrupts.

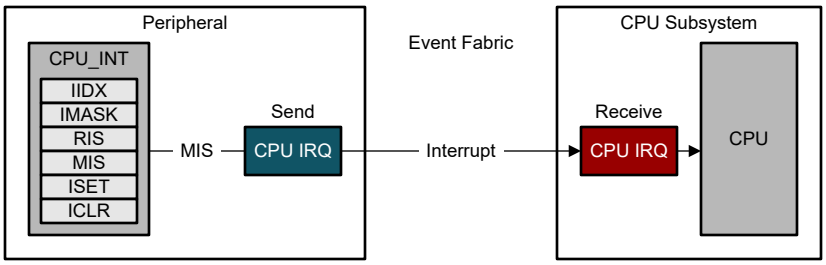



**Figure 7-1. CPU Interrupt (Fixed Event Route)**



Copyright © 2026 Texas Instruments Incorporated

For each peripheral which is capable of generating a CPU interrupt, a fixed route is provided from the peripheral's masked interrupt status (MIS) register to the CPU subsystem's interrupt management logic.

If software does not clear the interrupt request in the peripheral's event management registers, the request will remain pending to the CPU subsystem. See Section 7.2.5.3 for guidance on setting and clearing interrupt status with the event management registers.

#### 7.1.3.2 DMA Trigger Event Route (DMA_TRIGx)

A DMA route is a fixed route between a peripheral and the DMA controller, which optionally has additional side-band signals to pass a DMA done condition from the DMA controller back to the triggering peripheral to indicate when a DMA activity has run to completion. The DMA trigger route is shown in Figure 7-2.

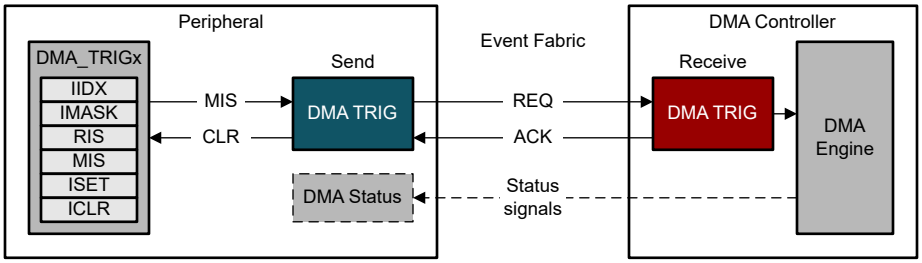



**Figure 7-2. DMA Route**

Most peripherals capable of generating a DMA trigger have an additional set of event management registers (in addition to the CPU_INT registers used for the CPU interrupt and any GEN_EVENTx generic route publishers). These registers can be used to select which peripheral condition to use for generating the DMA trigger.

When a trigger is received by the DMA, the DMA acknowledges the request and the peripheral clears the request. The DMA also acknowledges the cleared request, after which a new request can be asserted by the peripheral.

The DMA route can also contain status signals (for specific peripherals) to indicate to the triggering peripheral that a DMA transfer sequence has completed. For example, the DMA can be set up to transfer *N* number of bytes from an SRAM buffer into the UART TX data register based on the UART TX DMA trigger. Upon each trigger from the UART, the DMA will acknowledge that the transfer was successful. On the *N*th byte, the DMA will send a complete status signal to the UART, which the UART can use to propagate a transfer completion interrupt to the CPU.

### Special Cases

Certain peripherals (for example, the 12-bit DAC) do not implement an event management register set for managing their DMA triggers. In these cases, the peripheral implements specific DMA configuration logic such that the management registers are not needed to interface with the DMA. Figure 7-3 shows the model when the event management registers are not implemented. See the peripheral-specific section of this document for guidance on how to configure DMA channels in this case.

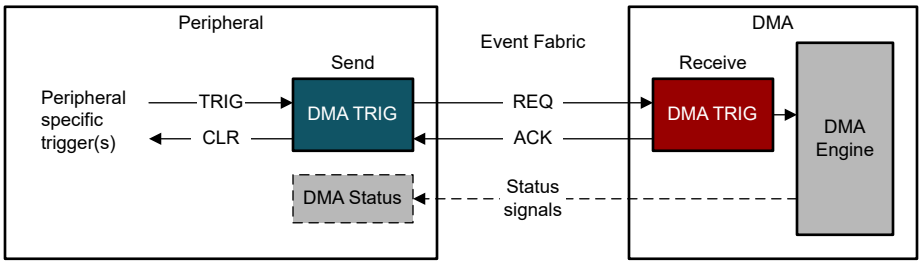



**Figure 7-3. DMA Route without Event Management Registers**



Copyright © 2026 Texas Instruments Incorporated

#### 7.1.3.3 Generic Event Route (GEN_EVENTx)

A generic route is either a point-to-point (1:1) route or a point-to-two (1:2) splitter route in which the peripheral publishing the event uses one of several available generic route channels to publish its event to another entity (or entities, in the case of a splitter route), where an entity can be another peripheral, a generic DMA trigger event, or a generic CPU event, as shown in Figure 7-4.

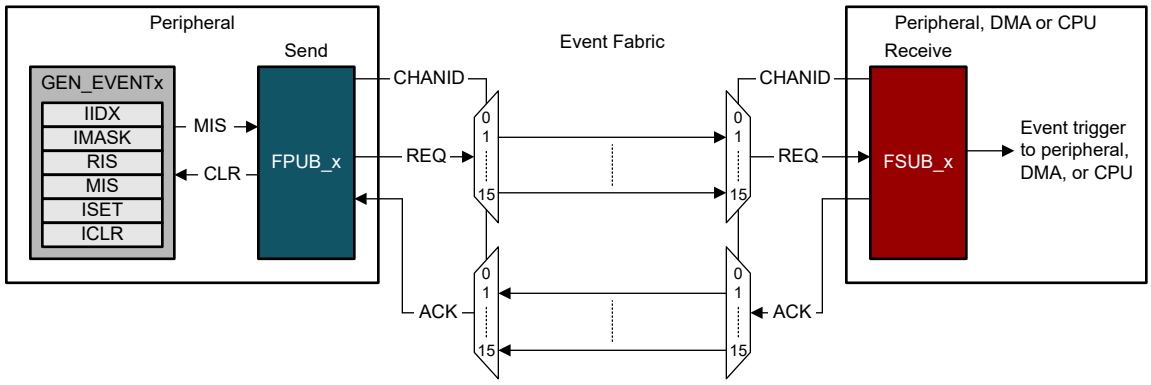



**Figure 7-4. Generic Route**

Peripherals capable of generating a generic event have an additional group (our groups) of GEN_EVENTx event management registers (in addition to the CPU_INT registers used for the CPU interrupt or DMA_TRIGx for DMA, if present). These registers can be used to select the peripheral condition to use for publishing a generic event. When configured, the event will broadcast out to the generic route channel selected by the FPUB_x register. A second peripheral, the DMA, or the CPU can subscribe to this event by configuring its generic subscriber port (FSUB_x) to listen on the same generic route channel to which the publishing peripheral is connected to.

Generic route channels can be configured with one subscriber (1:1 route) or two subscribers (1:2 splitter route), depending on which channel is selected. See the device data sheet for a complete listing of the available generic route channels and their type (1:1 or 1:2). Generic route channels can only be configured with one publishing peripheral at a time. Once a peripheral subscribes to a 1:1 generic route channel, no other peripheral will be able to select that channel to subscribe to, unless the originally connected peripheral is disconnected first. Generic route channels with splitter capability (1:2) enable exactly two peripherals to subscribe to the channel, after which additional attempts to add subscribers will be blocked by hardware until both of the two connected peripherals are disconnected from the splitter channel.

Each peripheral type has unique capabilities in terms of what can generate an event to publish, and what a subscribed event is capable of triggering within the peripheral. Review the chapter of this guide which corresponds to the peripheral of interest to understand what the publisher and subscriber ports on a given peripheral are capable of.

### *7.1.4 Event Routing Map*

The event capabilities of each peripheral type are shown in Figure 7-5. Peripherals such as UART, SPI, and I2C generate CPU interrupt events which are routed to the CPU, and they also generate DMA trigger events which are routed to the DMA. Peripherals such as GPIO and ADC generate CPU interrupt events as well, but they also support generating and receiving events routed through a generic channel. For example, through the use of a generic event channel, it is possible to directly start an ADC conversion from a GPIO event by connecting a GPIO FPUB_x and ADC FSUB_0 to the same generic event channel.



Copyright © 2026 Texas Instruments Incorporated

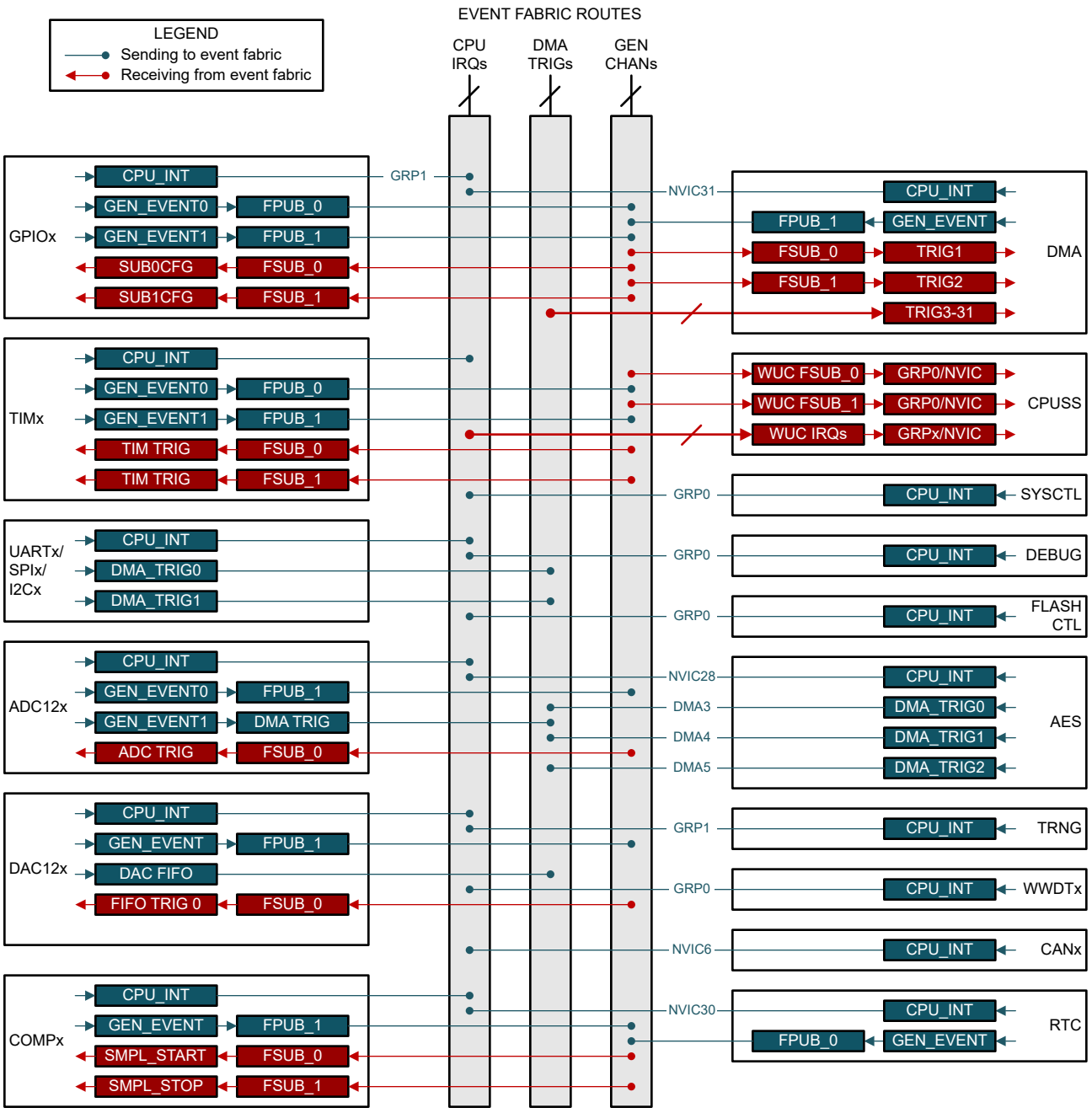



**Figure 7-5. Event Map**

### *7.1.5 Event Propagation Latency*

Generic route channels implement a four-way hardware handshake between the publishing entity and the subscribing entity. This handshake requires four ULPCLK cycles to complete:

1. Request from publisher to subscriber
2. Acknowledge from subscriber to publisher
3. De-assert of request from publisher to subscriber
4. Acknowledge of de-assert from subscriber to publisher

If the publishing peripheral sends two requests and the first request has not cleared the handshake, the second request is dropped.



Copyright © 2026 Texas Instruments Incorporated

## 7.2 Events Operation

This section describes how to configure peripherals to use the event manager. Note that the event manager does not contain any configuration registers. All event configuration is done through the publishing and subscribing peripherals.

### *7.2.1 CPU Interrupt*

Peripheral interrupt requests (IRQs) are propagated to the CPU subsystem through the event manager. Peripheral interrupt requests use fixed routes, but in addition the CPU subsystem may provide two generic event subscriber ports which can be used to trigger CPU interrupts through a generic route. See the device-specific data sheet for the complete list of interrupt assignments for a given device.

**Standard CPU Interrupt Requests**

No special configuration is required for fixed route interrupts. Interrupts can be managed through the peripheral's CPU_INT event management registers (IIDX, IMASK, RIS, MIS, ISET, and ICLR) and through the CPU subsystem interrupt configuration (see *Interrupts and Exceptions*).

**Generic Event Based CPU Interrupt Requests**

The CPU subsystem contains two generic event subscriber ports (FSUB_x) which can be used to source a CPU interrupt from any of the device's generic event channels. This can be used to enable special cases where a particular function on a peripheral generates a dedicated interrupt to the CPU subsystem which is independent from, and in addition to, that peripheral's standard interrupt mechanism.

Consider the GPIO peripheral, which has a standard interrupt request as well as 2 publishers which can route a GPIO event to any of the generic event channels based on a defined state in the GPIO. For example, it can be desirable to have most GPIO events configured to source the standard interrupt, while a single specific GPIO event sources a second dedicated CPU interrupt through a generic route. This enables the application software to have two completely independent interrupt handlers for the GPIO.

To configure the event manager to trigger a CPU interrupt from a generic route, follow the steps below:

1. Configure the GEN_EVENT registers of the peripheral generating the event to select the desired peripheral state as an event generator.
2. Configure the FPUB_x register of the peripheral generating the event with the generic route channel ID which is to be used. This channel must not be in use by another peripheral.
3. Configure the FSUB_x register of the wake up controller (WUC), which captures generic route channel events to forward to the CPU subsystem.
4. Configure the CPU subsystem interrupt management to enable the GENSUBx interrupt.

Note that when generating a CPU interrupt through a generic route, the generic event logic will automatically clear the pending interrupt request as a part of the four-way event handshake. Application software will not be able to read the cause of the interrupt from the peripheral registers, and it does not need to clear any interrupt status bits. Software can only read that the FSUB_x generic event generated an interrupt. This reduces the interrupt overhead.

### *7.2.2 DMA Trigger*

DMA triggers are propagated to the DMA through the event manager. Most DMA triggers use fixed routes, but the DMA does provide two generic event subscriber ports which can be used to trigger DMA transfers through a generic route channel. See the device-specific data sheet for the complete list of DMA trigger assignments for a given device.

**Standard DMA Triggers**

To determine if a particular peripheral on a device provides a fixed DMA trigger (DMA_TRIGx) from the peripheral directly to the DMA, review the DMA triggers table in the detailed description section of the device-



Copyright © 2026 Texas Instruments Incorporated

specific data sheet. Certain peripherals can have more than one DMA trigger (for example, to enable a TX trigger and an RX trigger on a serial communication peripheral).

To select the specific peripheral event which triggers a static DMA route, configure the peripheral's DMA_TRIGx event management register set (IIDX, IMASK, RIS, MIS, ISET, and ICLR) which corresponds to the targeted DMA route. To determine which DMA_TRIGx register set corresponds with which DMA trigger, review the relevant chapter of this guide for the corresponding peripheral, or review Section 7.1.4.

Certain peripherals (such as the 12-bit DAC) do not implement a DMA_TRIGx register set for managing DMA triggers. In these cases, the DMA trigger configuration is done through peripheral-specific configuration registers.

#### Generic Event Based DMA Triggers

The DMA contains two generic event subscriber ports (FSUB_x) which can be used to source a DMA trigger from any of the device's generic event channels. This can be used to enable special cases where a particular function on a peripheral generates a DMA trigger. For example, it can be desirable to trigger a DMA transfer from a timer.

To configure the event manager to trigger a DMA channel from a generic route, follow the steps below:

1. Configure the GEN_EVENTx registers of the peripheral generating the event to select the desired peripheral state as an event generator.
2. Configure the FPUB_x register of the peripheral generating the event with the generic event channel ID which is to be used. This channel must not be in use by another peripheral.
3. Configure the FSUB_x register of the DMA, which captures generic event channel events to be used as triggers in the DMA.
4. Configure the DMA according to the configuration instructions in the DMA chapter.

### *7.2.3 Peripheral to Peripheral Event*

Peripheral to peripheral events enable a condition in one peripheral to trigger an action in a second (or third) peripheral, completely in hardware without any CPU interaction. The device provides a certain number of generic route channels which can be either published to or subscribed to by peripherals which include publisher and subscriber ports. Before establishing a configuration, follow these steps:

1. Review the device specific data sheet to determine the generic route channel count and channel type available on the target device. Select an appropriate channel type (point to point or splitter) based on the desired functionality, and determine the channel number to use for the connection (the channel must not already be used by other peripherals).
2. Review the publisher and subscriber capabilities of the peripherals which are to be connected. Some peripherals have more than one publisher and/or more than one subscriber port, and some peripherals have no publisher or subscriber ports. To understand the available ports for a peripheral, review the peripheral's reference chapter in this guide, or check the generic event channel connections in *Event Routing Map*.

Once the channel to be used is determined, and both the publisher and subscriber ports for the peripherals being connected are known, use the steps below to establish the event connection. In this example, a timer triggered ADC application will be configured, using TIMG0 to publish an event to generic channel 1, with ADC0 subscribing to generic channel 1 as a start-of-conversion trigger.

1. Configure the GEN_EVENTx event management registers of TIMG0 to set the event request based on the appropriate timer event (for example, a zero event).
2. Store 0x1 into the FPUB_0 register of TIMG0 to publish the TIMG0 event selected by the GEN_EVENTx registers to generic route channel 1. Channel 1 must not be in use by another peripheral.
3. Store 0x1 the FSUB_0 register of ADC0 so that ADC0 is listening for events published by the timer to channel 1.
4. Configure ADC0 to trigger from the subscriber port according to the configuration instructions in *Sampling Trigger Sources and Sampling Modes*.
5. Configure and enable TIMG0.



Copyright © 2026 Texas Instruments Incorporated

### 7.2.4 Extended Module Description Register

The DESC_EX register is a read-only register in the event manager which can be read by application software to determine how many point to point (single) generic route channels and splitter (dual) generic route channels are available on a given device.

### 7.2.5 Using Event Registers

The event management register group is a set of standard registers which are implemented by all peripherals capable of generating events (CPU interrupts, DMA triggers, or generic events). Each event generator in a peripheral contains its own event management register set. For example, if a peripheral supports generating a CPU interrupt and a DMA trigger, it will have an event management register set for the CPU interrupt (with the group name of CPU_INT) as well as a second event management register set for the DMA trigger (with the group name of DMA_TRIG).

The event management registers are used to:

- Configure which peripheral conditions are used to generate the event (masking)
- Communicate raw and masked peripheral event status
- Set or clear peripheral event status by software

In the "Registers" section for a given peripheral, the "Group" column displays the Group Name to indicate what functionality is mapped to each event management register group. See Table 7-1 for which event management groups are mapped to specific functions in the group name for a peripheral's "Registers" section.

**Table 7-1. Event Management Group Functionality and Mapping**

| Group Name (in Registers) | Functionality |
|---|---|
| CPU_INT | CPU interrupt (fixed route to the CPU subsystem) |
| DMA_TRIGx | DMA trigger (fixed route to the DMA controller) |
| GEN_EVENT | Generic event (programmable route for other module-to-module connections) |

#### 7.2.5.1 Event Registers

The event management register set contains 6 standard registers: RIS, IMASK, MIS, ISET, ICLR, and IIDX, given in Table 7-2. The event registers are interconnected as shown in Figure 7-6.

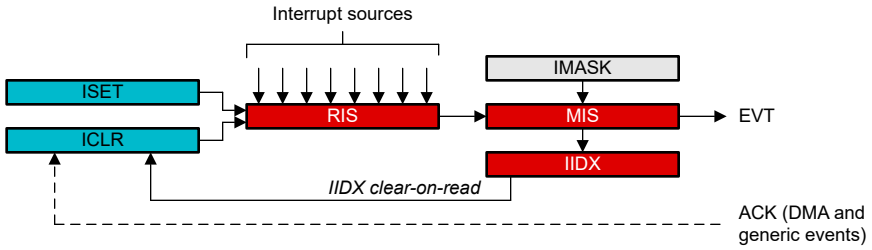



**Figure 7-6. Event Management Register Relationship**

The peripheral generating the event will contain one or more interrupt source signals which connect to the raw interrupt status (RIS) register. Software can poll RIS at any time to check the raw interrupt status. Software can also clear pending interrupts in the RIS register by writing to the corresponding bit position in the ICLR register. The RIS and IMASK registers are combined through a bit-wise AND function in the MIS register (masked interrupt status). To unmask an interrupt, set the corresponding bit in the IMASK register. Once unmasked, a pending interrupt will be indicated in both the RIS and MIS registers, and an event will be generated. The IIDX register will also be updated with the index of the highest priority pending interrupt.

In the case of a CPU interrupt (CPU_INT) with a CPU interrupt event route, a read of the IIDX register will clear the highest priority pending interrupt in the RIS and MIS registers and return the index of the highest priority pending interrupt to application software.



Copyright © 2026 Texas Instruments Incorporated

In the case of a hardware event DMA trigger route (DMA_TRIGx) or generic event route (GEN_EVENTx), the hardware four-way handshake will send an ACK signal to the ICLR mechanism which will clear the pending interrupt in the RIS and MIS registers.

**Table 7-2. Standardized Event Management Registers (Used for CPU_INT, DMA_TRIGx, GEN_EVENTx Configuration)**

| Register | Description | R/W | Functionality |
|---|---|---|---|
| RIS | Raw interrupt status | R | Indicates the current pending interrupt status, with one bit provided per interrupt condition. Writing to ICLR will clear the corresponding bit in the RIS register if the interrupt condition is no longer present. |
| IMASK | Interrupt mask | RW | Used by application software to configure which interrupt conditions propagate into an event, with one bit provided per interrupt condition. |
| MIS | Masked interrupt status | R | Indicates the current pending masked interrupt status to software and hardware, with one bit provided per interrupt condition. MIS is the bit-wise AND of the RIS and IMASK registers. Writing to ICLR will clear the corresponding bit in the RIS register if the interrupt condition is no longer present. If RIS is cleared, the corresponding bit in the MIS register is also automatically cleared. |
| ISET | Software interrupt set control | W | Used by application software to force an interrupt condition for diagnostics. Writing to ISET will set the corresponding bit in the RIS register. If the interrupt condition is enabled in IMASK, the corresponding bit in the MIS register is also set. Writing a '1' to a bit location in ISET sets the respective interrupt status. |
| ICLR | Software interrupt clear control | W | Used by application software to clear a pending interrupt status in RIS. Writing a '1' to a bit location in ICLR clears the respective interrupt status. If an interrupt is enabled in IMASK, the corresponding bit location in MIS is also cleared automatically when RIS clears. If the interrupt condition is still present, clearing the status has no effect and the RIS will remain set. |
| IIDX | Pending interrupt index | R | Used by application software to read the highest priority pending interrupt while simultaneously clearing the highest priority interrupt status in RIS and MIS. A read of IIDX returns 0 if no unmasked interrupts are pending (MIS==0), else it returns an index value indicating the highest priority pending interrupt. |

#### 7.2.5.2 Configuring Events

To configure which peripheral interrupt source is to be used to trigger an event, set the bit which corresponds to the desired interrupt source in the IMASK register which corresponds to the desired event. Setting a bit in IMASK will cause the raw interrupt status in the RIS register to propagate to the MIS register. When an interrupt status bit in the MIS register is set (due to the interrupt being unmasked in the IMASK register and a raw interrupt being pending in the RIS register), an event is generated.

Multiple interrupt sources can be enabled for CPU interrupt events, as application software can determine the cause of the interrupt by reading the IIDX or MIS register.

For hardware events such as DMA triggers and generic event publishers, only one interrupt source should be unmasked in IMASK.

#### 7.2.5.3 Responding to CPU Interrupts in Application Software

In the case of an event which generates a CPU interrupt, application software can determine which peripheral interrupt triggered the generation of the event by either reading the IIDX register or by reading the MIS and writing the ICLR registers.

### CPU IRQ Interrupt Service Routine using CPU_INT IIDX Register

Application software can read the IIDX register in CPU_INT group to determine and clear the highest priority pending interrupt. A read to IIDX will return an index corresponding to the highest priority interrupt which was both set and unmasked. The read action will also simultaneously clear the RIS and MIS bits corresponding to the



Copyright © 2026 Texas Instruments Incorporated

highest priority interrupt whose index was returned by the read. The read value from the IIDX register can then be used in a case statement, as shown below.

```
void ISR_IIDX(void)
{
  switch(IIDX)
  {
    case 0:         // no IRQ pending
      break;
    case 1:         // IRQ[0]
      do_irq0();
      break;
    case 2:         // IRQ[1]
      do_irq1();
      break;
    default:        // out of range
      illegal();
  }
}
```

**Note**

A read of IIDX clears only the single highest-priority pending interrupt. If additional unmasked interrupts remain pending (MIS ≠ 0) after the read, the interrupt request to the CPU subsystem is asserted again, and this repeats until all pending interrupts have been cleared (IIDX returns 0). Software can therefore either read IIDX once per ISR invocation (relying on re-entry for remaining interrupts) or read IIDX in a loop until 0 is returned.

#### CPU IRQ Interrupt Service Routine using CPU_INT MIS and ICLR Registers

Alternatively, application software can read the MIS register to determine which bits are set, followed by using the ICLR register to clear the pending interrupt status bits.

```
void ISR(void)
{
  uint32_t pending = MIS;
  ICLR = pending;           // clear pending IRQ
  if (pending & 0x01)       // IRQ[0]
  {
      do_irq0();
  }
  if (pending & 0x02)       // IRQ[1]
  {
      do_irq1();
  }
}
```

#### 7.2.5.4 Hardware Event Handling

In the case of an event which sources a DMA trigger (DMA_TRIG) or a generic event (GEN_EVENT), the IIDX register is not used. A four-way event handshake is performed between the peripheral generating the event and the hardware entity which is subscribed to the event (for example, the DMA or a secondary peripheral). The four-way event handshake will clear the corresponding interrupt status bits in the RIS and MIS registers automatically.



Copyright © 2026 Texas Instruments Incorporated

Chapter 8

# IOMUX

The IOMUX controls the configuration of all device pins with digital input-output (IO) functions, including: digital function selection, inversion control, drive strength (if applicable), the pullup or pulldown resistor (if applicable), and wake-up configuration (if applicable on certain IOs for wakeup from SHUTDOWN mode).


**8.1 IOMUX Overview** ..... 651
**8.2 IOMUX Operation** ..... 654
**8.3 IOMUX Registers** ..... 658





Copyright © 2026 Texas Instruments Incorporated

## 8.1 IOMUX Overview

The IOMUX manages the configuration of the digital IO. Refer to the device specific datasheet for supported features. Key functions configured by IOMUX include:

- Selection of which peripheral is multiplexed to each digital IO pin (for example, a GPIO or UART peripheral)
- Digital input path configuration
  - Hysteresis control
  - Input path enable/disable
  - Input logic inversion control
- Digital output path configuration
  - Drive strength control
  - Output connection enable/disable
  - Output logic inversion (control shared with input logic inversion)
  - Logic-high to High-Z output conversion (for open-drain style interfaces)
  - Retention of "last state" when a peripheral connected to an IO is disabled
- Wakeup configuration (for wakeup from SHUTDOWN mode if present)
  - Read wake up source from the wakestat bit from the PINCM Register
  - Wake up compare level
  - Release SHDNIOREL
  - Wakeup enable/disable
- Pullup and pulldown resistor control (if present)

### *8.1.1 IO Types and Analog Sharing*

The IOMUX manages the selection of which peripheral function is to be used on a digital IO. It also provides the controls for the output driver, input path, and the wakeup logic for wakeup from SHUTDOWN mode.

#### Digital IO Types

There are several digital IO types which can be included on a given device. Each digital IO type supports different features. Table 8-1 lists the features which are included with each IO type. See the device-specific data sheet for which IO type is used on a given package pin.

**Table 8-1. Digital IO Features by IO Type**

| IO Structure | Inversion Control | Drive Strength Control | Hysteresis Control | Pullup Resistor | Pulldown Resistor | Wakeup Logic |
|---|---|---|---|---|---|---|
| Low-drive (SDIO) | Y | | | Y | Y | |
| Low-drive with wake (SDIO) | Y | | | Y | Y | Y |
| Standard-drive (SDIO) | Y | | | Y | Y | |
| Standard-drive with wake (SDIO) | Y | | | Y | Y | Y |
| High-drive (HDIO) | Y | Y | | Y | Y | Y |
| High-speed (HSIO) | Y | Y | | Y | Y | |
| 5V tolerant open drain (ODIO) | Y | | Y | | Y | Y |

Please note that the IOMUX will not support the inversion control and pseudo-open drain (output-high translated to high-impedance) setting on the SPI POCI pins. Also the IOMUX will not support inversion control on the SPI SCLK pins connected on an HSIO pin. Not all devices support pulldown resistors refer to the device specific data sheet.

#### Digital IO Shared with Analog Functions

Certain pins on a device will be digital only and will not have any analog functions connected to the pin. Other pins can have one or more analog functions connected to the pin in addition to the digital IO functions. Analog



Copyright © 2026 Texas Instruments Incorporated

functions are never selected within the IOMUX; they are always configured within of the respective analog peripheral. Analog peripherals have no knowledge of, or interaction with, the IOMUX.

In general, when analog functionality is used on a pin which also has digital functions, the IOMUX configuration for that pin should be left in its default (high-Z) state so as to not interfere with the proper operation of the analog function. However, it is possible to have the IOMUX active on a pin when an analog peripheral is also interacting with the pin, provided that the application software ensures that there is not a conflict between the functions. For example, it is possible to have the pullup or pulldown resistor on an IO enabled at the same time that the ADC is running a conversion on the same IO. However, an invalid configuration would be enabling the output driver on an IO at the same time that an analog peripheral is driving the IO (for example, a DAC or OPA output). This would create an IO conflict.

Application software is responsible for ensuring that the IOMUX settings do not conflict with any analog peripheral functions which can be enabled on a shared pad.

### IO Slice

The mixed-signal IO pin slice diagram for a full featured IO pin is shown in Figure 8-1. Not all pins will have analog functions, wake-up logic, drive strength control, and pullup or pulldown resistors available. See the device-specific data sheet for detailed information on what features are supported for a specific pin.



Copyright © 2026 Texas Instruments Incorporated

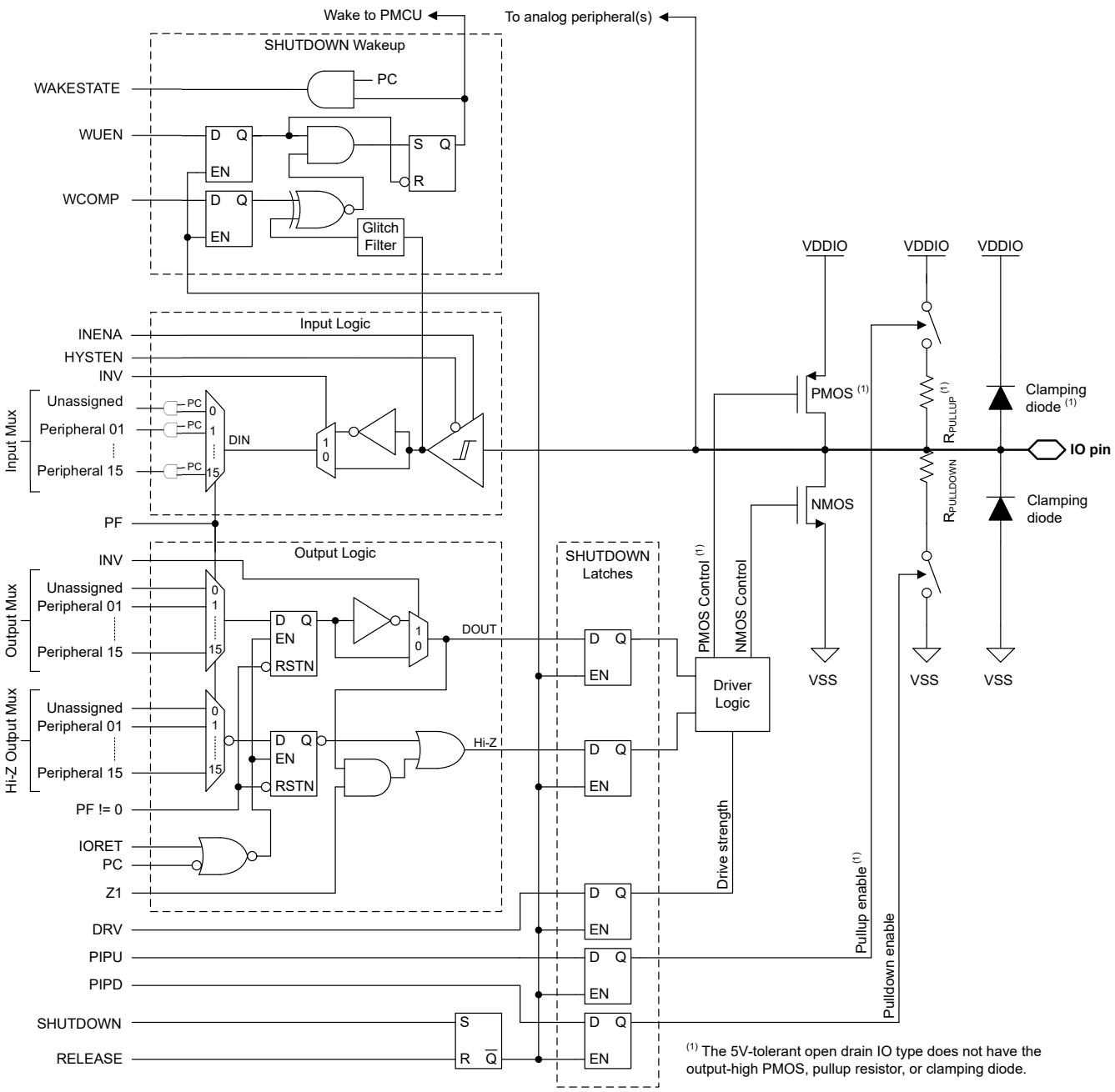



**Figure 8-1. Superset IO Slice**

### Default IOMUX State

The initial state of the IOMUX pin slice for all digital IO after a BOOTRST is as follows:

- The digital IO is in a high-impedance state
- The peripheral function selection field (PF) is cleared (no peripheral function selected) and the peripheral connect (PC) and input enable states are cleared (disabled)
- The inversion logic is disabled
- The Hi-Z output high mode is disabled
- The pullup/pulldown resistors (if present) are disabled
- The input hysteresis control (if present) is disabled to save power
- The drive strength control (if present) is reset



Copyright © 2026 Texas Instruments Incorporated

- The wakeup logic (if present) is disabled

**Note**

The SWD pins are a specific exception to the above default state. The SWD debug pins are configured in SWD mode by default, and may be switched to an alternate setting after start-up. See *SWD Pins* in the SYSCTL section of the PMCU chapter.

## 8.2 IOMUX Operation

Each digital IO on a device has a dedicated 32-bit PINCM register in the IOMUX peripheral register space which is used to configure the digital functions of the respective IO. See the device specific data sheet for determining the PINCM register index which corresponds to the IO to be configured.

### *8.2.1 Peripheral Function (PF) Assignment*

When setting up the initial IOMUX configuration for an IO after a BOOTRST, application software can select which digital peripheral from the supported options is to be connect to an IO by writing the appropriate peripheral select value to the PF field while simultaneously setting the PC and INENA bits in the PINCMx register corresponding to the targeted pin. The IOMUX configurations for a given peripheral must be set before the peripheral connected to the IOs has been initialized for operation.

To change the peripheral function selection for a digital IO at runtime after a peripheral function has already been configured for that IO, the following procedure should be followed:

1. Disable the currently connected peripheral function
2. Clear the PC bit (input/output connect bit) and INENA (input connect bit) in the corresponding PINCMx register
3. Write 0x0 to the PF field in the PINCMx to clear the logic in the data path
4. Select the new peripheral function by writing the peripheral function ID to the PF register
5. Set the PC and INENA bits in the PINCMx register to connect the newly selected peripheral
6. Enable the newly selected peripheral for operation

At runtime, the INENA bit can be used to mask the input from the IO to the peripheral, if desired. When INENA is cleared, a connected peripheral function will see logic low (0) from the IO, regardless of the external state of the IO. If an IO supports wakeup from SHUTDOWN mode, the INENA bit also controls propagation of the IO state to the SHUTDOWN mode wakeup logic.

If a peripheral is assigned to an IO, but the peripheral is itself in a disabled state, the last valid output conditions (output logic level and Hi-Z state) are latched in the IOMUX output logic. When the peripheral is enabled, the IOMUX will release the latched state to allow the (now enabled) peripheral's output state to propagate to the IO. Note that when the GPIO peripheral is selected by the IOMUX, the DOE[x] field also needs to be enabled to set DOUT and conversely cleared to configure a Hi-Z state. The PMCU indicates to the IOMUX when a peripheral is entering a disabled state via the IORET signal, which is combined with the PC signal via a logic OR to control the output state latches. This mechanism handles preservation of the last valid output state of peripherals in power domain 1 (PD1) when entering STOP or STANDBY mode, as PD1 peripherals are always temporarily disabled upon entry to STOP/STANDBY, and re-enabled upon exit from STOP/STANDBY modes.

When no peripheral function is selected (PF==0) the output latches are put into a reset state, causing the output NMOS and PMOS to be disabled (leaving the IO pin in a Hi-Z state with the exception of any enabled pullup/pulldown resistors). Note that the pullup/pulldown resistors are never controlled by either a connected peripheral or the peripheral muxing logic. They are only controlled by the IOMUX control bits (see pullup/pulldown).

### *8.2.2 Logic High to Hi-Z Conversion*

The IOMUX supports translating an output high signal from a connected peripheral into a Hi-Z output state at the IO pin. This functionality is particularly useful for open-drain digital input/output applications. When this functionality is enabled, the IO pin state as a function of the peripheral output is as shown in Table 8-2.



Copyright © 2026 Texas Instruments Incorporated

**Table 8-2. Logic High to Hi-Z Truth Table**

| Connected Peripheral Output | IO Pin State (Z1 = 0x0) | IO Pin State (Z1 = 0x1) |
|---|---|---|
| Logic low (0) | Output low | Output low |
| Logic high (1) | Output high | High impedance (Hi-Z) |

To enable logic high to Hi-Z conversion on a digital IO, set the Z1 bit in the corresponding PINCMx register.

Note that for 5V tolerant open-drain IO pins, the Z1 control has no effect as there is no high-side driver present. On these pins, a logic high output from the peripheral to the IO pin always results in a Hi-Z state.

### *8.2.3 Logic Inversion*

The IOMUX supports logic inversion of the digital input/output path. Logic inversion is useful for scenarios where opposite polarity is required for UART functions or SPI chip select functions.

To enable logic inversion on a digital IO, set the INV bit in the corresponding PINCMx register. To disable logic inversion, clear the corresponding bit. Logic inversion is disabled by default.

When logic inversion is enabled for a 5V tolerant open drain IO, a connected peripheral which outputs a logic low state will cause the IO pin to go to a Hi-Z state. When the peripheral applies a logic high state, the IO pin will go to an output low state.

### *8.2.4 SHUTDOWN Mode Wakeup Logic*

In SHUTDOWN mode, the entire regulated core supply of the device is disabled and the device wakes only from a wake-capable IO that is configured for wakeup, from NRST, or from a debug connection. The IO wake mechanism for exiting SHUTDOWN is managed by IOMUX and is level based. The 5V-tolerant open-drain IOs, high-drive IOs, and certain standard-drive IOs include the additional wakeup logic that can be used to wake the device from a SHUTDOWN operating mode upon a level match.

To configure a wake-capable IO for wakeup from SHUTDOWN mode:

1. Set the INENA bit to let the input state propagate from the IO to the wakeup logic.
2. Select the compare level to use for wake by setting or clearing the WCOMP bit in the PINCMx register corresponding to the targeted pin.
3. Enable wakeup by setting the WUEN bit in the PINCMx register corresponding to the targeted pin.

After the previous configuration, SHUTDOWN mode can be entered through the appropriate command in SYSCTL. Pins on the device that contain digital IO controlled by IOMUX retain their current state when the device enters into SHUTDOWN. While the digital IO state is latched upon entry into SHUTDOWN mode, the IOMUX configuration registers (all PINCMx registers) lose their contents as the regulated core supply is shut down.

Once the IOMUX is configured for wakeup, a level match on any wakeup-configured pin immediately triggers a wakeup event. Therefore, before entering SHUTDOWN mode, application code must disable and reenable WUEN to prevent unintended wakeup events. After SHUTDOWN mode is entered, a level match triggers the exit sequence from SHUTDOWN. When the device exits SHUTDOWN, a BOR-level reset occurs but the state of the digital IO remains latched through the reset, keeping the IO state that was present upon entry into SHUTDOWN. This state is held until the IO are released in SYSCTL. After the BOR, SYSCTL captures the cause of the reset as a SHUTDOWN exit so that software can identify this and take appropriate action to reconfigure the device.

If multiple pins were configured for wakeup from SHUTDOWN, application software can determine which wakeup-configured IO generated the wake by polling the WAKESTATE bit in any IOs that were enabled for wake before the SHUTDOWN exit.

Application software must apply the following process to restore the IO state upon exit from SHUTDOWN:

1. Check which IO triggered the wakeup from SHUTDOWN, if necessary, as follows:
   a. Reconfigure the PINCMx register corresponding to the IO to be tested for wakeup status and set the peripheral connect (PC) bit (the PC bit gates the WAKESTATE indication).



Copyright © 2026 Texas Instruments Incorporated

b. Test the WAKESTATE bit in the PINCMx register corresponding to the IO to be tested to determine if that particular IO received a WAKE status based on the previously configured WCOMP and WUEN configuration.

2. Reconfigure any remaining IOMUX PINCM registers to the correct states.
3. Reconfigure the peripherals that are connected to pins through IOMUX, and enable them.
4. Release the SHUTDOWN IO lock in SYSCTL.
5. Clear the WUEN bit in the PINCMx register to reset the WAKESTATE status.

**Note**

After waking from SHUTDOWN, if the WUEN bit not cleared and the shutdown release bit in SYSCTL is not set, then reentering SHUTDOWN results in an immediate wake event, because the WAKESTATE status was not cleared from the previous wake event.

### *8.2.5 Pullup/Pulldown Resistors*

Software enabled pullup/pulldown resistors are provided on most digital IO types, and are connected to VDD/VSS, respectively. The 5V tolerant open drain digital IO does not provide a pullup resistor due to the open drain configuration.

To enable the pullup or pulldown resistor on a digital IO, set the PIPU or PIPD bit, respectively, in the corresponding PINCMx register. To disable the pullup or pulldown resistor, clear the corresponding bit.

The pullup/pulldown resistors can be enabled at any time, and their configuration is independent from the peripheral function configuration. It is possible to enable a pullup/pulldown resistor while changing the selected peripheral function.

### *8.2.6 Drive Strength Control*

The high-drive and high-speed digital IO types have programmable drive strength (low drive and high drive). The default drive strength is low drive. Application software can request high drive by setting the DRV bit in the PINCMx register corresponding to the target digital IO. Drive strength control is not available for standard drive and open drain IO types.

The drive strength control is completely independent of the selected peripheral function (PF) and can be changed by application software at any time.

For detailed electrical specifications on the drive performance in each drive mode for a given IO, see the Digital IO parameters in the device-specific data sheet.

### *8.2.7 Hysteresis and Logic Level Control*

The 5V-tolerant open drain digital IOs provide a hysteresis and logic level control to enable operation in input mode with standard CMOS logic (hysteresis enabled, CMOS logic levels) and TTL logic (hysteresis disabled, TTL logic levels).

The default mode for the 5V-tolerant open drain digital IO is TLL mode (HYSTEN bit in the PINCMx register is cleared). To use a 5V-tolerant open drain digital IO in CMOS mode with hysteresis enabled, set the HYSTEN bit in the PINCMx register which corresponds to the targeted IO.

The input logic level differences between TTL mode (left) and CMOS mode (right) are shown in Figure 8-2.



Copyright © 2026 Texas Instruments Incorporated

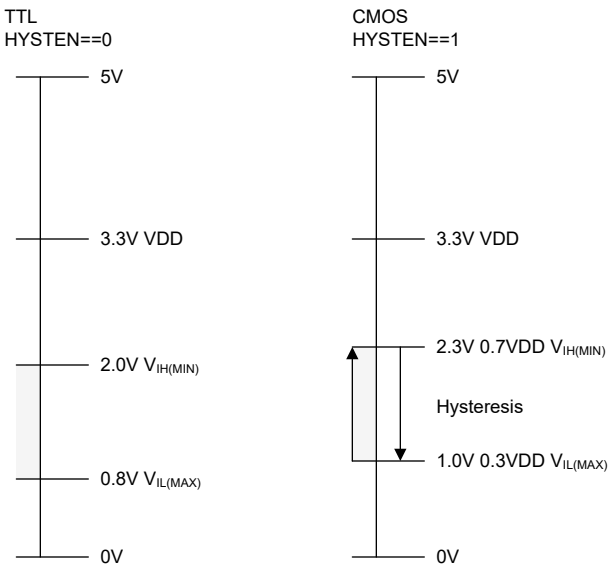



**Figure 8-2. Input Logic Levels - 5V Tolerant Open Drain Digital IO**



Copyright © 2026 Texas Instruments Incorporated

## 8.3 IOMUX Registers

Table 8-3 lists the memory-mapped registers for the IOMUX registers. All register offset addresses not listed in Table 8-3 should be considered as reserved locations and the register contents should not be modified.

**Table 8-3. IOMUX Registers**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 4h | PINCM | Pin Control Management Register in SECCFG region | | Go |

Complex bit access types are encoded to fit into small table cells. Table 8-4 shows the codes that are used for access types in this section.

**Table 8-4. IOMUX Access Type Codes**

| Access Type | Code | Description |
|---|---|---|
| **Read Type** | | |
| R | R | Read |
| **Write Type** | | |
| W | W | Write |
| **Reset or Default Value** | | |
| *-n* | | Value after reset or the default value |



Copyright © 2026 Texas Instruments Incorporated

### 8.3.1 PINCM (Offset = 4h) [Reset = X]

PINCM is shown in Figure 8-3 and described in Table 8-5.

Return to the Summary Table.

Pin Control Management Register

**Figure 8-3. PINCM**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | WCOMP | WUEN | INV | HIZ1 | RESERVED |
| R/W-0h | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | | RESERVED | DRV | HYSTEN | INENA | PIPU | PIPD |
| | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | WAKESTAT | RESERVED | | | | |
| R/W-0h | | R-0h | R/W-0h | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PC | RESERVED | PF | | | | | |
| R/W-0h | R/W-0h | R/W-0h | | | | | |

**Table 8-5. PINCM Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | RESERVED | R/W | 0h | |
| 28 | WCOMP | R/W | 0h | Wakeup Compare Value bit<br>0h = Wakeup on a match of 0<br>1h = Wakeup on a match of 1 |
| 27 | WUEN | R/W | 0h | Wakeup Enable bit<br>0h = wakeup is disabled.<br>1h = wakeup is enabled |
| 26 | INV | R/W | 0h | Data inversion selection<br>0h = Data inversion is disabled.<br>1h = Data inversion is enabled |
| 25 | HIZ1 | R/W | 0h | High output value will tri-state the output when this bit is enabled<br>0h = open-drain is disabled.<br>1h = open-drain is enabled. |
| 24 | RESERVED | R/W | 0h | |
| 21 | RESERVED | R/W | 0h | |
| 20 | DRV | R/W | 0h | Drive strength control selection, for HS IOCELL only<br>0h = Drive setting of 0 selected<br>1h = Drive setting of 1 selected |
| 19 | HYSTEN | R/W | 0h | Hysteresis Enable Control Selection<br>0h = hysteresis is disabled.<br>1h = hysteresis is enabled |
| 18 | INENA | R/W | 0h | Input Enable Control Selection<br>0h = Input enable is disabled.<br>1h = Input enable is enabled. |
| 17 | PIPU | R/W | 0h | Pull Up control selection<br>0h = Pull up is disabled.<br>1h = Pull up is enabled |
| 16 | PIPD | R/W | 0h | Pull Down control selection<br>0h = Pull down is disabled.<br>1h = Pull down is enabled |
| 15-14 | RESERVED | R/W | 0h | |



Copyright © 2026 Texas Instruments Incorporated

**Table 8-5. PINCM Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 13 | WAKESTAT | R | 0h | This has the IOPAD WAKEUP signal as status bit.<br>0h = wakeup source is NOT from this IOCELL<br>1h = wakeup source is from this IOCELL |
| 12-8 | RESERVED | R/W | 0h | |
| 7 | PC | R/W | 0h | Peripheral is “Connected”<br>0h = The output of the peripheral (and its output enable) will not propagate to the IOCELL<br>1h = The output latch of the dataflow will be “transparent” |
| 6 | RESERVED | R/W | 0h | |
| 5-0 | PF | R/W | 0h | Peripheral Function selection bits<br>0h = Reserved as unconnected<br>3Fh = An encoding per function that can be connected to this pin. |



Copyright © 2026 Texas Instruments Incorporated

Chapter 9

# General-Purpose Input/Output (GPIO)

The GPIO peripheral provides the user with a means to write data out and read data in to and from the device pins. The GPIO also provides a way to detect wakeup events while the device is in a low power state. This chapter describes the operation of the GPIO peripheral.


**9.1 GPIO Overview** ..... 662
**9.2 GPIO Operation** ..... 662
**9.3 GPIO Registers** ..... 666





Copyright © 2026 Texas Instruments Incorporated

## 9.1 GPIO Overview

The GPIO is used to read in digital data from the device pins and to send digital data out to the device pins.

The GPIO module features include:

- Zero wait state MMR access from CPU
- Set/clear/toggle multiple bits without the need of a read-modify-write construct in software
- Direct writes to individual GPIO output bits (DOUT) without the need of a read-modify-write construct in software
- Direct read comparisons of individual GPIO input bits (DIN) without the need to use masking in software
- DOUT serviceable by DMA to generate a predefined output sequence on specified pins
- "FastWake" feature enables low-power wakeup from STOP and STANDBY modes for any GPIO port
- User controlled input filtering (configurable per IO)
- Interconnection to the device event fabric through event publishers and event subscribers

GPIO Block Diagram shows the block diagram of the GPIO peripheral.

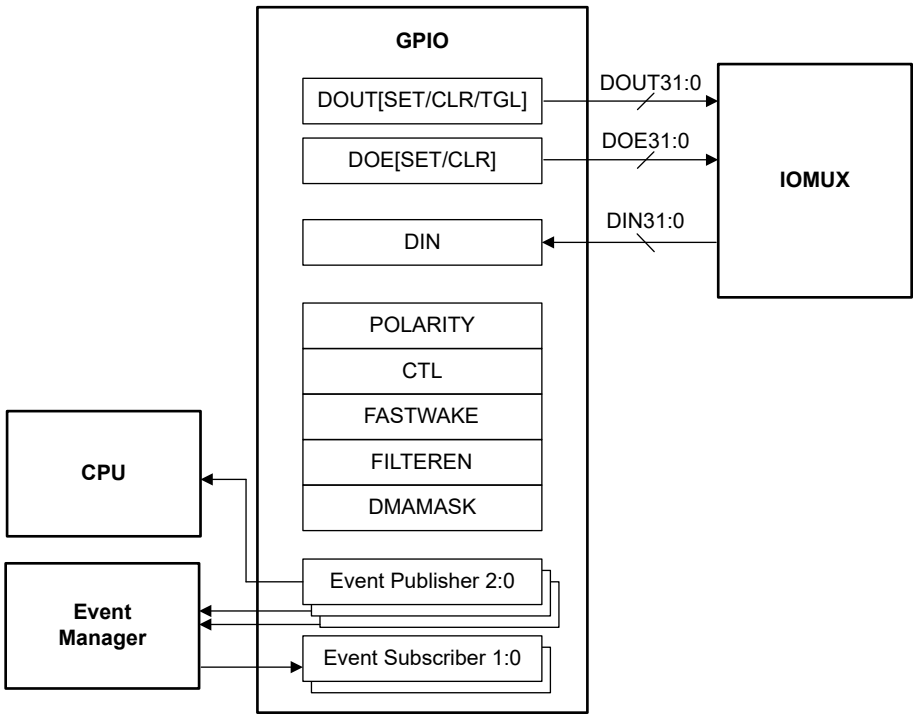



**Figure 9-1. GPIO Block Diagram**

**Note**

The GPIO module for the MSP platform does not manage the complete digital IO functionality (for example, pullup, pulldown, or other functional muxing). For complete digital IO control details, refer to Chapter 8. Similar to any other peripheral, the GPIO has inputs and outputs (with output enable) that allow the GPIO to interface with the IOMUX to make connections to the IO pins.

## 9.2 GPIO Operation

The GPIO peripheral is configured with user software. The setup and operation of the GPIO is discussed in the following sections.



Copyright © 2026 Texas Instruments Incorporated

### 9.2.1 GPIO Ports

An instance of the GPIO peripheral in the MSP platform supports up to 32 data input/output (DIO) bits. For devices with greater than 32 GPIOs, multiple instances of the GPIO peripheral are used to address all of the device pins. The GPIO port and bit names are directly mapped to the signal names associated with each device pin in the *Pin Configuration and Functions* section of the device data sheet.

**Table 9-1. GPIO Port and Device Pin Mapping**

| GPIO Port and Bit Name | Device Pin Signal Name |
|---|---|
| GPIO Port A Bit 0 (DIO0) | PA0 |
| GPIO Port A Bit 1 (DIO1) | PA1 |
| GPIO Port B Bit 0 (DIO0) | PB0 |
| GPIO Port B Bit 1 (DIO1) | PB1 |
| GPIO Port x Bit y (DIOy) | Pxy |

### 9.2.2 GPIO Read/Write Interface

The GPIO peripheral has features and dedicated registers to allow for advanced bit manipulations without the need to execute a read-modify-write construct in software. These features are outlined below.

### 9.2.3 GPIO Input Glitch Filtering and Synchronization

The GPIO module evaluates the state of input pins at the ULPCLK (PD0 bus clock) rate, synchronizing the pin state to ULPCLK through a 2-stage synchronizer before passing the GPIO state to the input glitch filter.

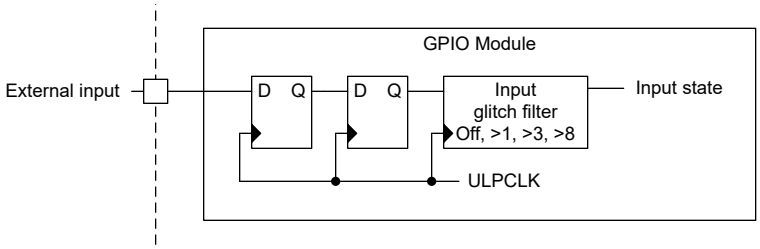



**Figure 9-2. GPIO Input Synchronizer**

A programmable input glitch filter is provided for suppressing noise on digital input pins. The glitch filter runs at ULPCLK rate. Four levels of user-specified input filtering are possible:

- Sampled input without filtering (the minimum reliably detected pulse width is one ULPCLK cycle due to synchronization of the pin state with ULPCLK +Delay time from edge of asynchronous request to first MCLK edge in case of fast wake enable for STANDBY0/1, STOP1/2 and SLEEP2 modes)
- Synchronized inputs which are not greater than 1 ULPCLK periods are filtered out
- Synchronized inputs which are not greater than 3 ULPCLK periods are filtered out
- Synchronized inputs which are not greater than 8 ULPCLK periods are filtered out

This feature allows users to easily implement input filtering in hardware for cases where fast switching on the input pin is needed to be filtered out. The bit fields in the FILTEREN31_16 and FILTEREN15_0 registers allow users to configure the level of filtering needed for the corresponding GPIO bit.

Input pulses of the same pulse length can be passed in some cases while being filtered in other cases, due to 1 ULPCLK cycle of uncertainty in the synchronization.



Copyright © 2026 Texas Instruments Incorporated

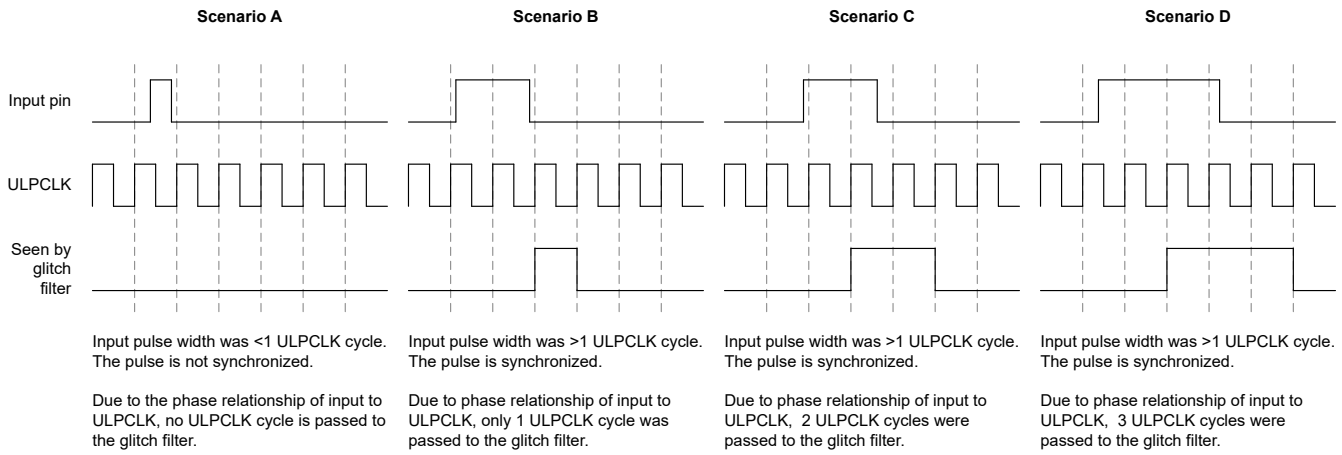



**Figure 9-3. GPIO Input Synchronizer and Glitch Filtering Scenarios**

- In Scenario A, the input pulse is less than one ULPCLK cycle. Pulses less than one ULPCLK cycle may not be captured. To ensure that GPIO inputs are always captured, the GPIO input pulse width must be greater than the ULPCLK period.
- In Scenario B, the input pulse is nearly two ULPCLK cycles in length, but because the rising edge occurs just after the ULPCLK edge, the GPIO synchronizer only views the input pin as having been high for 1 ULPCLK period. This scenario would not be filtered out by the glitch filter when the glitch filter is disabled, but a glitch filter value of >1 would result in this pulse being filtered. Conversely, the same input pulse width in Scenario C results in the input pin being considered high for two ULPCLK periods, as the rising edge occurred just before the ULPCLK edge. In this case, this scenario would not be filtered out when a glitch filter value of >1 is specified.
- In Scenario D, three ULPCLK cycles are passed to the glitch filter. In this case, the scenario would not be filtered out when a glitch filter value of >1 is specified, but it would be filtered out for a glitch filter value of >3 or >8.

**Note**

When the fast wake mode is enabled (SYSOSC is requested asynchronously upon input pin activity), the ULPCLK will switch from off (as would be the case in STANDBY1) or 32kHz (as would be the case in STANDBY0) to 24Mhz/32MHz, resulting in the input synchronization logic and glitch filter running at 24Mhz/32MHz after some latency. See the device specific data sheet for the asynchronous fast clock request wake time, and budget this time into any minimum pulse width calculations when using fast wake.

### *9.2.4 GPIO Fast Wake*

The fast wake feature in the MSP GPIO peripheral allows the GPIO module to stay in a low-power state and detect interrupt events on the device pins without requiring a high-speed clock. This allows the device to support fast wakeup from low-power modes, such as STOP and STANDBY, on any GPIO pin.

Fast wake can be enabled on a bit-wise basis using the FASTWAKE register. Setting a bit in the FASTWAKE register enables the corresponding device pin signal to support fast wakeup functionality. The CTL register contains a bit field named FASTWAKEONLY which allows for global control of the fast wake feature. Setting the FASTWAKEONLY bit enables all of the bits in the corresponding GPIO port to support fast wakeup functionality.

**Note**

Do not enable fast wake in the GPIO while simultaneously blocking asynchronous fast clock requests in SYSCTL. When fast wake is enabled, the GPIO expects to handshake with SYSCTL for the fast clock. If SYSCTL ignores the request, the GPIO does not receive a clock until SYSCTL completes the asynchronous fast clock request handshake.



Copyright © 2026 Texas Instruments Incorporated

### 9.2.5 GPIO DMA Interface

The GPIO peripheral allows the DMA write-access to the DOUT31_0 register. This functionality allows users to generate predefined output sequences on specified device pins. Some applications require preloaded sequences of GPIO pin changes and the MSP platform allows for the DMA to run that sequence so that the CPU can remain asleep and conserve energy.

The DMAMASK register is used to indicate which GPIO bits the DMA is allowed to modify. Setting a bit in the DMAMASK register enables the corresponding DOUT bit to be modified by the DMA.

**Note**

The CPU can write to any DOUT31_0 bit regardless of the DMAMASK value.

In cases where the DMA and the CPU both attempt to access and modify the DOUT31_0 register concurrently, it is the user's responsibility to manage the DMA and CPU bus transactions that are targeting the same bit to be modified.

- If a DMAMASK bit is set, the DMA will be prioritized to modify the corresponding DOUT bit.
- If a DMAMASK bit is cleared, the CPU will be prioritized to modify the corresponding DOUT bit.

### 9.2.6 Event Publishers and Subscribers

There are three independent event publishers available for GPIOx peripherals:

1. First Event Publisher (CPU_INT)
   - Used for generating CPU interrupt
   - Interrupt (RIS) flags are cleared upon software reading the IIDX register or writing to the respective ICLR register bits.
   - An event to the CPU can be individually specified for each GPIO bit through the POLARITY register:
     - 0: Disabled
     - 1: Rise Event
     - 2: Fall Event
     - 3: Rise or Fall Event
2. Second Event Publisher (GEN_EVENT0)
   - Uses the same POLARITY register definition as CPU_INT
   - Applies to GPIO bits 15 down to 0 (DIO15:0)
3. Third Event Publisher (GEN_EVENT1),
   - Uses the same POLARITY register definition as CPU_INT
   - Applies to GPIO bits 31 down to 16 (DIO31:16)

There are **two event subscribers**

1. First Event Subscriber (FSUB_0)
   - A specific pin can be directed to change state on an event
   - A subscriber event can only cause one single bit to have an action
   - Applies to GPIO bits 15 down to 0 (DIO15:0)
   - SUB0CFG register is used to enable the FSUB_0 event and define the output policy for a specific GPIO pin
2. Second Event Subscriber (FSUB_1)
   - A specific pin can be directed to change state on an event
   - A subscriber event can only cause one single bit to have an action
   - Applies to GPIO bits 31 down to 16 (DIO31:16)
   - SUB1CFG register is used to enable the FSUB_1 event and define the output policy for a specific GPIO pin



Copyright © 2026 Texas Instruments Incorporated

## 9.3 GPIO Registers

Table 9-2 lists the memory-mapped registers for the GPIO registers. All register offset addresses not listed in Table 9-2 should be considered as reserved locations and the register contents should not be modified.

**Table 9-2. GPIO Registers**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 400h | FSUB_0 | Subsciber Port 0 | | Go |
| 404h | FSUB_1 | Subscriber Port 1 | | Go |
| 444h | FPUB_0 | Publisher Port 0 | | Go |
| 448h | FPUB_1 | Publisher Port 1 | | Go |
| 800h | PWREN | Power enable | | Go |
| 804h | RSTCTL | Reset Control | | Go |
| 814h | STAT | Status Register | | Go |
| 1010h | CLKOVR | Clock Override | | Go |
| 1018h | PDBGCTL | Peripheral Debug Control | | Go |
| 1020h | IIDX | Interrupt index | CPU_INT | Go |
| 1028h | IMASK | Interrupt mask | CPU_INT | Go |
| 1030h | RIS | Raw interrupt status | CPU_INT | Go |
| 1038h | MIS | Masked interrupt status | CPU_INT | Go |
| 1040h | ISET | Interrupt set | CPU_INT | Go |
| 1048h | ICLR | Interrupt clear | CPU_INT | Go |
| 1050h | IIDX | Interrupt index | GEN_EVENT 0 | Go |
| 1058h | IMASK | Interrupt mask | GEN_EVENT 0 | Go |
| 1060h | RIS | Raw interrupt status | GEN_EVENT 0 | Go |
| 1068h | MIS | Masked interrupt status | GEN_EVENT 0 | Go |
| 1070h | ISET | Interrupt set | GEN_EVENT 0 | Go |
| 1078h | ICLR | Interrupt clear | GEN_EVENT 0 | Go |
| 1080h | IIDX | Interrupt index | GEN_EVENT 1 | Go |
| 1088h | IMASK | Interrupt mask | GEN_EVENT 1 | Go |
| 1090h | RIS | Raw interrupt status | GEN_EVENT 1 | Go |
| 1098h | MIS | Masked interrupt status | GEN_EVENT 1 | Go |
| 10A0h | ISET | Interrupt set | GEN_EVENT 1 | Go |
| 10A8h | ICLR | Interrupt clear | GEN_EVENT 1 | Go |
| 10E0h | EVT_MODE | Event Mode | | Go |
| 10FCh | DESC | Module Description | | Go |
| 1200h | DOUT3_0 | Data output 3 to 0 | | Go |
| 1204h | DOUT7_4 | Data output 7 to 4 | | Go |
| 1208h | DOUT11_8 | Data output 11 to 8 | | Go |
| 120Ch | DOUT15_12 | Data output 15 to 12 | | Go |



Copyright © 2026 Texas Instruments Incorporated

**Table 9-2. GPIO Registers (continued)**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 1210h | DOUT19_16 | Data output 19 to 16 | | Go |
| 1214h | DOUT23_20 | Data output 23 to 20 | | Go |
| 1218h | DOUT27_24 | Data output 27 to 24 | | Go |
| 121Ch | DOUT31_28 | Data output 31 to 28 | | Go |
| 1280h | DOUT31_0 | Data output 31 to 0 | | Go |
| 1290h | DOUTSET31_0 | Data output set 31 to 0 | | Go |
| 12A0h | DOUTCLR31_0 | Data output clear 31 to 0 | | Go |
| 12B0h | DOUTTGL31_0 | Data output toggle 31 to 0 | | Go |
| 12C0h | DOE31_0 | Data output enable 31 to 0 | | Go |
| 12D0h | DOESET31_0 | Data output enable set 31 to 0 | | Go |
| 12E0h | DOECLR31_0 | Data output enable clear 31 to 0 | | Go |
| 1300h | DIN3_0 | Data input 3 to 0 | | Go |
| 1304h | DIN7_4 | Data input 7 to 4 | | Go |
| 1308h | DIN11_8 | Data input 11 to 8 | | Go |
| 130Ch | DIN15_12 | Data input 15 to 12 | | Go |
| 1310h | DIN19_16 | Data input 19 to 16 | | Go |
| 1314h | DIN23_20 | Data input 23 to 20 | | Go |
| 1318h | DIN27_24 | Data input 27 to 24 | | Go |
| 131Ch | DIN31_28 | Data input 31 to 28 | | Go |
| 1380h | DIN31_0 | Data input 31 to 0 | | Go |
| 1390h | POLARITY15_0 | Polarity 15 to 0 | | Go |
| 13A0h | POLARITY31_16 | Polarity 31 to 16 | | Go |
| 1400h | CTL | FAST WAKE GLOBAL EN | | Go |
| 1404h | FASTWAKE | FAST WAKE ENABLE | | Go |
| 1500h | SUB0CFG | Subscriber 0 configuration | | Go |
| 1508h | FILTEREN15_0 | Filter Enable 15 to 0 | | Go |
| 150Ch | FILTEREN31_16 | Filter Enable 31 to 16 | | Go |
| 1510h | DMAMASK | DMA Write MASK | | Go |
| 1520h | SUB1CFG | Subscriber 1 configuration | | Go |

Complex bit access types are encoded to fit into small table cells. Table 9-3 shows the codes that are used for access types in this section.

**Table 9-3. GPIO Access Type Codes**

| Access Type | Code | Description |
|---|---|---|
| **Read Type** | | |
| R | R | Read |
| **Write Type** | | |
| K | K | Write protected by a key |
| W | W | Write |
| WK | W<br>K | Write<br>Write protected by a key |
| **Reset or Default Value** | | |
| *-n* | | Value after reset or the default value |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.1 FSUB_0 (Offset = 400h) [Reset = 00000000h]

FSUB_0 is shown in Figure 9-4 and described in Table 9-4.

Return to the Summary Table.

Subscriber port

**Figure 9-4. FSUB_0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | CHANID | | | |
| R/W-0h | | | | | | | | | | | | R/W-0h | | | |

**Table 9-4. FSUB_0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R/W | 0h | |
| 3-0 | CHANID | R/W | 0h | 0 = disconnected.<br>1-15 = connected to channelID = CHANID.<br>0h = A value of 0 specifies that the event is not connected<br>Fh = Consult your device data sheet as the actual allowed maximum may be less than 15. |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.2 FSUB_1 (Offset = 404h) [Reset = 00000000h]

FSUB_1 is shown in Figure 9-5 and described in Table 9-5.

Return to the Summary Table.

Subscriber port

**Figure 9-5. FSUB_1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | CHANID | | | |
| R/W-0h | | | | | | | | | | | | R/W-0h | | | |

**Table 9-5. FSUB_1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R/W | 0h | |
| 3-0 | CHANID | R/W | 0h | 0 = disconnected.<br>1-15 = connected to channelID = CHANID.<br>0h = A value of 0 specifies that the event is not connected<br>Fh = Consult your device data sheet as the actual allowed maximum may be less than 15. |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.3 FPUB_0 (Offset = 444h) [Reset = 00000000h]

FPUB_0 is shown in Figure 9-6 and described in Table 9-6.

Return to the Summary Table.

Publisher port

**Figure 9-6. FPUB_0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | CHANID | | | |
| R/W-0h | | | | | | | | | | | | R/W-0h | | | |

**Table 9-6. FPUB_0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R/W | 0h | |
| 3-0 | CHANID | R/W | 0h | 0 = disconnected.<br>1-15 = connected to channelID = CHANID.<br>0h = A value of 0 specifies that the event is not connected<br>Fh = Consult your device data sheet as the actual allowed maximum may be less than 15. |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.4 FPUB_1 (Offset = 448h) [Reset = 00000000h]

FPUB_1 is shown in Figure 9-7 and described in Table 9-7.

Return to the Summary Table.

Publisher port

**Figure 9-7. FPUB_1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | CHANID | | | |
| R/W-0h | | | | | | | | | | | | R/W-0h | | | |

**Table 9-7. FPUB_1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R/W | 0h | |
| 3-0 | CHANID | R/W | 0h | 0 = disconnected.<br>1-15 = connected to channelID = CHANID.<br>0h = A value of 0 specifies that the event is not connected<br>Fh = Consult your device data sheet as the actual allowed maximum may be less than 15. |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.5 PWREN (Offset = 800h) [Reset = 00000000h]

PWREN is shown in Figure 9-8 and described in Table 9-8.

Return to the Summary Table.

Register to control the power state

**Figure 9-8. PWREN**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R/W- | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R/W- | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | ENABLE |
| R/W- | | | | | | | K-0h |

**Table 9-8. PWREN Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | KEY to allow Power State Change<br>26h = KEY to allow write access to this register |
| 23-1 | RESERVED | R/W | 0h | |
| 0 | ENABLE | K | 0h | Enable the power<br>KEY must be set to 26h to write to this bit.<br>0h = Disable Power<br>1h = Enable Power |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.6 RSTCTL (Offset = 804h) [Reset = 00000000h]

RSTCTL is shown in Figure 9-9 and described in Table 9-9.

Return to the Summary Table.

Register to control reset assertion and de-assertion

**Figure 9-9. RSTCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| W- | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| W- | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | RESETSTKYCLR | RESETASSERT |
| W- | | | | | | WK-0h | WK-0h |

**Table 9-9. RSTCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | Unlock key<br>B1h = KEY to allow write access to this register |
| 23-2 | RESERVED | W | 0h | |
| 1 | RESETSTKYCLR | WK | 0h | Clear the RESETSTKY bit in the STAT register<br>KEY must be set to B1h to write to this bit.<br>0h = Writing 0 has no effect<br>1h = Clear reset sticky bit |
| 0 | RESETASSERT | WK | 0h | Assert reset to the peripheral<br>KEY must be set to B1h to write to this bit.<br>0h = Writing 0 has no effect<br>1h = Assert reset |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.7 STAT (Offset = 814h) [Reset = 00000000h]

STAT is shown in Figure 9-10 and described in Table 9-10.

Return to the Summary Table.

peripheral enable and reset status

**Figure 9-10. STAT**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R- | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | RESETSTKY |
| R- | | | | | | | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R- | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | |
| R- | | | | | | | |

**Table 9-10. STAT Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-17 | RESERVED | R | 0h | |
| 16 | RESETSTKY | R | 0h | This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register<br>0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register<br>1h = The peripheral was reset since the last bit clear |
| 15-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.8 CLKOVR (Offset = 1010h) [Reset = 00000000h]

CLKOVR is shown in Figure 9-11 and described in Table 9-11.

Return to the Summary Table.

This register overrides the functional clock request by this peripheral to the system

**Figure 9-11. CLKOVR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | RUN_STOP | OVERRIDE |
| R/W-0h | | | | | | R/W-0h | R/W-0h |

**Table 9-11. CLKOVR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | R/W | 0h | |
| 1 | RUN_STOP | R/W | 0h | If OVERRIDE is enabled, this register is used to manually control the peripheral's clock request to the system<br>0h = Run/ungate functional clock<br>1h = Stop/gate functional clock |
| 0 | OVERRIDE | R/W | 0h | Unlocks the functionality of RUN_STOP to override the automatic peripheral clock request<br>0h = Override disabled<br>1h = Override enabled |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.9 PDBGCTL (Offset = 1018h) [Reset = 00000001h]

PDBGCTL is shown in Figure 9-12 and described in Table 9-12.

Return to the Summary Table.

This register can be used by the software developer to control the behavior of the peripheral relative to the 'Core Halted' input

**Figure 9-12. PDBGCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | FREE |
| R/W-0h | | | | | | | R/W-1h |

**Table 9-12. PDBGCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R/W | 0h | |
| 0 | FREE | R/W | 1h | Free run control<br>0h = The peripheral freezes functionality while the Core Halted input is asserted and resumes when it is deasserted.<br>1h = The peripheral ignores the state of the Core Halted input |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.10 IIDX (Offset = 1020h) [Reset = 00000000h]

IIDX is shown in Figure 9-13 and described in Table 9-13.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. Value 0x00 means no event pending. Interrupt 1 is the highest priority, IIDX next highest, 4, 8, … IIDX^31 is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred. On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in [RIS] and [MIS] are cleared as well. After a read from the CPU (not from the debug interface), the register is updated with the next highest priority interrupt, if none are pending, then it should display 0x0.

**Figure 9-13. IIDX**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | STAT | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | |

**Table 9-13. IIDX Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | STAT | R | 0h | Interrupt index status<br>0h = No bit is set means there is no pending interrupt request<br>1h = DIO0 interrupt<br>2h = DIO1 interrupt<br>3h = DIO2 interrupt<br>4h = DIO3 interrupt<br>5h = DIO4 interrupt<br>6h = DIO5 interrupt<br>7h = DIO6 interrupt<br>8h = DIO7 interrupt<br>9h = DIO8 interrupt<br>Ah = DIO9 interrupt<br>Bh = DIO10 interrupt<br>Ch = DIO11 interrupt<br>Dh = DIO12 interrupt<br>Eh = DIO13 interrupt<br>Fh = DIO14 interrupt<br>10h = DIO15 interrupt<br>11h = DIO16 interrupt<br>12h = DIO17 interrupt<br>13h = DIO18 interrupt<br>14h = DIO19 interrupt<br>15h = DIO20 interrupt<br>16h = DIO21 interrupt<br>17h = DIO22 interrupt<br>18h = DIO23 interrupt<br>19h = DIO24 interrupt<br>1Ah = DIO25 interrupt<br>1Bh = DIO26 interrupt<br>1Ch = DIO27 interrupt<br>1Dh = DIO28 interrupt<br>1Eh = DIO29 interrupt<br>1Fh = DIO30 interrupt<br>20h = DIO31 interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.11 IMASK (Offset = 1028h) [Reset = 00000000h]

IMASK is shown in Figure 9-14 and described in Table 9-14.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is unmasked. Unmasking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

**Figure 9-14. IMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DIO31 | DIO30 | DIO29 | DIO28 | DIO27 | DIO26 | DIO25 | DIO24 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DIO23 | DIO22 | DIO21 | DIO20 | DIO19 | DIO18 | DIO17 | DIO16 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DIO15 | DIO14 | DIO13 | DIO12 | DIO11 | DIO10 | DIO9 | DIO8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DIO7 | DIO6 | DIO5 | DIO4 | DIO3 | DIO2 | DIO1 | DIO0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

**Table 9-14. IMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | DIO31 | R/W | 0h | DIO31 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 30 | DIO30 | R/W | 0h | DIO30 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 29 | DIO29 | R/W | 0h | DIO29 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 28 | DIO28 | R/W | 0h | DIO28 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 27 | DIO27 | R/W | 0h | DIO27 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 26 | DIO26 | R/W | 0h | DIO26 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 25 | DIO25 | R/W | 0h | DIO25 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 24 | DIO24 | R/W | 0h | DIO24 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 23 | DIO23 | R/W | 0h | DIO23 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 22 | DIO22 | R/W | 0h | DIO22 event mask<br>0h = Event is masked<br>1h = Event is unmasked |



Copyright © 2026 Texas Instruments Incorporated

**Table 9-14. IMASK Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 21 | DIO21 | R/W | 0h | DIO21 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 20 | DIO20 | R/W | 0h | DIO20 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 19 | DIO19 | R/W | 0h | DIO19 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 18 | DIO18 | R/W | 0h | DIO18 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 17 | DIO17 | R/W | 0h | DIO17 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 16 | DIO16 | R/W | 0h | DIO16 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 15 | DIO15 | R/W | 0h | DIO15 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 14 | DIO14 | R/W | 0h | DIO14 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 13 | DIO13 | R/W | 0h | DIO13 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 12 | DIO12 | R/W | 0h | DIO12 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 11 | DIO11 | R/W | 0h | DIO11 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 10 | DIO10 | R/W | 0h | DIO10 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 9 | DIO9 | R/W | 0h | DIO9 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 8 | DIO8 | R/W | 0h | DIO8 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 7 | DIO7 | R/W | 0h | DIO7 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 6 | DIO6 | R/W | 0h | DIO6 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 5 | DIO5 | R/W | 0h | DIO5 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 4 | DIO4 | R/W | 0h | DIO4 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 3 | DIO3 | R/W | 0h | DIO3 event mask<br>0h = Event is masked<br>1h = Event is unmasked |



Copyright © 2026 Texas Instruments Incorporated

**Table 9-14. IMASK Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | DIO2 | R/W | 0h | DIO2 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 1 | DIO1 | R/W | 0h | DIO1 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 0 | DIO0 | R/W | 0h | DIO0 event mask<br>0h = Event is masked<br>1h = Event is unmasked |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.12 RIS (Offset = 1030h) [Reset = 00000000h]

RIS is shown in Figure 9-15 and described in Table 9-15.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Figure 9-15. RIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DIO31 | DIO30 | DIO29 | DIO28 | DIO27 | DIO26 | DIO25 | DIO24 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DIO23 | DIO22 | DIO21 | DIO20 | DIO19 | DIO18 | DIO17 | DIO16 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DIO15 | DIO14 | DIO13 | DIO12 | DIO11 | DIO10 | DIO9 | DIO8 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DIO7 | DIO6 | DIO5 | DIO4 | DIO3 | DIO2 | DIO1 | DIO0 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 9-15. RIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | DIO31 | R | 0h | DIO31 event<br>0h = DIO31 event did not occur<br>1h = DIO31 event occurred |
| 30 | DIO30 | R | 0h | DIO30 event<br>0h = DIO30 event did not occur<br>1h = DIO30 event occurred |
| 29 | DIO29 | R | 0h | DIO29 event<br>0h = DIO29 event did not occur<br>1h = DIO29 event occurred |
| 28 | DIO28 | R | 0h | DIO28 event<br>0h = DIO28 event did not occur<br>1h = DIO28 event occurred |
| 27 | DIO27 | R | 0h | DIO27 event<br>0h = DIO27 event did not occur<br>1h = DIO27 event occurred |
| 26 | DIO26 | R | 0h | DIO26 event<br>0h = DIO26 event did not occur<br>1h = DIO26 event occurred |
| 25 | DIO25 | R | 0h | DIO25 event<br>0h = DIO25 event did not occur<br>1h = DIO25 event occurred |
| 24 | DIO24 | R | 0h | DIO24 event<br>0h = DIO24 event did not occur<br>1h = DIO24 event occurred |
| 23 | DIO23 | R | 0h | DIO23 event<br>0h = DIO23 event did not occur<br>1h = DIO23 event occurred |
| 22 | DIO22 | R | 0h | DIO22 event<br>0h = DIO22 event did not occur<br>1h = DIO22 event occurred |



Copyright © 2026 Texas Instruments Incorporated

**Table 9-15. RIS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 21 | DIO21 | R | 0h | DIO21 event<br>0h = DIO21 event did not occur<br>1h = DIO21 event occurred |
| 20 | DIO20 | R | 0h | DIO20 event<br>0h = DIO20 event did not occur<br>1h = DIO20 event occurred |
| 19 | DIO19 | R | 0h | DIO19 event<br>0h = DIO19 event did not occur<br>1h = DIO19 event occurred |
| 18 | DIO18 | R | 0h | DIO18 event<br>0h = DIO18 event did not occur<br>1h = DIO18 event occurred |
| 17 | DIO17 | R | 0h | DIO17 event<br>0h = DIO17 event did not occur<br>1h = DIO17 event occurred |
| 16 | DIO16 | R | 0h | DIO16 event<br>0h = DIO16 event did not occur<br>1h = DIO16 event occurred |
| 15 | DIO15 | R | 0h | DIO15 event<br>0h = DIO15 event did not occur<br>1h = DIO15 event occurred |
| 14 | DIO14 | R | 0h | DIO14 event<br>0h = DIO14 event did not occur<br>1h = DIO14 event occurred |
| 13 | DIO13 | R | 0h | DIO13 event<br>0h = DIO13 event did not occur<br>1h = DIO13 event occurred |
| 12 | DIO12 | R | 0h | DIO12 event<br>0h = DIO12 event did not occur<br>1h = DIO12 event occurred |
| 11 | DIO11 | R | 0h | DIO11 event<br>0h = DIO11 event did not occur<br>1h = DIO11 event occurred |
| 10 | DIO10 | R | 0h | DIO10 event<br>0h = DIO10 event did not occur<br>1h = DIO10 event occurred |
| 9 | DIO9 | R | 0h | DIO9 event<br>0h = DIO9 event did not occur<br>1h = DIO9 event occurred |
| 8 | DIO8 | R | 0h | DIO8 event<br>0h = DIO8 event did not occur<br>1h = DIO8 event occurred |
| 7 | DIO7 | R | 0h | DIO7 event<br>0h = DIO7 event did not occur<br>1h = DIO7 event occurred |
| 6 | DIO6 | R | 0h | DIO6 event<br>0h = DIO6 event did not occur<br>1h = DIO6 event occurred |
| 5 | DIO5 | R | 0h | DIO5 event<br>0h = DIO5 event did not occur<br>1h = DIO5 event occurred |
| 4 | DIO4 | R | 0h | DIO4 event<br>0h = DIO4 event did not occur<br>1h = DIO4 event occurred |
| 3 | DIO3 | R | 0h | DIO3 event<br>0h = DIO3 event did not occur<br>1h = DIO3 event occurred |



Copyright © 2026 Texas Instruments Incorporated

**Table 9-15. RIS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | DIO2 | R | 0h | DIO2 event<br>0h = DIO2 event did not occur<br>1h = DIO2 event occurred |
| 1 | DIO1 | R | 0h | DIO1 event<br>0h = DIO1 event did not occur<br>1h = DIO1 event occurred |
| 0 | DIO0 | R | 0h | DIO0 event<br>0h = DIO0 event did not occur<br>1h = DIO0 event occurred |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.13 MIS (Offset = 1038h) [Reset = 00000000h]

MIS is shown in Figure 9-16 and described in Table 9-16.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 9-16. MIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DIO31 | DIO30 | DIO29 | DIO28 | DIO27 | DIO26 | DIO25 | DIO24 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DIO23 | DIO22 | DIO21 | DIO20 | DIO19 | DIO18 | DIO17 | DIO16 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DIO15 | DIO14 | DIO13 | DIO12 | DIO11 | DIO10 | DIO9 | DIO8 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DIO7 | DIO6 | DIO5 | DIO4 | DIO3 | DIO2 | DIO1 | DIO0 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 9-16. MIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | DIO31 | R | 0h | DIO31 event<br>0h = DIO31 event did not occur<br>1h = DIO31 event occurred |
| 30 | DIO30 | R | 0h | DIO30 event<br>0h = DIO30 event did not occur<br>1h = DIO30 event occurred |
| 29 | DIO29 | R | 0h | DIO29 event<br>0h = DIO29 event did not occur<br>1h = DIO29 event occurred |
| 28 | DIO28 | R | 0h | DIO28 event<br>0h = DIO28 event did not occur<br>1h = DIO28 event occurred |
| 27 | DIO27 | R | 0h | DIO27 event<br>0h = DIO27 event did not occur<br>1h = DIO27 event occurred |
| 26 | DIO26 | R | 0h | DIO26 event<br>0h = DIO26 event did not occur<br>1h = DIO26 event occurred |
| 25 | DIO25 | R | 0h | DIO25 event<br>0h = DIO25 event did not occur<br>1h = DIO25 event occurred |
| 24 | DIO24 | R | 0h | DIO24 event<br>0h = DIO24 event did not occur<br>1h = DIO24 event occurred |
| 23 | DIO23 | R | 0h | DIO23 event<br>0h = DIO23 event did not occur<br>1h = DIO23 event occurred |
| 22 | DIO22 | R | 0h | DIO22 event<br>0h = DIO22 event did not occur<br>1h = DIO22 event occurred |



Copyright © 2026 Texas Instruments Incorporated

**Table 9-16. MIS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 21 | DIO21 | R | 0h | DIO21 event<br>0h = DIO21 event did not occur<br>1h = DIO21 event occurred |
| 20 | DIO20 | R | 0h | DIO20 event<br>0h = DIO20 event did not occur<br>1h = DIO20 event occurred |
| 19 | DIO19 | R | 0h | DIO19 event<br>0h = DIO19 event did not occur<br>1h = DIO19 event occurred |
| 18 | DIO18 | R | 0h | DIO18 event<br>0h = DIO18 event did not occur<br>1h = DIO18 event occurred |
| 17 | DIO17 | R | 0h | DIO17 event<br>0h = DIO17 event did not occur<br>1h = DIO17 event occurred |
| 16 | DIO16 | R | 0h | DIO16 event<br>0h = DIO16 event did not occur<br>1h = DIO16 event occurred |
| 15 | DIO15 | R | 0h | DIO15 event<br>0h = DIO15 event did not occur<br>1h = DIO15 event occurred |
| 14 | DIO14 | R | 0h | DIO14 event<br>0h = DIO14 event did not occur<br>1h = DIO14 event occurred |
| 13 | DIO13 | R | 0h | DIO13 event<br>0h = DIO13 event did not occur<br>1h = DIO13 event occurred |
| 12 | DIO12 | R | 0h | DIO12 event<br>0h = DIO12 event did not occur<br>1h = DIO12 event occurred |
| 11 | DIO11 | R | 0h | DIO11 event<br>0h = DIO11 event did not occur<br>1h = DIO11 event occurred |
| 10 | DIO10 | R | 0h | DIO10 event<br>0h = DIO10 event did not occur<br>1h = DIO10 event occurred |
| 9 | DIO9 | R | 0h | DIO9 event<br>0h = DIO9 event did not occur<br>1h = DIO9 event occurred |
| 8 | DIO8 | R | 0h | DIO8 event<br>0h = DIO8 event did not occur<br>1h = DIO8 event occurred |
| 7 | DIO7 | R | 0h | DIO7 event<br>0h = DIO7 event did not occur<br>1h = DIO7 event occurred |
| 6 | DIO6 | R | 0h | DIO6 event<br>0h = DIO6 event did not occur<br>1h = DIO6 event occurred |
| 5 | DIO5 | R | 0h | DIO5 event<br>0h = DIO5 event did not occur<br>1h = DIO5 event occurred |
| 4 | DIO4 | R | 0h | DIO4 event<br>0h = DIO4 event did not occur<br>1h = DIO4 event occurred |
| 3 | DIO3 | R | 0h | DIO3 event<br>0h = DIO3 event did not occur<br>1h = DIO3 event occurred |



Copyright © 2026 Texas Instruments Incorporated

**Table 9-16. MIS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | DIO2 | R | 0h | DIO2 event<br>0h = DIO2 event did not occur<br>1h = DIO2 event occurred |
| 1 | DIO1 | R | 0h | DIO1 event<br>0h = DIO1 event did not occur<br>1h = DIO1 event occurred |
| 0 | DIO0 | R | 0h | DIO0 event<br>0h = DIO0 event did not occur<br>1h = DIO0 event occurred |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.14 ISET (Offset = 1040h) [Reset = 00000000h]

ISET is shown in Figure 9-17 and described in Table 9-17.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 9-17. ISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DIO31 | DIO30 | DIO29 | DIO28 | DIO27 | DIO26 | DIO25 | DIO24 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DIO23 | DIO22 | DIO21 | DIO20 | DIO19 | DIO18 | DIO17 | DIO16 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DIO15 | DIO14 | DIO13 | DIO12 | DIO11 | DIO10 | DIO9 | DIO8 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DIO7 | DIO6 | DIO5 | DIO4 | DIO3 | DIO2 | DIO1 | DIO0 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

**Table 9-17. ISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | DIO31 | W | 0h | DIO31 event<br>0h = No effect<br>1h = Sets DIO31 in RIS register |
| 30 | DIO30 | W | 0h | DIO30 event<br>0h = No effect<br>1h = Sets DIO30 in RIS register |
| 29 | DIO29 | W | 0h | DIO29 event<br>0h = No effect<br>1h = Sets DIO29 in RIS register |
| 28 | DIO28 | W | 0h | DIO28 event<br>0h = No effect<br>1h = Sets DIO28 in RIS register |
| 27 | DIO27 | W | 0h | DIO27 event<br>0h = No effect<br>1h = Sets DIO27 in RIS register |
| 26 | DIO26 | W | 0h | DIO26 event<br>0h = No effect<br>1h = Sets DIO26 in RIS register |
| 25 | DIO25 | W | 0h | DIO25 event<br>0h = No effect<br>1h = Sets DIO25 in RIS register |
| 24 | DIO24 | W | 0h | DIO24 event<br>0h = No effect<br>1h = Sets DIO24 in RIS register |
| 23 | DIO23 | W | 0h | DIO23 event<br>0h = No effect<br>1h = Sets DIO23 in RIS register |
| 22 | DIO22 | W | 0h | DIO22 event<br>0h = No effect<br>1h = Sets DIO22 in RIS register |



Copyright © 2026 Texas Instruments Incorporated

**Table 9-17. ISET Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 21 | DIO21 | W | 0h | DIO21 event<br>0h = No effect<br>1h = Sets DIO21 in RIS register |
| 20 | DIO20 | W | 0h | DIO20 event<br>0h = No effect<br>1h = Sets DIO20 in RIS register |
| 19 | DIO19 | W | 0h | DIO19 event<br>0h = No effect<br>1h = Sets DIO19 in RIS register |
| 18 | DIO18 | W | 0h | DIO18 event<br>0h = No effect<br>1h = Sets DIO18 in RIS register |
| 17 | DIO17 | W | 0h | DIO17 event<br>0h = No effect<br>1h = Sets DIO17 in RIS register |
| 16 | DIO16 | W | 0h | DIO16 event<br>0h = No effect<br>1h = Sets DIO16 in RIS register |
| 15 | DIO15 | W | 0h | DIO15 event<br>0h = No effect<br>1h = Sets DIO15 in RIS register |
| 14 | DIO14 | W | 0h | DIO14 event<br>0h = No effect<br>1h = Sets DIO14 in RIS register |
| 13 | DIO13 | W | 0h | DIO13 event<br>0h = No effect<br>1h = Sets DIO13 in RIS register |
| 12 | DIO12 | W | 0h | DIO12 event<br>0h = No effect<br>1h = Sets DIO12 in RIS register |
| 11 | DIO11 | W | 0h | DIO11 event<br>0h = No effect<br>1h = Sets DIO11 in RIS register |
| 10 | DIO10 | W | 0h | DIO10 event<br>0h = No effect<br>1h = Sets DIO10 in RIS register |
| 9 | DIO9 | W | 0h | DIO9 event<br>0h = No effect<br>1h = Sets DIO9 in RIS register |
| 8 | DIO8 | W | 0h | DIO8 event<br>0h = No effect<br>1h = Sets DIO8 in RIS register |
| 7 | DIO7 | W | 0h | DIO7 event<br>0h = No effect<br>1h = Sets DIO7 in RIS register |
| 6 | DIO6 | W | 0h | DIO6 event<br>0h = No effect<br>1h = Sets DIO6 in RIS register |
| 5 | DIO5 | W | 0h | DIO5 event<br>0h = No effect<br>1h = Sets DIO5 in RIS register |
| 4 | DIO4 | W | 0h | DIO4 event<br>0h = No effect<br>1h = Sets DIO4 in RIS register |
| 3 | DIO3 | W | 0h | DIO3 event<br>0h = No effect<br>1h = Sets DIO3 in RIS register |



Copyright © 2026 Texas Instruments Incorporated

**Table 9-17. ISET Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | DIO2 | W | 0h | DIO2 event<br>0h = No effect<br>1h = Sets DIO2 in RIS register |
| 1 | DIO1 | W | 0h | DIO1 event<br>0h = No effect<br>1h = Sets DIO1 in RIS register |
| 0 | DIO0 | W | 0h | DIO0 event<br>0h = No effect<br>1h = Sets DIO0 in RIS register |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.15 ICLR (Offset = 1048h) [Reset = 00000000h]

ICLR is shown in Figure 9-18 and described in Table 9-18.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

**Figure 9-18. ICLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DIO31 | DIO30 | DIO29 | DIO28 | DIO27 | DIO26 | DIO25 | DIO24 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DIO23 | DIO22 | DIO21 | DIO20 | DIO19 | DIO18 | DIO17 | DIO16 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DIO15 | DIO14 | DIO13 | DIO12 | DIO11 | DIO10 | DIO9 | DIO8 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DIO7 | DIO6 | DIO5 | DIO4 | DIO3 | DIO2 | DIO1 | DIO0 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

**Table 9-18. ICLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | DIO31 | W | 0h | DIO31 event<br>0h = No effect<br>1h = Clears DIO31 in RIS register |
| 30 | DIO30 | W | 0h | DIO30 event<br>0h = No effect<br>1h = Clears DIO30 in RIS register |
| 29 | DIO29 | W | 0h | DIO29 event<br>0h = No effect<br>1h = Clears DIO29 in RIS register |
| 28 | DIO28 | W | 0h | DIO28 event<br>0h = No effect<br>1h = Clears DIO28 in RIS register |
| 27 | DIO27 | W | 0h | DIO27 event<br>0h = No effect<br>1h = Clears DIO27 in RIS register |
| 26 | DIO26 | W | 0h | DIO26 event<br>0h = No effect<br>1h = Clears DIO26 in RIS register |
| 25 | DIO25 | W | 0h | DIO25 event<br>0h = No effect<br>1h = Clears DIO25 in RIS register |
| 24 | DIO24 | W | 0h | DIO24 event<br>0h = No effect<br>1h = Clears DIO24 in RIS register |
| 23 | DIO23 | W | 0h | DIO23 event<br>0h = No effect<br>1h = Clears DIO23 in RIS register |
| 22 | DIO22 | W | 0h | DIO22 event<br>0h = No effect<br>1h = Clears DIO22 in RIS register |



Copyright © 2026 Texas Instruments Incorporated

**Table 9-18. ICLR Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 21 | DIO21 | W | 0h | DIO21 event<br>0h = No effect<br>1h = Clears DIO21 in RIS register |
| 20 | DIO20 | W | 0h | DIO20 event<br>0h = No effect<br>1h = Clears DIO20 in RIS register |
| 19 | DIO19 | W | 0h | DIO19 event<br>0h = No effect<br>1h = Clears DIO19 in RIS register |
| 18 | DIO18 | W | 0h | DIO18 event<br>0h = No effect<br>1h = Clears DIO18 in RIS register |
| 17 | DIO17 | W | 0h | DIO17 event<br>0h = No effect<br>1h = Clears DIO17 in RIS register |
| 16 | DIO16 | W | 0h | DIO16 event<br>0h = No effect<br>1h = Clears DIO16 in RIS register |
| 15 | DIO15 | W | 0h | DIO15 event<br>0h = No effect<br>1h = Clears DIO15 in RIS register |
| 14 | DIO14 | W | 0h | DIO14 event<br>0h = No effect<br>1h = Clears DIO14 in RIS register |
| 13 | DIO13 | W | 0h | DIO13 event<br>0h = No effect<br>1h = Clears DIO13 in RIS register |
| 12 | DIO12 | W | 0h | DIO12 event<br>0h = No effect<br>1h = Clears DIO12 in RIS register |
| 11 | DIO11 | W | 0h | DIO11 event<br>0h = No effect<br>1h = Clears DIO11 in RIS register |
| 10 | DIO10 | W | 0h | DIO10 event<br>0h = No effect<br>1h = Clears DIO10 in RIS register |
| 9 | DIO9 | W | 0h | DIO9 event<br>0h = No effect<br>1h = Clears DIO9 in RIS register |
| 8 | DIO8 | W | 0h | DIO8 event<br>0h = No effect<br>1h = Clears DIO8 in RIS register |
| 7 | DIO7 | W | 0h | DIO7 event<br>0h = No effect<br>1h = Clears DIO7 in RIS register |
| 6 | DIO6 | W | 0h | DIO6 event<br>0h = No effect<br>1h = Clears DIO6 in RIS register |
| 5 | DIO5 | W | 0h | DIO5 event<br>0h = No effect<br>1h = Clears DIO5 in RIS register |
| 4 | DIO4 | W | 0h | DIO4 event<br>0h = No effect<br>1h = Clears DIO4 in RIS register |
| 3 | DIO3 | W | 0h | DIO3 event<br>0h = No effect<br>1h = Clears DIO3 in RIS register |



Copyright © 2026 Texas Instruments Incorporated

**Table 9-18. ICLR Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | DIO2 | W | 0h | DIO2 event<br>0h = No effect<br>1h = Clears DIO2 in RIS register |
| 1 | DIO1 | W | 0h | DIO1 event<br>0h = No effect<br>1h = Clears DIO1 in RIS register |
| 0 | DIO0 | W | 0h | DIO0 event<br>0h = No effect<br>1h = Clears DIO0 in RIS register |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.16 IIDX (Offset = 1050h) [Reset = 00000000h]

IIDX is shown in Figure 9-19 and described in Table 9-19.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. Value 0x00 means no event pending. Interrupt 1 is the highest priority, IIDX next highest, 4, 8, … IIDX^31 is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred. On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in [RIS] and [MIS] are cleared as well. After a read from the CPU (not from the debug interface), the register is updated with the next highest priority interrupt, if none are pending, then it should display 0x0.

**Figure 9-19. IIDX**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | STAT | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | |

**Table 9-19. IIDX Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | STAT | R | 0h | Interrupt index status<br>0h = No bit is set means there is no pending interrupt request<br>1h = DIO0 interrupt<br>2h = DIO1 interrupt<br>3h = DIO2 interrupt<br>4h = DIO3 interrupt<br>5h = DIO4 interrupt<br>6h = DIO5 interrupt<br>7h = DIO6 interrupt<br>8h = DIO7 interrupt<br>9h = DIO8 interrupt<br>Ah = DIO9 interrupt<br>Bh = DIO10 interrupt<br>Ch = DIO11 interrupt<br>Dh = DIO12 interrupt<br>Eh = DIO13 interrupt<br>Fh = DIO14 interrupt<br>10h = DIO15 interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.17 IMASK (Offset = 1058h) [Reset = 00000000h]

IMASK is shown in Figure 9-20 and described in Table 9-20.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is unmasked. Unmasking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

**Figure 9-20. IMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DIO15 | DIO14 | DIO13 | DIO12 | DIO11 | DIO10 | DIO9 | DIO8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DIO7 | DIO6 | DIO5 | DIO4 | DIO3 | DIO2 | DIO1 | DIO0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

**Table 9-20. IMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R/W | 0h | |
| 15 | DIO15 | R/W | 0h | DIO15 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 14 | DIO14 | R/W | 0h | DIO14 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 13 | DIO13 | R/W | 0h | DIO13 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 12 | DIO12 | R/W | 0h | DIO12 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 11 | DIO11 | R/W | 0h | DIO11 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 10 | DIO10 | R/W | 0h | DIO10 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 9 | DIO9 | R/W | 0h | DIO9 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 8 | DIO8 | R/W | 0h | DIO8 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 7 | DIO7 | R/W | 0h | DIO7 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 6 | DIO6 | R/W | 0h | DIO6 event mask<br>0h = Event is masked<br>1h = Event is unmasked |



Copyright © 2026 Texas Instruments Incorporated

**Table 9-20. IMASK Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 5 | DIO5 | R/W | 0h | DIO5 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 4 | DIO4 | R/W | 0h | DIO4 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 3 | DIO3 | R/W | 0h | DIO3 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 2 | DIO2 | R/W | 0h | DIO2 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 1 | DIO1 | R/W | 0h | DIO1 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 0 | DIO0 | R/W | 0h | DIO0 event mask<br>0h = Event is masked<br>1h = Event is unmasked |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.18 RIS (Offset = 1060h) [Reset = 00000000h]

RIS is shown in Figure 9-21 and described in Table 9-21.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Figure 9-21. RIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DIO15 | DIO14 | DIO13 | DIO12 | DIO11 | DIO10 | DIO9 | DIO8 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DIO7 | DIO6 | DIO5 | DIO4 | DIO3 | DIO2 | DIO1 | DIO0 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 9-21. RIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15 | DIO15 | R | 0h | DIO15 event<br>0h = DIO15 event did not occur<br>1h = DIO15 event occurred |
| 14 | DIO14 | R | 0h | DIO14 event<br>0h = DIO14 event did not occur<br>1h = DIO14 event occurred |
| 13 | DIO13 | R | 0h | DIO13 event<br>0h = DIO13 event did not occur<br>1h = DIO13 event occurred |
| 12 | DIO12 | R | 0h | DIO12 event<br>0h = DIO12 event did not occur<br>1h = DIO12 event occurred |
| 11 | DIO11 | R | 0h | DIO11 event<br>0h = DIO11 event did not occur<br>1h = DIO11 event occurred |
| 10 | DIO10 | R | 0h | DIO10 event<br>0h = DIO10 event did not occur<br>1h = DIO10 event occurred |
| 9 | DIO9 | R | 0h | DIO9 event<br>0h = DIO9 event did not occur<br>1h = DIO9 event occurred |
| 8 | DIO8 | R | 0h | DIO8 event<br>0h = DIO8 event did not occur<br>1h = DIO8 event occurred |
| 7 | DIO7 | R | 0h | DIO7 event<br>0h = DIO7 event did not occur<br>1h = DIO7 event occurred |



Copyright © 2026 Texas Instruments Incorporated

**Table 9-21. RIS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 6 | DIO6 | R | 0h | DIO6 event<br>0h = DIO6 event did not occur<br>1h = DIO6 event occurred |
| 5 | DIO5 | R | 0h | DIO5 event<br>0h = DIO5 event did not occur<br>1h = DIO5 event occurred |
| 4 | DIO4 | R | 0h | DIO4 event<br>0h = DIO4 event did not occur<br>1h = DIO4 event occurred |
| 3 | DIO3 | R | 0h | DIO3 event<br>0h = DIO3 event did not occur<br>1h = DIO3 event occurred |
| 2 | DIO2 | R | 0h | DIO2 event<br>0h = DIO2 event did not occur<br>1h = DIO2 event occurred |
| 1 | DIO1 | R | 0h | DIO1 event<br>0h = DIO1 event did not occur<br>1h = DIO1 event occurred |
| 0 | DIO0 | R | 0h | DIO0 event<br>0h = DIO0 event did not occur<br>1h = DIO0 event occurred |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.19 MIS (Offset = 1068h) [Reset = 00000000h]

MIS is shown in Figure 9-22 and described in Table 9-22.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 9-22. MIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DIO15 | DIO14 | DIO13 | DIO12 | DIO11 | DIO10 | DIO9 | DIO8 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DIO7 | DIO6 | DIO5 | DIO4 | DIO3 | DIO2 | DIO1 | DIO0 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 9-22. MIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15 | DIO15 | R | 0h | DIO15 event<br>0h = DIO15 event did not occur<br>1h = DIO15 event occurred |
| 14 | DIO14 | R | 0h | DIO14 event<br>0h = DIO14 event did not occur<br>1h = DIO14 event occurred |
| 13 | DIO13 | R | 0h | DIO13 event<br>0h = DIO13 event did not occur<br>1h = DIO13 event occurred |
| 12 | DIO12 | R | 0h | DIO12 event<br>0h = DIO12 event did not occur<br>1h = DIO12 event occurred |
| 11 | DIO11 | R | 0h | DIO11 event<br>0h = DIO11 event did not occur<br>1h = DIO11 event occurred |
| 10 | DIO10 | R | 0h | DIO10 event<br>0h = DIO10 event did not occur<br>1h = DIO10 event occurred |
| 9 | DIO9 | R | 0h | DIO9 event<br>0h = DIO9 event did not occur<br>1h = DIO9 event occurred |
| 8 | DIO8 | R | 0h | DIO8 event<br>0h = DIO8 event did not occur<br>1h = DIO8 event occurred |
| 7 | DIO7 | R | 0h | DIO7 event<br>0h = DIO7 event did not occur<br>1h = DIO7 event occurred |
| 6 | DIO6 | R | 0h | DIO6 event<br>0h = DIO6 event did not occur<br>1h = DIO6 event occurred |



Copyright © 2026 Texas Instruments Incorporated

**Table 9-22. MIS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 5 | DIO5 | R | 0h | DIO5 event<br>0h = DIO5 event did not occur<br>1h = DIO5 event occurred |
| 4 | DIO4 | R | 0h | DIO4 event<br>0h = DIO4 event did not occur<br>1h = DIO4 event occurred |
| 3 | DIO3 | R | 0h | DIO3 event<br>0h = DIO3 event did not occur<br>1h = DIO3 event occurred |
| 2 | DIO2 | R | 0h | DIO2 event<br>0h = DIO2 event did not occur<br>1h = DIO2 event occurred |
| 1 | DIO1 | R | 0h | DIO1 event<br>0h = DIO1 event did not occur<br>1h = DIO1 event occurred |
| 0 | DIO0 | R | 0h | DIO0 event<br>0h = DIO0 event did not occur<br>1h = DIO0 event occurred |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.20 ISET (Offset = 1070h) [Reset = 00000000h]

ISET is shown in Figure 9-23 and described in Table 9-23.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 9-23. ISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DIO15 | DIO14 | DIO13 | DIO12 | DIO11 | DIO10 | DIO9 | DIO8 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DIO7 | DIO6 | DIO5 | DIO4 | DIO3 | DIO2 | DIO1 | DIO0 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

**Table 9-23. ISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | W | 0h | |
| 15 | DIO15 | W | 0h | DIO15 event<br>0h = No effect<br>1h = Sets DIO15 in RIS register |
| 14 | DIO14 | W | 0h | DIO14 event<br>0h = No effect<br>1h = Sets DIO14 in RIS register |
| 13 | DIO13 | W | 0h | DIO13 event<br>0h = No effect<br>1h = Sets DIO13 in RIS register |
| 12 | DIO12 | W | 0h | DIO12 event<br>0h = No effect<br>1h = Sets DIO12 in RIS register |
| 11 | DIO11 | W | 0h | DIO11 event<br>0h = No effect<br>1h = Sets DIO11 in RIS register |
| 10 | DIO10 | W | 0h | DIO10 event<br>0h = No effect<br>1h = Sets DIO10 in RIS register |
| 9 | DIO9 | W | 0h | DIO9 event<br>0h = No effect<br>1h = Sets DIO9 in RIS register |
| 8 | DIO8 | W | 0h | DIO8 event<br>0h = No effect<br>1h = Sets DIO8 in RIS register |
| 7 | DIO7 | W | 0h | DIO7 event<br>0h = No effect<br>1h = Sets DIO7 in RIS register |



Copyright © 2026 Texas Instruments Incorporated

**Table 9-23. ISET Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 6 | DIO6 | W | 0h | DIO6 event<br>0h = No effect<br>1h = Sets DIO6 in RIS register |
| 5 | DIO5 | W | 0h | DIO5 event<br>0h = No effect<br>1h = Sets DIO5 in RIS register |
| 4 | DIO4 | W | 0h | DIO4 event<br>0h = No effect<br>1h = Sets DIO4 in RIS register |
| 3 | DIO3 | W | 0h | DIO3 event<br>0h = No effect<br>1h = Sets DIO3 in RIS register |
| 2 | DIO2 | W | 0h | DIO2 event<br>0h = No effect<br>1h = Sets DIO2 in RIS register |
| 1 | DIO1 | W | 0h | DIO1 event<br>0h = No effect<br>1h = Sets DIO1 in RIS register |
| 0 | DIO0 | W | 0h | DIO0 event<br>0h = No effect<br>1h = Sets DIO0 in RIS register |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.21 ICLR (Offset = 1078h) [Reset = 00000000h]

ICLR is shown in Figure 9-24 and described in Table 9-24.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

**Figure 9-24. ICLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DIO15 | DIO14 | DIO13 | DIO12 | DIO11 | DIO10 | DIO9 | DIO8 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DIO7 | DIO6 | DIO5 | DIO4 | DIO3 | DIO2 | DIO1 | DIO0 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

**Table 9-24. ICLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | W | 0h | |
| 15 | DIO15 | W | 0h | DIO15 event<br>0h = No effect<br>1h = Clears DIO15 in RIS register |
| 14 | DIO14 | W | 0h | DIO14 event<br>0h = No effect<br>1h = Clears DIO14 in RIS register |
| 13 | DIO13 | W | 0h | DIO13 event<br>0h = No effect<br>1h = Clears DIO13 in RIS register |
| 12 | DIO12 | W | 0h | DIO12 event<br>0h = No effect<br>1h = Clears DIO12 in RIS register |
| 11 | DIO11 | W | 0h | DIO11 event<br>0h = No effect<br>1h = Clears DIO11 in RIS register |
| 10 | DIO10 | W | 0h | DIO10 event<br>0h = No effect<br>1h = Clears DIO10 in RIS register |
| 9 | DIO9 | W | 0h | DIO9 event<br>0h = No effect<br>1h = Clears DIO9 in RIS register |
| 8 | DIO8 | W | 0h | DIO8 event<br>0h = No effect<br>1h = Clears DIO8 in RIS register |
| 7 | DIO7 | W | 0h | DIO7 event<br>0h = No effect<br>1h = Clears DIO7 in RIS register |
| 6 | DIO6 | W | 0h | DIO6 event<br>0h = No effect<br>1h = Clears DIO6 in RIS register |



Copyright © 2026 Texas Instruments Incorporated

**Table 9-24. ICLR Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 5 | DIO5 | W | 0h | DIO5 event<br>0h = No effect<br>1h = Clears DIO5 in RIS register |
| 4 | DIO4 | W | 0h | DIO4 event<br>0h = No effect<br>1h = Clears DIO4 in RIS register |
| 3 | DIO3 | W | 0h | DIO3 event<br>0h = No effect<br>1h = Clears DIO3 in RIS register |
| 2 | DIO2 | W | 0h | DIO2 event<br>0h = No effect<br>1h = Clears DIO2 in RIS register |
| 1 | DIO1 | W | 0h | DIO1 event<br>0h = No effect<br>1h = Clears DIO1 in RIS register |
| 0 | DIO0 | W | 0h | DIO0 event<br>0h = No effect<br>1h = Clears DIO0 in RIS register |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.22 IIDX (Offset = 1080h) [Reset = 00000000h]

IIDX is shown in Figure 9-25 and described in Table 9-25.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. Value 0x00 means no event pending. Interrupt 1 is the highest priority, IIDX next highest, 4, 8, … IIDX^31 is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred. On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in [RIS] and [MIS] are cleared as well. After a read from the CPU (not from the debug interface), the register is updated with the next highest priority interrupt, if none are pending, then it should display 0x0.

**Figure 9-25. IIDX**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | STAT | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | |

**Table 9-25. IIDX Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | STAT | R | 0h | Interrupt index status<br>0h = No bit is set means there is no pending interrupt request<br>1h = DIO0 interrupt<br>2h = DIO1 interrupt<br>3h = DIO2 interrupt<br>4h = DIO3 interrupt<br>5h = DIO4 interrupt<br>6h = DIO5 interrupt<br>7h = DIO6 interrupt<br>8h = DIO7 interrupt<br>9h = DIO8 interrupt<br>Ah = DIO9 interrupt<br>Bh = DIO10 interrupt<br>Ch = DIO11 interrupt<br>Dh = DIO12 interrupt<br>Eh = DIO13 interrupt<br>Fh = DIO14 interrupt<br>10h = DIO15 interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.23 IMASK (Offset = 1088h) [Reset = 00000000h]

IMASK is shown in Figure 9-26 and described in Table 9-26.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is unmasked. Unmasking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

**Figure 9-26. IMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DIO31 | DIO30 | DIO29 | DIO28 | DIO27 | DIO26 | DIO25 | DIO24 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DIO23 | DIO22 | DIO21 | DIO20 | DIO19 | DIO18 | DIO17 | DIO16 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |

**Table 9-26. IMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | DIO31 | R/W | 0h | DIO31 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 30 | DIO30 | R/W | 0h | DIO30 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 29 | DIO29 | R/W | 0h | DIO29 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 28 | DIO28 | R/W | 0h | DIO28 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 27 | DIO27 | R/W | 0h | DIO27 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 26 | DIO26 | R/W | 0h | DIO26 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 25 | DIO25 | R/W | 0h | DIO25 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 24 | DIO24 | R/W | 0h | DIO24 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 23 | DIO23 | R/W | 0h | DIO23 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 22 | DIO22 | R/W | 0h | DIO22 event mask<br>0h = Event is masked<br>1h = Event is unmasked |



Copyright © 2026 Texas Instruments Incorporated

**Table 9-26. IMASK Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 21 | DIO21 | R/W | 0h | DIO21 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 20 | DIO20 | R/W | 0h | DIO20 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 19 | DIO19 | R/W | 0h | DIO19 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 18 | DIO18 | R/W | 0h | DIO18 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 17 | DIO17 | R/W | 0h | DIO17 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 16 | DIO16 | R/W | 0h | DIO16 event mask<br>0h = Event is masked<br>1h = Event is unmasked |
| 15-0 | RESERVED | R/W | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.24 RIS (Offset = 1090h) [Reset = 00000000h]

RIS is shown in Figure 9-27 and described in Table 9-27.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Figure 9-27. RIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DIO31 | DIO30 | DIO29 | DIO28 | DIO27 | DIO26 | DIO25 | DIO24 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DIO23 | DIO22 | DIO21 | DIO20 | DIO19 | DIO18 | DIO17 | DIO16 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |

**Table 9-27. RIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | DIO31 | R | 0h | DIO31 event<br>0h = DIO31 event did not occur<br>1h = DIO31 event occurred |
| 30 | DIO30 | R | 0h | DIO30 event<br>0h = DIO30 event did not occur<br>1h = DIO30 event occurred |
| 29 | DIO29 | R | 0h | DIO29 event<br>0h = DIO29 event did not occur<br>1h = DIO29 event occurred |
| 28 | DIO28 | R | 0h | DIO28 event<br>0h = DIO28 event did not occur<br>1h = DIO28 event occurred |
| 27 | DIO27 | R | 0h | DIO27 event<br>0h = DIO27 event did not occur<br>1h = DIO27 event occurred |
| 26 | DIO26 | R | 0h | DIO26 event<br>0h = DIO26 event did not occur<br>1h = DIO26 event occurred |
| 25 | DIO25 | R | 0h | DIO25 event<br>0h = DIO25 event did not occur<br>1h = DIO25 event occurred |
| 24 | DIO24 | R | 0h | DIO24 event<br>0h = DIO24 event did not occur<br>1h = DIO24 event occurred |
| 23 | DIO23 | R | 0h | DIO23 event<br>0h = DIO23 event did not occur<br>1h = DIO23 event occurred |
| 22 | DIO22 | R | 0h | DIO22 event<br>0h = DIO22 event did not occur<br>1h = DIO22 event occurred |



Copyright © 2026 Texas Instruments Incorporated

**Table 9-27. RIS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 21 | DIO21 | R | 0h | DIO21 event<br>0h = DIO21 event did not occur<br>1h = DIO21 event occurred |
| 20 | DIO20 | R | 0h | DIO20 event<br>0h = DIO20 event did not occur<br>1h = DIO20 event occurred |
| 19 | DIO19 | R | 0h | DIO19 event<br>0h = DIO19 event did not occur<br>1h = DIO19 event occurred |
| 18 | DIO18 | R | 0h | DIO18 event<br>0h = DIO18 event did not occur<br>1h = DIO18 event occurred |
| 17 | DIO17 | R | 0h | DIO17 event<br>0h = DIO17 event did not occur<br>1h = DIO17 event occurred |
| 16 | DIO16 | R | 0h | DIO16 event<br>0h = DIO16 event did not occur<br>1h = DIO16 event occurred |
| 15-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.25 MIS (Offset = 1098h) [Reset = 00000000h]

MIS is shown in Figure 9-28 and described in Table 9-28.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 9-28. MIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DIO31 | DIO30 | DIO29 | DIO28 | DIO27 | DIO26 | DIO25 | DIO24 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DIO23 | DIO22 | DIO21 | DIO20 | DIO19 | DIO18 | DIO17 | DIO16 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |

**Table 9-28. MIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | DIO31 | R | 0h | DIO31 event<br>0h = DIO31 event did not occur<br>1h = DIO31 event occurred |
| 30 | DIO30 | R | 0h | DIO30 event<br>0h = DIO30 event did not occur<br>1h = DIO30 event occurred |
| 29 | DIO29 | R | 0h | DIO29 event<br>0h = DIO29 event did not occur<br>1h = DIO29 event occurred |
| 28 | DIO28 | R | 0h | DIO28 event<br>0h = DIO28 event did not occur<br>1h = DIO28 event occurred |
| 27 | DIO27 | R | 0h | DIO27 event<br>0h = DIO27 event did not occur<br>1h = DIO27 event occurred |
| 26 | DIO26 | R | 0h | DIO26 event<br>0h = DIO26 event did not occur<br>1h = DIO26 event occurred |
| 25 | DIO25 | R | 0h | DIO25 event<br>0h = DIO25 event did not occur<br>1h = DIO25 event occurred |
| 24 | DIO24 | R | 0h | DIO24 event<br>0h = DIO24 event did not occur<br>1h = DIO24 event occurred |
| 23 | DIO23 | R | 0h | DIO23 event<br>0h = DIO23 event did not occur<br>1h = DIO23 event occurred |
| 22 | DIO22 | R | 0h | DIO22 event<br>0h = DIO22 event did not occur<br>1h = DIO22 event occurred |



Copyright © 2026 Texas Instruments Incorporated

**Table 9-28. MIS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 21 | DIO21 | R | 0h | DIO21 event<br>0h = DIO21 event did not occur<br>1h = DIO21 event occurred |
| 20 | DIO20 | R | 0h | DIO20 event<br>0h = DIO20 event did not occur<br>1h = DIO20 event occurred |
| 19 | DIO19 | R | 0h | DIO19 event<br>0h = DIO19 event did not occur<br>1h = DIO19 event occurred |
| 18 | DIO18 | R | 0h | DIO18 event<br>0h = DIO18 event did not occur<br>1h = DIO18 event occurred |
| 17 | DIO17 | R | 0h | DIO17 event<br>0h = DIO17 event did not occur<br>1h = DIO17 event occurred |
| 16 | DIO16 | R | 0h | DIO16 event<br>0h = DIO16 event did not occur<br>1h = DIO16 event occurred |
| 15-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.26 ISET (Offset = 10A0h) [Reset = 00000000h]

ISET is shown in Figure 9-29 and described in Table 9-29.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 9-29. ISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DIO31 | DIO30 | DIO29 | DIO28 | DIO27 | DIO26 | DIO25 | DIO24 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DIO23 | DIO22 | DIO21 | DIO20 | DIO19 | DIO18 | DIO17 | DIO16 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |

**Table 9-29. ISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | DIO31 | W | 0h | DIO31 event<br>0h = No effect<br>1h = Sets DIO31 in RIS register |
| 30 | DIO30 | W | 0h | DIO30 event<br>0h = No effect<br>1h = Sets DIO30 in RIS register |
| 29 | DIO29 | W | 0h | DIO29 event<br>0h = No effect<br>1h = Sets DIO29 in RIS register |
| 28 | DIO28 | W | 0h | DIO28 event<br>0h = No effect<br>1h = Sets DIO28 in RIS register |
| 27 | DIO27 | W | 0h | DIO27 event<br>0h = No effect<br>1h = Sets DIO27 in RIS register |
| 26 | DIO26 | W | 0h | DIO26 event<br>0h = No effect<br>1h = Sets DIO26 in RIS register |
| 25 | DIO25 | W | 0h | DIO25 event<br>0h = No effect<br>1h = Sets DIO25 in RIS register |
| 24 | DIO24 | W | 0h | DIO24 event<br>0h = No effect<br>1h = Sets DIO24 in RIS register |
| 23 | DIO23 | W | 0h | DIO23 event<br>0h = No effect<br>1h = Sets DIO23 in RIS register |
| 22 | DIO22 | W | 0h | DIO22 event<br>0h = No effect<br>1h = Sets DIO22 in RIS register |



Copyright © 2026 Texas Instruments Incorporated

**Table 9-29. ISET Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 21 | DIO21 | W | 0h | DIO21 event<br>0h = No effect<br>1h = Sets DIO21 in RIS register |
| 20 | DIO20 | W | 0h | DIO20 event<br>0h = No effect<br>1h = Sets DIO20 in RIS register |
| 19 | DIO19 | W | 0h | DIO19 event<br>0h = No effect<br>1h = Sets DIO19 in RIS register |
| 18 | DIO18 | W | 0h | DIO18 event<br>0h = No effect<br>1h = Sets DIO18 in RIS register |
| 17 | DIO17 | W | 0h | DIO17 event<br>0h = No effect<br>1h = Sets DIO17 in RIS register |
| 16 | DIO16 | W | 0h | DIO16 event<br>0h = No effect<br>1h = Sets DIO16 in RIS register |
| 15-0 | RESERVED | W | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.27 ICLR (Offset = 10A8h) [Reset = 00000000h]

ICLR is shown in Figure 9-30 and described in Table 9-30.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

**Figure 9-30. ICLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DIO31 | DIO30 | DIO29 | DIO28 | DIO27 | DIO26 | DIO25 | DIO24 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DIO23 | DIO22 | DIO21 | DIO20 | DIO19 | DIO18 | DIO17 | DIO16 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |

**Table 9-30. ICLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | DIO31 | W | 0h | DIO31 event<br>0h = No effect<br>1h = Clears DIO31 in RIS register |
| 30 | DIO30 | W | 0h | DIO30 event<br>0h = No effect<br>1h = Clears DIO30 in RIS register |
| 29 | DIO29 | W | 0h | DIO29 event<br>0h = No effect<br>1h = Clears DIO29 in RIS register |
| 28 | DIO28 | W | 0h | DIO28 event<br>0h = No effect<br>1h = Clears DIO28 in RIS register |
| 27 | DIO27 | W | 0h | DIO27 event<br>0h = No effect<br>1h = Clears DIO27 in RIS register |
| 26 | DIO26 | W | 0h | DIO26 event<br>0h = No effect<br>1h = Clears DIO26 in RIS register |
| 25 | DIO25 | W | 0h | DIO25 event<br>0h = No effect<br>1h = Clears DIO25 in RIS register |
| 24 | DIO24 | W | 0h | DIO24 event<br>0h = No effect<br>1h = Clears DIO24 in RIS register |
| 23 | DIO23 | W | 0h | DIO23 event<br>0h = No effect<br>1h = Clears DIO23 in RIS register |
| 22 | DIO22 | W | 0h | DIO22 event<br>0h = No effect<br>1h = Clears DIO22 in RIS register |



Copyright © 2026 Texas Instruments Incorporated

**Table 9-30. ICLR Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 21 | DIO21 | W | 0h | DIO21 event<br>0h = No effect<br>1h = Clears DIO21 in RIS register |
| 20 | DIO20 | W | 0h | DIO20 event<br>0h = No effect<br>1h = Clears DIO20 in RIS register |
| 19 | DIO19 | W | 0h | DIO19 event<br>0h = No effect<br>1h = Clears DIO19 in RIS register |
| 18 | DIO18 | W | 0h | DIO18 event<br>0h = No effect<br>1h = Clears DIO18 in RIS register |
| 17 | DIO17 | W | 0h | DIO17 event<br>0h = No effect<br>1h = Clears DIO17 in RIS register |
| 16 | DIO16 | W | 0h | DIO16 event<br>0h = No effect<br>1h = Clears DIO16 in RIS register |
| 15-0 | RESERVED | W | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.28 EVT_MODE (Offset = 10E0h) [Reset = 00000029h]

EVT_MODE is shown in Figure 9-31 and described in Table 9-31.

Return to the Summary Table.

Event mode register. It is used to select whether each line is disabled, in software mode (software clears the RIS) or in hardware mode (hardware clears the RIS)

**Figure 9-31. EVT_MODE**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R/W- | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R/W- | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R/W- | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | EVT2_CFG | | EVT1_CFG | | INT0_CFG | |
| R/W- | | R-2h | | R-2h | | R-1h | |

**Table 9-31. EVT_MODE Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | RESERVED | R/W | 0h | |
| 5-4 | EVT2_CFG | R | 2h | Event line mode select for event corresponding to none.GEN_EVENT1<br>0h = The interrupt or event line is disabled.<br>1h = The interrupt or event line is in software mode. Software must clear the RIS.<br>2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag. |
| 3-2 | EVT1_CFG | R | 2h | Event line mode select for event corresponding to none.GEN_EVENT0<br>0h = The interrupt or event line is disabled.<br>1h = The interrupt or event line is in software mode. Software must clear the RIS.<br>2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag. |
| 1-0 | INT0_CFG | R | 1h | Event line mode select for event corresponding to none.CPU_INT<br>0h = The interrupt or event line is disabled.<br>1h = The interrupt or event line is in software mode. Software must clear the RIS.<br>2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag. |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.29 DESC (Offset = 10FCh) [Reset = 16110000h]

DESC is shown in Figure 9-32 and described in Table 9-32.

Return to the Summary Table.

This register identifies the peripheral and its exact version.

**Figure 9-32. DESC**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MODULEID | | | | | | | | | | | | | | | |
| R-1611h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| FEATUREVER | | | | RESERVED | | | | MAJREV | | | | MINREV | | | |
| R- | | | | R- | | | | R- | | | | R- | | | |

**Table 9-32. DESC Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | MODULEID | R | 1611h | Module identification contains a unique peripheral identification number. The assignments are maintained in a central database for all of the platform modules to ensure uniqueness.<br>0h = Smallest value<br>FFFFh = Highest possible value |
| 15-12 | FEATUREVER | R | 0h | Feature Set for the module *instance*<br>0h = Smallest value<br>Fh = Highest possible value |
| 11-8 | RESERVED | R | 0h | |
| 7-4 | MAJREV | R | 0h | Major rev of the IP<br>0h = Smallest value<br>Fh = Highest possible value |
| 3-0 | MINREV | R | 0h | Minor rev of the IP<br>0h = Smallest value<br>Fh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.30 DOUT3_0 (Offset = 1200h) [Reset = 00000000h]

DOUT3_0 is shown in Figure 9-33 and described in Table 9-33.

Return to the Summary Table.

Data output for pins configured as DIO3 to DIO0. This is an alias register for byte access to bits 3 to 0 in DOUT31_0 register.

**Figure 9-33. DOUT3_0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | DIO3 |
| W-0h | | | | | | | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | DIO2 |
| W-0h | | | | | | | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | DIO1 |
| W-0h | | | | | | | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | DIO0 |
| W-0h | | | | | | | W-0h |

**Table 9-33. DOUT3_0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-25 | RESERVED | W | 0h | |
| 24 | DIO3 | W | 0h | This bit sets the value of the pin configured as DIO3 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 23-17 | RESERVED | W | 0h | |
| 16 | DIO2 | W | 0h | This bit sets the value of the pin configured as DIO2 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 15-9 | RESERVED | W | 0h | |
| 8 | DIO1 | W | 0h | This bit sets the value of the pin configured as DIO1 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 7-1 | RESERVED | W | 0h | |
| 0 | DIO0 | W | 0h | This bit sets the value of the pin configured as DIO0 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.31 DOUT7_4 (Offset = 1204h) [Reset = 00000000h]

DOUT7_4 is shown in Figure 9-34 and described in Table 9-34.

Return to the Summary Table.

Data output for pins configured as DIO7 to DIO4. This is an alias register for byte access to bits 7 to 4 in DOUT31_0 register.

**Figure 9-34. DOUT7_4**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | DIO7 |
| W-0h | | | | | | | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | DIO6 |
| W-0h | | | | | | | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | DIO5 |
| W-0h | | | | | | | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | DIO4 |
| W-0h | | | | | | | W-0h |

**Table 9-34. DOUT7_4 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-25 | RESERVED | W | 0h | |
| 24 | DIO7 | W | 0h | This bit sets the value of the pin configured as DIO7 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 23-17 | RESERVED | W | 0h | |
| 16 | DIO6 | W | 0h | This bit sets the value of the pin configured as DIO6 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 15-9 | RESERVED | W | 0h | |
| 8 | DIO5 | W | 0h | This bit sets the value of the pin configured as DIO5 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 7-1 | RESERVED | W | 0h | |
| 0 | DIO4 | W | 0h | This bit sets the value of the pin configured as DIO4 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.32 DOUT11_8 (Offset = 1208h) [Reset = 00000000h]

DOUT11_8 is shown in Figure 9-35 and described in Table 9-35.

Return to the Summary Table.

Data output for pins configured as DIO11 to DIO8. This is an alias register for byte access to bits 11 to 8 in DOUT31_0 register.

**Figure 9-35. DOUT11_8**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | DIO11 |
| W-0h | | | | | | | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | DIO10 |
| W-0h | | | | | | | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | DIO9 |
| W-0h | | | | | | | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | DIO8 |
| W-0h | | | | | | | W-0h |

**Table 9-35. DOUT11_8 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-25 | RESERVED | W | 0h | |
| 24 | DIO11 | W | 0h | This bit sets the value of the pin configured as DIO11 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 23-17 | RESERVED | W | 0h | |
| 16 | DIO10 | W | 0h | This bit sets the value of the pin configured as DIO10 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 15-9 | RESERVED | W | 0h | |
| 8 | DIO9 | W | 0h | This bit sets the value of the pin configured as DIO9 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 7-1 | RESERVED | W | 0h | |
| 0 | DIO8 | W | 0h | This bit sets the value of the pin configured as DIO8 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.33 DOUT15_12 (Offset = 120Ch) [Reset = 00000000h]

DOUT15_12 is shown in Figure 9-36 and described in Table 9-36.

Return to the Summary Table.

Data output for pins configured as DIO15 to DIO12. This is an alias register for byte access to bits 15 to 12 in DOUT31_0 register.

**Figure 9-36. DOUT15_12**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | DIO15 |
| W-0h | | | | | | | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | DIO14 |
| W-0h | | | | | | | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | DIO13 |
| W-0h | | | | | | | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | DIO12 |
| W-0h | | | | | | | W-0h |

**Table 9-36. DOUT15_12 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-25 | RESERVED | W | 0h | |
| 24 | DIO15 | W | 0h | This bit sets the value of the pin configured as DIO15 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 23-17 | RESERVED | W | 0h | |
| 16 | DIO14 | W | 0h | This bit sets the value of the pin configured as DIO14 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 15-9 | RESERVED | W | 0h | |
| 8 | DIO13 | W | 0h | This bit sets the value of the pin configured as DIO13 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 7-1 | RESERVED | W | 0h | |
| 0 | DIO12 | W | 0h | This bit sets the value of the pin configured as DIO12 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.34 DOUT19_16 (Offset = 1210h) [Reset = 00000000h]

DOUT19_16 is shown in Figure 9-37 and described in Table 9-37.

Return to the Summary Table.

Data output for pins configured as DIO19 to DIO16. This is an alias register for byte access to bits 19 to 16 in DOUT31_0 register.

**Figure 9-37. DOUT19_16**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | DIO19 |
| W-0h | | | | | | | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | DIO18 |
| W-0h | | | | | | | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | DIO17 |
| W-0h | | | | | | | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | DIO16 |
| W-0h | | | | | | | W-0h |

**Table 9-37. DOUT19_16 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-25 | RESERVED | W | 0h | |
| 24 | DIO19 | W | 0h | This bit sets the value of the pin configured as DIO19 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 23-17 | RESERVED | W | 0h | |
| 16 | DIO18 | W | 0h | This bit sets the value of the pin configured as DIO18 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 15-9 | RESERVED | W | 0h | |
| 8 | DIO17 | W | 0h | This bit sets the value of the pin configured as DIO17 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 7-1 | RESERVED | W | 0h | |
| 0 | DIO16 | W | 0h | This bit sets the value of the pin configured as DIO16 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.35 DOUT23_20 (Offset = 1214h) [Reset = 00000000h]

DOUT23_20 is shown in Figure 9-38 and described in Table 9-38.

Return to the Summary Table.

Data output for pins configured as DIO23 to DIO20. This is an alias register for byte access to bits 23 to 20 in DOUT31_0 register.

**Figure 9-38. DOUT23_20**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | DIO23 |
| W-0h | | | | | | | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | DIO22 |
| W-0h | | | | | | | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | DIO21 |
| W-0h | | | | | | | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | DIO20 |
| W-0h | | | | | | | W-0h |

**Table 9-38. DOUT23_20 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-25 | RESERVED | W | 0h | |
| 24 | DIO23 | W | 0h | This bit sets the value of the pin configured as DIO23 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 23-17 | RESERVED | W | 0h | |
| 16 | DIO22 | W | 0h | This bit sets the value of the pin configured as DIO22 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 15-9 | RESERVED | W | 0h | |
| 8 | DIO21 | W | 0h | This bit sets the value of the pin configured as DIO21 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 7-1 | RESERVED | W | 0h | |
| 0 | DIO20 | W | 0h | This bit sets the value of the pin configured as DIO20 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.36 DOUT27_24 (Offset = 1218h) [Reset = 00000000h]

DOUT27_24 is shown in Figure 9-39 and described in Table 9-39.

Return to the Summary Table.

Data output for pins configured as DIO27 to DIO24. This is an alias register for byte access to bits 27 to 24 in DOUT31_0 register.

**Figure 9-39. DOUT27_24**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | DIO27 |
| W-0h | | | | | | | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | DIO26 |
| W-0h | | | | | | | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | DIO25 |
| W-0h | | | | | | | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | DIO24 |
| W-0h | | | | | | | W-0h |

**Table 9-39. DOUT27_24 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-25 | RESERVED | W | 0h | |
| 24 | DIO27 | W | 0h | This bit sets the value of the pin configured as DIO27 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 23-17 | RESERVED | W | 0h | |
| 16 | DIO26 | W | 0h | This bit sets the value of the pin configured as DIO26 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 15-9 | RESERVED | W | 0h | |
| 8 | DIO25 | W | 0h | This bit sets the value of the pin configured as DIO25 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 7-1 | RESERVED | W | 0h | |
| 0 | DIO24 | W | 0h | This bit sets the value of the pin configured as DIO24 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.37 DOUT31_28 (Offset = 121Ch) [Reset = 00000000h]

DOUT31_28 is shown in Figure 9-40 and described in Table 9-40.

Return to the Summary Table.

Data output for pins configured as DIO31 to DIO28. This is an alias register for byte access to bits 31 to 28 in DOUT31_0 register.

**Figure 9-40. DOUT31_28**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | DIO31 |
| W-0h | | | | | | | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | DIO30 |
| W-0h | | | | | | | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | DIO29 |
| W-0h | | | | | | | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | DIO28 |
| W-0h | | | | | | | W-0h |

**Table 9-40. DOUT31_28 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-25 | RESERVED | W | 0h | |
| 24 | DIO31 | W | 0h | This bit sets the value of the pin configured as DIO31 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 23-17 | RESERVED | W | 0h | |
| 16 | DIO30 | W | 0h | This bit sets the value of the pin configured as DIO30 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 15-9 | RESERVED | W | 0h | |
| 8 | DIO29 | W | 0h | This bit sets the value of the pin configured as DIO29 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 7-1 | RESERVED | W | 0h | |
| 0 | DIO28 | W | 0h | This bit sets the value of the pin configured as DIO28 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.38 DOUT31_0 (Offset = 1280h) [Reset = 00000000h]

DOUT31_0 is shown in Figure 9-41 and described in Table 9-41.

Return to the Summary Table.

Data output for pins configured as DIO31 to DIO0.

**Figure 9-41. DOUT31_0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DIO31 | DIO30 | DIO29 | DIO28 | DIO27 | DIO26 | DIO25 | DIO24 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DIO23 | DIO22 | DIO21 | DIO20 | DIO19 | DIO18 | DIO17 | DIO16 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DIO15 | DIO14 | DIO13 | DIO12 | DIO11 | DIO10 | DIO9 | DIO8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DIO7 | DIO6 | DIO5 | DIO4 | DIO3 | DIO2 | DIO1 | DIO0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

**Table 9-41. DOUT31_0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | DIO31 | R/W | 0h | This bit sets the value of the pin configured as DIO31 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 30 | DIO30 | R/W | 0h | This bit sets the value of the pin configured as DIO30 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 29 | DIO29 | R/W | 0h | This bit sets the value of the pin configured as DIO29 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 28 | DIO28 | R/W | 0h | This bit sets the value of the pin configured as DIO28 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 27 | DIO27 | R/W | 0h | This bit sets the value of the pin configured as DIO27 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 26 | DIO26 | R/W | 0h | This bit sets the value of the pin configured as DIO26 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 25 | DIO25 | R/W | 0h | This bit sets the value of the pin configured as DIO25 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 24 | DIO24 | R/W | 0h | This bit sets the value of the pin configured as DIO24 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |



Copyright © 2026 Texas Instruments Incorporated

**Table 9-41. DOUT31_0 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 23 | DIO23 | R/W | 0h | This bit sets the value of the pin configured as DIO23 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 22 | DIO22 | R/W | 0h | This bit sets the value of the pin configured as DIO22 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 21 | DIO21 | R/W | 0h | This bit sets the value of the pin configured as DIO21 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 20 | DIO20 | R/W | 0h | This bit sets the value of the pin configured as DIO20 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 19 | DIO19 | R/W | 0h | This bit sets the value of the pin configured as DIO19 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 18 | DIO18 | R/W | 0h | This bit sets the value of the pin configured as DIO18 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 17 | DIO17 | R/W | 0h | This bit sets the value of the pin configured as DIO17 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 16 | DIO16 | R/W | 0h | This bit sets the value of the pin configured as DIO16 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 15 | DIO15 | R/W | 0h | This bit sets the value of the pin configured as DIO15 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 14 | DIO14 | R/W | 0h | This bit sets the value of the pin configured as DIO14 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 13 | DIO13 | R/W | 0h | This bit sets the value of the pin configured as DIO13 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 12 | DIO12 | R/W | 0h | This bit sets the value of the pin configured as DIO12 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 11 | DIO11 | R/W | 0h | This bit sets the value of the pin configured as DIO11 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 10 | DIO10 | R/W | 0h | This bit sets the value of the pin configured as DIO10 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |



Copyright © 2026 Texas Instruments Incorporated

**Table 9-41. DOUT31_0 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 9 | DIO9 | R/W | 0h | This bit sets the value of the pin configured as DIO9 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 8 | DIO8 | R/W | 0h | This bit sets the value of the pin configured as DIO8 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 7 | DIO7 | R/W | 0h | This bit sets the value of the pin configured as DIO7 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 6 | DIO6 | R/W | 0h | This bit sets the value of the pin configured as DIO6 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 5 | DIO5 | R/W | 0h | This bit sets the value of the pin configured as DIO5 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 4 | DIO4 | R/W | 0h | This bit sets the value of the pin configured as DIO4 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 3 | DIO3 | R/W | 0h | This bit sets the value of the pin configured as DIO3 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 2 | DIO2 | R/W | 0h | This bit sets the value of the pin configured as DIO2 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 1 | DIO1 | R/W | 0h | This bit sets the value of the pin configured as DIO1 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 0 | DIO0 | R/W | 0h | This bit sets the value of the pin configured as DIO0 when the output is enabled through DOE31_0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.39 DOUTSET31_0 (Offset = 1290h) [Reset = 00000000h]

DOUTSET31_0 is shown in Figure 9-42 and described in Table 9-42.

Return to the Summary Table.

Writing 1 to a bit position in this register sets the corresponding bit in the DOUT31_0 register.

**Figure 9-42. DOUTSET31_0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DIO31 | DIO30 | DIO29 | DIO28 | DIO27 | DIO26 | DIO25 | DIO24 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DIO23 | DIO22 | DIO21 | DIO20 | DIO19 | DIO18 | DIO17 | DIO16 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DIO15 | DIO14 | DIO13 | DIO12 | DIO11 | DIO10 | DIO9 | DIO8 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DIO7 | DIO6 | DIO5 | DIO4 | DIO3 | DIO2 | DIO1 | DIO0 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

**Table 9-42. DOUTSET31_0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | DIO31 | W | 0h | Writing 1 to this bit sets the DIO31 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO31 in DOUT31_0 |
| 30 | DIO30 | W | 0h | Writing 1 to this bit sets the DIO30 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO30 in DOUT31_0 |
| 29 | DIO29 | W | 0h | Writing 1 to this bit sets the DIO29 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO29 in DOUT31_0 |
| 28 | DIO28 | W | 0h | Writing 1 to this bit sets the DIO28 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO28 in DOUT31_0 |
| 27 | DIO27 | W | 0h | Writing 1 to this bit sets the DIO27 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO27 in DOUT31_0 |
| 26 | DIO26 | W | 0h | Writing 1 to this bit sets the DIO26 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO26 in DOUT31_0 |
| 25 | DIO25 | W | 0h | Writing 1 to this bit sets the DIO25 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO25 in DOUT31_0 |
| 24 | DIO24 | W | 0h | Writing 1 to this bit sets the DIO24 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO24 in DOUT31_0 |



Copyright © 2026 Texas Instruments Incorporated

**Table 9-42. DOUTSET31_0 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 23 | DIO23 | W | 0h | Writing 1 to this bit sets the DIO23 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO23 in DOUT31_0 |
| 22 | DIO22 | W | 0h | Writing 1 to this bit sets the DIO22 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO22 in DOUT31_0 |
| 21 | DIO21 | W | 0h | Writing 1 to this bit sets the DIO21 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO21 in DOUT31_0 |
| 20 | DIO20 | W | 0h | Writing 1 to this bit sets the DIO20 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO20 in DOUT31_0 |
| 19 | DIO19 | W | 0h | Writing 1 to this bit sets the DIO19 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO19 in DOUT31_0 |
| 18 | DIO18 | W | 0h | Writing 1 to this bit sets the DIO18 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO18 in DOUT31_0 |
| 17 | DIO17 | W | 0h | Writing 1 to this bit sets the DIO17 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO17 in DOUT31_0 |
| 16 | DIO16 | W | 0h | Writing 1 to this bit sets the DIO16 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO16 in DOUT31_0 |
| 15 | DIO15 | W | 0h | Writing 1 to this bit sets the DIO15 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO15 in DOUT31_0 |
| 14 | DIO14 | W | 0h | Writing 1 to this bit sets the DIO14 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO14 in DOUT31_0 |
| 13 | DIO13 | W | 0h | Writing 1 to this bit sets the DIO13 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO13 in DOUT31_0 |
| 12 | DIO12 | W | 0h | Writing 1 to this bit sets the DIO12 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO12 in DOUT31_0 |
| 11 | DIO11 | W | 0h | Writing 1 to this bit sets the DIO11 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO11 in DOUT31_0 |
| 10 | DIO10 | W | 0h | Writing 1 to this bit sets the DIO10 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO10 in DOUT31_0 |



Copyright © 2026 Texas Instruments Incorporated

**Table 9-42. DOUTSET31_0 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 9 | DIO9 | W | 0h | Writing 1 to this bit sets the DIO9 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO9 in DOUT31_0 |
| 8 | DIO8 | W | 0h | Writing 1 to this bit sets the DIO8 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO8 in DOUT31_0 |
| 7 | DIO7 | W | 0h | Writing 1 to this bit sets the DIO7 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO7 in DOUT31_0 |
| 6 | DIO6 | W | 0h | Writing 1 to this bit sets the DIO6 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO6 in DOUT31_0 |
| 5 | DIO5 | W | 0h | Writing 1 to this bit sets the DIO5 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO5 in DOUT31_0 |
| 4 | DIO4 | W | 0h | Writing 1 to this bit sets the DIO4 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO4 in DOUT31_0 |
| 3 | DIO3 | W | 0h | Writing 1 to this bit sets the DIO3 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO3 in DOUT31_0 |
| 2 | DIO2 | W | 0h | Writing 1 to this bit sets the DIO2 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO2 in DOUT31_0 |
| 1 | DIO1 | W | 0h | Writing 1 to this bit sets the DIO1 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO1 in DOUT31_0 |
| 0 | DIO0 | W | 0h | Writing 1 to this bit sets the DIO0 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO0 in DOUT31_0 |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.40 DOUTCLR31_0 (Offset = 12A0h) [Reset = 00000000h]

DOUTCLR31_0 is shown in Figure 9-43 and described in Table 9-43.

Return to the Summary Table.

Writing 1 to a bit position in this register clears the corresponding bit in the DOUT31_0 register.

**Figure 9-43. DOUTCLR31_0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DIO31 | DIO30 | DIO29 | DIO28 | DIO27 | DIO26 | DIO25 | DIO24 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DIO23 | DIO22 | DIO21 | DIO20 | DIO19 | DIO18 | DIO17 | DIO16 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DIO15 | DIO14 | DIO13 | DIO12 | DIO11 | DIO10 | DIO9 | DIO8 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DIO7 | DIO6 | DIO5 | DIO4 | DIO3 | DIO2 | DIO1 | DIO0 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

**Table 9-43. DOUTCLR31_0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | DIO31 | W | 0h | Writing 1 to this bit clears the DIO31 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO31 in DOUT31_0 |
| 30 | DIO30 | W | 0h | Writing 1 to this bit clears the DIO30 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO30 in DOUT31_0 |
| 29 | DIO29 | W | 0h | Writing 1 to this bit clears the DIO29 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO29 in DOUT31_0 |
| 28 | DIO28 | W | 0h | Writing 1 to this bit clears the DIO28 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO28 in DOUT31_0 |
| 27 | DIO27 | W | 0h | Writing 1 to this bit clears the DIO27 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO27 in DOUT31_0 |
| 26 | DIO26 | W | 0h | Writing 1 to this bit clears the DIO26 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO26 in DOUT31_0 |
| 25 | DIO25 | W | 0h | Writing 1 to this bit clears the DIO25 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO25 in DOUT31_0 |
| 24 | DIO24 | W | 0h | Writing 1 to this bit clears the DIO24 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO24 in DOUT31_0 |



Copyright © 2026 Texas Instruments Incorporated

**Table 9-43. DOUTCLR31_0 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 23 | DIO23 | W | 0h | Writing 1 to this bit clears the DIO23 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO23 in DOUT31_0 |
| 22 | DIO22 | W | 0h | Writing 1 to this bit clears the DIO22 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO22 in DOUT31_0 |
| 21 | DIO21 | W | 0h | Writing 1 to this bit clears the DIO21 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO21 in DOUT31_0 |
| 20 | DIO20 | W | 0h | Writing 1 to this bit clears the DIO20 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO20 in DOUT31_0 |
| 19 | DIO19 | W | 0h | Writing 1 to this bit clears the DIO19 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO19 in DOUT31_0 |
| 18 | DIO18 | W | 0h | Writing 1 to this bit clears the DIO18 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO18 in DOUT31_0 |
| 17 | DIO17 | W | 0h | Writing 1 to this bit clears the DIO17 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO17 in DOUT31_0 |
| 16 | DIO16 | W | 0h | Writing 1 to this bit clears the DIO16 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO16 in DOUT31_0 |
| 15 | DIO15 | W | 0h | Writing 1 to this bit clears the DIO15 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO15 in DOUT31_0 |
| 14 | DIO14 | W | 0h | Writing 1 to this bit clears the DIO14 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO14 in DOUT31_0 |
| 13 | DIO13 | W | 0h | Writing 1 to this bit clears the DIO13 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO13 in DOUT31_0 |
| 12 | DIO12 | W | 0h | Writing 1 to this bit clears the DIO12 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO12 in DOUT31_0 |
| 11 | DIO11 | W | 0h | Writing 1 to this bit clears the DIO11 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO11 in DOUT31_0 |
| 10 | DIO10 | W | 0h | Writing 1 to this bit clears the DIO10 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO10 in DOUT31_0 |



Copyright © 2026 Texas Instruments Incorporated

**Table 9-43. DOUTCLR31_0 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 9 | DIO9 | W | 0h | Writing 1 to this bit clears the DIO9 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO9 in DOUT31_0 |
| 8 | DIO8 | W | 0h | Writing 1 to this bit clears the DIO8 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO8 in DOUT31_0 |
| 7 | DIO7 | W | 0h | Writing 1 to this bit clears the DIO7 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO7 in DOUT31_0 |
| 6 | DIO6 | W | 0h | Writing 1 to this bit clears the DIO6 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO6 in DOUT31_0 |
| 5 | DIO5 | W | 0h | Writing 1 to this bit clears the DIO5 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO5 in DOUT31_0 |
| 4 | DIO4 | W | 0h | Writing 1 to this bit clears the DIO4 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO4 in DOUT31_0 |
| 3 | DIO3 | W | 0h | Writing 1 to this bit clears the DIO3 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO3 in DOUT31_0 |
| 2 | DIO2 | W | 0h | Writing 1 to this bit clears the DIO2 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO2 in DOUT31_0 |
| 1 | DIO1 | W | 0h | Writing 1 to this bit clears the DIO1 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO1 in DOUT31_0 |
| 0 | DIO0 | W | 0h | Writing 1 to this bit clears the DIO0 bit in the DOUT31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO0 in DOUT31_0 |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.41 DOUTTGL31_0 (Offset = 12B0h) [Reset = 00000000h]

DOUTTGL31_0 is shown in Figure 9-44 and described in Table 9-44.

Return to the Summary Table.

Writing 1 to a bit position in this register will invert the corresponding DIO output.

**Figure 9-44. DOUTTGL31_0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DIO31 | DIO30 | DIO29 | DIO28 | DIO27 | DIO26 | DIO25 | DIO24 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DIO23 | DIO22 | DIO21 | DIO20 | DIO19 | DIO18 | DIO17 | DIO16 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DIO15 | DIO14 | DIO13 | DIO12 | DIO11 | DIO10 | DIO9 | DIO8 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DIO7 | DIO6 | DIO5 | DIO4 | DIO3 | DIO2 | DIO1 | DIO0 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

**Table 9-44. DOUTTGL31_0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | DIO31 | W | 0h | This bit is used to toggle DIO31 output.<br>0h = No effect<br>1h = Toggle output |
| 30 | DIO30 | W | 0h | This bit is used to toggle DIO30 output.<br>0h = No effect<br>1h = Toggle output |
| 29 | DIO29 | W | 0h | This bit is used to toggle DIO29 output.<br>0h = No effect<br>1h = Toggle output |
| 28 | DIO28 | W | 0h | This bit is used to toggle DIO28 output.<br>0h = No effect<br>1h = Toggle output |
| 27 | DIO27 | W | 0h | This bit is used to toggle DIO27 output.<br>0h = No effect<br>1h = Toggle output |
| 26 | DIO26 | W | 0h | This bit is used to toggle DIO26 output.<br>0h = No effect<br>1h = Toggle output |
| 25 | DIO25 | W | 0h | This bit is used to toggle DIO25 output.<br>0h = No effect<br>1h = Toggle output |
| 24 | DIO24 | W | 0h | This bit is used to toggle DIO24 output.<br>0h = No effect<br>1h = Toggle output |
| 23 | DIO23 | W | 0h | This bit is used to toggle DIO23 output.<br>0h = No effect<br>1h = Toggle output |
| 22 | DIO22 | W | 0h | This bit is used to toggle DIO22 output.<br>0h = No effect<br>1h = Toggle output |



Copyright © 2026 Texas Instruments Incorporated

**Table 9-44. DOUTTGL31_0 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 21 | DIO21 | W | 0h | This bit is used to toggle DIO21 output.<br>0h = No effect<br>1h = Toggle output |
| 20 | DIO20 | W | 0h | This bit is used to toggle DIO20 output.<br>0h = No effect<br>1h = Toggle output |
| 19 | DIO19 | W | 0h | This bit is used to toggle DIO19 output.<br>0h = No effect<br>1h = Toggle output |
| 18 | DIO18 | W | 0h | This bit is used to toggle DIO18 output.<br>0h = No effect<br>1h = Toggle output |
| 17 | DIO17 | W | 0h | This bit is used to toggle DIO17 output.<br>0h = No effect<br>1h = Toggle output |
| 16 | DIO16 | W | 0h | This bit is used to toggle DIO16 output.<br>0h = No effect<br>1h = Toggle output |
| 15 | DIO15 | W | 0h | This bit is used to toggle DIO15 output.<br>0h = No effect<br>1h = Toggle output |
| 14 | DIO14 | W | 0h | This bit is used to toggle DIO14 output.<br>0h = No effect<br>1h = Toggle output |
| 13 | DIO13 | W | 0h | This bit is used to toggle DIO13 output.<br>0h = No effect<br>1h = Toggle output |
| 12 | DIO12 | W | 0h | This bit is used to toggle DIO12 output.<br>0h = No effect<br>1h = Toggle output |
| 11 | DIO11 | W | 0h | This bit is used to toggle DIO11 output.<br>0h = No effect<br>1h = Toggle output |
| 10 | DIO10 | W | 0h | This bit is used to toggle DIO10 output.<br>0h = No effect<br>1h = Toggle output |
| 9 | DIO9 | W | 0h | This bit is used to toggle DIO9 output.<br>0h = No effect<br>1h = Toggle output |
| 8 | DIO8 | W | 0h | This bit is used to toggle DIO8 output.<br>0h = No effect<br>1h = Toggle output |
| 7 | DIO7 | W | 0h | This bit is used to toggle DIO7 output.<br>0h = No effect<br>1h = Toggle output |
| 6 | DIO6 | W | 0h | This bit is used to toggle DIO6 output.<br>0h = No effect<br>1h = Toggle output |
| 5 | DIO5 | W | 0h | This bit is used to toggle DIO5 output.<br>0h = No effect<br>1h = Toggle output |
| 4 | DIO4 | W | 0h | This bit is used to toggle DIO4 output.<br>0h = No effect<br>1h = Toggle output |
| 3 | DIO3 | W | 0h | This bit is used to toggle DIO3 output.<br>0h = No effect<br>1h = Toggle output |



Copyright © 2026 Texas Instruments Incorporated

**Table 9-44. DOUTTGL31_0 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | DIO2 | W | 0h | This bit is used to toggle DIO2 output.<br>0h = No effect<br>1h = Toggle output |
| 1 | DIO1 | W | 0h | This bit is used to toggle DIO1 output.<br>0h = No effect<br>1h = Toggle output |
| 0 | DIO0 | W | 0h | This bit is used to toggle DIO0 output.<br>0h = No effect<br>1h = Toggle output |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.42 DOE31_0 (Offset = 12C0h) [Reset = 00000000h]

DOE31_0 is shown in Figure 9-45 and described in Table 9-45.

Return to the Summary Table.

This register is used to enable the data outputs for DIO31 to DIO0.

**Figure 9-45. DOE31_0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DIO31 | DIO30 | DIO29 | DIO28 | DIO27 | DIO26 | DIO25 | DIO24 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DIO23 | DIO22 | DIO21 | DIO20 | DIO19 | DIO18 | DIO17 | DIO16 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DIO15 | DIO14 | DIO13 | DIO12 | DIO11 | DIO10 | DIO9 | DIO8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DIO7 | DIO6 | DIO5 | DIO4 | DIO3 | DIO2 | DIO1 | DIO0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

**Table 9-45. DOE31_0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | DIO31 | R/W | 0h | Enables data output for DIO31.<br>0h = Output disabled<br>1h = Output enabled |
| 30 | DIO30 | R/W | 0h | Enables data output for DIO30.<br>0h = Output disabled<br>1h = Output enabled |
| 29 | DIO29 | R/W | 0h | Enables data output for DIO29.<br>0h = Output disabled<br>1h = Output enabled |
| 28 | DIO28 | R/W | 0h | Enables data output for DIO28.<br>0h = Output disabled<br>1h = Output enabled |
| 27 | DIO27 | R/W | 0h | Enables data output for DIO27.<br>0h = Output disabled<br>1h = Output enabled |
| 26 | DIO26 | R/W | 0h | Enables data output for DIO26.<br>0h = Output disabled<br>1h = Output enabled |
| 25 | DIO25 | R/W | 0h | Enables data output for DIO25.<br>0h = Output disabled<br>1h = Output enabled |
| 24 | DIO24 | R/W | 0h | Enables data output for DIO24.<br>0h = Output disabled<br>1h = Output enabled |
| 23 | DIO23 | R/W | 0h | Enables data output for DIO23.<br>0h = Output disabled<br>1h = Output enabled |
| 22 | DIO22 | R/W | 0h | Enables data output for DIO22.<br>0h = Output disabled<br>1h = Output enabled |



Copyright © 2026 Texas Instruments Incorporated

**Table 9-45. DOE31_0 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 21 | DIO21 | R/W | 0h | Enables data output for DIO21.<br>0h = Output disabled<br>1h = Output enabled |
| 20 | DIO20 | R/W | 0h | Enables data output for DIO20.<br>0h = Output disabled<br>1h = Output enabled |
| 19 | DIO19 | R/W | 0h | Enables data output for DIO19.<br>0h = Output disabled<br>1h = Output enabled |
| 18 | DIO18 | R/W | 0h | Enables data output for DIO18.<br>0h = Output disabled<br>1h = Output enabled |
| 17 | DIO17 | R/W | 0h | Enables data output for DIO17.<br>0h = Output disabled<br>1h = Output enabled |
| 16 | DIO16 | R/W | 0h | Enables data output for DIO16.<br>0h = Output disabled<br>1h = Output enabled |
| 15 | DIO15 | R/W | 0h | Enables data output for DIO15.<br>0h = Output disabled<br>1h = Output enabled |
| 14 | DIO14 | R/W | 0h | Enables data output for DIO14.<br>0h = Output disabled<br>1h = Output enabled |
| 13 | DIO13 | R/W | 0h | Enables data output for DIO13.<br>0h = Output disabled<br>1h = Output enabled |
| 12 | DIO12 | R/W | 0h | Enables data output for DIO12.<br>0h = Output disabled<br>1h = Output enabled |
| 11 | DIO11 | R/W | 0h | Enables data output for DIO11.<br>0h = Output disabled<br>1h = Output enabled |
| 10 | DIO10 | R/W | 0h | Enables data output for DIO10.<br>0h = Output disabled<br>1h = Output enabled |
| 9 | DIO9 | R/W | 0h | Enables data output for DIO9.<br>0h = Output disabled<br>1h = Output enabled |
| 8 | DIO8 | R/W | 0h | Enables data output for DIO8.<br>0h = Output disabled<br>1h = Output enabled |
| 7 | DIO7 | R/W | 0h | Enables data output for DIO7.<br>0h = Output disabled<br>1h = Output enabled |
| 6 | DIO6 | R/W | 0h | Enables data output for DIO6.<br>0h = Output disabled<br>1h = Output enabled |
| 5 | DIO5 | R/W | 0h | Enables data output for DIO5.<br>0h = Output disabled<br>1h = Output enabled |
| 4 | DIO4 | R/W | 0h | Enables data output for DIO4.<br>0h = Output disabled<br>1h = Output enabled |
| 3 | DIO3 | R/W | 0h | Enables data output for DIO3.<br>0h = Output disabled<br>1h = Output enabled |



Copyright © 2026 Texas Instruments Incorporated

**Table 9-45. DOE31_0 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | DIO2 | R/W | 0h | Enables data output for DIO2.<br>0h = Output disabled<br>1h = Output enabled |
| 1 | DIO1 | R/W | 0h | Enables data output for DIO1.<br>0h = Output disabled<br>1h = Output enabled |
| 0 | DIO0 | R/W | 0h | Enables data output for DIO0.<br>0h = Output disabled<br>1h = Output enabled |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.43 DOESET31_0 (Offset = 12D0h) [Reset = 00000000h]

DOESET31_0 is shown in Figure 9-46 and described in Table 9-46.

Return to the Summary Table.

Writing 1 to a bit position in this register sets the corresponding bit in the DOE31_0 register.

**Figure 9-46. DOESET31_0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DIO31 | DIO30 | DIO29 | DIO28 | DIO27 | DIO26 | DIO25 | DIO24 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DIO23 | DIO22 | DIO21 | DIO20 | DIO19 | DIO18 | DIO17 | DIO16 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DIO15 | DIO14 | DIO13 | DIO12 | DIO11 | DIO10 | DIO9 | DIO8 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DIO7 | DIO6 | DIO5 | DIO4 | DIO3 | DIO2 | DIO1 | DIO0 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

**Table 9-46. DOESET31_0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | DIO31 | W | 0h | Writing 1 to this bit sets the DIO31 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO31 in DOE31_0 |
| 30 | DIO30 | W | 0h | Writing 1 to this bit sets the DIO30 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO30 in DOE31_0 |
| 29 | DIO29 | W | 0h | Writing 1 to this bit sets the DIO29 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO29 in DOE31_0 |
| 28 | DIO28 | W | 0h | Writing 1 to this bit sets the DIO28 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO28 in DOE31_0 |
| 27 | DIO27 | W | 0h | Writing 1 to this bit sets the DIO27 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO27 in DOE31_0 |
| 26 | DIO26 | W | 0h | Writing 1 to this bit sets the DIO26 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO26 in DOE31_0 |
| 25 | DIO25 | W | 0h | Writing 1 to this bit sets the DIO25 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO25 in DOE31_0 |
| 24 | DIO24 | W | 0h | Writing 1 to this bit sets the DIO24 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO24 in DOE31_0 |



Copyright © 2026 Texas Instruments Incorporated

**Table 9-46. DOESET31_0 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 23 | DIO23 | W | 0h | Writing 1 to this bit sets the DIO23 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO23 in DOE31_0 |
| 22 | DIO22 | W | 0h | Writing 1 to this bit sets the DIO22 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO22 in DOE31_0 |
| 21 | DIO21 | W | 0h | Writing 1 to this bit sets the DIO21 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO21 in DOE31_0 |
| 20 | DIO20 | W | 0h | Writing 1 to this bit sets the DIO20 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO20 in DOE31_0 |
| 19 | DIO19 | W | 0h | Writing 1 to this bit sets the DIO19 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO19 in DOE31_0 |
| 18 | DIO18 | W | 0h | Writing 1 to this bit sets the DIO18 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO18 in DOE31_0 |
| 17 | DIO17 | W | 0h | Writing 1 to this bit sets the DIO17 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO17 in DOE31_0 |
| 16 | DIO16 | W | 0h | Writing 1 to this bit sets the DIO16 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO16 in DOE31_0 |
| 15 | DIO15 | W | 0h | Writing 1 to this bit sets the DIO15 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO15 in DOE31_0 |
| 14 | DIO14 | W | 0h | Writing 1 to this bit sets the DIO14 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO14 in DOE31_0 |
| 13 | DIO13 | W | 0h | Writing 1 to this bit sets the DIO13 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO13 in DOE31_0 |
| 12 | DIO12 | W | 0h | Writing 1 to this bit sets the DIO12 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO12 in DOE31_0 |
| 11 | DIO11 | W | 0h | Writing 1 to this bit sets the DIO11 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO11 in DOE31_0 |
| 10 | DIO10 | W | 0h | Writing 1 to this bit sets the DIO10 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO10 in DOE31_0 |



Copyright © 2026 Texas Instruments Incorporated

**Table 9-46. DOESET31_0 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 9 | DIO9 | W | 0h | Writing 1 to this bit sets the DIO9 bit in the DOE31_0 register. Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO9 in DOE31_0 |
| 8 | DIO8 | W | 0h | Writing 1 to this bit sets the DIO8 bit in the DOE31_0 register. Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO8 in DOE31_0 |
| 7 | DIO7 | W | 0h | Writing 1 to this bit sets the DIO7 bit in the DOE31_0 register. Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO7 in DOE31_0 |
| 6 | DIO6 | W | 0h | Writing 1 to this bit sets the DIO6 bit in the DOE31_0 register. Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO6 in DOE31_0 |
| 5 | DIO5 | W | 0h | Writing 1 to this bit sets the DIO5 bit in the DOE31_0 register. Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO5 in DOE31_0 |
| 4 | DIO4 | W | 0h | Writing 1 to this bit sets the DIO4 bit in the DOE31_0 register. Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO4 in DOE31_0 |
| 3 | DIO3 | W | 0h | Writing 1 to this bit sets the DIO3 bit in the DOE31_0 register. Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO3 in DOE31_0 |
| 2 | DIO2 | W | 0h | Writing 1 to this bit sets the DIO2 bit in the DOE31_0 register. Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO2 in DOE31_0 |
| 1 | DIO1 | W | 0h | Writing 1 to this bit sets the DIO1 bit in the DOE31_0 register. Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO1 in DOE31_0 |
| 0 | DIO0 | W | 0h | Writing 1 to this bit sets the DIO0 bit in the DOE31_0 register. Writing 0 has no effect.<br>0h = No effect<br>1h = Sets DIO0 in DOE31_0 |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.44 DOECLR31_0 (Offset = 12E0h) [Reset = 00000000h]

DOECLR31_0 is shown in Figure 9-47 and described in Table 9-47.

Return to the Summary Table.

Writing 1 to a bit position in this register clears the corresponding bit in the DOE31_0 register.

**Figure 9-47. DOECLR31_0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DIO31 | DIO30 | DIO29 | DIO28 | DIO27 | DIO26 | DIO25 | DIO24 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DIO23 | DIO22 | DIO21 | DIO20 | DIO19 | DIO18 | DIO17 | DIO16 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DIO15 | DIO14 | DIO13 | DIO12 | DIO11 | DIO10 | DIO9 | DIO8 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DIO7 | DIO6 | DIO5 | DIO4 | DIO3 | DIO2 | DIO1 | DIO0 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

**Table 9-47. DOECLR31_0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | DIO31 | W | 0h | Writing 1 to this bit clears the DIO31 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO31 in DOE31_0 |
| 30 | DIO30 | W | 0h | Writing 1 to this bit clears the DIO30 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO30 in DOE31_0 |
| 29 | DIO29 | W | 0h | Writing 1 to this bit clears the DIO29 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO29 in DOE31_0 |
| 28 | DIO28 | W | 0h | Writing 1 to this bit clears the DIO28 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO28 in DOE31_0 |
| 27 | DIO27 | W | 0h | Writing 1 to this bit clears the DIO27 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO27 in DOE31_0 |
| 26 | DIO26 | W | 0h | Writing 1 to this bit clears the DIO26 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO26 in DOE31_0 |
| 25 | DIO25 | W | 0h | Writing 1 to this bit clears the DIO25 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO25 in DOE31_0 |
| 24 | DIO24 | W | 0h | Writing 1 to this bit clears the DIO24 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO24 in DOE31_0 |



Copyright © 2026 Texas Instruments Incorporated

**Table 9-47. DOECLR31_0 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 23 | DIO23 | W | 0h | Writing 1 to this bit clears the DIO23 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO23 in DOE31_0 |
| 22 | DIO22 | W | 0h | Writing 1 to this bit clears the DIO22 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO22 in DOE31_0 |
| 21 | DIO21 | W | 0h | Writing 1 to this bit clears the DIO21 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO21 in DOE31_0 |
| 20 | DIO20 | W | 0h | Writing 1 to this bit clears the DIO20 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO20 in DOE31_0 |
| 19 | DIO19 | W | 0h | Writing 1 to this bit clears the DIO19 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO19 in DOE31_0 |
| 18 | DIO18 | W | 0h | Writing 1 to this bit clears the DIO18 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO18 in DOE31_0 |
| 17 | DIO17 | W | 0h | Writing 1 to this bit clears the DIO17 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO17 in DOE31_0 |
| 16 | DIO16 | W | 0h | Writing 1 to this bit clears the DIO16 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO16 in DOE31_0 |
| 15 | DIO15 | W | 0h | Writing 1 to this bit clears the DIO15 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO15 in DOE31_0 |
| 14 | DIO14 | W | 0h | Writing 1 to this bit clears the DIO14 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO14 in DOE31_0 |
| 13 | DIO13 | W | 0h | Writing 1 to this bit clears the DIO13 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO13 in DOE31_0 |
| 12 | DIO12 | W | 0h | Writing 1 to this bit clears the DIO12 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO12 in DOE31_0 |
| 11 | DIO11 | W | 0h | Writing 1 to this bit clears the DIO11 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO11 in DOE31_0 |
| 10 | DIO10 | W | 0h | Writing 1 to this bit clears the DIO10 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO10 in DOE31_0 |



Copyright © 2026 Texas Instruments Incorporated

**Table 9-47. DOECLR31_0 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 9 | DIO9 | W | 0h | Writing 1 to this bit clears the DIO9 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO9 in DOE31_0 |
| 8 | DIO8 | W | 0h | Writing 1 to this bit clears the DIO8 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO8 in DOE31_0 |
| 7 | DIO7 | W | 0h | Writing 1 to this bit clears the DIO7 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO7 in DOE31_0 |
| 6 | DIO6 | W | 0h | Writing 1 to this bit clears the DIO6 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO6 in DOE31_0 |
| 5 | DIO5 | W | 0h | Writing 1 to this bit clears the DIO5 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO5 in DOE31_0 |
| 4 | DIO4 | W | 0h | Writing 1 to this bit clears the DIO4 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO4 in DOE31_0 |
| 3 | DIO3 | W | 0h | Writing 1 to this bit clears the DIO3 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO3 in DOE31_0 |
| 2 | DIO2 | W | 0h | Writing 1 to this bit clears the DIO2 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO2 in DOE31_0 |
| 1 | DIO1 | W | 0h | Writing 1 to this bit clears the DIO1 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO1 in DOE31_0 |
| 0 | DIO0 | W | 0h | Writing 1 to this bit clears the DIO0 bit in the DOE31_0 register.<br>Writing 0 has no effect.<br>0h = No effect<br>1h = Clears DIO0 in DOE31_0 |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.45 DIN3_0 (Offset = 1300h) [Reset = 00000000h]

DIN3_0 is shown in Figure 9-48 and described in Table 9-48.

Return to the Summary Table.

Data input from pins configured as DIO3 to DIO0. This is an alias register for byte access to bits 3 to 0 in DIN31_0 register.

**Figure 9-48. DIN3_0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | DIO3 |
| R-0h | | | | | | | R-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | DIO2 |
| R-0h | | | | | | | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | DIO1 |
| R-0h | | | | | | | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | DIO0 |
| R-0h | | | | | | | R-0h |

**Table 9-48. DIN3_0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-25 | RESERVED | R | 0h | |
| 24 | DIO3 | R | 0h | This bit reads the data input value of DIO3.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 23-17 | RESERVED | R | 0h | |
| 16 | DIO2 | R | 0h | This bit reads the data input value of DIO2.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 15-9 | RESERVED | R | 0h | |
| 8 | DIO1 | R | 0h | This bit reads the data input value of DIO1.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 7-1 | RESERVED | R | 0h | |
| 0 | DIO0 | R | 0h | This bit reads the data input value of DIO0.<br>0h = Input value is 0<br>1h = Input value is 1 |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.46 DIN7_4 (Offset = 1304h) [Reset = 00000000h]

DIN7_4 is shown in Figure 9-49 and described in Table 9-49.

Return to the Summary Table.

Data input from pins configured as DIO7 to DIO4. This is an alias register for byte access to bits 7 to 4 in DIN31_0 register.

**Figure 9-49. DIN7_4**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | DIO7 |
| R-0h | | | | | | | R-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | DIO6 |
| R-0h | | | | | | | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | DIO5 |
| R-0h | | | | | | | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | DIO4 |
| R-0h | | | | | | | R-0h |

**Table 9-49. DIN7_4 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-25 | RESERVED | R | 0h | |
| 24 | DIO7 | R | 0h | This bit reads the data input value of DIO7.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 23-17 | RESERVED | R | 0h | |
| 16 | DIO6 | R | 0h | This bit reads the data input value of DIO6.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 15-9 | RESERVED | R | 0h | |
| 8 | DIO5 | R | 0h | This bit reads the data input value of DIO5.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 7-1 | RESERVED | R | 0h | |
| 0 | DIO4 | R | 0h | This bit reads the data input value of DIO4.<br>0h = Input value is 0<br>1h = Input value is 1 |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.47 DIN11_8 (Offset = 1308h) [Reset = 00000000h]

DIN11_8 is shown in Figure 9-50 and described in Table 9-50.

Return to the Summary Table.

Data input from pins configured as DIO11 to DIO8. This is an alias register for byte access to bits 11 to 8 in DIN31_0 register.

**Figure 9-50. DIN11_8**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | DIO11 |
| R-0h | | | | | | | R-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | DIO10 |
| R-0h | | | | | | | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | DIO9 |
| R-0h | | | | | | | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | DIO8 |
| R-0h | | | | | | | R-0h |

**Table 9-50. DIN11_8 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-25 | RESERVED | R | 0h | |
| 24 | DIO11 | R | 0h | This bit reads the data input value of DIO11.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 23-17 | RESERVED | R | 0h | |
| 16 | DIO10 | R | 0h | This bit reads the data input value of DIO10.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 15-9 | RESERVED | R | 0h | |
| 8 | DIO9 | R | 0h | This bit reads the data input value of DIO9.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 7-1 | RESERVED | R | 0h | |
| 0 | DIO8 | R | 0h | This bit reads the data input value of DIO8.<br>0h = Input value is 0<br>1h = Input value is 1 |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.48 DIN15_12 (Offset = 130Ch) [Reset = 00000000h]

DIN15_12 is shown in Figure 9-51 and described in Table 9-51.

Return to the Summary Table.

Data input from pins configured as DIO15 to DIO12. This is an alias register for byte access to bits 15 to 12 in DIN31_0 register.

**Figure 9-51. DIN15_12**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | DIO15 |
| R-0h | | | | | | | R-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | DIO14 |
| R-0h | | | | | | | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | DIO13 |
| R-0h | | | | | | | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | DIO12 |
| R-0h | | | | | | | R-0h |

**Table 9-51. DIN15_12 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-25 | RESERVED | R | 0h | |
| 24 | DIO15 | R | 0h | This bit reads the data input value of DIO15.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 23-17 | RESERVED | R | 0h | |
| 16 | DIO14 | R | 0h | This bit reads the data input value of DIO14.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 15-9 | RESERVED | R | 0h | |
| 8 | DIO13 | R | 0h | This bit reads the data input value of DIO13.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 7-1 | RESERVED | R | 0h | |
| 0 | DIO12 | R | 0h | This bit reads the data input value of DIO12.<br>0h = Input value is 0<br>1h = Input value is 1 |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.49 DIN19_16 (Offset = 1310h) [Reset = 00000000h]

DIN19_16 is shown in Figure 9-52 and described in Table 9-52.

Return to the Summary Table.

Data input from pins configured as DIO19 to DIO16. This is an alias register for byte access to bits 19 to 16 in DIN31_0 register.

**Figure 9-52. DIN19_16**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | DIO19 |
| R-0h | | | | | | | R-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | DIO18 |
| R-0h | | | | | | | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | DIO17 |
| R-0h | | | | | | | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | DIO16 |
| R-0h | | | | | | | R-0h |

**Table 9-52. DIN19_16 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-25 | RESERVED | R | 0h | |
| 24 | DIO19 | R | 0h | This bit reads the data input value of DIO19.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 23-17 | RESERVED | R | 0h | |
| 16 | DIO18 | R | 0h | This bit reads the data input value of DIO18.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 15-9 | RESERVED | R | 0h | |
| 8 | DIO17 | R | 0h | This bit reads the data input value of DIO17.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 7-1 | RESERVED | R | 0h | |
| 0 | DIO16 | R | 0h | This bit reads the data input value of DIO16.<br>0h = Input value is 0<br>1h = Input value is 1 |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.50 DIN23_20 (Offset = 1314h) [Reset = 00000000h]

DIN23_20 is shown in Figure 9-53 and described in Table 9-53.

Return to the Summary Table.

Data input from pins configured as DIO23 to DIO20. This is an alias register for byte access to bits 23 to 20 in DIN31_0 register.

**Figure 9-53. DIN23_20**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | DIO23 |
| R-0h | | | | | | | R-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | DIO22 |
| R-0h | | | | | | | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | DIO21 |
| R-0h | | | | | | | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | DIO20 |
| R-0h | | | | | | | R-0h |

**Table 9-53. DIN23_20 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-25 | RESERVED | R | 0h | |
| 24 | DIO23 | R | 0h | This bit reads the data input value of DIO23.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 23-17 | RESERVED | R | 0h | |
| 16 | DIO22 | R | 0h | This bit reads the data input value of DIO22.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 15-9 | RESERVED | R | 0h | |
| 8 | DIO21 | R | 0h | This bit reads the data input value of DIO21.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 7-1 | RESERVED | R | 0h | |
| 0 | DIO20 | R | 0h | This bit reads the data input value of DIO20.<br>0h = Input value is 0<br>1h = Input value is 1 |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.51 DIN27_24 (Offset = 1318h) [Reset = 00000000h]

DIN27_24 is shown in Figure 9-54 and described in Table 9-54.

Return to the Summary Table.

Data input from pins configured as DIO27 to DIO24. This is an alias register for byte access to bits 27 to 24 in DIN31_0 register.

**Figure 9-54. DIN27_24**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | DIO27 |
| R-0h | | | | | | | R-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | DIO26 |
| R-0h | | | | | | | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | DIO25 |
| R-0h | | | | | | | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | DIO24 |
| R-0h | | | | | | | R-0h |

**Table 9-54. DIN27_24 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-25 | RESERVED | R | 0h | |
| 24 | DIO27 | R | 0h | This bit reads the data input value of DIO27.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 23-17 | RESERVED | R | 0h | |
| 16 | DIO26 | R | 0h | This bit reads the data input value of DIO26.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 15-9 | RESERVED | R | 0h | |
| 8 | DIO25 | R | 0h | This bit reads the data input value of DIO25.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 7-1 | RESERVED | R | 0h | |
| 0 | DIO24 | R | 0h | This bit reads the data input value of DIO24.<br>0h = Input value is 0<br>1h = Input value is 1 |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.52 DIN31_28 (Offset = 131Ch) [Reset = 00000000h]

DIN31_28 is shown in Figure 9-55 and described in Table 9-55.

Return to the Summary Table.

Data input from pins configured as DIO31 to DIO28. This is an alias register for byte access to bits 31 to 28 in DIN31_0 register.

**Figure 9-55. DIN31_28**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | DIO31 |
| R-0h | | | | | | | R-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | DIO30 |
| R-0h | | | | | | | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | DIO29 |
| R-0h | | | | | | | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | DIO28 |
| R-0h | | | | | | | R-0h |

**Table 9-55. DIN31_28 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-25 | RESERVED | R | 0h | |
| 24 | DIO31 | R | 0h | This bit reads the data input value of DIO31.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 23-17 | RESERVED | R | 0h | |
| 16 | DIO30 | R | 0h | This bit reads the data input value of DIO30.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 15-9 | RESERVED | R | 0h | |
| 8 | DIO29 | R | 0h | This bit reads the data input value of DIO29.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 7-1 | RESERVED | R | 0h | |
| 0 | DIO28 | R | 0h | This bit reads the data input value of DIO28.<br>0h = Input value is 0<br>1h = Input value is 1 |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.53 DIN31_0 (Offset = 1380h) [Reset = 00000000h]

DIN31_0 is shown in Figure 9-56 and described in Table 9-56.

Return to the Summary Table.

Data input value for pins configured as DIO31 to DIO0.

**Figure 9-56. DIN31_0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DIO31 | DIO30 | DIO29 | DIO28 | DIO27 | DIO26 | DIO25 | DIO24 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DIO23 | DIO22 | DIO21 | DIO20 | DIO19 | DIO18 | DIO17 | DIO16 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DIO15 | DIO14 | DIO13 | DIO12 | DIO11 | DIO10 | DIO9 | DIO8 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DIO7 | DIO6 | DIO5 | DIO4 | DIO3 | DIO2 | DIO1 | DIO0 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 9-56. DIN31_0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | DIO31 | R | 0h | This bit reads the data input value of DIO31.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 30 | DIO30 | R | 0h | This bit reads the data input value of DIO30.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 29 | DIO29 | R | 0h | This bit reads the data input value of DIO29.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 28 | DIO28 | R | 0h | This bit reads the data input value of DIO28.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 27 | DIO27 | R | 0h | This bit reads the data input value of DIO27.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 26 | DIO26 | R | 0h | This bit reads the data input value of DIO26.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 25 | DIO25 | R | 0h | This bit reads the data input value of DIO25.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 24 | DIO24 | R | 0h | This bit reads the data input value of DIO24.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 23 | DIO23 | R | 0h | This bit reads the data input value of DIO23.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 22 | DIO22 | R | 0h | This bit reads the data input value of DIO22.<br>0h = Input value is 0<br>1h = Input value is 1 |



Copyright © 2026 Texas Instruments Incorporated

**Table 9-56. DIN31_0 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 21 | DIO21 | R | 0h | This bit reads the data input value of DIO21.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 20 | DIO20 | R | 0h | This bit reads the data input value of DIO20.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 19 | DIO19 | R | 0h | This bit reads the data input value of DIO19.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 18 | DIO18 | R | 0h | This bit reads the data input value of DIO18.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 17 | DIO17 | R | 0h | This bit reads the data input value of DIO17.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 16 | DIO16 | R | 0h | This bit reads the data input value of DIO16.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 15 | DIO15 | R | 0h | This bit reads the data input value of DIO15.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 14 | DIO14 | R | 0h | This bit reads the data input value of DIO14.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 13 | DIO13 | R | 0h | This bit reads the data input value of DIO13.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 12 | DIO12 | R | 0h | This bit reads the data input value of DIO12.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 11 | DIO11 | R | 0h | This bit reads the data input value of DIO11.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 10 | DIO10 | R | 0h | This bit reads the data input value of DIO10.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 9 | DIO9 | R | 0h | This bit reads the data input value of DIO9.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 8 | DIO8 | R | 0h | This bit reads the data input value of DIO8.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 7 | DIO7 | R | 0h | This bit reads the data input value of DIO7.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 6 | DIO6 | R | 0h | This bit reads the data input value of DIO6.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 5 | DIO5 | R | 0h | This bit reads the data input value of DIO5.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 4 | DIO4 | R | 0h | This bit reads the data input value of DIO4.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 3 | DIO3 | R | 0h | This bit reads the data input value of DIO3.<br>0h = Input value is 0<br>1h = Input value is 1 |



Copyright © 2026 Texas Instruments Incorporated

**Table 9-56. DIN31_0 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | DIO2 | R | 0h | This bit reads the data input value of DIO2.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 1 | DIO1 | R | 0h | This bit reads the data input value of DIO1.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 0 | DIO0 | R | 0h | This bit reads the data input value of DIO0.<br>0h = Input value is 0<br>1h = Input value is 1 |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.54 POLARITY15_0 (Offset = 1390h) [Reset = 00000000h]

POLARITY15_0 is shown in Figure 9-57 and described in Table 9-57.

Return to the Summary Table.

This register is used to enable and configure the polarity for input edge detection on DIO15 to DIO0. The corresponding DIO bits in RIS register will be set when the input event matches the configured polarity.

**Figure 9-57. POLARITY15_0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DIO15 | | DIO14 | | DIO13 | | DIO12 | | DIO11 | | DIO10 | | DIO9 | | DIO8 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DIO7 | | DIO6 | | DIO5 | | DIO4 | | DIO3 | | DIO2 | | DIO1 | | DIO0 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

**Table 9-57. POLARITY15_0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | DIO15 | R/W | 0h | Enables and configures edge detection polarity for DIO15.<br>0h = Edge detection disabled<br>1h = Detects rising edge of input event<br>2h = Detects falling edge of input event<br>3h = Detects both rising and falling edge of input event |
| 29-28 | DIO14 | R/W | 0h | Enables and configures edge detection polarity for DIO14.<br>0h = Edge detection disabled<br>1h = Detects rising edge of input event<br>2h = Detects falling edge of input event<br>3h = Detects both rising and falling edge of input event |
| 27-26 | DIO13 | R/W | 0h | Enables and configures edge detection polarity for DIO13.<br>0h = Edge detection disabled<br>1h = Detects rising edge of input event<br>2h = Detects falling edge of input event<br>3h = Detects both rising and falling edge of input event |
| 25-24 | DIO12 | R/W | 0h | Enables and configures edge detection polarity for DIO12.<br>0h = Edge detection disabled<br>1h = Detects rising edge of input event<br>2h = Detects falling edge of input event<br>3h = Detects both rising and falling edge of input event |
| 23-22 | DIO11 | R/W | 0h | Enables and configures edge detection polarity for DIO11.<br>0h = Edge detection disabled<br>1h = Detects rising edge of input event<br>2h = Detects falling edge of input event<br>3h = Detects both rising and falling edge of input event |
| 21-20 | DIO10 | R/W | 0h | Enables and configures edge detection polarity for DIO10.<br>0h = Edge detection disabled<br>1h = Detects rising edge of input event<br>2h = Detects falling edge of input event<br>3h = Detects both rising and falling edge of input event |
| 19-18 | DIO9 | R/W | 0h | Enables and configures edge detection polarity for DIO9.<br>0h = Edge detection disabled<br>1h = Detects rising edge of input event<br>2h = Detects falling edge of input event<br>3h = Detects both rising and falling edge of input event |
| 17-16 | DIO8 | R/W | 0h | Enables and configures edge detection polarity for DIO8.<br>0h = Edge detection disabled<br>1h = Detects rising edge of input event<br>2h = Detects falling edge of input event<br>3h = Detects both rising and falling edge of input event |



Copyright © 2026 Texas Instruments Incorporated

**Table 9-57. POLARITY15_0 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15-14 | DIO7 | R/W | 0h | Enables and configures edge detection polarity for DIO7.<br>0h = Edge detection disabled<br>1h = Detects rising edge of input event<br>2h = Detects falling edge of input event<br>3h = Detects both rising and falling edge of input event |
| 13-12 | DIO6 | R/W | 0h | Enables and configures edge detection polarity for DIO6.<br>0h = Edge detection disabled<br>1h = Detects rising edge of input event<br>2h = Detects falling edge of input event<br>3h = Detects both rising and falling edge of input event |
| 11-10 | DIO5 | R/W | 0h | Enables and configures edge detection polarity for DIO5.<br>0h = Edge detection disabled<br>1h = Detects rising edge of input event<br>2h = Detects falling edge of input event<br>3h = Detects both rising and falling edge of input event |
| 9-8 | DIO4 | R/W | 0h | Enables and configures edge detection polarity for DIO4.<br>0h = Edge detection disabled<br>1h = Detects rising edge of input event<br>2h = Detects falling edge of input event<br>3h = Detects both rising and falling edge of input event |
| 7-6 | DIO3 | R/W | 0h | Enables and configures edge detection polarity for DIO3.<br>0h = Edge detection disabled<br>1h = Detects rising edge of input event<br>2h = Detects falling edge of input event<br>3h = Detects both rising and falling edge of input event |
| 5-4 | DIO2 | R/W | 0h | Enables and configures edge detection polarity for DIO2.<br>0h = Edge detection disabled<br>1h = Detects rising edge of input event<br>2h = Detects falling edge of input event<br>3h = Detects both rising and falling edge of input event |
| 3-2 | DIO1 | R/W | 0h | Enables and configures edge detection polarity for DIO1.<br>0h = Edge detection disabled<br>1h = Detects rising edge of input event<br>2h = Detects falling edge of input event<br>3h = Detects both rising and falling edge of input event |
| 1-0 | DIO0 | R/W | 0h | Enables and configures edge detection polarity for DIO0.<br>0h = Edge detection disabled<br>1h = Detects rising edge of input event<br>2h = Detects falling edge of input event<br>3h = Detects both rising and falling edge of input event |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.55 POLARITY31_16 (Offset = 13A0h) [Reset = 00000000h]

POLARITY31_16 is shown in Figure 9-58 and described in Table 9-58.

Return to the Summary Table.

This register is used to enable and configure the polarity for input edge detection on DIO31 to DIO16. The corresponding DIO bits in RIS register will be set when the input event matches the configured polarity.

**Figure 9-58. POLARITY31_16**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DIO31 | | DIO30 | | DIO29 | | DIO28 | | DIO27 | | DIO26 | | DIO25 | | DIO24 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DIO23 | | DIO22 | | DIO21 | | DIO20 | | DIO19 | | DIO18 | | DIO17 | | DIO16 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

**Table 9-58. POLARITY31_16 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | DIO31 | R/W | 0h | Enables and configures edge detection polarity for DIO31.<br>0h = Edge detection disabled<br>1h = Detects rising edge of input event<br>2h = Detects falling edge of input event<br>3h = Detects both rising and falling edge of input event |
| 29-28 | DIO30 | R/W | 0h | Enables and configures edge detection polarity for DIO30.<br>0h = Edge detection disabled<br>1h = Detects rising edge of input event<br>2h = Detects falling edge of input event<br>3h = Detects both rising and falling edge of input event |
| 27-26 | DIO29 | R/W | 0h | Enables and configures edge detection polarity for DIO29.<br>0h = Edge detection disabled<br>1h = Detects rising edge of input event<br>2h = Detects falling edge of input event<br>3h = Detects both rising and falling edge of input event |
| 25-24 | DIO28 | R/W | 0h | Enables and configures edge detection polarity for DIO28.<br>0h = Edge detection disabled<br>1h = Detects rising edge of input event<br>2h = Detects falling edge of input event<br>3h = Detects both rising and falling edge of input event |
| 23-22 | DIO27 | R/W | 0h | Enables and configures edge detection polarity for DIO27.<br>0h = Edge detection disabled<br>1h = Detects rising edge of input event<br>2h = Detects falling edge of input event<br>3h = Detects both rising and falling edge of input event |
| 21-20 | DIO26 | R/W | 0h | Enables and configures edge detection polarity for DIO26.<br>0h = Edge detection disabled<br>1h = Detects rising edge of input event<br>2h = Detects falling edge of input event<br>3h = Detects both rising and falling edge of input event |
| 19-18 | DIO25 | R/W | 0h | Enables and configures edge detection polarity for DIO25.<br>0h = Edge detection disabled<br>1h = Detects rising edge of input event<br>2h = Detects falling edge of input event<br>3h = Detects both rising and falling edge of input event |
| 17-16 | DIO24 | R/W | 0h | Enables and configures edge detection polarity for DIO24.<br>0h = Edge detection disabled<br>1h = Detects rising edge of input event<br>2h = Detects falling edge of input event<br>3h = Detects both rising and falling edge of input event |



Copyright © 2026 Texas Instruments Incorporated

**Table 9-58. POLARITY31_16 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15-14 | DIO23 | R/W | 0h | Enables and configures edge detection polarity for DIO23.<br>0h = Edge detection disabled<br>1h = Detects rising edge of input event<br>2h = Detects falling edge of input event<br>3h = Detects both rising and falling edge of input event |
| 13-12 | DIO22 | R/W | 0h | Enables and configures edge detection polarity for DIO22.<br>0h = Edge detection disabled<br>1h = Detects rising edge of input event<br>2h = Detects falling edge of input event<br>3h = Detects both rising and falling edge of input event |
| 11-10 | DIO21 | R/W | 0h | Enables and configures edge detection polarity for DIO21.<br>0h = Edge detection disabled<br>1h = Detects rising edge of input event<br>2h = Detects falling edge of input event<br>3h = Detects both rising and falling edge of input event |
| 9-8 | DIO20 | R/W | 0h | Enables and configures edge detection polarity for DIO20.<br>0h = Edge detection disabled<br>1h = Detects rising edge of input event<br>2h = Detects falling edge of input event<br>3h = Detects both rising and falling edge of input event |
| 7-6 | DIO19 | R/W | 0h | Enables and configures edge detection polarity for DIO19.<br>0h = Edge detection disabled<br>1h = Detects rising edge of input event<br>2h = Detects falling edge of input event<br>3h = Detects both rising and falling edge of input event |
| 5-4 | DIO18 | R/W | 0h | Enables and configures edge detection polarity for DIO18.<br>0h = Edge detection disabled<br>1h = Detects rising edge of input event<br>2h = Detects falling edge of input event<br>3h = Detects both rising and falling edge of input event |
| 3-2 | DIO17 | R/W | 0h | Enables and configures edge detection polarity for DIO17.<br>0h = Edge detection disabled<br>1h = Detects rising edge of input event<br>2h = Detects falling edge of input event<br>3h = Detects both rising and falling edge of input event |
| 1-0 | DIO16 | R/W | 0h | Enables and configures edge detection polarity for DIO16.<br>0h = Edge detection disabled<br>1h = Detects rising edge of input event<br>2h = Detects falling edge of input event<br>3h = Detects both rising and falling edge of input event |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.56 CTL (Offset = 1400h) [Reset = 00000000h]

CTL is shown in Figure 9-59 and described in Table 9-59.

Return to the Summary Table.

GPIO Control Register

**Figure 9-59. CTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | FASTWAKEONLY |
| R/W-0h | | | | | | | R/W-0h |

**Table 9-59. CTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R/W | 0h | |
| 0 | FASTWAKEONLY | R/W | 0h | FASTWAKEONLY for the global control of fastwake<br>0h = The global control of fastwake is not enabled, per bit fast wake feature depends on FASTWAKE.DIN<br>1h = The global control of fastwake is enabled |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.57 FASTWAKE (Offset = 1404h) [Reset = 00000000h]

FASTWAKE is shown in Figure 9-60 and described in Table 9-60.

Return to the Summary Table.

This is per bit fast wake enable for the bit slice, allows the GPIO module to stay in a low power state and not require high speed clocking of the input synchronizer or filter

**Figure 9-60. FASTWAKE**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DIN31 | DIN30 | DIN29 | DIN28 | DIN27 | DIN26 | DIN25 | DIN24 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DIN23 | DIN22 | DIN21 | DIN20 | DIN19 | DIN18 | DIN17 | DIN16 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DIN15 | DIN14 | DIN13 | DIN12 | DIN11 | DIN10 | DIN9 | DIN8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DIN7 | DIN6 | DIN5 | DIN4 | DIN3 | DIN2 | DIN1 | DIN0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

**Table 9-60. FASTWAKE Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | DIN31 | R/W | 0h | Enable fastwake feature for DIN31<br>0h = fastwake feature is disabled<br>1h = fastwake feature is enabled |
| 30 | DIN30 | R/W | 0h | Enable fastwake feature for DIN30<br>0h = fastwake feature is disabled<br>1h = fastwake feature is enabled |
| 29 | DIN29 | R/W | 0h | Enable fastwake feature for DIN29<br>0h = fastwake feature is disabled<br>1h = fastwake feature is enabled |
| 28 | DIN28 | R/W | 0h | Enable fastwake feature for DIN29<br>0h = fastwake feature is disabled<br>1h = fastwake feature is enabled |
| 27 | DIN27 | R/W | 0h | Enable fastwake feature for DIN27<br>0h = fastwake feature is disabled<br>1h = fastwake feature is enabled |
| 26 | DIN26 | R/W | 0h | Enable fastwake feature for DIN26<br>0h = fastwake feature is disabled<br>1h = fastwake feature is enabled |
| 25 | DIN25 | R/W | 0h | Enable fastwake feature for DIN25<br>0h = fastwake feature is disabled<br>1h = fastwake feature is enabled |
| 24 | DIN24 | R/W | 0h | Enable fastwake feature for DIN24<br>0h = fastwake feature is disabled<br>1h = fastwake feature is enabled |
| 23 | DIN23 | R/W | 0h | Enable fastwake feature for DIN23<br>0h = fastwake feature is disabled<br>1h = fastwake feature is enabled |
| 22 | DIN22 | R/W | 0h | Enable fastwake feature for DIN22<br>0h = fastwake feature is disabled<br>1h = fastwake feature is enabled |



Copyright © 2026 Texas Instruments Incorporated

**Table 9-60. FASTWAKE Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 21 | DIN21 | R/W | 0h | Enable fastwake feature for DIN21<br>0h = fastwake feature is disabled<br>1h = fastwake feature is enabled |
| 20 | DIN20 | R/W | 0h | Enable fastwake feature for DIN20<br>0h = fastwake feature is disabled<br>1h = fastwake feature is enabled |
| 19 | DIN19 | R/W | 0h | Enable fastwake feature for DIN19<br>0h = fastwake feature is disabled<br>1h = fastwake feature is enabled |
| 18 | DIN18 | R/W | 0h | Enable fastwake feature for DIN18<br>0h = fastwake feature is disabled<br>1h = fastwake feature is enabled |
| 17 | DIN17 | R/W | 0h | Enable fastwake feature for DIN17<br>0h = fastwake feature is disabled<br>1h = fastwake feature is enabled |
| 16 | DIN16 | R/W | 0h | Enable fastwake feature for DIN16<br>0h = fastwake feature is disabled<br>1h = fastwake feature is enabled |
| 15 | DIN15 | R/W | 0h | Enable fastwake feature for DIN15<br>0h = fastwake feature is disabled<br>1h = fastwake feature is enabled |
| 14 | DIN14 | R/W | 0h | Enable fastwake feature for DIN14<br>0h = fastwake feature is disabled<br>1h = fastwake feature is enabled |
| 13 | DIN13 | R/W | 0h | Enable fastwake feature for DIN13<br>0h = fastwake feature is disabled<br>1h = fastwake feature is enabled |
| 12 | DIN12 | R/W | 0h | Enable fastwake feature for DIN12<br>0h = fastwake feature is disabled<br>1h = fastwake feature is enabled |
| 11 | DIN11 | R/W | 0h | Enable fastwake feature for DIN11<br>0h = fastwake feature is disabled<br>1h = fastwake feature is enabled |
| 10 | DIN10 | R/W | 0h | Enable fastwake feature for DIN10<br>0h = fastwake feature is disabled<br>1h = fastwake feature is enabled |
| 9 | DIN9 | R/W | 0h | Enable fastwake feature for DIN9<br>0h = fastwake feature is disabled<br>1h = fastwake feature is enabled |
| 8 | DIN8 | R/W | 0h | Enable fastwake feature for DIN8<br>0h = fastwake feature is disabled<br>1h = fastwake feature is enabled |
| 7 | DIN7 | R/W | 0h | Enable fastwake feature for DIN7<br>0h = fastwake feature is disabled<br>1h = fastwake feature is enabled |
| 6 | DIN6 | R/W | 0h | Enable fastwake feature for DIN6<br>0h = fastwake feature is disabled<br>1h = fastwake feature is enabled |
| 5 | DIN5 | R/W | 0h | Enable fastwake feature for DIN5<br>0h = fastwake feature is disabled<br>1h = fastwake feature is enabled |
| 4 | DIN4 | R/W | 0h | Enable fastwake feature for DIN4<br>0h = fastwake feature is disabled<br>1h = fastwake feature is enabled |
| 3 | DIN3 | R/W | 0h | Enable fastwake feature for DIN3<br>0h = fastwake feature is disabled<br>1h = fastwake feature is enabled |



Copyright © 2026 Texas Instruments Incorporated

**Table 9-60. FASTWAKE Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | DIN2 | R/W | 0h | Enable fastwake feature for DIN2<br>0h = fastwake feature is disabled<br>1h = fastwake feature is enabled |
| 1 | DIN1 | R/W | 0h | Enable fastwake feature for DIN1<br>0h = fastwake feature is disabled<br>1h = fastwake feature is enabled |
| 0 | DIN0 | R/W | 0h | Enable fastwake feature for DIN0<br>0h = fastwake feature is disabled<br>1h = fastwake feature is enabled |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.58 SUB0CFG (Offset = 1500h) [Reset = 00000000h]

SUB0CFG is shown in Figure 9-61 and described in Table 9-61.

Return to the Summary Table.

This register is used to enable the subscriber 0 event and define the output policy on the selected DIO 0-15 pins.

**Figure 9-61. SUB0CFG**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | INDEX | | | |
| R/W-0h | | | | R/W-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | OUTPOLICY | |
| R/W-0h | | | | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | ENABLE |
| R/W-0h | | | | | | | R/W-0h |

**Table 9-61. SUB0CFG Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | RESERVED | R/W | 0h | |
| 19-16 | INDEX | R/W | 0h | Indicates the specific bit among lower 16 bits that is targeted by the subscriber action<br>0h = specific bit targeted by the subscriber action is bit0<br>Fh = specific bit targeted by the subscriber action is bit15 |
| 15-10 | RESERVED | R/W | 0h | |
| 9-8 | OUTPOLICY | R/W | 0h | These bits configure the output policy for subscriber 0 event.<br>0h = Selected DIO pins are set<br>1h = Selected DIO pins are cleared<br>2h = Selected DIO pins are toggled |
| 7-1 | RESERVED | R/W | 0h | |
| 0 | ENABLE | R/W | 0h | This bit is used to enable subscriber 0 event.<br>0h = Subscriber 0 event is disabled<br>1h = Subscriber 0 event is enabled |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.59 FILTEREN15_0 (Offset = 1508h) [Reset = 00000000h]

FILTEREN15_0 is shown in Figure 9-62 and described in Table 9-62.

Return to the Summary Table.

Programmable counter length of digital glitch filter for DIN0-DIN15

**Figure 9-62. FILTEREN15_0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DIN15 | | DIN14 | | DIN13 | | DIN12 | | DIN11 | | DIN10 | | DIN9 | | DIN8 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DIN7 | | DIN6 | | DIN5 | | DIN4 | | DIN3 | | DIN2 | | DIN1 | | DIN0 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

**Table 9-62. FILTEREN15_0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | DIN15 | R/W | 0h | Programmable counter length of digital glitch filter for DIN15<br>0h = No additional filter beyond the CDC synchronization sample<br>1h = 1 ULPCLK minimum sample<br>2h = 3 ULPCLK minimum sample<br>3h = 8 ULPCLK minimum sample |
| 29-28 | DIN14 | R/W | 0h | Programmable counter length of digital glitch filter for DIN14<br>0h = No additional filter beyond the CDC synchronization sample<br>1h = 1 ULPCLK minimum sample<br>2h = 3 ULPCLK minimum sample<br>3h = 8 ULPCLK minimum sample |
| 27-26 | DIN13 | R/W | 0h | Programmable counter length of digital glitch filter for DIN13<br>0h = No additional filter beyond the CDC synchronization sample<br>1h = 1 ULPCLK minimum sample<br>2h = 3 ULPCLK minimum sample<br>3h = 8 ULPCLK minimum sample |
| 25-24 | DIN12 | R/W | 0h | Programmable counter length of digital glitch filter for DIN12<br>0h = No additional filter beyond the CDC synchronization sample<br>1h = 1 ULPCLK minimum sample<br>2h = 3 ULPCLK minimum sample<br>3h = 8 ULPCLK minimum sample |
| 23-22 | DIN11 | R/W | 0h | Programmable counter length of digital glitch filter for DIN11<br>0h = No additional filter beyond the CDC synchronization sample<br>1h = 1 ULPCLK minimum sample<br>2h = 3 ULPCLK minimum sample<br>3h = 8 ULPCLK minimum sample |
| 21-20 | DIN10 | R/W | 0h | Programmable counter length of digital glitch filter for DIN10<br>0h = No additional filter beyond the CDC synchronization sample<br>1h = 1 ULPCLK minimum sample<br>2h = 3 ULPCLK minimum sample<br>3h = 8 ULPCLK minimum sample |
| 19-18 | DIN9 | R/W | 0h | Programmable counter length of digital glitch filter for DIN9<br>0h = No additional filter beyond the CDC synchronization sample<br>1h = 1 ULPCLK minimum sample<br>2h = 3 ULPCLK minimum sample<br>3h = 8 ULPCLK minimum sample |
| 17-16 | DIN8 | R/W | 0h | Programmable counter length of digital glitch filter for DIN8<br>0h = No additional filter beyond the CDC synchronization sample<br>1h = 1 ULPCLK minimum sample<br>2h = 3 ULPCLK minimum sample<br>3h = 8 ULPCLK minimum sample |



Copyright © 2026 Texas Instruments Incorporated

**Table 9-62. FILTEREN15_0 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15-14 | DIN7 | R/W | 0h | Programmable counter length of digital glitch filter for DIN7<br>0h = No additional filter beyond the CDC synchronization sample<br>1h = 1 ULPCLK minimum sample<br>2h = 3 ULPCLK minimum sample<br>3h = 8 ULPCLK minimum sample |
| 13-12 | DIN6 | R/W | 0h | Programmable counter length of digital glitch filter for DIN6<br>0h = No additional filter beyond the CDC synchronization sample<br>1h = 1 ULPCLK minimum sample<br>2h = 3 ULPCLK minimum sample<br>3h = 8 ULPCLK minimum sample |
| 11-10 | DIN5 | R/W | 0h | Programmable counter length of digital glitch filter for DIN5<br>0h = No additional filter beyond the CDC synchronization sample<br>1h = 1 ULPCLK minimum sample<br>2h = 3 ULPCLK minimum sample<br>3h = 8 ULPCLK minimum sample |
| 9-8 | DIN4 | R/W | 0h | Programmable counter length of digital glitch filter for DIN4<br>0h = No additional filter beyond the CDC synchronization sample<br>1h = 1 ULPCLK minimum sample<br>2h = 3 ULPCLK minimum sample<br>3h = 8 ULPCLK minimum sample |
| 7-6 | DIN3 | R/W | 0h | Programmable counter length of digital glitch filter for DIN3<br>0h = No additional filter beyond the CDC synchronization sample<br>1h = 1 ULPCLK minimum sample<br>2h = 3 ULPCLK minimum sample<br>3h = 8 ULPCLK minimum sample |
| 5-4 | DIN2 | R/W | 0h | Programmable counter length of digital glitch filter for DIN2<br>0h = No additional filter beyond the CDC synchronization sample<br>1h = 1 ULPCLK minimum sample<br>2h = 3 ULPCLK minimum sample<br>3h = 8 ULPCLK minimum sample |
| 3-2 | DIN1 | R/W | 0h | Programmable counter length of digital glitch filter for DIN1<br>0h = No additional filter beyond the CDC synchronization sample<br>1h = 1 ULPCLK minimum sample<br>2h = 3 ULPCLK minimum sample<br>3h = 8 ULPCLK minimum sample |
| 1-0 | DIN0 | R/W | 0h | Programmable counter length of digital glitch filter for DIN0<br>0h = No additional filter beyond the CDC synchronization sample<br>1h = 1 ULPCLK minimum sample<br>2h = 3 ULPCLK minimum sample<br>3h = 8 ULPCLK minimum sample |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.60 FILTEREN31_16 (Offset = 150Ch) [Reset = 00000000h]

FILTEREN31_16 is shown in Figure 9-63 and described in Table 9-63.

Return to the Summary Table.

Programmable counter length of digital glitch filter for DIN16-DIN31

**Figure 9-63. FILTEREN31_16**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DIN31 | | DIN30 | | DIN29 | | DIN28 | | DIN27 | | DIN26 | | DIN25 | | DIN24 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DIN23 | | DIN22 | | DIN21 | | DIN20 | | DIN19 | | DIN18 | | DIN17 | | DIN16 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

**Table 9-63. FILTEREN31_16 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | DIN31 | R/W | 0h | Programmable counter length of digital glitch filter for DIN31<br>0h = No additional filter beyond the CDC synchronization sample<br>1h = 1 ULPCLK minimum sample<br>2h = 3 ULPCLK minimum sample<br>3h = 8 ULPCLK minimum sample |
| 29-28 | DIN30 | R/W | 0h | Programmable counter length of digital glitch filter for DIN30<br>0h = No additional filter beyond the CDC synchronization sample<br>1h = 1 ULPCLK minimum sample<br>2h = 3 ULPCLK minimum sample<br>3h = 8 ULPCLK minimum sample |
| 27-26 | DIN29 | R/W | 0h | Programmable counter length of digital glitch filter for DIN29<br>0h = No additional filter beyond the CDC synchronization sample<br>1h = 1 ULPCLK minimum sample<br>2h = 3 ULPCLK minimum sample<br>3h = 8 ULPCLK minimum sample |
| 25-24 | DIN28 | R/W | 0h | Programmable counter length of digital glitch filter for DIN28<br>0h = No additional filter beyond the CDC synchronization sample<br>1h = 1 ULPCLK minimum sample<br>2h = 3 ULPCLK minimum sample<br>3h = 8 ULPCLK minimum sample |
| 23-22 | DIN27 | R/W | 0h | Programmable counter length of digital glitch filter for DIN27<br>0h = No additional filter beyond the CDC synchronization sample<br>1h = 1 ULPCLK minimum sample<br>2h = 3 ULPCLK minimum sample<br>3h = 8 ULPCLK minimum sample |
| 21-20 | DIN26 | R/W | 0h | Programmable counter length of digital glitch filter for DIN26<br>0h = No additional filter beyond the CDC synchronization sample<br>1h = 1 ULPCLK minimum sample<br>2h = 3 ULPCLK minimum sample<br>3h = 8 ULPCLK minimum sample |
| 19-18 | DIN25 | R/W | 0h | Programmable counter length of digital glitch filter for DIN25<br>0h = No additional filter beyond the CDC synchronization sample<br>1h = 1 ULPCLK minimum sample<br>2h = 3 ULPCLK minimum sample<br>3h = 8 ULPCLK minimum sample |
| 17-16 | DIN24 | R/W | 0h | Programmable counter length of digital glitch filter for DIN24<br>0h = No additional filter beyond the CDC synchronization sample<br>1h = 1 ULPCLK minimum sample<br>2h = 3 ULPCLK minimum sample<br>3h = 8 ULPCLK minimum sample |



Copyright © 2026 Texas Instruments Incorporated

**Table 9-63. FILTEREN31_16 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15-14 | DIN23 | R/W | 0h | Programmable counter length of digital glitch filter for DIN23<br>0h = No additional filter beyond the CDC synchronization sample<br>1h = 1 ULPCLK minimum sample<br>2h = 3 ULPCLK minimum sample<br>3h = 8 ULPCLK minimum sample |
| 13-12 | DIN22 | R/W | 0h | Programmable counter length of digital glitch filter for DIN22<br>0h = No additional filter beyond the CDC synchronization sample<br>1h = 1 ULPCLK minimum sample<br>2h = 3 ULPCLK minimum sample<br>3h = 8 ULPCLK minimum sample |
| 11-10 | DIN21 | R/W | 0h | Programmable counter length of digital glitch filter for DIN21<br>0h = No additional filter beyond the CDC synchronization sample<br>1h = 1 ULPCLK minimum sample<br>2h = 3 ULPCLK minimum sample<br>3h = 8 ULPCLK minimum sample |
| 9-8 | DIN20 | R/W | 0h | Programmable counter length of digital glitch filter for DIN20<br>0h = No additional filter beyond the CDC synchronization sample<br>1h = 1 ULPCLK minimum sample<br>2h = 3 ULPCLK minimum sample<br>3h = 8 ULPCLK minimum sample |
| 7-6 | DIN19 | R/W | 0h | Programmable counter length of digital glitch filter for DIN19<br>0h = No additional filter beyond the CDC synchronization sample<br>1h = 1 ULPCLK minimum sample<br>2h = 3 ULPCLK minimum sample<br>3h = 8 ULPCLK minimum sample |
| 5-4 | DIN18 | R/W | 0h | Programmable counter length of digital glitch filter for DIN18<br>0h = No additional filter beyond the CDC synchronization sample<br>1h = 1 ULPCLK minimum sample<br>2h = 3 ULPCLK minimum sample<br>3h = 8 ULPCLK minimum sample |
| 3-2 | DIN17 | R/W | 0h | Programmable counter length of digital glitch filter for DIN17<br>0h = No additional filter beyond the CDC synchronization sample<br>1h = 1 ULPCLK minimum sample<br>2h = 3 ULPCLK minimum sample<br>3h = 8 ULPCLK minimum sample |
| 1-0 | DIN16 | R/W | 0h | Programmable counter length of digital glitch filter for DIN16<br>0h = No additional filter beyond the CDC synchronization sample<br>1h = 1 ULPCLK minimum sample<br>2h = 3 ULPCLK minimum sample<br>3h = 8 ULPCLK minimum sample |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.61 DMAMASK (Offset = 1510h) [Reset = 00000000h]

DMAMASK is shown in Figure 9-64 and described in Table 9-64.

Return to the Summary Table.

DMA MASK which indicates which bit lanes the DMA is allowed to modify.

**Figure 9-64. DMAMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DOUT31 | DOUT30 | DOUT29 | DOUT28 | DOUT27 | DOUT26 | DOUT25 | DOUT24 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DOUT23 | DOUT22 | DOUT21 | DOUT20 | DOUT19 | DOUT18 | DOUT17 | DOUT16 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DOUT15 | DOUT14 | DOUT13 | DOUT12 | DOUT11 | DOUT10 | DOUT9 | DOUT8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DOUT7 | DOUT6 | DOUT5 | DOUT4 | DOUT3 | DOUT2 | DOUT1 | DOUT0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

**Table 9-64. DMAMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | DOUT31 | R/W | 0h | DMA is allowed to modify DOUT31<br>0h = DMA is not allowed to modify this bit lane<br>1h = DMA is allowed to modify this bit lane |
| 30 | DOUT30 | R/W | 0h | DMA is allowed to modify DOUT30<br>0h = DMA is not allowed to modify this bit lane<br>1h = DMA is allowed to modify this bit lane |
| 29 | DOUT29 | R/W | 0h | DMA is allowed to modify DOUT29<br>0h = DMA is not allowed to modify this bit lane<br>1h = DMA is allowed to modify this bit lane |
| 28 | DOUT28 | R/W | 0h | DMA is allowed to modify DOUT28<br>0h = DMA is not allowed to modify this bit lane<br>1h = DMA is allowed to modify this bit lane |
| 27 | DOUT27 | R/W | 0h | DMA is allowed to modify DOUT27<br>0h = DMA is not allowed to modify this bit lane<br>1h = DMA is allowed to modify this bit lane |
| 26 | DOUT26 | R/W | 0h | DMA is allowed to modify DOUT26<br>0h = DMA is not allowed to modify this bit lane<br>1h = DMA is allowed to modify this bit lane |
| 25 | DOUT25 | R/W | 0h | DMA is allowed to modify DOUT25<br>0h = DMA is not allowed to modify this bit lane<br>1h = DMA is allowed to modify this bit lane |
| 24 | DOUT24 | R/W | 0h | DMA is allowed to modify DOUT24<br>0h = DMA is not allowed to modify this bit lane<br>1h = DMA is allowed to modify this bit lane |
| 23 | DOUT23 | R/W | 0h | DMA is allowed to modify DOUT23<br>0h = DMA is not allowed to modify this bit lane<br>1h = DMA is allowed to modify this bit lane |
| 22 | DOUT22 | R/W | 0h | DMA is allowed to modify DOUT22<br>0h = DMA is not allowed to modify this bit lane<br>1h = DMA is allowed to modify this bit lane |



Copyright © 2026 Texas Instruments Incorporated

**Table 9-64. DMAMASK Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 21 | DOUT21 | R/W | 0h | DMA is allowed to modify DOUT21<br>0h = DMA is not allowed to modify this bit lane<br>1h = DMA is allowed to modify this bit lane |
| 20 | DOUT20 | R/W | 0h | DMA is allowed to modify DOUT20<br>0h = DMA is not allowed to modify this bit lane<br>1h = DMA is allowed to modify this bit lane |
| 19 | DOUT19 | R/W | 0h | DMA is allowed to modify DOUT19<br>0h = DMA is not allowed to modify this bit lane<br>1h = DMA is allowed to modify this bit lane |
| 18 | DOUT18 | R/W | 0h | DMA is allowed to modify DOUT18<br>0h = DMA is not allowed to modify this bit lane<br>1h = DMA is allowed to modify this bit lane |
| 17 | DOUT17 | R/W | 0h | DMA is allowed to modify DOUT17<br>0h = DMA is not allowed to modify this bit lane<br>1h = DMA is allowed to modify this bit lane |
| 16 | DOUT16 | R/W | 0h | DMA is allowed to modify DOUT16<br>0h = DMA is not allowed to modify this bit lane<br>1h = DMA is allowed to modify this bit lane |
| 15 | DOUT15 | R/W | 0h | DMA is allowed to modify DOUT15<br>0h = DMA is not allowed to modify this bit lane<br>1h = DMA is allowed to modify this bit lane |
| 14 | DOUT14 | R/W | 0h | DMA is allowed to modify DOUT14<br>0h = DMA is not allowed to modify this bit lane<br>1h = DMA is allowed to modify this bit lane |
| 13 | DOUT13 | R/W | 0h | DMA is allowed to modify DOUT13<br>0h = DMA is not allowed to modify this bit lane<br>1h = DMA is allowed to modify this bit lane |
| 12 | DOUT12 | R/W | 0h | DMA is allowed to modify DOUT12<br>0h = DMA is not allowed to modify this bit lane<br>1h = DMA is allowed to modify this bit lane |
| 11 | DOUT11 | R/W | 0h | DMA is allowed to modify DOUT11<br>0h = DMA is not allowed to modify this bit lane<br>1h = DMA is allowed to modify this bit lane |
| 10 | DOUT10 | R/W | 0h | DMA is allowed to modify DOUT10<br>0h = DMA is not allowed to modify this bit lane<br>1h = DMA is allowed to modify this bit lane |
| 9 | DOUT9 | R/W | 0h | DMA is allowed to modify DOUT9<br>0h = DMA is not allowed to modify this bit lane<br>1h = DMA is allowed to modify this bit lane |
| 8 | DOUT8 | R/W | 0h | DMA is allowed to modify DOUT8<br>0h = DMA is not allowed to modify this bit lane<br>1h = DMA is allowed to modify this bit lane |
| 7 | DOUT7 | R/W | 0h | DMA is allowed to modify DOUT7<br>0h = DMA is not allowed to modify this bit lane<br>1h = DMA is allowed to modify this bit lane |
| 6 | DOUT6 | R/W | 0h | DMA is allowed to modify DOUT6<br>0h = DMA is not allowed to modify this bit lane<br>1h = DMA is allowed to modify this bit lane |
| 5 | DOUT5 | R/W | 0h | DMA is allowed to modify DOUT5<br>0h = DMA is not allowed to modify this bit lane<br>1h = DMA is allowed to modify this bit lane |
| 4 | DOUT4 | R/W | 0h | DMA is allowed to modify DOUT4<br>0h = DMA is not allowed to modify this bit lane<br>1h = DMA is allowed to modify this bit lane |
| 3 | DOUT3 | R/W | 0h | DMA is allowed to modify DOUT3<br>0h = DMA is not allowed to modify this bit lane<br>1h = DMA is allowed to modify this bit lane |



Copyright © 2026 Texas Instruments Incorporated

**Table 9-64. DMAMASK Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | DOUT2 | R/W | 0h | DMA is allowed to modify DOUT2<br>0h = DMA is not allowed to modify this bit lane<br>1h = DMA is allowed to modify this bit lane |
| 1 | DOUT1 | R/W | 0h | DMA is allowed to modify DOUT1<br>0h = DMA is not allowed to modify this bit lane<br>1h = DMA is allowed to modify this bit lane |
| 0 | DOUT0 | R/W | 0h | DMA is allowed to modify DOUT0<br>0h = DMA is not allowed to modify this bit lane<br>1h = DMA is allowed to modify this bit lane |



Copyright © 2026 Texas Instruments Incorporated

### 9.3.62 SUB1CFG (Offset = 1520h) [Reset = 00000000h]

SUB1CFG is shown in Figure 9-65 and described in Table 9-65.

Return to the Summary Table.

This register is used to enable the subscriber 1 event and define the output policy on the selected DIO 16-31 pins.

**Figure 9-65. SUB1CFG**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | INDEX | | | |
| R/W-0h | | | | R/W-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | OUTPOLICY | |
| R/W-0h | | | | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | ENABLE |
| R/W-0h | | | | | | | R/W-0h |

**Table 9-65. SUB1CFG Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | RESERVED | R/W | 0h | |
| 19-16 | INDEX | R/W | 0h | indicates the specific bit in the upper 16 bits that is targeted by the subscriber action<br>0h = specific bit targeted by the subscriber action is bit16<br>Fh = specific bit targeted by the subscriber action is bit31 |
| 15-10 | RESERVED | R/W | 0h | |
| 9-8 | OUTPOLICY | R/W | 0h | These bits configure the output policy for subscriber 1 event.<br>0h = Selected DIO pins are set<br>1h = Selected DIO pins are cleared<br>2h = Selected DIO pins are toggled |
| 7-1 | RESERVED | R/W | 0h | |
| 0 | ENABLE | R/W | 0h | This bit is used to enable subscriber 1 event.<br>0h = Subscriber 1 event is disabled<br>1h = Subscriber 1 event is enabled |



Copyright © 2026 Texas Instruments Incorporated

Chapter 10
# MATHACL

The Math accelerator (MATHACL) is a collection of hardware-accelerated 32-bit math functions to improve system computational throughput. Software running on the CPU invokes MATHACL by writing to the MATHACL registers.

## 10.1 Overview

The MATHACL offloads mathematical calculations performed by the CPU to improve efficiency. The following hardware functions are available in the MATHACL:

- Sine/cosine (SINCOS)
- Arc tangent (ATAN2)
- Square root (SQRT)
- Divide (DIV)
- Multiply with 32-bit result (MPY32)
- Square with 32-bit result (SQUARE32)
- Multiply with 64-bit result (MPY64)
- Square with 64-bit results (SQUARE64)
- Multiply-accumulate (MAC)
- Square-accumulate (SAC)

## 10.2 Data Format

The operands that are used in MATCHACL functions are 32-bit fixed point numbers. Operands can be expressed as unsigned 32-bit integers, signed 32-bit integers, unsigned 32-bit numbers, and signed 32-bit numbers.

**Note**

8 and 16-bit access is not allowed

Table 10-1 summarizes the data formats that are used in the MATHACL.

**Table 10-1. MATHACL data formats**

| Data format | Data type | Signed format | Data range |
|---|---|---|---|
| Unsigned 32-bit integers | Uint32_t | 2's compliment | 0 to $2^{32}$ - 1 |
| Signed 32-bit integers | int32_t | 2's compliment | $-2^{31}$ to $2^{31}$– 1 |
| Unsigned 32-bit numbers | UQm.n | Signed magnitude | 0 to $(2^{32})/2^{n}$ |
| Signed 32-bit numbers | SQm.n | Signed magnitude | $(-2^{31})/2^{n}$ to $(2^{31})/2^{n}$ |

### *10.2.1 Unsigned 32-bit integers*

An unsigned 32-bit integer is a 32-bit positive number ranging from 0 to $2^{32}$-1, and it is expressed as type uint32_t.

For instance, the number 123456 is equal to 0b 00000000 00000001 11100010 01000000, or 0x0001E240.



Copyright © 2026 Texas Instruments Incorporated

### 10.2.2 Signed 32-bit integers

A signed 32-bit integer is a 32-bit positive or negative number ranging from $-2^{31}$ to $2^{31} - 1$, and it is expressed as type int32_t. To convert a signed 32-bit integer from decimal:

- Convert the integer to an unsigned 32-bit integer
- Invert the number
- Add 1 to the least significant bit (LSB)

For instance, the number -123456 is equal to 0b 11111111 11111110 00011101 11000000, or 0xFFFE1DC0.

### 10.2.3 Unsigned 32-bit numbers

Unsigned 32-bit numbers include an integer component and fractional component using Q number format. It is expressed as UQ**m**.**n**, where:

- **m** is the integer component number of bits
- **n** is the fractional component number of bits

The m and n components vary depending on the operand type. For instance, an operand in UQ16.16 format has the signed number components as shown in Table 10-2

**Table 10-2. Unsigned number components**

| Description | Value | Number of Bits | Bits |
|---|---|---|---|
| Integer component | I | m = 16 | 31:16 |
| Fractional component | F | n = 16 | 15:0 |

To generate an unsigned number, such as 4.75 in UQ16.16:

1. Express the integer component (I = 4) as a number in m bits (0x0004)
2. Express the fractional component (F = 0.75) as a number in n bits (0xC000)
3. Express the resulting hexadecimal value as an unsigned value (0x0100C000)

### 10.2.4 Signed 32-bit numbers

Signed 32-bit numbers include a signed magnitude component, integer component and fractional component using Q number format. It is expressed as SQ**m**.**n**, where:

- S is the Sign Bit
- **m** is the integer component number of bits
- **n** is the fractional component number of bits

The m and n components vary depending on the operand type. For instance, an operand in SQ15.16 format has the signed number components as shown in Table 10-3.

**Table 10-3. Signed number components**

| Description | Value | Number of Bits | Bits |
|---|---|---|---|
| Sign bit | + or - | S = 1 | 31 |
| Integer component | I | m = 15 | 30:16 |
| Fractional component | F | n = 16 | 15:0 |

To generate a signed number, such as -1.5 in SQ15.16:

1. Express the integer component (I = 1) as a number in m bits (0x0001)
2. Express the fractional component (F = 0.5) as a number in n bits (0x8000)
3. Express the resulting hexadecimal value as an unsigned value (0x00018000)
4. If the number is signed, take the 2's compliment (0xFFFE7FFF)



Copyright © 2026 Texas Instruments Incorporated

## 10.3 Basic Operation

### Power Enable

To power on the MATHACL, write the KEY to the PWREN register and set the ENABLE bit to power on the MATHACL. To reset the MATHACL, write the KEY to the RSTCTL register, set the RESETSTKYCLR bit, and set the RESETASSERT bit.

### MATHACL Operation

All MATHACL operations follow a similar code sequence to call the functions, begin the compute operations, and read the results:

1. Write to the FUNC, QVAL, OPTYPE, SFACTOR, and NUMITER fields in the CTL register using a single write operation. See Section 10.4 for which fields are required by the function being called.
2. Trigger the function to start computing based on the number of operands required in the operation.
   a. In case of two operand functions, write to OP2 first, then write to OP1 to trigger the function to start computing.
   b. In case of single operand functions, write to OP1 to trigger the function to start computing.
3. To read the computed results, read the RES1 and RES2 registers to check when the new data has been returned. Optionally, the STATUS.BUSY flag can be polled until it is 0 to indicate the computation is complete.

**Note**

The MATHACL control registers should not be changed while the current operation is ongoing, or else the operation will result in undefined behavior.

**Note**

The BUSY bit exists so the application can check to see if the MATHACL operation is complete before requesting the result. If the application requests the result before the operation is complete, then the bus will stall until the operation is complete. The BUSY bit provides the application with a way to check the status of the MATHACL.

### MATHACL Timing

Timing for each MATHACL calculation depends on the number of fractional bits (n), number of iterations (NUMITER) in SINCOS, ATAN2, and SQRT functions, or software overheard. See each function example in Section 10.4 for the number of hardware cycles (if available).

## 10.4 Configuration Details with Examples

### *10.4.1 Sine and Cosine (SINCOS)*

Sine and Cosine (SINCOS) functions are computed using the CORDIC algorithm. The algorithm relies on expressing the angle as a sum of angles in per unit format with a configurable number of iterations to incrementally calculate the result. The sine and cosine of the given angle is computed simultaneously for the number of iterations specified in CTL.NUMITER field and stored in the RES1 and RES2 registers.

Table 10-4 shows the configuration registers for SINCOS.

**Table 10-4. Sine/Cosine Configuration registers**

| Register (Bit Field) | Value | Description |
|---|---|---|
| OP1 | User value | Angle in degrees per unit (SQ0.31 format). The angle in per unit is calculated by n = angle/180, where n is the angle in per unit between [-1,1) and angle is between [-180,180) degrees. |



Copyright © 2026 Texas Instruments Incorporated

**Table 10-4. Sine/Cosine Configuration registers (continued)**

| Register (Bit Field) | Value | Description |
|---|---|---|
| CTL.FUNC | 1h | SINCOS |
| CTL.NUMITER | User value | Number of iterations |
| RES1 | 32-bit result | Cosine of the angle (SQ0.31 format) |
| RES2 | 32-bit result | Sine of the angle (SQ0.31 format) |

A higher number of iterations (NUMITER) calculates more accurate results but will take longer to compute.

**Status, Errors, and Overflow**

There are no status, error, or overflow bits for SINCOS.

**Configuration**

To perform SINCOS:

1. In the CTL register:
   a. Set FUNC to 1h.
   b. Set NUMITER to the number of iterations for the calculation.
2. Set the angle per unit (SQ0.31 format) in OP1 to begin the computation.
3. Read the cosine of the angle in RES1 (SQ0.31 format).
4. Read the sine of the angle in RES2 (SQ0.31 format).

### *10.4.2 Arc Tangent (ATAN2)*

The arctangent function (ATAN2) computes the arctangent of (y/x), where y and x are the normalized coordinates. The ATAN2 algorithm relies on expressing the resulting angle as a sum of constituent angles computed by incrementally accumulating the constituent angles until the y-coordinate value is 0. The number of iterations to calculate the arctangent is specified using the CTL.NUMITER field, and the final result for the angle in per unit is stored in RES1 as a signed number.

Before performing the ATAN2 function, the x- and y- coordinates should be normalized to a unit vector, which can be done with various types of normalization algorithms. One normalization algorithm is shown in Figure 10-1.



Copyright © 2026 Texas Instruments Incorporated

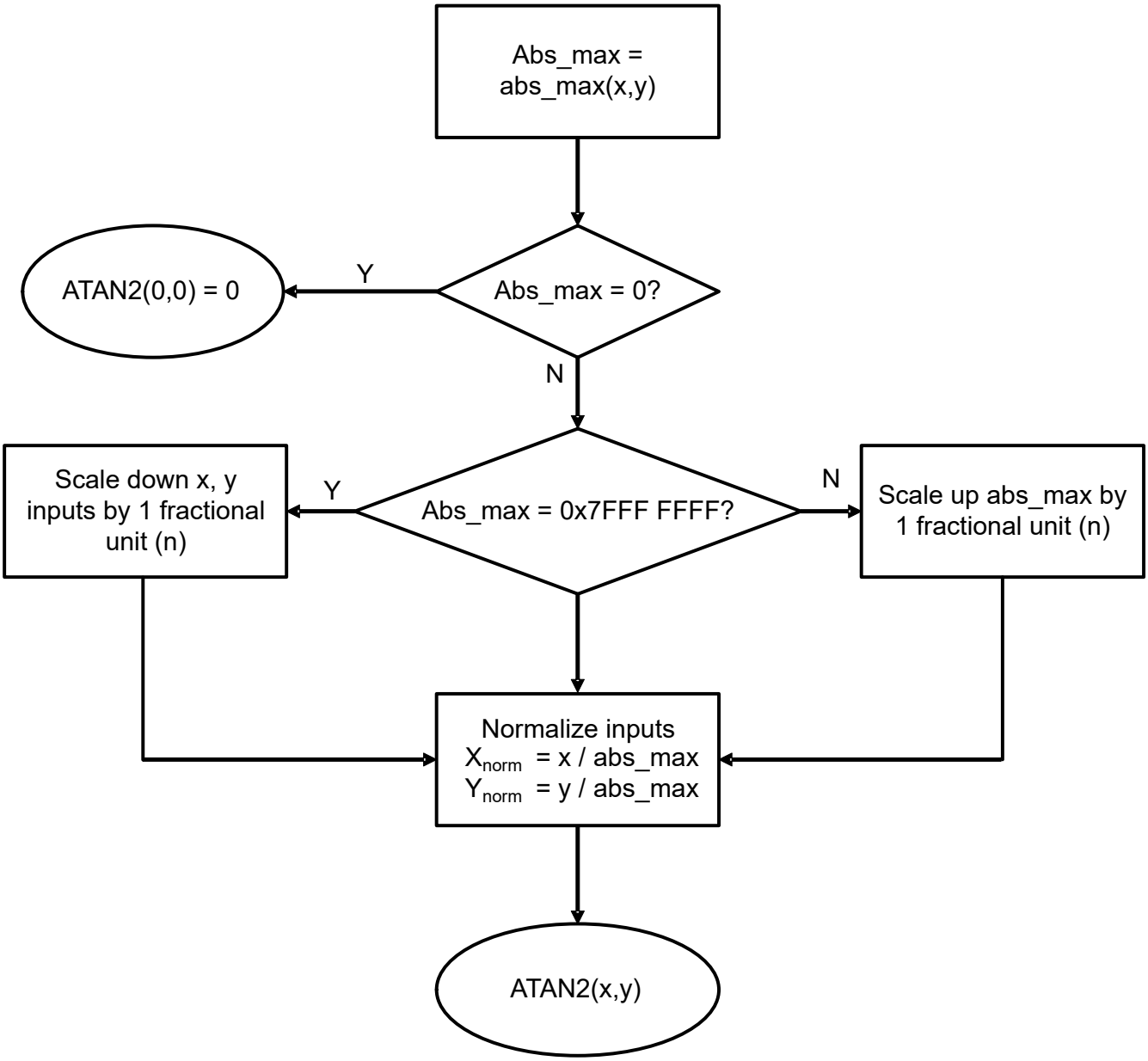



**Figure 10-1. Normalization algorithm for ATAN2 inputs**

Table 10-5 shows the configuration registers for the ATAN2 function.

**Table 10-5. ATAN2 configuration registers**

| Register (Bit Field) | Value | Description |
|---|---|---|
| OP1 | User value | X coordinate (SQ0.31 format) |
| OP2 | User value | Y coordinate (SQ0.31 format) |
| CTL.FUNC | 2h | ATAN2 |
| CTL.NUMITER | User value | Number of iterations |



Copyright © 2026 Texas Instruments Incorporated

**Table 10-5. ATAN2 configuration registers (continued)**

| Register (Bit Field) | Value | Description |
|---|---|---|
| RES1 | 32-bit value | Angle in degrees per unit (SQ0.31 format) . The angle in per unit is calculated by n = angle/180, where n is the angle in per unit between [-1,1) and angle is between [-180,180) degrees. |

A higher number of iterations (NUMITER) calculates more accurate results but will take longer to compute.

**Status, Errors, and Overflow**

There are no status, error, or overflow bits for ATAN2.

**Configuration**

To perform ATAN2:

1. In the CTL register:
   a. Set FUNC to 2h.
   b. Set NUMITER to the number of iterations for the calculation.
2. Set the normalized y-coordinate (SQ0.31 format) in OP2.
3. Set the normalized x-coordinate (SQ0.31 format) in OP1 to begin the computation.
4. Read the normalized arctangent result in RES1 (SQ0.31 format).

### *10.4.3 Square Root (SQRT)*

The square root function (SQRT) computes the square root of a number (radicand). The square root is computed incrementally by the number of iterations in the CTL.NUMITER field. A higher number of iterations calculates more accurate results but takes longer to compute.

The radicand must be in UQm.n data type, and it is broken down into two components: an unscaled number (UN) and a scaled number (SN). The unscaled number is the floor of the number, and the scaled number is between 1.0 and 2.0 in UQ2.30 format. The relationship between the scaled number (SN) and unscaled number (UN) is the scale factor (SFACTOR), which is shown in Equation 11.

$$Scaled\ number\ \left(SN\right) = \frac{Unscaled\ number\ (UN)}{2^{SFACTOR}} \tag{11}$$

To calculate the SN and SFACTOR from a radicand in UQm.n format, the algorithm below can be used.

```
//Compute scale factor and scaled number from radicand

UN = floor(radicand);          //unsigned number
SFACTOR = 0;                   //scale factor
count = 2;                     //scaled number must be between 1 and 2
do {
   SFACTOR++;                  //incease SFACTOR
   count <<= 1;                //multiply by 2
} while (count < UN)           //repeat until UN is less than count

SN = radicand / (count >> 1); //scale number = radicand / 2^SN
```

For example, if the 10.375 is the radicand, the unscaled number is 10, SFACTOR = 3 (10.375 / $2^3$ = 1.297), and 1.297 is the scaled number between 1.0 and 2.0.

Table 10-6 shows the configuration registers for the SQRT function.



Copyright © 2026 Texas Instruments Incorporated

**Table 10-6. SQRT configuration registers**

| Register (Bit Field) | Value | Description |
|---|---|---|
| OP1 | User value | Scaled number (SN) in UQ2.30 format. The scaled number is scaled from the unscaled number, and the scaled number has to be between 1.0 and 2.0. |
| CTL.FUNC | 5h | SQRT |
| CTL.SFACTOR | User value | Scaling factor n. The scaled number (SN) = (Unscaled number)/$2^n$. This is a signed integer and takes a positive value or negative value in two's compliment. |
| CTL.NUMITER | User value | Number of iterations |
| RES1 | 32-bit result | Square root of unscaled number in UQ16.16 format. |

A higher number of iterations (NUMITER) calculates more accurate results but takes longer to compute.

**Status, Errors, and Overflow**

There are no status, error, or overflow bits for SQRT.

**Configuration**

To perform a SQRT:

1. In the CTL register:
   a. Set FUNC to 5h.
   b. Set NUMITER to the number of iterations for the calculation.
   c. Set SFACTOR to the scaling factor n.
2. Set the scaled number (UQ2.30 format) in OP1 to begin the computation.
3. Read the square root result in RES1 (UQ16.16 format).

### *10.4.4 Division (DIV)*

The divide function (DIV) computes with a known dividend and divisor.

$$Dividend = Divisor*Quotient + Remainder \quad (12)$$

$$mod(Remainder) < mod(Divisor) \quad (13)$$

Table 10-7 shows the data types that the DIV function can perform.

**Table 10-7. DIV Data Types Supported**

| Data Type (Dividend and Divisor) | Data Type (Quotient) | Data Type (Remainder) |
|---|---|---|
| Uint32_t | Uint32_t | Uint32_t |
| Int32_t | Int32_t | Int32_t |
| SQm.n | SQm.n | N/A |
| UQm.n | UQm.n | N/A |

Table 10-8 shows the configuration registers for the DIV function.

**Table 10-8. DIV Configuration Registers**

| Register (Bit Field) | Value | Description |
|---|---|---|
| OP1 | User value | Dividend |
| OP2 | User value | Divisor |
| CTL.FUNC | 4h | DIV |



Copyright © 2026 Texas Instruments Incorporated

**Table 10-8. DIV Configuration Registers (continued)**

| Register (Bit Field) | Value | Description |
|---|---|---|
| CTL.QVAL | User value | Number of fractional bits (unsigned or signed number data type only) |
| CTL.OPTYPE | 0 = unsigned operands<br>1 = signed operands | Operand sign |
| RES1 | 32-bit result | Quotient (same data type as dividend and divisor) |
| RES2 | 32-bit result | Remainder (same data type as dividend and divisor) |

**Note**

OP1 and OP2 must be the same data type or use incorrect results will occur.

Signed integer (int32_t) calculations requires 4 cycles to compute results in RES1 and RES2 from an OP2 write. Examples of DIV for int32_t are in Table 10-9.

**Table 10-9. Division Examples for int32_t Data Type**

| Dividend | Divisor | Quotient | Remainder |
|---|---|---|---|
| 15 (0x0000000F) | 2 (0x00000002) | 7 (0x00000007) | 1 (0x00000001) |
| -15 (0xFFFFFFF1) | 2 (0x00000002) | -7 (0xFFFFFFF9) | -1 (0xFFFFFFFF) |
| 15 (0x0000000F) | -2 (0xFFFFFFFE) | -7 (0xFFFFFFF9) | 1 (0x00000001) |
| -15 (0xFFFFFFF1) | -2 (0xFFFFFFFE) | 7 (0x00000007) | -1 (0xFFFFFFFF) |

Signed number (SQm.n) calculations requires n cycles to compute results in RES1 and RES2 from an OP2 write, where n is the number of fractional bits. Examples of DIV for SQ15.16 are in Table 10-10.

**Table 10-10. Division Examples for SQ15.16 Data Type**

| Dividend | Divisor | Quotient | Remainder |
|---|---|---|---|
| 7.5 (0x00078000) | 4.5 (0x00048000) | 1.666657 (0x0001AAAA) | N/A |
| -7.5 (0xFFF88000) | 4.5 (0x00048000) | -1.666657 (0xFFFE5556) | N/A |
| 7.5 (0x00078000) | -4.5 (0xFFFB8000) | -1.666657 (0xFFFE5556) | N/A |
| -7.5 (0xFFF88000) | -4.5 (0xFFFB8000) | 1.666657 (0x0001AAAA) | N/A |

### Status, Errors, and Overflow

If the divisor is 0, then the STATUS.ERR flag is set.

If the result of the operation is more than 32 bits, STATUS.OVF is set to indicate that an overflow has occurred. It will remain set until cleared by writing a 1 to CLR.CLR_OVF.

For signed operation, if result saturation is enabled (CTL.SATEN = 1) in the case of an overflow, the result will be saturated to the maximum positive result (0x7FFFFFFF) and the minimum negative result (0x80000000).

For unsigned operation, if result saturation is enabled (CTL.SATEN = 1) in the case of an overflow, the result will be saturated to the maximum positive result (0xFFFFFFFF).

### Configuration

To perform a DIV:

1. In the CTL register:
   a. Set FUNC to 4h.
   b. Set OPTYPE to 0 if the data type is unsigned, or 1 if the data type is signed.
   c. Set QVAL to the number of fractional bits (n) for a signed (SQm.n) or unsigned number (UQm.n)



Copyright © 2026 Texas Instruments Incorporated

2. Set the divisor in OP2.
3. Set the dividend in OP1 to begin the computation.
4. Read the quotient result in RES1 and remainder result in RES2.

### *10.4.5 Multiplication*

Multiplication functions gives the result of multiplying two numbers. There are variations of the multiply function, including multiplying and squaring with 32-bit and 64-bit results.

#### 10.4.5.1 Multiply32 (MPY32)

The Multiply32 (MPY32) function returns the results of multiplying two 32-bit numbers with the 32-bit result in RES1.

The MPY32 function can perform the data types shown in Table 10-11.

**Table 10-11. MPY32 Data Types Supported**

| Data type (Multiplicand and Multiplier) | Data type (Product) | Condition |
|---|---|---|
| Uint32_t | Uint32_t | Product is < $2^{32}$ - 1 |
| Int32_t | Int32_t | Product is > $-2^{31}$ and < $2^{31}$ - 1 |
| SQm.n | SQm.n | Product is not outside the range representable in the chosen data format |
| UQm.n | UQm.n | |

Table 10-12 shows the operands and descriptions for the MPY32 function.

**Table 10-12. MPY32 Configuration Registers**

| Register (Bit Field) | Value | Description |
|---|---|---|
| OP1 | User value | Multiplicand |
| OP2 | User value | Multiplier |
| CTL.FUNC | 6h | MPY32 |
| CTL.QVAL | User value | Number of fractional bits (unsigned or signed number data type only) |
| CTL.OPTYPE | 0h = unsigned operands<br>1h = signed operands | Operand sign |
| RES1 | 32-bit product | Product |

**Note**

OP1 and OP2 must be the same data type or use incorrect results will occur.

Signed integer (int32_t) calculations requires 1 cycles to compute results in RES1 from an OP2 write. Examples of MPY32 for int32_t are in Table 10-13.

**Table 10-13. MPY32 Examples Using Signed Integers (int32_t)**

| Multiplicand | Multiplier | Product |
|---|---|---|
| 15 (0x0000000F) | 2 (0x00000002) | 30 (0x0000001E) |
| -15 (0xFFFFFFF1) | 2 (0x00000002) | -30 (0xFFFFFFE2) |
| 15 (0x0000000F) | -2 (0xFFFFFFFE) | -30 (0xFFFFFFE2) |
| -15 (0xFFFFFFF1) | -2 (0xFFFFFFFE) | 30 (0x0000001E) |

Signed number (SQm.n) calculations requires n cycles to compute results in RES1 from an OP2 write, where n is the number of fractional bits. Examples of MPY32 for SQ15.16 are in Table 10-14.



Copyright © 2026 Texas Instruments Incorporated

**Table 10-14. MPY32 Examples Using SQ15.16 Signed Numbers**

| Multiplicand | Multiplier | Product |
|---|---|---|
| 1.5 (0x00018000) | 2.5 (0x00028000) | 3.75 (0x0003C000) |
| -1.5 (0xFFFE8000) | 2.5 (0x00028000) | -3.75 (0xFFFCC000) |
| 1.5 (0x00018000) | -2.5 (0xFFFD8000) | -3.75 (0xFFFCC000) |
| -1.5 (0xFFFE8000) | -2.5 (0xFFFD8000) | 3.75 (0x0003C000) |

### Status, Errors, and Overflow

If the result of the operation is more than 32 bits, STAT.OVF is set to indicate that an overflow has occurred. It will remain set until cleared by writing setting CLR_OVF = 1 in the CLR register.

For signed operation, if result saturation is enabled (CTL.SATEN = 1), in the case of an overflow, the STAT.OVF bit is set and the result saturates to the maximum positive result (0x7FFFFFFF).

For unsigned operation, if result saturation is enabled (CTL.SATEN = 1) in the case of an overflow, the result will be saturated to the maximum positive result (0xFFFFFFFF).

### Configuration

To perform an MPY32:

1. In the CTL register:
   a. Set FUNC to 6h.
   b. Set OPTYPE to 0 if the data type is unsigned, or 1 if the data type is signed.
   c. Set QVAL to the number of fractional bits (n) for a signed (SQm.n) or unsigned number (UQm.n)
2. Set the multiplicand in OP2.
3. Set the multiplier in OP1 to begin the computation.
4. Read the product result in RES1.

#### 10.4.5.2 Square32 (SQUARE32)

The Square32 (SQUARE32) function is a variation of the MPY32 function and performs a square operation on operand OP2 with the 32-bit result in RES1.

The SQUARE32 function can perform the data types shown inTable 10-15.

**Table 10-15. SQUARE32 Data Types Supported**

| Data type (Base) | Data type (Result) | Condition |
|---|---|---|
| Uint32_t | Uint32_t | Result is < $2^{32}$ - 1 |
| Int32_t | Int32_t | Result is > $-2^{31}$ and < $2^{31}$ - 1 |
| SQm.n | SQm.n | Result is not outside the range representable in the chosen data format |
| UQm.n | UQm.n | |

Table 10-12 shows the operands and descriptions for the SQUARE32 function.

**Table 10-16. SQUARE32 Configuration Registers**

| Register (Bit Field) | Value | Description |
|---|---|---|
| OP1 | User value | Base |
| CTL.FUNC | 7h | MPY32 |
| CTL.QVAL | User value | Number of fractional bits (unsigned or signed number data type only) |
| CTL.OPTYPE | 0h = unsigned operands<br>1h = signed operands | Operand sign |
| RES1 | 32-bit square result | Square result |



Copyright © 2026 Texas Instruments Incorporated

Signed integer (int32_t) calculations requires 1 cycle to compute results in RES1 from an OP2 write. Examples of SQUARE32 for int32_t data type are in Table 10-17.

**Table 10-17. SQUARE32 Examples Using int32_t**

| Multiplicand/Multiplier | Square |
|---|---|
| 4 (0x00000004) | 16 (0x00000010) |
| -4 (0xFFFFFFFB) | 16 (0x00000010) |

Signed number (SQm.n) calculations requires n cycles to compute results in RES1 from an OP2 write, where n is the number of fractional bits. Examples of division for SQ15.16 are in Table 10-18.

**Table 10-18. SQUARE32 Examples Using SQ15.16 Signed Numbers**

| Multiplicand/Multiplier | Square |
|---|---|
| 1.5 (0x00018000) | 2.25 (0x00024000) |
| -1.5 (0xFFFE8000) | 2.25 (0x00024000) |

### Status, Errors, and Overflow

If the result of the operation is more than 32 bits, STAT.OVF is set to indicate that an overflow has occurred. It will remain set until cleared by writing setting CLR_OVF = 1 in the CLR register.

For signed operation, if result saturation is enabled (CTL.SATEN = 1), in the case of an overflow, the STAT.OVF bit is set and the result saturates to the maximum positive result (0x7FFFFFFF).

For unsigned operation, if result saturation is enabled (CTL.SATEN = 1) in the case of an overflow, the result will be saturated to the maximum positive result (0xFFFFFFFF).

### Configuration

To perform a SQUARE32:

1. In the CTL register:
   a. Set FUNC to 6h.
   b. Set OPTYPE to 0 if the data type is unsigned, or 1 if the data type is signed.
   c. Set QVAL to the number of fractional bits (n) for a signed (SQm.n) or unsigned number (UQm.n)
2. Set the base in OP1 to begin the computation.
3. Read the product result in RES1.

#### 10.4.5.3 Multiply64 (MPY64)

The Multiply64 function (MPY64) returns the results of multiplying two 32-bit numbers with the 64-bit result in RES1 and RES2.

The MPY64 function can perform the types of multiply shown in Table 10-11.

**Table 10-19. MPY64 Data Types Supported**

| Data type (Multiplicand and Multiplier) | Data type (Product) | Condition |
|---|---|---|
| Uint32_t | Uint64_t | Product is < $2^{64}$ - 1 |
| Int32_t | Int64_t | Product is > $-2^{63}$ and < $2^{63}$ - 1 |

Table 10-12 shows the operands and descriptions for the multiply function.

**Table 10-20. MPY64 Configuration Registers**

| Register (Bit Field) | Value | Description |
|---|---|---|
| OP1 | User value | Multiplicand |
| OP2 | User value | Multiplier |



Copyright © 2026 Texas Instruments Incorporated

**Table 10-20. MPY64 Configuration Registers (continued)**

| Register (Bit Field) | Value | Description |
|---|---|---|
| CTL.FUNC | 8h | MPY64 |
| CTL.OPTYPE | 0h = unsigned operands<br>1h = signed operands | Operand sign |
| RES1 | Lower 32 bits of product | Product (64-bit) |
| RES2 | Upper 32 bits of product | |

**Note**

OP1 and OP2 must be the same data type or use incorrect results will occur.

**Status, Errors, and Overflow**

There are no status, error, or overflow bits for MPY64.

**Configuration**

To perform an MPY64:

1. In the CTL register:
   a. Set FUNC to 8h.
   b. Set OPTYPE to 0 if the data type is unsigned, or 1 if the data type is signed.
2. Set the multiplicand in OP2.
3. Set the multiplier in OP1 to begin the computation.
4. Read the product result in RES1 and RES2.

#### 10.4.5.4 Square64 (SQUARE64)

The Square64 function (SQUARE64) is a variation of the MPY32 function and performs a square operation on operand OP2 with the 64-bit result in RES1 and RES2.

The SQUARE64 function can perform the types of multiply shown in Table 10-11.

**Table 10-21. SQUARE64 data types supported**

| Data type (Base) | Data type (Square Result) | Condition |
|---|---|---|
| Uint32_t | Uint64_t | Result is < $2^{64}$ - 1 |
| Int32_t | Int64_t | Result is > $-2^{63}$ and < $2^{63}$ - 1 |

Table 10-12 shows the operands and descriptions for the SQUARE64 function.

**Table 10-22. SQUARE64 configuration registers**

| Register (Bit Field) | Value | Description |
|---|---|---|
| OP1 | User value | Base |
| CTL.FUNC | 9h | MPY64 |
| CTL.OPTYPE | 0h = unsigned operands<br>1h = signed operands | Operand sign |
| RES1 | Lower 32 bits of product | Product (64-bit) |
| RES2 | Upper 32 bits of product | |

**Status, Errors, and Overflow**

There are no status, error, or overflow bits for SQUARE64.

**Configuration**



Copyright © 2026 Texas Instruments Incorporated

To perform an SQUARE64:

1. In the CTL register:
   a. Set FUNC to 8h.
   b. Set OPTYPE to 0 if the data type is unsigned, or 1 if the data type is signed.
2. Set the base in OP1 to begin the computation.
3. Read the product result in RES1 and RES2.

### *10.4.6 Multiply-Accumulate (MAC)*

Multiply-accumulate (MAC) functions accumulates the results of the product of OP1 and OP2 inside result registers RES1 and RES2 over multiple iterations until the MATHACL is reprogrammed for another function. The MATHACL supports only a 32-bit multiplier and multiplicand operand size, and a 64-bit accumulated result size.

Figure 10-2 shows the operation of the MATHACL in MAC mode.

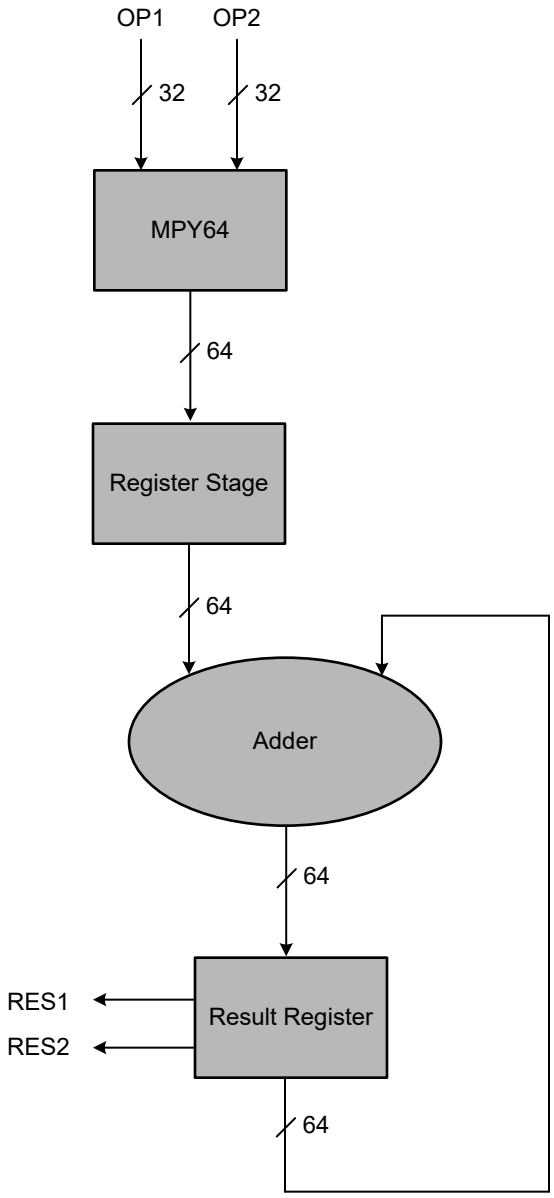



**Figure 10-2. MAC Operation**



Copyright © 2026 Texas Instruments Incorporated

Internal MAC operation performs the following steps:

- OP1 and OP2 are multiplied together, and the result is a 64-bit product
  - If using SQm.n or UQm.n data type, the product has the same number of fractional bits (n) as the operands
- The product result is stored into a Register Stage for correct timing
- The registered value is fed into a 64-bit Adder with the current Result Register value as the other input (Result Register += Register Stage)
- The result of the add operation is stored into RES1 and RES2

**Note**

Write 0 into RES1 and RES2 before starting a MAC operation to get the correct result.

The MAC function can perform the following data types of multiply-accumulate shown in Table 10-23.

**Table 10-23. MAC Data Types Supported**

| Data Type (Multiplicand and Multiplier) | Data Type (Product) | Condition |
|---|---|---|
| Uint32_t | Uint64_t | MAC result is < $2^{64}$ - 1 |
| Int32_t | Int64_t | MAC result is > $-2^{63}$ and < $2^{63}$ - 1 |
| SQm.n | SQ(32+m).n | MAC result is not outside the range representable in the chosen data format |
| UQm.n | UQ(32+m).n | |

Table 10-24 shows the configuration registers for the MAC function.

**Table 10-24. MAC Configuration Registers**

| Register (Bit Field) | Value | Description |
|---|---|---|
| OP1 | User value | Multiplicand |
| OP2 | User value | Multiplier |
| CTL.FUNC | Ah | MAC |
| CTL.QVAL | User value | Number of fractional bits (unsigned or signed number data type only) |
| CTL.OPTYPE | 0 = unsigned operands<br>1 = signed operands | Operand sign |
| RES1 | Lower 32 bits of result | Multiply-accumulate result (64-bit) |
| RES2 | Upper 32 bits of result | |

**Note**

OP1 and OP2 must be the same data type or use incorrect results will occur.

The final result of the MAC operation can be read after 2 cycles after writing OP2.

### Status, Errors, and Overflow

If the result of the operation is more than 64 bits, STAT.OVF is set to indicate that an overflow has occurred. It will remain set until cleared by writing a 1 to CLR.CLR_OVF.

For signed operation, if result saturation is enabled (CTL.SATEN = 1) in the case of an overflow, the result will be saturated to the maximum positive result (0x7FFF_FFFF_FFFF_FFFF) and the minimum negative result (0x8000_0000_0000_0000).

For unsigned operation, if result saturation is enabled (CTL.SATEN = 1) in the case of an overflow, the result will be saturated to the maximum positive result (0xFFFF_FFFF_FFFF_FFFF).

### Configuration



Copyright © 2026 Texas Instruments Incorporated

To configure the MAC for operation:

1. In the CTL register:
   a. Set FUNC to Ah.
   b. Set OPTYPE to 0 if the data type is unsigned, or 1 if the data type is signed.
   c. Set QVAL to the number of fractional bits (n) for a signed (SQm.n) or unsigned number (UQm.n)
2. Clear the result registers by setting RES1 and RES2 to 0.
3. Set the multiplicand in OP2.
4. Set the multiplier in OP1 to begin the computation.
5. Repeat step 3-4 to continue multiplying and accumulating data in RES1 and RES2.
6. Read the 64-bit product result in RES1 and RES2.

### *10.4.7 Square Accumulate (SAC)*

Square-accumulate (SAC) functions accumulates the results of the square of operand OP2 inside result registers RES1 and RES2 over multiple iterations until the MATHACL is reprogrammed for another function. It is a variation of the MAC function and operation details remain the same.

**Note**

Write 0 into RES1 and RES2 before starting a SAC operation to get the correct result.

The SAC function can perform the following data types of square-accumulate shown in Table 10-25.

**Table 10-25. SAC Data Types Supported**

| Data Type (Base) | Data Type (Result) | Condition |
|---|---|---|
| Uint32_t | Uint64_t | SAC result should be < $2^{64}$ - 1 |
| Int32_t | Int64_t | SAC result should be > $-2^{63}$ and < $2^{63}$ - 1 |
| SQm.n | SQ(32+m).n | SAC result should not be outside the range representable in the chosen data format |
| UQm.n | UQ(32+m).n | |

Table 10-26 shows the configuration registers for the SAC function.

**Table 10-26. SAC configuration registers**

| Register (Bit Field) | Value | Description |
|---|---|---|
| OP1 | User value | Base |
| CTL.FUNC | Bh | SAC |
| CTL.QVAL | User value | Number of fractional bits (unsigned or signed number data type only) |
| CTL.OPTYPE | 0 = unsigned operands<br>1 = signed operands | Operand sign |
| RES1 | Lower 32 bits of result | Square-accumulate result (64-bit) |
| RES2 | Upper 32 bits of result | |

The final result of the SAC operation can be read after 2 cycles after writing OP2.

**Status, Errors, and Overflow**

If the result of the operation is more than 64 bits, STAT.OVF is set to indicate that an overflow has occurred. It will remain set until cleared by writing a 1 to CLR.CLR_OVF.



Copyright © 2026 Texas Instruments Incorporated

For signed operation, if result saturation is enabled (CTL.SATEN = 1) in the case of an overflow, the result will be saturated to the maximum positive result (0x7FFF_FFFF_FFFF_FFFF) and the minimum negative result (0x8000_0000_0000_0000).

For unsigned operation, if result saturation is enabled (CTL.SATEN = 1) in the case of an overflow, the result will be saturated to the maximum positive result (0xFFFF_FFFF_FFFF_FFFF).

**Configuration**

To configure the SAC for operation:

1. In the CTL register:
   a. Set FUNC to Bh.
   b. Set OPTYPE to 0 if the data type is unsigned, or 1 if the data type is signed.
   c. Set QVAL to the number of fractional bits (n) for a signed (SQm.n) or unsigned number (UQm.n)
2. Clear the result registers by setting RES1 and RES2 to 0.
3. Set the base in OP1 to begin the computation.
4. Repeat step 3 to continue squaring and accumulating data in RES1 and RES2.
5. Read the 64-bit square result in RES1 and RES2.



Copyright © 2026 Texas Instruments Incorporated

## 10.5 MATHACL Registers

Table 10-27 lists the memory-mapped registers for the MATHACL registers. All register offset addresses not listed in Table 10-27 should be considered as reserved locations and the register contents should not be modified.

**Table 10-27. MATHACL Registers**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 800h | PWREN | Power enable | | Go |
| 804h | RSTCTL | Reset Control | | Go |
| 814h | STAT | Status Register | | Go |
| 1100h | CTL | Control Register | | Go |
| 1118h | OP2 | Operand 2 register. | | Go |
| 111Ch | OP1 | Operand 1 register. | | Go |
| 1120h | RES1 | Result 1 register. | | Go |
| 1124h | RES2 | Result 2 register. | | Go |
| 1130h | STATUS | Status Register | | Go |
| 1140h | STATUSCLR | Status flag clear register | | Go |

Complex bit access types are encoded to fit into small table cells. Table 10-28 shows the codes that are used for access types in this section.

**Table 10-28. MATHACL Access Type Codes**

| Access Type | Code | Description |
|---|---|---|
| Read Type | | |
| R | R | Read |
| RH | R<br>H | Read<br>Set or cleared by hardware |
| Write Type | | |
| W | W | Write |
| WK | W<br>K | Write<br>Write protected by a key |
| Reset or Default Value | | |
| *-n* | | Value after reset or the default value |



Copyright © 2026 Texas Instruments Incorporated

### 10.5.1 PWREN (Offset = 800h) [Reset = 00000000h]

PWREN is shown in Figure 10-3 and described in Table 10-29.

Return to the Summary Table.

Register to control the power state

**Figure 10-3. PWREN**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | ENABLE |
| R-0h | | | | | | | R/WK-0h |

**Table 10-29. PWREN Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | KEY to allow Power State Change<br>26h = KEY to allow write access to this register |
| 23-1 | RESERVED | R | 0h | |
| 0 | ENABLE | R/WK | 0h | Enable the power<br>[EXT_GPRCM.GPRCM.PWREN.KEY] must be set to 26h to write to this bit.<br>0h = Disable Power<br>1h = Enable Power |



Copyright © 2026 Texas Instruments Incorporated

### 10.5.2 RSTCTL (Offset = 804h) [Reset = 00000000h]

RSTCTL is shown in Figure 10-4 and described in Table 10-30.

Return to the Summary Table.

Register to control reset assertion and de-assertion

**Figure 10-4. RSTCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | RESETSTKYCLR | RESETASSERT |
| R-0h | | | | | | WK-0h | WK-0h |

**Table 10-30. RSTCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | Unlock key<br>B1h = KEY to allow write access to this register |
| 23-2 | RESERVED | R | 0h | |
| 1 | RESETSTKYCLR | WK | 0h | Clear the RESETSTKY bit in the STAT register<br>[EXT_GPRCM.GPRCM.RSTCTL.KEY] must be set to B1h to write to this bit.<br>0h = Writing 0 has no effect<br>1h = Clear reset sticky bit |
| 0 | RESETASSERT | WK | 0h | Assert reset to the peripheral<br>[EXT_GPRCM.GPRCM.RSTCTL.KEY] must be set to B1h to write to this bit.<br>0h = Writing 0 has no effect<br>1h = Assert reset |



Copyright © 2026 Texas Instruments Incorporated

### 10.5.3 STAT (Offset = 814h) [Reset = 00000000h]

STAT is shown in Figure 10-5 and described in Table 10-31.

Return to the Summary Table.

peripheral enable and reset status

**Figure 10-5. STAT**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | RESETSTKY |
| R-0h | | | | | | | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |

**Table 10-31. STAT Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-17 | RESERVED | R | 0h | |
| 16 | RESETSTKY | R | 0h | This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register<br>0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register<br>1h = The peripheral was reset since the last bit clear |
| 15-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 10.5.4 CTL (Offset = 1100h) [Reset = 00000000h]

CTL is shown in Figure 10-6 and described in Table 10-32.

Return to the Summary Table.

Control Register

**Figure 10-6. CTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | NUMITER | | | | |
| R-0h | | | R/W-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | SATEN | SFACTOR | | | | | |
| R-0h | R/W-0h | R/W-0h | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | QVAL | | | | |
| R-0h | | | R/W-0h | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | OPTYPE | FUNC | | | | |
| R-0h | | R/W-0h | R/W-0h | | | | |

**Table 10-32. CTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | RESERVED | R | 0h | |
| 28-24 | NUMITER | R/W | 0h | Number of iterations, applicable if the function does the computations iteratively, for example sine/cosine/atan2/sqrt.<br>Note: A value of 0 is interpreted as 31. |
| 23 | RESERVED | R | 0h | |
| 22 | SATEN | R/W | 0h | Saturation enable<br>This bit is shared among DIV, SQUARE32, MPY32, MAC and SAC functions.<br>When enabled, it will make the result to saturate to maximum value in case of an overflow event<br>When disabled, the result will overflow to an unknown value.<br>0h = Saturation is disabled<br>1h = Saturation is enabled |
| 21-16 | SFACTOR | R/W | 0h | Scaling factor. In case of SQRT function, the input operand needs to be in a range. If not it has to be scaled to $2^{+/-n}$. This field should be written with the value 'n'. |
| 15-13 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

**Table 10-32. CTL Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 12-8 | QVAL | R/W | 0h | Indicates the fractional bits in the operands, ranges from 0 to 31. Applicable to DIV function.<br>0h = Q0 operands<br>1h = Q1 operands<br>2h = Q2 operands<br>3h = Q3 operands<br>4h = Q4 operands<br>5h = Q5 operands<br>6h = Q6 operands<br>7h = Q7 operands<br>8h = Q8 operands<br>9h = Q9 operands<br>Ah = Q10 operands<br>Bh = Q11 operands<br>Ch = Q12 operands<br>Dh = Q13 operands<br>Eh = Q14 operands<br>Fh = Q15 operands<br>10h = Q16 operands<br>11h = Q17 operands<br>12h = Q18 operands<br>13h = Q19 operands<br>14h = Q20 operands<br>15h = Q21 operands<br>16h = Q22 operands<br>17h = Q23 operands<br>18h = Q24 operands<br>19h = Q25 operands<br>1Ah = Q26 operands<br>1Bh = Q27 operands<br>1Ch = Q28 operands<br>1Dh = Q29 operands<br>1Eh = Q30 operands<br>1Fh = Q31 operands |
| 7-6 | RESERVED | R | 0h | |
| 5 | OPTYPE | R/W | 0h | Operand type, could signed or unsigned. applicable to DIV function.<br>0h = Unsigned operands<br>1h = Signed operands. |
| 4-0 | FUNC | R/W | 0h | ULP_ADCHP Enable Conversions.<br>0h = No operation<br>1h = Sine and Cosine operation<br>2h = Arc tangent with x and y values as operands.<br>4h = Divide, the operands are numerator, denominator, and the divide type. Result is the quotient and reminder.<br>5h = Do square root. Operand is the number whoose square root needs to be computed. The number if outside the range needs to be scaled up down by 2 power 2n to bring it with in the range.<br>6h = 32-bit Multiply Result<br>7h = 32-bit square result<br>8h = 64-bit multiply result<br>9h = 64-bit multiply result<br>Ah = Multiply and accumulate operation<br>Bh = Square and accumulate operation |



Copyright © 2026 Texas Instruments Incorporated

### 10.5.5 OP2 (Offset = 1118h) [Reset = 00000000h]

OP2 is shown in Figure 10-7 and described in Table 10-33.

Return to the Summary Table.

Operand 2 register.

**Figure 10-7. OP2**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 10-33. OP2 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | R/W | 0h | Operand 2 Register |



Copyright © 2026 Texas Instruments Incorporated

### 10.5.6 OP1 (Offset = 111Ch) [Reset = 00000000h]

OP1 is shown in Figure 10-8 and described in Table 10-34.

Return to the Summary Table.

Operand 1 register

**Figure 10-8. OP1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 10-34. OP1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | R/W | 0h | Operand 1 register. |



Copyright © 2026 Texas Instruments Incorporated

### 10.5.7 RES1 (Offset = 1120h) [Reset = 00000000h]

RES1 is shown in Figure 10-9 and described in Table 10-35.

Return to the Summary Table.

Result 1 register.

**Figure 10-9. RES1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| RH/W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 10-35. RES1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | RH/W | 0h | Result 1 register |



Copyright © 2026 Texas Instruments Incorporated

### 10.5.8 RES2 (Offset = 1124h) [Reset = 00000000h]

RES2 is shown in Figure 10-10 and described in Table 10-36.

Return to the Summary Table.

Result 2 register.

**Figure 10-10. RES2**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| RH/W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 10-36. RES2 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | RH/W | 0h | Result 2 register |



Copyright © 2026 Texas Instruments Incorporated

### 10.5.9 STATUS (Offset = 1130h) [Reset = 00000000h]

STATUS is shown in Figure 10-11 and described in Table 10-37.

Return to the Summary Table.

Status Register

**Figure 10-11. STATUS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | BUSY |
| R-0h | | | | | | | RH-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | ERR | | OVF | RESERVED |
| R-0h | | | | RH-0h | | R-0h | R-0h |

**Table 10-37. STATUS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-9 | RESERVED | R | 0h | |
| 8 | BUSY | RH | 0h | MATHACL busy bit.<br>0h = Compute has completed.<br>1h = Compute ongoing |
| 7-4 | RESERVED | R | 0h | |
| 3-2 | ERR | RH | 0h | Incorrect inputs/outputs.<br>0h = No Error in computation.<br>1h = DIVBY0 error |
| 1 | OVF | R | 0h | Overflow bit for MPY32, SQUARE32, DIV, MAC, and SAC functions<br>This bit will be set on overflow and will retain its value until cleared by writing 1 into CLR.CLR_OVF<br>0h = Overflow error. |
| 0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 10.5.10 STATUSCLR (Offset = 1140h) [Reset = 00000000h]

STATUSCLR is shown in Figure 10-12 and described in Table 10-38.

Return to the Summary Table.

Clear register for clearing flags in STATUS register

**Figure 10-12. STATUSCLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | CLR_ERR | CLR_OVF | RESERVED |
| R-0h | | | | | W-0h | W-0h | R-0h |

**Table 10-38. STATUSCLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | RESERVED | R | 0h | |
| 2 | CLR_ERR | W | 0h | Write 1 to this bit to clear STATUS.ERR field<br>0h = Writing 0 has no effect<br>1h = Clear STATUS.ERR |
| 1 | CLR_OVF | W | 0h | Write 1 to this bit to clear STATUS.OVF bit<br>0h = Writing 0 has no effect<br>1h = Clear STATUS.OVF |
| 0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

Chapter 11

# TinyEngine™ NPU

## 11.1 Overview

The TinyEngine™ Neural-network Processing Unit (NPU) module is a neural-network processing unit capable of supporting Artificial Intelligence (AI) and Machine Learning (ML) applications.

This NPU is a highly optimized core for deep convolutional neural networks (CNNs), supporting machine learning inference using pre-trained models. The NPU works in conjunction with the on-chip CPU to provide higher performance and lower power consumption for CNNs inference. With the capable of 640–2560MOPS (Mega Operations Per Second), the NPU provides up to two orders of magnitude improvement compared to a purely software-based implementation.

NPU features include:

- Up to 16 MACs/cycle (Multiply-Accumulate Operations Per Cycle) for 4-bit x 4-bit or 8-bit x 2-bit operation, equals to 2560 MOPS
- Batch normalization datapath supporting scaling and shifting operation
- Input data supports signed/unsigned data types and 8-bit/4-bit data length
- Weights support signed data with 8-bit/4-bit/2-bit data length
- Programmable engine with dedicated program and data memory (RAM)
- Read/write from system memory (RAM)
- Interrupt support on certain instruction
- Support 1D, 2D and Transposed Convolution layer
- Support Depthwise and Pointwise Convolution
- Support different strides, padding and kernel dimensions
- Support Average / Max Pooling model
- Support Residual Network
- Support mixed precision models – different layers can be implemented with different input and weight precision

## 11.2 NPU Operation

Figure 11-1 shows the top-level view of the modules within the chip.



Copyright © 2026 Texas Instruments Incorporated

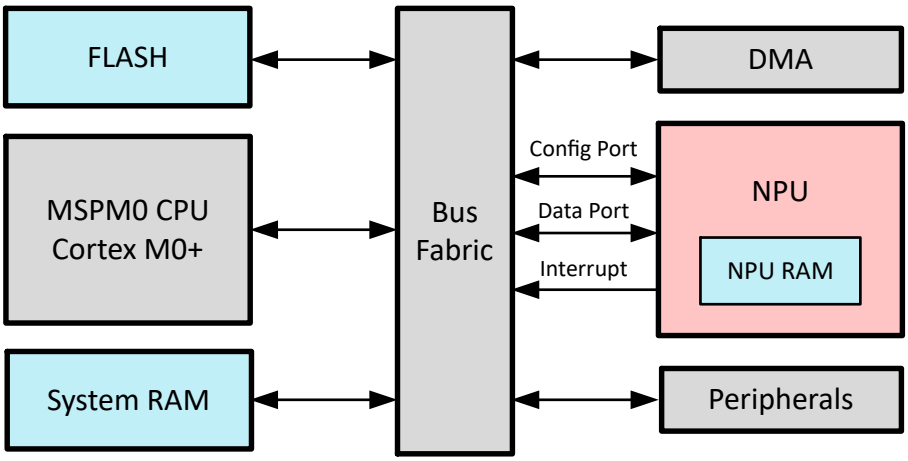



**Figure 11-1. MSP CPU/NPU System Structure**

MSPM0 with NPU system shared the on-chip memory including non-volatile FLASH memory and volatile RAM memory. The system RAM supports read/write with CPU or NPU. Additionally, NPU has a dedicated RAM for CNNs inference. The Edge AI model need fit within available FLASH memory range, and the peak RAM requirements need fit within available system RAM.

The overall task of CNNs inference is split between the MCU and the NPU. Table 11-1 shows each component's main task during CNNs inference:

**Table 11-1. Component Task Overview**

| Component | Main Task |
|---|---|
| FLASH | • NPU program, and ML parameter storage |
| System RAM | • Shared with CPU and NPU<br>• Storage for input data, ML parameters, and output data<br>• Storage for NPU inference instructions |
| NPU RAM | • Dedicated program and data memory (RAM) for NPU<br>• Storage for NPU inference instructions |
| CPU | • Copies inference code from FLASH to NPU RAM<br>• Copies ML parameters from FLASH or System RAM to NPU parameter buffer (RAM)<br>• Executes data processing to extract input features from raw data (optional) |
| NPU | • Process the CNNs inference with input data and ML parameters |

**Note**

Upon exit from an LPM such as STOP or STANDBY, the NPU needs to be re-initialized.

Figure 11-2 shows the ML flow where NPU works in conjunction with the on-chip CPU to process CNNs inference.



Copyright © 2026 Texas Instruments Incorporated

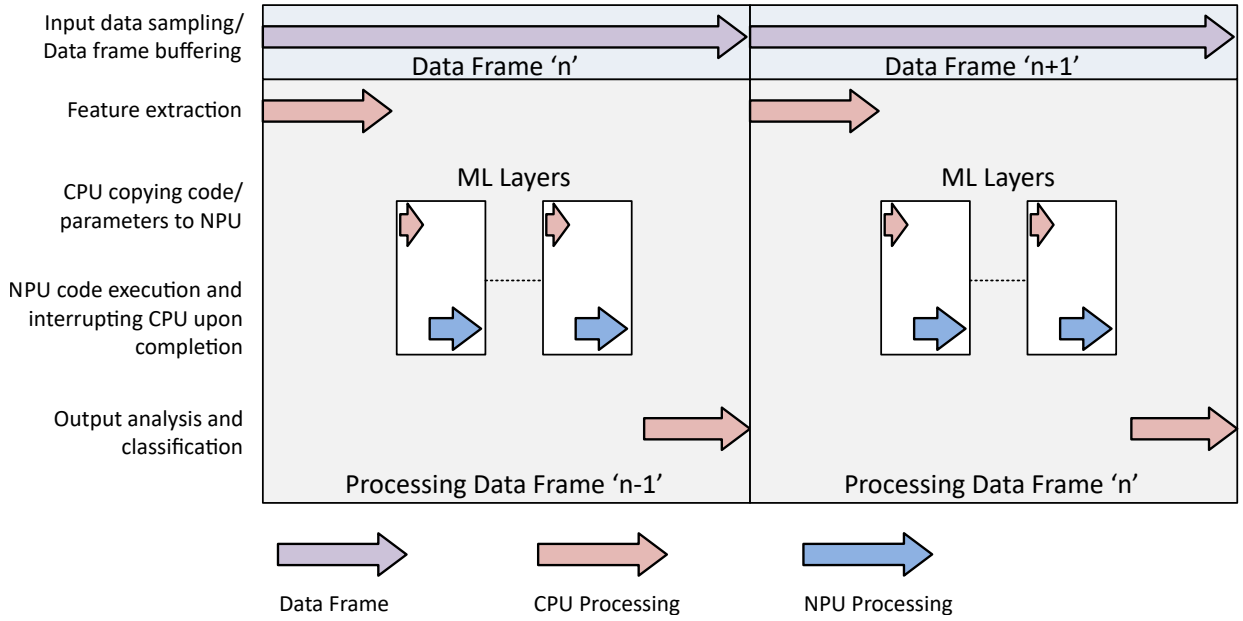



**Figure 11-2. Data/Program Flow Diagram**

The data is generated in real time and is buffered in system RAM (as data frames). When one data frame is ready for processing, the CPU can do the input feature map extraction (for example, computing the spectral components using FFT). The input features extracted (or raw data in some cases) is then processed by one or more ML layers. This phase can involve multiple sub-phases, in which CPU copies the inference code and ML parameters to NPU RAM and then on NPU completing one sub-phase copies the next set of code and parameters until all the ML layers are computed for the data frame being processed.

In the final phase, CPU can do the analysis of output and necessary classification. As can be seen in the flow diagram, while the processing is being done on the current frame, next frame is being constructed. The processing has to be completed within the time required for one data-frame construction in the real-time application.

## 11.3 NPU Toolchain

For the expert users, train your own model and compile the model with TI Neural Network Compiler. For the catalog users, load and train models with tools from TI: Model Composer GUI or TI's command-line Modelmaker tool for an advanced set of capabilities. Both options automatically generate source code for the MSPM0, eliminating the need to manually write code.

Figure 11-3 shows the toolchain overview and steps to add NPU support to a project, starting with importing or using existing models from TI, training the models, generating the associated software libraries, and integrating into an existing Code Composer Studio™ IDE project.



Copyright © 2026 Texas Instruments Incorporated

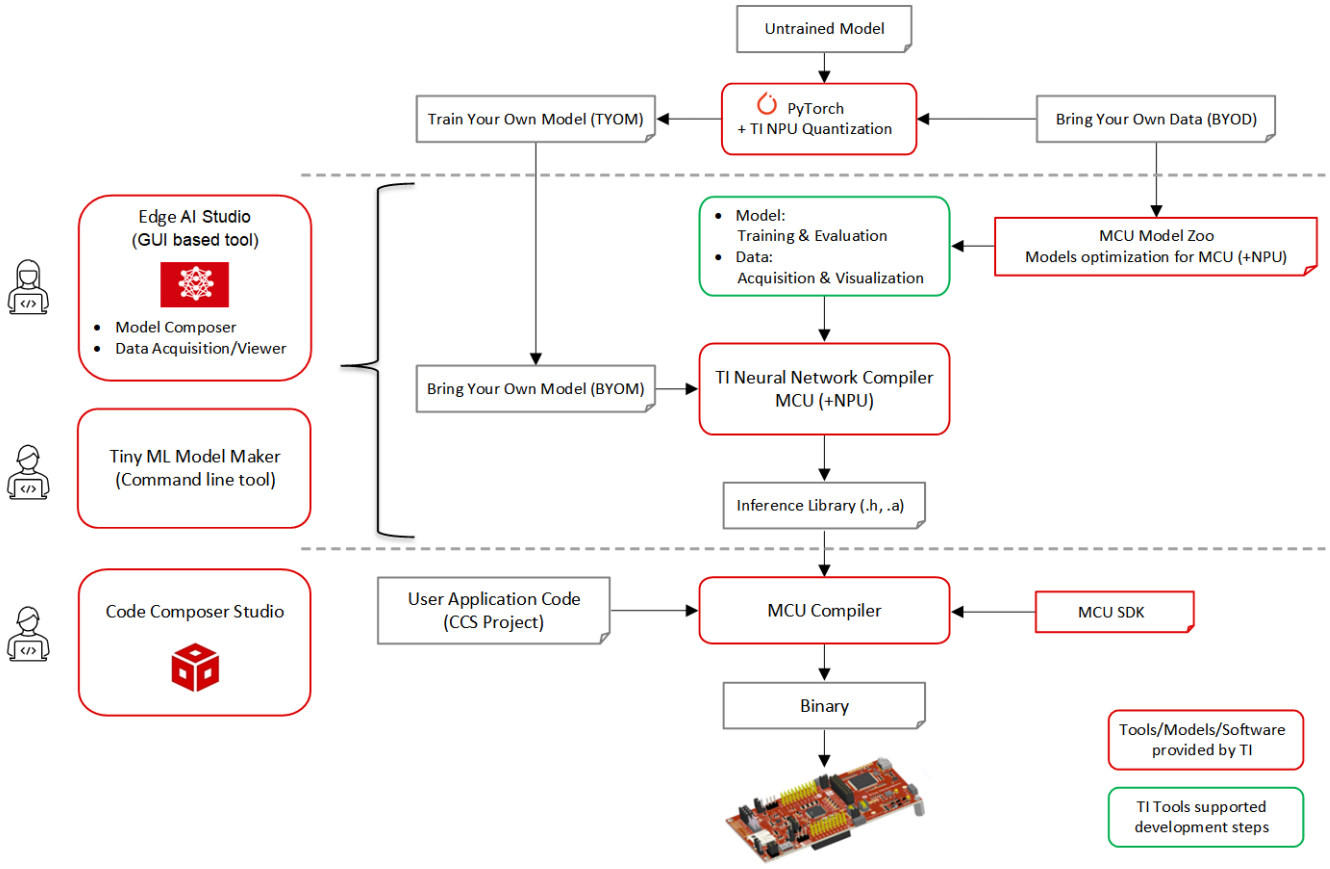



**Figure 11-3. NPU Development Flow**

NPU toolchain related collaterals:

- MSPM0 EdgeAI User Guide
- Edge AI Studio
- Neural Network Compiler
- Tiny ML Tensorlab
- Tiny ML Modelmaker



Copyright © 2026 Texas Instruments Incorporated

Chapter 12

# AES

The AES accelerator module accelerates encryption and decryption operations in hardware based on the FIPS PUB 197 advanced encryption standard (AES).


**12.1 AES Overview** ..... 807
**12.2 AES Operation** ..... 808
**12.3 AES Registers** ..... 829





Copyright © 2026 Texas Instruments Incorporated

## 12.1 AES Overview

**Note**

Not all the MSPM0G devices support support AES. See device-specific datasheet to determine if AES is present.

The AES accelerator module performs encryption and decryption of 128-bit data blocks with a 128-bit or 256-bit key in hardware according to the advanced encryption standard (AES). AES is a symmetric-key block cipher algorithm specified in FIPS PUB 197.

The AES accelerator features include:

- AES 128-bit block encryption and decryption
- DMA trigger support for automating ECB, CBC, OFB, and CFB block cipher modes as defined in NIST SP 800-38
- Support for accelerating CTR cipher mode by encrypting precalculated (nonce || counter) blocks and accelerating XOR of plaintext with the generated key stream
- Support for accelerating CBC-MAC tag computation (CBC DMA mode with zero initialization vector)
- On-the-fly key expansion for encryption and decryption
- Offline key generation for decryption
- Shadow register storing the initial key for all key lengths
- 8-bit byte or 32-bit word access to provide key data, input data, and output data
- AES ready interrupt
- Supported in RUN and SLEEP (see the *Operating Modes* section of the device technical reference manual)

The AES accelerator hardware consists of the 128-bit state memory and associated input/output registers, the AES encryption/decryption core and control logic, and the 256-bit AES key memory and associated input register. The AES hardware is shown in Figure 12-1.

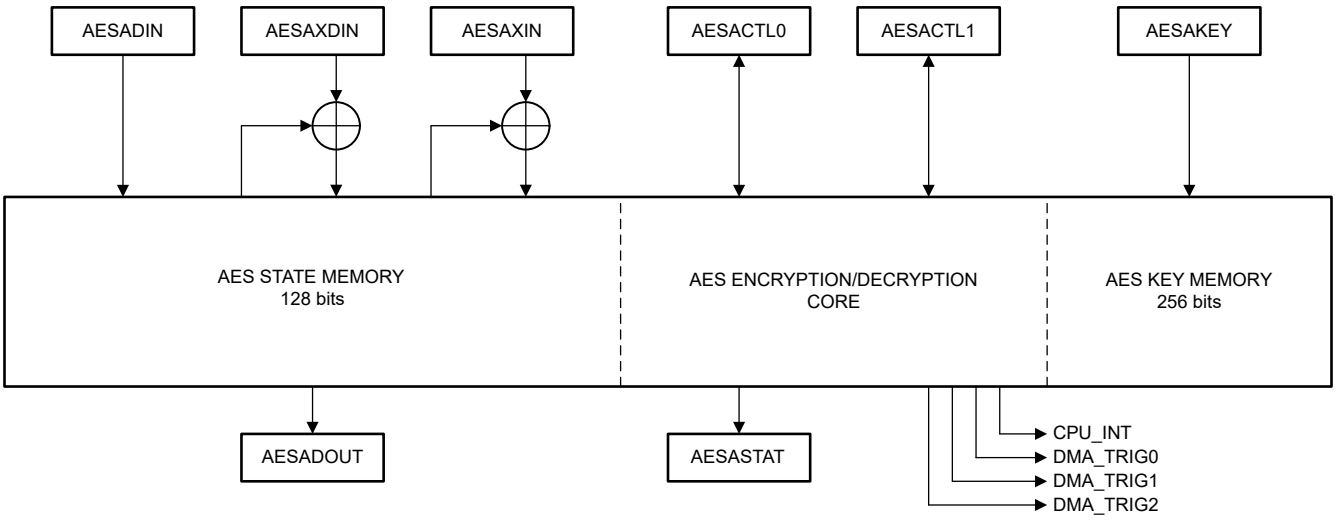



**Figure 12-1. AES Accelerator Block Diagram**

### *12.1.1 AES Performance*

The AES accelerator provides fast encryption and decryption of 128-bit blocks. AES accelerator performance in both cycles and execution time for block encryption and block decryption (with pregenerated decryption key) is given in Table 12-1.



Copyright © 2026 Texas Instruments Incorporated

**Table 12-1. AES Hardware Accelerator Key Performance Metrics**

| AES Key Length | Encryption (OPx==0x0) | | | Decryption (OPx==0x3) | | |
|---|---|---|---|---|---|---|
| | Cycles | Time (32MHz) | Time (80MHz) | Cycles | Time (32MHz) | Time (80MHz) |
| 128-bit | 168 | 5.25 µs | 2.10 µs | 168 | 5.25 µs | 2.10 µs |
| 256-bit | 234 | 7.31 µs | 2.93 µs | 234 | 7.31 µs | 2.93 µs |

## 12.2 AES Operation

The AES accelerator is configured with user software, and has two general modes of operation (configured with the AESACTL0 register):

- **Single block mode**
  - One 128-bit data block is encrypted or decrypted at a time under software control
  - No DMA triggers are generated directly by the AES hardware; application software writes and reads all data or configures DMA to load data with an alternate trigger
  - This mode can also be used to pregenerate the last round key needed for automated decryption in block cipher mode
- **Automated block cipher mode**
  - Multiple 128-bit data blocks can be encrypted or decrypted in an automated way using ECB, CBC, CFB, or OFB block cipher modes of operation
  - DMA triggers are generated to move data in and out of the accelerator automatically for a specified number of blocks with the assistance of 2 or 3 DMA channels (depending on the selected block cipher mode)

Regardless of single block mode or block cipher mode, the key length and operation type are specified by the KL and OP fields in the AESACTL0 register. The KLx and OPx fields are not reset by the AES software reset control (the SWRST bit in the AESACTL0 register). Whenever the KLx or OPx fields are changed, the key memory is reset.

- The KLx field selects the AES key size which is to be used (128 or 256 bits).
- The OPx field selects the AES operation to perform (encrypt, decrypt, generate last round decryption key, decrypt with pregenerated last round decryption key).

The KLx and OPx combination determines the cycle count per block operation, as shown in Table 12-2.

**Table 12-2. AES Operation Overview by OPx (operation) and KLx (key length) Configuration**

| AESOPx | AESKLx | Operation | Clock Cycles |
|---|---|---|---|
| 00 | 00 | AES128 encryption | 168 |
| | 10 | AES256 encryption | 234 |
| 01 | 00 | AES128 decryption | 215 |
| | 10 | AES256 decryption | 292 |
| 10 | 00 | AES128 encryption key schedule is performed (to generate last roundkey) | 53 |
| | 10 | AES256 encryption key schedule is performed (to generate last roundkey) | 68 |
| 11 | 00 | AES128 decryption (with last roundkey) is performed | 168 |
| | 10 | AES256 decryption (with last roundkey) is performed | 234 |

Internally, the AES algorithm's operations are performed on a two-dimensional array of bytes called the STATE. The STATE consists of four rows of bytes, each containing four bytes, independent of whether AES128 or AES256 is performed. The input is assigned to the STATE array as shown in Figure 12-2, with in[0] being the first data byte written into one of the AES accelerators input registers (AESADIN, AESAXDIN, and AESXIN). The encrypt or decrypt operations are then conducted on the STATE array, after which its final values can be read from the output with out[0] being the first data byte read from the AES accelerator data output register (AESADOUT).



Copyright © 2026 Texas Instruments Incorporated

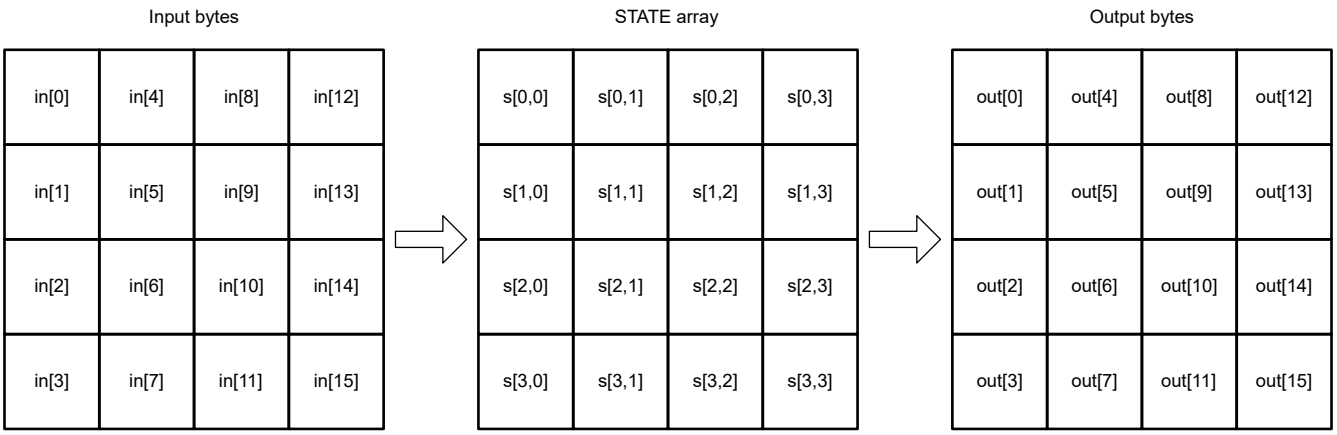



**Figure 12-2. AES State Array Input and Output**

If an encryption is to be performed, the initial state is called plaintext. If a decryption is to be performed, the initial state is called ciphertext.

### *12.2.1 AES Register Access Rules*

The AES accelerator must be enabled before being configured for use through the PWREN register (see peripheral power enable). The AES accelerator is located in power domain 1 (PD1) and as such is only available in RUN and SLEEP modes. If the AES accelerator is enabled by application software, entry into STOP or STANDBY low-power modes will force the AES accelerator to be disabled while the device is in STOP or STANDBY mode, and the AES accelerator register contents will be lost (including any key or state contents).

#### Data Sizes for Reading and Writing Keys and Data

The module allows 32-bit word or 8-bit byte access to all key and data registers (AESAKEY, AESADIN, AESAXDIN, AESAXIN, and AESADOUT). Word and byte access must not be mixed while reading from or writing into one of the registers. However, it is possible to write one of the registers using byte access and another using word access.

#### General Access Restrictions

While the AES accelerator is busy (AESBUSY bit is set in the AESASTAT register):

- AESADOUT always reads as zero.
- The DOUTCNTx counter, the DOUTRD flag, and the DINWR flag are reset.
- Any attempt to change OPx, KLx, DINWR, or KEYWR is ignored.
- Writing to AESAKEY, AESADIN, AESAXDIN, or AESAXIN aborts the current operation, the complete module is reset (except for the IMASK registers, OPx, and KLx), and the AES error flag ERRFG is set.

AESADIN, AESAXDIN, AESAXIN, and AESAKEY are write-only registers and always read as zero.

Writing data into AESADIN, AESAXDIN, or AESAXIN influences the content of the corresponding output data; for example, writing in[0] alters out[0], writing in[1] alters out[1], and so on. However, interleaved operation is possible; for example, first reading out[0] and then writing in[0], and continuing with reading out[1] and then writing in[1], and so on. This interleaved operation must be either byte or word access on in[x] and out[x].

#### Access Restriction With Block Cipher Modes Enabled

When automated block cipher modes are enabled (CMEN is set in AESACTL0) and a cipher block operation is being processed (BLKCNTx > 0), writes to the following bits are ignored, independent of the BUSY bit status: CMEN, CMx, KLx, OPx, and BLKCNTx. Writing to AESAKEY aborts the cipher block mode operation if BUSY = 1, and the complete module is reset (except for the IMASK registers, OPx, and KLx).



Copyright © 2026 Texas Instruments Incorporated

### 12.2.2 Loading the Key

The key can be loaded by writing to the AESAKEY register or by setting the KEYWR bit. Depending on the selected key length (KLx in the AESCTL0 register), a different number of bits must be loaded:

- If KLx = 00, the 128-bit key must be loaded using either 16 byte-writes or 4 word-writes to AESAKEY
- If KLx = 10, the 256-bit key must be loaded using either 32 byte-writes or 8 word-writes to AESAKEY

The key memory is reset after changing the KLx bits.

If a key was loaded previously without changing the OPx bits, the KEYWR flag is cleared with the first write access to AESAKEY.

If an operation is triggered without writing a new key, the last key is used. The key must always be written before writing the data.

The AES algorithm operates not only on the STATE but also on the key. To avoid the need to reload the key for each operation, a key buffer is included such that the key expansion operations for round key generation to not remove the cipher key loaded in AESAKEY.

### 12.2.3 Loading Data

The state can be loaded by writing to AESADIN, AESAXDIN, or AESAXIN with 16 byte writes or 4 word writes. Do not mix byte and word mode when writing the state. Writing to a mixture of AESADIN, AESAXDIN, and AESAXIN using the same byte or word data format is allowed. When the 16th byte or 4th word of the state is written, DINWR is set in the AESASTAT register.

When writing to AESADIN, the corresponding byte or word of the state is overwritten. If AESADIN is used to write the last byte or word of the state, encryption or decryption starts automatically.

When writing to AESAXDIN, the corresponding byte or word is XORed with the current byte or word of the state. If AESAXDIN is used to write the last byte or word of the state, encryption or decryption starts automatically.

Writing to AESAXIN has the same behavior as writing to AESAXDIN: the corresponding byte or word is XORed with the current byte or word of the state; however, writing the last byte or word of the state using AESAXIN does not start encryption or decryption.

**Note**

The AESAXIN register can be used to accelerate the XOR operation of plaintext with an encrypted key stream (nonce || counter) when using the AES accelerator to assist with implementation of a CTR cipher mode. In a CTR cipher mode scenario, a plaintext block can be loaded to AESAXIN after encryption of the (nonce || counter) value has completed, as the encrypted (nonce || counter) value will be held in the STATE. After loading data into AESAXIN, the (plaintext □ (nonce || counter) output can be read from AESADOUT.

### 12.2.4 Reading Data

The STATE can be read when the BUSY bit in AESASTAT is cleared. The STATE is read using 16 byte reads or 4 word reads from AESADOUT. When all 16 bytes are read, the DOUTRD flag in AESASTAT indicates completion of the read.

### 12.2.5 Triggering an Encryption or Decryption

An encrypt or decrypt operation is triggered if the STATE was completely written in the AESADIN or AESAXDIN registers. Alternatively, the bit DINWR bit in the AESASTAT register can be set to trigger an operation if CMEN is cleared.

### 12.2.6 Single Block Operations

Single block operations include encryption, decryption, and decryption key generation.



Copyright © 2026 Texas Instruments Incorporated

#### 12.2.6.1 Encryption

To perform encryption:

1. Set OPx = 0x0 to select encryption. Note that changing the OPx bits clears the KEYWR flag and a new key must be loaded in the next step.
2. Load the key as described in Section 12.2.2.
3. Load the STATE (data) as described in Section 12.2.3. After the data is loaded, the AES module starts the encryption process.
4. Wait for the operation to complete (wait for the BUSY flag to clear).
5. After the encryption is complete, the result can be read from AESADOUT as described in Section 12.2.4.
6. To encrypt additional data with the same key loaded in step 2, write the new data into AESADIN after the results of the operation on the previous data were read from AESADOUT. When an additional 16 data bytes are written, the module automatically starts the encryption using the key loaded in step 2.

**Note**

Setting the DINWR bit in the AESASTAT register triggers another encryption, and the AES accelerator starts the encryption using the output data from the previous encryption as the input data. This is useful in certain block cipher modes with feedback.

The encryption process is given in Figure 12-3 (128-bit key with 10 rounds shown).

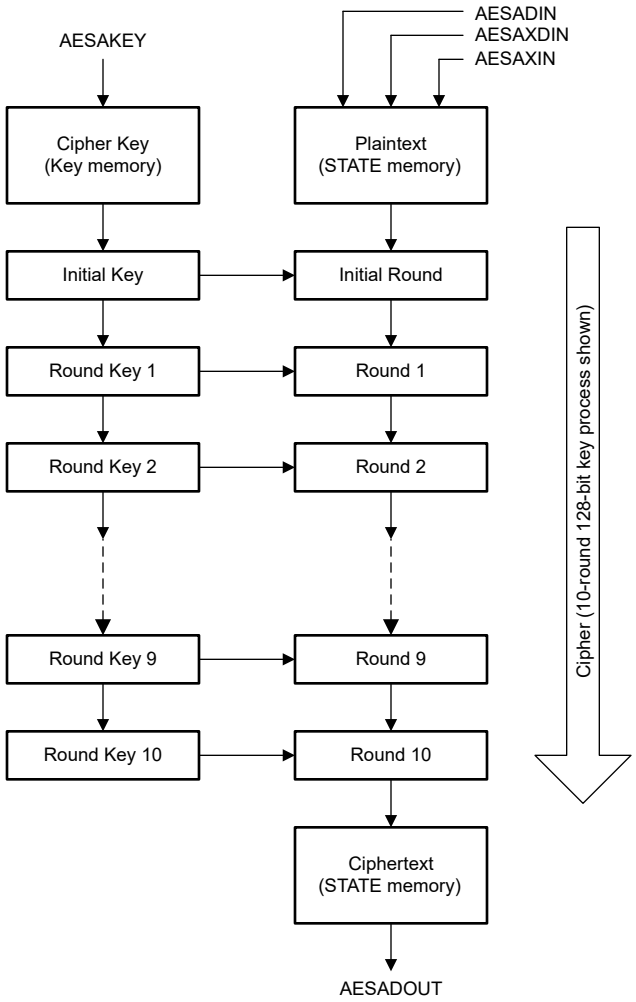



**Figure 12-3. AES Encryption (OPx=0x0, 128-bit Key)**



Copyright © 2026 Texas Instruments Incorporated

#### 12.2.6.2 Decryption

To perform decryption:

1. Set OPx = 0x1 to select decryption using the same key used for encryption, or set OPx = 0x3 if the first round key required for decryption (the last round key used for encryption) is already pregenerated and will be loaded in step 2. Changing the OPx bits clears the KEYWR flag, and a new key must be loaded in step 2.
2. Load the key according to Section 12.2.2.
3. Load the STATE (data) according to Section 12.2.3. After the data is loaded, the AES module starts the decryption.
4. Wait for the operation to complete (wait for the BUSY flag to clear).
5. After the decryption is ready, the result can be read from AESADOUT as described in Section 12.2.4.
6. If additional data should be decrypted with the same key loaded in step 2, new data can be written into AESADIN after the results of the operation on the previous data were read from AESADOUT. When additional 16 data bytes are written, the module automatically starts the decryption using the key loaded in step 2.

The decryption process with decryption round key generation is given in Figure 12-4 (128-bit key with 10 rounds shown).

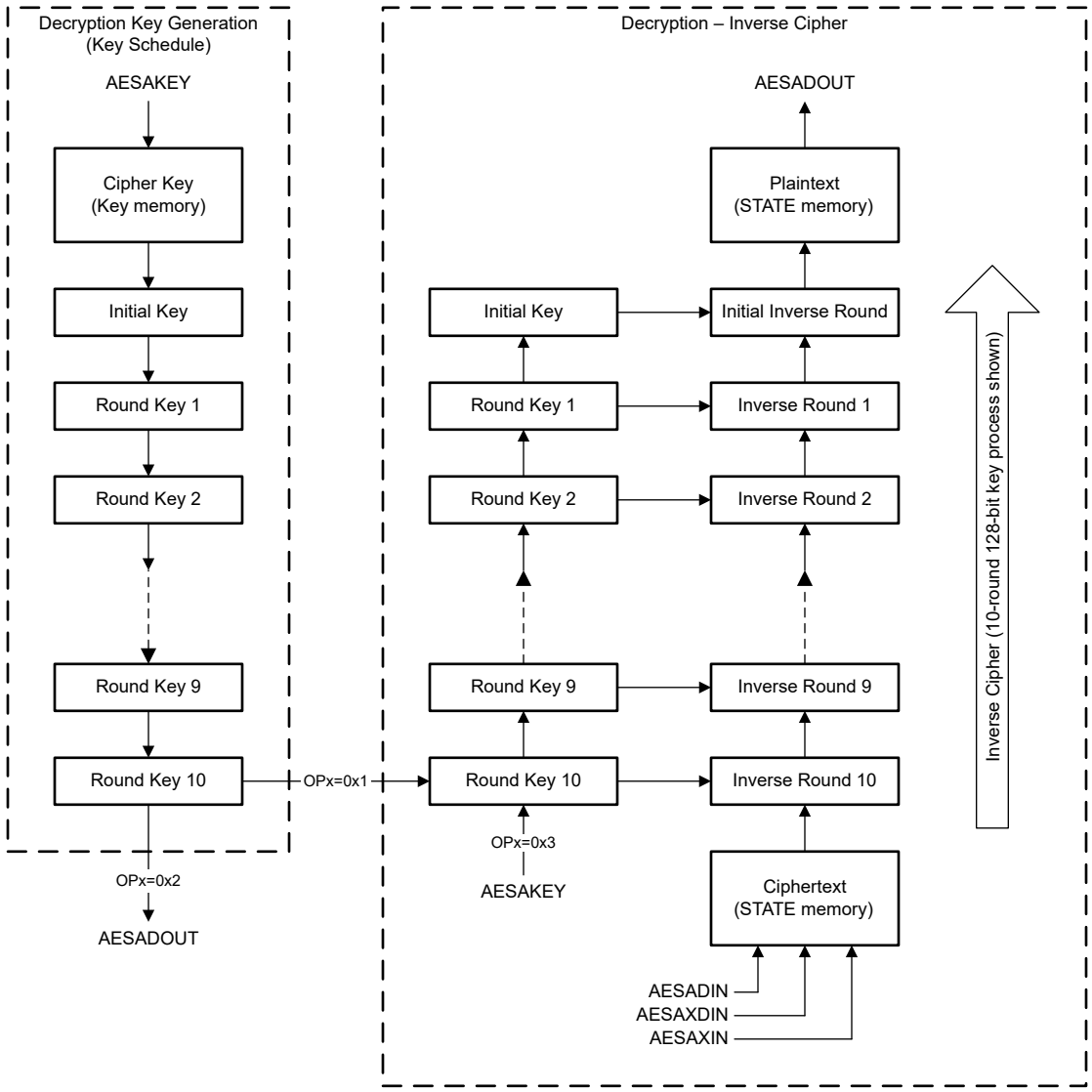



**Figure 12-4. AES Decryption (OPx = 0x1, 128-bit Key)**



Copyright © 2026 Texas Instruments Incorporated

#### 12.2.6.2.1 Pregenerating a Decryption Key

When decryption is performed using OPx=0x1, the original cipher key used for encryption is loaded into AESAKEY. Decryption requires execution of the key schedule to obtain the round key used for the last round of encryption. That final encryption round key is then used by the decryption logic as the first round key for decryption. In the case of OPx=0x1, before the decryption begins the key schedule will be executed on the key material loaded into AESAKEY to obtain the last round key. Once the last round key used for encryption is generated, the decryption process begins automatically.

While OPx=0x1 is convenient for single block cases (as the original encryption key can be loaded without special handling), it is not efficient for handling multiple blocks, because the key schedule will be executed for each block to obtain the last round key which is needed to start the decryption process. To reduce the cycle count required for decryption, it is possible to pregenerate the last round encryption key using OPx=0x2. Then that pregenerated round key can be used to immediately start any decryption operations by loading the pregenerated key material and running the OPx=0x3 decryption operation, which assumes that the key material is the round key from the last round of encryption.

To execute the key schedule and obtain the last round key used for encryption (first round key used for decryption) without actually running a decryption operation, follow the steps below:

1. Set OPx to 0x2 to select decryption key generation.
2. Load the cipher key used for encryption into AESAKEY as described in Section 12.2.2. Generation will start as soon as the key material is loaded.
3. While the AES module is busy executing the key schedule, the AESBUSY bit is set. 53 clock cycles are required to complete the key generation for a 128-bit cipher key. After completion, the AESRDY interrupt is set and the 128-bit result can be read from the AESDOUT. AESRDY is cleared when reading AESADOUT or writing to AESAKEY or AESADIN.

Once a decryption first round key is generated, it can be loaded into AESAKEY and used for decryption with OPx=3.

### 12.2.7 Block Cipher Mode Operations

Block cipher modes are used to implement the electronic code book (ECB), cipher block chaining (CBC), output feedback (OFB) and cipher feedback (CFB) block cipher modes using AES as the underlying block cipher. These modes work together with the DMA through 2 or 3 DMA triggers to support easy and fast encryption or decryption of more than 128 bits.

#### 12.2.7.1 Electronic Codebook (ECB) Mode

The electronic codebook (ECB) cipher is the simplest block cipher mode. The plaintext data is divided into 128-bit blocks, and each block is encrypted and decrypted independently from any other block. The ECB cipher is shown in Figure 12-5. Note that each 128-bit data block can be encrypted or decrypted individually without any knowledge of the other blocks of plaintext or ciphertext.

While ECB is simple to understand and implement, it has a key disadvantage: the same 128-bit plaintext block is always encrypted into the same ciphertext block, allowing patterns in the ciphertext to be detected.



Copyright © 2026 Texas Instruments Incorporated

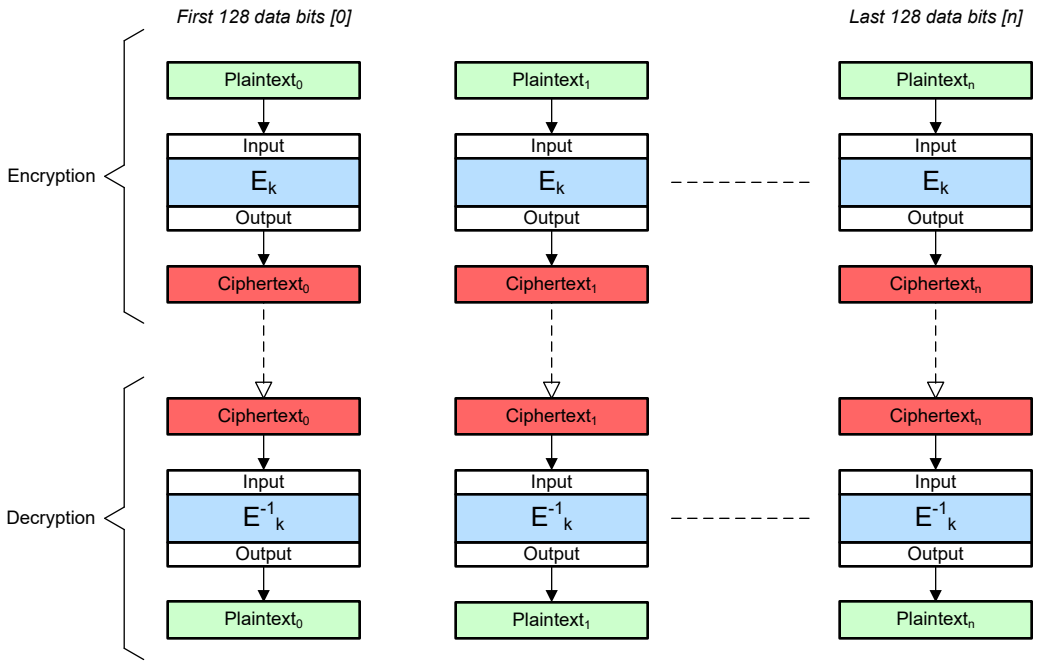



**Figure 12-5. ECB Cipher**

The AES accelerator supports automated encryption and decryption of more than 128 bits of data in ECB block cipher mode through the use of DMA together with the AES accelerator. ECB mode utilizes two DMA channels (referred to as DMA_A, DMA_B in this section) for output and input to the AES accelerator, respectively. The DMA channel usage for ECB mode encryption and decryption operations is given in Table 12-3.

**Table 12-3. ECB Block Cipher DMA Behavior**

| AES Operation | | | AES DMA Triggers | | |
|---|---|---|---|---|---|
| **CMEN** | **CMx** | **OPx** | **DMA_A (DMA_TRIG0)** | **DMA_B (DMA_TRIG1)** | **DMA_C (DMA_TRIG2)** |
| 0x1 | 0x0 (ECB) | 0x0 (encryption) | Used to read ciphertext from AESADOUT | Used to write plaintext to AESADIN, which also triggers start of block encryption when 128 bits are written to AESADIN | Not used |
| | | 0x1/0x2 (decryption) | Used to read plaintext from AESADOUT | Used to write ciphertext to AESADIN, which also triggers start of block decryption when 128 bits are written to AESADIN | Not used |

#### *12.2.7.1.1 ECB Encryption*

ECB mode encryption of *N* blocks of plaintext into *N* blocks of ciphertext without CPU interaction is achieved through the use of 2 DMA channels (referred to as DMA_A and DMA_B). To implement ECB encryption follow these steps:

1. Configure the AESACTL0 register for block cipher encryption mode using ECB:
   a. Set CMEN to enable block cipher mode
   b. Set CMx to ECB mode
   c. Set OPx to 0x0 (encryption mode)
2. Load key as described in Section 12.2.2
3. Configure DMA_A channel for saving ciphertext:
   a. Set DMA channel trigger selection to AES0 trigger
   b. Set DMA channel source address to AESADOUT
   c. Set DMA channel destination address to location where ciphertext is to be stored (for example, SRAM)
   d. Set DMA channel transfer size to *N*∗4



Copyright © 2026 Texas Instruments Incorporated

e. Set DMA channel mode to single transfer mode
f. In the AES event registers, unmask DMA0 in the IMASK register of DMA_TRIG0

4. Configure DMA_B channel for loading plaintext:
   a. Set DMA channel trigger selection to AES1 trigger
   b. Set DMA channel source address to location where plaintext is stored (for example, SRAM)
   c. Set DMA channel destination address to AESADIN
   d. Set DMA channel transfer size to *N*∗4
   e. Set DMA channel mode to single transfer mode
   f. In the AES event registers, unmask DMA1 in the IMASK register of DMA_TRIG

      1

5. Configure and enable the DMA interrupt for the DMA_A channel in the DMA controller
6. Start encryption by writing the block count *N* to BLKCNTx in the AESACTL1 register
7. Wait for the DMA_A channel interrupt which indicates completion of the operation. The ciphertext output will be stored in the location configured in step 3c.

#### *12.2.7.1.2 ECB Decryption*

ECB mode decryption of *N* blocks of ciphertext into *N* blocks of plaintext without CPU interaction is achieved through the use of 2 DMA channels (referred to as DMA_A and DMA_B). To implement ECB decryption follow these steps:

1. Configure the AESACTL0 register to pregenerate the decryption key (last round key):
   a. Clear CMEN to disable block cipher mode
   b. Set OPx to 0x2 (decryption key generation)
   c. Write key into AESAKEY register as described in Section 12.2.2
   d. Wait for the BUSY status in the AESSTAT register to clear, indicating key generation has completed
2. Configure the AESACTL0 register for block cipher decryption mode using ECB:
   a. Set CMEN to enable block cipher mode
   b. Set CMx to ECB mode
   c. Set OPx to 0x3 (decryption mode)
3. Set AESKEYWR bit in AESSTAT (to use pregenerated key from step 1)
4. Configure DMA_A channel for saving plaintext:
   a. Set DMA channel trigger selection to AES0 trigger
   b. Set DMA channel source address to AESADOUT
   c. Set DMA channel destination address to location where plaintext is to be stored (for example, SRAM)
   d. Set DMA channel transfer size to *N*∗4
   e. Set DMA channel mode to single transfer mode
   f. In the AES event registers, unmask DMA0 in the IMASK register of DMA_TRIG

      0

5. Configure DMA_B channel for loading ciphertext:
   a. Set DMA channel trigger selection to AES1 trigger
   b. Set DMA channel source address to location where ciphertext is stored (for example, SRAM)
   c. Set DMA channel destination address to AESADIN
   d. Set DMA channel transfer size to *N*∗4
   e. Set DMA channel mode to single transfer mode
   f. In the AES event registers, unmask DMA1 in the IMASK register of DMA_TRIG

      1

6. Configure and enable the DMA interrupt for the DMA_A channel in the DMA controller
7. Start decryption by writing the block count *N* to BLKCNTx in the AESACTL1 register



Copyright © 2026 Texas Instruments Incorporated

8. Wait for the DMA_A channel interrupt which indicates completion of the operation. The plaintext output will be stored in the location configured in step 3c.

#### 12.2.7.2 Cipher Block Chaining (CBC) Mode

The cipher block chaining (CBC) cipher mode builds upon the ECB cipher mode to make the ciphertext output for each block dependent not only on the plaintext and the cipher key *k*, but also upon the ciphertext of the previous block. The CBC cipher is shown in Figure 12-6. Like ECB mode, the plaintext data is divided into 128-bit blocks. Unlike ECB mode, in CBC mode each new plaintext block is bit-wise XORed with the previous ciphertext block to create the input to the AES block cipher.

In CBC mode, an unpredictable initialization vector (IV) must be provided. The initialization vector is XORed with the first plaintext block, as there is no "previous" ciphertext block to XOR the first plaintext block with.

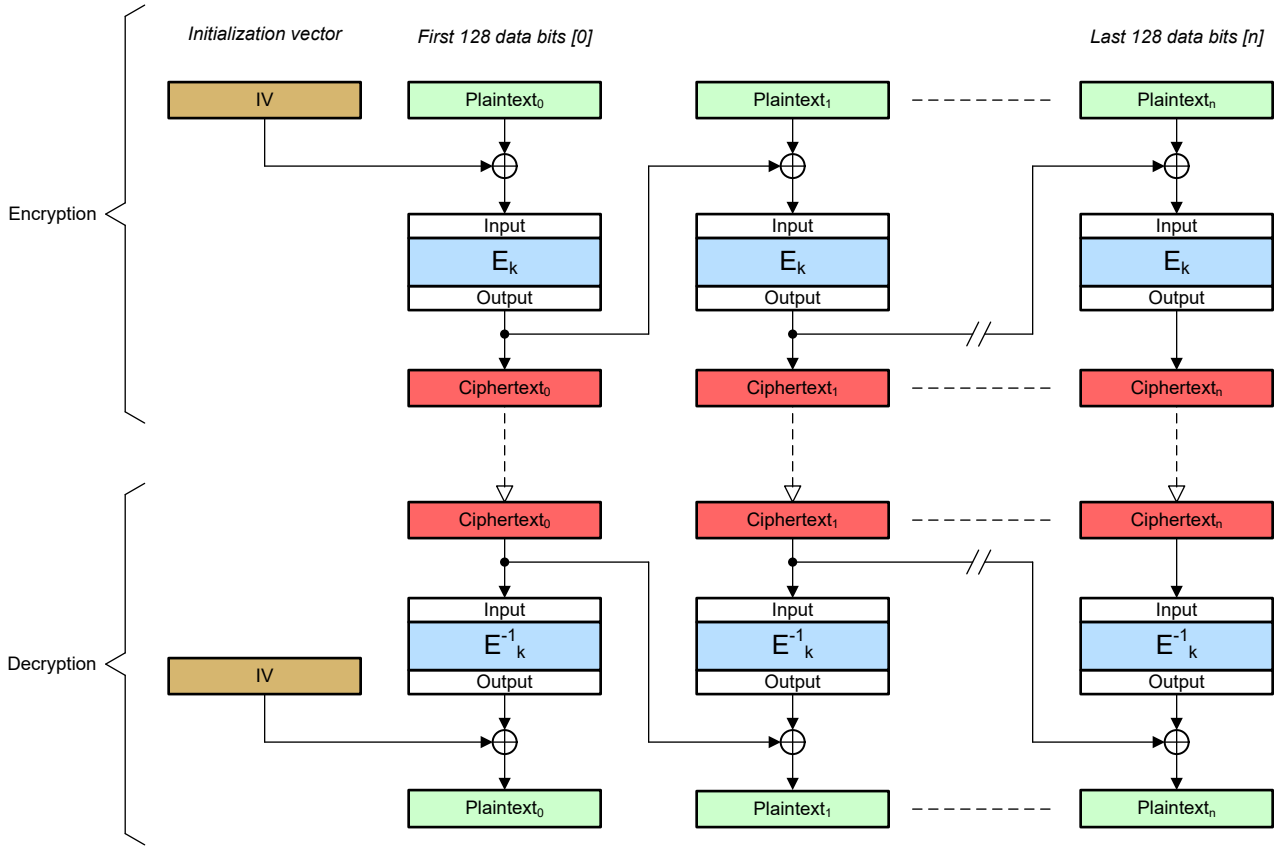



**Figure 12-6. CBC Cipher**

The AES accelerator supports automated encryption and decryption of more than 128 bits of data in CBC block cipher mode through the use of DMA together with the AES accelerator. CBC mode utilizes two DMA channels (referred to as DMA_A, DMA_B in this section) for output and input to the AES accelerator, respectively, when encrypting. CBC mode utilizes three DMA channels (DMA_A, DMA_B, and DMA_C) for input, output, and input to the AES accelerator when decrypting. The DMA channel usage for CBC mode encryption and decryption operations is given in Table 12-4.



Copyright © 2026 Texas Instruments Incorporated

**Table 12-4. CBC Block Cipher DMA Behavior**

| AES Operation | | | AES DMA Triggers | | |
|---|---|---|---|---|---|
| **CMEN** | **CMx** | **OPx** | **DMA_A (DMA_TRIG0)** | **DMA_B (DMA_TRIG1)** | **DMA_C (DMA_TRIG2)** |
| 0x1 | 0x1 (CBC) | 0x0 (encryption) | Used to read ciphertext from AESADOUT | Used to write plaintext to AESAXIN, which also triggers start of block encryption when 128 bits are written to AESADIN | Not used |
| | | 0x1/0x2 (decryption) | Used to write the initialization vector and previous ciphertext block to AESAXIN | Used to read plaintext from AESADOUT | Used to write next ciphertext to AESADIN, which also triggers start of block decryption when 128 bits are written to AESADIN |

#### *12.2.7.2.1 CBC Encryption*

CBC-mode encryption of *N* blocks of plaintext into *N* blocks of ciphertext without CPU interaction is achieved through the use of 2 DMA channels (referred to as DMA_A and DMA_B). To implement CBC encryption follow these steps:

1. Configure the AESACTL0 register for block cipher encryption mode using CBC:
   a. Reset AES module to clear internal state memory (set SWRST in the AESACTL0 register)
   b. Set CMEN to enable block cipher mode
   c. Set CMx to CBC mode
   d. Set OPx to 0x0 (encryption mode)
2. Load key as described in Section 12.2.2
3. Load initialization vector (IV) into AESAXIN
   a. Loading to AESAXIN does not start any encryption
   b. The STATE must be reset (step 1a above) so that the IV is XORed with zeros when loaded
4. Configure DMA_A channel for saving ciphertext:
   a. Set DMA channel trigger selection to AES0 trigger
   b. Set DMA channel source address to AESADOUT
   c. Set DMA channel destination address to location where ciphertext is to be stored (for example, SRAM)
   d. Set DMA channel transfer size to *N*∗4 words
   e. Set DMA channel mode to single transfer mode
   f. In the AES event registers, unmask DMA0 in the IMASK register of DMA_TRIG

      0
5. Configure DMA_B channel for loading plaintext:
   a. Set DMA channel trigger selection to AES1 trigger
   b. Set DMA channel source address to location where plaintext is stored (for example, SRAM)
   c. Set DMA channel destination address to AESAXDIN
   d. Set DMA channel transfer size to *N*∗4 words
   e. Set DMA channel mode to single transfer mode
   f. In the AES event registers, unmask DMA1 in the IMASK register of DMA_TRIG

      1
6. Configure and enable the DMA interrupt for the DMA_A channel in the DMA controller
7. Start encryption by writing the block count *N* to BLKCNTx in the AESACTL1 register
8. Wait for the DMA_A channel interrupt which indicates completion of the operation. The ciphertext output will be stored in the location configured in step 4c.

#### *12.2.7.2.2 CBC Decryption*

CBC-mode decryption of *N* blocks of ciphertext into *N* blocks of plaintext without CPU interaction is achieved through the use of 3 DMA channels (referred to as DMA_A, DMA_B, and DMA_C). To implement CBC decryption follow these steps:



Copyright © 2026 Texas Instruments Incorporated

1. Configure the AESACTL0 register to pre-generate the decryption key (last round key):
   a. Clear CMEN to disable block cipher mode
   b. Set OPx to 0x2 (decryption key generation)
   c. Write key into AESAKEY register as described in Section 12.2.2
   d. Wait for the BUSY status in the AESSTAT register to clear, indicating key generation has completed
2. Configure the AESACTL0 register for block cipher decryption mode using CBC:
   a. Set CMEN to enable block cipher mode (0x1)
   b. Set CMx to 0x1 (CBC mode)
   c. Set OPx to 0x3 (decryption mode)
3. Set the AESKEYWR bit in AESSTAT (to use pre-generate key from step 1)
4. Configure DMA_A channel for loading the initialization vector and ciphertext:
   a. Set up the ciphertext buffer in memory and pre-pend the initialization vector (IV) to the ciphertext
   b. Set DMA channel trigger selection to AES0 trigger
   c. Set DMA channel source address to location where initialization vector is stored in front of the following ciphertext (for example, SRAM)
   d. Set DMA channel destination address to AESAXIN
   e. Set DMA channel transfer size to *N*∗4 words
   f. Set DMA channel mode to single transfer mode
   g. In the AES event registers, unmask DMA0 in the IMASK register of DMA_TRIG
5. Configure DMA_B channel for saving plaintext:
   a. Set DMA channel trigger selection to AES1 trigger
   b. Set DMA channel source address to AESADOUT
   c. Set DMA channel destination address to location where plaintext is to be stored (for example, SRAM)
   d. Set DMA channel transfer size to *N*∗4 words
   e. Set DMA channel mode to single transfer mode
   f. In the AES event registers, unmask DMA1 in the IMASK register of DMA_TRIG
6. Configure DMA_C channel for loading ciphertext:
   a. Set DMA channel trigger selection to AES2 trigger
   b. Set DMA channel source address to location where ciphertext is stored after the IV (for example, SRAM)
   c. Set DMA channel destination address to AESADIN
   d. Set DMA channel transfer size to *N*∗4 words
   e. Set DMA channel mode to single transfer mode
   f. In the AES event registers, unmask DMA2 in the IMASK register of DMA_TRIG
7. Configure and enable the DMA interrupt for the DMA_A and DMA_B channels in the DMA controller
8. Start decryption by writing the block count *N* to BLKCNTx in the AESACTL1 register
9. Wait for the DMA_B channel interrupt which indicates completion of the operation. The plaintext output will be stored in the location configured in step 5c.

#### 12.2.7.3 Output Feedback (OFB) Mode

The output feedback mode leverages an initialization vector (IV) to generate a keystream by repeatedly encrypting the IV with the cipher key. The output ciphertext is obtained by XORing plaintext with the encrypted and re-encrypted versions of the initialization vector. The OFB cipher is shown in Figure 12-7.

In OFB mode, the initialization vector must be a nonce (number used once). To prevent loss of confidentiality, each IV must only be used one time with a given key, and any value passed into the cipher $E_k$ for a given key $k$ must not be used as an initialization vector with the same key $k$.

**Note**

As OFB is a stream cipher, the AES block function is used in the forward (encryption) mode when performing OFB encryption or OFB decryption. When decrypting, the keystream is simply regenerated again to be XORed with the ciphertext, giving back the plaintext.



Copyright © 2026 Texas Instruments Incorporated

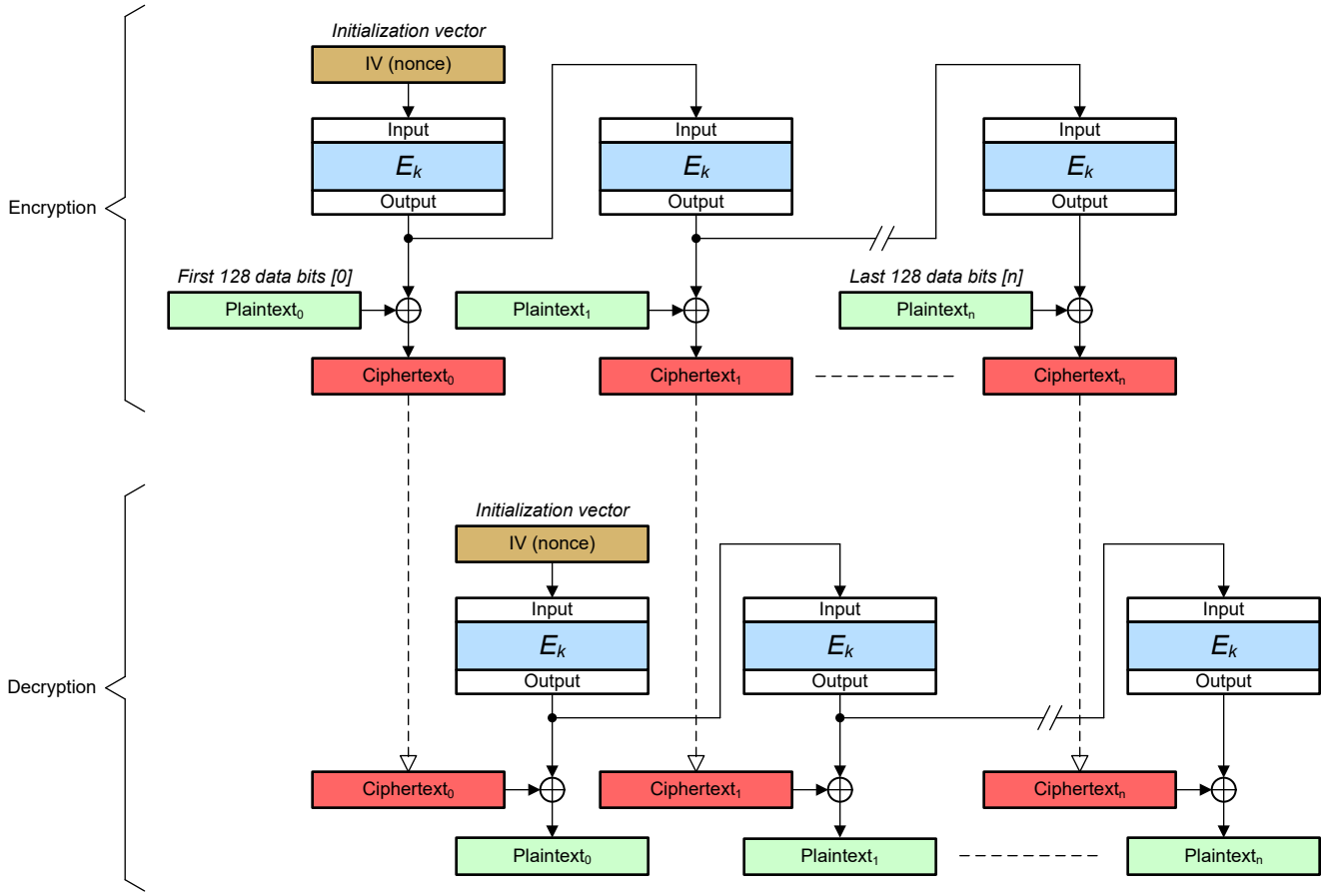



**Figure 12-7. OFB Cipher**

The AES accelerator supports automated encryption and decryption of more than 128 bits of data in OFB block cipher mode through the use of DMA together with the AES accelerator. OFB mode utilizes three DMA channels (referred to as DMA_A, DMA_B, and DMA_C in this section) for input, output, and input to the AES accelerator, respectively. The DMA channel usage for OFB mode encryption and decryption operations is given in Table 12-5.

**Table 12-5. OFB Cipher DMA Behavior**

| AES Operation | | | AES DMA Triggers | | |
|---|---|---|---|---|---|
| **CMEN** | **CMx** | **OPx** | **DMA_A (DMA_TRIG0)** | **DMA_B (DMA_TRIG1)** | **DMA_C (DMA_TRIG2DMA_TRIG)** |
| 0x1 | 0x2 (OFB) | 0x0 (encryption) | Used to write plaintext of the current block to AESAXIN | Used to read ciphertext from AESADOUT | Used to write the plaintext of the current block to AESAXDIN, which also triggers the next encryption |
| | | 0x1 (decryption) | Used to write the ciphertext of the current block to AESAXIN | Used to read the plaintext from AESADOUT | Used to write the ciphertext of the current block to AESAXDIN, which also triggers the next decryption |

#### *12.2.7.3.1 OFB Encryption*

OFB mode encryption of *N* blocks of plaintext into *N* blocks of ciphertext without CPU interaction is achieved through the use of 3 DMA channels (referred to as DMA_A, DMA_B, and DMA_C). To implement OFB encryption follow these steps:

1. Configure the AESACTL0 register for block cipher encryption mode using OFB:



Copyright © 2026 Texas Instruments Incorporated

a. Reset AES module to clear the internal state memory (set SWRST in the AESACTL0 register)
b. Set CMEN to enable block cipher mode
c. Set CMx to OFB mode (0x2)
d. Set OPx to encryption mode (0x0)

2. Load key as described in Section 12.2.2
3. Load initialization vector (IV) into AESAXIN
   a. Loading to AESAXIN does not start any encryption
   b. The STATE must be reset (step 1a above) so that the IV is XORed with zeros when loaded
4. Configure the DMA_A channel for loading plaintext:
   a. Set the DMA channel trigger selection to AES0 trigger
   b. Set the DMA channel source address to the location where plaintext is stored (for example, SRAM)
   c. Set the DMA channel destination address to AESAXIN
   d. Set the DMA channel transfer size to *N**4 words
   e. Set the DMA channel mode to single transfer mode
   f. In the AES event registers, unmask DMA0 in the IMASK register of DMA_TRIG

      0
5. Configure the DMA_B channel for reading ciphertext:
   a. Set the DMA channel trigger selection for AES1 trigger
   b. Set the DMA channel source address to AESADOUT
   c. Set the DMA channel destination address to the location where ciphertext is to be stored (for example, SRAM)
   d. Set the DMA channel transfer size to *N**4 words
   e. Set the DMA channel mode to single transfer mode
   f. In the AES event registers, unmask DMA1 in the IMASK register of DMA_TRIG

      1
6. Configure the DMA_C channel for loading plaintext:
   a. Set the DMA channel trigger selection for AES2 trigger
   b. Set the DMA channel source address to the location where plaintext is stored (for example, SRAM)
   c. Set the DMA channel destination address to AESAXDIN
   d. Set the DMA channel transfer size to *N**4 words
   e. Set the DMA channel mode to single transfer mode
7. Configure and enable the DMA interrupt for the DMA_B channel in the DMA controller
8. Start encryption by writing the block count *N* to BLKCNTx in the AESACTL1 register and setting the DINWR bit in the AESASTAT register
9. Wait for the DMA_B channel interrupt which indicates completion of the operation. The ciphertext output will be stored in the location configured in step 5c.

#### *12.2.7.3.2 OFB Decryption*

OFB mode decryption of *N* blocks of ciphertext into *N* blocks of ciphertext without CPU interaction is achieved through the use of 3 DMA channels (referred to as DMA_A, DMA_B, and DMA_C). To implement OFB decryption follow these steps:

1. Configure the AESACTL0 register for block cipher decryption mode using OFB:
   a. Reset the AES module to clear the internal state memory (set SWRST in the AESCTL0 register)
   b. Set the CMEN to enable block cipher mode
   c. Set CMx to OFB mode (0x2)
   d. Set OPx to decryption mode (0x1)
2. Load key as described in Section 12.2.2
3. Load initialization vector IV into AESAXIN
   a. Loading to AESAXIN does not start any decryption



Copyright © 2026 Texas Instruments Incorporated

b. The STATE must be reset (step 1a above) so that the IV is XORed with zeros when loaded

4. Configure the DMA_A channel for loading ciphertext:
   a. Set the DMA channel trigger selection to AES0 trigger
   b. Set the DMA channel source address to the location where ciphertext is stored (for example, SRAM)
   c. Set the DMA channel destination address to AESAXIN
   d. Set the DMA channel transfer size to *N**4 words
   e. Set the DMA channel mode to single transfer mode
5. Configure the DMA_B channel for reading plaintext:
   a. Set the DMA channel trigger selection for AES1 trigger
   b. Set the DMA channel source address to AESADOUT
   c. Set the DMA channel destination address to the location where plaintext is to be stored (for example, SRAM)
   d. Set the DMA channel transfer size to *N**4 words
   e. Set the DMA channel mode to single transfer mode
   f. In the AES event registers, unmask DMA1 in the IMASK register of DMA_TRIG

      1
6. Configure the DMA_C channel for loading ciphertext:
   a. Set the DMA channel trigger selection for AES2 trigger
   b. Set the DMA channel source address to the location where ciphertext is stored (for example, SRAM)
   c. Set the DMA channel destination address to AESAXDIN
   d. Set the DMA channel transfer size to *N**4 words
   e. Set the DMA channel mode to single transfer mode
7. Configure and enable the DMA interrupt for the DMA_B channel in the DMA controller
8. Start decryption by writing the block count *N* to BLKCNTx in the AESACTL1 register and setting the DINWR bit in the AESASTAT register
9. Wait for the DMA_B channel interrupt which indicates completion of the operation. The plaintext output will be stored in the location configured in step 5c.

#### 12.2.7.4 Cipher Feedback (CFB) Mode

The cipher feedback (CFB) mode is similar to the output feedback (OFB) mode, with the key difference being that the input block to the block cipher *E* used to generate the key stream is taken from the previous ciphertext block (after being XORed with the plaintext), vs. taken before being XORed with the plaintext (as in the case of OFB). As a result, the keystream is dependent upon the plaintext, which is not true of OFB. The CFB cipher is shown in Figure 12-8.

Like OFB, CFB requires an initialization vector (IV). In CFB mode, the initialization vector (IV) must be unpredictable.

**Note**

As CFB is a stream cipher, the AES block function is used in the forward (encryption) mode when performing CFB encryption or CFB decryption.



Copyright © 2026 Texas Instruments Incorporated

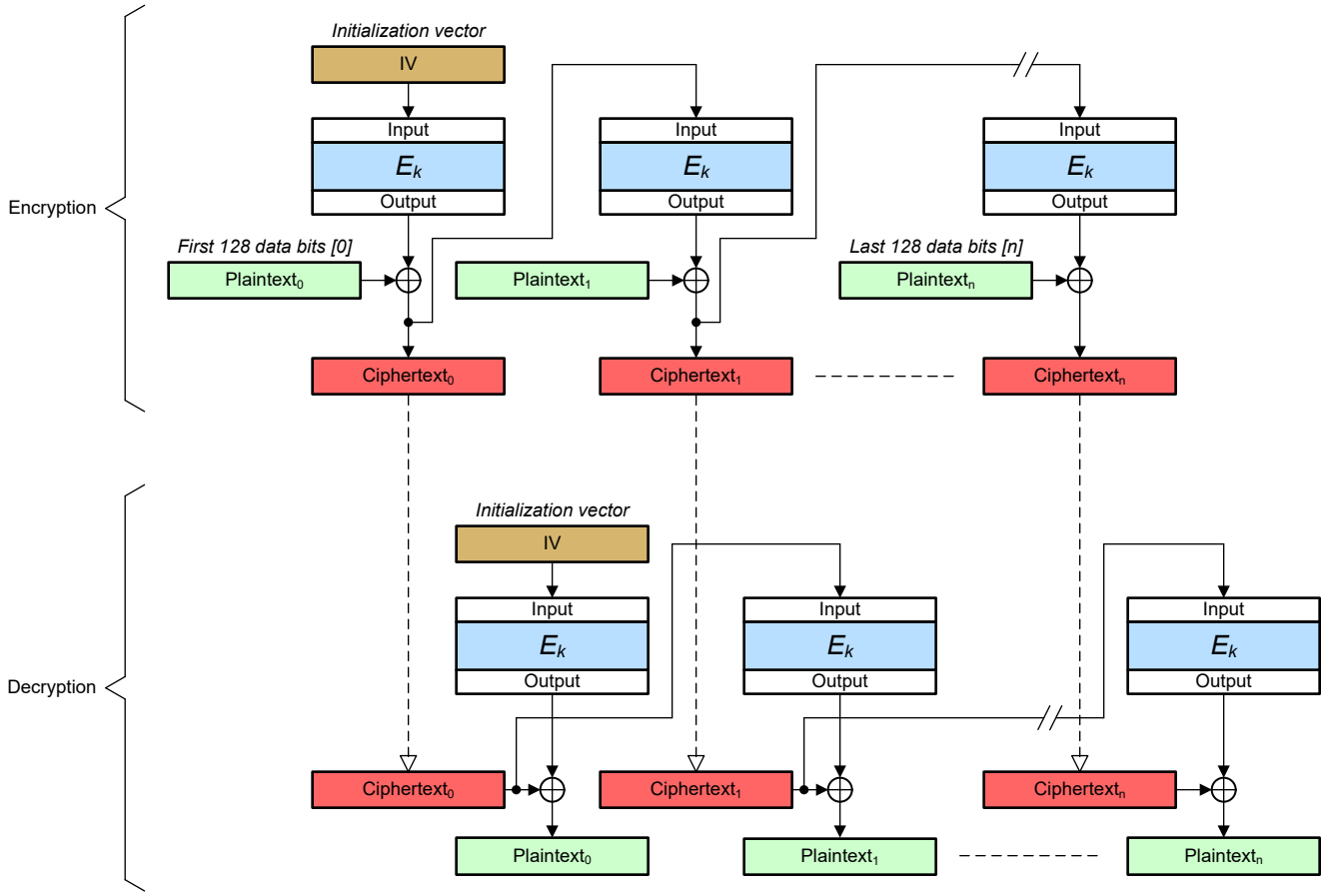



**Figure 12-8. CFB Cipher**

The AES accelerator supports automated encryption and decryption of more than 128 bits of data in CFB block cipher mode through the use of DMA together with the AES accelerator. CFB mode utilizes three DMA channels (referred to as DMA_A, DMA_B, and DMA_C in this section) for input, output, and input to the AES accelerator, respectively. The DMA channel usage for CFB mode encryption and decryption operations is given in Table 12-6.

**Table 12-6. CFB Cipher DMA Behavior**

| AES Operation | | | AES DMA Triggers | | |
|---|---|---|---|---|---|
| **CMEN** | **CMx** | **OPx** | **DMA_A (DMA_TRIG0)** | **DMA_B (DMA_TRIG1)** | **DMA_C (DMA_TRIG2)** |
| 0x1 | 0x3 (CFB) | 0x0 (encryption) | Used to write plaintext of the current block to AESAXIN | Used to read ciphertext from AESADOUT, which also triggers the next encryption | Not used |
| | | 0x1 (decryption) | Used to write the ciphertext of the current block to AESAXIN | Used to read the plaintext from AESADOUT | Used to write the ciphertext of the current block to AESADIN, which also triggers the next decryption |

#### *12.2.7.4.1 CFB Encryption*

CFB mode encryption of *N* blocks of plaintext into *N* blocks of ciphertext without CPU interaction is achieved through the use of two DMA channels (referred to as DMA_A, DMA_B). To implement CFB encryption follow these steps:

1. Configure the AESACTL0 register for block cipher encryption mode using OFB:
   a. Reset AES module to clear the internal state memory (set SWRST in the AESACTL0 register)
   b. Set CMEN to enable block cipher mode



Copyright © 2026 Texas Instruments Incorporated

c. Set CMx to CFB mode (0x3)
d. Set OPx to encryption mode (0x0)

2. Load key as described in Section 12.2.2
3. Load initialization vector (IV) into AESAXIN
   a. Loading to AESAXIN does not start any encryption
   b. The STATE must be reset (step 1a above) so that the IV is XORed with zeros when loaded
4. Configure the DMA_A channel for loading plaintext:
   a. Set the DMA channel trigger selection to AES0 trigger
   b. Set the DMA channel source address to the location where plaintext is stored (for example, SRAM)
   c. Set the DMA channel destination address to AESAXIN
   d. Set the DMA channel transfer size to *N**4 words
   e. Set the DMA channel mode to single transfer mode
5. Configure the DMA_B channel for reading ciphertext:
   a. Set the DMA channel trigger selection for AES1 trigger
   b. Set the DMA channel source address to AESADOUT
   c. Set the DMA channel destination address to the location where ciphertext is to be stored (for example, SRAM)
   d. Set the DMA channel transfer size to *N**4 words
   e. Set the DMA channel mode to single transfer mode
   f. In the AES event registers, unmask DMA1 in the IMASK register of DMA_TRIG

      1
6. Configure and enable the DMA interrupt for the DMA_B channel in the DMA controller
7. Start encryption by writing the block count *N* to BLKCNTx in the AESACTL1 register and setting the DINWR bit in the AESASTAT register
8. Wait for the DMA_B channel interrupt which indicates completion of the operation. The ciphertext output will be stored in the location configured in step 5c.

#### *12.2.7.4.2 CFB Decryption*

CFB mode decryption of *N* blocks of ciphertext into *N* blocks of ciphertext without CPU interaction is achieved through the use of 3 DMA channels (referred to as DMA_A, DMA_B, and DMA_C). To implement CFB decryption follow these steps:

1. Configure the AESACTL0 register for block cipher decryption mode using OFB:
   a. Reset the AES module to clear the internal state memory (set SWRST in the AESCTL0 register)
   b. Set the CMEN to enable block cipher mode
   c. Set CMx to CFB mode (0x3)
   d. Set OPx to decryption mode (0x1)
2. Load key as described in Section 12.2.2
3. Load initialization vector IV into AESAXIN
   a. Loading to AESAXIN does not start any decryption
   b. The STATE must be reset (step 1a above) so that the IV is XORed with zeros when loaded
4. Configure the DMA_A channel for loading ciphertext:
   a. Set the DMA channel trigger selection to AES0 trigger
   b. Set the DMA channel source address to the location where ciphertext is stored (for example, SRAM)
   c. Set the DMA channel destination address to AESAXIN
   d. Set the DMA channel transfer size to *N**4 words
   e. Set the DMA channel mode to single transfer mode
5. Configure the DMA_B channel for reading plaintext:
   a. Set the DMA channel trigger selection for AES1 trigger
   b. Set the DMA channel source address to AESADOUT



Copyright © 2026 Texas Instruments Incorporated

c. Set the DMA channel destination address to the location where plaintext is to be stored (for example, SRAM)
d. Set the DMA channel transfer size to *N**4 words
e. Set the DMA channel mode to single transfer mode
f. In the AES event registers, unmask DMA1 in the IMASK register of DMA_TRIG

1

6. Configure the DMA_C channel for loading ciphertext:
   a. Set the DMA channel trigger selection for AES2 trigger
   b. Set the DMA channel source address to the location where ciphertext is stored (for example, SRAM)
   c. Set the DMA channel destination address to AESADIN
   d. Set the DMA channel transfer size to *N**4 words
   e. Set the DMA channel mode to single transfer mode
7. Configure and enable the DMA interrupt for the DMA_B channel in the DMA controller
8. Start decryption by writing the block count *N* to BLKCNTx in the AESACTL1 register and setting the DINWR bit in the AESASTAT register
9. Wait for the DMA_B channel interrupt which indicates completion of the operation. The plaintext output will be stored in the location configured in step 5c.

#### 12.2.7.5 Counter (CTR) Mode

The counter mode leverages a nonce (number used once) and a counting integer to generate a keystream by appending the counter to the nonce and encrypting the combined nonce || counter value with the cipher key.

The nonce must only be used once with a given key *k*. The counter value can start from any value and is incremented for each 128-bit block of data.

The keystream is derived be encrypting the nonce || counter value for each 128-bit data block with the cipher key *k*. The output ciphertext is then obtained by XORing the plaintext with the encrypted nonce || counter value for each data block. The CTR cipher is shown in Figure 12-9.

**Note**

As CTR mode is a stream cipher, the AES block function is used in the forward (encryption) mode when performing CTR encryption or CTR decryption. When decrypting, the keystream is simply regenerated again to be XORed with the ciphertext, giving back the plaintext.



Copyright © 2026 Texas Instruments Incorporated

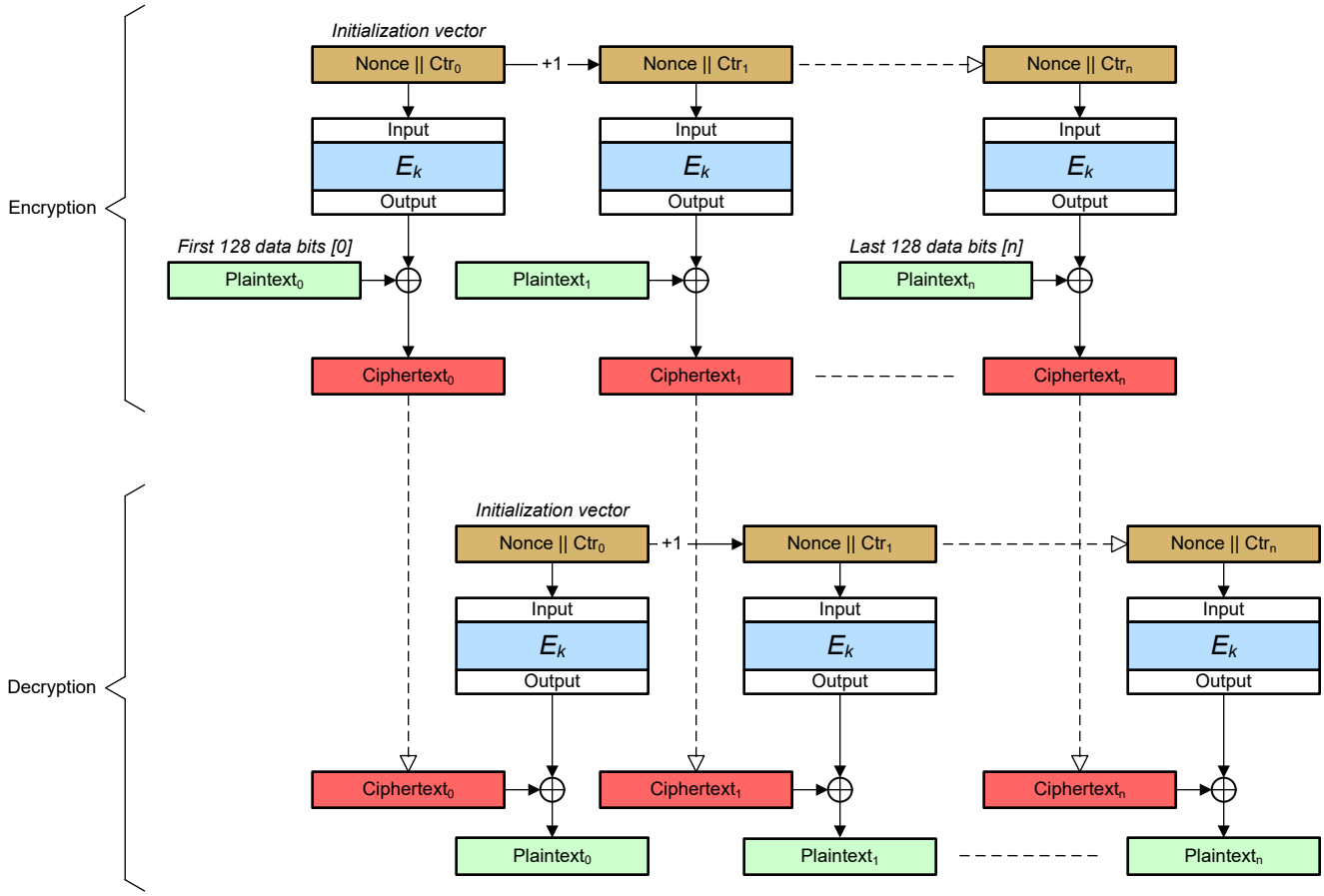



**Figure 12-9. CTR Cipher**

The AES accelerator can be used to accelerate CTR ciphers by accelerating the encryption of each nonce || counter value with the cipher key *k*, and accelerating the XOR operation of the plaintext (in the case of encryption) or ciphertext (in the case of decryption) with the keystream. DMA trigger generation is not supported when implementing a CTR cipher.

#### *12.2.7.5.1 CTR Encryption*

CTR-mode encryption of *N* blocks of plaintext into *N* blocks of ciphertext is achieved by following the steps below:

1. Configure the AESACTL0 register for single block encryption mode:
   a. Clear CMEN to disable block cipher mode (if previously enabled)
   b. Set OPx to encryption mode (0x0)
2. Load key as described in Section 12.2.2
3. Load first (nonce || counter) value into AESADIN as described in Section 12.2.3
4. Wait for the encryption process to complete (wait for the BUSY flag to clear)
5. The encrypted (nonce || counter) value is now in the STATE. The first ciphertext block can be obtained by XORing the first plaintext block with the keystream in the STATE.
   a. Load the first plaintext block into AESAXIN (XORing the plaintext with the keystream, but not starting any encryption operation)
6. Read the first block of ciphertext from AESADOUT as described in Section 12.2.4
7. Repeat steps 3-6 for additional data blocks, incrementing the counter value each time.



Copyright © 2026 Texas Instruments Incorporated

#### 12.2.7.5.2 CTR Decryption

CTR-mode decryption of *N* blocks of ciphertext into *N* blocks of plaintext is achieved by following the same steps given for encryption (Section 12.2.7.5.1), but substituting the ciphertext for the plaintext and the reverse. Note that only the forward cipher is used in CTR mode; the encryption and decryption operations are identical and the same software implementation can be used for both, with substitution of ciphertext for plaintext in the case of decryption.

### 12.2.8 AES Events

The AES module contains four event publishers and no event subscribers. One event publisher (CPU_INT) manages AES interrupt requests (IRQs) to the CPU subsystem through a static event route. The second, third, and fourth event publishers (DMA_TRIG0, DMA_TRIG1, and DMA_TRIG2) can be used to publish AES events to the DMA through DMA event routes.

The AES events are summarized in Table 12-7.

**Table 12-7. AES Events**

| Event | Type | Source | Destination | Route | Configuration | Functionality |
|---|---|---|---|---|---|---|
| CPU Interrupt Event | Publisher | AES | CPU Subsystem | Static route | CPU_INT registers | Fixed interrupt route from RTC to CPU |
| DMA Trigger Event 0 | Publisher | AES | DMA | DMA route | DMA_TRIG 0 registers | DMA trigger 0 for block cipher modes |
| DMA Trigger Event 1 | Publisher | AES | DMA | DMA route | DMA_TRIG 1 registers | DMA trigger 1 for block cipher modes |
| DMA Trigger Event 2 | Publisher | AES | DMA | DMA route | DMA_TRIG 2 registers | DMA trigger 2 for block cipher modes |

In general, the CPU interrupt event is used to communicate completion of an AES operation to the CPU, and the DMA triggers are used together to implement the block cipher modes (ECB, CBC, OFB, and CFB) using the DMA together with the AES accelerator.

When the CMEN bit is set in AESACTL0, the AES module will trigger DMA trigger event 0, DMA trigger event 1, and DMA trigger event 2 to execute different block cipher modes of operation together with the DMA.

For example, when using ECB encryption with AESCMEN=0x1, DMA trigger event 0 is triggered four times for DMA word access to read out AESADOUT, and AES trigger 1 is triggered 4 times to fill the next data into AESADIN. The AES module generates a trigger for each word, and so the DMA must be configured in repeated single channel mode. See the block cipher mode section for details on how the DMA must be configured to support the ECB, CBC, OFB, and CFB block cipher modes.

#### 12.2.8.1 CPU Interrupt Event Publisher (CPU_EVENT)

The AES module provides four interrupt sources which can be configured to source a CPU interrupt event. In order of decreasing interrupt priority, the CPU interrupt events from the AES are given inTable 12-8 .

**Table 12-8. AES CPU Interrupt Event Conditions (CPU_EVENT)**

| Index (IIDX) | Name | Description |
|---|---|---|
| 0 | AESRDY | Indicates that the AES module has completed the selected operation and the results can be read from AESADOUT (if applicable). |
| 1 | DMA0 | Not to be used for normal operation. |
| 2 | DMA1 | Not to be used for normal operation. |
| 3 | DMA2 | Not to be used for normal operation. |



Copyright © 2026 Texas Instruments Incorporated

The CPU interrupt event configuration is managed with the event management registers. See Section 7.2.5 for guidance on configuring these registers for CPU interrupts.

#### 12.2.8.2 DMA Trigger Event Publisher (DMA_TRIG0)

The AES module provides four trigger sources which can be configured to source DMA trigger 0. In order of decreasing interrupt priority, the DMA0 trigger events from the AES are given in Table 12-9. When the DMA0 channel is needed by the AES for block cipher operations, the DMA0 trigger should be unmasked in the IMASK register of DMA_TRIG0 and the DMA should be configured as needed to support the AES operation.

**Table 12-9. AES DMA Trigger 0 Event Conditions (DMA_TRIG0)**

| Index (IIDX) | Name | Description |
|---|---|---|
| 0 | AESRDY | Not to be used for normal operation. |
| 1 | DMA0 | DMA0 trigger indication |
| 2 | DMA1 | Not to be used for normal operation. |
| 3 | DMA2 | Not to be used for normal operation. |

The DMA trigger 0 event configuration is managed with the DMA_TRIG0 event management registers. See Section 7.2.5 for guidance on configuring the event registers for DMA triggers.

#### 12.2.8.3 DMA Trigger Event Publisher (DMA_TRIG1)

The AES module provides four trigger sources which can be configured to source DMA trigger 1. In order of decreasing interrupt priority, the DMA1 trigger events from the AES are given in Table 12-10. When the DMA1 channel is needed by the AES for block cipher operations, the DMA1 trigger should be unmasked in the IMASK register of DMA_TRIG1 and the DMA should be configured as needed to support the AES operation.

**Table 12-10. AES DMA Trigger 1 Event Conditions (DMA_TRIG1)**

| Index (IIDX) | Name | Description |
|---|---|---|
| 0 | AESRDY | Not to be used for normal operation. |
| 1 | DMA0 | Not to be used for normal operation. |
| 2 | DMA1 | DMA1 trigger indication |
| 3 | DMA2 | Not to be used for normal operation. |

The DMA trigger 1 event configuration is managed with the DMA_TRIG1 event management registers. See Section 7.2.5 for guidance on configuring the event registers for DMA triggers.

#### 12.2.8.4 DMA Trigger Event Publisher (DMA_TRIG2)

The AES module provides four trigger sources which can be configured to source DMA trigger 2. In order of decreasing interrupt priority, the DMA2 trigger events from the AES are given in Table 12-11. When the DMA2 channel is needed by the AES for block cipher operations, the DMA2 trigger should be unmasked in the IMASK register of DMA_TRIG2 and the DMA should be configured as needed to support the AES operation.

**Table 12-11. AES DMA Trigger 2 Event Conditions (DMA_TRIG2)**

| Index (IIDX) | Name | Description |
|---|---|---|
| 0 | AESRDY | Not to be used for normal operation. |
| 1 | DMA0 | Not to be used for normal operation. |
| 2 | DMA1 | Not to be used for normal operation. |
| 3 | DMA2 | DMA2 trigger indication |



Copyright © 2026 Texas Instruments Incorporated

The DMA trigger 2 event configuration is managed with the DMA_TRIG2 event management registers. See Section 7.2.5 for guidance on configuring the event registers for DMA triggers.



Copyright © 2026 Texas Instruments Incorporated

## 12.3 AES Registers

Table 12-12 lists the memory-mapped registers for the AES registers. All register offset addresses not listed in Table 12-12 should be considered as reserved locations and the register contents should not be modified.

**Table 12-12. AES Registers**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 800h | PWREN | Power enable | | Go |
| 804h | RSTCTL | Reset Control | | Go |
| 814h | STAT | Status Register | | Go |
| 1018h | PDBGCTL | Peripheral Debug Control | | Go |
| 1020h | IIDX | Interrupt Index Register | CPU_INT | Go |
| 1028h | IMASK | Interrupt mask | CPU_INT | Go |
| 1030h | RIS | Raw interrupt status | CPU_INT | Go |
| 1038h | MIS | Masked interrupt status | CPU_INT | Go |
| 1040h | ISET | Interrupt set | CPU_INT | Go |
| 1048h | ICLR | Interrupt clear | CPU_INT | Go |
| 1050h | IIDX | Interrupt Index Register | DMA_TRIG0 | Go |
| 1058h | IMASK | Interrupt mask | DMA_TRIG0 | Go |
| 1060h | RIS | Raw interrupt status | DMA_TRIG0 | Go |
| 1068h | MIS | Masked interrupt status | DMA_TRIG0 | Go |
| 1070h | ISET | Interrupt set | DMA_TRIG0 | Go |
| 1078h | ICLR | Interrupt clear | DMA_TRIG0 | Go |
| 1080h | IIDX | Interrupt Index Register | DMA_TRIG1 | Go |
| 1088h | IMASK | Interrupt mask | DMA_TRIG1 | Go |
| 1090h | RIS | Raw interrupt status | DMA_TRIG1 | Go |
| 1098h | MIS | Masked interrupt status | DMA_TRIG1 | Go |
| 10A0h | ISET | Interrupt set | DMA_TRIG1 | Go |
| 10A8h | ICLR | Interrupt clear | DMA_TRIG1 | Go |
| 10B0h | IIDX | Interrupt Index Register | DMA_TRIG2 | Go |
| 10B8h | IMASK | Interrupt mask | DMA_TRIG2 | Go |
| 10C0h | RIS | Raw interrupt status | DMA_TRIG2 | Go |
| 10C8h | MIS | Masked interrupt status | DMA_TRIG2 | Go |
| 10D0h | ISET | Interrupt set | DMA_TRIG2 | Go |
| 10D8h | ICLR | Interrupt clear | DMA_TRIG2 | Go |
| 10E0h | EVT_MODE | Event Mode | | Go |
| 1100h | AESACTL0 | AES accelerator control register 0 | | Go |
| 1104h | AESACTL1 | AES accelerator control register 1 | | Go |
| 1108h | AESASTAT | AES accelerator status register | | Go |
| 110Ch | AESAKEY | AES accelerator key register | | Go |
| 1110h | AESADIN | AES accelerator data in register | | Go |
| 1114h | AESADOUT | AES accelerator data out register | | Go |
| 1118h | AESAXDIN | AES accelerator XORed data in register | | Go |
| 111Ch | AESAXIN | AES accelerator XORed data in register (no trigger) | | Go |

Complex bit access types are encoded to fit into small table cells. Table 12-13 shows the codes that are used for access types in this section.



Copyright © 2026 Texas Instruments Incorporated

**Table 12-13. AES Access Type Codes**

| Access Type | Code | Description |
|---|---|---|
| **Read Type** | | |
| R | R | Read |
| **Write Type** | | |
| W | W | Write |
| WK | W<br>K | Write<br>Write protected by a key |
| **Reset or Default Value** | | |
| *-n* | | Value after reset or the default value |



Copyright © 2026 Texas Instruments Incorporated

### 12.3.1 PWREN (Offset = 800h) [Reset = 00000000h]

PWREN is shown in Figure 12-10 and described in Table 12-14.

Return to the Summary Table.

Register to control the power state

**Figure 12-10. PWREN**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | ENABLE |
| R/W-0h | | | | | | | R/WK-0h |

**Table 12-14. PWREN Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | KEY to allow Power State Change<br>26h = KEY to allow write access to this register |
| 23-1 | RESERVED | R/W | 0h | |
| 0 | ENABLE | R/WK | 0h | Enable the power<br>KEY must be set to 26h to write to this bit.<br>0h = Disable Power<br>1h = Enable Power |



Copyright © 2026 Texas Instruments Incorporated

### 12.3.2 RSTCTL (Offset = 804h) [Reset = 00000000h]

RSTCTL is shown in Figure 12-11 and described in Table 12-15.

Return to the Summary Table.

Register to control reset assertion and de-assertion

**Figure 12-11. RSTCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | RESETSTKYCLR | RESETASSERT |
| W-0h | | | | | | WK-0h | WK-0h |

**Table 12-15. RSTCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | Unlock key<br>B1h = KEY to allow write access to this register |
| 23-2 | RESERVED | W | 0h | |
| 1 | RESETSTKYCLR | WK | 0h | Clear the RESETSTKY bit in the STAT register<br>KEY must be set to B1h to write to this bit.<br>0h = Writing 0 has no effect<br>1h = Clear reset sticky bit |
| 0 | RESETASSERT | WK | 0h | Assert reset to the peripheral<br>KEY must be set to B1h to write to this bit.<br>0h = Writing 0 has no effect<br>1h = Assert reset |



Copyright © 2026 Texas Instruments Incorporated

### 12.3.3 STAT (Offset = 814h) [Reset = 00000000h]

STAT is shown in Figure 12-12 and described in Table 12-16.

Return to the Summary Table.

peripheral enable and reset status

**Figure 12-12. STAT**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R- | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | RESETSTKY |
| R- | | | | | | | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R- | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | |
| R- | | | | | | | |

**Table 12-16. STAT Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-17 | RESERVED | R | 0h | |
| 16 | RESETSTKY | R | 0h | This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register<br>0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register<br>1h = The peripheral was reset since the last bit clear |
| 15-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 12.3.4 PDBGCTL (Offset = 1018h) [Reset = 00000003h]

PDBGCTL is shown in Figure 12-13 and described in Table 12-17.

Return to the Summary Table.

This register can be used by the software developer to control the behavior of the peripheral relative to the 'Core Halted' input

**Figure 12-13. PDBGCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R/W- | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R/W- | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R/W- | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | SOFT | FREE |
| R/W- | | | | | | R/W-1h | R/W-1h |

**Table 12-17. PDBGCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | R/W | 0h | |
| 1 | SOFT | R/W | 1h | Soft halt boundary control. This function is only available, if FREE is set to 'STOP'<br>0h = The peripheral will halt immediately, even if the resultant state will result in corruption if the system is restarted<br>1h = The peripheral blocks the debug freeze until it has reached a boundary where it can resume without corruption |
| 0 | FREE | R/W | 1h | Free run control<br>0h = The peripheral freezes functionality while the Core Halted input is asserted and resumes when it is deasserted.<br>1h = The peripheral ignores the state of the Core Halted input |



Copyright © 2026 Texas Instruments Incorporated

### 12.3.5 IIDX (Offset = 1020h) [Reset = 00000000h]

IIDX is shown in Figure 12-14 and described in Table 12-18.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. Value 0x00 means no event pending. Interrupt 1 is the highest priority, IIDX next highest, 4, 8, … IIDX^31 is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred. On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in [RIS] and [MIS] are cleared as well. After a read from the CPU (not from the debug interface), the register is updated with the next highest priority interrupt, if none are pending, then it should display 0x0.

**Figure 12-14. IIDX**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | STAT | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | |

**Table 12-18. IIDX Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | STAT | R | 0h | Interrupt index status<br>00h = No interrupt pending<br>1h = AES ready interrupt, set when the selected AES operation was completed and the result can be read from AESADOUT |



Copyright © 2026 Texas Instruments Incorporated

### 12.3.6 IMASK (Offset = 1028h) [Reset = 00000000h]

IMASK is shown in Figure 12-15 and described in Table 12-19.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is unmasked. Unmasking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

**Figure 12-15. IMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | AESRDY |
| R/W-0h | | | | | | | R/W-0h |

**Table 12-19. IMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R/W | 0h | |
| 0 | AESRDY | R/W | 0h | AES ready interrupt, set when the selected AES operation was completed and the result can be read from AESADOUT.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |



Copyright © 2026 Texas Instruments Incorporated

### 12.3.7 RIS (Offset = 1030h) [Reset = 00000000h]

RIS is shown in Figure 12-16 and described in Table 12-20.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Figure 12-16. RIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | AESRDY |
| R-0h | | | | | | | R-0h |

**Table 12-20. RIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | AESRDY | R | 0h | AES ready interrupt, set when the selected AES operation was completed and the result can be read from AESADOUT.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

### 12.3.8 MIS (Offset = 1038h) [Reset = 00000000h]

MIS is shown in Figure 12-17 and described in Table 12-21.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 12-17. MIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | AESRDY |
| R-0h | | | | | | | R-0h |

**Table 12-21. MIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | AESRDY | R | 0h | AES ready interrupt, set when the selected AES operation was completed and the result can be read from AESADOUT.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

### 12.3.9 ISET (Offset = 1040h) [Reset = 00000000h]

ISET is shown in Figure 12-18 and described in Table 12-22.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 12-18. ISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | AESRDY |
| W-0h | | | | | | | W-0h |

**Table 12-22. ISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | W | 0h | |
| 0 | AESRDY | W | 0h | AES ready interrupt, set when the selected AES operation was completed and the result can be read from AESADOUT.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 12.3.10 ICLR (Offset = 1048h) [Reset = 00000000h]

ICLR is shown in Figure 12-19 and described in Table 12-23.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

**Figure 12-19. ICLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | AESRDY |
| W-0h | | | | | | | W-0h |

**Table 12-23. ICLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | W | 0h | |
| 0 | AESRDY | W | 0h | AES ready interrupt, set when the selected AES operation was completed and the result can be read from AESADOUT.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 12.3.11 IIDX (Offset = 1050h) [Reset = 00000000h]

IIDX is shown in Figure 12-20 and described in Table 12-24.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. Value 0x00 means no event pending. Interrupt 1 is the highest priority, IIDX next highest, 4, 8, … IIDX^31 is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred. On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in [RIS] and [MIS] are cleared as well. After a read from the CPU (not from the debug interface), the register is updated with the next highest priority interrupt, if none are pending, then it should display 0x0.

**Figure 12-20. IIDX**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | STAT | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | |

**Table 12-24. IIDX Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | STAT | R | 0h | Interrupt index status<br>00h = No interrupt pending<br>2h = AES trigger 0 DMA |



Copyright © 2026 Texas Instruments Incorporated

### 12.3.12 IMASK (Offset = 1058h) [Reset = 00000000h]

IMASK is shown in Figure 12-21 and described in Table 12-25.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is unmasked. Unmasking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

**Figure 12-21. IMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | DMA0 | RESERVED |
| R/W-0h | | | | | | R/W-0h | R/W-0h |

**Table 12-25. IMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | R/W | 0h | |
| 1 | DMA0 | R/W | 0h | DMA0 event mask.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 0 | RESERVED | R/W | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 12.3.13 RIS (Offset = 1060h) [Reset = 00000000h]

RIS is shown in Figure 12-22 and described in Table 12-26.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Figure 12-22. RIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | DMA0 | RESERVED |
| R-0h | | | | | | R-0h | R-0h |

**Table 12-26. RIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | |
| 1 | DMA0 | R | 0h | DMA0 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 12.3.14 MIS (Offset = 1068h) [Reset = 00000000h]

MIS is shown in Figure 12-23 and described in Table 12-27.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 12-23. MIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | DMA0 | RESERVED |
| R-0h | | | | | | R-0h | R-0h |

**Table 12-27. MIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | |
| 1 | DMA0 | R | 0h | DMA0 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 12.3.15 ISET (Offset = 1070h) [Reset = 00000000h]

ISET is shown in Figure 12-24 and described in Table 12-28.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 12-24. ISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | DMA0 | RESERVED |
| W-0h | | | | | | W-0h | W-0h |

**Table 12-28. ISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | W | 0h | |
| 1 | DMA0 | W | 0h | DMA0<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 0 | RESERVED | W | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 12.3.16 ICLR (Offset = 1078h) [Reset = 00000000h]

ICLR is shown in Figure 12-25 and described in Table 12-29.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

**Figure 12-25. ICLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | DMA0 | RESERVED |
| W-0h | | | | | | W-0h | W-0h |

**Table 12-29. ICLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | W | 0h | |
| 1 | DMA0 | W | 0h | DMA0 event<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 0 | RESERVED | W | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 12.3.17 IIDX (Offset = 1080h) [Reset = 00000000h]

IIDX is shown in Figure 12-26 and described in Table 12-30.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. Value 0x00 means no event pending. Interrupt 1 is the highest priority, IIDX next highest, 4, 8, … IIDX^31 is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred. On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in [RIS] and [MIS] are cleared as well. After a read from the CPU (not from the debug interface), the register is updated with the next highest priority interrupt, if none are pending, then it should display 0x0.

**Figure 12-26. IIDX**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | STAT | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | |

**Table 12-30. IIDX Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | STAT | R | 0h | Interrupt index status<br>00h = No interrupt pending<br>3h = AES trigger 1 DMA |



Copyright © 2026 Texas Instruments Incorporated

### 12.3.18 IMASK (Offset = 1088h) [Reset = 00000000h]

IMASK is shown in Figure 12-27 and described in Table 12-31.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is unmasked. Unmasking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

**Figure 12-27. IMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | DMA1 | RESERVED | |
| R/W-0h | | | | | R/W-0h | R/W-0h | |

**Table 12-31. IMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | RESERVED | R/W | 0h | |
| 2 | DMA1 | R/W | 0h | DMA1 event mask.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 1-0 | RESERVED | R/W | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 12.3.19 RIS (Offset = 1090h) [Reset = 00000000h]

RIS is shown in Figure 12-28 and described in Table 12-32.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Figure 12-28. RIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | DMA1 | RESERVED | |
| R-0h | | | | | R-0h | R-0h | |

**Table 12-32. RIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | RESERVED | R | 0h | |
| 2 | DMA1 | R | 0h | DMA1 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 1-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 12.3.20 MIS (Offset = 1098h) [Reset = 00000000h]

MIS is shown in Figure 12-29 and described in Table 12-33.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 12-29. MIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | DMA1 | RESERVED | |
| R-0h | | | | | R-0h | R-0h | |

**Table 12-33. MIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | RESERVED | R | 0h | |
| 2 | DMA1 | R | 0h | DMA1 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 1-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 12.3.21 ISET (Offset = 10A0h) [Reset = 00000000h]

ISET is shown in Figure 12-30 and described in Table 12-34.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 12-30. ISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | DMA1 | RESERVED | |
| W-0h | | | | | W-0h | W-0h | |

**Table 12-34. ISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | RESERVED | W | 0h | |
| 2 | DMA1 | W | 0h | DMA1 event<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 1-0 | RESERVED | W | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 12.3.22 ICLR (Offset = 10A8h) [Reset = 00000000h]

ICLR is shown in Figure 12-31 and described in Table 12-35.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

**Figure 12-31. ICLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | DMA1 | RESERVED | |
| W-0h | | | | | W-0h | W-0h | |

**Table 12-35. ICLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | RESERVED | W | 0h | |
| 2 | DMA1 | W | 0h | DMA1 event<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 1-0 | RESERVED | W | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 12.3.23 IIDX (Offset = 10B0h) [Reset = 00000000h]

IIDX is shown in Figure 12-32 and described in Table 12-36.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. Value 0x00 means no event pending. Interrupt 1 is the highest priority, IIDX next highest, 4, 8, … IIDX^31 is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred. On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in [RIS] and [MIS] are cleared as well. After a read from the CPU (not from the debug interface), the register is updated with the next highest priority interrupt, if none are pending, then it should display 0x0.

**Figure 12-32. IIDX**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | STAT | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | |

**Table 12-36. IIDX Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | STAT | R | 0h | Interrupt index status<br>00h = No interrupt pending<br>4h = AES trigger 2 DMA |



Copyright © 2026 Texas Instruments Incorporated

### 12.3.24 IMASK (Offset = 10B8h) [Reset = 00000000h]

IMASK is shown in Figure 12-33 and described in Table 12-37.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is unmasked. Unmasking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

**Figure 12-33. IMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | DMA2 | RESERVED | | |
| R/W-0h | | | | R/W-0h | R/W-0h | | |

**Table 12-37. IMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R/W | 0h | |
| 3 | DMA2 | R/W | 0h | DMA2 event mask.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 2-0 | RESERVED | R/W | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 12.3.25 RIS (Offset = 10C0h) [Reset = 00000000h]

RIS is shown in Figure 12-34 and described in Table 12-38.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Figure 12-34. RIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | DMA2 | RESERVED | | |
| R-0h | | | | R-0h | R-0h | | |

**Table 12-38. RIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | DMA2 | R | 0h | DMA2 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 2-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 12.3.26 MIS (Offset = 10C8h) [Reset = 00000000h]

MIS is shown in Figure 12-35 and described in Table 12-39.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 12-35. MIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | DMA2 | RESERVED | | |
| R-0h | | | | R-0h | R-0h | | |

**Table 12-39. MIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | DMA2 | R | 0h | DMA2 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 2-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 12.3.27 ISET (Offset = 10D0h) [Reset = 00000000h]

ISET is shown in Figure 12-36 and described in Table 12-40.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 12-36. ISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | DMA2 | RESERVED | | |
| W-0h | | | | W-0h | W-0h | | |

**Table 12-40. ISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | W | 0h | |
| 3 | DMA2 | W | 0h | DMA2 event<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 2-0 | RESERVED | W | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 12.3.28 ICLR (Offset = 10D8h) [Reset = 00000000h]

ICLR is shown in Figure 12-37 and described in Table 12-41.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

**Figure 12-37. ICLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | DMA2 | RESERVED | | |
| W-0h | | | | W-0h | W-0h | | |

**Table 12-41. ICLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | W | 0h | |
| 3 | DMA2 | W | 0h | DMA2 event<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 2-0 | RESERVED | W | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 12.3.29 EVT_MODE (Offset = 10E0h) [Reset = 000000A9h]

EVT_MODE is shown in Figure 12-38 and described in Table 12-42.

Return to the Summary Table.

Event mode register. It is used to select whether each line is disabled, in software mode (software clears the RIS) or in hardware mode (hardware clears the RIS)

**Figure 12-38. EVT_MODE**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R/W- | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R/W- | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R/W- | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| EVT3_CFG | | EVT2_CFG | | EVT1_CFG | | INT0_CFG | |
| R-2h | | R-2h | | R-2h | | R-1h | |

**Table 12-42. EVT_MODE Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R/W | 0h | |
| 7-6 | EVT3_CFG | R | 2h | Event line mode select for event corresponding to none.DMA_TRIG2<br>0h = The interrupt or event line is disabled.<br>1h = The interrupt or event line is in software mode. Software must clear the RIS.<br>2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag. |
| 5-4 | EVT2_CFG | R | 2h | Event line mode select for event corresponding to none.DMA_TRIG1<br>0h = The interrupt or event line is disabled.<br>1h = The interrupt or event line is in software mode. Software must clear the RIS.<br>2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag. |
| 3-2 | EVT1_CFG | R | 2h | Event line mode select for event corresponding to none.DMA_TRIG0<br>0h = The interrupt or event line is disabled.<br>1h = The interrupt or event line is in software mode. Software must clear the RIS.<br>2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag. |
| 1-0 | INT0_CFG | R | 1h | Event line mode select for event corresponding to none.CPU_INT<br>0h = The interrupt or event line is disabled.<br>1h = The interrupt or event line is in software mode. Software must clear the RIS.<br>2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag. |



Copyright © 2026 Texas Instruments Incorporated

### 12.3.30 AESACTL0 (Offset = 1100h) [Reset = 00000000h]

AESACTL0 is shown in Figure 12-39 and described in Table 12-43.

Return to the Summary Table.

AES accelerator control register 0

**Figure 12-39. AESACTL0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CMEN | RESERVED | | | ERRFG | RESERVED | | |
| R/W-0h | R/W-0h | | | R/W-0h | R/W-0h | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SWRST | CMx | | RESERVED | KLx | | OPx | |
| R/W-0h | R/W-0h | | R/W-0h | R/W-0h | | R/W-0h | |

**Table 12-43. AESACTL0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R/W | 0h | |
| 15 | CMEN | R/W | 0h | AESCMEN enables the support of the cipher modes ECB, CBC, OFB and CFB together with the DMA. Writes are ignored when AESCMEN = 1 and AESBLKCNTx > 0. 0 = No DMA triggers are generated. 1 = DMA cipher mode support operation is enabled and the corresponding DMA triggers are generated.<br>0h = No DMA triggers are generated.<br>1h = DMA cipher mode support operation is enabled and the corresponding DMA triggers are generated. |
| 14-12 | RESERVED | R/W | 0h | |
| 11 | ERRFG | R/W | 0h | AES error flag. AESAKEY or AESADIN were written while an AES operation was in progress. The bit must be cleared by software. 0b = No error 1b = Error occurred<br>0h (R/W) = No error<br>1h (R/W) = Error occurred |
| 10-8 | RESERVED | R/W | 0h | |
| 7 | SWRST | R/W | 0h | AES software reset. Immediately resets the complete AES accelerator module even when busy except for the AESRDYIE, the AESKLx and the AESOPx bits. It also clears the (internal) state memory. The AESSWRST bit is automatically reset and is always read as zero. 0b = No reset 1b = Reset AES accelerator module<br>0h = No reset.<br>1h = Reset AES accelerator module. |
| 6-5 | CMx | R/W | 0h | AES cipher mode select. These bits are ignored for AESCMEN = 0. Writes are ignored when AESCMEN = 1 and AESBLKCNTx > 0. 00b = ECB 01b = CBC 10b = OFB 11b = CFB<br>0h = ECB<br>1h = CBC<br>2h = OFB<br>3h = CFB |
| 4 | RESERVED | R/W | 0h | |



Copyright © 2026 Texas Instruments Incorporated

**Table 12-43. AESACTL0 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 3-2 | KLx | R/W | 0h | AES key length.<br>These bits define which of the 1 AES standards is performed. The AESKLx bits are not reset by AESSWRST = 1. Writes are ignored when AESCMEN = 1 and AESBLKCNTx > 0.<br>0h = The key size is 128 bit.<br>2h = The key size is 256 bit. |
| 1-0 | OPx | R/W | 0h | AES operation. The AESOPx bits are not reset by AESSWRST = 1. Writes are ignored when AESCMEN = 1 and AESBLKCNTx > 0. 00b = Encryption. 01b = Decryption. The provided key is the same key used for encryption. 10b = Generate first round key required for decryption. 11b = Decryption. The provided key is the first round key required for decryption.<br>0h (R/W) = Encryption<br>1h (R/W) = Decryption. The provided key is the same key used for encryption.<br>2h (R/W) = Generate first round key required for decryption.<br>3h (R/W) = Decryption. The provided key is the first round key required for decryption. |



Copyright © 2026 Texas Instruments Incorporated

### 12.3.31 AESACTL1 (Offset = 1104h) [Reset = 00000000h]

AESACTL1 is shown in Figure 12-40 and described in Table 12-44.

Return to the Summary Table.

AES accelerator control register 1

**Figure 12-40. AESACTL1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | BLKCNTx | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | | | | | | | | | R/W-0h | | | | | | | |

**Table 12-44. AESACTL1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R/W | 0h | |
| 7-0 | BLKCNTx | R/W | 0h | Cipher Block Counter. Number of blocks to be encrypted or decrypted with block cipher modes enabled (AESCMEN = 1). Ignored if AESCMEN = 0. The block counter decrements with each performed encryption or decryption. Writes are ignored when AESCMEN = 1 and AESBLKCNTx > 0.<br>00h = Smallest value<br>FFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 12.3.32 AESASTAT (Offset = 1108h) [Reset = 00000000h]

AESASTAT is shown in Figure 12-41 and described in Table 12-45.

Return to the Summary Table.

AES accelerator Status Register.

**Figure 12-41. AESASTAT**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DOUTCNTx | | | | DINCNTx | | | |
| R-0h | | | | R-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| KEYCNTx | | | | DOUTRD | DINWR | KEYWR | BUSY |
| R-0h | | | | R-0h | R/W-0h | R/W-0h | R-0h |

**Table 12-45. AESASTAT Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15-12 | DOUTCNTx | R | 0h | Bytes read from AESADOUT. Reset when AESDOUTRD is reset. If AESDOUTCNTx = 0 and AESDOUTRD = 0, no bytes were read. If AESDOUTCNTx = 0 and AESDOUTRD = 1, all bytes were read.<br>0h = Smallest possible value<br>Fh = Highest possible value |
| 11-8 | DINCNTx | R | 0h | Bytes written to AESADIN, AESAXDIN or AESAXIN. Reset when AESDINWR is reset. If AESDINCNTx = 0 and AESDINWR = 0, no bytes were written. If AESDINCNTx = 0 and AESDINWR = 1, all bytes were written.<br>0h = Smallest possible value<br>Fh = Highest possible value |
| 7-4 | KEYCNTx | R | 0h | Bytes written to AESAKEY when AESKLx = 00, half-words written to AESAKEY if AESKLx = b10. Reset when AESKEYWR is reset. If AESKEYCNTx = 0 and AESKEYWR = 0, no bytes were written. If AESKEYCNTx = 0 and AESKEYWR = 1, all bytes were written.<br>0h = Smallest possible value<br>Fh = Highest possible value |
| 3 | DOUTRD | R | 0h | All 16 bytes read from AESADOUT. AESDOUTRD is reset by PUC, AESSWRST, an error condition, changing AESOPx, changing AESKLx, when the AES accelerator is busy, and when the output data is read again. 0 = Not all bytes read 1 = All bytes read<br>0h = Not all bytes read<br>1h = All bytes read |
| 2 | DINWR | R/W | 0h | All 16 bytes written to AESADIN, AESAXDIN or AESAXIN. Changing its state by software also resets the AESDINCNTx bits. AESDINWR is reset by PUC, AESSWRST, an error condition, changing AESOPx, changing AESKLx, the start to (over)write the data, and when the AES accelerator is busy. Because it is reset when AESOPx or AESKLx is changed it can be set by software again to indicate that the current data is still valid. 0 = Not all bytes written 1 = All bytes written<br>0h = Not all bytes written<br>1h = All bytes written |



Copyright © 2026 Texas Instruments Incorporated

**Table 12-45. AESASTAT Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 1 | KEYWR | R/W | 0h | All bytes written to AESAKEY. This bit can be modified by software but it must not be reset by software (1→0) if AESCMEN=1. Changing its state by software also resets the AESKEYCNTx bits. AESKEYWR is reset by PUC, AESSWRST, an error condition, changing AESOPx, changing AESKLx, and the start to (over)write a new key. Because it is reset when AESOPx is changed it can be set by software again to indicate that the loaded key is still valid.<br>0h = Not all bytes written<br>1h = All bytes written |
| 0 | BUSY | R | 0h | AES accelerator module busy; encryption, decryption, or key generation in progress. 0 = Not busy 1 = Busy<br>0h = Not busy<br>1h = Busy |



Copyright © 2026 Texas Instruments Incorporated

### 12.3.33 AESAKEY (Offset = 110Ch) [Reset = 00000000h]

AESAKEY is shown in Figure 12-42 and described in Table 12-46.

Return to the Summary Table.

aes accelerator key register

**Figure 12-42. AESAKEY**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| KEY3x | | | | | | | | KEY2x | | | | | | | | KEY1x | | | | | | | | KEY0x | | | | | | | |
| W-0h | | | | | | | | W-0h | | | | | | | | W-0h | | | | | | | | W-0h | | | | | | | |

**Table 12-46. AESAKEY Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY3x | W | 0h | AES key byte n+3 when AESAKEY is written as word. Do not use these bits for byte access. Do not mix word and byte access. Always reads as zero. The key is reset by PUC or by AESSWRST = 1.<br>00h = Smallest possible value<br>FFh = Highest possible value |
| 23-16 | KEY2x | W | 0h | AES key byte n+2 when AESAKEY is written as word. Do not use these bits for byte access. Do not mix word and byte access. Always reads as zero. The key is reset by PUC or by AESSWRST = 1.<br>00h = Smallest possible value<br>FFh = Highest possible value |
| 15-8 | KEY1x | W | 0h | AES key byte n+1 when AESAKEY is written as word. Do not use these bits for byte access. Do not mix word and byte access. Always reads as zero. The key is reset by PUC or by AESSWRST = 1.<br>00h = Smallest possible value<br>FFh = Highest possible value |
| 7-0 | KEY0x | W | 0h | AES key byte n when AESAKEY is written as word. AES next key byte when AESAKEY is written as byte. Do not mix word and byte access. Always reads as zero. The key is reset by PUC or by AESSWRST = 1.<br>00h = Smallest possible value<br>FFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 12.3.34 AESADIN (Offset = 1110h) [Reset = 00000000h]

AESADIN is shown in Figure 12-43 and described in Table 12-47.

Return to the Summary Table.

aes accelerator data in register

**Figure 12-43. AESADIN**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DIN3x | | | | | | | | DIN2x | | | | | | | | DIN1x | | | | | | | | DIN0x | | | | | | | |
| W-0h | | | | | | | | W-0h | | | | | | | | W-0h | | | | | | | | W-0h | | | | | | | |

**Table 12-47. AESADIN Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | DIN3x | W | 0h | AES data in byte n+3 when AESADIN is written as word. Do not use these bits for byte access. Do not mix word and byte access. Always reads as zero.<br>00h = Smallest possible value<br>FFh = Highest possible value |
| 23-16 | DIN2x | W | 0h | AES data in byte n+2 when AESADIN is written as word. Do not use these bits for byte access. Do not mix word and byte access. Always reads as zero.<br>00h = Smallest possible value<br>FFh = Highest possible value |
| 15-8 | DIN1x | W | 0h | AES data in byte n+1 when AESADIN is written as word. Do not use these bits for byte access. Do not mix word and byte access. Always reads as zero.<br>00h = Smallest possible value<br>FFh = Highest possible value |
| 7-0 | DIN0x | W | 0h | AES data in byte n when AESADIN is written as word. AES next data in byte when AESADIN is written as byte. Do not mix word and byte access. Always reads as zero.<br>00h = Smallest possible value<br>FFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 12.3.35 AESADOUT (Offset = 1114h) [Reset = 00000000h]

AESADOUT is shown in Figure 12-44 and described in Table 12-48.

Return to the Summary Table.

aes accelerator data out register

**Figure 12-44. AESADOUT**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DOUT3x | | | | | | | | DOUT2x | | | | | | | | DOUT1x | | | | | | | | DOUT0x | | | | | | | |
| R-0h | | | | | | | | R-0h | | | | | | | | R-0h | | | | | | | | R-0h | | | | | | | |

**Table 12-48. AESADOUT Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | DOUT3x | R | 0h | AES data out byte n+3 when AESADOUT is read as word. Do not use these bits for byte access. Do not mix word and byte access.<br>00h = Smallest possible value<br>FFh = Highest possible value |
| 23-16 | DOUT2x | R | 0h | AES data out byte n+2 when AESADOUT is read as word. Do not use these bits for byte access. Do not mix word and byte access.<br>00h = Smallest possible value<br>FFh = Highest possible value |
| 15-8 | DOUT1x | R | 0h | AES data out byte n+1 when AESADOUT is read as word. Do not use these bits for byte access. Do not mix word and byte access.<br>00h = Smallest possible value<br>FFh = Highest possible value |
| 7-0 | DOUT0x | R | 0h | AES data out byte n when AESADOUT is read as word. AES next data out byte when AESADOUT is read as byte. Do not mix word and byte access.<br>00h = Smallest possible value<br>FFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 12.3.36 AESAXDIN (Offset = 1118h) [Reset = 00000000h]

AESAXDIN is shown in Figure 12-45 and described in Table 12-49.

Return to the Summary Table.

aes accelerator xored data in register

**Figure 12-45. AESAXDIN**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| XDIN3x | | | | | | | | XDIN2x | | | | | | | | XDIN1x | | | | | | | | XDIN0x | | | | | | | |
| W-0h | | | | | | | | W-0h | | | | | | | | W-0h | | | | | | | | W-0h | | | | | | | |

**Table 12-49. AESAXDIN Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | XDIN3x | W | 0h | AES data in byte n+3 when AESAXDIN is written as word. Do not use these bits for byte access. Do not mix word and byte access. Always reads as zero.<br>00h = Smallest possible value<br>FFh = Highest possible value |
| 23-16 | XDIN2x | W | 0h | AES data in byte n+2 when AESAXDIN is written as word. Do not use these bits for byte access. Do not mix word and byte access. Always reads as zero.<br>00h = Smallest possible value<br>FFh = Highest possible value |
| 15-8 | XDIN1x | W | 0h | AES data in byte n+1 when AESAXDIN is written as word. Do not use these bits for byte access. Do not mix word and byte access. Always reads as zero.<br>00h = Smallest possible value<br>FFh = Highest possible value |
| 7-0 | XDIN0x | W | 0h | AES data in byte n when AESAXDIN is written as word. AES next data in byte when AESAXDIN is written as byte. Do not mix word and byte access. Always reads as zero.<br>00h = Smallest possible value<br>FFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 12.3.37 AESAXIN (Offset = 111Ch) [Reset = 00000000h]

AESAXIN is shown in Figure 12-46 and described in Table 12-50.

Return to the Summary Table.

aes accelerator xored data in register (no trigger)

**Figure 12-46. AESAXIN**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| XIN3x | | | | | | | | XIN2x | | | | | | | | XIN1x | | | | | | | | XIN0x | | | | | | | |
| W-0h | | | | | | | | W-0h | | | | | | | | W-0h | | | | | | | | W-0h | | | | | | | |

**Table 12-50. AESAXIN Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | XIN3x | W | 0h | AES data in byte n+3 when AESAXIN is written as word. Do not use these bits for byte access. Do not mix word and byte access. Always reads as zero.<br>00h = Smallest possible value<br>FFh = Highest possible value |
| 23-16 | XIN2x | W | 0h | AES data in byte n+2 when AESAXIN is written as word. Do not use these bits for byte access. Do not mix word and byte access. Always reads as zero.<br>00h = Smallest possible value<br>FFh = Highest possible value |
| 15-8 | XIN1x | W | 0h | AES data in byte n+1 when AESAXIN is written as word. Do not use these bits for byte access. Do not mix word and byte access. Always reads as zero.<br>00h = Smallest possible value<br>FFh = Highest possible value |
| 7-0 | XIN0x | W | 0h | AES data in byte n when AESAXIN is written as word. AES next data in byte when AESAXIN is written as byte. Do not mix word and byte access. Always reads as zero.<br>00h = Smallest possible value<br>FFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

Chapter 13

# AESADV

The AESADV accelerator module accelerates encryption and decryption operations in hardware based on the FIPS PUB 197 advanced encryption standard (AES).


**13.1 AES Overview** ..... 871
**13.2 AESADV Operation** ..... 872
**13.3 AESADV Registers** ..... 893





Copyright © 2026 Texas Instruments Incorporated

## 13.1 AES Overview

**Note**

Not all the MSPM0G devices support AES. See device-specific datasheet to determine if AES is present.

The AES accelerator module performs encryption and decryption of 128-bit data blocks with a 128-bit or 256-bit key in hardware. AES is a symmetric-key block cipher algorithm specified in FIPS PUB 197.

The AES accelerator features include:

- AES 128-bit block encryption and decryption
- Key scheduling in hardware
- Enc/decrypt only modes: CBC, CFB-1, CFB-8, CFB-128, OFB-128, CTR/ICM
- Authentication only modes: CBC-MAC, CMAC
- AES-CCM
- AES-GCM
- AES-CCM and AES-GCM modes support continuation with hold/resume of payload data
- 32-bit word access to provide key data, input data, and output data
- AES ready interrupt
- DMA triggers for input/output data
- Supported in RUN and SLEEP (see the *Operating Modes* section of the device technical reference manual)

A high level block diagram of the AES engine is shown in Figure 13-1. The AES engine consists of a processing core that performs both encryption/decryption as well as Galois field multiplication. The core is driven with configuration and data inputs that software will configure via memory mapped registers.

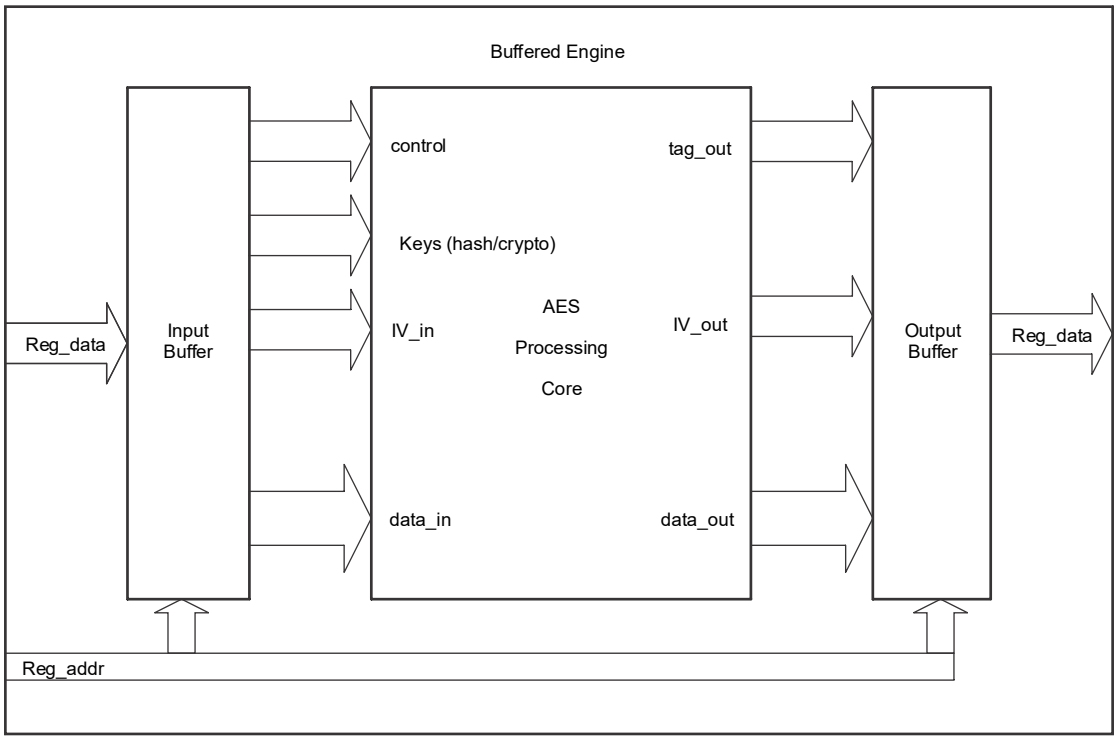



**Figure 13-1. AES Block Diagram**



Copyright © 2026 Texas Instruments Incorporated

### 13.1.1 AESADV Performance

The AESADV accelerator provides fast encryption and decryption of 128-bit blocks. AESADV accelerator performance in both cycles and execution time for block encryption and block decryption (with pregenerated decryption key) is given in Section 13.1. This table assumes that there are no system overheads (delays in supplying next input or reading out available output) that stall the engine.

**Table 13-1. AESADV Hardware Accelerator Key Performance Metrics**

| AESADV Key Length | Encryption | | | Decryption | | |
|---|---|---|---|---|---|---|
| | Cycles | Time (32MHz) | Time (80MHz) | Cycles | Time (32MHz) | Time (80MHz) |
| 128-bit | 76 | 2.38us | 0.95us | 76 | 2.38us | 0.95us |
| 256-bit | 81 | 2.53us | 1.01us | 81 | 2.53us | 1.01us |

## 13.2 AESADV Operation

The AESADV engine provides an efficient implementation of the Rijndael cipher (the AES algorithm) and a 128-bit polynomial multiplication (referred here to as 'GHASH', as per the AES-GCM specification). Rijndael is a block cipher with each data block consisting of 128-bits.

AES encryption requires a specific number of rounds depending on the key length. Supported key lengths are 128-bit, and 256 bit, requiring 10, and 14 rounds respectively.

The AESADV engine contains the AES ECB core and a dedicated 32-cycle polynomial multiplication module for performing GHASH operations (when GCM is configured). The polynomial multiplication operates in parallel with the AES core, if there is data available for both modules. This is the case after encryption of the first data block.

### 13.2.1 Loading the Key

Keys can be configured into the engine in one of two ways.

1. Secure key initialization via keystore controller: In this method, the intended AES key is securely transferred from the keystore controller into the engine via a secure private bus connecting the keystore controller and the AES engine. The keystore controller initiates the key transfer and waits for the AES engine to acknowledge transfer completion.
2. Software explicitly configures keys: In this method, software configures the key data into the engine by writing to the KEY0--KEY7 registers, writing 32-bits at a time starting with KEY0. For 128-bit keys, KEY0--KEY3 register will need to be written. For 256-bit keys, KEY0--KEY7 will need to be written.

In order to prevent key stealing attacks by partial modification method, the engine ensures that once a secure key transfer has completed via the keystore, software can no longer explicitly configure/modify the key. This status is provided by the STATUS.KEYWR field. If this field is 0, then software is allowed to write key data. If this field is 1, then software is not allowed to write key data. In order to allow software to write key data, the module has to be reset. The reset operation clears existing key data before new key data can be written by software.

### 13.2.2 Writing Input Data

Input data is written into the engine either through the DATA0, DATA1, DATA2, and DATA3 registers or through the DATA_IN register. If DMA is not being used to automate the input/output transfers (DMA_HS is 0), then CPU software can perform the 128-bit data input by writing 32-bit data to each of the registers DATA0, DATA1, DATA2 and DATA3 in sequence.

If DMA is being used to automate the input/output transfers (DMA_HS is 1), then the DMA channel that is associated with DMA Trigger 0 event must be configured to perform four 32-bit writes to the DATA_IN register.

### 13.2.3 Reading Output Data

Output data is read from the engine either via the DATA0/1/2/3 registers or via the DATA_OUT register. If DMA is not being used to automate the input/output transfers (DMA_HS is 0), then CPU software can read



Copyright © 2026 Texas Instruments Incorporated

out the 128-bit data output by reading 32-bit data from each of DATA0, DATA1, DATA2 and DATA3 registers in sequence.

If DMA is being used to automate the input/output transfers (DMA_HS is 1), then the DMA channel that is associated with DMA Trigger 1 event will need to be configured to perform 4 32-bit reads from the DATA_OUT register.

### *13.2.4 Operation Descriptions*

Both single block and block cipher mode operations are configured by writing the appropriate context to the AESADV registers. Independently, the input and output data can be transferred via CPU software or via DMA.

Block cipher modes are used to implement the ECB, CBC, OFB, and CFB block cipher modes using AES as the underlying block cipher. These modes work together with the DMA using the DMA triggers to support easy and fast encryption or decryption of more than 128 bits.

#### 13.2.4.1 Single Block Operation

A single 128-bit block of data can be encrypted or decrypted by first configuring the context registers and then initiating input data transfer. The following pseudo code describes the actions that are typically executed by CPU software for a basic encryption operation with 128-bit key.

```
wait AES_CTRL.CNTXT_RDY == '1' // wait for CNTXT_RDY to become 1
wait AES_CTRL.INPUT_RDY == '10' // wait for INPUT_RDY to become 1
write AES_KEY0 // first 32 bits of key
write AES_KEY1 // next 32 bits of key
write AES_KEY2 // next 32 bits of key
write AES_KEY3 // final 32 bits of key
write AES_CTRL.SAVE_CNTXT == '0' // clear AES save context
write AES_CTRL.KEYSIZE == '01' // Write AES 128 bit mode
write AES_CTRL.DIR == '1'      //Write AES encrypt mode
write AES_DATA0 // write 32-bit data LSW to supply 128-bit block
write AES_DATA1 // write 32-bit data (LSW + 1) to supply 128-bit block
write AES_DATA2 // write 32-bit data (LSW + 2) to supply 128-bit block
write AES_DATA3 // write 32-bit data MSW to supply 128-bit block
wait AES_CTRL.OUTPUT_RDY == '1' // wait for OUTPUT_RDY to become 1
read AES_DATA0 // read 32-bit data LSW to read out 128-bit encrypted data
read AES_DATA1 // read 32-bit data (LSW + 1) to read out 128-bit encrypted data
read AES_DATA2 // read 32-bit data (LSW + 2) to read out 128-bit encrypted data
read AES_DATA3 // read 32-bit data MSW to read out 128-bit encrypted data
```

#### 13.2.4.2 Electronic Codebook (ECB) Mode

The electronic codebook (ECB) cipher is the simplest block cipher mode. The plaintext data is divided into 128-bit blocks, and each block is encrypted and decrypted independently from any other block. The ECB cipher is shown in Figure 13-2. Note that each 128-bit data block can be encrypted or decrypted individually without any knowledge of the other blocks of plaintext or ciphertext.

While ECB is simple to understand and implement, it has a key disadvantage: the same 128-bit plaintext block is always encrypted into the same ciphertext block, allowing patterns in the ciphertext to be detected.



Copyright © 2026 Texas Instruments Incorporated

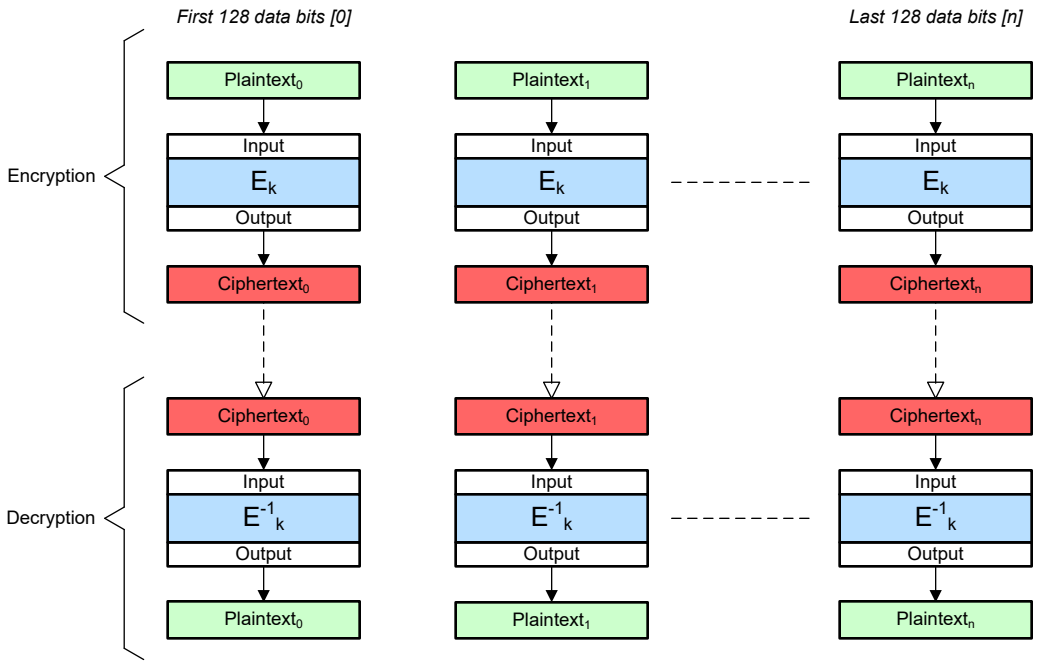



**Figure 13-2. ECB Cipher**

The AESADV accelerator supports automated encryption and decryption of more than 128 bits of data in ECB block cipher mode either through interrupts or through the use of DMA. Software interrupt-based multi-block handling uses the AESADV interrupt condition reported in the CPU_INT.IIDX.STAT field. If this field reads 0x2 (INPUTRDY), then the next block of input can be written (as 4 32-bit writes to DATA0/1/2/3). If this field reads 0x1 (OUTPUTRDY), then the output block can be read (as 4 32-bit reads from DATA0/1/2/3).

In DMA mode, ECB uses two DMA channels. Channel bound to DMA_TRIG0 is the input channel. Channel bound to DMA_TRIG1 is the output channel. The channels need to be configured to perform one 32-bit read/write per trigger.

#### *13.2.4.2.1 ECB Encryption*

ECB mode encryption of *N* blocks of plaintext into *N* blocks of ciphertext without CPU interaction is achieved through the use of 2 DMA channels (referred to as DMA_A and DMA_B). To implement ECB encryption follow these steps:

1. Configure Output DMA channel for saving ciphertext:
   a. Set DMA channel trigger selection to AES Trig1
   b. Set DMA channel source address to DATA_OUT
   c. Set DMA channel destination address to location where ciphertext is to be stored (for example, SRAM)
   d. Set DMA channel transfer size to *N*∗4
   e. Set DMA channel mode to single transfer mode
   f. In the AESADV event registers, unmask Trig1 in the IMASK register of DMA_TRIG1
2. Configure Input DMA channel for loading plaintext:
   a. Set DMA channel trigger selection to AES Trig0
   b. Set DMA channel source address to location where plaintext is stored (for example, SRAM)
   c. Set DMA channel destination address to DATA_IN
   d. Set DMA channel transfer size to *N*∗4
   e. Set DMA channel mode to single transfer mode
   f. In the AES event registers, unmask Trig0 in the IMASK register of DMA_TRIG0
3. Configure and enable the DMA interrupt for the Output DMA channel in the DMA controller
4. Configure DMA_HS for DMA based handshake: set DMA_HS[DMA_DATA_ACK] = 1



Copyright © 2026 Texas Instruments Incorporated

5. Configure the CTRL register for block cipher encryption mode for ECB:
   a. Set key size in CTRL[KEY_SIZ]
   b. Select Direction for Encryption by CTRL[DIR] = 1
   c. Select ECB mode by setting rest of the bits in CTRL to 0
6. Load key as described in Section 13.2.1.
7. Start encryption by writing the number of bytes *N×4* to AES C_LENGTH_0 and C_LENGTH_1 registers
8. Wait for the DMA channel interrupt which indicates completion of the entire operation. The ciphertext output will be stored in the location configured in step 1c.

#### *13.2.4.2.2 ECB Decryption*

ECB Mode Decryption configuration is nearly identical to ECB Mode Encryption. The only difference is that the direction bit (CTRL.DIR) has to be set to 0.

### 13.2.4.3 Cipher Block Chaining (CBC) Mode

The cipher block chaining (CBC) cipher mode builds upon the ECB cipher mode to make the ciphertext output for each block dependent not only on the plaintext and the cipher key *k*, but also upon the ciphertext of the previous block. The CBC cipher is shown in Figure 13-3. Like ECB mode, the plaintext data is divided into 128-bit blocks. Unlike ECB mode, in CBC mode each new plaintext block is bit-wise XORed with the previous ciphertext block to create the input to the AES block cipher.

In CBC mode, an unpredictable initialization vector (IV) must be provided. The initialization vector is XORed with the first plaintext block, as there is no "previous" ciphertext block to XOR the first plaintext block with.

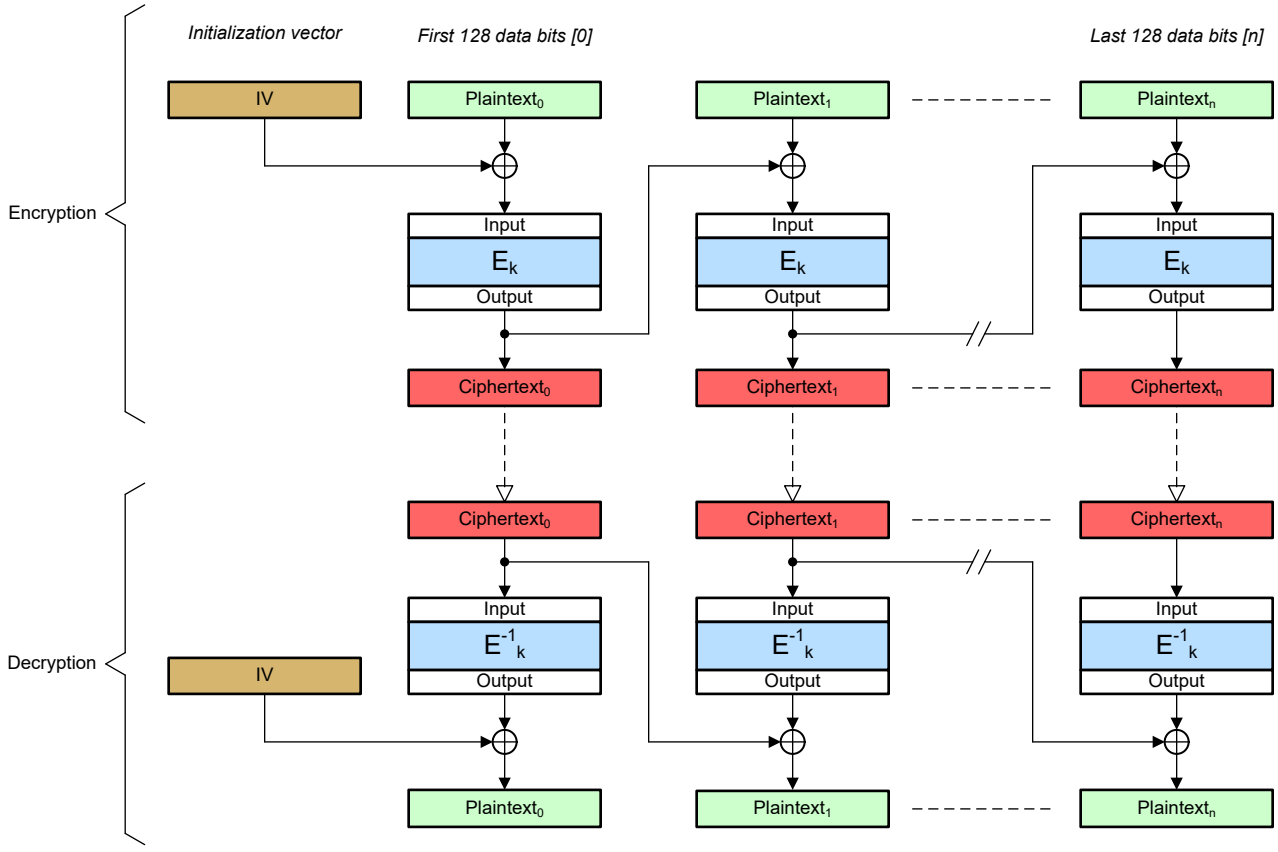



**Figure 13-3. CBC Cipher**

The AESADV accelerator supports automated CBC mode operation of more than 128 bits of data either through interrupts or through the use of DMA. Software interrupt-based multi-block handling uses the AES interrupt



Copyright © 2026 Texas Instruments Incorporated

condition reported in the CPU_INT.IIDX.STAT field. If this field reads 0x2 (INPUTRDY), then the next block of input can be written (as 4 32-bit writes). If this field reads 0x1 (OUTPUTRDY), then the output block can be read (as 4 32-bit reads).

In DMA mode, CBC utilizes two DMA channels. Channel bound to DMA_TRIG0 is the input channel. Channel bound to DMA_TRIG1 is the output channel. The channels need to be configured to perform one 32-bit read/write per trigger.

#### *13.2.4.3.1 CBC Encryption*

CBC mode encryption of *N* blocks of plaintext into *N* blocks of ciphertext without CPU interaction is achieved through the use of 2 DMA channels. To implement CBC encryption follow these steps:

1. Configure Output DMA channel for saving ciphertext:
   a. Set DMA channel trigger selection to AES Trig1
   b. Set DMA channel source address to DATA_OUT
   c. Set DMA channel destination address to location where ciphertext is to be stored (for example, SRAM)
   d. Set DMA channel transfer size to *N*∗4
   e. Set DMA channel mode to single transfer mode
   f. In the AES event registers, unmask Trig1 in the IMASK register of DMA_TRIG1
2. Configure Input DMA channel for loading plaintext:
   a. Set DMA channel trigger selection to AES Trig0
   b. Set DMA channel source address to location where plaintext is stored (for example, SRAM)
   c. Set DMA channel destination address to DATA_IN
   d. Set DMA channel transfer size to *N*∗4
   e. Set DMA channel mode to single transfer mode
   f. In the AES event registers, unmask Trig0 in the IMASK register of DMA_TRIG0
3. Configure and enable the DMA interrupt for the Output DMA channel in the DMA controller
4. Configure DMA_HS for DMA based handshake: set DMA_HS[DMA_DATA_ACK] = 1
5. Load key as described in Section 13.2.1
6. Load Initialization vector (IV) by writing to IV0, IV1, IV2 and IV3 registers
7. Configure the CTRL register for block cipher encryption mode for CBC
   a. Select key size via CTRL[KEY_SIZ]
   b. Select Direction for Encryption by CTRL[DIR] = 1
   c. Select CBC mode by setting CTRL[CBC]=1
8. Start encryption by writing the number of bytes N×4 to AES C_LENGTH_0 and C_LENGTH_1 registers
9. Wait for the DMA channel interrupt that indicates completion of the entire operation. The ciphertext output is stored in the location configured in step 1c.

#### *13.2.4.3.2 CBC Decryption*

CBC-mode decryption is nearly identical to CBC-mode encryption setup, with the exception of configuring CTRL[DIR] = 0 for decryption.

### 13.2.4.4 Output Feedback (OFB) Mode

The output feedback mode leverages an initialization vector (IV) to generate a keystream by repeatedly encrypting the IV with the cipher key. The output ciphertext is obtained by XORing plaintext with the encrypted and re-encrypted versions of the initialization vector. The OFB cipher is shown in Figure 13-4.

In OFB mode, the initialization vector must be a nonce (number used once). To prevent loss of confidentiality, each IV must only be used one time with a given key, and any value passed into the cipher $E_k$ for a given key $k$ must not be used as an initialization vector with the same key $k$.



Copyright © 2026 Texas Instruments Incorporated

**Note**

As OFB is a stream cipher, the AES block function is used in the forward (encryption) mode when performing OFB encryption or OFB decryption. When decrypting, the keystream is simply regenerated again to be XORed with the ciphertext, giving back the plaintext.

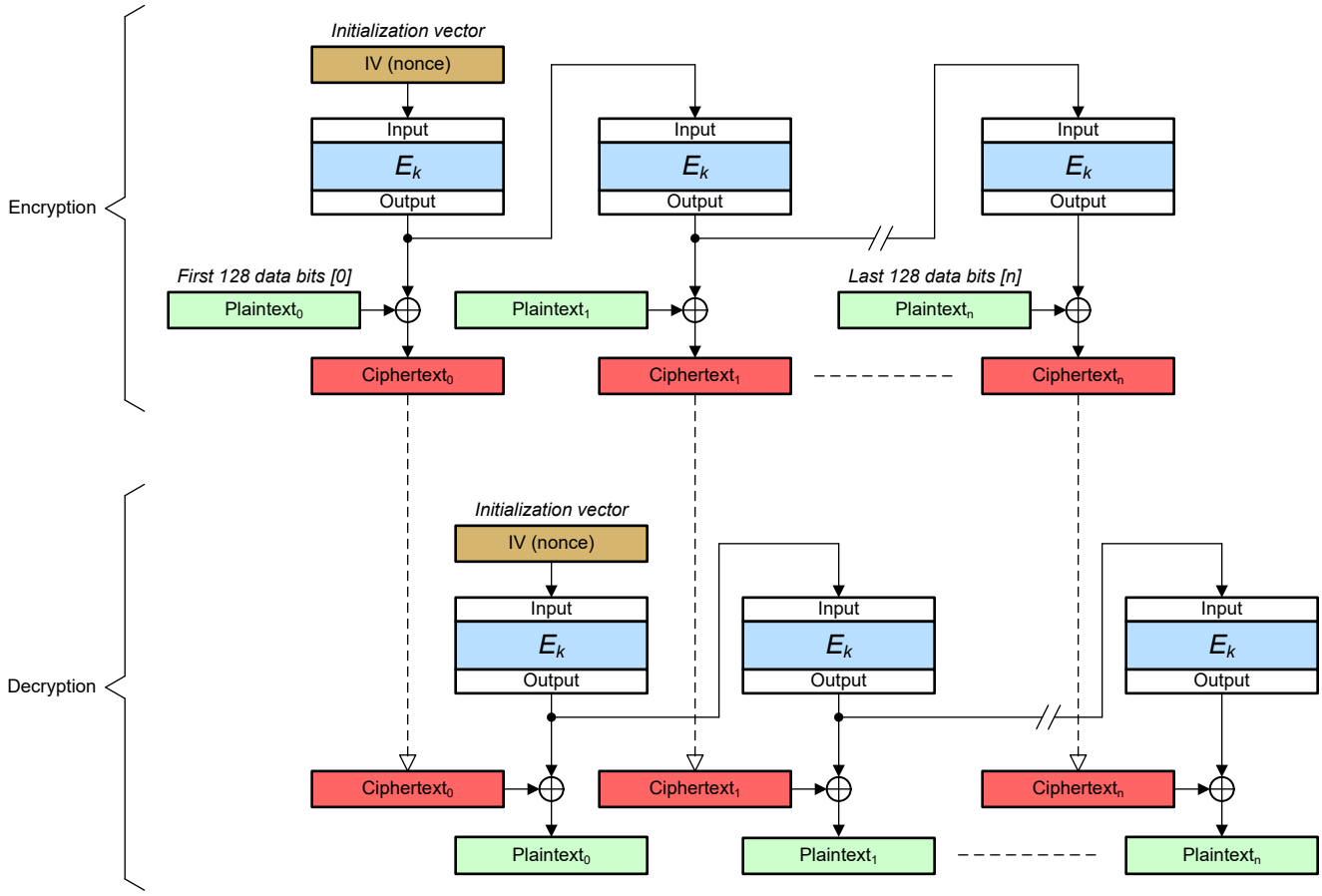



**Figure 13-4. OFB Cipher**

The AES accelerator supports automated OFB mode operation of more than 128 bits of data either through interrupts or through the use of DMA. Software interrupt-based multi-block handling uses the AES interrupt condition reported in the CPU_INT.IIDX.STAT field. If this field reads 0x2 (INPUTRDY), then the next block of input can be written (as 4 32-bit writes). If this field reads 0x1 (OUTPUTRDY), then the output block can be read (as 4 32-bit reads).

In DMA mode, OFB utilizes two DMA channels. Channel bound to DMA_TRIG0 is the input channel. Channel bound to DMA_TRIG1 is the output channel. The channels need to be configured to perform one 32-bit read/write per trigger.

#### *13.2.4.4.1 OFB Encryption*

OFB mode encryption of *N* blocks of plaintext into *N* blocks of ciphertext without CPU interaction is achieved through the use of 2 DMA channels. To implement OFB encryption follow these steps:

1. Configure Output DMA channel for saving ciphertext:
   a. Set DMA channel trigger selection to AES Trig1
   b. Set DMA channel source address to DATA_OUT
   c. Set DMA channel destination address to location where ciphertext is to be stored (for example, SRAM)
   d. Set DMA channel transfer size to *N*∗4



Copyright © 2026 Texas Instruments Incorporated

e. Set DMA channel mode to single transfer mode
f. In the AES event registers, unmask Trig1 in the IMASK register of DMA_TRIG_DATAOUT

2. Configure Input DMA channel for loading plaintext:
   a. Set DMA channel trigger selection to AES Trig0
   b. Set DMA channel source address to location where plaintext is stored (for example, SRAM)
   c. Set DMA channel destination address to DATA_IN
   d. Set DMA channel transfer size to *N**4
   e. Set DMA channel mode to single transfer mode
   f. In the AES event registers, unmask Trig0 in the IMASK register of DMA_TRIG0
3. Configure and enable the DMA interrupt for the Output DMA channel in the DMA controller
4. Configure DMA_HS for DMA based handshake: set DMA_HS[DMA_DATA_ACK] = 1
5. Load key as described in Section 13.2.1
6. Load Initialization vector (IV) by writing to IV0, IV1, IV2 and IV3 registers
7. Configure the CTRL register for block cipher encryption mode for OFB
   a. Select key size via CTRL[KEY_SIZ]
   b. Select Direction for Encryption by CTRL[DIR] = 1
   c. Select OFB mode by setting CTRL[OFB_GCM_CCM_CONT]=1
8. Start encryption by writing number of bytes N×4 to AES C_LENGTH_0 and C_LENGTH_1 registers
9. Wait for the DMA channel interrupt which indicates completion of the entire operation. The ciphertext output will be stored in the location configured in step 1c.

**Note**

OFB_GCM_CCM_CONT bit has dual use. When CCM/GCM modes are not selected, then this bit serves to select OFB mode.

#### *13.2.4.4.2 OFB Decryption*

OFB-mode decryption is nearly identical to OFB-mode encryption setup, with the exception of configuring CTRL[DIR] = 0 for decryption.

#### 13.2.4.5 Cipher Feedback (CFB) Mode

The cipher feedback (CFB) mode is similar to the output feedback (OFB) mode, with the key difference being that the input block to the block cipher *E* used to generate the key stream is taken from the previous ciphertext block (after being XORed with the plaintext), vs. taken before being XORed with the plaintext (as in the case of OFB). As a result, the keystream is dependent upon the plaintext, which is not true of OFB. The CFB cipher is shown in Figure 13-5.

Like OFB, CFB requires an initialization vector (IV). In CFB mode, the initialization vector (IV) must be unpredictable.



Copyright © 2026 Texas Instruments Incorporated

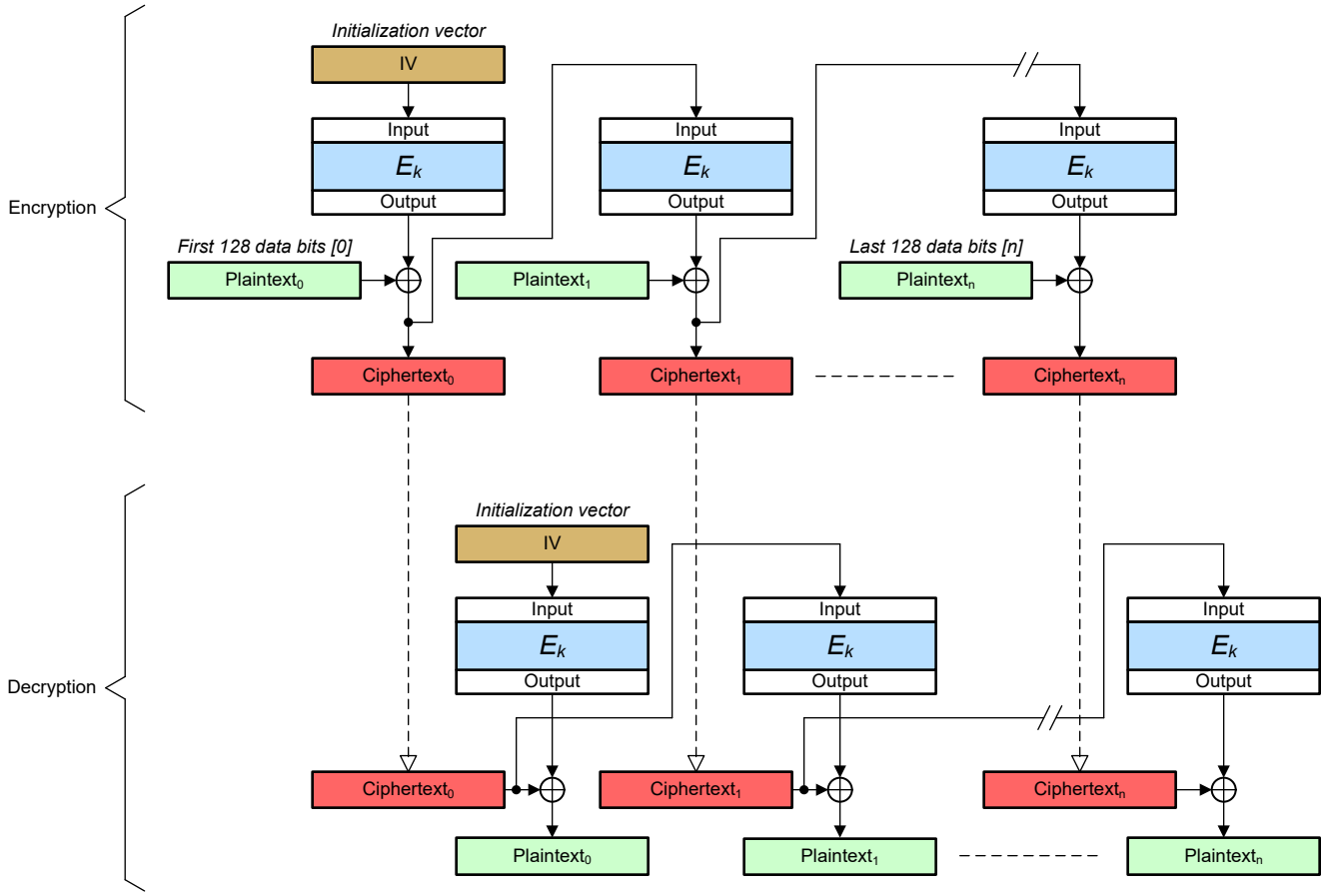



**Figure 13-5. CFB Cipher**

The AESADV accelerator supports automated CFB mode operation of more than 128 bits of data either through interrupts or through the use of DMA. Software interrupt-based multi-block handling uses the AES interrupt condition reported in the CPU_INT.IIDX.STAT field. If this field reads 0x2 (INPUTRDY), then the next block of input can be written (as 4 32-bit writes). If this field reads 0x1 (OUTPUTRDY), then the output block can be read (as 4 32-bit reads).

In DMA mode, CFB utilizes two DMA channels. Channel bound to DMA_TRIG_DATAIN is the input channel. Channel bound to DMA_TRIG_DATAOUT is the output channel. The channels need to be configured to perform one 32-bit read/write per trigger.

#### *13.2.4.5.1 CFB Encryption*

CFB mode encryption of *N* blocks of plaintext into *N* blocks of ciphertext without CPU interaction is achieved through the use of 2 DMA channels. To implement CFB encryption follow these steps:

1. Configure Output DMA channel for saving ciphertext:
   a. Set DMA channel trigger selection to AES Trig1
   b. Set DMA channel source address to DATA_OUT
   c. Set DMA channel destination address to location where ciphertext is to be stored (for example, SRAM)
   d. Set DMA channel transfer size to *N*∗4
   e. Set DMA channel mode to single transfer mode
   f. In the AES event registers, unmask Trig1 in the IMASK register of DMA_TRIG_DATAOUT
2. Configure Input DMA channel for loading plaintext:
   a. Set DMA channel trigger selection to AES Trig0
   b. Set DMA channel source address to location where plaintext is stored (for example, SRAM)
   c. Set DMA channel destination address to DATA_IN



Copyright © 2026 Texas Instruments Incorporated

   d. Set DMA channel transfer size to *N**4
   e. Set DMA channel mode to single transfer mode
   f. In the AES event registers, unmask Trig0 in the IMASK register of DMA_TRIG0
3. Configure and enable the DMA interrupt for the Output DMA channel in the DMA controller
4. Configure DMA_HS for DMA based handshake: set DMA_HS[DMA_DATA_ACK] = 1
5. Load key as described in Section 13.2.1
6. Load Initialization vector (IV) by writing to IV0, IV1, IV2 and IV3 registers
7. Configure the CTRL register for block cipher encryption mode for CFB
   a. Select key size via CTRL[KEY_SIZ]
   b. Select Direction for Encryption by CTRL[DIR] = 1
   c. Select CFB mode by setting CTRL[CFB]=1
   d. Select feedback width by setting CTRL[CTR_WIDTH]
      i. 00b - CFB-128
      ii. 01b - CFB-1
      iii. 10b - CFB-8
8. Start encryption by writing number of bytes N×4 to AES C_LENGTH_0 and C_LENGTH_1 registers
9. Wait for the DMA channel interrupt that indicates completion of the entire operation. The ciphertext output is stored in the location configured in step 1c.

#### *13.2.4.5.2 CFB Decryption*

CFB-mode decryption is nearly identical to CFB-mode encryption setup, with the exception of configuring CTRL[DIR] = 0 for decryption.

### 13.2.4.6 Counter (CTR) Mode

The counter mode leverages a nonce (number used once) and a counting integer to generate a keystream by appending the counter to the nonce and encrypting the combined nonce || counter value with the cipher key.

The nonce must only be used once with a given key *k*. The counter value can start from any value and is incremented for each 128-bit block of data.

The keystream is derived be encrypting the nonce || counter value for each 128-bit data block with the cipher key *k*. The output ciphertext is then obtained by XORing the plaintext with the encrypted nonce || counter value for each data block. The CTR cipher is shown in Figure 13-6.



Copyright © 2026 Texas Instruments Incorporated

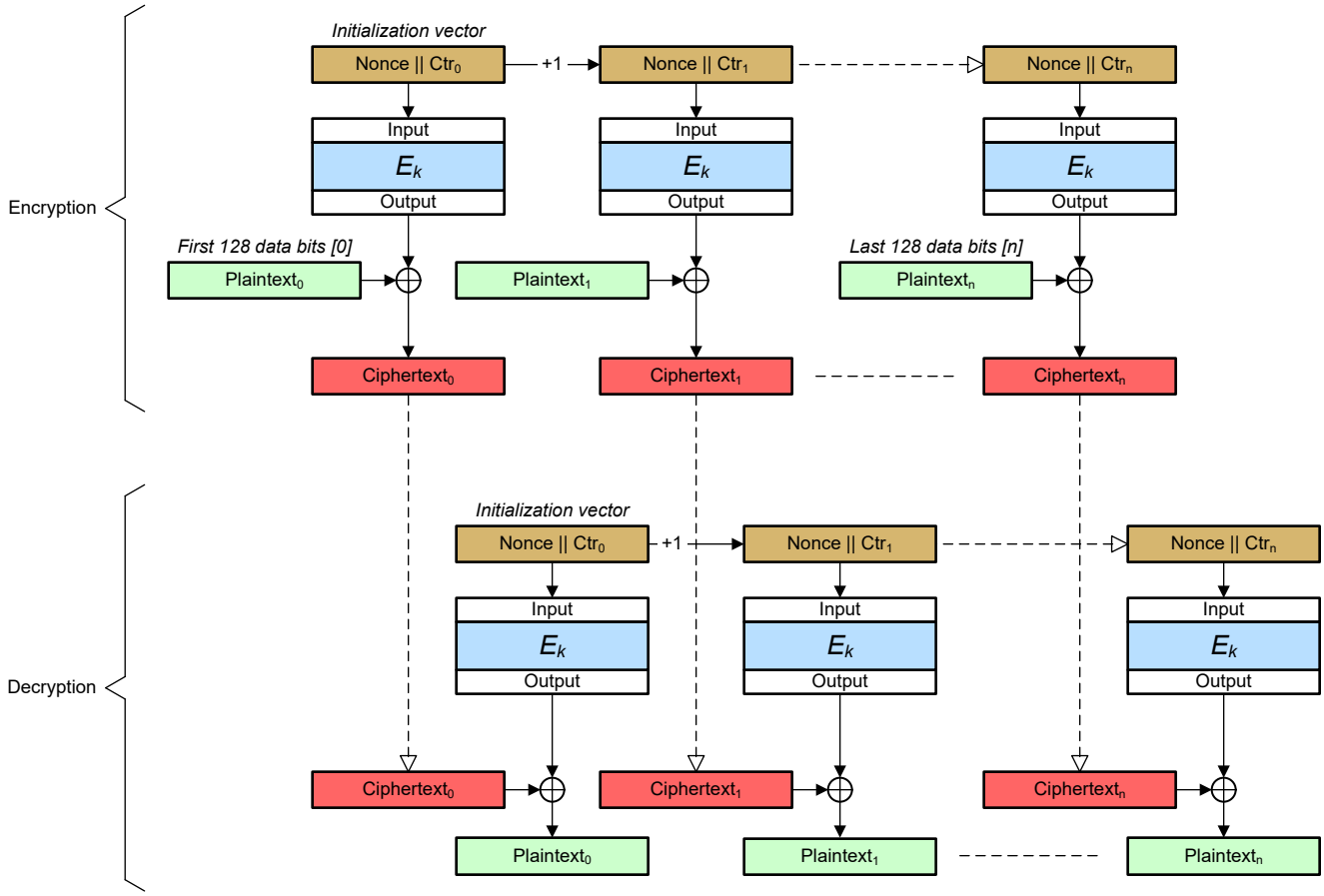



**Figure 13-6. CTR Cipher**

The AES accelerator implements logic and storage for incrementing and storing nonce || counter from one block to the next.

#### *13.2.4.6.1 CTR Encryption*

CTR mode encryption of *N* blocks of plaintext into *N* blocks of ciphertext without CPU interaction is achieved through the use of 2 DMA channels. To implement CTR encryption follow these steps:

1. Configure Output DMA channel for saving ciphertext:
   a. Set DMA channel trigger selection to AES Trig1
   b. Set DMA channel source address to DATA_OUT
   c. Set DMA channel destination address to location where ciphertext is to be stored (for example, SRAM)
   d. Set DMA channel transfer size to *N*∗4
   e. Set DMA channel mode to single transfer mode
   f. In the AES event registers, unmask Trig1 in the IMASK register of DMA_TRIG_DATAOUT
2. Configure Input DMA channel for loading plaintext:
   a. Set DMA channel trigger selection to AES Trig0
   b. Set DMA channel source address to location where plaintext is stored (for example, SRAM)
   c. Set DMA channel destination address to DATA_IN
   d. Set DMA channel transfer size to *N*∗4
   e. Set DMA channel mode to single transfer mode
   f. In the AES event registers, unmask Trig0 in the IMASK register of DMA_TRIG0
3. Configure and enable the DMA interrupt for the Output DMA channel in the DMA controller
4. Configure DMA_HS for DMA based handshake: set DMA_HS[DMA_DATA_ACK] = 1
5. Load key as described in Section 13.2.1



Copyright © 2026 Texas Instruments Incorporated

6. Load Initialization vector (IV) by writing to IV0, IV1, IV2 and IV3 registers
7. Configure the CTRL register for block cipher encryption mode for CTR
   a. Select key size via CTRL[KEY_SIZ]
   b. Select Direction for Encryption by CTRL[DIR] = 1
   c. Select CTR mode by setting CTRL[CTR]=1
   d. Select CTR width by setting CTRL[CTR_WIDTH]
      i. 00b -CTR32
      ii. 01b -CTR64
      iii. 10b -CTR96
      iv. 11b -CTR128

8. Start encryption by writing number of bytes N×4 to AES C_LENGTH_0 and C_LENGTH_1 registers
9. Wait for the DMA channel interrupt which indicates completion of the entire operation. The ciphertext output will be stored in the location configured in step 1c.

#### *13.2.4.6.2 CTR Decryption*

CTR-mode decryption is nearly identical to CTR-mode encryption setup, with the exception of configuring CTRL[DIR] = 0 for decryption.

### 13.2.4.7 Galois Counter (GCM) Mode

A GCM protocol operation is a combined operation, consisting of encryption/decryption and authentication. Figure 13-7 illustrates an overview of the GCM operation.

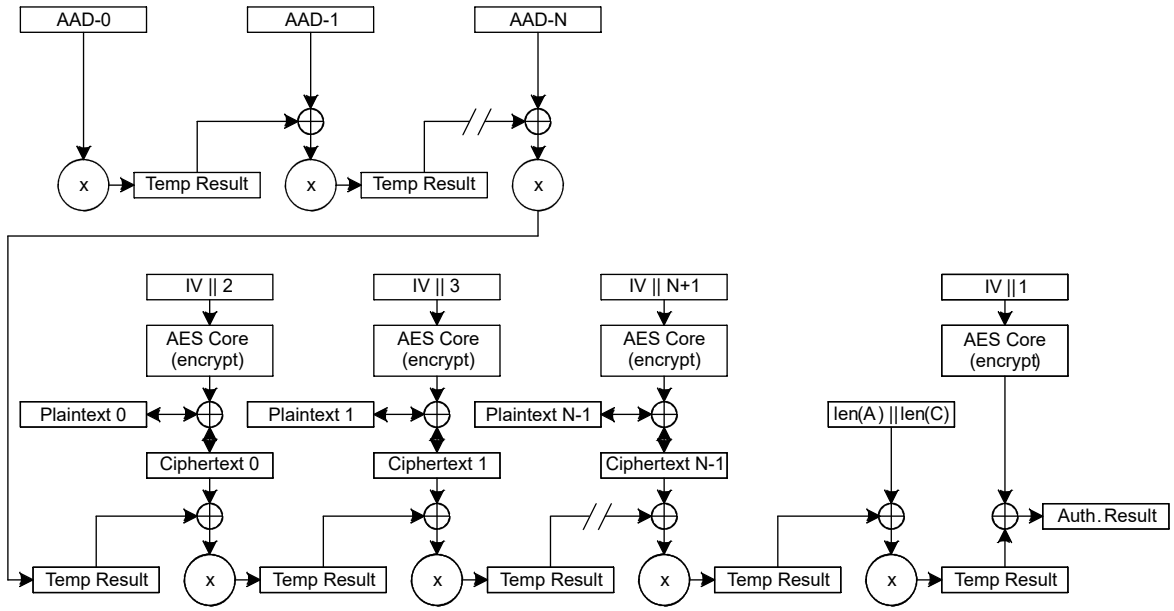



**Figure 13-7. GCM Protocol Operation**

A part of the input data stream can be authenticated only, while normally most of the input data is encrypted/decrypted and authenticated. The authentication only data always needs to be in front of the data that requires encryption. Within GCM, the authentication only data is called the AAD (Additional Authentication Data). The AAD is fetched independently of the other data.

The intermediate (temp) result data is used as input for the remaining authentication operation. Since the authentication operation does not require the encryption core but only the polynomial multiplication, both encryption/decryption and authentication are performed in parallel. After encryption of the last data block, an additional polynomial multiplication and encryption are required to respectively authenticate a 128-bit length vector and finally encrypt the authentication result.



Copyright © 2026 Texas Instruments Incorporated

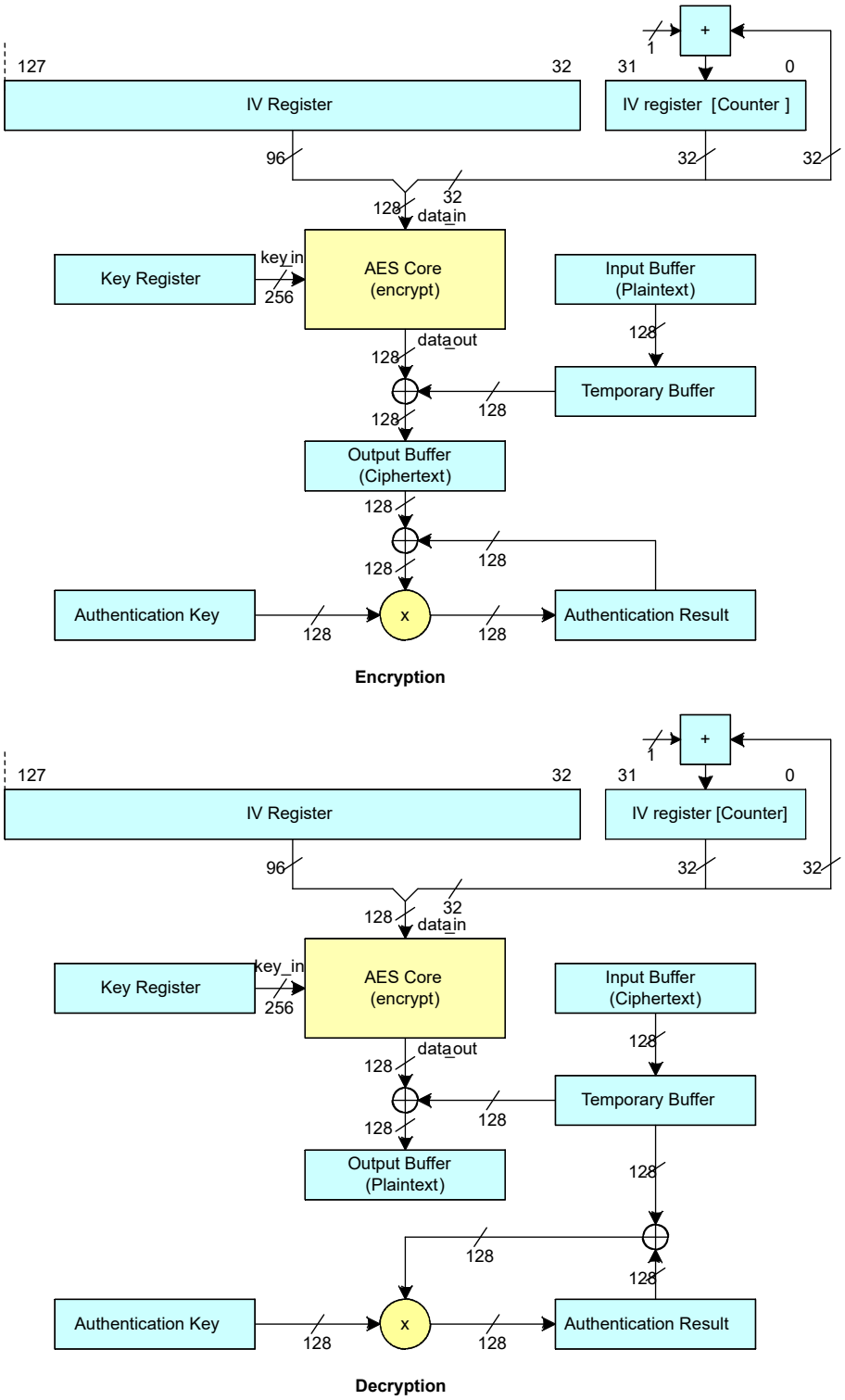



**Figure 13-8. GCM Operation on a Block**

GMAC operations, as specified in [NIST-SP800-38D], are also supported via the GCM operation. GMAC is a special use of GCM where no crypto data is processed and only AAD data is provided to produce a MAC of the input data. The crypto data length is set to zero for this case.



Copyright © 2026 Texas Instruments Incorporated

Figure 13-8 illustrates the operations performed in one round. In one round of a GCM operation for both encryption and decryption, a 32-bit counter is used as IV (as it is for CTR mode). The data is encrypted the same way as CTR mode, by XOR-ing the crypto output with the input. After the encryption/decryption, the ciphertext is XOR-ed with the intermediate authentication result. The XOR-ed result is used as input for the polynomial multiplication to create the next (intermediate) authentication result.

#### *13.2.4.7.1 GHASH Operation*

For GHASH operations, the engine performs a modular polynomial multiplication in the GF(2128) field. The result is XOR-ed with the encrypted GCM initialization vector (referred to as 'Y0-encrypted'). Y0-encrypted is only relevant when the engine is performing a complete GCM operation, therefore in other modes the value of Y0-encrypted is forced to zero.

Also, for GCM mode only, the GCM 'Hash key' or 'H' input can be pre-calculated and supplied to the engine directly, or it can be calculated by the engine internally, by encrypting the value '0', using the encryption key.

#### *13.2.4.7.2 GCM Operating Modes*

For GCM three cases need to be distinguished.

- The first one is Autonomous GCM Mode where both H and Y0-encrypted are calculated internally. This mode requires that a 128-bit Y0 be provided to the core via the IV together with the mode.

  **Note**

  GCM mode bits must be set to 2'b11.

- The second case is the scenario where H is pre-calculated and Y0 still needs to be encrypted by the engine on a per packet basis. This can be useful when multiple packets use the same AES-key. Since H is constant for all packets using the same key, a pre-calculation saves cycles for each packet using that key. H can simply be calculated by performing a basic AES-ECB encryption with the AES-key and a data block containing all zeros or more formally: H=E(K, {}). Once H is calculated, it can be loaded with the control data every time a packet is processed that requires the same AES-key.

  **Note**

  GCM mode bits must be set to 2'b10.

- In the last case, neither H nor Y0-encrypted are calculated by the core. In this case, Y0-encrypted is forced to zeros, such that the hash result is not encrypted but provided plain via the TAG output registers. This scenario can be selected if a hash (GHASH) only operation needs to be performed. A scenario where this setting can be used is GCM IV-truncation. The GCM specification [GCM] allows an IV that has a length other than 96-bits. In this case, a basic GHASH operation needs to be performed to calculate a 128-bit Y0. For a basic GHASH operation, H needs to be pre-calculated (as explained in the previous paragraph). If H is available, the GHASH operation is similar to that of a general GCM operation with H pre-calculated, only the crypto input data will not be encrypted or decrypted.

  **Note**

  In the default case a 96-bit IV is combined with a 32-bit counter to create a 128-bit Y0 (meaning: Y0 = {IV||031||1}).

  **Note**

  GCM mode bits must be set to 2'b01.

#### *13.2.4.7.2.1 Autonomous GCM Operation*

In this mode, the engine calculates H and Y0-encrypted internally. At the high-level, the programming involves these steps:

- Provide GCM context (key, IV, lengths and mode).
- Provide AAD data (and wait for calculation of H and encryption of Y0).
- Provide next AAD data.



Copyright © 2026 Texas Instruments Incorporated

- Provide last AAD data.
- Provide first crypto data.
- Provide next crypto data.
- Read result data and provide next crypto data.
- ...
- Read result data and provide last crypto data.
- Read result data.
- Read result data.
- Read authentication result (TAG).

To implement GCM encryption over N blocks of plaintext and M blocks of AAD, follow these steps:

1. Configure Output DMA channel for saving ciphertext:
   a. Set DMA channel trigger selection to AES Trig1
   b. Set DMA channel source address to DATA_OUT
   c. Set DMA channel destination address to location where ciphertext is to be stored (for example, SRAM)
   d. Set DMA channel transfer size to *N×4*
   e. Set DMA channel mode to single transfer mode
   f. In the AES event registers, unmask Trig1 in the IMASK register of DMA_TRIG_DATAOUT
2. Configure Input DMA channel for loading AAD and plaintext:
   a. Set DMA channel trigger selection to AES Trig0
   b. Set DMA channel source address to location where plaintext is stored (for example, SRAM)
   c. Set DMA channel destination address to DATA_IN
   d. Set DMA channel transfer size to *(N+M)*4*
   e. Set DMA channel mode to single transfer mode
   f. In the AES event registers, unmask Trig0 in the IMASK register of DMA_TRIG0
3. Configure and enable the DMA interrupt for the Output DMA channel in the DMA controller
4. Configure DMA_HS for DMA based handshake: set DMA_HS[DMA_DATA_ACK] = 1
5. Load Encryption/Decryption key as described in Section 13.2.1
6. Load GCMCCM_TAGn (0,1,2,3) registers with 0s.
7. Load Initialization vector (IV) by writing to IV0, IV1, IV2 and IV3 registers
8. Configure the CTRL register for block cipher encryption mode for GCM
   a. Select key size via CTRL[KEY_SIZ]
   b. Select Direction for Encryption by CTRL[DIR] = 1
   c. Select GCM mode by setting CTRL[GCM] = 3
   d. Select CTR mode by setting CTRL[CTR] = 1
   e. Enable saving of TAG by setting CTRL[SAVE_CNTXT] = 1
9. Write encryption/decryption byte count N×4 to AES C_LENGTH_0 and C_LENGTH_1 registers
10. Write authentication data (AAD) byte count M*4 to AES AAD_LENGTH register
11. Wait for the DMA output channel interrupt to indicate completion of the entire operation. The output is stored started at the location configured in step 1c.
12. Read out the final TAG from the TAG0, TAG1, TAG2, TAG3 registers
13. GCM Continued Operation:
    a. SW guidelines for suspend resume operation of GCM through DMA:
       i. Before suspending the GCM operation (before writing to the GET_DIGEST bit in CTRL register) stop the DMA transfer
       ii. After stopping the DMA transfer, request for the GET_DIGEST
       iii. Read all required fields - note where the DMA transfer is halted
       iv. Use the above information as input to the engine before the resume operation starts
    b. SW guidelines for saved context ready interrupt:
       i. Ensure SAVE_CNTXT bit is true to enable SAVEDCNTXTRDY usage



Copyright © 2026 Texas Instruments Incorporated

ii. Poll the RIS register continuously for SAVEDCNTXTRDY status instead of waiting for interrupt. CPU will go into sleep mode, causing interrupt to not be triggered. SAVECNTXTRDY is mutually exclusive to the CNTXT_RDY bit

```
while(MIS != SAVEDCNTXTRDY);
// read out AES context MMRs and save to SRAM
```

**Note**

The AAD and cryptographic data can end misaligned. The CPU must pad both to a 128-bit boundary with zeroes. More formally, the AAD and crypto data padding must satisfy the bit string: 0n, with 0 <= n <= 127. This means that the AAD must be provided as separate blocks to the engine, such that the cryptographic data starts 128-bit aligned. If the AAD/cryptographic data stream is 128-bit aligned no padding is required. Because the engine only supports bytes, n must be such that (n MOD 8) = 0. Further, because a single DMA channel supplies both AAD and plaintext, the entire data must be organized contiguously in memory with the first M blocks of AAD followed by N blocks of plaintext. This memory contiguity restriction is not applicable when CPU software directly provides inputs via interrupt handling.

Do not load both length values with zeroes. If a data stream is done and the next data stream uses the same key and control, only the IV and length values need be re-loaded.

#### *13.2.4.7.2.1.1 GMAC*

GMAC is a special case of GCM wherein there is no payload (N=0). AES C_LENGTH_0 and C_LENGTH_1 registers are set to 0 in this case.

#### *13.2.4.7.2.2 GCM With Pre-Calculations*

Pre-calculation of H:

1. Provide AES-ECB context (key and mode)
2. Provide zeros as data
3. Read result data (H)

IV truncation/pre-calculation (GHASH only operation):

- Provide GHASH context (H, lengths and mode)

**Note**

GHASH only: Y0-encrypted forced to zero, H loaded, and no crypto mode selected

1. Provide IV-data-block
2. Provide next IV-data-block
3. Read dummy result and provide next IV-data-block
4. ...
5. Read dummy result data and provide last IV-data-block
6. Read dummy result data
7. Read dummy result data
8. Read authentication result (TAG... this data is Y0)

GCM operation with pre-calculated H:

1. Provide GCM context (key, Y0 as IV, H, lengths and mode)
2. Provide AAD data (and wait for encryption of Y0)
3. Provide next AAD data
4. Provide last AAD data
5. Provide first crypto data
6. Provide next crypto data



Copyright © 2026 Texas Instruments Incorporated

7. Read result data and provide next crypto data
8. ...
9. Read result data and provide last crypto data
10. Read result data
11. Read result data
12. Read authentication result (TAG)

For GCM with pre-calculated H, select GCM mode by setting CTRL[GCM] = 2

#### *13.2.4.7.2.3 GCM Operation With Precalculated H- and Y0-Encrypted Forced to Zero*

The outline of operations is listed below:

1. Provide GCM context (key, H, lengths and mode)
2. Provide AAD data
3. Provide next AAD data
4. Provide last AAD data
5. Provide first crypto data
6. Provide next crypto data
7. Read result data and provide next crypto data
8. ...
9. Read result data and provide last crypto data
10. Read result data
11. Read result data
12. Read plaintext authentication result

### 13.2.4.8 Counter With Cipher Block Chaining Message Authentication Code (CCM)

The CCM (Counter with CBC-MAC) protocol operation is a combined operation, consisting of encryption/decryption and authentication. Both the authentication and encryption/decryption operations use the crypto core; these are executed sequentially on the AES core. A part of the data stream can require authentication only. The authentication only data always needs to be in front of the data that requires encryption.

Figure 13-9 illustrates the CCM protocol. The authentication starts with the encryption of a pre-defined block B0. This block consists of flags, nonce and message length. The next blocks contain the authentication data length concatenated with the authentication only data. After processing the authentication only data, the encryption/decryption operations are performed, each followed by the related authentication of the plaintext data block (which equals the input in the case of encryption and the output in the case of decryption). The final authentication result needs to be encrypted using the output of the encryption of the IV block A0. This block contains the IV (consisting of flags and nonce) concatenated with the counter, which is zero for A0.



Copyright © 2026 Texas Instruments Incorporated

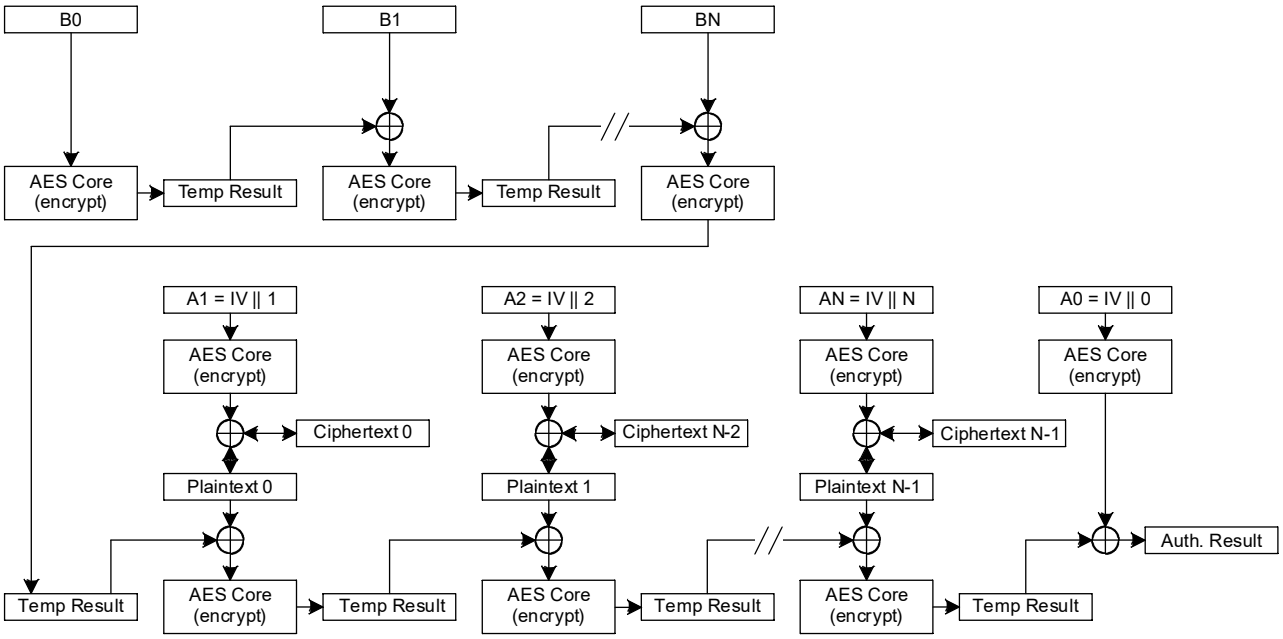



**Figure 13-9. CCM Protocol Operation**

Figure 13-10 shows one round of a CCM operation for both encryption and decryption. A 32-bit counter is used as IV (as it is for CTR mode). The data is encrypted in the same way as CTR mode, by XOR-ing the crypto core output with the input. Directly after the encrypt-operation, the plaintext is XOR-ed with the intermediate authentication result. The XOR result is used as input for a second encrypt-operation to calculate the next (intermediate) authentication result.



Copyright © 2026 Texas Instruments Incorporated

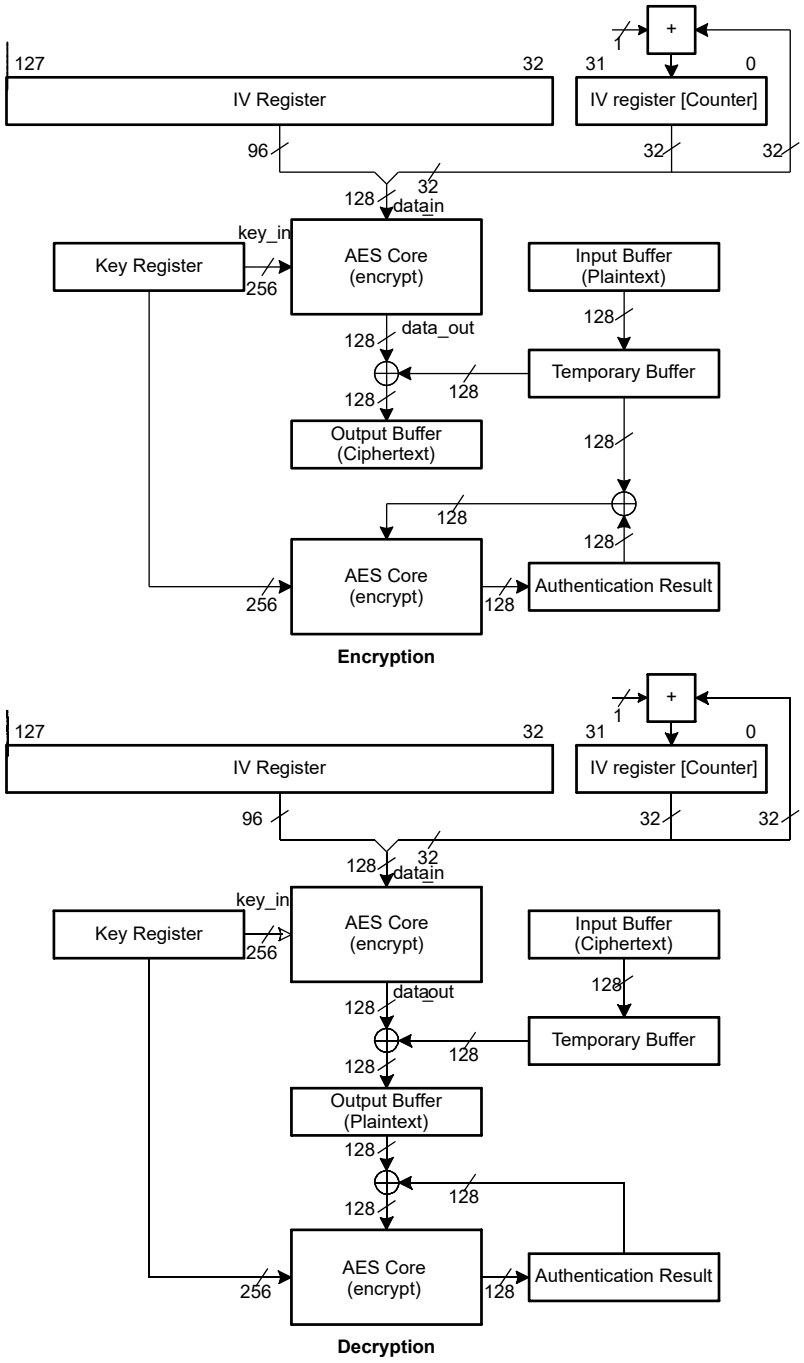



**Figure 13-10. CCM Operation**

#### *13.2.4.8.1 CCM Operation*

At the high-level, CCM operation is performed in the following sequence:

- Provide CCM context (key, IV including flags, lengths and mode)
- Provide hash only data
- Provide next hash only data
- Provide last hash only data
- Provide first crypto data
- Provide next crypto data



Copyright © 2026 Texas Instruments Incorporated

- Read result data and provide next crypto data
- ...
- Read result data and provide last crypto data
- Read result data
- Read result data
- Read authentication result (TAG)

The read/write of data can be managed via DMA or by CPU software.

To implement CCM encryption over N blocks of plaintext and M blocks of AAD, follow these steps:

1. Configure Output DMA channel for saving ciphertext:
   a. Set DMA channel trigger selection to AES Trig1
   b. Set DMA channel source address to DATA_OUT
   c. Set DMA channel destination address to location where ciphertext is to be stored (for example, SRAM)
   d. Set DMA channel transfer size to *N×4*
   e. Set DMA channel mode to single transfer mode
   f. In the AES event registers, unmask Trig1 in the IMASK register of DMA_TRIG_DATAOUT
2. Configure Input DMA channel for loading AAD and plaintext:
   a. Set DMA channel trigger selection to AES Trig0
   b. Set DMA channel source address to location where plaintext is stored (for example, SRAM)
   c. Set DMA channel destination address to DATA_IN
   d. Set DMA channel transfer size to *(N+M)*4*
   e. Set DMA channel mode to single transfer mode
   f. In the AES event registers, unmask Trig0 in the IMASK register of DMA_TRIG0
3. Configure and enable the DMA interrupt for the Output DMA channel in the DMA controller
4. Configure DMA_HS for DMA based handshake: set DMA_HS[DMA_DATA_ACK] = 1
5. Load Encryption/Decryption key as described in Section 13.2.1
6. Load Initialization vector (IV) by writing to IV0, IV1, IV2 and IV3 registers (must contain the flags for the cryptographic operation and the NONCE bytes, for both authentication and encryption)
7. Configure the CTRL register for block cipher encryption mode for CCM
   a. Select key size via CTRL[KEY_SIZ]
   b. Select Direction for Encryption by CTRL[DIR] = 1
   c. Select CCM mode by setting CTRL[CCM] = 1
   d. Select CTR mode by setting CTRL[CTR] =

      1
   e. Enabke saving of TAG by setting CTRL[SAVE_CNTXT] = 1
   f. Configure CTRL[CCML] -- CCM-L can be set to any value, representing a crypto data length field of CCML plus one Bytes.
   g. Configure CTRL[CTR_WIDTH] -- Note: CCM-L sets the actual counter field width in the IV register for CCM operations, in the range from 2 (CCM-L = 1) up to and including 8 (CCM-L = 7) Bytes. The actual counter width chosen with the CTR_WIDTH must be long enough to cover this field. The counter field width must be chosen so that the counter cannot overflow.
   h. Configure CTRL[CCMM] -- CCM-M can be set to any value and has no effect on the actual processing (other than being present in the special ‘B0’ block that is encrypted at the start of the operation). The CPU must select the valid TAG bytes from the 128-bit TAG (which are in the least significant 2 * (CCMM + 1) Bytes).
8. Write encryption/decryption byte count N×4 to AES C_LENGTH_0 and C_LENGTH_1 registers
9. Write authentication data (AAD) byte count M*4 to AES AAD_LENGTH register
10. Wait for the DMA output channel interrupt that indicates completion of the entire operation. The output is stored started at the location configured in step 1c.
11. Read out the final TAG from the TAG0/1/2/3 registers



Copyright © 2026 Texas Instruments Incorporated

**Note**

The AAD and cryptographic data can end misaligned. The CPU must pad both to a 128-bit boundary with zeroes. More formally, the AAD and crypto data padding must satisfy the bit string: 0n, with 0<= n <= 127. This means that the AAD must be provided as separate blocks to the engine, such that the cryptographic data starts 128-bit aligned. If the AAD/cryptographic data stream is 128-bit aligned, no padding is required. Because the engine only supports bytes, n must be such that (n MOD 8) = 0. Further, since a single DMA channel is used to supply both AAD and plaintext, the entire data must be organized contiguously in memory with the first M blocks of AAD followed by N blocks of plaintext. This memory contiguity restriction is not applicable when CPU software directly provides inputs via interrupt handling.

Do not write both length values with zeroes. If a data stream is done and the next data stream uses the same key and control, only the IV and length values need be re-loaded.

### *13.2.5 AES Events*

The AES module contains three event publishers and no event subscribers. One event publisher (CPU_INT) manages AES interrupt requests (IRQs) to the CPU subsystem through a static event route. The second, and third event publishers (DMA_TRIG_DATAIN, and DMA_TRIG_DATAOUT) can be used to publish AES events to the DMA through DMA event routes.

The AES events are summarized in Table 13-2.

**Table 13-2. AES Events**

| Event | Type | Source | Destination | Route | Configuration | Functionality |
|---|---|---|---|---|---|---|
| CPU Interrupt Event | Publisher | AES | CPU Subsystem | Static route | CPU_INT registers | Fixed interrupt route from RTC to CPU |
| DMA Trigger Event 0 | Publisher | AES | DMA | DMA route | DMA_TRIG 0 registers | DMA trigger 0: Data Input into engine |
| DMA Trigger Event 1 | Publisher | AES | DMA | DMA route | DMA_TRIG 1 registers | DMA trigger 1: Data Output from engine |

In general, the CPU interrupt event is used to communicate completion of an AES operation to the CPU, and the DMA triggers are used together to implement the block cipher modes (ECB, CBC, OFB, CFB, GCM/GMAC, CCM/CMAC) using the DMA together with the AES accelerator.

#### 13.2.5.1 CPU Interrupt Event Publisher (CPU_EVENT)

The AESADV module provides four interrupt sources which can be configured to source a CPU interrupt event. The CPU interrupt events from the AES are given inTable 13-3 .

**Table 13-3. AES CPU Interrupt Event Conditions (CPU_EVENT)**

| Index (IIDX) | Name | Description |
|---|---|---|
| 0 | NO_INTR | No interrupt pending. |
| 1 | OUTPUTRDY | This indicates that the engine has an output available to be read out. This should not be used if DMA handshake is used (AES_DMA_HS.DMA_DATA_ACK set to 1) |
| 2 | INPUTRDY | This indicates that the engine can take new input. This should not be used if DMA handshake is used (AES_DMA_HS.DMA_DATA_ACK set to 1) |
| 3 | SAVEDCNTXTRDY | This bit indicates that an AES authentication TAG and/or IV block(s) is/are available for the CPU to retrieve. This bit is only asserted if the ‘save_context’ bit is set to ‘1b’. The bit is mutually exclusive with the ‘context_ready’ bit. |



Copyright © 2026 Texas Instruments Incorporated

**Table 13-3. AES CPU Interrupt Event Conditions (CPU_EVENT) (continued)**

| Index (IIDX) | Name | Description |
|---|---|---|
| 4 | CNTXTRDY | This bit indicates that the context data registers can be overwritten, and the CPU is permitted to write new context. |

The CPU interrupt event configuration is managed with the event management registers. See *Using Event Registers* for guidance on configuring these registers for CPU interrupts.

#### 13.2.5.2 DMA Trigger Event Publisher (DMA_TRIG_DATAIN)

The AESADV module provides a trigger source which can be configured to source DMA trigger 0. The DMA0 trigger events from the AES are given in Table 13-4. When the DMA0 channel is needed by the AES for block cipher operations, the DMA0 trigger should be unmasked in the IMASK register of DMA_TRIG_DATAIN and the DMA should be configured as needed to support the AES operation.

**Table 13-4. AES DMA Trigger 0 Event Conditions (DMA_TRIG_DATAIN)**

| Index (IIDX) | Name | Description |
|---|---|---|
| 0 | NO_INTR | No DMA Trig0 event pending |
| 1 | TRIG0 | DMA Trigger for Data Input |

The DMA trigger 0 event configuration is managed with the DMA_TRIG_DATAIN event management registers. See *Using Event Registers* for guidance on configuring the event registers for DMA triggers.

#### 13.2.5.3 DMA Trigger Event Publisher (DMA_TRIG_DATAOUT)

The AES module provides a trigger source which can be configured to source DMA trigger 1. The DMA1 trigger events from the AES are given in Table 13-5. When the DMA1 channel is needed by the AES for block cipher operations, the DMA1 trigger should be unmasked in the IMASK register of DMA_TRIG_DATAOUT and the DMA should be configured as needed to support the AES operation.

**Table 13-5. AES DMA Trigger 1 Event Conditions (DMA_TRIG_DATAOUT)**

| Index (IIDX) | Name | Description |
|---|---|---|
| 0 | NO_INTR | No DMA Trig1 Event Pending |
| 1 | TRIG1 | DMA Trigger for Data Output |

The DMA trigger 1 event configuration is managed with the DMA_TRIG_DATAOUT event management registers. See *Using Event Registers* for guidance on configuring the event registers for DMA triggers.



Copyright © 2026 Texas Instruments Incorporated

## 13.3 AESADV Registers

Table 13-6 lists the memory-mapped registers for the AESADV registers. All register offset addresses not listed in Table 13-6 should be considered as reserved locations and the register contents should not be modified.

**Table 13-6. AESADV Registers**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 480h | CPU_CONNECT_0 | CPU Connect | | Go |
| 800h | PWREN | Power enable | | Go |
| 804h | RSTCTL | Reset Control | | Go |
| 814h | STAT | Status Register | | Go |
| 1018h | PDBGCTL | Peripheral Debug Control | | Go |
| 1020h | IIDX | Interrupt Index Register | | Go |
| 1028h | IMASK | Interrupt mask | | Go |
| 1030h | RIS | Raw interrupt status | | Go |
| 1038h | MIS | Masked interrupt status | | Go |
| 1040h | ISET | Interrupt set | | Go |
| 1048h | ICLR | Interrupt clear | | Go |
| 1050h | IIDX | Interrupt Index Register | DMA_TRIG_DATAIN | Go |
| 1058h | IMASK | Interrupt mask | DMA_TRIG_DATAIN | Go |
| 1060h | RIS | Raw interrupt status | DMA_TRIG_DATAIN | Go |
| 1068h | MIS | Masked interrupt status | DMA_TRIG_DATAIN | Go |
| 1070h | ISET | Interrupt set | DMA_TRIG_DATAIN | Go |
| 1078h | ICLR | Interrupt clear | DMA_TRIG_DATAIN | Go |
| 1080h | IIDX | Interrupt Index Register | DMA_TRIG_DATAOUT | Go |
| 1088h | IMASK | Interrupt mask | DMA_TRIG_DATAOUT | Go |
| 1090h | RIS | Raw interrupt status | DMA_TRIG_DATAOUT | Go |
| 1098h | MIS | Masked interrupt status | DMA_TRIG_DATAOUT | Go |
| 10A0h | ISET | Interrupt set | DMA_TRIG_DATAOUT | Go |
| 10A8h | ICLR | Interrupt clear | DMA_TRIG_DATAOUT | Go |
| 10E0h | EVT_MODE | Event Mode | | Go |
| 1100h | GCMCCM_TAG0 | CBC-MAC third key (LSW) / GCM & CCM Intermediate TAG (LSW) | | Go |
| 1104h | GCMCCM_TAG1 | CBC-MAC third key / GCM & CCM Intermediate TAG | | Go |
| 1108h | GCMCCM_TAG2 | CBC-MAC third key / GCM & CCM Intermediate TAG | | Go |
| 110Ch | GCMCCM_TAG3 | CBC-MAC third key (MSW) / GCM & CCM Intermediate TAG (MSW) | | Go |
| 1110h | GHASH_H0 | CCM & CBC-MAC second key (LSW) / GCM Hash Key input (LSW) | | Go |



Copyright © 2026 Texas Instruments Incorporated

**Table 13-6. AESADV Registers (continued)**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 1114h | GHASH_H1 | CCM & CBC-MAC second key / GCM Hash Key input | | Go |
| 1118h | GHASH_H2 | CCM & CBC-MAC second key / GCM Hash Key input | | Go |
| 111Ch | GHASH_H3 | CCM & CBC-MAC second key (MSW) / GCM Hash Key input (MSW) | | Go |
| 1120h | KEY0 | KEY (LSW) | | Go |
| 1124h | KEY1 | KEY | | Go |
| 1128h | KEY2 | KEY | | Go |
| 112Ch | KEY3 | KEY | | Go |
| 1130h | KEY4 | KEY | | Go |
| 1134h | KEY5 | KEY | | Go |
| 1138h | KEY6 | KEY | | Go |
| 113Ch | KEY7 | KEY (MSW) | | Go |
| 1140h | IV0 | IV (LSW) | | Go |
| 1144h | IV1 | IV | | Go |
| 1148h | IV2 | IV | | Go |
| 114Ch | IV3 | IV | | Go |
| 1150h | CTRL | Input/Output Buffer Control and Mode selection | | Go |
| 1154h | C_LENGTH_0 | Crypto data length (LSW) | | Go |
| 1158h | C_LENGTH_1 | Crypto data length (MSW) | | Go |
| 115Ch | AAD_LENGTH | AAD Data Length | | Go |
| 1160h | DATA0 | Data input (LSW) / Data output (LSW) | | Go |
| 1164h | DATA1 | Data input / Data output | | Go |
| 1168h | DATA2 | Data input / Data output | | Go |
| 116Ch | DATA3 | Data input (LSW) / Data output (MSW) | | Go |
| 1170h | TAG0 | Hash result (LSW) | | Go |
| 1174h | TAG1 | Hash result | | Go |
| 1178h | TAG2 | Hash result | | Go |
| 117Ch | TAG3 | Hash result (MSW) | | Go |
| 1180h | STATUS | Status | | Go |
| 1184h | DATA_IN | Data in alias register | | Go |
| 1188h | DATA_OUT | Data out alias register | | Go |
| 11D0h | FORCE_IN_AV | Data control register for input data | | Go |
| 11D4h | CCM_ALN_WRD | AES-CCM AAD alignment data word | | Go |
| 11D8h | BLK_CNT0 | Internal block counter (LSW) | | Go |
| 11DCh | BLK_CNT1 | Internal block counter (MSW) | | Go |
| 11F4h | DMA_HS | Control register for DMA handshaking | | Go |

Complex bit access types are encoded to fit into small table cells. Table 13-7 shows the codes that are used for access types in this section.

**Table 13-7. AESADV Access Type Codes**

| Access Type | Code | Description |
|---|---|---|
| Read Type | | |
| R | R | Read |



Copyright © 2026 Texas Instruments Incorporated

**Table 13-7. AESADV Access Type Codes (continued)**

| Access Type | Code | Description |
|---|---|---|
| Write Type | | |
| W | W | Write |
| WK | W<br>K | Write<br>Write protected by a key |
| Reset or Default Value | | |
| *-n* | | Value after reset or the default value |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.1 CPU_CONNECT_0 (Offset = 480h) [Reset = 00000000h]

CPU_CONNECT_0 is shown in Figure 13-11 and described in Table 13-8.

Return to the Summary Table.

Directly connect peripheral publisher port to application processor

**Figure 13-11. CPU_CONNECT_0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | CPUSS0_CONN | RESERVED |
| R-0h | | | | | | R/W-0h | R-0h |

**Table 13-8. CPU_CONNECT_0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | |
| 1 | CPUSS0_CONN | R/W | 0h | CPUSS0 connect bit.<br>0h = The CPU is not connected.<br>1h = The CPU is connected. |
| 0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.2 PWREN (Offset = 800h) [Reset = 00000000h]

PWREN is shown in Figure 13-12 and described in Table 13-9.

Return to the Summary Table.

Register to control the power state

**Figure 13-12. PWREN**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | ENABLE |
| R-0h | | | | | | | R/WK-0h |

**Table 13-9. PWREN Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | KEY to allow Power State Change<br>26h = KEY to allow write access to this register |
| 23-1 | RESERVED | R | 0h | |
| 0 | ENABLE | R/WK | 0h | Enable the power<br>KEY must be set to 26h to write to this bit.<br>0h = Disable Power<br>1h = Enable Power |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.3 RSTCTL (Offset = 804h) [Reset = 00000000h]

RSTCTL is shown in Figure 13-13 and described in Table 13-10.

Return to the Summary Table.

Register to control reset assertion and de-assertion

**Figure 13-13. RSTCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | RESETSTKYCLR | RESETASSERT |
| R-0h | | | | | | WK-0h | WK-0h |

**Table 13-10. RSTCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | Unlock key<br>B1h = KEY to allow write access to this register |
| 23-2 | RESERVED | R | 0h | |
| 1 | RESETSTKYCLR | WK | 0h | Clear the RESETSTKY bit in the STAT register<br>KEY must be set to B1h to write to this bit.<br>0h = Writing 0 has no effect<br>1h = Clear reset sticky bit |
| 0 | RESETASSERT | WK | 0h | Assert reset to the peripheral<br>KEY must be set to B1h to write to this bit.<br>0h = Writing 0 has no effect<br>1h = Assert reset |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.4 STAT (Offset = 814h) [Reset = 00000000h]

STAT is shown in Figure 13-14 and described in Table 13-11.

Return to the Summary Table.

peripheral enable and reset status

**Figure 13-14. STAT**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | RESETSTKY |
| R-0h | | | | | | | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |

**Table 13-11. STAT Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-17 | RESERVED | R | 0h | |
| 16 | RESETSTKY | R | 0h | This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register<br>0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register<br>1h = The peripheral was reset since the last bit clear |
| 15-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.5 PDBGCTL (Offset = 1018h) [Reset = 00000000h]

PDBGCTL is shown in Figure 13-15 and described in Table 13-12.

Return to the Summary Table.

AES can not be halted when the core is halted. In order to halt the AES, the DMA shall be halted.
This achieves the same effect as a halt feature in the AES: when the AES submits the next DMA trigger, if the DMA is halted, then the AES will automatically halt.

**Figure 13-15. PDBGCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | FREE |
| R-0h | | | | | | | R-0h |

**Table 13-12. PDBGCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | FREE | R | 0h | Free run control<br>1h = The peripheral ignores the state of the Core Halted input |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.6 IIDX (Offset = 1020h) [Reset = 00000000h]

IIDX is shown in Figure 13-16 and described in Table 13-13.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. Value 0x00 means no event pending. Interrupt 1 is the highest priority, IIDX next highest, 4, 8, … IIDX^31 is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred. On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in [RIS] and [MIS] are cleared as well. After a read from the CPU (not from the debug interface), the register is updated with the next highest priority interrupt, if none are pending, then it should display 0x0.

**Figure 13-16. IIDX**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | STAT | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | |

**Table 13-13. IIDX Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | STAT | R | 0h | Interrupt index status<br>00h = No interrupt pending<br>1h = This indicates that the core has an output available to be read out. This should not be used if DMA handshake is used (AES_DMA_HS.DMA_DATA_ACK set to 1)<br>2h = This indicates that the engine can take new input. This should not be used if DMA handshake is used (AES_DMA_HS.DMA_DATA_ACK set to 1)<br>3h = This bit indicates that an AES authentication TAG and/or IV block(s) is/are available for the CPU to retrieve. This bit is only asserted if the ‘save_context’ bit is set to ‘1b’. The bit is mutually exclusive with the ‘context_ready’ bit.<br>4h = This bit indicates that the context data registers can be overwritten, and the CPU is permitted to write new context. |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.7 IMASK (Offset = 1028h) [Reset = 00000000h]

IMASK is shown in Figure 13-17 and described in Table 13-14.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is unmasked. Unmasking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

**Figure 13-17. IMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | CNTXTRDY | SAVEDCNTXTRDY | INPUTRDY | OUTPUTRDY |
| R-0h | | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

**Table 13-14. IMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | CNTXTRDY | R/W | 0h | This bit indicates that the context data registers can be overwritten, and the CPU is permitted to write next context.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 2 | SAVEDCNTXTRDY | R/W | 0h | This bit indicates that an AES authentication TAG and/or IV block(s) is/are available for the CPU to retrieve. This bit is only asserted if the ‘save_context’ bit is set to ‘1b’. The bit is mutually exclusive with the ‘context_ready’ bit.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 1 | INPUTRDY | R/W | 0h | This indicates that the engine can take new input. This should not be used if DMA handshake is used (AES_DMA_HS.DMA_DATA_ACK set to 1)<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 0 | OUTPUTRDY | R/W | 0h | This indicates that the core has an output available to be read out. This should not be used if DMA handshake is used (AES_DMA_HS.DMA_DATA_ACK set to 1)<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.8 RIS (Offset = 1030h) [Reset = 00000000h]

RIS is shown in Figure 13-18 and described in Table 13-15.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Figure 13-18. RIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | CNTXTRDY | SAVEDCNTXTRDY | INPUTRDY | OUTPUTRDY |
| R-0h | | | | R-0h | R-0h | R-0h | R-0h |

**Table 13-15. RIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | CNTXTRDY | R | 0h | This bit indicates that the context data registers can be overwritten, and the CPU is permitted to write next context.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 2 | SAVEDCNTXTRDY | R | 0h | This bit indicates that an AES authentication TAG and/or IV block(s) is/are available for the CPU to retrieve. This bit is only asserted if the ‘save_context’ bit is set to ‘1b’. The bit is mutually exclusive with the ‘context_ready’ bit.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 1 | INPUTRDY | R | 0h | This indicates that the engine can take new input. This should not be used if DMA handshake is used (AES_DMA_HS.DMA_DATA_ACK set to 1)<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 0 | OUTPUTRDY | R | 0h | This indicates that the core has an output available to be read out. This should not be used if DMA handshake is used (AES_DMA_HS.DMA_DATA_ACK set to 1)<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.9 MIS (Offset = 1038h) [Reset = 00000000h]

MIS is shown in Figure 13-19 and described in Table 13-16.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 13-19. MIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | CNTXTRDY | SAVEDCNTXTRDY | INPUTRDY | OUTPUTRDY |
| R-0h | | | | R-0h | R-0h | R-0h | R-0h |

**Table 13-16. MIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | CNTXTRDY | R | 0h | This bit indicates that the context data registers can be overwritten, and the CPU is permitted to write next context.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 2 | SAVEDCNTXTRDY | R | 0h | This bit indicates that an AES authentication TAG and/or IV block(s) is/are available for the CPU to retrieve. This bit is only asserted if the ‘save_context’ bit is set to ‘1b’. The bit is mutually exclusive with the ‘context_ready’ bit.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 1 | INPUTRDY | R | 0h | This indicates that the engine can take new input. This should not be used if DMA handshake is used (AES_DMA_HS.DMA_DATA_ACK set to 1)<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 0 | OUTPUTRDY | R | 0h | This indicates that the core has an output available to be read out. This should not be used if DMA handshake is used (AES_DMA_HS.DMA_DATA_ACK set to 1)<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.10 ISET (Offset = 1040h) [Reset = 00000000h]

ISET is shown in Figure 13-20 and described in Table 13-17.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 13-20. ISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | CNTXTRDY | SAVEDCNTXTRDY | INPUTRDY | OUTPUTRDY |
| R-0h | | | | W-0h | W-0h | W-0h | W-0h |

**Table 13-17. ISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | CNTXTRDY | W | 0h | This bit indicates that the context data registers can be overwritten, and the CPU is permitted to write next context.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 2 | SAVEDCNTXTRDY | W | 0h | This bit indicates that an AES authentication TAG and/or IV block(s) is/are available for the CPU to retrieve. This bit is only asserted if the ‘save_context’ bit is set to ‘1b’. The bit is mutually exclusive with the ‘context_ready’ bit.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 1 | INPUTRDY | W | 0h | This indicates that the engine can take new input. This should not be used if DMA handshake is used (AES_DMA_HS.DMA_DATA_ACK set to 1)<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 0 | OUTPUTRDY | W | 0h | This indicates that the core has an output available to be read out. This should not be used if DMA handshake is used (AES_DMA_HS.DMA_DATA_ACK set to 1)<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.11 ICLR (Offset = 1048h) [Reset = 00000000h]

ICLR is shown in Figure 13-21 and described in Table 13-18.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

**Figure 13-21. ICLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | CNTXTRDY | SAVEDCNTXTRDY | INPUTRDY | OUTPUTRDY |
| R-0h | | | | W-0h | W-0h | W-0h | W-0h |

**Table 13-18. ICLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | CNTXTRDY | W | 0h | This bit indicates that the context data registers can be overwritten, and the CPU is permitted to write next context.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 2 | SAVEDCNTXTRDY | W | 0h | This bit indicates that an AES authentication TAG and/or IV block(s) is/are available for the CPU to retrieve. This bit is only asserted if the ‘save_context’ bit is set to ‘1b’. The bit is mutually exclusive with the ‘context_ready’ bit.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 1 | INPUTRDY | W | 0h | This indicates that the engine can take new input. This should not be used if DMA handshake is used (AES_DMA_HS.DMA_DATA_ACK set to 1)<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 0 | OUTPUTRDY | W | 0h | This indicates that the core has an output available to be read out. This should not be used if DMA handshake is used (AES_DMA_HS.DMA_DATA_ACK set to 1)<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.12 IIDX (Offset = 1050h) [Reset = 00000000h]

IIDX is shown in Figure 13-22 and described in Table 13-19.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. Value 0x00 means no event pending. Interrupt 1 is the highest priority, IIDX next highest, 4, 8, … IIDX^31 is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred. On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in [RIS] and [MIS] are cleared as well. After a read from the CPU (not from the debug interface), the register is updated with the next highest priority interrupt, if none are pending, then it should display 0x0.

**Figure 13-22. IIDX**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | STAT | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | |

**Table 13-19. IIDX Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | STAT | R | 0h | Interrupt index status<br>00h = No interrupt pending<br>1h = AES trigger 0 DMA (Data Input trigger) |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.13 IMASK (Offset = 1058h) [Reset = 00000000h]

IMASK is shown in Figure 13-23 and described in Table 13-20.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is unmasked. Unmasking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

**Figure 13-23. IMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | TRIG0 |
| R-0h | | | | | | | R/W-0h |

**Table 13-20. IMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | TRIG0 | R/W | 0h | TRIG0 event mask.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.14 RIS (Offset = 1060h) [Reset = 00000000h]

RIS is shown in Figure 13-24 and described in Table 13-21.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Figure 13-24. RIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | TRIG0 |
| R-0h | | | | | | | R-0h |

**Table 13-21. RIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | TRIG0 | R | 0h | TRIG0 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.15 MIS (Offset = 1068h) [Reset = 00000000h]

MIS is shown in Figure 13-25 and described in Table 13-22.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 13-25. MIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | TRIG0 |
| R-0h | | | | | | | R-0h |

**Table 13-22. MIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | TRIG0 | R | 0h | TRIG0 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.16 ISET (Offset = 1070h) [Reset = 00000000h]

ISET is shown in Figure 13-26 and described in Table 13-23.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 13-26. ISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | TRIG0 |
| R-0h | | | | | | | W-0h |

**Table 13-23. ISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | TRIG0 | W | 0h | TRIG0<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.17 ICLR (Offset = 1078h) [Reset = 00000000h]

ICLR is shown in Figure 13-27 and described in Table 13-24.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

**Figure 13-27. ICLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | TRIG0 |
| R-0h | | | | | | | W-0h |

**Table 13-24. ICLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | TRIG0 | W | 0h | TRIG0 event<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.18 IIDX (Offset = 1080h) [Reset = 00000000h]

IIDX is shown in Figure 13-28 and described in Table 13-25.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. Value 0x00 means no event pending. Interrupt 1 is the highest priority, IIDX next highest, 4, 8, … IIDX^31 is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred. On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in [RIS] and [MIS] are cleared as well. After a read from the CPU (not from the debug interface), the register is updated with the next highest priority interrupt, if none are pending, then it should display 0x0.

**Figure 13-28. IIDX**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | STAT | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | |

**Table 13-25. IIDX Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | STAT | R | 0h | Interrupt index status<br>00h = No interrupt pending<br>1h = AES DMA Trigger 1 (Data Output trigger) |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.19 IMASK (Offset = 1088h) [Reset = 00000000h]

IMASK is shown in Figure 13-29 and described in Table 13-26.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is unmasked. Unmasking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

**Figure 13-29. IMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | TRIG1 |
| R-0h | | | | | | | R/W-0h |

**Table 13-26. IMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | TRIG1 | R/W | 0h | TRIG1 event mask.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.20 RIS (Offset = 1090h) [Reset = 00000000h]

RIS is shown in Figure 13-30 and described in Table 13-27.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Figure 13-30. RIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | TRIG1 |
| R-0h | | | | | | | R-0h |

**Table 13-27. RIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | TRIG1 | R | 0h | TRIG1 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.21 MIS (Offset = 1098h) [Reset = 00000000h]

MIS is shown in Figure 13-31 and described in Table 13-28.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 13-31. MIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | TRIG1 |
| R-0h | | | | | | | R-0h |

**Table 13-28. MIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | TRIG1 | R | 0h | TRIG1 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.22 ISET (Offset = 10A0h) [Reset = 00000000h]

ISET is shown in Figure 13-32 and described in Table 13-29.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 13-32. ISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | TRIG1 |
| R-0h | | | | | | | W-0h |

**Table 13-29. ISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | TRIG1 | W | 0h | TRIG1 event<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.23 ICLR (Offset = 10A8h) [Reset = 00000000h]

ICLR is shown in Figure 13-33 and described in Table 13-30.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

**Figure 13-33. ICLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | TRIG1 |
| R-0h | | | | | | | W-0h |

**Table 13-30. ICLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | TRIG1 | W | 0h | TRIG1 event<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.24 EVT_MODE (Offset = 10E0h) [Reset = 00000000h]

EVT_MODE is shown in Figure 13-34 and described in Table 13-31.

Return to the Summary Table.

Event mode register. It is used to select whether each line is disabled, in software mode (software clears the RIS) or in hardware mode (hardware clears the RIS)

**Figure 13-34. EVT_MODE**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | EVT2_CFG | | EVT1_CFG | | INT0_CFG | |
| R-0h | | R-0h | | R-0h | | R-0h | |

**Table 13-31. EVT_MODE Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | RESERVED | R | 0h | |
| 5-4 | EVT2_CFG | R | 0h | Event line mode select for event corresponding to INT_EVENT2<br>0h = The interrupt or event line is disabled.<br>1h = The interrupt or event line is in software mode. Software must clear the RIS.<br>2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag. |
| 3-2 | EVT1_CFG | R | 0h | Event line mode select for event corresponding to INT_EVENT1<br>0h = The interrupt or event line is disabled.<br>1h = The interrupt or event line is in software mode. Software must clear the RIS.<br>2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag. |
| 1-0 | INT0_CFG | R | 0h | Event line mode select for event corresponding to INT_EVENT0<br>0h = The interrupt or event line is disabled.<br>1h = The interrupt or event line is in software mode. Software must clear the RIS.<br>2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag. |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.25 GCMCCM_TAG0 (Offset = 1100h) [Reset = 00000000h]

GCMCCM_TAG0 is shown in Figure 13-35 and described in Table 13-32.

Return to the Summary Table.

CBC-MAC third key (LSW) / GCM & CCM Intermediate TAG (LSW)
For CBC-MAC: Pre-calculated CBC-MAC third key used to perform a final XOR operation on the last input data block.
For CCM: This register is internally used to store intermediate values. This register must be initialized with zeroes when writing a new CCM context. To restore an interrupted CCM operation, this register needs to be loaded with the intermediate TAG.
For GCM: This register is internally used to store intermediate values. This register must be initialized with zeroes when writing a new GCM context. To restore an interrupted GCM operation, this register needs to be loaded with the intermediate TAG.

**Figure 13-35. GCMCCM_TAG0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 13-32. GCMCCM_TAG0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | W | 0h | Key data |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.26 GCMCCM_TAG1 (Offset = 1104h) [Reset = 00000000h]

GCMCCM_TAG1 is shown in Figure 13-36 and described in Table 13-33.

Return to the Summary Table.

CBC-MAC third key / GCM & CCM Intermediate TAG
For CBC-MAC: Pre-calculated CBC-MAC third key used to perform a final XOR operation on the last input data block.
For CCM: This register is internally used to store intermediate values. This register must be initialized with zeroes when writing a new CCM context. To restore an interrupted CCM operation, this register needs to be loaded with the intermediate TAG.
For GCM: This register is internally used to store intermediate values. This register must be initialized with zeroes when writing a new GCM context. To restore an interrupted GCM operation, this register needs to be loaded with the intermediate TAG.

**Figure 13-36. GCMCCM_TAG1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 13-33. GCMCCM_TAG1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | W | 0h | Key data |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.27 GCMCCM_TAG2 (Offset = 1108h) [Reset = 00000000h]

GCMCCM_TAG2 is shown in Figure 13-37 and described in Table 13-34.

Return to the Summary Table.

CBC-MAC third key / GCM & CCM Intermediate TAG
For CBC-MAC: Pre-calculated CBC-MAC third key used to perform a final XOR operation on the last input data block.
For CCM: This register is internally used to store intermediate values. This register must be initialized with zeroes when writing a new CCM context. To restore an interrupted CCM operation, this register needs to be loaded with the intermediate TAG.
For GCM: This register is internally used to store intermediate values. This register must be initialized with zeroes when writing a new GCM context. To restore an interrupted GCM operation, this register needs to be loaded with the intermediate TAG.

**Figure 13-37. GCMCCM_TAG2**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 13-34. GCMCCM_TAG2 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | W | 0h | Key data |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.28 GCMCCM_TAG3 (Offset = 110Ch) [Reset = 00000000h]

GCMCCM_TAG3 is shown in Figure 13-38 and described in Table 13-35.

Return to the Summary Table.

CBC-MAC third key (MSW) / GCM & CCM Intermediate TAG (MSW)
For CBC-MAC: Pre-calculated CBC-MAC third key used to perform a final XOR operation on the last input data block.
For CCM: This register is internally used to store intermediate values. This register must be initialized with zeroes when writing a new CCM context. To restore an interrupted CCM operation, this register needs to be loaded with the intermediate TAG.
For GCM: This register is internally used to store intermediate values. This register must be initialized with zeroes when writing a new GCM context. To restore an interrupted GCM operation, this register needs to be loaded with the intermediate TAG.

**Figure 13-38. GCMCCM_TAG3**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 13-35. GCMCCM_TAG3 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | W | 0h | Key data |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.29 GHASH_H0 (Offset = 1110h) [Reset = 00000000h]

GHASH_H0 is shown in Figure 13-39 and described in Table 13-36.

Return to the Summary Table.

CCM & CBC-MAC second key (LSW) / GCM Hash Key input (LSW)
For CBC-MAC: Pre-calculated CBC-MAC second key used to perform a final XOR operation on the last input data block.
For GCM: Hash key; can be calculated internal or written via these registers. Only used for GHASH (GCM) modes.
For a CPU write operation, these registers must be written with the new values to be subsequently transferred to the engine.

**Figure 13-39. GHASH_H0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 13-36. GHASH_H0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | W | 0h | Key data |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.30 GHASH_H1 (Offset = 1114h) [Reset = 00000000h]

GHASH_H1 is shown in Figure 13-40 and described in Table 13-37.

Return to the Summary Table.

CCM & CBC-MAC second key / GCM Hash Key input
For CBC-MAC: Pre-calculated CBC-MAC second key used to perform a final XOR operation on the last input data block.
For GCM: Hash key; can be calculated internal or written via these registers. Only used for GHASH (GCM) modes.
For a CPU write operation, these registers must be written with the new values to be subsequently transferred to the engine.

**Figure 13-40. GHASH_H1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 13-37. GHASH_H1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | W | 0h | Key data |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.31 GHASH_H2 (Offset = 1118h) [Reset = 00000000h]

GHASH_H2 is shown in Figure 13-41 and described in Table 13-38.

Return to the Summary Table.

CCM & CBC-MAC second key / GCM Hash Key input
For CBC-MAC: Pre-calculated CBC-MAC second key used to perform a final XOR operation on the last input data block.
For GCM: Hash key; can be calculated internal or written via these registers. Only used for GHASH (GCM) modes.
For a CPU write operation, these registers must be written with the new values to be subsequently transferred to the engine.

**Figure 13-41. GHASH_H2**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 13-38. GHASH_H2 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | W | 0h | Key data |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.32 GHASH_H3 (Offset = 111Ch) [Reset = 00000000h]

GHASH_H3 is shown in Figure 13-42 and described in Table 13-39.

Return to the Summary Table.

CCM & CBC-MAC second key (MSW) / GCM Hash Key input (MSW)
For CBC-MAC: Pre-calculated CBC-MAC second key used to perform a final XOR operation on the last input data block.
For GCM: Hash key; can be calculated internal or written via these registers. Only used for GHASH (GCM) modes.
For a CPU write operation, these registers must be written with the new values to be subsequently transferred to the engine.

**Figure 13-42. GHASH_H3**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 13-39. GHASH_H3 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | W | 0h | Key data |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.33 KEY0 (Offset = 1120h) [Reset = 00000000h]

KEY0 is shown in Figure 13-43 and described in Table 13-40.

Return to the Summary Table.

KEY (LSW)

**Figure 13-43. KEY0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 13-40. KEY0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | W | 0h | Key data |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.34 KEY1 (Offset = 1124h) [Reset = 00000000h]

KEY1 is shown in Figure 13-44 and described in Table 13-41.

Return to the Summary Table.

KEY

**Figure 13-44. KEY1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 13-41. KEY1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | W | 0h | Key data |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.35 KEY2 (Offset = 1128h) [Reset = 00000000h]

KEY2 is shown in Figure 13-45 and described in Table 13-42.

Return to the Summary Table.

KEY

**Figure 13-45. KEY2**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 13-42. KEY2 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | W | 0h | Key data |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.36 KEY3 (Offset = 112Ch) [Reset = 00000000h]

KEY3 is shown in Figure 13-46 and described in Table 13-43.

Return to the Summary Table.

KEY

**Figure 13-46. KEY3**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 13-43. KEY3 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | W | 0h | Key data |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.37 KEY4 (Offset = 1130h) [Reset = 00000000h]

KEY4 is shown in Figure 13-47 and described in Table 13-44.

Return to the Summary Table.

KEY

**Figure 13-47. KEY4**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 13-44. KEY4 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | W | 0h | Key data |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.38 KEY5 (Offset = 1134h) [Reset = 00000000h]

KEY5 is shown in Figure 13-48 and described in Table 13-45.

Return to the Summary Table.

KEY

**Figure 13-48. KEY5**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 13-45. KEY5 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | W | 0h | Key data |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.39 KEY6 (Offset = 1138h) [Reset = 00000000h]

KEY6 is shown in Figure 13-49 and described in Table 13-46.

Return to the Summary Table.

KEY

**Figure 13-49. KEY6**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 13-46. KEY6 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | W | 0h | Key data |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.40 KEY7 (Offset = 113Ch) [Reset = 00000000h]

KEY7 is shown in Figure 13-50 and described in Table 13-47.

Return to the Summary Table.

KEY (MSW)

**Figure 13-50. KEY7**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 13-47. KEY7 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | W | 0h | Key data |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.41 IV0 (Offset = 1140h) [Reset = 00000000h]

IV0 is shown in Figure 13-51 and described in Table 13-48.

Return to the Summary Table.

IV (LSW)

**Figure 13-51. IV0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 13-48. IV0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | W | 0h | Key data |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.42 IV1 (Offset = 1144h) [Reset = 00000000h]

IV1 is shown in Figure 13-52 and described in Table 13-49.

Return to the Summary Table.

IV

**Figure 13-52. IV1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 13-49. IV1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | W | 0h | Key data |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.43 IV2 (Offset = 1148h) [Reset = 00000000h]

IV2 is shown in Figure 13-53 and described in Table 13-50.

Return to the Summary Table.

IV

**Figure 13-53. IV2**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 13-50. IV2 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | W | 0h | Key data |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.44 IV3 (Offset = 114Ch) [Reset = 00000000h]

IV3 is shown in Figure 13-54 and described in Table 13-51.

Return to the Summary Table.

IV

**Figure 13-54. IV3**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 13-51. IV3 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | W | 0h | Key data |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.45 CTRL (Offset = 1150h) [Reset = 80000000h]

CTRL is shown in Figure 13-55 and described in Table 13-52.

Return to the Summary Table.

Input/Output Buffer Control and Mode selection. The content of this register determines the mode of operation of the engine.

**Figure 13-55. CTRL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CNTXT_RDY | SAVED_CNTXT_RDY | SAVE_CNTXT | GCM_CONT | GET_DIGEST | OFB_GCM_CCM_CONT | RESERVED | CCMM |
| R-1h | R-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CCMM | | CCML | | | CCM | GCM | |
| R/W-0h | | R/W-0h | | | R/W-0h | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CBCMAC | RESERVED | | | | CFB | ICM | CTR_WIDTH |
| R/W-0h | R-0h | | | | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CTR_WIDTH | CTR | CBC | KEYSIZE | | DIR | INPUT_RDY | OUTPUT_RDY |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | | R/W-0h | R-0h | R-0h |

**Table 13-52. CTRL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | CNTXT_RDY | R | 1h | If ‘1b’, this read-only status bit indicates that the context data registers can be overwritten, and the CPU is permitted to write the next context.<br>0h = Not ready<br>1h = Ready |
| 30 | SAVED_CNTXT_RDY | R | 0h | If ‘1b’, this read-only status bit indicates that an AES authentication TAG and/or IV block(s) is/are available for the Host to retrieve. This bit is only asserted if the ‘save_context’ bit is set to ‘1b’. The bit is mutually exclusive with the ‘context_ready’ bit.<br>0h = Not ready<br>1h = Ready |
| 29 | SAVE_CNTXT | R/W | 0h | This bit is used to indicate that an authentication TAG or result IV needs to be stored as a result context. If this bit is set, context output DMA and/or interrupt will be asserted if the operation is finished, and related signals are enabled.<br>Typically, this value must be set for authentication modes returning a TAG (CBC-MAC, GCM and CCM), or for basic encryption modes that require future continuation with the current result IV.<br>If this bit is set, the engine will hold its full context until the TAG and/or IV registers are read. Only after reading the TAG or IV, a new DMA request for a new (input) context will be asserted.<br>If this bit is not set, the engine will assert the context input DMA request signal directly after starting to process the last block with the current context.<br>0h = No effect<br>1h = Enable |



Copyright © 2026 Texas Instruments Incorporated

**Table 13-52. CTRL Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 28 | GCM_CONT | R/W | 0h | Continue processing of an interrupted AES-GCM or AES-CCM operation in the crypto/payload phase.<br>Set this write-only signal to ‘1b’ together with the regular mode bit settings for a GCM or CCM operation, to continue processing from the next full block (128 bits) boundary.<br>Before setting this bit all applicable context to resume processing must have been loaded into the engine: Keys, IV, intermediate digest/TAG and block counter. The mode can be written together with this bit, as it is part of the same register.<br>0h = No effect<br>1h = Enable |
| 27 | GET_DIGEST | R/W | 0h | Interrupt processing and generate an intermediate digest during an AES-GCM or AES-CCM operation.<br>Set this write-only signal to ‘1b’ to interrupt GCM or CCM processing at the next full block (128 bits) boundary. An intermediate digest may be requested during the encryption/decryption data phase or in the AAD phase.<br>Note: Interruption can only be done on full block (128 bits) boundaries. The minimum number of remaining bytes to resume and finalize the operation, must be greater than or equal to 1.<br>0h = No effect<br>1h = Enable |
| 26 | OFB_GCM_CCM_CONT | R/W | 0h | This bit has a dual use, depending on the selection of CCM/GCM, see bits [18:16].<br>If CCM/GCM is not selected:<br>If this bit is set to ‘1b’, full block AES output feedback mode (OFB-128) is selected.<br>If CCM/GCM is selected:<br>Continue processing of an interrupted AES-GCM or AES-CCM operation in the AAD phase.<br>Set this write-only signal to ‘1b’ together with the regular mode bit settings for a GCM or CCM operation, to continue processing from the next full AAD block (128 bits) boundary.<br>Before setting this bit all applicable context to resume processing must have been loaded into the engine: Keys, IV, intermediate digest/TAG, block counter and the CCM align data word (the latter is for CCM mode only). The mode can be written together with this bit, as it is part of the same register.<br>1h = Continue GCM/CCM processing in AAD phase |
| 25 | RESERVED | R | 0h | |
| 24-22 | CCMM | R/W | 0h | Defines “M” that indicates the length of the authentication field for CCM operations; the authentication field length equals two times (the value of CCM-M plus one). Note: The engine always returns a 128-bit authentication field, of which the M least significant bytes are valid. All values are supported.<br>0h = Length is 1<br>7h = Length is 8 |
| 21-19 | CCML | R/W | 0h | Defines “L” that indicates the width of the length field for CCM operations; the length field in bytes equals the value of CMM-L plus one. All values are supported.<br>0h = Length is 1<br>7h = Length is 8 |
| 18 | CCM | R/W | 0h | If set to ‘1b’, AES-CCM is selected, this is a combined mode, using AES for both authentication and encryption. In addition to the CCM bit, the CTR mode bit must be set such that AES-CTR is enabled. Other combinations with CCM are invalid.<br>0h = Disable CBC mode<br>1h = Select CBC mode |



Copyright © 2026 Texas Instruments Incorporated

**Table 13-52. CTRL Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 17-16 | GCM | R/W | 0h | If not set to ‘00b’, AES-GCM mode is selected, this is a combined mode, using the Galois field multiplier GF(2128) for authentication and AES-CTR mode for encryption, the bits specify the GCM mode: 01b = GHASH with H loaded and Y0-encrypted forced to zero 10b = GHASH with H loaded and Y0-encrypted calculated internally 11b = Autonomous GHASH (both H and Y0-encrypted calculated internally) Note: Besides GCM, the CTR mode bits must also be set to ‘1b’ to enable GCM with AES-CTR; if the CTR bit is not set a GHASH (authentication) only operation is performed. A GHASH only operation is only allowed if the GCM mode is set to '01b' and the direction bit is set to '0b'. Other modes may not be selected in combination with GCM. Table 14 below shows the valid combinations for the GCM and CTR mode bits, all other options are invalid and must not be selected.<br>1h = GHASH with H loaded and Y0-encrypted forced to 0.<br>2h = GHASH with H loaded and Y0-encrypted calculated internally<br>3h = Autonomous GHASH (both H and Y0-encrypted calculated internally) |
| 15 | CBCMAC | R/W | 0h | If set to ‘1b’, AES-CBC MAC is selected, the Direction bit must be set to ‘1’ for this mode.<br>0h = Disable CBC mode<br>1h = Select CBC mode |
| 14-11 | RESERVED | R | 0h | |
| 10 | CFB | R/W | 0h | If set to ‘1b’, AES cipher feedback mode CFB is selected. Use the ctr_width field to specify the feedback width.<br>0h = Disable CBC mode<br>1h = Select CBC mode |
| 9 | ICM | R/W | 0h | When the CFB bit is set, specifies the CFB mode feedback width:<br>0h = Disable CBC mode<br>1h = Select CBC mode |
| 8-7 | CTR_WIDTH | R/W | 0h | When the CTR bit is set, specifies the counter width for AES-CTR mode.<br>When the CFB bit is set, specifies the CFB mode feedback width:<br>0h = CFB-128 mode<br>1h = CFB-1 mode<br>2h = CFB-8 mode<br>3h = 128-bit counter |
| 6 | CTR | R/W | 0h | If set to ‘1b’, AES counter mode (CTR) is selected. Note: This bit must also be set for GCM and CCM, when encryption/decryption is required.<br>0h = Disable CBC mode<br>1h = Select CBC mode |
| 5 | CBC | R/W | 0h | If set to ‘1b’, cipher-block-chaining (CBC) mode is selected.<br>0h = Disable CBC mode<br>1h = Select CBC mode |
| 4-3 | KEYSIZE | R/W | 0h | Specifies the encryption strength / key width<br>1h = 128-bit key<br>3h = 256-bit key |
| 2 | DIR | R/W | 0h | Direction. If set to ‘1b’ an encrypt operation is performed. If set to ‘0b’ a decrypt operation is performed. Note: This bit must be written with a ‘1b’ when CBC-MAC is selected.<br>0h = Decryption<br>1h = Encryption |
| 1 | INPUT_RDY | R | 0h | Ready for input. If ‘1b’, this read-only status bit indicates that the 16-byte input buffer is empty, and the CPU is permitted to write the next block of data. After reset, this bit is ‘0’. After writing a context, this bit will become ‘1b’.<br>0h = Not Ready<br>1h = Ready |



Copyright © 2026 Texas Instruments Incorporated

**Table 13-52. CTRL Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 0 | OUTPUT_RDY | R | 0h | Output Ready. If ‘1b’, this read-only status bit indicates that an AES output block is available for the CPU to retrieve.<br>0h = Not Ready<br>1h = Ready |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.46 C_LENGTH_0 (Offset = 1154h) [Reset = 00000000h]

C_LENGTH_0 is shown in Figure 13-56 and described in Table 13-53.

Return to the Summary Table.

Crypto data length (LSW). These registers buffer the Length values to the engine. While processing, the length values decrement to zero. If both lengths are zero, the data stream is finished, and a new context is requested. For basic AES modes (ECB/CBC/CTR/ICM/CFB/OFB), a crypto length of '0' can be written if the context DMA is disabled. Writing a zero length results in continued data requests until a new context is written. For the other modes (GCM and CCM) no (new) data requests are done if the length decrements to or equals zero.
It is advised to write a new length per packet. If the length registers decrement to zero, no new data is processed until a new context or length value is written.
When writing a new context without writing the length registers, the length register values from the previous context are reused.

**Figure 13-56. C_LENGTH_0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 13-53. C_LENGTH_0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | W | 0h | Bits [60:0] of the crypto length registers (LSW and MSW) store the cryptographic data length in bytes for all modes. Once processing with this context is started, this length decrements to zero. Data lengths up to (261-1) bytes are allowed.<br>For GCM, any value up to 236-32 bytes can be used. This is because a 32-bit counter mode is used; the maximum number of 128-bit blocks is 232-2, resulting in a maximum number of bytes of 236-32.<br>A write to this register triggers the engine to start using this context. This is valid for all modes except GCM and CCM.<br>Note that for the combined modes, this length does not include the authentication only data; the authentication length is specified in the AES_AAD_LENGTH register below.<br>All modes must have a length > 0. For the combined modes, it is allowed to have one of the lengths equal to zero.<br>For the basic encryption modes (ECB/CBC/CTR/ICM/CFB/OFB) it is allowed to program zero to the length field; in that case the length is assumed infinite.<br>All data must be byte (8-bit) aligned for stream cipher modes; bit aligned data streams are not supported. For block cipher modes, the data length must be programmed in multiples of the block cipher size, 16 bytes. |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.47 C_LENGTH_1 (Offset = 1158h) [Reset = 00000000h]

C_LENGTH_1 is shown in Figure 13-57 and described in Table 13-54.

Return to the Summary Table.

Crypto data length (MSW). These registers buffer the Length values to the engine. While processing, the length values decrement to zero. If both lengths are zero, the data stream is finished, and a new context is requested. For basic AES modes (ECB/CBC/CTR/ICM/CFB/OFB), a crypto length of '0' can be written if the context DMA is disabled. Writing a zero length results in continued data requests until a new context is written. For the other modes (GCM and CCM) no (new) data requests are done if the length decrements to or equals zero.
It is advised to write a new length per packet. If the length registers decrement to zero, no new data is processed until a new context or length value is written.
When writing a new context without writing the length registers, the length register values from the previous context are reused.

**Figure 13-57. C_LENGTH_1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 13-54. C_LENGTH_1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | RESERVED | R | 0h | |
| 28-0 | DATA | W | 0h | Bits [60:0] of the crypto length registers (LSW and MSW) store the cryptographic data length in bytes for all modes. Once processing with this context is started, this length decrements to zero. Data lengths up to (261-1) bytes are allowed.<br>For GCM, any value up to 236-32 bytes can be used. This is because a 32-bit counter mode is used; the maximum number of 128-bit blocks is 232-2, resulting in a maximum number of bytes of 236-32.<br>A write to this register triggers the engine to start using this context. This is valid for all modes except GCM and CCM.<br>Note that for the combined modes, this length does not include the authentication only data; the authentication length is specified in the AES_AAD_LENGTH register below.<br>All modes must have a length > 0. For the combined modes, it is allowed to have one of the lengths equal to zero.<br>For the basic encryption modes (ECB/CBC/CTR/ICM/CFB/OFB) it is allowed to program zero to the length field; in that case the length is assumed infinite.<br>All data must be byte (8-bit) aligned for stream cipher modes; bit aligned data streams are not supported. For block cipher modes, the data length must be programmed in multiples of the block cipher size, 16 bytes. |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.48 AAD_LENGTH (Offset = 115Ch) [Reset = 00000000h]

AAD_LENGTH is shown in Figure 13-58 and described in Table 13-55.

Return to the Summary Table.

AAD Data Length

**Figure 13-58. AAD_LENGTH**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 13-55. AAD_LENGTH Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | W | 0h | Bits [31:0] of the authentication length register store the authentication data length in bytes for combined modes only (GCM or CCM)<br>Supported AAD-lengths for CCM are from 0 to (216-28) bytes. For GCM any value up to (232-1) bytes can be used. Once processing with this context is started, this length decrements to zero.<br>A write to this register triggers the engine to start using this context for GCM and CCM. |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.49 DATA0 (Offset = 1160h) [Reset = 00000000h]

DATA0 is shown in Figure 13-59 and described in Table 13-56.

Return to the Summary Table.

Data input (LSW) / Data output (LSW)\. The Data Input/Output Registers buffer the input/output data blocks to/from the engine. Notice that the data input buffer (AES_DATA_IN_n) and data output buffer (AES_DATA_OUT_n) are mapped to the same address locations. Writes to these addresses load the Input Buffer while reads pull from the Output Buffer. Therefore, for write access, the data input buffer is written; for read access, the data output buffer is read. The data input buffer must be written prior to starting an operation. The data output buffer contains valid data on completion of an operation. All writes from, and reads to, these registers are tracked independently per direction. Therefore, any 128-bit data block can be split over multiple 32-bit word transfers, which can be mixed with other transfers over the external interface.

**Figure 13-59. DATA0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 13-56. DATA0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | R/W | 0h | Data |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.50 DATA1 (Offset = 1164h) [Reset = 00000000h]

DATA1 is shown in Figure 13-60 and described in Table 13-57.

Return to the Summary Table.

Data input / Data output. The Data Input/Output Registers buffer the input/output data blocks to/from the engine. Notice that the data input buffer (AES_DATA_IN_n) and data output buffer (AES_DATA_OUT_n) are mapped to the same address locations. Writes to these addresses load the Input Buffer while reads pull from the Output Buffer. Therefore, for write access, the data input buffer is written; for read access, the data output buffer is read. The data input buffer must be written prior to starting an operation. The data output buffer contains valid data on completion of an operation. All writes from, and reads to, these registers are tracked independently per direction. Therefore, any 128-bit data block can be split over multiple 32-bit word transfers, which can be mixed with other transfers over the external interface.

**Figure 13-60. DATA1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 13-57. DATA1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | R/W | 0h | Data |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.51 DATA2 (Offset = 1168h) [Reset = 00000000h]

DATA2 is shown in Figure 13-61 and described in Table 13-58.

Return to the Summary Table.

Data input / Data output. The Data Input/Output Registers buffer the input/output data blocks to/from the engine. Notice that the data input buffer (AES_DATA_IN_n) and data output buffer (AES_DATA_OUT_n) are mapped to the same address locations. Writes to these addresses load the Input Buffer while reads pull from the Output Buffer. Therefore, for write access, the data input buffer is written; for read access, the data output buffer is read. The data input buffer must be written prior to starting an operation. The data output buffer contains valid data on completion of an operation. All writes from, and reads to, these registers are tracked independently per direction. Therefore, any 128-bit data block can be split over multiple 32-bit word transfers, which can be mixed with other transfers over the external interface.

**Figure 13-61. DATA2**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 13-58. DATA2 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | R/W | 0h | Data |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.52 DATA3 (Offset = 116Ch) [Reset = 00000000h]

DATA3 is shown in Figure 13-62 and described in Table 13-59.

Return to the Summary Table.

Data input (MSW) / Data output (MSW). The Data Input/Output Registers buffer the input/output data blocks to/from the engine. Notice that the data input buffer (AES_DATA_IN_n) and data output buffer (AES_DATA_OUT_n) are mapped to the same address locations. Writes to these addresses load the Input Buffer while reads pull from the Output Buffer. Therefore, for write access, the data input buffer is written; for read access, the data output buffer is read. The data input buffer must be written prior to starting an operation. The data output buffer contains valid data on completion of an operation. All writes from, and reads to, these registers are tracked independently per direction. Therefore, any 128-bit data block can be split over multiple 32-bit word transfers, which can be mixed with other transfers over the external interface.

**Figure 13-62. DATA3**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 13-59. DATA3 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | R/W | 0h | Data |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.53 TAG0 (Offset = 1170h) [Reset = 00000000h]

TAG0 is shown in Figure 13-63 and described in Table 13-60.

Return to the Summary Table.

Hash result (LSW). These registers buffer the TAG from the engine. The registers are shared with the intermediate authentication result registers but cannot be read until the processing is finished.

**Figure 13-63. TAG0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 13-60. TAG0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | R | 0h | For a CPU read operation, these registers contain the last 128-bit TAG output of the engine; the TAG is available until the next context is written.<br>This register will only contain valid data if the TAG is available, when the ‘saved_context_ready’ or ‘get_digest’ bit from AES_CTRL register is set. In case of get_digest, the output will be an intermediate TAG for CCM or GCM operation continuation. During processing or for operations/modes that do not return a TAG, reads from this register returns data from the IV register. For operations that do return a TAG in the IV register, the IV register must be accessed directly. |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.54 TAG1 (Offset = 1174h) [Reset = 00000000h]

TAG1 is shown in Figure 13-64 and described in Table 13-61.

Return to the Summary Table.

Hash result. These registers buffer the TAG from the engine. The registers are shared with the intermediate authentication result registers but cannot be read until the processing is finished.

**Figure 13-64. TAG1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 13-61. TAG1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | R | 0h | For a CPU read operation, these registers contain the last 128-bit TAG output of the engine; the TAG is available until the next context is written.<br>This register will only contain valid data if the TAG is available, when the ‘saved_context_ready’ or ‘get_digest’ bit from AES_CTRL register is set. In case of get_digest, the output will be an intermediate TAG for CCM or GCM operation continuation. During processing or for operations/modes that do not return a TAG, reads from this register returns data from the IV register. For operations that do return a TAG in the IV register, the IV register must be accessed directly. |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.55 TAG2 (Offset = 1178h) [Reset = 00000000h]

TAG2 is shown in Figure 13-65 and described in Table 13-62.

Return to the Summary Table.

Hash result. These registers buffer the TAG from the engine. The registers are shared with the intermediate authentication result registers but cannot be read until the processing is finished.

**Figure 13-65. TAG2**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 13-62. TAG2 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | R | 0h | For a CPU read operation, these registers contain the last 128-bit TAG output of the engine; the TAG is available until the next context is written.<br>This register will only contain valid data if the TAG is available, when the 'saved_context_ready' or 'get_digest' bit from AES_CTRL register is set. In case of get_digest, the output will be an intermediate TAG for CCM or GCM operation continuation. During processing or for operations/modes that do not return a TAG, reads from this register returns data from the IV register. For operations that do return a TAG in the IV register, the IV register must be accessed directly. |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.56 TAG3 (Offset = 117Ch) [Reset = 00000000h]

TAG3 is shown in Figure 13-66 and described in Table 13-63.

Return to the Summary Table.

Hash result (MSW). These registers buffer the TAG from the engine. The registers are shared with the intermediate authentication result registers but cannot be read until the processing is finished.

**Figure 13-66. TAG3**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 13-63. TAG3 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | R | 0h | For a CPU read operation, these registers contain the last 128-bit TAG output of the engine; the TAG is available until the next context is written.<br>This register will only contain valid data if the TAG is available, when the ‘saved_context_ready’ or ‘get_digest’ bit from AES_CTRL register is set. In case of get_digest, the output will be an intermediate TAG for CCM or GCM operation continuation. During processing or for operations/modes that do not return a TAG, reads from this register returns data from the IV register. For operations that do return a TAG in the IV register, the IV register must be accessed directly. |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.57 STATUS (Offset = 1180h) [Reset = 00000000h]

STATUS is shown in Figure 13-67 and described in Table 13-64.

Return to the Summary Table.

Status register

**Figure 13-67. STATUS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | KEYWR |
| R-0h | | | | | | | R-0h |

**Table 13-64. STATUS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | KEYWR | R | 0h | Key write status. 0 - user write to KEY register is allowed. 1 - user write to KEY register is ignored.<br>In order to allow user write, perform a module reset.<br>0h = User write to KEY MMR is allowed<br>1h = User write to KEY MMR is disabled. Writing has no effect. |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.58 DATA_IN (Offset = 1184h) [Reset = 00000000h]

DATA_IN is shown in Figure 13-68 and described in Table 13-65.

Return to the Summary Table.

Data-in register: alias for DATA0/1/2/3 at a single address for DMA addressing

**Figure 13-68. DATA_IN**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 13-65. DATA_IN Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | W | 0h | Data input word |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.59 DATA_OUT (Offset = 1188h) [Reset = 00000000h]

DATA_OUT is shown in Figure 13-69 and described in Table 13-66.

Return to the Summary Table.

Data-out register: alias for DATA0/1/2/3 at a single address for DMA addressing

**Figure 13-69. DATA_OUT**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 13-66. DATA_OUT Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | R | 0h | Data output word |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.60 FORCE_IN_AV (Offset = 11D0h) [Reset = 00000000h]

FORCE_IN_AV is shown in Figure 13-70 and described in Table 13-67.

Return to the Summary Table.

Data control register for input data. This write-only register provides a means to force the availability of the input data buffer of the engine.

**Figure 13-70. FORCE_IN_AV**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 13-67. FORCE_IN_AV Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | W | 0h | Any write to this register forces the input data buffer to valid and will force the engine to start processing this data. The data written here is not used. The core must be configured to have input and output data acknowledge be I/O register based |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.61 CCM_ALN_WRD (Offset = 11D4h) [Reset = 00000000h]

CCM_ALN_WRD is shown in Figure 13-71 and described in Table 13-68.

Return to the Summary Table.

AES-CCM AAD alignment data word. This register provides a means to access an internal register that stores alignment data bytes during the AAD phase of AES-CCM processing. This register needs to be read and stored when an AES-CCM operation is interrupted during the AAD phase. This value needs to be restored by writing this register, when resuming that AES-CCM operation in a later session.

**Figure 13-71. CCM_ALN_WRD**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 13-68. CCM_ALN_WRD Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | W | 0h | This register provides a means to access an internal register that stores alignment data bytes during the AAD phase of AES-CCM processing. This register needs to be read and stored when an AES-CCM operation is interrupted during the AAD phase. This value needs to be restored by writing this register, when resuming that AES-CCM operation in a later session. |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.62 BLK_CNT0 (Offset = 11D8h) [Reset = 00000000h]

BLK_CNT0 is shown in Figure 13-72 and described in Table 13-69.

Return to the Summary Table.

Internal block counter (LSW). This register along with BLK_CNT1 register provides access to the internal data block counter of the engine. This counter keeps track of the number of data blocks during AES-CCM and AES-GCM operations. Reading and writing this counter allows interruption and resuming of long CCM or GCM operations. Note that internally, the block counter is used for AAD data as well as encryption/decryption data. Interruption and resuming is only supported in the encryption/decryption data phase and not during AAD.

**Figure 13-72. BLK_CNT0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 13-69. BLK_CNT0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | R/W | 0h | Internal block counter for AES GCM and CCM operations.<br>These bits read the block count value that represents the number of blocks to go. This value is valid with saved_context_ready after a request for an intermediate GCM/CCM digest.<br>Writing these registers will restore the internal block counter to the programmed value. This only needs to be done to prepare the engine to continue processing of an interrupted GCM or CCM operation.<br>Also refer to the get_digest and gcm_ccm_continue bits in AES_CTRL register. |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.63 BLK_CNT1 (Offset = 11DCh) [Reset = 00000000h]

BLK_CNT1 is shown in Figure 13-73 and described in Table 13-70.

Return to the Summary Table.

Internal block counter (MSW). This register along with BLK_CNT0 register provides access to the internal data block counter of the engine. This counter keeps track of the number of data blocks during AES-CCM and AES-GCM operations. Reading and writing this counter allows interruption and resuming of long CCM or GCM operations. Note that internally, the block counter is used for AAD data as well as encryption/decryption data. Interruption and resuming is only supported in the encryption/decryption data phase and not during AAD.

**Figure 13-73. BLK_CNT1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | DATA | | | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | R/W-0h | | | | | | | | | | | | | | | | | | | | | | | |

**Table 13-70. BLK_CNT1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | RESERVED | R | 0h | |
| 23-0 | DATA | R/W | 0h | Internal block counter for AES GCM and CCM operations.<br>These bits read the block count value that represents the number of blocks to go. This value is valid with saved_context_ready after a request for an intermediate GCM/CCM digest.<br>Writing these registers will restore the internal block counter to the programmed value. This only needs to be done to prepare the engine to continue processing of an interrupted GCM or CCM operation.<br>Also refer to the get_digest and gcm_ccm_continue bits in AES_CTRL register. |



Copyright © 2026 Texas Instruments Incorporated

### 13.3.64 DMA_HS (Offset = 11F4h) [Reset = 00000000h]

DMA_HS is shown in Figure 13-74 and described in Table 13-71.

Return to the Summary Table.

Control register for DMA handshaking

**Figure 13-74. DMA_HS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | DMA_DATA_ACK |
| R-0h | | | | | | | R/W-0h |

**Table 13-71. DMA_HS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | DMA_DATA_ACK | R/W | 0h | When this bit is 0b, input and output data acknowledge is I/O register based, as specified in the description of the AES_DATA_IN_n / AES_DATA_OUT_n registers.<br>When this bit is 1b, input and output data acknowledge is based on DMA handshake signals.<br>0h = Disable DMA based data handshake<br>1h = Enables DMA based handshake |



Copyright © 2026 Texas Instruments Incorporated

Chapter 14

# CRC

The cyclic redundancy check (CRC) accelerator generates signatures for a given data sequence based on the CRC16-CCITT polynomial or the CRC32-ISO3309 polynomial.


**14.1 CRC Overview** ..... 964
**14.2 CRC Operation** ..... 964
**14.3 CRCP0 Registers** ..... 967





Copyright © 2026 Texas Instruments Incorporated

## 14.1 CRC Overview

The CRC accelerator produces CRC signatures for given sequences of data. The CRC16-CCITT and CRC32-ISO3309 polynomial functions are supported. Identical input data sequences result in identical CRC signatures when the CRC is initialized with a fixed seed value. Different sequences of input data, in general, result in different signatures for a given CRC function.

| | CRC | CRC-P |
|---|---|---|
| Support for CRC16-CCITT | ✓ | ✓ |
| Support for CRC32-ISO3309 Polynomial | ✓ | ✓ |
| Fast single cycle computation of new CRC output for each data input (no wait states) | ✓ | X |
| Support for input / output bit reversal | ✓ | ✓ |
| Support for big endian and little endian operation | ✓ | ✓ |
| Byte, Half-word, or word input to CRCIN | ✓ | ✓ |
| 512-word CRCIN_IDX input field in which all addresses are mapped to CRCIN, supporting use of a standard C-style memcpy() routine to load data into the CRC module for data lengths up to 2KB | ✓ | ✓ |
| User-selectable CRC32 polynomial | X | ✓ |

**Figure 14-1. CRC and CRC-P Key Features**

### *14.1.1 CRC16-CCITT*

For CRC16-CCITT, the CRC signature is generated based on the polynomial given in the 16-bit CCITT standard, as shown in the equation below.

$$f(x)=x^{16}+x^{12}+x^{5}+1 \quad (14)$$

The CRC16-CCITT digest size is 16 bits (half-word).

### *14.1.2 CRC32-ISO3309*

For CRC32-ISO3309, the CRC signature is generated based on the polynomial given in the ISO3309 Ethernet standard, as shown in the equation below.

$$f(x)=x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^{8}+x^{7}+x^{5}+x^{4}+x^{2}+x+1 \quad (15)$$

The digest size for CRC32-ISO3309 is 32 bits.

## 14.2 CRC Operation

The CRC generator is initialized by configuring the desired mode of operation in the CRCCTRL register, followed by writing the seed value to the CRCSEED register. After the seed is loaded to the CRCSEED register, the CRCOUT register will reflect the SEED value loaded to CRCSEED.

**Note**

If the endianness mode is configured to big endian before the seed value is written to CRCSEED, then the byte order of the seed value written to CRCSEED is swapped when the seed value is loaded into the CRC module.



Copyright © 2026 Texas Instruments Incorporated

Once initialized, data can be input into the CRC generator by writing to the CRCIN register using byte (8-bit), half-word (16-bit), or word (32-bit) writes. The CPU or the DMA can be used to move input data into the CRC accelerator.

**Note**

Byte writes need not be word aligned; a byte write to any byte location in CRCIN will be interpreted the same way (adding the 8 written bits to the computed CRC). Half-word writes also need not be word aligned, but they must be half-word aligned. For example, a half-word can be written to BIT0-BIT15 or BIT16-BIT32 of CRCIN, but not to BIT8-BIT23.

When using the CRC generator to verify a data set, all data to be included in the CRC calculation must be written to the CRCIN register in the same order as was used to calculate the original CRC signature. When using the CRC generator to create a new signature to be used in the future for verification, be sure to load the data the same way and with the same settings when performing verification.

The current CRC output can be read at any time by reading the CRCOUT register.

### *14.2.1 CRC Generator Implementation*

The CRC generator is implemented with a set of XOR trees. After a set of 8, 16, or 32 bits is provided to the CRC accelerator by writing to the CRCIN register, a calculation for the whole set of input bits is performed. When new data is written to CRCIN, the CRC generator updates the CRC output in a single cycle when using the CRC peripheral. CRC-P versions require additional cycles. Bus wait states are not required to load data back-to-back into the CRC generator.

### *14.2.2 Configuration*

The CRC accelerator supports polynomial selection, bit reversal selection, and byte order (endianness) selection. This section describes these configuration aspects.

The CRC accelerator must be enabled before being used through the PWREN register (see peripheral power enable).

The CRC accelerator is in power domain 1 (PD1), and as such can only be active in RUN or SLEEP mode. If the CRC accelerator is configured to be enabled by application software, and the device enters STOP or STANDBY mode, SYSCTL will force the CRC into a disabled state until the device exists STOP or STANDBY mode. All CRC register contents are retained when the CRC is forced to a disabled state in STOP or STANDBY mode.

The CRC module only runs from the PD1 bus clock (MCLK).

#### 14.2.2.1 Polynomial Selection

The desired polynomial (CRC16-CCITT or CRC32-ISO3309) is selected with the POLYSIZE bit in the CRCCTRL register. The polynomial to be used must be selected before loading the seed value and any data values.

When the CRC generator is configured for a 16-bit digest (CRC16-CCITT), the following conditions apply:

- The upper half-word (16 bits) of the CRCSEED register are ignored and only the lower half-word is used
- The upper half-word (16 bits) of the CRCOUT register read back as zero and only the lower half-word is valid

When the CRC generator is configured for CRC32-ISO3309, all 32 bits of CRCSEED and CRCOUT are valid.

#### 14.2.2.2 Bit Order

The various CRC standards were defined in the era of main frame computers. At that time, BIT0 was treated as the MSB. In modern computing, BIT0 is typically the LSB.

The Arm Cortex-M0+M33 CPU treats BIT0 as the LSB, as is typical in modern CPUs and MCUs. This sometimes causes confusion, because BIT0 has been treated as the LSB in some cases and as the MSB in other cases. Therefore, the CRC accelerator provides a bit order reversal capability to support both conventions.

Bit order reversal can be enabled by setting the BITREVERSE bit in the CRCCTRL register, giving the following behavior for input and output data:



Copyright © 2026 Texas Instruments Incorporated

- **Input data:** The bit order of each input byte written to the CRCIN register is reversed before it is passed to the CRC generator to be used for CRC calculation
- **Output data:** The bit order of the 16-bit or 32-bit CRC result is reversed when read from the CRCOUT register

**Note**

If input data must be provided bit-reversed, but the output is to be read not reversed, the BITREVERSE bit can be set before loading data to CRCIN and then cleared after all data is written to CRCIN but before the resulting signature is read from CRCOUT. Likewise, it is possible to load input data to CRCIN with BITREVERSE cleared (not reversed), and flip the output before reading it (by setting BITREVERSE before reading CRCOUT).

#### 14.2.2.3 Byte Swap

The bit OUTPUT_BYTESWAP in the register CRCCTRL can be used to enable or disable CRC output byte swap. This bit controls whether the output is byte-swapped upon a read of the CRCOUT register. If CRCOUT is accessed as a half-word, and the OUTPUT_BYTESWAP is set to 1, then the two bytes in the 16-bit access are swapped and returned. Using B0, B1, B2 and B3 to identify Byte 0, Byte 1, Byte 2, Byte 3. B1 is returned as B0 and B0 is returned as B1. If CRCOUT is accessed as a word, and the OUTPUT_BYTESWAP is set to 1, then the four bytes in the 32-bit read are swapped. B3 is returned as B0, B2 is returned as B1, B1 is returned as B2 and B0 is returned as B3.

Note that if the CRC POLYSIZE is 16-bit and a 32-bit read of CRCOUT is performed with OUTPUT_BYTESWAP enabled, then the output is: MSB LSB 0x0 0x0 B0 B1. If the CRC POLYSIZE is 16-bit and a 32-bit read of CRCOUT is performed with OUTPUT_BYTESWAP disabled, then the output is: MSB LSB 0x0 0x0 B1 B0.

#### 14.2.2.4 Byte Order

When working with half-word or word input data, the input byte order can be configured as either little endian or big endian. The default configuration is little endian. To reverse the byte order when using half-word or word inputs, set the INPUT_ENDIANNESS bit in the CRCCTRL register.

Reversing the endianness will cause the following translation for half-word and word writes:

**Table 14-1. CRCIN Byte Order Translation**

| Endianness | Data Written to CRCIN | Data Applied to CRC Logic |
|---|---|---|
| 0 (little) | 0x1234 | 0x1234 |
| 1 (big) | 0x1234 | 0x3412 |
| 0 (little) | 0x12345678 | 0x12345678 |
| 1 (big) | 0x12345678 | 0x78563412 |

**Note**

If the ENDIANNESS bit is set before the seed value is written to CRCSEED, then the byte order of the seed value written to CRCSEED is also reversed when it is loaded into the CRC, and the seed value will read back reversed when reading the CRCOUT register after writing to the CRCSEED register.

#### 14.2.2.5 CRC C Library Compatibility

To simplify loading of data by software into the CRC, the CRC accelerator provides a 512-word (2KB) CRCIN_IDX region. Within the CRCIN_IDX region, a write to any word is re-mapped as, and functionally equivalent to, a write to the CRCIN register. This mechanism enables the use of the standard C library *memcpy()* routine to copy data from SRAM or flash into the CRC, provided that the source data is less than 2KB (2,048 bytes).



Copyright © 2026 Texas Instruments Incorporated

## 14.3 CRCP0 Registers

Table 14-2 lists the memory-mapped registers for the CRCP0 registers. All register offset addresses not listed in Table 14-2 should be considered as reserved locations and the register contents should not be modified.

**Table 14-2. CRCP0 Registers**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 800h | PWREN | Power enable | | Go |
| 804h | RSTCTL | Reset Control | | Go |
| 814h | STAT | Status Register | | Go |
| 1004h | CLKSEL | Clock Select | | Go |
| 10FCh | DESC | Module Description | | Go |
| 1100h | CRCCTRL | CRC Control Register | | Go |
| 1104h | CRCSEED | CRC Seed Register | | Go |
| 1108h | CRCIN | CRC Input Data Register | | Go |
| 110Ch | CRCOUT | CRC Output Result Register | | Go |
| 1110h | CRCPOLY | CRC Polynomial configuration register | | Go |
| 1800h + formula | CRCIN_IDX[y] | CRC Input Data Array Register | | Go |

Complex bit access types are encoded to fit into small table cells. Table 14-3 shows the codes that are used for access types in this section.

**Table 14-3. CRCP0 Access Type Codes**

| Access Type | Code | Description |
|---|---|---|
| Read Type | | |
| H | H | Set or cleared by hardware |
| R | R | Read |
| Write Type | | |
| W | W | Write |
| WK | W<br>K | Write<br>Write protected by a key |
| Reset or Default Value | | |
| *-n* | | Value after reset or the default value |
| Register Array Variables | | |
| i,j,k,l,m,n | | When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula. |
| y | | When this variable is used in a register name, an offset, or an address it refers to the value of a register array. |



Copyright © 2026 Texas Instruments Incorporated

### 14.3.1 PWREN (Offset = 800h) [Reset = 00000000h]

PWREN is shown in Figure 14-2 and described in Table 14-4.

Return to the Summary Table.

Register to control the power state

**Figure 14-2. PWREN**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | ENABLE |
| R-0h | | | | | | | R/WK-0h |

**Table 14-4. PWREN Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | KEY to allow Power State Change<br>26h = KEY to allow write access to this register |
| 23-1 | RESERVED | R | 0h | |
| 0 | ENABLE | R/WK | 0h | Enable the power<br>KEY must be set to 26h to write to this bit.<br>0h = Disable Power<br>1h = Enable Power |



Copyright © 2026 Texas Instruments Incorporated

### 14.3.2 RSTCTL (Offset = 804h) [Reset = 00000000h]

RSTCTL is shown in Figure 14-3 and described in Table 14-5.

Return to the Summary Table.

Register to control reset assertion and de-assertion

**Figure 14-3. RSTCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | RESETSTKYCLR | RESETASSERT |
| R-0h | | | | | | WK-0h | WK-0h |

**Table 14-5. RSTCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | Unlock key<br>B1h = KEY to allow write access to this register |
| 23-2 | RESERVED | R | 0h | |
| 1 | RESETSTKYCLR | WK | 0h | Clear the RESETSTKY bit in the STAT register<br>KEY must be set to B1h to write to this bit.<br>0h = Writing 0 has no effect<br>1h = Clear reset sticky bit |
| 0 | RESETASSERT | WK | 0h | Assert reset to the peripheral<br>KEY must be set to B1h to write to this bit.<br>0h = Writing 0 has no effect<br>1h = Assert reset |



Copyright © 2026 Texas Instruments Incorporated

### 14.3.3 STAT (Offset = 814h) [Reset = 00000000h]

STAT is shown in Figure 14-4 and described in Table 14-6.

Return to the Summary Table.

peripheral enable and reset status

**Figure 14-4. STAT**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | RESETSTKY |
| R-0h | | | | | | | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |

**Table 14-6. STAT Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-17 | RESERVED | R | 0h | |
| 16 | RESETSTKY | R | 0h | This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register<br>0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register<br>1h = The peripheral was reset since the last bit clear |
| 15-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 14.3.4 CLKSEL (Offset = 1004h) [Reset = 00000001h]

CLKSEL is shown in Figure 14-5 and described in Table 14-7.

Return to the Summary Table.

Clock source selection

**Figure 14-5. CLKSEL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | MCLK_SEL |
| R-0h | | | | | | | R-1h |

**Table 14-7. CLKSEL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | MCLK_SEL | R | 1h | Selects main clock (MCLK) if enabled<br>0h = Does not select this clock as a source<br>1h = Select this clock as a source |



Copyright © 2026 Texas Instruments Incorporated

### 14.3.5 DESC (Offset = 10FCh) [Reset = 00000000h]

DESC is shown in Figure 14-6 and described in Table 14-8.

Return to the Summary Table.

This register identifies the peripheral and its exact version.

**Figure 14-6. DESC**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MODULEID | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| FEATUREVER | | | | INSTNUM | | | | MAJREV | | | | MINREV | | | |
| R-0h | | | | R-0h | | | | R-0h | | | | R-0h | | | |

**Table 14-8. DESC Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | MODULEID | R | 2011h | Module identification contains a unique peripheral identification number. The assignments are maintained in a central database for all of the platform modules to ensure uniqueness.<br>0h = Smallest value<br>FFFFh = Highest possible value |
| 15-12 | FEATUREVER | R | 8h | Feature Set for the module *instance*<br>0h = Smallest value<br>Fh = Highest possible value |
| 11-8 | INSTNUM | R | 0h | Instance Number within the device. This will be a parameter to the RTL for modules that can have multiple instances<br>0h = Smallest value<br>Fh = Highest possible value |
| 7-4 | MAJREV | R | 1h | Major rev of the IP<br>0h = Smallest value<br>Fh = Highest possible value |
| 3-0 | MINREV | R | 0h | Minor rev of the IP<br>0h = Smallest value<br>Fh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 14.3.6 CRCCTRL (Offset = 1100h) [Reset = 00000000h]

CRCCTRL is shown in Figure 14-7 and described in Table 14-9.

Return to the Summary Table.

CRC Control Register. Configuration control of the CRC.

**Figure 14-7. CRCCTRL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | OUTPUT_BYTESWAP | RESERVED | INPUT_ENDIANNESS | BITREVERSE | POLYSIZE |
| R-0h | | | R/W-0h | R-0h | R/W-0h | R/W-0h | R/W-0h |

**Table 14-9. CRCCTRL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-5 | RESERVED | R | 0h | |
| 4 | OUTPUT_BYTESWAP | R/W | 0h | CRC Output Byteswap Enable. This bit controls whether the output is byte-swapped upon a read of the CRCOUT register.<br>If CRCOUT is accessed as a half-word, and the OUTPUT_BYTESWAP is set to to 1, then the two bytes in the 16-bit access are swapped and returned.<br>B1 is returned as B0<br>B0 is returned as B1<br>If CRCOUT is accessed as a word, and the OUTPUT_BYTESWAP is set to 1, then the four bytes in the 32-bit read are swapped.<br>B3 is returned as B0<br>B2 is returned as B1<br>B1 is returned as B2<br>B0 is returned as B3<br>Note that if the CRC POLYSIZE is 16-bit and a 32-bit read of CRCOUT is performed with OUTPUT_BYTESWAP enabled, then the output is:<br>MSB LSB<br>0x0 0x0 B0 B1<br>If the CRC POLYSIZE is 16-bit and a 32-bit read of CRCOUT is performed with OUTPUT_BYTESWAP disabled, then the output is:<br>MSB LSB<br>0x0 0x0 B1 B0<br>0h = Output byteswapping is disabled<br>1h = Output byteswapping is enabled. |
| 3 | RESERVED | R | 0h | |
| 2 | INPUT_ENDIANNESS | R/W | 0h | CRC Endian. This bit indicates the byte order within a word or half word of input data.<br>0h = LSB is lowest memory address and first to be processed.<br>1h = LSB is highest memory address and last to be processed. |



Copyright © 2026 Texas Instruments Incorporated

**Table 14-9. CRCCTRL Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 1 | BITREVERSE | R/W | 0h | CRC Bit Input and output Reverse. This bit indictes that the bit order of each input byte used for the CRC calculation is reversed before it is passed to the generator, and that the bit order of the calculated CRC is be reversed when read from CRC_RESULT.<br>0h = Bit order is not reversed.<br>1h = Bit order is reversed. |
| 0 | POLYSIZE | R/W | 0h | This bit indicates which CRC calculation is performed by the generator.<br>0h = CRC-32 ISO-3309 calulation is performed<br>1h = CRC-16 CCITT is performed |



Copyright © 2026 Texas Instruments Incorporated

### 14.3.7 CRCSEED (Offset = 1104h) [Reset = 00000000h]

CRCSEED is shown in Figure 14-8 and described in Table 14-10.

Return to the Summary Table.

CRC Seed Register. The Data written to this register is used to initialize the CRC result with this SEED value. Note that in 16-bit mode only the lower 16-bits of this value are used. After writing this register the CRC Output Result Register will reflect this value.

**Figure 14-8. CRCSEED**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SEED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 14-10. CRCSEED Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | SEED | W | 0h | Seed Data<br>00000000h = Mnimum value<br>FFFFFFFFh = Maximum value |



Copyright © 2026 Texas Instruments Incorporated

### 14.3.8 CRCIN (Offset = 1108h) [Reset = 00000000h]

CRCIN is shown in Figure 14-9 and described in Table 14-11.

Return to the Summary Table.

CRC Input Data Register. The Data writen to this register is used along with the current CRC result to calculate the next CRC result. This is done in a single clock cycle and requires no wait states. This register can by written as a byte, half word or word transfer and the correct number of bits will be used for the next CRC result.
This register is also mapped to a range of registers starting at 0xTDB_X000 and ending at 0xTDB_XFFF to allow memcpy to be used instead of DMA for CRC calculations that do not exceed the bounds of the memory range of this register.

**Figure 14-9. CRCIN**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 14-11. CRCIN Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | W | 0h | Input Data<br>00000000h = Mnimum value<br>FFFFFFFFh = Maximum value |



Copyright © 2026 Texas Instruments Incorporated

### 14.3.9 CRCOUT (Offset = 110Ch) [Reset = 00000000h]

CRCOUT is shown in Figure 14-10 and described in Table 14-12.

Return to the Summary Table.

CRC Output Result Register. This register stores the result of the currect CRC calculation. Note when configured for 16-bit mode the upper bits will read back 0. Note that if output inversion is set in the CRC Control register it will be applied.

**Figure 14-10. CRCOUT**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESULT | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 14-12. CRCOUT Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | RESULT | R | 0h | Result<br>00000000h = Mnimum value<br>FFFFFFFFh = Maximum value |



Copyright © 2026 Texas Instruments Incorporated

### 14.3.10 CRCPOLY (Offset = 1110h) [Reset = 00000000h]

CRCPOLY is shown in Figure 14-11 and described in Table 14-13.

Return to the Summary Table.

CRC Polynomial configuration register. This register is present only in devices that support custom polynomials. Consult the device datasheet.

**Figure 14-11. CRCPOLY**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| -0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 14-13. CRCPOLY Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | 04C11DB7h | Polynomial definition | |



Copyright © 2026 Texas Instruments Incorporated

### 14.3.11 CRCIN_IDX[y] (Offset = 1800h + formula) [Reset = 00000000h]

CRCIN_IDX[y] is shown in Figure 14-12 and described in Table 14-14.

Return to the Summary Table.

This register is dual mapped to CRCIN and is intended to allow operation with C memcpy routine.

Offset = 1800h + (y * 4h); where y = 0h to 1FFh

**Figure 14-12. CRCIN_IDX[y]**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| HW-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 14-14. CRCIN_IDX[y] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | HW | 0h | Input Data<br>00000000h = Minimum value<br>FFFFFFFFh = Maximum value |



Copyright © 2026 Texas Instruments Incorporated

Chapter 15

# Keystore

**15.1 Overview**....981
**15.2 Detailed Description**....981
**15.3 KEYSTORECTL Registers**....982



Copyright © 2026 Texas Instruments Incorporated

## 15.1 Overview

The Keystore controller provides secure management of the Advanced Encryption Engine (AES) keys. The use model of the keystore controller is to securely deposit keys into it during the execution of customer secure code and to have the AES engine access them subsequently in a secure manner without leaking any key data to observers. Both 128 and 256-bit keys can be stored in the keystore key slots. The keystore and its interaction with the AES engine are designed for secure operation including thwarting partial key modification attacks.

## 15.2 Detailed Description

The keystore supports storage of AES keys. Up to four 128-bit key slots are provisioned and can be used in different software-managed configurations, such as:

- Four 128-bit keys
- Two 256-bit keys
- One 256-bit key and two 128-bit keys

**Note**

Refer to the device-specific data sheet to determine the number of AES keys supported.

This configuration selection is the first to be performed by customer secure code before depositing keys into slots. Configuration is done by writing the number of 256-bit keys into the NK256 field of the CFG register.

After configuration, the key bytes can be written to selected slots in the keystore controller register. This is done by programming the KEYSZSEL (key size selection) and KEYSLOTSEL (key slot selection) fields of the KEYWR register. Note that the hardware expects any 256-bit keys to be store in lower-numbered slots. For example, if the store is configured for one 256-bit key and two 128-bit keys, the 256-bit key must be stored in slots 0 and 1. The IP reports an error (invalid configuration) if any other slot is used for the 256-bit key. If the store is configured for two 256-bit keys, then key 0 occupies slots 0 and 1 and key 1 occupies slots 2 and 3.

Depositing the key bytes into slots is done by writing them to the KEYIN register. Once a key write transaction is initiated, until and unless all the key bytes are written (4 words for a 128-bit key and 8 words for a 256-bit key), the key is not considered valid. Validity of slots is presented via a status register for the application. When customer secure code has deposited the required number of bytes, the key becomes valid and can be used by the application. The STATUS register provides status and validity fields to indicate status of the keystore controller as well as the validity of key slots.

Note that key configuration operations are permitted only while customer secure code is executing. Subsequently, the keystore configuration is locked and can not be modified. The signaling of the end of customer secure code is discussed in the Security Architecture chapter.

At run-time (after the end of customer secure code execution), the application is able to use one of the valid keys for an AES encryption/decryption operation. To configure the AES engine to use a specific key, the application references a key slot and initiates a transfer of the key data into the AES engine via a secure internal bus. This is performed by writing to the KEYSZSEL (key size selection), and KEYSLOTSEL (Key slot selection) fields of the KEYRD register. The selected key data is securely transferred to the AES engine over a private bus that is not accessible by software or the debugger.

The keystore holds the state and key data until a subsequent boot reset. At boot reset, customer startup code configures the keystore. If one or more slots are not used at the end of customer startup code execution, those slots remain unusable for the rest of the application execution.



Copyright © 2026 Texas Instruments Incorporated

## 15.3 KEYSTORECTL Registers

Table 15-1 lists the memory-mapped registers for the KEYSTORECTL registers. All register offset addresses not listed in Table 15-1 should be considered as reserved locations and the register contents should not be modified.

**Table 15-1. KEYSTORECTL Registers**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 1100h | CFG | Keystore configuration | | Go |
| 1104h | KEYWR | Key write configuration | | Go |
| 1108h | KEYRD | Key read configuration | | Go |
| 110Ch | STATUS | Status | | Go |
| 1110h | KEYIN | Input key | | Go |
| 1114h | KEYLOCK | Keylock | | Go |

Complex bit access types are encoded to fit into small table cells. Table 15-2 shows the codes that are used for access types in this section.

**Table 15-2. KEYSTORECTL Access Type Codes**

| Access Type | Code | Description |
|---|---|---|
| Read Type | | |
| R | R | Read |
| Write Type | | |
| W | W | Write |
| Reset or Default Value | | |
| *-n* | | Value after reset or the default value |



Copyright © 2026 Texas Instruments Incorporated

### 15.3.1 CFG (Offset = 1100h) [Reset = 00000001h]

CFG is shown in Figure 15-1 and described in Table 15-3.

Return to the Summary Table.

Key-store configuration register.

**Figure 15-1. CFG**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | NK256 | | | |
| R-0h | | | | | | | | | | | | R/W-1h | | | |

**Table 15-3. CFG Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3-0 | NK256 | R/W | 1h | Number of 256 bit keys to be held in the key-store.<br>Can not exceed the total number of slots present in the hardware / 2.<br>For example, if SYS_N_SLOTS = 4, then at most 2 256-bit keys can be held in the key-store.<br>Incorrect setting of this field will be reported via STATUS register<br>0h = No 256-bit keys<br>1h = One 256-bit key<br>2h = Two 256-bit keys<br>3h = Three 256-bit keys<br>4h = Four 256-bit keys |



Copyright © 2026 Texas Instruments Incorporated

### 15.3.2 KEYWR (Offset = 1104h) [Reset = 00000000h]

KEYWR is shown in Figure 15-2 and described in Table 15-4.

Return to the Summary Table.

Register to configure a key write operation

**Figure 15-2. KEYWR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| KEYSLOTSEL | | | | RESERVED | KEYSZSEL | | |
| R/W-0h | | | | R-0h | R/W-0h | | |

**Table 15-4. KEYWR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-4 | KEYSLOTSEL | R/W | 0h | Select the key slot to write the key into.<br>NOTE: SW needs to ensure that it is writing to the correct slots. The slot numbering is from 0 through SYS_N_SLOTS-1.<br>Each slot is a 128-bit slot. Therefore, when writing a 256-bit key, two slots will need to budgeted.<br>The selected slot for the 256-bit key needs to be an even numbered slot (0 or 2).<br>Also all the 256-bit keys need to be stored in lower numbered slots and 128-bit keys stored in higher numbered slots.<br>0h = Slot 0<br>1h = Slot 1<br>2h = Slot 2<br>3h = Slot 3<br>4h = Slot 4<br>5h = Slot 5<br>6h = Slot 6<br>7h = Slot 7 |
| 3 | RESERVED | R | 0h | |
| 2-0 | KEYSZSEL | R/W | 0h | Key size selection. Selection of 128 or 256 bit keys<br>0h = 256 bit key<br>1h = 128 bit key |



Copyright © 2026 Texas Instruments Incorporated

### 15.3.3 KEYRD (Offset = 1108h) [Reset = 00000000h]

KEYRD is shown in Figure 15-3 and described in Table 15-5.

Return to the Summary Table.

Register to configure a key transfer operation - transfer from keystore into a crypto engine

**Figure 15-3. KEYRD**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | CRYPTOSEL | |
| R-0h | | | | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| KEYSLOTSEL | | | | RESERVED | KEYSZSEL | | |
| R/W-0h | | | | R-0h | R/W-0h | | |

**Table 15-5. KEYRD Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-10 | RESERVED | R | 0h | |
| 9-8 | CRYPTOSEL | R/W | 0h | Crypto engine selector |
| 7-4 | KEYSLOTSEL | R/W | 0h | Select the key slot to read the key from.<br>NOTE: SW needs to ensure that it is reading from the correct slots.<br>The slot numbering is from 0 through SYS_N_SLOTS-1.<br>Each slot is a 128-bit slot. Therefore, when reading a 256-bit key, two adjacent slots will be read.<br>For a 256-bit key, an even numbered slot index must be specified. |
| 3 | RESERVED | R | 0h | |
| 2-0 | KEYSZSEL | R/W | 0h | Key size selection. Selection of 128 or 256 bit keys<br>0h = 256 bit key<br>1h = 128 bit key |



Copyright © 2026 Texas Instruments Incorporated

### 15.3.4 STATUS (Offset = 110Ch) [Reset = 00020001h]

STATUS is shown in Figure 15-4 and described in Table 15-6.

Return to the Summary Table.

Status register

**Figure 15-4. STATUS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | NKEYSLOTS | |
| R-0h | | | | | | R-2h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | VALID | | | |
| R-0h | | | | R-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| VALID | | | | STAT | | | |
| R-0h | | | | R-1h | | | |

**Table 15-6. STATUS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-18 | RESERVED | R | 0h | |
| 17-16 | NKEYSLOTS | R | 2h | Size of key storage: Number of 128-bit key slots<br>0h = Two slots<br>1h = Three slots<br>2h = Four slots |
| 15-12 | RESERVED | R | 0h | |
| 11-4 | VALID | R | 0h | Bit vector of valid bits to indicate which slots have been configured |
| 3-0 | STAT | R | 1h | Status information<br>0h = Valid configuration<br>1h = Key-store has not been configured. NK256 has not been set.<br>2h = Invalid value for NK256 field in CFG.<br>3h = Busy receiving a key deposit<br>4h = Busy transmitting a key to a crypto engine<br>5h = Invalid key slot selection for writing<br>6h = Invalid key slot selection for reading |



Copyright © 2026 Texas Instruments Incorporated

### 15.3.5 KEYIN (Offset = 1110h) [Reset = 00000000h]

KEYIN is shown in Figure 15-5 and described in Table 15-7.

Return to the Summary Table.

Write 32-bit key value by writing to this register. This is not readable.

**Figure 15-5. KEYIN**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 15-7. KEYIN Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | W | 0h | 32-bit key data |



Copyright © 2026 Texas Instruments Incorporated

### 15.3.6 KEYLOCK (Offset = 1114h) [Reset = 00000000h]

KEYLOCK is shown in Figure 15-6 and described in Table 15-8.

Return to the Summary Table.

Writing 1 to this will lock the key access to the particular one hot encoded slot

**Figure 15-6. KEYLOCK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | LOCKBIT | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R/W-0h | | | | | | | |

**Table 15-8. KEYLOCK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | LOCKBIT | R/W | 0h | Bit vector of lock bits to lock certain slots from being used by application. |



Copyright © 2026 Texas Instruments Incorporated

Chapter 16

# TRNG

The true random number generator (TRNG) block is an entropy source which can be used to generate random bit sequences.


**16.1 TRNG Overview** ..... 990
**16.2 TRNG Operation** ..... 990
**16.3 TRNG Registers** ..... 997





Copyright © 2026 Texas Instruments Incorporated

## 16.1 TRNG Overview

The true random number generator (TRNG) module securely generates random 32-bit numbers through an internal analog entropy source and digital conditioning/decimation blocks. The TRNG can be used as an entropy source to derive true random seed values when implementing a deterministic random bit generator (DRBG). The TRNG is designed for use in creating a deterministic random bit generator (DRBG) system which can pass the NIST SP800-22 statistical test suite for cryptographic random number generators.

Key features of the TRNG module include:

- 32-bit true random number output
- Functional across the entire device supply range (voltage and temperature)
- ΔΣ-modulator based analog entropy source
- Conditioning block
- Configurable decimation block
- Integrated startup and continuous health tests, compliant to NIST SP800-22
- Dedicated internal LDO regulator to defend against power manipulation attacks

## 16.2 TRNG Operation

The TRNG contains two main components: the analog block (containing the LDO regulator and entropy source), and the digital block (containing the state machine, startup logic, conditioning, decimation, health tests, and register interface). Figure 16-1 shows the TRNG blocks.

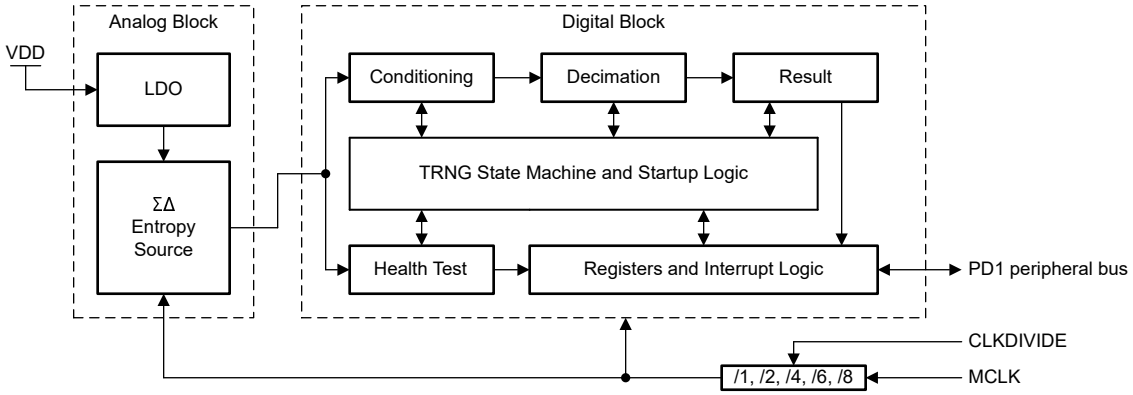



**Figure 16-1. TRNG Block Diagram**

### *16.2.1 TRNG Generation Data Path*

The random number generation data path begins in the analog block. The analog block contains a dedicated low drop-out regulator which supplies power to the entropy source, improving resistance to power manipulation attacks on the MCU supply. The sigma-delta entropy source itself derives entropy through delta-sigma modulation of Johnson-Nyquist noise.

During operation, the entropy source output is digitized and the digital output is sent to the digital conditioning block to generate a random bit stream. The conditioning block implements a stream cipher scheme. Following the conditioning block, a decimation block is provided to boost entropy by accumulating the entropy of a configurable number of samples. The output of the decimation block is finally captured in a result word holding register to be read by application software.

**Note**

The purpose of the decimation block is to increase the overall entropy of the TRNG by merging entropy from multiple samples. The decimation block will accumulate bits from the conditioning block, and decimate down to 1 sample every *n* samples, where *n* is the decimation value. Decimation is implemented as a bit-wise exclusive-or function of *n* conditioned bits. The decimation integer *n* can be set between 1 and 8.



Copyright © 2026 Texas Instruments Incorporated

### 16.2.2 Clock Configuration and Output Rate

The TRNG functional clock is derived from MCLK. The TRNG requires a functional clock which is within a specified frequency range for proper operation. Review the device-specific data sheet for the allowed TRNG frequency range on a given device. A clock divider is included in the TRNG to derive the frequency for the TRNG to use. The clock divider is specified through the RATIO field in the CLKDIVIDE register in the TRNG. This field must be set after the TRNG is enabled (through the PWREN register) but before the TRNG state is moved from the OFF state. Running the TRNG at a frequency outside the range specified in the device data sheet can lead to unexpected behavior.

The output rate of the TRNG module is dependent upon the TRNG functional clock frequency and the selected decimation rate. A minimum of 32 cycles of the TRNG functional clock are required to capture 32 random bits when the decimation rate is set to DECIM_RATE=0x0 (decimate-by-1, or no decimation). When the decimation rate is set to DECIM_RATE=0x3 (decimate-by-4), 128 cycles are required to capture 32 random bits. At a 10MHz clock rate with the decimation rate set to decimate-by-4, the time required to capture 32 random data bits is 12.8µs. Equation 16 gives the formula for calculating the time required to generate 32 random bits based on the TRNG functional clock frequency and the selected decimation rate.

$$t_{GENERATE}=(32*(DECIM\_RATE+1) / f_{TRNG} \quad (16)$$

**Note**

A decimation rate of decimate-by-4 (DECIM_RATE=0x3) or greater is required to obtain the minimum entropy which is required to pass the NIST SP800-22 statistical test suite. TI recommends using decimate-by-4 or greater when using the TRNG in cryptographic applications.

**Note**

When the DECIM_RATE field is changed, a NORM_FUNC command must be re-sent to the TRNG for the new rate to take effect (by writing 0x3 to the CMD field in the CTL register).

When 32 bits of random data are captured and available in the DATA_CAPTURE register, the TRNG asserts the IRQ_CAPTURED_RDY interrupt to indicate to the processor that data is ready. Once the data is read by the processor, capture of the next 32 bits of random data begins.

### 16.2.3 Behavior in Low Power Modes

The TRNG is available for use in RUN and SLEEP modes. It is not available in any other operating mode, and all TRNG configuration settings and data are lost when the device transitions to any operating mode below SLEEP. When using low power operating modes below SLEEP, the TRNG must be re-configured for use as needed when waking the device from STOP, STANDBY, or SHUTDOWN mode.

### 16.2.4 Health Tests

Three health test mechanisms are provided: a digital block power-on self-test, analog block power-on self-test, and a runtime self-test. The digital and analog power-on self-tests are executed by sending the corresponding CMD to the TRNG state machine when the TRNG is in the NORM_FUNC state. The runtime self-test is always running when the TRNG is in the NORM_FUNC state.

#### 16.2.4.1 Digital Block Startup Self-Test

The digital startup health test is run by application software when powering up the TRNG module. This built-in self-test verifies that the digital block is functioning properly by running predefined sequences of digital samples through the complete digital block while checking for expected outputs. The test sequence includes eight tests. Each test requires 1024 TRNG clock cycles to complete, as 1024 samples are input to the digital block for each test.

The results of the digital startup self-test are reported in the DIG_TEST field of the TEST_RESULTS register. Each test reports its status in an individual result bit. As each test passes, the corresponding bit in DIG_TEST is set by hardware. Upon completion of the digital start up self test, a pass condition is indicated if all eight bits



Copyright © 2026 Texas Instruments Incorporated

in the DIG_TEST field of the TEST_RESULTS register are set. If any bit in the DIG_TEST field is not set, this indicates a failing test. The TRNG is not usable if any of the digital startup health tests fail.

**Note**

The digital self-test changes the decimation rate when executing. This is a part of the self-test feature.

**Note**

The digital self-test injects deterministic values into the data path as a part of the test sequence. Following completion of the digital block startup self-test, the first data read from the DATA_CAPTURE register in normal (NORM_FUNC) operating mode is a deterministic value from the digital self-test, and not a true random number. **Always discard the first DATA_CAPTURE value after running the digital startup self-test.**

#### 16.2.4.2 Analog Block Startup Self-Test

The analog startup health test is run by application software when powering up the TRNG module. This test verifies that the analog block is functioning properly by capturing 4,096 consecutive analog samples and verifying that the samples pass a health test.

If the health test fails, the health fail (IRQ_HEALTH_FAIL) and command done (IRQ_CMD_DONE) interrupts are both asserted, and the TRNG state machine enters the ERROR state. The result of the analog start up self-test is also reported in the ANA_TEST bit in the TEST_RESULTS register. If the test passed, the ANA_TEST bit is set. If the test failed, the ANA_TEST bit is left cleared.

If the test does fail, this indicates that a severe loss of entropy in the analog block was detected during the test. This means that the entropy source can not be used in its current state. The probability of this test returning a false positive (probability that a failure was asserted but the analog block is operating correctly) is statistically very low, but not zero. This means that there is a small probability that a failure could be indicated in certain cases even if the TRNG analog block is operating correctly. To ensure that a test failure is legitimate and that there is not a loss of entropy, the analog startup health test can be repeated to validate the failure. In the event of a failure, the following procedure is recommended:

1. Clear the IRQ_HEALTH_FAIL interrupt
2. Power off the TRNG
3. Run the analog startup health test again
4. Follow the appropriate action below based on the test result:
   a. If the test passed, the TRNG can be used
   b. If the test failed a second time, go to step 1 and attempt to run the test a third time
   c. If the test failed a third time, there is catastrophic entropy loss and the TRNG should not be used

#### 16.2.4.3 Runtime Health Test

During normal operation of the TRNG, raw data from the entropy source is sent through the health test logic to ensure that there is sufficient entropy present on an ongoing basis. The tests are designed to statistically check for a minimum of 0.3 bits of entropy per sample. The TRNG implements both repetition count and adaptive proportion continuous self-tests.

##### *16.2.4.3.1 Repetition Count Test*

The repetition count test quickly detects failures that cause the entropy source to remain stuck on a single output value for an extended period of time. The repetition count test fails if the entropy source outputs the same bit value for 135 consecutive samples.

##### *16.2.4.3.2 Adaptive Proportion Test*

The adaptive proportion test detects failures that cause a disproportionate number of samples to be the same bit value and/or bit pattern over a window of 1024 samples. The adaptive proportion test fails any of the conditions



Copyright © 2026 Texas Instruments Incorporated

in Table 16-1 are violated. For example, the test will fail if more than 912 samples within a 1024 sample window are the same bit value.

**Table 16-1. Adaptive Proportion Test Bounds**

| Bit Pattern | Allowable Range of Bit Pattern Occurrences |
|---|---|
| 1 | $112 < n < 912$ |
| 10 | $211 < n < 341$ |
| 001 | $97 < n < 192$ |
| 1011 | $11 < n < 104$ |

The bit patterns in Table 16-1 used for adaptive proportion testing are designed to effectively detect catastrophic loss of entropy in the TRNG while also minimizing the occurrence of false positive failure detections.

#### *16.2.4.3.3 Handling Runtime Health Test Failures*

In the event that the run-time health test identifies that insufficient entropy can be present, the health fail (IRQ_HEALTH_FAIL) interrupt is asserted and the TRNG state machine enters the ERROR state. Due to the random nature of the TRNG module, it is possible for a health failure to be indicated in some cases even when the TRNG is working properly. To determine if there is a true loss of entropy, the following procedure is recommended:

1. Clear the IRQ_HEALTH_FAIL interrupt
2. Power off the TRNG
3. Power on the TRNG to normal mode again
   a. If the health failure is not asserted again, the TRNG can be used
   b. If the health test fails a second time, go to step 1 and attempt to run the test a third time
   c. If the health test fails a third time, there is catastrophic entropy loss and the TRNG should not be used

The TRNG does provide diagnostic information to indicate which test(s) failed. To determine which runtime test(s) are failing, check the REP_FAIL and ADAP_FAIL bits in the STAT register for the repetition test and adaptive proportion test, respectively.

### *16.2.5 Configuration*

The TRNG module is configured and used by application software and is accessed through the PD1 CPU-only peripheral bus. The DMA does not have access to the TRNG.

The TRNG contains a state machine which manages the current operating mode of the TRNG. Application software can configure the TRNG through commands to change the current state. The TRNG also sources an interrupt to the CPU to communicate status.

**Note**

Writing to a TRNG configuration register before a CMD is written to the TRNG can cause an unexpected command fail interrupt request (IRQ_CMD_FAIL) to be asserted. Mask the TRNG command fail interrupt when writing to TRNG configuration registers until a CMD is written to the TRNG. Clear the IRQ_CMD_FAIL interrupt status after writing the CMD and before enabling the IRQ_CMD_FAIL interrupt.

#### 16.2.5.1 TRNG State Machine

The TRNG includes a state machine which manages the current operating state of the TRNG. The TRNG is configured for use by sending commands to the TRNG to change its state. The TRNG has seven states, given in Table 16-2.



Copyright © 2026 Texas Instruments Incorporated

**Table 16-2. TRNG States**

| Code | State Name | Description |
|---|---|---|
| 0x0 | OFF | Analog block is powered off, digital block is disabled |
| 0x1 | PWRUP_ES | TRNG is in the process of powering up the entropy source and will transition to NORM_FUNC. |
| 0x2 | PWRDOWN_ES | TRNG is in the process of powering down the entropy source and will transition to OFF. |
| 0x3 | NORM_FUNC | TRNG is running and generating random bits normally. |
| 0x7 | TEST_DIG | TRNG is running the digital power-on self-test. |
| 0xB | TEST_ANA | TRNG is running the analog power-on self-test. |
| 0xA | ERROR | Operation halted due to an error condition. The analog remains powered but the health and conditioning logic is stopped. |

The current TRNG state can be determined by application software. To check the current state, read the FSM_STATE field in the STAT register of the TRNG.

#### *16.2.5.1.1 Changing TRNG States*

Application software can set the operating state of the TRNG by writing a new command value to the CMD field in the CTL register. While the TRNG state machine has 7 states, only 4 of the 7 states are entered by user control with CMDs (OFF, TEST_DIG, TEST_ANA, and NORM_FUNC). The other 3 states (PWRUP_ES, PWRDOWN_ES, and ERR) are entered by a hardware only when powering up or down the analog block or when an error condition occurs. The operating state change commands for the TRNG are given in Table 16-3.

**Table 16-3. TRNG Operating Mode Commands**

| CMD | Mode | Description |
|---|---|---|
| 0x0 | PWROFF | The TRNG analog blocks are powered off, and the TRNG digital blocks are clock gated. |
| 0x1 | TEST_DIG | The TRNG digital power-on self-tests are executed and the result is reported in the TEST_RESULTS register. |
| 0x2 | TEST_ANA | The TRNG analog power-on self-tests are executed and the result is reported in the TEST_RESULTS register. |
| 0x3 | NORM_FUNC | The TRNG is put into normal operating mode, in which samples from the entropy source are collected, conditioned, decimated, and placed in the DATA_CAPTURE register. The continuous statistical health tests run in this mode. |

From a given state, only certain state changes are possible. For example, from OFF only a CMD to switch to NORM_FUNC is supported. From NORM_FUNC, CMDs to switch to TEST_DIG, TEST_ANA, and OFF are supported. From the ERR state, only a switch to OFF is supported. The state flow for the TRNG is given in Figure 16-2.



Copyright © 2026 Texas Instruments Incorporated

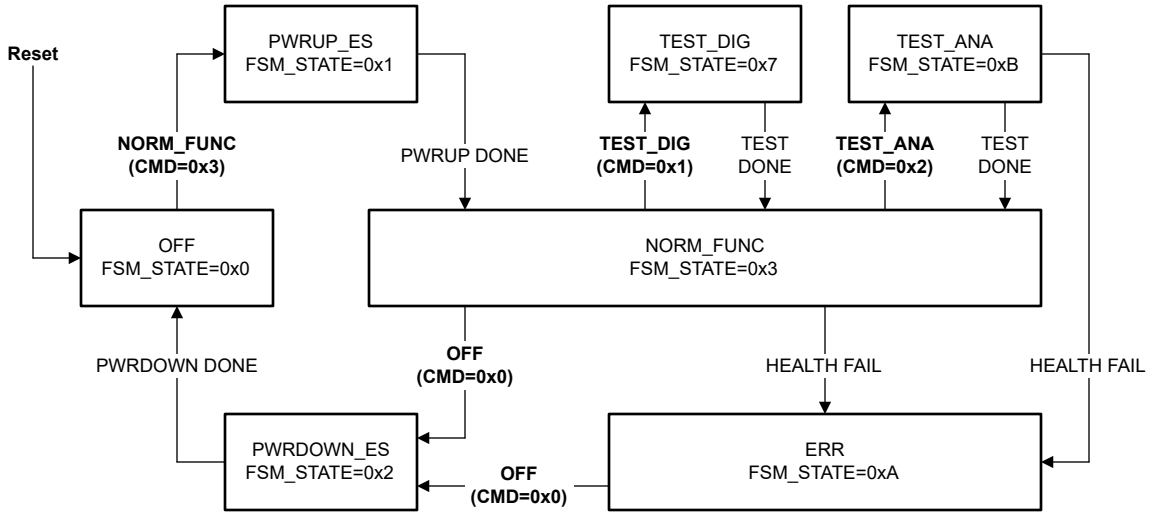



**Figure 16-2. TRNG State Flow**

A new command can only be written to the CMD field after any previously issued commands have run to completion. If a command is written at an invalid time, it will be rejected and a command fail interrupt will be asserted.

#### 16.2.5.2 Using the TRNG

Use the following procedure to start the TRNG:

1. Enable the TRNG by setting the ENABLE bit together with the KEY in the TRNG PWREN register.
2. Configure the TRNG clock divider to ensure that the TRNG functional clock is within the allowable range (10MHz typical, see the device data sheet for additional detail). The clock divider is configured by programming the required value to the RATIO field of the CLKDIVIDE register. As an example, if MCLK is 80MHz, the RATIO field shall be set to 0x7 (divide-by-8) to provide a 10MHz functional clock to the TRNG module.
3. Verify that the TRNG interrupts are disabled (interrupt mask bits are cleared to mask interrupts).
4. Move the TRNG from the default OFF state to the NORM_FUNC state by writing the NORM_FUNC command (0x3) to the CMD field in the CTL register of the TRNG. Wait for the IRQ_CMD_DONE interrupt flag to be set, indicating that the CMD completed.
5. Run the digital block start-up self-test routine to ensure the TRNG digital is functioning properly:
   a. Move the TRNG from the NORM_FUNC state to the TEST_DIG state by writing the TEST_DIG command (0x1) to the CMD field in the CTL register. Wait for the IRQ_CMD_DONE interrupt flag to be set, indicating that the digital self-test has completed.
   b. Check that all 8 digital tests passed be ensuring the DIG_TEST field in the TEST_RESULTS register are set (DIG_TEST=0xFF).
   c. After the digital test, the TRNG will return to the NORM_FUNC state automatically.
6. Run the analog block start-up self-test routine to ensure that the TRNG analog is functioning properly:
   a. Move the TRNG from the NORM_FUNC state to the TEST_ANA state by writing the TEST_ANA command (0x2) to the CMD field in the CTL register. Wait for the IRQ_CMD_DONE interrupt flag to be set, indicating that the analog self-test has completed.
   b. Check that the analog test passed by verifying that the ANA_TEST bit in the TEST_RESULTS register was set.
   c. After the analog test, if the test passed the TRNG will return to the NORM_FUNC state automatically. If the test failed, the TRNG enters the ERR state and must be brought to the OFF state before attempting to use it again.
7. Configure the TRNG for normal operation after running start-up self-tests:



Copyright © 2026 Texas Instruments Incorporated

a. Clear the IRQ_CAPTURED_RDY_IRQ status by setting the corresponding ICLR bit, as this may have been set during the self-tests performed earlier.
b. Set the decimation rate to the desired value by programming the new decimation rate into the DECIM_RATE field of the CTL register, followed by sending the NORM_FUNC command again (by writing 0x3 to the CMD field in the CTL register). A decimation rate of 4 (DECIM_RATE=0x3) or greater is recommended.
c. Enable the health fail interrupt by setting the IRQ_HEALTH_FAIL bit in the IMASK register.
d. Enable the data captured interrupt by setting the IRQ_CAPTURED_RDY bit in the IMASK register.

8. Wait for the first IRQ_CAPTURED_RDY IRQ. After the IRQ_CAPTURED_RDY IRQ is raised, clear the RIS register using the ICLR register and then read the DATA_CAPTURE register. This value (the first value read from DATA_CAPTURE after running a startup self-test) is not a true random value and must be read and discarded before collecting true random data from the DATA_CAPTURE register.
9. When the IRQ_CAPTURED_RDY IRQ is again asserted, clear the RIS register using the ICLR register and then read the DATA_CAPTURE register. Random bits will now be available for read-out in the DATA_CAPTURE register.
10. If the IRQ_HEALTH_FAIL IRQ is asserted, a low entropy condition was found and the TRNG will have automatically switched to the ERR state to stop operation. To exit the ERR state, clear the IRQ_HEALTH_FAIL interrupt. Then, transition the TRNG to the OFF state by sending an OFF command (0x0) to the CMD field in the CTL register. Wait for the IRQ_CMD_DONE interrupt flag to be set, then return to step #2 to power up the TRNG again to test if sufficient entropy is again available.

#### 16.2.5.3 TRNG Events

The TRNG module contains one event publisher and no event subscribers. One event publisher (CPU_INT) manages TRNG interrupt requests (IRQs) to the CPU subsystem through a static event route.

Table 16-4 lists the TRNG events.

**Table 16-4. TRNG Events**

| Event | Type | Source | Destination | Route | Configuration | Functionality |
|---|---|---|---|---|---|---|
| CPU Interrupt Event | Publisher | TRNG | CPU Subsystem | Static route | CPU_INT registers | Fixed interrupt route from TRNG to CPU |

##### *16.2.5.3.1 CPU Interrupt Event Publisher (CPU_INT)*

The TRNG module provides four interrupt sources which can be configured to source a CPU interrupt event. Table 16-5 lists the TRNG interrupt conditions.

**Table 16-5. TRNG CPU Interrupt Conditions (CPU_INT)**

| Index (IIDX) | Name | Description |
|---|---|---|
| 1 | IRQ_HEALTH_FAIL | Indicates that a health test has failed. |
| 2 | IRQ_CMD_FAIL | Indicates that a CMD which was issued has failed. |
| 3 | IRQ_CMD_DONE | Indicates that a CMD which was issued has completed. |
| 4 | IRQ_CAPTURED_RDY | Indicates that a new 32-bit data word containing random bits is available to be read by the processor. |

The CPU interrupt event configuration is managed with the CPU_INT event management registers. See Section 7.2.5 for guidance on configuring these registers for CPU interrupts.



Copyright © 2026 Texas Instruments Incorporated

## 16.3 TRNG Registers

Table 16-6 lists the memory-mapped registers for the TRNG registers. All register offset addresses not listed in Table 16-6 should be considered as reserved locations and the register contents should not be modified.

**Table 16-6. TRNG Registers**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 800h | PWREN | Power enable | | Go |
| 804h | RSTCTL | Reset Control | | Go |
| 814h | STAT | Status Register | | Go |
| 1020h | IIDX | Interrupt index | CPU_INT | Go |
| 1028h | IMASK | Interrupt mask | CPU_INT | Go |
| 1030h | RIS | Raw interrupt status | CPU_INT | Go |
| 1038h | MIS | Masked interrupt status | CPU_INT | Go |
| 1040h | ISET | Interrupt set | CPU_INT | Go |
| 1048h | ICLR | Interrupt clear | CPU_INT | Go |
| 10FCh | DESC | Module descriptions | | Go |
| 1100h | CTL | Controls the command and decimation rate | | Go |
| 1104h | STAT | Status register that informs health test results and last issued command | | Go |
| 1108h | DATA_CAPTURE | Captured word buffer of RNG data | | Go |
| 110Ch | TEST_RESULTS | Test results from TEST_ANA and TEST_DIG | | Go |
| 1110h | CLKDIVIDE | Clock Divider | | Go |

Complex bit access types are encoded to fit into small table cells. Table 16-7 shows the codes that are used for access types in this section.

**Table 16-7. TRNG Access Type Codes**

| Access Type | Code | Description |
|---|---|---|
| **Read Type** | | |
| R | R | Read |
| **Write Type** | | |
| W | W | Write |
| WK | W<br>K | Write<br>Write protected by a key |
| **Reset or Default Value** | | |
| *-n* | | Value after reset or the default value |



Copyright © 2026 Texas Instruments Incorporated

### 16.3.1 PWREN (Offset = 800h) [Reset = 00000000h]

PWREN is shown in Figure 16-3 and described in Table 16-8.

Return to the Summary Table.

Register to control the power state

**Figure 16-3. PWREN**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | ENABLE |
| R/W-0h | | | | | | | R/WK-0h |

**Table 16-8. PWREN Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | KEY to allow Power State Change<br>26h = KEY to allow write access to this register |
| 23-1 | RESERVED | R/W | 0h | |
| 0 | ENABLE | R/WK | 0h | Enable the power<br>KEY must be set to 26h to write to this bit.<br>0h = Disable Power<br>1h = Enable Power |



Copyright © 2026 Texas Instruments Incorporated

### 16.3.2 RSTCTL (Offset = 804h) [Reset = 00000000h]

RSTCTL is shown in Figure 16-4 and described in Table 16-9.

Return to the Summary Table.

Register to control reset assertion and de-assertion

**Figure 16-4. RSTCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | RESETSTKYCLR | RESETASSERT |
| W-0h | | | | | | WK-0h | WK-0h |

**Table 16-9. RSTCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | Unlock key<br>B1h = KEY to allow write access to this register |
| 23-2 | RESERVED | W | 0h | |
| 1 | RESETSTKYCLR | WK | 0h | Clear the RESETSTKY bit in the STAT register<br>KEY must be set to B1h to write to this bit.<br>0h = Writing 0 has no effect<br>1h = Clear reset sticky bit |
| 0 | RESETASSERT | WK | 0h | Assert reset to the peripheral<br>KEY must be set to B1h to write to this bit.<br>0h = Writing 0 has no effect<br>1h = Assert reset |



Copyright © 2026 Texas Instruments Incorporated

### 16.3.3 STAT (Offset = 814h) [Reset = 00000000h]

STAT is shown in Figure 16-5 and described in Table 16-10.

Return to the Summary Table.

peripheral enable and reset status

**Figure 16-5. STAT**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R- | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | RESETSTKY |
| R- | | | | | | | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R- | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | |
| R- | | | | | | | |

**Table 16-10. STAT Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-17 | RESERVED | R | 0h | |
| 16 | RESETSTKY | R | 0h | This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register<br>0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register<br>1h = The peripheral was reset since the last bit clear |
| 15-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 16.3.4 IIDX (Offset = 1020h) [Reset = X]

IIDX is shown in Figure 16-6 and described in Table 16-11.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. 0h means no event pending. Interrupt 1 is the highest priority, 2 next highest, 4, 8, … $2^{31}$ is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred.
On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in the RIS and MIS are cleared as well. After a read from the CPU (not from the debug interface), the register must be updated with the next highest priority interrupt, if none are pending, then it should display 0h.

**Figure 16-6. IIDX**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | STAT | | | | | | | |
| R-X | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | |

**Table 16-11. IIDX Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | X | |
| 7-0 | STAT | R | 0h | Interrupt index status<br>0h = No bit is set means there is no pending interrupt request<br>1h = Indicates that a health test has failed. The TRNG is in an error state until the interrupt is cleared.<br>2h = Indicates that the just issued command was rejected and is not being performed.<br>3h = Indicates that the current command/mode is done. This may have different meanings based on the mode:<br>OFF --> Power has been turned off<br>PWRUP_DIG --> Digital powerup tests are done<br>PWRUP_ANA --> Analog powerup tests are done<br>NORM_FUNC --> No IRQ, since mode runs indefinitely until a new command is issued<br>4h = Indicates that the captured word buffer is ready to be copied to memory |



Copyright © 2026 Texas Instruments Incorporated

### 16.3.5 IMASK (Offset = 1028h) [Reset = X]

IMASK is shown in Figure 16-7 and described in Table 16-12.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt mask is set.

**Figure 16-7. IMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R/W-X | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R/W-X | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R/W-X | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | IRQ_CAPTURED_RDY | IRQ_CMD_DONE | IRQ_CMD_FAIL | IRQ_HEALTH_FAIL |
| R/W-X | | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

**Table 16-12. IMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R/W | X | |
| 3 | IRQ_CAPTURED_RDY | R/W | 0h | Mask for IRQ_CAPTURED_RDY. Indicates to the CPU that the Captured Word is ready to be read.<br>0h = Interrupt is masked out<br>1h = Interrupt will request an interrupt service routine and corresponding bit in MIS will be set |
| 2 | IRQ_CMD_DONE | R/W | 0h | Mask for IRQ_CMD_DONE. Indicates that a command has finished<br>0h = Interrupt is masked out<br>1h = Interrupt will request an interrupt service routine and corresponding bit in MIS will be set |
| 1 | IRQ_CMD_FAIL | R/W | 0h | Masked interrupt source for IRQ_CMD_FAIL. Indicates that the just issued command/mode has been rejected.<br>0h = Interrupt is masked out<br>1h = Interrupt will request an interrupt service routine and corresponding bit in MIS will be set |
| 0 | IRQ_HEALTH_FAIL | R/W | 0h | Mask for IRQ_HEALTH_FAIL. Indicates that a health test has failed.<br>0h = Interrupt is masked out<br>1h = Interrupt will request an interrupt service routine and corresponding bit in MIS will be set |



Copyright © 2026 Texas Instruments Incorporated

### 16.3.6 RIS (Offset = 1030h) [Reset = X]

RIS is shown in Figure 16-8 and described in Table 16-13.

Return to the Summary Table.

Raw interrupt status reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Figure 16-8. RIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-X | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-X | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-X | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | IRQ_CAPTURED_RDY | IRQ_CMD_DONE | IRQ_CMD_FAIL | IRQ_HEALTH_FAIL |
| R-X | | | | R-0h | R-0h | R-0h | R-0h |

**Table 16-13. RIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | X | |
| 3 | IRQ_CAPTURED_RDY | R | 0h | Indicates to the CPU that the Captured Word is ready to be read. Reading the IIDX will clear this interrupt.<br>0h = IRQ_CAPTURED_READY did not occur<br>1h = IRQ_CAPTURED_READY occurred |
| 2 | IRQ_CMD_DONE | R | 0h | Raw interrupt source for IRQ_CMD_DONE. Indicates that the issued command/mode has completed.<br>0h = IRQ_CMD_DONE did not occur<br>1h = IRQ_CMD_DONE occurred |
| 1 | IRQ_CMD_FAIL | R | 0h | Masked interrupt source for IRQ_CMD_FAIL. Indicates that the just issued command/mode has been rejected.<br>0h = IRQ_CMD_FAIL did not occur<br>1h = IRQ_CMD_FAIL occurred |
| 0 | IRQ_HEALTH_FAIL | R | 0h | Indicates to the CPU that any of the health tests have failed. Reading the IIDX will clear this interrupt.<br>0h = IRQ_CAPTURED_READY did not occur<br>1h = IRQ_CAPTURED_READY occurred |



Copyright © 2026 Texas Instruments Incorporated

### 16.3.7 MIS (Offset = 1038h) [Reset = X]

MIS is shown in Figure 16-9 and described in Table 16-14.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 16-9. MIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-X | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-X | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-X | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | IRQ_CAPTURED_RDY | IRQ_CMD_DONE | IRQ_CMD_FAIL | IRQ_HEALTH_FAIL |
| R-X | | | | R-0h | R-0h | R-0h | R-0h |

**Table 16-14. MIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | X | |
| 3 | IRQ_CAPTURED_RDY | R | 0h | Masked interrupt result for CAPTURED_READY. Indicates to the CPU that the Captured Word is ready to be read. Reading the IIDX will clear this interrupt.<br>0h = IRQ_CAPTURED_READY did not request an interrupt service routine<br>1h = IRQ_CAPTURED_READY requests an interrupt service routine |
| 2 | IRQ_CMD_DONE | R | 0h | Masked interrupt source for IRQ_CMD_DONE. Indicates that the issued command/mode has completed.<br>0h = IRQ_CAPTURED_READY did not request an interrupt service routine<br>1h = IRQ_CMD_DONE requests an interrupt service routine |
| 1 | IRQ_CMD_FAIL | R | 0h | Masked interrupt source for IRQ_CMD_FAIL. Indicates that the just issued command/mode has been rejected.<br>0h = IRQ_CMD_FAIL did not request an interrupt service routine<br>1h = IRQ_CMD_FAIL requests an interrupt service routine |
| 0 | IRQ_HEALTH_FAIL | R | 0h | Masked interrupt result for HEALTH_FAIL. Indicates to the CPU that any of the health tests have failed for the latest 1024-bit window.<br>0h = IRQ_CAPTURED_READY did not request an interrupt service routine<br>1h = IRQ_CAPTURED_READY requests an interrupt service routine |



Copyright © 2026 Texas Instruments Incorporated

### 16.3.8 ISET (Offset = 1040h) [Reset = 00000000h]

ISET is shown in Figure 16-10 and described in Table 16-15.

Return to the Summary Table.

ISET allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 16-10. ISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| W- | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| W- | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| W- | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | IRQ_CAPTURED_RDY | IRQ_CMD_DONE | IRQ_CMD_FAIL | IRQ_HEALTH_FAIL |
| W- | | | | W- | W- | W- | W- |

**Table 16-15. ISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | W | 0h | |
| 3 | IRQ_CAPTURED_RDY | W | 0h | Indicates to the CPU that the Captured Word is ready to be read. Reading the IIDX or DATA_CAPTURE registers will clear this interrupt.<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to CAPTURED_READY is set |
| 2 | IRQ_CMD_DONE | W | 0h | Write to turn on CMD_DONE IRQ. Indicates that the last issued TRNG command has finished.<br>0h = Writing a 0 has no effect.<br>1h = RIS bit corresponding to CMD_DONE is set |
| 1 | IRQ_CMD_FAIL | W | 0h | Masked interrupt source for IRQ_CMD_FAIL. Indicates that the just issued command/mode has been rejected.<br>0h = Writing a 0 has no effect.<br>1h = RIS bit corresponding to CMD_FAIL is set |
| 0 | IRQ_HEALTH_FAIL | W | 0h | Indicates to the CPU that any of the health tests have failed. Reading the IIDX or DATA_CAPTURE registers will clear this interrupt.<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to HEALTH_FAIL is set |



Copyright © 2026 Texas Instruments Incorporated

### 16.3.9 ICLR (Offset = 1048h) [Reset = 00000000h]

ICLR is shown in Figure 16-11 and described in Table 16-16.

Return to the Summary Table.

Write a 1 to clear corresponding Interrupt.

**Figure 16-11. ICLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| W- | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| W- | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| W- | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | IRQ_CAPTURED_RDY | IRQ_CMD_DONE | IRQ_CMD_FAIL | IRQ_HEALTH_FAIL |
| W- | | | | W- | W- | W- | W- |

**Table 16-16. ICLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | W | 0h | |
| 3 | IRQ_CAPTURED_RDY | W | 0h | Indicates to the CPU that the Captured Word is ready to be read. Reading the IIDX or DATA_CAPTURE registers will clear this interrupt.<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to CAPTURED_READY is cleared |
| 2 | IRQ_CMD_DONE | W | 0h | Write to turn off CMD_DONE IRQ. Indicates that the last issued TRNG command has finished.<br>0h = Writing a 0 has no effect.<br>1h = RIS bit corresponding to CMD_DONE is cleared |
| 1 | IRQ_CMD_FAIL | W | 0h | Masked interrupt source for IRQ_CMD_FAIL. Indicates that the just issued command/mode has been rejected.<br>0h = Writing a 0 has no effect.<br>1h = RIS bit corresponding to CMD_FAIL is cleared |
| 0 | IRQ_HEALTH_FAIL | W | 0h | Indicates to the CPU that any of the health tests have failed. Reading the IIDX or DATA_CAPTURE registers will clear this interrupt.<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to CAPTURED_READY is cleared |



Copyright © 2026 Texas Instruments Incorporated

### 16.3.10 DESC (Offset = 10FCh) [Reset = 05110000h]

DESC is shown in Figure 16-12 and described in Table 16-17.

Return to the Summary Table.

This register is used to specify clock source selection for any special modules that need to choose between MCLK and another special clock source. The register is not expected to be present on most standard SVT peripherals but may be present on modules such as the ADC. The register is not present on VDDCOREULP domain peripherals and will be reserved, reading 0.

**Figure 16-12. DESC**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MODULEID | | | | | | | | | | | | | | | |
| R-511h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| FEATUREVER | | | | INSTNUM | | | | MAJREV | | | | MINREV | | | |
| R-0h | | | | R-0h | | | | R-0h | | | | R-0h | | | |

**Table 16-17. DESC Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | MODULEID | R | 511h | Module Identifier - An internal TI page has been created to request unique module IDs |
| 15-12 | FEATUREVER | R | 0h | Feature Set for the module *instance* |
| 11-8 | INSTNUM | R | 0h | Instance Number within the device. This will be a parameter to the RTL for modules that can have multiple instances |
| 7-4 | MAJREV | R | 0h | Major rev of the IP |
| 3-0 | MINREV | R | 0h | Minor rev of the IP |



Copyright © 2026 Texas Instruments Incorporated

### 16.3.11 CTL (Offset = 1100h) [Reset = X]

CTL is shown in Figure 16-13 and described in Table 16-18.

Return to the Summary Table.

Affects various parameters of the TRNG system

**Figure 16-13. CTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R/W-X | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | PWRUP_PSTART_CFG | | PWRUP_PCHRG_CFG | | PWRUP_CLKDIV |
| R/W-X | | | R/W-2h | | R/W-2h | | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | DECIM_RATE | | |
| R/W-X | | | | | R/W-0h | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | CMD | |
| R/W-X | | | | | | R/W-0h | |

**Table 16-18. CTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-21 | RESERVED | R/W | X | |
| 20-19 | PWRUP_PSTART_CFG | R/W | 2h | Configure pulse startup sequence length<br>b00 = Disabled<br>b01 = rise at 10us, fall at 50us<br>b10 = rise at 10us, fall at 70us (default)<br>b11 = rise at 10us, fall at 90us |
| 18-17 | PWRUP_PCHRG_CFG | R/W | 2h | Configure PCHARGE sequence length<br>b00 = Disabled<br>b01 = 20 us PCHARGE<br>b10 = 30 us PCHARGE (default)<br>b11 = 40 us PCHARGE |
| 16 | PWRUP_CLKDIV | R/W | 0h | When '1', the power-up sequence takes twice as long (clock frequency halved) |
| 15-11 | RESERVED | R/W | X | |
| 10-8 | DECIM_RATE | R/W | 0h | Set decimation rate. Decimate by n<br>0x0 = Decimation by 1 (no decimation)<br>0x1 = Decimation by 2 (Skip every other sample)<br>…<br>0x7 = Decimation by 8 (Take every 8th sample) |
| 7-2 | RESERVED | R/W | X | |



Copyright © 2026 Texas Instruments Incorporated

**Table 16-18. CTL Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 1-0 | CMD | R/W | 0h | Sets the TRNG mode through a command. The mode will not be updated until the previous command is done, as indicated by IRQ_CMD_DONE.<br>00 --> OFF<br>01 --> PWRUP_DIG<br>10 --> PWRUP_ANA<br>11 --> NORM_FUNC<br>0h = Turns the power off of the analog source and clocks the digital interface<br>1h = Initiates the powerup test sequence for the digital components. This verifies that the digital components are properly working. IRQ_CMD_DONE indicates that the test is done. The results of this test are in bits 0:6 in TEST_RESULTS register<br>2h = Initiates the powerup test sequence for the analog TRNG. This verifies that the analog component is generating sequences with enough entropy. IRQ_CMD_DONE indicates that the test is done. The results of this test are in bit 7 in TEST_RESULTS register<br>3h = Normal operating mode for TRNG. All components are turned on. |



Copyright © 2026 Texas Instruments Incorporated

### 16.3.12 STAT (Offset = 1104h) [Reset = X]

STAT is shown in Figure 16-14 and described in Table 16-19.

Return to the Summary Table.

Status register that informs health test result, last issued command, and current FSM state

**Figure 16-14. STAT**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-X | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | FSM_STATE | | | |
| R-X | | | | R-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | ISSUED_CMD | |
| R-X | | | | | | R-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | REP_FAIL | ADAP_FAIL |
| R-X | | | | | | R-0h | R-0h |

**Table 16-19. STAT Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | RESERVED | R | X | |
| 19-16 | FSM_STATE | R | 0h | Current state of the front end FSM (behind a clock domain crossing). 2 reads are REQUIRED as there is a chance of metastability when reading this<br>States:<br>0000: OFF<br>0001: PWRUP_ES<br>0011: NORM_FUNC<br>0111: TEST_DIG<br>1011: TEST_ANA<br>1010: ERROR<br>0010: PWRDOWN_ES |
| 15-10 | RESERVED | R | X | |
| 9-8 | ISSUED_CMD | R | 0h | Indicates the last accepted command that is issued to the TRNG interface.<br>Upon writing a valid command, this register will update and the command will be in progress until CMD_DONE_IRQ is set. CMD_DONE_IRQ indicates that the state is in PWROFF, NORM_FUNC, or ERROR. These states will accept new commands.<br>00 --> OFF<br>01 --> PWRUP_DIG<br>10 --> PWRUP_ANA<br>11 --> NORM_FUNC |
| 7-2 | RESERVED | R | X | |
| 1 | REP_FAIL | R | 0h | Indicates that the repetition counter test caused the most recent failure. Thus, the health count numbers are most likely not for a complete 1024-bit window. |
| 0 | ADAP_FAIL | R | 0h | Indicates that the Adaptive Proportion Test (1,2,3, or 4-bit counters) failed by having too many or too few counted samples in the last 1024 bit window. |



Copyright © 2026 Texas Instruments Incorporated

### 16.3.13 DATA_CAPTURE (Offset = 1108h) [Reset = 00000000h]

DATA_CAPTURE is shown in Figure 16-15 and described in Table 16-20.

Return to the Summary Table.

Captured data from decimation block

**Figure 16-15. DATA_CAPTURE**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BUFFER | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 16-20. DATA_CAPTURE Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | BUFFER | R | 0h | Captured Data from the Decimation Block |



Copyright © 2026 Texas Instruments Incorporated

### 16.3.14 TEST_RESULTS (Offset = 110Ch) [Reset = X]

TEST_RESULTS is shown in Figure 16-16 and described in Table 16-21.

Return to the Summary Table.

Includes a bit describing which startup test failed. The first 8 bits check the condition of the digital components in the digital startup validation check and the 9th bit checks that the entropy source is producing samples that have enough entropy. This register will read all '0s when a command is not finished. Each of the tests are active low, indicating a failed test with '0' and a passed test with '1'.

**Figure 16-16. TEST_RESULTS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-X | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-X | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | ANA_TEST |
| R-X | | | | | | | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DIG_TEST | | | | | | | |
| R-0h | | | | | | | |

**Table 16-21. TEST_RESULTS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-9 | RESERVED | R | X | |
| 8 | ANA_TEST | R | 0h | Runs through 4096 samples from an enabled entropy source and verifies that none of the health tests failed, indicating sufficient entropy was produced by the analog components |
| 7-0 | DIG_TEST | R | 0h | Bit 0 indicates if the first decimation rate test and health test(verifies conditioning, decimation, and captured buffer) fails and Bit 1 indicates if the second decimation test and health test fails<br>Bit 0 - decim_test0 (decim = 0x0)<br>Bit 1 - decim_test1 (decim = 0x1) ... |



Copyright © 2026 Texas Instruments Incorporated

### 16.3.15 CLKDIVIDE (Offset = 1110h) [Reset = 00000000h]

CLKDIVIDE is shown in Figure 16-17 and described in Table 16-22.

Return to the Summary Table.

This register is used to specify module-specific divide ratio of the functional clock and it only supports even division for TRNG.

**Figure 16-17. CLKDIVIDE**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | | RATIO | | |
| R/W-0h | | | | | | | | | | | | | R/W-0h | | |

**Table 16-22. CLKDIVIDE Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | RESERVED | R/W | 0h | |
| 2-0 | RATIO | R/W | 0h | Selects divide ratio of module clock<br>0h = Do not divide clock source<br>1h = Divide clock source by 2<br>3h = Divide clock source by 4<br>5h = Divide clock source by 6<br>7h = Divide clock source by 8 |



Copyright © 2026 Texas Instruments Incorporated

Chapter 17

# Temperature Sensor

The PMU provides a basic temperature sensor which can be used to approximate the temperature of the device. The temperature sensor is connected internally to the ADC, and the ADC must be used to perform a temperature measurement. See the device-specific data sheet to determine the correct internal ADC channel to use when measuring the temperature sensor.

The temperature sensor outputs a voltage which has a linear relationship with temperature. The temperature coefficient ($TS_c$) is the slope of the temperature-voltage relationship (given in mV/C), and is given in the specifications section of the device-specific data sheet.

A unit-specific single-point trim value (TEMP_SENSE0.DATA) is provided in the factory constants memory of each device. This value indicates the temperature sensor output voltage at the factory trim temperature ($TS_{TRIM}$), in ADC result code format. The ADC result code in TEMP_SENSE0.DATA is based upon 12-bit sampling mode together with the a voltage reference used during factory calibration. The $TS_{TRIM}$ temperature is also given in the specifications section of the device-specific data sheet.

The approximate temperature of the device can be computed through the use of the following parameters:

- $TS_c$, taken from the device data sheet
- TEMP_SENSE0.DATA, taken from the unit-specific factory constants memory
- $TS_{TRIM}$, taken from the device data sheet
- $V_{SAMPLE}$ (voltage sample of the temperature sensor, converted from the ADC results and voltage reference)

The temperature is computed through the linear relationship given in Equation 17, where $V_{SAMPLE}$ is the temperature sensor voltage, and $V_{TRIM}$ is the factory calibrated temperature sensor voltage at the $TS_{TRIM}$ temperature (derived from TEMP_SENSE0.DATA):

$$\mathbf{T_{SAMPLE}} = (1 / TS_c) * (V_{SAMPLE} - V_{TRIM}) + TS_{TRIM} \quad (17)$$

The $ADC_{CODE}$ raw result can be converted to a voltage equivalent ($V_{SAMPLE}$) as shown in the relationship in Equation 18, where RES is the ADC resolution in bits, and VREF is the ADC reference voltage.

$$\mathbf{V_{SAMPLE}} = (VREF / 2^{RES}) * (ADC_{CODE} - 0.5) \quad (18)$$

### Example

To illustrate the process of converting an ADC sample of the temperature sensor into an approximate device temperature, an example is given below.

Example parameters (see device datasheet for actual parameters):

- $TS_c$ = -2.04mV/C
- TEMP_SENSE0.DATA = 1857 (ADC result code based on 12-bit mode and a 1.4V reference)
- $TS_{TRIM}$ = 30C
- $ADC_{CODE}$ = 1677 (ADC result code based on 12-bit mode and a 1.4V reference)

First, the current temperature sensor sample voltage is calculated using Equation 19:

$$\mathbf{V_{SAMPLE}} = (1.4V / 4096) * (1677 – 0.5) = \mathbf{0.5730V} \quad (19)$$



Copyright © 2026 Texas Instruments Incorporated

Then, the single-point calibration voltage is calculated using the same means:

$$\mathbf{V_{TRIM}} = (1.4V / 4096) * (1857 - 0.5) = \mathbf{0.6345V} \quad (20)$$

Then, the temperature is approximated using Equation 21:

$$\mathbf{T_{SAMPLE}} = (1 / -0.002044) * (0.5730V - 0.6345) + 30°C = \mathbf{60°C} \quad (21)$$

**Note**

The temperature sensor is not available in STANDBY and SHUTDOWN operating modes.



Copyright © 2026 Texas Instruments Incorporated

Chapter 18
# ADC

The ADC is a high-performance successive-approximation-register (SAR) analog-to-digital converter. This chapter describes the features and operation of the ADC peripheral.


**18.1 ADC Overview** ..... 1017
**18.2 ADC Operation** ..... 1018
**18.3 ADC12 Registers** ..... 1036





Copyright © 2026 Texas Instruments Incorporated

## 18.1 ADC Overview

The ADC supports measure analog signals and convert them to a digital representation with minimal CPU intervention.

The ADC supports fast 12-, 10-, and 8-bit analog-to-digital conversions. It implements a 12-bit SAR core, sample and conversion mode control, and up to 12 independent conversion-and-control buffers. The ADC allows up to 12 independent analog-to-digital converter (ADC) samples to be converted and stored without any CPU intervention.

ADC features include:

- Up to 4-Msps conversion rate at a resolution of 12 bits
- Integrated hardware oversampling for averaging up to 128 samples
- Full-scale ADC operating voltage range
- Sample-and-hold with programmable sampling periods controlled by software or timers
- Two sampling trigger sources: software trigger and event trigger
- Software-selectable on-chip reference voltage of 1.4V or 2.5V
- Configurable ADC reference source: VDD, internal reference, or external reference
- Individually configurable analog input channels
- Internal conversion channels for temperature sensing, supply monitoring, and analog signal chain (see the device-specific data sheet for availability and channel mapping)
- Configurable ADC clock source
- Single-channel, repeat-single-channel, sequence (autoscan), and repeat-sequence (repeated autoscan) conversion modes
- Conversion-result storage registers (MEMRES)
- Window comparator for low-power monitoring of input signals from conversion-result registers
- DMA support with interrupt event generation on completion of transfer
- Operates in RUN, SLEEP, and STOP modes
- Can be triggered in any operating mode except for SHUTDOWN
- Automatic power, reference, and clock control for low-power operation
- Support for simultaneous and synchronous operation (sample and conversion) of two ADCs in parallel

**Note**

Refer to the device-specific datasheet for the supported ADC features.

Figure 18-1 shows the functional block diagram of the ADC peripheral.



Copyright © 2026 Texas Instruments Incorporated

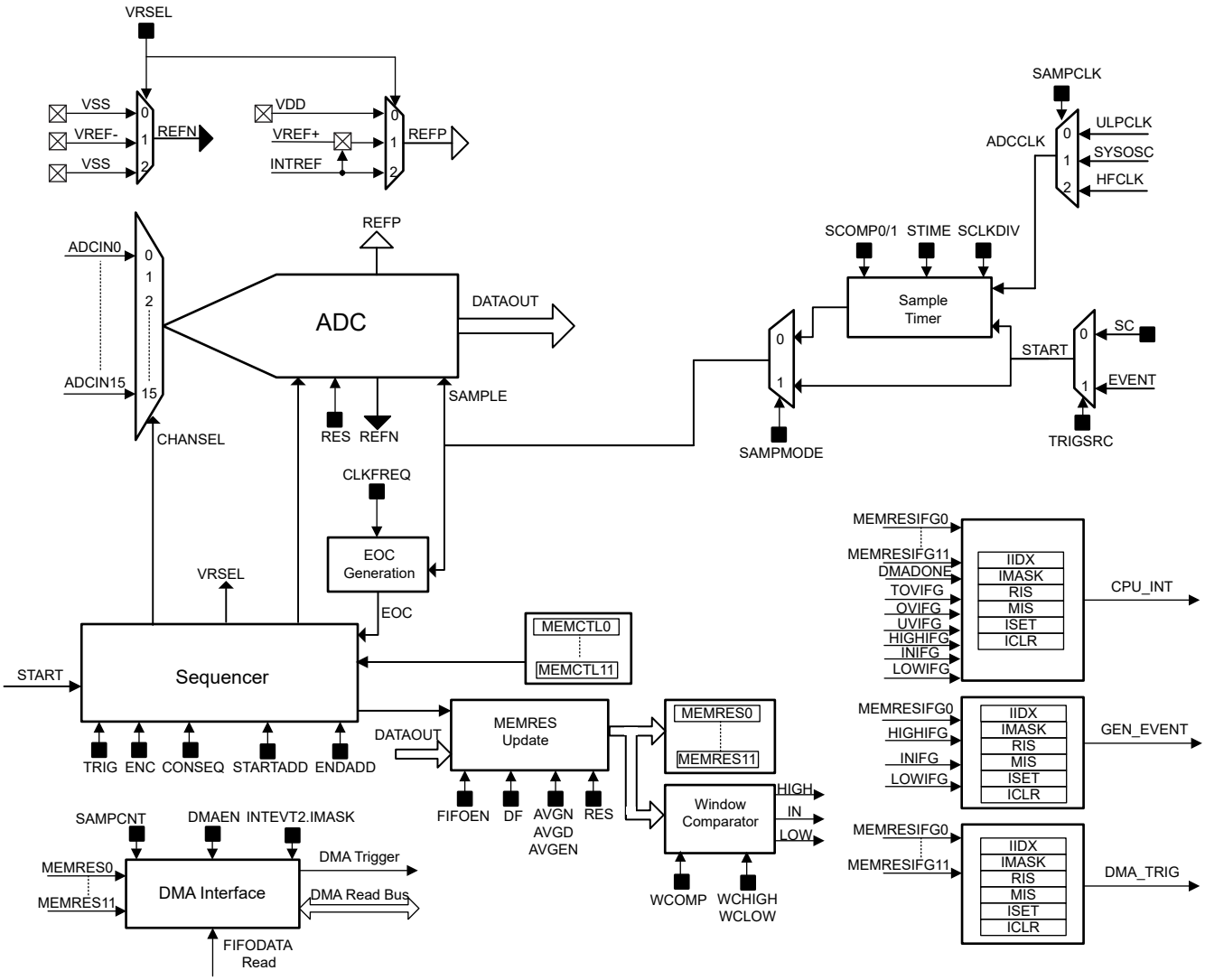



**Figure 18-1. ADC Block Diagram**

**Note**

The sample clock (SAMPCLK) is sourced from ULPCLK, SYSOSC or HFCLK, the conversion clock (CONVCLK) is sourced from an 80Mhz oscillator within the ADC IP.

## 18.2 ADC Operation

The ADC is configured with user software. The following sections describe the setup and operation of the ADC.



Copyright © 2026 Texas Instruments Incorporated

**Note**

The ADC result registers (MEMRES) are aliased between two address regions on the device:

- The primary region, through which all ADC registers can be accessed by the CPU or DMA at the ULPCLK rate
- The aliased region, through which the ADC MEMRES registers can be accessed by the CPU or DMA at the MCLK rate for fast read-out of ADC results. The aliased region is in the PD1 power domain.

On devices which run MCLK and ULPCLK at the same frequency, there is not a performance benefit to accessing the ADC MEMRES registers through the aliased region. On device configurations where MCLK is greater than ULPCLK, the aliased region is recommended to be used. However, application software can use the aliased region on all devices for MEMRES access to keep the software implementation consistent.

### *18.2.1 ADC Core*

The ADC core converts an analog input to its digital representation. The core uses two voltage levels ($V_{R+}$ and $V_{R-}$) to define the upper and lower limits of the conversion. The digital output ($N_{ADC}$) is full scale when the input signal is equal to or higher than $V_{R+}$ (VREFP), and is zero when the input signal is equal to or lower than $V_{R-}$(VREFN). The input channel and the positive reference voltage level ($V_{R+}$) are defined in the conversion-control memory.

Equation 22 below shows the conversion formula for the ADC result, $N_{ADC}$, for n-bit resolution mode.

$$N_{ADC} = \left(2^n - 1\right) \times \frac{(V_{in} + 0.5LSB) - V_{R-}}{V_{R+} - V_{R-}} \qquad \text{Where,} \quad LSB = \frac{V_{R+} - V_{R-}}{2^n} \tag{22}$$

**Note**

The supported $V_{R-}$ for this ADC is 0V. All subsequent equations and sections will reflect this inherent property.

Given that $V_{R-}$ is 0V in this ADC, the equation for $N_{ADC}$ becomes:

$$N_{ADC} = \left(2^n - 1\right) \times \frac{(V_{in} + 0.5LSB)}{V_{R+}} \qquad \text{Where,} \quad LSB = \frac{V_{R+}}{2^n} \tag{23}$$

Equation 24 below describes the input voltage at which the ADC output saturates:

$$V_{in} = V_{R+} - 1.5LSB \tag{24}$$

**Note**

The ADC is NOT functional in STANDBY or SHUTDOWN mode.

### *18.2.2 Voltage Reference Options*

The ADC voltage reference ($V_{R+}$) can be configured through the VRSEL bits in the MEMCTL register. Different reference sources can be selected for conversion on different channels. There are three options available for supplying a reference voltage to the ADC (refer to the device specific datasheet for the supported options):

- External reference supplied to the ADC through the VREF+ and VREF- pins
- MCU supply voltage (VDD)
- Configurable internal reference voltage (INTREF) of 1.4V and 2.5V from Vref module

When supplying an external reference to the ADC, connect the VREF+ pin to the reference source with the appropriate decoupling circuitry, and connect the VREF- pin to ground.



Copyright © 2026 Texas Instruments Incorporated

**Note**

Using reference options with VREF- will provide better noise performance, as the VREF- provides a dedicated ground for the reference.

The internal reference (VREF) is a voltage reference module that is shared by multiple analog peripherals including the ADC peripheral. The internal reference is available on the VREF+ pin and it is recommended to connect a decoupling capacitor of specified value on the device pin. Refer to the device-specific data sheet for the recommended capacitor value. When supplying the internal reference as the reference to the ADC, there is no restriction on the ADC sampling rate and thus can be operated at full speed.

### *18.2.3 Generic Resolution Modes*

The ADC supports operation in 12-bit (default), 10-bit, and 8-bit resolution modes. The resolution mode is configured using the RES bits in the CTL2 register.

- When 12-bit mode is selected, the conversion phase requires a total of 12 conversion clock cycles
- When 10-bit mode is selected, the conversion phase requires a total of 10 conversion clock cycles
- When 8-bit mode is selected, the conversion phase requires a total of 8 conversion clock cycles

The conversion clock (CONVCLK) is sourced from an 80MHz oscillator within the ADC IP. The conversion window is based on the resolution mode and the frequency of ADCCLK. For more details, refer to Figure 18-2 and Figure 18-3.

### *18.2.4 Hardware Averaging*

This ADC implements digital sample averaging in hardware (HW averaging) to efficiently increase the effective resolution of the ADC without the need for SW or CPU intervention. The HW averaging functionality is configured using the AVGN and AVGD bits in the CTL1 register.

- AVGN defines the number of conversions to accumulate for the current MEMCTLx
- AVGD defines what the accumulated value gets divided by using bit shifting

**Note**

The MEMRES result register is a maximum of 16 bits long. If not shifted appropriately, the result will be truncated.

**Table 18-1. Available Hardware Averaging Settings**

| Bit Field Value | AVGN Settings (number of samples accumulated) | AVGD Settings (number of bits to right shift) |
|---|---|---|
| 0x0 | 0 | 0 |
| 0x1 | 2 | 1 |
| 0x2 | 4 | 2 |
| 0x3 | 8 | 3 |
| 0x4 | 16 | 4 |
| 0x5 | 32 | 5 |
| 0x6 | 64 | 6 |
| 0x7 | 128 | 7 |

The averaging configuration is global and it holds for any channel that enables the averaging feature. It is not possible to have different averaging configurations defined per channel. The averaging feature for each individual channel can be enabled though the AVGEN bit in the MEMCTL register. When the sample trigger is received for a channel with averaging enabled, the required number of conversions are performed automatically back-to-back and the final averaged value is stored in the MEMRES register or FIFODAT.



Copyright © 2026 Texas Instruments Incorporated

**Note**

The data format must be selected as unsigned binary while using the hardware averaging feature.

### *18.2.5 ADC Clocking*

The ADC peripheral clock (ADCCLK) is provided by the Section 2.4 and is used for the sampling clock (SAMPCLK). SYSOSC , HFCLK and ULPCLK are the available clock sources available for ADCCLK, which can support up to 48Mhz,. Refer to the device-specific data sheet for supported ADCCLK frequencies. Using the ULPCLK, which is the bus clock for all peripherals, is very useful for deterministic start of samplingand simultaneous sampling. Using the HFCLK as the clock source for ADCCLK is useful for when a very accurate, low-jitter, sampling period is needed. The ADC clock source can be selected by programming the SAMPCLK bits in the CLKCFG register. The conversion clock is sourced from a dedicated 80MHz local oscillator which enables high speed 12-bit conversions up to 4Msps.

SYSOSC needs to be active for the ADC to operate properly. If SYSOSC is not running and the ADC is triggered, the ADC will automatically request SYSCTL to enable and set SYSOSC to base frequency during the conversion. If SYSOSC is already enabled, it will remain the same frequency. The only exception to this is in STOP1 operating mode where SYSOSC will go to base frequency when the ADC is triggered.

In order to provide a way to ensure predictable sample rate operation between power modes, the CCONRUN and CCONSTOP bits can be set to signal the ADC that it can expect that the SYSOSC will already be ON when the device is in RUN and STOP modes respectively. When these bits are set, the ADC will not wait for an ACK from SYSCTL to make sure SYSOSC is running before starting sampling. This feature gives users the flexibility to save power in applications where deterministic sample timing is not a requirement. Refer to Section 18.2.6 for examples on how to properly use the CCONRUN and CCONSTOP control bits.

The user must configure the FRANGE bits in the CLKFREQ register to the appropriate setting based on the expected ADCCLK frequency. The ADC uses the CLKFREQ.FRANGE value and the resolution mode, configured by CTL2.RES bits, to determine how many ADCCLK clock cycles a conversion will take before signaling the End of Conversion (EOC). See Table 18-2 below for more details on how to properly configure the CLKFREQ register and how that impacts the EOC signal generation.

**Table 18-2. CLKFREQ Register Configuration**

| CLKFREQ.FRANGE Values | ADCCLK Frequency Range (MHz) | Conversion Duration (Clock cycles) | | |
|---|---|---|---|---|
| | | 12-bit | 10-bit | 8-bit |
| 0 | >1 to 4 | 1 | 1 | 1 |
| 1 | >4 to 8 | 2 | 2 | 1 |
| 2 | >8 to 16 | 3 | 3 | 2 |
| 3 | >16 to 20 | 4 | 4 | 3 |
| 4 | >20 to 24 | 5 | 4 | 3 |
| 5 | >24 to 32 | 6 | 6 | 4 |
| 6 | >32 to 40 | 8 | 7 | 5 |
| 7 | >40 to 48 | 9 | 8 | 6 |

**Note**

If the CLKFREQ.FRANGE is not selected to the correct frequency than unintended results can occur. Make sure the value the user sets in FRANGE is correct for the input ADCCLK.

### *18.2.6 Common ADC Use Cases*

There are many ADC use-cases from an operating mode and clocking standpoint but the majority of them fit into one of the items below:

- Triggers in RUN or SLEEP mode



Copyright © 2026 Texas Instruments Incorporated

  - If ADC is triggered to start a conversion (software or Event), and the device is in RUN0 or RUN1 or SLEEP0 or SLEEP1 mode(SYSOSC is already running at any frequency), then:
    - Sample clock can be ULPCLK, HFCLK, or SYSOSC in this mode
    - The conversion runs without changing SYSOSC frequency
    - 4MHz, 16MHz, 24MHz, or 32MHz SYSOSC frequencies are allowed
  - If ADC is triggered to start a conversion (software or Event), and the device is in RUN2 or SLEEP2 mode (SYSOSC is disabled, MCLK = LFCLK = 32kHz), then:
    - Sample clock can be ULPCLK or SYSOSC in this mode
    - SYSCTL interprets the ADC CLK REQ as an asynchronous fast clock request, enabling SYSOSC at 32MHz and forcing MCLK or ULPCLK to 32MHz MHz until the ADC de-asserts the request
    - CCONRUN must be cleared in this use case
    - CCONSTOP must be cleared in this use case
- Triggers in STOP mode
  - Sample clock can be ULPCLK or SYSOSC in this mode
  - If ADC is triggered to start a conversion (Event), and the device is in STOP0 mode (SYSOSC runs at any frequency,ULPCLK = 4MHz), then:
    - The conversion runs without changing SYSOSC frequency
    - 4MHz, 16MHz, 24MHz, or 32MHz SYSOSC frequencies are allowed
  - If ADC is triggered to start a conversion (Event), and the device is in STOP1 mode (SYSOSC was gear shifted to 4MHz), then:
    - SYSCTL forces SYSOSC to BASE (consistent with RUN mode because USE4MHZSTOP was set) when receiving ADC CLK REQ, and SYSCTL releases SYSOSC back to 4MHz after the ADC CLK REQ is removed
    - CCONRUN must be cleared
    - CCONSTOP must be cleared
  - If ADC is triggered to start a conversion (Event), and the device is in STOP2 mode (SYSOSC disabled), then:
    - The trigger event propagates through the event fabric at 32kHz, the ADC receives the trigger and assert the ADC CLK REQ (CPCLK REQ) to SYSCTL, and SYSCTL receives the ADC CLK REQ as an asynchronous fast clock request, suspending STOP, enabling SYSOSC at 32MHz, and forcing MCLK or ULPCLK to 32MHz until the ADC de-asserts the ADC CLK REQ
    - CCONRUN must be cleared
    - CCONSTOP must be cleared
- Triggers in STANDBY mode
  - Sample clock can be ULPCLK or SYSOSC in this mode
  - If ADC is triggered to start a conversion (Event), and the device is in STANDBY0 mode (SYSOSC is disabled but ULPCLK is running), then:
    - The trigger event propagates through event fabric at 32kHz, the ADC receives the trigger and asserts the ADC CLK REQ (CPCLK REQ) to SYSCTL, and SYSCTL interprets the ADC CLK REQ as an asynchronous fast clock request, suspending STANDBY, enabling SYSOSC at 32MHz, and forcing MCLK/ULPCLK to 32MHz until the ADC de-asserts the ADC CLK REQ
    - CCONRUN must be cleared
    - CCONSTOP must be cleared
  - If ADC is triggered to start a conversion (Event-TIMG0 or TIMG1), and the device is in STANDBY1 (ULPCLK is gated with STOPCLKSTBY set), then:
    - The TIMG0 or TIMG1 event triggers an asynchronous fast clock request to suspend STANDBY mode, start SYSOSC at 32MHz, and force MCLK or ULPCLK to 32MHzz; there are then 32 SYSOSC cycles for the TIMG0 or TIMG1 event to proceed through the event fabric and for the ADC to capture the timer event and assert the ADC CLK REQ to hold the SYSOSC enabled to run the conversion
    - When the ADC de-asserts the ADC CLK REQ, ULPCLK runs for 32 additional cycles to allow any ADC event (DMA request or IRQ) to propagate, after which SYSCTL resumes STANDBY with STOPCLKSTBY (STANDBY1)
    - CCONRUN must be cleared



Copyright © 2026 Texas Instruments Incorporated

- CCONSTOP must be cleared

### 18.2.7 Power Down Behavior

The ENABLE bit in the PWREN register enables or disables the ADC peripheral. The ADC should be disabled when it is not in use to save power. The PWRDN bit in the CTL0 register selects the ADC power down policy between AUTO and MANUAL. This takes effect when the ADC operates in RUN, SLEEP, and STOP MCU power modes.

PWRDN should be configured based on the max ADC sampling rate required and the operational needs in different MCU power modes. ADC hardware does not force the power down policy to AUTO during operation in STOP mode. It follows the user setting irrespective of the device power modes.

The reset value of PWRDN is 0, which has the default behavior of automatic power down of the ADC peripheral at the end of a conversion and when the next sample signal is not required to be asserted immediately. When the PWRDN bit is set to '1' it selects manual power down behavior. In this setting, the ADC is not powered down at the end of a conversion and remains enabled. This means that the ADC peripheral would only be powered down using the PWREN register.

**Note**

Refer to the device-specific data sheet for specifications on the ADC wakeup and enable time.

### 18.2.8 Sampling Trigger Sources and Sampling Modes

#### Sample Triggers

There are two sampling trigger sources available which can be selected through the TRIGSRC bit in the CTL1 register; one is a software trigger and the other is an event trigger.

When the software trigger is selected as the source, the application software can set the SC bit in the CTL1 register to initiate the sample phase. When the event trigger is selected as the source, a rising edge on the selected event from the event manager will initiate the sample phase. An event is always edge triggered.

#### Sampling Modes

There are two sampling modes available, AUTO and MANUAL, which can be selected through the SAMPMODE bit in the CTL1 register.

##### 18.2.8.1 AUTO Sampling Mode

In AUTO mode, the sample signal is generated synchronous to the sampling clock (SAMPCLK) and can be programmed using an internal sampling timer to determine the duration of the sampling window. The sample timer is 10-bits wide and there are two sample time compare registers (SCOMPx) available to account for various source impedances to measure signals from. One of these two SCOMP registers can be selected using the STIME bit in the MEMCTL register.

There is a 2-3 cycle latency from when the sampling is triggered and when the sampling period starts. This latency can be bypassed by setting ULPCLK as the source for ADCCLK. This synchronization bypass feature is very useful for deterministic samplingand for simultaneous operation of two ADC peripherals.

Figure 18-2 shows the ADC sample and conversion timing diagram when the ADC is configured in AUTO sampling mode.



Copyright © 2026 Texas Instruments Incorporated

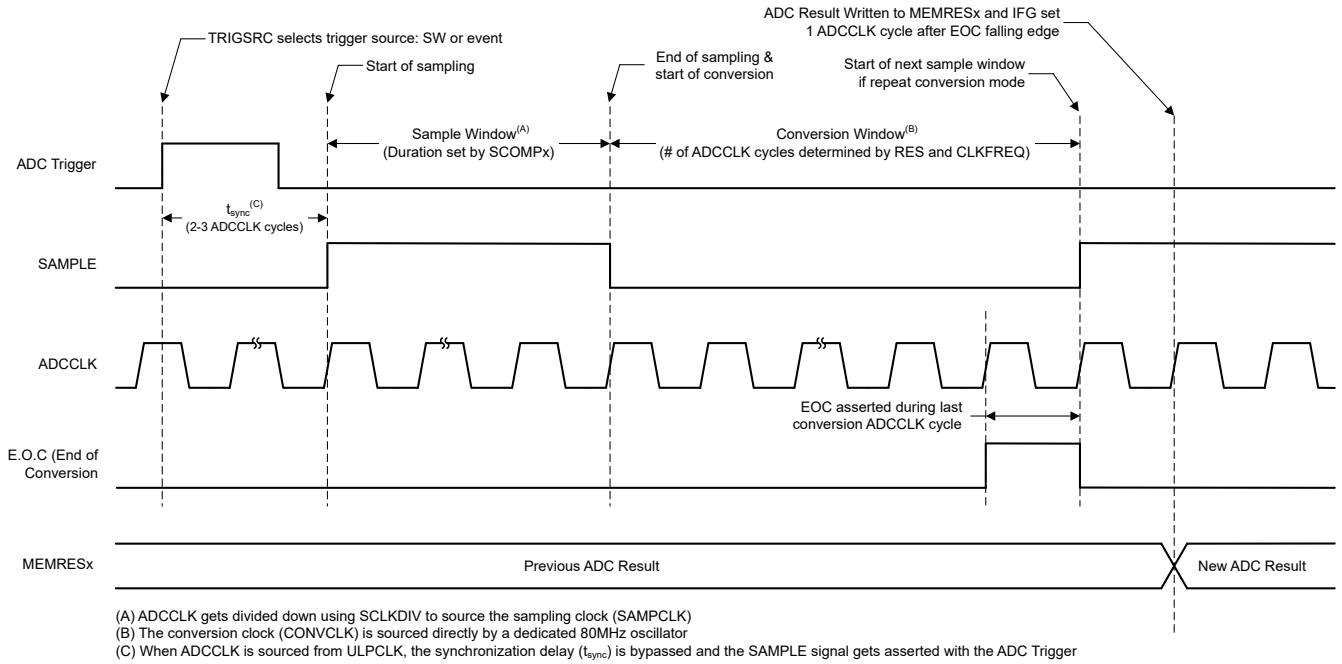



**Figure 18-2. AUTO Sampling Mode ADC Sample and Conversion Timing Diagram**

**Note**

When the reset value of PWRDN is set as ‘0’ which has the default behavior of automatic power down, ADC wakeup/enable time needs to be considered in each sample window. Refer to the device-specific data sheet for specifications on the ADC wakeup/enable time. For example, if the maximum ADC wakeup/enable time is 5uS, it means the duration set by SCOMPx should be > (5uS + Duration for sample window).

#### 18.2.8.2 MANUAL Sampling Mode

In MANUAL mode, the sample signal is generated when the SC bit is set which can be asynchronous to the sampling clock. The duration of the sampling window is controlled by software by holding the SC bit high. Because an event is always edge triggered, manual mode with event trigger is not supported for any of the conversion modes. Software trigger with manual sampling mode is supported only for single channel single conversion mode and it is not supported for any of the other three conversion modes.

There is a 2-3 cycle synchronization latency from when the sample window ends to when the conversion window begins.

Figure 18-3 shows the ADC sample and conversion timing diagram when the ADC is configured in MANUAL sampling mode:



Copyright © 2026 Texas Instruments Incorporated

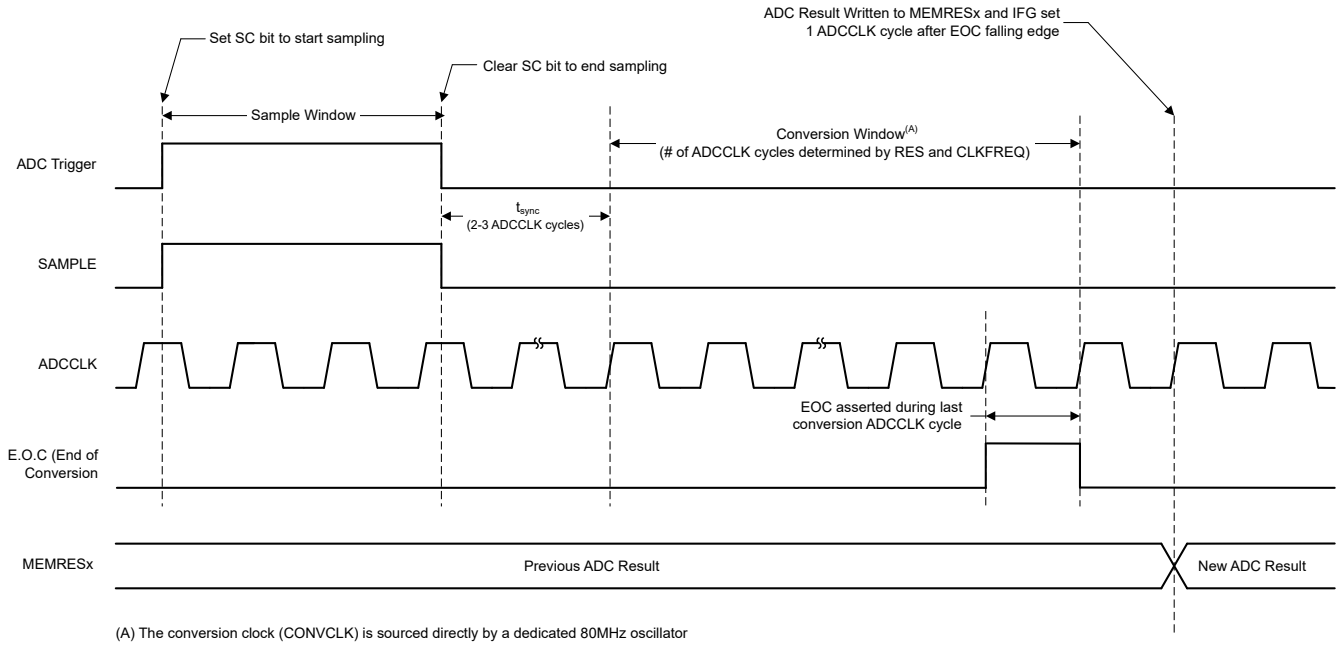



**Figure 18-3. MANUAL Sampling Mode ADC Sample and Conversion Timing Diagram**

**Note**

1. In MANUAL sampling mode the CCONRUN bit in the CLKCFG register must be set to 1.
2. When the reset value of PWRDN is set as '0' which has the default behavior of automatic power down, ADC wakeup/enable time needs to be considered before sample window. It means that after setting ENABLE bit in the PWREN register to enable ADC, then application software should set up a delay to wait the ADC wakeup/enable time before starting sampling. Refer to the device-specific data sheet for specifications on the ADC wakeup/enable time.
3. Once SC bit set by software, the SC bit is automatically cleared by hardware after the sample duration, which is the duration during which the analog signal is sampled. If the software attempts to set the SC bit before the sample duration + 2-3 cycles synchronization latency, TOVIFG flag will not be set.

### *18.2.9 Sampling Period*

The sampling clock source is selected in the SYSCTL module using the SAMPCLK bits in the CLKCFG register. The desired sampling period for ADC operation can be generated using the internal clock divider and/or the sample timer, which applies to AUTO sampling mode. The internal clock divider is configured using the SCLKDIV bits in the CTL0 register and has divide options of 1, 2, 4, 8, 16, 24, 32, and 48.

The duration of the sampling period can be programmed to one of two user-defined values set by the SCOMP0 and SCOMP1 sample timer registers. The value in SCOMPx configures the sampling period by defining the number of sample time clocks to set the sample window to. The default SCOMPx sample timer value translates to 1 cycle wide sample pulse which allows the sampling period to be solely based on the sample clock and SCLKDIV. In general, there are three knobs that can be leveraged to control the sample period: SCOMPx, SCLKDIV, and the source of the sample clock.

When AUTO power down mode is selected using PWRDN=0, the module enable signal to the ADC peripheral is generated one sampling clock cycle after the sample signal is asserted. This should be considered by the user in the sample window calculation in addition to the ADC power time or settling time needs of other analog modules such as the Temperature Sensor, VREF, etc.



Copyright © 2026 Texas Instruments Incorporated

### 18.2.10 Conversion Modes

There are four conversion modes available in the ADC:

1. Single channel single conversion
   - The channel can be selected using MEMCTL
   - The selected channel is sampled and converted only once
   - Multiple conversions are performed when HW averaging is enabled
2. Repeat single channel conversion
   - The channel can be selected using MEMCTL
   - The selected channel is repeatedly sampled and converted until ENC is cleared by software
     - If the TRIG bit is set, a trigger is needed to move to the next conversion
   - Multiple conversions are performed when HW averaging is enabled
3. Sequence of channels conversion
   - Groups of channels can be formed using STARTADD, ENDADD, and MEMCTL registers
   - Each of the channels in the group is sampled and converted only once
   - Multiple conversions are performed on a channel during the sequence when HW averaging is enabled
   - The sequence will complete even if ENC is cleared in the middle of the sequence
4. Repeat sequence of channels conversion
   - Groups of channels can be formed using STARTADD, ENDADD, and MEMCTL registers
   - The group of channels are sampled and converted repeatedly until ENC is cleared by software
     - If the TRIG bit is set, a trigger is needed to move to the next conversion
   - When ENC is cleared the operation stops at the end of the last conversion
   - Multiple conversions are performed on a channel during a sequence when averaging is enabled

The following steps outline the recommended process for configuring the ADC for a desired conversion mode:

1. Use the CONSEQ bits in the CTL1 register to select the desired ADC conversion mode
2. Use the STARTADD bits in the CTL2 register to select which MEMCTLx is used for single conversion or as first MEMCTL for a sequence mode
3. If using a sequence mode, use the ENDADD bits in the CTL2 register to select which MEMCTLx is used for the last conversion of the sequence
4. Assign an ADC input channel to the appropriate MEMCTLx register using the CHANSEL bits
   - For sequence modes, you must assign an ADC input channel for each MEMCTLx that is part of the configured sequence
5. Select HW or SW trigger using the TRIGSRC bit in the CTL1 register
6. Select AUTO or MANUAL sampling mode using the SAMPMODE bit in the CTL1 register
   - If using AUTO mode, program the desired sample timer value in the SCOMPx register and use the STIME bits in the MEMCTLx register to select the appropriate sample timer source (SCOMP0 or SCOMP1)
7. If using repeat single channel or sequence conversion modes, program the TRIG bit in each MEMCTLx register to indicate if a trigger will be needed to step to the next MEMCTL in the sequence
8. Set the ENC bit in the CTL0 register to enable ADC conversions
9. The following table matrix depicts the next step of ADC configuration and usage based on the selected trigger and sampling modes:

**Table 18-3. Trigger and Sample Mode ADC Usage Matrix**

| | Trigger Mode | |
|---|---|---|
| | **SW Trigger** | **Event Trigger** |



Copyright © 2026 Texas Instruments Incorporated

**Table 18-3. Trigger and Sample Mode ADC Usage Matrix (continued)**

| | | | |
|---|---|---|---|
| **Sampling Mode** | **AUTO** | • Set SC bit to start the sample phase (duration determined by sample timer)<br>• Conversion starts once sample phase is over<br>• In single channel single conversion, ENC is cleared when conversion is over<br>• SC bit is automatically cleared once the trigger is captured<br>• For repeat and sequence modes, if TRIG is set in MEMCTL, the SC bit needs to be set for the next conversion to proceed | • HW trigger starts the sample phase (duration determined by sample timer)<br>• Conversion starts once sample phase is over<br>• In single channel single conversion, ENC is cleared when conversion is over<br>• For repeat and sequence modes, ADC waits for HW trigger or automatically starts the next conversion based on TRIG setting |
| | **MANUAL** | • Set SC bit to start the sample phase (SC bit is not automatically reset)<br>• Clear the SC bit to end the sample phase and start the conversion<br>• In single channel single conversion, ENC bit is cleared when conversion is over<br>• Repeated/sequential conversion modes and HW averaging are NOT supported in this configuration | ADC operation is NOT supported in this configuration |

10. The ADC results are stored in the MEMRES register of the associated MEMCTL (for example, the MEMCTL0 result is stored in MEMRES0).
    - For repeat conversion modes, the result in MEMRES is updated after every associated MEMCTL conversion
11. For repeated conversion modes, clear the ENC bit to stop ADC operation

### *18.2.11 Data Format*

The ADC supports two data formats – unsigned binary and 2's complement signed binary. Unsigned binary results are stored right-justified in the MEMRES register or FIFO. Signed binary results are stored left justified in the MEMRES register or FIFO.

**Table 18-4. ADC Data Formats**

| Data Format | Resolution | Result Range (decimal) | Result Range (hex) |
|---|---|---|---|
| Unsigned | 8-bit | 0 to 255 | 0000h to 00FFh |
| | 10-bit | 0 to 1023 | 0000h to 03FFh |
| | 12-bit | 0 to 4095 | 0000h 0FFFh |
| Signed | 8-bit | -128 to 127 | 8000h to 7F00h |
| | 10-bit | -512 to 511 | 8000h to 7FC0h |
| | 12-bit | -2048 to 2047 | 8000h to 7FF0h |

### *18.2.12 Advanced Features*

The following sections describe the additional features and benefits provided with the ADC peripheral and how to leverage them in an application.

#### 18.2.12.1 Simultaneous Sampling

Some applications, such as current and voltage sensing, require measurement from multiple analog signals at the exact same time. In these circumstances, using multiple ADCs on a single MCU to perform simultaneous sampling is a requirement. Any device with multiple ADC peripherals in the MSPM0xx platform supports simultaneous sampling.



Copyright © 2026 Texas Instruments Incorporated

For this use-case, the ULPCLK should be used as the source for ADCCLK because it is also the clock for all of the peripherals in PD0. This means that it is already synchronized with the bus clock and therefore can ensure deterministic timing for the start of sampling. In addition to using the ULPCLK as the sample clock source for both ADC peripherals, the user is expected to select the same sample trigger and program the clock prescaler and/or SCOMP with the same values on both ADCs.

#### 18.2.12.2 Window Comparator

There is one window comparator unit available in the ADC which can be used to check if the input signal is within predefined threshold values set by software. The ADC result that goes into MEMRES or FIFO is what gets checked against the threshold values of the window comparator.

Based on the comparison it can generate 3 interrupt conditions:

1. LOWIFG – Conversion result is below the Low threshold (WCLOW)
2. HIGHIFG – Conversion result is above the High threshold (WCHIGH)
3. INIFG – Conversion result is in between or equal to the Low and High thresholds

The window comparator low and high threshold values are global for all channels and the window comparison feature can be enabled for each channel as needed using the WINCOMP bit in the MEMCTL register.

When the ADC result data format (CTL2.DF) or resolution (CTL2.RES) configuration is changed, the window comparator threshold values are not reset by hardware and are retained as is. The software application is expected to reconfigure the threshold values as appropriate after changing the data format and/or resolution configuration.

#### 18.2.12.3 DMA and FIFO Operation

The ADC has a dedicated interface for communicating to and from the DMA. This interface is useful to offload work from the CPU by using the DMA to store ADC results to memory automatically. Figure 18-4 shows the signals that make up this interface:

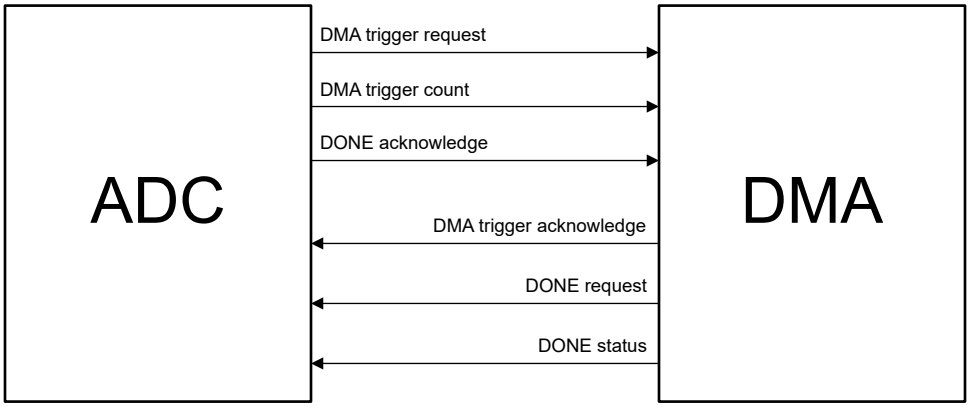



**Figure 18-4. Internal ADC-DMA Interface**

**Note**

The “DMA trigger count” signal indicates the number of samples that can be transferred by the DMA upon one trigger request. The “DONE status” signal is used by the ADC to generate the DMA DONE interrupt and it indicates if the DMA data transfer of programmed block size is completed.

The DMAEN bit in the CTL2 register is used to enable the DMA for ADC data transfer. The DMAEN bit is cleared by ADC hardware when the DMA “DONE status” signal is asserted. Software is expected to re-enable the DMA using DMAEN to arm the ADC to generate the next DMA trigger.

The ADC also incorporates an optional First-In-First-Out buffer to provide a way for ADC results to be stored for future use, such as transferring to memory by the DMA. Either the CPU or the DMA can be used to move data



Copyright © 2026 Texas Instruments Incorporated

from the ADC regardless of whether the FIFO is enabled or disabled. The memory result flags in the RIS register of the third event publisher serve as the FIFO threshold and can be unmasked to generate the DMA trigger.

The following sections explain the details of using the ADC+DMA/CPU in various conversion modes and with the FIFO enabled or disabled

### ADC-DMA/CPU Operation in Non-FIFO Mode (FIFOEN=0)

ADC-DMA/CPU Operation in **Non-FIFO Mode (FIFOEN=0)**

- Single Conversion and Repeat Single Conversion
  - Configure STARTADD bits to select the desired MEMCTLx register
    - MEMCTLx **is correlated to** MEMRESx
    - MEMRESx **is correlated to** MEMRESIFGx
  - Configure MEMCTL CHANSEL bits to select the desired ADC channel
  - Conversion data is available in MEMRESx
  - MEMRESIFGx can be set to generate a CPU interrupt or the DMA trigger
  - SAMPCNT **must be programmed to 1** by SW for DMA operation
  - The conversion overflow flag OVIFG is set when the ADC updates MEMRESx before the previous sample is read by the CPU or DMA
  - The conversion underflow flag UVIFG is set when the CPU or DMA reads the MEMRESx register before the next conversion result is available
- Sequence Conversion and Repeat Sequence Conversion
  - Configure STARTADD bits to select the first MEMCTL in the sequence
  - Configure ENDADD bits to select the last MEMCTL in the sequence
    - MEMCTLx **is correlated to** MEMRESx
    - MEMRESx **is correlated to** MEMRESIFGx
  - Configure each MEMCTLx CHANSEL bits to select the desired ADC channels
  - Conversion data is available in MEMRESx
  - MEMRESIFGx can be set to generate a CPU interrupt or the DMA trigger
  - SAMPCNT must be programmed by SW to a **suitable value based on number of ADC samples** for DMA operation. Each DMA transfer will be a single ADC result in non-FIFO mode.
  - The conversion overflow flag OVIFG is set when the ADC updates MEMRESx before the previous sample is read by the CPU or DMA
  - The conversion underflow flag UVIFG is set when the CPU or DMA reads the MEMRESx register before the next conversion result is available

**Note**

For DMA-based operation in ADC sequence conversion mode, the MEMCTL start address should be smaller than the end address as the DMA source address does not roll back. Repeat sequence conversion mode cannot fill a buffer greater than the number of channels because the DMA does not have a circular mode. Once a sequence is completed, the DMA would need to be manually reconfigured to set the new destination address in order to fill a buffer larger than the number of channels in the sequence.

### ADC-DMA/CPU Operation in FIFO Mode (FIFOEN=1)

- Single Conversion and Repeat Single Conversion
  - Configure STARTADD bits to select the desired MEMCTLx register
    - MEMCTLx **is NOT correlated to** MEMRESx
    - MEMRESx **is correlated to** MEMRESIFGx
  - Configure MEMCTL CHANSEL bits to select the desired ADC channel
  - Conversion data is loaded sequentially into MEMRES0,1,2,….N (organized as a FIFO) where N is the number of MEMRES channels in the device.



Copyright © 2026 Texas Instruments Incorporated

  - The CPU or DMA must read ADC samples from the dedicated FIFODAT register and not from MEMRES registers directly
    - Data in the FIFO is always compacted with two samples and provided as 32-bit data upon a FIFODATA read by CPU or DMA
  - MEMRESIFGx should be used as a threshold condition to generate a CPU interrupt or DMA trigger
    - Any MEMRESIFGx can be selected but improper selections can lead to an UVIFG or OVIFG getting set based on DMA speeds and ADC sampling speeds.
  - SAMPCNT must be programmed by SW to a **suitable value based on threshold setting** for DMA operation
  - The conversion overflow flag OVIFG is set when the ADC updates MEMRESx before the previous sample is read by the CPU or DMA
  - The conversion underflow flag UVIFG is set when the CPU or DMA reads the FIFODAT register before the conversion result is available in the MEMRESx registers.

**Note**

Single conversion mode with FIFO enabled is not recommended for CPU or DMA based operation. It will lead to underflow condition and unwanted 16-bit data will have to be discarded in software.

- Sequence Conversion and Repeat Sequence Conversion
  - Configure STARTADD bits to select the first MEMCTL in the sequence
  - Configure ENDADD bits to select the last MEMCTL in the sequence
    - MEMCTLx **is NOT correlated to** MEMRESx
    - MEMRESx **is correlated to** MEMRESIFGx
  - Configure each MEMCTLx CHANSEL bits to select the desired ADC channels
  - Conversion data is loaded sequentially into MEMRES0,1,2,….N (organized as a FIFO) where N is the number of MEMRES channels in the device.
  - The CPU or DMA must read ADC samples from the dedicated FIFODAT register and not from MEMRES registers directly
    - Data in the FIFO is always compacted with two samples and provided as 32-bit data upon a FIFODATA read by CPU or DMA
  - MEMRESIFGx can be used as a threshold condition to generate a CPU interrupt or DMA trigger
    - Any MEMRESIFGx can be selected but improper selections can lead to an UVIFG or OVIFG getting set based on DMA speeds and ADC sampling speeds.
  - SAMPCNT must be programmed by SW to a **suitable value based on threshold setting** for DMA operation
  - The conversion overflow flag OVIFG is set when the ADC updates MEMRESx before the previous sample is read by the CPU or DMA
  - The conversion underflow flag UVIFG is set when the CPU or DMA reads the FIFODAT register before the conversion result is available in the MEMRESx registers



Copyright © 2026 Texas Instruments Incorporated

**Note**

- The data in FIFODATA register will shift to the next result upon a read, if there is no new result then the populated value will be 0.
- All available MEMRES are used sequentially in FIFO mode.
- To ensure synchronized reading of bytes from the 32-bit FIFO, which stores 16-bit samples, specific DMA triggers can be used. In particular, selecting MEMRES1, MEMRES3, MEMRES5, MEMRES7, MEMRES9 and MEMRES11 will synchronize the reading of bytes from the FIFO with the corresponding MEMRESx bytes.
  - Set the DMA samples to be transferred upon a trigger to be 1 and set the DMA to repeat single transfer mode. Each trigger will then send 2 ADC samples be synchronized based on when the 2 samples are ready. The synchronization is beneficial as the DMA is commonly faster than the ADC sample and conversion time.
- If the ADC is disabled during either the repeat sequence mode or normal repeat mode, an additional conversion may occur before the ADC completely stops.
- If stopping conversions in the middle of a sequence, it is recommended to flush the FIFO by toggling the FIFOEN bit before restarting the conversions. By toggling the FIFO it will return to the configured start MEMRES.

**Table 18-5. ADC-DMA/CPU Operation Summary Matrix**

| Conversion Mode | FIFO Disabled (FIFOEN=0)<br>Samples not compacted<br>Read from MEMRESx registers directly | | FIFO Enabled (FIFOEN=1)<br>Samples always compacted<br>Read from FIFODAT register only | |
|---|---|---|---|---|
| | CPU Read/Write | DMA Read/Write | CPU Read/Write | DMA Read/Write |
| **Single** | Supported | Supported<br>SAMPCNT=1<br>Sample in 16 bits | Not recommended<br>Underflow flag will set<br>Unwanted 16-bits should be ignored | Not recommended<br>Underflow flag will set<br>Unwanted 16-bits should be ignored |
| **Repeat Single** | Supported | Supported<br>SAMPCNT=1<br>Sample in 16 bits | Supported<br>MEMRESIFG=CPU interrupt<br>FIFODAT read in 32-bits | Supported<br>MEMRESIFG=DMA trigger<br>SAMPCNT=Samples in 32-bits |
| **Sequence** | Supported | Supported<br>SAMPCNT=Sample in 16 bits<br>STARTADD<ENDADD | Supported<br>MEMRESIFG=CPU interrupt<br>FIFODAT read in 32-bits | Supported<br>MEMRESIFG=DMA trigger<br>SAMPCNT=Samples in 32-bits |
| **Repeat Sequence** | Supported | Not Supported | Supported<br>MEMRESIFG=CPU interrupt<br>FIFODAT read in 32-bits | Supported<br>MEMRESIFG=DMA trigger<br>SAMPCNT=Samples in 32-bits |

When using the DMA to read from the ADC, take into consideration the timing for the DMA and ADC to prevent underflow/overflow conditions. The DMA will typically run faster than the ADC. If the user has setup the DMA to transfer data for other peripherals, take into consideration the DMA being delayed before transfering ADC results, or properly assign DMA priorites.

#### 18.2.12.4 Analog Peripheral Interconnection

The MSPM0 platform of MCUs provides a rich set of high-performance analog peripherals which can interact with each other to perform various analog signal chain functions. The items below describe how the ADC interacts with the other on-board analog peripherals:

**ADC with Internal Reference Module (VREF)**

The ADC has a dedicated enable request and ready interface with the internal voltage reference module. VREF enable is asserted upon sample trigger while internal reference buffer is selected for ADC operation. The ready response from VREF is captured in the ADC status register (REFBUFRDY).

The ready response from VREF **does not** gate the sample phase in the ADC. The ADC sample window is started upon the sample trigger and the settling time of the internal reference buffer needs to be considered in



Copyright © 2026 Texas Instruments Incorporated

the sample period as appropriate. Software can enable the VREF module using the software enable bit so that it is already settled by the time the ADC starts sampling the input channel. In this case, the sampling time can be smaller and does not have to account for the enable time of VREF.

#### ADC with OPA

The ADC provides a burnout current source (BCS) enable signal to an OPA peripheral when it is selected as the ADC input channel and the BCSEN bit is set in the MEMCTL register. This signal allows the ADC peripheral to work in unison with the OPA peripheral to monitor sensor health and detect faults. The BCS enable signal remains high for all conversions during hardware averaging operation. The ADC receives a chop request signal and provides a chop state control signal to the OPA peripheral when ADC-assisted OPA chopping is activated and the hardware averaging feature is enabled. The chop state control signal toggles at the end of each conversion during hardware averaging operation. The number of samples averaged should be programmed to an even value when ADC-assisted OPA chopping is activated.

#### ADC with Temperature Sensor

The temperature sensor module enable signal is generated by the ADC when the temp sense channel is selected. The settling time of the temperature sensor should be accounted for in the sample period as there is no ready response from the temperature sensor.

### *18.2.13 Status Register*

The ADC status register, STATUS, contains two bits – REFBUFRDY and BUSY.

- REFBUFRDY is set when the ADC receives the ready signal from the internal reference buffer (VREF/ REFBUF) after asserting the enable request
- BUSY equaling ‘1’ indicates that the ADC is busy performing a sample or convert operation
  - For **single channel single conversion**, it signals that a trigger has been received and sample or conversion is ongoing. BUSY will be cleared when the conversion completes.
  - For **repeat single conversion**, it signals that repeat single operation has begun and has not ended. BUSY will be cleared when ENC is written ‘0’ and the last conversion completes. When reenabling the ENC bit, wait until the BUSY status is 0 before setting ENC to 1.
  - For **sequence of channels conversion**, it signals that the sequence of channels conversion has started. BUSY will be cleared at the end of the sequence.
  - For **repeat sequence of channels conversion**, it signals the repeat sequence is ongoing. BUSY will be cleared when ENC is written ‘0’ and the last conversion completes. . When reenabling the ENC bit, wait until the BUSY status is 0 before setting ENC to 1.

**Note**

When the ADC is in auto power down mode, the ADC requires a wakeup time before sampling begins. Provide a software delay for the ADC wakeup time, see device datasheet,, before checking the BUSY bit; otherwise a 0 will be read for the BUSY bit.

### *18.2.14 ADC Events*

The ADC peripheral contains three event publishers and one event subscriber.

One event publisher (CPU_INT) manages ADC interrupt requests (IRQs) to the CPU subsystem through a static event route. The second event publisher (GEN_EVENT) can be used to publish ADC events to a subscriber through a generic event route channel. The third event publisher (DMA_TRIG) can be used as an ADC-to- DMA trigger to send ADC events directly to the DMA through a DMA event route.

The event subscriber (FSUB_0) can be used to subscribe to events which are published to the event fabric through a generic event route channel.

The ADC events are summarized in Table 18-6.



Copyright © 2026 Texas Instruments Incorporated

**Table 18-6. ADC Events**

| Event | Type | Source | Destination | Route | Configuration | Functionality |
|---|---|---|---|---|---|---|
| CPU interrupt event | Publisher | ADC | CPU Subsystem | Static route | CPU_INT registers | Fixed interrupt route from ADC to CPU |
| Generic publisher event | Publisher | ADC | Generic event channel | Generic route (FPUB_0) | GEN_EVENT registers, FPUB_0 register | Trigger generic event channel from ADC |
| DMA trigger event | Publisher | ADC | DMA | DMA route | DMA_TRIG registers | Fixed trigger route from ADC to DMA |
| Generic subscriber event | Subscriber | Other peripherals | ADC | Generic route(FSUB_0) | FSUB_0 | ADC subscription to generic event channel |

#### 18.2.14.1 CPU Interrupt Event Publisher (CPU_INT)

The ADC peripheral provides many interrupt sources which can be configured to source a CPU interrupt event. In order of decreasing interrupt priority, the CPU interrupt events from the ADC are given in Table 18-7.

**Table 18-7. ADC CPU Interrupt Event Conditions (CPU_INT)**

| Index (IIDX) | Name | Description |
|---|---|---|
| 0x0 | NO_INTR | No bit set (IIDX.STAT = 0) means there is no pending interrupt request |
| 0x1 | OVIFG | Conversion overflow interrupt flag is set when the ADC updates MEMRESx before the previous sample is read by the CPU or DMA |
| 0x2 | TOVIFG | Sequence conversion time overflow interrupt flag is set when the ADC receives a new sampling trigger while the previous sample+conversion is still in progress |
| 0x3 | HIGHIFG | High threshold compare interrupt flag is set when the MEMRESx result register is higher than the WCHIGH threshold of the window comparator |
| 0x4 | LOWIFG | Low threshold compare interrupt flag is set when the MEMRESx result register is lower than the WCLOW threshold of the window comparator |
| 0x5 | INIFG | In-range comparator interrupt flag is set when the MEMRESx result register is within the range of WCLOW and WCHIGH of the window comparator |
| 0x6 | DMADONE | DMA done interrupt flag is set when the DMA data transfer of programmed block size is completed |
| 0x7 | UVIFG | Conversion underflow interrupt flag, the UVIFG flag is set when the CPU or DMA reads the MEMRESx register before the next conversion result is available |
| 0x9 up to 0x20 | MEMRESIFG[0 up to 24](1) | Memory register interrupt flag is set when MEMRESx is loaded with a new conversion result |

(1) Refer to the device-specific data sheet to see how many conversion-result storage registers (MEMRES) your device supports.

The CPU interrupt event configuration is managed with the CPU_INT event management registers. Interrupt (RIS) flags are cleared upon software reading the IIDX register or writing to the respective ICLR register bits. See Section 7.2.5 for guidance on configuring the Event registers for CPU interrupts.

#### 18.2.14.2 Generic Event Publisher (GEN_EVENT)

The ADC peripheral provides 4 interrupt sources, one of which can be configured to publish an event to a generic event route channel. Table 18-8 lists these interrupt sources.

**Table 18-8. ADC Generic Event Publisher Conditions (GEN_EVENT)**

| Index | Name | Description |
|---|---|---|
| 0x0 | NO_INTR | No bit set means there is no pending interrupt request |



Copyright © 2026 Texas Instruments Incorporated

**Table 18-8. ADC Generic Event Publisher Conditions (GEN_EVENT) (continued)**

| Index | Name | Description |
|---|---|---|
| 0x3 | HIGHIFG | High threshold compare interrupt flag is set when the MEMRESx result register is higher than the WCHIGH threshold of the window comparator |
| 0x4 | LOWIFG | Low threshold compare interrupt flag is set when the MEMRESx result register is lower than the WCLOW threshold of the window comparator |
| 0x5 | INIFG | In-range comparator interrupt flag is set when the MEMRESx result register is within the range of WCLOW and WCHIGH of the window comparator |
| 0x9 | MEMRESIFG0 | Memory register interrupt flag is set when MEMRES0 is loaded with a new conversion result |

The generic event publisher configuration is managed with the GEN_EVENT event management registers. Interrupt (RIS) flags are cleared based on acknowledgment (ACK) signal from the subscriber module received over the event fabric. See Section 7.2.5 for guidance on configuring the Event registers for generic event publishers.

The generic event channel which GEN_EVENT is to publish to must be selected by writing the target generic channel ID to the **FPUB_0** register in the ADC. See Section 7.1.3.3 for guidance on configuring generic event routes.

If this publisher is not used in an application, the FPUB_0 register can be left in a disconnected state (set equal to zero) and no events should be unmasked through the MIS register in the ADC GEN_EVENT register set.

#### 18.2.14.3 DMA Trigger Event Publisher (DMA_TRIG)

The ADC module provides many interrupt sources which can be configured to source the DMA trigger. In order of decreasing interrupt priority, the DMA trigger events from the ADC are given in Table 18-9. When the DMA channel is needed by the ADC, the DMA trigger should be unmasked in the IMASK register of DMA_TRIG and the DMA should be configured as needed to support the ADC operation.

**Table 18-9. ADC DMA Trigger Event Conditions (DMA_TRIG)**

| Index (IIDX) | Name | Description |
|---|---|---|
| 0x0 | NO_INTR | No bit set (IIDX.STAT = 0) means there is no pending interrupt request |
| 0x9 up to 0x20 | MEMRESIFG[0 up to 24](1) | Memory register interrupt flag is set when MEMRESx is loaded with a new conversion result |

(1) Check the device-specific data sheet to see how many conversion-result storage registers (MEMRES) your device supports.

The DMA trigger event configuration is managed with the DMA_TRIG event management registers. The interrupt (RIS) flags are cleared based on ACK from DMA. See Section 7.2.5 for guidance on configuring the Event registers for DMA triggers.

#### 18.2.14.4 Generic Event Subscriber (FSUB_0)

The ADC peripheral supports receiving events routed through a generic channel from other peripherals. Refer to Section 7.1.3.3 and Section 7.2.3 for how generic event route works.

Once the channel to be used is determined, and both the publisher and subscriber ports for the peripherals being connected are known, use the steps below to establish the event connection. In this example, a GPIO triggered ADC application will be configured, using GPIO Port A to publish an event to generic channel 1, with ADC0 subscribing to generic channel 1 as a start-of-conversion trigger.

1. Configure the GEN_EVENT registers of GPIO Port A to set the event request based on the appropriate event (for example, a DIN rise event).
2. Store 0x1 into the FPUB_0 register of GPIO Port A to publish the GPIO event selected by the GEN_EVENT registers to generic route channel 1. Channel 1 must not be in use by another peripheral.
3. Store 0x1 the FSUB_0 register of ADC0 so that ADC0 is listening for events published by the timer to channel 1.



Copyright © 2026 Texas Instruments Incorporated

4. Configure ADC0 to trigger from the subscriber port according to the configuration instructions in Section 18.2.8
5. Configure and enable the appropriate GPIO pin to monitor input voltage events



Copyright © 2026 Texas Instruments Incorporated

## 18.3 ADC12 Registers

Table 18-10 lists the memory-mapped registers for the ADC12 registers. All register offset addresses not listed in Table 18-10 should be considered as reserved locations and the register contents should not be modified.

**Table 18-10. ADC12 Registers**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 400h | FSUB_0 | Subscriber Configuration Register. | Section 18.3.1 |
| 444h | FPUB_1 | Publisher Configuration Register. | Section 18.3.2 |
| 800h | PWREN | Power enable | Section 18.3.3 |
| 804h | RSTCTL | Reset Control | Section 18.3.4 |
| 808h | CLKCFG | ADC clock configuration Register | Section 18.3.5 |
| 814h | STAT | Status Register | Section 18.3.6 |
| 1020h | IIDX | Interrupt index | Section 18.3.7 |
| 1028h | IMASK | Interrupt mask | Section 18.3.8 |
| 1030h | RIS | Raw interrupt status | Section 18.3.9 |
| 1038h | MIS | Masked interrupt status | Section 18.3.10 |
| 1040h | ISET | Interrupt set | Section 18.3.11 |
| 1048h | ICLR | Interrupt clear | Section 18.3.12 |
| 1050h | IIDX | Interrupt index | Section 18.3.13 |
| 1058h | IMASK | Interrupt mask | Section 18.3.14 |
| 1060h | RIS | Raw interrupt status | Section 18.3.15 |
| 1068h | MIS | Masked interrupt status | Section 18.3.16 |
| 1070h | ISET | Interrupt set | Section 18.3.17 |
| 1078h | ICLR | Interrupt clear | Section 18.3.18 |
| 1080h | IIDX | Interrupt index | Section 18.3.19 |
| 1088h | IMASK | Interrupt mask extension | Section 18.3.20 |
| 1090h | RIS | Raw interrupt status extension | Section 18.3.21 |
| 1098h | MIS | Masked interrupt status extension | Section 18.3.22 |
| 10A0h | ISET | Interrupt set extension | Section 18.3.23 |
| 10A8h | ICLR | Interrupt clear extension | Section 18.3.24 |
| 10E0h | EVT_MODE | Event Mode | Section 18.3.25 |
| 10FCh | DESC | Module Description | Section 18.3.26 |
| 1100h | CTL0 | Control Register 0 | Section 18.3.27 |
| 1104h | CTL1 | Control Register 1 | Section 18.3.28 |
| 1108h | CTL2 | Control Register 2 | Section 18.3.29 |
| 1110h | CLKFREQ | Sample Clock Frequency Range Register | Section 18.3.30 |
| 1114h | SCOMP0 | Sample Time Compare 0 Register | Section 18.3.31 |
| 1118h | SCOMP1 | Sample Time Compare 1 Register | Section 18.3.32 |
| 1148h | WCLOW | Window Comparator Low Threshold Register | Section 18.3.33 |
| 1150h | WCHIGH | Window Comparator High Threshold Register | Section 18.3.34 |
| 1180h + formula | MEMCTL_y | Conversion Memory Control Register | Section 18.3.35 |
| 1340h | STATUS | Status Register | Section 18.3.36 |

Complex bit access types are encoded to fit into small table cells. Table 18-11 shows the codes that are used for access types in this section.



Copyright © 2026 Texas Instruments Incorporated

**Table 18-11. ADC12 Access Type Codes**

| Access Type | Code | Description |
|---|---|---|
| Read Type | | |
| R | R | Read |
| RH | R<br>H | Read<br>Set or cleared by hardware |
| Write Type | | |
| W | W | Write |
| WK | W<br>K | Write<br>Write protected by a key |
| Reset or Default Value | | |
| *-n* | | Value after reset or the default value |
| Register Array Variables | | |
| i,j,k,l,m,n | | When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula. |
| y | | When this variable is used in a register name, an offset, or an address it refers to the value of a register array. |



Copyright © 2026 Texas Instruments Incorporated

### 18.3.1 FSUB_0 Register (Offset = 400h) [Reset = 00000000h]

FSUB_0 is shown in Figure 18-5 and described in Table 18-12.

Return to the Summary Table.

Subscriber port

**Figure 18-5. FSUB_0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | CHANID | | | |
| R-0h | | | | | | | | | | | | R/W-0h | | | |

**Table 18-12. FSUB_0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3-0 | CHANID | R/W | 0h | 0 = disconnected.<br>1-15 = connected to channelID = CHANID.<br>0h = A value of 0 specifies that the event is not connected<br>Fh = Consult your device datasheet as the actual allowed maximum may be less than 255. |



Copyright © 2026 Texas Instruments Incorporated

### 18.3.2 FPUB_1 Register (Offset = 444h) [Reset = 00000000h]

FPUB_1 is shown in Figure 18-6 and described in Table 18-13.

Return to the Summary Table.

Publisher port

**Figure 18-6. FPUB_1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | CHANID | | | |
| R-0h | | | | | | | | | | | | R/W-0h | | | |

**Table 18-13. FPUB_1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3-0 | CHANID | R/W | 0h | 0 = disconnected.<br>1-15 = connected to channelID = CHANID.<br>0h = A value of 0 specifies that the event is not connected<br>Fh = Consult your device datasheet as the actual allowed maximum may be less than 255. |



Copyright © 2026 Texas Instruments Incorporated

### 18.3.3 PWREN Register (Offset = 800h) [Reset = 00000000h]

PWREN is shown in Figure 18-7 and described in Table 18-14.

Return to the Summary Table.

Register to control the power state

**Figure 18-7. PWREN Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | ENABLE |
| R-0h | | | | | | | R/WK-0h |

**Table 18-14. PWREN Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | KEY to allow Power State Change<br>26h = KEY to allow write access to this register |
| 23-1 | RESERVED | R | 0h | |
| 0 | ENABLE | R/WK | 0h | Enable the power<br>KEY must be set to 26h to write to this bit.<br>0h = Disable Power<br>1h = Enable Power |



Copyright © 2026 Texas Instruments Incorporated

### 18.3.4 RSTCTL Register (Offset = 804h) [Reset = 00000000h]

RSTCTL is shown in Figure 18-8 and described in Table 18-15.

Return to the Summary Table.

Register to control reset assertion and de-assertion

**Figure 18-8. RSTCTL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | RESETSTKYCLR | RESETASSERT |
| R-0h | | | | | | WK-0h | WK-0h |

**Table 18-15. RSTCTL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | Unlock key<br>B1h = KEY to allow write access to this register |
| 23-2 | RESERVED | R | 0h | |
| 1 | RESETSTKYCLR | WK | 0h | Clear the RESETSTKY bit in the STAT register<br>KEY must be set to B1h to write to this bit.<br>0h = Writing 0 has no effect<br>1h = Clear reset sticky bit |
| 0 | RESETASSERT | WK | 0h | Assert reset to the peripheral<br>KEY must be set to B1h to write to this bit.<br>0h = Writing 0 has no effect<br>1h = Assert reset |



Copyright © 2026 Texas Instruments Incorporated

### 18.3.5 CLKCFG Register (Offset = 808h) [Reset = 00000000h]

CLKCFG is shown in Figure 18-9 and described in Table 18-16.

Return to the Summary Table.

ADC clock configuration

**Figure 18-9. CLKCFG Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | CCONSTOP | CCONRUN | RESERVED | | SAMPCLK | |
| R-0h | | R/W-0h | R/W-0h | R-0h | | R/W-0h | |

**Table 18-16. CLKCFG Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | Unlock key<br>A9h = KEY to allow write access to this register |
| 23-6 | RESERVED | R | 0h | |
| 5 | CCONSTOP | R/W | 0h | CCONSTOP: Forces SYSOSC to run at base frequency when device is in STOP mode which can be used as ADC sample or conversion clock source.<br>0h = ADC conversion clock source is not kept continuously on during STOP mode.<br>1h = ADC conversion clock source kept continuously on during STOP mode. |
| 4 | CCONRUN | R/W | 0h | CCONRUN: Forces SYSOSC to run at base frequency when device is in RUN mode which can be used as ADC sample or conversion clock source.<br>0h = ADC conversion clock source is not kept continuously on during RUN mode.<br>1h = ADC conversion clock source kept continuously on during RUN mode. |
| 3-2 | RESERVED | R | 0h | |
| 1-0 | SAMPCLK | R/W | 0h | ADC sample clock source selection.<br>0h = ULPCLK is the source of ADC sample clock.<br>1h = SYSOSC is the source of ADC sample clock.<br>2h = HFCLK clock is the source of ADC sample clock.<br>Note : HFCLK may not be available on all the devices. |



Copyright © 2026 Texas Instruments Incorporated

### 18.3.6 STAT Register (Offset = 814h) [Reset = 00000000h]

STAT is shown in Figure 18-10 and described in Table 18-17.

Return to the Summary Table.

peripheral enable and reset status

**Figure 18-10. STAT Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | RESETSTKY |
| R-0h | | | | | | | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |

**Table 18-17. STAT Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-17 | RESERVED | R | 0h | |
| 16 | RESETSTKY | R | 0h | This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register<br>0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register<br>1h = The peripheral was reset since the last bit clear |
| 15-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 18.3.7 IIDX Register (Offset = 1020h) [Reset = 00000000h]

IIDX is shown in Figure 18-11 and described in Table 18-18.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. 0x0 means no event pending. Interrupt 1 is the highest priority, 2 next highest, 4, 8, … $2^{31}$ is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred.
On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in the RIS and MIS are cleared as well. After a read from the CPU (not from the debug interface), the register must be updated with the next highest priority interrupt, if none are pending, then it should display 0x0.

**Figure 18-11. IIDX Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | STAT | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | | | |

**Table 18-18. IIDX Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-10 | RESERVED | R | 0h | |
| 9-0 | STAT | R | 0h | Interrupt index status<br>00h = No bit is set means there is no pending interrupt request<br>01h = MEMRESx overflow interrupt<br>02h = Sequence Conversion time overflow interrupt<br>03h = High threshold compare interrupt<br>04h = Low threshold compare interrupt<br>05h = Primary Sequence In range comparator interrupt<br>6h = DMA done interrupt, generated on DMA transfer completion,<br>07h = MEMRESx underflow interrupt<br>9h = MEMRES0 data loaded interrupt<br>Ah = MEMRES1 data loaded interrupt<br>Bh = MEMRES2 data loaded interrupt<br>Ch = MEMRES3 data loaded interrupt<br>Dh = MEMRES4 data loaded interrupt<br>Eh = MEMRES5 data loaded interrupt<br>Fh = MEMRES6 data loaded interrupt<br>10h = MEMRES7 data loaded interrupt<br>11h = MEMRES8 data loaded interrupt<br>12h = MEMRES9 data loaded interrupt<br>13h = MEMRES10 data loaded interrupt<br>14h = MEMRES11 data loaded interrupt<br>15h = MEMRES12 data loaded interrupt<br>16h = MEMRES13 data loaded interrupt<br>17h = MEMRES14 data loaded interrupt<br>18h = MEMRES15 data loaded interrupt<br>19h = MEMRES16 data loaded interrupt<br>1Ah = MEMRES17 data loaded interrupt<br>1Bh = MEMRES18 data loaded interrupt<br>1Ch = MEMRES19 data loaded interrupt<br>1Dh = MEMRES20 data loaded interrupt<br>1Eh = MEMRES21 data loaded interrupt<br>1Fh = MEMRES22 data loaded interrupt<br>20h = MEMRES23 data loaded interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 18.3.8 IMASK Register (Offset = 1028h) [Reset = 00000000h]

IMASK is shown in Figure 18-12 and described in Table 18-19.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

**Figure 18-12. IMASK Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MEMRESIFG23 | MEMRESIFG22 | MEMRESIFG21 | MEMRESIFG20 | MEMRESIFG19 | MEMRESIFG18 | MEMRESIFG17 | MEMRESIFG16 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MEMRESIFG15 | MEMRESIFG14 | MEMRESIFG13 | MEMRESIFG12 | MEMRESIFG11 | MEMRESIFG10 | MEMRESIFG9 | MEMRESIFG8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MEMRESIFG7 | MEMRESIFG6 | MEMRESIFG5 | MEMRESIFG4 | MEMRESIFG3 | MEMRESIFG2 | MEMRESIFG1 | MEMRESIFG0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | UVIFG | DMADONE | INIFG | LOWIFG | HIGHIFG | TOVIFG | OVIFG |
| R-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

**Table 18-19. IMASK Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | MEMRESIFG23 | R/W | 0h | Raw interrupt status for MEMRES23.<br>This bit is set to 1 when MEMRES23 is loaded with a new conversion result.<br>Reading MEMRES23 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 30 | MEMRESIFG22 | R/W | 0h | Raw interrupt status for MEMRES22.<br>This bit is set to 1 when MEMRES22 is loaded with a new conversion result.<br>Reading MEMRES22 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 29 | MEMRESIFG21 | R/W | 0h | Raw interrupt status for MEMRES21.<br>This bit is set to 1 when MEMRES21 is loaded with a new conversion result.<br>Reading MEMRES21 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 28 | MEMRESIFG20 | R/W | 0h | Raw interrupt status for MEMRES20.<br>This bit is set to 1 when MEMRES20 is loaded with a new conversion result.<br>Reading MEMRES20 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |



Copyright © 2026 Texas Instruments Incorporated

**Table 18-19. IMASK Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 27 | MEMRESIFG19 | R/W | 0h | Raw interrupt status for MEMRES19.<br>This bit is set to 1 when MEMRES19 is loaded with a new conversion result.<br>Reading MEMRES19 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 26 | MEMRESIFG18 | R/W | 0h | Raw interrupt status for MEMRES18.<br>This bit is set to 1 when MEMRES18 is loaded with a new conversion result.<br>Reading MEMRES18 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 25 | MEMRESIFG17 | R/W | 0h | Raw interrupt status for MEMRES17.<br>This bit is set to 1 when MEMRES17 is loaded with a new conversion result.<br>Reading MEMRES17 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 24 | MEMRESIFG16 | R/W | 0h | Raw interrupt status for MEMRES16.<br>This bit is set to 1 when MEMRES16 is loaded with a new conversion result.<br>Reading MEMRES16 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 23 | MEMRESIFG15 | R/W | 0h | Raw interrupt status for MEMRES15.<br>This bit is set to 1 when MEMRES15 is loaded with a new conversion result.<br>Reading MEMRES15 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 22 | MEMRESIFG14 | R/W | 0h | Raw interrupt status for MEMRES14.<br>This bit is set to 1 when MEMRES14 is loaded with a new conversion result.<br>Reading MEMRES14 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 21 | MEMRESIFG13 | R/W | 0h | Raw interrupt status for MEMRES13.<br>This bit is set to 1 when MEMRES13 is loaded with a new conversion result.<br>Reading MEMRES13 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 20 | MEMRESIFG12 | R/W | 0h | Raw interrupt status for MEMRES12.<br>This bit is set to 1 when MEMRES12 is loaded with a new conversion result.<br>Reading MEMRES12 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |



Copyright © 2026 Texas Instruments Incorporated

**Table 18-19. IMASK Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 19 | MEMRESIFG11 | R/W | 0h | Raw interrupt status for MEMRES11.<br>This bit is set to 1 when MEMRES11 is loaded with a new conversion result.<br>Reading MEMRES11 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 18 | MEMRESIFG10 | R/W | 0h | Raw interrupt status for MEMRES10.<br>This bit is set to 1 when MEMRES10 is loaded with a new conversion result.<br>Reading MEMRES10 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 17 | MEMRESIFG9 | R/W | 0h | Raw interrupt status for MEMRES9.<br>This bit is set to 1 when MEMRES9 is loaded with a new conversion result.<br>Reading MEMRES9 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 16 | MEMRESIFG8 | R/W | 0h | Raw interrupt status for MEMRES8.<br>This bit is set to 1 when MEMRES8 is loaded with a new conversion result.<br>Reading MEMRES8 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 15 | MEMRESIFG7 | R/W | 0h | Raw interrupt status for MEMRES7.<br>This bit is set to 1 when MEMRES7 is loaded with a new conversion result.<br>Reading MEMRES7 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 14 | MEMRESIFG6 | R/W | 0h | Raw interrupt status for MEMRES6.<br>This bit is set to 1 when MEMRES6 is loaded with a new conversion result.<br>Reading MEMRES6 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 13 | MEMRESIFG5 | R/W | 0h | Raw interrupt status for MEMRES5.<br>This bit is set to 1 when MEMRES5 is loaded with a new conversion result.<br>Reading MEMRES5 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 12 | MEMRESIFG4 | R/W | 0h | Raw interrupt status for MEMRES4.<br>This bit is set to 1 when MEMRES4 is loaded with a new conversion result.<br>Reading MEMRES4 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |



Copyright © 2026 Texas Instruments Incorporated

**Table 18-19. IMASK Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 11 | MEMRESIFG3 | R/W | 0h | Raw interrupt status for MEMRES3.<br>This bit is set to 1 when MEMRES3 is loaded with a new conversion result.<br>Reading MEMRES3 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 10 | MEMRESIFG2 | R/W | 0h | Raw interrupt status for MEMRES2.<br>This bit is set to 1 when MEMRES2 is loaded with a new conversion result.<br>Reading MEMRES2 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 9 | MEMRESIFG1 | R/W | 0h | Raw interrupt status for MEMRES1.<br>This bit is set to 1 when MEMRES1 is loaded with a new conversion result.<br>Reading MEMRES1 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 8 | MEMRESIFG0 | R/W | 0h | Raw interrupt status for MEMRES0.<br>This bit is set to 1 when MEMRES0 is loaded with a new conversion result.<br>Reading MEMRES0 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 7 | RESERVED | R | 0h | |
| 6 | UVIFG | R/W | 0h | Raw interrupt flag for MEMRESx underflow.<br>This bit is reset to 0 by IIDX read or when corresponding bit in ICLR is set to 1.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 5 | DMADONE | R/W | 0h | Raw interrupt flag for DMADONE.<br>This bit is reset to 0 by IIDX read or when corresponding bit in ICLR_EX is set to 1.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 4 | INIFG | R/W | 0h | Mask INIFG in MIS_EX register.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 3 | LOWIFG | R/W | 0h | Raw interrupt flag for the MEMRESx result register being below than the WCLOWx threshold of the window comparator.<br>This bit is reset to 0 by IIDX read or when corresponding bit in ICLR_EX is set to 1.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 2 | HIGHIFG | R/W | 0h | Raw interrupt flag for the MEMRESx result register being higher than the WCHIGHx threshold of the window comparator.<br>This bit is reset to 0 by IIDX read or when corresponding bit in ICLR_EX is set to 1.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 1 | TOVIFG | R/W | 0h | Raw interrupt flag for sequence conversion timeout overflow.<br>This bit is reset to 0 by IIDX read or when corresponding bit in ICLR_EX is set to 1.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |



Copyright © 2026 Texas Instruments Incorporated

**Table 18-19. IMASK Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 0 | OVIFG | R/W | 0h | Raw interrupt flag for MEMRESx overflow.<br>This bit is reset to 0 by IIDX read or when corresponding bit in ICLR_EX is set to 1.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |



Copyright © 2026 Texas Instruments Incorporated

### 18.3.9 RIS Register (Offset = 1030h) [Reset = 00000000h]

RIS is shown in Figure 18-13 and described in Table 18-20.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Figure 18-13. RIS Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MEMRESIFG23 | MEMRESIFG22 | MEMRESIFG21 | MEMRESIFG20 | MEMRESIFG19 | MEMRESIFG18 | MEMRESIFG17 | MEMRESIFG16 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MEMRESIFG15 | MEMRESIFG14 | MEMRESIFG13 | MEMRESIFG12 | MEMRESIFG11 | MEMRESIFG10 | MEMRESIFG9 | MEMRESIFG8 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MEMRESIFG7 | MEMRESIFG6 | MEMRESIFG5 | MEMRESIFG4 | MEMRESIFG3 | MEMRESIFG2 | MEMRESIFG1 | MEMRESIFG0 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | UVIFG | DMADONE | INIFG | LOWIFG | HIGHIFG | TOVIFG | OVIFG |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 18-20. RIS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | MEMRESIFG23 | R | 0h | Raw interrupt status for MEMRES23.<br>This bit is set to 1 when MEMRES23 is loaded with a new conversion result.<br>Reading MEMRES23 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 30 | MEMRESIFG22 | R | 0h | Raw interrupt status for MEMRES22.<br>This bit is set to 1 when MEMRES22 is loaded with a new conversion result.<br>Reading MEMRES22 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 29 | MEMRESIFG21 | R | 0h | Raw interrupt status for MEMRES21.<br>This bit is set to 1 when MEMRES21 is loaded with a new conversion result.<br>Reading MEMRES21 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 28 | MEMRESIFG20 | R | 0h | Raw interrupt status for MEMRES20.<br>This bit is set to 1 when MEMRES20 is loaded with a new conversion result.<br>Reading MEMRES20 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |



Copyright © 2026 Texas Instruments Incorporated

**Table 18-20. RIS Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 27 | MEMRESIFG19 | R | 0h | Raw interrupt status for MEMRES19.<br>This bit is set to 1 when MEMRES19 is loaded with a new conversion result.<br>Reading MEMRES19 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 26 | MEMRESIFG18 | R | 0h | Raw interrupt status for MEMRES18.<br>This bit is set to 1 when MEMRES18 is loaded with a new conversion result.<br>Reading MEMRES18 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 25 | MEMRESIFG17 | R | 0h | Raw interrupt status for MEMRES17.<br>This bit is set to 1 when MEMRES17 is loaded with a new conversion result.<br>Reading MEMRES17 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 24 | MEMRESIFG16 | R | 0h | Raw interrupt status for MEMRES16.<br>This bit is set to 1 when MEMRES16 is loaded with a new conversion result.<br>Reading MEMRES16 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 23 | MEMRESIFG15 | R | 0h | Raw interrupt status for MEMRES15.<br>This bit is set to 1 when MEMRES15 is loaded with a new conversion result.<br>Reading MEMRES15 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 22 | MEMRESIFG14 | R | 0h | Raw interrupt status for MEMRES14.<br>This bit is set to 1 when MEMRES14 is loaded with a new conversion result.<br>Reading MEMRES14 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 21 | MEMRESIFG13 | R | 0h | Raw interrupt status for MEMRES13.<br>This bit is set to 1 when MEMRES13 is loaded with a new conversion result.<br>Reading MEMRES13 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 20 | MEMRESIFG12 | R | 0h | Raw interrupt status for MEMRES12.<br>This bit is set to 1 when MEMRES12 is loaded with a new conversion result.<br>Reading MEMRES12 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |



Copyright © 2026 Texas Instruments Incorporated

**Table 18-20. RIS Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 19 | MEMRESIFG11 | R | 0h | Raw interrupt status for MEMRES11.<br>This bit is set to 1 when MEMRES11 is loaded with a new conversion result.<br>Reading MEMRES11 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 18 | MEMRESIFG10 | R | 0h | Raw interrupt status for MEMRES10.<br>This bit is set to 1 when MEMRES10 is loaded with a new conversion result.<br>Reading MEMRES10 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 17 | MEMRESIFG9 | R | 0h | Raw interrupt status for MEMRES9.<br>This bit is set to 1 when MEMRES9 is loaded with a new conversion result.<br>Reading MEMRES9 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 16 | MEMRESIFG8 | R | 0h | Raw interrupt status for MEMRES8.<br>This bit is set to 1 when MEMRES8 is loaded with a new conversion result.<br>Reading MEMRES8 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 15 | MEMRESIFG7 | R | 0h | Raw interrupt status for MEMRES7.<br>This bit is set to 1 when MEMRES7 is loaded with a new conversion result.<br>Reading MEMRES7 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 14 | MEMRESIFG6 | R | 0h | Raw interrupt status for MEMRES6.<br>This bit is set to 1 when MEMRES6 is loaded with a new conversion result.<br>Reading MEMRES6 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 13 | MEMRESIFG5 | R | 0h | Raw interrupt status for MEMRES5.<br>This bit is set to 1 when MEMRES5 is loaded with a new conversion result.<br>Reading MEMRES5 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 12 | MEMRESIFG4 | R | 0h | Raw interrupt status for MEMRES4.<br>This bit is set to 1 when MEMRES4 is loaded with a new conversion result.<br>Reading MEMRES4 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |



Copyright © 2026 Texas Instruments Incorporated

**Table 18-20. RIS Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 11 | MEMRESIFG3 | R | 0h | Raw interrupt status for MEMRES3.<br>This bit is set to 1 when MEMRES3 is loaded with a new conversion result.<br>Reading MEMRES3 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 10 | MEMRESIFG2 | R | 0h | Raw interrupt status for MEMRES2.<br>This bit is set to 1 when MEMRES2 is loaded with a new conversion result.<br>Reading MEMRES2 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 9 | MEMRESIFG1 | R | 0h | Raw interrupt status for MEMRES1.<br>This bit is set to 1 when MEMRES1 is loaded with a new conversion result.<br>Reading MEMRES1 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 8 | MEMRESIFG0 | R | 0h | Raw interrupt status for MEMRES0.<br>This bit is set to 1 when MEMRES0 is loaded with a new conversion result.<br>Reading MEMRES0 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 7 | RESERVED | R | 0h | |
| 6 | UVIFG | R | 0h | Raw interrupt flag for MEMRESx underflow.<br>This bit is reset to 0 by IIDX read or when corresponding bit in ICLR is set to 1.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 5 | DMADONE | R | 0h | Raw interrupt flag for DMADONE.<br>This bit is reset to 0 by IIDX read or when corresponding bit in ICLR_EX is set to 1.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 4 | INIFG | R | 0h | Mask INIFG in MIS_EX register.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 3 | LOWIFG | R | 0h | Raw interrupt flag for the MEMRESx result register being below than the WCLOWx threshold of the window comparator.<br>This bit is reset to 0 by IIDX read or when corresponding bit in ICLR_EX is set to 1.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 2 | HIGHIFG | R | 0h | Raw interrupt flag for the MEMRESx result register being higher than the WCHIGHx threshold of the window comparator.<br>This bit is reset to 0 by IIDX read or when corresponding bit in ICLR_EX is set to 1.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 1 | TOVIFG | R | 0h | Raw interrupt flag for sequence conversion trigger overflow.<br>This bit is reset to 0 by IIDX read or when corresponding bit in ICLR_EX is set to 1.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |



Copyright © 2026 Texas Instruments Incorporated

**Table 18-20. RIS Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 0 | OVIFG | R | 0h | Raw interrupt flag for MEMRESx overflow.<br>This bit is reset to 0 by IIDX read or when corresponding bit in ICLR_EX is set to 1.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |



Copyright © 2026 Texas Instruments Incorporated

### 18.3.10 MIS Register (Offset = 1038h) [Reset = 00000000h]

MIS is shown in Figure 18-14 and described in Table 18-21.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 18-14. MIS Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MEMRESIFG23 | MEMRESIFG22 | MEMRESIFG21 | MEMRESIFG20 | MEMRESIFG19 | MEMRESIFG18 | MEMRESIFG17 | MEMRESIFG16 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MEMRESIFG15 | MEMRESIFG14 | MEMRESIFG13 | MEMRESIFG12 | MEMRESIFG11 | MEMRESIFG10 | MEMRESIFG9 | MEMRESIFG8 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MEMRESIFG7 | MEMRESIFG6 | MEMRESIFG5 | MEMRESIFG4 | MEMRESIFG3 | MEMRESIFG2 | MEMRESIFG1 | MEMRESIFG0 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | UVIFG | DMADONE | INIFG | LOWIFG | HIGHIFG | TOVIFG | OVIFG |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 18-21. MIS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | MEMRESIFG23 | R | 0h | Raw interrupt status for MEMRES23.<br>This bit is set to 1 when MEMRES23 is loaded with a new conversion result.<br>Reading MEMRES23 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 30 | MEMRESIFG22 | R | 0h | Raw interrupt status for MEMRES22.<br>This bit is set to 1 when MEMRES22 is loaded with a new conversion result.<br>Reading MEMRES22 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 29 | MEMRESIFG21 | R | 0h | Raw interrupt status for MEMRES21.<br>This bit is set to 1 when MEMRES21 is loaded with a new conversion result.<br>Reading MEMRES21 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 28 | MEMRESIFG20 | R | 0h | Raw interrupt status for MEMRES20.<br>This bit is set to 1 when MEMRES20 is loaded with a new conversion result.<br>Reading MEMRES20 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |



Copyright © 2026 Texas Instruments Incorporated

**Table 18-21. MIS Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 27 | MEMRESIFG19 | R | 0h | Raw interrupt status for MEMRES19.<br>This bit is set to 1 when MEMRES19 is loaded with a new conversion result.<br>Reading MEMRES19 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 26 | MEMRESIFG18 | R | 0h | Raw interrupt status for MEMRES18.<br>This bit is set to 1 when MEMRES18 is loaded with a new conversion result.<br>Reading MEMRES18 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 25 | MEMRESIFG17 | R | 0h | Raw interrupt status for MEMRES17.<br>This bit is set to 1 when MEMRES17 is loaded with a new conversion result.<br>Reading MEMRES17 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 24 | MEMRESIFG16 | R | 0h | Raw interrupt status for MEMRES16.<br>This bit is set to 1 when MEMRES16 is loaded with a new conversion result.<br>Reading MEMRES16 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 23 | MEMRESIFG15 | R | 0h | Raw interrupt status for MEMRES15.<br>This bit is set to 1 when MEMRES15 is loaded with a new conversion result.<br>Reading MEMRES15 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 22 | MEMRESIFG14 | R | 0h | Raw interrupt status for MEMRES14.<br>This bit is set to 1 when MEMRES14 is loaded with a new conversion result.<br>Reading MEMRES14 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 21 | MEMRESIFG13 | R | 0h | Raw interrupt status for MEMRES13.<br>This bit is set to 1 when MEMRES13 is loaded with a new conversion result.<br>Reading MEMRES13 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 20 | MEMRESIFG12 | R | 0h | Raw interrupt status for MEMRES12.<br>This bit is set to 1 when MEMRES12 is loaded with a new conversion result.<br>Reading MEMRES12 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |



Copyright © 2026 Texas Instruments Incorporated

**Table 18-21. MIS Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 19 | MEMRESIFG11 | R | 0h | Raw interrupt status for MEMRES11.<br>This bit is set to 1 when MEMRES11 is loaded with a new conversion result.<br>Reading MEMRES11 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 18 | MEMRESIFG10 | R | 0h | Raw interrupt status for MEMRES10.<br>This bit is set to 1 when MEMRES10 is loaded with a new conversion result.<br>Reading MEMRES10 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 17 | MEMRESIFG9 | R | 0h | Raw interrupt status for MEMRES9.<br>This bit is set to 1 when MEMRES9 is loaded with a new conversion result.<br>Reading MEMRES9 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 16 | MEMRESIFG8 | R | 0h | Raw interrupt status for MEMRES8.<br>This bit is set to 1 when MEMRES8 is loaded with a new conversion result.<br>Reading MEMRES8 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 15 | MEMRESIFG7 | R | 0h | Raw interrupt status for MEMRES7.<br>This bit is set to 1 when MEMRES7 is loaded with a new conversion result.<br>Reading MEMRES7 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 14 | MEMRESIFG6 | R | 0h | Raw interrupt status for MEMRES6.<br>This bit is set to 1 when MEMRES6 is loaded with a new conversion result.<br>Reading MEMRES6 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 13 | MEMRESIFG5 | R | 0h | Raw interrupt status for MEMRES5.<br>This bit is set to 1 when MEMRES5 is loaded with a new conversion result.<br>Reading MEMRES5 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 12 | MEMRESIFG4 | R | 0h | Raw interrupt status for MEMRES4.<br>This bit is set to 1 when MEMRES4 is loaded with a new conversion result.<br>Reading MEMRES4 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |



Copyright © 2026 Texas Instruments Incorporated

**Table 18-21. MIS Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 11 | MEMRESIFG3 | R | 0h | Raw interrupt status for MEMRES3.<br>This bit is set to 1 when MEMRES3 is loaded with a new conversion result.<br>Reading MEMRES3 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 10 | MEMRESIFG2 | R | 0h | Raw interrupt status for MEMRES2.<br>This bit is set to 1 when MEMRES2 is loaded with a new conversion result.<br>Reading MEMRES2 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 9 | MEMRESIFG1 | R | 0h | Raw interrupt status for MEMRES1.<br>This bit is set to 1 when MEMRES1 is loaded with a new conversion result.<br>Reading MEMRES1 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 8 | MEMRESIFG0 | R | 0h | Raw interrupt status for MEMRES0.<br>This bit is set to 1 when MEMRES0 is loaded with a new conversion result.<br>Reading MEMRES0 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 7 | RESERVED | R | 0h | |
| 6 | UVIFG | R | 0h | Raw interrupt flag for MEMRESx underflow.<br>This bit is reset to 0 by IIDX read or when corresponding bit in ICLR is set to 1.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 5 | DMADONE | R | 0h | Raw interrupt flag for DMADONE.<br>This bit is reset to 0 by IIDX read or when corresponding bit in ICLR_EX is set to 1.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 4 | INIFG | R | 0h | Mask INIFG in MIS_EX register.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 3 | LOWIFG | R | 0h | Raw interrupt flag for the MEMRESx result register being below than the WCLOWx threshold of the window comparator.<br>This bit is reset to 0 by IIDX read or when corresponding bit in ICLR_EX is set to 1.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 2 | HIGHIFG | R | 0h | Raw interrupt flag for the MEMRESx result register being higher than the WCHIGHx threshold of the window comparator.<br>This bit is reset to 0 by IIDX read or when corresponding bit in ICLR_EX is set to 1.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 1 | TOVIFG | R | 0h | Raw interrupt flag for sequence conversion timeout overflow.<br>This bit is reset to 0 by IIDX read or when corresponding bit in ICLR_EX is set to 1.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |



Copyright © 2026 Texas Instruments Incorporated

**Table 18-21. MIS Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 0 | OVIFG | R | 0h | Raw interrupt flag for MEMRESx overflow.<br>This bit is reset to 0 by IIDX read or when corresponding bit in ICLR_EX is set to 1.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |



Copyright © 2026 Texas Instruments Incorporated

### 18.3.11 ISET Register (Offset = 1040h) [Reset = 00000000h]

ISET is shown in Figure 18-15 and described in Table 18-22.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 18-15. ISET Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MEMRESIFG23 | MEMRESIFG22 | MEMRESIFG21 | MEMRESIFG20 | MEMRESIFG19 | MEMRESIFG18 | MEMRESIFG17 | MEMRESIFG16 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MEMRESIFG15 | MEMRESIFG14 | MEMRESIFG13 | MEMRESIFG12 | MEMRESIFG11 | MEMRESIFG10 | MEMRESIFG9 | MEMRESIFG8 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MEMRESIFG7 | MEMRESIFG6 | MEMRESIFG5 | MEMRESIFG4 | MEMRESIFG3 | MEMRESIFG2 | MEMRESIFG1 | MEMRESIFG0 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | UVIFG | DMADONE | INIFG | LOWIFG | HIGHIFG | TOVIFG | OVIFG |
| R-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

**Table 18-22. ISET Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | MEMRESIFG23 | W | 0h | Raw interrupt status for MEMRES23.<br>This bit is set to 1 when MEMRES23 is loaded with a new conversion result.<br>Reading MEMRES23 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 30 | MEMRESIFG22 | W | 0h | Raw interrupt status for MEMRES22.<br>This bit is set to 1 when MEMRES22 is loaded with a new conversion result.<br>Reading MEMRES22 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 29 | MEMRESIFG21 | W | 0h | Raw interrupt status for MEMRES21.<br>This bit is set to 1 when MEMRES21 is loaded with a new conversion result.<br>Reading MEMRES21 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 28 | MEMRESIFG20 | W | 0h | Raw interrupt status for MEMRES20.<br>This bit is set to 1 when MEMRES20 is loaded with a new conversion result.<br>Reading MEMRES20 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |



Copyright © 2026 Texas Instruments Incorporated

**Table 18-22. ISET Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 27 | MEMRESIFG19 | W | 0h | Raw interrupt status for MEMRES19.<br>This bit is set to 1 when MEMRES19 is loaded with a new conversion result.<br>Reading MEMRES19 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 26 | MEMRESIFG18 | W | 0h | Raw interrupt status for MEMRES18.<br>This bit is set to 1 when MEMRES18 is loaded with a new conversion result.<br>Reading MEMRES18 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 25 | MEMRESIFG17 | W | 0h | Raw interrupt status for MEMRES17.<br>This bit is set to 1 when MEMRES17 is loaded with a new conversion result.<br>Reading MEMRES17 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 24 | MEMRESIFG16 | W | 0h | Raw interrupt status for MEMRES16.<br>This bit is set to 1 when MEMRES16 is loaded with a new conversion result.<br>Reading MEMRES16 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 23 | MEMRESIFG15 | W | 0h | Raw interrupt status for MEMRES15.<br>This bit is set to 1 when MEMRES15 is loaded with a new conversion result.<br>Reading MEMRES15 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 22 | MEMRESIFG14 | W | 0h | Raw interrupt status for MEMRES14.<br>This bit is set to 1 when MEMRES14 is loaded with a new conversion result.<br>Reading MEMRES14 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 21 | MEMRESIFG13 | W | 0h | Raw interrupt status for MEMRES13.<br>This bit is set to 1 when MEMRES13 is loaded with a new conversion result.<br>Reading MEMRES13 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 20 | MEMRESIFG12 | W | 0h | Raw interrupt status for MEMRES12.<br>This bit is set to 1 when MEMRES12 is loaded with a new conversion result.<br>Reading MEMRES12 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |



Copyright © 2026 Texas Instruments Incorporated

**Table 18-22. ISET Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 19 | MEMRESIFG11 | W | 0h | Raw interrupt status for MEMRES11.<br>This bit is set to 1 when MEMRES11 is loaded with a new conversion result.<br>Reading MEMRES11 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 18 | MEMRESIFG10 | W | 0h | Raw interrupt status for MEMRES10.<br>This bit is set to 1 when MEMRES10 is loaded with a new conversion result.<br>Reading MEMRES10 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 17 | MEMRESIFG9 | W | 0h | Raw interrupt status for MEMRES9.<br>This bit is set to 1 when MEMRES9 is loaded with a new conversion result.<br>Reading MEMRES9 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 16 | MEMRESIFG8 | W | 0h | Raw interrupt status for MEMRES8.<br>This bit is set to 1 when MEMRES8 is loaded with a new conversion result.<br>Reading MEMRES8 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 15 | MEMRESIFG7 | W | 0h | Raw interrupt status for MEMRES7.<br>This bit is set to 1 when MEMRES7 is loaded with a new conversion result.<br>Reading MEMRES7 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 14 | MEMRESIFG6 | W | 0h | Raw interrupt status for MEMRES6.<br>This bit is set to 1 when MEMRES6 is loaded with a new conversion result.<br>Reading MEMRES6 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 13 | MEMRESIFG5 | W | 0h | Raw interrupt status for MEMRES5.<br>This bit is set to 1 when MEMRES5 is loaded with a new conversion result.<br>Reading MEMRES5 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 12 | MEMRESIFG4 | W | 0h | Raw interrupt status for MEMRES4.<br>This bit is set to 1 when MEMRES4 is loaded with a new conversion result.<br>Reading MEMRES4 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |



Copyright © 2026 Texas Instruments Incorporated

**Table 18-22. ISET Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 11 | MEMRESIFG3 | W | 0h | Raw interrupt status for MEMRES3.<br>This bit is set to 1 when MEMRES3 is loaded with a new conversion result.<br>Reading MEMRES3 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 10 | MEMRESIFG2 | W | 0h | Raw interrupt status for MEMRES2.<br>This bit is set to 1 when MEMRES2 is loaded with a new conversion result.<br>Reading MEMRES2 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 9 | MEMRESIFG1 | W | 0h | Raw interrupt status for MEMRES1.<br>This bit is set to 1 when MEMRES1 is loaded with a new conversion result.<br>Reading MEMRES1 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 8 | MEMRESIFG0 | W | 0h | Raw interrupt status for MEMRES0.<br>This bit is set to 1 when MEMRES0 is loaded with a new conversion result.<br>Reading MEMRES0 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 7 | RESERVED | R | 0h | |
| 6 | UVIFG | W | 0h | Raw interrupt flag for MEMRESx underflow.<br>This bit is reset to 0 by IIDX read or when corresponding bit in ICLR_EX is set to 1.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 5 | DMADONE | W | 0h | Raw interrupt flag for DMADONE.<br>This bit is reset to 0 by IIDX read or when corresponding bit in ICLR_EX is set to 1.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 4 | INIFG | W | 0h | Mask INIFG in MIS_EX register.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 3 | LOWIFG | W | 0h | Raw interrupt flag for the MEMRESx result register being below than the WCLOWx threshold of the window comparator.<br>This bit is reset to 0 by IIDX read or when corresponding bit in ICLR_EX is set to 1.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 2 | HIGHIFG | W | 0h | Raw interrupt flag for the MEMRESx result register being higher than the WCHIGHx threshold of the window comparator.<br>This bit is reset to 0 by IIDX read or when corresponding bit in ICLR_EX is set to 1.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 1 | TOVIFG | W | 0h | Raw interrupt flag for sequence conversion timeout overflow.<br>This bit is reset to 0 by IIDX read or when corresponding bit in ICLR_EX is set to 1.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |



Copyright © 2026 Texas Instruments Incorporated

**Table 18-22. ISET Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 0 | OVIFG | W | 0h | Raw interrupt flag for MEMRESx overflow.<br>This bit is reset to 0 by IIDX read or when corresponding bit in ICLR_EX is set to 1.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |



Copyright © 2026 Texas Instruments Incorporated

### 18.3.12 ICLR Register (Offset = 1048h) [Reset = 00000000h]

ICLR is shown in Figure 18-16 and described in Table 18-23.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

**Figure 18-16. ICLR Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MEMRESIFG23 | MEMRESIFG22 | MEMRESIFG21 | MEMRESIFG20 | MEMRESIFG19 | MEMRESIFG18 | MEMRESIFG17 | MEMRESIFG16 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MEMRESIFG15 | MEMRESIFG14 | MEMRESIFG13 | MEMRESIFG12 | MEMRESIFG11 | MEMRESIFG10 | MEMRESIFG9 | MEMRESIFG8 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MEMRESIFG7 | MEMRESIFG6 | MEMRESIFG5 | MEMRESIFG4 | MEMRESIFG3 | MEMRESIFG2 | MEMRESIFG1 | MEMRESIFG0 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | UVIFG | DMADONE | INIFG | LOWIFG | HIGHIFG | TOVIFG | OVIFG |
| R-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

**Table 18-23. ICLR Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | MEMRESIFG23 | W | 0h | Raw interrupt status for MEMRES23.<br>This bit is set to 1 when MEMRES23 is loaded with a new conversion result.<br>Reading MEMRES23 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 30 | MEMRESIFG22 | W | 0h | Raw interrupt status for MEMRES22.<br>This bit is set to 1 when MEMRES22 is loaded with a new conversion result.<br>Reading MEMRES22 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 29 | MEMRESIFG21 | W | 0h | Raw interrupt status for MEMRES21.<br>This bit is set to 1 when MEMRES21 is loaded with a new conversion result.<br>Reading MEMRES21 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 28 | MEMRESIFG20 | W | 0h | Raw interrupt status for MEMRES20.<br>This bit is set to 1 when MEMRES20 is loaded with a new conversion result.<br>Reading MEMRES20 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |



Copyright © 2026 Texas Instruments Incorporated

**Table 18-23. ICLR Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 27 | MEMRESIFG19 | W | 0h | Raw interrupt status for MEMRES19.<br>This bit is set to 1 when MEMRES19 is loaded with a new conversion result.<br>Reading MEMRES19 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 26 | MEMRESIFG18 | W | 0h | Raw interrupt status for MEMRES18.<br>This bit is set to 1 when MEMRES18 is loaded with a new conversion result.<br>Reading MEMRES18 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 25 | MEMRESIFG17 | W | 0h | Raw interrupt status for MEMRES17.<br>This bit is set to 1 when MEMRES17 is loaded with a new conversion result.<br>Reading MEMRES17 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 24 | MEMRESIFG16 | W | 0h | Raw interrupt status for MEMRES16.<br>This bit is set to 1 when MEMRES16 is loaded with a new conversion result.<br>Reading MEMRES16 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 23 | MEMRESIFG15 | W | 0h | Raw interrupt status for MEMRES15.<br>This bit is set to 1 when MEMRES15 is loaded with a new conversion result.<br>Reading MEMRES15 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 22 | MEMRESIFG14 | W | 0h | Raw interrupt status for MEMRES14.<br>This bit is set to 1 when MEMRES14 is loaded with a new conversion result.<br>Reading MEMRES14 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 21 | MEMRESIFG13 | W | 0h | Raw interrupt status for MEMRES13.<br>This bit is set to 1 when MEMRES13 is loaded with a new conversion result.<br>Reading MEMRES13 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 20 | MEMRESIFG12 | W | 0h | Raw interrupt status for MEMRES12.<br>This bit is set to 1 when MEMRES12 is loaded with a new conversion result.<br>Reading MEMRES12 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |



Copyright © 2026 Texas Instruments Incorporated

**Table 18-23. ICLR Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 19 | MEMRESIFG11 | W | 0h | Raw interrupt status for MEMRES11.<br>This bit is set to 1 when MEMRES11 is loaded with a new conversion result.<br>Reading MEMRES11 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 18 | MEMRESIFG10 | W | 0h | Raw interrupt status for MEMRES10.<br>This bit is set to 1 when MEMRES10 is loaded with a new conversion result.<br>Reading MEMRES10 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 17 | MEMRESIFG9 | W | 0h | Raw interrupt status for MEMRES9.<br>This bit is set to 1 when MEMRES9 is loaded with a new conversion result.<br>Reading MEMRES9 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 16 | MEMRESIFG8 | W | 0h | Raw interrupt status for MEMRES8.<br>This bit is set to 1 when MEMRES8 is loaded with a new conversion result.<br>Reading MEMRES8 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 15 | MEMRESIFG7 | W | 0h | Raw interrupt status for MEMRES7.<br>This bit is set to 1 when MEMRES7 is loaded with a new conversion result.<br>Reading MEMRES7 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 14 | MEMRESIFG6 | W | 0h | Raw interrupt status for MEMRES6.<br>This bit is set to 1 when MEMRES6 is loaded with a new conversion result.<br>Reading MEMRES6 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 13 | MEMRESIFG5 | W | 0h | Raw interrupt status for MEMRES5.<br>This bit is set to 1 when MEMRES5 is loaded with a new conversion result.<br>Reading MEMRES5 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 12 | MEMRESIFG4 | W | 0h | Raw interrupt status for MEMRES4.<br>This bit is set to 1 when MEMRES4 is loaded with a new conversion result.<br>Reading MEMRES4 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |



Copyright © 2026 Texas Instruments Incorporated

**Table 18-23. ICLR Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 11 | MEMRESIFG3 | W | 0h | Raw interrupt status for MEMRES3.<br>This bit is set to 1 when MEMRES3 is loaded with a new conversion result.<br>Reading MEMRES3 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 10 | MEMRESIFG2 | W | 0h | Raw interrupt status for MEMRES2.<br>This bit is set to 1 when MEMRES2 is loaded with a new conversion result.<br>Reading MEMRES2 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 9 | MEMRESIFG1 | W | 0h | Raw interrupt status for MEMRES1.<br>This bit is set to 1 when MEMRES1 is loaded with a new conversion result.<br>Reading MEMRES1 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 8 | MEMRESIFG0 | W | 0h | Raw interrupt status for MEMRES0.<br>This bit is set to 1 when MEMRES0 is loaded with a new conversion result.<br>Reading MEMRES0 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 7 | RESERVED | R | 0h | |
| 6 | UVIFG | W | 0h | Raw interrupt flag for MEMRESx underflow.<br>This bit is reset to 0 by IIDX read or when corresponding bit in ICLR_EX is set to 1.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 5 | DMADONE | W | 0h | Raw interrupt flag for DMADONE.<br>This bit is reset to 0 by IIDX read or when corresponding bit in ICLR_EX is set to 1.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 4 | INIFG | W | 0h | Mask INIFG in MIS_EX register.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 3 | LOWIFG | W | 0h | Raw interrupt flag for the MEMRESx result register being below than the WCLOWx threshold of the window comparator.<br>This bit is reset to 0 by IIDX read or when corresponding bit in ICLR_EX is set to 1.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 2 | HIGHIFG | W | 0h | Raw interrupt flag for the MEMRESx result register being higher than the WCHIGHx threshold of the window comparator.<br>This bit is reset to 0 by IIDX read or when corresponding bit in ICLR_EX is set to 1.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 1 | TOVIFG | W | 0h | Raw interrupt flag for sequence conversion timeout overflow.<br>This bit is reset to 0 by IIDX read or when corresponding bit in ICLR_EX is set to 1.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |



Copyright © 2026 Texas Instruments Incorporated

**Table 18-23. ICLR Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 0 | OVIFG | W | 0h | Raw interrupt flag for MEMRESx overflow.<br>This bit is reset to 0 by IIDX read or when corresponding bit in ICLR_EX is set to 1.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |



Copyright © 2026 Texas Instruments Incorporated

### 18.3.13 IIDX Register (Offset = 1050h) [Reset = 00000000h]

IIDX is shown in Figure 18-17 and described in Table 18-24.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. 0x0 means no event pending. Interrupt 1 is the highest priority, 2 next highest, 4, 8, … $2^{31}$ is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred.
On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in the RIS and MIS are cleared as well. After a read from the CPU (not from the debug interface), the register must be updated with the next highest priority interrupt, if none are pending, then it should display 0x0.

**Figure 18-17. IIDX Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | STAT | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | | | |

**Table 18-24. IIDX Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-10 | RESERVED | R | 0h | |
| 9-0 | STAT | R | 0h | Interrupt index status<br>00h = No bit is set means there is no pending interrupt request<br>03h = High threshold compare interrupt<br>04h = Low threshold compare interrupt<br>05h = Primary Sequence In range comparator interrupt<br>9h = MEMRES0 data loaded interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 18.3.14 IMASK Register (Offset = 1058h) [Reset = 00000000h]

IMASK is shown in Figure 18-18 and described in Table 18-25.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

**Figure 18-18. IMASK Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | MEMRESIFG0 |
| R-0h | | | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | INIFG | LOWIFG | HIGHIFG | RESERVED | |
| R-0h | | | R/W-0h | R/W-0h | R/W-0h | R-0h | |

**Table 18-25. IMASK Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-9 | RESERVED | R | 0h | |
| 8 | MEMRESIFG0 | R/W | 0h | Raw interrupt status for MEMRES0.<br>This bit is set to 1 when MEMRES0 is loaded with a new conversion result.<br>Reading MEMRES0 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 7-5 | RESERVED | R | 0h | |
| 4 | INIFG | R/W | 0h | Mask INIFG in MIS_EX register.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 3 | LOWIFG | R/W | 0h | Raw interrupt flag for the MEMRESx result register being below than the WCLOWx threshold of the window comparator.<br>This bit is reset to 0 by IIDX read or when corresponding bit in ICLR_EX is set to 1.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 2 | HIGHIFG | R/W | 0h | Raw interrupt flag for the MEMRESx result register being higher than the WCHIGHx threshold of the window comparator.<br>This bit is reset to 0 by IIDX read or when corresponding bit in ICLR_EX is set to 1.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 1-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 18.3.15 RIS Register (Offset = 1060h) [Reset = 00000000h]

RIS is shown in Figure 18-19 and described in Table 18-26.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Figure 18-19. RIS Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | MEMRESIFG0 |
| R-0h | | | | | | | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | INIFG | LOWIFG | HIGHIFG | RESERVED | |
| R-0h | | | R-0h | R-0h | R-0h | R-0h | |

**Table 18-26. RIS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-9 | RESERVED | R | 0h | |
| 8 | MEMRESIFG0 | R | 0h | Raw interrupt status for MEMRES0.<br>This bit is set to 1 when MEMRES0 is loaded with a new conversion result.<br>Reading MEMRES0 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 7-5 | RESERVED | R | 0h | |
| 4 | INIFG | R | 0h | Mask INIFG in MIS_EX register.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 3 | LOWIFG | R | 0h | Raw interrupt flag for the MEMRESx result register being below than the WCLOWx threshold of the window comparator.<br>This bit is reset to 0 by IIDX read or when corresponding bit in ICLR_EX is set to 1.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 2 | HIGHIFG | R | 0h | Raw interrupt flag for the MEMRESx result register being higher than the WCHIGHx threshold of the window comparator.<br>This bit is reset to 0 by IIDX read or when corresponding bit in ICLR_EX is set to 1.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 1-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 18.3.16 MIS Register (Offset = 1068h) [Reset = 00000000h]

MIS is shown in Figure 18-20 and described in Table 18-27.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 18-20. MIS Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | MEMRESIFG0 |
| R-0h | | | | | | | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | INIFG | LOWIFG | HIGHIFG | RESERVED | |
| R-0h | | | R-0h | R-0h | R-0h | R-0h | |

**Table 18-27. MIS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-9 | RESERVED | R | 0h | |
| 8 | MEMRESIFG0 | R | 0h | Raw interrupt status for MEMRES0.<br>This bit is set to 1 when MEMRES0 is loaded with a new conversion result.<br>Reading MEMRES0 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 7-5 | RESERVED | R | 0h | |
| 4 | INIFG | R | 0h | Mask INIFG in MIS_EX register.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 3 | LOWIFG | R | 0h | Raw interrupt flag for the MEMRESx result register being below than the WCLOWx threshold of the window comparator.<br>This bit is reset to 0 by IIDX read or when corresponding bit in ICLR_EX is set to 1.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 2 | HIGHIFG | R | 0h | Raw interrupt flag for the MEMRESx result register being higher than the WCHIGHx threshold of the window comparator.<br>This bit is reset to 0 by IIDX read or when corresponding bit in ICLR_EX is set to 1.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 1-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 18.3.17 ISET Register (Offset = 1070h) [Reset = 00000000h]

ISET is shown in Figure 18-21 and described in Table 18-28.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 18-21. ISET Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | MEMRESIFG0 |
| R-0h | | | | | | | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | INIFG | LOWIFG | HIGHIFG | RESERVED | |
| R-0h | | | W-0h | W-0h | W-0h | R-0h | |

**Table 18-28. ISET Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-9 | RESERVED | R | 0h | |
| 8 | MEMRESIFG0 | W | 0h | Raw interrupt status for MEMRES0.<br>This bit is set to 1 when MEMRES0 is loaded with a new conversion result.<br>Reading MEMRES0 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 7-5 | RESERVED | R | 0h | |
| 4 | INIFG | W | 0h | Mask INIFG in MIS_EX register.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 3 | LOWIFG | W | 0h | Raw interrupt flag for the MEMRESx result register being below than the WCLOWx threshold of the window comparator.<br>This bit is reset to 0 by IIDX read or when corresponding bit in ICLR_EX is set to 1.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 2 | HIGHIFG | W | 0h | Raw interrupt flag for the MEMRESx result register being higher than the WCHIGHx threshold of the window comparator.<br>This bit is reset to 0 by IIDX read or when corresponding bit in ICLR_EX is set to 1.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 1-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 18.3.18 ICLR Register (Offset = 1078h) [Reset = 00000000h]

ICLR is shown in Figure 18-22 and described in Table 18-29.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

**Figure 18-22. ICLR Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | MEMRESIFG0 |
| R-0h | | | | | | | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | INIFG | LOWIFG | HIGHIFG | RESERVED | |
| R-0h | | | W-0h | W-0h | W-0h | R-0h | |

**Table 18-29. ICLR Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-9 | RESERVED | R | 0h | |
| 8 | MEMRESIFG0 | W | 0h | Raw interrupt status for MEMRES0.<br>This bit is set to 1 when MEMRES0 is loaded with a new conversion result.<br>Reading MEMRES0 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 7-5 | RESERVED | R | 0h | |
| 4 | INIFG | W | 0h | Mask INIFG in MIS_EX register.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 3 | LOWIFG | W | 0h | Raw interrupt flag for the MEMRESx result register being below than the WCLOWx threshold of the window comparator.<br>This bit is reset to 0 by IIDX read or when corresponding bit in ICLR_EX is set to 1.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 2 | HIGHIFG | W | 0h | Raw interrupt flag for the MEMRESx result register being higher than the WCHIGHx threshold of the window comparator.<br>This bit is reset to 0 by IIDX read or when corresponding bit in ICLR_EX is set to 1.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 1-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 18.3.19 IIDX Register (Offset = 1080h) [Reset = 00000000h]

IIDX is shown in Figure 18-23 and described in Table 18-30.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. 0x0 means no event pending. Interrupt 1 is the highest priority, 2 next highest, 4, 8, … $2^{31}$ is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred.
On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in the RIS and MIS are cleared as well. After a read from the CPU (not from the debug interface), the register must be updated with the next highest priority interrupt, if none are pending, then it should display 0x0.

**Figure 18-23. IIDX Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | STAT | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | | | |

**Table 18-30. IIDX Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-10 | RESERVED | R | 0h | |
| 9-0 | STAT | R | 0h | Interrupt index status<br>00h = No bit is set means there is no pending interrupt request<br>9h = MEMRES0 data loaded interrupt<br>Ah = MEMRES1 data loaded interrupt<br>Bh = MEMRES2 data loaded interrupt<br>Ch = MEMRES3 data loaded interrupt<br>Dh = MEMRES4 data loaded interrupt<br>Eh = MEMRES5 data loaded interrupt<br>Fh = MEMRES6 data loaded interrupt<br>10h = MEMRES7 data loaded interrupt<br>11h = MEMRES8 data loaded interrupt<br>12h = MEMRES9 data loaded interrupt<br>13h = MEMRES10 data loaded interrupt<br>14h = MEMRES11 data loaded interrupt<br>15h = MEMRES12 data loaded interrupt<br>16h = MEMRES13 data loaded interrupt<br>17h = MEMRES14 data loaded interrupt<br>18h = MEMRES15 data loaded interrupt<br>19h = MEMRES16 data loaded interrupt<br>1Ah = MEMRES17 data loaded interrupt<br>1Bh = MEMRES18 data loaded interrupt<br>1Ch = MEMRES19 data loaded interrupt<br>1Dh = MEMRES20 data loaded interrupt<br>1Eh = MEMRES21 data loaded interrupt<br>1Fh = MEMRES22 data loaded interrupt<br>20h = MEMRES23 data loaded interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 18.3.20 IMASK Register (Offset = 1088h) [Reset = 00000000h]

IMASK is shown in Figure 18-24 and described in Table 18-31.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

**Figure 18-24. IMASK Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MEMRESIFG23 | MEMRESIFG22 | MEMRESIFG21 | MEMRESIFG20 | MEMRESIFG19 | MEMRESIFG18 | MEMRESIFG17 | MEMRESIFG16 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MEMRESIFG15 | MEMRESIFG14 | MEMRESIFG13 | MEMRESIFG12 | MEMRESIFG11 | MEMRESIFG10 | MEMRESIFG9 | MEMRESIFG8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MEMRESIFG7 | MEMRESIFG6 | MEMRESIFG5 | MEMRESIFG4 | MEMRESIFG3 | MEMRESIFG2 | MEMRESIFG1 | MEMRESIFG0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |

**Table 18-31. IMASK Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | MEMRESIFG23 | R/W | 0h | Raw interrupt status for MEMRES23.<br>This bit is set to 1 when MEMRES23 is loaded with a new conversion result.<br>Reading MEMRES23 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 30 | MEMRESIFG22 | R/W | 0h | Raw interrupt status for MEMRES22.<br>This bit is set to 1 when MEMRES22 is loaded with a new conversion result.<br>Reading MEMRES22 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 29 | MEMRESIFG21 | R/W | 0h | Raw interrupt status for MEMRES21.<br>This bit is set to 1 when MEMRES21 is loaded with a new conversion result.<br>Reading MEMRES21 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 28 | MEMRESIFG20 | R/W | 0h | Raw interrupt status for MEMRES20.<br>This bit is set to 1 when MEMRES20 is loaded with a new conversion result.<br>Reading MEMRES20 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |



Copyright © 2026 Texas Instruments Incorporated

**Table 18-31. IMASK Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 27 | MEMRESIFG19 | R/W | 0h | Raw interrupt status for MEMRES19.<br>This bit is set to 1 when MEMRES19 is loaded with a new conversion result.<br>Reading MEMRES19 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 26 | MEMRESIFG18 | R/W | 0h | Raw interrupt status for MEMRES18.<br>This bit is set to 1 when MEMRES18 is loaded with a new conversion result.<br>Reading MEMRES18 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 25 | MEMRESIFG17 | R/W | 0h | Raw interrupt status for MEMRES17.<br>This bit is set to 1 when MEMRES17 is loaded with a new conversion result.<br>Reading MEMRES17 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 24 | MEMRESIFG16 | R/W | 0h | Raw interrupt status for MEMRES16.<br>This bit is set to 1 when MEMRES16 is loaded with a new conversion result.<br>Reading MEMRES16 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 23 | MEMRESIFG15 | R/W | 0h | Raw interrupt status for MEMRES15.<br>This bit is set to 1 when MEMRES15 is loaded with a new conversion result.<br>Reading MEMRES15 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 22 | MEMRESIFG14 | R/W | 0h | Raw interrupt status for MEMRES14.<br>This bit is set to 1 when MEMRES14 is loaded with a new conversion result.<br>Reading MEMRES14 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 21 | MEMRESIFG13 | R/W | 0h | Raw interrupt status for MEMRES13.<br>This bit is set to 1 when MEMRES13 is loaded with a new conversion result.<br>Reading MEMRES13 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 20 | MEMRESIFG12 | R/W | 0h | Raw interrupt status for MEMRES12.<br>This bit is set to 1 when MEMRES12 is loaded with a new conversion result.<br>Reading MEMRES12 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |



Copyright © 2026 Texas Instruments Incorporated

**Table 18-31. IMASK Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 19 | MEMRESIFG11 | R/W | 0h | Raw interrupt status for MEMRES11.<br>This bit is set to 1 when MEMRES11 is loaded with a new conversion result.<br>Reading MEMRES11 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 18 | MEMRESIFG10 | R/W | 0h | Raw interrupt status for MEMRES10.<br>This bit is set to 1 when MEMRES10 is loaded with a new conversion result.<br>Reading MEMRES10 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 17 | MEMRESIFG9 | R/W | 0h | Raw interrupt status for MEMRES9.<br>This bit is set to 1 when MEMRES9 is loaded with a new conversion result.<br>Reading MEMRES9 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 16 | MEMRESIFG8 | R/W | 0h | Raw interrupt status for MEMRES8.<br>This bit is set to 1 when MEMRES8 is loaded with a new conversion result.<br>Reading MEMRES8 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 15 | MEMRESIFG7 | R/W | 0h | Raw interrupt status for MEMRES7.<br>This bit is set to 1 when MEMRES7 is loaded with a new conversion result.<br>Reading MEMRES7 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 14 | MEMRESIFG6 | R/W | 0h | Raw interrupt status for MEMRES6.<br>This bit is set to 1 when MEMRES6 is loaded with a new conversion result.<br>Reading MEMRES6 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 13 | MEMRESIFG5 | R/W | 0h | Raw interrupt status for MEMRES5.<br>This bit is set to 1 when MEMRES5 is loaded with a new conversion result.<br>Reading MEMRES5 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 12 | MEMRESIFG4 | R/W | 0h | Raw interrupt status for MEMRES4.<br>This bit is set to 1 when MEMRES4 is loaded with a new conversion result.<br>Reading MEMRES4 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |



Copyright © 2026 Texas Instruments Incorporated

**Table 18-31. IMASK Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 11 | MEMRESIFG3 | R/W | 0h | Raw interrupt status for MEMRES3.<br>This bit is set to 1 when MEMRES3 is loaded with a new conversion result.<br>Reading MEMRES3 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 10 | MEMRESIFG2 | R/W | 0h | Raw interrupt status for MEMRES2.<br>This bit is set to 1 when MEMRES2 is loaded with a new conversion result.<br>Reading MEMRES2 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 9 | MEMRESIFG1 | R/W | 0h | Raw interrupt status for MEMRES1.<br>This bit is set to 1 when MEMRES1 is loaded with a new conversion result.<br>Reading MEMRES1 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 8 | MEMRESIFG0 | R/W | 0h | Raw interrupt status for MEMRES0.<br>This bit is set to 1 when MEMRES0 is loaded with a new conversion result.<br>Reading MEMRES0 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 7-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 18.3.21 RIS Register (Offset = 1090h) [Reset = 00000000h]

RIS is shown in Figure 18-25 and described in Table 18-32.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Figure 18-25. RIS Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MEMRESIFG23 | MEMRESIFG22 | MEMRESIFG21 | MEMRESIFG20 | MEMRESIFG19 | MEMRESIFG18 | MEMRESIFG17 | MEMRESIFG16 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MEMRESIFG15 | MEMRESIFG14 | MEMRESIFG13 | MEMRESIFG12 | MEMRESIFG11 | MEMRESIFG10 | MEMRESIFG9 | MEMRESIFG8 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MEMRESIFG7 | MEMRESIFG6 | MEMRESIFG5 | MEMRESIFG4 | MEMRESIFG3 | MEMRESIFG2 | MEMRESIFG1 | MEMRESIFG0 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |

**Table 18-32. RIS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | MEMRESIFG23 | R | 0h | Raw interrupt status for MEMRES23.<br>This bit is set to 1 when MEMRES23 is loaded with a new conversion result.<br>Reading MEMRES23 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 30 | MEMRESIFG22 | R | 0h | Raw interrupt status for MEMRES22.<br>This bit is set to 1 when MEMRES22 is loaded with a new conversion result.<br>Reading MEMRES22 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 29 | MEMRESIFG21 | R | 0h | Raw interrupt status for MEMRES21.<br>This bit is set to 1 when MEMRES21 is loaded with a new conversion result.<br>Reading MEMRES21 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 28 | MEMRESIFG20 | R | 0h | Raw interrupt status for MEMRES20.<br>This bit is set to 1 when MEMRES20 is loaded with a new conversion result.<br>Reading MEMRES20 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |



Copyright © 2026 Texas Instruments Incorporated

**Table 18-32. RIS Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 27 | MEMRESIFG19 | R | 0h | Raw interrupt status for MEMRES19.<br>This bit is set to 1 when MEMRES19 is loaded with a new conversion result.<br>Reading MEMRES19 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 26 | MEMRESIFG18 | R | 0h | Raw interrupt status for MEMRES18.<br>This bit is set to 1 when MEMRES18 is loaded with a new conversion result.<br>Reading MEMRES18 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 25 | MEMRESIFG17 | R | 0h | Raw interrupt status for MEMRES17.<br>This bit is set to 1 when MEMRES17 is loaded with a new conversion result.<br>Reading MEMRES17 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 24 | MEMRESIFG16 | R | 0h | Raw interrupt status for MEMRES16.<br>This bit is set to 1 when MEMRES16 is loaded with a new conversion result.<br>Reading MEMRES16 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 23 | MEMRESIFG15 | R | 0h | Raw interrupt status for MEMRES15.<br>This bit is set to 1 when MEMRES15 is loaded with a new conversion result.<br>Reading MEMRES15 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 22 | MEMRESIFG14 | R | 0h | Raw interrupt status for MEMRES14.<br>This bit is set to 1 when MEMRES14 is loaded with a new conversion result.<br>Reading MEMRES14 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 21 | MEMRESIFG13 | R | 0h | Raw interrupt status for MEMRES13.<br>This bit is set to 1 when MEMRES13 is loaded with a new conversion result.<br>Reading MEMRES13 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 20 | MEMRESIFG12 | R | 0h | Raw interrupt status for MEMRES12.<br>This bit is set to 1 when MEMRES12 is loaded with a new conversion result.<br>Reading MEMRES12 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |



Copyright © 2026 Texas Instruments Incorporated

**Table 18-32. RIS Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 19 | MEMRESIFG11 | R | 0h | Raw interrupt status for MEMRES11.<br>This bit is set to 1 when MEMRES11 is loaded with a new conversion result.<br>Reading MEMRES11 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 18 | MEMRESIFG10 | R | 0h | Raw interrupt status for MEMRES10.<br>This bit is set to 1 when MEMRES10 is loaded with a new conversion result.<br>Reading MEMRES10 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 17 | MEMRESIFG9 | R | 0h | Raw interrupt status for MEMRES9.<br>This bit is set to 1 when MEMRES9 is loaded with a new conversion result.<br>Reading MEMRES9 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 16 | MEMRESIFG8 | R | 0h | Raw interrupt status for MEMRES8.<br>This bit is set to 1 when MEMRES8 is loaded with a new conversion result.<br>Reading MEMRES8 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 15 | MEMRESIFG7 | R | 0h | Raw interrupt status for MEMRES7.<br>This bit is set to 1 when MEMRES7 is loaded with a new conversion result.<br>Reading MEMRES7 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 14 | MEMRESIFG6 | R | 0h | Raw interrupt status for MEMRES6.<br>This bit is set to 1 when MEMRES6 is loaded with a new conversion result.<br>Reading MEMRES6 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 13 | MEMRESIFG5 | R | 0h | Raw interrupt status for MEMRES5.<br>This bit is set to 1 when MEMRES5 is loaded with a new conversion result.<br>Reading MEMRES5 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 12 | MEMRESIFG4 | R | 0h | Raw interrupt status for MEMRES4.<br>This bit is set to 1 when MEMRES4 is loaded with a new conversion result.<br>Reading MEMRES4 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |



Copyright © 2026 Texas Instruments Incorporated

**Table 18-32. RIS Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 11 | MEMRESIFG3 | R | 0h | Raw interrupt status for MEMRES3.<br>This bit is set to 1 when MEMRES3 is loaded with a new conversion result.<br>Reading MEMRES3 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 10 | MEMRESIFG2 | R | 0h | Raw interrupt status for MEMRES2.<br>This bit is set to 1 when MEMRES2 is loaded with a new conversion result.<br>Reading MEMRES2 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 9 | MEMRESIFG1 | R | 0h | Raw interrupt status for MEMRES1.<br>This bit is set to 1 when MEMRES1 is loaded with a new conversion result.<br>Reading MEMRES1 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 8 | MEMRESIFG0 | R | 0h | Raw interrupt status for MEMRES0.<br>This bit is set to 1 when MEMRES0 is loaded with a new conversion result.<br>Reading MEMRES0 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 7-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 18.3.22 MIS Register (Offset = 1098h) [Reset = 00000000h]

MIS is shown in Figure 18-26 and described in Table 18-33.

Return to the Summary Table.

Extension of Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 18-26. MIS Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MEMRESIFG23 | MEMRESIFG22 | MEMRESIFG21 | MEMRESIFG20 | MEMRESIFG19 | MEMRESIFG18 | MEMRESIFG17 | MEMRESIFG16 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MEMRESIFG15 | MEMRESIFG14 | MEMRESIFG13 | MEMRESIFG12 | MEMRESIFG11 | MEMRESIFG10 | MEMRESIFG9 | MEMRESIFG8 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MEMRESIFG7 | MEMRESIFG6 | MEMRESIFG5 | MEMRESIFG4 | MEMRESIFG3 | MEMRESIFG2 | MEMRESIFG1 | MEMRESIFG0 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |

**Table 18-33. MIS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | MEMRESIFG23 | R | 0h | Raw interrupt status for MEMRES23.<br>This bit is set to 1 when MEMRES23 is loaded with a new conversion result.<br>Reading MEMRES23 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 30 | MEMRESIFG22 | R | 0h | Raw interrupt status for MEMRES22.<br>This bit is set to 1 when MEMRES22 is loaded with a new conversion result.<br>Reading MEMRES22 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 29 | MEMRESIFG21 | R | 0h | Raw interrupt status for MEMRES21.<br>This bit is set to 1 when MEMRES21 is loaded with a new conversion result.<br>Reading MEMRES21 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 28 | MEMRESIFG20 | R | 0h | Raw interrupt status for MEMRES20.<br>This bit is set to 1 when MEMRES20 is loaded with a new conversion result.<br>Reading MEMRES20 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |



Copyright © 2026 Texas Instruments Incorporated

**Table 18-33. MIS Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 27 | MEMRESIFG19 | R | 0h | Raw interrupt status for MEMRES19.<br>This bit is set to 1 when MEMRES19 is loaded with a new conversion result.<br>Reading MEMRES19 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 26 | MEMRESIFG18 | R | 0h | Raw interrupt status for MEMRES18.<br>This bit is set to 1 when MEMRES18 is loaded with a new conversion result.<br>Reading MEMRES18 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 25 | MEMRESIFG17 | R | 0h | Raw interrupt status for MEMRES17.<br>This bit is set to 1 when MEMRES17 is loaded with a new conversion result.<br>Reading MEMRES17 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 24 | MEMRESIFG16 | R | 0h | Raw interrupt status for MEMRES16.<br>This bit is set to 1 when MEMRES16 is loaded with a new conversion result.<br>Reading MEMRES16 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 23 | MEMRESIFG15 | R | 0h | Raw interrupt status for MEMRES15.<br>This bit is set to 1 when MEMRES15 is loaded with a new conversion result.<br>Reading MEMRES15 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 22 | MEMRESIFG14 | R | 0h | Raw interrupt status for MEMRES14.<br>This bit is set to 1 when MEMRES14 is loaded with a new conversion result.<br>Reading MEMRES14 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 21 | MEMRESIFG13 | R | 0h | Raw interrupt status for MEMRES13.<br>This bit is set to 1 when MEMRES13 is loaded with a new conversion result.<br>Reading MEMRES13 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 20 | MEMRESIFG12 | R | 0h | Raw interrupt status for MEMRES12.<br>This bit is set to 1 when MEMRES12 is loaded with a new conversion result.<br>Reading MEMRES12 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |



Copyright © 2026 Texas Instruments Incorporated

**Table 18-33. MIS Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 19 | MEMRESIFG11 | R | 0h | Raw interrupt status for MEMRES11.<br>This bit is set to 1 when MEMRES11 is loaded with a new conversion result.<br>Reading MEMRES11 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 18 | MEMRESIFG10 | R | 0h | Raw interrupt status for MEMRES10.<br>This bit is set to 1 when MEMRES10 is loaded with a new conversion result.<br>Reading MEMRES10 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 17 | MEMRESIFG9 | R | 0h | Raw interrupt status for MEMRES9.<br>This bit is set to 1 when MEMRES9 is loaded with a new conversion result.<br>Reading MEMRES9 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 16 | MEMRESIFG8 | R | 0h | Raw interrupt status for MEMRES8.<br>This bit is set to 1 when MEMRES8 is loaded with a new conversion result.<br>Reading MEMRES8 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 15 | MEMRESIFG7 | R | 0h | Raw interrupt status for MEMRES7.<br>This bit is set to 1 when MEMRES7 is loaded with a new conversion result.<br>Reading MEMRES7 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 14 | MEMRESIFG6 | R | 0h | Raw interrupt status for MEMRES6.<br>This bit is set to 1 when MEMRES6 is loaded with a new conversion result.<br>Reading MEMRES6 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 13 | MEMRESIFG5 | R | 0h | Raw interrupt status for MEMRES5.<br>This bit is set to 1 when MEMRES5 is loaded with a new conversion result.<br>Reading MEMRES5 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 12 | MEMRESIFG4 | R | 0h | Raw interrupt status for MEMRES4.<br>This bit is set to 1 when MEMRES4 is loaded with a new conversion result.<br>Reading MEMRES4 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |



Copyright © 2026 Texas Instruments Incorporated

**Table 18-33. MIS Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 11 | MEMRESIFG3 | R | 0h | Raw interrupt status for MEMRES3.<br>This bit is set to 1 when MEMRES3 is loaded with a new conversion result.<br>Reading MEMRES3 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 10 | MEMRESIFG2 | R | 0h | Raw interrupt status for MEMRES2.<br>This bit is set to 1 when MEMRES2 is loaded with a new conversion result.<br>Reading MEMRES2 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 9 | MEMRESIFG1 | R | 0h | Raw interrupt status for MEMRES1.<br>This bit is set to 1 when MEMRES1 is loaded with a new conversion result.<br>Reading MEMRES1 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 8 | MEMRESIFG0 | R | 0h | Raw interrupt status for MEMRES0.<br>This bit is set to 1 when MEMRES0 is loaded with a new conversion result.<br>Reading MEMRES0 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 7-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 18.3.23 ISET Register (Offset = 10A0h) [Reset = 00000000h]

ISET is shown in Figure 18-27 and described in Table 18-34.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 18-27. ISET Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MEMRESIFG23 | MEMRESIFG22 | MEMRESIFG21 | MEMRESIFG20 | MEMRESIFG19 | MEMRESIFG18 | MEMRESIFG17 | MEMRESIFG16 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MEMRESIFG15 | MEMRESIFG14 | MEMRESIFG13 | MEMRESIFG12 | MEMRESIFG11 | MEMRESIFG10 | MEMRESIFG9 | MEMRESIFG8 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MEMRESIFG7 | MEMRESIFG6 | MEMRESIFG5 | MEMRESIFG4 | MEMRESIFG3 | MEMRESIFG2 | MEMRESIFG1 | MEMRESIFG0 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |

**Table 18-34. ISET Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | MEMRESIFG23 | W | 0h | Raw interrupt status for MEMRES23.<br>This bit is set to 1 when MEMRES23 is loaded with a new conversion result.<br>Reading MEMRES23 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 30 | MEMRESIFG22 | W | 0h | Raw interrupt status for MEMRES22.<br>This bit is set to 1 when MEMRES22 is loaded with a new conversion result.<br>Reading MEMRES22 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 29 | MEMRESIFG21 | W | 0h | Raw interrupt status for MEMRES21.<br>This bit is set to 1 when MEMRES21 is loaded with a new conversion result.<br>Reading MEMRES21 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 28 | MEMRESIFG20 | W | 0h | Raw interrupt status for MEMRES20.<br>This bit is set to 1 when MEMRES20 is loaded with a new conversion result.<br>Reading MEMRES20 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |



Copyright © 2026 Texas Instruments Incorporated

**Table 18-34. ISET Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 27 | MEMRESIFG19 | W | 0h | Raw interrupt status for MEMRES19.<br>This bit is set to 1 when MEMRES19 is loaded with a new conversion result.<br>Reading MEMRES19 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 26 | MEMRESIFG18 | W | 0h | Raw interrupt status for MEMRES18.<br>This bit is set to 1 when MEMRES18 is loaded with a new conversion result.<br>Reading MEMRES18 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 25 | MEMRESIFG17 | W | 0h | Raw interrupt status for MEMRES17.<br>This bit is set to 1 when MEMRES17 is loaded with a new conversion result.<br>Reading MEMRES17 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 24 | MEMRESIFG16 | W | 0h | Raw interrupt status for MEMRES16.<br>This bit is set to 1 when MEMRES16 is loaded with a new conversion result.<br>Reading MEMRES16 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 23 | MEMRESIFG15 | W | 0h | Raw interrupt status for MEMRES15.<br>This bit is set to 1 when MEMRES15 is loaded with a new conversion result.<br>Reading MEMRES15 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 22 | MEMRESIFG14 | W | 0h | Raw interrupt status for MEMRES14.<br>This bit is set to 1 when MEMRES14 is loaded with a new conversion result.<br>Reading MEMRES14 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 21 | MEMRESIFG13 | W | 0h | Raw interrupt status for MEMRES13.<br>This bit is set to 1 when MEMRES13 is loaded with a new conversion result.<br>Reading MEMRES13 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 20 | MEMRESIFG12 | W | 0h | Raw interrupt status for MEMRES12.<br>This bit is set to 1 when MEMRES12 is loaded with a new conversion result.<br>Reading MEMRES12 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |



Copyright © 2026 Texas Instruments Incorporated

**Table 18-34. ISET Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 19 | MEMRESIFG11 | W | 0h | Raw interrupt status for MEMRES11.<br>This bit is set to 1 when MEMRES11 is loaded with a new conversion result.<br>Reading MEMRES11 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 18 | MEMRESIFG10 | W | 0h | Raw interrupt status for MEMRES10.<br>This bit is set to 1 when MEMRES10 is loaded with a new conversion result.<br>Reading MEMRES10 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 17 | MEMRESIFG9 | W | 0h | Raw interrupt status for MEMRES9.<br>This bit is set to 1 when MEMRES9 is loaded with a new conversion result.<br>Reading MEMRES9 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 16 | MEMRESIFG8 | W | 0h | Raw interrupt status for MEMRES8.<br>This bit is set to 1 when MEMRES8 is loaded with a new conversion result.<br>Reading MEMRES8 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 15 | MEMRESIFG7 | W | 0h | Raw interrupt status for MEMRES7.<br>This bit is set to 1 when MEMRES7 is loaded with a new conversion result.<br>Reading MEMRES7 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 14 | MEMRESIFG6 | W | 0h | Raw interrupt status for MEMRES6.<br>This bit is set to 1 when MEMRES6 is loaded with a new conversion result.<br>Reading MEMRES6 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 13 | MEMRESIFG5 | W | 0h | Raw interrupt status for MEMRES5.<br>This bit is set to 1 when MEMRES5 is loaded with a new conversion result.<br>Reading MEMRES5 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 12 | MEMRESIFG4 | W | 0h | Raw interrupt status for MEMRES4.<br>This bit is set to 1 when MEMRES4 is loaded with a new conversion result.<br>Reading MEMRES4 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |



Copyright © 2026 Texas Instruments Incorporated

**Table 18-34. ISET Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 11 | MEMRESIFG3 | W | 0h | Raw interrupt status for MEMRES3.<br>This bit is set to 1 when MEMRES3 is loaded with a new conversion result.<br>Reading MEMRES3 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 10 | MEMRESIFG2 | W | 0h | Raw interrupt status for MEMRES2.<br>This bit is set to 1 when MEMRES2 is loaded with a new conversion result.<br>Reading MEMRES2 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 9 | MEMRESIFG1 | W | 0h | Raw interrupt status for MEMRES1.<br>This bit is set to 1 when MEMRES1 is loaded with a new conversion result.<br>Reading MEMRES1 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 8 | MEMRESIFG0 | W | 0h | Raw interrupt status for MEMRES0.<br>This bit is set to 1 when MEMRES0 is loaded with a new conversion result.<br>Reading MEMRES0 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 7-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 18.3.24 ICLR Register (Offset = 10A8h) [Reset = 00000000h]

ICLR is shown in Figure 18-28 and described in Table 18-35.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

**Figure 18-28. ICLR Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MEMRESIFG23 | MEMRESIFG22 | MEMRESIFG21 | MEMRESIFG20 | MEMRESIFG19 | MEMRESIFG18 | MEMRESIFG17 | MEMRESIFG16 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MEMRESIFG15 | MEMRESIFG14 | MEMRESIFG13 | MEMRESIFG12 | MEMRESIFG11 | MEMRESIFG10 | MEMRESIFG9 | MEMRESIFG8 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MEMRESIFG7 | MEMRESIFG6 | MEMRESIFG5 | MEMRESIFG4 | MEMRESIFG3 | MEMRESIFG2 | MEMRESIFG1 | MEMRESIFG0 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |

**Table 18-35. ICLR Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | MEMRESIFG23 | W | 0h | Raw interrupt status for MEMRES23.<br>This bit is set to 1 when MEMRES23 is loaded with a new conversion result.<br>Reading MEMRES23 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 30 | MEMRESIFG22 | W | 0h | Raw interrupt status for MEMRES22.<br>This bit is set to 1 when MEMRES22 is loaded with a new conversion result.<br>Reading MEMRES22 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 29 | MEMRESIFG21 | W | 0h | Raw interrupt status for MEMRES21.<br>This bit is set to 1 when MEMRES21 is loaded with a new conversion result.<br>Reading MEMRES21 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 28 | MEMRESIFG20 | W | 0h | Raw interrupt status for MEMRES20.<br>This bit is set to 1 when MEMRES20 is loaded with a new conversion result.<br>Reading MEMRES20 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |



Copyright © 2026 Texas Instruments Incorporated

**Table 18-35. ICLR Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 27 | MEMRESIFG19 | W | 0h | Raw interrupt status for MEMRES19.<br>This bit is set to 1 when MEMRES19 is loaded with a new conversion result.<br>Reading MEMRES19 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 26 | MEMRESIFG18 | W | 0h | Raw interrupt status for MEMRES18.<br>This bit is set to 1 when MEMRES18 is loaded with a new conversion result.<br>Reading MEMRES18 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 25 | MEMRESIFG17 | W | 0h | Raw interrupt status for MEMRES17.<br>This bit is set to 1 when MEMRES17 is loaded with a new conversion result.<br>Reading MEMRES17 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 24 | MEMRESIFG16 | W | 0h | Raw interrupt status for MEMRES16.<br>This bit is set to 1 when MEMRES16 is loaded with a new conversion result.<br>Reading MEMRES16 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 23 | MEMRESIFG15 | W | 0h | Raw interrupt status for MEMRES15.<br>This bit is set to 1 when MEMRES15 is loaded with a new conversion result.<br>Reading MEMRES15 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 22 | MEMRESIFG14 | W | 0h | Raw interrupt status for MEMRES14.<br>This bit is set to 1 when MEMRES14 is loaded with a new conversion result.<br>Reading MEMRES14 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 21 | MEMRESIFG13 | W | 0h | Raw interrupt status for MEMRES13.<br>This bit is set to 1 when MEMRES13 is loaded with a new conversion result.<br>Reading MEMRES13 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 20 | MEMRESIFG12 | W | 0h | Raw interrupt status for MEMRES12.<br>This bit is set to 1 when MEMRES12 is loaded with a new conversion result.<br>Reading MEMRES12 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |



Copyright © 2026 Texas Instruments Incorporated

**Table 18-35. ICLR Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 19 | MEMRESIFG11 | W | 0h | Raw interrupt status for MEMRES11.<br>This bit is set to 1 when MEMRES11 is loaded with a new conversion result.<br>Reading MEMRES11 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 18 | MEMRESIFG10 | W | 0h | Raw interrupt status for MEMRES10.<br>This bit is set to 1 when MEMRES10 is loaded with a new conversion result.<br>Reading MEMRES10 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 17 | MEMRESIFG9 | W | 0h | Raw interrupt status for MEMRES9.<br>This bit is set to 1 when MEMRES9 is loaded with a new conversion result.<br>Reading MEMRES9 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 16 | MEMRESIFG8 | W | 0h | Raw interrupt status for MEMRES8.<br>This bit is set to 1 when MEMRES8 is loaded with a new conversion result.<br>Reading MEMRES8 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 15 | MEMRESIFG7 | W | 0h | Raw interrupt status for MEMRES7.<br>This bit is set to 1 when MEMRES7 is loaded with a new conversion result.<br>Reading MEMRES7 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 14 | MEMRESIFG6 | W | 0h | Raw interrupt status for MEMRES6.<br>This bit is set to 1 when MEMRES6 is loaded with a new conversion result.<br>Reading MEMRES6 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 13 | MEMRESIFG5 | W | 0h | Raw interrupt status for MEMRES5.<br>This bit is set to 1 when MEMRES5 is loaded with a new conversion result.<br>Reading MEMRES5 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 12 | MEMRESIFG4 | W | 0h | Raw interrupt status for MEMRES4.<br>This bit is set to 1 when MEMRES4 is loaded with a new conversion result.<br>Reading MEMRES4 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |



Copyright © 2026 Texas Instruments Incorporated

**Table 18-35. ICLR Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 11 | MEMRESIFG3 | W | 0h | Raw interrupt status for MEMRES3.<br>This bit is set to 1 when MEMRES3 is loaded with a new conversion result.<br>Reading MEMRES3 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 10 | MEMRESIFG2 | W | 0h | Raw interrupt status for MEMRES2.<br>This bit is set to 1 when MEMRES2 is loaded with a new conversion result.<br>Reading MEMRES2 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 9 | MEMRESIFG1 | W | 0h | Raw interrupt status for MEMRES1.<br>This bit is set to 1 when MEMRES1 is loaded with a new conversion result.<br>Reading MEMRES1 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 8 | MEMRESIFG0 | W | 0h | Raw interrupt status for MEMRES0.<br>This bit is set to 1 when MEMRES0 is loaded with a new conversion result.<br>Reading MEMRES0 register will clear this bit, or when the corresponding bit in ICLR is set to 1<br>0h = No new data ready.<br>1h = A new data is ready to be read. |
| 7-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 18.3.25 EVT_MODE Register (Offset = 10E0h) [Reset = 00000000h]

EVT_MODE is shown in Figure 18-29 and described in Table 18-36.

Return to the Summary Table.

Event mode register. It is used to select whether each line is disabled, in software mode (software clears the RIS) or in hardware mode (hardware clears the RIS)

**Figure 18-29. EVT_MODE Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | EVT1_CFG | | INT0_CFG | |
| R-0h | | | | R-0h | | R-0h | |

**Table 18-36. EVT_MODE Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3-2 | EVT1_CFG | R | 2h | Event line mode select for event corresponding to GEN_EVENT<br>0h = The interrupt or event line is disabled.<br>1h = The interrupt or event line is in software mode. Software must clear the RIS.<br>2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag. |
| 1-0 | INT0_CFG | R | 1h | Event line mode select for event corresponding to CPU_INT<br>0h = The interrupt or event line is disabled.<br>1h = The interrupt or event line is in software mode. Software must clear the RIS.<br>2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag. |



Copyright © 2026 Texas Instruments Incorporated

### 18.3.26 DESC Register (Offset = 10FCh) [Reset = 00000000h]

DESC is shown in Figure 18-30 and described in Table 18-37.

Return to the Summary Table.

This register identifies the peripheral and its exact version.

**Figure 18-30. DESC Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MODULEID | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| FEATUREVER | | | | INSTNUM | | | | MAJREV | | | | MINREV | | | |
| R-0h | | | | R-0h | | | | R-0h | | | | R-0h | | | |

**Table 18-37. DESC Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | MODULEID | R | 2611h | Module identification contains a unique peripheral identification number. The assignments are maintained in a central database for all of the platform modules to ensure uniqueness.<br>0h = Smallest value<br>FFFFh = Highest possible value |
| 15-12 | FEATUREVER | R | 0h | Feature Set for the module *instance*<br>0h = Smallest value<br>Fh = Highest possible value |
| 11-8 | INSTNUM | R | 0h | Instance Number within the device. This will be a parameter to the RTL for modules that can have multiple instances |
| 7-4 | MAJREV | R | 1h | Major rev of the IP<br>0h = Smallest value<br>Fh = Highest possible value |
| 3-0 | MINREV | R | 0h | Minor rev of the IP<br>0h = Smallest value<br>Fh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 18.3.27 CTL0 Register (Offset = 1100h) [Reset = 00000000h]

CTL0 is shown in Figure 18-31 and described in Table 18-38.

Return to the Summary Table.

Control Register 0

**Figure 18-31. CTL0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | SCLKDIV | | |
| R-0h | | | | | R/W-0h | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | PWRDN |
| R-0h | | | | | | | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | ENC |
| R-0h | | | | | | | RH/W-0h |

**Table 18-38. CTL0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | RESERVED | R | 0h | |
| 26-24 | SCLKDIV | R/W | 0h | Sample clock divider<br>0h = Do not divide clock source<br>1h = Divide clock source by 2<br>2h = Divide clock source by 4<br>3h = Divide clock source by 8<br>4h = Divide clock source by 16<br>5h = Divide clock source by 24<br>6h = Divide clock source by 32<br>7h = Divide clock source by 48 |
| 23-17 | RESERVED | R | 0h | |
| 16 | PWRDN | R/W | 0h | Power down policy<br>0h = ADC is powered down on completion of a conversion if there is no pending trigger<br>1h = ADC remains powered on as long as it is enabled through software. |
| 15-1 | RESERVED | R | 0h | |
| 0 | ENC | RH/W | 0h | Enable conversion<br>0h = Conversion disabled. ENC change from ON to OFF will abort single or repeat sequence on a MEMCTLx boundary. The current conversion will finish and result stored in corresponding MEMRESx.<br>1h = Conversion enabled. ADC sequencer waits for valid trigger (software or hardware). |



Copyright © 2026 Texas Instruments Incorporated

### 18.3.28 CTL1 Register (Offset = 1104h) [Reset = 00000000h]

CTL1 is shown in Figure 18-32 and described in Table 18-39.

Return to the Summary Table.

Control Register 1

**Figure 18-32. CTL1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | AVGD | | | RESERVED | AVGN | | |
| R-0h | R/W-0h | | | R-0h | R/W-0h | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | SAMPMODE | RESERVED | | CONSEQ | |
| R-0h | | | R/W-0h | R-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | SC |
| R-0h | | | | | | | RH/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | TRIGSRC |
| R-0h | | | | | | | R/W-0h |

**Table 18-39. CTL1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | RESERVED | R | 0h | |
| 30-28 | AVGD | R/W | 0h | Hardware averager denominator. The number to divide the accumulated value by (this is a shift). Note result register is maximum of 16-bits long so if not shifted appropriately result will be truncated.<br>0h (R/W) = No shift<br>1h (R/W) = 1 bit shift<br>2h (R/W) = 2 bit shift<br>3h (R/W) = 3 bit shift<br>4h (R/W) = 4 bit shift<br>5h (R/W) = 5 bit shift<br>6h (R/W) = 6 bit shift<br>7h (R/W) = 7 bit shift |
| 27 | RESERVED | R | 0h | |
| 26-24 | AVGN | R/W | 0h | Hardware averager numerator. Selects number of conversions to accumulate for current MEMCTLx and then it is divided by AVGD. Result will be stored in MEMRESx.<br>0h (R/W) = Disables averager<br>1h (R/W) = Averages 2 conversions before storing in MEMRESx register<br>2h (R/W) = Averages 4 conversions before storing in MEMRESx register<br>3h (R/W) = Averages 8 conversions before storing in MEMRESx register<br>4h (R/W) = Averages 16 conversions before storing in MEMRESx register<br>5h (R/W) = Averages 32 conversions before storing in MEMRESx register<br>6h (R/W) = Averages 64 conversions before storing in MEMRESx register<br>7h (R/W) = Averages 128 conversions before storing in MEMRESx register |
| 23-21 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

**Table 18-39. CTL1 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 20 | SAMPMODE | R/W | 0h | Sample mode. This bit selects the source of the sampling signal. MANUAL option is not valid when TRIGSRC is selected as hardware event trigger.<br>0h = Sample timer high phase is used as sample signal<br>1h = Software trigger is used as sample signal |
| 19-18 | RESERVED | R | 0h | |
| 17-16 | CONSEQ | R/W | 0h | Conversion sequence mode<br>0h = ADC channel in MEMCTLx pointed by STARTADD will be converted once<br>1h = ADC channel sequence pointed by STARTADD and ENDADD will be converted once<br>2h = ADC channel in MEMCTLx pointed by STARTADD will be converted repeatedly<br>3h = ADC channel sequence pointed by STARTADD and ENDADD will be converted repeatedly |
| 15-9 | RESERVED | R | 0h | |
| 8 | SC | RH/W | 0h | Start of conversion<br>0h = When SAMPMODE is set to MANUAL, clearing this bit will end the sample phase and the conversion phase will start.<br>When SAMPMODE is set to AUTO, writing 0 has no effect.<br>1h = When SAMPMODE is set to MANUAL, setting this bit will start the sample phase. Sample phase will last as long as this bit is set.<br>When SAMPMODE is set to AUTO, setting this bit will trigger the timer based sample time. |
| 7-1 | RESERVED | R | 0h | |
| 0 | TRIGSRC | R/W | 0h | Sample trigger source<br>0h = Software trigger<br>1h = Hardware event trigger |



Copyright © 2026 Texas Instruments Incorporated

### 18.3.29 CTL2 Register (Offset = 1108h) [Reset = 00000000h]

CTL2 is shown in Figure 18-33 and described in Table 18-40.

Return to the Summary Table.

Control Register 2

**Figure 18-33. CTL2 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | ENDADD | | | | |
| R-0h | | | R/W-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | STARTADD | | | | |
| R-0h | | | R/W-0h | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| SAMPCNT | | | | | FIFOEN | RESERVED | DMAEN |
| R/W-0h | | | | | R/W-0h | R-0h | RH/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | RSTSAMPCAPEN | RESERVED | RES | | DF |
| R-0h | | | R/W-0h | R-0h | R/W-0h | | R/W-0h |

**Table 18-40. CTL2 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | RESERVED | R | 0h | |
| 28-24 | ENDADD | R/W | 0h | Sequence end address. These bits select which MEMCTLx is the last one for the sequence mode.<br>The value of ENDADD is 0x00 to 0x17, corresponding to MEMRES0 to MEMRES23.<br>00h = MEMCTL0 is selected as end address of sequence.<br>01h = MEMCTL1 is selected as end address of sequence.<br>02h = MEMCTL2 is selected as end address of sequence.<br>03h = MEMCTL3 is selected as end address of sequence.<br>04h = MEMCTL4 is selected as end address of sequence.<br>05h = MEMCTL5 is selected as end address of sequence.<br>06h = MEMCTL6 is selected as end address of sequence.<br>07h = MEMCTL7 is selected as end address of sequence.<br>08h = MEMCTL8 is selected as end address of sequence.<br>09h = MEMCTL9 is selected as end address of sequence.<br>0Ah = MEMCTL10 is selected as end address of sequence.<br>0Bh = MEMCTL11 is selected as end address of sequence.<br>0Ch = MEMCTL12 is selected as end address of sequence.<br>0Dh = MEMCTL13 is selected as end address of sequence.<br>0Eh = MEMCTL14 is selected as end address of sequence.<br>0Fh = MEMCTL15 is selected as end address of sequence.<br>10h = MEMCTL16 is selected as end address of sequence.<br>11h = MEMCTL17 is selected as end address of sequence.<br>12h = MEMCTL18 is selected as end address of sequence.<br>13h = MEMCTL19 is selected as end address of sequence.<br>14h = MEMCTL20 is selected as end address of sequence.<br>15h = MEMCTL21 is selected as end address of sequence.<br>16h = MEMCTL22 is selected as end address of sequence.<br>17h = MEMCTL23 is selected as end address of sequence. |
| 23-21 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

**Table 18-40. CTL2 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 20-16 | STARTADD | R/W | 0h | Sequencer start address. These bits select which MEMCTLx is used for single conversion or as first MEMCTL for sequence mode.<br>The value of STARTADD is 0x00 to 0x17, corresponding to MEMRES0 to MEMRES23.<br>00h = MEMCTL0 is selected as start address of a sequence or for a single conversion.<br>01h = MEMCTL1 is selected as start address of a sequence or for a single conversion.<br>02h = MEMCTL2 is selected as start address of a sequence or for a single conversion.<br>03h = MEMCTL3 is selected as start address of a sequence or for a single conversion.<br>04h = MEMCTL4 is selected as start address of a sequence or for a single conversion.<br>05h = MEMCTL5 is selected as start address of a sequence or for a single conversion.<br>06h = MEMCTL6 is selected as start address of a sequence or for a single conversion.<br>07h = MEMCTL7 is selected as start address of a sequence or for a single conversion.<br>08h = MEMCTL8 is selected as start address of a sequence or for a single conversion.<br>09h = MEMCTL9 is selected as start address of a sequence or for a single conversion.<br>0Ah = MEMCTL10 is selected as start address of a sequence or for a single conversion.<br>0Bh = MEMCTL11 is selected as start address of a sequence or for a single conversion.<br>0Ch = MEMCTL12 is selected as start address of a sequence or for a single conversion.<br>0Dh = MEMCTL13 is selected as start address of a sequence or for a single conversion.<br>0Eh = MEMCTL14 is selected as start address of a sequence or for a single conversion.<br>0Fh = MEMCTL15 is selected as start address of a sequence or for a single conversion.<br>10h = MEMCTL16 is selected as start address of a sequence or for a single conversion.<br>11h = MEMCTL17 is selected as start address of a sequence or for a single conversion.<br>12h = MEMCTL18 is selected as start address of a sequence or for a single conversion.<br>13h = MEMCTL19 is selected as start address of a sequence or for a single conversion.<br>14h = MEMCTL20 is selected as start address of a sequence or for a single conversion.<br>15h = MEMCTL21 is selected as start address of a sequence or for a single conversion.<br>16h = MEMCTL22 is selected as start address of a sequence or for a single conversion.<br>17h = MEMCTL23 is selected as start address of a sequence or for a single conversion. |
| 15-11 | SAMPCNT | R/W | 0h | Number of ADC converted samples to be transferred on a DMA trigger<br>0h = Minimum value<br>18h = Maximum value |
| 10 | FIFOEN | R/W | 0h | Enable FIFO based operation<br>0h = Disable<br>1h = Enable |
| 9 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

**Table 18-40. CTL2 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 8 | DMAEN | RH/W | 0h | Enable DMA trigger for data transfer.<br>Note: DMAEN bit is cleared by hardware based on DMA done signal at the end of data transfer. Software has to re-enable DMAEN bit for ADC to generate DMA triggers.<br>0h (R/W) = DMA trigger not enabled<br>1h (R/W) = DMA trigger enabled |
| 7-5 | RESERVED | R | 0h | |
| 4 | RSTSAMPCAPEN | R/W | 0h | 0: Sample and hold capacitor is not explicitly discharged at the end of conversion.<br>1: Sample and hold capacitor is discharged at the end of conversion. This incurs one additional conversion clock cycle.<br>0h = Disable sample capacitor discharge feature.<br>1h = Enable sample capacitor discharge feature. |
| 3 | RESERVED | R | 0h | |
| 2-1 | RES | R/W | 0h | Resolution. These bits define the resolution of ADC conversion result.<br>Note : A value of 3 defaults to 12-bits resolution.<br>0h = 12-bits resolution<br>1h = 10-bits resolution<br>2h = 8-bits resolution |
| 0 | DF | R/W | 0h | Data read-back format. Data is always stored in binary unsigned format.<br>0h = Digital result reads as Binary Unsigned.<br>1h = Digital result reads Signed Binary. (2s complement), left aligned. |



Copyright © 2026 Texas Instruments Incorporated

### 18.3.30 CLKFREQ Register (Offset = 1110h) [Reset = 00000000h]

CLKFREQ is shown in Figure 18-34 and described in Table 18-41.

Return to the Summary Table.

Sampling clock frequency range register.

**Figure 18-34. CLKFREQ Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | | FRANGE | | |
| R-0h | | | | | | | | | | | | | R/W-0h | | |

**Table 18-41. CLKFREQ Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | RESERVED | R | 0h | |
| 2-0 | FRANGE | R/W | 0h | Frequency Range.<br>0h = 1 to 4 MHz<br>1h = >4 to 8 MHz<br>2h = >8 to 16 MHz<br>3h = >16 to 20 MHz<br>4h = >20 to 24 MHz<br>5h = >24 to 32 MHz<br>6h = >32 to 40 MHz<br>7h = >40 to 48 MHz |



Copyright © 2026 Texas Instruments Incorporated

### 18.3.31 SCOMP0 Register (Offset = 1114h) [Reset = 00000000h]

SCOMP0 is shown in Figure 18-35 and described in Table 18-42.

Return to the Summary Table.

Sample time compare 0 register. Specifies the sample time, in number of ADC sample clock cycles. CTL0.ENC must be 0 to write to this register.

**Figure 18-35. SCOMP0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | VAL | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | R/W-0h | | | | | | | | | |

**Table 18-42. SCOMP0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-10 | RESERVED | R | 0h | |
| 9-0 | VAL | R/W | 0h | Specifies the number of sample clocks.<br>When VAL = 0 or 1, number of sample clocks = Sample clock divide value.<br>When VAL > 1, number of sample clocks = VAL x Sample clock divide value.<br>Note: Sample clock divide value is not the value written to SCLKDIV but the actual divide value (SCLKDIV = 2 implies divide value is 4).<br>Example: VAL = 4, SCLKDIV = 3 implies 32 sample clock cycles. |



Copyright © 2026 Texas Instruments Incorporated

### 18.3.32 SCOMP1 Register (Offset = 1118h) [Reset = 00000000h]

SCOMP1 is shown in Figure 18-36 and described in Table 18-43.

Return to the Summary Table.

Sample time compare 1 register. Specifies the sample time, in number of ADC sample clock cycles. CTL0.ENC must be 0 to write to this register.

**Figure 18-36. SCOMP1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | VAL | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | R/W-0h | | | | | | | | | |

**Table 18-43. SCOMP1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-10 | RESERVED | R | 0h | |
| 9-0 | VAL | R/W | 0h | Specifies the number of sample clocks.<br>When VAL = 0 or 1, number of sample clocks = Sample clock divide value.<br>When VAL > 1, number of sample clocks = VAL x Sample clock divide value.<br>Note: Sample clock divide value is not the value written to SCLKDIV but the actual divide value (SCLKDIV = 2 implies divide value is 4).<br>Example: VAL = 4, SCLKDIV = 3 implies 32 sample clock cycles. |



Copyright © 2026 Texas Instruments Incorporated

### 18.3.33 WCLOW Register (Offset = 1148h) [Reset = 00000000h]

WCLOW is shown in Figure 18-37 and described in Table 18-44.

Return to the Summary Table.

Window Comparator Low Threshold Register.
The data format that is used to write and read WCLOW depends on the value of DF bit in CTL2 register.
CTL0.ENC must be 0 to write to this register.
Note: Change in ADC data format or resolution does not reset WCLOW.

**Figure 18-37. WCLOW Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | DATA | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | R/W-0h | | | | | | | | | | | | | | | |

**Table 18-44. WCLOW Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15-0 | DATA | R/W | 0h | If DF = 0, unsigned binary format has to be used.<br>The value based on the resolution has to be right aligned with the MSB on the left.<br>For 10-bits and 8-bits resolution, unused bits have to be 0s.<br>If DF = 1, 2s-complement format has to be used.<br>The value based on the resolution has to be left aligned with the LSB on the right.<br>For 10-bits and 8-bits resolution, unused bits have to be 0s. |



Copyright © 2026 Texas Instruments Incorporated

### 18.3.34 WCHIGH Register (Offset = 1150h) [Reset = 00000000h]

WCHIGH is shown in Figure 18-38 and described in Table 18-45.

Return to the Summary Table.

Window Comparator High Threshold Register.
The data format that is used to write and read WCHIGH depends on the value of DF bit in CTL2 register.
CTL0.ENC must be 0 to write to this register.
Note: Change in ADC data format or resolution does not reset WCHIGH.

**Figure 18-38. WCHIGH Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | DATA | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | R/W-0h | | | | | | | | | | | | | | | |

**Table 18-45. WCHIGH Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15-0 | DATA | R/W | 0h | If DF = 0, unsigned binary format has to be used.<br>The threshold value has to be right aligned, with the MSB on the left.<br>For 10-bits and 8-bits resolution, unused bit have to be 0s.<br>If DF = 1, 2s-complement format has to be used.<br>The value based on the resolution has to be left aligned with the LSB on the right.<br>For 10-bits and 8-bits resolution, unused bit have to be 0s. |



Copyright © 2026 Texas Instruments Incorporated

### 18.3.35 MEMCTL_y Register (Offset = 1180h + formula) [Reset = 00000000h]

MEMCTL_y is shown in Figure 18-39 and described in Table 18-46.

Return to the Summary Table.

Conversion Memory Control Register.
CTL0.ENC must be 0 to write to this register.

Offset = 1180h + (y * 4h); where y = 0h to 17h

**Figure 18-39. MEMCTL_y Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | WINCOMP | RESERVED | | | TRIG |
| R-0h | | | R/W-0h | R-0h | | | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | BCSEN | RESERVED | | | AVGEN |
| R-0h | | | R/W-0h | R-0h | | | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | STIME | RESERVED | VRSEL | | |
| R-0h | | | R/W-0h | R-0h | R/W-0h | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | CHANSEL | | | | |
| R-0h | | | R/W-0h | | | | |

**Table 18-46. MEMCTL_y Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | RESERVED | R | 0h | |
| 28 | WINCOMP | R/W | 0h | Enable window comparator.<br>0h = Disable<br>1h = Enable |
| 27-25 | RESERVED | R | 0h | |
| 24 | TRIG | R/W | 0h | Trigger policy. Indicates if a trigger will be needed to step to the next MEMCTL in the sequence or to perform next conversion in the case of repeat single channel conversions.<br>0h = Next conversion is automatic<br>1h = Next conversion requires a trigger |
| 23-21 | RESERVED | R | 0h | |
| 20 | BCSEN | R/W | 0h | Enable burn out current source.<br>0h = Disable<br>1h = Enable |
| 19-17 | RESERVED | R | 0h | |
| 16 | AVGEN | R/W | 0h | Enable hardware averaging.<br>0h (R/W) = Averaging disabled.<br>1h = Averaging enabled. |
| 15-13 | RESERVED | R | 0h | |
| 12 | STIME | R/W | 0h | Selects the source of sample timer period between SCOMP0 and SCOMP1.<br>0h = Select SCOMP0<br>1h = Select SCOMP1 |
| 11 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

**Table 18-46. MEMCTL_y Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 10-8 | VRSEL | R/W | 0h | Voltage reference selection. VEREFM must be connected to on-board ground when external reference option is selected.<br>Note: Writing value 0x3 defaults to INTREF.<br>0h = Positive reference: VDD reference<br>Negative reference: VSS<br>1h = Positive reference: VREF+ pin. If CTL0.ENABLE0 of the VREF IP equals 1, then the internal reference will be used; otherwise the external reference will be used.<br>Negative reference: VREF- pin<br>2h = Positive reference: Internal reference. Set CTL0.ENABLE0 to 1 in the VREF IP.<br>Negative reference: VSS<br>3h = Positive reference: VDD reference<br>Negative reference: VREF- pin<br>Note: Refer to the device specific datasheet if this feature is supported.<br>4h = Positive reference: Internal reference. Set CTL0.ENABLE0 to 1 in the VREF IP.<br>Negative reference: VREF- pin<br>Note: Refer to the device specific datasheet if this feature is supported. |
| 7-5 | RESERVED | R | 0h | |
| 4-0 | CHANSEL | R/W | 0h | Input channel select.<br>00h = Selects channel 0<br>01h = Selects channel 1<br>02h = Selects channel 2<br>03h = Selects channel 3<br>04h = Selects channel 4<br>05h = Selects channel 5<br>06h = Selects channel 6<br>07h = Selects channel 7<br>08h = Selects channel 8<br>09h = Selects channel 9<br>0Ah = Selects channel 10<br>0Bh = Selects channel 11<br>0Ch = Selects channel 12<br>0Dh = Selects channel 13<br>0Eh = Selects channel 14<br>0Fh = Selects channel 15<br>10h = Selects channel 16<br>11h = Selects channel 17<br>12h = Selects channel 18<br>13h = Selects channel 19<br>14h = Selects channel 20<br>15h = Selects channel 21<br>16h = Selects channel 22<br>17h = Selects channel 23<br>18h = Selects channel 24<br>19h = Selects channel 25<br>1Ah = Selects channel 26<br>1Bh = Selects channel 27<br>1Ch = Selects channel 28<br>1Dh = Selects channel 29<br>1Eh = Selects channel 30<br>1Fh = Selects channel 31 |



Copyright © 2026 Texas Instruments Incorporated

### 18.3.36 STATUS Register (Offset = 1340h) [Reset = 00000000h]

STATUS is shown in Figure 18-40 and described in Table 18-47.

Return to the Summary Table.

Status Register

**Figure 18-40. STATUS Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | REFBUFRDY | BUSY |
| R-0h | | | | | | R-0h | R-0h |

**Table 18-47. STATUS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | |
| 1 | REFBUFRDY | R | 0h | Indicates reference buffer is powered up and ready.<br>0h = Not ready<br>1h = Ready |
| 0 | BUSY | R | 0h | Busy. This bit indicates that an active ADC sample or conversion operation is in progress.<br>0h = No ADC sampling or conversion in progress.<br>1h = ADC sampling or conversion is in progress. |



Copyright © 2026 Texas Instruments Incorporated

Chapter 19
# COMP

The comparator module (COMP) is an analog voltage comparator. This chapter describes the operation of the comparator module.


**19.1 Comparator Overview** ..... 1114
**19.2 Comparator Operation** ..... 1119
**19.3 COMP0 Registers** ..... 1131





Copyright © 2026 Texas Instruments Incorporated

## 19.1 Comparator Overview

**Ultra Low Power 3V Comparator Features:**

- Ultra low power
- Inverting and noninverting terminal input multiplexer
- Inverting and noninverting terminal short and exchange capabilities
- Software-selectable output filter
- Programmable hysteresis
- Use externally capped VREF or VDD as a voltage reference for internal DAC
- DAC can serve as a reference to either comparator terminal
- Configurable 8-bit reference DAC with two input codes
- Output connected to event system
- Window comparator mode
- Interrupt driven measurement system for low-power operation
- Exchange DAC output reference between inverting and noninverting inputs
- Operating DAC in sample and hold mode will reduce DAC power usage

**High Speed 3V Comparator Features:**

- High speed operation
- Inverting and noninverting terminal input multiplexer
- Inverting and noninverting terminal short and exchange capabilities
- Software-selectable output filter
- Programmable hysteresis
- Use externally capped VREF or VDD as a voltage reference for internal DAC
- DAC can serve as a reference to either comparator terminal
- Input of DAC can connect directly to either comparator terminal
- Configurable 8-bit reference DAC with two input codes
- Output connected to event system
- Window comparator mode
- Interrupt driven measurement system for low-power operation
- Exchange DAC output reference between inverting and noninverting inputs

**Ultra Low Power 5V Comparator Features:**

- Ultra low power
- Inverting and noninverting terminal input multiplexer
- Software-selectable output filter
- Programmable hysteresis
- Use externally capped VREF or VDD as a voltage reference for internal DAC
- DAC can serve as a reference to the comparator inverting terminal
- Configurable 8-bit reference DAC with two input codes
- Output connected to event system
- Window comparator mode
- Interrupt driven measurement system for low-power operation

**High Speed 5V Comparator Features:**

- High speed operation
- Inverting and noninverting terminal input multiplexer
- Software-selectable output filter
- Programmable hysteresis
- Use externally capped VREF or VDD as a voltage reference for internal DAC
- DAC can serve as a reference to either comparator terminal
- Input of DAC can connect directly to either comparator terminal
- Configurable 8-bit reference DAC with two input codes
- Output connected to event system



Copyright © 2026 Texas Instruments Incorporated

- Window comparator mode
- Interrupt driven measurement system for low-power operation
- Exchange DAC output reference between inverting and noninverting inputs

Note: Some devices bring the 8-bit reference DAC out to a pin

**Note**

If Comp and DAC pin output both are not being used, set the refsrc to 0 to avoid any unintentional leakage

Comparator Block Diagrams

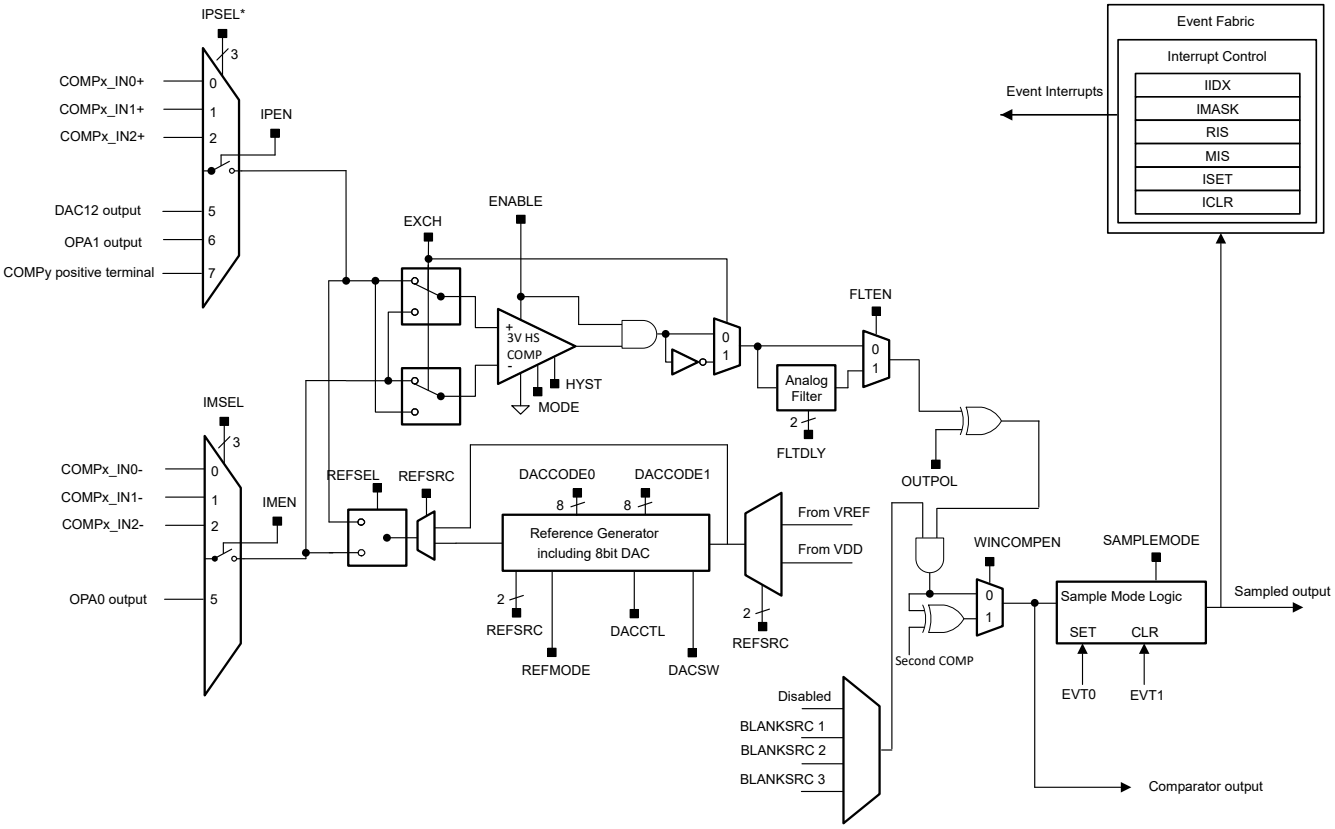



**Figure 19-1. COMP High Speed 3V**



Copyright © 2026 Texas Instruments Incorporated

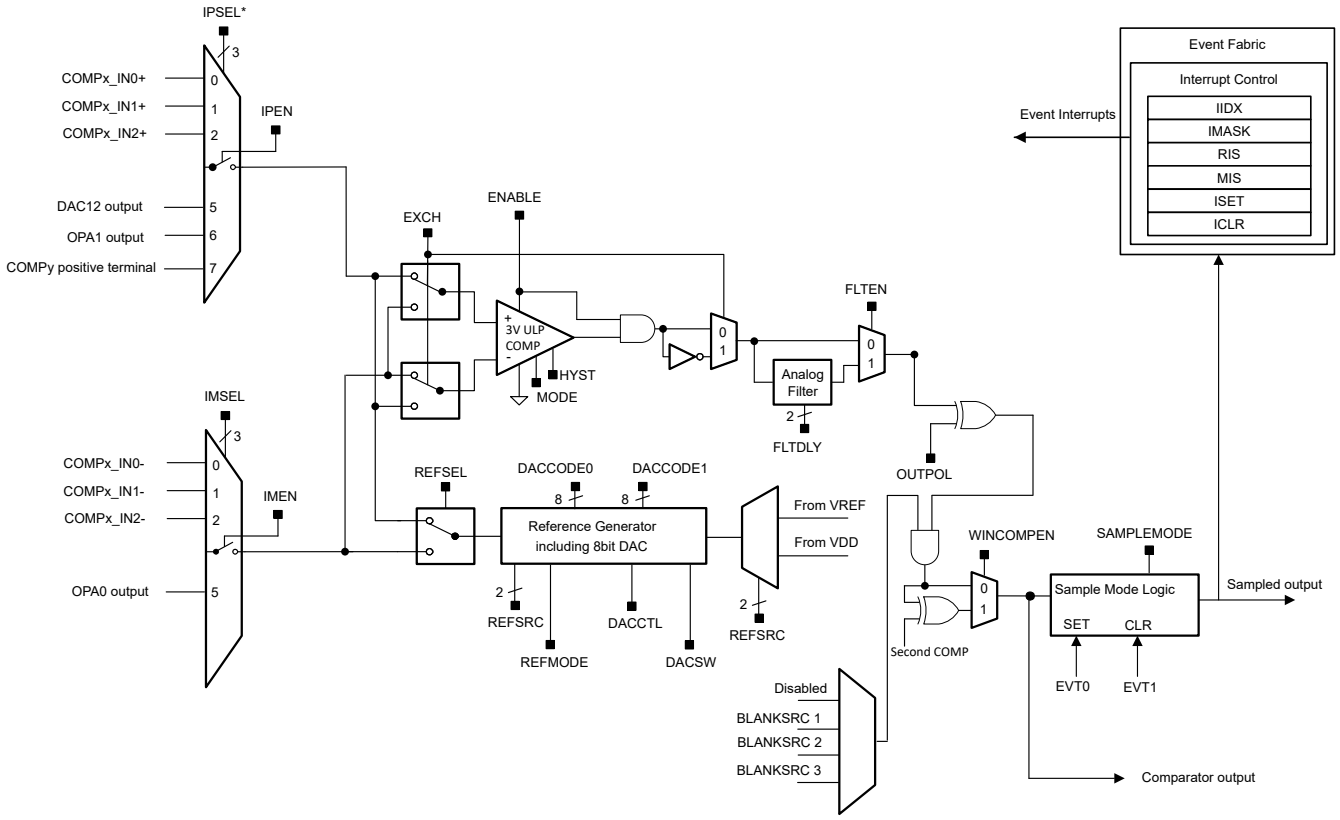



**Figure 19-2. COMP Ultra Low Power 3V**



Copyright © 2026 Texas Instruments Incorporated

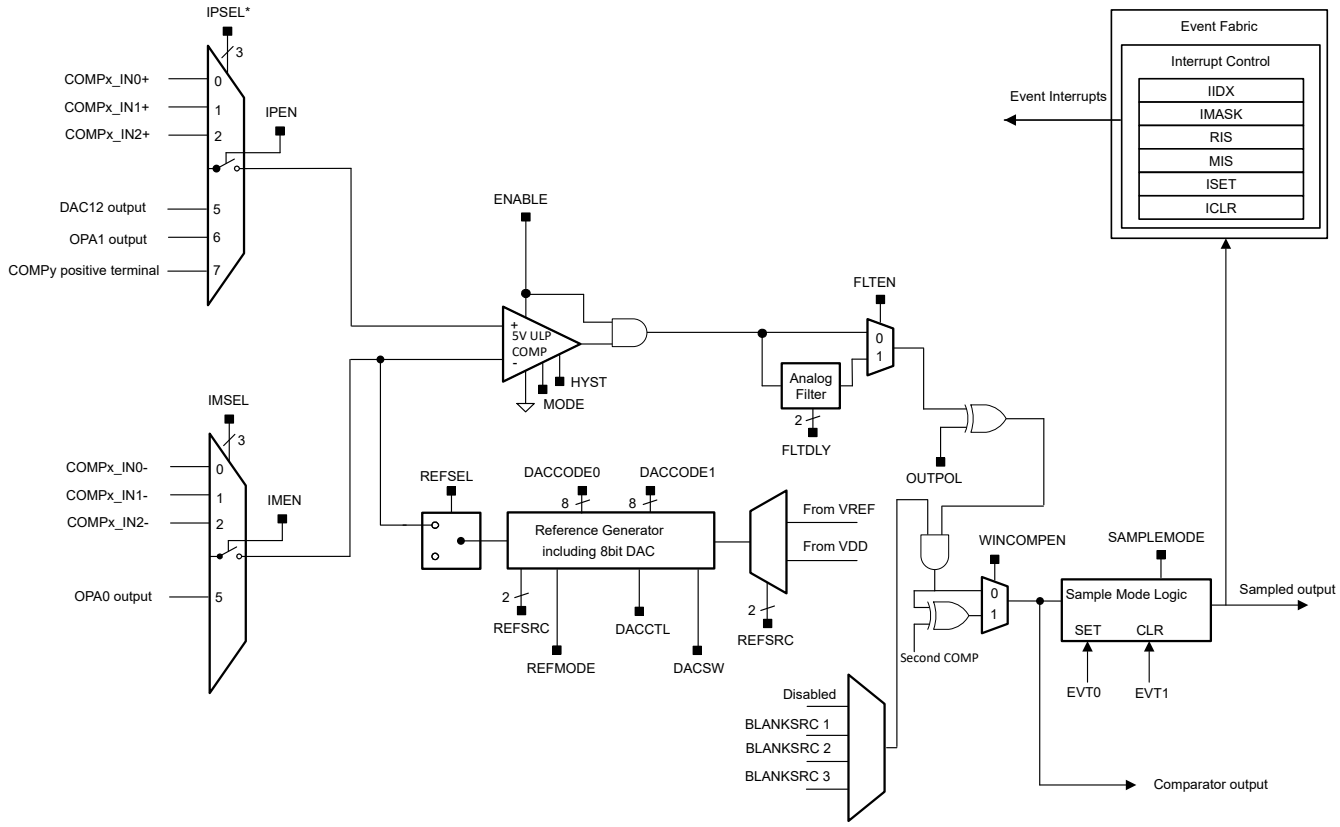



**Figure 19-3. COMP Ultra Low Power 5V**



Copyright © 2026 Texas Instruments Incorporated

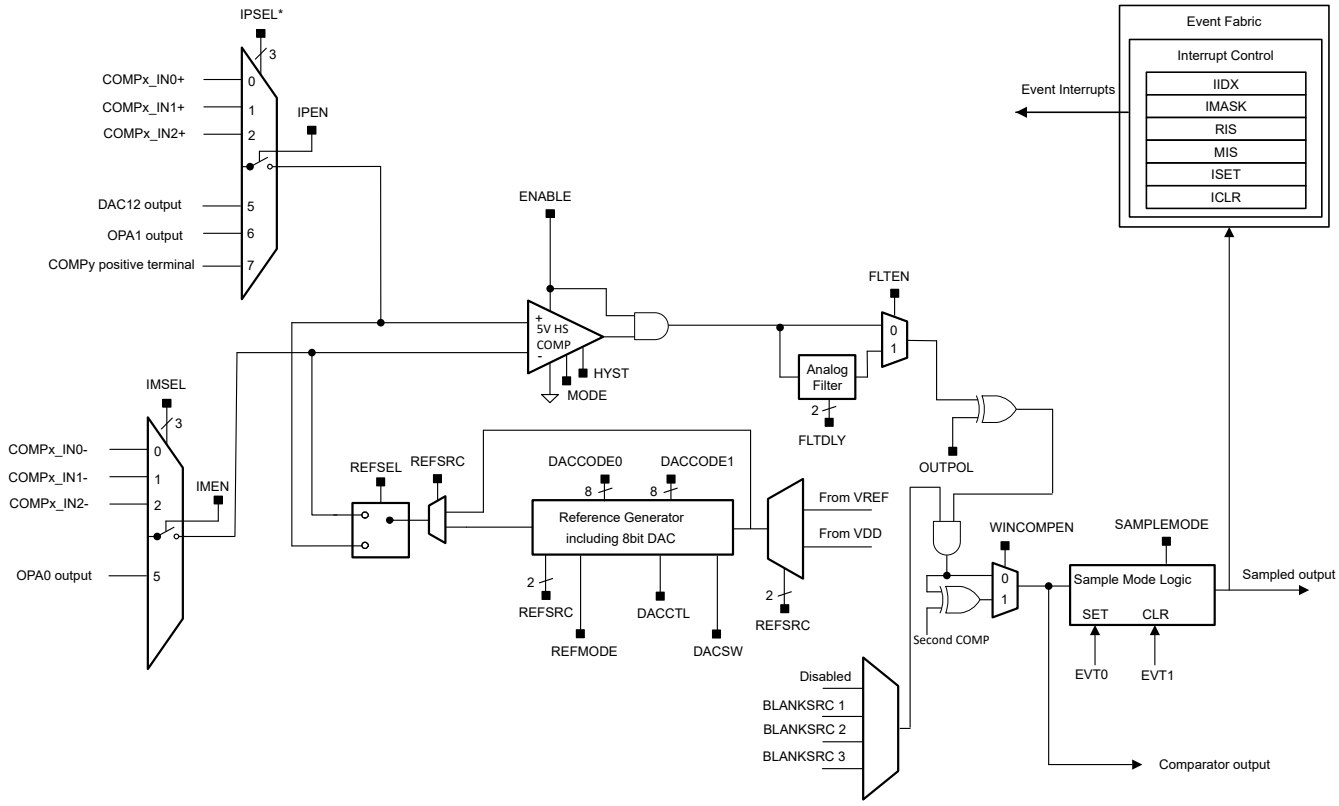



**Figure 19-4. COMP High Speed 5V**



Copyright © 2026 Texas Instruments Incorporated

## 19.2 Comparator Operation

The comparator module is configured by user software. The setup and operation of comparator is discussed in the following sections.

### *19.2.1 Comparator Configuration*

The comparator compares the analog voltages at the positive (+) and negative (–) input terminals. If the + terminal is higher than the – terminal, then the output is high. The polarity of the comparator output can be configured by using the OUTPOL bit in COMPx.CTL1 register.

**Enable Bit**

The comparator can be switched on or off using ENABLE bit in COMPx.CTL1 register. The comparator should be switched off when not in use to reduce current consumption.

**5V COMP Enable Bit Behavior**

COMP output is low when COMP is off

**3V COMP Enable Bit Behavior**

COMP output is high when COMP is off

**Fast vs. Ultra Low Power Mode**

The comparator can be configured in fast or ultra-low-power mode using MODE bit in COMPx.CTL1 register. The default value of MODE bit is 0 (fast mode). The comparator is configured in ultra-low-power mode when MODE is set to 1.

Fast mode consumes higher current, but response time is faster.

Ultra-low power mode consumes very low current, but response time is slower.

*Please see the device specific data sheets for speed and power specifications, so the best mode can be selected for the application.*

**Clock Control**

The clock control for comparator is managed by System Controller (SYSCTL), SYSCTL knows if comparator module is enabled and it also knows if it is in ultra-low-power mode or fast mode. User needs to ensure the proper bus clock is selected for different comparator operation mode:

- Ultra-low-power mode comparator: Bus clock can be LFCLK, or any of the high speed clocks.
- Fast mode comparator: Bus clock cannot be LFCLK. A clock error interrupt is generated in SYSCTL if you enable a comparator in fast mode and the bus clock is LFCLK.

**Using COMP in Low Power Mode**

Suggested configuration sequence:

1. Configure comparator registers and enable COMP when MCLK is SYSOSC
2. Wait for OUTRDY interrupt to be set
3. Move to low power mode or change MCLK to LFCLK

### *19.2.2 Comparator Channels Selection*

The comparator channels on positive terminal is selected through IPSEL bits and enabled through IPEN bit in the register COMPx.CTL0. The comparator channels on negative terminal is selected through IMSEL bits and enabled through IMEN bit. The IPSEL and IMSEL bits can be used to select the comparator channel inputs from device pins or from internal analog modules. The switches in the comparator input channels multiplexer on positive and negative terminals are implemented using break-before-make arrangement to minimize the cross talk when the channel selection changes.

The IPSEL and IMSEL bits allow:



Copyright © 2026 Texas Instruments Incorporated

- Connection of an external signal to the positive and negative terminals of the comparator
- Connection of an internal analog signal (like DAC or OPA) to the positive and negative terminals of the comparator
- Connection of one comparator positive terminal signal to another comparator positive terminal in window compare mode

The EXCH bit in COMPx.CTL1 register controls the input multiplexer, exchange the input signals of the comparator positive and negative terminals. Additionally, when the comparator terminals are exchanged, the output signal from the comparator is inverted too. This allows the user to determine or compensate for the comparator input offset voltage.

**Note**

**Comparator Input Connection**

Please see the device data sheet COMP section for the input channel selection configuration.

When the comparator is on, the input terminals should be connected to a signal, power, or ground. Otherwise, floating levels can cause unexpected interrupts and increased current consumption.

**Comparator Configuration**

The configuration of the comparator must not change while it is in operation except for changes in IPSEL and IMSEL in the register COMPx.CTL0. To change the configuration of the comparator it must be first disabled, reconfigured as needed and then re-enabled.

**COMP Settling**

There can be a glitch on the COMP outpur in certain situations:

- Changing IPSEL or IMSEL
- Changing EXCH
- Enabling COMP

In any of the above situations, wait until the OUTRDY interrupt fires before assuming the output of the COMP is stable.

**Special Cases**

**5V Comparator**

If IPEN == 1 and IMEN == 0 and REFSEL == 0 then the internal reference will automatically switch to the negative side. This is intended to protect any inputs from being shorted with the DAC or VREF output. In this example, IMEN is disabled, so if the user shorts that pin, then nothing will happen.

If IPEN == 0 and IMEN == 1 and REFSEL == 1 then the internal reference will automatically switch to the positive side. This is intended to protect any inputs from being shorted with the DAC or VREF output. In this example, IPEN is disabled, so if the user shorts that pin, then nothing will happen.

**3V Comparator**

If IPEN == 1 and IMEN == 0 and REFSEL == 0 then the internal reference will automatically switch to the negative side using the exchange bits. This is intended to protect any inputs from being shorted with the DAC or VREF output. In this example, IMEN is disabled, so if the user shorts that pin, then nothing will happen.

If IPEN == 0 and IMEN == 1 and REFSEL == 1 then the internal reference will automatically switch to the positive side using the exchange bits. This is intended to protect any inputs from being shorted with the DAC or VREF output. In this example, IPEN is disabled, so if the user shorts that pin, then nothing will happen.



Copyright © 2026 Texas Instruments Incorporated

### 19.2.3 Comparator Output

The comparator output is readable from the OUT bit in COMPx.STAT register. The output is also brought out on device pin. For successful connection of comparator output with the device pin, it is necessary for the software to configure the corresponding device pin in IOMUX module. The comparator output also goes to the Event Interface to generate CPU interrupt event and generic publisher event for internal connects with other modules.

### 19.2.4 Output Filter

The output of the comparator can be used with or without internal filtering. Output filtering can only be enabled while Comparator is in fast mode (COMPx.CTL1.MODE = 0). When FLTEN bit in COMPx.CTL1 register is set, the output is filtered with an on-chip analog filter. The delay of the filter can be adjusted in four different steps using the FLTDLY bits in COMPx.CTL1 register (see Table 19-1).

**Table 19-1. Typical Output Filter Delay Settings**

| FLTDLY bit in COMPx.CTL1 | Typical filter delay |
|---|---|
| 0 | 60ns |
| 1 | 440ns |
| 2 | 1.01us |
| 3 | 2.2us |

Filter delay settings are an addition to propagation delay outlined in device specific data sheet.

The comparator output oscillates if the voltage difference across the input terminals is small (see Figure 19-5 ). Internal and external parasitic effects and cross coupling on and between signal lines, power supply lines, and other parts of the system result in this behavior. The comparator output oscillation reduces the accuracy and resolution of the comparison result. Selecting the output filter can reduce errors associated with comparator oscillation.

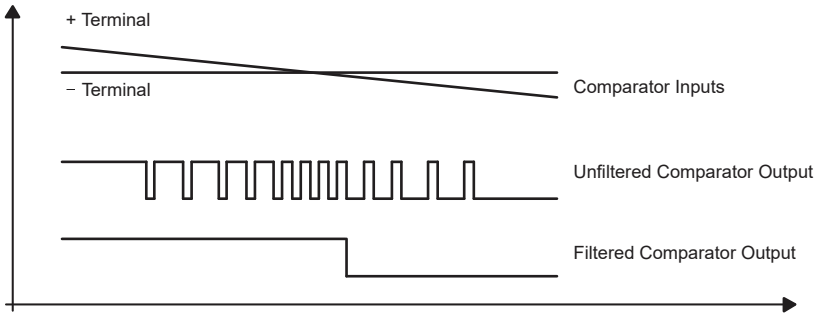



**Figure 19-5. Analog Filter Response at the Output of the Comparator**

**Note**

The output filter only works for the comparator when it operates in fast mode.

### 19.2.5 Sampled Output Mode

Sampled mode is a method of operation for the comparator, which compares input signals at discrete time intervals or samples, producing output signals that change only at the sampling points. The SAMPMODE bit in the register CTL2 enables or disables the sampled mode.

To define the sampling window, two events, EVT0 and EVT1, are generated from the timer. The sampling window can be aligned to less noisy phases with the use of these events. EVT0 sets the sampling window, while EVT1 clears it.

In sampled mode, COMP.STAT is only sampled when the sampling window is high, and the output is not captured at other times. The captured output is used for interrupt and event generation.



Copyright © 2026 Texas Instruments Incorporated

By using sampled mode, the comparator can reduce the effects of noise in the input signals, resulting in more accurate and reliable comparisons. The use of EVT0 and EVT1 to define the sampling window allows for greater flexibility in aligning the window to the least noisy phases of the input signals, further improving the accuracy of the comparisons.

Please note following when internal DAC8 is used in sampled mode and user wants to enter Standby0 mode; for comparator output to work as expected, it is to be ensured that COMP is ready by polling the OUTRDYIFG flag before SoC enters Standby0 mode. 3V devices have a 6ms period and 64us on time. 5V devices have a 3ms period and 64us on time.

**Note**

In sampled mode, DACCODE change should take place after 1 SnH period. If DACCODE is changed faster than this, then hysteresis may happen randomly

User can follow the below sequence for this scenario.

1. In run/active mode, do all the necessary COMP configurations (DAC8 in sampled mode, COMP in ULP mode, etc.)
2. Set power policy to standby0 (DL_SYSCTL_setPowerPolicySTANDBY0)
3. Enable COMP (CTL1.EN bit to be written)
4. Wait for COMP OUTRDYIFG to be set (wait for bit number 3 in RIS register to be 1)
5. Enter Standby0 low power mode (using __WFI() function)

**Note**

Sample and hold mode should only be used with the Ultra Low Power Comparator

### *19.2.6 Blanking Mode*

Due to the noise or other interference, the input signal may fluctuate around the threshold voltage, causing the output to switch rapidly and erratically. This can lead to false triggering. Comparator blanking mode introduces a brief delay between the time the input signal crosses the threshold voltage and the time the output changes state.

When enabled, the blanking source signal acts as an inverted enable for the comparator output. Specifically, when the blanking source is asserted (logic high), the comparator output is forced to its current output state and is prevented from changing state, regardless of the comparator input voltages. Conversely, when the blanking source is de-asserted (logic low), the comparator output is enabled and free to respond to the comparator input voltages. This ensures stable operation during potentially noisy periods. See Figure 19-6 for a block diagram and Figure 19-7 for an example waveform demonstrating this functionality.



Copyright © 2026 Texas Instruments Incorporated

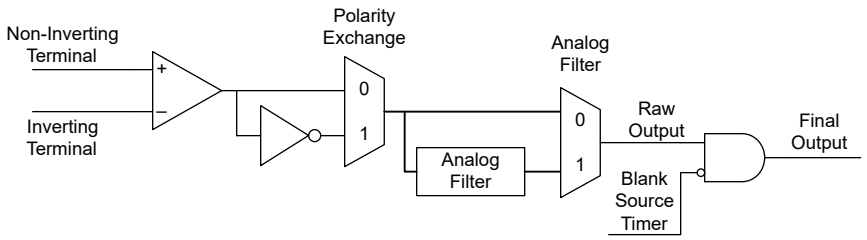



**Figure 19-6. Blanking Mode Block Diagram**

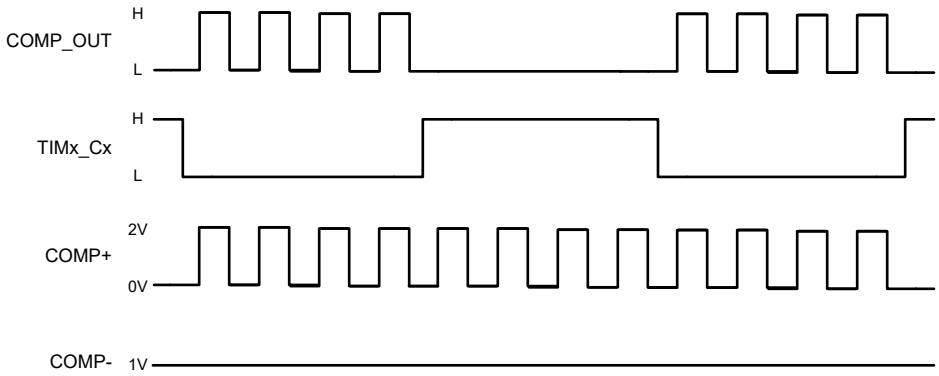



**Figure 19-7. Blanking Mode Example Waveform**

During this delay period, the comparator ignores any further changes in the input signal and drives its output low. This feature is mainly for uses cases like PWM waves indicated by a timer in motor drive, power control, and so on.

The control bits BLANKSRC in the register CTL2 can be used to configure disable or which source should be working on blanking mode. Refer to the data sheet of the device to see the specific blanking sources supported.

**Note**

The timer is used as the source of comparator blanking mode, there is typically a 1 TIMCLK cycle delay between the timer output and the comparator output.

### *19.2.7 Reference Voltage Generator*

Figure 19-8 shows the block diagram of the comparator reference voltage generator. The comparator reference voltage generator consists of a 8-bit DAC along with some configuration options. The REFSRC bits in COMPx.CTL2 register are used to select the reference source for the comparator.

- When REFSRC = 0, the reference voltage generator is disabled and the reference voltage generator cannot be used for comparator operation.
- When REFSRC = 1, the analog supply VDDA is selected as the reference input for the DAC and the DAC output is used as reference voltage for the comparator.
- When REFSRC = 2, the VREF from internal reference module output is selected as the reference input for the DAC and the DAC output is used as reference voltage for the comparator.
- When REFSRC = 3, the VREF from internal reference module output is used directly as reference voltage for the comparator and the DAC is switched off.

The reference voltage generator output can be applied to either the positive terminal or negative terminal of the comparator using the REFSEL bit COMPx.CTL2 register. If external signals are applied to both comparator input terminals, turn off the internal reference voltage generator to reduce current consumption.

When the reference voltage generator output is applied on a comparator terminal using REFSEL bit COMPx.CTL2 register and the comparator channel (from device pins or from internal analog modules) is



Copyright © 2026 Texas Instruments Incorporated

also selected on the same terminal using IPSEL/IPEN or IMSEL/IMEN bits in COMPx.CTL0 register then the comparator channel selection takes precedence.

The DAC cannot be bypassed when REFMODE = 1. If CTL1.REFMODE = 1 (Sampled Mode), then the comparator input will be connected to the DAC regardless of REFSRC setting. E.g. If REFSRC = VDDA and REFMODE = 1 then the comparator input is still connected to the DAC.

**Note**

If COMP and DAC are both not used set the REFSRC = 0, otherwise VREF will get turned on and leak current

DAC8 output is also connected to the internal analog OPA module through OPAx (see *OPA Block Diagram*).

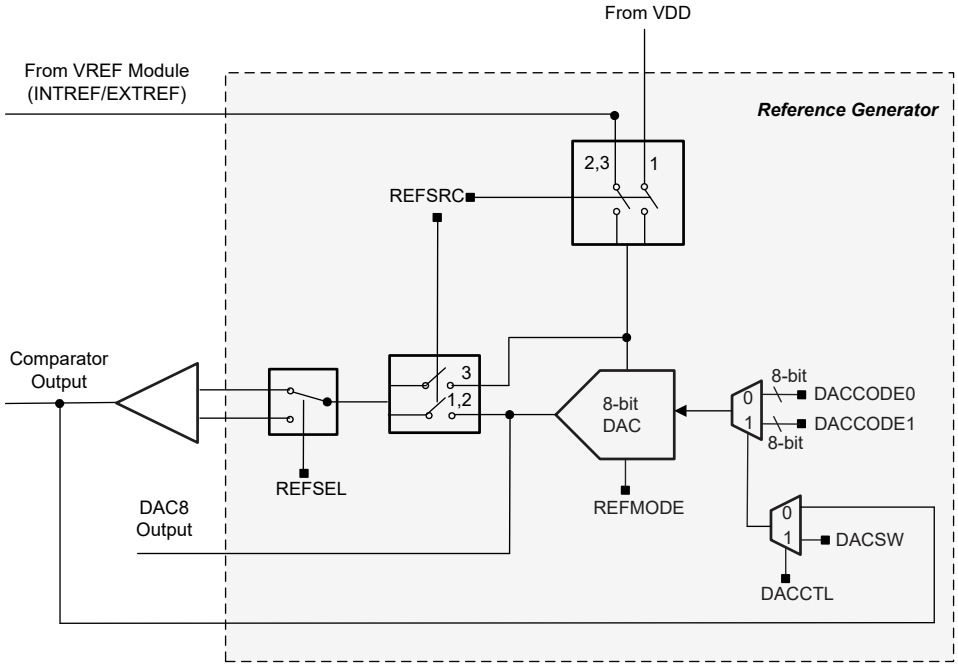



**Figure 19-8. Reference Voltage Generator Block Diagram**

**Note**

The comparator output in the diagrams above is not the comparator output on the pin. It is the comparator output after the output filter but before the output processing stage. This is important in modes such as blanking because the internal DACCODE will continue to toggle (and change the state of the comparator in the image above), but the output on the **pin** will not be toggled while the comparator is in blank mode.

### Integrated 8-bit DAC

The 8-bit DAC input code can be provided through DACCODE0 or DACCODE1 bits in COMPx.CTL3 register. The DACCTL bit in COMPx.CTL2 register determines if the comparator output or a software control bit DACSW in COMPx.CTL2 register selects DACCODE0 or DACCODE1 bits as input to DAC.

- When DACCTL is 0 the comparator output value selects DACCODE0 or DACCODE1. If the comparator output value is 0, DACCODE0 will be the input to DAC. And DACCODE1 will be the input to DAC if the comparator output value is 1.
- When DACCTL is 1 the DACSW bit value that is programmed by software selects DACCODE0 or DACCODE1. If DACSW is 0, DACCODE0 will be the input to DAC. And DACCODE1 will be the input to DAC if DACSW is 1. With this provision it is possible to generate desired hysteresis levels for the comparator without using external components.



Copyright © 2026 Texas Instruments Incorporated

- The REFMODE bit in COMPx.CTL2 register determines if the comparator requests for internal VREF operation in fast or ultra-low-power and also the mode of operation of the 8-bit DAC. When REFMODE bit is 0, the internal VREF is requested for operation in fast mode and the 8-bit DAC in comparator is configured in fast mode as well. When REFMODE bit is 1, the internal VREF is requested for operation in ultra-low-power and the 8-bit DAC in comparator in comparator is configured in ultra-low-power. Operation in fast mode offers higher accuracy but consumes higher current while the ultra-low-power operation consumes lower current but with relaxed reference voltage accuracy. Refer to comparator electrical specifications in device-specific data sheet for details.

### *19.2.8 Window Comparator Mode*

The purpose of window comparator mode is to monitor the input signal within a specified threshold range defined by lower and upper thresholds.

As Figure 19-9 shows, COMP0 and COMP1 can combine to create a window comparator. The input signal is connected to the positive terminal of comparators connected together and the upper and lower threshold voltages are connected to the negative terminal of the comparators.

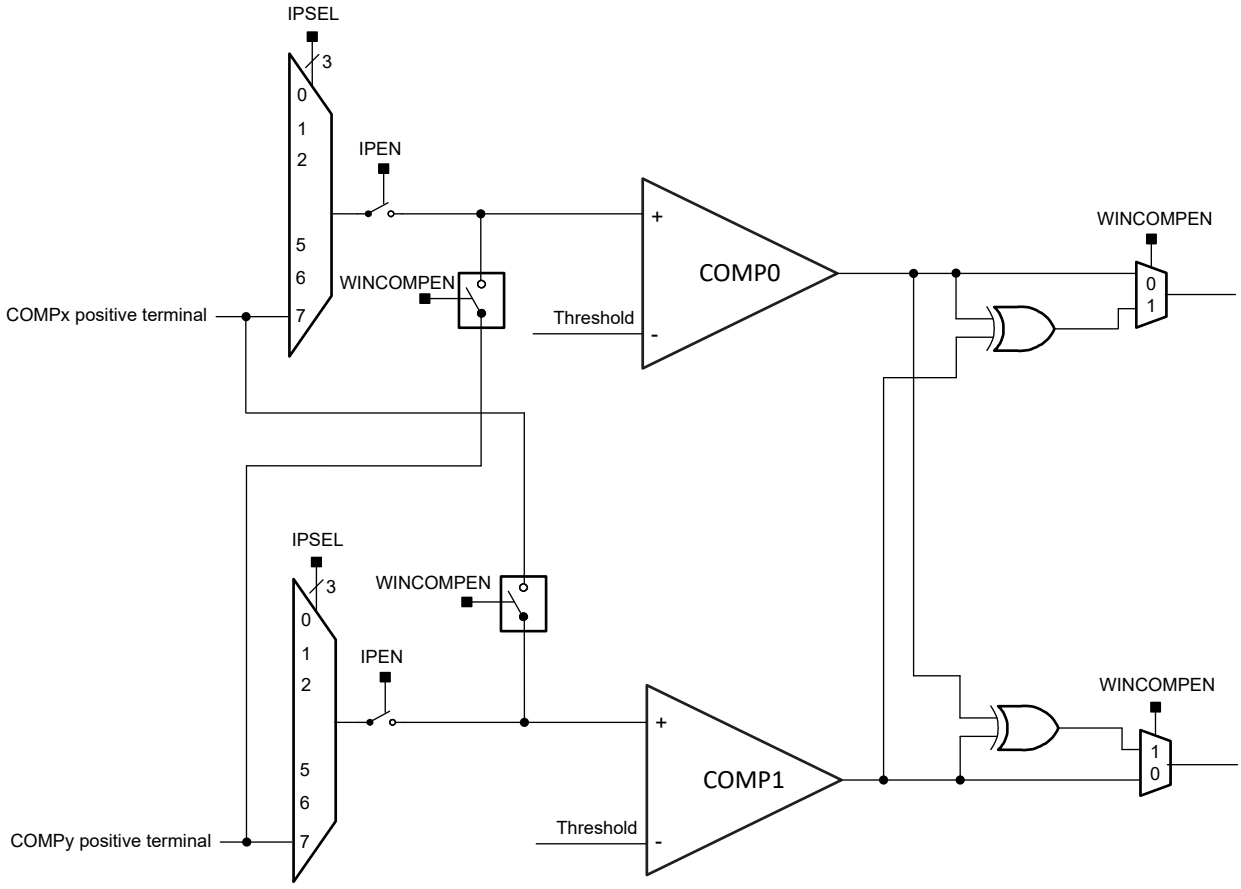



**Figure 19-9. Window Comparator Mode**

Window comparator mode can be enabled by set the WINCOMPEN bit in COMPx.CTL1 register. Figure 19-10 below shows a example of a window mode configuration:

- Configure IPSEL bits in COMP0.CTL0 register to the corresponding input signal pin.
- Set WINCOMPEN bit in COMP0.CTL1 register to 1.
- Clear the WINCOMPEN bit in COMP1.CTL1 register.
- Configure IPSEL bits in COMP1.CTL0 register to 0x07, select the COMP0 positive terminal as the input.



Copyright © 2026 Texas Instruments Incorporated

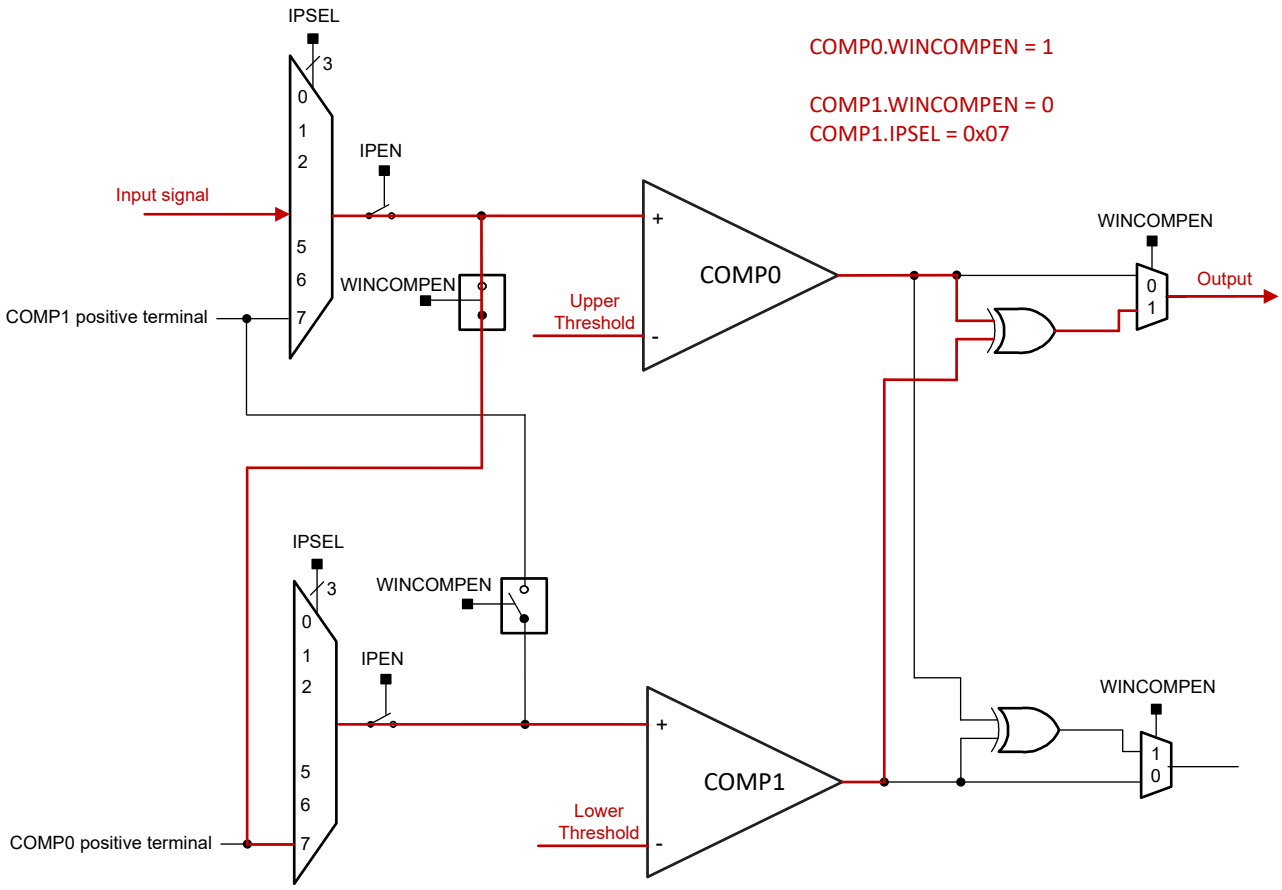



**Figure 19-10. Window Comparator Mode Configuration**

### *19.2.9 Comparator Hysteresis*

The comparator supports programmable hysteresis voltages for fast and ultra-low-power modes to avoid spurious output transitions in case of noisy input signals. When HYST bits in COMPx.CTL1 register are 0, the comparator generates no hysteresis. 10mV, 20mV, and 30mV typical hysteresis voltages are generated by the comparator for HYST bits values 1, 2, and 3 respectively.

The hysteresis can also be implemented by using the internal 8-bit DAC. The input of the reference generator 8-bit DAC can be provided by two values DACCODE0 and DACCODE1 in COMPx.CTL3 register. User can configure the DACCTL bit in COMPx.CTL2 register to 0 and the comparator output value will select the DAC input between DACCODE0 and DACCODE1 values. With this provision it is possible to generate desired hysteresis levels for the comparator without using external components. For more information about this configuration please refer to Integrated 8-bit DAC section.



Copyright © 2026 Texas Instruments Incorporated

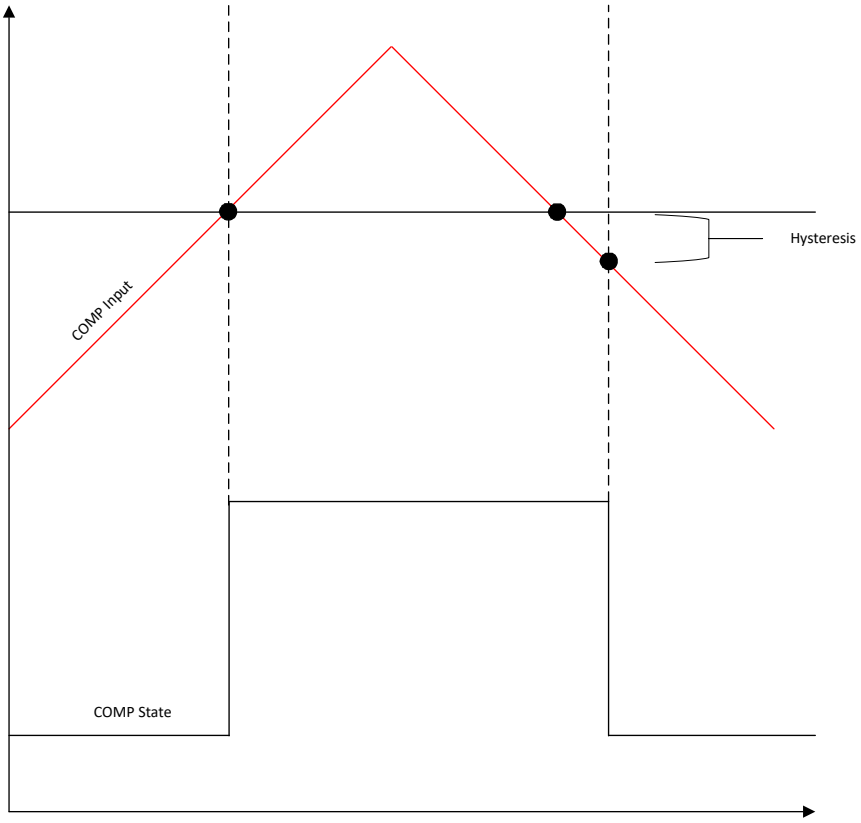



**Figure 19-11. COMP Hysteresis**

**Note**

It is recommended to use a high hysteresis setting when the comparator input signal has a low slew rate

### *19.2.10 Input SHORT Switch*

The SHORT bit shorts the comparator inputs. This can be used to build a simple sample-and-hold for the comparator (see Figure 19-12)



Copyright © 2026 Texas Instruments Incorporated

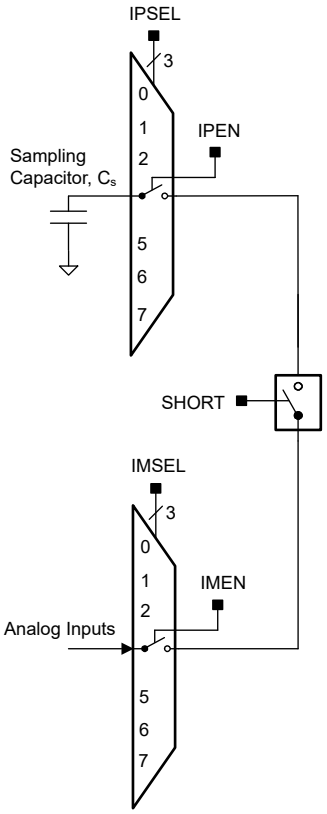



**Figure 19-12. Comparator Short Switch**

The required sampling time is proportional to the size of the sampling capacitor (CS), the resistance of the input switches in series with the short switch (RI) and the resistance of the external source (RS). The sampling capacitor CS should be greater than 100pF. The time constant, Tau, to charge the sampling capacitor CS can be calculated with Equation 25 .

$$T_{au} = (R_I + R_S)\text{x}C_S \quad (25)$$

Depending on the required accuracy, use 3 to 10 Tau as the sampling time. With 3 Tau the sampling capacitor is charged to approximately 95% of the input signals voltage level, with 5 Tau it is charged to more than 99%, and with 10 Tau the sampled voltage is sufficient for 12-bit accuracy.

### *19.2.11 Interrupt and Events Support*

The comparator module contains two event publishers and two event subscribers. One event publisher (CPU_INT) manages comparator interrupt requests (IRQs) to the CPU subsystem through a static event route. The second event publisher (GEN_EVENT) is used to setup the generic event publisher through Generic route.

The two event subscribers are used to define the start and stop points in sampled output mode.

The comparator events are summarized in Table 19-2.

**Table 19-2. Comparator Events**

| Event | Type | Source | Destination | Route | Configuration | Functionality |
|---|---|---|---|---|---|---|
| CPU interrupt | Publisher | Comparator | CPU Subsystem | Static route | CPU_INT registers | Fixed interrupt route from comparator to CPU |



Copyright © 2026 Texas Instruments Incorporated

**Table 19-2. Comparator Events (continued)**

| Event | Type | Source | Destination | Route | Configuration | Functionality |
|---|---|---|---|---|---|---|
| Generic publisher event | Publisher | Comparator | Other peripherals | Generic route | GEN_EVENT FPUB_1 registers | Configurable route from comparator to other peripherals |
| Generic subscriber event | Subscriber | Other peripherals | Comparator | Generic route | GEN_EVENT FSUB_0 registers | Configurable route from other peripherals to the comparator start of sample window |
| Generic subscriber event | Subscriber | Other peripherals | Comparator | Generic route | GEN_EVENT FSUB_1 registers | Configurable route from other peripherals to the comparator stop sampling window |

#### 19.2.11.1 CPU Interrupt Event Publisher (CPU_INT)

The comparator module provides 3 interrupt sources which can be configured to source a CPU interrupt event. In order of decreasing interrupt priority, the CPU interrupt events from the comparator are:

**Table 19-3. Comparator CPU Interrupt Event Conditions (CPU_INT)**

| IIDX STAT | Name | Description |
|---|---|---|
| 0x01 | COMPIFG | The interrupt flags COMPIFG and COMPINVIFG are set either on the rising or falling edge of the comparator output, selected by the IES bit. When IES bit is 0, rising edge of the comparator output sets the COMPIFG and falling edge sets the COMPINVIFG. When IES bit is 1, falling edge of the comparator output sets the COMPIFG and rising edge sets the COMPINVIFG. |
| 0x02 | COMPINVIFG | The interrupt flags COMPIFG and COMPINVIFG are set either on the rising or falling edge of the comparator output, selected by the IES bit. When IES bit is 0, rising edge of the comparator output sets the COMPIFG and falling edge sets the COMPINVIFG. When IES bit is 1, falling edge of the comparator output sets the COMPIFG and rising edge sets the COMPINVIFG. |
| 0x03 | OUTRDYIFG | Comparator output ready interrupt. This bit is set when the comparator output is valid. |

The CPU interrupt event configuration is managed with the CPU_INT event management registers. See Section 7.2.5 for guidance on configuring the Event registers for CPU interrupts.

#### 19.2.11.2 Generic Event Publisher (GEN_EVENT)

A generic route is a route in which the comparator peripheral publishing the event is configured to use one of several available generic route channels to publish its event to another entity (or entities, in the case of a splitter route), where an entity can be another peripheral, a generic DMA trigger event, or a generic CPU event.

The GEN_EVENT register is used to select a peripheral condition (Table 19-4) to use for publishing an event. FPUB_1 is the publisher port register and it is used to configure which generic route channel to use to broadcast the event. A second peripheral, the DMA, or the CPU can subscribe to this event by configuring its subscriber port to listen on the same generic route channel which the publishing peripheral is connected to.

For example, through the use of a generic event channel, it is possible to directly trigger a timer capture from a comparator event, by connecting a comparator FPUB_1 and TIM FSUB_x to the same generic event channel. Refer to Section 7.1.3.3 and Section 7.2.3 for how generic event route works.



Copyright © 2026 Texas Instruments Incorporated

**Table 19-4. Comparator Generic Event Conditions (GEN_EVENT)**

| IIDX STAT | Name | Description |
|---|---|---|
| 0x01 | COMPIFG | The interrupt flags COMPIFG and COMPINVIFG are set either on the rising or falling edge of the comparator output, selected by the IES bit. When IES bit is 0, rising edge of the comparator output sets the COMPIFG and falling edge sets the COMPINVIFG. When IES bit is 1, falling edge of the comparator output sets the COMPIFG and rising edge sets the COMPINVIFG. |
| 0x02 | COMPINVIFG | The interrupt flags COMPIFG and COMPINVIFG are set either on the rising or falling edge of the comparator output, selected by the IES bit. When IES bit is 0, rising edge of the comparator output sets the COMPIFG and falling edge sets the COMPINVIFG. When IES bit is 1, falling edge of the comparator output sets the COMPIFG and rising edge sets the COMPINVIFG. |
| 0x03 | OUTRDYIFG | Comparator output ready interrupt. This bit is set when the comparator output is valid. |

See Section 7.2.5 for guidance on configuring the Event registers.

#### 19.2.11.3 Generic Event Subscribers

The two event subscribers are used to establish the sampling start and stop times when the comparator is used in the sampled output mode.

- FSUB_0 selects the generic event route at the device level on which the start of sampling window event is sent.
- FSUB_1 selects the generic event route at the device level on which the stop of sampling window event is sent.

As the sampling window is driven from generic event routes, any peripheral module with generic publisher capabilities can set the comparator sampling window, including timers and GPIO.



Copyright © 2026 Texas Instruments Incorporated

## 19.3 COMP0 Registers

Table 19-5 lists the memory-mapped registers for the COMP0 registers. All register offset addresses not listed in Table 19-5 should be considered as reserved locations and the register contents should not be modified.

**Table 19-5. COMP0 Registers**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 400h | FSUB_0 | Subscriber Port 0 | Section 19.3.1 |
| 404h | FSUB_1 | Subscriber Port 1 | Section 19.3.2 |
| 444h | FPUB_1 | Publisher port 1 | Section 19.3.3 |
| 800h | PWREN | Power enable | Section 19.3.4 |
| 804h | RSTCTL | Reset Control | Section 19.3.5 |
| 808h | CLKCFG | Peripheral Clock Configuration Register | Section 19.3.6 |
| 814h | STAT | Status Register | Section 19.3.7 |
| 1020h | IIDX | Interrupt index | Section 19.3.8 |
| 1028h | IMASK | Interrupt mask | Section 19.3.9 |
| 1030h | RIS | Raw interrupt status | Section 19.3.10 |
| 1038h | MIS | Masked interrupt status | Section 19.3.11 |
| 1040h | ISET | Interrupt set | Section 19.3.12 |
| 1048h | ICLR | Interrupt clear | Section 19.3.13 |
| 1050h | IIDX | Interrupt index | Section 19.3.14 |
| 1058h | IMASK | Interrupt mask | Section 19.3.15 |
| 1060h | RIS | Raw interrupt status | Section 19.3.16 |
| 1068h | MIS | Masked interrupt status | Section 19.3.17 |
| 1070h | ISET | Interrupt set | Section 19.3.18 |
| 1078h | ICLR | Interrupt clear | Section 19.3.19 |
| 10E0h | EVT_MODE | Event Mode | Section 19.3.20 |
| 10FCh | DESC | Module Description | Section 19.3.21 |
| 1100h | CTL0 | Control 0 | Section 19.3.22 |
| 1104h | CTL1 | Control 1 | Section 19.3.23 |
| 1108h | CTL2 | Control 2 | Section 19.3.24 |
| 110Ch | CTL3 | Control 3 | Section 19.3.25 |
| 1120h | STAT | Status | Section 19.3.26 |

Complex bit access types are encoded to fit into small table cells. Table 19-6 shows the codes that are used for access types in this section.

**Table 19-6. COMP0 Access Type Codes**

| Access Type | Code | Description |
|---|---|---|
| Read Type | | |
| R | R | Read |
| Write Type | | |
| W | W | Write |
| WK | W<br>K | Write<br>Write protected by a key |
| Reset or Default Value | | |
| *-n* | | Value after reset or the default value |



Copyright © 2026 Texas Instruments Incorporated

### 19.3.1 FSUB_0 Register (Offset = 400h) [Reset = 00000000h]

FSUB_0 is shown in Figure 19-13 and described in Table 19-7.

Return to the Summary Table.

Subscriber port

**Figure 19-13. FSUB_0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | CHANID | | | |
| R-0h | | | | | | | | | | | | R/W-0h | | | |

**Table 19-7. FSUB_0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3-0 | CHANID | R/W | 0h | 0 = disconnected.<br>1-15 = connected to channelID = CHANID.<br>0h = A value of 0 specifies that the event is not connected<br>Fh = Consult your device datasheet as the actual allowed maximum may be less than 15. |



Copyright © 2026 Texas Instruments Incorporated

### 19.3.2 FSUB_1 Register (Offset = 404h) [Reset = 00000000h]

FSUB_1 is shown in Figure 19-14 and described in Table 19-8.

Return to the Summary Table.

Subscriber port

**Figure 19-14. FSUB_1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | CHANID | | | |
| R-0h | | | | | | | | | | | | R/W-0h | | | |

**Table 19-8. FSUB_1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3-0 | CHANID | R/W | 0h | 0 = disconnected.<br>1-15 = connected to channelID = CHANID.<br>0h = A value of 0 specifies that the event is not connected<br>Fh = Consult your device datasheet as the actual allowed maximum may be less than 15. |



Copyright © 2026 Texas Instruments Incorporated

### 19.3.3 FPUB_1 Register (Offset = 444h) [Reset = 00000000h]

FPUB_1 is shown in Figure 19-15 and described in Table 19-9.

Return to the Summary Table.

Publisher port

**Figure 19-15. FPUB_1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | CHANID | | | |
| R-0h | | | | | | | | | | | | R/W-0h | | | |

**Table 19-9. FPUB_1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3-0 | CHANID | R/W | 0h | 0 = disconnected.<br>1-15 = connected to channelID = CHANID.<br>0h = A value of 0 specifies that the event is not connected<br>Fh = Consult your device datasheet as the actual allowed maximum may be less than 15. |



Copyright © 2026 Texas Instruments Incorporated

### 19.3.4 PWREN Register (Offset = 800h) [Reset = 00000000h]

PWREN is shown in Figure 19-16 and described in Table 19-10.

Return to the Summary Table.

Register to control the power state

**Figure 19-16. PWREN Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | ENABLE |
| R-0h | | | | | | | R/WK-0h |

**Table 19-10. PWREN Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | KEY to allow Power State Change<br>26h = KEY to allow write access to this register |
| 23-1 | RESERVED | R | 0h | |
| 0 | ENABLE | R/WK | 0h | Enable the power<br>KEY must be set to 26h to write to this bit.<br>0h = Disable Power<br>1h = Enable Power |



Copyright © 2026 Texas Instruments Incorporated

### 19.3.5 RSTCTL Register (Offset = 804h) [Reset = 00000000h]

RSTCTL is shown in Figure 19-17 and described in Table 19-11.

Return to the Summary Table.

Register to control reset assertion and de-assertion

**Figure 19-17. RSTCTL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | RESETSTKYCLR | RESETASSERT |
| R-0h | | | | | | WK-0h | WK-0h |

**Table 19-11. RSTCTL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | Unlock key<br>B1h = KEY to allow write access to this register |
| 23-2 | RESERVED | R | 0h | |
| 1 | RESETSTKYCLR | WK | 0h | Clear the RESETSTKY bit in the STAT register<br>KEY must be set to B1h to write to this bit.<br>0h = Writing 0 has no effect<br>1h = Clear reset sticky bit |
| 0 | RESETASSERT | WK | 0h | Assert reset to the peripheral<br>KEY must be set to B1h to write to this bit.<br>0h = Writing 0 has no effect<br>1h = Assert reset |



Copyright © 2026 Texas Instruments Incorporated

### 19.3.6 CLKCFG Register (Offset = 808h) [Reset = 00000000h]

CLKCFG is shown in Figure 19-18 and described in Table 19-12.

Return to the Summary Table.

Peripheral Clock Configuration Register

**Figure 19-18. CLKCFG Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | BLOCKASYNC |
| R-0h | | | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |

**Table 19-12. CLKCFG Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | KEY to Allow State Change -- 0xA9<br>A9h = Key value to be used in writing to this register for the write to take effect. |
| 23-9 | RESERVED | R | 0h | |
| 8 | BLOCKASYNC | R/W | 0h | Async Clock Request is blocked from starting SYSOSC or forcing bus clock to 32MHz<br>0h = disable COMP to request SYSOSC<br>1h = enable COMP to request SYSOSC |
| 7-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 19.3.7 STAT Register (Offset = 814h) [Reset = 00000000h]

STAT is shown in Figure 19-19 and described in Table 19-13.

Return to the Summary Table.

peripheral enable and reset status

**Figure 19-19. STAT Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | RESETSTKY |
| R-0h | | | | | | | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |

**Table 19-13. STAT Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-17 | RESERVED | R | 0h | |
| 16 | RESETSTKY | R | 0h | This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register<br>0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register<br>1h = The peripheral was reset since the last bit clear |
| 15-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 19.3.8 IIDX Register (Offset = 1020h) [Reset = 00000000h]

IIDX is shown in Figure 19-20 and described in Table 19-14.

Return to the Summary Table.

Interrupt index register. This read-only register provides the interrupt index of the pending interrupt with the highest priority. It also indicates if no interrupt is pending. The priority order is fixed: lower index equals higher priority. Alternatively to the use of IIDX, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred.
On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flags in the RIS and MIS are cleared as well. After a read from the CPU (not from the debug interface), the register must be updated with the next highest priority interrupt or indicate that no interrupt is pending. Only interrupts which are selected via IMASK are indicated.

**Figure 19-20. IIDX Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | | | STAT | |
| R-0h | | | | | | | | | | | | | | R-0h | |

**Table 19-14. IIDX Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | |
| 1-0 | STAT | R | 0h | Interrupt index status<br>0h = No pending interrupt<br>2h = Comparator output interrupt<br>3h = Comparator output inverted interrupt<br>4h = Comparator output ready interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 19.3.9 IMASK Register (Offset = 1028h) [Reset = 00000000h]

IMASK is shown in Figure 19-21 and described in Table 19-15.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS."

**Figure 19-21. IMASK Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | OUTRDYIFG | COMPINVIFG | COMPIFG | RESERVED |
| R-0h | | | | R/W-0h | R/W-0h | R/W-0h | R-0h |

**Table 19-15. IMASK Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | OUTRDYIFG | R/W | 0h | Masks OUTRDYIFG<br>0h = Interrupt is masked out<br>1h = Interrupt will request an interrupt service routine and corresponding bit in MIS will be set |
| 2 | COMPINVIFG | R/W | 0h | Masks COMPINVIFG<br>0h = Interrupt is masked out<br>1h = Interrupt will request an interrupt service routine and corresponding bit in MIS will be set |
| 1 | COMPIFG | R/W | 0h | Masks COMPIFG<br>0h = Interrupt is masked out<br>1h = Interrupt will request an interrupt service routine and corresponding bit in MIS will be set |
| 0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 19.3.10 RIS Register (Offset = 1030h) [Reset = 00000000h]

RIS is shown in Figure 19-22 and described in Table 19-16.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Figure 19-22. RIS Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | OUTRDYIFG | COMPINVIFG | COMPIFG | RESERVED |
| R-0h | | | | R-0h | R-0h | R-0h | R-0h |

**Table 19-16. RIS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | OUTRDYIFG | R | 0h | Raw interrupt status for comparator output ready interrupt flag. This bit is set when the comparator output is valid.<br>0h = No interrupt pending<br>1h = Interrupt pending |
| 2 | COMPINVIFG | R | 0h | Raw interrupt status for comparator output inverted interrupt flag. The IES bit defines the transition of the comparator output setting this bit.<br>0h = No interrupt pending<br>1h = Interrupt pending |
| 1 | COMPIFG | R | 0h | Raw interrupt status for comparator output interrupt flag. The IES bit defines the transition of the comparator output setting this bit.<br>0h = No interrupt pending<br>1h = Interrupt pending |
| 0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 19.3.11 MIS Register (Offset = 1038h) [Reset = 00000000h]

MIS is shown in Figure 19-23 and described in Table 19-17.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 19-23. MIS Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | OUTRDYIFG | COMPINVIFG | COMPIFG | RESERVED |
| R-0h | | | | R-0h | R-0h | R-0h | R-0h |

**Table 19-17. MIS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | OUTRDYIFG | R | 0h | Masked interrupt status for OUTRDYIFG<br>0h = OUTRDYIFG does not request an interrupt service routine<br>1h = OUTRDYIFG requests an interrupt service routine |
| 2 | COMPINVIFG | R | 0h | Masked interrupt status for COMPINVIFG<br>0h = COMPINVIFG does not request an interrupt service routine<br>1h = COMPINVIFG requests an interrupt service routine |
| 1 | COMPIFG | R | 0h | Masked interrupt status for COMPIFG<br>0h = COMPIFG does not request an interrupt service routine<br>1h = COMPIFG requests an interrupt service routine |
| 0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 19.3.12 ISET Register (Offset = 1040h) [Reset = 00000000h]

ISET is shown in Figure 19-24 and described in Table 19-18.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 19-24. ISET Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | OUTRDYIFG | COMPINVIFG | COMPIFG | RESERVED |
| R-0h | | | | W-0h | W-0h | W-0h | R-0h |

**Table 19-18. ISET Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | OUTRDYIFG | W | 0h | Sets OUTRDYIFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to OUTRDYIFG is set |
| 2 | COMPINVIFG | W | 0h | Sets COMPINVIFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to COMPINVIFG is set |
| 1 | COMPIFG | W | 0h | Sets COMPIFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to COMPIFG is set |
| 0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 19.3.13 ICLR Register (Offset = 1048h) [Reset = 00000000h]

ICLR is shown in Figure 19-25 and described in Table 19-19.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

**Figure 19-25. ICLR Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | OUTRDYIFG | COMPINVIFG | COMPIFG | RESERVED |
| R-0h | | | | W-0h | W-0h | W-0h | R-0h |

**Table 19-19. ICLR Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | OUTRDYIFG | W | 0h | Clears OUTRDYIFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to OUTRDYIFG is cleared |
| 2 | COMPINVIFG | W | 0h | Clears COMPINVIFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to COMPINVIFG is cleared |
| 1 | COMPIFG | W | 0h | Clears COMPIFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to COMPIFG is cleared |
| 0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 19.3.14 IIDX Register (Offset = 1050h) [Reset = 00000000h]

IIDX is shown in Figure 19-26 and described in Table 19-20.

Return to the Summary Table.

Interrupt index register. This read-only register provides the interrupt index of the pending interrupt with the highest priority. It also indicates if no interrupt is pending. The priority order is fixed: lower index equals higher priority. Alternatively to the use of IIDX, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred.
On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flags in the RIS and MIS are cleared as well. After a read from the CPU (not from the debug interface), the register must be updated with the next highest priority interrupt or indicate that no interrupt is pending. Only interrupts which are selected via IMASK are indicated.

**Figure 19-26. IIDX Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | | | STAT | |
| R-0h | | | | | | | | | | | | | | R-0h | |

**Table 19-20. IIDX Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | |
| 1-0 | STAT | R | 0h | Interrupt index status<br>0h = No pending interrupt<br>2h = Comparator output interrupt<br>3h = Comparator output inverted interrupt<br>4h = Comparator output ready interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 19.3.15 IMASK Register (Offset = 1058h) [Reset = 00000000h]

IMASK is shown in Figure 19-27 and described in Table 19-21.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS."

**Figure 19-27. IMASK Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | OUTRDYIFG | COMPINVIFG | COMPIFG | RESERVED |
| R-0h | | | | R/W-0h | R/W-0h | R/W-0h | R-0h |

**Table 19-21. IMASK Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | OUTRDYIFG | R/W | 0h | Masks OUTRDYIFG<br>0h = Interrupt is masked out<br>1h = Interrupt will request an interrupt service routine and corresponding bit in MIS will be set |
| 2 | COMPINVIFG | R/W | 0h | Masks COMPINVIFG<br>0h = Interrupt is masked out<br>1h = Interrupt will request an interrupt service routine and corresponding bit in MIS will be set |
| 1 | COMPIFG | R/W | 0h | Masks COMPIFG<br>0h = Interrupt is masked out<br>1h = Interrupt will request an interrupt service routine and corresponding bit in MIS will be set |
| 0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 19.3.16 RIS Register (Offset = 1060h) [Reset = 00000000h]

RIS is shown in Figure 19-28 and described in Table 19-22.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Figure 19-28. RIS Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | OUTRDYIFG | COMPINVIFG | COMPIFG | RESERVED |
| R-0h | | | | R-0h | R-0h | R-0h | R-0h |

**Table 19-22. RIS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | OUTRDYIFG | R | 0h | Raw interrupt status for comparator output ready interrupt flag. This bit is set when the comparator output is valid.<br>0h = No interrupt pending<br>1h = Interrupt pending |
| 2 | COMPINVIFG | R | 0h | Raw interrupt status for comparator output inverted interrupt flag. The IES bit defines the transition of the comparator output setting this bit.<br>0h = No interrupt pending<br>1h = Interrupt pending |
| 1 | COMPIFG | R | 0h | Raw interrupt status for comparator output interrupt flag. The IES bit defines the transition of the comparator output setting this bit.<br>0h = No interrupt pending<br>1h = Interrupt pending |
| 0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 19.3.17 MIS Register (Offset = 1068h) [Reset = 00000000h]

MIS is shown in Figure 19-29 and described in Table 19-23.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 19-29. MIS Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | OUTRDYIFG | COMPINVIFG | COMPIFG | RESERVED |
| R-0h | | | | R-0h | R-0h | R-0h | R-0h |

**Table 19-23. MIS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | OUTRDYIFG | R | 0h | Masked interrupt status for OUTRDYIFG<br>0h = OUTRDYIFG does not request an interrupt service routine<br>1h = OUTRDYIFG requests an interrupt service routine |
| 2 | COMPINVIFG | R | 0h | Masked interrupt status for COMPINVIFG<br>0h = COMPINVIFG does not request an interrupt service routine<br>1h = COMPINVIFG requests an interrupt service routine |
| 1 | COMPIFG | R | 0h | Masked interrupt status for COMPIFG<br>0h = COMPIFG does not request an interrupt service routine<br>1h = COMPIFG requests an interrupt service routine |
| 0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 19.3.18 ISET Register (Offset = 1070h) [Reset = 00000000h]

ISET is shown in Figure 19-30 and described in Table 19-24.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 19-30. ISET Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | OUTRDYIFG | COMPINVIFG | COMPIFG | RESERVED |
| R-0h | | | | W-0h | W-0h | W-0h | R-0h |

**Table 19-24. ISET Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | OUTRDYIFG | W | 0h | Sets OUTRDYIFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to OUTRDYIFG is set |
| 2 | COMPINVIFG | W | 0h | Sets COMPINVIFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to COMPINVIFG is set |
| 1 | COMPIFG | W | 0h | Sets COMPIFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to COMPIFG is set |
| 0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 19.3.19 ICLR Register (Offset = 1078h) [Reset = 00000000h]

ICLR is shown in Figure 19-31 and described in Table 19-25.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

**Figure 19-31. ICLR Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | OUTRDYIFG | COMPINVIFG | COMPIFG | RESERVED |
| R-0h | | | | W-0h | W-0h | W-0h | R-0h |

**Table 19-25. ICLR Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | OUTRDYIFG | W | 0h | Clears OUTRDYIFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to OUTRDYIFG is cleared |
| 2 | COMPINVIFG | W | 0h | Clears COMPINVIFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to COMPINVIFG is cleared |
| 1 | COMPIFG | W | 0h | Clears COMPIFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to COMPIFG is cleared |
| 0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 19.3.20 EVT_MODE Register (Offset = 10E0h) [Reset = 00000009h]

EVT_MODE is shown in Figure 19-32 and described in Table 19-26.

Return to the Summary Table.

Event mode register. It is used to select whether each line is disabled, in software mode (software clears the RIS) or in
hardware mode (hardware clears the RIS)

**Figure 19-32. EVT_MODE Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | EVT1_CFG | | INT0_CFG | |
| R-0h | | | | R-2h | | R-1h | |

**Table 19-26. EVT_MODE Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3-2 | EVT1_CFG | R | 2h | Event line mode select for event corresponding to GEN_EVENT<br>0h = The interrupt or event line is disabled.<br>1h = Event handled by software. Software must clear the associated RIS flag.<br>2h = Event handled by hardware. The hardware (another module) clears automatically the associated RIS flag. |
| 1-0 | INT0_CFG | R | 1h | Event line mode select for event corresponding to CPU_INT<br>0h = The interrupt or event line is disabled.<br>1h = Event handled by software. Software must clear the associated RIS flag.<br>2h = Event handled by hardware. The hardware (another module) clears automatically the associated RIS flag. |



Copyright © 2026 Texas Instruments Incorporated

### 19.3.21 DESC Register (Offset = 10FCh) [Reset = 00000000h]

DESC is shown in Figure 19-33 and described in Table 19-27.

Return to the Summary Table.

This register identifies the peripheral and its exact version.

**Figure 19-33. DESC Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MODULEID | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| FEATUREVER | | | | RESERVED | | | | MAJREV | | | | MINREV | | | |
| R-0h | | | | R-0h | | | | R-0h | | | | R-0h | | | |

**Table 19-27. DESC Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | MODULEID | R | 611h | Module identification contains a unique peripheral identification number. The assignments are maintained in a central database for all of the platform modules to ensure uniqueness. |
| 15-12 | FEATUREVER | R | 0h | Feature Set for the module *instance* |
| 11-8 | RESERVED | R | 0h | |
| 7-4 | MAJREV | R | 0h | Major rev of the IP |
| 3-0 | MINREV | R | 0h | Minor rev of the IP |



Copyright © 2026 Texas Instruments Incorporated

### 19.3.22 CTL0 Register (Offset = 1100h) [Reset = 00000000h]

CTL0 is shown in Figure 19-34 and described in Table 19-28.

Return to the Summary Table.

Control 0 register.

**Figure 19-34. CTL0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| IMEN | RESERVED | | | | | | |
| R/W-0h | R-0h | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | IMSEL | | |
| R-0h | | | | | R/W-0h | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| IPEN | RESERVED | | | | | | |
| R/W-0h | R-0h | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | IPSEL | | |
| R-0h | | | | | R/W-0h | | |

**Table 19-28. CTL0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | IMEN | R/W | 0h | Channel input enable for the negative terminal of the comparator.<br>0h = Selected analog input channel for negative terminal is disabled<br>1h = Selected analog input channel for negative terminal is enabled |
| 30-19 | RESERVED | R | 0h | |
| 18-16 | IMSEL | R/W | 0h | Channel input selected for the negative terminal of the comparator if IMEN is set to 1.<br>0h = Channel 0 selected<br>1h = Channel 1 selected<br>2h = Channel 2 selected<br>3h = Channel 3 selected<br>4h = Channel 4 selected<br>5h = Channel 5 selected<br>6h = Channel 6 selected<br>7h = Channel 7 selected |
| 15 | IPEN | R/W | 0h | Channel input enable for the positive terminal of the comparator.<br>0h = Selected analog input channel for positive terminal is disabled<br>1h = Selected analog input channel for positive terminal is enabled |
| 14-3 | RESERVED | R | 0h | |
| 2-0 | IPSEL | R/W | 0h | Channel input selected for the positive terminal of the comparator if IPEN is set to 1.<br>0h = Channel 0 selected<br>1h = Channel 1 selected<br>2h = Channel 2 selected<br>3h = Channel 3 selected<br>4h = Channel 4 selected<br>5h = Channel 5 selected<br>6h = Channel 6 selected<br>7h = Channel 7 selected |



Copyright © 2026 Texas Instruments Incorporated

### 19.3.23 CTL1 Register (Offset = 1104h) [Reset = 00000000h]

CTL1 is shown in Figure 19-35 and described in Table 19-29.

Return to the Summary Table.

Control 1 register.

**Figure 19-35. CTL1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | WINCOMPEN | DACOUTEN | FLTDLY | | FLTEN |
| R-0h | | | R/W-0h | R/W-0h | R/W-0h | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| OUTPOL | HYST | | IES | SHORT | EXCH | MODE | ENABLE |
| R/W-0h | R/W-0h | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

**Table 19-29. CTL1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-13 | RESERVED | R | 0h | |
| 12 | WINCOMPEN | R/W | 0h | This bit enables window comparator operation of comparator.<br>0h = window comparator is disable<br>1h = window comparator is enable |
| 11 | DACOUTEN | R/W | 0h | This bit enables DAC8 output to be connected to the pin. Please consult the device-specific datasheet to determine if this feature is present.<br>0h = Disable the DAC8 output connectivity to pin.<br>1h = Enable the DAC8 output connectivity to pin. |
| 10-9 | FLTDLY | R/W | 0h | These bits select the comparator output filter delay. See the device-specific data sheet for specific values on comparator propagation delay for different filter delay settings.<br>0h = Typical filter delay of 70 ns<br>1h = Typical filter delay of 500 ns<br>2h = Typical filter delay of 1200 ns<br>3h = Typical filter delay of 2700 ns |
| 8 | FLTEN | R/W | 0h | This bit enables the analog filter at comparator output.<br>0h = Comparator output filter is disabled<br>1h = Comparator output filter is enabled |
| 7 | OUTPOL | R/W | 0h | This bit selects the comparator output polarity.<br>0h = Comparator output is non-inverted<br>1h = Comparator output is inverted |
| 6-5 | HYST | R/W | 0h | These bits select the hysteresis setting of the comparator.<br>0h = No hysteresis<br>1h = Low hysteresis, typical 10mV<br>2h = Medium hysteresis, typical 20mV<br>3h = High hysteresis, typical 30mV |
| 4 | IES | R/W | 0h | This bit selected the interrupt edge for COMPIFG and COMPINVIFG.<br>0h = Rising edge sets COMPIFG and falling edge sets COMPINVIFG<br>1h = Falling edge sets COMPIFG and rising edge sets COMPINVIFG |



Copyright © 2026 Texas Instruments Incorporated

**Table 19-29. CTL1 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 3 | SHORT | R/W | 0h | This bit shorts the positive and negative input terminals of the comparator.<br>0h = Comparator positive and negative input terminals are not shorted<br>1h = Comparator positive and negative input terminals are shorted |
| 2 | EXCH | R/W | 0h | This bit exchanges the comparator inputs and inverts the comparator output.<br>0h = Comparator inputs not exchanged and output not inverted<br>1h = Comparator inputs exchanged and output inverted |
| 1 | MODE | R/W | 0h | This bit selects the comparator operating mode.<br>0h = Comparator is in fast mode<br>1h = Comparator is in ultra-low power mode |
| 0 | ENABLE | R/W | 0h | This bit turns on the comparator. When the comparator is turned off it consumes no power.<br>0h = Comparator is off<br>1h = Comparator is on |



Copyright © 2026 Texas Instruments Incorporated

### 19.3.24 CTL2 Register (Offset = 1108h) [Reset = 00000000h]

CTL2 is shown in Figure 19-36 and described in Table 19-30.

Return to the Summary Table.

Control 2 register.

**Figure 19-36. CTL2 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | SAMPMODE |
| R-0h | | | | | | | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | DACSW | DACCTL |
| R-0h | | | | | | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | BLANKSRC | | |
| R-0h | | | | | R/W-0h | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| REFSEL | RESERVED | REFSRC | | | RESERVED | | REFMODE |
| R/W-0h | R-0h | R/W-0h | | | R-0h | | R/W-0h |

**Table 19-30. CTL2 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-25 | RESERVED | R | 0h | |
| 24 | SAMPMODE | R/W | 0h | Enable sampled mode of comparator.<br>0h = Sampled mode disabled<br>1h = Sampled mode enabled |
| 23-18 | RESERVED | R | 0h | |
| 17 | DACSW | R/W | 0h | This bit selects between DACCODE0 and DACCODE1 to 8-bit DAC when DACCTL bit is 1.<br>0h = DACCODE0 selected for 8-bit DAC<br>1h = DACCODE1 selected for 8-bit DAC |
| 16 | DACCTL | R/W | 0h | This bit determines if the comparator output or DACSW bit controls the selection between DACCODE0 and DACCODE1.<br>0h = Comparator output controls selection between DACCODE0 and DACCODE1<br>1h = DACSW bit controls selection between DACCODE0 and DACCODE1 |
| 15-11 | RESERVED | R | 0h | |
| 10-8 | BLANKSRC | R/W | 0h | These bits select the blanking source for the comparator.<br>0h = Blanking source disabled<br>1h = Select Blanking Source 1<br>2h = Select Blanking Source 2<br>3h = Select Blanking Source 3<br>4h = Select Blanking Source 4<br>5h = Select Blanking Source 5<br>6h = Select Blanking Source 6 |



Copyright © 2026 Texas Instruments Incorporated

**Table 19-30. CTL2 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7 | REFSEL | R/W | 0h | This bit selects if the selected reference voltage is applied to positive or negative terminal of the comparator.<br>0h = If EXCH bit is 0, the selected reference is applied to positive terminal.<br>If EXCH bit is 1, the selected reference is applied to negative terminal.<br>1h = If EXCH bit is 0, the selected reference is applied to negative terminal.<br>If EXCH bit is 1, the selected reference is applied to positive terminal. |
| 6 | RESERVED | R | 0h | |
| 5-3 | REFSRC | R/W | 0h | These bits select the reference source for the comparator.<br>0h = Reference voltage generator is disabled (local reference buffer as well as DAC).<br>1h = VDDA selected as the reference source to DAC and DAC output applied as reference to comparator.<br>2h = VREF selected as reference to DAC and DAC output applied as reference to comparator.<br>If the device supports internal VREF, then configuring CTL0 register of VREF module to 1 selects internal VREF.<br>Otherwise, external VREF is selected.<br>For more details, refer to the Reference Voltage Generator section of the COMP chapter of the TRM.<br>3h = VREF is applied as reference to comparator. DAC is switched off.<br>If the device supports internal VREF, then configuring CTL0 register of VREF module to 1 selects internal VREF.<br>Otherwise, external VREF is selected.<br>For more details, refer to the Reference Voltage Generator section of the COMP chapter of the TRM.<br>5h = VDDA is used as comparator reference.<br>6h = VDDA selected as the reference source to DAC and DAC output applied as reference to comparator.<br>Note: This feature is present only in certain devices. Please consult the device-specific datasheet to confirm if this feature is present.<br>7h = VDDA is used as the source of comparator.<br>Note: This feature is present only in certain devices. Please consult the device-specific datasheet to confirm if this feature is present. |
| 2-1 | RESERVED | R | 0h | |
| 0 | REFMODE | R/W | 0h | This bit requests ULP_REF bandgap operation in fast mode(static) or low power mode (sampled). The local reference buffer and 8-bit DAC inside comparator module are also configured accordingly. Fast mode operation offers higher accuracy but consumes higher current. Low power operation consumes lower current but with relaxed reference voltage accuracy. Comparator requests for reference voltage from ULP_REF only when REFLVL > 0.<br>0h = ULP_REF bandgap, local reference buffer and 8-bit DAC inside comparator operate in static mode.<br>1h = ULP_REF bandgap, local reference buffer and 8-bit DAC inside comparator operate in sampled mode. |



Copyright © 2026 Texas Instruments Incorporated

### 19.3.25 CTL3 Register (Offset = 110Ch) [Reset = 00000000h]

CTL3 is shown in Figure 19-37 and described in Table 19-31.

Return to the Summary Table.

Control 3 register.

**Figure 19-37. CTL3 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | DACCODE1 | | | | | | | | RESERVED | | | | | | | | DACCODE0 | | | | | | | |
| R-0h | | | | | | | | R/W-0h | | | | | | | | R-0h | | | | | | | | R/W-0h | | | | | | | |

**Table 19-31. CTL3 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | RESERVED | R | 0h | |
| 23-16 | DACCODE1 | R/W | 0h | This is the second 8-bit DAC code. When the DAC code is 0x0 the DAC output will be selected reference voltage x 1/256 V. When the DAC code is 0xFF the DAC output will be selected reference voltage x 255/256 V.<br>0h = Minimum DAC code value<br>FFh = Minimum DAC code value |
| 15-8 | RESERVED | R | 0h | |
| 7-0 | DACCODE0 | R/W | 0h | This is the first 8-bit DAC code. When the DAC code is 0x0 the DAC output will be selected reference voltage x 1/256 V. When the DAC code is 0xFF the DAC output will be selected reference voltage x 255/256 V.<br>0h = Minimum DAC code value<br>FFh = Minimum DAC code value |



Copyright © 2026 Texas Instruments Incorporated

### 19.3.26 STAT Register (Offset = 1120h) [Reset = 00000000h]

STAT is shown in Figure 19-38 and described in Table 19-32.

Return to the Summary Table.

Status register.

**Figure 19-38. STAT Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | | | | OUT |
| R-0h | | | | | | | | | | | | | | | R-0h |

**Table 19-32. STAT Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | OUT | R | 0h | This bit reflects the value of the comparator output. Writing to this bit has no effect on the comparator output.<br>0h = Comparator output is low<br>1h = Comparator output is high |



Copyright © 2026 Texas Instruments Incorporated

Chapter 20
# DAC

The DAC module is a 12-bit voltage-output digital-to-analog converter (DAC). This chapter describes the operation of the DAC module.


**20.1 DAC Introduction** ..... 1161
**20.2 DAC Operation** ..... 1162
**20.3 DAC12 Registers** ..... 1171





Copyright © 2026 Texas Instruments Incorporated

## 20.1 DAC Introduction

The DAC module is a 12-bit voltage-output DAC. The DAC can be configured in 8-bit or 12-bit resolution setting and can be used in conjunction with the DMA controller.

Features of the DAC include:

- 8-bit or 12-bit voltage-output resolution
- Straight binary or twos-complement data format
- Internal sample time generator for generation of predefined sampling rates
- Two hardware triggers from event fabric for D/A conversion
- 4x12-bits internal FIFO for data flow rate control
- Operation with DMA controller without CPU Intervention
- Selectable voltage reference options
- Output internally connected to analog modules like OPA, ADC and COMP
- Self-calibration option for offset error correction

Figure 20-1 shows the block diagram of DAC module.

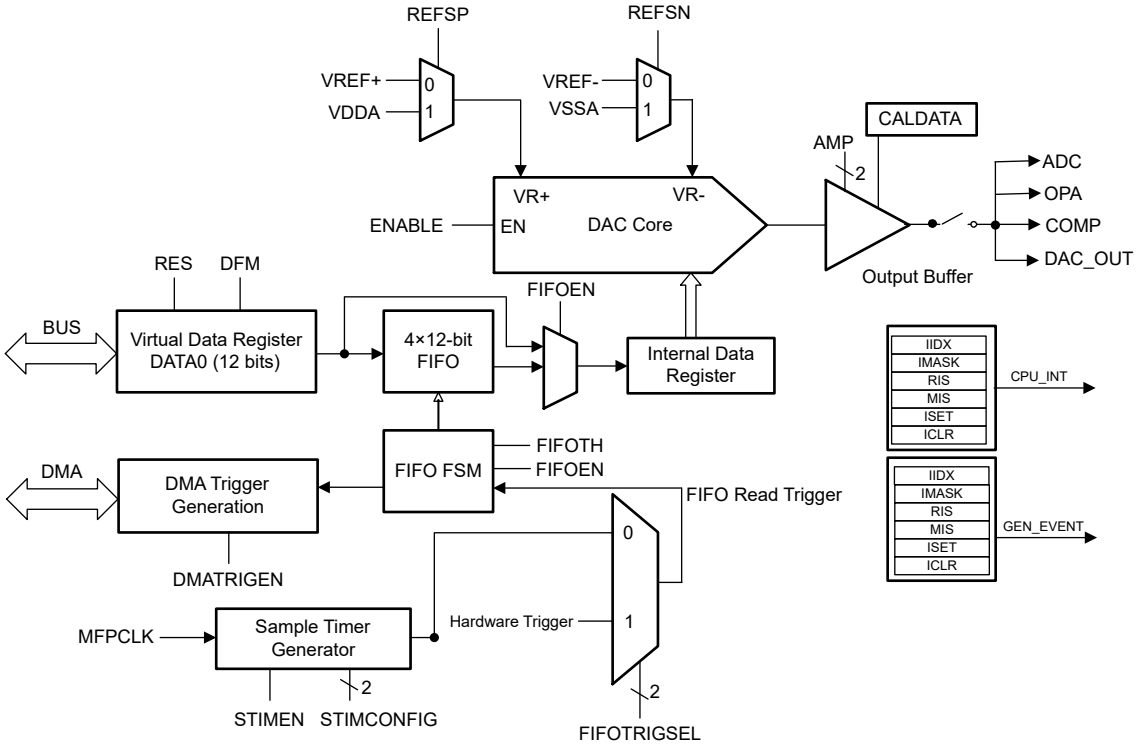



**Figure 20-1. DAC Block Diagram**



Copyright © 2026 Texas Instruments Incorporated

## 20.2 DAC Operation

The DAC module is configured with user software. The setup and operation of the DAC is discussed in the following sections.

### *20.2.1 DAC Core*

The DAC can be configured to operate in 8-bit or 12-bit resolution setting using the CTL0.RES bit. The CTL0.DFM bit can be used to select the data format to be straight-binary or twos-complement. When using straight-binary data format, the formula for the output voltage is given in Table 20-1.

**Table 20-1. DAC Full-Scale Range**

| Resolution | CTL0.RES | Output Voltage Formula |
|---|---|---|
| 12-bit | 1 | $V_{out} = V_{ref} \times (DATA\_VALUE / 4096)$ |
| 8-bit | 0 | $V_{out} = V_{ref} \times (DATA\_VALUE / 256)$ |

- In 8-bit resolution setting, the maximum useable value for DATA_VALUE in the DATA0 register is 255 (0xFF).
- In 12-bit resolution setting, the maximum useable value for DATA_VALUE in the DATA0 register is 4095(0xFFF).

The DAC can be enabled by setting the CTL0.ENABLE bit . When the DAC is enabled, the DAC core and output buffer start to settle and the module ready interrupt condition (MODRDYIFG) is generated once to indicate that the DAC is ready for use.

**Note**

The DAC must be enabled after the configuration of all control registers as required in the application. If a configuration change is required when the DAC is running, the DAC must be disabled first and re-enabled after the new configuration is programmed into the control registers. Any change in control registers while the DAC is running can cause unpredictable results.

### *20.2.2 DAC Output*

DAC output can be connected to internal analog modules OPA, ADC & COMP or DAC_OUT pin by setting CTL1.OPS bit. Reset value of bits CTL1.OPS is 0. There are the following scenarios:

- If CTL1.OPS bit is set, DAC output is connected to OPA, ADC, COMP and DAC_OUT.
- If CTL1.OPS bit is not set, there is no DAC output. The pin DAC_OUT can be connected to ADC, OPA or COMP.

**Note**

The CTL1.OPS bit must be set as 0 when pin DAC_OUT is configured as the input for ADC, OPA or COMP.

### *20.2.3 DAC Voltage Reference*

The CTL1.REFSP and CTL1.REFSN bits select the voltage reference for the DAC from the following options.

- Analog supply (VDDA) (see *Power Management (PMU)*).
- An external reference voltage that can be provided on VREF+ and VREF- pins of the device.
- The internal reference voltage from the output of VREF module.

### *20.2.4 DAC Output Buffers*

The voltage output buffers of the DAC can be configured to be enabled or disabled by setting the bit CTL1.AMPEN. When disabled, the output amplifier can be set either to ground or high impedance using the bit CTL1.AMPHIZ.



Copyright © 2026 Texas Instruments Incorporated

### 20.2.5 DAC Data Formats

The DAC supports both straight-binary and twos-complement data formats. When using straight-binary data format, the full-scale output value is 0xFFF in 12-bit resolution setting and 0xFF in 8-bit resolution settings (see Figure 20-2).

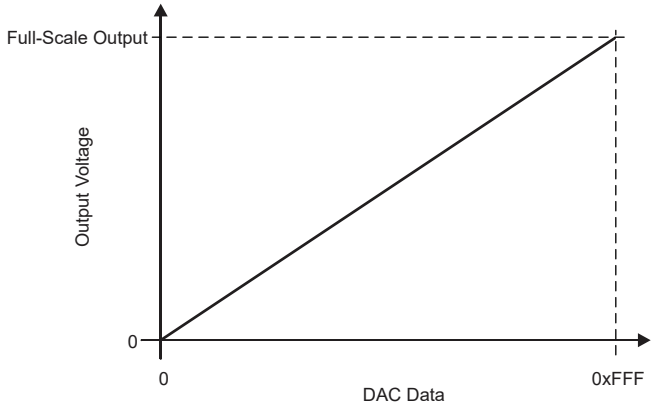



**Figure 20-2. Output Voltage vs DAC Data, 12-Bit Straight-Binary Mode**

When using twos-complement data format, the range is shifted so that a DAC data value of 0x800 in 12-bit resolution setting and 0x80 in 8-bit resolution settings results in a zero output voltage, 0x0 is the mid-scale output voltage, and 0x7FF in 12-bit resolution setting and 0x7F in 8-bit resolution settings is the full-scale voltage output (see Figure 20-3).

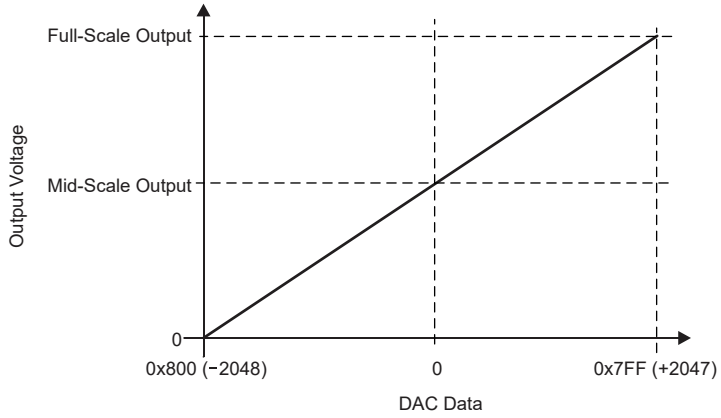



**Figure 20-3. Output Voltage vs DAC Data, 12-Bit Twos-Complement Mode**

### 20.2.6 Sample Time Generator

There is a sample time generator available in the DAC module. This can be used to generate DAC conversion trigger for certain predefined sampling rates. The sample time generator is disabled by default and it can be enabled by setting the CTL3.STIMEN bit. When the sample time generator is enabled, it automatically requests for the MFPCLK (4MHz) and generates trigger according to the programmed trigger rate. The MFPCLK needs to be enabled in order to operate the sample time generator. The predefined sampling rates supported are 500sps, 1Ksps, 2Ksps, 4Ksps, 8Ksps, 16Ksps, 100Ksps, 200Ksps, 500Ksps, and 1Msps. These trigger rates are programmed by setting the bit CTL3.STIMCONFIG. The predefined sampling rates are given in Table 20-2

**Table 20-2. Predefined Sampling Rates**

| CTL3.STIMCONFIG | Sampling Rate |
|---|---|
| 0 | 500sps |
| 1 | 1Ksps |



Copyright © 2026 Texas Instruments Incorporated

**Table 20-2. Predefined Sampling Rates (continued)**

| CTL3.STIMCONFIG | Sampling Rate |
|---|---|
| 2 | 2Ksps |
| 3 | 4Ksps |
| 4 | 8Ksps |
| 5 | 16Ksps |
| 6 | 100Ksps |
| 7 | 200Ksps |
| 8 | 500Ksps |
| 9 | 1Msps |

### *20.2.7 DAC FIFO Structure*

The DAC module has an 4x12-bits FIFO. By default the FIFO operation is disabled and it is enabled by setting the CTL2.FIFOEN bit. When the CPU or DMA controller writes into the memory mapped data register DATA0, the data is written into FIFO at the location pointed by the write pointer. The data is read from FIFO and loaded to internal DAC data register when the FIFO read trigger is asserted. The data written into the FIFO can be in binary or twos complement format.

#### 20.2.7.1 Loading Data From FIFO to Internal DAC Data Register

The CTL2.FIFOTRIGSEL register is used to select the trigger source for loading data from FIFO into internal DAC data register. There are three triggers sources:

- The sample time generator output can be selected as the trigger source.
- Hardware Trigger, listening for events published through GEN_EVENT configured in the register FSUB_0.

For the hardware trigger source, the DAC module can subscribe to the event through the event fabric. Setting CTR2.FIFOTRIGSEL as 1 is to subscribe to the event on FSUB_0.

When the sample time generator is used, the MFPCLK is used for trigger generation and for transferring the data from FIFO into internal DAC data register. In this scenario, the DAC sampling rate is predictable and has no connection with ULPCLK.

When the sample time generator is not used, the ULPCLK is used for transferring the data from FIFO into internal DAC data register upon the external event trigger. For a predictable DAC sampling rate, the application software must manage ULPCLK frequency to be deterministic. The sampling rate is affected if there is any change in ULPCLK frequency during DAC operation in this scenario.

Whenever the selected trigger source is asserted, the data in FIFO pointed by the read pointer is read and loaded into the internal DAC data register for conversion.

### *20.2.8 DAC Operation With DMA Controller*

The DMA controller can move data into the DAC from the SRAM. Program the CTL2.DMATRIGEN and CTL2.FIFOEN bits to enable DAC operation with the DMA controller. When the DMA trigger mechanism is enabled in the DAC along with the FIFO, the FIFO hardware state machine evaluates the available empty locations in the FIFO and generates a DMA trigger. The FIFO threshold level can be selected using the CTL2.FIFOTH bits. The available FIFO level settings are empty, 1/4, 1/2, and 3/4.

During operation, empty locations in FIFO are continuously evaluated by the FIFO hardware state machine and are compared with the selected FIFO threshold level. The DAC generates a trigger to the DMA when the number of empty locations matches the programmed FIFO threshold.

The DAC-to-DMA controller interface is shown in Figure 20-4 and explained in the following sections.



Copyright © 2026 Texas Instruments Incorporated

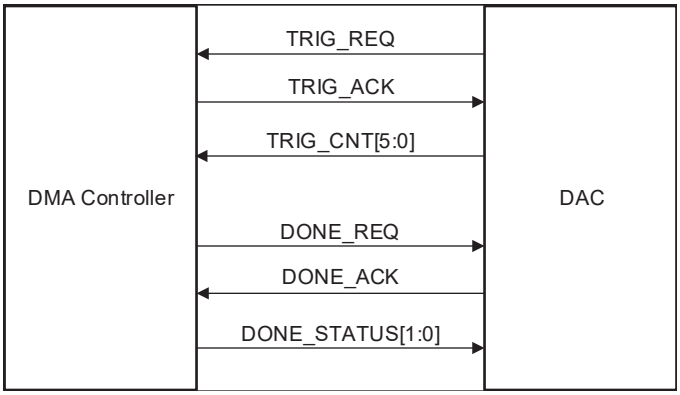



**Figure 20-4. DAC Interface With DMA Controller**

#### 20.2.8.1 DMA Trigger Interface

There is a defined protocol between DAC and DMA:

1. The DAC generates a trigger TRIG_REQ to the DMA together with the trigger count TRIG_CNT. TRIG_CNT is a 6-bit signal that indicates the number of empty locations available in the DAC FIFO for the DMA to fill. The value of TRIG_CNT is in accordance with the FIFO element size.
2. When TRIG_REQ is asserted, the DMA provides a trigger acknowledge signal TRIG_ACK back to the DAC.

The trigger event and TRIG_ACK are routed through the event fabric and follow the standard 4-way request, acknowledge handshake protocol.

#### 20.2.8.2 DMA Status Interface

When the DAC trigger is captured by DMA and trigger acknowledgment has been provided, both the DAC and the DMA are in triggered state. In this state, the DAC must not generate any new DMA trigger. When DMA starts to service the DAC, the DMA can perform data transfers to the actual value indicated by the trigger count or to a value less than the trigger count (it can be even no transfer). This is not deterministic and depends on factors like DMA bandwidth and channel priorities.

When the DMA performs a certain number of transfers, it asserts a DMA done signal DONE_REQ to the DAC along with the DMA status side band signal DONE_STATUS. The DAC module provides a DMA done acknowledgment DONE_ACK as a response. The DMA done event and DONE_ACK are routed through the event fabric and follow the standard 4-way request, acknowledge handshake protocol.

#### 20.2.8.3 DMA Trigger Generation Scheme

DMA status signal DONE_STATUS value 0 indicates that DMA has more data elements to transfer to DAC. When DAC observes DONE_STATUS to be 0, it performs fresh assessment of the number of empty locations in the FIFO and if that matches with the programmed FIFO threshold level it makes the next TRIG_REQ along with TRIG_CNT. Then the whole sequence of operations mentioned above repeats.

When the DMA has finished transferring the complete data then it asserts DMA done signal DONE_REQ along with DMA status signal DONE_STATUS with a non-zero value. When DAC captures DMA status signal with a non-zero value it sets the DMADONE interrupt flag. This interrupt flag can be used to generate an interrupt to the CPU that owns the DAC for the appropriate action to be taken. CTL2.DMATRIGEN bit should be cleared by software to stop further DMA triggers. This interrupt flag can be used to generate an interrupt to the CPU that owns the DAC for the appropriate action to be taken.

The following section explain the details of using the DAC and DMA with the FIFO enabled

**DAC-DMA Operation in FIFO Mode (CTL2.FIFOEN=1) and Sample Time Generator enabled**

- Configure the clock, the MFPCLK should be enabled as Sample Time Generator will be used
- Configure DMA source and destination address. Destination address is FIFO data register DATA0



Copyright © 2026 Texas Instruments Incorporated

- Choose the transfer mode and DMA transfer size between byte (8-bit), short word (16-bit), word (32-bit) or long word (64-bit)
- Set DAC resolution (8-bit or 12-bit)
- Configure FIFO Trigger Threshold via the CTL2.FIFOTH bits which supports ¼, ½ and ¾
- Configure and enable DAC and DMA interrupt
- Use CTL2.FIFOTRIGSEL to configure Sample Time Generator as the trigger source for loading data to FIFO data register. Ensure that the clock configurations of MCLK and ULPCLK are properly set to avoid FIFO underrun issues and distortion in the DAC output
  - Make sure that ULPCLK is equal to MFPCLK to maintain synchronization between the clock domains
  - Adjust the FIFO threshold setting to ½ or ¾ when the source of MCLK is HFCLK or SYSPLL and its frequency is equal to or greater than 32MHz
- Enable DMA Trigger

### *20.2.9 DAC Operation With CPU*

The CPU can directly load data into DAC for conversion which has the following scenarios:

- **Fixed DC voltage output and FIFO disabled:** CPU writes the data into the data register DATA0. This data will pass on to the internal DAC data register for conversion.
- **Fixed DC voltage output and FIFO enabled:** CPU writes the data into the data register DATA0 which is read from FIFO and loaded to internal DAC data register when the FIFO read trigger is asserted.
- **Fixed AC voltage output and FIFO enabled:** CPU writes the data into the data register DATA0 which is read from FIFO and loaded to internal DAC data register when the FIFO read trigger is asserted.

**Note**

1. If a fixed AC voltage output is desired, the FIFO must be enabled.
2. CPU must write into the data register DATA0 with byte-0 for 8-bit DAC resolution and lower half-word (bytes 0 and 1) for 12-bit DAC resolution.
3. DMA trigger generation mechanism must be kept disabled when CPU is used to load data into DAC.

For the FIFO read triggers, it is possible to use the sample time generator or the external event available for the event fabric.

#### 20.2.9.1 Interrupt conditions for DAC operation with CPU

When the FIFO is enabled and CPU is used to load data into DAC, there are certain FIFO specific interrupt conditions generated in the DAC. These are FIFO empty, FIFO full, FIFO 1/4 empty, FIFO 1/2 empty and FIFO 3/4 empty. The reset value of all these interrupt flags is 0. When the DAC module is enabled along with the FIFO, the internal logic evaluates the FIFO status and updates the FIFO specific interrupt flags accordingly. These interrupt conditions can be enabled appropriately by software so that CPU is notified about the empty locations available in FIFO for loading data into DAC.

### *20.2.10 Data Register Format*

The DAC supports programmable resolution setting and data format . The figures given in this section take DATA0 register as an example and describe the format in which CPU or DMA controller is expected to write data into DATA0 register for one data sample.

The CPU always writes only into DATA0 register. If FIFO is disabled then CPU can write into only byte-0 for 8-bit DAC resolution and lower half-word (bytes 0 and 1) for 12-bit DAC resolution. The upper half-word of DATA0 register is never written.

When the FIFO is enabled, the CPU can write up to 4 data samples into DATA0 register for 8-bit resolution or 12-bit resolution.



Copyright © 2026 Texas Instruments Incorporated

**Note**

If changing between 8-bit and 12-bit modes, rewrite to the data register (DATA0) to achieve the intended voltage output.

Figure 20-5 shows the data format in DATA0 register for 8-bit binary or two's complement data.

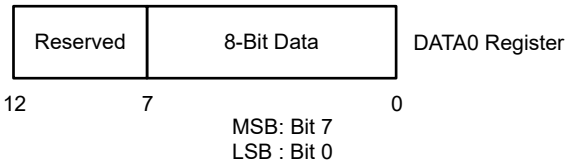



**Figure 20-5. 8-Bit Data**

### *20.2.11 DAC Output Amplifier Offset Calibration*

The offset voltage of the DAC output amplifier can be positive or negative. When the offset is negative, the output amplifier attempts to drive the voltage negative but cannot do so. The output voltage remains at zero until the DAC digital input produces a sufficient positive output voltage to overcome the negative offset voltage, resulting in the transfer function in Figure 20-6.

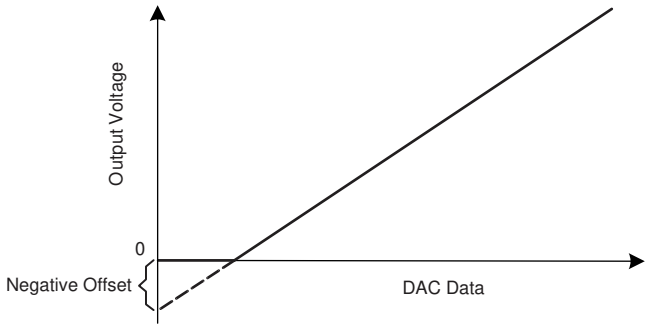



**Figure 20-6. Negative Offset**

When the output amplifier has a positive offset, a digital input of zero does not result in a zero output voltage. The DAC output voltage reaches the maximum output level before the DAC data reaches the maximum code (see Figure 20-7).



Copyright © 2026 Texas Instruments Incorporated

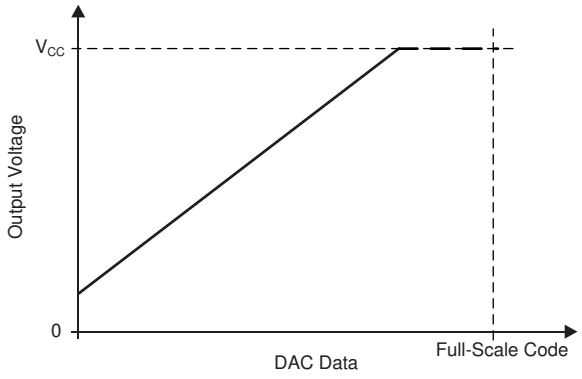



**Figure 20-7. Positive Offset**

The DAC can self-calibrate the offset voltage of the output amplifier. The application can perform self-calibration of the DAC output buffer to achieve the offset error given in the electrical specification. The CALCTL.CALON bit is used to initiate the offset error calibration. The DAC output is high impedance while the calibration is active. When the calibration is complete, the CALCTL.CALON bit is automatically reset. Software must configure the CTL1.AMP bits before calibration.

For best calibration results, activity on the device pins and in the CPU must be minimized during calibration. Calibration data is loaded into CALDATA; when DAC self-calibration is initiated using the CALCTL.CALON bit, the CALDATA register is continuously updated during the course of calibration. Read the CALDATA register only after the CALCTL.CALON bit is cleared, otherwise incorrect values can be read. The DAC calibration data format is twos complement. Only the lowest byte is used and the upper bytes have no effect on the offset calibration.

**Note**

Recalibrate the DAC output buffer after changing the reference voltage level in the VREF module when DAC is in use.

### *20.2.12 Interrupt and Event Support*

The DAC module contains three event publishers and two event subscribers. One event publisher (CPU_INT) manages DAC interrupt requests (IRQs) to the CPU subsystem via a static event route. The second event publisher (GEN_EVENT) can be used to setup the generic event publisher via Generic route. A direct DMA trigger can be used as a DAC-to-DMA trigger to send DAC events to the DMA through DMA event route.

The event subscriber (FSUB_0) can be used to subscribe to events which are published to the event fabric through a generic event route channel.

The DAC events are summarized in Table 20-3.

**Table 20-3. DAC Events**

| Event | Type | Source | Destination | Route | Configuration | Functionality |
|---|---|---|---|---|---|---|
| CPU interrupt | Publisher | DAC | CPU Subsystem | Static route | CPU_INT registers | Fixed interrupt route from DAC to CPU |
| Generic publisher event | Publisher | DAC | Other peripherals | Generic route (FPUB_0) | GEN_EVENT and FPUB_0 registers | Trigger generic event channel from DAC |
| DMA Trigger event | Publisher | DAC | DMA | DMA route | DMA_TRIG and FPUB_1 registers | Fixed trigger route from ADC to DMA |



Copyright © 2026 Texas Instruments Incorporated

**Table 20-3. DAC Events (continued)**

| Event | Type | Source | Destination | Route | Configuration | Functionality |
|---|---|---|---|---|---|---|
| Generic subscriber event | Subscriber | Other peripherals | DAC | Generic route (FSUB_0) | FSUB_0 | DAC subscription to generic event channel |

#### 20.2.12.1 CPU Interrupt Event Publisher (CPU_INT)

The DAC module provides different interrupt sources which can be configured to source a CPU interrupt event. In order of decreasing interrupt priority, the CPU interrupt events from the DAC are:

**Table 20-4. DAC Interrupt Event Conditions (CPU_INT)**

| Index (IIDX) | Name | Description |
|---|---|---|
| 0x0 | NO_INTR | No bit set (IIDX.STAT = 0) means there is no pending interrupt request |
| 0x2 | MODRDYIFG | The DAC output is settled after it is turned on using ENABLE in the CTL0 register |
| 0x9 | FIFOFULLIFG | FIFO full interrupt flag is set when the FIFO is full |
| 0xA | FIFO1B4IFG | FIFO one fourth empty interrupt flag is set when one fourth of FIFO locations are empty. |
| 0xB | FIFO1B2IFG | FIFO half empty interrupt flag is set when half of the FIFO locations are empty. |
| 0xC | FIFO3B4IFG | FIFO three fourth empty interrupt flag is set when all the locations in FIFO are empty. |
| 0xD | FIFOEMPTYIFG | FIFO empty interrupt flag is set when all data in the FIFO have been shifted out. |
| 0xE | FIFOURUNIFG | FIFO underrun flag is set when the FIFO read trigger is asserted while the FIFO is empty. |
| 0xF | DMADONEIFG | DMA done interrupt flag is set when the DMA data transfer of programmed block size is completed |

The CPU interrupt event configuration is managed with the CPU_INT event management registers. Interrupt (RIS) flags are cleared upon software reading the IIDX register or writing to the respective ICLR register bits.

See *Section 7.2.5* section for guidance on configuring the event registers for CPU interrupts.

#### 20.2.12.2 Generic Event Publisher (GEN_EVENT)

The DAC module provides the following interrupt sources, one of which can be configured to publish an event to a generic event route channel.

**Table 20-5. DAC Generic Event Publisher Conditions (GEN_EVENT)**

| Index (IIDX) | Name | Description |
|---|---|---|
| 0x0 | NO_INTR | No bit set (IIDX.STAT = 0) means there is no pending interrupt request |
| 0x2 | MODRDYIFG | Module ready interrupt once DAC is enabled and DAC output is settled |
| 0x9 | FIFOFULLIFG | FIFO full interrupt flag is set when the FIFO is full |
| 0xA | FIFO1B4IFG | FIFO one fourth empty interrupt flag is set when one fourth of FIFO locations are empty. |
| 0xB | FIFO1B2IFG | FIFO half empty interrupt flag is set when half of the FIFO locations are empty. |
| 0xC | FIFO3B4IFG | FIFO three fourth empty interrupt flag is set when all the locations in FIFO are empty. |
| 0xD | FIFOEMPTYIFG | FIFO empty interrupt flag is set when all data in the FIFO have been shifted out. |
| 0xE | FIFOURUNIFG | FIFO underrun flag is set when the FIFO read trigger is asserted while the FIFO is empty. |
| 0xF | DMADONEIFG | DMA done interrupt flag is set when the DMA data transfer of programmed block size is completed |

The generic event publisher configuration is managed with the GEN_EVENT event management registers. Interrupt (RIS) flags are cleared based on acknowledgment (ACK) signal from the subscriber module received over the event fabric. See [Using Event Registers] for guidance on configuring the event registers for generic event publishers. The generic event channel which GEN_EVENT is to publish to must be selected by writing the target generic channel ID to the **FPUB_0** register in the DAC. See [Generic Event Route] for guidance on configuring generic event routes.If this publisher is not used in an application, the FPUB_0 register can be left in



Copyright © 2026 Texas Instruments Incorporated

a disconnected state (set equal to zero) and no events should be unmasked through the MIS register in the ADC GEN_EVENT register set.

#### 20.2.12.3 DMA Trigger Event Publisher

The DAC module provides different interrupt sources which can be configured to source the DMA trigger. In order of decreasing interrupt priority, the DMA trigger events from the DAC are given as the following table. When the DMA channel is needed by the DAC, the DMA trigger should be unmasked in the IMASK register.

**Note**

The DMA trigger from the DAC is a direct trigger in hardware and not a part of the DMA_TRIG group of configurable event management registers.

To configure the DMA controller to trigger from the DAC, please see Section 20.2.8.

**Table 20-6. DAC DMA Trigger Event Publisher Conditions**

| Index (IIDX) | Name | Description |
|---|---|---|
| 0x0 | NO_INTR | No bit set (IIDX.STAT = 0) means there is no pending interrupt request |
| 0x2 | MODRDYIFG | Module ready interrupt once DAC is enabled and DAC output is settled |
| 0x9 | FIFOFULLIFG | FIFO full interrupt flag is set when the FIFO is full |
| 0xA | FIFO1B4IFG | FIFO one fourth empty interrupt flag is set when one fourth of FIFO locations are empty. |
| 0xB | FIFO1B2IFG | FIFO half empty interrupt flag is set when half of the FIFO locations are empty. |
| 0xC | FIFO3B4IFG | FIFO three fourth empty interrupt flag is set when all the locations in FIFO are empty. |
| 0xD | FIFOEMPTYIFG | FIFO empty interrupt flag is set when all data in the FIFO have been shifted out. |
| 0xE | FIFOURUNIFG | FIFO underrun flag is set when the FIFO read trigger is asserted while the FIFO is empty. |
| 0xF | DMADONEIFG | DMA done interrupt flag is set when the DMA data transfer of programmed block size is completed |

#### 20.2.12.4 Generic Event Subscriber (FSUB_0)

The DAC module can trigger FIFO transfers by receiving events routed through a generic channel from other peripherals. When the event is received, the data in the FIFO is loaded into internal DAC data register. Refer to Generic Event Route and Peripheral to Peripheral Event for details on how the generic event route works. When the channel to be used is determined, and both the publisher and subscriber ports for the peripherals to connect are known, use the steps below to establish the event connection.

In this example, a GPIO-triggered DAC FIFO transfer is configured, using GPIO Port A to publish an event to generic channel 1, with DAC subscribing to generic channel 1 as a start-of-conversion trigger.

1. Configure the GEN_EVENT registers of GPIO Port A to set the event request based on the appropriate event (for example, a DIN rise event).
2. Store 0x1 into the FPUB_0 register of GPIO Port A to publish the GPIO event selected by the GEN_EVENT registers to generic route channel 1. Channel 1 must not be in use by another peripheral.
3. Store 0x1 the FSUB_0 register of DAC so that DAC is listening for events published by the timer to channel 1.
4. Configure DAC to trigger from the subscriber port according to the configuration instructions in Loading Data From FIFO to Internal DAC Data Register.
5. Configure and enable the appropriate GPIO pin to monitor input voltage events.



Copyright © 2026 Texas Instruments Incorporated

## 20.3 DAC12 Registers

Table 20-7 lists the memory-mapped registers for the DAC12 registers. All register offset addresses not listed in Table 20-7 should be considered as reserved locations and the register contents should not be modified.

**Table 20-7. DAC12 Registers**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 400h | FSUB_0 | Subscriber Port 0 | | Go |
| 444h | FPUB_1 | Publisher port 1 | | Go |
| 800h | PWREN | Power enable | | Go |
| 804h | RSTCTL | Reset Control | | Go |
| 814h | STAT | Status Register | | Go |
| 1020h | IIDX | Interrupt index | CPU_INT | Go |
| 1028h | IMASK | Interrupt mask | CPU_INT | Go |
| 1030h | RIS | Raw interrupt status | CPU_INT | Go |
| 1038h | MIS | Masked interrupt status | CPU_INT | Go |
| 1040h | ISET | Interrupt set | CPU_INT | Go |
| 1048h | ICLR | Interrupt clear | CPU_INT | Go |
| 1050h | IIDX | Interrupt index | GEN_EVENT | Go |
| 1058h | IMASK | Interrupt mask | GEN_EVENT | Go |
| 1060h | RIS | Raw interrupt status | GEN_EVENT | Go |
| 1068h | MIS | Masked interrupt status | GEN_EVENT | Go |
| 1070h | ISET | Interrupt set | GEN_EVENT | Go |
| 1078h | ICLR | Interrupt clear | GEN_EVENT | Go |
| 10E0h | EVT_MODE | Event Mode | | Go |
| 10FCh | DESC | Module Description | | Go |
| 1100h | CTL0 | Control 0 | | Go |
| 1110h | CTL1 | Control 1 | | Go |
| 1120h | CTL2 | Control 2 | | Go |
| 1130h | CTL3 | Control 3 | | Go |
| 1140h | CALCTL | Calibration control | | Go |
| 1160h | CALDATA | Calibration data | | Go |
| 1200h | DATA0 | Data 0 | | Go |

Complex bit access types are encoded to fit into small table cells. Table 20-8 shows the codes that are used for access types in this section.

**Table 20-8. DAC12 Access Type Codes**

| Access Type | Code | Description |
|---|---|---|
| **Read Type** | | |
| R | R | Read |
| RH | R<br>H | Read<br>Set or cleared by hardware |
| **Write Type** | | |
| K | K | Write protected by a key |
| W | W | Write |
| WK | W<br>K | Write<br>Write protected by a key |
| **Reset or Default Value** | | |



Copyright © 2026 Texas Instruments Incorporated

**Table 20-8. DAC12 Access Type Codes (continued)**

| Access Type | Code | Description |
|---|---|---|
| *-n* | | Value after reset or the default value |



Copyright © 2026 Texas Instruments Incorporated

### 20.3.1 FSUB_0 (Offset = 400h) [Reset = 00000000h]

FSUB_0 is shown in Figure 20-8 and described in Table 20-9.

Return to the Summary Table.

Subscriber port 0

**Figure 20-8. FSUB_0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R/W- | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | CHANID | | | |
| R/W- | | | | | | | | | | | | R/W-0h | | | |

**Table 20-9. FSUB_0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R/W | 0h | |
| 3-0 | CHANID | R/W | 0h | 0 = disconnected.<br>others = connected to channel_ID = CHANID.<br>0h = A value of 0 specifies that the event is not connected<br>Fh = Consult your device data sheet as the actual allowed maximum may be less than 15. |



Copyright © 2026 Texas Instruments Incorporated

### 20.3.2 FPUB_1 (Offset = 444h) [Reset = 00000000h]

FPUB_1 is shown in Figure 20-9 and described in Table 20-10.

Return to the Summary Table.

Publisher port 1

**Figure 20-9. FPUB_1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R/W- | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | CHANID | | | |
| R/W- | | | | | | | | | | | | R/W-0h | | | |

**Table 20-10. FPUB_1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R/W | 0h | |
| 3-0 | CHANID | R/W | 0h | 0 = disconnected.<br>others = connected to channel_ID = CHANID.<br>0h = A value of 0 specifies that the event is not connected<br>Fh = Consult your device data sheet as the actual allowed maximum may be less than 15. |



Copyright © 2026 Texas Instruments Incorporated

### 20.3.3 PWREN (Offset = 800h) [Reset = 00000000h]

PWREN is shown in Figure 20-10 and described in Table 20-11.

Return to the Summary Table.

Register to control the power state

**Figure 20-10. PWREN**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R/W- | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R/W- | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | ENABLE |
| R/W- | | | | | | | K-0h |

**Table 20-11. PWREN Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | KEY to allow Power State Change<br>26h = KEY to allow write access to this register |
| 23-1 | RESERVED | R/W | 0h | |
| 0 | ENABLE | K | 0h | Enable the power<br>KEY must be set to 26h to write to this bit.<br>0h = Disable Power<br>1h = Enable Power |



Copyright © 2026 Texas Instruments Incorporated

### 20.3.4 RSTCTL (Offset = 804h) [Reset = 00000000h]

RSTCTL is shown in Figure 20-11 and described in Table 20-12.

Return to the Summary Table.

Register to control reset assertion and de-assertion

**Figure 20-11. RSTCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| W- | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| W- | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | RESETSTKYCLR | RESETASSERT |
| W- | | | | | | WK-0h | WK-0h |

**Table 20-12. RSTCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | Unlock key<br>B1h = KEY to allow write access to this register |
| 23-2 | RESERVED | W | 0h | |
| 1 | RESETSTKYCLR | WK | 0h | Clear the RESETSTKY bit in the STAT register<br>KEY must be set to B1h to write to this bit.<br>0h = Writing 0 has no effect<br>1h = Clear reset sticky bit |
| 0 | RESETASSERT | WK | 0h | Assert reset to the peripheral<br>KEY must be set to B1h to write to this bit.<br>0h = Writing 0 has no effect<br>1h = Assert reset |



Copyright © 2026 Texas Instruments Incorporated

### 20.3.5 STAT (Offset = 814h) [Reset = 00000000h]

STAT is shown in Figure 20-12 and described in Table 20-13.

Return to the Summary Table.

peripheral enable and reset status

**Figure 20-12. STAT**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R- | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | RESETSTKY |
| R- | | | | | | | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R- | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | |
| R- | | | | | | | |

**Table 20-13. STAT Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-17 | RESERVED | R | 0h | |
| 16 | RESETSTKY | R | 0h | This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register<br>0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register<br>1h = The peripheral was reset since the last bit clear |
| 15-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 20.3.6 IIDX (Offset = 1020h) [Reset = 00000000h]

IIDX is shown in Figure 20-13 and described in Table 20-14.

Return to the Summary Table.

Interrupt index register. This read-only register provides the interrupt index of the pending interrupt with the highest priority. It also indicates if no interrupt is pending. The priority order is fixed: lower index equals higher priority. Alternatively, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred.
On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flags in the RIS and MIS are cleared as well. After a read from the CPU (not from the debug interface), the register must be updated with the next highest priority interrupt or indicate that no interrupt is pending. Only interrupts which are selected via IMASK are indicated.

**Figure 20-13. IIDX**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | STAT | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | |

**Table 20-14. IIDX Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3-0 | STAT | R | 0h | Interrupt index status<br>0h = No pending interrupt<br>2h = Module ready interrupt<br>9h = FIFO full interrupt<br>Ah = FIFO one fourth empty interrupt<br>Bh = FIFO half empty interrupt<br>Ch = FIFO three fourth empty interrupt<br>Dh = FIFO empty interrupt<br>Eh = FIFO underrun interrupt<br>Fh = DMA done interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 20.3.7 IMASK (Offset = 1028h) [Reset = 00000000h]

IMASK is shown in Figure 20-14 and described in Table 20-15.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is unmasked. Unmasking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS

**Figure 20-14. IMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | DMADONEIFG | FIFOURUNIFG | FIFOEMPTYIFG | FIFO3B4IFG | FIFO1B2IFG | FIFO1B4IFG | FIFOFULLIFG |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | MODRDYIFG | RESERVED |
| R/W-0h | | | | | | R/W-0h | R/W-0h |

**Table 20-15. IMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-15 | RESERVED | R/W | 0h | |
| 14 | DMADONEIFG | R/W | 0h | Masks DMADONEIFG<br>0h = Interrupt is masked out<br>1h = Interrupt will request an interrupt service routine and corresponding bit in MIS will be set |
| 13 | FIFOURUNIFG | R/W | 0h | Masks FIFOURUNIFG<br>0h = Interrupt is masked out<br>1h = Interrupt will request an interrupt service routine and corresponding bit in MIS will be set |
| 12 | FIFOEMPTYIFG | R/W | 0h | Masks FIFOEMPTYIFG<br>0h = Interrupt is masked out<br>1h = Interrupt will request an interrupt service routine and corresponding bit in MIS will be set |
| 11 | FIFO3B4IFG | R/W | 0h | Masks FIFO3B4IFG<br>0h = Interrupt is masked out<br>1h = Interrupt will request an interrupt service routine and corresponding bit in MIS will be set |
| 10 | FIFO1B2IFG | R/W | 0h | Masks FIFO1B2IFG<br>0h = Interrupt is masked out<br>1h = Interrupt will request an interrupt service routine and corresponding bit in MIS will be set |
| 9 | FIFO1B4IFG | R/W | 0h | Masks FIFO1B4IFG<br>0h = Interrupt is masked out<br>1h = Interrupt will request an interrupt service routine and corresponding bit in MIS will be set |
| 8 | FIFOFULLIFG | R/W | 0h | Masks FIFOFULLIFG<br>0h = Interrupt is masked out<br>1h = Interrupt will request an interrupt service routine and corresponding bit in MIS will be set |
| 7-2 | RESERVED | R/W | 0h | |



Copyright © 2026 Texas Instruments Incorporated

**Table 20-15. IMASK Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 1 | MODRDYIFG | R/W | 0h | Masks MODRDYIFG<br>0h = Interrupt is masked out<br>1h = Interrupt will request an interrupt service routine and corresponding bit in MIS will be set |
| 0 | RESERVED | R/W | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 20.3.8 RIS (Offset = 1030h) [Reset = 00001E00h]

RIS is shown in Figure 20-15 and described in Table 20-16.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Figure 20-15. RIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | DMADONEIFG | FIFOURUNIFG | FIFOEMPTYIFG | FIFO3B4IFG | FIFO1B2IFG | FIFO1B4IFG | FIFOFULLIFG |
| R-0h | R-0h | R-0h | R-1h | R-1h | R-1h | R-1h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | MODRDYIFG | RESERVED |
| R-0h | | | | | | R-0h | R-0h |

**Table 20-16. RIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-15 | RESERVED | R | 0h | |
| 14 | DMADONEIFG | R | 0h | Raw interrupt status for DMADONEIFG<br>0h = DMA done condition did not occur<br>1h = DMA done condition occurred |
| 13 | FIFOURUNIFG | R | 0h | Raw interrupt status for FIFOURUNIFG<br>0h = FIFO underrun condition did not occur<br>1h = FIFO underrun condition occurred |
| 12 | FIFOEMPTYIFG | R | 1h | Raw interrupt status for FIFOEMPTYIFG<br>0h = FIFO empty condition did not occur<br>1h = FIFO empty condition occurred |
| 11 | FIFO3B4IFG | R | 1h | Raw interrupt status for FIFO3B4IFG<br>0h = FIFO three fourth empty condition did not occur<br>1h = FIFO three fourth empty condition occurred |
| 10 | FIFO1B2IFG | R | 1h | Raw interrupt status for FIFO1B2IFG<br>0h = FIFO half empty condition did not occur<br>1h = FIFO half empty condition occurred |
| 9 | FIFO1B4IFG | R | 1h | Raw interrupt status for FIFO1B4IFG<br>0h = FIFO one fourth empty condition did not occur<br>1h = FIFO one fourth empty condition occurred |
| 8 | FIFOFULLIFG | R | 0h | Raw interrupt status for FIFOFULLIFG<br>0h = FIFO full condition did not occur<br>1h = FIFO full condition occurred |
| 7-2 | RESERVED | R | 0h | |
| 1 | MODRDYIFG | R | 0h | Raw interrupt status for MODRDYIFG<br>0h = DAC module ready event did not occur<br>1h = DAC module ready event occurred |
| 0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 20.3.9 MIS (Offset = 1038h) [Reset = 00000000h]

MIS is shown in Figure 20-16 and described in Table 20-17.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 20-16. MIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | DMADONEIFG | FIFOURUNIFG | FIFOEMPTYIFG | FIFO3B4IFG | FIFO1B2IFG | FIFO1B4IFG | FIFOFULLIFG |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | MODRDYIFG | RESERVED |
| R-0h | | | | | | R-0h | R-0h |

**Table 20-17. MIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-15 | RESERVED | R | 0h | |
| 14 | DMADONEIFG | R | 0h | Masked interrupt status for DMADONEIFG<br>0h = DMADONEIFG does not request an interrupt service routine<br>1h = DMADONEIFG requests an interrupt service routine |
| 13 | FIFOURUNIFG | R | 0h | Masked interrupt status for FIFOURUNIFG<br>0h = FIFOURUNIFG does not request an interrupt service routine<br>1h = FIFOURUNIFG requests an interrupt service routine |
| 12 | FIFOEMPTYIFG | R | 0h | Masked interrupt status for FIFOEMPTYIFG<br>0h = FIFOEMPTYIFG does not request an interrupt service routine<br>1h = FIFOEMPTYIFG requests an interrupt service routine |
| 11 | FIFO3B4IFG | R | 0h | Masked interrupt status for FIFO3B4IFG<br>0h = FIFO3B4IFG does not request an interrupt service routine<br>1h = FIFO3B4IFG requests an interrupt service routine |
| 10 | FIFO1B2IFG | R | 0h | Masked interrupt status for FIFO1B2IFG<br>0h = FIFO1B2IFG does not request an interrupt service routine<br>1h = FIFO1B2IFG requests an interrupt service routine |
| 9 | FIFO1B4IFG | R | 0h | Masked interrupt status for FIFO1B4IFG<br>0h = FIFO1B4IFG does not request an interrupt service routine<br>1h = FIFO1B4IFG requests an interrupt service routine |
| 8 | FIFOFULLIFG | R | 0h | Masked interrupt status for FIFOFULLIFG<br>0h = FIFOFULLIFG does not request an interrupt service routine<br>1h = FIFOFULLIFG requests an interrupt service routine |
| 7-2 | RESERVED | R | 0h | |
| 1 | MODRDYIFG | R | 0h | Masked interrupt status for MODRDYIFG<br>0h = MODRDYIFG does not request an interrupt service routine<br>1h = MODRDYIFG requests an interrupt service routine |
| 0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 20.3.10 ISET (Offset = 1040h) [Reset = 00000000h]

ISET is shown in Figure 20-17 and described in Table 20-18.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 20-17. ISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | DMADONEIFG | FIFOURUNIFG | FIFOEMPTYIFG | FIFO3B4IFG | FIFO1B2IFG | FIFO1B4IFG | FIFOFULLIFG |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | MODRDYIFG | RESERVED |
| W-0h | | | | | | W-0h | W-0h |

**Table 20-18. ISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-15 | RESERVED | W | 0h | |
| 14 | DMADONEIFG | W | 0h | Sets DMADONEIFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to DMADONEIFG is set |
| 13 | FIFOURUNIFG | W | 0h | Sets FIFOURUNIFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to FIFOURUNIFG is set |
| 12 | FIFOEMPTYIFG | W | 0h | Sets FIFOEMPTYIFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to FIFOEMPTYIFG is set |
| 11 | FIFO3B4IFG | W | 0h | Sets FIFO3B4IFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to FIFO3B4IFG is set |
| 10 | FIFO1B2IFG | W | 0h | Sets FIFO1B2IFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to FIFO1B2IFG is set |
| 9 | FIFO1B4IFG | W | 0h | Sets FIFO1B4IFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to FIFO1B4IFG is set |
| 8 | FIFOFULLIFG | W | 0h | Sets FIFOFULLIFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to FIFOFULLIFG is set |
| 7-2 | RESERVED | W | 0h | |
| 1 | MODRDYIFG | W | 0h | Sets MODRDYIFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to MODRDYIFG is set |
| 0 | RESERVED | W | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 20.3.11 ICLR (Offset = 1048h) [Reset = 00000000h]

ICLR is shown in Figure 20-18 and described in Table 20-19.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

**Figure 20-18. ICLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | DMADONEIFG | FIFOURUNIFG | FIFOEMPTYIFG | FIFO3B4IFG | FIFO1B2IFG | FIFO1B4IFG | FIFOFULLIFG |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | MODRDYIFG | RESERVED |
| W-0h | | | | | | W-0h | W-0h |

**Table 20-19. ICLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-15 | RESERVED | W | 0h | |
| 14 | DMADONEIFG | W | 0h | Clears DMADONEIFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to DMADONEIFG is cleared |
| 13 | FIFOURUNIFG | W | 0h | Clears FIFOURUNIFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to FIFOURUNIFG is cleared |
| 12 | FIFOEMPTYIFG | W | 0h | Clears FIFOEMPTYIFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to FIFOEMPTYIFG is cleared |
| 11 | FIFO3B4IFG | W | 0h | Clears FIFO3B4IFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to FIFO3B4IFG is cleared |
| 10 | FIFO1B2IFG | W | 0h | Clears FIFO1B2IFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to FIFO1B2IFG is cleared |
| 9 | FIFO1B4IFG | W | 0h | Clears FIFO1B4IFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to FIFO1B4IFG is cleared |
| 8 | FIFOFULLIFG | W | 0h | Clears FIFOFULLIFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to FIFOFULLIFG is cleared |
| 7-2 | RESERVED | W | 0h | |
| 1 | MODRDYIFG | W | 0h | Clears MODRDYIFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to MODRDYIFG is cleared |
| 0 | RESERVED | W | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 20.3.12 IIDX (Offset = 1050h) [Reset = 00000000h]

IIDX is shown in Figure 20-19 and described in Table 20-20.

Return to the Summary Table.

Interrupt index register. This read-only register provides the interrupt index of the pending interrupt with the highest priority. It also indicates if no interrupt is pending. The priority order is fixed: lower index equals higher priority. Alternatively, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred.
On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flags in the RIS and MIS are cleared as well. After a read from the CPU (not from the debug interface), the register must be updated with the next highest priority interrupt or indicate that no interrupt is pending. Only interrupts which are selected via IMASK are indicated.

**Figure 20-19. IIDX**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | | STAT | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | |

**Table 20-20. IIDX Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3-0 | STAT | R | 0h | Interrupt index status<br>0h = No pending interrupt<br>2h = Module ready interrupt<br>9h = FIFO full interrupt<br>Ah = FIFO one fourth empty interrupt<br>Bh = FIFO half empty interrupt<br>Ch = FIFO three fourth empty interrupt<br>Dh = FIFO empty interrupt<br>Eh = FIFO underrun interrupt<br>Fh = DMA done interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 20.3.13 IMASK (Offset = 1058h) [Reset = 00000000h]

IMASK is shown in Figure 20-20 and described in Table 20-21.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is unmasked. Unmasking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS

**Figure 20-20. IMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | DMADONEIFG | FIFOURUNIFG | FIFOEMPTYIFG | FIFO3B4IFG | FIFO1B2IFG | FIFO1B4IFG | FIFOFULLIFG |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | MODRDYIFG | RESERVED |
| R/W-0h | | | | | | R/W-0h | R/W-0h |

**Table 20-21. IMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-15 | RESERVED | R/W | 0h | |
| 14 | DMADONEIFG | R/W | 0h | Masks DMADONEIFG<br>0h = Interrupt is masked out<br>1h = Interrupt will request an interrupt service routine and corresponding bit in MIS will be set |
| 13 | FIFOURUNIFG | R/W | 0h | Masks FIFOURUNIFG<br>0h = Interrupt is masked out<br>1h = Interrupt will request an interrupt service routine and corresponding bit in MIS will be set |
| 12 | FIFOEMPTYIFG | R/W | 0h | Masks FIFOEMPTYIFG<br>0h = Interrupt is masked out<br>1h = Interrupt will request an interrupt service routine and corresponding bit in MIS will be set |
| 11 | FIFO3B4IFG | R/W | 0h | Masks FIFO3B4IFG<br>0h = Interrupt is masked out<br>1h = Interrupt will request an interrupt service routine and corresponding bit in MIS will be set |
| 10 | FIFO1B2IFG | R/W | 0h | Masks FIFO1B2IFG<br>0h = Interrupt is masked out<br>1h = Interrupt will request an interrupt service routine and corresponding bit in MIS will be set |
| 9 | FIFO1B4IFG | R/W | 0h | Masks FIFO1B4IFG<br>0h = Interrupt is masked out<br>1h = Interrupt will request an interrupt service routine and corresponding bit in MIS will be set |
| 8 | FIFOFULLIFG | R/W | 0h | Masks FIFOFULLIFG<br>0h = Interrupt is masked out<br>1h = Interrupt will request an interrupt service routine and corresponding bit in MIS will be set |
| 7-2 | RESERVED | R/W | 0h | |



Copyright © 2026 Texas Instruments Incorporated

**Table 20-21. IMASK Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 1 | MODRDYIFG | R/W | 0h | Masks MODRDYIFG<br>0h = Interrupt is masked out<br>1h = Interrupt will request an interrupt service routine and corresponding bit in MIS will be set |
| 0 | RESERVED | R/W | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 20.3.14 RIS (Offset = 1060h) [Reset = 00001E00h]

RIS is shown in Figure 20-21 and described in Table 20-22.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Figure 20-21. RIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | DMADONEIFG | FIFOURUNIFG | FIFOEMPTYIFG | FIFO3B4IFG | FIFO1B2IFG | FIFO1B4IFG | FIFOFULLIFG |
| R-0h | R-0h | R-0h | R-1h | R-1h | R-1h | R-1h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | MODRDYIFG | RESERVED |
| R-0h | | | | | | R-0h | R-0h |

**Table 20-22. RIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-15 | RESERVED | R | 0h | |
| 14 | DMADONEIFG | R | 0h | Raw interrupt status for DMADONEIFG<br>0h = DMA done condition did not occur<br>1h = DMA done condition occurred |
| 13 | FIFOURUNIFG | R | 0h | Raw interrupt status for FIFOURUNIFG<br>0h = FIFO underrun condition did not occur<br>1h = FIFO underrun condition occurred |
| 12 | FIFOEMPTYIFG | R | 1h | Raw interrupt status for FIFOEMPTYIFG<br>0h = FIFO empty condition did not occur<br>1h = FIFO empty condition occurred |
| 11 | FIFO3B4IFG | R | 1h | Raw interrupt status for FIFO3B4IFG<br>0h = FIFO three fourth empty condition did not occur<br>1h = FIFO three fourth empty condition occurred |
| 10 | FIFO1B2IFG | R | 1h | Raw interrupt status for FIFO1B2IFG<br>0h = FIFO half empty condition did not occur<br>1h = FIFO half empty condition occurred |
| 9 | FIFO1B4IFG | R | 1h | Raw interrupt status for FIFO1B4IFG<br>0h = FIFO one fourth empty condition did not occur<br>1h = FIFO one fourth empty condition occurred |
| 8 | FIFOFULLIFG | R | 0h | Raw interrupt status for FIFOFULLIFG<br>0h = FIFO full condition did not occur<br>1h = FIFO full condition occurred |
| 7-2 | RESERVED | R | 0h | |
| 1 | MODRDYIFG | R | 0h | Raw interrupt status for MODRDYIFG<br>0h = DAC module ready event did not occur<br>1h = DAC module ready event occurred |
| 0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 20.3.15 MIS (Offset = 1068h) [Reset = 00000000h]

MIS is shown in Figure 20-22 and described in Table 20-23.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 20-22. MIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | DMADONEIFG | FIFOURUNIFG | FIFOEMPTYIFG | FIFO3B4IFG | FIFO1B2IFG | FIFO1B4IFG | FIFOFULLIFG |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | MODRDYIFG | RESERVED |
| R-0h | | | | | | R-0h | R-0h |

**Table 20-23. MIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-15 | RESERVED | R | 0h | |
| 14 | DMADONEIFG | R | 0h | Masked interrupt status for DMADONEIFG<br>0h = DMADONEIFG does not request an interrupt service routine<br>1h = DMADONEIFG requests an interrupt service routine |
| 13 | FIFOURUNIFG | R | 0h | Masked interrupt status for FIFOURUNIFG<br>0h = FIFOURUNIFG does not request an interrupt service routine<br>1h = FIFOURUNIFG requests an interrupt service routine |
| 12 | FIFOEMPTYIFG | R | 0h | Masked interrupt status for FIFOEMPTYIFG<br>0h = FIFOEMPTYIFG does not request an interrupt service routine<br>1h = FIFOEMPTYIFG requests an interrupt service routine |
| 11 | FIFO3B4IFG | R | 0h | Masked interrupt status for FIFO3B4IFG<br>0h = FIFO3B4IFG does not request an interrupt service routine<br>1h = FIFO3B4IFG requests an interrupt service routine |
| 10 | FIFO1B2IFG | R | 0h | Masked interrupt status for FIFO1B2IFG<br>0h = FIFO1B2IFG does not request an interrupt service routine<br>1h = FIFO1B2IFG requests an interrupt service routine |
| 9 | FIFO1B4IFG | R | 0h | Masked interrupt status for FIFO1B4IFG<br>0h = FIFO1B4IFG does not request an interrupt service routine<br>1h = FIFO1B4IFG requests an interrupt service routine |
| 8 | FIFOFULLIFG | R | 0h | Masked interrupt status for FIFOFULLIFG<br>0h = FIFOFULLIFG does not request an interrupt service routine<br>1h = FIFOFULLIFG requests an interrupt service routine |
| 7-2 | RESERVED | R | 0h | |
| 1 | MODRDYIFG | R | 0h | Masked interrupt status for MODRDYIFG<br>0h = MODRDYIFG does not request an interrupt service routine<br>1h = MODRDYIFG requests an interrupt service routine |
| 0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 20.3.16 ISET (Offset = 1070h) [Reset = 00000000h]

ISET is shown in Figure 20-23 and described in Table 20-24.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 20-23. ISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | DMADONEIFG | FIFOURUNIFG | FIFOEMPTYIFG | FIFO3B4IFG | FIFO1B2IFG | FIFO1B4IFG | FIFOFULLIFG |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | MODRDYIFG | RESERVED |
| W-0h | | | | | | W-0h | W-0h |

**Table 20-24. ISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-15 | RESERVED | W | 0h | |
| 14 | DMADONEIFG | W | 0h | Sets DMADONEIFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to DMADONEIFG is set |
| 13 | FIFOURUNIFG | W | 0h | Sets FIFOURUNIFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to FIFOURUNIFG is set |
| 12 | FIFOEMPTYIFG | W | 0h | Sets FIFOEMPTYIFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to FIFOEMPTYIFG is set |
| 11 | FIFO3B4IFG | W | 0h | Sets FIFO3B4IFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to FIFO3B4IFG is set |
| 10 | FIFO1B2IFG | W | 0h | Sets FIFO1B2IFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to FIFO1B2IFG is set |
| 9 | FIFO1B4IFG | W | 0h | Sets FIFO1B4IFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to FIFO1B4IFG is set |
| 8 | FIFOFULLIFG | W | 0h | Sets FIFOFULLIFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to FIFOFULLIFG is set |
| 7-2 | RESERVED | W | 0h | |
| 1 | MODRDYIFG | W | 0h | Sets MODRDYIFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to MODRDYIFG is set |
| 0 | RESERVED | W | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 20.3.17 ICLR (Offset = 1078h) [Reset = 00000000h]

ICLR is shown in Figure 20-24 and described in Table 20-25.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

**Figure 20-24. ICLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | DMADONEIFG | FIFOURUNIFG | FIFOEMPTYIFG | FIFO3B4IFG | FIFO1B2IFG | FIFO1B4IFG | FIFOFULLIFG |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | MODRDYIFG | RESERVED |
| W-0h | | | | | | W-0h | W-0h |

**Table 20-25. ICLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-15 | RESERVED | W | 0h | |
| 14 | DMADONEIFG | W | 0h | Clears DMADONEIFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to DMADONEIFG is cleared |
| 13 | FIFOURUNIFG | W | 0h | Clears FIFOURUNIFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to FIFOURUNIFG is cleared |
| 12 | FIFOEMPTYIFG | W | 0h | Clears FIFOEMPTYIFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to FIFOEMPTYIFG is cleared |
| 11 | FIFO3B4IFG | W | 0h | Clears FIFO3B4IFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to FIFO3B4IFG is cleared |
| 10 | FIFO1B2IFG | W | 0h | Clears FIFO1B2IFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to FIFO1B2IFG is cleared |
| 9 | FIFO1B4IFG | W | 0h | Clears FIFO1B4IFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to FIFO1B4IFG is cleared |
| 8 | FIFOFULLIFG | W | 0h | Clears FIFOFULLIFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to FIFOFULLIFG is cleared |
| 7-2 | RESERVED | W | 0h | |
| 1 | MODRDYIFG | W | 0h | Clears MODRDYIFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to MODRDYIFG is cleared |
| 0 | RESERVED | W | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 20.3.18 EVT_MODE (Offset = 10E0h) [Reset = 00000009h]

EVT_MODE is shown in Figure 20-25 and described in Table 20-26.

Return to the Summary Table.

Event mode register. It is used to select whether each line is disabled, in software mode (software clears the RIS) or in hardware mode (hardware clears the RIS)

**Figure 20-25. EVT_MODE**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | EVT1_CFG | | INT0_CFG | |
| R/W-0h | | | | R-2h | | R-1h | |

**Table 20-26. EVT_MODE Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R/W | 0h | |
| 3-2 | EVT1_CFG | R | 2h | Event line mode select for event corresponding to none.GEN_EVENT<br>0h = The interrupt or event line is disabled.<br>1h = Event handled by software. Software must clear the associated RIS flag.<br>2h = Event handled by hardware. The hardware (another module) clears automatically the associated RIS flag. |
| 1-0 | INT0_CFG | R | 1h | Event line mode select for event corresponding to none.CPU_INT<br>0h = The interrupt or event line is disabled.<br>1h = Event handled by software. Software must clear the associated RIS flag.<br>2h = Event handled by hardware. The hardware (another module) clears automatically the associated RIS flag. |



Copyright © 2026 Texas Instruments Incorporated

### 20.3.19 DESC (Offset = 10FCh) [Reset = 03110000h]

DESC is shown in Figure 20-26 and described in Table 20-27.

Return to the Summary Table.

This register identifies the peripheral and its exact version.

**Figure 20-26. DESC**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MODULEID | | | | | | | | | | | | | | | |
| R-311h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| FEATUREVER | | | | RESERVED | | | | MAJREV | | | | MINREV | | | |
| R-0h | | | | R- | | | | R-0h | | | | R-0h | | | |

**Table 20-27. DESC Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | MODULEID | R | 311h | Module identification contains a unique peripheral identification number. The assignments are maintained in a central database for all of the platform modules to ensure uniqueness. |
| 15-12 | FEATUREVER | R | 0h | Feature Set for the module *instance* |
| 11-8 | RESERVED | R | 0h | |
| 7-4 | MAJREV | R | 0h | Major rev of the IP |
| 3-0 | MINREV | R | 0h | Minor rev of the IP |



Copyright © 2026 Texas Instruments Incorporated

### 20.3.20 CTL0 (Offset = 1100h) [Reset = 00000000h]

CTL0 is shown in Figure 20-27 and described in Table 20-28.

Return to the Summary Table.

Control 0 register.

**Figure 20-27. CTL0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | DFM |
| R/W-0h | | | | | | | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | RES |
| R/W-0h | | | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | ENABLE |
| R/W-0h | | | | | | | R/W-0h |

**Table 20-28. CTL0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-17 | RESERVED | R/W | 0h | |
| 16 | DFM | R/W | 0h | This bit defines the DAC input data format.<br>0h = Straight binary<br>1h = Twos complement |
| 15-9 | RESERVED | R/W | 0h | |
| 8 | RES | R/W | 0h | These bits define the DAC output voltage resolution.<br>0h = 8-bits resolution<br>1h = 12-bit resolution |
| 7-1 | RESERVED | R/W | 0h | |
| 0 | ENABLE | R/W | 0h | This bit enables the DAC module.<br>0h = DAC is disabled<br>1h = DAC is enabled |



Copyright © 2026 Texas Instruments Incorporated

### 20.3.21 CTL1 (Offset = 1110h) [Reset = 00000000h]

CTL1 is shown in Figure 20-28 and described in Table 20-29.

Return to the Summary Table.

Control 1 register.

**Figure 20-28. CTL1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | OPS |
| R/W-0h | | | | | | | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | REFSN | REFSP |
| R/W-0h | | | | | | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | AMPHIZ | AMPEN |
| R/W-0h | | | | | | R/W-0h | R/W-0h |

**Table 20-29. CTL1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-25 | RESERVED | R/W | 0h | |
| 24 | OPS | R/W | 0h | These bits select the DAC output on device pin.<br>0h = No connect. Both DAC output switches are open.<br>1h = OUT0 output is selected |
| 23-10 | RESERVED | R/W | 0h | |
| 9 | REFSN | R/W | 0h | This bit selects the DAC voltage reference source + input.<br>0h = VEREFN pin as VR-<br>1h = Analog supply (VSSA) as VR- |
| 8 | REFSP | R/W | 0h | This bit selects the DAC voltage reference source + input.<br>0h = Analog supply (VDDA) as VR+<br>1h = VEREFP pin as VR+ |
| 7-2 | RESERVED | R/W | 0h | |
| 1 | AMPHIZ | R/W | 0h | AMPHIZ - amplifier output value<br>0 : amplifier output is high impedance<br>1 : amplifier output is pulled down to ground<br>0h = HiZ when disable<br>1h = dacout pulldown when disable |
| 0 | AMPEN | R/W | 0h | AMP_EN - output amplifier enabled or disabled<br>0 : disabled<br>1 : enabled<br>0h = disable<br>1h = enable |



Copyright © 2026 Texas Instruments Incorporated

### 20.3.22 CTL2 (Offset = 1120h) [Reset = 00000000h]

CTL2 is shown in Figure 20-29 and described in Table 20-30.

Return to the Summary Table.

Control 2 register.

**Figure 20-29. CTL2**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | DMATRIGEN |
| R/W-0h | | | | | | | RH/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | FIFOTRIGSEL | |
| R/W-0h | | | | | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | FIFOTH | |
| R/W-0h | | | | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | FIFOEN |
| R/W-0h | | | | | | | R/W-0h |

**Table 20-30. CTL2 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-25 | RESERVED | R/W | 0h | |
| 24 | DMATRIGEN | RH/W | 0h | This bit enables the DMA trigger generation mechanism. When this bit is set along with FIFOEN, the DMA trigger is generated based on the empty FIFO locations qualified by FIFOTH settings. This bit should be cleared by software to stop further DMA triggers.<br>0h = DMA trigger generation mechanism is disabled<br>1h = DMA trigger generation mechanism is enabled |
| 23-18 | RESERVED | R/W | 0h | |
| 17-16 | FIFOTRIGSEL | R/W | 0h | These bits select the source for FIFO read trigger. When the selected FIFO read trigger is asserted, the data from FIFO (as indicated by read pointer) is moved into internal DAC data register.<br>0h = Sample time generator output<br>1h = Hardware trigger-0 from event fabric<br>2h = Reserved - unimplemented<br>3h = Reserved - unimplemented |
| 15-10 | RESERVED | R/W | 0h | |
| 9-8 | FIFOTH | R/W | 0h | These bits determine the FIFO threshold. In case of DMA based operation, DAC generates new DMA trigger when the number of empty locations in FIFO match the selected FIFO threshold level. In case of CPU based operation, the FIFO threshold bits are don't care and FIFO level is directly indicated through the respective bits in the RIS register.<br>0h = One fourth of the FIFO locations are empty<br>1h = Half of the FIFO locations are empty<br>2h = Three fourth of the FIFO locations are empty<br>3h = Reserved value. Defaults to same effect as FIFOTH = 0 (One fourth of the FIFO locations are empty). |
| 7-1 | RESERVED | R/W | 0h | |
| 0 | FIFOEN | R/W | 0h | This bit enables the FIFO and the FIFO hardware control state machine.<br>0h = FIFO is disabled<br>1h = FIFO is enabled |



Copyright © 2026 Texas Instruments Incorporated

### 20.3.23 CTL3 (Offset = 1130h) [Reset = 00000000h]

CTL3 is shown in Figure 20-30 and described in Table 20-31.

Return to the Summary Table.

Control 3 register.

**Figure 20-30. CTL3**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | STIMCONFIG | | | |
| R/W-0h | | | | R/W-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | STIMEN |
| R/W-0h | | | | | | | R/W-0h |

**Table 20-31. CTL3 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-12 | RESERVED | R/W | 0h | |
| 11-8 | STIMCONFIG | R/W | 0h | These bits are used to configure the trigger rate from the sample time generator. The STIMCONFIG values 10 to 15 are reserved and default to same effect as value 0 (500SPS).<br>0h = Trigger rate is 500 sps (clock divide value is 4000)<br>1h = Trigger rate is 1 ksps (clock divide value is 2000)<br>2h = Trigger rate is 2 ksps (clock divide value is 1000)<br>3h = Trigger rate is 4 ksps (clock divide value is 500)<br>4h = Trigger rate is 8 ksps (clock divide value is 250)<br>5h = Trigger rate is 16 ksps (clock divide value is 125)<br>6h = Trigger rate is 100 ksps (clock divide value is 20)<br>7h = Trigger rate is 200 ksps (clock divide value is 10)<br>8h = Trigger rate is 500 ksps (clock divide value is 4)<br>9h = Trigger rate is 1 Msps (clock divide value is 2) |
| 7-1 | RESERVED | R/W | 0h | |
| 0 | STIMEN | R/W | 0h | This bit enables the sample time generator.<br>0h = Sample time generator is disabled<br>1h = Sample time generator is enabled |



Copyright © 2026 Texas Instruments Incorporated

### 20.3.24 CALCTL (Offset = 1140h) [Reset = 00000000h]

CALCTL is shown in Figure 20-31 and described in Table 20-32.

Return to the Summary Table.

Calibration control register.

**Figure 20-31. CALCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | CALSEL | CALON |
| R/W-0h | | | | | | RH/W-0h | RH/W-0h |

**Table 20-32. CALCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | R/W | 0h | |
| 1 | CALSEL | RH/W | 0h | This bit is used to select between factory trim or self calibration trim.<br>0h (R/W) = Factory Trim : Factory Trim Calibration Values are used when calibration is enabled<br>1h (R/W) = Self Calibration Trim : Self Calibration Trim Values are used when calibration is enabled |
| 0 | CALON | RH/W | 0h | This bit when set initiates the DAC offset error calibration sequence and is automatically reset when the offset error calibration completes.<br>0h = Offset error calibration is not active<br>1h = Initiate offset error calibration or offset error calibration is already in progress |



Copyright © 2026 Texas Instruments Incorporated

### 20.3.25 CALDATA (Offset = 1160h) [Reset = 00000000h]

CALDATA is shown in Figure 20-32 and described in Table 20-33.

Return to the Summary Table.

This is the offset error calibration data register.

**Figure 20-32. CALDATA**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | DATA | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | RH-0h | | | | | | |

**Table 20-33. CALDATA Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-7 | RESERVED | R | 0h | |
| 6-0 | DATA | RH | 0h | DAC offset error calibration data. The DAC offset error calibration data is represented in twos complement format providing a range of –64 to +63.<br>This is read-only bit, reflecting the calibration data. Writing to this register will have no effect, it will not change the calibration value. |



Copyright © 2026 Texas Instruments Incorporated

### 20.3.26 DATA0 (Offset = 1200h) [Reset = 00000000h]

DATA0 is shown in Figure 20-33 and described in Table 20-34.

Return to the Summary Table.

Data 0 register. This register can be written with one 8-bit or one 12-bit digital input data.

**Figure 20-33. DATA0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | DATA_VALUE | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | | | | | R/W-0h | | | | | | | | | | | |

**Table 20-34. DATA0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-12 | RESERVED | R/W | 0h | |
| 11-0 | DATA_VALUE | R/W | 0h | This is the data written for digital to analog conversion. |



Copyright © 2026 Texas Instruments Incorporated

Chapter 21
# OPA

The OPA is a zero-drift chopper stabilized operational amplifier with a programmable gain stage. This chapter describes the features and operation of the OPA peripheral.


**21.1 OPA Overview** ..... 1202
**21.2 OPA Operation** ..... 1203
**21.3 OA Registers** ..... 1213





Copyright © 2026 Texas Instruments Incorporated

## 21.1 OPA Overview

The purpose of the OPA peripheral integrated into the MSPM0xx MCU platform is to provide a way to buffer or amplify analog signals in a system without the need of discrete op-amps.

The OPA is based off of a high-performance factory trimmed amplifier core accompanied with a programmable gain stage feedback loop, configurable input muxes, and a burnout current source used to monitor sensor health.

OPA features include:

- Software selectable zero-drift chopper stabilization
- Factory trimming to remove offset error
- Rail-to-rail input & output
- Operation supported in full-scale supply voltage range
- Dual power modes for added flexibility
- Programmable gain amplifier (PGA) up to 32x
- Configurable amplifier modes include General Purpose Mode, Buffer Mode, Inverting/Non-inverting PGA Mode, Difference Amplifier Mode and Cascade Amplifier Mode
- Rich selection of external/internal op-amp terminal input options
- Dual-differential input pairs to connect multiple sensors to a single OPA
- Burnout current source (BCS) integrated to monitor sensor health
- Operates in RUN, SLEEP, and STOP modes

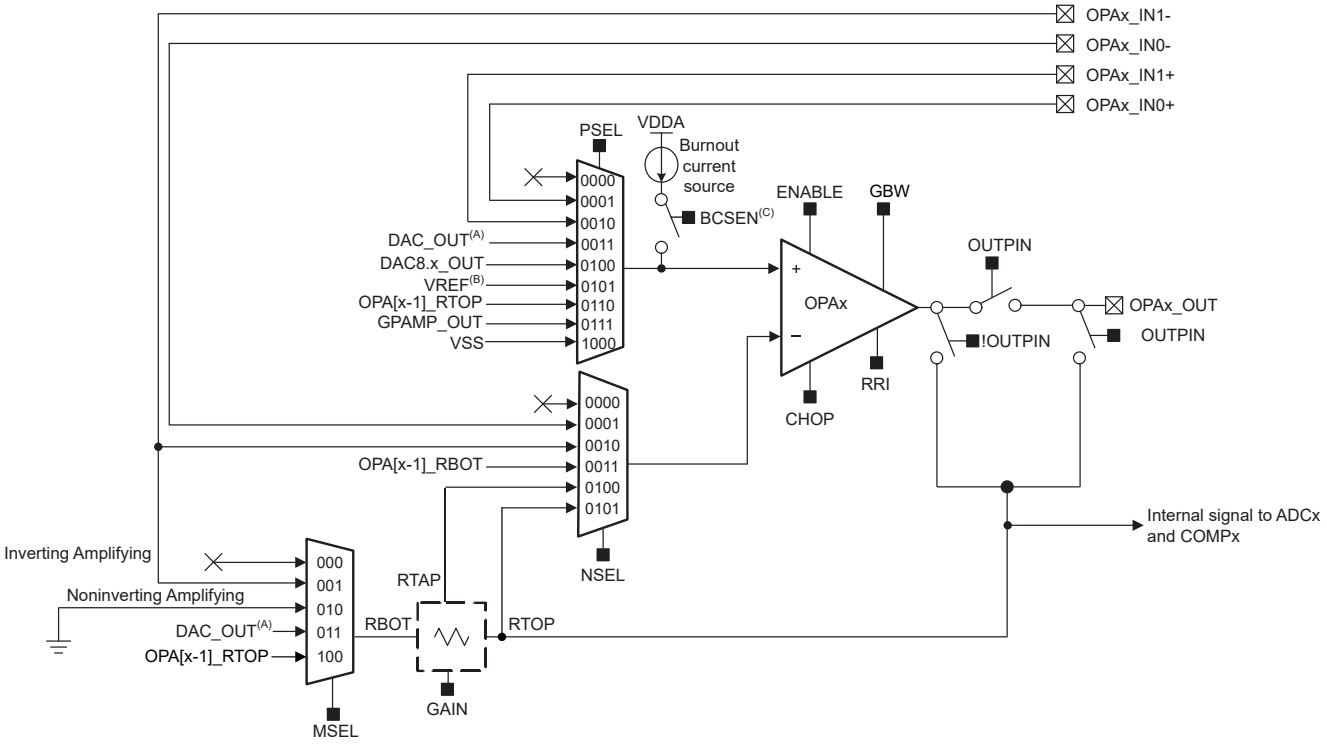
**Note**

Most devices integrate more than one OPA peripheral. If more than one OPA is present on a device, they all operate identically. Throughout this chapter, nomenclature appears such as OPAx and OPA[x-1] to describe different instances of the OPA peripheral.

Figure 21-1 shows the functional block diagram of the OPA peripheral.



Copyright © 2026 Texas Instruments Incorporated




(A) Available on select devices. See the device-specific data sheet for availability.
(B) From VREF module
(C) From ADC MMR

**Figure 21-1. OPA Block Diagram**

## 21.2 OPA Operation

The OPA is configured with application software. The following sections describe the setup and operation of the OPA peripheral.

### *21.2.1 Analog Core*

The OPA integrates a high-performance chopper stabilized low-power rail-to-rail input/output operational amplifier. It can be reconfigured in real-time to meet the performance requirements of a specific application. The following table depicts all of the op-amp analog core features and parameters that are configurable:

**Table 21-1. OPA Configurable Parameters**

| Parameter | Configuration Options |
|---|---|
| Rail-to-Rail Input (RRI) | Enabled / Disabled |
| Gain Bandwidth Product (GBW) | 6MHz (STD Mode) / 1MHz (LP Mode) |
| Chopping (CHOP) | Disabled/Standard Chop/ADC-assisted Chop |

In some basic signal conditioning applications, rail-to-rail input is not required and thus can be disabled to reduce overall power consumption. Rail-to-rail input can be disabled by clearing the RRI bit field in the PMUOPAMP register. The same concept applies to the gain bandwidth product for general purpose applications which require <1MHz GBW. The OPA can be configured to the low power operating mode (LP mode) by clearing the GBW bit in the CFGBASE register.

In some analog sensing applications, high accuracy and low offset drift is a requirement. For these use-cases, chopping can be enabled to significantly improve offset voltage and offset drift performance. Please refer to Section 21.2.6 for more details on how to properly configure and leverage the chopping feature of the OPA.



Copyright © 2026 Texas Instruments Incorporated

### 21.2.2 Power Up Behavior

The ENABLE bit in the PWREN register connects the OPA peripheral to the bus and the clock system. The ENABLE bit in the CTL register activates the OPA analog and digital core. There is a ready bit (RDY) in the STAT register which is set when the OPA is ready for use. Please see the device-specific data sheet for the enable time of the OPA peripheral.

**Note**

PWREN.ENABLE must be set before writing to CTL.ENABLE or any other OPA memory mapped registers (MMRs).

### 21.2.3 Inputs

The OPA uses input muxes P-MUX and N-MUX to provide a selection of input channels for the noninverting and inverting terminals of the amplifier. These inputs are configurable using the PSEL and NSEL control bits.

A generalized depiction of the input channel mapping is included in Table 21-2 below. Please refer to the device-specific data sheet for a more detailed depiction of the available OPAx inputs.

**Table 21-2. General OPAx Input Channels**

| PSEL | OPAx Non-inverting Channels | NSEL | OPAx Inverting Input Channels |
|---|---|---|---|
| 0x0 | Open | 0x0 | Open |
| 0x1 | OPAx_IN0+ | 0x1 | OPAx_IN0- |
| 0x2 | OPAx_IN1+ | 0x2 | OPAx_IN1- |
| 0x3 | DAC_OUT | 0x3 | OPA[x-1]_RBOT |
| 0x4 | DAC8.x_OUT | 0x4 | RTAP |
| 0x5 | VREF | 0x5 | RTOP |
| 0x6 | OPA[x-1]_RTOP | | |
| 0x7 | GPAMP Output | | |
| 0x8 | VSS | | |

**Note**

OPAx and OPA[x-1] refer to instances where there are 2 OPA peripherals in a device. In these cases, the peripherals are paired and the input/outputs are shared as shown in Table 21-2. DAC8.x refers to the paired COMP reference DAC which can be routed to the noninverting input of OPAx.

### 21.2.4 Output

The OPA output is connected to a network of complementary switches to allow the amplifier output to be a purely internal signal or allow the signal to be accessed through a device pin. If OUTPIN = 0, the output is internally connected to the programmable gain stage and capable of being routed to other analog peripherals. If OUTPIN = 1, the output goes to the device pin while still maintaining connection to the programmable gain stage and other analog peripherals. Having access to the amplifier output at the device pin allows for external filtering circuitry.

### 21.2.5 Clock Requirements

The OPA requires SYSOSC to be active for proper operation to support features such as chopping and rail-to-rail input. There are no provisions in place for the OPA to automatically request the SYSOSC from SYSCTL so the user must ensure that SYSOSC is enabled to use the OPA. The minimum requirements for the input clock depend on the OPA configuration as shown in the table below.



Copyright © 2026 Texas Instruments Incorporated

**Table 21-3.**

| RRI | Supported SYSOSC Frequencies |
|---|---|
| 0 | 4/16/24/32MHz |
| 1 | 32MHz |

**Note**

Application software needs to ensure that SYSOSC is programmed to an appropriate frequency for rail-to-rail input.

### *21.2.6 Chopping*

The OPA peripheral implements chopper stabilization to reduce offset, drift, and 1/f noise. CFG.CHOP can be set to 0x1 for standard chopping mode which requires an external filter. Note that the OPA automatically scales the chopping frequency based on CFG.GAIN which is used to select the OPA PGA gain but is also needed with external filter when the standard chopping is used as shown in Table 21-4.

**Table 21-4. Standard Chopping Frequency for OPA**

| GAIN | Inverting / Non-inverting Gain | Chopping Frequency (Hz) | Resistor(kΩ) | Capacitor(nF) |
|---|---|---|---|---|
| 0x0 | 1x | 125k | 1 | 10 |
| 0x1 | -1x / 2x | 62.5k | 1 | 30 |
| 0x2 | -3x / 4x | 31.25k | 1 | 90 |
| 0x3 | -7x / 8x | 15.6k | 1 | 270 |
| 0x4 | -15x / 16x | 7.8k | 1 | 810 |
| 0x5 | -31x / 32x | 3.9k | 1 | 2400 |

### *21.2.7 OPA Amplifier Modes*

The OPA uses an input mux, M-MUX, connected to the programmable gain stage resistor ladder to allow the user to program the OPA for various analog signal chain configurations. These configurations include unity gain buffer, inverting/noninverting amplification, PGA, cascade amplification, and more.

A generalized depiction of the input channel mapping for M-MUX is included in Table 21-5 below. Please refer to the device-specific data sheet for a mode detailed depiction of the available M-MUX inputs.

**Table 21-5. General OPA M-MUX Channel Mapping**

| MSEL | OPAx M-MUX Input Channels |
|---|---|
| 0x0 | Open |
| 0x1 | OPAx_IN1- |
| 0x2 | GND |
| 0x3 | DAC_OUT |
| 0x4 | OPA[x-1]_RTOP |

The following sections provide examples of how P-MUX, N-MUX, and M-MUX can be configured for various amplifier modes.

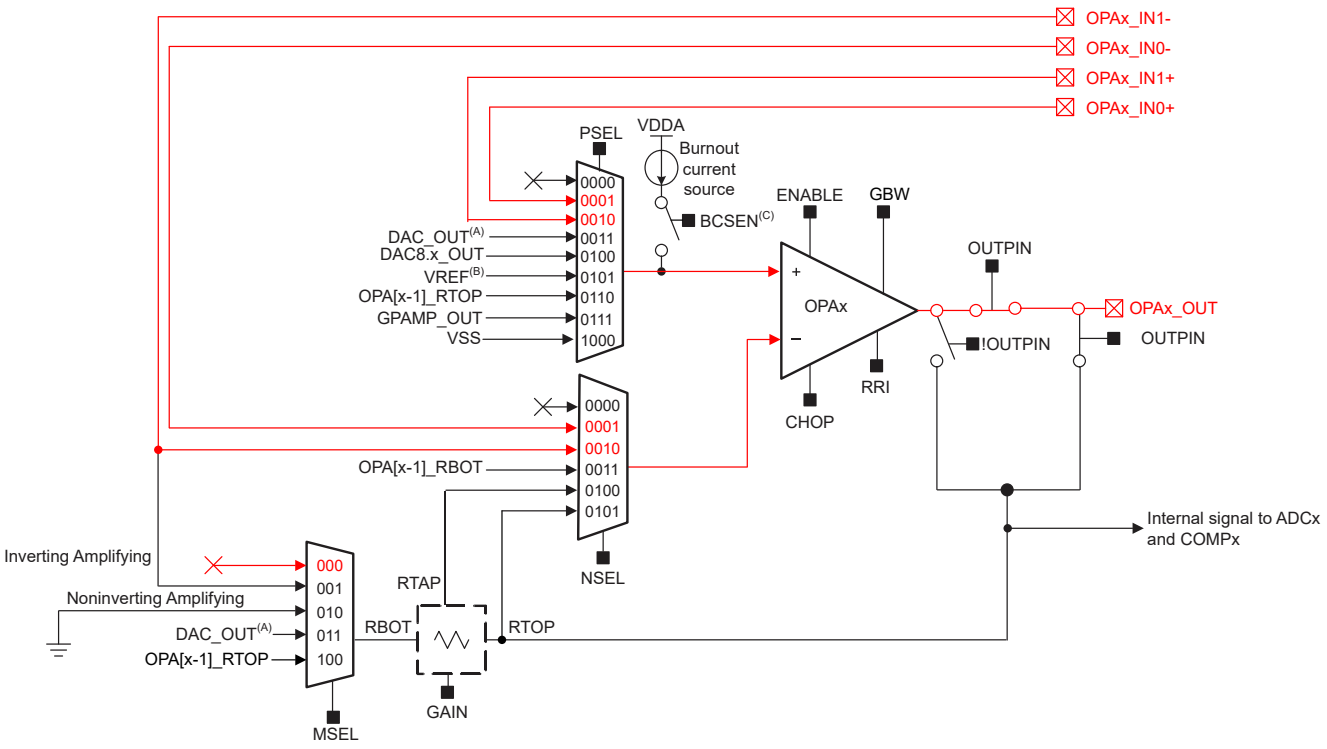
#### 21.2.7.1 General-Purpose Mode

The OPA internal muxes support GP mode which allows all input and output terminals to be accessed from device pins. Using this mode with external components enables the amplifier to realize any analog signal chain circuitry that a discrete op-amp can. Dual differential inputs IN0+/- and IN1+/- allow for two different input networks to be connected to the same OPA peripheral.

Figure 21-2 shows the block diagram of the OPA in GP mode.



Copyright © 2026 Texas Instruments Incorporated




(A) Available on select devices. See the device-specific data sheet for availability.
(B) From VREF module
(C) From ADC MMR

**Figure 21-2. OPA in General Purpose (GP) Mode**

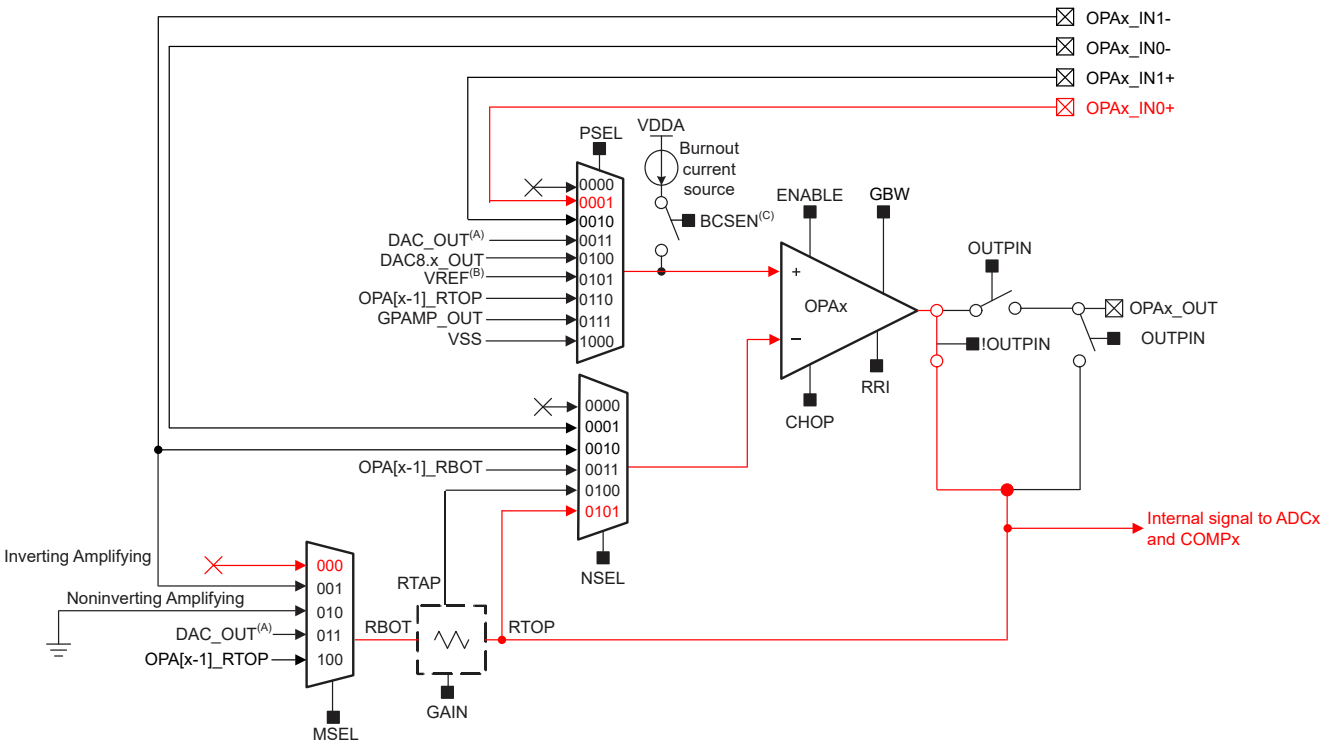
#### 21.2.7.2 Buffer Mode

The internal feedback loop can be programmed to a unity gain configuration to support OPA buffer mode. The noninverting input can be sourced from a variety of signals such as the external OPAx_INx or the on-board DAC and VREF peripherals. The OPA can output to an external pin by setting OUTPIN = 0x1 or be internally routed to on-board analog peripherals such as the ADC, COMP, and/or paired OPA by setting OUTPIN = 0x0.

Figure 21-3 shows the block diagram of the OPA in buffer mode with external OPAx_IN0+ as the noninverting input and the output internally routed to the ADC.



Copyright © 2026 Texas Instruments Incorporated



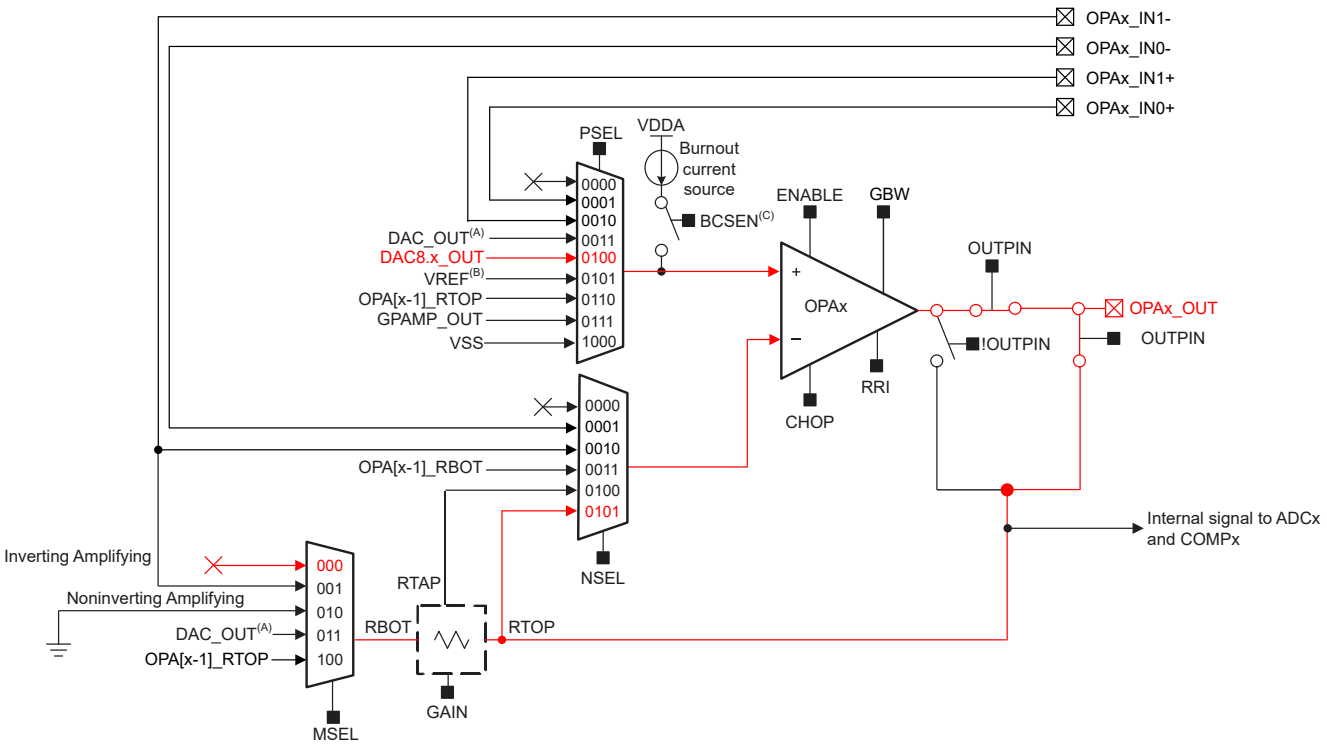

(A) Available on select devices. See the device-specific data sheet for availability.
(B) From VREF module
(C) From ADC MMR

**Figure 21-3. OPA in ADC Buffer Configuration**

Figure 21-4 shows the block diagram of the OPA in buffer mode with DAC8.x_OUT as the noninverting input and the output routed out to a device pin. This mode enables the user to use the OPA peripheral as an output buffer for the comparator onboard reference DAC.



Copyright © 2026 Texas Instruments Incorporated




(A) Available on select devices. See the device-specific data sheet for availability.
(B) From VREF module
(C) From ADC MMR

**Figure 21-4. OPA in DAC Output Buffer Configuration**

### 21.2.7.3 OPA PGA Mode

The OPA integrates a programmable gain stage in the feedback loop to configure the OPA as a PGA (programmable gain amplifier). The gain stage is a feedback resistance ladder and it supports up to 32x amplification. Table 21-6 lists the OPA PGA mode gain settings.

**Table 21-6. PGA Mode and Gain Settings**

| PGA Mode Configuration | GAIN | Gain Value from PGA (Internal Gain) |
|---|---|---|
| Inverting PGA Mode | 0x0 | Not Valid |
| | 0x1 | -1x |
| | 0x2 | -3x |
| | 0x3 | -7x |
| | 0x4 | -15x |
| | 0x5 | -31x |
| Non-inverting PGA Mode | 0x0 | Not valid |
| | 0x1 | 2x |
| | 0x2 | 4x |
| | 0x3 | 8x |
| | 0x4 | 16x |
| | 0x5 | 32x |



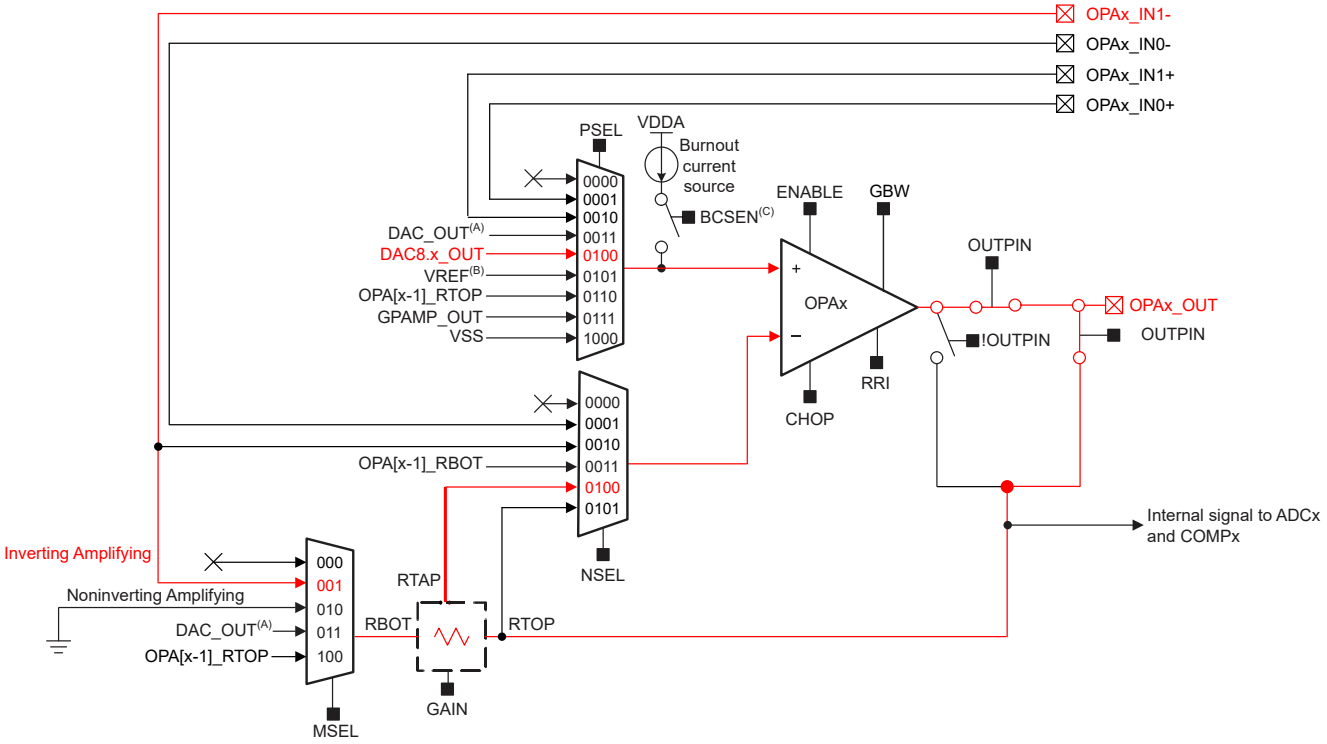
Copyright © 2026 Texas Instruments Incorporated

The programmable gain stage works with P-MUX, N-MUX, and M-MUX to support various internal amplifier configurations which are detailed in the following sections.

#### *21.2.7.3.1 Inverting PGA Mode*

In this mode, the inverting input is sourced from the external OPAx_IN1- crossing the resistor ladder of the PGA. The MSEL bits select OPAx_IN1- as the source (MSEL = 0x1) and the NSEL bits must select the RTAP as the inverting input for the amplifier (NSEL = 0x4). Because the OPA is a single supply amplifier, the noninverting input needs to be biased up to ensure that the output stays within the linear range. The noninverting input can be sourced from the external OPAx_INx+ input or from a DAC input.

Figure 21-5 shows an example of the OPA inverting PGA mode with the 8-bit reference DAC as bias cancellation.




(A) Available on select devices. See the device-specific data sheet for availability.
(B) From VREF module
(C) From ADC MMR

**Figure 21-5. OPA in Inverting PGA Configuration**

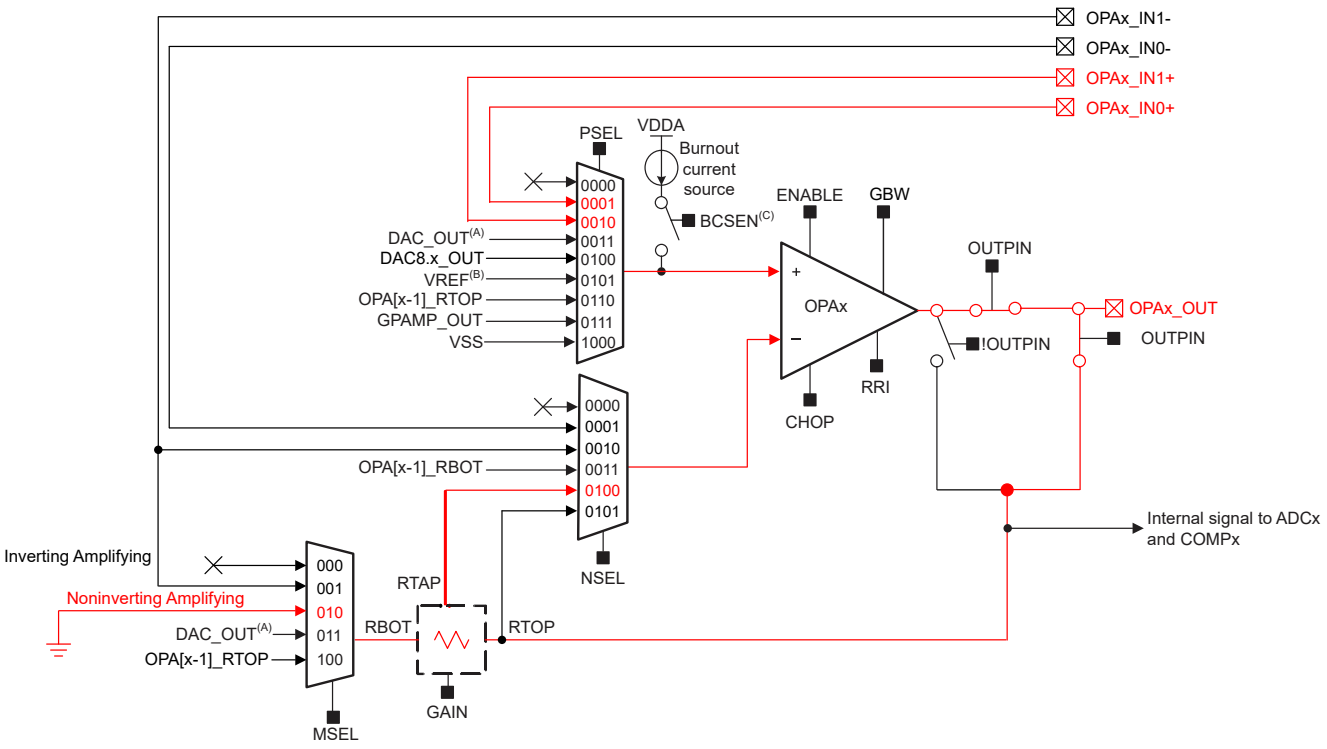
#### *21.2.7.3.2 Non-inverting PGA Mode*

In this mode, the noninverting input is sourced from the external OPAx_INx+ input (PSEL = 0x1 or 0x2). The NSEL bits must select RTAP as the inverting input for the amplifier (NSEL = 0x4) and from the MSEL bits a bias is chosen from either ground or the 12-bit buffered DAC (DAC_OUT).

Figure 21-6 shows an example of the OPA in noninverting PGA mode.



Copyright © 2026 Texas Instruments Incorporated




(A) Available on select devices. See the device-specific data sheet for availability.
(B) From VREF module
(C) From ADC MMR

**Figure 21-6. OPA in Non-inverting PGA Configuration**

#### 21.2.7.4 Difference Amplifier Mode

When two or more OPAs are available on a device, two can be combined to form a difference amplifier. The output equation for the difference amplifier is given by the Vdiff equation in Figure 21-7. The internally connected configuration for two OPAs with the output being sampled by the ADC is shown in Figure 21-8.

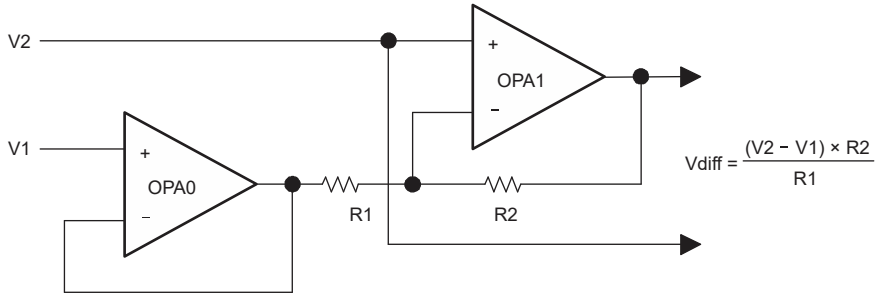



**Figure 21-7. Two OPA Difference Amplifier Block Diagram and Equation**



Copyright © 2026 Texas Instruments Incorporated

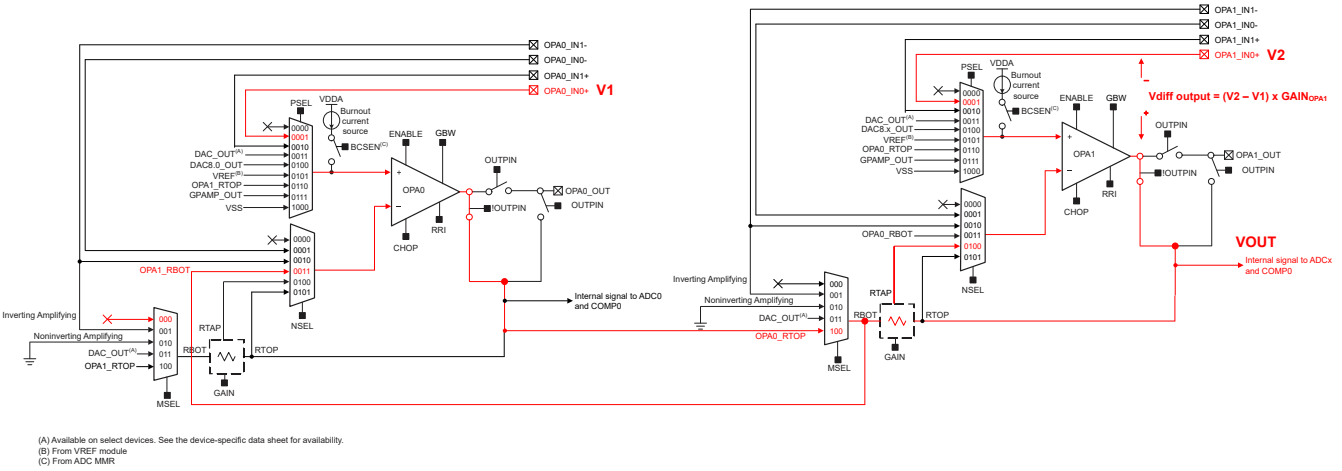



**Figure 21-8. Two OPAs in Difference Amplifier Configuration**

### 21.2.7.5 Cascade Amplifier Mode

When two or more OPAs are available on a device, two can be combined to form a multistage or cascaded amplifier. Using the programmable input muxes, all combinations of inverting and noninverting multistage amplifiers can be implemented. The output equation for the noninverting to noninverting cascaded amplifier is given by the $V_{out}$ equation in Figure 21-9. The internally connected configuration for two OPAs is shown in Figure 21-10.

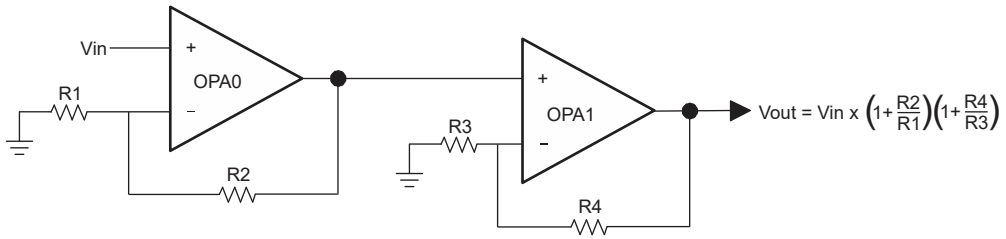



**Figure 21-9. Two OPA Non-inverting to Non-inverting Cascade Amplifier Block Diagram and Equation**

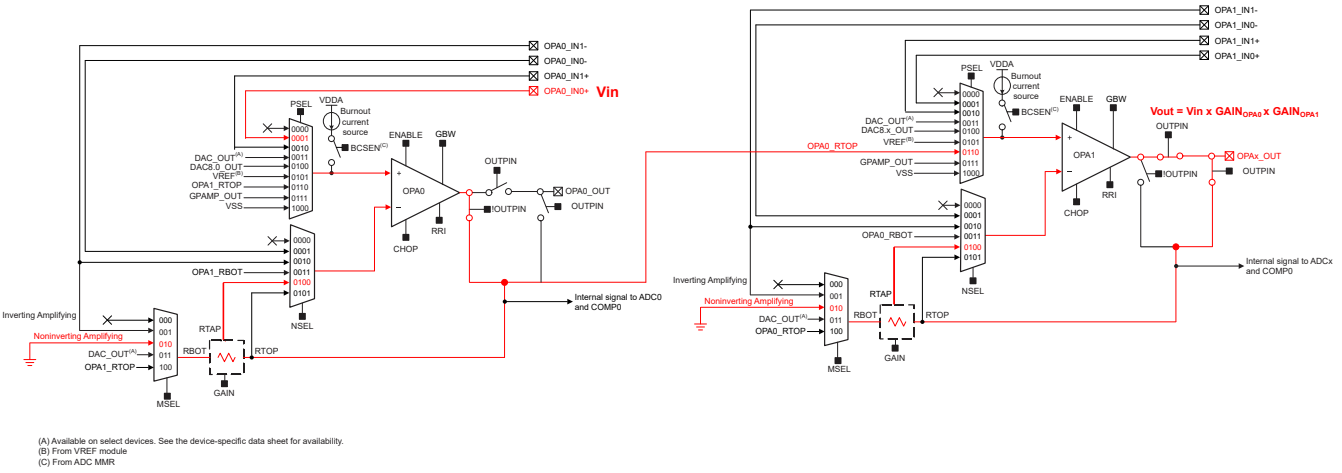



**Figure 21-10. Two OPAs in Non-inverting to Non-inverting Cascade Amplifier Configuration**

## *21.2.8 OPA Configuration Selection*

The OPA peripheral allows customers to dynamically change the OPA gain without disabling OPA functionality by using the register CFG.GAIN control bits. When using the control bits CFG.GAIN to select between different



Copyright © 2026 Texas Instruments Incorporated

gain values, the user must ensure that the additional CFG control bits remain unchanged, and that the CFG.GAIN control bits do not get cleared to 0x0.

To ensure proper operation of the OPA, the configuration settings except the CFG.GAIN bits governed by the CFG register can only be changed when the OPA is disabled. This means that to change OPA inputs or mode within an application, a user must perform the following:

- Clear CTL.ENABLE bit
- Update CFG.MSEL or CFG.NSEL or CFG.PSEL bits
- Set CTL.ENABLE bit

**Note**

Applications must take into account OPA enable time when performing this procedure. Please see OPA Specification within the device-specific data sheet for details.

**Note**

This procedure must also be followed if changing the CFG.GAIN bits to or from a value of 0x0.

### *21.2.9 Burnout Current Source*

The OPA has a burnout current source (BCS) that can be used when requested by the ADC. The ADC will provide a burnout current source enable signal to the OPA peripheral when it is selected as the ADC input channel and the BCSEN bit is set in the ADC MEMCTL register. When enabled, the burnout current source will inject current into the positive input of the OPA. The ADC should then be used to measure the voltage across the sensor or resistive Wheatstone bridge with the input signal going through the OPA configured in buffer mode (or whichever OPA configuration is normally used). An ADC reading with the burnout current source enabled can then detect if the sensor is open (that is, a broken wire resulting in full range signal) or shorted (0V). The user can periodically enable the burnout current source to check if the sensor is operating properly. It should be turned off during measurements.



Copyright © 2026 Texas Instruments Incorporated

## 21.3 OA Registers

Table 21-7 lists the memory-mapped registers for the OA registers. All register offset addresses not listed in Table 21-7 should be considered as reserved locations and the register contents should not be modified.

**Table 21-7. OA Registers**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 800h | PWREN | Power enable | | Go |
| 804h | RSTCTL | Reset Control | | Go |
| 814h | STAT | Status Register | | Go |
| 1010h | CLKOVR | Clock Override | | Go |
| 101Ch | PWRCTL | Power Control | | Go |
| 1100h | CTL | Control Register | | Go |
| 1108h | CFG | Configuration Register | | Go |
| 1118h | STAT | Status Register | | Go |

Complex bit access types are encoded to fit into small table cells. Table 21-8 shows the codes that are used for access types in this section.

**Table 21-8. OA Access Type Codes**

| Access Type | Code | Description |
|---|---|---|
| **Read Type** | | |
| R | R | Read |
| **Write Type** | | |
| W | W | Write |
| WK | W<br>K | Write<br>Write protected by a key |
| **Reset or Default Value** | | |
| *-n* | | Value after reset or the default value |



Copyright © 2026 Texas Instruments Incorporated

### 21.3.1 PWREN (Offset = 800h) [Reset = 00000000h]

PWREN is shown in Figure 21-11 and described in Table 21-9.

Return to the Summary Table.

Register to control the power state

**Figure 21-11. PWREN**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | ENABLE |
| R/W-0h | | | | | | | R/WK-0h |

**Table 21-9. PWREN Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | KEY to allow Power State Change<br>26h = KEY to allow write access to this register |
| 23-1 | RESERVED | R/W | 0h | |
| 0 | ENABLE | R/WK | 0h | Enable the power<br>KEY must be set to 26h to write to this bit.<br>0h = Disable Power<br>1h = Enable Power |



Copyright © 2026 Texas Instruments Incorporated

### 21.3.2 RSTCTL (Offset = 804h) [Reset = 00000000h]

RSTCTL is shown in Figure 21-12 and described in Table 21-10.

Return to the Summary Table.

Register to control reset assertion and de-assertion

**Figure 21-12. RSTCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | RESETSTKYCLR | RESETASSERT |
| W-0h | | | | | | WK-0h | WK-0h |

**Table 21-10. RSTCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | Unlock key<br>B1h = KEY to allow write access to this register |
| 23-2 | RESERVED | W | 0h | |
| 1 | RESETSTKYCLR | WK | 0h | Clear the RESETSTKY bit in the STAT register<br>KEY must be set to B1h to write to this bit.<br>0h = Writing 0 has no effect<br>1h = Clear reset sticky bit |
| 0 | RESETASSERT | WK | 0h | Assert reset to the peripheral<br>KEY must be set to B1h to write to this bit.<br>0h = Writing 0 has no effect<br>1h = Assert reset |



Copyright © 2026 Texas Instruments Incorporated

### 21.3.3 STAT (Offset = 814h) [Reset = 00000000h]

STAT is shown in Figure 21-13 and described in Table 21-11.

Return to the Summary Table.

peripheral enable and reset status

**Figure 21-13. STAT**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R- | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | RESETSTKY |
| R- | | | | | | | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R- | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | |
| R- | | | | | | | |

**Table 21-11. STAT Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-17 | RESERVED | R | 0h | |
| 16 | RESETSTKY | R | 0h | This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register<br>0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register<br>1h = The peripheral was reset since the last bit clear |
| 15-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 21.3.4 CLKOVR (Offset = 1010h) [Reset = 00000000h]

CLKOVR is shown in Figure 21-14 and described in Table 21-12.

Return to the Summary Table.

This register overrides the functional clock request by this peripheral to the system

**Figure 21-14. CLKOVR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | RUN_STOP | OVERRIDE |
| R/W-0h | | | | | | R/W-0h | R/W-0h |

**Table 21-12. CLKOVR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | R/W | 0h | |
| 1 | RUN_STOP | R/W | 0h | If OVERRIDE is enabled, this register is used to manually control the peripheral's clock request to the system<br>0h = Run/ungate functional clock<br>1h = Stop/gate functional clock |
| 0 | OVERRIDE | R/W | 0h | Unlocks the functionality of RUN_STOP to override the automatic peripheral clock request<br>0h = Override disabled<br>1h = Override enabled |



Copyright © 2026 Texas Instruments Incorporated

### 21.3.5 PWRCTL (Offset = 101Ch) [Reset = 00000001h]

PWRCTL is shown in Figure 21-15 and described in Table 21-13.

Return to the Summary Table.

This register controls if a peripheral is disabled if it is in idle state.

**Figure 21-15. PWRCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R/W- | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R/W- | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R/W- | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | AUTO_OFF |
| R/W- | | | | | | | R/W-1h |

**Table 21-13. PWRCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R/W | 0h | |
| 0 | AUTO_OFF | R/W | 1h | When set the peripheral will remove its local IP “request for enable” so that it can be disabled if no other entities in the system are requesting it to be enabled.<br>0h = Disable automatic power off function<br>1h = Enable automatic power off function |



Copyright © 2026 Texas Instruments Incorporated

### 21.3.6 CTL (Offset = 1100h) [Reset = X]

CTL is shown in Figure 21-16 and described in Table 21-14.

Return to the Summary Table.

Control Register

**Figure 21-16. CTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | ENABLE |
| R/W-0h | | | | | | | R/W-0h |

**Table 21-14. CTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R/W | 0h | |
| 0 | ENABLE | R/W | 0h | OAxn Enable.<br>0h (R/W) = OAxn OFF<br>1h (R/W) = OAxn ON |



Copyright © 2026 Texas Instruments Incorporated

### 21.3.7 CFG (Offset = 1108h) [Reset = 00000000h]

CFG is shown in Figure 21-17 and described in Table 21-15.

Return to the Summary Table.

Configuration Register

**Figure 21-17. CFG**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| GAIN | | | MSEL | | | NSEL | |
| R/W-0h | | | R/W-0h | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| NSEL | PSEL | | | | OUTPIN | CHOP | |
| R/W-0h | R/W-0h | | | | R/W-0h | R/W-0h | |

**Table 21-15. CFG Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R/W | 0h | |
| 15-13 | GAIN | R/W | 0h | Gain setting. Refer to TRM for enumeration information.<br>0h = Minimum gain value<br>7h = Maximum gain value. |
| 12-10 | MSEL | R/W | 0h | MSEL Mux selection.<br>Please refer to the device specific data sheet for exact channels available.<br>0h (R/W) = no connect<br>1h (R/W) = external pin OAn-1<br>2h (R/W) = VSS<br>3h (R/W) = DAC12 Output<br>4h (R/W) = OA[n-1]Rtop |
| 9-7 | NSEL | R/W | 0h | Negative OA input selection.<br>Please refer to the device specific data sheet for exact channels available.<br>0h (R/W) = no connect<br>1h (R/W) = external pin OAn-0<br>2h (R/W) = external pin OAn-1<br>3h (R/W) = OA[n+1]Rbot<br>4h (R/W) = OA[n]Rtap<br>5h (R/W) = OA[n]Rtop<br>6h (R/W) = Spare input |
| 6-3 | PSEL | R/W | 0h | Positive OA input selection.<br>Please refer to the device specific data sheet for exact channels available.<br>0h (R/W) = No connect<br>1h (R/W) = external pin OA+0<br>2h (R/W) = external pin OAn+1<br>3h (R/W) = DAC12OUT<br>4h (R/W) = DAC8OUT<br>5h (R/W) = VREF Channel<br>6h (R/W) = OA[n-1]Rtop<br>7h (R/W) = GPAMP_OUT_INT Input<br>8h = Internal Ground Connection |



Copyright © 2026 Texas Instruments Incorporated

**Table 21-15. CFG Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | OUTPIN | R/W | 0h | Enable output pin<br>0h (R/W) = Output pin disabled<br>1h (R/W) = Output pin enabled |
| 1-0 | CHOP | R/W | 0h | Chopping enable.<br>0h (R/W) = Chopping disable.<br>1h (R/W) = Standard chopping enable.<br>2h (R/W) = Chop with post average on. Requires output to be connect to ADC in average mode. |



Copyright © 2026 Texas Instruments Incorporated

### 21.3.8 STAT (Offset = 1118h) [Reset = X]

STAT is shown in Figure 21-18 and described in Table 21-16.

Return to the Summary Table.

Status Register

**Figure 21-18. STAT**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | | | | RDY |
| R-0h | | | | | | | | | | | | | | | R-0h |

**Table 21-16. STAT Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | RDY | R | 0h | OA ready status.<br>0h (R) = OAxn is not ready.<br>1h (R) = OAxn is ready. |



Copyright © 2026 Texas Instruments Incorporated

Chapter 22

# GPAMP

The GPAMP is a chopper stabilized general purpose operational amplifier with rail-to-rail input and output. This chapter describes the features and operation of the GPAMP peripheral.


**22.1 GPAMP Overview** ..... 1224
**22.2 GPAMP Operation** ..... 1224
**22.3 GPAMP Registers** ..... 1227





Copyright © 2026 Texas Instruments Incorporated

## 22.1 GPAMP Overview

The purpose of the GPAMP peripheral integrated into the MSPM0xx MCU platform is to provide a way to buffer or amplify analog signals in a system without the need of discrete op-amps.

The GPAMP is based off of a factory trimmed amplifier core accompanied with a configurable inverting input mux which allows for various analog signal chain configurations.

GPAMP features include:

- Software selectable chopper stabilization
- Rail-to-rail input & output
- Operation supported in full-scale supply voltage range
- Programmable internal unity gain feedback loop
- Operates in RUN, SLEEP, and STOP modes

Figure 22-1 show the functional block diagram of the GPAMP peripheral.

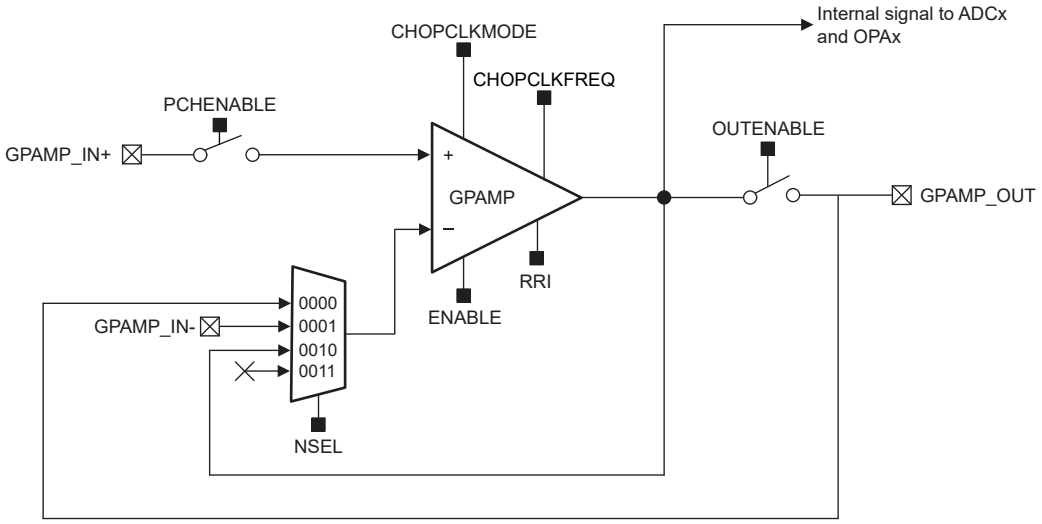



**Figure 22-1. GPAMP Block Diagram**

## 22.2 GPAMP Operation

The GPAMP peripheral is configured with user software using the PMUOPAMP register which is located in the PMCU region of memory mapped registers. For bit field definitions, you can refer to the end of this chapter or the register section of PMCU. The setup and operation of GPAMP is discussed in the following sections.

### *22.2.1 Analog Core*

The GPAMP integrates a chopper stabilized low-power rail-to-rail input/output operational amplifier. It can be reconfigured in real-time to meet the performance requirements of various applications. The following table depicts all of the amplifier analog core features and parameters that are configurable:

**Table 22-1. GPAMP Configurable Parameters**

| Parameter | Configuration Options |
|---|---|
| Rail-to-Rail Input (RRI) | Mode 0 / Mode 1 / Mode 2 |
| Chopping (CHOPCLKMODE) | Standard Chop / Disabled |

In some basic signal conditioning applications, rail-to-rail input is not required and thus can be disabled to reduce overall power consumption. Rail-to-rail input can be disabled by programming the RRI bit field to 0x0 (Mode 0). Refer to the GPAMP specifications in the device specific data sheet for supported common-mode input ranges in the various RRI modes.



Copyright © 2026 Texas Instruments Incorporated

In some analog sensing applications, high accuracy and low offset drift is a requirement. For these use-cases, chopping can be enabled to significantly improve offset voltage and offset drift performance. Please refer to Section 22.2.6 for more details on how to properly configure and leverage the chopping feature of the GPAMP.

### *22.2.2 Power Up Behavior*

The ENABLE bit in the PMUOPAMP register activates the GPAMP analog and digital core. The GPAMP requires the VBOOST circuit in the PMU to be stabilized before use. Refer to Section 2.2.5 for more details on what the VBOOST circuit is and why it is needed for proper GPAMP operation. Please see the device-specific data sheet for the enable time of the GPAMP peripheral.

### *22.2.3 Inputs*

The GPAMP uses an input mux, N-MUX, to provide a selection of input channels for the inverting terminal of the amplifier. These inputs are configurable using the NSEL control bits. The noninverting terminal of the amplifier is enabled using the PCHENABLE control bit and allows the input to be driven from the GPAMP_IN+ signal on a device pin.

A depiction of the input channel mapping is included in Table 22-2 below.

**Table 22-2. GPAMP Input Channels**

| PCHENABLE | GPAMP Non-inverting Input | NSEL | GPAMP Inverting Input Channels |
|---|---|---|---|
| 0x0 | Open | 0x0 | GPAMP output pad (GPAMP_OUT) |
| 0x1 | GPAMP_IN+ | 0x1 | GPAMP_IN- |
| | | 0x2 | GPAMP internal amplifier output |
| | | 0x3 | Open |

### *22.2.4 Output*

The GPAMP output is connected to a switch to allow the amplifier output to be a purely internal signal or allow the signal to be accessed through a device pin. If OUTENABLE = 1, the amplifier output goes to the device pin while still maintaining connection to the inverting input mux and other analog peripherals. If OUTENABLE = 0, the amplifier output is disconnected from the device pin but is still internally connected to the inverting input mux and other analog peripherals on-board. Having access to the amplifier output at the device pin allows for external filtering circuitry. Refer to the GPAMP Block Diagram for a detailed schematic of the GPAMP output circuitry.

### *22.2.5 GPAMP Amplifier Modes*

The GPAMP uses N-MUX and an internal unity gain feedback loop to allow the user to program the GPAMP for various analog signal chain configurations. These configurations include general purpose mode, ADC buffer, and unity gain mode.

The following sections provide examples of how GPAMP can be configured for these various amplifier modes.

#### 22.2.5.1 General-Purpose Mode

The GPAMP supports general-purpose (GP) mode which allows all input and output terminals to be accessed from device pins. To realize this GP mode NSEL, PCHENABLE, and OUTENABLE must all be set to 0x1. Using this mode requires external components and an appropriate feedback loop to build the circuit desired.

Figure 22-2 shows the block diagram of the GPAMP in GP mode.



Copyright © 2026 Texas Instruments Incorporated

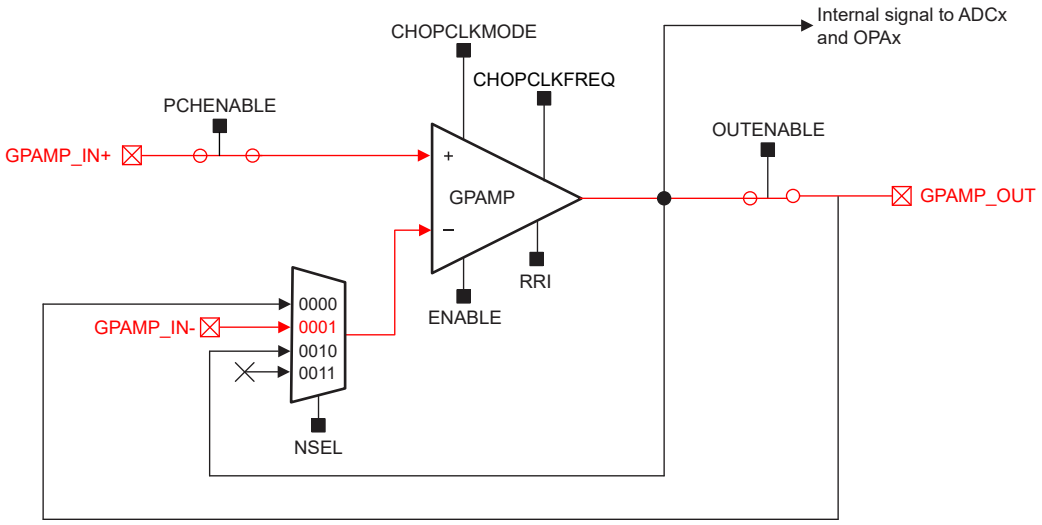



**Figure 22-2. GPAMP in General Purpose (GP) Mode**

#### 22.2.5.2 ADC Buffer Mode

The GPAMP can be used to internally buffer an analog signal into an on-board ADC. To realize this ADC buffer mode, NSEL must be set to 0x2, PCHENABLE must be set to 0x1, and OUTENABLE must be set to 0x0.

Figure 22-3 shows the block diagram of the GPAMP in ADC buffer mode with GPAMP_IN+ as the noninverting input and the output internally routed to the ADC.

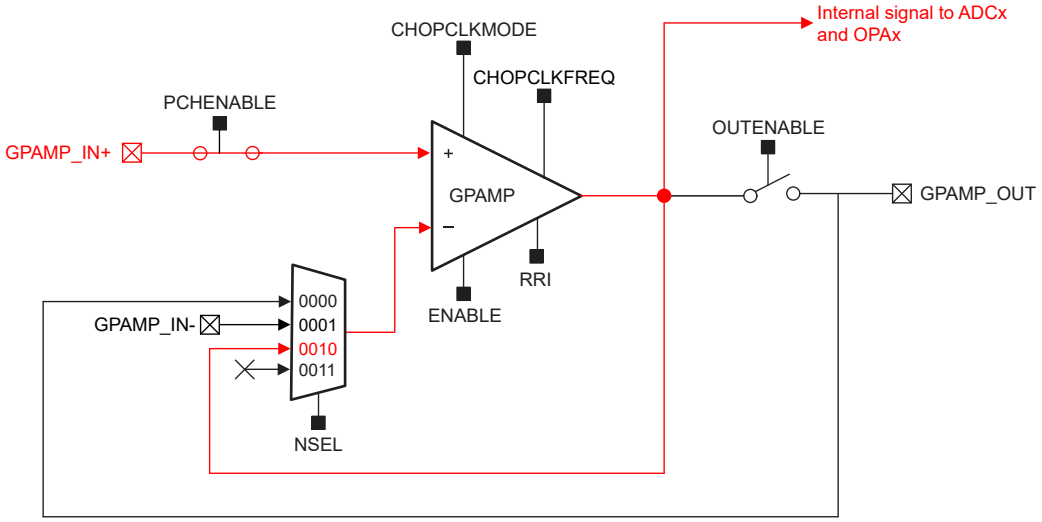



**Figure 22-3. GPAMP in ADC Buffer Configuration**

#### 22.2.5.3 Unity Gain Mode

The internal feedback loop of the GPAMP can be used to implement a unity gain amplifier which can buffer an analog signal to other devices in a system. To realize this unity gain mode, NSEL must be set to 0x0, PCHENABLE must be set to 0x1, and OUTENABLE must be set to 0x1.

Figure 22-4 shows the block diagram of the GPAMP in unity gain mode.



Copyright © 2026 Texas Instruments Incorporated

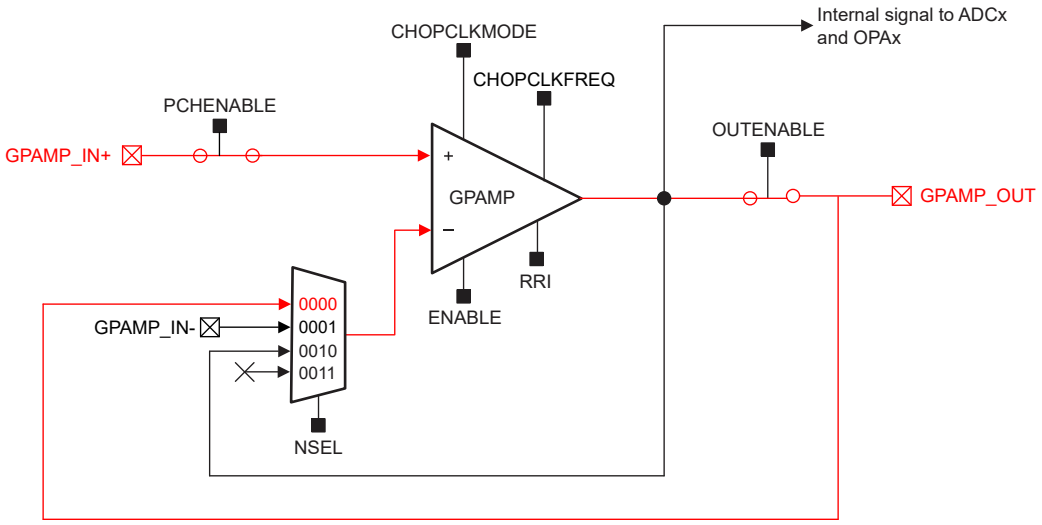



**Figure 22-4. GPAMP in Unity Gain Configuration**

### *22.2.6 Chopping*

The GPAMP peripheral implements chopper stabilization to reduce offset, drift, and 1/f noise. CHOPCLKMODE can be set to 0x1 for standard chopping mode which requires an external filter. See device-specific data sheet for recommended values. The chopping frequency, which is set by CHOPCLKFREQ, needs to scale down as the gain increases. Refer to Table 22-3 for info on what chopping frequency is required for a given gain.

**Table 22-3. Standard Chopping Frequency for GPAMP**

| CHOPCLKFREQ | Supported External Gain Configuration | Chopping Frequency (Hz) |
|---|---|---|
| 0x0 | -1x / 2x | 16k |
| 0x1 | -3x / 4x | 8k |
| 0x2 | -7x / 8x | 4k |
| 0x3 | -15x / 16x | 2k |

## 22.3 GPAMP Registers

The GPAMP control register (PMUOPAMP) is in the SYSCTL registers.



Copyright © 2026 Texas Instruments Incorporated

Chapter 23

# VREF

The VREF module contains a configurable voltage reference buffer which allows users to supply a stable internal reference to on-board analog peripherals. It also supports bringing in an external reference for applications where higher accuracy is required. This chapter describes the features and operation of the VREF module.


**23.1 VREF Overview** ..... 1229
**23.2 VREF Operation** ..... 1230
**23.3 VREF Registers** ..... 1233





Copyright © 2026 Texas Instruments Incorporated

## 23.1 VREF Overview

The VREF module for the MSPM0Gxx family is a shared voltage reference module which can be leveraged by a variety on on-board analog peripherals. VREF allows users to choose between using an internally generated reference voltage or using an externally provided reference voltage from outside the MCU. Please reference the device data sheet for specifics about its VREF configuration.

The VREF module features include:

- 1.4V and 2.5V user-selectable internal references
- Support for receiving external reference on VREF+/- device pins
- Internal reference output available on VREF+ device pin
- Sample and hold mode supports VREF operation down to STANDBY operating mode
- Internal reference supports full speed ADC operation



Copyright © 2026 Texas Instruments Incorporated

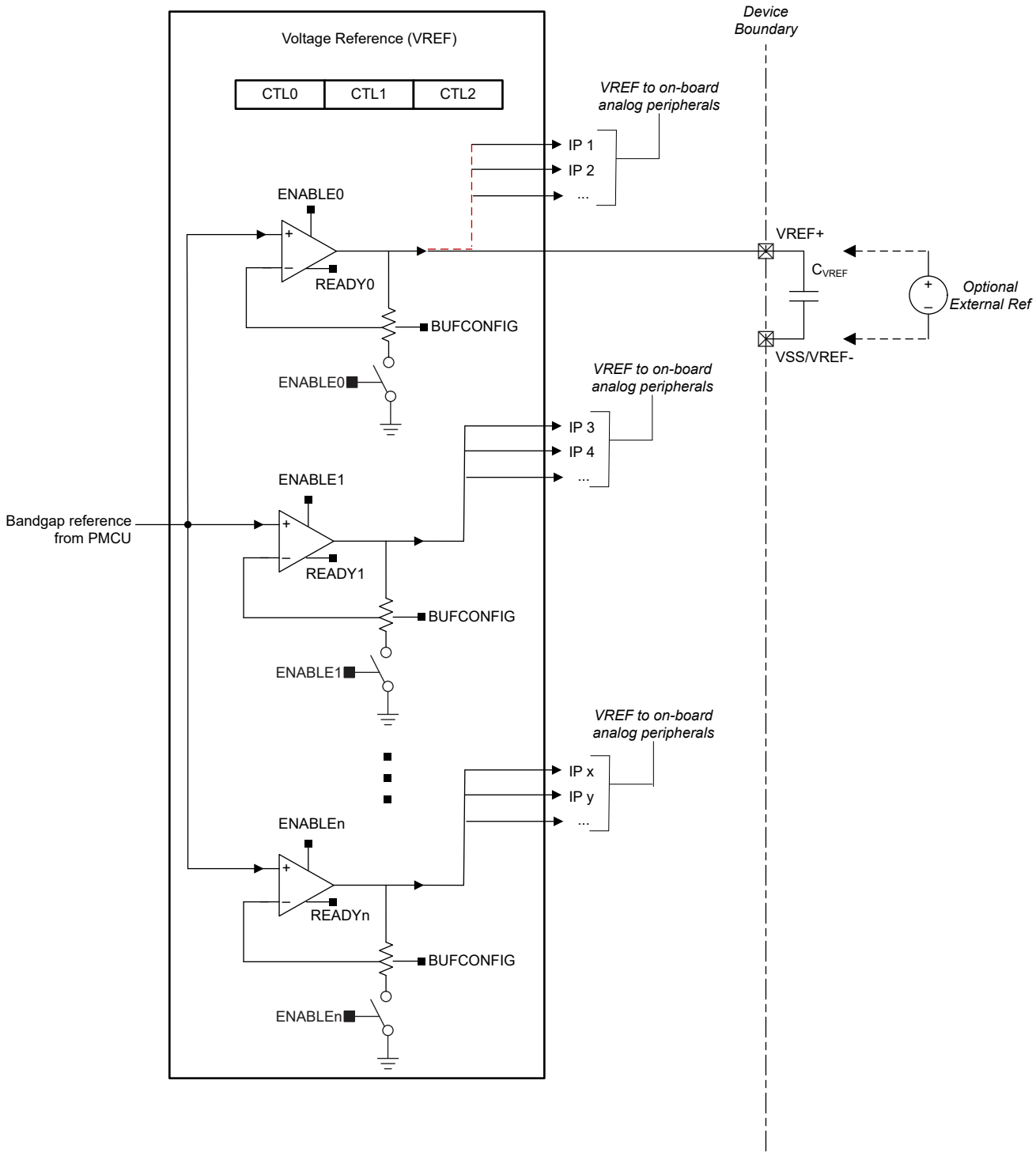



**Figure 23-1. MSPM0 VREF Superset Block Diagram**

## 23.2 VREF Operation

The VREF module is configured with user software. The setup and operation of VREF is discussed in the following sections.



Copyright © 2026 Texas Instruments Incorporated

### 23.2.1 Internal Reference Generation

To use VREF to generate an internal voltage reference, the user must first enable register write access to the module using the ENABLE control bit in the PWREN register, and then enable the reference buffer using the ENABLE control bit in the CTL0 register. The VREF module generates voltage references based on the factory trimmed bandgap coming from the PMU. The bandgap reference is buffered through a noninverting amplifier to generate an internal reference voltage (1.4V or 2.5V). Only one voltage can be selected at a time using the BUFCONFIG control bit in CTL0.

**Note**

The internal reference generated by VREF requires an external decoupling capacitor ($C_{VREF}$) for proper operation. Please refer to the device specific data sheet for more information on the value and required placement of $C_{VREF}$.

After it is enabled and settled, the internal reference can be used as an accurate and stable voltage reference for on-board analog peripherals. Refer to the analog peripheral chapters for more info on how can leverage this reference voltage.

The VREF provides a READY indication bit in the CTL1 register. CTL1 READY bit is set to 1 even after disabling and later re-enabling of VREF IP.

**Note**

VREF configuration change from 2.5V to 1.4V mode has a very slow slew rate. If this change is needed; then follow below procedure:

- Disable VREF using the ENABLE bit in the CTL0 register
- Configure pin PA23(VREF+) as GPIO output and drive this pin to logic low for 100us. A 1uF external capacitor is assumed on the VREF+ pin
- Re-enable VREF in 1.4V mode by setting the BUGCONFIG bit in the CTL0 register to 1. After VREF is enabled and settled, the internal reference of 1.4V can be used

**Note**

The VREF ready bit will remain high throughout this process, so the settling time will need to be manually tracked

### 23.2.2 External Reference Input

For circumstances where accuracy is a key requirement, an external reference can be brought in to the MCU from the VREF+ and VREF- device pins. Only one type of reference (internal or external) can be selected at a time. When supplying the MCU with an external reference, it is recommended to connect a decoupling capacitor on the reference pins with a value based on the voltage source.

In situations where the external VREF is being enabled prior to entering Standby1, be sure to wait for the EXTVREF ready signal (CTL1) to become 1. Alternatively, wait for EXTVREF powerup time before entering Standby1

On devices where there are two VREFs, the ready bit only applies to the first VREF that is enabled. For the second VREF, the user will need to use a counter to manually wait for the settling time.

**Note**

Be sure the VREF reference buffer is disabled by clearing the ENABLE control bit in the CTL0 register before applying the external reference to the device pins.



Copyright © 2026 Texas Instruments Incorporated

### 23.2.3 Analog Peripheral Interface

#### VREF to ADC

The ADC peripheral can leverage the internal configurable reference or an external reference using the VRSEL control bits in the ADC MEMCTL register.

The user must make sure that the VREF reference buffer has settled before triggering an ADC conversion when using an internal reference as the ADC voltage reference (VR+). Refer to the device-specific data sheet for VREF settling times and refer to the ADC chapter for more details on all of the reference options available for the ADC peripheral.

#### VREF to COMP

The comparator peripheral can leverage the internal configurable reference or an external reference using the REFSRC control bits within the CTL2 register associated with the COMP.

Refer to the COMP chapter for more details on all of the reference options available for the comparator peripheral.

#### VREF to DAC

The DAC peripheral can leverage the internal configurable reference or an external reference using the REFSP control bits in the DAC CTL1 register.

Refer to the DAC chapter for more details on all of the reference options available for the DAC peripheral.

#### VREF to OPA

The OPA peripheral can leverage the internal configurable reference or an external reference for setting a voltage bias on the noninverting input terminal of the amplifier using the PSEL control bits in the CFG register. In cases where other devices in a system need to use the same reference as the MCU, the OPA can be used in unity gain buffer configuration to drive the VREF internal reference voltage to the OPA_OUT pin which can be used as a reference for other devices on the PCB.

Refer to the OPA chapter for more details on all of the channels available for the noninverting amplifier input terminal of the amplifier.

### 23.2.4 Sample and Hold Mode

Sample and hold mode for the VREF module charges the external VREF capacitor to the selected voltage during the sample phase. During the hold phase, the VREF capacitor is allowed to discharge. The value of the charged capacitor is used as the reference value. This mode helps to conserve power.

VREF sample and hold feature is not available in STOP2/STANDBY0/STANDBY1 modes if MFCLK is chosen as VREF CLKSEL because MFCLK will not be available when SYSOSC is absent.

VREF sample and hold feature is not available in STANDBY1 mode if we select BUSCLK or LFCLK as VREF CLKSEL because LFCLK will not be available.



Copyright © 2026 Texas Instruments Incorporated

## 23.3 VREF Registers

Table 23-1 lists the memory-mapped registers for the VREF registers. All register offset addresses not listed in Table 23-1 should be considered as reserved locations and the register contents should not be modified.

**Table 23-1. VREF Registers**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 800h | PWREN | Power enable | | Go |
| 804h | RSTCTL | Reset Control | | Go |
| 814h | STAT | Status Register | | Go |
| 1000h | CLKDIV | Clock Divider | | Go |
| 1008h | CLKSEL | Clock Selection | | Go |
| 1100h | CTL0 | Control 0 | | Go |
| 1104h | CTL1 | Control 1 | | Go |
| 1108h | CTL2 | Control 2 | | Go |

Complex bit access types are encoded to fit into small table cells. Table 23-2 shows the codes that are used for access types in this section.

**Table 23-2. VREF Access Type Codes**

| Access Type | Code | Description |
|---|---|---|
| Read Type | | |
| R | R | Read |
| Write Type | | |
| W | W | Write |
| WK | W<br>K | Write<br>Write protected by a key |
| Reset or Default Value | | |
| *-n* | | Value after reset or the default value |



Copyright © 2026 Texas Instruments Incorporated

### 23.3.1 PWREN (Offset = 800h) [Reset = 00000000h]

PWREN is shown in Figure 23-2 and described in Table 23-3.

Return to the Summary Table.

Register to control the power state

**Figure 23-2. PWREN**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | ENABLE |
| R-0h | | | | | | | R/WK-0h |

**Table 23-3. PWREN Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | KEY to allow Power State Change<br>26h = KEY to allow write access to this register |
| 23-1 | RESERVED | R | 0h | |
| 0 | ENABLE | R/WK | 0h | Enable the power<br>KEY must be set to 26h to write to this bit.<br>0h = Disable Power<br>1h = Enable Power |



Copyright © 2026 Texas Instruments Incorporated

### 23.3.2 RSTCTL (Offset = 804h) [Reset = 00000000h]

RSTCTL is shown in Figure 23-3 and described in Table 23-4.

Return to the Summary Table.

Register to control reset assertion and de-assertion

**Figure 23-3. RSTCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | RESETSTKYCLR | RESETASSERT |
| R-0h | | | | | | WK-0h | WK-0h |

**Table 23-4. RSTCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | Unlock key<br>B1h = KEY to allow write access to this register |
| 23-2 | RESERVED | R | 0h | |
| 1 | RESETSTKYCLR | WK | 0h | Clear the RESETSTKY bit in the STAT register<br>KEY must be set to B1h to write to this bit.<br>0h = Writing 0 has no effect<br>1h = Clear reset sticky bit |
| 0 | RESETASSERT | WK | 0h | Assert reset to the peripheral<br>KEY must be set to B1h to write to this bit.<br>0h = Writing 0 has no effect<br>1h = Assert reset |



Copyright © 2026 Texas Instruments Incorporated

### 23.3.3 STAT (Offset = 814h) [Reset = 00000000h]

STAT is shown in Figure 23-4 and described in Table 23-5.

Return to the Summary Table.

peripheral enable and reset status

**Figure 23-4. STAT**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | RESETSTKY |
| R-0h | | | | | | | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |

**Table 23-5. STAT Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-17 | RESERVED | R | 0h | |
| 16 | RESETSTKY | R | 0h | This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register<br>0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register<br>1h = The peripheral was reset since the last bit clear |
| 15-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 23.3.4 CLKDIV (Offset = 1000h) [Reset = 00000000h]

CLKDIV is shown in Figure 23-5 and described in Table 23-6.

Return to the Summary Table.

Clock Divider

**Figure 23-5. CLKDIV**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | | RATIO | | |
| R-0h | | | | | | | | | | | | | R/W-0h | | |

**Table 23-6. CLKDIV Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | RESERVED | R | 0h | |
| 2-0 | RATIO | R/W | 0h | Selects divide ratio of module clock to be used in sample and hold logic |



Copyright © 2026 Texas Instruments Incorporated

### 23.3.5 CLKSEL (Offset = 1008h) [Reset = 00000000h]

CLKSEL is shown in Figure 23-6 and described in Table 23-7.

Return to the Summary Table.

Clock Selection

**Figure 23-6. CLKSEL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | BUSCLK_SEL | MFCLK_SEL | LFCLK_SEL | RESERVED |
| R-0h | | | | R/W-0h | R/W-0h | R/W-0h | R-0h |

**Table 23-7. CLKSEL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | BUSCLK_SEL | R/W | 0h | Selects BUSCLK as clock source if enabled |
| 2 | MFCLK_SEL | R/W | 0h | Selects MFCLK as clock source if enabled |
| 1 | LFCLK_SEL | R/W | 0h | Selects LFCLK as clock source if enabled |
| 0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 23.3.6 CTL0 (Offset = 1100h) [Reset = 00000000h]

CTL0 is shown in Figure 23-7 and described in Table 23-8.

Return to the Summary Table.

Control 0 register.

**Figure 23-7. CTL0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | SHMODE |
| R-0h | | | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BUFCONFIG | RESERVED | | | | ENABLE2 | ENABLE1 | ENABLE0 |
| R/W-0h | R-0h | | | | R/W-0h | R/W-0h | R/W-0h |

**Table 23-8. CTL0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-9 | RESERVED | R | 0h | |
| 8 | SHMODE | R/W | 0h | This bit enable sample and hold mode<br>0h = Sample and hold mode is disable<br>1h = Sample and hold mode is enable |
| 7 | BUFCONFIG | R/W | 0h | These bits configure output buffer.<br>0h = output 2p5v : Configure Output Buffer to 2.5v<br>1h = output 1p4v : Configure Output Buffer to 1.4v |
| 6-3 | RESERVED | R | 0h | |
| 2 | ENABLE2 | R/W | 0h | This bit enables the VREF reference buffer 2.<br>0h = VREF buffer is disabled<br>1h = VREF buffer is enabled |
| 1 | ENABLE1 | R/W | 0h | This bit enables the VREF reference buffer 1.<br>0h = VREF buffer is disabled<br>1h = VREF buffer is enabled |
| 0 | ENABLE0 | R/W | 0h | This bit enables the VREF reference buffer 0.<br>0h = VREF buffer is disabled<br>1h = VREF buffer is enabled |



Copyright © 2026 Texas Instruments Incorporated

### 23.3.7 CTL1 (Offset = 1104h) [Reset = 00000000h]

CTL1 is shown in Figure 23-8 and described in Table 23-9.

Return to the Summary Table.

Control 1 register.

**Figure 23-8. CTL1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | READY |
| R-0h | | | | | | | R-0h |

**Table 23-9. CTL1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | READY | R | 0h | These bits defines status of VREF<br>0h = VREF output is not ready<br>1h = VREF output is ready |



Copyright © 2026 Texas Instruments Incorporated

### 23.3.8 CTL2 (Offset = 1108h) [Reset = 00000000h]

CTL2 is shown in Figure 23-9 and described in Table 23-10.

Return to the Summary Table.

Control 2 register.

**Figure 23-9. CTL2**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| HCYCLE | | | | | | | | | | | | | | | | SHCYCLE | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | R/W-0h | | | | | | | | | | | | | | | |

**Table 23-10. CTL2 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | HCYCLE | R/W | 0h | Hold cycle count<br>Total cycles of module clock for hold phase when VREF is working in sample and hold mode in STANDBY to save power.<br>Please refer VREF section of datasheet for recommended values of sample and hold times.<br>0h = smallest hold cycle<br>FFFFh = largest hold cycle |
| 15-0 | SHCYCLE | R/W | 0h | Sample and Hold cycle count<br>Total cycles of module clock for sample and hold phase when VREF is working in sample and hold mode in STANDBY to save power.<br>This field should be greater than HCYCLE field. The difference between this field and HCYCLE gives the number of cycles of sample phase.<br>Please refer VREF section of datasheet for recommended values of sample and hold times.<br>0h = smallest sample and hold cycle count<br>FFFFh = largest sample and hold cycle |



Copyright © 2026 Texas Instruments Incorporated

Chapter 24

# UART

The universal asynchronous receiver-transmitter (UART) module provides an interface for serial communication protocols.


**24.1 UART Overview** ..... 1243
**24.2 UART Operation** ..... 1245
**24.3 UART Registers** ..... 1265





Copyright © 2026 Texas Instruments Incorporated

## 24.1 UART Overview

### 24.1.1 Purpose of the Peripheral

The UART module can share data between devices with asynchronous serial communication protocols such as LIN (local interconnection network), ISO7816 (Smart card protocol), IrDA (infrared data association), and hardware flow control (CTS/RTS).

### 24.1.2 Features

Refer to the device-specific datasheet for the supported features, as some devices may not include support for select features. The following list describes the full complement of features available:

- Fully programmable serial interface:
  - 5, 6, 7 or 8 data bits
  - Even, odd, stick, or no-parity bit generation and detection
  - 1 or 2 stop bit generation
  - LSB-first or MSB-first data transmit and receive
  - Line-break detection
  - Glitch filter on the input signals
  - Programmable baud-rate generation with oversampling (16, 8, or 3)
- Separate transmit and receive 4-byte FIFOs to reduce CPU interrupt servicing
- Supports DMA data transfer
- Standard FIFO-level and End-of-Transmission interrupts
- Active in all low-power modes
- SYSOSC wakeup via an asynchronous fast clock request when operating in low power modes (supports up to 19200 baud rates when using SYSOSC in FCL mode (1% accuracy)
- Loopback mode operation
- Hardware flow control
- 9-bit multi-drop configuration
- Supported protocols:
  - LIN
  - DALI
  - IrDA
  - ISO7816 Smart card
  - RS485
  - Manchester encoding
  - Idle-Line Multiprocessor

There are two types of UART instances: Extend and Main. The following table compares the features of UART Extend and UART Main.

**Table 24-1. UART Extend and Main Features** (1)

| Features | UART Extend | UART Main (Low-Power Mode Support) | UART Main |
|---|---|---|---|
| Instances | UART0, UART7 | UART1, UART2 | UART3-UART6 |
| Hardware flow control | Yes | Yes | Yes |
| Oversampling options | 3, 8, 16 | 3, 8, 16 | 3, 8, 16 |
| Separate transmit and receive FIFOs | Yes | Yes | Yes |
| Active in STOP and STANDBY modes | Yes(2) | Yes(2) | No |
| Wake-up with start bit | Yes(3) | Yes(3) | Yes(3) |
| Digital Glitch filter | Yes | No | No |
| Analog Glitch filter | Yes | Sometimes | No |
| 9-bit multi-drop configuration | Yes | Yes | Yes |
| Idle-Line Multiprocessor | Yes | Yes | Yes |



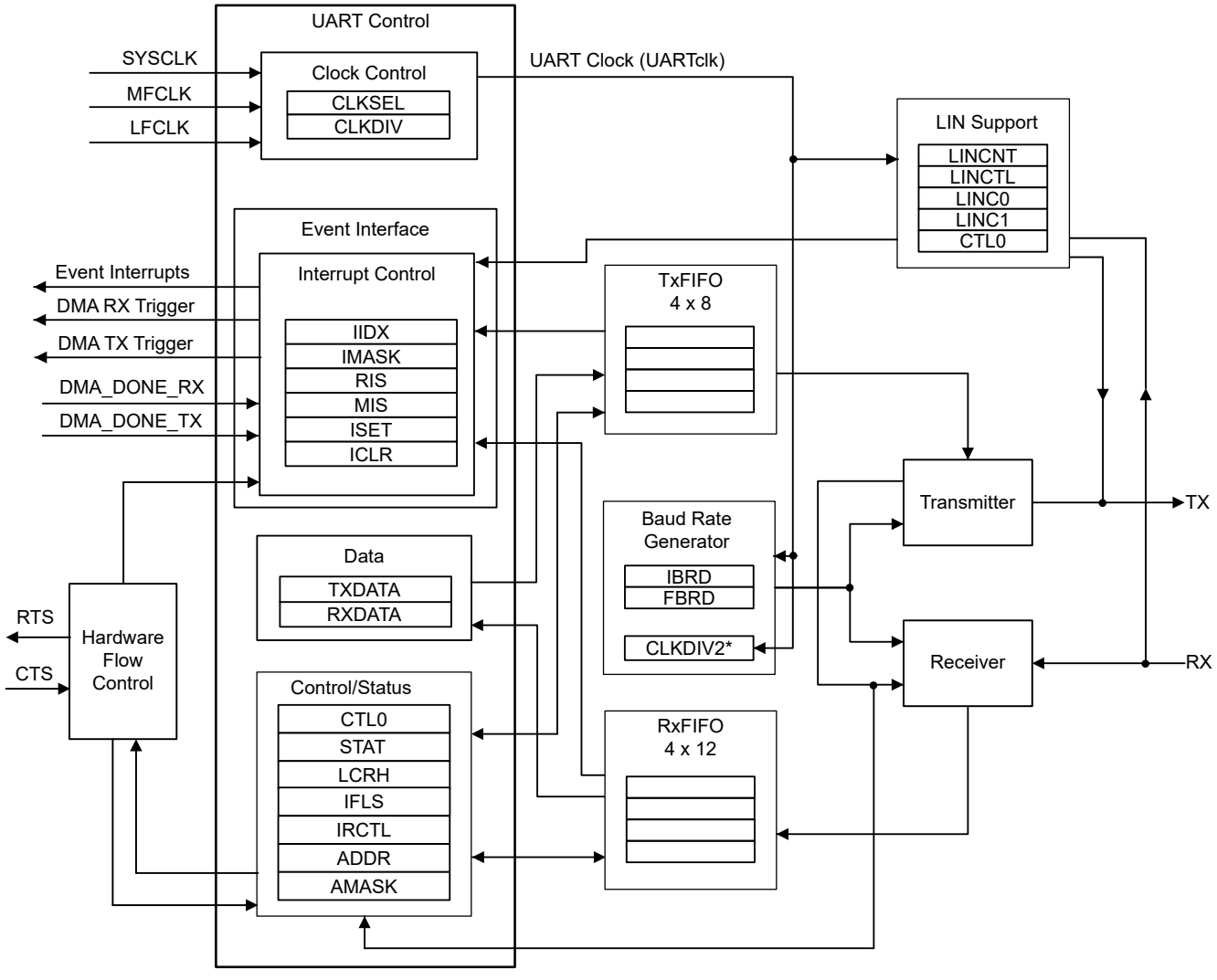
Copyright © 2026 Texas Instruments Incorporated

**Table 24-1. UART Extend and Main Features (1) (continued)**

| Features | UART Extend | UART Main (Low-Power Mode Support) | UART Main |
|---|---|---|---|
| RS-485 | Yes | Yes | Yes |
| Support LIN mode | Yes | - | - |
| Support DALI | Yes | - | - |
| Support IrDA | Yes | - | - |
| Support ISO7816 Smart card | Yes | - | - |
| Support Manchester code | Yes | - | - |

(1) Refer to the device specific data sheet for configuration and power domain details.,
(2) UART can be active in all low-power modes including STOP and STANDBY unless it is in power domain 1 (PD1). Refer to the device specific data sheet for configuration and power domain details.
(3) Only UART instances in power domain 0 (PD0) are active in STOP and STANDBY.

### 24.1.3 Functional Block Diagram




*CLKDIV2 is for IrDA mode only

**Figure 24-1. UART Functional Block Diagram**



Copyright © 2026 Texas Instruments Incorporated

## 24.2 UART Operation

This section describes the operation of the UART peripheral.

### *24.2.1 Clock Control*

The UART internal functional clock is selected and divided from the functional clock of the IP.

- Use UARTx.CLKSEL register to select the source of the UART functional clock.
  - BUSSCLK: the current bus clock is selected as the source for UART. The current bus clock depends on power domain. If the UART instance is in power domain 1 (PD1) refer to MCLK, if the UART instance is in power domain 0 (PD0) refer to ULPCLK.
  - MFCLK: MFCLK is selected as the source for UART, refer to MFCLK.
  - LFCLK: LFCLK is selected as the source for UART, refer to LFCLK
- Use UARTx.CLKDIV register to select the divide ratio of the UART function clock,options are from divide by 1 to 8. For UART Extend, there is a CLKDIV2 register to further divide the UART function clock to support the IrDA mode. When using IrDA mode, CLKDIV2.RATIO must be set to 1h for proper IrDA clocking.

The source clock and frequency depends on the power mode. See the Clock Module (CKM) section for more information. The module is ready to start receiving and transmitting data when the ENABLE bit is set.

### *24.2.2 Signal Descriptions*

UART communications require two pins: Receive Data (RX) and Transmit Data (TX):

- RX (Receive Data): RX is the serial data input. Glitch filter and oversampling are available.
- TX (Transmit Data): TX is the serial data output. The TX pin is held high if idling while the module is enabled. Bidirectional protocols like ISO7816 Smart card can use this pin to receive data.

The following pins are available in hardware flow control modes:

- CTS (Clear To Send): This signal blocks data transmission at the end of the current transfer when driven high by external signal
- RTS (Request To Send): This signal indicates the UART is ready to receive data when low.

### *24.2.3 General Architecture and Protocol*

UART transmits and receives characters asynchronously. Timing for each character is based on the selected baud-rate.

Control registers should only be programmed when the UART is disabled (ENABLE bit in the UARTx.CTL0 register is cleared). The current transaction gets completed before the UART stops if it is disabled during a transmission or reception. The baud-rate divisor registers (UARTx.IBRD and UARTx.FBRD) can be modified without disabling the UART.

#### 24.2.3.1 Transmit Receive Logic

The transmit logic performs parallel-to-serial conversion on the data read from the transmit FIFO. The control logic outputs the serial bit stream beginning with a start bit and followed by the data bits (LSB first), parity bit, and the stop bits according to the programmed configuration.

The receiver performs serial-to-parallel conversion on the received bit stream after a valid start pulse has been detected. Overrun, parity, frame error checking, and line-break detection are also performed, and their status accompanies the data that is written to the receive FIFO, this is the reason for 12-bit receive FIFO.

UARTx.CTL0.ENABLE bit is used to enable and disable the UART module, UARTx.CTL0.TXE and RXE bits are used to enable the transmit and receive mode, UARTx.LCRH.WLEN bit is used to configure the number of data bits transmitted or received in a frame, UARTx.LCRH.PEN is used to enable parity and UARTx.LCRH.STP2 is used to send two stop bits.



Copyright © 2026 Texas Instruments Incorporated

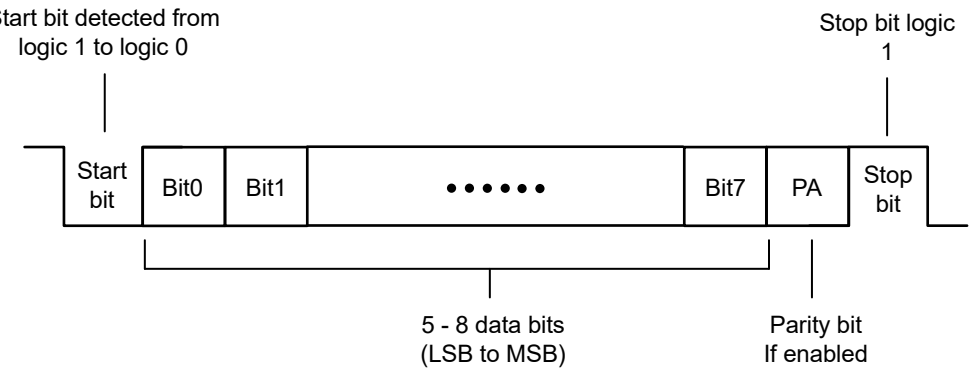



**Figure 24-2. UART Character Frame**

#### 24.2.3.2 Bit Sampling

UARTx.CTL0.HSE is set to 0, and 16 oversampling is selected by default. Received bits are expected to have the length of 16 UARTclk cycles. Data is sampled on the 8th UARTclk cycle.

Setting the UARTx.CTL0.HSE bit to 1 selects the 8 oversampling. Received bits are expected to have the length of 8 UARTclk cycles. Data is sampled on the 4th UARTclk cycle.

Setting the UARTx.CTL0.HSE bit to 2 selects the 3 oversampling. Received bits are expected to have the length of 3 UARTclk cycles. Data is sampled on the 2nd UARTclk cycle.

The previous scenarios assume IBRD = 1 and FBRD = 0.

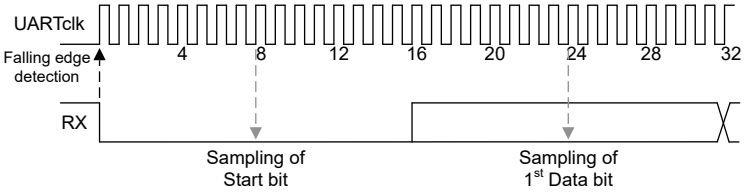
Depending on the application:

- Oversampling by 3 or 8 to achieves higher speeds with UARTclk/8 or UARTclk/3. This reduces the receiver's tolerance to clock deviation.
- Oversampling by 16 to increases the tolerance of the receiver to clock deviations. The maximum speed is limited to UARTclk/16.

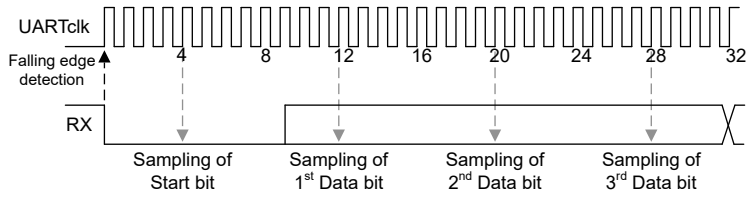


Copyright © 2026 Texas Instruments Incorporated

16x oversampling mode (HSE = 0)




8x oversampling mode (HSE = 1)




3x oversampling mode (HSE = 2)

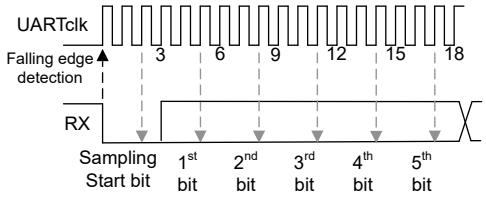



**Figure 24-3. UART Oversampling mode**

#### 24.2.3.3 Majority Voting Feature

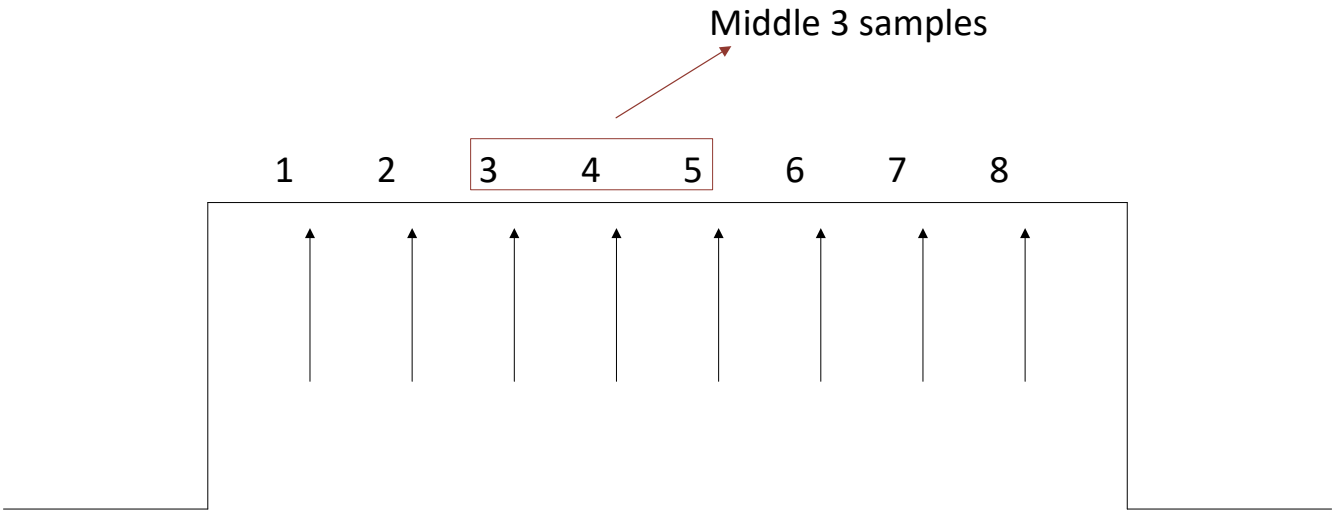



**Figure 24-4. Majority voting for 8 oversampling**

Majority voting provides noise immunity by sampling each bit 3 times in the center of the bit period.

Majority voting is only for data bits. Select majority voting or single sample using the MAJVOTE control bit in the CTL0 register. Disable majority voting if IrDA or Manchester Encoding modes are used.

**Example 1:** When the UARTx.CTL0.HSE is set to 0 with 16 oversampling; the 7th, 8th and 9th bit are sampled.



Copyright © 2026 Texas Instruments Incorporated

**Example 2:** When the UARTx.CTL0.HSE is set to 1 with 8 oversampling; then the 3rd ,4th and 5th bits are sampled and majority value is considered as final value to be sampled. Oversampling with majority voting is only applicable for 16 and 8 oversampling.

In both examples, the majority value is considered as the final value to be sampled.

When the 3 samples used for majority vote are not equal; the RIS.NE (noise error bit) is set. The received data is transferred despite the noise error. The NE bit appends to the received data before storing NE in the RXDATA register at bit position 12.

**Note**

Majority voting provides an additional layer of noise immunity on top of hardware noise filters built into the UART module.

#### 24.2.3.4 Baud Rate Generation

The baud-rate divisor is a 22-bit number consisting of a 16-bit integer, and a 6-bit fractional part. The number formed by these two values determines the bit sample period. The fractional baud-rate divisor allows the UART to accurately generate standard baud-rates.

The 16-bit integer is loaded into the UART Integer Baud-Rate Divisor UARTx.IBRD register, and the 6-bit fractional part is loaded into the UART Fractional Baud-Rate Divisor UARTx.FBRD register.

The baud-rate divisor can be calculated by using the following formula:

BRD = UART Clock / (Oversampling x Baud rate) (26)

The UART clock is configured by CLKSEL and CLKDIV. Oversampling (by 16, 8 or 3) is selected by the HSE bit in the UARTx.CTL0 register.

- UARTx.IBRD = INT(BRD), integer part of the BRD
- UARTx.FBRD = BRD % 64 , fractional part

**Note**

FBRD is ignored when IBRD = 0, and no data gets transferred by the UART. Similarly, when IBRD = 65535 (0xFFFF), then FBRD must not be greater than zero. Exceeding 65535 in FBRD results in an aborted transmission or reception.

The following example shows a simple method to calculate IBRD.DIVINT and FBRD.DIVFRAC for a baud rate of 19200 bit/s:

UART Clock = 40 MHz
Oversampling = 16
Baudrate = 19200 bit/s

$$BRD = \frac{UARTclk}{OVS \times Baudrate} = \frac{40MHz}{16 \times 19200\ bit/s} = 130.2083333$$

→ UARTx.IBRD.DIVINT= 130 (=82h)

→ UARTx.FBRRD.DIFRAC
= INT((.2083333 x 64) + 0.5)
= INT(13.833333)
= 13d (= Dh)

Note: The adder '+0.5' ensures rounding to the closest integer value to keep the rounding error as small as possible

**Figure 24-5. Baud Rate Configuration**



Copyright © 2026 Texas Instruments Incorporated

The UART.LCRH register must also be written when updating the baud-rate divisor (UARTx.IBRD or UARTx.IFRD). Changes to the baud-rate divisor must be followed by a write to the LCRH register for the changes to take effect.

The contents of the UART.IBRD and UART.FBRD registers are not updated until transmission or reception of the current character is complete.

**Note**

The UART receive interrupt timeout highest possible value (RXTOSEL) is calculated using the following formula: Timeout = F * 1/baud rate

#### 24.2.3.5 Data Transmission

Data received or transmitted is stored in two FIFOs. The receive FIFO has an extra four bits per character for status information.

**Transmit data:**

If the UART is enabled, and data is written to the FIFO, then a data frame starts transmitting with the parameters indicated in the UARTx.LCRH register. Data continues to be transmitted until there is no data left in the transmit FIFO. The BUSY bit in the UARTx.STAT register is asserted when data is written to the transmit FIFO (if the FIFO is not empty), and remains asserted while data is being transmitted. The BUSY bit is cleared when the transmit FIFO is empty (including the stop bits). The UART can indicate BUSY even though the UART can no longer be enabled. BUSY also is set during the generation of a BREAK signal.

**Receive data:**

When the receiver is idle (the RX signal is continuously 1), and the data input goes low (a start bit has been received), the receive counter begins running. Data is sampled on the different cycle based on the oversampling setting of the HSE bit in UARTx.CTL0 register.

The start bit is recognized if the RX signal is still low after certain number for cycles based on the oversampling setting. Successive data bits are sampled after a valid start bit is detected. The parity bit is checked if parity mode is enabled. Data length and parity are defined in the UART.LCRH register. Oversampling is explained in Section 24.2.3.2.

The valid stop bit is confirmed if the RXD signal is high, otherwise a framing error has occurred. The data is stored in the receive FIFO along with any error bits associated with that word when a full word is received.

#### 24.2.3.6 Error and Status

**For received data:** the data byte and the 4-bit status (break, frame, parity, and overrun) is pushed into the 12-bit wide receive FIFO. The error and status can be can be retrieved by reading UARTx.RXDATA as showed in Table 24-2.

**Table 24-2. UART Error and Conditions**

| Error Condition(1) | Bit Field | Description |
|---|---|---|
| Framing error | FRMERR | A framing error occurs when a low stop bit is detected. When two stop bits are used, both stop bits are checked for framing error. When a framing error is detected, the FRMERR bit is set. |
| Parity error | PARERR | A parity error is a mismatch between the number of 1s in a character and the value of the parity bit. When an address bit is included in the character, the address bit is included in the parity calculation. When a parity error is detected, the PARERR bit is set. |
| Receive overrun | OVRERR | An overrun error occurs when a character is loaded into RXDATA/FIFO before the prior character has been read. When an overrun occurs, the OVRERR bit is set. |
| Break condition | BRKERR | A break is detected when all received data, parity, and stop bits are 0. When a break condition is detected, the BRKERR bit is set. A break condition can also set the interrupt flag RXINT if the break interrupt enable IMASK.BRKERR bit is set. |

(1) Framing errors and break conditions are not set when LIN mode is enabled. This pattern is used to signal a Sync frame for LIN. Break detection is not available for IRDA, Manchester and Dali mode.



Copyright © 2026 Texas Instruments Incorporated

UART module flag status can also be checked by reading UARTx.STAT register as showing in Table 24-3.

**Table 24-3. UART Flag Status**

| Bit Field | Description |
|---|---|
| BUSY | This bit is set as soon as the transmit FIFO becomes nonempty (regardless of whether UART is enabled). In IDLE_Line mode the busy signal also stays set during the idle time generation. |
| RXFE | This bit is set when receive FIFO is empty. If the FIFO is disabled (FEN is 0), the receive holding register is full. If the FIFO is enabled (FEN is 1), the receive FIFO is full. |
| RXFF | This bit is set when receive FIFO is full. If the FIFO is disabled (FEN is 0), the receive holding register is empty. If the FIFO is enabled (FEN is 1), the receive FIFO is empty. |
| TXFE | This bit is set when transmit FIFO is empty. If the FIFO is disabled (FEN is 0), the transmit holding register is full. If the FIFO is enabled (FEN is 1), the transmit FIFO is full. |
| TXFF | This bit is set when transmit FIFO is full. If the FIFO is disabled (FEN is 0), the transmit holding register is empty. If the FIFO is enabled (FEN is 1), the transmit FIFO is empty. |
| CTS | This bit is set when CTS signal is asserted (low) and cleared when CTS signal is not asserted (high). |
| IDLE | IDLE mode has been detected in idle line multiprocessor mode. The IDLE bit is used as an address tag for each block of characters. In idle line multiprocessor format, this bit is set when a received character is an address. |

#### 24.2.3.7 Local Interconnect Network (LIN) Support

This section is only relevant for UART Extend, which supports LIN mode. Refer to the device data sheet for the device-specific configuration of UART extend and UART main.

For supporting local interconnect network (LIN) protocol, the following hardware enhancements are implemented in the UART module:

- 16 bit up-counter (LINCNT) clocked by the UART clock.
- Interrupt capability on counter overflow (CPU_INT.IMASK.LINOVF).
- 16 bit capture register (LINC0) with two configurable modes
  - Capture of LINCNT value on RXD falling edge. Interrupt capability on capture.
  - Compare of LINCNT with interrupt capability on match.
- 16 bit capture register (LINC1) can be configured:
  - Capture LINCNT value on RXD rising edge. Interrupt capability on capture.

**LIN transmission**

Sending the break signal can be done by setting the BRK bit in UARTx.LCRH register. This bit needs to be set before the data is written into the FIFO or transmit data register TXDATA.

To generate LIN responder signals such as wake signals, the TX pin can be configured by TXD_OUT and TXD_CTL_EN bits in register UARTx.CTL0 to be software controlled. By setting TXD_CTL_EN bit to 1, the TX output pin can be controlled by the TXD_OUT bit if the UART transmit is disabled (CTL0.TXE is cleared).

**LIN reception**

LIN commander issues a break field and sync field at the start of every frame. Hardware must be added such that the LIN responder software driver can reasonably detect BREAK-SYNC and measure the necessary timing parameters to adjust the baud rate or determine an error.

For LIN reception, break field detection and compare mode are needed. To configure these features:

1. Initialize LIN counter to 0 (UARTx.LINCNT = 0)
2. Enable counter compare match mode (UARTx.LINCTL.LINC0_MATCH = 1)
3. Load UARTx.LINC0 (counter capture 0 register) with counter value corresponding to 9.5 x $T_{bit}$ (refer to LIN Specification for details on this timing).
4. Enable LINC0 match interrupt (CPU_INT.IMASK.LINC0 = 1)
5. Setup LIN count control (UARTx.LINCTL):
   - Enable count while low signal on RXD (LINCTL.CNTRXLOW = 1)
   - Enable LIN counter clearing on RXD falling edge (LINCTL.ZERONE = 1)



Copyright © 2026 Texas Instruments Incorporated

- Enable LIN counter (LINCTL.CTRENA = 1)

Optional:

- User can enable the rising edge on UART TX signal interrupt (CPU_INT.IMASK.RXPE = 1), when the RXPE interrupt fires, the software can read the LINCTR directly to see the BREAK field timing.
- User can enable the LIN counter overflow interrupt (CPU_INT.IMASK.LINOVF = 1) to detect the BREAK field is too long and overflows 16bit counter. The timeout can be calculated as $t_{Timeout}$= $2^{16}$/ UART clock.

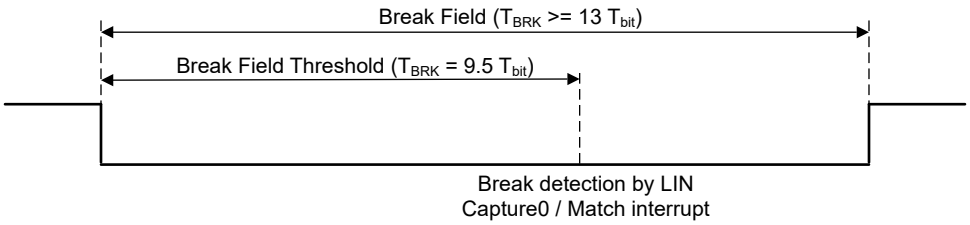



**Figure 24-6. LIN Break Field Detection**

Sync field validation is required to ensure accuracy of this LIN header part and to calculate the commander baud rate. The synch field consists of the data 0x55 inside a byte field (see Figure 24-7). After software detects a valid BREAK, it can then set the counter to measure the SYNC field. Both capture registers in LIN counter are used so that software sees fewer interrupts. Figure 24-7 shows the SYNC byte format. The LINCTR should be set to 0 on the start bit falling edge and count continuously. The LINC0 capture or RX falling edge interrupts fire at the falling edges of the RX line. During the interrupt processing, software can measure the individual HIGH-LOW times of the bits themselves using the values in the LINC0 and LINC1 registers to make sure all of the timings are valid.

The following flow describes a possible LIN sync field validation procedure:

1. Initialize LIN counter to 0 (UARTx.LINCNT = 0) after detecting a valid break field.
2. Enable interrupt on RX falling edge (CPU_INT.IMASK.RXNE = 1)
3. Setup LIN count control (LINCTL):
   - Enable LIN counter capture on raising RX edge (LINCTL.LINC1CAP = 1)
   - Enable LIN counter capture on falling RX edge (LINCTL.LINC0CAP = 1)
   - Enable LIN counter clearing on RX falling edge (LINCTL.ZERONE = 1)
   - Enable LIN counter (LINCTL.CTRENA = 1)

**Note**

If the device receives 0x0 data followed by a frame error then it can be mistaken for a break condition, and the data will not be moved to the FIFO.

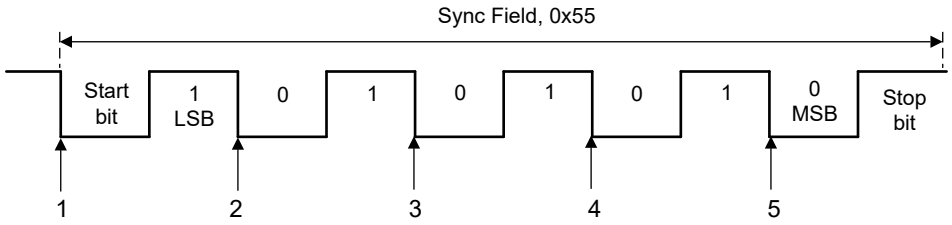



**Figure 24-7. LIN SYNC Field Detection**

Actions at each falling edge of the RX line for the sync field as showing below:

1. LIN counter is set to 0 and start counting on the falling RX edge. (LINCTL.ZERONE = 1)
2. RX falling edge interrupt trigger (RXNE):
   - Read capture register LINC0 (falling edge) and LINC1 (rising edge) values
   - Verify bit times



Copyright © 2026 Texas Instruments Incorporated

3. RX falling edge interrupt trigger (RXNE):
   - Read capture register LINC0 (falling edge) and LINC1 (rising edge) values
   - Verify bit times
4. RX falling edge interrupt trigger (RXNE):
   - Read capture register LINC0 (falling edge) and LINC1 (rising edge) values
   - Verify bit times
5. RX falling edge interrupt trigger (RXNE):
   - Read capture register LINC0 (falling edge) and LINC1 (rising edge) values
   - Verify bit times
   - Calculate the proper baud rate to set. Software must set the baud rate before the start bit of the PID field after sync field.

On each interrupt occurrence, the capture registers must be read and the bit times need to be validated by the application software. In case of a bit time verification error, the sync field analysis process must be aborted and the application software must switch back to break detection.

In case of errors like a breaking commander communication during sync field detection, a timeout interrupt can be generated by enabling the LIN counter overflow (IMASK.LINOVF = 1). When the counter overflows, the interrupt handler can abort the sync field analysis and switch back to break detection. The time the counter overflow interrupt occurs can be calculated as $t_{Timeout} = 2^{16}$/ UART clock.

**Note**

The sync field is automatically stored in the RX FIFO and can be misread as the PID. Therefore, flush the RX FIFO after the sync field is received and before the PID is received.

#### *24.2.3.7.1 LIN Responder Transmission Delay*

The interrupt RXINT for starting the transmission on the responder line always occurs at the mid-point of the commander STOP bit. Depending on the BUSCLK and baud-rate; there might not be enough response space between the STOP bit of commander and START bit of the responder; and the data transmission can start before end of STOP bit.

To overcome this, a delay of half the STOP bit period can be added as response space time to make sure that there is sufficient delay time between STOP and START bit, before start of responder data transmission.

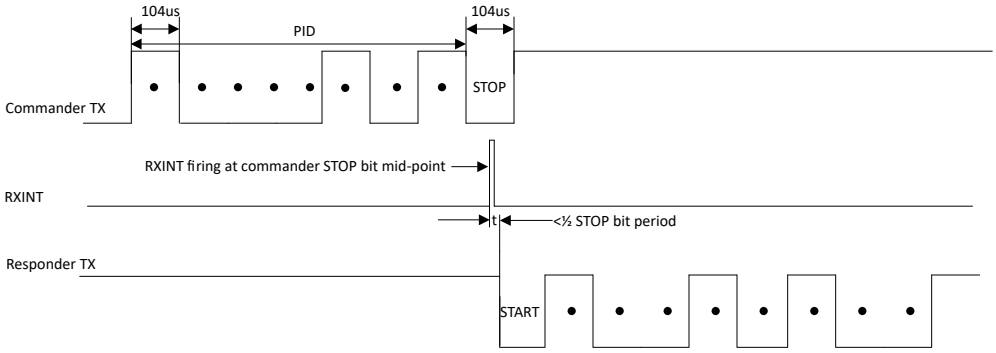



**Figure 24-8. LIN Responder Transmission Delay**

#### 24.2.3.8 Flow Control

Hardware flow control provides a mechanism for slow receivers to tell a transmitter to temporarily stop transmitting. This gives the receiver time to catch up, and prevent data loss.

Hardware flow control between two devices is accomplished in UART mode (CTL0.MODE set to 0), by connecting the RTS output to the CTS input on the receiving device, and connecting the RTS output on the receiving device to the CTS input. The RTS output signal is active low, so the CTS input expects a low signal on a send request as shown in Figure 24-9.



Copyright © 2026 Texas Instruments Incorporated

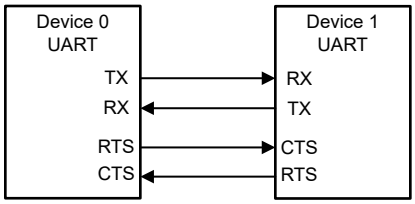



**Figure 24-9. Flow Control**

The CTS input controls the transmitter. Data can only be transmitted when the CTS input is set low. When the receiving device's buffer is full, then the receiver's RTS pin is set high. A high RTS pin (connected to CTS of the transmitter) causes transmission to pause.

**For example:** CTS of Device 1 remains low until the preprogrammed RX FIFO level of Device 0 is reached, indicating that the receive FIFO of Device 0 has no space to store additional characters.

The CTSEN and RTSEN bits in the UART.CTL0 register specify the flow control mode as shown in Table 24-4.

**Table 24-4. Flow Control Enable**

| CTSEN | RTSEN | Description |
|---|---|---|
| 1 | 1 | RTS and CTS flow control enabled |
| 1 | 0 | Only CTS flow control enabled |
| 0 | 1 | Only RTS flow control enabled |
| 0 | 0 | Both RTS and CTS flow control disabled |

When RTSEN is set to 1, the value of the CTL0.RTS bit is ignored and the RTS output signal is generated by the hardware trigger levels as described below. When RTSEN bit is cleared, the RTS signal output is controlled by the CTL0.RTS bit for SW control.

**RTS flow control:**

The RTS flow control logic is linked to the programmable receive FIFO watermark levels, it can be configured using UARTx.IFLS register. When RTS flow control is enabled, the RTS is asserted (low) until the receive FIFO is filled up to the watermark level. When the receive FIFO watermark level is reached, the RTS signal is de-asserted (high), indicating that there is no more room to receive any more data. The transmission of data is expected to cease after the current character has been transmitted. The RTS signal is reasserted (low) when data has been read out of the receive FIFO so that it is filled to less than the watermark level. If RTS flow control is disabled and the UART is still enabled, then data is received until the receive FIFO is full, or no more data is transmitted to it.

As the RTS signal is de-asserted when the FIFO watermark level is reached by putting the last received character into the FIFO. This means on a back to back transmit another character transfer could already been started by the sender. Therefore, in such cases the watermark level should be set to one level lower to ensure all data can be received and put into the FIFO.

**CTS flow control:**

If CTS flow control is enabled, then the transmitter checks the CTS signal before transmitting the next byte. If the CTS signal is asserted (low), it transmits the byte otherwise transmission does not occur. The data continues to be transmitted while CTS is asserted (low), and the transmit FIFO is not empty. If the transmit FIFO is empty and the CTS signal is asserted (low) no data is transmitted. If the CTS signal is de-asserted (high) and CTS flow control is enabled, then the current character transmission is completed before stopping. If CTS flow control is disabled and the UART is enabled, then the data continues to be transmitted until the transmit FIFO is empty.

#### 24.2.3.9 Idle-Line Multiprocessor

When IDLELINE is set in the CTL0.MODE register bits, the idle-line multiprocessor format is selected. Blocks of data are separated by an idle time on the transmit or receive lines (Figure 24-10). An idle receive line is



Copyright © 2026 Texas Instruments Incorporated

detected when ten or more continuous ones (marks) are received after the one or two stop bits of a character. The baud-rate generator is switched off after reception of an idle line until the next start edge is detected. When an idle line is detected, the IDLE bit in UARTx.STAT is set. In Idle-Line mode the UART receiver operates in no parity mode and the UART word length (UARTx.LCRH.WLEN) must be set to 8bit.

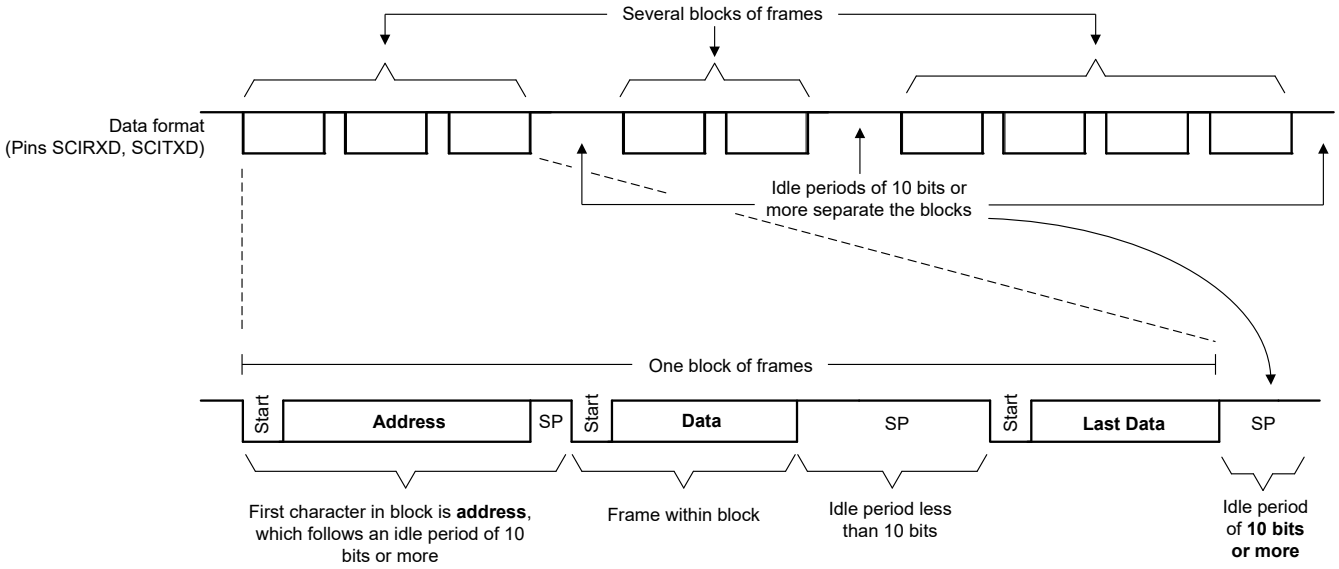



**Figure 24-10. Idle-Line Multiprocessor**

The first character received after an idle period is an address character. The IDLE bit in UARTx.STAT register is used as an address tag for each block of characters.

Address characters are compared against the ADDR register with the AMASK applied. If the received character matches, the address character and all following received characters are transferred into the RXDATA buffer. Interrupts/flags are generated until the next address without a match is received. The IDLE bit in UARTx.STAT register is automatically cleared when the address does match; otherwise the IDLE bit is set until the address is matched.

When there is an ongoing transaction, then a write into SENDIDLE is ignored. Application needs to set SENDIDLE when the UART is idle.

The following procedure sends out an idle frame to indicate an address character followed by associated data:

1. Set SENDIDLE then write the address character to TXDATA. TXDATA must be ready for new data (TXINT interrupt must be 1). This generates an idle period of exactly 11 bits followed by the address character. SENDIDLE is reset automatically when the address character has been transferred (all bits are sent out of shift register).
2. Write desired data characters to TXDATA. TXDATA must be ready for new data (TXINT interrupt must be 1). The data written to TXDATA is transferred to the shift register and transmitted as soon as the shift register is ready for new data.

The idle-line time must not be exceeded between address and data transmission or between data transmissions. Otherwise, the transmitted data is misinterpreted as an address.

BUSY bit in IDLELINE mode:

- TX: BUSY is set during the generation of an IDLELINE signal and while the address and data bytes are sent
- RX: The BUSY signal set during the receive of data and till first 10 idle bits are received.



Copyright © 2026 Texas Instruments Incorporated

#### 24.2.3.10 9-Bit UART Mode

9-bit mode allows the UART module to communicate in a multidrop mode. The additional bit tells the controller if the transmission is interpretted as an address or data.

9-bit mode is enabled by setting the MODE bit to ADDR9BIT in the UARTx.CTL0 register. In 9-bit UART mode, the parity enable/mode bits are ignored, and the UART word length (UARTx.LCRH.WLEN) must be set to 8 bit.

**Receive Transaction (9-bit mode):**

UART checks for the address qualifier at the location of the parity bit. If set, the received byte is compared with the preprogrammed address in UARTx.ADDR register:

- If the address matches, an interrupt (ADDR_MATCH) is generated, and the data becomes eligible for reception.
- If the address does not match, then the address and subsequent data bytes are dropped.

The address is set in the UART.ADDR register to match with the received byte. The matching can be extended to a set of addresses using the address mask in the UART.AMASK register. The UART.AMASK is 0xFF by default, so only the set address must match

**Transmit Transaction (9-bit mode):**

All send transactions in 9-bit UART mode are interpreted as either address or data bytes:

- Address bytes (if the 9th bit is set)
- Data bytes (if the 9th bit is cleared)

The 9th bit can be controlled by software. The EPS bit setting of the LCRH register reflects the 9th bit for transmit transactions. To indicate an address byte, the software must set the EPS bit before the byte transmission. To indicate a data byte, the EPS bit must be cleared before the byte transmission. For a complete transmit transaction, the address byte must be transmitted with EPS bit set followed by a data byte burst with EPS bit cleared.

**$9^{th}$ bit handling:**

**Table 24-5. 9th Bit Handling**

| PEN | SPS | EPS | $9^{th}$ Bit (Transmitted or Verified) |
|---|---|---|---|
| 0 | X | X | Not transmitted or verified |
| 1 | 1 | 0 | 0 (= Data) |
| 1 | 1 | 1 | 1 (= Address) |

#### 24.2.3.11 RS-485 Support

RS-485 can be used effectively over long distances, and in electrically noisy environments. Multiple receivers can be connected to such a network.

An external RS-485 PHY can be controlled using RTS as a direction signal. The signal is automatically set to high when a data transmit is started. It will stay set between bytes if they are sent back to back.

Data exchange sequence as show in Figure 24-11:

- Wait till an ongoing receive has been finished.
- Activate direction signal to transmit on RTS Pin
- Send data (one or more Bytes)
- Wait till an ongoing receive has been finished.
- Deactivate direction signal to transmit on RTS Pin



Copyright © 2026 Texas Instruments Incorporated

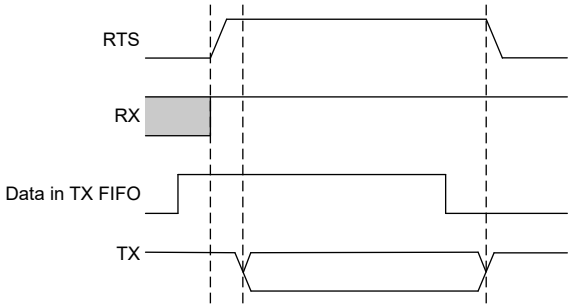



**Figure 24-11. RS485 Data Exchange**

Two bit fields in the LCRH register define the setup and hold time of the external driver direction control:

- EXTDIR_SETUP defines the number of UART clock ticks the signal to control the external driver for the RS485 will be set before the START bit is send. The generated setup time will be between EXDIR_SETUP value and EXDIR_SETUP + one baud rate cycle
- EXTDIR_HOLD bits defines the number of UART clock ticks the signal to control the external driver for the RS485 will be reset after the beginning of the stop bit. (If 2 STOP bits are enabled the beginning of the 2nd STOP bit.)

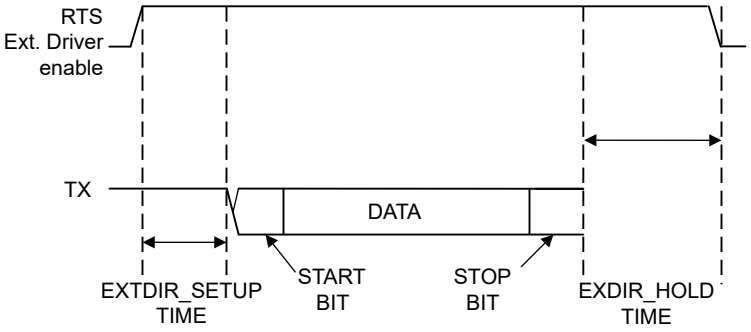



**Figure 24-12. RS485 External Control**

#### 24.2.3.12 DALI Protocol

Digital Addressable Lighting Interface (DALI) is an International Standard (IEC-62386) lighting control system for providing a single interface for lighting controllers.

The UART module supports sending and receiving forward and backward frames. The timing between any forward and backward frame sequence needs to be handled and checked by software.

**Transmitting a forward or backward frame:**

When transmitting a forward frame, a second byte must be written to the buffer before first byte has been shifted out (Otherwise, stop bits are sent like a backward frame). The hardware then sends the two bytes without inserting the stop bits.

**Receiving data:**

The UART module in DALI mode checks the 9th bit after the start bit to detect a forward or backward frame. If this bit has a change of phase (= no stop bit) a forward frame is detected and:

- Address compare is done in hardware based on AMASK
- Address and data are transferred into the RX buffer if there is an address match

Otherwise a backward frame is detected and the data is transferred into the RX Buffer without setting the ADDR_MATCH interrupt.



Copyright © 2026 Texas Instruments Incorporated

The AMASK register can be used as group assignment used during multicast operation with the MSB to indicate if the device is part of a DALI group. To enable the UART in DALI mode to respond on all ADDRESS the AMASK register needs to be cleared.

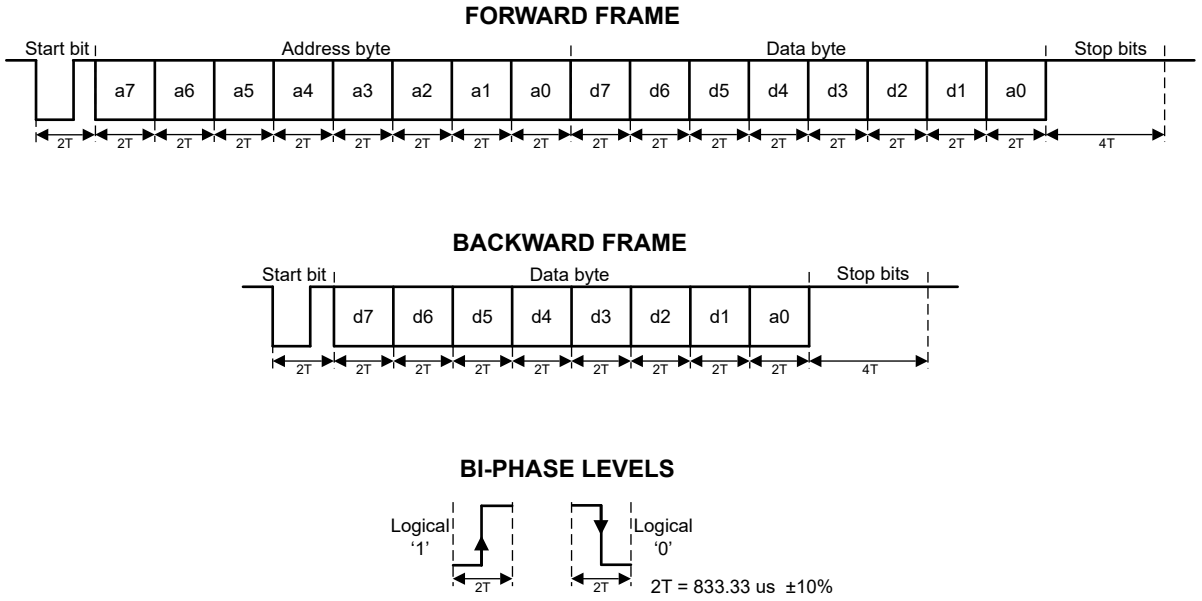



**Figure 24-13. DALI Protocol**

The limits for the times T shall be: 334 µs < T < 500 µs according to standard IEC 62386-102. DALI mode requires Manchester encoding to be enabled. When using DALI mode (CTL0.MODE set to DALI), the CTL0 and LCRH register must be set to:

- 8-bit word length (WLEN bit)
- No parity / 2 Stop Bit
- Manchester encoding enabled
- Baud-rate configuration set to match 2T = 833.33us
- DALI mode requires the FIFO to be enabled

**Note**

When UART is configured as a DALI Control Device: backward frame is stored in the device only if address match is enabled and there is an address match. Upon an address match, both forward frame and backward frame are stored in the receive FIFO and ADDR_MATCH interrupt flag is set.

**Note**

DALI timing support is restricted to +/- 2.0%

#### 24.2.3.13 Manchester Encoding and Decoding

UART can receive and transmit Manchester encoded data. Enable Manchester encoding using the MENC bit in the CTL0 register to generate the IEEE 802.3 compliant waveform. The GPIO control module can invert the signal to generate the G. E. Thomas compliant waveform.

The output signal is generated by XORing the data with UART clock signal. The UART clock needs to be two times faster than the baud-rate. There is an edge at the beginning and the middle of each transmitted data bit. The edge in the middle of the bit is detected to decode the received data.

#### 24.2.3.14 IrDA Encoding and Decoding

The IrDA encoder and decoder are enabled when IREN bit in UARTx.IRCTL register is set. IrDA can only be used with the UART mode (UARTx.CTL0.MODE is 0)



Copyright © 2026 Texas Instruments Incorporated

### IrDA Encoding

The encoder sends a pulse for every zero bit in the transmit bit stream coming from the UART (see Figure 24-14). The pulse duration is defined by IRTXPL. IRTXPL specifyies the number of one-half clock periods of the clock selected by IRTXCLK.

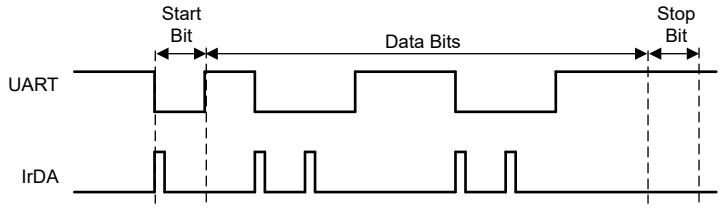



**Figure 24-14. IrDA Protocol**

### IrDA Decoding

The decoder detects high pulses when IRRXPL = 0, and the decoder detects low pulses when IRRXPL = 1.

A programmable digital filter stage can be enabled by setting UARTx.GFCTL.DGFSEL > 0. The digital glitch filter can be set when IRCTL.IREN is set, so only pulses longer than the programmed filter length are passed, and shorter pulses are discarded. (See the Glitch Suppression chapter on how to set the filter.)

#### 24.2.3.15 ISO7816 Smart Card Support

The UART offers basic support to allow communication with an ISO7816 smart card. The TXD signal is used as a bit clock when the MODE bits of UARTx.CTL0 are configured for smart card. The RXD signal is used as the half-duplex communication line connected to the smart card. Further smart card signals are not supported by the UART.

The clock rate of the UART clock in ISO7816 mode must be within 1MHz to 5MHz when using ISO7816 mode, and the UARTx.LCRH register must be set to:

- 8-bit word length (WLEN bits configured to 0x3)
- Even parity (PEN set and EPS set)
- No stick parity (SPS cleared)

The UART automatically uses 2 stop bits in ISO7816 mode; therefore the STP2 bit of the LCRH register is ignored.

If a parity error is detected during a transmission, then RXD is pulled low during the second stop bit. In that case, the UART aborts the transmission, flushes the transmit FIFO, discards any data it contains, and raises a parity error interrupt. The software can then initiate retransmission of the data.

The UART does not support automatic retransmission on parity errors. If a parity error is detected on transmission, then all further transmit operations are aborted, and software must handle retransmission.

If a parity error occurs, then the receiver drives the line low, and a parity interrupt flag is asserted. The transmitter responds based on the value of this bit.

#### 24.2.3.16 Address Detection

The UARTx.ADDR register sets the address that matches with the receiving address byte. This register is used in conjunction with UARTx.AMASK register. Only bits where the AMASK is set to '1' are considered. This feature is used in DALI, UART 9-Bit, or Idle-Line mode.

**Table 24-6. Address Detection**

| Condition | DALI Mode | Idle Line Mode | 9-Bit Mode |
|---|---|---|---|
| **Address match** | Address and Data is moved to RXDATA | Address and Data is moved to RXDATA | Address and Data is moved to RXDATA |



Copyright © 2026 Texas Instruments Incorporated

**Table 24-6. Address Detection (continued)**

| Condition | DALI Mode | Idle Line Mode | 9-Bit Mode |
|---|---|---|---|
| **Address mismatch** | Address and Data are be dropped | Address and Data are be dropped | Address and Data are be dropped |

#### 24.2.3.17 FIFO Operation

The UART has two FIFOs – one for receiving (RXDATA), and one for transmitting (TXDATA). Each FIFO has a depth of 4 entries. Reading RXDATA returns a 12-bit value consisting of 8 data bits and 4 error flags. Writing to TXDATA places 8-bit data bytes into the transmit FIFO.

Both FIFOs are disabled and act as 1-byte-deep holding registers upon leaving reset. The FIFOs are enabled by setting the FEN bit in UARTx.CTL0. FIFO status can be monitored through the UARTx.STAT register and interrupt events.

**Hardware monitors empty, full and overrun conditions**

The UARTx.STAT register contains empty and full flags (TXFE, TXFF, RXFE, and RXFF bits). The CPU_INT.RIS register shows overrun status of the receive FIFO through the OVRERR bit. There is no indicator for a transmit FIFO overrun. Writes are lost if data overruns the transmit FIFO. The empty and full flags are set according to the status of the 1-byte-deep holding registers if the FIFOs are disabled. The oldest data is overwritten with the new received data when receiving more data than the FIFO can capture.

The FIFOs generate interrupts at trigger points controlled through the UARTx.IFLS register. Both FIFOs can be individually configured to trigger interrupts at different points. Available configurations for the transmit FIFO are ¾, ½, ¼, and empty. Available configurations for the receive FIFO are ¼, ½, ¾, and full.

**For example:**If the 3/4 option is selected for the receive FIFO, the UART generates a receive interrupt after 3 data bytes are received.

Both FIFOs are configured to trigger an interrupt at the 1/2 mark directly following a reset. The FIFO integrity is indeterminate under the following conditions:

- After a UART Send Break has been initiated (LCRH.BRK = 1)
- Software disables the UART in the middle of a transmission with data in the FIFO, and then re-enables the UART.

#### 24.2.3.18 Loopback Operation

The UART can be placed into an internal loopback mode for diagnostic or debug work by setting the LBE bit in the UART.CTL0 register. The UART operates with the following behavior when in loopback mode:

- Data transmitted on the TX output is received on the RX input
- Data transmitted on the TX output is not propagated to the TX IO pin
- Data received on the RX IO pin is ignored

**Note**

If loopback mode is enabled, then the TX pin will still toggle



Copyright © 2026 Texas Instruments Incorporated

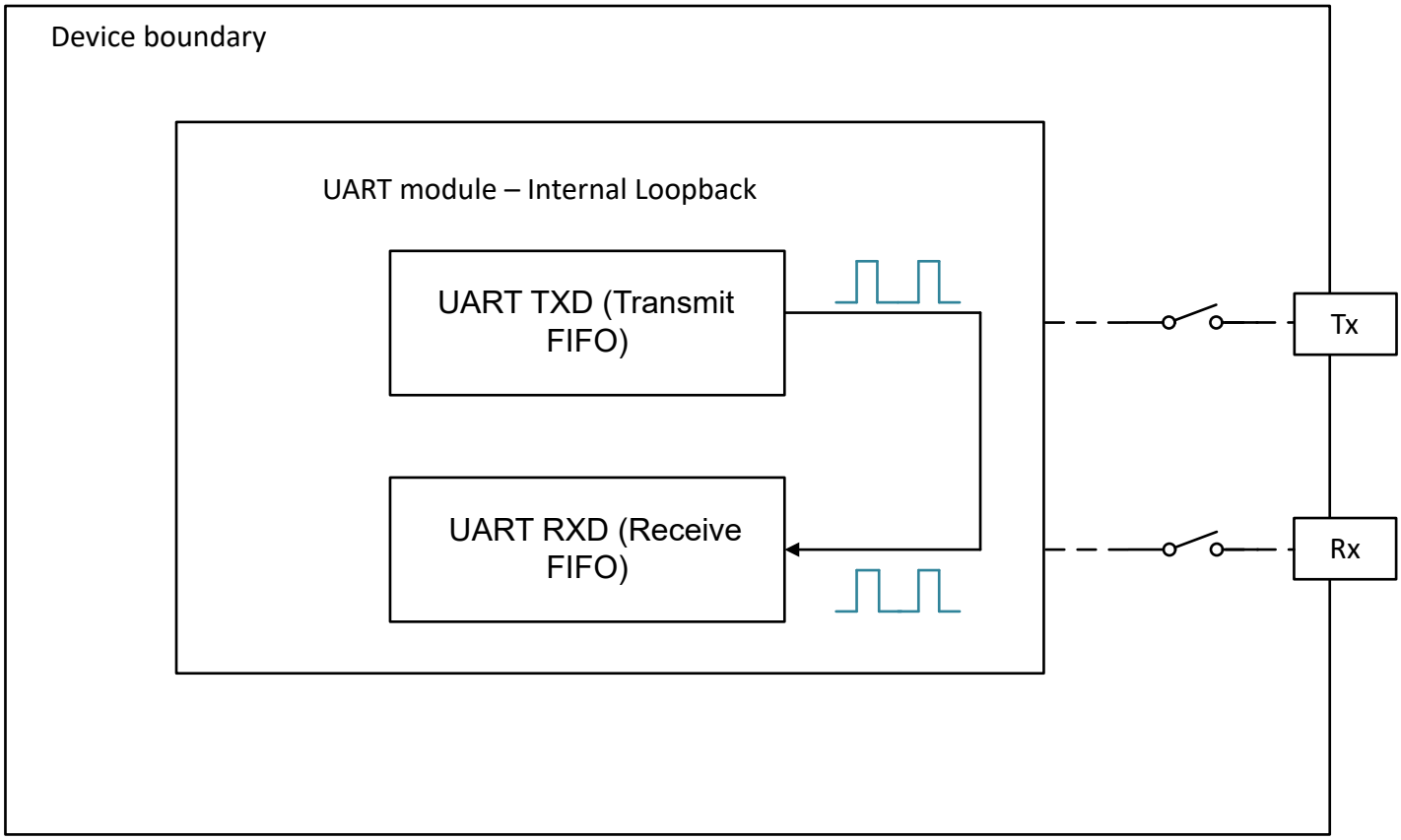



**Figure 24-15. UART Loopback Mode**

#### 24.2.3.19 Glitch Suppression

**Digital filter**

The digital filter is based on the UART functional clock. The DGFSEL bits in the UARTx.GFCTL register can be programmed to provide glitch suppression on the RX line. All signals are delayed internally when glitch suppression is nonzero.

**For Example:** If DGFSEL is set to 0x5, then 5 clocks are added onto the calculation for the expected transaction time. The DGFSEL needs to be configured for the glitch suppression pulse width to be shorter than 1/3 of a normal data pulse to avoid filtering a normal pulse.

**Analog filter**

The analog glitch suppression on the RX line is based on the analog glitch filter, and is selected with the AGFSEL bits in the UARTx.GFCTL register. See the device specific data sheet for selectable glitch filter values. The analog glitch filter is enabled with AGFEN. If AGFEN is not set, then input signals can not be filtered.

### *24.2.4 Low Power Operation*

When the module is instantiated in PD0 domain (always on); UART receive is supported in STOP/STANDBY.

1. Baud rate supported depends on the configured clock. *Refer to TRM chapter PMCU section Operating Modes*
2. The module can also request for a fast clock by asserting an asynchronous clock request upon detecting a START condition. Upon receiving an asynchronous clock request, SYSCTL provides SYSOSC at 32MHz in M0+ devices; for SYSOSC wakeup time refer to device datasheet
   a. Asynchronous clock request can be blocked by assert BLOCK_ASYNC_CLK_REQ; in this case, the module will not request for a fast clock and UART data reception will continue at the clock available in that particular power mode



Copyright © 2026 Texas Instruments Incorporated

b. Asynchronous clock is provided to the module as long as async_clk_req signal is asserted; typically, the duration of the UART frame and a few extra cycles for loading into FIFO and for FSM transition.
c. In case of glitches on the line, it is possible that the UART detects a START condition and requests for a fast clock. In this case, when the receive line is sampled again and an idle condition is observed, FSM moves back to idle and fast clock request is de-asserted and sysosc is turned off.

3. For M0+ devices, power mode transition from low power mode to ACTIVE is only upon receiving an enabled interrupt.

### *24.2.5 Reset Considerations*

#### Software Reset Considerations

A software reset can be executed by setting RESETASSERT with the KEY in the RSTCTL register. An ongoing transfer is terminated immediately, but leaves the software in an undefined state. Stop transmitting before initiating a software reset.

#### Hardware Reset Considerations

A hardware reset also initializes the IO configuration. This sets the IOs to a high impedance state, and leaves the data lines floating. Pull up or pull down resistors can be required if the application is sensitive to floating data lines.

### *24.2.6 Initialization*

Clear the ENABLE bit before setup to avoid unpredictable behavior:

Use the following steps to configure and enable the UART:

1. Configure RX and TX pin functions by using the IOMUX registers.
2. Reset the peripheral using UARTx.RSTCTL register
3. Enable the power to UART peripheral using the UARTx.PWREN register
4. Select the UART function clock source and divide options using UART.CLKSEL and UART.CLKDIV registers.
5. Disable the UART by clearing the UART.CTL0.ENABLE bit.
6. Use the baud-rate equation in Section 24.2.3.4 to calculate the UARTx.IBRD and UARTx.FBRD registers.
7. Write the integer portion of the BRD to the UART.IBRD register.
8. Write the fractional portion of the BRD to the UART.FBRD register.
9. Write the desired oversampling and FIFO configuration to the UART.CTL0 register
10. Write the desired serial parameters to the UART.LCRH register.
11. Enable the UART by setting the UART.CTL0.ENABLE bit.

### *24.2.7 Interrupt and Events Support*

The UART module contains three event publishers and no event subscribers. One event publisher (CPU_INT) manages UART interrupt requests (IRQs) to the CPU subsystem through a static event route. The second and third event publishers (DMA_TRIG_RX, DMA_TRIG_TX) are used to setup the trigger signaling for the DMA through DMA event route.

The UART events are summarized in Table 24-7.

**Table 24-7. UART Events**

| Event | Type | Source | Destination | Route | Configuration | Functionality |
|---|---|---|---|---|---|---|
| CPU interrupt | Publisher | UART | CPU Subsystem | Static route | CPU_INT registers | Fixed interrupt route from UART to CPU |
| DMA trigger | Publisher | UART | DMA | DMA event route | DMA_TRIG_RX registers | Fixed interrupt route from UART to DMA |



Copyright © 2026 Texas Instruments Incorporated

**Table 24-7. UART Events (continued)**

| Event | Type | Source | Destination | Route | Configuration | Functionality |
|---|---|---|---|---|---|---|
| DMA trigger | Publisher | UART | DMA | DMA event route | DMA_TRIG_TX registers | Fixed interrupt route from UART to DMA |

#### 24.2.7.1 CPU Interrupt Event Publisher (CPU_INT)

The UART module provides 18 interrupt sources which can be configured to source a CPU interrupt event. In order of decreasing interrupt priority, the CPU interrupt events from the UART are:

**Table 24-8. UART CPU Interrupt Event Conditions (CPU_INT)**

| IIDX STAT | Name | Description |
|---|---|---|
| 0x01 | RTOUT | UART receive timeout interrupt, This interrupt is asserted when the receive FIFO is not empty, and no further data is received specified time in the UARTx.IFLS.RXTOSEL bits. More information provided below. |
| 0x02 | FRMERR | UART framing error interrupt, see Section 24.2.3.6 for more information. |
| 0x03 | PARERR | UART parity error interrupt, see Section 24.2.3.6 for more information. |
| 0x04 | BRKERR | UART break error interrupt, see Section 24.2.3.6 for more information. |
| 0x05 | OVRERR | UART receive overrun error interrupt, see Section 24.2.3.6 for more information. |
| 0x06 | RXNE | Falling edge on RX interrupt, this interrupt triggers when there is a falling edge on RX line. |
| 0x07 | RXPE | Rising edge on RX interrupt, this interrupt triggers when there is a rising edge on RX line. |
| 0x08 | LINC0 | LIN capture 0 match interrupt, this interrupt triggers when the defined capture 0 value is reached in LIN counter. |
| 0x09 | LINC1 | LIN capture 1 match interrupt, this interrupt triggers when the defined capture 1 value is reached in LIN counter. |
| 0x0A | LINOVF | LIN counter overflow interrupt, this interrupt triggers when the 16bit LIN counter overflows. |
| 0x0B | RXINT | UART receive interrupt. More information provided below. |
| 0x0C | TXINT | UART transmit interrupt. More information provided below. |
| 0x0D | EOT | UART end of transmission interrupt, it indicates that the last bit of all transmitted data and status has left the serializer and without any further data in the TX FIFO. |
| 0x0E | ADDR_MATCH | Address match interrupt, used in protocols with address to indicate address match happened. |
| 0x0F | CTS | UART clear to send interrupt, indicate the CTS signal status. |
| 0x10 | DMA_DONE_RX | This interrupt is set if the RX DMA channel sends the DONE signal. |
| 0x11 | DMA_DONE_TX | This interrupt is set if the TX DMA channel sends the DONE signal. |

The CPU interrupt event configuration is managed with the CPU_INT event management registers. See Section 7.2.5 for guidance on configuring the Event registers for CPU interrupts.

The receive timeout interrupt is asserted when the receive FIFO is not empty, and no further data is received specified time in the IFLS.RXTOSEL bits. The receive timeout interrupt is cleared either when the FIFO becomes empty through reading all the data (or by reading the holding register), by reading the interrupt index from IIDX or when a 1 is written to the corresponding bit in the ICLR register.

The receive interrupt (RXINT, 0x0B) changes state when one of the following events occurs:

- If the FIFOs are enabled and the receive FIFO reaches the programmed trigger level, the RXINT bit is set. The receive interrupt is cleared by reading data from the receive FIFO until it becomes less than the trigger level, by reading the interrupt index from IIDX or by writing a 1 to the RXINT bit in ICLR.
- If the FIFOs are disabled (have a depth of one location) and data is received thereby filling the location, the RXINT bit is set. The receive interrupt is cleared by performing a single read of the receive FIFO, by reading the interrupt index from IIDX or by writing a 1 to the RXINT bit in ICLR.

The transmit interrupt (TXINT, 0x0C) changes state when one of the following events occurs:



Copyright © 2026 Texas Instruments Incorporated

- If the FIFOs are enabled and the transmit FIFO progresses through the programmed trigger level, the TXINT bit is set. The transmit interrupt is based on a transition through level, therefore the FIFO must be written past the programmed trigger level otherwise no further transmit interrupts will be generated. The transmit interrupt is cleared by writing data to the transmit FIFO until it becomes greater than the trigger level, by reading the interrupt index from IIDX or by writing a 1 to the TXINT bit in ICLR.
- If the FIFOs are disabled (have a depth of one location) and there is no data present in the transmitters single location, the TXINT bit is set. It is cleared by performing a single write to the transmit FIFO, by reading the interrupt index from IIDX or by writing a 1 to the TXINT bit in ICLR.

#### 24.2.7.2 DMA Trigger Publisher (DMA_TRIG_RX, DMA_TRIG_TX)

DMA_TRIG_RX and DMA_TRIG_TX registers are used to setup the trigger signaling for the DMA. This can be setup in a flexible way to trigger the DMA for receive or transmit events with the trigger conditions in Table 24-9and Table 24-10.

DMA_TRIG_RX is used for triggering the DMA to do a receive data transfer and DMA_TRIG_TX is used for triggering the DMA to do a transmit data transfer.

**Table 24-9. UART DMA Trigger Condition (DMA_TRIG_RX)**

| IIDX STAT | Name | Description |
|---|---|---|
| 0x01 | RTOUT | UART receive timeout interrupt, This interrupt is asserted when the receive FIFO is not empty, and no further data is received specified time in the UARTx.IFLS.RXTOSEL bits. More information provided below. |
| 0x0B | RXINT | UART receive interrupt. More information provided below. |

The receive timeout interrupt is asserted when the receive FIFO is not empty, and no further data is received specified time in the IFLS.RXTOSEL bits. The receive timeout interrupt is cleared either when the FIFO becomes empty through reading all the data (or by reading the holding register), by reading the interrupt index from IIDX or when a 1 is written to the corresponding bit in the ICLR register.

The receive interrupt (RXINT, 0x0B) changes state when one of the following events occurs:

- If the FIFOs are enabled and the receive FIFO reaches the programmed trigger level, the RXINT bit is set. The receive interrupt is cleared by reading data from the receive FIFO until it becomes less than the trigger level, by reading the interrupt index from IIDX or by writing a 1 to the RXINT bit in ICLR.
- If the FIFOs are disabled (have a depth of one location) and data is received thereby filling the location, the RXINT bit is set. The receive interrupt is cleared by performing a single read of the receive FIFO, by reading the interrupt index from IIDX or by writing a 1 to the RXINT bit in ICLR.

**Table 24-10. UART DMA Trigger Condition (DMA_TRIG_TX)**

| IIDX STAT | Name | Description |
|---|---|---|
| 0x0C | TXINT | UART transmit interrupt. More information provided below. |

The transmit interrupt (TXINT, 0x0C) changes state when one of the following events occurs:

- If the FIFOs are enabled and the transmit FIFO progresses through the programmed trigger level, the TXINT bit is set. The transmit interrupt is based on a transition through level, therefore the FIFO must be written past the programmed trigger level otherwise no further transmit interrupts will be generated. The transmit interrupt is cleared by writing data to the transmit FIFO until it becomes greater than the trigger level, by reading the interrupt index from IIDX or by writing a 1 to the TXINT bit in ICLR.
- If the FIFOs are disabled (have a depth of one location) and there is no data present in the transmitters single location, the TXINT bit is set. It is cleared by performing a single write to the transmit FIFO, by reading the interrupt index from IIDX or by writing a 1 to the TXINT bit in ICLR.

The DMA trigger event configuration is managed with the DMA_TRIG_RX and DMA_TRIG_TX event management registers. See Section 7.2.5 for guidance on configuring the Event registers and Section 7.1.3.2 for on how DMA trigger event works.



Copyright © 2026 Texas Instruments Incorporated

### 24.2.8 Emulation Modes

The module behavior while the device is in debug mode is controlled by the FREE and SOFT bits in PDBGCTL register.

When the device is in debug mode and set into halt mode below behavior can be configured.

**Table 24-11. Debug Mode Peripheral Behavior**

| PDBGCTL.FREE | PDBGCTL.SOFT | Function |
|---|---|---|
| **1** | **x** | Modules continues operation |
| **0** | **0** | Module stops immediately |
| **0** | **1** | Module stops after the next transfer has been finished |



Copyright © 2026 Texas Instruments Incorporated

## 24.3 UART Registers

Table 24-12 lists the memory-mapped registers for the UART registers. All register offset addresses not listed in Table 24-12 should be considered as reserved locations and the register contents should not be modified.

**Table 24-12. UART Registers**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 800h | PWREN | Power enable | | Go |
| 804h | RSTCTL | Reset Control | | Go |
| 808h | CLKCFG | Peripheral Clock Configuration Register | | Go |
| 814h | STAT | Status Register | | Go |
| 1000h | CLKDIV | Clock Divider | | Go |
| 1008h | CLKSEL | Clock Select for Ultra Low Power peripherals | | Go |
| 1018h | PDBGCTL | Peripheral Debug Control | | Go |
| 1020h | IIDX | Interrupt index | CPU_INT | Go |
| 1028h | IMASK | Interrupt mask | CPU_INT | Go |
| 1030h | RIS | Raw interrupt status | CPU_INT | Go |
| 1038h | MIS | Masked interrupt status | CPU_INT | Go |
| 1040h | ISET | Interrupt set | CPU_INT | Go |
| 1048h | ICLR | Interrupt clear | CPU_INT | Go |
| 1050h | IIDX | Interrupt index | DMA_TRIG_RX | Go |
| 1058h | IMASK | Interrupt mask | DMA_TRIG_RX | Go |
| 1060h | RIS | Raw interrupt status | DMA_TRIG_RX | Go |
| 1068h | MIS | Masked interrupt status | DMA_TRIG_RX | Go |
| 1070h | ISET | Interrupt set | DMA_TRIG_RX | Go |
| 1078h | ICLR | Interrupt clear | DMA_TRIG_RX | Go |
| 1080h | IIDX | Interrupt index | DMA_TRIG_TX | Go |
| 1088h | IMASK | Interrupt mask | DMA_TRIG_TX | Go |
| 1090h | RIS | Raw interrupt status | DMA_TRIG_TX | Go |
| 1098h | MIS | Masked interrupt status | DMA_TRIG_TX | Go |
| 10A0h | ISET | Interrupt set | DMA_TRIG_TX | Go |
| 10A8h | ICLR | Interrupt clear | DMA_TRIG_TX | Go |
| 10E0h | EVT_MODE | Event Mode | | Go |
| 10E4h | INTCTL | Interrupt control register | | Go |
| 1100h | CTL0 | UART Control Register 0 | | Go |
| 1104h | LCRH | UART Line Control Register | | Go |
| 1108h | STAT | UART Status Register | | Go |
| 110Ch | IFLS | UART Interrupt FIFO Level Select Register | | Go |
| 1110h | IBRD | UART Integer Baud-Rate Divisor Register | | Go |



Copyright © 2026 Texas Instruments Incorporated

**Table 24-12. UART Registers (continued)**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 1114h | FBRD | UART Fractional Baud-Rate Divisor Register | | Go |
| 1118h | GFCTL | Glitch Filter Control | | Go |
| 1120h | TXDATA | UART Transmit Data Register | | Go |
| 1124h | RXDATA | UART Receive Data Register | | Go |
| 1130h | LINCNT | UART LIN Mode Counter Register | | Go |
| 1134h | LINCTL | UART LIN Mode Control Register | | Go |
| 1138h | LINC0 | UART LIN Mode Capture 0 Register | | Go |
| 113Ch | LINC1 | UART LIN Mode Capture 1 Register | | Go |
| 1140h | IRCTL | eUSCI_Ax IrDA Control Word Register | | Go |
| 1148h | AMASK | Self Address Mask Register | | Go |
| 114Ch | ADDR | Self Address Register | | Go |
| 1160h | CLKDIV2 | Clock Divider | | Go |

Complex bit access types are encoded to fit into small table cells. Table 24-13 shows the codes that are used for access types in this section.

**Table 24-13. UART Access Type Codes**

| Access Type | Code | Description |
|---|---|---|
| **Read Type** | | |
| R | R | Read |
| **Write Type** | | |
| W | W | Write |
| WK | W<br>K | Write<br>Write protected by a key |
| **Reset or Default Value** | | |
| *-n* | | Value after reset or the default value |



Copyright © 2026 Texas Instruments Incorporated

### 24.3.1 PWREN (Offset = 800h) [Reset = 00000000h]

PWREN is shown in Figure 24-16 and described in Table 24-14.

Return to the Summary Table.

Register to control the power state

**Figure 24-16. PWREN**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | ENABLE |
| R/W-0h | | | | | | | R/WK-0h |

**Table 24-14. PWREN Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | KEY to allow Power State Change<br>26h = KEY to allow write access to this register |
| 23-1 | RESERVED | R/W | 0h | |
| 0 | ENABLE | R/WK | 0h | Enable the power<br>KEY must be set to 26h to write to this bit.<br>0h = Disable Power<br>1h = Enable Power |



Copyright © 2026 Texas Instruments Incorporated

### 24.3.2 RSTCTL (Offset = 804h) [Reset = 00000000h]

RSTCTL is shown in Figure 24-17 and described in Table 24-15.

Return to the Summary Table.

Register to control reset assertion and de-assertion

**Figure 24-17. RSTCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | RESETSTKYCLR | RESETASSERT |
| W-0h | | | | | | WK-0h | WK-0h |

**Table 24-15. RSTCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | Unlock key<br>B1h = KEY to allow write access to this register |
| 23-2 | RESERVED | W | 0h | |
| 1 | RESETSTKYCLR | WK | 0h | Clear the RESETSTKY bit in the STAT register<br>KEY must be set to B1h to write to this bit.<br>0h = Writing 0 has no effect<br>1h = Clear reset sticky bit |
| 0 | RESETASSERT | WK | 0h | Assert reset to the peripheral<br>KEY must be set to B1h to write to this bit.<br>0h = Writing 0 has no effect<br>1h = Assert reset |



Copyright © 2026 Texas Instruments Incorporated

### 24.3.3 CLKCFG (Offset = 808h) [Reset = 00000000h]

CLKCFG is shown in Figure 24-18 and described in Table 24-16.

Return to the Summary Table.

Peripheral Clock Configuration Register

**Figure 24-18. CLKCFG**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | BLOCKASYNC |
| R/W-0h | | | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |

**Table 24-16. CLKCFG Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | KEY to Allow State Change -- 0xA9<br>A9h = 0xA9 |
| 23-9 | RESERVED | R/W | 0h | |
| 8 | BLOCKASYNC | R/W | 0h | Async Clock Request is blocked from starting SYSOSC or forcing bus clock to 32MHz<br>0h = 0<br>1h = 1 |
| 7-0 | RESERVED | R/W | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 24.3.4 STAT (Offset = 814h) [Reset = 00000000h]

STAT is shown in Figure 24-19 and described in Table 24-17.

Return to the Summary Table.

Reset status register

**Figure 24-19. STAT**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R- | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | RESETSTKY |
| R- | | | | | | | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R- | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | |
| R- | | | | | | | |

**Table 24-17. STAT Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-17 | RESERVED | R | 0h | |
| 16 | RESETSTKY | R | 0h | This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register<br>0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register<br>1h = The peripheral was reset since the last bit clear |
| 15-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 24.3.5 CLKDIV (Offset = 1000h) [Reset = 00000000h]

CLKDIV is shown in Figure 24-20 and described in Table 24-18.

Return to the Summary Table.

This register is used to specify module-specific divide ratio of the functional clock

**Figure 24-20. CLKDIV**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | | RATIO | | |
| R/W-0h | | | | | | | | | | | | | R/W-0h | | |

**Table 24-18. CLKDIV Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | RESERVED | R/W | 0h | |
| 2-0 | RATIO | R/W | 0h | Selects divide ratio of module clock<br>0h = Do not divide clock source<br>1h = Divide clock source by 2<br>2h = Divide clock source by 3<br>3h = Divide clock source by 4<br>4h = Divide clock source by 5<br>5h = Divide clock source by 6<br>6h = Divide clock source by 7<br>7h = Divide clock source by 8 |



Copyright © 2026 Texas Instruments Incorporated

### 24.3.6 CLKSEL (Offset = 1008h) [Reset = 00000000h]

CLKSEL is shown in Figure 24-21 and described in Table 24-19.

Return to the Summary Table.

Clock source selection for peripherals

**Figure 24-21. CLKSEL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | BUSCLK_SEL | MFCLK_SEL | LFCLK_SEL | RESERVED |
| R/W-0h | | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

**Table 24-19. CLKSEL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R/W | 0h | |
| 3 | BUSCLK_SEL | R/W | 0h | Selects BUS CLK as clock source if enabled<br>0h = Does not select this clock as a source<br>1h = Select this clock as a source |
| 2 | MFCLK_SEL | R/W | 0h | Selects MFCLK as clock source if enabled<br>0h = Does not select this clock as a source<br>1h = Select this clock as a source |
| 1 | LFCLK_SEL | R/W | 0h | Selects LFCLK as clock source if enabled<br>0h = Does not select this clock as a source<br>1h = Select this clock as a source |
| 0 | RESERVED | R/W | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 24.3.7 PDBGCTL (Offset = 1018h) [Reset = 00000003h]

PDBGCTL is shown in Figure 24-22 and described in Table 24-20.

Return to the Summary Table.

This register can be used by the software developer to control the behavior of the peripheral relative to the 'Core Halted' input

**Figure 24-22. PDBGCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R/W- | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R/W- | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R/W- | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | SOFT | FREE |
| R/W- | | | | | | R/W-1h | R/W-1h |

**Table 24-20. PDBGCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | R/W | 0h | |
| 1 | SOFT | R/W | 1h | Soft halt boundary control. This function is only available, if FREE is set to 'STOP'<br>0h = The peripheral will halt immediately, even if the resultant state will result in corruption if the system is restarted<br>1h = The peripheral blocks the debug freeze until it has reached a boundary where it can resume without corruption |
| 0 | FREE | R/W | 1h | Free run control<br>0h = The peripheral freezes functionality while the Core Halted input is asserted and resumes when it is deasserted.<br>1h = The peripheral ignores the state of the Core Halted input |



Copyright © 2026 Texas Instruments Incorporated

### 24.3.8 IIDX (Offset = 1020h) [Reset = 00000000h]

IIDX is shown in Figure 24-23 and described in Table 24-21.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. Value 0x00 means no event pending. Interrupt 1 is the highest priority, IIDX next highest, 4, 8, … IIDX^31 is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred. On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in [RIS] and [MIS] are cleared as well. After a read from the CPU (not from the debug interface), the register is updated with the next highest priority interrupt, if none are pending, then it should display 0x0.

**Figure 24-23. IIDX**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | STAT | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | |

**Table 24-21. IIDX Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | STAT | R | 0h | UART Module Interrupt Vector Value. This register provides the highest priority interrupt index. A read clears the corresponding interrupt flag in RIS and MIS registers. 15h-1Fh = Reserved<br>00h = No interrupt pending<br>01h = UART receive time-out interrupt; Interrupt Flag: RT; Interrupt Priority: Highest<br>02h = UART framing error interrupt; Interrupt Flag: FE<br>03h = UART parity error interrupt; Interrupt Flag: PE<br>04h = UART break error interrupt; Interrupt Flag: BE<br>05h = UART receive overrun error interrupt; Interrupt Flag: OE<br>06h = Negative edge on UARTxRXD interrupt; Interrupt Flag: RXNE<br>07h = Positive edge on UARTxRXD interrupt; Interrupt Flag: RXPE<br>08h = LIN capture 0 / match interrupt; Interrupt Flag: LINC0<br>09h = LIN capture 1 interrupt; Interrupt Flag: LINC1<br>0Ah = LIN hardware counter overflow interrupt; Interrupt Flag: LINOVF<br>0Bh = UART receive interrupt; Interrupt Flag: RX<br>0Ch = UART transmit interrupt; Interrupt Flag: TX<br>0Dh = UART end of transmission interrupt (transmit serializer empty); Interrupt Flag: EOT<br>0Eh = 9-bit mode address match interrupt; Interrupt Flag: MODE_9B<br>Fh = UART Clear to Send Modem interrupt; Interrupt Flag: CTS<br>10h = DMA DONE on RX<br>11h = DMA DONE on TX<br>12h = Noise Error Event |



Copyright © 2026 Texas Instruments Incorporated

### 24.3.9 IMASK (Offset = 1028h) [Reset = 00000000h]

IMASK is shown in Figure 24-24 and described in Table 24-22.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is unmasked. Unmasking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

**Figure 24-24. IMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | NERR | DMA_DONE_TX |
| R/W-0h | | | | | | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DMA_DONE_RX | CTS | ADDR_MATCH | EOT | TXINT | RXINT | LINOVF | LINC1 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LINC0 | RXPE | RXNE | OVRERR | BRKERR | PARERR | FRMERR | RTOUT |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

**Table 24-22. IMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-18 | RESERVED | R/W | 0h | |
| 17 | NERR | R/W | 0h | Noise Error on triple voting. Asserted when the 3 samples of majority voting are not equal<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 16 | DMA_DONE_TX | R/W | 0h | Enable DMA Done on TX Event Channel Interrupt<br>0h = Interrupt disabled<br>1h = Set Interrupt Mask |
| 15 | DMA_DONE_RX | R/W | 0h | Enable DMA Done on RX Event Channel Interrupt<br>0h = Interrupt disabled<br>1h = Set Interrupt Mask |
| 14 | CTS | R/W | 0h | Enable UART Clear to Send Modem Interrupt.<br>0h = Interrupt disabled<br>1h = Set Interrupt Mask |
| 13 | ADDR_MATCH | R/W | 0h | Enable Address Match Interrupt.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 12 | EOT | R/W | 0h | Enable UART End of Transmission Interrupt Indicates that the last bit of all transmitted data and flags has left the serializer and without any further Data in the TX FIFO or Buffer.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 11 | TXINT | R/W | 0h | Enable UART Transmit Interrupt.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 10 | RXINT | R/W | 0h | Enable UART Receive Interrupt.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |



Copyright © 2026 Texas Instruments Incorporated

**Table 24-22. IMASK Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 9 | LINOVF | R/W | 0h | Enable LIN Hardware Counter Overflow Interrupt.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 8 | LINC1 | R/W | 0h | Enable LIN Capture 1 Interrupt.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 7 | LINC0 | R/W | 0h | Enable LIN Capture 0 / Match Interrupt .<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 6 | RXPE | R/W | 0h | Enable Positive Edge on UARTxRXD Interrupt.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 5 | RXNE | R/W | 0h | Enable Negative Edge on UARTxRXD Interrupt.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 4 | OVRERR | R/W | 0h | Enable UART Receive Overrun Error Interrupt.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 3 | BRKERR | R/W | 0h | Enable UART Break Error Interrupt.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 2 | PARERR | R/W | 0h | Enable UART Parity Error Interrupt.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 1 | FRMERR | R/W | 0h | Enable UART Framing Error Interrupt.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 0 | RTOUT | R/W | 0h | Enable UARTOUT Receive Time-Out Interrupt.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |



Copyright © 2026 Texas Instruments Incorporated

### 24.3.10 RIS (Offset = 1030h) [Reset = 00000000h]

RIS is shown in Figure 24-25 and described in Table 24-23.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Figure 24-25. RIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R- | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | NERR | DMA_DONE_TX |
| R- | | | | | | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DMA_DONE_RX | CTS | ADDR_MATCH | EOT | TXINT | RXINT | LINOVF | LINC1 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LINC0 | RXPE | RXNE | OVRERR | BRKERR | PARERR | FRMERR | RTOUT |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 24-23. RIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-18 | RESERVED | R | 0h | |
| 17 | NERR | R | 0h | Noise Error on triple voting. Asserted when the 3 samples of majority voting are not equal<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 16 | DMA_DONE_TX | R | 0h | DMA Done on TX Event Channel Interrupt<br>0h = Interrupt disabled<br>1h = Interrupt occurred |
| 15 | DMA_DONE_RX | R | 0h | DMA Done on RX Event Channel Interrupt<br>0h = Interrupt disabled<br>1h = Interrupt occurred |
| 14 | CTS | R | 0h | UART Clear to Send Modem Interrupt.<br>0h = Interrupt disabled<br>1h = Interrupt occurred |
| 13 | ADDR_MATCH | R | 0h | Address Match Interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 12 | EOT | R | 0h | UART End of Transmission Interrupt Indicates that the last bit of all transmitted data and flags has left the serializer and without any further Data in the TX FIFO or Buffer.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 11 | TXINT | R | 0h | UART Transmit Interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 10 | RXINT | R | 0h | UART Receive Interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

**Table 24-23. RIS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 9 | LINOVF | R | 0h | LIN Hardware Counter Overflow Interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 8 | LINC1 | R | 0h | LIN Capture 1 Interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 7 | LINC0 | R | 0h | LIN Capture 0 / Match Interrupt .<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 6 | RXPE | R | 0h | Positive Edge on UARTxRXD Interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 5 | RXNE | R | 0h | Negative Edge on UARTxRXD Interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 4 | OVRERR | R | 0h | UART Receive Overrun Error Interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 3 | BRKERR | R | 0h | UART Break Error Interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 2 | PARERR | R | 0h | UART Parity Error Interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 1 | FRMERR | R | 0h | UART Framing Error Interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 0 | RTOUT | R | 0h | UARTOUT Receive Time-Out Interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

### 24.3.11 MIS (Offset = 1038h) [Reset = 00000000h]

MIS is shown in Figure 24-26 and described in Table 24-24.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 24-26. MIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | NERR | DMA_DONE_TX |
| R-0h | | | | | | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DMA_DONE_RX | CTS | ADDR_MATCH | EOT | TXINT | RXINT | LINOVF | LINC1 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LINC0 | RXPE | RXNE | OVRERR | BRKERR | PARERR | FRMERR | RTOUT |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 24-24. MIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-18 | RESERVED | R | 0h | |
| 17 | NERR | R | 0h | Noise Error on triple voting. Asserted when the 3 samples of majority voting are not equal<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 16 | DMA_DONE_TX | R | 0h | Masked DMA Done on TX Event Channel Interrupt<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 15 | DMA_DONE_RX | R | 0h | Masked DMA Done on RX Event Channel Interrupt<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 14 | CTS | R | 0h | Masked UART Clear to Send Modem Interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 13 | ADDR_MATCH | R | 0h | Masked Address Match Interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 12 | EOT | R | 0h | UART End of Transmission Interrupt Indicates that the last bit of all transmitted data and flags has left the serializer and without any further Data in the TX FIFO or Buffer.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 11 | TXINT | R | 0h | Masked UART Transmit Interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 10 | RXINT | R | 0h | Masked UART Receive Interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

**Table 24-24. MIS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 9 | LINOVF | R | 0h | Masked LIN Hardware Counter Overflow Interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 8 | LINC1 | R | 0h | Masked LIN Capture 1 Interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 7 | LINC0 | R | 0h | Masked LIN Capture 0 / Match Interrupt .<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 6 | RXPE | R | 0h | Masked Positive Edge on UARTxRXD Interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 5 | RXNE | R | 0h | Masked Negative Edge on UARTxRXD Interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 4 | OVRERR | R | 0h | Masked UART Receive Overrun Error Interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 3 | BRKERR | R | 0h | Masked UART Break Error Interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 2 | PARERR | R | 0h | Masked UART Parity Error Interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 1 | FRMERR | R | 0h | Masked UART Framing Error Interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 0 | RTOUT | R | 0h | Masked UARTOUT Receive Time-Out Interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

### 24.3.12 ISET (Offset = 1040h) [Reset = 00000000h]

ISET is shown in Figure 24-27 and described in Table 24-25.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 24-27. ISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | NERR | DMA_DONE_TX |
| W-0h | | | | | | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DMA_DONE_RX | CTS | ADDR_MATCH | EOT | TXINT | RXINT | LINOVF | LINC1 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LINC0 | RXPE | RXNE | OVRERR | BRKERR | PARERR | FRMERR | RTOUT |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

**Table 24-25. ISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-18 | RESERVED | W | 0h | |
| 17 | NERR | W | 0h | Noise Error on triple voting. Asserted when the 3 samples of majority voting are not equal<br>0h = Writing this has no effect<br>1h = Set the interrupt |
| 16 | DMA_DONE_TX | W | 0h | Set DMA Done on TX Event Channel Interrupt<br>0h = Interrupt disabled<br>1h = Set Interrupt |
| 15 | DMA_DONE_RX | W | 0h | Set DMA Done on RX Event Channel Interrupt<br>0h = Interrupt disabled<br>1h = Set Interrupt |
| 14 | CTS | W | 0h | Set UART Clear to Send Modem Interrupt.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 13 | ADDR_MATCH | W | 0h | Set Address Match Interrupt.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 12 | EOT | W | 0h | Set UART End of Transmission Interrupt Indicates that the last bit of all transmitted data and flags has left the serializer and without any further Data in the TX FIFO or Buffer.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 11 | TXINT | W | 0h | Set UART Transmit Interrupt.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 10 | RXINT | W | 0h | Set UART Receive Interrupt.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |



Copyright © 2026 Texas Instruments Incorporated

**Table 24-25. ISET Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 9 | LINOVF | W | 0h | Set LIN Hardware Counter Overflow Interrupt.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 8 | LINC1 | W | 0h | Set LIN Capture 1 Interrupt.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 7 | LINC0 | W | 0h | Set LIN Capture 0 / Match Interrupt .<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 6 | RXPE | W | 0h | Set Positive Edge on UARTxRXD Interrupt.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 5 | RXNE | W | 0h | Set Negative Edge on UARTxRXD Interrupt.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 4 | OVRERR | W | 0h | Set UART Receive Overrun Error Interrupt.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 3 | BRKERR | W | 0h | Set UART Break Error Interrupt.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 2 | PARERR | W | 0h | Set UART Parity Error Interrupt.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 1 | FRMERR | W | 0h | Set UART Framing Error Interrupt.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 0 | RTOUT | W | 0h | Set UARTOUT Receive Time-Out Interrupt.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 24.3.13 ICLR (Offset = 1048h) [Reset = 00000000h]

ICLR is shown in Figure 24-28 and described in Table 24-26.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

**Figure 24-28. ICLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | NERR | DMA_DONE_TX |
| W-0h | | | | | | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DMA_DONE_RX | CTS | ADDR_MATCH | EOT | TXINT | RXINT | LINOVF | LINC1 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LINC0 | RXPE | RXNE | OVRERR | BRKERR | PARERR | FRMERR | RTOUT |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

**Table 24-26. ICLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-18 | RESERVED | W | 0h | |
| 17 | NERR | W | 0h | Noise Error on triple voting. Asserted when the 3 samples of majority voting are not equal<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 16 | DMA_DONE_TX | W | 0h | Clear DMA Done on TX Event Channel Interrupt<br>0h = Interrupt disabled<br>1h = Clear Interrupt |
| 15 | DMA_DONE_RX | W | 0h | Clear DMA Done on RX Event Channel Interrupt<br>0h = Interrupt disabled<br>1h = Clear Interrupt |
| 14 | CTS | W | 0h | Clear UART Clear to Send Modem Interrupt.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 13 | ADDR_MATCH | W | 0h | Clear Address Match Interrupt.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 12 | EOT | W | 0h | Clear UART End of Transmission Interrupt Indicates that the last bit of all transmitted data and flags has left the serializer and without any further Data in the TX FIFO or Buffer.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 11 | TXINT | W | 0h | Clear UART Transmit Interrupt.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 10 | RXINT | W | 0h | Clear UART Receive Interrupt.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |



Copyright © 2026 Texas Instruments Incorporated

**Table 24-26. ICLR Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 9 | LINOVF | W | 0h | Clear LIN Hardware Counter Overflow Interrupt.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 8 | LINC1 | W | 0h | Clear LIN Capture 1 Interrupt.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 7 | LINC0 | W | 0h | Clear LIN Capture 0 / Match Interrupt .<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 6 | RXPE | W | 0h | Clear Positive Edge on UARTxRXD Interrupt.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 5 | RXNE | W | 0h | Clear Negative Edge on UARTxRXD Interrupt.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 4 | OVRERR | W | 0h | Clear UART Receive Overrun Error Interrupt.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 3 | BRKERR | W | 0h | Clear UART Break Error Interrupt.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 2 | PARERR | W | 0h | Clear UART Parity Error Interrupt.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 1 | FRMERR | W | 0h | Clear UART Framing Error Interrupt.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 0 | RTOUT | W | 0h | Clear UARTOUT Receive Time-Out Interrupt.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 24.3.14 IIDX (Offset = 1050h) [Reset = 00000000h]

IIDX is shown in Figure 24-29 and described in Table 24-27.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. Value 0x00 means no event pending. Interrupt 1 is the highest priority, IIDX next highest, 4, 8, … IIDX^31 is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred. On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in [RIS] and [MIS] are cleared as well. After a read from the CPU (not from the debug interface), the register is updated with the next highest priority interrupt, if none are pending, then it should display 0x0.

**Figure 24-29. IIDX**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | STAT | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | |

**Table 24-27. IIDX Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | STAT | R | 0h | UART Module Interrupt Vector Value. This register provides the highest priority interrupt index. A read clears the corresponding interrupt flag in UARTRIS and UARTMISC. 15h-1Fh = Reserved<br>00h = No interrupt pending<br>01h = UART receive time-out interrupt; Interrupt Flag: RT; Interrupt Priority: Highest<br>0Bh = UART receive interrupt; Interrupt Flag: RX |



Copyright © 2026 Texas Instruments Incorporated

### 24.3.15 IMASK (Offset = 1058h) [Reset = 00000000h]

IMASK is shown in Figure 24-30 and described in Table 24-28.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is unmasked. Unmasking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

**Figure 24-30. IMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | RXINT | RESERVED | |
| R/W-0h | | | | | R/W-0h | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | RTOUT |
| R/W-0h | | | | | | | R/W-0h |

**Table 24-28. IMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-11 | RESERVED | R/W | 0h | |
| 10 | RXINT | R/W | 0h | Enable UART Receive Interrupt.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 9-1 | RESERVED | R/W | 0h | |
| 0 | RTOUT | R/W | 0h | Enable UARTOUT Receive Time-Out Interrupt.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |



Copyright © 2026 Texas Instruments Incorporated

### 24.3.16 RIS (Offset = 1060h) [Reset = 00000000h]

RIS is shown in Figure 24-31 and described in Table 24-29.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Figure 24-31. RIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R- | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R- | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | RXINT | RESERVED | |
| R- | | | | | R-0h | R- | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | RTOUT |
| R- | | | | | | | R-0h |

**Table 24-29. RIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-11 | RESERVED | R | 0h | |
| 10 | RXINT | R | 0h | UART Receive Interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 9-1 | RESERVED | R | 0h | |
| 0 | RTOUT | R | 0h | UARTOUT Receive Time-Out Interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

### 24.3.17 MIS (Offset = 1068h) [Reset = 00000000h]

MIS is shown in Figure 24-32 and described in Table 24-30.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 24-32. MIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | RXINT | RESERVED | |
| R-0h | | | | | R-0h | R-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | RTOUT |
| R-0h | | | | | | | R-0h |

**Table 24-30. MIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-11 | RESERVED | R | 0h | |
| 10 | RXINT | R | 0h | Masked UART Receive Interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 9-1 | RESERVED | R | 0h | |
| 0 | RTOUT | R | 0h | Masked UARTOUT Receive Time-Out Interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

### 24.3.18 ISET (Offset = 1070h) [Reset = 00000000h]

ISET is shown in Figure 24-33 and described in Table 24-31.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 24-33. ISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | RXINT | RESERVED | |
| W-0h | | | | | W-0h | W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | RTOUT |
| W-0h | | | | | | | W-0h |

**Table 24-31. ISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-11 | RESERVED | W | 0h | |
| 10 | RXINT | W | 0h | Set UART Receive Interrupt.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 9-1 | RESERVED | W | 0h | |
| 0 | RTOUT | W | 0h | Set UARTOUT Receive Time-Out Interrupt.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 24.3.19 ICLR (Offset = 1078h) [Reset = 00000000h]

ICLR is shown in Figure 24-34 and described in Table 24-32.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

**Figure 24-34. ICLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | RXINT | RESERVED | |
| W-0h | | | | | W-0h | W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | RTOUT |
| W-0h | | | | | | | W-0h |

**Table 24-32. ICLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-11 | RESERVED | W | 0h | |
| 10 | RXINT | W | 0h | Clear UART Receive Interrupt.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 9-1 | RESERVED | W | 0h | |
| 0 | RTOUT | W | 0h | Clear UARTOUT Receive Time-Out Interrupt.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 24.3.20 IIDX (Offset = 1080h) [Reset = 00000000h]

IIDX is shown in Figure 24-35 and described in Table 24-33.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. Value 0x00 means no event pending. Interrupt 1 is the highest priority, IIDX next highest, 4, 8, … IIDX^31 is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred. On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in [RIS] and [MIS] are cleared as well. After a read from the CPU (not from the debug interface), the register is updated with the next highest priority interrupt, if none are pending, then it should display 0x0.

**Figure 24-35. IIDX**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | STAT | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | |

**Table 24-33. IIDX Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | STAT | R | 0h | UART Module Interrupt Vector Value. This register provides the highest priority interrupt index. A read clears the corresponding interrupt flag in UARTRIS and UARTMISC. 15h-1Fh = Reserved<br>00h = No interrupt pending<br>0Ch = UART transmit interrupt; Interrupt Flag: TX |



Copyright © 2026 Texas Instruments Incorporated

### 24.3.21 IMASK (Offset = 1088h) [Reset = 00000000h]

IMASK is shown in Figure 24-36 and described in Table 24-34.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is unmasked. Unmasking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

**Figure 24-36. IMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | TXINT | RESERVED | | |
| R/W-0h | | | | R/W-0h | R/W-0h | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |

**Table 24-34. IMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-12 | RESERVED | R/W | 0h | |
| 11 | TXINT | R/W | 0h | Enable UART Transmit Interrupt.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 10-0 | RESERVED | R/W | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 24.3.22 RIS (Offset = 1090h) [Reset = 00000000h]

RIS is shown in Figure 24-37 and described in Table 24-35.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Figure 24-37. RIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R- | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R- | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | TXINT | RESERVED | | |
| R- | | | | R-0h | R- | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | |
| R- | | | | | | | |

**Table 24-35. RIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-12 | RESERVED | R | 0h | |
| 11 | TXINT | R | 0h | UART Transmit Interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 10-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 24.3.23 MIS (Offset = 1098h) [Reset = 00000000h]

MIS is shown in Figure 24-38 and described in Table 24-36.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 24-38. MIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | TXINT | RESERVED | | |
| R-0h | | | | R-0h | R-0h | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |

**Table 24-36. MIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-12 | RESERVED | R | 0h | |
| 11 | TXINT | R | 0h | Masked UART Transmit Interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 10-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 24.3.24 ISET (Offset = 10A0h) [Reset = 00000000h]

ISET is shown in Figure 24-39 and described in Table 24-37.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 24-39. ISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | TXINT | RESERVED | | |
| W-0h | | | | W-0h | W-0h | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |

**Table 24-37. ISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-12 | RESERVED | W | 0h | |
| 11 | TXINT | W | 0h | Set UART Transmit Interrupt.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 10-0 | RESERVED | W | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 24.3.25 ICLR (Offset = 10A8h) [Reset = 00000000h]

ICLR is shown in Figure 24-40 and described in Table 24-38.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

**Figure 24-40. ICLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | TXINT | RESERVED | | |
| W-0h | | | | W-0h | W-0h | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |

**Table 24-38. ICLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-12 | RESERVED | W | 0h | |
| 11 | TXINT | W | 0h | Clear UART Transmit Interrupt.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 10-0 | RESERVED | W | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 24.3.26 EVT_MODE (Offset = 10E0h) [Reset = 00000029h]

EVT_MODE is shown in Figure 24-41 and described in Table 24-39.

Return to the Summary Table.

Event mode register. It is used to select whether each line is disabled, in software mode (software clears the RIS) or in hardware mode (hardware clears the RIS)

**Figure 24-41. EVT_MODE**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R/W- | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R/W- | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R/W- | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | INT2_CFG | | INT1_CFG | | INT0_CFG | |
| R/W- | | R-2h | | R-2h | | R-1h | |

**Table 24-39. EVT_MODE Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | RESERVED | R/W | 0h | |
| 5-4 | INT2_CFG | R | 2h | Event line mode select for event corresponding to DMA_TRIG_TX<br>0h = The interrupt or event line is disabled.<br>1h = The interrupt or event line is in software mode. Software must clear the RIS.<br>2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag. |
| 3-2 | INT1_CFG | R | 2h | Event line mode select for event corresponding to DMA_TRIG_RX<br>0h = The interrupt or event line is disabled.<br>1h = The interrupt or event line is in software mode. Software must clear the RIS.<br>2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag. |
| 1-0 | INT0_CFG | R | 1h | Event line mode select for event corresponding to CPU_INT<br>0h = The interrupt or event line is disabled.<br>1h = The interrupt or event line is in software mode. Software must clear the RIS.<br>2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag. |



Copyright © 2026 Texas Instruments Incorporated

### 24.3.27 INTCTL (Offset = 10E4h) [Reset = 00000000h]

INTCTL is shown in Figure 24-42 and described in Table 24-40.

Return to the Summary Table.

Interrupt control register

**Figure 24-42. INTCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R/W- | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R/W- | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R/W- | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | INTEVAL |
| R/W- | | | | | | | W-0h |

**Table 24-40. INTCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R/W | 0h | |
| 0 | INTEVAL | W | 0h | Writing a 1 to this field re-evaluates the interrupt sources.<br>0h = The interrupt or event line is disabled.<br>1h = The interrupt or event line is in software mode. Software must clear the RIS. |



Copyright © 2026 Texas Instruments Incorporated

### 24.3.28 CTL0 (Offset = 1100h) [Reset = 00000038h]

CTL0 is shown in Figure 24-43 and described in Table 24-41.

Return to the Summary Table.

UART Control Register
The CTL0 register is the control register. All the bits are cleared on reset except for the Transmit Enable (TXE) and Receive Enable (RXE) bits, which are set. To enable the UART module, the UARTEN bit must be set. If software requires a configuration change in the module, the UARTEN bit must be cleared before the configuration changes are written. If the UART is disabled during a transmit or receive operation, the current transaction is completed prior to the UART stopping. NOTE: The CTL0 register should not be changed while the UART is enabled or else the results are unpredictable. The following sequence is recommended for making changes to the CTL0 register.
1. Disable the UART.
2. Wait for the end of transmission or reception of the current character.
3. Flush the transmit FIFO by clearing bit FEN in the UART control register CTL0.
4. Reprogram the control register.
5. Enable the UART.

**Figure 24-43. CTL0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | MSBFIRST | MAJVOTE | FEN | HSE |
| R/W-0h | | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| HSE | CTSEN | RTSEN | RTS | RESERVED | MODE | | |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MENC | TXD_OUT | TXD_OUT_EN | TXE | RXE | LBE | RESERVED | ENABLE |
| R/W-0h | R/W-0h | R/W-1h | R/W-1h | R/W-1h | R/W-0h | R/W-0h | R/W-0h |

**Table 24-41. CTL0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | RESERVED | R/W | 0h | |
| 19 | MSBFIRST | R/W | 0h | Most Significant Bit First<br>This bit has effect both on the way protocol byte is transmitted and received.<br>Notes: User needs to match the protocol to the correct value of this bit to send MSb or LSb first. The hardware engine will send the byte entirely based on this bit.<br>0h = Least significant bit is sent first in the protocol packet<br>1h = Most significant bit is sent first in the protocol packet |
| 18 | MAJVOTE | R/W | 0h | Majority Vote Enable<br>When Majority Voting is enabled, the three center bits are used to determine received sample value. In case of error (all 3 bits are not the same), noise error is detected and bits RIS.NERR and register RXDATA.NERR are set.<br>Oversampling of 16 : bits 7, 8, 9 are used<br>Oversampling of 8 : bits 3, 4, 5 are used<br>Disabled : Single sample value (center value) used<br>0h = Majority voting is disabled<br>1h = Majority voting is enabled |



Copyright © 2026 Texas Instruments Incorporated

**Table 24-41. CTL0 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 17 | FEN | R/W | 0h | UART Enable FIFOs<br>0h = The FIFOs are disabled (Character mode). The FIFOs become 1-byte-deep holding registers.<br>1h = The transmit and receive FIFO buffers are enabled (FIFO mode). |
| 16-15 | HSE | R/W | 0h | High-Speed Bit Oversampling Enable **NOTE:** The bit oversampling influences the UART baud-rate configuration. The state of this bit has no effect on clock generation in ISO7816 smart card mode (the SMART bit is set).<br>0h = 16x oversampling.<br>1h = 8x oversampling.<br>2h = 3x oversampling. IrDA, Manchester and DALI not supported when 3x oversampling is enabled. |
| 14 | CTSEN | R/W | 0h | Enable Clear To Send<br>0h = CTS hardware flow control is disabled.<br>1h = CTS hardware flow control is enabled. Data is only transmitted when the UARTxCTS signal is asserted. |
| 13 | RTSEN | R/W | 0h | Enable hardware controlled Request to Send<br>0h = RTS hardware flow control is disabled.<br>1h = RTS hardware flow control is enabled. Data is only requested (by asserting UARTxRTS) when the receive FIFO has available entries. |
| 12 | RTS | R/W | 0h | Request to Send<br>If RTSEN is set the RTS output signals is controlled by the hardware logic using the FIFO fill level or TXDATA buffer.<br>If RTSEN is cleared the RTS output is controlled by the RTS bit. The bit is the complement of the UART request to send, RTS modem status output.<br>0h = Signal not RTS<br>1h = Signal RTS |
| 11 | RESERVED | R/W | 0h | |
| 10-8 | MODE | R/W | 0h | Set the communication mode and protocol used.<br>(Not defined settings uses the default setting: 0)<br>0h = Normal operation<br>1h = RS485 mode: UART needs to be IDLE with receiving data for the in EXTDIR_HOLD set time. EXTDIR_SETUP defines the time the RTS line is set to high before sending. When the buffer is empty the RTS line is set low again. A transmit will be delayed as long the UART is receiving data.<br>2h = The UART operates in IDLE Line Mode<br>3h = The UART operates in 9 Bit Address mode<br>4h = ISO7816 Smart Card Support The application must ensure that it sets 8-bit word length (WLEN set to 3h) and even parity (PEN set to 1, EPS set to 1, SPS set to 0) in UARTLCRH when using ISO7816 mode. The value of the STP2 bit in UARTLCRH is ignored and the number of stop bits is forced to 2.<br>5h = DALI Mode: |
| 7 | MENC | R/W | 0h | Manchester Encode enable<br>0h = Disable Manchester Encoding<br>1h = Enable Manchester Encoding |
| 6 | TXD_OUT | R/W | 0h | TXD Pin Control Controls the TXD pin when TXD_OUT_EN = 1 and TXE = 0.<br>0h = TXD pin is low<br>1h = TXD pin is high |
| 5 | TXD_OUT_EN | R/W | 1h | TXD Pin Control Enable. When the transmit section of the UART is disabled (TXE = 0), the TXD pin can be controlled by the TXD_OUT bit.<br>0h = TXD pin can not be controlled by TXD_OUT<br>1h = TXD pin can be controlled by TXD_OUT |



Copyright © 2026 Texas Instruments Incorporated

**Table 24-41. CTL0 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 4 | TXE | R/W | 1h | UART Transmit Enable If the UART is disabled in the middle of a transmission, it completes the current character before stopping.<br>**NOTE:** To enable transmission, the UARTEN bit must be set.<br>0h = The transmit section of the UART is disabled. The UARTxTXD pin of the UART can be controlled by the TXD_CTL bit when enabled.<br>1h = The transmit section of the UART is enabled. |
| 3 | RXE | R/W | 1h | UART Receive Enable If the UART is disabled in the middle of a receive, it completes the current character before stopping. **NOTE:** To enable reception, the UARTEN bit must be set.<br>0h = The receive section of the UART is disabled.<br>1h = The receive section of the UART is enabled. |
| 2 | LBE | R/W | 0h | UART Loop Back Enable<br>0h = Normal operation.<br>1h = The UARTxTX path is fed through the UARTxRX path internally. |
| 1 | RESERVED | R/W | 0h | |
| 0 | ENABLE | R/W | 0h | UART Module Enable. If the UART is disabled in the middle of transmission or reception, it completes the current character before stopping.<br>If the ENABLE bit is not set, all registers can still be accessed and updated. It is recommended to setup and change the UART operation mode with having the ENABLE bit cleared to avoid unpredictable behavior during the setup or update.<br>If disabled the UART module will not send or receive any data and the logic is held in reset state.<br>0h = Disable Module<br>1h = Enable module |



Copyright © 2026 Texas Instruments Incorporated

### 24.3.29 LCRH (Offset = 1104h) [Reset = 00000000h]

LCRH is shown in Figure 24-44 and described in Table 24-42.

Return to the Summary Table.

UART Line Control Register The LCRH register is the line control register. Serial parameters such as data length, parity, and stop bit selection are implemented in this register. When updating the baud-rate divisor (UARTIBRD or UARTIFRD), the LCRH register must also be written. The write strobe for the baud-rate divisor registers is tied to the LCRH register.

**Figure 24-44. LCRH**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | EXTDIR_HOLD | |
| R/W-0h | | | | | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| EXTDIR_HOLD | | | EXTDIR_SETUP | | | | |
| R/W-0h | | | R/W-0h | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SENDIDLE | SPS | WLEN | | STP2 | EPS | PEN | BRK |
| R/W-0h | R/W-0h | R/W-0h | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

**Table 24-42. LCRH Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | RESERVED | R/W | 0h | |
| 25-21 | EXTDIR_HOLD | R/W | 0h | Defines the number of UARTclk ticks the signal to control the external driver for the RS485 will be reset after the beginning of the stop bit. (If 2 STOP bits are enabled the beginning of the 2nd STOP bit.)<br>0h = Smallest value<br>1Fh = Highest possible value |
| 20-16 | EXTDIR_SETUP | R/W | 0h | Defines the number of UARTclk ticks the signal to control the external driver for the RS485 will be set before the START bit is send<br>0h = Smallest value<br>1Fh = Highest possible value |
| 15-8 | RESERVED | R/W | 0h | |
| 7 | SENDIDLE | R/W | 0h | UART send IDLE pattern. When this bit is set an SENDIDLE period of 11 bit times will be sent on the TX line. The bit is cleared by hardware afterward.<br>0h = Disable Send Idle Pattern<br>1h = Enable Send Idle Pattern |
| 6 | SPS | R/W | 0h | UART Stick Parity Select<br>The Stick Parity Select (SPS) bit is used to set either a permanent '1' or a permanent '0' as parity when transmitting or receiving data. Its purpose is to typically indicate the first byte of a package or to mark an address byte, for example in a multi-drop RS-485 network.<br>When bits PEN, EPS, and SPS of UARTLCRH are set, the parity bit is transmitted and checked as a 0.<br>When bits PEN and SPS are set and EPS is cleared, the parity bit is transmitted and checked as a 1.<br>0h = Disable Stick Parity<br>1h = Enable Stick Parity |



Copyright © 2026 Texas Instruments Incorporated

**Table 24-42. LCRH Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 5-4 | WLEN | R/W | 0h | UART Word Length The bits indicate the number of data bits transmitted or received in a frame as follows:<br>0h = 5 bits (default)<br>1h = 6 bits<br>2h = 7 bits<br>3h = 8 bits |
| 3 | STP2 | R/W | 0h | UART Two Stop Bits Select When in 7816 smart card mode (the SMART bit is set in the UARTCTL register), the number of stop bits is forced to 2.<br>0h = One stop bit is transmitted at the end of a frame.<br>1h = Two stop bits are transmitted at the end of a frame. The receive logic checks for two stop bits being received and provide Frame Error if either is invalid. |
| 2 | EPS | R/W | 0h | UART Even Parity Select This bit has no effect when parity is disabled by the PEN bit. For 9-Bit UART Mode transmissions, this bit controls the address byte and data byte indication (9th bit). 0 = The transferred byte is a data byte 1 = The transferred byte is an address byte<br>0h = Odd parity is performed, which checks for an odd number of 1s.<br>1h = Even parity generation and checking is performed during transmission and reception, which checks for an even number of 1s in data and parity bits. |
| 1 | PEN | R/W | 0h | UART Parity Enable<br>0h = Parity is disabled and no parity bit is added to the data frame.<br>1h = Parity checking and generation is enabled. |
| 0 | BRK | R/W | 0h | UART Send Break (for LIN Protocol)<br>0h = Normal use.<br>1h = A low level is continually output on the UARTxTXD signal, after completing transmission of the current character. For the proper execution of the break command, software must set this bit for at least two frames (character periods). |



Copyright © 2026 Texas Instruments Incorporated

### 24.3.30 STAT (Offset = 1108h) [Reset = 00000144h]

STAT is shown in Figure 24-45 and described in Table 24-43.

Return to the Summary Table.

UART Status Register

**Figure 24-45. STAT**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R- | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R- | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | IDLE | CTS |
| R- | | | | | | R-0h | R-1h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TXFF | TXFE | RESERVED | | RXFF | RXFE | RESERVED | BUSY |
| R-0h | R-1h | R- | | R-0h | R-1h | R- | R-0h |

**Table 24-43. STAT Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-10 | RESERVED | R | 0h | |
| 9 | IDLE | R | 0h | IDLE mode has been detected in Idleline-Multiprocessor-Mode.<br>The IDLE bit is used as an address tag for each block of characters.<br>In idle-line multiprocessor format, this bit is set when a received character is an address.<br>0h = IDLE has not been detected before last received character. (In idle-line multiprocessor mode).<br>1h = IDLE has been detected before last received character. (In idle-line multiprocessor mode). |
| 8 | CTS | R | 1h | Clear To Send<br>0h = The CTS signal is not asserted (high).<br>1h = The CTS signal is asserted (low). |
| 7 | TXFF | R | 0h | UART Transmit FIFO Full The meaning of this bit depends on the state of the FEN bit in the CTL0 register.<br>0h = The transmitter is not full.<br>1h = If the FIFO is disabled (FEN is 0), the transmit holding register is full. If the FIFO is enabled (FEN is 1), the transmit FIFO is full. |
| 6 | TXFE | R | 1h | UART Transmit FIFO Empty The meaning of this bit depends on the state of the FEN bit in the CTL0 register.<br>0h = The transmitter has data to transmit.<br>1h = If the FIFO is disabled (FEN is 0), the transmit holding register is empty. If the FIFO is enabled (FEN is 1), the transmit FIFO is empty. |
| 5-4 | RESERVED | R | 0h | |
| 3 | RXFF | R | 0h | UART Receive FIFO Full The meaning of this bit depends on the state of the FEN bit in the CTL0 register.<br>0h = The receiver can receive data.<br>1h = If the FIFO is disabled (FEN is 0), the receive holding register is full. If the FIFO is enabled (FEN is 1), the receive FIFO is full. |



Copyright © 2026 Texas Instruments Incorporated

**Table 24-43. STAT Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | RXFE | R | 1h | UART Receive FIFO Empty The meaning of this bit depends on the state of the FEN bit in the CTL0 register.<br>0h = The receiver is not empty.<br>1h = If the FIFO is disabled (FEN is 0), the receive holding register is empty. If the FIFO is enabled (FEN is 1), the receive FIFO is empty. |
| 1 | RESERVED | R | 0h | |
| 0 | BUSY | R | 0h | UART Busy<br>This bit is set as soon as the transmit FIFO or TXDATA register becomes nonempty (regardless of whether UART is enabled) or if a receive data is currently ongoing (after the start edge have been detected until a complete byte, including all stop bits, has been received by the shift register).<br>In IDLE_Line mode the Busy signal also stays set during the idle time generation.<br>0h = The UART is not busy.<br>1h = The UART is busy transmitting data. This bit remains set until the complete byte, including all stop bits, has been sent/received from/into the shift register. |



Copyright © 2026 Texas Instruments Incorporated

### 24.3.31 IFLS (Offset = 110Ch) [Reset = 00000022h]

IFLS is shown in Figure 24-46 and described in Table 24-44.

Return to the Summary Table.

The IFLS register is the interrupt FIFO level select register. You can use this register to define the levels at which the TX, RX and timeout interrupt flags are triggered. The interrupts are generated based on a transition through a level rather than being based on the level. That is, the interrupts are generated when the fill level progresses through the trigger level. For example, if the receive trigger level is set to the half-way mark, the interrupt is triggered when the receive FIFO is filled with two or more characters. Out of reset, the TXIFLSEL and RXIFLSEL bits are configured so that the FIFOs trigger an interrupt at the half-way mark.

**Figure 24-46. IFLS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | RXTOSEL | | | |
| R/W-0h | | | | R/W-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | RXIFLSEL | | | RESERVED | TXIFLSEL | | |
| R/W-0h | R/W-2h | | | R/W-0h | R/W-2h | | |

**Table 24-44. IFLS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-12 | RESERVED | R/W | 0h | |
| 11-8 | RXTOSEL | R/W | 0h | UART Receive Interrupt Timeout Select. When receiving no start edge for an additional character within the set bit times a RX interrupt is set even if the FIFO level is not reached. A value of 0 disables this function.<br>0h = Smallest value<br>Fh = Highest possible value |
| 7 | RESERVED | R/W | 0h | |
| 6-4 | RXIFLSEL | R/W | 2h | UART Receive Interrupt FIFO Level Select The trigger points for the receive interrupt are as follows:<br>Note:<br>In ULP domain the trigger levels are used for:<br>0: LVL_1_4<br>4: LVL_FULL<br>For undefined settings the default configuration is used.<br>0h = RX FIFO >= 1/4 full<br>Note: For ULP Domain<br>1h = RX FIFO >= 1/4 full<br>2h = RX FIFO >= 1/2 full (default)<br>3h = RX FIFO >= 3/4 full<br>4h = RX FIFO is full<br>Note: For ULP Domain<br>5h = RX FIFO is full<br>7h = RX FIFO >= 1 entry available<br>Note: esp. required for DMA Trigger |
| 3 | RESERVED | R/W | 0h | |



Copyright © 2026 Texas Instruments Incorporated

**Table 24-44. IFLS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2-0 | TXIFLSEL | R/W | 2h | UART Transmit Interrupt FIFO Level Select The trigger points for the transmit interrupt are as follows:<br>Note: for undefined settings the default configuration is used.<br>1h = TX FIFO <= 3/4 empty<br>2h = TX FIFO <= 1/2 empty (default)<br>3h = TX FIFO <= 1/4 empty<br>5h = TX FIFO is empty<br>7h = TX FIFO >= 1 entry free<br>Note: esp. required for DMA Trigger |



Copyright © 2026 Texas Instruments Incorporated

### 24.3.32 IBRD (Offset = 1110h) [Reset = 00000000h]

IBRD is shown in Figure 24-47 and described in Table 24-45.

Return to the Summary Table.

When changing the IBRD register, the new value does not take effect until transmission/reception of the current character is complete. Any changes to the baud-rate divisor must be followed by a write to the UARTLCRH register. See Baud-Rate Generation chapter for configuration details.

**Figure 24-47. IBRD**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | DIVINT | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | R/W-0h | | | | | | | | | | | | | | | |

**Table 24-45. IBRD Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R/W | 0h | |
| 15-0 | DIVINT | R/W | 0h | Integer Baud-Rate Divisor<br>0h = Smallest value<br>FFFFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 24.3.33 FBRD (Offset = 1114h) [Reset = 00000000h]

FBRD is shown in Figure 24-48 and described in Table 24-46.

Return to the Summary Table.

UART Fractional Baud-Rate Divisor Register The FBRD register is the fractional part of the baud-rate divisor value. All the bits are cleared on reset. When changing the FBRD register, the new value does not take effect until transmission/reception of the current character is complete. Any changes to the baud-rate divisor must be followed by a write to the UARTLCRH register. See Baud-Rate Generation chapter for configuration details.

**Figure 24-48. FBRD**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | DIVFRAC | | | | | |
| R/W-0h | | | | | | | | | | R/W-0h | | | | | |

**Table 24-46. FBRD Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | RESERVED | R/W | 0h | |
| 5-0 | DIVFRAC | R/W | 0h | Fractional Baud-Rate Divisor<br>0h = Smallest value<br>3Fh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 24.3.34 GFCTL (Offset = 1118h) [Reset = X]

GFCTL is shown in Figure 24-49 and described in Table 24-47.

Return to the Summary Table.

This register control the glitch filter on the RX input.

**Figure 24-49. GFCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | CHAIN | AGFSEL | | AGFEN |
| R/W-0h | | | | R/W-0h | R/W-0h | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | DGFSEL | | | | | |
| R/W-0h | | R/W-0h | | | | | |

**Table 24-47. GFCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-12 | RESERVED | R/W | 0h | |
| 11 | CHAIN | R/W | 0h | Analog and digital noise filters chaining enable.<br>0 DISABLE: When 0, chaining is disabled and only digital filter output is available to IP logic for sampling<br>1 ENABLE: When 1, analog and digital glitch filters are chained and the output of the combination is made available to IP logic for sampling<br>0h = Disabled<br>1h = Enabled |
| 10-9 | AGFSEL | R/W | 0h | Analog Glitch Suppression Pulse Width<br>This field controls the pulse width select for the analog glitch suppression on the RX line.<br>See device data sheet for exact values.<br>0h = Pulses shorter then 5ns length are filtered.<br>1h = Pulses shorter then 10ns length are filtered.<br>2h = Pulses shorter then 25ns length are filtered.<br>3h = Pulses shorter then 50ns length are filtered. |
| 8 | AGFEN | R/W | 0h | Analog Glitch Suppression Enable<br>0h = Analog Glitch Filter disable<br>1h = Analog Glitch Filter enable |
| 7-6 | RESERVED | R/W | 0h | |
| 5-0 | DGFSEL | R/W | 0h | Glitch Suppression Pulse Width<br>This field controls the pulse width select for glitch suppression on the RX line.<br>The value programmed in this field gives the number of cycles of functional clock up to which the glitch has to be suppressed on the RX line.<br>In IRDA mode:<br>The minimum pulse length for receive is given by:<br>t(MIN) = (DGFSEL) / f(IRTXCLK)<br>0h = Bypass GF<br>3Fh = Highest Possible Value |



Copyright © 2026 Texas Instruments Incorporated

### 24.3.35 TXDATA (Offset = 1120h) [Reset = 00000000h]

TXDATA is shown in Figure 24-50 and described in Table 24-48.

Return to the Summary Table.

UART Transmit Data Register. This register is the transmit data register (the interface to the FIFOs). For transmitted data, if the FIFO is enabled, data written to this location is pushed onto the transmit FIFO. If the FIFO is disabled, data is stored in the transmitter holding register (the bottom word of the transmit FIFO). A write to this register initiates a transmission from the UART.

**Figure 24-50. TXDATA**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | DATA | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | | | | | | | | | R/W-0h | | | | | | | |

**Table 24-48. TXDATA Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R/W | 0h | |
| 7-0 | DATA | R/W | 0h | Data Transmitted or Received Data that is to be transmitted via the UART is written to this field. When read, this field contains the data that was received by the UART.<br>0h = Smallest value<br>FFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 24.3.36 RXDATA (Offset = 1124h) [Reset = 00000000h]

RXDATA is shown in Figure 24-51 and described in Table 24-49.

Return to the Summary Table.

UART Receive Data Register. This register is the data receive register (the interface to the FIFOs). For received data, if the FIFO is enabled, the data byte and the 4-bit status (break, frame, parity, and overrun) is pushed onto the 12-bit wide receive FIFO. If the FIFO is disabled, the data byte and status are stored in the receiving holding register (the bottom word of the receive FIFO). The received data can be retrieved by reading this register.

**Figure 24-51. RXDATA**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | NERR | OVRERR | BRKERR | PARERR | FRMERR |
| R-0h | | | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DATA | | | | | | | |
| R-0h | | | | | | | |

**Table 24-49. RXDATA Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-13 | RESERVED | R | 0h | |
| 12 | NERR | R | 0h | Noise Error.<br>Writing to this bit has no effect. The flag is cleared by writing 1 to the NERR bit in the UART EVENT ICLR register.<br>0h = No noise error occurred<br>1h = Noise error occurred during majority voting |
| 11 | OVRERR | R | 0h | UART Receive Overrun Error Writing to this bit has no effect. The flag is cleared by writing 1 to the OVRERR bit in the UART EVENT ICLR register. In case of a receive FIFO overflow, the FIFO contents remain valid because no further data is written when the FIFO is full. Only the contents of the shift register are overwritten. The CPU must read the data to empty the FIFO.<br>0h = No data has been lost due to a receive overrun.<br>1h = New data was received but could not be stored, because the previous data was not read (resulting in data loss). |
| 10 | BRKERR | R | 0h | UART Break Error Writing to this bit has no effect. The flag is cleared by writing 1 to the BRKERR bit in the UART EVENT ICLR register. This error is associated with the character at the top of the FIFO. When a break occurs, only one 0 character is loaded into the FIFO. The next character is only enabled after the receive data input goes to a 1 (marking state) and the next valid start bit is received.<br>0h = No break condition has occurred<br>1h = A break condition has been detected, indicating that the receive data input was held low for longer than a full-word transmission time (defined as start, data, parity, and stop bits). |
| 9 | PARERR | R | 0h | UART Parity Error Writing to this bit has no effect. The flag is cleared by writing 1 to the PARERR bit in the UART EVENT ICLR register.<br>0h = No parity error has occurred<br>1h = The parity of the received data character does not match the parity defined by bits 2 and 7 of the UARTLCRH register. |



Copyright © 2026 Texas Instruments Incorporated

**Table 24-49. RXDATA Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 8 | FRMERR | R | 0h | UART Framing Error Writing to this bit has no effect. The flag is cleared by writing 1 to the FRMERR bit in the UART EVENT ICLR register. This error is associated with the character at the top of the FIFO.<br>0h = No framing error has occurred<br>1h = The received character does not have a valid stop bit sequence, which is one or two stop bits depending on the UARTLCRH.STP2 setting (a valid stop bit is 1). |
| 7-0 | DATA | R | 0h | Received Data. When read, this field contains the data that was received by the UART.<br>0h = Smallest value<br>FFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 24.3.37 LINCNT (Offset = 1130h) [Reset = 00000000h]

LINCNT is shown in Figure 24-52 and described in Table 24-50.

Return to the Summary Table.

UART LIN Mode Counter Register

**Figure 24-52. LINCNT**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | VALUE | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | R/W-0h | | | | | | | | | | | | | | | |

**Table 24-50. LINCNT Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R/W | 0h | |
| 15-0 | VALUE | R/W | 0h | 16 bit up counter clocked by the functional clock of the UART.<br>0h = Smallest value<br>FFFFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 24.3.38 LINCTL (Offset = 1134h) [Reset = 00000000h]

LINCTL is shown in Figure 24-53 and described in Table 24-51.

Return to the Summary Table.

UART LIN Mode Control Register

**Figure 24-53. LINCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | LINC0_MATCH | LINC1CAP | LINC0CAP | RESERVED | CNTRXLOW | ZERONE | CTRENA |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

**Table 24-51. LINCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-7 | RESERVED | R/W | 0h | |
| 6 | LINC0_MATCH | R/W | 0h | Counter Compare Match Mode When this bit is set to 1 a counter compare match with LINC0 register triggers an LINC0 interrupt when enabled.<br>0h = Counter compare match mode disabled (capture mode enabled)<br>1h = Counter compare match enabled (capture mode disabled) |
| 5 | LINC1CAP | R/W | 0h | Capture Counter on positive RXD Edge. When enabled the counter value is captured to LINC1 register on each positive RXD edge. A LINC1 interrupt is triggered when enabled.<br>0h = Capture counter on positive UARTxRXD edge disabled<br>1h = Capture counter on positive UARTxRXD edge enabled |
| 4 | LINC0CAP | R/W | 0h | Capture Counter on negative RXD Edge. When enabled the counter value is captured to LINC0 register on each negative RXD edge. A LINC0 interrupt is triggered when enabled.<br>0h = Capture counter on negative UARTxRXD edge disabled<br>1h = Capture counter on negative UARTxRXD edge enabled |
| 3 | RESERVED | R/W | 0h | |
| 2 | CNTRXLOW | R/W | 0h | Count while low Signal on RXD When counter is enabled and the signal on RXD is low, the counter increments.<br>0h = Count while low Signal on UARTxRXD disabled<br>1h = Count while low Signal on UARTxRXD enabled |
| 1 | ZERONE | R/W | 0h | Zero on negative Edge of RXD. When enabled the counter is set to 0 and starts counting on a negative edge of RXD<br>0h = Zero on negative edge disabled<br>1h = Zero on negative edge enabled |
| 0 | CTRENA | R/W | 0h | LIN Counter Enable. LIN counter will only count when enabled.<br>0h = Counter disabled<br>1h = Counter enabled |



Copyright © 2026 Texas Instruments Incorporated

### 24.3.39 LINC0 (Offset = 1138h) [Reset = 00000000h]

LINC0 is shown in Figure 24-54 and described in Table 24-52.

Return to the Summary Table.

UART LIN Mode Capture 0 Register

**Figure 24-54. LINC0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | DATA | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | R/W-0h | | | | | | | | | | | | | | | |

**Table 24-52. LINC0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R/W | 0h | |
| 15-0 | DATA | R/W | 0h | 16 Bit Capture / Compare Register<br>Captures current LINCTR value on RXD falling edge and can generate a LINC0 interrupt when capture is enabled (LINC0CAP = 1).<br>If compare mode is enabled (LINC0_MATCH = 1), a counter match can generate a LINC0 interrupt.<br>0h = Smallest value<br>FFFFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 24.3.40 LINC1 (Offset = 113Ch) [Reset = 00000000h]

LINC1 is shown in Figure 24-55 and described in Table 24-53.

Return to the Summary Table.

UART LIN Mode Capture 1 Register

**Figure 24-55. LINC1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | DATA | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | R/W-0h | | | | | | | | | | | | | | | |

**Table 24-53. LINC1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R/W | 0h | |
| 15-0 | DATA | R/W | 0h | 16 Bit Capture / Compare Register<br>Captures current LINCTR value on RXD rising edge and can generate a LINC1 interrupt when capture is enabled (LINC1CAP = 1)<br>0h = Smallest value<br>FFFFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 24.3.41 IRCTL (Offset = 1140h) [Reset = X]

IRCTL is shown in Figure 24-56 and described in Table 24-54.

Return to the Summary Table.

IrDA Control Register

**Figure 24-56. IRCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | IRRXPL | RESERVED |
| R/W-0h | | | | | | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| IRTXPL | | | | | | IRTXCLK | IREN |
| R/W-0h | | | | | | R/W-0h | R/W-0h |

**Table 24-54. IRCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-10 | RESERVED | R/W | 0h | |
| 9 | IRRXPL | R/W | 0h | IrDA receive input UCAxRXD polarity<br>0h = HIGH : IrDA transceiver delivers a high pulse when a light pulse is seen<br>1h = LOW : IrDA transceiver delivers a low pulse when a light pulse is seen |
| 8 | RESERVED | R/W | 0h | |
| 7-2 | IRTXPL | R/W | 0h | Transmit pulse length.<br>Pulse length t(PULSE) = (IRTXPLx + 1) / [2 * f(IRTXCLK)]<br>(IRTXCLK = functional clock of the UART)<br>0h = Smallest value<br>3Fh = Highest possible value |
| 1 | IRTXCLK | R/W | 0h | IrDA transmit pulse clock select<br>0h (R/W) = IrDA encode data is based on the functional clock.<br>1h (R/W) = IrDA encode data is based on the Baud Rate clock<br>when select 8x oversampling, the IRTXPL cycle should less 8;<br>when select 16x oversampling, the IRTXPL cycle should less 16. |
| 0 | IREN | R/W | 0h | IrDA encoder/decoder enable<br>0h (R/W) = IrDA encoder/decoder disabled<br>1h (R/W) = IrDA encoder/decoder enabled |



Copyright © 2026 Texas Instruments Incorporated

### 24.3.42 AMASK (Offset = 1148h) [Reset = 000000FFh]

AMASK is shown in Figure 24-57 and described in Table 24-55.

Return to the Summary Table.

Self Address Mask Register The AMASK register is used to enable the address mask for 9-bit or Idle-Line mode. The address bits are masked to create a set of addresses to be matched with the received address byte. Used in DALI, UART 9-Bit or Idle-Line mode.

**Figure 24-57. AMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | VALUE | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | | | | | | | | | R/W-FFh | | | | | | | |

**Table 24-55. AMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R/W | 0h | |
| 7-0 | VALUE | R/W | FFh | Self Address Mask for 9-Bit Mode This field contains the address mask that creates a set of addresses that should be matched. A 0 bit in the MSK bit field configures, that the corresponding bit in the ADDR bit field of the UARTxADDR register is don't care. A 1 bit in the MSK bit field configures, that the corresponding bit in the ADDR bit field of the UARTxADDR register must match.<br>0h = Smallest value<br>FFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 24.3.43 ADDR (Offset = 114Ch) [Reset = 00000000h]

ADDR is shown in Figure 24-58 and described in Table 24-56.

Return to the Summary Table.

Self Address Register The ADDR register is used to write the specific address that should be matched with the receiving byte when the Address Mask (AMASK) is set to FFh. This register is used in conjunction with AMASK to form a match for address-byte received.
Used in DALI, UART 9-Bit or Idle-Line mode.

**Figure 24-58. ADDR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | VALUE | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | | | | | | | | | R/W-0h | | | | | | | |

**Table 24-56. ADDR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R/W | 0h | |
| 7-0 | VALUE | R/W | 0h | Self Address for 9-Bit Mode This field contains the address that should be matched when UARTxAMASK is FFh.<br>0h = Smallest value<br>FFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 24.3.44 CLKDIV2 (Offset = 1160h) [Reset = 00000000h]

CLKDIV2 is shown in Figure 24-59 and described in Table 24-57.

Return to the Summary Table.

This register is used to specify module-specific divide ratio of the functional clock (only in UART extended).

**Figure 24-59. CLKDIV2**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | | RATIO | | |
| R/W-0h | | | | | | | | | | | | | R/W-0h | | |

**Table 24-57. CLKDIV2 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | RESERVED | R/W | 0h | |
| 2-0 | RATIO | R/W | 0h | Selects divide ratio of module clock<br>0h = Do not divide clock source<br>1h = Divide clock source by 2<br>2h = Divide clock source by 3<br>3h = Divide clock source by 4<br>4h = Divide clock source by 5<br>5h = Divide clock source by 6<br>6h = Divide clock source by 7<br>7h = Divide clock source by 8 |



Copyright © 2026 Texas Instruments Incorporated

Chapter 25

# I²C

The I²C module provides a standardized serial interface to transfer data between MSP devices and other external I²C devices (such as a sensors, memory, or DACs).


**25.1 I²C Overview**.....1323
**25.2 I²C Operation**.....1325
**25.3 I2C Registers**.....1350





Copyright © 2026 Texas Instruments Incorporated

## 25.1 I²C Overview

The I²C module provides a standardized serial interface to transfer data between MSP devices and other external I²C devices (such as a sensors, memory, or DACs).

### *25.1.1 Purpose of the Peripheral*

The I²C peripheral provides bidirectional data transfer through a two-wire serial bus consisting of a data (SDA) and clock (SCL) line. The I²C bus is widely used to interface with devices such as battery management ICs, sensors, other MCUs and so on. This I²C peripheral provides the ability to both transmit to and receive from other I²C devices on the bus.

### *25.1.2 Features*

The controller includes I²C modules with the following features:

- Devices on the I²C bus can be designated as either a controller or a target with 7-bit addressing.
- Support four I²C modes
  - Controller transmit
  - Controller receive
  - Target transmit
  - Target receive
- Supported transmission speeds:
  - Standard-mode (Sm) with a bit rate up to 100 kbps
  - Fast-mode (Fm) with a bit rate up to 400 kbps
  - Fast-mode Plus (Fm+) with a bit rate up to 1 Mbps
- Independent 8-byte FIFOs for reception and transmission
- Dual target address capability
- Glitch suppression
- Independent controller and target interrupt generation
- Controller operation with arbitration, clock synchronization, multiple controller support
- Hardware support for SMBus and PMBus
  - Clock low timeout detection and interrupt
  - Quick command capability
- Hardware support for DMA with separate channels for transmit and receive



Copyright © 2026 Texas Instruments Incorporated

### 25.1.3 Functional Block Diagram

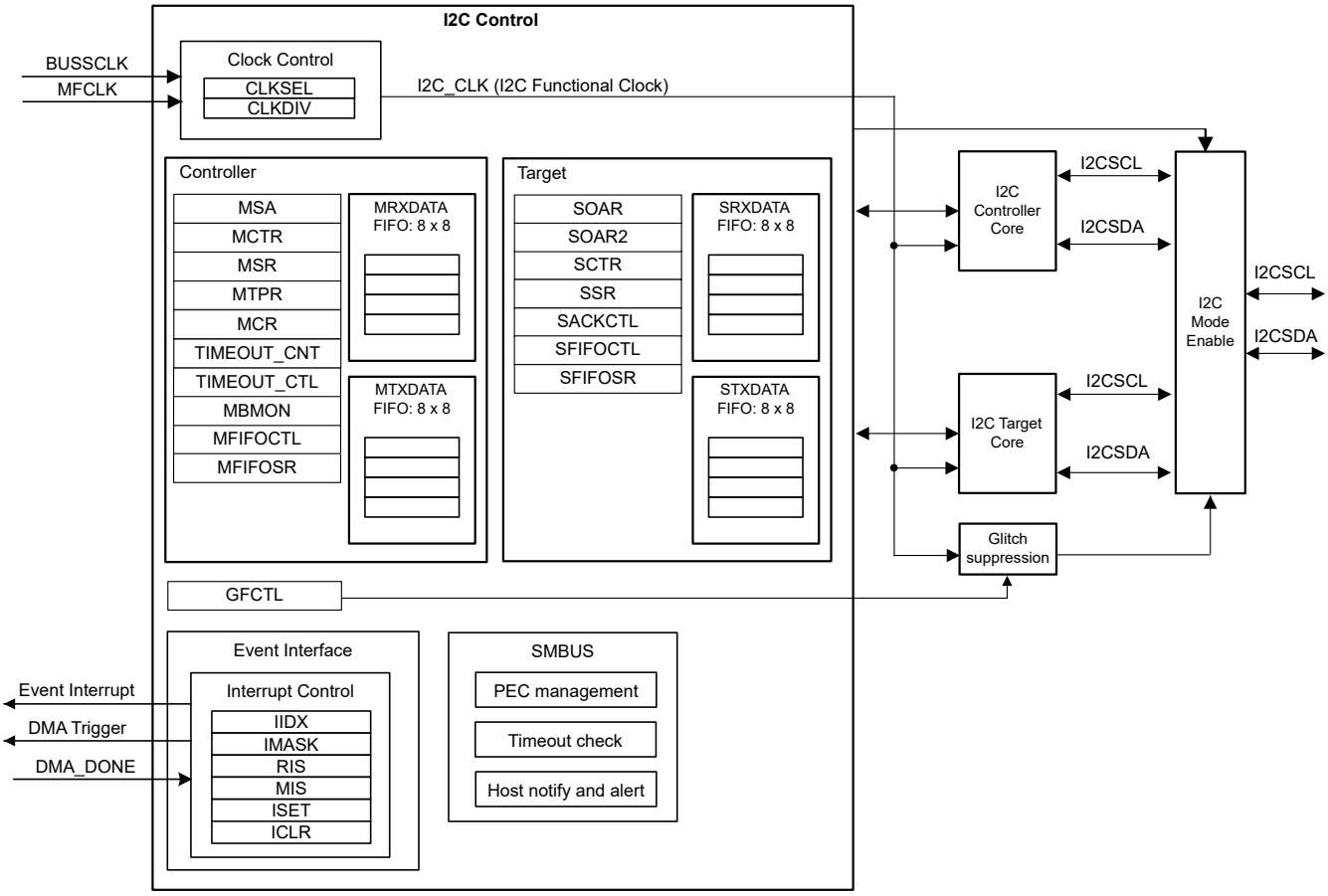



**Figure 25-1. I2C Functional Block Diagram**

### 25.1.4 Environment and External Connections

The I²C mode supports any target or controller I²C-compatible device. Figure 25-2 shows an example of an I²C bus. Each I²C peripheral instance is comprised of both controller and target functions where the target can be addressed with 2 independent user defined addresses. A device connected to the I²C bus can be considered as the controller or the target when performing data transfers. A controller initiates a data transfer and generates the clock signal SCL. Any device addressed by a controller is considered a target.

I²C data is communicated using the serial data (SDA) pin and the serial clock (SCL) pin. Both SDA and SCL are open-drain bidirectional and must be connected to a positive supply voltage using a pullup resistor.

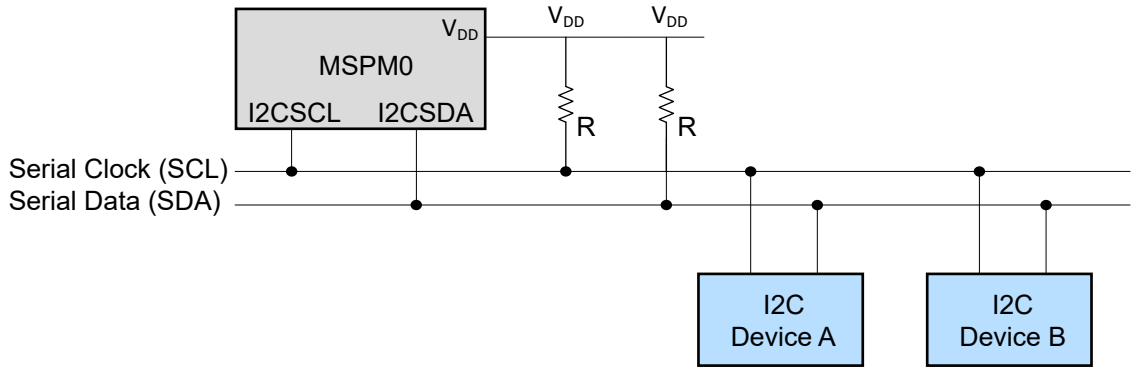



**Figure 25-2. Bus Connection Diagram**



Copyright © 2026 Texas Instruments Incorporated

## 25.2 I²C Operation

This section describes the operation of the I²C peripheral.

### *25.2.1 Clock Control*

#### 25.2.1.1 Clock Select and I²C Speed

Standard, Fast, and Fast Plus modes are selected using a value in the I²C Controller Timer Period (I2CMTPR) register that results in a maximum SCL frequency of:

- 100kbps for Standard mode
- 400kbps for Fast mode
- 1Mbps for Fast mode Plus

The I²C frequency I2C_FREQ is determined by the I2C_CLK frequency and bit fields TPR , SCL_LP , and SCL_HP where:

- I2C_CLK is the functional clock frequency to the I²C module. Note that the I²C internal functional clock is first divided from the source clock:
  - Use I2Cx.CLKSEL register to select the source of the I²C functional clock
    - BUSSCLK: the current bus clock is selected as the source for I²C. The current bus clock depends on power domain. If the I²C instance is in power domain 1 (PD1) refer to MCLK, if the I²C instance is in power domain 0 (PD0) refer to ULPCLK.
    - MFCLK: MFCLK is selected as the source for I²C, refer to MFCLK.
  - Use I2Cx.CLKDIV register to select divide ratio of I²C function clock, options are from divide by 1 to 8.
- SCL_LP is the low phase of SCL (which must be fixed at 6)
- SCL_HP is the high phase of SCL (which must be fixed at 4)
- TPR is the programmed value of the TPR bits in the I2Cx.MTPR register. This value is determined by replacing the known variables in the equation below and solving for TPR

The I²C frequency is calculated as follows:

$$I2C\_FREQ = I2C\_CLK / ((1+TPR) \times (SCL\_LP + SCL\_HP)) \quad (27)$$

For example, if the I²C functional clock frequency is 32MHz and target SCL frequency is 400kHz:

I2C_CLK = 32MHz

I2C_FREQ = 400kHz

SCL_LP = 6 , SCL_HP = 4

TPR = (I2C_CLK / (I2C_FREQ * (4 + 6))) - 1

TPR = 7 (0x07)

**Table 25-1. Examples of Controller Clock Setting for Typical Clock Configurations**

| Functional Clock | TPR Bits Standard Mode 100kHz SCL | TPR Bits Fast Mode 400kHz SCL | TPR Bits Fast Mode Plus 1000kHz SCL |
|---|---|---|---|
| 4MHz | 0x03 | - | - |
| 8MHz | 0x07 | 0x01 | - |
| 20MHz | 0x13 | 0x04 | 0x01 |
| 32MHz | 0x1F | 0x07 | 0x02(1) |
| 40MHz | 0x27 | 0x09 | 0x03 |

(1) With 32MHz functional clock, TPR = 0x01 generates 1.6MHz SCL frequency, and TPR = 0x02 generates 1.067MHz SCL frequency.

I²C functional clock must be greater than or equal to 20 times the SCL frequency, I2C_CLK ≥ 20 × I2C_FREQ.



Copyright © 2026 Texas Instruments Incorporated

The following minimum functional clock frequencies are required when running certain I2C clock speeds:

- I2C_CLK ≥ 2MHz when working with I²C speed 0 to 100kHz
- I2C_CLK ≥ 8MHz when working with I²C speed 100 to 400kHz
- I2C_CLK ≥ 20MHz when working with I²C speed 400 kHz to 1MHz

#### 25.2.1.2 Clock Startup

The selected clock source is always available and the frequency depends on the power mode, for more information refer to the PMU/Clock section. After enabling the I²C module by setting the I2C.PWREN.ENABLE bit, the module is ready to start receiving and transmitting data.

### *25.2.2 Signal Descriptions*

The I²C bus consists of a clock signal and a data signal. The clock signal can be generated either internally (during controller operation) or externally (during target operation).

**Table 25-2. I²C signal descriptions**

| Device Pin | Function |
|---|---|
| I2Cx_SCL | I²C clock signal |
| I2Cx_SDA | I²C data signal |

### *25.2.3 General Architecture*

#### 25.2.3.1 I²C Bus Functional Overview

The I²C bus uses only two signals: SDA and SCL. SDA is the bidirectional serial data line and SCL is the bidirectional serial clock line. The bus is considered idle when both lines are in high state and no transfer is ongoing.

Every transaction on the I²C bus is 9-bits long, consisting of 8 data bits and 1 acknowledge bit. A transfer is defined as the time between a valid START and STOP condition—as described in Figure 25-3. The number of bytes per transfer is unrestricted; however, each data byte must be followed by an acknowledge bit and data must be transferred MSB first. When a receiver cannot receive another complete byte, it can hold the clock line SCL Low and force the transmitter into a wait state, this process is commonly known as clock stretching. The data transfer continues when the receiver releases the clock SCL.

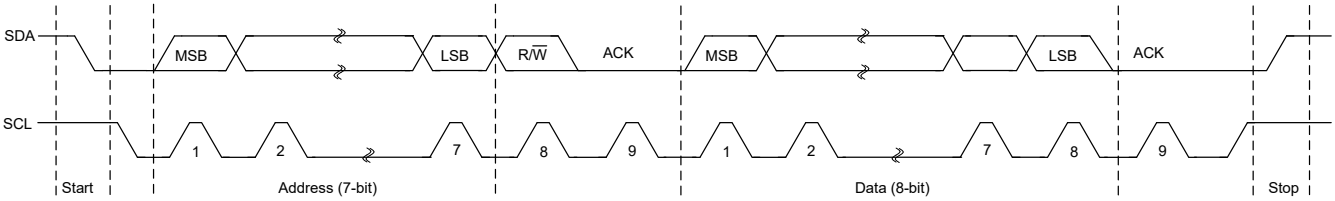



**Figure 25-3. Module Data Transfer**

The data on the SDA line must be stable during the high period of the clock, and the data line can only change when SCL is low (see Figure 25-4), otherwise START or STOP conditions are generated.

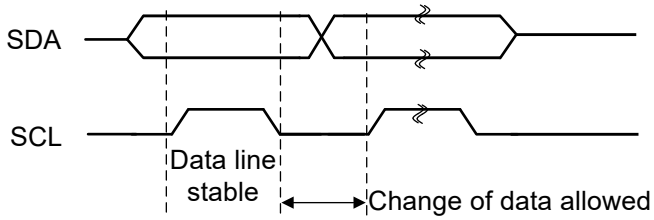



**Figure 25-4. Data Change Period**



Copyright © 2026 Texas Instruments Incorporated

#### 25.2.3.2 START and STOP Conditions

The protocol of the I²C bus defines two states to begin and end a transaction: START and STOP. A high-to-low transition on the SDA line while the SCL is high is defined as a START condition, and a low-to-high transition on the SDA line while SCL is high is defined as a STOP condition. START and STOP conditions are always generated by the controller. The bus is considered busy after a START condition and free after a STOP condition.

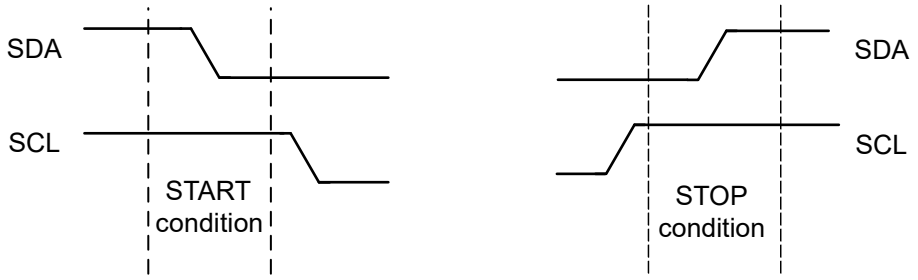



**Figure 25-5. START and STOP Conditions**

The STOP bit determines if the transaction stops at the end of the data cycle or continues on to a repeated START condition.

To generate a single transaction, the I²C controller target address I2Cx.MSA.SADDR register is written with the desired address, the I2Cx.MSA.DIR bit should be set to 1 to start a receive operation and 0 to start a transmit operation , and the control register (I2Cx.MCTR) is written with ACK = X (0 or 1), STOP = 1, START = 1, and RUN = 1 to perform the operation and generate a STOP at the end. When the operation is completed (or aborted due an error) the interrupt flags are set. When the I²C module operates in Controller receiver mode, the ACK bit is normally (in case of no error) set causing the I²C bus controller to transmit an acknowledge automatically after each byte. This bit must be cleared when the I²C bus controller requires no further data to be transmitted from the target transmitter. For more information on I²C controller mode configuration please refer to Section 25.2.4.1

When operating in target mode, the START and STOP bits in the CPU_INT.RIS register indicate detection of start and stop conditions on the bus and the CPU_INT.IMASK can be configured to allow START and STOP to be promoted to controller interrupts (when interrupts are enabled). For more information on I²C target mode configuration please refer to Section 25.2.4.2

Bus Status Flags:

- BUSBSY is set when a START is detected on the bus and sequentially cleared when a STOP condition is detected on the bus. It is also cleared after a clock timeout I2Cx.MSR.CLKTO is set once both SCL and SDA are pulled to high.
- BUSY is also set after a START/RESTART, and cleared after I2Cx.MCTR.MBLEN bytes of data is transferred. It is also cleared after data is NACK ‘d or a STOP.
- IDLE is set when the Controller I²C State machine is in the IDLE state indicating no ongoing transfer.

#### 25.2.3.3 Data Format with 7-Bit Address

Data transfers follow the format shown in Figure 25-6. After the START condition, a target address is transmitted. This address is 7-bits long followed by an eighth bit, which is a data direction bit (this bit is only as Controller mode, DIR bit in the I2Cx.MSA register). If the I2Cx.MSA.DIR bit is 0, it indicates a transmit operation (send), and if it is set to 1, it indicates a request to receive data (receive). A data transfer is always terminated by a STOP condition generated by the controller; however, a controller can initiate communications with another device on the bus by generating a repeated START condition and addressing another target without first generating a STOP condition, see section Repeated Start. Various combinations of receive/transmit formats are then possible within a single transfer. The ninth bit is the Acknowledge bit, which is described in Section 25.2.3.5 .



Copyright © 2026 Texas Instruments Incorporated

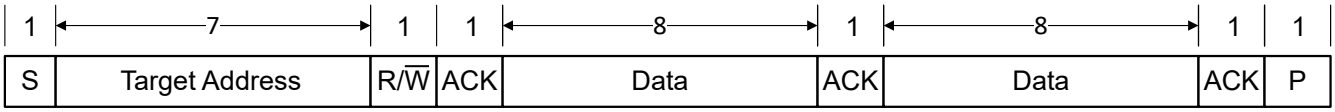



**Figure 25-6. Data Format with 7-Bit Address**

With the I2Cx.SCTR.GENCALL bit the I²C module can be enabled to respond on a General Call on the I²C bus. The General Call is identified by address of 0x00 and the R/W bit set to 0. The General Call interrupt can be enabled with the CPU_INT.IMASK.GENCALL bit.

#### 25.2.3.4 Data Format with 10-Bit Address

Data transfers with 10-bit address follow the format shown in below. After the START condition, a leading ***'11110b'*** introduces the 10-bit addressing scheme. The last two address bits of the first byte concatenated with the eight bits of the second byte of the whole 10-bit address. Devices which only use 7-bit addressing simply ignore messages with the leading '11110b'. The data direction bit (this bit is only as Controller mode, DIR bit in the CSA register) is located in the first 8-bit data.

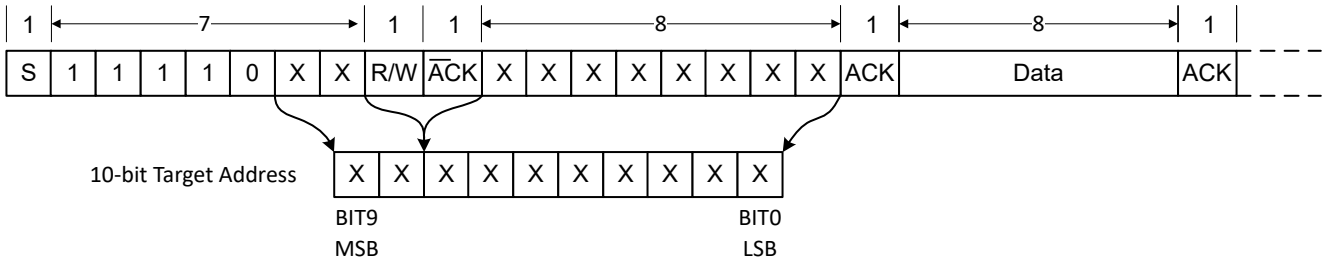



**Figure 25-7. Data Format with 10-Bit Address**

A ***Controller transmitter*** addresses the Target with two address bytes as described below with the RW-Bit='0' followed by data bytes from the Controller.




**Figure 25-8. I2C Controller Transmit with 10-Bit Address**

The ***Controller receiver*** transfer is only possible with a Combined Transfer due to the fact that the second address byte can only be transmitted if the RW-Bit of the first address byte is '0'. Hence, the start of a Controller receiver transfer will be the same as a Controller transmitter transfer followed by a repeated start condition and the first byte of address byte with RW-Bit='1' (switching to Target transmitter mode).

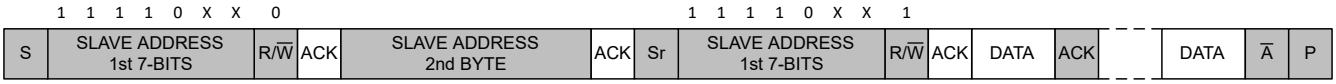



**Figure 25-9. I2C Controller Receive with 10-Bit Address**

#### 25.2.3.5 Acknowledge

All bus transactions have a required acknowledge clock cycle that is generated by the controller. During the acknowledge cycle, the transmitter (which can be the controller or target) releases the SDA line. To acknowledge the transaction, the receiver must pull down SDA during the acknowledge clock cycle. The acknowledge cycle must comply with the data validity requirements.

When a target receiver does not acknowledge the target address, SDA must be left high by the target so that the controller can generate a STOP condition and abort the current transfer or generate a repeated START condition to start a new transfer. If the controller device is acting as a receiver during a transfer, it is responsible



Copyright © 2026 Texas Instruments Incorporated

for acknowledging each transfer made by the target. Because the controller controls the number of bytes in the transfer, it signals the end of data to the target transmitter by not generating an acknowledge on the last data byte. The target transmitter must then release SDA to allow the controller to generate the STOP or a repeated START condition.

A target can generate ACK/NACK manually or automatically. When I2Cx.SACKCTL.ACKOEN=0, the target will send automatic ACK. Note that due to the FIFO, the device will receive and ACK all bytes automatically until the RX FIFO is full. Setting SACKCTL.ACKOEN =1 enables manual ACK. Manual ACK override can be used to evaluate each received byte or to slow down the communication when automatic FIFO reception is not desired. When manual ACK override operation is enabled, the I²C target module's clock is pulled low after the last data bit until this SACKCTL.ACKOVAL is written with the indicated response. The reception of new data is indicated by the SRXDONE interrupt flag.

If the controller receives a NACK while transmitting data the NACK and MTXDONE bit will be set in the RIS registers. If there is still data in the FIFO the TXEMPTY bit will not be set to inform software that a TX FIFO flush may be required.

#### 25.2.3.6 Repeated Start

The direction of data flow on SDA can be changed by the controller, without first stopping a transfer, by issuing a repeated START condition. This is called a RESTART, see data format in Figure 25-10. After a RESTART is issued, the target address is again sent out with the new data direction specified by the R/W bit.

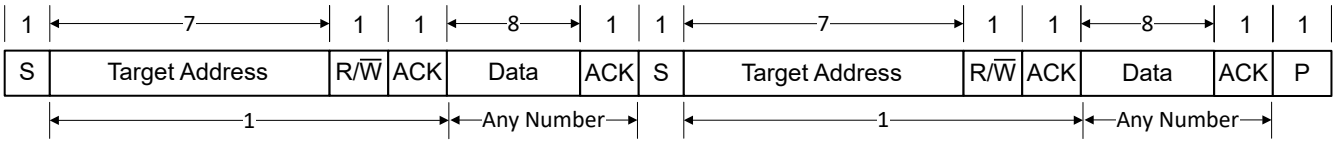



**Figure 25-10. Data Format with Repeated Start**

A repeated start sequence for a Controller transmit is as follows:

1. When the device is in the idle state, the Controller writes the target address to the I2Cx.MSA register and configures the DIR bit for the desired transfer type.
2. Data is written to the I2Cx.MTXDATA register.
3. When the BUSY bit in the MSR register is 0, the BURSTRUN and START bit in the I2Cx.MCTR register need to be set to initiate a transfer.
4. Wait till the BUSY bit in the I2Cx.MSR register gets 0.
5. The Controller does not generate a STOP condition but instead writes another target address to the I2Cx.MSA register and then sets the BURSTRUN and START bit again with a write operation to initiate the repeated START.

A repeated start sequence for a Controller receive is similar:

1. When the device is in idle, the Controller writes the target address to the I2Cx.MSA register and configures the DIR bit for the desired transfer type.
2. When the BUSY bit in the I2Cx.MSR register is 0, the BURSTRUN and START bit in the I2Cx.MCTR register need to be set to initiate a transfer.
3. The controller reads data from the I2Cx.MRXDATA register.
4. Wait till the BUSY bit in the I2Cx.MSR register gets 0.
5. The Controller does not generate a STOP condition but instead writes another target address to the I2Cx.MSA register and then sets the BURSTRUN and START bit again with a write operation to initiate the repeated START.

#### 25.2.3.7 SCL Clock Low Timeout

The I²C target can extend the transaction by pulling the clock low to slow down communication. The I²C module has a 12-bit programmable counter that is used to track how long the clock has been held low. The upper 8



Copyright © 2026 Texas Instruments Incorporated

bits of the count value are software programmable through the I2CTIMEOUT_CTL register. The CNTL value programmed in the I2CTIMEOUT_CTL.TCNTLA register has to be greater than 0x01 to enable the timeout feature . Please note that the low timeout configuration needs to be set only during initialization and not during active state.

The application can program the eight most significant bits of the counter to reflect the acceptable cumulative low period in a transaction. Each count is equal to a timeout period of (1 + TPR) × 12 of functional clocks FCLK where the TPR is the programmable timer period. The Timeout counter A counts for the entire time SCL is held Low continuously. When SCL is high, the Timeout counter A is reloaded with the value in the I2CTIMEOUT_CTL.TCNTLA register bit, and begins counting down from this value at the falling edge of SCL.

The BUSSCLK clock generated for the timeout counter keeps running irrespective of the programmed I²C speed even if SCL is held low on the bus.

The I²C clock low timeout period is calculated as follows:

- Cumulative clock low period = Timeout counter * One timeout period
- One timeout period = BUSSCLK period * (1 + TPR) * 12
- Timeout counter = I2CTIMEOUT_CTL.TCNTLA register (this register contains the upper 8-bits of a 12-bit counter value, the lower 4-bits are set to 0)

As an example, if the BUSSCLK: clock is 20MHz and the I²C module was operating at 100kHz speed, the TPR would be equal to 19. See Section 25.2.1.1 on how TPR is calculated. One timeout period is equal to (1 / 20MHz) × ( 1 + 19) × 12 or 12µs. Programming the I2CTIMEOUT_CTL.TCNTLA register to 0xDA would translate to the value 0xDA0, because the lower 4-bits are set to 0x0. This would translate to a decimal value of 3488 clocks or a cumulative clock low period of 3488 × 12µs, or 41.856ms at 100kHz.

The TIMEOUTA bit in the I²C Controller Raw Interrupt Status (RIS) register is set when the clock timeout period is reached, allowing the controller to start corrective action to resolve the remote target state. In addition, the BUSBSY bit in the I²C Controller Status MSR register is set. This bit is cleared after I²C goes to idle or during the I²C controller reset. The status of the raw SDA and SCL signals are readable by software through the SDA and SCL bits in the I²C Controller Bus Monitor MBMON register to help determine the state of the remote target.

In the event of a timeout condition, application software must choose how it intends to attempt bus recovery. If a timeout is detected before the end of a transfer (receive or transmit), software should flush the FIFO before initializing the next transfer. The clock low timeout is needed for SMBus and PMBus implementation.

**Note**

The Controller clock low timeout counter register I2CTIMEOUT_CNT.TCNTA counts for the entire time SCL is held low continuously. When SCL is high, the counter is reloaded with the value in the I2CTIMEOUT_CTL.TCNTLA register, and begins counting down from this value at the falling edge of SCL.

#### 25.2.3.8 Clock Stretching

In controller mode, the clock stretching can be disabled if no targets on the bus support it, allowing the controller to reach the maximum speed on the bus. Otherwise the clock can be slowed by a target keeping the clock low or due to the clock status detection delay within the I²C module.

To ensure compliance to the I²C specification, clock stretching needs to be enabled. Clock stretching is activated when either the RX FIFO full or TX FIFO empty is set. Clock stretching support can be enabled or disabled by configuring the CLKSTRETCH bit in I2Cx.MCR register.

In the target mode, clock stretching is enabled by default and it is signaled by the TREQ and RREQ bits of the I²C target status register I2Cx.SSR.

- When TREQ bit is set, it indicates the I²C controller has been addressed as a target transmitter and is using clock stretching to delay the controller until data has been written to the STXDATA FIFO (Target TX FIFO is empty).



Copyright © 2026 Texas Instruments Incorporated

- When RREQ bit is set, it indicates the I²C controller has outstanding receive data from the I2C controller and is using clock stretching to delay the controller until the data has been read from the SRXDATA FIFO (Target RX FIFO is full).

**Note**

Clock stretching in target mode may be used together with an asynchronous fast clock request to support bringing the device into a suspended low power mode state upon detection of an I2C start bit, enabling the I2C module to support 100kHz (standard mode) or 400kHz (fast mode) operation out of low power modes where the bus clock speed is below the minimum oversampling speed required by the respective mode. Refer to the clock selection and I²C speed section for the minimum frequency requirements. When clock stretching is used together with an asynchronous fast clock request, it is possible for the device to wait in STOP or STANDBY mode when the I²C is idle, and when an I²C bus edge is seen, the fast clock request will requests SYSOSC at base frequency and the bus clock will switch to SYSOSC at base frequency, allowing the I²C module to process the bus transaction and wake the processor if an interrupt is generated.

#### 25.2.3.9 Dual Address

The I²C target interface supports dual address capability for the target. An additional programmable I²C Target Own Address Register I2CX. SOAR is provided and can be matched if enabled. When dual address disabled (I2Cx.SOAR2.OAR2EN=0) , the I²C target provides an ACK on the bus if the address matches the OAR field in the I2Cx.SOAR register. In dual address mode (I2Cx.SOAR2.OAR2EN=1), the I²C target provides an ACK on the bus if either the OAR field in the I2Cx.SOAR register or the OAR2 field in the I2Cx.SOAR2 register is matched.

The OAR2SEL bit in the I2Cx.SSR register indicates if the address that was ACKed is the alternate address or not. When this bit is clear, it indicates either the primary address match or no OAR2 address match.

#### 25.2.3.10 Arbitration

A controller can start a transfer only if the bus is idle. It's possible for two or more controllers to generate a START condition within minimum hold time of the START condition. In these situations, an arbitration scheme takes place on the SDA line, while SCL is high (see Figure 25-11) . The first controller transmitter that generates a logic high is overruled by the opposing controller generating a logic low and the losing controller releases the bus until the bus is idle again.

Arbitration can take place over several bits. Its first stage is a comparison of address bits, and if both controllers are trying to address the same device, arbitration continues on to the comparison of data bits.

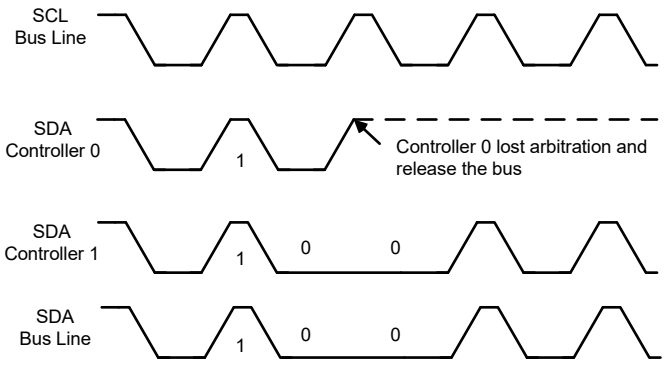



**Figure 25-11. Arbitration Procedure Between Two Controller Transmitters**

When an arbitration lost is detected the I2Cx.MSR.ARBLST flag is set. It will be reset by the hardware with the next STOP condition detected on the bus. Additionally the ARBLOST flags in CPU_INT.RIS registers are set.

If arbitration is lost when the I²C controller has initiated a transfer, the application should execute the following steps to correctly handle the arbitration loss:



Copyright © 2026 Texas Instruments Incorporated

- Flush TX FIFO
- Clear and mask the TX Empty interrupt by the TXEMPTY bit through the IMASK and ICLR register.
- Once the bus is IDLE, the TXFIFO can be filled and enabled, the TXEMPTY bit can be unmasked and a new transaction can be initiated.

#### 25.2.3.11 Multiple Controller Mode

When operating in a multiple controller system, the I2Cx.MCR.MMST bit must be set.

During the arbitration procedure, the clocks from the different controllers must be synchronized. A device that first generates a low period on SCL overrules the other devices, forcing them to start their own low periods. SCL is then held low by the device with the longest low period. The other devices must wait for SCL to be released before starting their high periods

In multiple-controller configuration, the clock synchronization (Figure 25-12) during the arbitration is enabled, the SCL high time counts once the SCL line has been detected high. If not enabled the high time counts as soon as the SCL line has been set high by the I²C controller which allows the I²C to reach the maximum speed by the I2Cx.MTPR register specified speed.

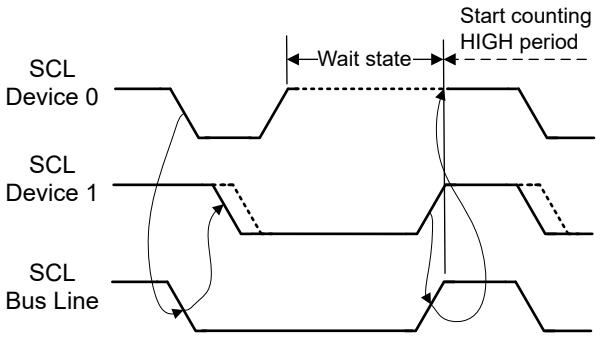



**Figure 25-12. Clock Synchronization**

#### 25.2.3.12 Glitch Suppression

The I²C module supports glitch suppression on the SCL and SDA lines to meet the 50ns glitch suppression as specified in the I²C specification.

### Analog glitch Filter

By default, an analog glitch filter is enabled and configured to suppress spikes with a pulse width up to 50ns. I²C spec advises to suppress noise spikes of less than 50ns. The user can disable this filter by clear the I2Cx.GFCTL.AGFEN bit and the suppression pulse width can also be configured by using I2Cx.GFCTL.AGFEN. The analog glitch filter can only be used to wake up the I²C in low-power mode.

### Digital glitch Filter

The DGFSEL bits in the I2Cx.GFCTL register can be programmed to provide glitch suppression on the SCL and SDA lines and assure proper signal values. The glitch suppression value is in terms of the I²C functional clocks. All signals are delayed internally when glitch suppression is nonzero. For example, if DGFSEL bit is set to 0x7, 31 clocks should be added onto the calculation for the expected transaction time, for more information please see register description. Digital glitch filter can't be used to wake up the I2C from low-power mode.

**Table 25-3. Glitch Suppression Filter**

| | Analog Glitch Filter | Digital Glitch Filter |
|---|---|---|
| **Default** | Default enabled with 50ns | Default bypassed |
| **Pulse width of suppressed spikes** | Configurable<br>5ns, 10ns, 25ns, 50ns | Programmable I²C clock cycle<br>1, 2, 3, 4, 8, 16, 31 |
| **Benefits** | Available without needing clock | • Programmable length to provide extra filtering<br>• Stable filtering length |



Copyright © 2026 Texas Instruments Incorporated

**Table 25-3. Glitch Suppression Filter (continued)**

| | Analog Glitch Filter | Digital Glitch Filter |
|---|---|---|
| **Limitation** | Variation with temperature, voltage, process | Does not work in low-power mode wakeup when there is no sufficient clock .<br>Enabled only after 3 clock cycles after start of I²C packet |

**Note**

The glitch filter configuration within the register I2Cx.GFCTL should only be modified while the I²C module for controller and target is not enabled.

#### 25.2.3.13 FIFO operation

The receive data register I2Cx.MRXDATA (for controller mode) and I2Cx.SRXDATA (for target mode) are user accessible and contains the current character to be read from the RX FIFO stack. The last received character from the receive shift register will be push to the end of the FIFO Stack.

The transmit data register I2Cx.MTXDATA (for controller mode) and I2Cx.STXDATA (for target mode) are user accessible and holds the data last written data to the TX FIFO. The TX FIFO contains the data waiting to be moved into the transmit shift register and transmitted on SDA.

FIFOs are available for the controller and target receive and transmit. Each FIFO entry has a width of 8 bits and should be accessed in byte mode. Each FIFO has a programmable threshold point (configured by RXTRIG and TXTRIG bits in the I2Cx.MFIFOCTL register for controller mode and I2Cx.SFIFOCTL register for target mode) which indicates when the FIFO service interrupt should be generated. Additionally, a FIFO receive full and transmit empty interrupt can be enabled in the interrupt mask (IMASK) registers for the controller and target.

The content of the FIFO can be flushed with setting TXFLUSH or RXFLUSH bit to 1 in the I2Cx.FIFOCTL registers. When the I²C gets reset the content of the FIFO needs also to be cleared. FIFO clear should only be executed while the I²C is in IDLE mode. Before triggering the flush the FIFO interrupts should be disabled and after flush has completed the interrupt flags needs to be checked.

##### *25.2.3.13.1 Flushing Stale Tx Data in Target Mode*

In most use cases, the user does not want to transmit leftover data of the I2C peripheral Tx FIFO from previous frame in the next frame. The device provides a mechanism for the software to choose whether to flush stale data or not.

A status bit SSR.STALE_TXFIFO represents whether the data present inside I2C peripheral TX FIFO is stale or not. A control bit SCTR.TXWAIT_STALE_TXFIFO is used to enable modified empty indication to target logic - indicate empty to I2C peripheral FSM when Tx FIFO is empty OR stale data present in Tx FIFO. The SCTR.TXEMPTY_ON_TREQ control bit allows the RIS.STXEMPTY interrupt to be used to indicate the TREQ condition, the condition when SCL is being stretched waiting for transmit data from the I2C peripheral.

**Recommended Sequence**

1. When SCTR.TXWAIT_STALE_TXFIFO is set, on STOP, restart, or timeout, I2C peripheral FSM gets an empty indication even though stale data is present inside I2C peripheral Tx FIFO.
2. I2C module doesn’t immediately generate an TX FIFO empty interrupt/DMA request. Instead, waits till the controller on the I2C bus attempts to read data from the I2C peripheral, at which point the I2C peripheral clock stretches.
3. The I2C peripheral issues the clock stretch (TREQ) interrupt to the CPU when SCTR.TXEMPTY_ON_TREQ is set.
4. Application software in the ISR checks for a SSR.STALE_TXFIFO flag and flushes the TX FIFO using SFIFOCTL.TXFLUSH, which also clears the status of SSR.STALE_TXFIFO.



Copyright © 2026 Texas Instruments Incorporated

#### 25.2.3.14 Loopback mode

The I²C modules can be placed into an internal loopback mode for diagnostic or debug work by setting the LPBK bit in the I²C controller configuration I2Cx.MCR register. In loopback mode, the SDA and SCL signals from the controller part of the I²C are tied to the SDA and SCL signals of the target part of the I²C module to allow internal testing of the device without having to connect the I/Os. The SWUEN bit should be set to 0 for internal loopback to work correctly.

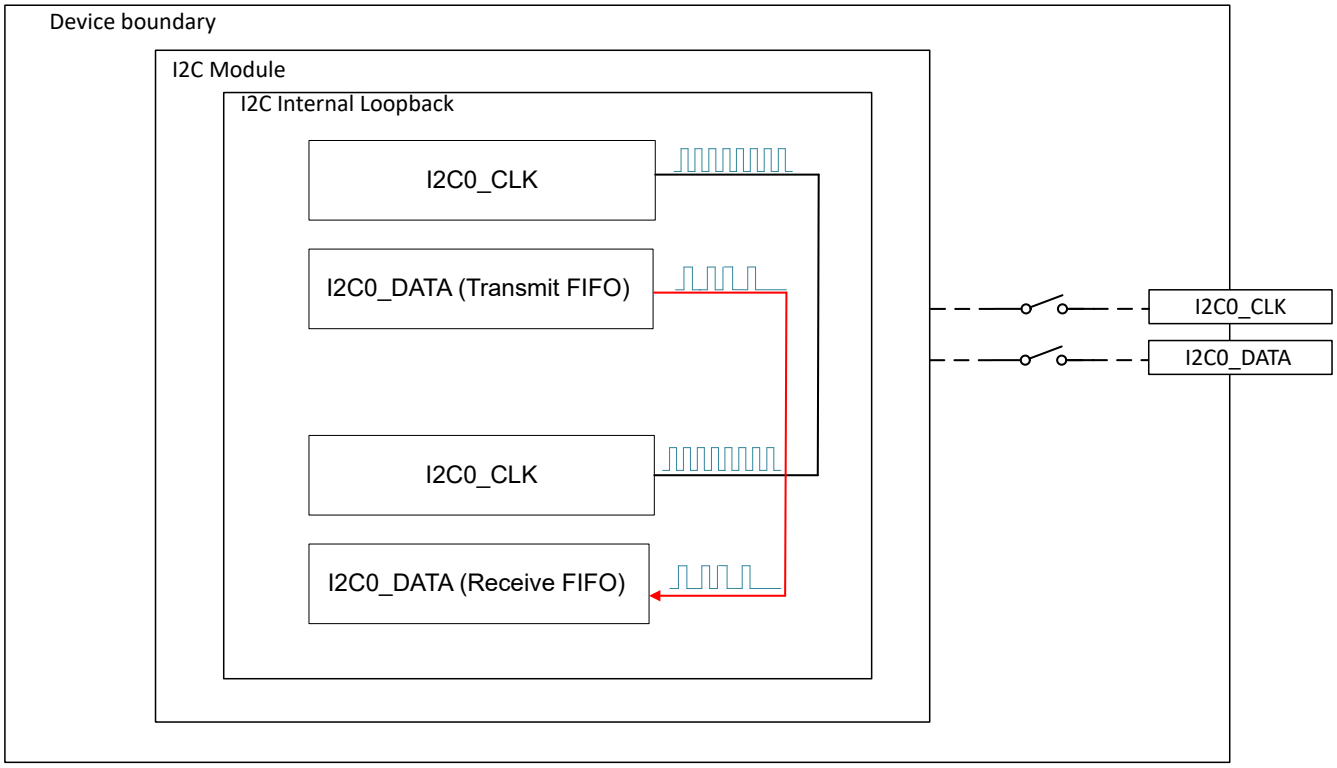



**Figure 25-13. I2C Internal Loopback mode**

#### 25.2.3.15 Burst Mode

A burst mode is provided for the controller module which allows a sequence of data transfers using the DMA or software to handle the data in the FIFO. The burst mode is enabled by setting the MBLEN bits in the controller control register I2Cx.MCTR to a value greater then '1'. This sets the number of bytes transferred by a burst. A copy of this value is automatically written to the MBCNT bits in the I²C controller status register I2Cx.MSR to be used as a down-counter during the burst transfer.

The bytes written to the I²C FIFO are transferred to the RX FIFO or TX FIFO depending on whether a transmit or receive is being executed. If data is not acknowledged (NACK) during a BURST and the STOP bit is set in the I2Cx.MCTR register, the transfer terminates. If the STOP bit is not set, software must issue a STOP or repeated START when a NACK interrupt is asserted. In the case of a NACK, the MBCNT bits in the I2Cx.MSR register can be used to determine the amount of data that was transferred prior to the burst termination. If the address is not acknowledged (NACK) during a transfer, then a STOP is issued.

#### 25.2.3.16 DMA Operation

The I²C provides an interface to the DMA controller with two channels. The DMA operation of the I²C is enabled through the I²C event register and DMA peripheral registers. When the DMA functionality is enabled, the I²C asserts a DMA request on the selected channel when the associated FIFO can transfer or receive data.

For more information about I²C event and DMA, please refer to Interrupt and Events Support and DMA Trigger Publisher sections.



Copyright © 2026 Texas Instruments Incorporated

**Note**

Per DMA Channel only one event source should be enabled by the IMASK register at the same time. The DMA transaction descriptor needs to match for the selected trigger with a correct setup for either Controller or Target and RX or TX configuration. The DMA trigger should only be changed when no I²C transfer is ongoing and a previous triggered DMA transfer has been finished. If this cannot be ensured the I²C and DMA channel should be disabled first.

#### 25.2.3.17 Low-Power Operation

I²C supports operation in low-power modes.

Supported power mode for controller mode:

- Controller mode with 100kHz speed can operate in RUN, SLEEP, STOP power mode
- Controller mode with 400kHz speed can operate in RUN, SLEEP power mode
- Controller mode with 1MHz speed can operate in RUN, SLEEP power mode

In low-power mode, the I²C interface automatically request the clock selected in the CLKSEL control register after detecting a start condition. The clock request is released after detection of the STOP condition.

Supported power mode and wakeup mode for target mode:

- Target mode with 100kHz speed can operate in RUN, SLEEP, STOP power mode
- Target mode with 100kHz speed can't operate in STANDBY mode because the maximum bus clock in STANDBY mode is 32kHz and support for 100kHz speed requires a minimum 400kHz clock. However, in STANDBY mode the I2C target can still wakeup from the start bit and perform the Asynchronous Fast Clock Requests to temporarily get a 24MHz/32MHz clock to receive the data until the FIFO or address match interrupt wakes up the CPU.
- Target mode with 400kHz speed in RUN, SLEEP power mode
- Target mode with 1MHz speed in RUN, SLEEP power mode

### *25.2.4 Protocol Descriptions*

#### 25.2.4.1 I²C Controller Mode

As showing in the function block diagram section, I²C peripheral has a set of specific controller registers to configure the operation when the module is configured as an I²C target mode.

##### *25.2.4.1.1 Controller Configuration*

The I²C Controller Control register I2Cx.MCTR and I²C Controller Target Address register I2Cx.MSA are used for controlling controller transmit and receive modes. The following settings can be modified to control the different transactions.

- Length indicates the number of bytes for the transaction and is configured by I2Cx.MCTR.MBLEN bit
- Direction (transmit or receive) is configured by I2Cx.MSA.DIR bit
- ACK generation is configured by I2Cx.MCTR.MBLEN bit
- STOP condition generation is configured by I2Cx.MCTR.STOP bit
- START or repeated START condition generation is configured by I2Cx.MCTR.START bit
- RUN enable the controller transaction and is configured by I2Cx.MCTR.BURSTRUN bit

**Controller send data transactions**

**Table 25-4. Start Transmit From Idle Mode**

| Length | Direction | ACK | STOP | START | RUN | Format | Comments |
|---|---|---|---|---|---|---|---|
| n (n>0) | 0 | x | 0 or 1 | 1 | 1 | START+ADDR+R/W+ DATA*n+(STOP) | Sending of STOP depends on STOP bit |



Copyright © 2026 Texas Instruments Incorporated

**Table 25-5. Continue Transmit When Last Transmission Finished Without stop**

| Length | Direction | ACK | STOP | START | RUN | Format | Comments |
|---|---|---|---|---|---|---|---|
| n (n>0) | 0 | x | 0 or 1 | 0 | 1 | DATA*n+ (ACK/NACK)+ (STOP) | Sending of STOP depends on STOP bit |

If there is a NACK response from the target, the controller will automatically send out a stop to finish the transmit. The Controller will be unable to send a RESTART after ADDR or DATA NACK.

### Controller receive data transactions

**Table 25-6. Start Receive From Idle Mode**

| Length | Direction | ACK | STOP | START | RUN | Format | Comments |
|---|---|---|---|---|---|---|---|
| n (n>0) | 1 | 0 or 1 | 0 or 1 | 1 | 1 | START+ADDR +R/ W+DATA *n +(ACK/NACK) +(STOP) | The last data ACK or NACK depend on ACK bit; additional sending of STOP depends on STOP bit |

**Table 25-7. Continue Receive When Last Receive Finished Without STOP or NACK**

| Length | Direction | ACK | STOP | START | RUN | Format | Comments |
|---|---|---|---|---|---|---|---|
| n (n>0) | 1 | 0 or 1 | 0 or 1 | 0 | 1 | DATA*n+ (ACK/NACK) + (STOP) | The last data followed by ACK or NACK depend on ACK bit; additional sending of STOP depends on STOP bit |

This configuration is not allowed if last transaction ended with NACK, as NACK can only be followed by STOP or RESTART. The ACK and STOP bits should not be set to 1 at the same time, as the target needs to be informed to release bus lines before sending out STOP.

### Controller repeated start transactions

If last transmit or receive finished without stop, a repeat start can be generated to initiate a new transaction

**Table 25-8. Repeated Start Transmit**

| Length | Direction | ACK | STOP | START | RUN | Format | Comments |
|---|---|---|---|---|---|---|---|
| n (n>0) | 0 | 0 or 1 | 0 or 1 | 1 | 1 | Restart+ADDR +R/ W+DATA*n +(STOP) | Additional sending of STOP depends on STOP bit |

If there is a NACK response from the target, the controller will automatically send out a stop to finish the transmit. The Controller will be unable to send a RESTART after ADDR or DATA NACK.

**Table 25-9. Repeated Start Receive**

| Length | Direction | ACK | STOP | START | RUN | Format | Comments |
|---|---|---|---|---|---|---|---|
| n (n>0) | 1 | 0 or 1 | 0 or 1 | 1 | 1 | Restart+ADDR +R/ W+DATA*n +(ACK/NACK) +(STOP) | The last data followed by ACK or NACK depend on ACK bit; additional sending of STOP depends on STOP bit |

The ACK and STOP bits should not be set to 1 at the same time, as the target needs to be informed to release bus lines before sending out STOP.



Copyright © 2026 Texas Instruments Incorporated

### Controller send STOP only transaction

**Table 25-10. Send STOP only**

| Length | Direction | ACK | STOP | START | RUN | Format | Comments |
|---|---|---|---|---|---|---|---|
| n (n>=0) | X | X | 1 | 0 | 1 | STOP | STOP command |

It is only allowed to send after previous transaction success finished, and STOP can't be sent without a NACK to the target if controller is currently in receive mode

### Controller Quick Command transaction

The Quick command could only be sent at the transaction beginning, not following other transactions (without stop) or repeat start.

**Table 25-11. Controller quick command**

| Length | Direction | ACK | STOP | START | RUN | Format | Comments |
|---|---|---|---|---|---|---|---|
| 0 | 0/1 | X | 1 | 1 | 1 | START+ADDR +R/ W+STOP | Quick command |

#### *25.2.4.1.2 Controller Mode Operation*

### I²C Controller Initialization

1. Configure SDA and SCL pin functions and select as input by using the IOMUX registers.
2. Reset the peripheral using I2Cx.RSTCTL register
3. Enable the power to peripheral using I2Cx.PWREN register
4. Select and configure the I²C clock using the CLKCTL and CLKDIV registers.
5. Set the desired SCL clock speed of by writing the TPR bit in I2Cx.MTPR register with the correct value. For more information about how to calculate TPR value refer to Clock Control section. For example, with 20MHz I²C clock to achieve 100 kbps SCL clock, TPR value will be 19 (0x13) so we can write the I2Cx.MTPR register with the value of 0x13.
6. Specify the target address and mode (transmit or receive) for the next operation by writing the I2Cx.MSA register. For example, if the target address is 0x3B (MSA.SADDR[7:1] = 0x3B) and we want to transmit (MSA.DIR[0] = 0x0) data then we can write the I2Cx.MSA register with a value of 0x76.
7. If controller is transmitting data, user can place data (byte) to be transmitted in the data register by writing the I2Cx.MTXDATA register with the desired data.
8. Configure the controller transmit or receive mode by writing the I2Cx.MCTR register. For more information on how to configure I2Cx.MCTR register for different mode please see Controller Configuration section above.
9. Enable desired interrupts and/or DMA event by using CPU_INT.IMASK register.

### I²C Controller Status

User can read the I2Cx.MSR register to check the current state of the I²C controller.

**Table 25-12. Controller Status Register**

| Bit Field | Description |
|---|---|
| BUSY | I2C controller busy. The BUSY bit is set during an ongoing transaction. |
| ERR | I2C error. The error can be from the target address not being acknowledged or the transmit data not being acknowledged. |
| ADRACK | Acknowledge address. This bit is set if the transmitted address was not acknowledged. |
| DATACK | Acknowledge data. This bit is set if the transmitted data was not acknowledged. |
| ARBLST | Arbitration lost. This bit is set if controller lost arbitration. |
| IDLE | I2C bus idle. |
| BUSBSY | I2C bus busy. The bit changes based on the START and STOP conditions, set it bus is busy. |
| CLKTO | Clock timeout error. This bit is set if the clock timeout error has occurred. |



Copyright © 2026 Texas Instruments Incorporated

**Table 25-12. Controller Status Register (continued)**

| Bit Field | Description |
|---|---|
| MBCNT | I2C controller transaction count. This field contains the current countdown value of the transaction. |

### I²C Controller Receiver Mode

For controller to start receive data out of the idle mode, user needs to set the START bit in I2Cx.MCTR register to generate the start condition. Then the controller automatically sends the START condition followed by the target address as soon as it detects that the bus is free. All the process below should be followed.

I2Cx.MSA.DIR is set to 1 to enable receive mode, I2Cx.MCTR.START is set to generate start condition, I2Cx.MBLEN can be programmed to indicate the number of bytes (n) for the receive operation. I2Cx.MCTR.ACK and I2Cx.MCTR.STOP bit can be set or clear based on user configuration. I2Cx.MCTR.BURSTRUN is set to start the operation. The packet format is START+ADDR+R+DATA*n +(ACK/NACK) + (STOP). The last data ACK/NACK depend on ACK bit, additional sending of STOP depends on STOP bit.

After last byte is received, the MRXDONE (0x01) interrupt in CPU_INT.IIDX register is set to indicate that controller receive transaction is completed. User can use the MRXFIFOTRG (0x03) interrupt in CPU_INT.IIDX register to read the data from the receive FIFO. This interrupt will trigger when controller RX FIFO contains >= defined bytes, the trigger level can be defined by using RXTRIG bit in I2Cx.MFIFOCTL register. The flow chart of controller receiver mode is shown in Figure 25-14.



Copyright © 2026 Texas Instruments Incorporated

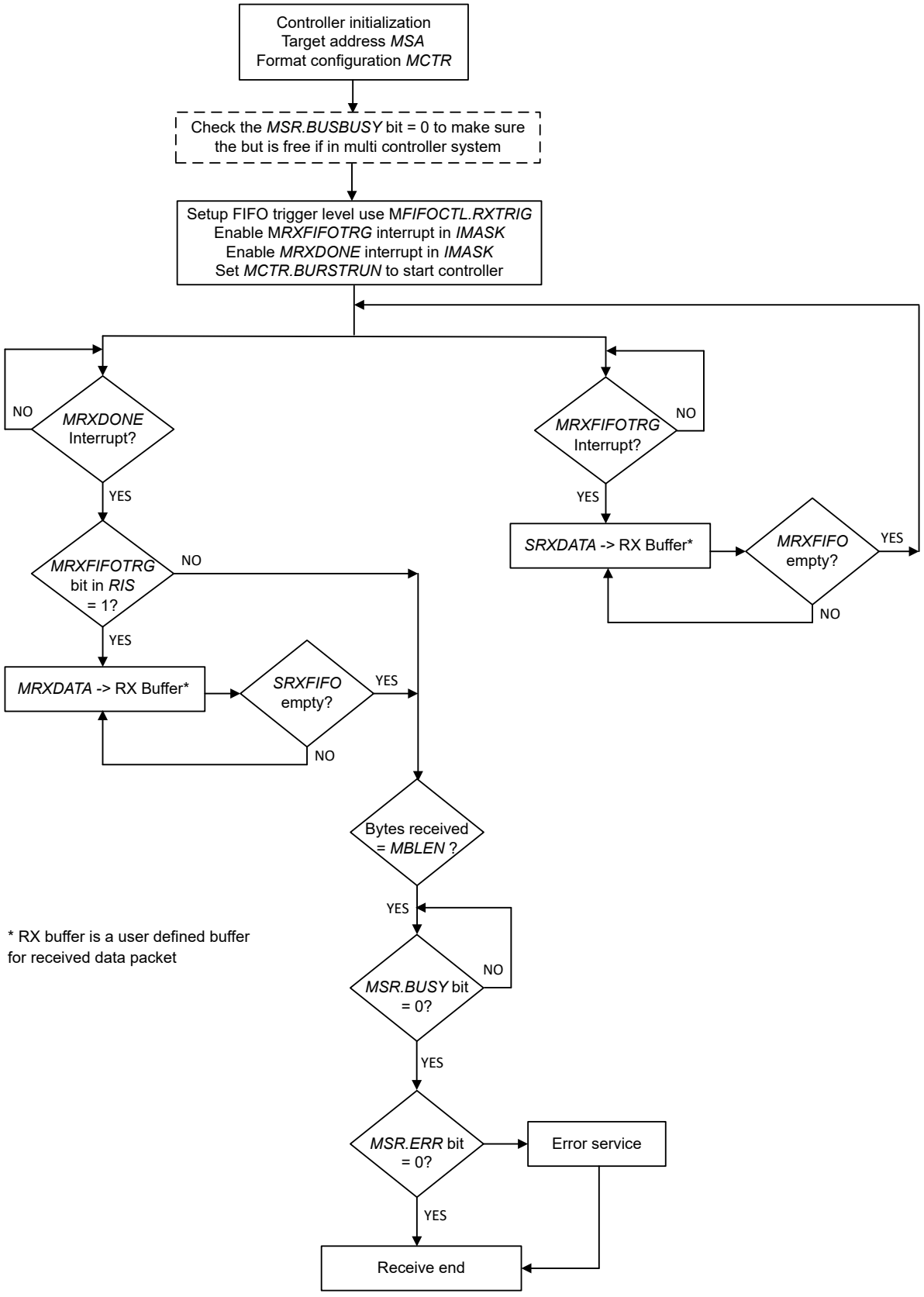



**Figure 25-14. Controller Receiver Mode**



Copyright © 2026 Texas Instruments Incorporated

### I²C Controller transmitter Mode

For controller, to start transmit data out of the idle mode, user need to set the START bit in I2Cx.MCTR register to generate the start condition. Then the controller automatically sends the START condition followed by the target address as soon as it detects that the bus is free. The data written into the MTXFIFO is transmitted if arbitration is not lost during transmission of the target address. All the process below should be followed.

I2Cx.MSA.DIR is cleared to enable transmit mode, I2Cx.MCTR.START is set to generate start condition, I2Cx.MBLEN can be programmed to indicate the number of bytes (n) for the transmit operation. I2Cx.MCTR.STOP bit can be set or clear based on user configuration. I2Cx.MCTR.BURSTRUN is set to start the operation. The packet format is START+ADDR+W+DATA*n + (STOP), sending of STOP depends on STOP bit.

After last byte is transmitted, the MTXDONE (0x02) interrupt in CPU_INT.IIDX register is set to indicate that controller transmit transaction is completed. User can also set use the MTXEMPTY (0x06) interrupt in CPU_INT.IIDX register to see if the MTXFIFO is empty and ready to load more data. This interrupt will trigger if all data in the transmit FIFO have been shifted out. The flow chart of controller transmitter mode is shown in Figure 25-15.



Copyright © 2026 Texas Instruments Incorporated

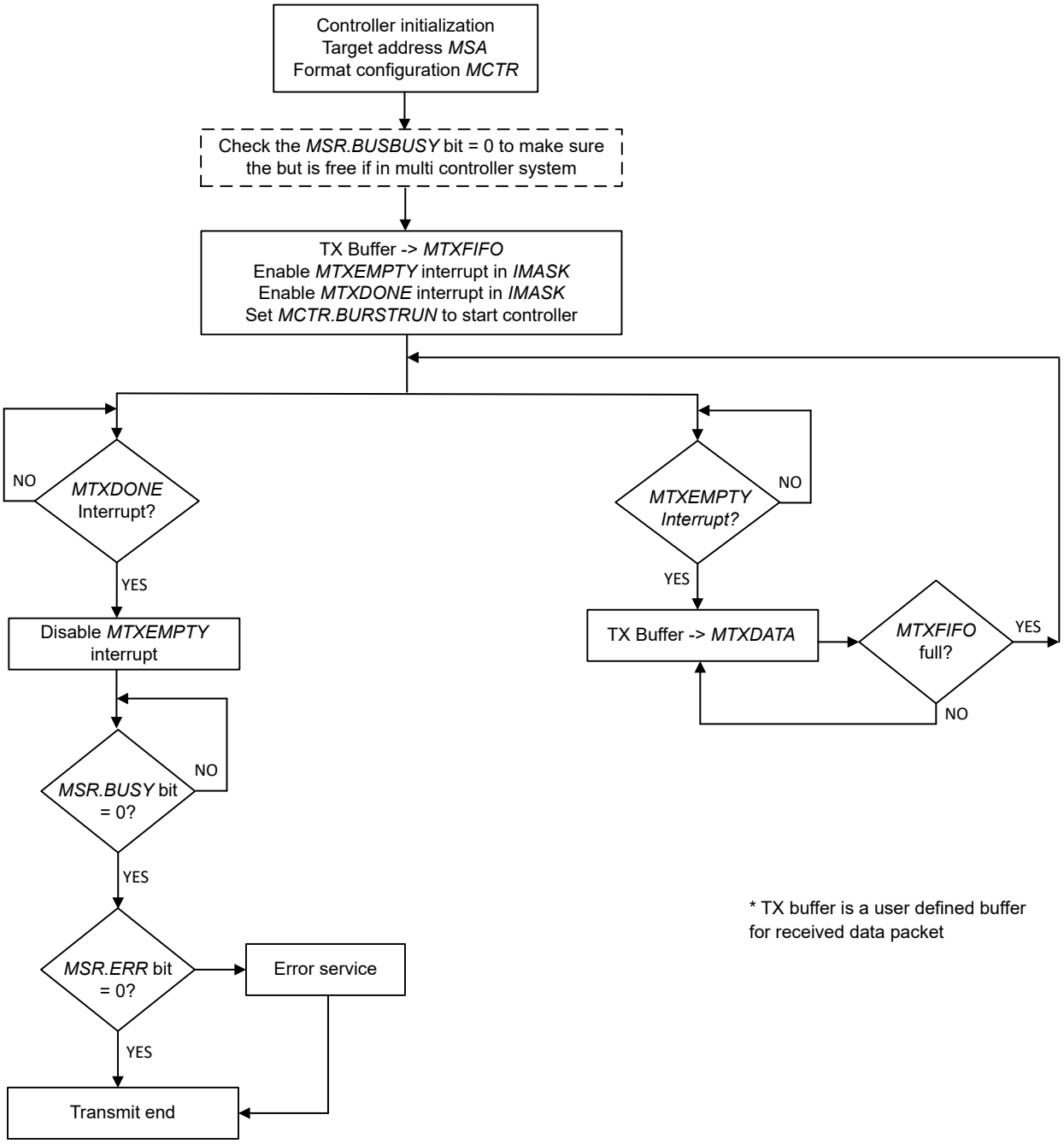



**Figure 25-15. Controller Transmitter Mode**

#### *25.2.4.1.3 Read On TX Empty*

A typical I2C transaction must first write the target to set the register number and then do a repeated start + read to read the data value. However, the RD_ON_EMPTY flag enables two I2C transactions with a single setup of the I2C controller. Software can set up both the write and read together, with the limitation being that the "write" phase cannot send more data than fits in the TX FIFO. This is an optimization to minimize interrupt processing requirements. The high-level set up is:

1. Set MSA to target address with RECEIVE = 1 (RD_ON_TXEMPTY must have I2C_MSR_DIR_RECEIVE set)
2. Set MCTR to initiate the transaction, in this use case write:



Copyright © 2026 Texas Instruments Incorporated

- MBLEN= size for READ phase
- RD_ON_TXEMPTY
- ACK_DISABLE (NACK+STOP after read)
- STOP_ENABLE
- START_ENABLE
- BURST_ENABLE

The expected behavior with this set up is that the controller does a START, transmits all TXFIFO data, and when TXFIFO is empty automatically does a repeated START, reads the MBLEN bytes, then STOP.

#### 25.2.4.2 I²C Target Mode

##### *25.2.4.2.1 Target Mode Operation*

**I²C Target Initialization**

1. Configure SDA and SCL pin functions and select as input by using the IOMUX registers.
2. Reset the peripheral using I2Cx.RSTCTL register
3. Enable the power to peripheral using I2Cx.PWREN register
4. Select and configure the I²C clock using the CLKCTL and CLKDIV registers.
5. Configure at least one target address by writing the 7-bit address to I2Cx.SOAR register. The additional target address can be enabled and configured by using I2Cx.SOAR2 register.
6. Enable desired interrupts and/or DMA event by using CPU_INT.IMASK register.
7. The general call response can be enabled by setting the GENCALL bit in I2Cx.SCTR register.
8. Enable the I²C target mode by setting the ACTIVE bit in I2Cx.SCTR register.

**I²C Target Status**

User can read the I2Cx.SSR register to check the current state of the I²C target.

**Table 25-13. Target Status Register**

| Bit Field | Description |
|---|---|
| RREQ | This bit is set if the I2C controller has outstanding receive data from the I2C controller and is using clock stretching to delay the controller until the data has been read from the SRXDATA FIFO (Target RX FIFO is full) |
| TREQ | This bit is set if the I2C controller has been addressed as a target transmitter and is using clock stretching to delay the controller until data has been written to the STXDATA FIFO (Target TX FIFO is empty). |
| OAR2SEL | This bit is set if SOAR2.OAR2 address matched and ACKed by the target. |
| QCMDST | Quick command status value. This bit is 0 if the last transaction was a normal transaction or a transaction has not occurred. This bit is set if the last transaction was a quick command transaction |
| QCMDRW | Quick command read / write status. This bit only has meaning when the QCMDST bit is set. This bit is 0 if quick command was a write. This bit is set if quick command was a read. |

**I²C Target Receiver Mode**

Target receiver mode is entered when the target address transmitted by the controller matches its own address and a cleared R/W bit is received. In target receiver mode, serial data bits received on SDA are shifted in with the clock pulses that are generated by the controller device. The target device does not generate the clock, but it can hold SCL low if intervention of the CPU is required after a byte has been received.

After the a data byte is received, the SRXDONE (0x11) interrupt in CPU_INT.IIDX register is set to indicate that a byte has been received. The I²C module automatically acknowledges the received data or user can choice to manually send acknowledge after each byte received by configuring the I2Cx.SACKCTL register.

When the controller generates a START condition, the SSTART (0x17) interrupt in CPU_INT.IIDX register is set. When the controller generates a STOP condition, the SSTOP (0x18) interrupt in CPU_INT.IIDX register is set.

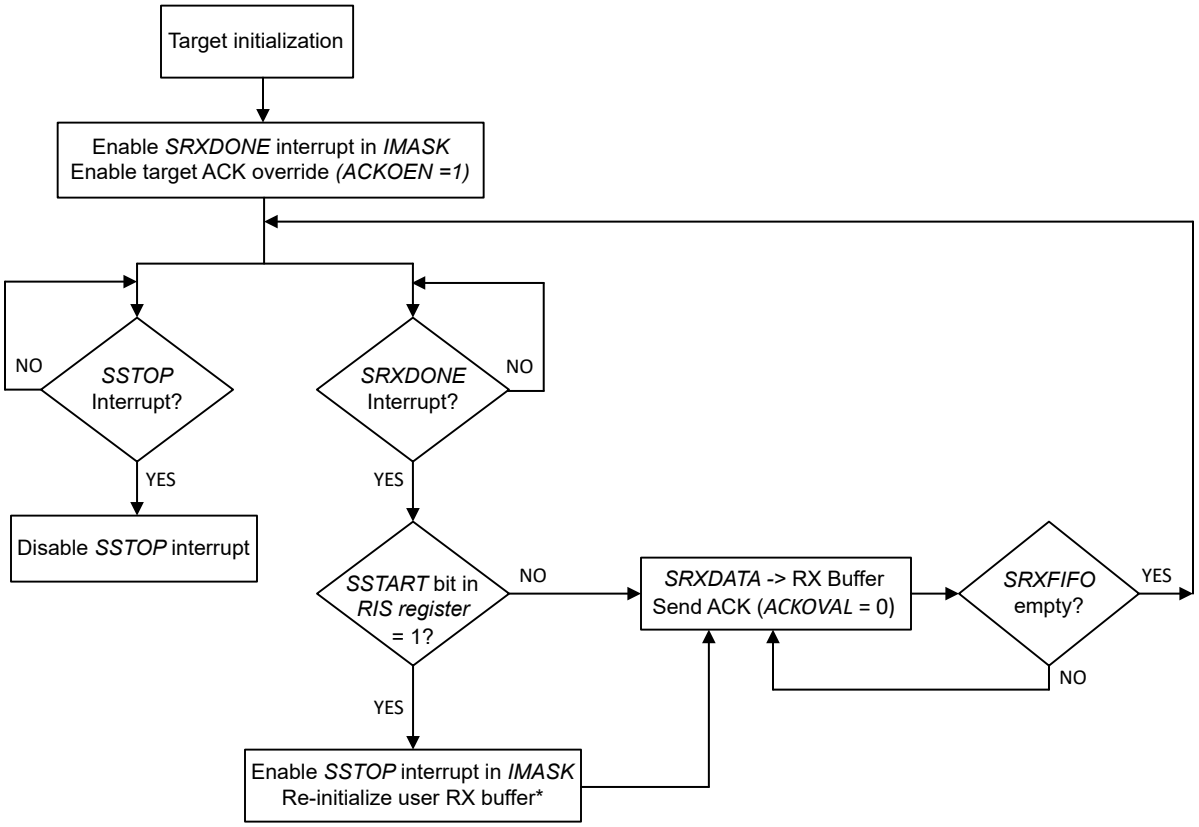
User can also set use the SRXFIFOTRG (0x13) interrupt in CPU_INT.IIDX register to read the data from the receive FIFO. This interrupt will trigger when receive FIFO contains >= defined bytes, the trigger level can be defined by using RXTRIG bit in I2Cx.SFIFOCTL register.



Copyright © 2026 Texas Instruments Incorporated

The SRXDONE approach could be used if target wants to slow down communication to evaluate reception of every byte, while the SRXFIFOTRG approach could be used to maximize throughput and avoid clock stretching.

The flow chart of using SRXDONE and SRXFIFOTRG interrupt to read the receive data are shown in Figure 25-16 and Figure 25-17 respectively.




* RX buffer is a user defined buffer for received data packet

**Figure 25-16. Target Receiver Mode using SRXDONE and ACK override**



Copyright © 2026 Texas Instruments Incorporated

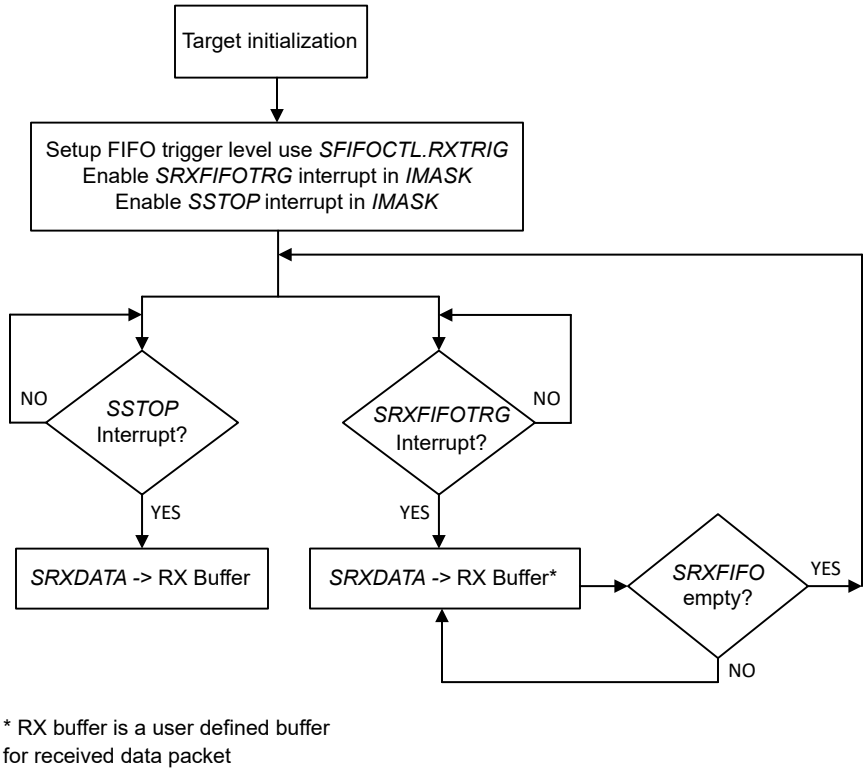



**Figure 25-17. Target Receiver Mode using SRXFIFOTRG and automatic ACK**

### I²C Target Transmitter Mode

Target transmitter mode is entered when the target address transmitted by the controller is identical to its own address with a set R/W bit. The target transmitter shifts the serial data out on SDA with the clock pulses that are generated by the controller device. The target device does not generate the clock, but it can hold SCL low if intervention of the CPU is required after a byte has been transmitted.

After a data byte is transmitted, the STXDONE (0x12) interrupt in CPU_INT.IIDX register is set to indicate that a byte has been transmitted.

When the controller generates a START condition, the SSTART (0x17) interrupt in CPU_INT.IIDX register is set. When the controller generates a STOP condition, the SSTOP (0x18) interrupt in CPU_INT.IIDX register is set.

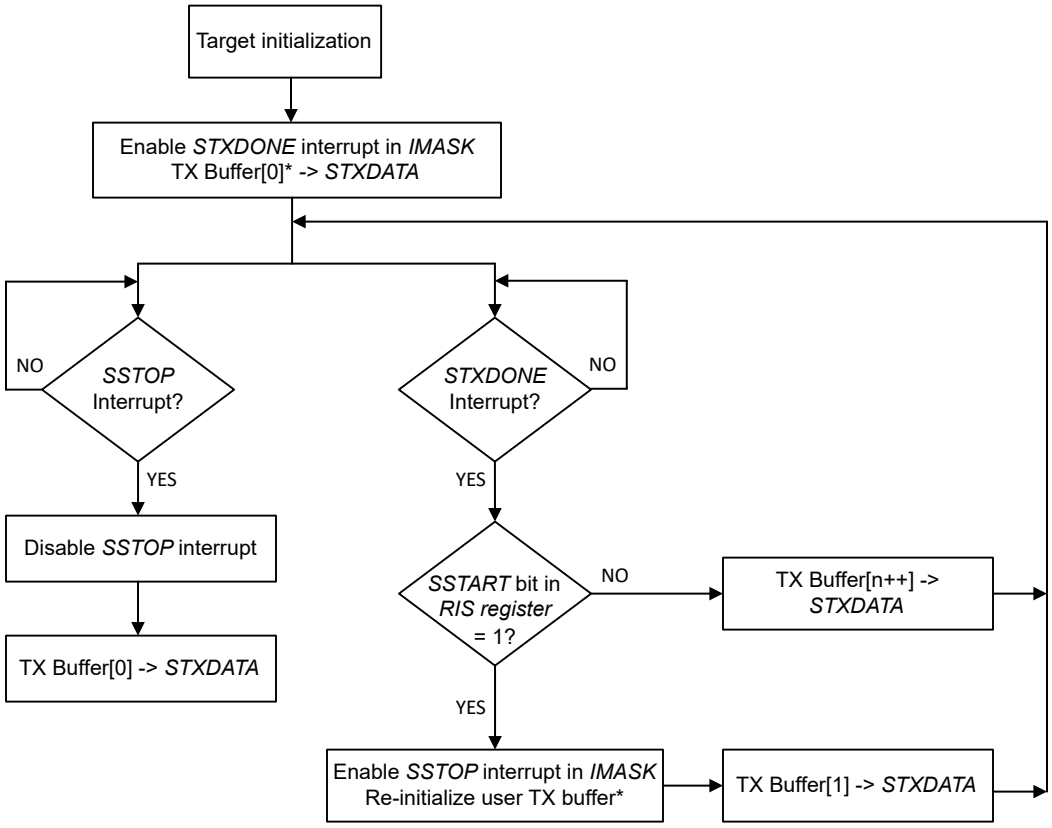
User can also set use the STXFIFOTRG (0x14) interrupt in CPU_INT.IIDX register to load the data to the transmit FIFO. This interrupt will trigger when transmit FIFO contains <= defined bytes, the trigger level can be defined by using TXTRIG bit in I2Cx.SFIFOCTL register.

The STXDONE approach could be used if target wants to slow down communication to evaluate reception of every byte, while the STXFIFOTRG approach could be used to maximize throughput and avoid clock stretching.

The flow chat of using STXDONE and STXFIFOTRG interrupt to transmit data are shown in Figure 25-18 and Figure 25-19.



Copyright © 2026 Texas Instruments Incorporated




* TX buffer is a user defined buffer for transmit data packet

**Figure 25-18. Target Transmitter Mode using STXDONE**



Copyright © 2026 Texas Instruments Incorporated

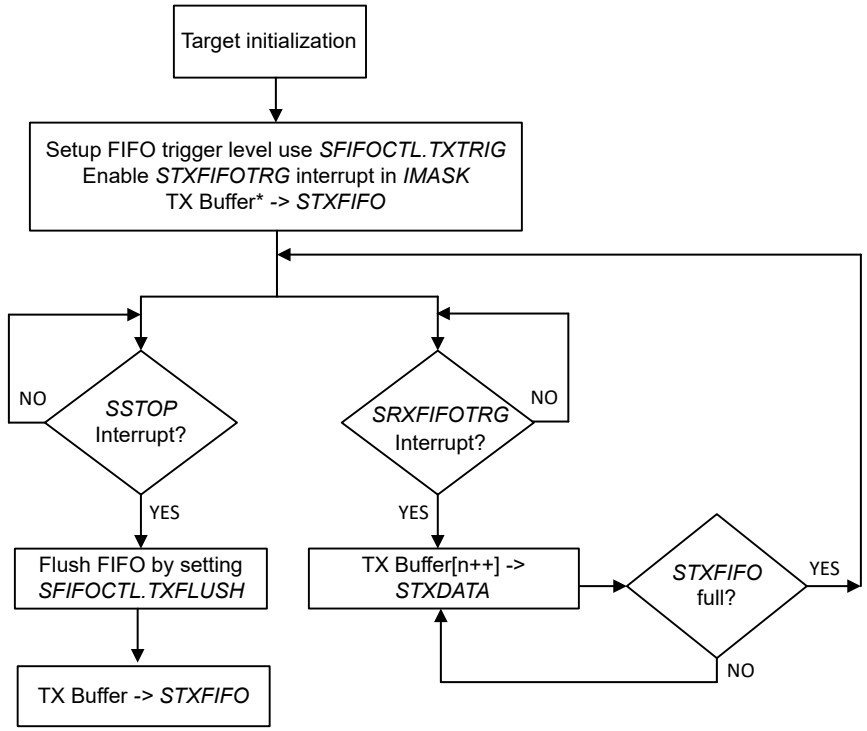



**Figure 25-19. Target Transmitter Mode using STXFIFOTRG**

### *25.2.5 Reset Considerations*

#### Software Reset Considerations

A Software reset can be executed by setting the RESETASSERT bit together with the KEY bit in the I2Cx.RSTCTL register. We recommend issues a reset only after terminating a transaction.

#### Hardware Reset Considerations

A hardware reset also initializes the IO configuration. This sets the IOs to a high-impedance state and with the external pullup resistors for I²C the lines pulled high.

Table 25-14 shows the behavior of status bits when Controller or Target gets disabled.

**Table 25-14. Status Bits When Controller is Disabled**

| Register | Bit | Behavior When Controller Disabled | Behavior When Target Disabled | Behavior After Controller/Target Enabled |
|---|---|---|---|---|
| I2Cx.MSR | Busy | Reset State | Don't care | updates Start condition sending |
| | ERR | Reset State | Don't care | updates on next event detected |
| | ADRACK | Reset State | Don't care | updates on next event detected |
| | DATACK | Reset State | Don't care | updates on next event detected |
| | ARBLST | Reset State | Don't care | updates on next event detected |
| | IDLE | Reset State | Don't care | updates on next event detected |
| | BUSBSY | Reset State | Don't care | updates on next Start detected on bus (or SDA or SCL is low) |
| | CLKTO | Reset State | Don't care | updates on next event detected |
| | MBCNT | Reset State | Don't care | updates on next event detected |
| I2Cx.MCLKOCNT | CLKCNT | Reset State | Don't care | updates with Controller enable when SCL is high |



Copyright © 2026 Texas Instruments Incorporated

**Table 25-14. Status Bits When Controller is Disabled (continued)**

| Register | Bit | Behavior When Controller Disabled | Behavior When Target Disabled | Behavior After Controller/Target Enabled |
|---|---|---|---|---|
| I2Cx.MBMON | SCL | Reset State | Don't care | updates with Controller enable |
| | SDA | Reset State | Don't care | updates with Controller enable |

**Table 25-15. Status Bits When Target is Disabled**

| Register | Bit | Behavior When Controller Disabled | Behavior When Target Disabled | Behavior After Controller/Target Enabled |
|---|---|---|---|---|
| I2Cx.SSR | RREQ | Don't care | Reset State | updates on next event detected |
| | TREQ | Don't care | Reset State | updates on next event detected |
| | OAR2SEL | Don't care | Reset State | updates on next event detected |
| | QCMDST | Don't care | Reset State | updates on next event detected |
| | QCMDRW | Don't care | Reset State | updates on next event detected |
| I2Cx.SFIFOSR | RXFIFOCNT | Don't care | Unchanged | updates on FIFO access |
| | TXFIFOCNT | Don't care | Unchanged | updates on FIFO access |

### *25.2.6 Initialization*

Please see Section 25.2.4.1.2 for controller initialization and Section 25.2.4.2.1 for target initialization.

**Note**

The configuration registers of the I2C module (including the clock configuration, mode configuration, and timeout count) must not be modified by application software when an I2C data transfer is in progress.

### *25.2.7 Interrupt and Events Support*

The I²C module contains three event publishers and no event subscribers. One event publisher (CPU_INT) manages I²C interrupt requests (IRQs) to the CPU subsystem through a static event route. The second and third event publishers (DMA_TRIG1, DMA_TRIG0) are used to setup the trigger signaling for the DMA through DMA event route.

The I²C events are summarized in Table 25-16.

**Table 25-16. I2C Events**

| Event | Type | Source | Destination | Route | Configuration | Functionality |
|---|---|---|---|---|---|---|
| CPU interrupt | Publisher | I²C | CPU Subsystem | Static route | CPU_INT registers | Fixed interrupt route from I²C to CPU |
| DMA trigger | Publisher | I²C | DMA | DMA event route | DMA_TRIG1 registers | Fixed interrupt route from I²C to DMA |
| DMA trigger | Publisher | I²C | DMA | DMA event route | DMA_TRIG0 registers | Fixed interrupt route from I²C to DMA |

#### 25.2.7.1 CPU Interrupt Event Publisher (CPU_INT)

The I²C module provides 24 interrupt sources which can be configured to source a CPU interrupt event. In order of decreasing interrupt priority, the CPU interrupt events from the I²C are:



Copyright © 2026 Texas Instruments Incorporated

**Table 25-17. I²C CPU Interrupt Event Conditions for Controller (CPU_INT)**

| IIDX STAT | Name | Description |
|---|---|---|
| 0x01 | MRXDONE | Controller receive transaction completed interrupt |
| 0x02 | MTXDONE | Controller transmit transaction completed interrupt |
| 0x03 | MRXFIFOTRG | Controller receive FIFO trigger. Trigger when RX FIFO contains >= defined bytes |
| 0x04 | MTXFIFOTRG | Controller transmit FIFO trigger. Trigger when Transmit FIFO contains <= defined bytes |
| 0x05 | MRXFIFOFULL | Controller RXFIFO full event. This interrupt is set if an RX FIFO is full. |
| 0x06 | MTXEMPTY | Controller transmit FIFO empty interrupt. This interrupt is set if all data in the Transmit FIFO have been shifted out and the transmit goes into idle mode. |
| 0x07 | MCLKTO | Controller clock timeout interrupt |
| 0x08 | MNACK | Address/Data NACK interrupt |
| 0x09 | MSTART | Controller START detection interrupt |
| 0x0A | MSTOP | Controller STOP detection interrupt |
| 0x0B | MARBLOST | Controller arbitration lost interrupt |
| 0x0C | MDMA_DONE_TX | Controller DMA TX done signal (see next section for more detail) |
| 0x0D | MDMA_DONE_RX | Controller DMA RX done signal (see next section for more detail) |

**Table 25-18. I²C CPU Interrupt Event Conditions for Target (CPU_INT)**

| IIDX STAT | Name | Description |
|---|---|---|
| 0x11 | SRXDONE | Target receive transaction completed interrupt |
| 0x12 | STXDONE | Target transmit transaction completed interrupt |
| 0x13 | SRXFIFOTRG | Target receive FIFO trigger. It will trigger when receive FIFO contains >= defined bytes |
| 0x14 | STXFIFOTRG | Target transmit FIFO trigger. It will trigger when transmit FIFO contains <= defined bytes |
| 0x15 | SXFIFOFULL | Target RXFIFO full event. This interrupt is set if an RX FIFO is full. |
| 0x16 | STXEMPTY | Target transmit FIFO empty interrupt. This interrupt is set if all data in the Target Transmit FIFO have been shifted out and the transmit goes into idle mode. |
| 0x17 | SSTART | Target START detection interrupt |
| 0x18 | SSTOP | Target STOP detection interrupt |
| 0x19 | SGENCALL | General call interrupt |
| 0x1A | SDMA_DONE_TX | Target DMA TX done signal (see next section for more detail) |
| 0x1B | SDMA_DONE_RX | Target DMA RX done signal (see next section for more detail) |

The CPU interrupt event configuration is managed with the CPU_INT event management registers. See Section 7.2.5 for guidance on configuring the Event registers for CPU interrupts.

#### 25.2.7.2 DMA Trigger Publisher (DMA_TRIG1, DMA_TRIG0)

DMA_TRIG1 and DMA_TRIG0 registers are used to setup the trigger signaling for the DMA. This can be setup in a flexible way to trigger the DMA for Controller or Target and receive or transmit events with the following four trigger conditions:

**Table 25-19. I²C DMA Trigger Condition (DMA_TRIG1 and DMA_TRIG0)**

| IIDX STAT | Name | Description |
|---|---|---|
| 0x01 | MRXFIFOTRG | Controller receive FIFO trigger. Trigger when RX FIFO contains >= defined bytes |
| 0x02 | MTXFIFOTRG | Controller transmit FIFO trigger. Trigger when Transmit FIFO contains <= defined bytes |
| 0x03 | SRXFIFOTRG | Target receive FIFO trigger. Trigger when RX FIFO contains >= defined bytes |
| 0x04 | STXFIFOTRG | Target transmit FIFO trigger. Trigger when Transmit FIFO contains <= defined bytes |



Copyright © 2026 Texas Instruments Incorporated

The DMA trigger event configuration is managed with the DMA_TRIG1 and DMA_TRIG0 event management registers. See Section 7.2.5 for guidance on configuring the Event registers and Section 7.1.3.2 for on how DMA trigger event works. DMA_TRIG1 and DMA_TRIG0 are two event management registers that correspond to two DMA channels.

As shown in Figure 25-20, each DMA channel can be triggered by any of the conditions listed in Table 25-19 and it can generate either the controller DMA done signal or target DMA done signal.

For example, the user can configure the DMA_TRIG1 trigger using MTXFIFOTRG and the DMA_TRIG0 trigger using SRXFIFOTRG. When the Channel 1 DMA status changes to done, the MDMA_DONE_TX and MDMA_DONE_RX interrupts will set, and when the Channel 2 DMA status change to done, the SDMA_DONE_TX and SDMA_DONE_RX interrupts will set.

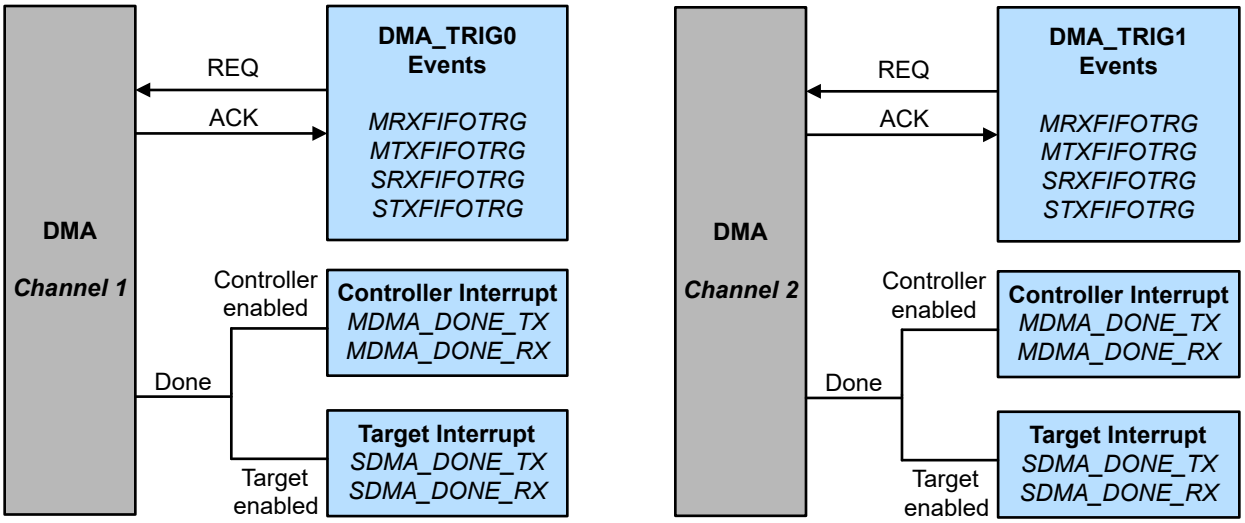



**Figure 25-20. I²C DMA Trigger and Status**

### *25.2.8 Emulation Modes*

The module behavior while the device is in debug mode is controlled by the FREE and SOFT bits in PDBGCTL register.

When the device is in debug mode and set into halt mode below behavior can be configured.

**Table 25-20. Debug Mode Peripheral Behavior**

| PDBGCTL.FREE | PDBGCTL.SOFT | Function |
|---|---|---|
| 1 | x | Modules continues operation |
| 0 | 0 | Module stops immediately |
| 0 | 1 | Module stops after the next transfer has been finished |



Copyright © 2026 Texas Instruments Incorporated

## 25.3 I2C Registers

Table 25-21 lists the memory-mapped registers for the I2C registers. All register offset addresses not listed in Table 25-21 should be considered as reserved locations and the register contents should not be modified.

**Table 25-21. I2C Registers**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 800h | PWREN | Power enable | Section 25.3.1 |
| 804h | RSTCTL | Reset Control | Section 25.3.2 |
| 808h | CLKCFG | Peripheral Clock Configuration Register | Section 25.3.3 |
| 814h | STAT | Status Register | Section 25.3.4 |
| 1000h | CLKDIV | Clock Divider | Section 25.3.5 |
| 1004h | CLKSEL | Clock Select for Ultra Low Power peripherals | Section 25.3.6 |
| 1018h | PDBGCTL | Peripheral Debug Control | Section 25.3.7 |
| 1020h | IIDX | Interrupt index | Section 25.3.8 |
| 1028h | IMASK | Interrupt mask | Section 25.3.9 |
| 1030h | RIS | Raw interrupt status | Section 25.3.10 |
| 1038h | MIS | Masked interrupt status | Section 25.3.11 |
| 1040h | ISET | Interrupt set | Section 25.3.12 |
| 1048h | ICLR | Interrupt clear | Section 25.3.13 |
| 1050h | IIDX | Interrupt index | Section 25.3.14 |
| 1058h | IMASK | Interrupt mask | Section 25.3.15 |
| 1060h | RIS | Raw interrupt status | Section 25.3.16 |
| 1068h | MIS | Masked interrupt status | Section 25.3.17 |
| 1070h | ISET | Interrupt set | Section 25.3.18 |
| 1078h | ICLR | Interrupt clear | Section 25.3.19 |
| 1080h | IIDX | Interrupt index | Section 25.3.20 |
| 1088h | IMASK | Interrupt mask | Section 25.3.21 |
| 1090h | RIS | Raw interrupt status | Section 25.3.22 |
| 1098h | MIS | Masked interrupt status | Section 25.3.23 |
| 10A0h | ISET | Interrupt set | Section 25.3.24 |
| 10A8h | ICLR | Interrupt clear | Section 25.3.25 |
| 10E0h | EVT_MODE | Event Mode | Section 25.3.26 |
| 10E4h | INTCTL | Interrupt control register | Section 25.3.27 |
| 10FCh | DESC | Module Description | Section 25.3.28 |
| 1200h | GFCTL | I2C Glitch Filter Control | Section 25.3.29 |
| 1204h | TIMEOUT_CTL | I2C Timeout Count Control Register | Section 25.3.30 |
| 1208h | TIMEOUT_CNT | I2C Timeout Count Register | Section 25.3.31 |
| 1210h | CSA | I2C Controller Target Address Register | Section 25.3.32 |
| 1214h | CCTR | I2C Controller Control Register | Section 25.3.33 |
| 1218h | CSR | I2C Controller Status Register | Section 25.3.34 |
| 121Ch | CRXDATA | I2C Controller RXData | Section 25.3.35 |
| 1220h | CTXDATA | I2C Controller TXData | Section 25.3.36 |
| 1224h | CTPR | I2C Controller Timer Period | Section 25.3.37 |
| 1228h | CCR | I2C Controller Configuration | Section 25.3.38 |
| 1234h | CBMON | I2C Controller Bus Monitor | Section 25.3.39 |
| 1238h | CFIFOCTL | I2C Controller FIFO Control | Section 25.3.40 |
| 123Ch | CFIFOSR | I2C Controller FIFO Status Register | Section 25.3.41 |



Copyright © 2026 Texas Instruments Incorporated

**Table 25-21. I2C Registers (continued)**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 1240h | _I2CPECCTL | I2C Controller PEC control register | Section 25.3.42 |
| 1244h | _PECSR | I2C Controller PEC status register | Section 25.3.43 |
| 1250h | TOAR | I2C Target Own Address | Section 25.3.44 |
| 1254h | TOAR2 | I2C Target Own Address 2 | Section 25.3.45 |
| 1258h | TCTR | I2C Target Control Register | Section 25.3.46 |
| 125Ch | TSR | I2C Target Status Register | Section 25.3.47 |
| 1260h | TRXDATA | I2C Target RXData | Section 25.3.48 |
| 1264h | TTXDATA | I2C Target TXData | Section 25.3.49 |
| 1268h | TACKCTL | I2C Target ACK Control | Section 25.3.50 |
| 126Ch | TFIFOCTL | I2C Target FIFO Control | Section 25.3.51 |
| 1270h | TFIFOSR | I2C Target FIFO Status Register | Section 25.3.52 |
| 1274h | _PECCTL | I2C Target PEC control register | Section 25.3.53 |
| 1278h | _PECSR | I2C Target PEC status register | Section 25.3.54 |

Complex bit access types are encoded to fit into small table cells. Table 25-22 shows the codes that are used for access types in this section.

**Table 25-22. I2C Access Type Codes**

| Access Type | Code | Description |
|---|---|---|
| Read Type | | |
| R | R | Read |
| Write Type | | |
| W | W | Write |
| WK | W<br>K | Write<br>Write protected by a key |
| Reset or Default Value | | |
| *-n* | | Value after reset or the default value |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.1 PWREN Register (Offset = 800h) [Reset = 00000000h]

PWREN is shown in Table 25-23.

Return to the Summary Table.

Register to control the power state

**Table 25-23. PWREN Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | KEY to allow Power State Change<br>26h = KEY to allow write access to this register |
| 23-1 | RESERVED | R | 0h | |
| 0 | ENABLE | R/WK | 0h | Enable the power<br>KEY must be set to 26h to write to this bit.<br>0h = Disable Power<br>1h = Enable Power |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.2 RSTCTL Register (Offset = 804h) [Reset = 00000000h]

RSTCTL is shown in Table 25-24.

Return to the Summary Table.

Register to control reset assertion and de-assertion

**Table 25-24. RSTCTL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | Unlock key<br>B1h = KEY to allow write access to this register |
| 23-2 | RESERVED | R | 0h | |
| 1 | RESETSTKYCLR | WK | 0h | Clear the RESETSTKY bit in the STAT register<br>KEY must be set to B1h to write to this bit.<br>0h = Writing 0 has no effect<br>1h = Clear reset sticky bit |
| 0 | RESETASSERT | WK | 0h | Assert reset to the peripheral<br>KEY must be set to B1h to write to this bit.<br>0h = Writing 0 has no effect<br>1h = Assert reset |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.3 CLKCFG Register (Offset = 808h) [Reset = 00000000h]

CLKCFG is shown in Table 25-25.

Return to the Summary Table.

Peripheral Clock Configuration Register

**Table 25-25. CLKCFG Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | KEY to Allow State Change -- 0xA9<br>A9h = key value to allow change field of GPRCM |
| 23-9 | RESERVED | R | 0h | |
| 8 | BLOCKASYNC | R/W | 0h | Async Clock Request is blocked from starting SYSOSC or forcing bus clock to 32MHz<br>0h = Not block async clock request<br>1h = Block async clock request |
| 7-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.4 STAT Register (Offset = 814h) [Reset = 00000000h]

STAT is shown in Table 25-26.

Return to the Summary Table.

peripheral enable and reset status

**Table 25-26. STAT Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-17 | RESERVED | R | 0h | |
| 16 | RESETSTKY | R | 0h | This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register<br>0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register<br>1h = The peripheral was reset since the last bit clear |
| 15-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.5 CLKDIV Register (Offset = 1000h) [Reset = 00000000h]

CLKDIV is shown in Table 25-27.

Return to the Summary Table.

This register is used to specify module-specific divide ratio of the functional clock

**Table 25-27. CLKDIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | RESERVED | R | 0h | |
| 2-0 | RATIO | R/W | 0h | Selects divide ratio of module clock<br>0h = Do not divide clock source<br>1h = Divide clock source by 2<br>2h = Divide clock source by 3<br>3h = Divide clock source by 4<br>4h = Divide clock source by 5<br>5h = Divide clock source by 6<br>6h = Divide clock source by 7<br>7h = Divide clock source by 8 |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.6 CLKSEL Register (Offset = 1004h) [Reset = 00000000h]

CLKSEL is shown in Table 25-28.

Return to the Summary Table.

Clock source selection.

**Table 25-28. CLKSEL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | BUSCLK_SEL | R/W | 0h | Selects BUSCLK as clock source if enabled<br>0h = Does not select this clock as a source<br>1h = Select this clock as a source |
| 2 | MFCLK_SEL | R/W | 0h | Selects MFCLK as clock source if enabled<br>0h = Does not select this clock as a source<br>1h = Select this clock as a source |
| 1-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.7 PDBGCTL Register (Offset = 1018h) [Reset = 00000000h]

PDBGCTL is shown in Table 25-29.

Return to the Summary Table.

This register can be used by the software developer to control the behavior of the peripheral relative to the 'Core Halted' input

**Table 25-29. PDBGCTL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | |
| 1 | SOFT | R/W | 1h | Soft halt boundary control. This function is only available, if FREE is set to 'STOP'<br>0h = The peripheral will halt immediately, even if the resultant state will result in corruption if the system is restarted<br>1h = The peripheral blocks the debug freeze until it has reached a boundary where it can resume without corruption |
| 0 | FREE | R/W | 1h | Free run control<br>0h = The peripheral freezes functionality while the Core Halted input is asserted and resumes when it is deasserted.<br>1h = The peripheral ignores the state of the Core Halted input |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.8 IIDX Register (Offset = 1020h) [Reset = 00000000h] (CPU_INT)

IIDX is shown in Table 25-30.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index.

**Table 25-30. IIDX Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | STAT | R | 0h | I2C Module Interrupt Vector Value. This register provides the highes priority interrupt index. A read clears the corresponding interrupt flag in RIS and MISC. 15h-1Fh = Reserved<br>00h = No interrupt pending<br>01h = Controller data received<br>02h = Controller data transmitted<br>03h = Controller receive FIFO Trigger Level<br>04h = Controller transmit FIFO Trigger level<br>5h = RX FIFO FULL Event/interrupt pending<br>6h = Transmit FIFO/Buffer Empty Event/interrupt pending<br>08h = Address/Data NACK<br>09h = Start Event<br>0Ah = Stop Event<br>0Bh = Arbitration Lost<br>Ch = DMA DONE on Channel TX<br>Dh = DMA DONE on Channel RX<br>Eh = Controller PEC Receive Error Event<br>Fh = Timeout A Event<br>10h = Timeout B Event<br>11h = Target Data Event<br>12h = Target Data Event<br>13h = Target receive FIFO Trigger Level<br>14h = Target transmit FIFO Trigger level<br>15h = RX FIFO FULL Event/interrupt pending<br>16h = Transmit FIFO/Buffer Empty Event/interrupt pending<br>17h = Start Event<br>18h = Stop Event<br>19h = General Call Event<br>1Ah = DMA DONE on Channel TX<br>1Bh = DMA DONE on Channel RX<br>1Ch = Target PEC receive error event<br>1Dh = Target TX FIFO underflow<br>1Eh = Target RX FIFO overflow event<br>1Fh = Target arbitration lost event<br>20h = Interrupt overflow event |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.9 IMASK Register (Offset = 1028h) [Reset = 00000000h] (CPU_INT)

IMASK is shown in Table 25-31.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

**Table 25-31. IMASK Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INTR_OVFL | R/W | 0h | Interrupt Overflow Interrupt Mask<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 30 | TARBLOST | R/W | 0h | Target Arbitration Lost<br>0h = Clear Set Interrupt Mask<br>1h = Set Interrupt Mask |
| 29 | TRX_OVFL | R/W | 0h | Target RX FIFO overflow<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 28 | TTX_UNFL | R/W | 0h | Target TX FIFO underflow<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 27 | TPEC_RX_ERR | R/W | 0h | Target RX Pec Error Interrupt<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 26 | TDMA_DONE_RX | R/W | 0h | Target DMA Done on Event Channel RX<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 25 | TDMA_DONE_TX | R/W | 0h | Target DMA Done on Event Channel TX<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 24 | TGENCALL | R/W | 0h | General Call Interrupt<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 23 | TSTOP | R/W | 0h | Stop Condition Interrupt<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 22 | TSTART | R/W | 0h | Start Condition Interrupt<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 21 | TTXEMPTY | R/W | 0h | Target Transmit FIFO Empty interrupt mask. This interrupt is set if all data in the Transmit FIFO have been shifted out and the transmit goes into idle mode.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 20 | TRXFIFOFULL | R/W | 0h | RXFIFO full event. This interrupt is set if an Target RX FIFO is full.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 19 | TTXFIFOTRG | R/W | 0h | Target Transmit FIFO Trigger<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 18 | TRXFIFOTRG | R/W | 0h | Target Receive FIFO Trigger<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 17 | TTXDONE | R/W | 0h | Target Transmit Transaction completed Interrupt<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |



Copyright © 2026 Texas Instruments Incorporated

**Table 25-31. IMASK Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 16 | TRXDONE | R/W | 0h | Target Receive Data Interrupt<br>Signals that a byte has been received<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 15 | TIMEOUTB | R/W | 0h | Timeout B Interrupt<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 14 | TIMEOUTA | R/W | 0h | Timeout A Interrupt<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 13 | CPEC_RX_ERR | R/W | 0h | Controller RX Pec Error Interrupt<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 12 | CDMA_DONE_RX | R/W | 0h | DMA Done on Event Channel RX<br>0h = Interrupt disabled<br>1h = Set Interrupt Mask |
| 11 | CDMA_DONE_TX | R/W | 0h | DMA Done on Event Channel TX<br>0h = Interrupt disabled<br>1h = Set Interrupt Mask |
| 10 | CARBLOST | R/W | 0h | Arbitration Lost Interrupt<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 9 | CSTOP | R/W | 0h | STOP Detection Interrupt<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 8 | CSTART | R/W | 0h | START Detection Interrupt<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 7 | CNACK | R/W | 0h | Address/Data NACK Interrupt<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 6 | RESERVED | R | 0h | |
| 5 | CTXEMPTY | R/W | 0h | Transmit FIFO Empty interrupt mask. This interrupt is set if all data in the Transmit FIFO have been shifted out and the transmit goes into idle mode.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 4 | CRXFIFOFULL | R/W | 0h | RXFIFO full event. This interrupt is set if an RX FIFO is full.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 3 | CTXFIFOTRG | R/W | 0h | Controller Transmit FIFO Trigger<br>Trigger when Transmit FIFO contains <= defined bytes<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 2 | CRXFIFOTRG | R/W | 0h | Controller Receive FIFO Trigger<br>Trigger when RX FIFO contains >= defined bytes<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 1 | CTXDONE | R/W | 0h | Controller Transmit Transaction completed Interrupt<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 0 | CRXDONE | R/W | 0h | Controller Receive Transaction completed Interrupt<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.10 RIS Register (Offset = 1030h) [Reset = 00000000h] (CPU_INT)

RIS is shown in Table 25-32.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Table 25-32. RIS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INTR_OVFL | R | 0h | Interrupt overflow interrupt<br>It is set when CSTART or CSTOP interrupts overflow i.e. occur twice without being serviced<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 30 | TARBLOST | R | 0 h | Target Arbitration Lost<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 29 | TRX_OVFL | R | 0h | Target RX FIFO overflow<br>0h = Interrupt did not occur<br>1h = Interrupt Occured |
| 28 | TTX_UNFL | R | 0h | Target TX FIFO underflow<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 27 | TPEC_RX_ERR | R | 0h | Target RX Pec Error Interrupt<br>0h = Interrupt did not occur<br>1h = Interrupt ocuured |
| 26 | TDMA_DONE_RX | R | 0h | DMA Done on Event Channel RX<br>0h = Clear interrupt<br>1h = Set interrupt |
| 25 | TDMA_DONE_TX | R | 0h | DMA Done on Event Channel TX<br>0h = Clear interrupt<br>1h = Set interrupt |
| 24 | TGENCALL | R | 0h | General Call Interrupt<br>0h = Clear Interrupt Mask<br>1h = Interrupt occurred |
| 23 | TSTOP | R | 0h | Stop Condition Interrupt<br>0h = Clear Interrupt<br>1h = Set interrupt |
| 22 | TSTART | R | 0h | Start Condition Interrupt<br>0h = Clear interrupt<br>1h = Set Interrupt |
| 21 | TTXEMPTY | R | 0h | Transmit FIFO Empty interrupt mask. This interrupt is set if all data in the Transmit FIFO have been shifted out and the transmit goes into idle mode.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 20 | TRXFIFOFULL | R | 0h | RXFIFO full event. This interrupt is set if an RX FIFO is full.<br>0h = Clear Interrupt Mask<br>1h = Interrupt occurred |
| 19 | TTXFIFOTRG | R | 0h | Target Transmit FIFO Trigger<br>0h = Clear Interrupt Mask<br>1h = Interrupt occurred |
| 18 | TRXFIFOTRG | R | 0h | Target Receive FIFO Trigger<br>0h = Clear Interrupt Mask<br>1h = Interrupt occurred |
| 17 | TTXDONE | R | 0h | Target Transmit Transaction completed Interrupt<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

**Table 25-32. RIS Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 16 | TRXDONE | R | 0h | Target Receive Data Interrupt<br>Signals that a byte has been received<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 15 | TIMEOUTB | R | 0h | Timeout B Interrupt<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 14 | TIMEOUTA | R | 0h | Timeout A Interrupt<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 13 | CPEC_RX_ERR | R | 0h | Controller RX Pec Error Interrupt<br>0h = Interrupt did not occur<br>1h = Interrupt Occured |
| 12 | CDMA_DONE_RX | R | 0h | DMA Done on Event Channel RX<br>0h = Interrupt disabled<br>1h = Interrupt occurred |
| 11 | CDMA_DONE_TX | R | 0h | DMA Done on Event Channel TX<br>0h = Interrupt disabled<br>1h = Interrupt occurred |
| 10 | CARBLOST | R | 0h | Arbitration Lost Interrupt<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 9 | CSTOP | R | 0h | STOP Detection Interrupt<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 8 | CSTART | R | 0h | START Detection Interrupt<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 7 | CNACK | R | 0h | Address/Data NACK Interrupt<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 6 | RESERVED | R | 0h | |
| 5 | CTXEMPTY | R | 0h | Transmit FIFO Empty interrupt mask. This interrupt is set if all data in the Transmit FIFO have been shifted out and the transmit goes into idle mode.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 4 | CRXFIFOFULL | R | 0h | RXFIFO full event. This interrupt is set if an RX FIFO is full.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 3 | CTXFIFOTRG | R | 0h | Controller Transmit FIFO Trigger<br>Trigger when Transmit FIFO contains <= defined bytes<br>0h = Clear Interrupt Mask<br>1h = Interrupt occurred |
| 2 | CRXFIFOTRG | R | 0h | Controller Receive FIFO Trigger<br>Trigger when RX FIFO contains >= defined bytes<br>0h = Clear Interrupt Mask<br>1h = Interrupt occurred |
| 1 | CTXDONE | R | 0h | Controller Transmit Transaction completed Interrupt<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 0 | CRXDONE | R | 0h | Controller Receive Transaction completed Interrupt<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.11 MIS Register (Offset = 1038h) [Reset = 00000000h] (CPU_INT)

MIS is shown in Table 25-33.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Table 25-33. MIS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INTR_OVFL | R | 0h | Interrupt overflow<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 30 | TARBLOST | R | 0h | Target Arbitration Lost<br>0h = Clear interrupt mask<br>1h = Set interrupt mask |
| 29 | TRX_OVFL | R | 0h | Target RX FIFO overflow<br>0h = Clear interrupt mask<br>1h = Set interrupt mask |
| 28 | TTX_UNFL | R | 0h | Target TX FIFO underflow<br>0h = Clear interrupt mask<br>1h = Set interrupt mask |
| 27 | TPEC_RX_ERR | R | 0h | Target RX Pec Error Interrupt<br>0h = Clear interrupt mask<br>1h = Set interrupt mask |
| 26 | TDMA_DONE_RX | R | 0h | DMA Done on Event Channel RX<br>0h = Clear MIS<br>1h = Set MIS |
| 25 | TDMA_DONE_TX | R | 0h | DMA Done on Event Channel TX<br>0h = Clear MIS<br>1h = Set MIS |
| 24 | TGENCALL | R | 0h | General Call Interrupt<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 23 | TSTOP | R | 0h | Target STOP Detection Interrupt<br>0h = Clear MIS<br>1h = Set MIS |
| 22 | TSTART | R | 0h | Target START Detection Interrupt<br>0h = Clear MIS<br>1h = Set MIS |
| 21 | TTXEMPTY | R | 0h | Transmit FIFO Empty interrupt mask. This interrupt is set if all data in the Transmit FIFO have been shifted out and the transmit goes into idle mode.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 20 | TRXFIFOFULL | R | 0h | RXFIFO full event. This interrupt is set if an RX FIFO is full.<br>0h = Clear Interrupt Mask<br>1h = Interrupt occurred |
| 19 | TTXFIFOTRG | R | 0h | Target Transmit FIFO Trigger<br>0h = Clear Interrupt Mask<br>1h = Interrupt occurred |
| 18 | TRXFIFOTRG | R | 0h | Target Receive FIFO Trigger<br>0h = Clear Interrupt Mask<br>1h = Interrupt occurred |
| 17 | TTXDONE | R | 0h | Target Transmit Transaction completed Interrupt<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 16 | TRXDONE | R | 0h | Target Receive Data Interrupt<br>Signals that a byte has been received<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

**Table 25-33. MIS Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15 | TIMEOUTB | R | 0h | Timeout B Interrupt<br>0h = Clear interrupt mask<br>1h = Set interrupt mask |
| 14 | TIMEOUTA | R | 0h | Timeout A Interrupt<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 13 | CPEC_RX_ERR | R | 0h | Controller RX Pec Error Interrupt<br>0h = Clear interrupt mask<br>1h = Set interrupt mask |
| 12 | CDMA_DONE_RX | R | 0h | DMA Done on Event Channel RX<br>0h = Interrupt disabled<br>1h = Interrupt occurred |
| 11 | CDMA_DONE_TX | R | 0h | DMA Done on Event Channel TX<br>0h = Interrupt disabled<br>1h = Interrupt occurred |
| 10 | CARBLOST | R | 0h | Arbitration Lost Interrupt<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 9 | CSTOP | R | 0h | STOP Detection Interrupt<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 8 | CSTART | R | 0h | START Detection Interrupt<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 7 | CNACK | R | 0h | Address/Data NACK Interrupt<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 6 | RESERVED | R | 0h | |
| 5 | CTXEMPTY | R | 0h | Transmit FIFO Empty interrupt mask. This interrupt is set if all data in the Transmit FIFO have been shifted out and the transmit goes into idle mode.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 4 | CRXFIFOFULL | R | 0h | RXFIFO full event. This interrupt is set if the RX FIFO is full.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 3 | CTXFIFOTRG | R | 0h | Controller Transmit FIFO Trigger<br>Trigger when Transmit FIFO contains <= defined bytes<br>0h = Clear Interrupt Mask<br>1h = Interrupt occurred |
| 2 | CRXFIFOTRG | R | 0h | Controller Receive FIFO Trigger<br>Trigger when RX FIFO contains >= defined bytes<br>0h = Clear Interrupt Mask<br>1h = Interrupt occurred |
| 1 | CTXDONE | R | 0h | Controller Transmit Transaction completed Interrupt<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 0 | CRXDONE | R | 0h | Controller Receive Data Interrupt<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.12 ISET Register (Offset = 1040h) [Reset = 00000000h] (CPU_INT)

ISET is shown in Table 25-34.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Table 25-34. ISET Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INTR_OVFL | W | 0h | Interrupt overflow<br>0h = No effect<br>1h = Set interrupt |
| 30 | TARBLOST | W | 0h | Target Arbitration Lost<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 29 | TRX_OVFL | W | 0h | Target RX FIFO overflow<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 28 | TTX_UNFL | W | 0h | Target TX FIFO underflow<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 27 | TPEC_RX_ERR | W | 0h | Target RX Pec Error Interrupt<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 26 | TDMA_DONE_RX | W | 0h | DMA Done on Event Channel RX<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 25 | TDMA_DONE_TX | W | 0h | DMA Done on Event Channel TX<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 24 | TGENCALL | W | 0h | General Call Interrupt<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 23 | TSTOP | W | 0h | Stop Condition Interrupt<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 22 | TSTART | W | 0h | Start Condition Interrupt<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 21 | TTXEMPTY | W | 0h | Transmit FIFO Empty interrupt mask. This interrupt is set if all data in the Transmit FIFO have been shifted out and the transmit goes into idle mode.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 20 | TRXFIFOFULL | W | 0h | RXFIFO full event. This interrupt is set if an RX FIFO is full.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt |
| 19 | TTXFIFOTRG | W | 0h | Target Transmit FIFO Trigger<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt |
| 18 | TRXFIFOTRG | W | 0h | Target Receive FIFO Trigger<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt |
| 17 | TTXDONE | W | 0h | Target Transmit Transaction completed Interrupt<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |



Copyright © 2026 Texas Instruments Incorporated

**Table 25-34. ISET Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 16 | TRXDONE | W | 0h | Target Receive Data Interrupt<br>Signals that a byte has been received<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 15 | TIMEOUTB | W | 0h | Timeout B Interrupt<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 14 | TIMEOUTA | W | 0h | Timeout A interrupt<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 13 | CPEC_RX_ERR | W | 0h | Controller RX Pec Error Interrupt<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 12 | CDMA_DONE_RX | W | 0h | DMA Done on Event Channel RX<br>0h = Interrupt disabled<br>1h = Set Interrupt |
| 11 | CDMA_DONE_TX | W | 0h | DMA Done on Event Channel TX<br>0h = Interrupt disabled<br>1h = Set Interrupt |
| 10 | CARBLOST | W | 0h | Arbitration Lost Interrupt<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 9 | CSTOP | W | 0h | STOP Detection Interrupt<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 8 | CSTART | W | 0h | START Detection Interrupt<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 7 | CNACK | W | 0h | Address/Data NACK Interrupt<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 6 | RESERVED | R | 0h | |
| 5 | CTXEMPTY | W | 0h | Transmit FIFO Empty interrupt mask. This interrupt is set if all data in the Transmit FIFO have been shifted out and the transmit goes into idle mode.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 4 | CRXFIFOFULL | W | 0h | RXFIFO full event.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 3 | CTXFIFOTRG | W | 0h | Controller Transmit FIFO Trigger<br>Trigger when Transmit FIFO contains <= defined bytes<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt |
| 2 | CRXFIFOTRG | W | 0h | Controller Receive FIFO Trigger<br>Trigger when RX FIFO contains >= defined bytes<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt |
| 1 | CTXDONE | W | 0h | Controller Transmit Transaction completed Interrupt<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 0 | CRXDONE | W | 0h | Controller Receive Data Interrupt<br>Signals that a byte has been received<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.13 ICLR Register (Offset = 1048h) [Reset = 00000000h] (CPU_INT)

ICLR is shown in Table 25-35.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

**Table 25-35. ICLR Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | INTR_OVFL | W | 0h | Interrupt overflow<br>0h = No effect<br>1h = Clear interrupt |
| 30 | TARBLOST | W | 0h | Target Arbitration Lost<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 29 | TRX_OVFL | W | 0h | Target RX FIFO overflow<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 28 | TTX_UNFL | W | 0h | Target TX FIFO underflow<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 27 | TPEC_RX_ERR | W | 0h | Target RX Pec Error Interrupt<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 26 | TDMA_DONE_RX | W | 0h | DMA Done on Event Channel RX<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 25 | TDMA_DONE_TX | W | 0h | DMA Done on Event Channel TX<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 24 | TGENCALL | W | 0h | General Call Interrupt<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 23 | TSTOP | W | 0h | Target STOP Detection Interrupt<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 22 | TSTART | W | 0h | Target START Detection Interrupt<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 21 | TTXEMPTY | W | 0h | Transmit FIFO Empty interrupt mask. This interrupt is set if all data in the Transmit FIFO have been shifted out and the transmit goes into idle mode.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 20 | TRXFIFOFULL | W | 0h | RXFIFO full event. This interrupt is set if an RX FIFO is full.<br>0h = Clear Interrupt Mask<br>1h = Clear Interrupt |
| 19 | TTXFIFOTRG | W | 0h | Target Transmit FIFO Trigger<br>0h = Clear Interrupt Mask<br>1h = Clear Interrupt |
| 18 | TRXFIFOTRG | W | 0h | Target Receive FIFO Trigger<br>0h = Clear Interrupt Mask<br>1h = Clear Interrupt |
| 17 | TTXDONE | W | 0h | Target Transmit Transaction completed Interrupt<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 16 | TRXDONE | W | 0h | Target Receive Data Interrupt<br>Signals that a byte has been received<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |



Copyright © 2026 Texas Instruments Incorporated

**Table 25-35. ICLR Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15 | TIMEOUTB | W | 0h | Timeout B Interrupt<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 14 | TIMEOUTA | W | 0h | Timeout A interrupt<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 13 | CPEC_RX_ERR | W | 0h | Controller RX Pec Error Interrupt<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 12 | CDMA_DONE_RX | W | 0h | DMA Done on Event Channel RX<br>0h = Interrupt disabled<br>1h = Clear Interrupt |
| 11 | CDMA_DONE_TX | W | 0h | DMA Done on Event Channel TX<br>0h = Interrupt disabled<br>1h = Clear Interrupt |
| 10 | CARBLOST | W | 0h | Arbitration Lost Interrupt<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 9 | CSTOP | W | 0h | STOP Detection Interrupt<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 8 | CSTART | W | 0h | START Detection Interrupt<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 7 | CNACK | W | 0h | Address/Data NACK Interrupt<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 6 | RESERVED | R | 0h | |
| 5 | CTXEMPTY | W | 0h | Transmit FIFO Empty interrupt mask. This interrupt is set if all data in the Transmit FIFO have been shifted out and the transmit goes into idle mode.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 4 | CRXFIFOFULL | W | 0h | RXFIFO full event.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 3 | CTXFIFOTRG | W | 0h | Controller Transmit FIFO Trigger<br>Trigger when Transmit FIFO contains <= defined bytes<br>0h = Clear Interrupt Mask<br>1h = Clear Interrupt |
| 2 | CRXFIFOTRG | W | 0h | Controller Receive FIFO Trigger<br>Trigger when RX FIFO contains >= defined bytes<br>0h = Clear Interrupt Mask<br>1h = Clear Interrupt |
| 1 | CTXDONE | W | 0h | Controller Transmit Transaction completed Interrupt<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 0 | CRXDONE | W | 0h | Controller Receive Data Interrupt<br>Signals that a byte has been received<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.14 IIDX Register (Offset = 1050h) [Reset = 00000000h] (DMA_TRIG1)

IIDX is shown in Table 25-36.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index.

**Table 25-36. IIDX Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | STAT | R | 0h | I2C Module Interrupt Vector Value. This register provides the highes priority interrupt index. A read clears the corresponding interrupt flag in RIS and MISC. 15h-1Fh = Reserved<br>00h = No interrupt pending<br>01h = Controller receive FIFO Trigger Level<br>02h = Controller transmit FIFO Trigger level<br>03h = Target receive FIFO Trigger Level<br>04h = Target transmit FIFO Trigger level |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.15 IMASK Register (Offset = 1058h) [Reset = 00000000h] (DMA_TRIG1)

IMASK is shown in Table 25-37.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

**Table 25-37. IMASK Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | TTXFIFOTRG | R/W | 0h | Target Transmit FIFO Trigger<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 2 | TRXFIFOTRG | R/W | 0h | Target Receive FIFO Trigger<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 1 | CTXFIFOTRG | R/W | 0h | Controller Transmit FIFO Trigger<br>Trigger when Transmit FIFO contains <= defined bytes<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 0 | CRXFIFOTRG | R/W | 0h | Controller Receive FIFO Trigger<br>Trigger when RX FIFO contains >= defined bytes<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.16 RIS Register (Offset = 1060h) [Reset = 00000000h] (DMA_TRIG1)

RIS is shown in Table 25-38.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Table 25-38. RIS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | TTXFIFOTRG | R | 0h | Target Transmit FIFO Trigger<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 2 | TRXFIFOTRG | R | 0h | Target Receive FIFO Trigger<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 1 | CTXFIFOTRG | R | 0h | Controller Transmit FIFO Trigger<br>Trigger when Transmit FIFO contains <= defined bytes<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 0 | CRXFIFOTRG | R | 0h | Controller Receive FIFO Trigger<br>Trigger when RX FIFO contains >= defined bytes<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.17 MIS Register (Offset = 1068h) [Reset = 00000000h] (DMA_TRIG1)

MIS is shown in Table 25-39.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Table 25-39. MIS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | TTXFIFOTRG | R | 0h | Target Transmit FIFO Trigger<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 2 | TRXFIFOTRG | R | 0h | Target Receive FIFO Trigger<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 1 | CTXFIFOTRG | R | 0h | Controller Transmit FIFO Trigger<br>Trigger when Transmit FIFO contains <= defined bytes<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 0 | CRXFIFOTRG | R | 0h | Controller Receive FIFO Trigger<br>Trigger when RX FIFO contains >= defined bytes<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.18 ISET Register (Offset = 1070h) [Reset = 00000000h] (DMA_TRIG1)

ISET is shown in Table 25-40.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Table 25-40. ISET Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | TTXFIFOTRG | W | 0h | Target Transmit FIFO Trigger<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 2 | TRXFIFOTRG | W | 0h | Target Receive FIFO Trigger<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 1 | CTXFIFOTRG | W | 0h | Controller Transmit FIFO Trigger<br>Trigger when Transmit FIFO contains <= defined bytes<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 0 | CRXFIFOTRG | W | 0h | Controller Receive FIFO Trigger<br>Trigger when RX FIFO contains >= defined bytes<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.19 ICLR Register (Offset = 1078h) [Reset = 00000000h] (DMA_TRIG1)

ICLR is shown in Table 25-41.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

**Table 25-41. ICLR Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | TTXFIFOTRG | W | 0h | Target Transmit FIFO Trigger<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 2 | TRXFIFOTRG | W | 0h | Target Receive FIFO Trigger<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 1 | CTXFIFOTRG | W | 0h | Controller Transmit FIFO Trigger<br>Trigger when Transmit FIFO contains <= defined bytes<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 0 | CRXFIFOTRG | W | 0h | Controller Receive FIFO Trigger<br>Trigger when RX FIFO contains >= defined bytes<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.20 IIDX Register (Offset = 1080h) [Reset = 00000000h] (DMA_TRIG0)

IIDX is shown in Table 25-42.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index.

**Table 25-42. IIDX Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | STAT | R | 0h | I2C Module Interrupt Vector Value. This register provides the highes priority interrupt index. A read clears the corresponding interrupt flag in RIS and MISC. 15h-1Fh = Reserved<br>00h = No interrupt pending<br>01h = Controller receive FIFO Trigger Level<br>02h = Controller transmit FIFO Trigger level<br>03h = Target receive FIFO Trigger Level<br>04h = Target transmit FIFO Trigger level |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.21 IMASK Register (Offset = 1088h) [Reset = 00000000h] (DMA_TRIG0)

IMASK is shown in Table 25-43.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

**Table 25-43. IMASK Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | TTXFIFOTRG | R/W | 0h | Target Transmit FIFO Trigger<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 2 | TRXFIFOTRG | R/W | 0h | Target Receive FIFO Trigger<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 1 | CTXFIFOTRG | R/W | 0h | Controller Transmit FIFO Trigger<br>Trigger when Transmit FIFO contains <= defined bytes<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 0 | CRXFIFOTRG | R/W | 0h | Controller Receive FIFO Trigger<br>Trigger when RX FIFO contains >= defined bytes<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.22 RIS Register (Offset = 1090h) [Reset = 00000000h] (DMA_TRIG0)

RIS is shown in Table 25-44.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Table 25-44. RIS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | TTXFIFOTRG | R | 0h | Target Transmit FIFO Trigger<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 2 | TRXFIFOTRG | R | 0h | Target Receive FIFO Trigger<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 1 | CTXFIFOTRG | R | 0h | Controller Transmit FIFO Trigger<br>Trigger when Transmit FIFO contains <= defined bytes<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 0 | CRXFIFOTRG | R | 0h | Controller Receive FIFO Trigger<br>Trigger when RX FIFO contains >= defined bytes<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.23 MIS Register (Offset = 1098h) [Reset = 00000000h] (DMA_TRIG0)

MIS is shown in Table 25-45.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Table 25-45. MIS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | TTXFIFOTRG | R | 0h | Target Transmit FIFO Trigger<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 2 | TRXFIFOTRG | R | 0h | Target Receive FIFO Trigger<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 1 | CTXFIFOTRG | R | 0h | Controller Transmit FIFO Trigger<br>Trigger when Transmit FIFO contains <= defined bytes<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 0 | CRXFIFOTRG | R | 0h | Controller Receive FIFO Trigger<br>Trigger when RX FIFO contains >= defined bytes<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.24 ISET Register (Offset = 10A0h) [Reset = 00000000h] (DMA_TRIG0)

ISET is shown in Table 25-46.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Table 25-46. ISET Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | TTXFIFOTRG | W | 0h | Target Transmit FIFO Trigger<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 2 | TRXFIFOTRG | W | 0h | Target Receive FIFO Trigger<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 1 | CTXFIFOTRG | W | 0h | Controller Transmit FIFO Trigger<br>Trigger when Transmit FIFO contains <= defined bytes<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 0 | CRXFIFOTRG | W | 0h | Controller Receive FIFO Trigger<br>Trigger when RX FIFO contains >= defined bytes<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.25 ICLR Register (Offset = 10A8h) [Reset = 00000000h] (DMA_TRIG0)

ICLR is shown in Table 25-47.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

**Table 25-47. ICLR Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | TTXFIFOTRG | W | 0h | Target Transmit FIFO Trigger<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 2 | TRXFIFOTRG | W | 0h | Target Receive FIFO Trigger<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 1 | CTXFIFOTRG | W | 0h | Controller Transmit FIFO Trigger<br>Trigger when Transmit FIFO contains <= defined bytes<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 0 | CRXFIFOTRG | W | 0h | Controller Receive FIFO Trigger<br>Trigger when RX FIFO contains >= defined bytes<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.26 EVT_MODE Register (Offset = 10E0h) [Reset = 00000000h]

EVT_MODE is shown in Table 25-48.

Return to the Summary Table.

Event mode register. It is used to select whether each line is disabled, in software mode (software clears the RIS) or in hardware mode (hardware clears the RIS)

**Table 25-48. EVT_MODE Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | RESERVED | R | 0h | |
| 5-4 | EVT2_CFG | R | 2h | Event line mode select for event corresponding to none.DMA_TRIG0<br>0h = The interrupt or event line is disabled.<br>1h = The interrupt or event line is in software mode. Software must clear the RIS.<br>2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag. |
| 3-2 | INT1_CFG | R | 2h | Event line mode select for event corresponding to none.DMA_TRIG1<br>0h = The interrupt or event line is disabled.<br>1h = The interrupt or event line is in software mode. Software must clear the RIS.<br>2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag. |
| 1-0 | INT0_CFG | R | 1h | Event line mode select for event corresponding to none.CPU_INT<br>0h = The interrupt or event line is disabled.<br>1h = The interrupt or event line is in software mode. Software must clear the RIS.<br>2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag. |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.27 INTCTL Register (Offset = 10E4h) [Reset = 00000000h]

INTCTL is shown in Table 25-49.

Return to the Summary Table.

Interrupt control register

**Table 25-49. INTCTL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | INTEVAL | W | 0h | Writing a 1 to this field re-evaluates the interrupt sources.<br>0h = The interrupt or event line is disabled.<br>1h = The interrupt or event line is in software mode. Software must clear the RIS. |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.28 DESC Register (Offset = 10FCh) [Reset = 00000000h]

DESC is shown in Table 25-50.

Return to the Summary Table.

This register identifies the peripheral and its exact version.

**Table 25-50. DESC Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | MODULEID | R | 1511h | Module identification contains a unique peripheral identification number. The assignments are maintained in a central database for all of the platform modules to ensure uniqueness.<br>0h = Smallest value<br>FFFFh = Highest possible value |
| 15-12 | FEATUREVER | R | 0h | Feature Set for the module *instance*<br>0h = Smallest value<br>Fh = Highest possible value |
| 11-8 | INSTNUM | R | 0h | Instance Number within the device. This will be a parameter to the RTL for modules that can have multiple instances<br>0h = Smallest value<br>Fh = Highest possible value |
| 7-4 | MAJREV | R | 1h | Major rev of the IP<br>0h = Smallest value<br>Fh = Highest possible value |
| 3-0 | MINREV | R | 0h | Minor rev of the IP<br>0h = Smallest value<br>Fh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.29 GFCTL Register (Offset = 1200h) [Reset = 00000000h]

GFCTL is shown in Table 25-51.

Return to the Summary Table.

This register controls the glitch filter on the SCL and SDA lines

**Table 25-51. GFCTL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-12 | RESERVED | R | 0h | |
| 11 | CHAIN | R/W | 1h | Analog and digital noise filters chaining enable.<br>0h = When 0, chaining is disabled and only digital filter output is available to IP logic for oversampling<br>1h = When 1, analog and digital glitch filters are chained and the output of the combination is made available to IP logic for oversampling |
| 10-9 | AGFSEL | R/W | 3h | Analog Glitch Suppression Pulse Width<br>This field controls the pulse width select for the analog glitch suppression on SCL and SDA lines.<br>See device datasheet for exact values.<br>(ULP I2C only)<br>0h = Pulses shorter then 5ns length are filtered.<br>1h = Pulses shorter then 10ns length are filtered.<br>2h = Pulses shorter then 25ns length are filtered.<br>3h = Pulses shorter then 50ns length are filtered. |
| 8 | AGFEN | R/W | 1h | Analog Glitch Suppression Enable<br>0h = Analog Glitch Filter disable<br>1h = Analog Glitch Filter enable |
| 7-3 | RESERVED | R | 0h | |
| 2-0 | DGFSEL | R/W | 0h | Glitch Suppression Pulse Width<br>This field controls the pulse width select for glitch suppression on the SCL and SDA lines. The following values are the glitch suppression values in terms of functional clocks.<br>(Core Domain only)<br>0h = Bypass<br>1h = 1 clock<br>2h = 2 clocks<br>3h = 3 clocks<br>4h = 4 clocks<br>5h = 8 clocks<br>6h = 16 clocks<br>7h = 31 clocks |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.30 TIMEOUT_CTL Register (Offset = 1204h) [Reset = 00020002h]

TIMEOUT_CTL is shown in Table 25-52.

Return to the Summary Table.

This register contains controls for Timeout Counters A and B

**Table 25-52. TIMEOUT_CTL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | TCNTBEN | R/W | 0h | Timeout Counter B Enable<br>0h = Disable Timeout Counter B<br>1h = Enable Timeout Counter B |
| 30-24 | RESERVED | R | 0h | |
| 23-16 | TCNTLB | R/W | 2h | Timeout Count B Load: Counter B is used for SCL High Detection. This field contains the upper 8 bits of a 12-bit pre-load value for the Timeout B count. NOTE: The value of CNTLB must be greater than 1h.<br>Each count is equal to 1* clock period. For example, with 10MHz functional clock one timeout period will be equal to1*100ns.<br>0h = Smallest possible value<br>FFh = Highest possible value |
| 15 | TCNTAEN | R/W | 0h | Timeout Counter A Enable<br>0h = Disable Timeout Counter B<br>1h = Enable Timeout Counter B |
| 14-8 | RESERVED | R | 0h | |
| 7-0 | TCNTLA | R/W | 2h | Timeout counter A load value<br>Counter A is used for SCL low detection. This field contains the upper 8 bits of a 12-bit pre-load value for the Timeout A count. NOTE: The value of CNTLA must be greater than 1h.<br>Each count is equal to 520 times the timeout period of functional clock. For example, with 8MHz functional clock and a 100KHz operating I2C clock, one timeout period will be equal to (1 / 8MHz) * 520 or 65 us.<br>0h = Smallest Value<br>FFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.31 TIMEOUT_CNT Register (Offset = 1208h) [Reset = 00020002h]

TIMEOUT_CNT is shown in Table 25-53.

Return to the Summary Table.

This register contains the upper 8 bits of a 12-bit current counter values for counter A and B. The lower four bits of the counter are not user visible and are always 0h.

**Table 25-53. TIMEOUT_CNT Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | RESERVED | R | 0h | |
| 23-16 | TCNTB | R | 2h | Timeout Count B Current Count: This field contains the upper 8 bits of a 12-bit current counter for timeout counter B<br>0h = Smallest Value<br>FFh = Highest possible value |
| 15-8 | RESERVED | R | 0h | |
| 7-0 | TCNTA | R | 2h | Timeout Count A Current Count: This field contains the upper 8 bits of a 12-bit current counter for timeout counter A<br>0h = Smallest Value<br>FFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.32 CSA Register (Offset = 1210h) [Reset = 00000000h]

CSA is shown in Table 25-54.

Return to the Summary Table.

I2C Controller Target Address Register

**Table 25-54. CSA Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15 | CMODE | R/W | 0h | This bit selects the adressing mode to be used in Controller mode<br>When 0, 7-bit addressing is used.<br>When 1, 10-bit addressing is used.<br>0h = 7-bit addressing mode<br>1h = 10-bit addressing mode |
| 14-11 | RESERVED | R | 0h | |
| 10-1 | TADDR | R/W | 0h | I2C Target Address This field specifies bits A9 through A0 of the Target address.<br>In 7-bit addressing mode as selected by MSA.MODE bit, the top 3 bits are don't care<br>0h = Smallest value<br>3FFh = Highest possible value |
| 0 | DIR | R/W | 0h | Receive/Send<br>The DIR bit specifies if the next Controller operation is a Receive (High) or Transmit (Low).<br>0h = Transmit<br>1h = Receive<br>0h = The Controller is in transmit mode.<br>1h = The Controller is in receive mode. |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.33 CCTR Register (Offset = 1214h) [Reset = 00000000h]

CCTR is shown in Table 25-55.

Return to the Summary Table.

This control register configures the I2C controller operation. The START bit generates the START or REPEATED START condition. The STOP bit determines if the cycle stops at the end of the data cycle or continues to the next transfer cycle, which could be a repeated START. To generate a single transmit cycle, the I2C Controller Target Address (MSA) register is written with the desired address, the RS bit is cleared, and this register is written with ACK = X (0 or 1), STOP = 1, START = 1, and RUN = 1 to perform the operation and stop. When the operation is completed (or aborted due an error), an byte transaction completed interrupt becomes active and the data may be read from the MRXDATA register. When the I2C module operates in Controller receiver mode, a set ACK bit causes the I2C bus controller to transmit an acknowledge automatically after each byte. This bit must be cleared when the I2C bus controller requires no further data to be transmitted from the Target transmitter.

**Table 25-55. CCTR Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-28 | RESERVED | R | 0h | |
| 27-16 | CBLEN | R/W | 0h | I2C transaction length<br>This field contains the programmed length of bytes of the Transaction.<br>0h = Smallest value<br>FFFh = Highest possible value |
| 15-6 | RESERVED | R | 0h | |
| 5 | RD_ON_TXEMPTY | R/W | 0h | Read on TX Empty<br>0h = No special behavior<br>1h = When 1 the Controller will transmit all bytes from the TX FIFO before continuing with the programmed Burst Run Read. If the DIR is not set to Read in the MSA then this bit is ignored. The Start must be set in the MCTR for proper I2C protocol. The Controller will first send the Start Condition, I2C Address with R/W bit set to write, before sending the bytes in the TX FIFO. When the TX FIFO is empty, the I2C transaction will continue as programmed in MTCR and MSA without sending a Stop Condition.<br>This is intended to be used to perform simple I2C command based reads transition that will complete after initiating them without having to get an interrupt to turn the bus around. |
| 4 | CACKOEN | R/W | 0h | Controller ACK overrride Enable<br>0h = No special behavior<br>1h = When 1 and the Controller is receiving data and the number of bytes indicated in MBLEN have been received, the state machine will generate an rxdone interrupt and wait at the start of the ACK for FW to indicate if an ACK or NACK should be sent. The ACK or NACK is selected by writing the MCTR register and setting ACK accordingly. The other fields in this register can also be written at this time to continue on with the transaction. If a NACK is sent the state machine will automatically send a Stop. |
| 3 | ACK | R/W | 0h | Data Acknowledge Enable.<br>Software needs to configure this bit to send the ACK or NACK.<br>0h = The last received data byte of a transaction is not acknowledged automatically by the Controller.<br>1h = The last received data byte of a transaction is acknowledged automatically by the Controller. |
| 2 | STOP | R/W | 0h | Generate STOP<br>0h = The controller does not generate the STOP condition.<br>1h = The controller generates the STOP condition. |
| 1 | START | R/W | 0h | Generate START<br>0h = The controller does not generate the START condition.<br>1h = The controller generates the START or repeated START condition. |



Copyright © 2026 Texas Instruments Incorporated

**Table 25-55. CCTR Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 0 | BURSTRUN | R/W | 0h | I2C Controller Enable<br>and start transaction<br>0h = In standard mode, this encoding means the Controller is unable to transmit or receive data.<br>1h = The Controller is able to transmit or receive data. |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.34 CSR Register (Offset = 1218h) [Reset = 00000000h]

CSR is shown in Table 25-56.

Return to the Summary Table.

The status register indicates the state of the I2C bus controller.

**Table 25-56. CSR Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-28 | RESERVED | R | 0h | |
| 27-16 | CBCNT | R | 0h | I2C Controller Transaction Count<br>This field contains the current count-down value of the transaction.<br>0h = Smallest value<br>FFFh = Highest possible value |
| 15-7 | RESERVED | R | 0h | |
| 6 | BUSBSY | R | 0h | I2C Bus is Busy<br>Controller State Machine will wait until this bit is cleared before starting a transaction. When first enabling the Controller in multi Controller environments, FW should wait for one I2C clock period after setting ACTIVE high before writing to the MTCR register to start the transaction so that if SCL goes low it will trigger the BUSBSY.<br>0h = The I2C bus is idle.<br>1h = 'This Status bit is set on a START or when SCL goes low. It is cleared on a STOP, or when a SCL high bus busy timeout occurs and SCL and SDA are both high. This status is cleared when the ACTIVE bit is low.<br>Note that the Controller State Machine will wait until this bit is cleared before starting an I2C transaction. When first enabling the Controller in multi Controller environments, FW should wait for one I2C clock period after setting ACTIVE high before writing to the MTCR register to start the transaction so that if SCL goes low it will trigger the BUSBSY. |
| 5 | IDLE | R | 1h | I2C Idle<br>0h = The I2C controller is not idle.<br>1h = The I2C controller is idle. |
| 4 | ARBLST | R | 0h | Arbitration Lost<br>0h = The I2C controller won arbitration.<br>1h = The I2C controller lost arbitration. |
| 3 | DATACK | R | 0h | Acknowledge Data<br>0h = The transmitted data was acknowledged<br>1h = The transmitted data was not acknowledged. |
| 2 | ADRACK | R | 0h | Acknowledge Address<br>0h = The transmitted address was acknowledged<br>1h = The transmitted address was not acknowledged. |
| 1 | ERR | R | 0h | Error<br>The error can be from the Target address not being acknowledged or the transmit data not being acknowledged.<br>0h = No error was detected on the last operation.<br>1h = An error occurred on the last operation. |
| 0 | BUSY | R | 0h | I2C Controller FSM Busy<br>The BUSY bit is set during an ongoing transaction, so is set during the transmit/receive of the amount of data set in MBLEN including START, RESTART, Address and STOP signal generation when required for the current transaction.<br>0h = The controller is idle.<br>1h = The controller is busy. |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.35 CRXDATA Register (Offset = 121Ch) [Reset = 00000000h]

CRXDATA is shown in Table 25-57.

Return to the Summary Table.

I2C Controller RX FIFO Read Data Byte
This field contains the current byte being read in the RX FIFO stack.
If the FIFO is disabled, the data byte and status are stored in the receiving holding register (the bottom word of the receive FIFO). The received data can be retrieved by reading this register.

**Table 25-57. CRXDATA Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | VALUE | R | 0h | Received Data.<br>This field contains the last received data.<br>0h = Smallest value<br>FFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.36 CTXDATA Register (Offset = 1220h) [Reset = 00000000h]

CTXDATA is shown in Table 25-58.

Return to the Summary Table.

I2C Controller Transmit Data Register.
This register is the transmit data register (the interface to the FIFOs). For transmitted data, if the FIFO is enabled, data written to this location is pushed onto the transmit FIFO. If the FIFO is disabled, data is stored in the transmitter holding register (the bottom word of the transmit FIFO).

**Table 25-58. CTXDATA Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | VALUE | R/W | 0h | Transmit Data<br>This byte contains the data to be transferred during the next transaction.<br>0h = Smallest value<br>FFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.37 CTPR Register (Offset = 1224h) [Reset = 00000001h]

CTPR is shown in Table 25-59.

Return to the Summary Table.

This register is programmed to set the timer period for the SCL clock and assign the SCL clock to standard mode.

**Table 25-59. CTPR Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-7 | RESERVED | R | 0h | |
| 6-0 | TPR | R/W | 1h | Timer Period<br>This field is used in the equation to configure SCL_PERIOD :<br>SCL_PERIOD = (1 + TPR ) × (SCL_LP + SCL_HP ) × INT_CLK_PRD<br>where:<br>SCL_PRD is the SCL line period (I2C clock).<br>TPR is the Timer Period register value (range of 1 to 127).<br>SCL_LP is the SCL Low period (fixed at 6).<br>SCL_HP is the SCL High period (fixed at 4).<br>CLK_PRD is the functional clock period in ns.<br>0h = Smallest value<br>7Fh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.38 CCR Register (Offset = 1228h) [Reset = 00000000h]

CCR is shown in Table 25-60.

Return to the Summary Table.

Controller configuration register

**Table 25-60. CCR Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-9 | RESERVED | R | 0h | |
| 8 | LPBK | R/W | 0h | I2C Loopback<br>0h = Normal operation.<br>1h = The controller in a test mode loopback configuration. |
| 7-3 | RESERVED | R | 0h | |
| 2 | CLKSTRETCH | R/W | 0h | Clock Stretching. This bit controls the support for clock stretching of the I2C bus.<br>0h = Disables the clock stretching detection.<br>This can be disabled if no Target on the bus does support clock stretching, so that the maximum speed on the bus can be reached.<br>1h = Enables the clock stretching detection.<br>Enabling the clock stretching ensures compliance to the I2C standard but could limit the speed due the clock stretching. |
| 1 | MCTL | R/W | 0h | MultiController mode. In MultiController mode the SCL high time counts once the SCL line has been detected high. If this is not enabled the high time counts as soon as the SCL line has been set high by the I2C controller.<br>0h = Disable MultiController mode.<br>1h = Enable MultiController mode. |
| 0 | ACTIVE | R/W | 0h | Device Active After this bit has been set, it should not be set again unless it has been cleared by writing a 0 or by a reset, otherwise transfer failures may occur.<br>0h = Disables the I2C Controller operation.<br>1h = Enables the I2C Controller operation. |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.39 CBMON Register (Offset = 1234h) [Reset = 00000003h]

CBMON is shown in Table 25-61.

Return to the Summary Table.

This register is used to determine the SCL and SDA signal status.

**Table 25-61. CBMON Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | |
| 1 | SDA | R | 1h | I2C SDA Status<br>0h = The I2CSDA signal is low.<br>1h = The I2CSDA signal is high. Note: During and right after reset, the SDA pin is in GPIO input mode without the internal pull enabled. For proper I2C operation, the user should have the external pull-up resistor in place before starting any I2C operations. |
| 0 | SCL | R | 1h | I2C SCL Status<br>0h = The I2CSCL signal is low.<br>1h = The I2CSCL signal is high. Note: During and right after reset, the SCL pin is in GPIO input mode without the internal pull enabled. For proper I2C operation, the user should have the external pull-up resistor in place before starting any I2C operations. |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.40 CFIFOCTL Register (Offset = 1238h) [Reset = 00000000h]

CFIFOCTL is shown in Table 25-62.

Return to the Summary Table.

I2C Controller FIFO Control

**Table 25-62. CFIFOCTL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15 | RXFLUSH | R/W | 0h | RX FIFO Flush<br>Setting this bit will Flush the RX FIFO.<br>Before clearing this bit to stop Flush the RXFIFOCNT should be checked to be 0 and indicating that the Flush has completed.<br>0h = Do not Flush FIFO<br>1h = Flush FIFO |
| 14-11 | RESERVED | R | 0h | |
| 10-8 | RXTRIG | R/W | 0h | RX FIFO Trigger<br>Indicates at what fill level in the RX FIFO a trigger will be generated.<br>Note: Programming RXTRIG to 0x0 has no effect since no data is present to transfer out of RX FIFO.<br>0h = Trigger when RX FIFO contains >= 1 byte<br>1h = Trigger when RX FIFO contains >= 2 byte<br>2h = Trigger when RX FIFO contains >= 3 byte<br>3h = Trigger when RX FIFO contains >= 4 byte<br>4h = Trigger when RX FIFO contains >= 5 byte<br>5h = Trigger when RX FIFO contains >= 6 byte<br>6h = Trigger when RX FIFO contains >= 7 byte<br>7h = Trigger when RX FIFO contains >= 8 byte |
| 7 | TXFLUSH | R/W | 0h | TX FIFO Flush<br>Setting this bit will Flush the TX FIFO.<br>Before clearing this bit to stop Flush the TXFIFOCNT should be checked to be 8 and indicating that the Flush has completed.<br>0h = Do not Flush FIFO<br>1h = Flush FIFO |
| 6-3 | RESERVED | R | 0h | |
| 2-0 | TXTRIG | R/W | 0h | TX FIFO Trigger<br>Indicates at what fill level in the TX FIFO a trigger will be generated.<br>0h = Trigger when the TX FIFO is empty.<br>1h = Trigger when TX FIFO contains ≤ 1 byte<br>2h = Trigger when TX FIFO contains ≤ 2 byte<br>3h = Trigger when TX FIFO contains ≤ 3 byte<br>4h = Trigger when TX FIFO contains ≤ 4 byte<br>5h = Trigger when TX FIFO contains ≤ 5 byte<br>6h = Trigger when TX FIFO contains ≤ 6 byte<br>7h = Trigger when TX FIFO contains ≤ 7 byte |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.41 CFIFOSR Register (Offset = 123Ch) [Reset = 00000800h]

CFIFOSR is shown in Table 25-63.

Return to the Summary Table.

I2C Controller FIFO Status Register
Note: this Register should only be read when BUSY is 0

**Table 25-63. CFIFOSR Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15 | TXFLUSH | R | 0h | TX FIFO Flush<br>When this bit is set a Flush operation for the TX FIFO is active. Clear the TXFLUSH bit in the control register to stop.<br>0h = FIFO Flush not active<br>1h = FIFO Flush active |
| 14-12 | RESERVED | R | 0h | |
| 11-8 | TXFIFOCNT | R | 8h | Number of Bytes which could be put into the TX FIFO<br>0h = Smallest value<br>8h = Highest possible value |
| 7 | RXFLUSH | R | 0h | RX FIFO Flush<br>When this bit is set a Flush operation for the RX FIFO is active. Clear the RXFLUSH bit in the control register to stop.<br>0h = FIFO Flush not active<br>1h = FIFO Flush active |
| 6-4 | RESERVED | R | 0h | |
| 3-0 | RXFIFOCNT | R | 0h | Number of Bytes which could be read from the RX FIFO<br>0h = Smallest value<br>8h = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.42 _I2CPECCTL Register (Offset = 1240h) [Reset = 00000000h]

_I2CPECCTL is shown in Table 25-64.

Return to the Summary Table.

I2C Controller PEC Control Register

**Table 25-64. _I2CPECCTL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-13 | RESERVED | R | 0h | |
| 12 | PECEN | R/W | 0h | PEC Enable<br>This bit enables the SMB Packet Error Checking (PEC). When enabled the PEC is calculated on all bits except the Start, Stop, Ack and Nack. The PEC LSFR and the Byte Counter is set to 0 when the State Machine is in the IDLE state, which occur following a Stop or when a timeout occurs. The Counter is also set to 0 after the PEC byte is sent or received. Note that the NACK is automatically send following a PEC byte that results in a PEC error.<br>The PEC Polynomial is x^8 + x^2 + x^1 + 1.<br>0h = PEC is disabled in Controller mode<br>1h = PEC is enabled in Controller mode |
| 11-9 | RESERVED | R | 0h | |
| 8-0 | PECCNT | R/W | 0h | PEC Count<br>When this field is non zero, the number of I2C bytes are counted (Note that although the PEC is calculated on the I2C address it is not counted at a byte). When the byte count = PECCNT and the state machine is transmitting, the contents of the LSFR is loaded into the shift register instead of the byte received from the Tx FIFO. When the state machine is receiving, after the last bit of this byte is received the LSFR is checked and if it is non-zero, a PEC RX Error interrupt is generated. The I2C packet must be padded to include the PEC byte for both transmit and receive. In transmit mode the FIFO must be loaded with a dummy PEC byte. In receive mode the PEC byte will be passed to the Rx FIFO.<br>In the normal Controller use case, FW would set PECEN=1 and PECCNT=SMB packet length (Not including Target Address byte, but including the PEC byte). FW would then configure DMA to allow the packet to complete unassisted and write MCTR to initiate the transaction.<br>Note that when the byte count = PEC CNT, the byte count is reset to 0 and multiple PEC calculation can automatically occur within a single I2C transaction.<br>Note that any write to the Controller_I2CPECCTL Register will clear the current PEC Byte Count in the Controller State Machine.<br>0h = Minimum Value<br>1FFh = Maximum Value |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.43 _PECSR Register (Offset = 1244h) [Reset = 00000000h]

_PECSR is shown in Table 25-65.

Return to the Summary Table.

Controller PEC Status Register

**Table 25-65. _PECSR Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-18 | RESERVED | R | 0h | |
| 17 | PECSTS_ERROR | R | 0h | This status bit indicates if a PEC check error occurred in the transaction that occurred before the last Stop. Latched on Stop.<br>0h = Indicates PEC check error did not occurr in the transaction that occurred before the last Stop<br>1h = Indicates if a PEC check error occurred in the transaction that occurred before the last Stop |
| 16 | PECSTS_CHECK | R | 0h | This status bit indicates if the PEC was checked in the transaction that occurred before the last Stop. Latched on Stop.<br>0h = Indicates PEC was not checked in the transaction that occurred before the last Stop<br>1h = Indicates if the PEC was checked in the transaction that occurred before the last Stop |
| 15-9 | RESERVED | R | 0h | |
| 8-0 | PECBYTECNT | R | 0h | PEC Byte Count<br>This is the current PEC Byte Count of the Controller State Machine.<br>0h = Minimum Value<br>1FFh = Maximum Value |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.44 TOAR Register (Offset = 1250h) [Reset = 00004000h]

TOAR is shown in Table 25-66.

Return to the Summary Table.

This register consists of seven address bits that identify the I2C device on the I2C bus.

**Table 25-66. TOAR Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15 | TMODE | R/W | 0h | This bit selects the adressing mode to be used in Target mode.<br>When 0, 7-bit addressing is used.<br>When 1, 10-bit addressing is used.<br>0h = Enable 7-bit addressing<br>1h = Enable 10-bit addressing |
| 14 | OAREN | R/W | 1h | I2C Target Own Address Enable<br>0h = Disable OAR address<br>1h = Enable OAR address |
| 13-10 | RESERVED | R | 0h | |
| 9-0 | OAR | R/W | 0h | I2C Target Own Address: This field specifies bits A9 through A0 of the Target address.<br>In 7-bit addressing mode as selected by I2CSOAR.MODE bit, the top 3 bits are don't care<br>0h = Smallest value<br>3FFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.45 TOAR2 Register (Offset = 1254h) [Reset = 00000000h]

TOAR2 is shown in Table 25-67.

Return to the Summary Table.

This register consists of seven address bits that identify the alternate address for the I2C device on the I2C bus.

**Table 25-67. TOAR2 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-23 | RESERVED | R | 0h | |
| 22-16 | OAR2_MASK | R/W | 0h | I2C Target Own Address 2 Mask: This field specifies bits A6 through A0 of the Target address.<br>The bits with value '1' in SOAR2.OAR2_MASK field will make the corresponding incoming address bits to match by default regardless of the value inside SOAR2.OAR2 i.e. corresponding SOAR2.OAR2 bit is a don't care.<br>0h = Minimum Value<br>7Fh = Maximum Value |
| 15-8 | RESERVED | R | 0h | |
| 7 | OAR2EN | R/W | 0h | I2C Target Own Address 2 Enable<br>0h = The alternate address is disabled.<br>1h = Enables the use of the alternate address in the OAR2 field. |
| 6-0 | OAR2 | R/W | 0h | I2C Target Own Address 2<br>This field specifies the alternate OAR2 address.<br>0h = Smallest value<br>7Fh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.46 TCTR Register (Offset = 1258h) [Reset = 00000404h]

TCTR is shown in Table 25-68.

Return to the Summary Table.

I2C Target Control Register

**Table 25-68. TCTR Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-11 | RESERVED | R | 0h | |
| 10 | TWUEN | R/W | 1h | Target Wakeup Enable<br>0h = When 0, the Target is not allowed to clock stretch on START detection<br>1h = When 1, the Target is allowed to clock stretch on START detection and wait for faster clock to be abvailable. This allows clean wake up support for I2C in low power mode use cases |
| 9 | EN_DEFDEVADR | R/W | 0h | Enable Deault device address<br>0h = When this bit is 0, the default device address is not matched. NOTE: it may still be matched if programmed inside SOAR/SOAR2.<br>1h = When this bit is 1, default device address of 7'h110_0001 is always matched by the Target address match logic. |
| 8 | EN_ALRESPADR | R/W | 0h | Enable Alert Response Address<br>0h = When this bit is 0, the alert response address is not matched. NOTE: it may still be matched if programmed inside SOAR/SOAR2<br>1h = When this bit is 1, alert response address of 7'h000_1100 is always matched by the Target address match logic. |
| 7 | EN_DEFHOSTADR | R/W | 0h | Enable Default Host Address<br>0h = When this bit is 0, the default host address is not matched<br>NOTE: it may still be matched if programmed inside SOAR/SOAR2<br>1h = When this bit is 1, default host address of 7'h000_1000 is always matched by the Target address match logic. |
| 6 | RXFULL_ON_RREQ | R/W | 0h | Rx full interrupt generated on RREQ condition as indicated in SSR<br>0h = When 0, RIS:TRXFULL will be set when only the Target RX FIFO is full.<br>This allows the TRXFULL interrupt to be used to indicate that the I2C bus is being clock stretched and that the FW must either read the RX FIFO or ACK/NACK the current Rx byte.<br>1h = When 1, RIS:TRXFULL will be set when the Target State Machine is in the RX_WAIT or RX_ACK_WAIT states which occurs when the I2C transaction is clock stretched because the RX FIFO is full or the ACKOEN has been set and the state machine is waiting for FW to ACK/NACK the current byte. |
| 5 | TXWAIT_STALE_TXFIFO | R/W | 0h | Tx transfer waits when stale data in Tx FIFO.<br>This prevents stale bytes left in the TX FIFO from automatically being sent on the next I2C packet. Note: this should be used with TXEMPTY_ON_TREQ set to prevent the Target State Machine from waiting for TX FIFO data without an interrupt notification when the FIFO data is stale.<br>0h = When 0, the TX FIFO empty signal to the Target State Machine indicates that the TX FIFO is empty.<br>1h = When 1, the TX FIFO empty signal to the Target State Machine will indicate that the TX FIFO is empty or that the TX FIFO data is stale. The TX FIFO data is determined to be stale when there is data in the TX FIFO when the Target State Machine leaves the TXMODE as defined in the SSR register. This can occur is a Stop or timeout occur when there are bytes left in the TX FIFO. |



Copyright © 2026 Texas Instruments Incorporated

**Table 25-68. TCTR Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 4 | TXTRIG_TXMODE | R/W | 0h | Tx Trigger when Target FSM is in Tx Mode<br>0h = No special behavior<br>1h = When 1, RIS:TXFIFOTRG will be set when the Target TX FIFO has reached the trigger level AND the Target State Machine is in the TXMODE as defined in the SSR register.<br>When cleared RIS:TXFIFOTRG will be set when the Target TX FIFO is at or above the trigger level.<br>This setting can be used to hold off the TX DMA until a transaction starts.<br>This allows the DMA to be configured when the I2C is idle but have it wait till the transaction starts to load the Target TX FIFO, so it can load from a memory buffer that might be changing over time. |
| 3 | TXEMPTY_ON_TREQ | R/W | 0h | Tx Empty Interrupt on TREQ<br>0h = When 0, RIS:TTXEMPTY will be set when only the Target TX FIFO is empty.<br>This allows the TTXEMPTY interrupt to be used to indicate that the I2C bus is being clock stretched and that Target TX data is required.<br>1h = When 1, RIS:TTXEMPTY will be set when the Target State Machine is in the TX_WAIT state which occurs when the TX FIFO is empty AND the I2C transaction is clock stretched waiting for the FIFO to receive data. |
| 2 | TCLKSTRETCH | R/W | 1h | Target Clock Stretch Enable<br>0h = Target clock stretching is disabled<br>1h = Target clock stretching is enabled |
| 1 | GENCALL | R/W | 0h | General call response enable<br>Modify only when UCSWRST = 1.<br>0b = Do not respond to a general call<br>1b = Respond to a general call<br>0h = Do not respond to a general call<br>1h = Respond to a general call |
| 0 | ACTIVE | R/W | 0h | Device Active. Setting this bit enables the Target functionality.<br>0h = Disables the I2C Target operation.<br>1h = Enables the I2C Target operation. |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.47 TSR Register (Offset = 125Ch) [Reset = 00000000h]

TSR is shown in Table 25-69.

Return to the Summary Table.

This register functions as a control register when written, and a status register when read.

**Table 25-69. TSR Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-19 | RESERVED | R | 0h | |
| 18-9 | ADDRMATCH | R | 0h | Indicates the address for which Target address match happened<br>0h = Minimum Value<br>3FFh = Maximum Value |
| 8 | STALE_TXFIFO | R | 0h | Stale Tx FIFO<br>0h = Tx FIFO is not stale<br>1h = The TX FIFO is stale. This occurs when the TX FIFO was not emptied during the previous I2C transaction. |
| 7 | TXMODE | R | 0h | Target FSM is in TX MODE<br>0h = The Target State Machine is not in TX_DATA, TX_WAIT, TX_ACK or ADDR_ACK state with the bus direction set to read.<br>1h = The Target State Machine is in TX_DATA, TX_WAIT, TX_ACK or ADDR_ACK state with the bus direction set to read. |
| 6 | BUSBSY | R | 0h | I2C bus is busy<br>0h = The I2C Bus is not busy<br>1h = The I2C Bus is busy. This is cleared on a timeout. |
| 5 | QCMDRW | R | 0h | Quick Command Read / Write<br>This bit only has meaning when the QCMDST bit is set.<br>Value Description:<br>0: Quick command was a write<br>1: Quick command was a read<br>0h = Quick command was a write<br>1h = Quick command was a read |
| 4 | QCMDST | R | 0h | Quick Command Status<br>Value Description:<br>0: The last transaction was a normal transaction or a transaction has not occurred.<br>1: The last transaction was a Quick Command transaction<br>0h = The last transaction was a normal transaction or a transaction has not occurred.<br>1h = The last transaction was a Quick Command transaction. |
| 3 | OAR2SEL | R | 0h | OAR2 Address Matched<br>This bit gets reevaluated after every address comparison.<br>0h = Either the OAR2 address is not matched or the match is in legacy mode.<br>1h = OAR2 address matched and ACKed by the Target. |
| 2 | RXMODE | R | 0h | Target FSM is in Rx MODE<br>0h = The Target State Machine is not in the RX_DATA, RX_ACK, RX_WAIT, RX_ACK_WAIT or ADDR_ACK state with the bus direction set to write.<br>1h = The Target State Machine is in the RX_DATA, RX_ACK, RX_WAIT, RX_ACK_WAIT or ADDR_ACK state with the bus direction set to write. |
| 1 | TREQ | R | 0h | Transmit Request<br>0h = No outstanding transmit request.<br>1h = The I2C controller has been addressed as a Target transmitter and is using clock stretching to delay the Controller until data has been written to the TTXDATA FIFO (Target TX FIFO is empty). |



Copyright © 2026 Texas Instruments Incorporated

**Table 25-69. TSR Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 0 | RREQ | R | 0h | Receive Request<br>0h = No outstanding receive data.<br>1h = The I2C controller has outstanding receive data from the I2C Controller and is using clock stretching to delay the Controller until the data has been read from the TRXDATA FIFO (Target RX FIFO is full). |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.48 TRXDATA Register (Offset = 1260h) [Reset = 00000000h]

TRXDATA is shown in Table 25-70.

Return to the Summary Table.

I2C Target RX FIFO Read Data Byte
This field contains the current byte being read in the RX FIFO stack.
If the FIFO is disabled, the data byte and status are stored in the receiving holding register (the bottom word of the receive FIFO). The received data can be retrieved by reading this register.

**Table 25-70. TRXDATA Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | VALUE | R | 0h | Received Data.<br>This field contains the last received data.<br>0h = Smallest value<br>FFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.49 TTXDATA Register (Offset = 1264h) [Reset = 00000000h]

TTXDATA is shown in Table 25-71.

Return to the Summary Table.

I2C Target Transmit Data Register.
This register is the transmit data register (the interface to the FIFOs). For transmitted data, if the FIFO is enabled, data written to this location is pushed onto the transmit FIFO. If the FIFO is disabled, data is stored in the transmitter holding register (the bottom word of the transmit FIFO).

**Table 25-71. TTXDATA Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | VALUE | R/W | 0h | Transmit Data<br>This byte contains the data to be transferred during the next transaction.<br>0h = Smallest value<br>FFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.50 TACKCTL Register (Offset = 1268h) [Reset = 00000000h]

TACKCTL is shown in Table 25-72.

Return to the Summary Table.

This register enables the I2C Target to Not Acknowledge (NACK) for invalid data or command or Acknowledge (ACK) for valid data or command. The I2C clock is pulled low after the last data bit until this register is written.

**Table 25-72. TACKCTL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-5 | RESERVED | R | 0h | |
| 4 | ACKOEN_ON_PECDONE | R/W | 0h | When set this bit will automatically turn on the Target ACKOEN field following the ACK/NACK of the received PEC byte.<br>0h = No special behavior<br>1h = When set this bit will automatically turn on the Target ACKOEN field following the ACK/NACK of the received PEC byte. |
| 3 | ACKOEN_ON_PECNEXT | R/W | 0h | When set this bit will automatically turn on the Target ACKOEN field following the ACK/NACK of the byte received just prior to the PEC byte.<br>Note that when ACKOEN is set the PEC byte will not automatically be ACKed/NACKed by the State Machine and FW must perform this function by writing Target_SACKCTL.<br>0h = No special behavior<br>1h = When set this bit will automatically turn on the Target ACKOEN field following the ACK/NACK of the byte received just prior to the PEC byte.<br>Note that when ACKOEN is set the PEC byte will not automatically be ACKed/NACKed by the State Machine and FW must perform this function by writing Target_SACKCTL. |
| 2 | ACKOEN_ON_START | R/W | 0h | When set this bit will automatically turn on the Target ACKOEN field following a Start Condition.<br>0h = No special behavior<br>1h = When set this bit will automatically turn on the Target ACKOEN field following a Start Condition. |
| 1 | ACKOVAL | R/W | 0h | I2C Target ACK Override Value<br>Note: for General Call this bit will be ignored if set to NACK and Target continues to receive data.<br>0h = An ACK is sent indicating valid data or command.<br>1h = A NACK is sent indicating invalid data or command. |
| 0 | ACKOEN | R/W | 0h | I2C Target ACK Override Enable<br>0h = A response in not provided.<br>1h = An ACK or NACK is sent according to the value written to the ACKOVAL bit. |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.51 TFIFOCTL Register (Offset = 126Ch) [Reset = 00000000h]

TFIFOCTL is shown in Table 25-73.

Return to the Summary Table.

I2C Target FIFO Control

**Table 25-73. TFIFOCTL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15 | RXFLUSH | R/W | 0h | RX FIFO Flush<br>Setting this bit will Flush the RX FIFO.<br>Before clearing this bit to stop Flush the RXFIFOCNT should be checked to be 0 and indicating that the Flush has completed.<br>0h = Do not Flush FIFO<br>1h = Flush FIFO |
| 14-11 | RESERVED | R | 0h | |
| 10-8 | RXTRIG | R/W | 0h | RX FIFO Trigger<br>Indicates at what fill level in the RX FIFO a trigger will be generated.<br>Note: Programming RXTRIG to 0x0 has no effect since no data is present to transfer out of RX FIFO.<br>4h = Trigger when RX FIFO contains >= 5 byte<br>5h = Trigger when RX FIFO contains >= 6 byte<br>6h = Trigger when RX FIFO contains >= 7 byte<br>7h = Trigger when RX FIFO contains >= 8 byte |
| 7 | TXFLUSH | R/W | 0h | TX FIFO Flush<br>Setting this bit will Flush the TX FIFO.<br>Before clearing this bit to stop Flush the TXFIFOCNT should be checked to be 8 and indicating that the Flush has completed.<br>0h = Do not Flush FIFO<br>1h = Flush FIFO |
| 6-3 | RESERVED | R | 0h | |
| 2-0 | TXTRIG | R/W | 0h | TX FIFO Trigger<br>Indicates at what fill level in the TX FIFO a trigger will be generated.<br>4h = Trigger when TX FIFO contains ≤ 4 byte<br>5h = Trigger when TX FIFO contains ≤ 5 byte<br>6h = Trigger when TX FIFO contains ≤ 6 byte<br>7h = Trigger when TX FIFO contains ≤ 7 byte |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.52 TFIFOSR Register (Offset = 1270h) [Reset = 00000800h]

TFIFOSR is shown in Table 25-74.

Return to the Summary Table.

I2C Target FIFO Status Register
Note: this Register should only be read when BUSY is 0

**Table 25-74. TFIFOSR Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15 | TXFLUSH | R | 0h | TX FIFO Flush<br>When this bit is set a Flush operation for the TX FIFO is active. Clear the TXFLUSH bit in the control register to stop.<br>0h = FIFO Flush not active<br>1h = FIFO Flush active |
| 14-12 | RESERVED | R | 0h | |
| 11-8 | TXFIFOCNT | R | 8h | Number of Bytes which could be put into the TX FIFO<br>0h = Smallest value<br>8h = Highest possible value |
| 7 | RXFLUSH | R | 0h | RX FIFO Flush<br>When this bit is set a Flush operation for the RX FIFO is active. Clear the RXFLUSH bit in the control register to stop.<br>0h = FIFOFlush not active<br>1h = FIFO Flush active |
| 6-4 | RESERVED | R | 0h | |
| 3-0 | RXFIFOCNT | R | 0h | Number of Bytes which could be read from the RX FIFO<br>0h = Smallest value<br>8h = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.53 _PECCTL Register (Offset = 1274h) [Reset = 00000000h]

_PECCTL is shown in Table 25-75.

Return to the Summary Table.

I2C Target PEC Control Register

**Table 25-75. _PECCTL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-13 | RESERVED | R | 0h | |
| 12 | PECEN | R/W | 0h | PEC Enable<br>This bit enables the SMB Packet Error Checking (PEC). When enabled the PEC is calculated on all bits except the Start, Stop, Ack and Nack. The PEC LSFR and the Byte Counter is set to 0 when the State Machine is in the IDLE state, which occur following a Stop or when a timeout occurs. The Counter is also set to 0 after the PEC byte is sent or received. Note that the NACK is automatically send following a PEC byte that results in a PEC error.<br>The PEC Polynomial is x^8 + x^2 + x^1 + 1.<br>0h = PEC transmission and check is disabled<br>1h = PEC transmission and check is enabled |
| 11-9 | RESERVED | R | 0h | |
| 8-0 | PECCNT | R/W | 0h | When this field is non zero, the number of I2C data bytes are counted. When the byte count = PECCNT and the state machine is transmitting, the contents of the LSFR is loaded into the shift register instead of the byte received from the Tx FIFO. When the state machine is receiving, after the last bit of this byte is received the LSFR is checked and if it is non-zero, a PEC RX Error interrupt is generated. The I2C packet must be padded to include the PEC byte for both transmit and receive. In transmit mode the FIFO must be loaded with a dummy PEC byte. In receive mode the PEC byte will be passed to the Rx FIFO.<br>In the normal Target use case, FW would set PECEN=1 and PECCNT=0 and use the ACKOEN until the remaining SMB packet length is known. FW would then set the PECCNT to the remaining packet length (Including PEC bye). FW would then configure DMA to allow the packet to complete unassisted and exit NoAck mode.<br>Note that when the byte count = PEC CNT, the byte count is reset to 0 and multiple PEC calculation can automatically occur within a single I2C transaction<br>0h = Minimum Value<br>1FFh = Maximum Value |



Copyright © 2026 Texas Instruments Incorporated

### 25.3.54 _PECSR Register (Offset = 1278h) [Reset = 00000000h]

_PECSR is shown in Table 25-76.

Return to the Summary Table.

Target PEC Status Register

**Table 25-76. _PECSR Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-18 | RESERVED | R | 0h | |
| 17 | PECSTS_ERROR | R | 0h | This status bit indicates if a PEC check error occurred in the transaction that occurred before the last Stop. Latched on Stop.<br>0h = Indicates PEC check error did not occurr in the transaction that occurred before the last Stop<br>1h = Indicates PEC check error occurred in the transaction that occurred before the last Stop |
| 16 | PECSTS_CHECK | R | 0h | This status bit indicates if the PEC was checked in the transaction that occurred before the last Stop. Latched on Stop.<br>0h = Indicates PEC was not checked in the transaction that occurred before the last Stop<br>1h = Indicates PEC was checked in the transaction that occurred before the last Stop |
| 15-9 | RESERVED | R | 0h | |
| 8-0 | PECBYTECNT | R | 0h | This is the current PEC Byte Count of the Target State Machine.<br>0h = Minimum Value<br>1FFh = Maximum Value |



Copyright © 2026 Texas Instruments Incorporated

Chapter 26

# SPI

The serial peripheral interface (SPI) module provides a standardized serial interface to transfer data between MSPM0 devices and other external devices with SPI interface.


**26.1 SPI Overview** ..... 1415
**26.2 SPI Operation** ..... 1418
**26.3 SPI Registers** ..... 1431





Copyright © 2026 Texas Instruments Incorporated

## 26.1 SPI Overview

The SPI module provides a standardized serial interface to transfer data between MSPM0 devices and other external devices using SPI protocols (such as a Sensors, Memory, ADCs, or DACs).

### *26.1.1 Purpose of the Peripheral*

The SPI module acts as a controller or peripheral interface for synchronous serial communication with peripheral devices and other controllers. The transmit and receive paths are buffered with internal, independent FIFO memories allowing up to 4 entries with 16-bit width. A DMA interface is also provided to allow the data exchange with the transmit and receive FIFOs.

### *26.1.2 Features*

The SPI modules have the following features:

- Configurable as a controller or a peripheral
- Programmable clock bit rate and prescaler
- Separate transmit (TX) and receive (RX) first-in first-out buffers (FIFOs)
- Programmable data frame size from 4-bits to 16-bits (Controller Mode)
- Programmable data frame size from 7-bits to 16-bits (Peripheral Mode)
- Interrupts for transmit and receive FIFOs, overrun and timeout interrupts, and DMA done
- Programmable SPI mode support Motorola SPI or Texas Instruments format
- Direct memory access controller interface (DMA):
  - Separate channels for transmit and receive
  - Transfer complete interrupt



Copyright © 2026 Texas Instruments Incorporated

### 26.1.3 Functional Block Diagram

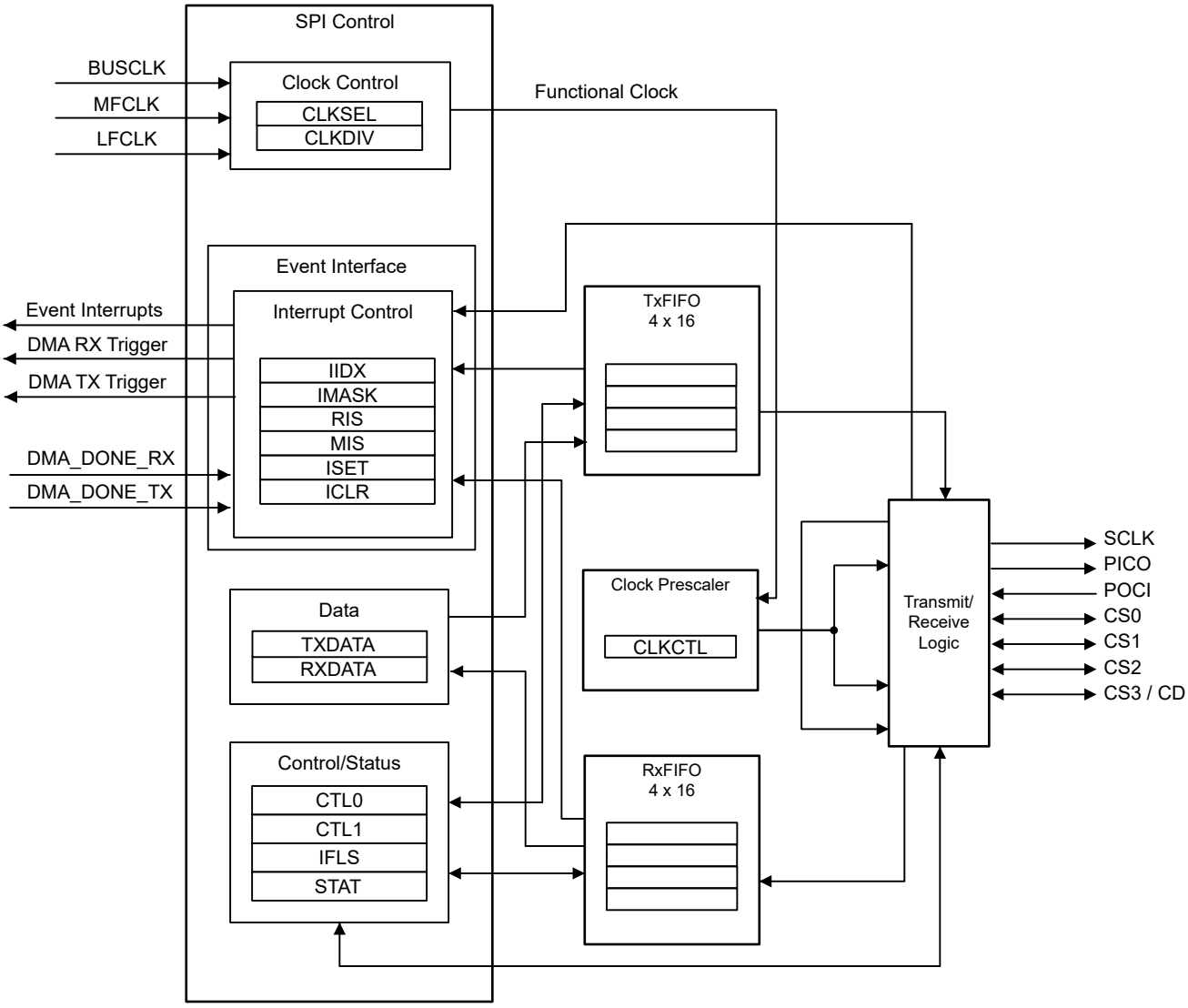



**Figure 26-1. SPI Functional Block Diagram**

### 26.1.4 External Connections and Signal Descriptions

Figure 26-2 and Table 26-1 show an overview of the pin functions for different operation modes of the SPI module.



Copyright © 2026 Texas Instruments Incorporated

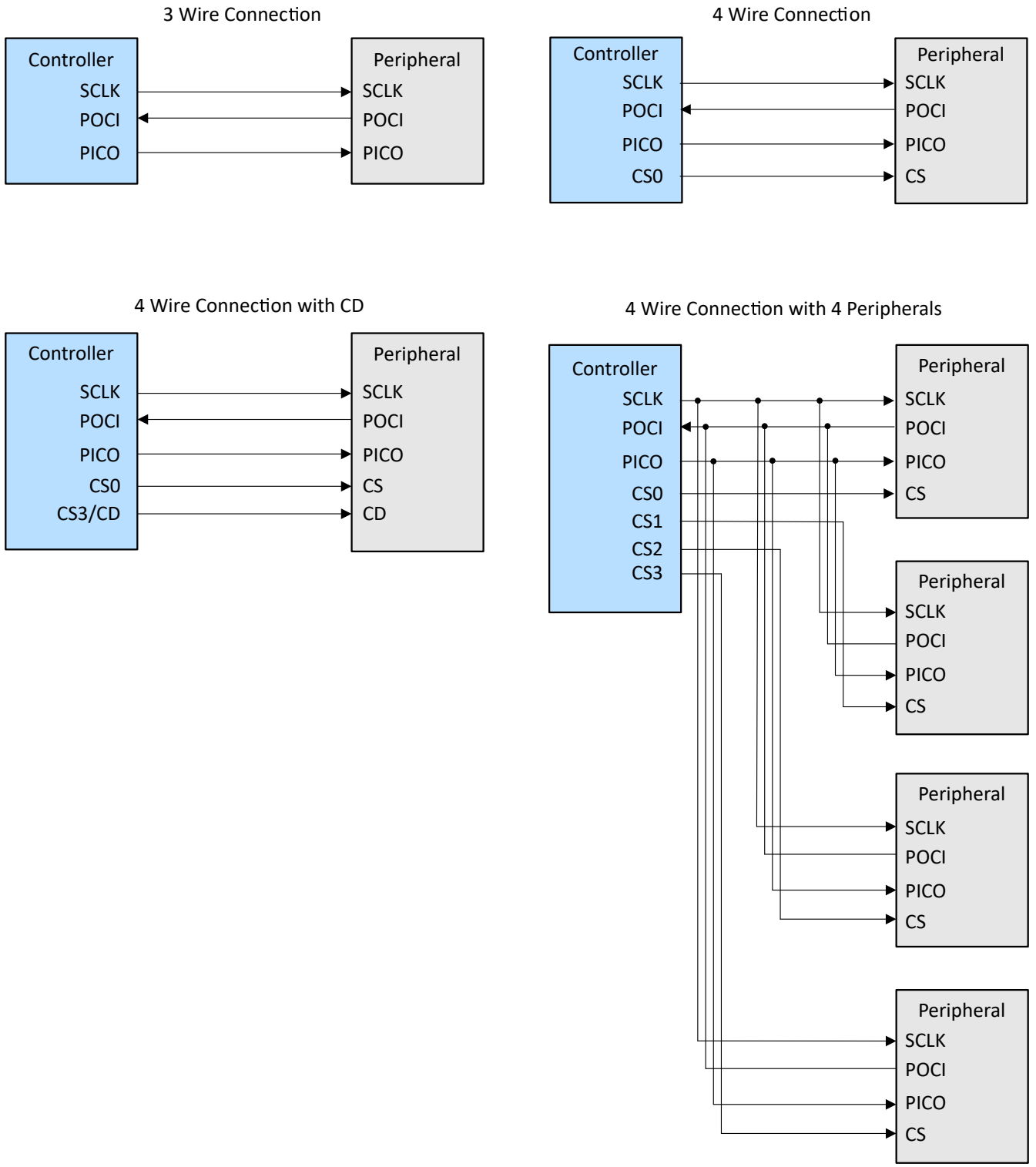



**Figure 26-2. External Connections for Different SPI Configurations**



Copyright © 2026 Texas Instruments Incorporated

**Table 26-1. Pin Function Overview**

| Standard SPI | Feature |
|---|---|
| **SCLK** | **SPI clock**<br>Controller mode: SCLK is an output<br>Peripheral mode: SCLK is an input |
| **PICO** | **Controller out, peripheral in**<br>Controller mode: PICO is the data output line<br>Peripheral mode: PICO is the data input line |
| **POCI** | **Controller in, peripheral out**<br>Controller mode: POCI is the data input line<br>Peripheral mode: POCI is the data output line |
| **CS0** | Chip select signal 0, used in 4-pin mode |
| **CS1** | Chip select signal 1, used in 4-pin mode |
| **CS2** | Chip select signal 2, used in 4-pin mode |
| **CS3/CD** | Chip select signal 3 or Command/Data, used in 4-pin mode |

For UNICOMM, to meet timing requirements in controller mode, chip select and clock should be looped back from IOs. This compensates for the IO delays within the SoC. At the device's IOMUX level, CS and SCLK INENA bits will need to be set.

## 26.2 SPI Operation

### *26.2.1 Clock Control*

The SPI internal functional clock is selected and divided from the clock sourced to this module.

- Use SPIx.CLKSEL register to select the source of the SPI functional clock.
  - BUSCLK: the current bus clock is selected as the source for SPI. The current bus clock depends on power domain. If the SPI instance is in power domain 1 (PD1) please refer to MCLK, if the SPI instance is in power domain 0 (PD0) refer to ULPCLK.
  - MFCLK: MFCLK is selected as the source for SPI, refer to MFCLK.
  - LFCLK: LFCLK is selected as the source for SPI, refer to LFCLK
- Use SPIx.CLKDIV register to select the divide ratio of the SPI function clock. Options are from divide by 1 to 8.

The SPI module must be enabled before being configured for use by using the CTL1.ENABLE bit in SPIx.PWREN register (see peripheral power enable). When the SPI is set up, or the configuration needs to be changed, after enabling power, the SPI peripheral must be configured to the desired configuration and then enabled via CTL1.ENABLE bit to avoid unpredictable behavior during the updates or for the first data received or transmitted afterward.

The maximum SPI frequency supported in the controller and peripheral mode depends on the device clock option and IO option. Please refer to specific device data sheet spec for more information.

### *26.2.2 General Architecture*

#### 26.2.2.1 Chip Select and Command Handling

##### *26.2.2.1.1 Chip Select Control*

To configure the SPI to be in controller mode by setting the CTL1.CP bit to 1, and to be in peripheral mode by clearing the CTL1.CP bit.

The CTL0.CSSEL bit selects which connected peripheral is addressed by the up to 4 CS signals. The bits are controlled by the SPI module in controller or target/peripheral mode. The selected signal is controlled during the transfers.



Copyright © 2026 Texas Instruments Incorporated

The chip select signal needs to be provided by the controller in four-wire mode and the chip select polarity can be inverted by configuring the PINCM.CSx.INV register.

In peripheral mode, the clock is provided by the controller and used by the peripheral to capture the data. The peripheral has the option to operate in 3-wire or 4-wire mode. 4-wire mode only accepts data transfers if the CS is activated.

The CTL0.CSCLR bit controls whether the transmit/receive shift register is automatically cleared when the chip select (CS) signal becomes inactive:

- CTL0.CSCLR = 0: The bit counter state is preserved when CS deactivates the peripheral.
- CTL0.CSCLR = 1: The bit counter is reset to zero when CS deactivates the peripheral.

When using CTL0.CSCLR = 1, observe these timing constraints to ensure proper synchronization:

- CS Inactive Period: Maintain CS in the inactive state for at least two functional clock cycles before reasserting it.
- CS Setup Time: Allow at least 2 SPI functional clock cycles between CS activation and the first clock edge.

These timing requirements help the peripheral resynchronize with the controller after clock-line disturbances or glitches, or during system initialization. This feature applies only when operating in peripheral mode.

**Note**

The CSCLR function requires the CS disable pulse to be longer than 2 SPI function clock cycles to properly detect and clear the SPI bit counter. The CS lead time (CS active to the first bit-clock edge) must be at least 2 SPI function clock cycles. Double-check the device's datasheet to see if this note applies to your device.

#### *26.2.2.1.2 Command Data Control*

When using the Motorola frame format, the CDMODE bit can be set to use the CS3/CD line as signal to distinguish between Command and Data information. This is often used for LCD or data storage devices.

- CD level low: command function
- CD level high: data function

**Command Function:** To send out COMMAND, load CDMODE with 0xF and populate the transmit FIFO with the COMMAND data to be sent; the module will drive CS3/CD low and transmit the COMMAND data on the TX line.

**Data Function with CDMODE as counter:** After the COMMAND data is sent, load CDMODE with the number of data bytes to be transmitted and populate the transmit FIFO with the data to be sent. CS3/CD line will be driven high & data will be transmitted on the TX line; CDMODE will be decremented by 1 after each transmission. CS3/CD line remains high after the counter reaches '0'.

Alternatively, the DATA function can also be implemented in MANUAL mode, by writing CDMODE with a value of '0' and then populating the transmit FIFO. This will pull CS3/CD high and transmit populated in the transmit FIFO.

**Note**

1. The CDMODE bit is available only when the CDENABLE configuration bit is set; while the CDENABLE bit can be updated only when the SPI IP is disabled, the CDMODE bit can be updated at any time. 2. The CDMODE bit is not write-locked when counting down; therefore, a write to the CDMODE bit will restart the counter with the new value. 3. Before writing a new value into the CDMODE bit, ensure that the FIFO is empty.

#### 26.2.2.2 Data Format

The control bit CTL1.MSB defines the direction of data input and output: most-significant-bit (MSB) or least-significant-bit (LSB) first. If the parity is enabled, the parity bits are always received as the last bit.



Copyright © 2026 Texas Instruments Incorporated

With the control register bits CTL0.DSS, the bit length per transfer will be defined between 4-16 bits for Controller mode and 7-16 bits for Peripheral mode.

A transfer will be triggered by writing to the TX buffer register. The data write needs to have at least the number of bits of the transfer. For example, if only a byte is written to the TX buffer, but the length of the transfer is > 8, the missing bits will be filled with 0s. On the receive path, the received data will be moved to the RXFIFO or RX buffer after the number of bits defined in the CTL0.DSS register has been received.

The RX and TX buffer shall be accessed with at least the bits covering one transfer.

- 4-8 bits : byte access (Peripheral mode: 7-8 Bits)
- 9-16 bits : 16 bit access

Clock polarity (CTL0.SPO) is used to control the clock polarity when data is not being transferred, and it is only used in theMotorola SPI frame mode.

- 0h = peripheral produces a steady state LOW value on the CLKOUT pin when data is not being transferred.
- 1h = peripheral produces a steady state HIGH value on the CLKOUT pin when data is not being transferred.

The Clock phase (CTL0.SPH) bit selects the clock edge that captures data and enables the data to change state. It has the greatest impact on the first bit transmitted, either by permitting or not permitting a clock transition before the first data capture edge. Please refer to the Motorola SPI frame mode section to check the diagrams. **(Add line here)**

- 0h = Data is captured on the first clock edge transition.
- 1h = Data is captured on the second clock edge transition.

The SPI can be configured to work in Peripheral mode with CTL1.CP bit = 0. In Peripheral mode, the clock is provided by the controller and is available to the peripheral on the CLK pins, which must be configured as inputs. The CS input signal selects/enables the peripheral's data receive path in 4-wire mode.

The SPI can be configured to work in Controller mode with CTL1.CP bit = 1. In Controller mode, the clock needs to be generated by selecting a clock source using the clock select bits. It also needs to control the CS signal depending on the selected protocol.

When setting the CTL1.PEN bit, the last bit will be used as parity to evaluate the integrity of the previous bits. The CTL1.PES bit selects the parity mode (even or odd). When detecting a fault, the interrupt flag RIS.PER is set to mark the data as invalid. Parity checking is a feature to improve the robustness of the communication.

#### 26.2.2.3 Delayed data sampling

When the SPI input data arrives late at the POCI pin due to:

- Signal propagation delays
- Slow peripheral response times
- Board routing delays

The controller might sample the wrong data, i.e., it could read the previous bit instead of the current, but at the normal sampling clock edge. The solution for this is to use the CLKCTL.DSAMPLE feature, which allows you to shift the sampling point later within each clock cycle.

.

#### 26.2.2.4 Clock Generation

The SPI includes a programmable bit-rate clock divider and a prescaler to generate the serial output clock (SCLK).

Bit rates supported are up to, the input clocks divided by 2. The input clock selection depends on the specific device, refer to the device data sheet and Clock Control section.

The SPI functionality can work with any of these selected inputs : BUSCLK, MFCLK and LFCLK

"Functional Clock" is the output after clock division performed according to ratio selected by the CLKDIV register. Functional clock = Selected input clock / (1 + CLKDIV)



Copyright © 2026 Texas Instruments Incorporated

Functional Clock x 2 (SCLK) is the output after dividing the Functional Clock by the Prescalar value. SCLK = Functional Clock / ((1 + SCR )*2)

#### 26.2.2.5 FIFO Operation

**Transmit FIFO**

The TX FIFO is a 16-bit-wide, 4-location-deep, first-in first-out memory buffer. The CPU writes data to the FIFO by writing the SPI Data Register TXDATA.DATA, and data is stored in the FIFO until the transmission logic reads it out.

When configured as a controller or a peripheral, parallel data is written into the TX FIFO before serial conversion and transmission to the attached peripheral or controller, respectively, through the PICO or POCI pin.

In peripheral mode, the SPI transmits data each time the controller initiates a transaction. If the TX FIFO is empty and the controller initiates a transfer, the peripheral sends undefined/random data to the transmit FIFO. The user or software is responsible for making valid data available to the FIFO as needed. The SPI can be configured to generate an interrupt or a DMA request as configured in the FIFO settings. The transmit FIFO has a TXFIFO_UNF interrupt to indicate a FIFO underflow condition.

**Receive FIFO**

The RX FIFO is a 16-bit-wide, 4-location-deep, first-in-first-out memory buffer. Data received from the serial interface is stored in the buffer until read by the CPU or DMA, which accesses the read FIFO via the SPIx.RXDATA register.

When configured as a controller or peripheral, serial data is received through the POCI or PICO pin.

With the FIFO fill level trigger signals located in the STAT register (TFE, TNF, RFE, RNF), if the FIFO is full and new data is written to it without first reading data, the RXFIFO overflow event is set. The receive FIFO has an RXFULL interrupt to indicate a FIFO full condition.

#### 26.2.2.6 Loopback mode

The SPI module can be placed into an internal loopback mode for diagnostic or debug work by setting the LBM bit in the CTL1 register. In loopback mode, the data from the TX FIFO can be serially transmitted into the RX FIFO. Data from the RX FIFO can be read to check whether the transmission was correct. External toggling of the IOs has no effect when the module is set to internal loopback mode.



Copyright © 2026 Texas Instruments Incorporated

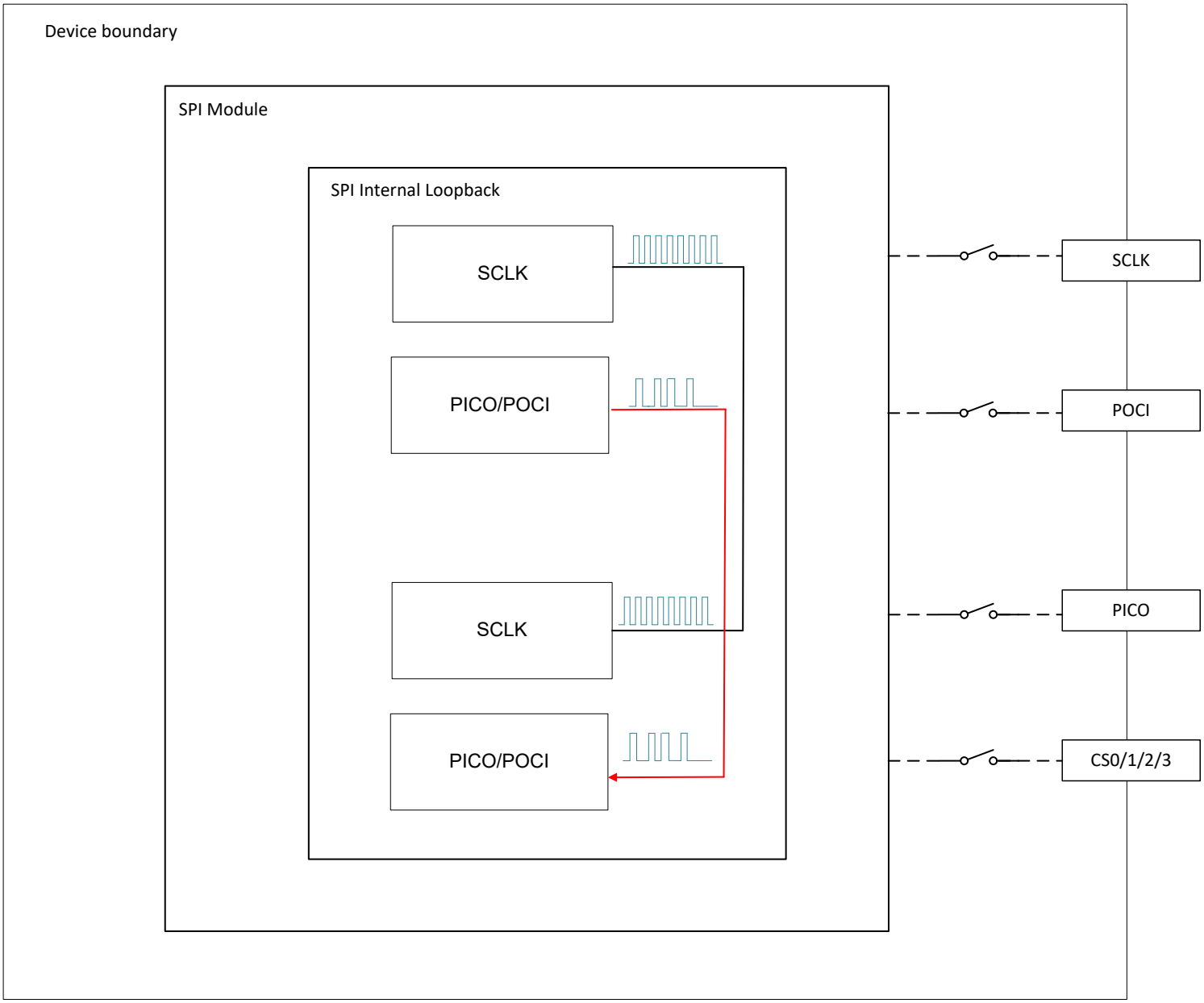



**Figure 26-3. Internal Loop Back Mode**

#### 26.2.2.7 DMA Operation

The SPI provides an interface to the DMA controller with separate channels for transmit and receive. The DMA operation of the SPI is enabled through the SPI Event and DMA register. When DMA operation is enabled, the SPI asserts a DMA request on the receive or transmit channel when the associated FIFO can transfer data.

For the receive channel a transfer request is asserted whenever the amount of data in the receive FIFO is at or above the FIFO trigger level configured using the RXIFLSEL bit in IFLS register or the receive timeout has triggered, in this case the amount of data received so far will be transmitted.

For the transmit channel a transfer request is asserted whenever the transmit FIFO contains fewer characters than the FIFO trigger level configured using the TXIFLSEL bit in IFLS register. The DMA transfer requests are handled automatically by the DMA controller depending on how the DMA channel is configured.



Copyright © 2026 Texas Instruments Incorporated

The DMA transfers can be configured and aligned between the data width of the SPI transfers and the bus accesses width of 8/16 bits to make an efficient usage of the bus. The trigger and transfers are independent for receive and transmit.

See more information about the interrupt and event in Section 26.2.6.2 section.

#### 26.2.2.8 Repeat Transfer mode

With the CTL1.REPEATTX bits the last character transmit will be repeated as defined by the register bits. A value of 0 in CTL1.REPEATTX bits will disable this mode and this function is only available in Controller mode. The transfer will be started with writing a data into the TX Buffer. Then the data will be repeatedly sent with the given value. The behavior is identical as if the data would be written into the TX Buffer that many times as defined by the value here. It can be used to clean a transfer or to pull a certain amount of data from a peripheral.

When REPEATTX is used it needs to be aligned with the data in the FIFO. So the below shown sequence should be used:

- Wait and check till FIFO is empty
- Setup REPEATTX
- Write to TXDATA / FIFO
- Wait till requested data has been received

#### 26.2.2.9 Low Power Mode

The SPI module is located in power domain 1 (PD1) and, as such, is only active in RUN and SLEEP modes. The SPI module needs to be disabled before entering low-power mode. After waking from a low-power mode, the SPI module must be enabled again. If the SPI module is enabled by application software, an entry into STOP or STANDBY low-power mode temporarily disables the SPI module while the device is in STOP or STANDBY mode.

### *26.2.3 Protocol Descriptions*

The protocol format mode can be selected by using CTL0.FRF register. The supported options include Motorola 3-wire, Motorola 4-wire, and Texas Instruments Synchronous.

#### 26.2.3.1 Motorola SPI Frame Format

The Motorola SPI interface is a 4-wire interface where the CS signal behaves as a peripheral select. In the 3-wire mode the CS signals is not required and the module behaves as if always selected. The main feature of the Motorola SPI format is that the inactive state and phase of the SCLK signal can be programmed through the SPO and SPH bits in the SPIx.CTL0 control register.

**SPO Clock Polarity Bit**

If the CTL0.SPO clock polarity control bit is clear; it produces a steady-state low value on the SCLK pin when data is not being transferred. If the CTL0.SPO bit is set; it places a steady-state high value on the SCLK pin when data is not being transferred. In 4-wire/controller mode, when CS is idle, SPICLK will be driven to its idle state. In 4-wire/peripheral mode, SCLK is expected to be in the idle state when CD is idle.

**SPH Phase-Control Bit**

The CTL0.SPH phase-control bit selects the clock edge that captures data, and allows it to change state. The state of this bit has the most significant impact on the first bit transmitted, either allowing or preventing a clock transition before the first data capture edge. If the CTL0.SPH phase-control bit is clear, data is captured on the first clock edge transition. If the SPH bit is set, data is captured on the second clock edge transition. In 4-wire mode with SPH =1, the first bit is driven on the CS falling edge and captured on the first clock edge.

**Motorola SPI Frame Format with SPO = 0 and SPH = 0**

Figure 26-4 shows signal sequences for Motorola SPI format with SPO = 0 and SPH = 0.



Copyright © 2026 Texas Instruments Incorporated

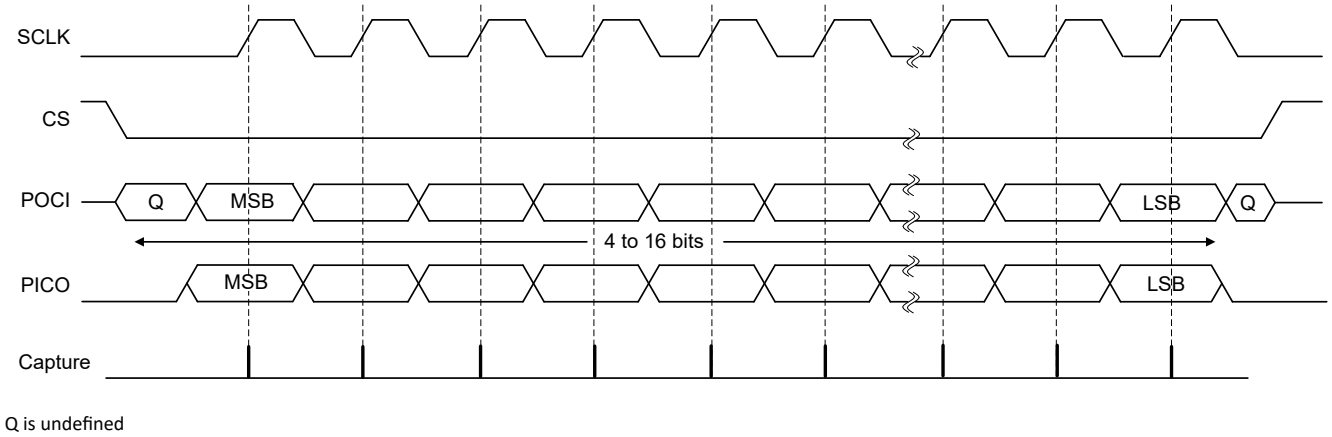



**Figure 26-4. Motorola SPI Format With SPO = 0 and SPH = 0**

In this configuration, the following occurs during idle periods:

- SCLK is forced low
- CS is forced high
- The transmit data line PICO is forced low
- When the SPI is configured as a controller, it enables the SCLK pin
- When the SPI is configured as a peripheral, it disables the SCLK pin

If the SPI is enabled and valid data is in the TX FIFO, the CS controller signal is driven low at the start of transmission which causes enabling of peripheral data onto the POCI input line of the controller. The controller PICO output pin is enabled.

In controller mode, valid controller data is transferred to the PICO pin after one functional clock. Once both the controller and peripheral data are set, the SCLK controller clock pin goes high after an additional one-half SCLK period. The data is captured on the rising edges and propagated on the falling edges of the SCLK signal.

For a single-frame transmission after all bits of the data word are transferred, the CS line is returned to its IDLE high state one SCLK period after the last bit is captured. For continuous back-to-back transmissions, the CS signal must pulse high between each frame word transfer because the peripheral-select pin freezes the data in its serial peripheral register and does not allow altering of the data if the SPH bit is clear. The controller device must raise the CS pin of the peripheral device between each data transfer to enable the serial peripheral data write. When the continuous transfer completes, the CS pin is returned to its IDLE state one SCLK period after the last bit is captured.

**Motorola SPI Frame Format with SPO = 0 and SPH = 1**

Figure 26-5 shows the signal sequence for Motorola SPI format with SPO = 0 and SPH = 1.



Copyright © 2026 Texas Instruments Incorporated

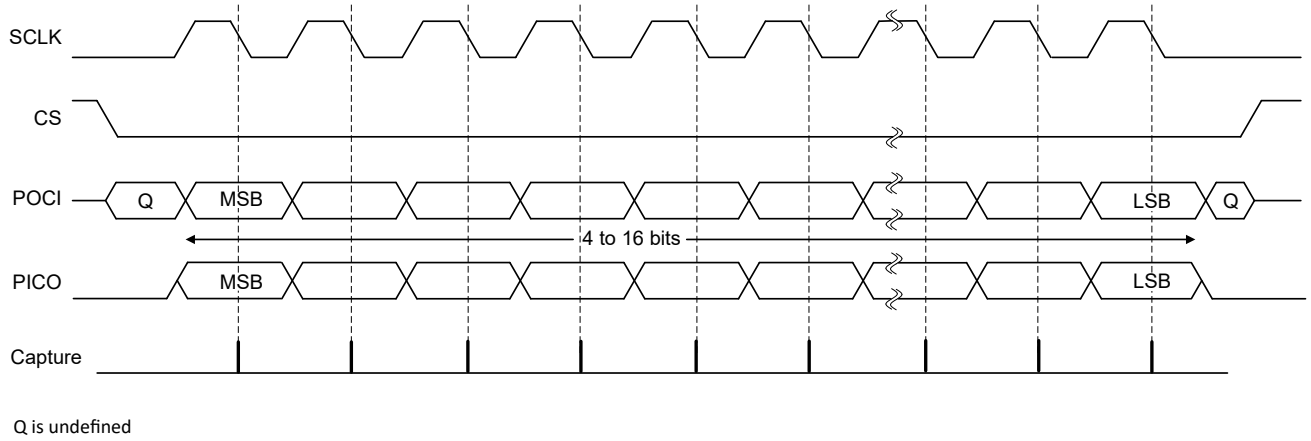



**Figure 26-5. Motorola SPI Frame Format With SPO = 0 and SPH = 1**

In this configuration, the following occurs during idle periods:

- SCLK is forced low
- CS is forced high
- The transmit data line PICO is forced low
- When the SPI is configured as a controller, it enables the SCLK pin
- When the SPI is configured as a peripheral, it disables the SCLK pin

If SPI is enabled and valid data is present in the TX FIFO, the CS controller signal goes low at the start of transmission. The controller PICO output is enabled.

For a single-frame transfer, after all bits are transferred, the CS line is returned to its IDLE high state one SCLK period after the last bit is captured. For continuous back-to-back transfers, the CS pin is held low between successive data frame and terminates like a single-word transfer.

**Motorola SPI Frame Format with SPO = 1 and SPH = 0**

Figure 26-6 shows signal sequences for Motorola SPI format with SPO = 1 and SPH = 0.

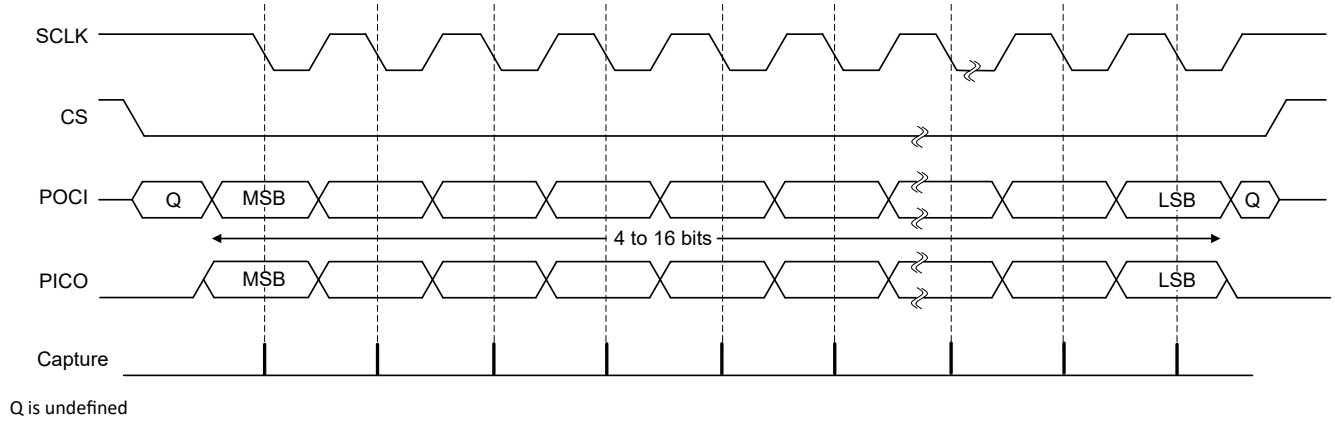



**Figure 26-6. Motorola SPI Frame Format With SPO = 1 and SPH = 0**

In this configuration, the following occurs during idle periods:

- SCLK is forced high
- CS is forced high
- The transmit data line PICO is arbitrarily forced low
- When the SPI is configured as a controller, it enables the SCLK pin
- When the SPI is configured as a peripheral, it disables the SCLK pin



Copyright © 2026 Texas Instruments Incorporated

If the SPI is enabled and valid data is in the TX FIFO, the SPI CS controller signal goes low at the start of transmission and transfers peripheral data onto the POCI line of the controller immediately. The controller PICO output pin is enabled.

After one Functional Clock later in controller mode, valid controller data is transferred to the PICO line. When both the controller and peripheral data have been set, the SCLK controller clock pin becomes low after one additional half SCLK period. Data is captured on the falling edges and propagated on the rising edges of the SCLK signal.

For a single-frame transmission after all bits of the data word are transferred, the CS line is returned to its IDLE high state one SCLK period after the last bit is captured. For continuous back-to-back transmissions, the CS signal must pulse high between each data word transfer as the peripheral-select pin freezes the data in its serial peripheral register and keeps it from being altered if the SPH bit is clear. The controller device must raise the CS pin of the peripheral device between each data transfer to enable the serial peripheral data write. When the continuous transfer completes, the CS pin returns to its IDLE state one SCLK period after the last bit is captured.

**Motorola SPI Frame Format with SPO = 1 and SPH = 1**

Figure 26-7shows the signal sequence for Motorola SPI format with SPO = 1 and SPH = 1.

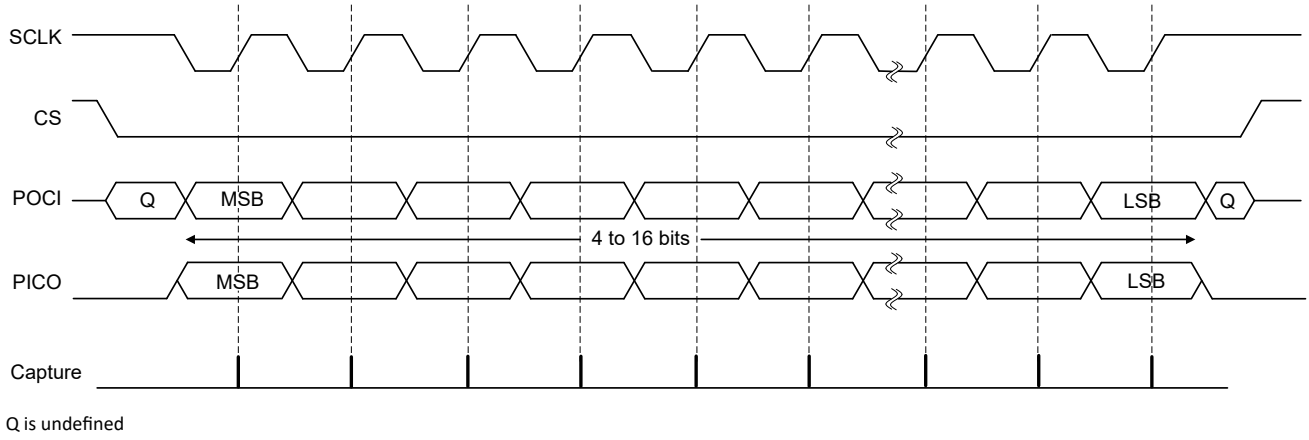



**Figure 26-7. Motorola SPI Frame Format With SPO = 1 and SPH = 1**

In this configuration, the following occurs during idle periods:

- SCLK is forced high
- CS is forced high
- The transmit data line PICO is arbitrarily forced low
- When the SPI is configured as a controller, it enables the SCLK pin
- When the SPI is configured as a peripheral, it disables the SCLK pin

If the SPI is enabled and valid data is in the TX FIFO, the start of transmission is signified by the CS controller signal going low. The controller PICO output pin is enabled. After an additional one-half SCLK period, both controller and peripheral data are enabled onto their respective transmission lines. At the same time, SCLK is enabled with a falling-edge transition. Data is then captured on the rising edges and propagated on the falling edges of the SCLK signal.

For a single word transmission, after all bits are transferred, the CS line returns to its IDLE high state one SCLK period after the last bit is captured. For continuous back-to-back transmissions, the CS pin remains in its active low state until the final bit of the last word is captured and then returns to its IDLE state. For continuous back-to-back transfers, the CS pin is held low between successive data words and terminates like a single-word transfer.



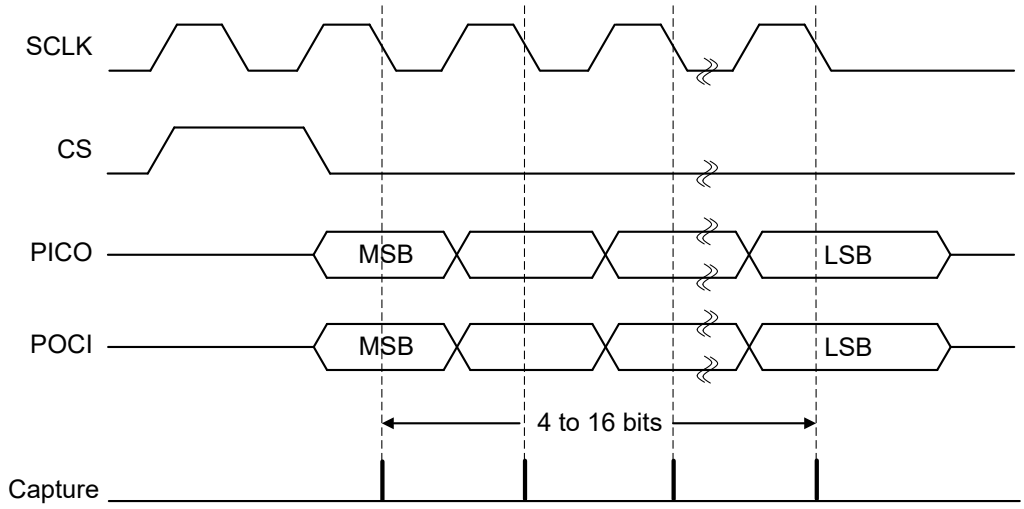
Copyright © 2026 Texas Instruments Incorporated

The serial clock (SCLK) is held inactive while the SPI is idle and SCLK transitions at the programmed frequency only during active transmission or reception of data. The IDLE state of SCLK provides a receive timeout indication that occurs when the RX FIFO still contains data after a timeout period.

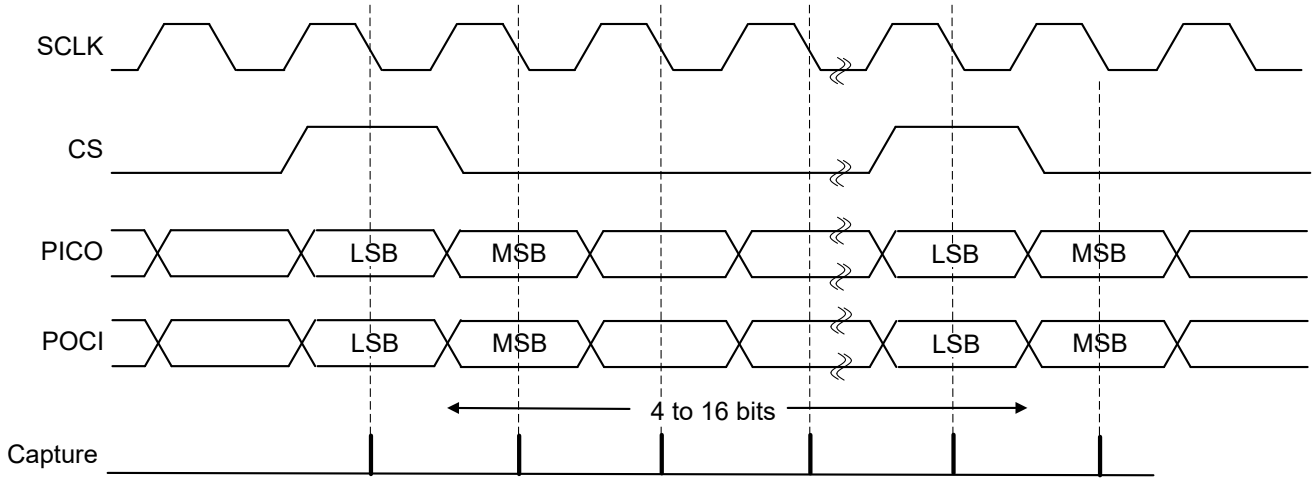
#### 26.2.3.2 Texas Instruments Synchronous Serial Frame Format

The SPI peripheral is compatible with Texas Instruments Synchronous Serial frame format.

Figure 26-8 shows the TI synchronous serial frame format for a single and continuous transmitted frame.




**Single Transmission Signal Sequences**




**Continuous Transmission Signal Sequences**

**Figure 26-8. TI Synchronous Serial Frame Format**

SCLK and CS are forced low and the transmit data line PICO is put in tristate whenever the SPI is idle. When the TX FIFO contains data, CS is pulsed high for one SCLK period. The transmitted value is also transferred from the TX FIFO to the serial shift register of the transmit logic. On the next rising edge of SCLK, the MSB of the 4- to 16-bit data frame is shifted out on the PICO pin. Likewise, the MSB of the received data is shifted onto the POCI pin by the off-chip serial peripheral device. Both the SPI and the off-chip serial peripheral device then



Copyright © 2026 Texas Instruments Incorporated

clock each data bit into their serial shifter on each falling edge of SCLK. The received data is transferred from the serial shifter to the RX FIFO on the first rising edge of SCLK after the least significant bit (LSB) is latched.

The serial clock (SCLK) is held inactive while the SPI is idle and SCLK transitions at the programmed frequency only during active transmission or reception of data. The IDLE state of SCLK provides a receive time-out indication that occurs when the RX FIFO still contains data after a time-out period.

### *26.2.4 Reset Considerations*

#### Software Reset Considerations

A Software reset can be executed with setting the RESETASSERT together with the KEY in the RSTCTL register. An ongoing transfer will be terminated immediately and can leave the software in an undefined state. Therefore, before requesting a Reset an ongoing Transfer should be terminated.

#### Hardware Reset Considerations

A hardware reset also initializes the IO configuration. This sets the IOs to a high impedance state and the data lines can float. If this is critical for the application or connected devices on the SPI interface external pull up or down resistors might be required.

### *26.2.5 Initialization*

To enable and initialize the SPI, the following steps are necessary:

1. Configure the IOMUX with the appropriate GPIO pins for which the SPI signals are multiplexed to

**Note**

Pull-ups can be used to avoid unnecessary toggles on the SPI pins, which can take the peripheral to a wrong state. In addition, if the SCLK signal is programmed to steady state High through the SPO bit in the CTL0 register, then software must also configure the GPIO port pin corresponding to the SCLK signal as a pull-up.

For each of the frame formats, the SPI is configured using the following steps:

1. Ensure that the ENABLE bit in the CTL1 register is clear before making any configuration changes.
2. Select and configure the clock prescale divisor by writing the CLKSEL and CLKDIV register.
3. Select whether the SPI is a controller or peripheral:
   - For controller operations, set the CP bit in the CTL1 register.
   - For peripheral mode, clear the CP bit in the CTL1 register.
4. Configure the clock divisor by writing the CLKCTL register.
5. Please note that a SPI Software reset ( See section 15.3.4) is required when switching SPI protocol format.
6. Configure the CTL0 and CTL1 register with based on the desired protocol, data width and other special configurations.
7. Optionally configure DMA
8. Enable the SPI by setting the ENABLE bit in the CTL1 register.

### *26.2.6 Interrupt and Events Support*

The SPI module contains three event publishers and no event subscribers. One event publisher (CPU_INT) manages SPI interrupt requests (IRQs) to the CPU subsystem through a static event route. The second and third event publishers (DMA_TRIG_RX, DMA_TRIG_TX) are used to setup the trigger signaling for the DMA through DMA event route.

The SPI events are summarized in Table 26-2.



Copyright © 2026 Texas Instruments Incorporated

**Table 26-2. SPI Events**

| Event | Type | Source | Destination | Route | Configuration | Functionality |
|---|---|---|---|---|---|---|
| CPU interrupt | Publisher | SPI | CPU Subsystem | Static route | CPU_INT registers | Fixed interrupt route from SPI to CPU |
| DMA trigger | Publisher | SPI | DMA | DMA event route | DMA_TRIG_RX registers | Fixed interrupt route from SPI RX to DMA |
| DMA trigger | Publisher | SPI | DMA | DMA event route | DMA_TRIG_TX registers | Fixed interrupt route from SPI TX to DMA |

#### 26.2.6.1 CPU Interrupt Event Publisher (CPU_INT)

The SPI module provides 9 interrupt sources that can source a CPU interrupt event. Table 26-3 lists the CPU interrupt events from the SPI in order of decreasing priority.

**Table 26-3. SPI CPU_INT Trigger Condition**

| IIDX STAT | Name | Description |
|---|---|---|
| 0x01 | RXFIFO_OVF | RXFIFO overflow event. This interrupt is set if an RX FIFO overflow has been detected. |
| 0x02 | PER | Parity error event. This bit if a Parity error has been detected. |
| 0x03 | RTOUT | Peripheral receive timeout event. When in peripheral mode and not receiving data for the CTL1.RXTIMEOUT selected number of functional clock cycles. |
| 0x04 | RX | Receive FIFO event. This interrupt is set if the selected receive FIFO level has been reached. |
| 0x05 | TX | Transmit FIFO event. This interrupt is set if the selected transmit FIFO level has been reached. |
| 0x06 | TXEMPTY | Transmit FIFO empty interrupt. This is set if all data in the transmit FIFO have been shifted out. |
| 0x07 | IDLE | SPI Idle. SPI has done finished transfers and changed into IDLE mode. This bit is set when STAT.BUSY bit goes low. |
| 0x08 | DMA_DONE1_RX | This interrupt is set if the RX DMA channel sends the DONE signal. |
| 0x09 | DMA_DONE1_TX | This interrupt is set if the TX DMA channel sends the DONE signal. |

The CPU interrupt event configuration is managed with the CPU_INT event management registers. See Section 7.2.5 for guidance on configuring the Event registers.

#### 26.2.6.2 DMA Trigger Publisher (DMA_TRIG_RX, DMA_TRIG_TX)

DMA_TRIG_RX and DMA_TRIG_TX registers are used to setup the trigger signaling for the DMA. This can be setup in a flexible way to trigger the DMA for receive or transmit events with the trigger conditions in Table 26-4 and Table 26-5.

DMA_TRIG_RX is used for triggering the DMA to do a receive data transfer and DMA_TRIG_TX is used for triggering the DMA to do a transmit data transfer.

**Table 26-4. SPI DMA_TRIG_RX DMA Trigger Condition**

| IIDX STAT | Name | Description |
|---|---|---|
| 0x03 | RTOUT | Peripheral receive timeout event. When in peripheral mode and not receiving data for the CTL1.RXTIMEOUT selected number of functional clock cycles. |
| 0x04 | RX | Receive FIFO event. This interrupt is set if the selected receive FIFO level has been reached. |



Copyright © 2026 Texas Instruments Incorporated

**Table 26-5. SPI DMA_TRIG_TX DMA Trigger Condition**

| IIDX STAT | Name | Description |
|---|---|---|
| 0x05 | TX | Transmit FIFO event. This interrupt is set if the selected transmit FIFO level has been reached. |

The DMA trigger event configuration is managed with the DMA_TRIG_RX and DMA_TRIG_TX event management registers. See Section 7.2.5 for guidance on configuring the Event registers and Section 7.1.3.2 for on how DMA trigger event works.

### *26.2.7 Emulation Modes*

The module behavior while the device is in debug mode is controlled by the FREE and SOFT bits in PDBGCTL register.

When the device is in debug mode and set into halt mode below behavior can be configured.

**Table 26-6. Debug Mode Peripheral Behavior**

| PDBGCTL.FREE | PDBGCTL.SOFT | Function |
|---|---|---|
| **1** | **x** | Modules continues operation |
| **0** | **0** | Module stops immediately |
| **0** | **1** | Module stops after the next transfer has been finished |



Copyright © 2026 Texas Instruments Incorporated

## 26.3 SPI Registers

Table 26-7 lists the memory-mapped registers for the SPI registers. All register offset addresses not listed in Table 26-7 should be considered as reserved locations and the register contents should not be modified.

**Table 26-7. SPI Registers**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 800h | PWREN | Power enable | | Go |
| 804h | RSTCTL | Reset Control | | Go |
| 808h | CLKCFG | Peripheral Clock Configuration Register | | Go |
| 814h | STAT | Status Register | | Go |
| 1000h | CLKDIV | Clock Divider | | Go |
| 1004h | CLKSEL | Clock Select for Ultra Low Power peripherals | | Go |
| 1018h | PDBGCTL | Peripheral Debug Control | | Go |
| 1020h | IIDX | Interrupt Index Register | CPU_INT | Go |
| 1028h | IMASK | Interrupt mask | CPU_INT | Go |
| 1030h | RIS | Raw interrupt status | CPU_INT | Go |
| 1038h | MIS | Masked interrupt status | CPU_INT | Go |
| 1040h | ISET | Interrupt set | CPU_INT | Go |
| 1048h | ICLR | Interrupt clear | CPU_INT | Go |
| 1050h | IIDX | Interrupt Index Register | DMA_TRIG_RX | Go |
| 1058h | IMASK | Interrupt mask | DMA_TRIG_RX | Go |
| 1060h | RIS | Raw interrupt status | DMA_TRIG_RX | Go |
| 1068h | MIS | Masked interrupt status | DMA_TRIG_RX | Go |
| 1070h | ISET | Interrupt set | DMA_TRIG_RX | Go |
| 1078h | ICLR | Interrupt clear | DMA_TRIG_RX | Go |
| 1080h | IIDX | Interrupt Index Register | DMA_TRIG_TX | Go |
| 1088h | IMASK | Interrupt mask | DMA_TRIG_TX | Go |
| 1090h | RIS | Raw interrupt status | DMA_TRIG_TX | Go |
| 1098h | MIS | Masked interrupt status | DMA_TRIG_TX | Go |
| 10A0h | ISET | Interrupt set | DMA_TRIG_TX | Go |
| 10A8h | ICLR | Interrupt clear | DMA_TRIG_TX | Go |
| 10E0h | EVT_MODE | Event Mode | | Go |
| 10E4h | INTCTL | Interrupt control register | | Go |
| 1100h | CTL0 | SPI control register 0 | | Go |
| 1104h | CTL1 | SPI control register 1 | | Go |
| 1108h | CLKCTL | Clock prescaler and divider register. | | Go |
| 110Ch | IFLS | Interrupt FIFO Level Select Register | | Go |
| 1110h | STAT | Status Register | | Go |



Copyright © 2026 Texas Instruments Incorporated

**Table 26-7. SPI Registers (continued)**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 1130h | RXDATA | RXDATA Register | | Go |
| 1140h | TXDATA | TXDATA Register | | Go |

Complex bit access types are encoded to fit into small table cells. Table 26-8 shows the codes that are used for access types in this section.

**Table 26-8. SPI Access Type Codes**

| Access Type | Code | Description |
|---|---|---|
| Read Type | | |
| R | R | Read |
| Write Type | | |
| W | W | Write |
| WK | W<br>K | Write<br>Write protected by a key |
| Reset or Default Value | | |
| *-n* | | Value after reset or the default value |



Copyright © 2026 Texas Instruments Incorporated

### 26.3.1 PWREN (Offset = 800h) [Reset = 00000000h]

PWREN is shown in Figure 26-9 and described in Table 26-9.

Return to the Summary Table.

Register to control the power state

**Figure 26-9. PWREN**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | ENABLE |
| R-0h | | | | | | | R/WK-0h |

**Table 26-9. PWREN Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | KEY to allow Power State Change<br>26h = KEY to allow write access to this register |
| 23-1 | RESERVED | R | 0h | |
| 0 | ENABLE | R/WK | 0h | Enable the power<br>KEY must be set to 26h to write to this bit.<br>0h = Disable Power<br>1h = Enable Power |



Copyright © 2026 Texas Instruments Incorporated

### 26.3.2 RSTCTL (Offset = 804h) [Reset = 00000000h]

RSTCTL is shown in Figure 26-10 and described in Table 26-10.

Return to the Summary Table.

Register to control reset assertion and de-assertion

**Figure 26-10. RSTCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | RESETSTKYCLR | RESETASSERT |
| R-0h | | | | | | WK-0h | WK-0h |

**Table 26-10. RSTCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | Unlock key<br>B1h = KEY to allow write access to this register |
| 23-2 | RESERVED | R | 0h | |
| 1 | RESETSTKYCLR | WK | 0h | Clear the RESETSTKY bit in the STAT register<br>KEY must be set to B1h to write to this bit.<br>0h = Writing 0 has no effect<br>1h = Clear reset sticky bit |
| 0 | RESETASSERT | WK | 0h | Assert reset to the peripheral<br>KEY must be set to B1h to write to this bit.<br>0h = Writing 0 has no effect<br>1h = Assert reset |



Copyright © 2026 Texas Instruments Incorporated

### 26.3.3 CLKCFG (Offset = 808h) [Reset = 00000000h]

CLKCFG is shown in Figure 26-11 and described in Table 26-11.

Return to the Summary Table.

Peripheral Clock Configuration Register

**Figure 26-11. CLKCFG**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | BLOCKASYNC |
| R-0h | | | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |

**Table 26-11. CLKCFG Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | KEY to Allow State Change -- 0xA9<br>A9h = key value to allow change field of GPRCM |
| 23-9 | RESERVED | R | 0h | |
| 8 | BLOCKASYNC | R/W | 0h | Async Clock Request is blocked from starting SYSOSC or forcing bus clock to 32MHz<br>0h = Not block async clock request<br>1h = Block async clock request |
| 7-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 26.3.4 STAT (Offset = 814h) [Reset = 00000000h]

STAT is shown in Figure 26-12 and described in Table 26-12.

Return to the Summary Table.

peripheral enable and reset status

**Figure 26-12. STAT**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | RESETSTKY |
| R-0h | | | | | | | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |

**Table 26-12. STAT Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-17 | RESERVED | R | 0h | |
| 16 | RESETSTKY | R | 0h | This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register<br>0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register<br>1h = The peripheral was reset since the last bit clear |
| 15-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 26.3.5 CLKDIV (Offset = 1000h) [Reset = 00000000h]

CLKDIV is shown in Figure 26-13 and described in Table 26-13.

Return to the Summary Table.

This register is used to specify module-specific divide ratio of the functional clock

**Figure 26-13. CLKDIV**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | | RATIO | | |
| R-0h | | | | | | | | | | | | | R/W-0h | | |

**Table 26-13. CLKDIV Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | RESERVED | R | 0h | |
| 2-0 | RATIO | R/W | 0h | Selects divide ratio of module clock<br>0h = Do not divide clock source<br>1h = Divide clock source by 2<br>2h = Divide clock source by 3<br>3h = Divide clock source by 4<br>4h = Divide clock source by 5<br>5h = Divide clock source by 6<br>6h = Divide clock source by 7<br>7h = Divide clock source by 8 |



Copyright © 2026 Texas Instruments Incorporated

### 26.3.6 CLKSEL (Offset = 1004h) [Reset = 00000000h]

CLKSEL is shown in Figure 26-14 and described in Table 26-14.

Return to the Summary Table.

Clock source selection for peripherals

**Figure 26-14. CLKSEL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | SYSCLK_SEL | MFCLK_SEL | LFCLK_SEL | RESERVED |
| R-0h | | | | R/W-0h | R/W-0h | R/W-0h | R-0h |

**Table 26-14. CLKSEL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | SYSCLK_SEL | R/W | 0h | Selects SYSCLK as clock source if enabled<br>0h = Does not select this clock as a source<br>1h = Select this clock as a source |
| 2 | MFCLK_SEL | R/W | 0h | Selects MFCLK as clock source if enabled<br>0h = Does not select this clock as a source<br>1h = Select this clock as a source |
| 1 | LFCLK_SEL | R/W | 0h | Selects LFCLK as clock source if enabled<br>0h = Does not select this clock as a source<br>1h = Select this clock as a source |
| 0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 26.3.7 PDBGCTL (Offset = 1018h) [Reset = 00000000h]

PDBGCTL is shown in Figure 26-15 and described in Table 26-15.

Return to the Summary Table.

This register can be used by the software developer to control the behavior of the peripheral relative to the 'Core Halted' input

**Figure 26-15. PDBGCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | SOFT | FREE |
| R-0h | | | | | | R/W-0h | R/W-0h |

**Table 26-15. PDBGCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | |
| 1 | SOFT | R/W | 1h | Soft halt boundary control. This function is only available, if FREE is set to 'STOP'<br>0h = The peripheral will halt immediately, even if the resultant state will result in corruption if the system is restarted<br>1h = The peripheral blocks the debug freeze until it has reached a boundary where it can resume without corruption |
| 0 | FREE | R/W | 1h | Free run control<br>0h = The peripheral freezes functionality while the Core Halted input is asserted and resumes when it is deasserted.<br>1h = The peripheral ignores the state of the Core Halted input |



Copyright © 2026 Texas Instruments Incorporated

### 26.3.8 IIDX (Offset = 1020h) [Reset = 00000000h]

IIDX is shown in Figure 26-16 and described in Table 26-16.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. Value 0x00 means no event pending. Interrupt 1 is the highest priority, IIDX next highest, 4, 8, … IIDX^31 is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred. On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in [RIS] and [MIS] are cleared as well. After a read from the CPU (not from the debug interface), the register is updated with the next highest priority interrupt, if none are pending, then it should display 0x0.

**Figure 26-16. IIDX**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | STAT | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | |

**Table 26-16. IIDX Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | STAT | R | 0h | Interrupt index status<br>00h = No interrupt pending<br>1h = RX FIFO Overflow Event/interrupt pending<br>2h = Transmit Parity Event/interrupt pending<br>3h = SPI receive time-out interrupt<br>4h = Receive Event/interrupt pending<br>5h = Transmit Event/interrupt pending<br>6h = Transmit Buffer Empty Event/interrupt pending<br>7h = End of Transmit Event/interrupt pending<br>8h = DMA Done for Receive Event/interrupt pending<br>9h = DMA Done for Transmit Event/interrupt pending<br>Ah = TX FIFO underflow interrupt<br>Bh = RX FIFO Full Interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 26.3.9 IMASK (Offset = 1028h) [Reset = 00000000h]

IMASK is shown in Figure 26-17 and described in Table 26-17.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

**Figure 26-17. IMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | RXFULL | TXFIFO_UNF | DMA_DONE_TX |
| R-0h | | | | | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DMA_DONE_RX | IDLE | TXEMPTY | TX | RX | RTOUT | PER | RXFIFO_OVF |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

**Table 26-17. IMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-11 | RESERVED | R | 0h | |
| 10 | RXFULL | R/W | 0h | RX FIFO Full Interrupt Mask<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 9 | TXFIFO_UNF | R/W | 0h | TX FIFO underflow interrupt mask<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 8 | DMA_DONE_TX | R/W | 0h | DMA Done 1 event for TX event mask.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 7 | DMA_DONE_RX | R/W | 0h | DMA Done 1 event for RX event mask.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 6 | IDLE | R/W | 0h | SPI Idle event mask.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 5 | TXEMPTY | R/W | 0h | Transmit FIFO Empty event mask.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 4 | TX | R/W | 0h | Transmit FIFO event mask.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 3 | RX | R/W | 0h | Receive FIFO event.This interrupt is set if the selected Receive FIFO level has been reached<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 2 | RTOUT | R/W | 0h | Enable SPI Receive Time-Out event mask.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |



Copyright © 2026 Texas Instruments Incorporated

**Table 26-17. IMASK Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 1 | PER | R/W | 0h | Parity error event mask.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 0 | RXFIFO_OVF | R/W | 0h | RXFIFO overflow event mask.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |



Copyright © 2026 Texas Instruments Incorporated

### 26.3.10 RIS (Offset = 1030h) [Reset = 00000000h]

RIS is shown in Figure 26-18 and described in Table 26-18.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Figure 26-18. RIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | RXFULL | TXFIFO_UNF | DMA_DONE_TX |
| R-0h | | | | | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DMA_DONE_RX | IDLE | TXEMPTY | TX | RX | RTOUT | PER | RXFIFO_OVF |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 26-18. RIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-11 | RESERVED | R | 0h | |
| 10 | RXFULL | R | 0h | RX FIFO Full Interrupt<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 9 | TXFIFO_UNF | R | 0h | TX FIFO Underflow Interrupt<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 8 | DMA_DONE_TX | R | 0h | DMA Done 1 event for TX. This interrupt is set if the TX DMA channel sends the DONE signal. This allows the handling of the DMA event inside the mapped peripheral.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 7 | DMA_DONE_RX | R | 0h | DMA Done 1 event for RX. This interrupt is set if the RX DMA channel sends the DONE signal. This allows the handling of the DMA event inside the mapped peripheral.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 6 | IDLE | R | 0h | SPI has done finished transfers and changed into IDLE mode. This bit is set when BUSY goes low.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 5 | TXEMPTY | R | 0h | Transmit FIFO Empty interrupt mask. This interrupt is set if all data in the Transmit FIFO have been move to the shift register.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 4 | TX | R | 0h | Transmit FIFO event..This interrupt is set if the selected Transmit FIFO level has been reached.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

**Table 26-18. RIS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 3 | RX | R | 0h | Receive FIFO event.This interrupt is set if the selected Receive FIFO level has been reached<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 2 | RTOUT | R | 0h | SPI Receive Time-Out event.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 1 | PER | R | 0h | Parity error event: this bit is set if a Parity error has been detected<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 0 | RXFIFO_OVF | R | 0h | RXFIFO overflow event. This interrupt is set if an RX FIFO overflow has been detected.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

### 26.3.11 MIS (Offset = 1038h) [Reset = 00000000h]

MIS is shown in Figure 26-19 and described in Table 26-19.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 26-19. MIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | RXFULL | TXFIFO_UNF | DMA_DONE_TX |
| R-0h | | | | | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DMA_DONE_RX | IDLE | TXEMPTY | TX | RX | RTOUT | PER | RXFIFO_OVF |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 26-19. MIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-11 | RESERVED | R | 0h | |
| 10 | RXFULL | R | 0h | RX FIFO Full Interrupt<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 9 | TXFIFO_UNF | R | 0h | TX FIFO underflow interrupt<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 8 | DMA_DONE_TX | R | 0h | Masked DMA Done 1 event for TX.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 7 | DMA_DONE_RX | R | 0h | Masked DMA Done 1 event for RX.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 6 | IDLE | R | 0h | Masked SPI IDLE mode event.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 5 | TXEMPTY | R | 0h | Masked Transmit FIFO Empty event.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 4 | TX | R | 0h | Masked Transmit FIFO event. This interrupt is set if the selected Transmit FIFO level has been reached.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 3 | RX | R | 0h | Masked receive FIFO event.This interrupt is set if the selected Receive FIFO level has been reached<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 2 | RTOUT | R | 0h | Masked SPI Receive Time-Out Interrupt.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |



Copyright © 2026 Texas Instruments Incorporated

**Table 26-19. MIS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 1 | PER | R | 0h | Masked Parity error event: this bit if a Parity error has been detected<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 0 | RXFIFO_OVF | R | 0h | Masked RXFIFO overflow event. This interrupt is set if an RX FIFO overflow has been detected.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

### 26.3.12 ISET (Offset = 1040h) [Reset = 00000000h]

ISET is shown in Figure 26-20 and described in Table 26-20.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 26-20. ISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | RXFULL | TXFIFO_UNF | DMA_DONE_TX |
| R-0h | | | | | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DMA_DONE_RX | IDLE | TXEMPTY | TX | RX | RTOUT | PER | RXFIFO_OVF |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

**Table 26-20. ISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-11 | RESERVED | R | 0h | |
| 10 | RXFULL | W | 0h | Set RX FIFO Full Event<br>0h = Writing has no effect<br>1h = Set Interrupt |
| 9 | TXFIFO_UNF | W | 0h | Set TX FIFO Underflow Event<br>0h = Writing has no effect<br>1h = Set interrupt |
| 8 | DMA_DONE_TX | W | 0h | Set DMA Done 1 event for TX.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 7 | DMA_DONE_RX | W | 0h | Set DMA Done 1 event for RX.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 6 | IDLE | W | 0h | Set SPI IDLE mode event.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 5 | TXEMPTY | W | 0h | Set Transmit FIFO Empty event.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 4 | TX | W | 0h | Set Transmit FIFO event.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 3 | RX | W | 0h | Set Receive FIFO event.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 2 | RTOUT | W | 0h | Set SPI Receive Time-Out Event.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt Mask |



Copyright © 2026 Texas Instruments Incorporated

**Table 26-20. ISET Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 1 | PER | W | 0h | Set Parity error event.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 0 | RXFIFO_OVF | W | 0h | Set RXFIFO overflow event.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 26.3.13 ICLR (Offset = 1048h) [Reset = 00000000h]

ICLR is shown in Figure 26-21 and described in Table 26-21.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

**Figure 26-21. ICLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | RXFULL | TXFIFO_UNF | DMA_DONE_TX |
| R-0h | | | | | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DMA_DONE_RX | IDLE | TXEMPTY | TX | RX | RTOUT | PER | RXFIFO_OVF |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

**Table 26-21. ICLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-11 | RESERVED | R | 0h | |
| 10 | RXFULL | W | 0h | Clear RX FIFO underflow event<br>0h = Writing has no effect<br>1h = Clear interrupt |
| 9 | TXFIFO_UNF | W | 0h | Clear TXFIFO underflow event<br>0h = Writing has no effect<br>1h = Clear interrupt |
| 8 | DMA_DONE_TX | W | 0h | Clear DMA Done 1 event for TX.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 7 | DMA_DONE_RX | W | 0h | Clear DMA Done 1 event for RX.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 6 | IDLE | W | 0h | Clear SPI IDLE mode event.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 5 | TXEMPTY | W | 0h | Clear Transmit FIFO Empty event.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 4 | TX | W | 0h | Clear Transmit FIFO event.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 3 | RX | W | 0h | Clear Receive FIFO event.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 2 | RTOUT | W | 0h | Clear SPI Receive Time-Out Event.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt Mask |



Copyright © 2026 Texas Instruments Incorporated

**Table 26-21. ICLR Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 1 | PER | W | 0h | Clear Parity error event.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 0 | RXFIFO_OVF | W | 0h | Clear RXFIFO overflow event.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 26.3.14 IIDX (Offset = 1050h) [Reset = 00000000h]

IIDX is shown in Figure 26-22 and described in Table 26-22.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. Value 0x00 means no event pending. Interrupt 1 is the highest priority, IIDX next highest, 4, 8, … IIDX^31 is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred. On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in [RIS] and [MIS] are cleared as well. After a read from the CPU (not from the debug interface), the register is updated with the next highest priority interrupt, if none are pending, then it should display 0x0.

**Figure 26-22. IIDX**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | STAT | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | |

**Table 26-22. IIDX Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | STAT | R | 0h | Interrupt index status<br>00h = No interrupt pending<br>3h = SPI receive time-out interrupt<br>4h = Receive Event/interrupt pending |



Copyright © 2026 Texas Instruments Incorporated

### 26.3.15 IMASK (Offset = 1058h) [Reset = 00000000h]

IMASK is shown in Figure 26-23 and described in Table 26-23.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

**Figure 26-23. IMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | RX | RTOUT | RESERVED | |
| R-0h | | | | R/W-0h | R/W-0h | R-0h | |

**Table 26-23. IMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | RX | R/W | 0h | Receive FIFO event mask.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 2 | RTOUT | R/W | 0h | SPI Receive Time-Out event mask.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 1-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 26.3.16 RIS (Offset = 1060h) [Reset = 00000000h]

RIS is shown in Figure 26-24 and described in Table 26-24.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Figure 26-24. RIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | RX | RTOUT | RESERVED | |
| R-0h | | | | R-0h | R-0h | R-0h | |

**Table 26-24. RIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | RX | R | 0h | Receive FIFO event.This interrupt is set if the selected Receive FIFO level has been reached<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 2 | RTOUT | R | 0h | SPI Receive Time-Out Event.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 1-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 26.3.17 MIS (Offset = 1068h) [Reset = 00000000h]

MIS is shown in Figure 26-25 and described in Table 26-25.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 26-25. MIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | RX | RTOUT | RESERVED | |
| R-0h | | | | R-0h | R-0h | R-0h | |

**Table 26-25. MIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | RX | R | 0h | Receive FIFO event mask.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 2 | RTOUT | R | 0h | SPI Receive Time-Out event mask.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 1-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 26.3.18 ISET (Offset = 1070h) [Reset = 00000000h]

ISET is shown in Figure 26-26 and described in Table 26-26.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 26-26. ISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | RX | RTOUT | RESERVED | |
| R-0h | | | | W-0h | W-0h | R-0h | |

**Table 26-26. ISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | RX | W | 0h | Set Receive FIFO event.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 2 | RTOUT | W | 0h | Set SPI Receive Time-Out event.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt Mask |
| 1-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 26.3.19 ICLR (Offset = 1078h) [Reset = 00000000h]

ICLR is shown in Figure 26-27 and described in Table 26-27.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

**Figure 26-27. ICLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | RX | RTOUT | RESERVED | |
| R-0h | | | | W-0h | W-0h | R-0h | |

**Table 26-27. ICLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | RX | W | 0h | Clear Receive FIFO event.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 2 | RTOUT | W | 0h | Clear SPI Receive Time-Out event.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt Mask |
| 1-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 26.3.20 IIDX (Offset = 1080h) [Reset = 00000000h]

IIDX is shown in Figure 26-28 and described in Table 26-28.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. Value 0x00 means no event pending. Interrupt 1 is the highest priority, IIDX next highest, 4, 8, … IIDX^31 is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred. On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in [RIS] and [MIS] are cleared as well. After a read from the CPU (not from the debug interface), the register is updated with the next highest priority interrupt, if none are pending, then it should display 0x0.

**Figure 26-28. IIDX**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | STAT | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | |

**Table 26-28. IIDX Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | STAT | R | 0h | Interrupt index status<br>00h = No interrupt pending<br>5h = Transmit Event/interrupt pending |



Copyright © 2026 Texas Instruments Incorporated

### 26.3.21 IMASK (Offset = 1088h) [Reset = 00000000h]

IMASK is shown in Figure 26-29 and described in Table 26-29.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

**Figure 26-29. IMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | TX | RESERVED | | | |
| R-0h | | | | | | | | | | | R/W-0h | R-0h | | | |

**Table 26-29. IMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-5 | RESERVED | R | 0h | |
| 4 | TX | R/W | 0h | Transmit FIFO event mask.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 3-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 26.3.22 RIS (Offset = 1090h) [Reset = 00000000h]

RIS is shown in Figure 26-30 and described in Table 26-30.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Figure 26-30. RIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | TX | RESERVED | | | |
| R-0h | | | | | | | | | | | R-0h | R-0h | | | |

**Table 26-30. RIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-5 | RESERVED | R | 0h | |
| 4 | TX | R | 0h | Transmit FIFO event: A read returns the current mask for transmit FIFO interrupt. On a write of 1, the mask for transmit FIFO interrupt is set which means the interrupt state will be reflected in MIS.TXMIS. A write of 0 clears the mask which means MIS.TXMIS will not reflect the interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 3-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 26.3.23 MIS (Offset = 1098h) [Reset = 00000000h]

MIS is shown in Figure 26-31 and described in Table 26-31.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 26-31. MIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | TX | RESERVED | | | |
| R-0h | | | | | | | | | | | R-0h | R-0h | | | |

**Table 26-31. MIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-5 | RESERVED | R | 0h | |
| 4 | TX | R | 0h | Masked Transmit FIFO event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 3-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 26.3.24 ISET (Offset = 10A0h) [Reset = 00000000h]

ISET is shown in Figure 26-32 and described in Table 26-32.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 26-32. ISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | TX | RESERVED | | | |
| R-0h | | | | | | | | | | | W-0h | R-0h | | | |

**Table 26-32. ISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-5 | RESERVED | R | 0h | |
| 4 | TX | W | 0h | Set Transmit FIFO event.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 3-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 26.3.25 ICLR (Offset = 10A8h) [Reset = 00000000h]

ICLR is shown in Figure 26-33 and described in Table 26-33.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

**Figure 26-33. ICLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | TX | RESERVED | | | |
| R-0h | | | | | | | | | | | W-0h | R-0h | | | |

**Table 26-33. ICLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-5 | RESERVED | R | 0h | |
| 4 | TX | W | 0h | Clear Transmit FIFO event.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 3-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 26.3.26 EVT_MODE (Offset = 10E0h) [Reset = 00000000h]

EVT_MODE is shown in Figure 26-34 and described in Table 26-34.

Return to the Summary Table.

Event mode register. It is used to select whether each line is disabled, in software mode (software clears the RIS) or in hardware mode (hardware clears the RIS)

**Figure 26-34. EVT_MODE**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | INT2_CFG | | INT1_CFG | | INT0_CFG | |
| R-0h | | R-0h | | R-0h | | R-0h | |

**Table 26-34. EVT_MODE Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | RESERVED | R | 0h | |
| 5-4 | INT2_CFG | R | 2h | Event line mode select for event corresponding to *none.DMA_TRIG_TX*<br>0h = The interrupt or event line is disabled.<br>1h = The interrupt or event line is in software mode. Software must clear the RIS.<br>2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag. |
| 3-2 | INT1_CFG | R | 2h | Event line mode select for event corresponding to *none.DMA_TRIG_RX*<br>0h = The interrupt or event line is disabled.<br>1h = The interrupt or event line is in software mode. Software must clear the RIS.<br>2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag. |
| 1-0 | INT0_CFG | R | 1h | Event line mode select for event corresponding to *none.CPU_INT*<br>0h = The interrupt or event line is disabled.<br>1h = The interrupt or event line is in software mode. Software must clear the RIS.<br>2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag. |



Copyright © 2026 Texas Instruments Incorporated

### 26.3.27 INTCTL (Offset = 10E4h) [Reset = 00000000h]

INTCTL is shown in Figure 26-35 and described in Table 26-35.

Return to the Summary Table.

Interrupt control register

**Figure 26-35. INTCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | INTEVAL |
| R-0h | | | | | | | W-0h |

**Table 26-35. INTCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | INTEVAL | W | 0h | Writing a 1 to this field re-evaluates the interrupt sources.<br>0h = The interrupt or event line is disabled.<br>1h = The interrupt or event line is in software mode. Software must clear the RIS. |



Copyright © 2026 Texas Instruments Incorporated

### 26.3.28 CTL0 (Offset = 1100h) [Reset = 00000000h]

CTL0 is shown in Figure 26-36 and described in Table 26-36.

Return to the Summary Table.

SPI control register 0

**Figure 26-36. CTL0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | CSCLR | CSSEL | | RESERVED | | SPH | SPO |
| R-0h | R/W-0h | R/W-0h | | R-0h | | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PACKEN | FRF | | DSS | | | | |
| R/W-0h | R/W-0h | | R/W-0h | | | | |

**Table 26-36. CTL0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-15 | RESERVED | R | 0h | |
| 14 | CSCLR | R/W | 0h | Clear shift register counter on CS inactive<br>This bit is relevant only in the peripheral, CTL1.CP=0.<br>0h = Disable automatic clear of shift register when CS goes to disable.<br>1h = Enable automatic clear of shift register when CS goes to disable. |
| 13-12 | CSSEL | R/W | 0h | Select the CS line to control on data transfer<br>This bit is applicable for both controller/target mode<br>0h (R/W) = CS line select: 0<br>1h (R/W) = CS line select: 1<br>2h (R/W) = CS line select: 2<br>3h (R/W) = CS line select: 3 |
| 11-10 | RESERVED | R | 0h | |
| 9 | SPH | R/W | 0h | CLKOUT phase (Motorola SPI frame format only)<br>This bit selects the clock edge that captures data and enables it to change state. It<br>has the most impact on the first bit transmitted by either permitting or not permitting a clock transition before the first data capture edge.<br>0h = Data is captured on the first clock edge transition.<br>1h = Data is captured on the second clock edge transition. |
| 8 | SPO | R/W | 0h | CLKOUT polarity (Motorola SPI frame format only)<br>0h = SPI produces a steady state LOW value on the CLKOUT<br>1h = SPI produces a steady state HIGH value on the CLKOUT |
| 7 | PACKEN | R/W | 0h | Packing Enable.<br>When 1, packing feature is enabled inside the IP<br>When 0, packing feature is disabled inside the IP<br>0h = Packing feature disabled<br>1h = Packing feature enabled |



Copyright © 2026 Texas Instruments Incorporated

**Table 26-36. CTL0 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 6-5 | FRF | R/W | 0h | Frame format Select<br>0h = Motorola SPI frame format (3 wire mode)<br>1h = Motorola SPI frame format (4 wire mode)<br>2h = TI synchronous serial frame format<br>3h = National Microwire frame format |
| 4-0 | DSS | R/W | 0h | Data Size Select.<br>Values 0 - 2 are reserved and shall not be used.<br>3h = 4_BIT : 4-bit data<br>SPI allows only values up to 16 Bit<br>3h (R/W) = Data Size Select bits: 4<br>4h (R/W) = Data Size Select bits: 5<br>5h (R/W) = Data Size Select bits: 6<br>6h (R/W) = Data Size Select bits: 7<br>7h (R/W) = Data Size Select bits: 8<br>8h (R/W) = Data Size Select bits: 9<br>9h (R/W) = Data Size Select bits: 10<br>Ah (R/W) = Data Size Select bits: 11<br>Bh (R/W) = Data Size Select bits: 12<br>Ch (R/W) = Data Size Select bits: 13<br>Dh (R/W) = Data Size Select bits: 14<br>Eh (R/W) = Data Size Select bits: 15<br>Fh (R/W) = Data Size Select bits: 16 |



Copyright © 2026 Texas Instruments Incorporated

### 26.3.29 CTL1 (Offset = 1104h) [Reset = 00000004h]

CTL1 is shown in Figure 26-37 and described in Table 26-37.

Return to the Summary Table.

SPI control register 1

**Figure 26-37. CTL1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | RXTIMEOUT | | | | | |
| R-0h | | R/W-0h | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| REPEATTX | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CDMODE | | | | CDENABLE | RESERVED | | PTEN |
| R/W-0h | | | | R/W-0h | R-0h | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | PES | PREN | MSB | POD | CP | LBM | ENABLE |
| R-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-1h | R/W-0h | R/W-0h |

**Table 26-37. CTL1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | RESERVED | R | 0h | |
| 29-24 | RXTIMEOUT | R/W | 0h | Receive Timeout (only for Peripheral mode) Defines the number of Clock Cycles before after which the Receive Timeout flag RTOUT is set. The time is calculated using the control register for the clock selection and divider in the Controller mode configuration. A value of 0 disables this function.<br>0h = Smallest value<br>3Fh = Highest possible value |
| 23-16 | REPEATTX | R/W | 0h | Counter to repeat last transfer 0: repeat last transfer is disabled. x: repeat the last transfer with the given number. The transfer will be started with writing a data into the TX Buffer. Sending the data will be repeated with the given value, so the data will be transferred X+1 times in total. The behavior is identical as if the data would be written into the TX Buffer that many times as defined by the value here. It can be used to clean a transfer or to pull a certain amount of data by a peripheral.<br>0h = Smallest value<br>FFh = Highest possible value |
| 15-12 | CDMODE | R/W | 0h | Command/Data Mode Value<br>When CTL1.CDENABLE is 1, CS3 line is used as C/D signal to distinguish between Command (C/D low) and Data (C/D high) information.<br>When a value is written into the CTL1.CDMODE bits, the C/D (CS3) line will go low for the given numbers of byte which are sent by the SPI, starting with the next value to be transmitted after which, C/D line will go high automatically<br>0: Manual mode with C/D signal as High<br>1-14: C/D is low while this number of bytes are being sent after which, this field sets to 0 and C/D goes high. Reading this field at any time returns the remaining number of command bytes.<br>15: Manual mode with C/D signal as Low.<br>0h = Manual mode: Data<br>0h = Smallest value<br>Fh = Manual mode: Command |



Copyright © 2026 Texas Instruments Incorporated

**Table 26-37. CTL1 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 11 | CDENABLE | R/W | 0h | Command/Data Mode enable<br>0h = CS3 is used for Chip Select<br>1h = CS3 is used as CD signal |
| 10-9 | RESERVED | R | 0h | |
| 8 | PTEN | R/W | 0h | Parity transmit enable<br>If enabled, parity transmission will be done for both controller and peripheral modes.<br>0h = Parity transmission is disabled<br>1h = Parity transmission is enabled |
| 7 | RESERVED | R | 0h | |
| 6 | PES | R/W | 0h | Even Parity Select<br>0h = Odd Parity mode<br>1h = Even Parity mode |
| 5 | PREN | R/W | 0h | Parity receive enable<br>If enabled, parity reception check will be done for both controller and peripheral modes<br>In case of a parity miss-match the parity error flag RIS.PER will be set.<br>0h = Disable Parity receive function<br>1h = Enable Parity receive function |
| 4 | MSB | R/W | 0h | MSB first select. Controls the direction of the receive and transmit shift register.<br>0h = LSB first<br>1h = MSB first |
| 3 | POD | R/W | 0h | Peripheral-mode: Data output disabled<br>This bit is relevant only in Peripheral mode. In multiple-peripheral system topologies, SPI controller can broadcast a message to all peripherals, while only one peripheral drives the line.<br>POD can be used by the SPI peripheral to disable driving data on the line.<br>0h = SPI can drive the MISO output in peripheral mode.<br>1h = SPI cannot drive the MISO output in peripheral mode. |
| 2 | CP | R/W | 1h | Controller or peripheral mode select. This bit can be modified only when SPI is disabled, CTL1.ENABLE=0.<br>0h = Select Peripheral mode<br>1h = Select Controller Mode |
| 1 | LBM | R/W | 0h | Loop back mode<br>0h = Disable loopback mode<br>1h = Enable loopback mode |
| 0 | ENABLE | R/W | 0h | SPI enable<br>0h = Disable module function<br>1h = Enable module function |



Copyright © 2026 Texas Instruments Incorporated

### 26.3.30 CLKCTL (Offset = 1108h) [Reset = 00000000h]

CLKCTL is shown in Figure 26-38 and described in Table 26-38.

Return to the Summary Table.

Clock prescaler and divider register. This register contains the settings for the Clock prescaler and divider settings.

**Figure 26-38. CLKCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DSAMPLE | | | | RESERVED | | | | | | | | | | | |
| R/W-0h | | | | R-0h | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | SCR | | | | | | | | | |
| R-0h | | | | | | R/W-0h | | | | | | | | | |

**Table 26-38. CLKCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-28 | DSAMPLE | R/W | 0h | Delayed sampling value.<br>In controller mode the data on the input pin will be delayed sampled by the defined clock cycles of internal functional clock hence relaxing the setup time of input data. This setting is useful in systems where the board delays and external peripheral delays are more than the input setup time of the controller. Please refer to the datasheet for values of controller input setup time and assess what DSAMPLE value meets the requirement of the system.<br>Note: High values of DSAMPLE can cause HOLD time violations and must be factored in the calculations.<br>0h = Smallest value<br>Fh = Highest possible value |
| 27-10 | RESERVED | R | 0h | |
| 9-0 | SCR | R/W | 0h | Serial clock divider: This is used to generate the transmit and receive bit rate of the SPI. The SPI bit rate is (SPI's functional clock frequency)/((SCR+1)*2). SCR is a value from 0-1023.<br>0h = Smallest value<br>3FFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 26.3.31 IFLS (Offset = 110Ch) [Reset = 00000012h]

IFLS is shown in Figure 26-39 and described in Table 26-39.

Return to the Summary Table.

The IFLS register is the interrupt FIFO level select register. You can use this register to define the levels at which the TX, RX and timeout interrupt flags are triggered. The interrupts are generated based on a transition through a level rather than being based on the level. That is, the interrupts are generated when the fill level progresses through the trigger level. For example, if the receive trigger level is set to the half-way mark, the interrupt is triggered when the receive FIFO is filled with two or more characters. Out of reset, the TXIFLSEL and RXIFLSEL bits are configured so that the FIFOs trigger an interrupt at the half-way mark.

**Figure 26-39. IFLS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | RXIFLSEL | | | TXIFLSEL | | |
| R-0h | | | | | | | | | | R/W-2h | | | R/W-2h | | |

**Table 26-39. IFLS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | RESERVED | R | 0h | |
| 5-3 | RXIFLSEL | R/W | 2h | SPI Receive Interrupt FIFO Level Select The trigger points for the receive interrupt are as follows:<br>0h = Reserved<br>1h = RX FIFO >= 1/4 full<br>2h = RX FIFO >= 1/2 full (default)<br>3h = RX FIFO >= 3/4 full<br>4h = Reserved<br>5h = RX FIFO is full<br>6h = Reserved<br>7h = Trigger when RX FIFO contains >= 1 frame |
| 2-0 | TXIFLSEL | R/W | 2h | SPI Transmit Interrupt FIFO Level Select The trigger points for the transmit interrupt are as follows:<br>0h = Reserved<br>1h = TX FIFO <= 3/4 empty<br>2h = TX FIFO <= 1/2 empty (default)<br>3h = TX FIFO <= 1/4 empty<br>4h = Reserved<br>5h = TX FIFO is empty<br>6h = Reserved<br>7h = Trigger when TX FIFO has >= 1 frame free. |



Copyright © 2026 Texas Instruments Incorporated

### 26.3.32 STAT (Offset = 1110h) [Reset = 00000000h]

STAT is shown in Figure 26-40 and described in Table 26-40.

Return to the Summary Table.

Status Register

**Figure 26-40. STAT**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | BUSY | RNF | RFE | TNF | TFE |
| R-0h | | | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 26-40. STAT Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-5 | RESERVED | R | 0h | |
| 4 | BUSY | R | 0h | Busy<br>0h = SPI is in idle mode.<br>1h = SPI is currently transmitting and/or receiving data, or transmit FIFO is not empty. |
| 3 | RNF | R | 1h | Receive FIFO not full<br>0h = Receive FIFO is full.<br>1h = Receive FIFO is not full. |
| 2 | RFE | R | 1h | Receive FIFO empty.<br>0h = Receive FIFO is not empty.<br>1h = Receive FIFO is empty. |
| 1 | TNF | R | 1h | Transmit FIFO not full<br>0h = Transmit FIFO is full.<br>1h = Transmit FIFO is not full. |
| 0 | TFE | R | 1h | Transmit FIFO empty.<br>0h = Transmit FIFO is not empty.<br>1h = Transmit FIFO is empty. |



Copyright © 2026 Texas Instruments Incorporated

### 26.3.33 RXDATA (Offset = 1130h) [Reset = 00000000h]

RXDATA is shown in Figure 26-41 and described in Table 26-41.

Return to the Summary Table.

RXDATA Register
Reading this register returns value(s) of FIFO. If the FIFO is empty the last read value is returned.
Writing has not effect and is ignored.
When PACKEN=1,two entries of the FIFO are returned as a 32-bit value. When PACKEN=0, 1 entry of FIFO is returned as 16-bit value.

**Figure 26-41. RXDATA**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 26-41. RXDATA Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | R | 0h | Received Data<br>When PACKEN=1,two entries of the FIFO are returned as a 32-bit value. When PACKEN=0, 1 entry of FIFO is returned as 16-bit value.<br>As data values are removed by the receive logic from the incoming data frame, they are placed into the entry in the receive FIFO, pointed to by the current FIFO write pointer.<br>Received data less than 16 bits is automatically right justified in the receive buffer.<br>0h = Smallest value<br>FFFFFFFFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 26.3.34 TXDATA (Offset = 1140h) [Reset = 00000000h]

TXDATA is shown in Figure 26-42 and described in Table 26-42.

Return to the Summary Table.

TXDATA Register
Writing puts the data into the TX FIFO. Reading this register returns the last written value.
When PACKEN=0, only the lower 16-bits of data written into the register is transferred to one 16-bits wide TX FIFO entry
When PACKEN=1, upper and lower 16-bits of 32-bit write data are transferred to two16-bits wide TX FIFO entry

**Figure 26-42. TXDATA**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 26-42. TXDATA Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | R/W | 0h | Transmit Data<br>When read, last written value will be returned. If the last write to this field was a 32-bit write (with PACKEN=1), 32-bits will be returned and if the last write was a 16-bit write (PACKEN=0), those 16-bits will be returned.<br>When written, one or two FIFO entries will be written depending on PACKEN value. Data values are removed from the transmit FIFO one value at a time by the transmit logic. It is loaded into the transmit serial shifter, then serially shifted out onto the TXD output pin at the programmed bit rate.<br>When a data size of less than 16 bits is selected, the user must right-justify data written to the transmit FIFO. The transmit logic ignores the unused bits.<br>0h = Smallest value<br>FFFFFFFFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

Chapter 27
# UNICOMM

The UNICOMM module is a run time configurable peripheral capable of supporting UART, $I^2C$ and SPI.


**27.1 Overview** ..... 1475
**27.2 Unicomm Architecture** ..... 1477
**27.3 High-Level Initialization** ..... 1479
**27.4 UNICOMM/SPGSS Registers** ..... 1479





Copyright © 2026 Texas Instruments Incorporated

## 27.1 Overview

UNICOMM is a unified serial communication peripheral capable of operating in multiple different standard serial protocols including UART, SPI, and I2C. Register settings configure which protocol is selected for a given UNICOMM peripheral instance via the IPMODE field. A UNICOMM instance can only be configured for one protocol mode at a time.

Each UNICOMM instance can be configured to operate in one of the following protocol modes, if available:

- UART - Universal Asychronous Receiver/Transmitter
- SPI - Serial Peripheral Interface
- I2CC - Inter Integrated Circuit Controller
- I2CT - Inter Integrated Circuit Target

This document commonly uses the abbreviation UCx to refer to a specific UNICOMM instance, with x being the unique instance number. Individual UNICOMM instances are grouped together into designated Scalable Peripheral Groupings (SPGs) for high level configurations such as I2C pairings. Each grouping uses the naming scheme SPGx (or Sx in the UNICOMM register blocks) where x represents the group number index that the UNICOMM instance belongs to.

The specific UNICOMM and SPG instances available in a device will vary, so it is best to refer to the device-specific data sheet for an exact representation of available instances and configurations. Each UNICOMM instance will also vary in what power domains they are supported in, and whether Direct Memory Access (DMA) is supported or not. Table 27-1 lists the configuration options and peripheral types available for each unique UNICOMM instance. The device-specific datasheet will indicate what SPGx index each UCx instance is grouped within.

**Table 27-1. Available UNICOMM Configurations Per Instance**

| Power Domain | UNICOMM Instance (UCx) | Available Peripheral Types | | | | DMA support |
|---|---|---|---|---|---|---|
| | | UART | SPI | I2C Controller | I2C Target | |
| PD0 | UC0 | Advanced | - | Advanced | Advanced | Y |
| | UC1 | Basic+LIN | - | Advanced | Advanced | Y |
| | UC4 | Advanced | Basic | - | - | Y |
| | UC5 | Minimum+Lin | - | Advanced | Advanced | Y |
| | UC6 | - | - | Advanced | - | Y |
| | UC7 | - | - | - | Advanced | Y |
| | UC10 | Advanced | - | - | - | Y |
| | UC16 | Advanced | Basic | Advanced | Advanced | Y |
| | UC17 | Basic+LIN | Basic | Advanced | Advanced | Y |



Copyright © 2026 Texas Instruments Incorporated

**Table 27-1. Available UNICOMM Configurations Per Instance (continued)**

| Power Domain | UNICOMM Instance (UCx) | Available Peripheral Types | | | | DMA support |
|---|---|---|---|---|---|---|
| | | UART | SPI | I2C Controller | I2C Target | |
| PD1 | UC2 | - | Advanced | - | - | Y |
| | UC3 | Basic | Basic | - | - | Y |
| | UC8 | Minimum | Basic | - | - | N |
| | UC9 | Minimum+LIN | - | - | - | N |
| | UC11 | Minimum | - | - | - | N |
| | UC12 | Advanced | - | - | - | Y |
| | UC13 | Basic | Basic | Minimum | Minimum | Y |
| | UC14 | Basic+LIN | - | Advanced | Advanced | Y |
| | UC15 | - | - | Advanced | Advanced | Y |
| | UC18 | Basic+LIN | Advanced | - | - | Y |
| | UC19 | Basic+LIN | - | Advanced | Advanced | Y |

This chapter describes all the features and configurations that are common across all UNICOMM instances regardless of the IPMODE configured. This includes FIFO operation, enabling and resetting a UNICOMM instance, the generic transmit and receive sequence, and generic initialization sequence for UNICOMM.

The detailed and protocol-specific operation of each UNICOMM protocol mode is described in each of the below dedicated chapters:

- UART - UNICOMM-UART
- SPI - UNICOMM-SPI
- I2CC / I2CT - UNICOMM-I2C



Copyright © 2026 Texas Instruments Incorporated

### 27.1.1 Block Diagram

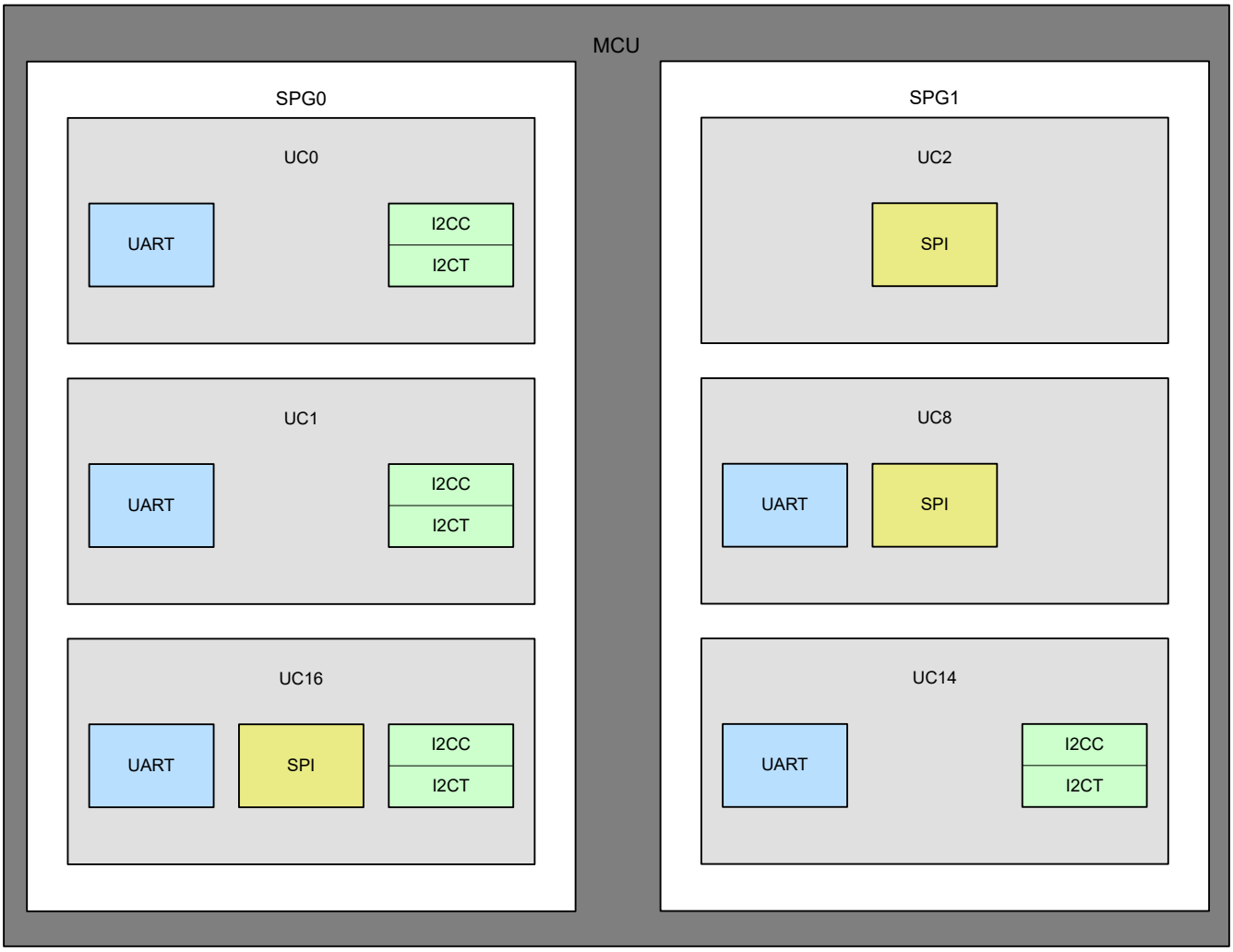



**Figure 27-1. Unicomm Block Diagram**

## 27.2 Unicomm Architecture

### 27.2.1 Serial Peripheral Group (SPG) Configurations

The Scalable Peripheral Group (SPG) level configurations consist of I2C pairing operation in compliance with the SMBUS protocol, as well as multi-controller systems.

**Note**

Users not employing either of the loopback or I2C pairing test features can skip to Enables & Resets .

#### 27.2.1.1 I2C Pairings

I2C pairing supports specialized I2C modes where a controller-target pair are connected to a single external bus. Multi-controller configurations and protocols like SMBus/PMBus require this setup. I2C pairing is available between two UNICOMM instances when both of the below are true:

1. Both UNICOMM modules are configured for I2C communication in the IPMODE register. Controller-Target pairings are supported.
2. Both UNICOMM modules belong to the same SPG instance. Check the *Unicomm Block Diagram* to see which SPG instance the UNICOMM instance belongs to.



Copyright © 2026 Texas Instruments Incorporated

I2C pairing connects SDA-to-SDA and SCL-to-SCL lines between two UNICOMM instances within the same SPGSS & they share a single set of SDA/SCL pins at the external pin interface. The I2C pairing control is set through PAIR0 register.

To create an I2C pair within an SPG:

1. Select the UNICOMM to be used as the first I2C (controller) in the loopback and find the local index number of the UNICOMM instance in the SPG.
2. Select the UNICOMM to be used as the second I2C (target) in the loopback and find the local index number of the UNICOMM instance in the SPG.
3. Write the first UNICOMM index to PAIR0.CTL
4. Write the second UNICOMM index to PAIR0.TARGET
5. Enable the loopback by writing a '1' to PAIR0.EN

**Note**

The *local* index number of the UNICOMM instance can be found in the UNICOMM detailed description section of the device-specific datasheet. Example: SPG1 may contain two UNICOMM instances (UC2 and UC3) in total. The UC2 would have a local index value of '0' because it is the lowest UNICOMM instance number within the SPG1 module.

The I2C pairing feature establishes a controller-target relationship between two UNICOMM modules while maintaining connection to external pins.

When an I2C pair is enabled:

- The SDA/SCL signals from the UNICOMM module specified in PAIR.TARGET are actively propagated to the device pins

- The SDA/SCL signals from the UNICOMM module specified in PAIR.CTL are automatically driven to their IDLE states by the module's internal logic

This configuration allows the TARGET module to control the physical I2C bus while preventing bus contention. Although the PAIR.CTL module's pins are driven to IDLE states internally, these physical pins can still be reassigned through the device's pin multiplexing (pinmux) configuration to serve alternative functions in other peripheral modules if needed.

### *27.2.2 Enables & Resets*

Each UNICOMM instance (UCx) has a dedicated reset and enable bitfield in the UNICOMM_REGS register group.

The UNICOMM instance can be reset by writing a 1 to the RSTCTL.RESETASSERT bit along with the KEY for RSTCTL. Writing to this field performs a complete reset of the UCx module, which:

- Restores all configuration registers to their default values
- Clears all internal state information
- Empties all FIFO buffers

The STAT.RESETKY bit is set once the UCx module has undergone a reset. To clear the RESETKY bit, write to RSTCTL.RESETSTKYCLR bit and the RESETKY bit won't be set until the next reset is performed.

A UNICOMM instance can be enabled at the UCx level by setting the PWREN.ENABLE bit. Clocking of the UC peripheral-specific registers is not enabled until this PWREN.ENABLE bit is set. Therefore, the IPMODE configuration must be configured after PWREN.ENABLE bit is asserted.

Configure the module by first setting the appropriate protocol mode in the IPMODE register, then proceed to write the specific configuration values into the corresponding protocol registers.



Copyright © 2026 Texas Instruments Incorporated

## 27.3 High-Level Initialization

Below are the generic initialization steps to configure a UNICOMM instance and the recommended order for software to execute these steps. Peripheral specific initializations are listed in the respective sections.

1. First, put the UNICOMM instance into reset by setting the RSCTL.RESETASSERT register bit from UNICOMM Registers.
2. Enable power to the UNICOMM instance by writing "0h26" to the KEY field and setting the EN bit for the PWREN register in UNICOMM Registers.
3. Configure the peripheral mode for the UCx instance in the IPMODE.SELECT field from the UNICOMM Registers to operate as a UART, SPI, I2C Controller, or I2C Target.
4. (Optionally) Configure the inter-module internal loopback/pairing settings in SPG Registers between two UCx modules in the same SPG. See the UNICOMM Block Diagram to understand how UCx pairings are available within a SPG module.
   a. Configure the loopback pairs (maximum of one pair per SPG module) in the LPBK0 register field.
   b. Configure the I2C pairs (maximum of one pair per SPG module) in the PAIR0 register field.
5. Enable the pin functionality for each GPIO being used in the peripheral configuration in IOMUX Registers.
6. Configure the below settings in the *UNICOMM-UART Registers* / *UNICOMM-SPI Registers* / *UNICOMM-I2C Registers*.
   a. To enable clocking of the UNICOMM module, configure the clock source in the CLKSEL register to decide the functional clock of UNICOMM.
   b. Configure the clock divider for the UNICOMM instance's peripheral-specific functional clock in the CLKDIV.RATIO register field. This decides how much to divide down the overall UNICOMM clock source for each specific UNICOMM instance clock.
   c. Configure the interrupt masks for the peripheral in the CPU_INT.IMASK register.
   d. Configure the emulation mode for the peripheral in the PDBGCTL register.
   e. Perform peripheral-specific initializations (example: SPI peripheral vs. controller mode, UART baud rate etc.) Follow the detailed steps in in the *UART Initialization*, *SPI Initialization*, *I2C Controller Initialization*, and *I2C Target Initialization* chapters.

## 27.4 UNICOMM/SPGSS Registers

### *27.4.1 UNICOMM Registers*

This section describes the UNICOMM Registers.



Copyright © 2026 Texas Instruments Incorporated

#### 27.4.1.1 UNICOMM Registers

Table 27-2 lists the memory-mapped registers for the UNICOMM registers. All register offset addresses not listed in Table 27-2 should be considered as reserved locations and the register contents should not be modified.

**Table 27-2. UNICOMM Registers**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 800h | PWREN | Power enable | | Go |
| 804h | RSTCTL | Reset Control | | Go |
| 808h | CLKCFG | Peripheral Clock Configuration Register | | Go |
| 814h | STAT | Status Register | | Go |
| 1100h | IPMODE | Mode Selection Register | | Go |

Complex bit access types are encoded to fit into small table cells. Table 27-3 shows the codes that are used for access types in this section.

**Table 27-3. UNICOMM Access Type Codes**

| Access Type | Code | Description |
|---|---|---|
| Read Type | | |
| R | R | Read |
| Write Type | | |
| W | W | Write |
| WK | W<br>K | Write<br>Write protected by a key |
| Reset or Default Value | | |
| *-n* | | Value after reset or the default value |



Copyright © 2026 Texas Instruments Incorporated

#### 27.4.1.1.1 PWREN (Offset = 800h) [Reset = 00000000h]

PWREN is shown in Figure 27-2 and described in Table 27-4.

Return to the Summary Table.

Register to control the power state

**Figure 27-2. PWREN**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | ENABLE |
| R-0h | | | | | | | R/WK-0h |

**Table 27-4. PWREN Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | KEY to allow Power State Change<br>26h = KEY to allow write access to this register |
| 23-1 | RESERVED | R | 0h | |
| 0 | ENABLE | R/WK | 0h | Enable the power<br>KEY must be set to 26h to write to this bit.<br>0h = Disable Power<br>1h = Enable Power |



Copyright © 2026 Texas Instruments Incorporated

#### 27.4.1.1.2 RSTCTL (Offset = 804h) [Reset = 00000000h]

RSTCTL is shown in Figure 27-3 and described in Table 27-5.

Return to the Summary Table.

Register to control reset assertion and de-assertion

**Figure 27-3. RSTCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | RESETSTKYCLR | RESETASSERT |
| R-0h | | | | | | WK-0h | WK-0h |

**Table 27-5. RSTCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | Unlock key<br>B1h = KEY to allow write access to this register |
| 23-2 | RESERVED | R | 0h | |
| 1 | RESETSTKYCLR | WK | 0h | Clear the RESETSTKY bit in the STAT register<br>KEY must be set to B1h to write to this bit.<br>0h = Writing 0 has no effect<br>1h = Clear reset sticky bit |
| 0 | RESETASSERT | WK | 0h | Assert reset to the peripheral<br>KEY must be set to B1h to write to this bit.<br>0h = Writing 0 has no effect<br>1h = Assert reset |



Copyright © 2026 Texas Instruments Incorporated

#### 27.4.1.1.3 CLKCFG (Offset = 808h) [Reset = 00000000h]

CLKCFG is shown in Figure 27-4 and described in Table 27-6.

Return to the Summary Table.

Peripheral Clock Configuration Register

**Figure 27-4. CLKCFG**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | BLOCKASYNC |
| R-0h | | | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |

**Table 27-6. CLKCFG Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | KEY to Allow State Change -- 0xA9<br>A9h = key value to allow change field of GPRCM |
| 23-9 | RESERVED | R | 0h | |
| 8 | BLOCKASYNC | R/W | 0h | Async Clock Request is blocked from starting SYSOSC or forcing bus clock to 32MHz<br>0h = Not block async clock request<br>1h = Block async clock request |
| 7-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

#### 27.4.1.1.4 STAT (Offset = 814h) [Reset = 00000000h]

STAT is shown in Figure 27-5 and described in Table 27-7.

Return to the Summary Table.

peripheral enable and reset status

**Figure 27-5. STAT**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | RESETSTKY |
| R-0h | | | | | | | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |

**Table 27-7. STAT Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-17 | RESERVED | R | 0h | |
| 16 | RESETSTKY | R | 0h | This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register<br>0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register<br>1h = The peripheral was reset since the last bit clear |
| 15-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

#### 27.4.1.1.5 IPMODE (Offset = 1100h) [Reset = 00000000h]

IPMODE is shown in Figure 27-6 and described in Table 27-8.

Return to the Summary Table.

Used for selecting mode - UART / I2C Controller / I2C Peripheral / SPI

**Figure 27-6. IPMODE**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | | | SELECT | |
| R-0h | | | | | | | | | | | | | | R/W-0h | |

**Table 27-8. IPMODE Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | |
| 1-0 | SELECT | R/W | 0h | Free run control<br>0h = Default functionality; UNICOMM in UART mode<br>1h = UNICOMM in SPI Mode<br>2h = UNICOMM in I2C Controller Mode<br>3h = UNICOMM in I2C Peripheral/Target Mode |

### *27.4.2 SPG Registers*

This section describes the SPG Registers.



Copyright © 2026 Texas Instruments Incorporated

#### 27.4.2.1 SPGSS Registers

Table 27-9 lists the memory-mapped registers for the SPGSS registers. All register offset addresses not listed in Table 27-9 should be considered as reserved locations and the register contents should not be modified.

**Table 27-9. SPGSS Registers**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 1C0h | PAIR0 | Pairing of I2C for Multi-Controller/ SMBUS applications | | Go |
| 1C4h | PAIR1 | Pairing of I2C for Multi-Controller/ SMBUS applications | | Go |
| 1C8h | PAIR2 | Pairing of I2C for Multi-Controller/ SMBUS applications | | Go |
| 1CCh | PAIR3 | Pairing of I2C for Multi-Controller/ SMBUS applications | | Go |
| 1D0h | PAIR4 | Pairing of I2C for Multi-Controller/ SMBUS applications | | Go |
| 1D4h | PAIR5 | Pairing of I2C for Multi-Controller/ SMBUS applications | | Go |
| 1D8h | PAIR6 | Pairing of I2C for Multi-Controller/ SMBUS applications | | Go |
| 1DCh | PAIR7 | Pairing of I2C for Multi-Controller/ SMBUS applications | | Go |

Complex bit access types are encoded to fit into small table cells. Table 27-10 shows the codes that are used for access types in this section.

**Table 27-10. SPGSS Access Type Codes**

| Access Type | Code | Description |
|---|---|---|
| Read Type | | |
| R | R | Read |
| Write Type | | |
| W | W | Write |
| Reset or Default Value | | |
| *-n* | | Value after reset or the default value |



Copyright © 2026 Texas Instruments Incorporated

#### 27.4.2.1.1 PAIR0 (Offset = 1C0h) [Reset = 00000000h]

PAIR0 is shown in Figure 27-7 and described in Table 27-11.

Return to the Summary Table.

Pairing of I2C for Multi-Controller/SMBUS applications

**Figure 27-7. PAIR0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TARGET | | | | CTL | | | | RESERVED | | | | | | | EN |
| R/W-0h | | | | R/W-0h | | | | R-0h | | | | | | | R/W-0h |

**Table 27-11. PAIR0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15-12 | TARGET | R/W | 0h | UNICOMM instance for loopback. This instance will be treated as the target<br>0h = UNICOMM peripheral/target for loopback.<br>1h = UNICOMM peripheral/target for loopback.<br>2h = UNICOMM peripheral/target for loopback.<br>3h = UNICOMM peripheral/target for loopback.<br>4h = UNICOMM peripheral/target for loopback.<br>5h = UNICOMM peripheral/target for loopback.<br>6h = UNICOMM peripheral/target for loopback.<br>7h = UNICOMM peripheral/target for loopback.<br>8h = UNICOMM peripheral/target for loopback.<br>9h = UNICOMM peripheral/target for loopback.<br>Ah = UNICOMM peripheral/target for loopback.<br>Bh = UNICOMM peripheral/target for loopback.<br>Ch = UNICOMM peripheral/target for loopback.<br>Dh = UNICOMM peripheral/target for loopback.<br>Eh = UNICOMM peripheral/target for loopback.<br>Fh = UNICOMM peripheral/target for loopback. |
| 11-8 | CTL | R/W | 0h | UNICOMM instance for loopback. This instance will be treated as the controller.<br>0h = UNICOMM controller for loopback.<br>1h = UNICOMM controller for loopback.<br>2h = UNICOMM controller for loopback.<br>3h = UNICOMM controller for loopback.<br>4h = UNICOMM controller for loopback.<br>5h = UNICOMM controller for loopback.<br>6h = UNICOMM controller for loopback.<br>7h = UNICOMM controller for loopback.<br>8h = UNICOMM controller for loopback.<br>9h = UNICOMM controller for loopback.<br>Ah = UNICOMM controller for loopback.<br>Bh = UNICOMM controller for loopback.<br>Ch = UNICOMM controller for loopback.<br>Dh = UNICOMM controller for loopback.<br>Eh = UNICOMM controller for loopback.<br>Fh = UNICOMM controller for loopback. |
| 7-1 | RESERVED | R | 0h | |
| 0 | EN | R/W | 0h | Enable<br>0h = Disable<br>1h = Enable |



Copyright © 2026 Texas Instruments Incorporated

#### 27.4.2.1.2 PAIR1 (Offset = 1C4h) [Reset = 00000000h]

PAIR1 is shown in Figure 27-8 and described in Table 27-12.

Return to the Summary Table.

Pairing of I2C for Multi-Controller/SMBUS applications

**Figure 27-8. PAIR1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TARGET | | | | CTL | | | | RESERVED | | | | | | | EN |
| R/W-0h | | | | R/W-0h | | | | R-0h | | | | | | | R/W-0h |

**Table 27-12. PAIR1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15-12 | TARGET | R/W | 0h | UNICOMM instance for loopback. This instance will be treated as the target<br>0h = UNICOMM peripheral/target for loopback.<br>1h = UNICOMM peripheral/target for loopback.<br>2h = UNICOMM peripheral/target for loopback.<br>3h = UNICOMM peripheral/target for loopback.<br>4h = UNICOMM peripheral/target for loopback.<br>5h = UNICOMM peripheral/target for loopback.<br>6h = UNICOMM peripheral/target for loopback.<br>7h = UNICOMM peripheral/target for loopback.<br>8h = UNICOMM peripheral/target for loopback.<br>9h = UNICOMM peripheral/target for loopback.<br>Ah = UNICOMM peripheral/target for loopback.<br>Bh = UNICOMM peripheral/target for loopback.<br>Ch = UNICOMM peripheral/target for loopback.<br>Dh = UNICOMM peripheral/target for loopback.<br>Eh = UNICOMM peripheral/target for loopback.<br>Fh = UNICOMM peripheral/target for loopback. |
| 11-8 | CTL | R/W | 0h | UNICOMM instance for loopback. This instance will be treated as the controller.<br>0h = UNICOMM controller for loopback.<br>1h = UNICOMM controller for loopback.<br>2h = UNICOMM controller for loopback.<br>3h = UNICOMM controller for loopback.<br>4h = UNICOMM controller for loopback.<br>5h = UNICOMM controller for loopback.<br>6h = UNICOMM controller for loopback.<br>7h = UNICOMM controller for loopback.<br>8h = UNICOMM controller for loopback.<br>9h = UNICOMM controller for loopback.<br>Ah = UNICOMM controller for loopback.<br>Bh = UNICOMM controller for loopback.<br>Ch = UNICOMM controller for loopback.<br>Dh = UNICOMM controller for loopback.<br>Eh = UNICOMM controller for loopback.<br>Fh = UNICOMM controller for loopback. |
| 7-1 | RESERVED | R | 0h | |
| 0 | EN | R/W | 0h | Enable<br>0h = Disable<br>1h = Enable |



Copyright © 2026 Texas Instruments Incorporated

#### 27.4.2.1.3 PAIR2 (Offset = 1C8h) [Reset = 00000000h]

PAIR2 is shown in Figure 27-9 and described in Table 27-13.

Return to the Summary Table.

Pairing of I2C for Multi-Controller/SMBUS applications

**Figure 27-9. PAIR2**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TARGET | | | | CTL | | | | RESERVED | | | | | | | EN |
| R/W-0h | | | | R/W-0h | | | | R-0h | | | | | | | R/W-0h |

**Table 27-13. PAIR2 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15-12 | TARGET | R/W | 0h | UNICOMM instance for loopback. This instance will be treated as the target<br>0h = UNICOMM peripheral/target for loopback.<br>1h = UNICOMM peripheral/target for loopback.<br>2h = UNICOMM peripheral/target for loopback.<br>3h = UNICOMM peripheral/target for loopback.<br>4h = UNICOMM peripheral/target for loopback.<br>5h = UNICOMM peripheral/target for loopback.<br>6h = UNICOMM peripheral/target for loopback.<br>7h = UNICOMM peripheral/target for loopback.<br>8h = UNICOMM peripheral/target for loopback.<br>9h = UNICOMM peripheral/target for loopback.<br>Ah = UNICOMM peripheral/target for loopback.<br>Bh = UNICOMM peripheral/target for loopback.<br>Ch = UNICOMM peripheral/target for loopback.<br>Dh = UNICOMM peripheral/target for loopback.<br>Eh = UNICOMM peripheral/target for loopback.<br>Fh = UNICOMM peripheral/target for loopback. |
| 11-8 | CTL | R/W | 0h | UNICOMM instance for loopback. This instance will be treated as the controller.<br>0h = UNICOMM controller for loopback.<br>1h = UNICOMM controller for loopback.<br>2h = UNICOMM controller for loopback.<br>3h = UNICOMM controller for loopback.<br>4h = UNICOMM controller for loopback.<br>5h = UNICOMM controller for loopback.<br>6h = UNICOMM controller for loopback.<br>7h = UNICOMM controller for loopback.<br>8h = UNICOMM controller for loopback.<br>9h = UNICOMM controller for loopback.<br>Ah = UNICOMM controller for loopback.<br>Bh = UNICOMM controller for loopback.<br>Ch = UNICOMM controller for loopback.<br>Dh = UNICOMM controller for loopback.<br>Eh = UNICOMM controller for loopback.<br>Fh = UNICOMM controller for loopback. |
| 7-1 | RESERVED | R | 0h | |
| 0 | EN | R/W | 0h | Enable<br>0h = Disable<br>1h = Enable |



Copyright © 2026 Texas Instruments Incorporated

#### 27.4.2.1.4 PAIR3 (Offset = 1CCh) [Reset = 00000000h]

PAIR3 is shown in Figure 27-10 and described in Table 27-14.

Return to the Summary Table.

Pairing of I2C for Multi-Controller/SMBUS applications

**Figure 27-10. PAIR3**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TARGET | | | | CTL | | | | RESERVED | | | | | | | EN |
| R/W-0h | | | | R/W-0h | | | | R-0h | | | | | | | R/W-0h |

**Table 27-14. PAIR3 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15-12 | TARGET | R/W | 0h | UNICOMM instance for loopback. This instance will be treated as the target<br>0h = UNICOMM peripheral/target for loopback.<br>1h = UNICOMM peripheral/target for loopback.<br>2h = UNICOMM peripheral/target for loopback.<br>3h = UNICOMM peripheral/target for loopback.<br>4h = UNICOMM peripheral/target for loopback.<br>5h = UNICOMM peripheral/target for loopback.<br>6h = UNICOMM peripheral/target for loopback.<br>7h = UNICOMM peripheral/target for loopback.<br>8h = UNICOMM peripheral/target for loopback.<br>9h = UNICOMM peripheral/target for loopback.<br>Ah = UNICOMM peripheral/target for loopback.<br>Bh = UNICOMM peripheral/target for loopback.<br>Ch = UNICOMM peripheral/target for loopback.<br>Dh = UNICOMM peripheral/target for loopback.<br>Eh = UNICOMM peripheral/target for loopback.<br>Fh = UNICOMM peripheral/target for loopback. |
| 11-8 | CTL | R/W | 0h | UNICOMM instance for loopback. This instance will be treated as the controller.<br>0h = UNICOMM controller for loopback.<br>1h = UNICOMM controller for loopback.<br>2h = UNICOMM controller for loopback.<br>3h = UNICOMM controller for loopback.<br>4h = UNICOMM controller for loopback.<br>5h = UNICOMM controller for loopback.<br>6h = UNICOMM controller for loopback.<br>7h = UNICOMM controller for loopback.<br>8h = UNICOMM controller for loopback.<br>9h = UNICOMM controller for loopback.<br>Ah = UNICOMM controller for loopback.<br>Bh = UNICOMM controller for loopback.<br>Ch = UNICOMM controller for loopback.<br>Dh = UNICOMM controller for loopback.<br>Eh = UNICOMM controller for loopback.<br>Fh = UNICOMM controller for loopback. |
| 7-1 | RESERVED | R | 0h | |
| 0 | EN | R/W | 0h | Enable<br>0h = Disable<br>1h = Enable |



Copyright © 2026 Texas Instruments Incorporated

#### 27.4.2.1.5 PAIR4 (Offset = 1D0h) [Reset = 00000000h]

PAIR4 is shown in Figure 27-11 and described in Table 27-15.

Return to the Summary Table.

Pairing of I2C for Multi-Controller/SMBUS applications

**Figure 27-11. PAIR4**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TARGET | | | | CTL | | | | RESERVED | | | | | | | EN |
| R/W-0h | | | | R/W-0h | | | | R-0h | | | | | | | R/W-0h |

**Table 27-15. PAIR4 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15-12 | TARGET | R/W | 0h | UNICOMM instance for loopback. This instance will be treated as the target<br>0h = UNICOMM peripheral/target for loopback.<br>1h = UNICOMM peripheral/target for loopback.<br>2h = UNICOMM peripheral/target for loopback.<br>3h = UNICOMM peripheral/target for loopback.<br>4h = UNICOMM peripheral/target for loopback.<br>5h = UNICOMM peripheral/target for loopback.<br>6h = UNICOMM peripheral/target for loopback.<br>7h = UNICOMM peripheral/target for loopback.<br>8h = UNICOMM peripheral/target for loopback.<br>9h = UNICOMM peripheral/target for loopback.<br>Ah = UNICOMM peripheral/target for loopback.<br>Bh = UNICOMM peripheral/target for loopback.<br>Ch = UNICOMM peripheral/target for loopback.<br>Dh = UNICOMM peripheral/target for loopback.<br>Eh = UNICOMM peripheral/target for loopback.<br>Fh = UNICOMM peripheral/target for loopback. |
| 11-8 | CTL | R/W | 0h | UNICOMM instance for loopback. This instance will be treated as the controller.<br>0h = UNICOMM controller for loopback.<br>1h = UNICOMM controller for loopback.<br>2h = UNICOMM controller for loopback.<br>3h = UNICOMM controller for loopback.<br>4h = UNICOMM controller for loopback.<br>5h = UNICOMM controller for loopback.<br>6h = UNICOMM controller for loopback.<br>7h = UNICOMM controller for loopback.<br>8h = UNICOMM controller for loopback.<br>9h = UNICOMM controller for loopback.<br>Ah = UNICOMM controller for loopback.<br>Bh = UNICOMM controller for loopback.<br>Ch = UNICOMM controller for loopback.<br>Dh = UNICOMM controller for loopback.<br>Eh = UNICOMM controller for loopback.<br>Fh = UNICOMM controller for loopback. |
| 7-1 | RESERVED | R | 0h | |
| 0 | EN | R/W | 0h | Enable<br>0h = Disable<br>1h = Enable |



Copyright © 2026 Texas Instruments Incorporated

#### 27.4.2.1.6 PAIR5 (Offset = 1D4h) [Reset = 00000000h]

PAIR5 is shown in Figure 27-12 and described in Table 27-16.

Return to the Summary Table.

Pairing of I2C for Multi-Controller/SMBUS applications

**Figure 27-12. PAIR5**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TARGET | | | | CTL | | | | RESERVED | | | | | | | EN |
| R/W-0h | | | | R/W-0h | | | | R-0h | | | | | | | R/W-0h |

**Table 27-16. PAIR5 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15-12 | TARGET | R/W | 0h | UNICOMM instance for loopback. This instance will be treated as the target<br>0h = UNICOMM peripheral/target for loopback.<br>1h = UNICOMM peripheral/target for loopback.<br>2h = UNICOMM peripheral/target for loopback.<br>3h = UNICOMM peripheral/target for loopback.<br>4h = UNICOMM peripheral/target for loopback.<br>5h = UNICOMM peripheral/target for loopback.<br>6h = UNICOMM peripheral/target for loopback.<br>7h = UNICOMM peripheral/target for loopback.<br>8h = UNICOMM peripheral/target for loopback.<br>9h = UNICOMM peripheral/target for loopback.<br>Ah = UNICOMM peripheral/target for loopback.<br>Bh = UNICOMM peripheral/target for loopback.<br>Ch = UNICOMM peripheral/target for loopback.<br>Dh = UNICOMM peripheral/target for loopback.<br>Eh = UNICOMM peripheral/target for loopback.<br>Fh = UNICOMM peripheral/target for loopback. |
| 11-8 | CTL | R/W | 0h | UNICOMM instance for loopback. This instance will be treated as the controller.<br>0h = UNICOMM controller for loopback.<br>1h = UNICOMM controller for loopback.<br>2h = UNICOMM controller for loopback.<br>3h = UNICOMM controller for loopback.<br>4h = UNICOMM controller for loopback.<br>5h = UNICOMM controller for loopback.<br>6h = UNICOMM controller for loopback.<br>7h = UNICOMM controller for loopback.<br>8h = UNICOMM controller for loopback.<br>9h = UNICOMM controller for loopback.<br>Ah = UNICOMM controller for loopback.<br>Bh = UNICOMM controller for loopback.<br>Ch = UNICOMM controller for loopback.<br>Dh = UNICOMM controller for loopback.<br>Eh = UNICOMM controller for loopback.<br>Fh = UNICOMM controller for loopback. |
| 7-1 | RESERVED | R | 0h | |
| 0 | EN | R/W | 0h | Enable<br>0h = Disable<br>1h = Enable |



Copyright © 2026 Texas Instruments Incorporated

#### 27.4.2.1.7 PAIR6 (Offset = 1D8h) [Reset = 00000000h]

PAIR6 is shown in Figure 27-13 and described in Table 27-17.

Return to the Summary Table.

Pairing of I2C for Multi-Controller/SMBUS applications

**Figure 27-13. PAIR6**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TARGET | | | | CTL | | | | RESERVED | | | | | | | EN |
| R/W-0h | | | | R/W-0h | | | | R-0h | | | | | | | R/W-0h |

**Table 27-17. PAIR6 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15-12 | TARGET | R/W | 0h | UNICOMM instance for loopback. This instance will be treated as the target<br>0h = UNICOMM peripheral/target for loopback.<br>1h = UNICOMM peripheral/target for loopback.<br>2h = UNICOMM peripheral/target for loopback.<br>3h = UNICOMM peripheral/target for loopback.<br>4h = UNICOMM peripheral/target for loopback.<br>5h = UNICOMM peripheral/target for loopback.<br>6h = UNICOMM peripheral/target for loopback.<br>7h = UNICOMM peripheral/target for loopback.<br>8h = UNICOMM peripheral/target for loopback.<br>9h = UNICOMM peripheral/target for loopback.<br>Ah = UNICOMM peripheral/target for loopback.<br>Bh = UNICOMM peripheral/target for loopback.<br>Ch = UNICOMM peripheral/target for loopback.<br>Dh = UNICOMM peripheral/target for loopback.<br>Eh = UNICOMM peripheral/target for loopback.<br>Fh = UNICOMM peripheral/target for loopback. |
| 11-8 | CTL | R/W | 0h | UNICOMM instance for loopback. This instance will be treated as the controller.<br>0h = UNICOMM controller for loopback.<br>1h = UNICOMM controller for loopback.<br>2h = UNICOMM controller for loopback.<br>3h = UNICOMM controller for loopback.<br>4h = UNICOMM controller for loopback.<br>5h = UNICOMM controller for loopback.<br>6h = UNICOMM controller for loopback.<br>7h = UNICOMM controller for loopback.<br>8h = UNICOMM controller for loopback.<br>9h = UNICOMM controller for loopback.<br>Ah = UNICOMM controller for loopback.<br>Bh = UNICOMM controller for loopback.<br>Ch = UNICOMM controller for loopback.<br>Dh = UNICOMM controller for loopback.<br>Eh = UNICOMM controller for loopback.<br>Fh = UNICOMM controller for loopback. |
| 7-1 | RESERVED | R | 0h | |
| 0 | EN | R/W | 0h | Enable<br>0h = Disable<br>1h = Enable |



Copyright © 2026 Texas Instruments Incorporated

#### 27.4.2.1.8 PAIR7 (Offset = 1DCh) [Reset = 00000000h]

PAIR7 is shown in Figure 27-14 and described in Table 27-18.

Return to the Summary Table.

Pairing of I2C for Multi-Controller/SMBUS applications

**Figure 27-14. PAIR7**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TARGET | | | | CTL | | | | RESERVED | | | | | | | EN |
| R/W-0h | | | | R/W-0h | | | | R-0h | | | | | | | R/W-0h |

**Table 27-18. PAIR7 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15-12 | TARGET | R/W | 0h | UNICOMM instance for loopback. This instance will be treated as the target<br>0h = UNICOMM peripheral/target for loopback.<br>1h = UNICOMM peripheral/target for loopback.<br>2h = UNICOMM peripheral/target for loopback.<br>3h = UNICOMM peripheral/target for loopback.<br>4h = UNICOMM peripheral/target for loopback.<br>5h = UNICOMM peripheral/target for loopback.<br>6h = UNICOMM peripheral/target for loopback.<br>7h = UNICOMM peripheral/target for loopback.<br>8h = UNICOMM peripheral/target for loopback.<br>9h = UNICOMM peripheral/target for loopback.<br>Ah = UNICOMM peripheral/target for loopback.<br>Bh = UNICOMM peripheral/target for loopback.<br>Ch = UNICOMM peripheral/target for loopback.<br>Dh = UNICOMM peripheral/target for loopback.<br>Eh = UNICOMM peripheral/target for loopback.<br>Fh = UNICOMM peripheral/target for loopback. |
| 11-8 | CTL | R/W | 0h | UNICOMM instance for loopback. This instance will be treated as the controller.<br>0h = UNICOMM controller for loopback.<br>1h = UNICOMM controller for loopback.<br>2h = UNICOMM controller for loopback.<br>3h = UNICOMM controller for loopback.<br>4h = UNICOMM controller for loopback.<br>5h = UNICOMM controller for loopback.<br>6h = UNICOMM controller for loopback.<br>7h = UNICOMM controller for loopback.<br>8h = UNICOMM controller for loopback.<br>9h = UNICOMM controller for loopback.<br>Ah = UNICOMM controller for loopback.<br>Bh = UNICOMM controller for loopback.<br>Ch = UNICOMM controller for loopback.<br>Dh = UNICOMM controller for loopback.<br>Eh = UNICOMM controller for loopback.<br>Fh = UNICOMM controller for loopback. |
| 7-1 | RESERVED | R | 0h | |
| 0 | EN | R/W | 0h | Enable<br>0h = Disable<br>1h = Enable |



Copyright © 2026 Texas Instruments Incorporated

Chapter 28

# UNICOMM-UART

**28.1 UNICOMM-UART Overview** ..... 1496
**28.2 UART Operation** ..... 1497



Copyright © 2026 Texas Instruments Incorporated

## 28.1 UNICOMM-UART Overview

The functionality of a UNICOMM module when configured to operate as a UART is described in this section. The protocol mode of a given UNICOMM instance is decided by the programmed SELECT field of the IPMODE register. If a UNICOMM instance is configured for another peripheral mode, the UART functionality on that UNICOMM instance is disabled and unusable.

### *28.1.1 Purpose of the Peripheral*

The Universal Asynchronous Receiver/Transmitter (UART) interface is a single-ended, asynchronous serial communication port that transmits (TX) and receives (RX) a programmable length serial bit stream into and out of the device at a programmable baud rate. The UNICOMM UART module can be used to communicate with another device supporting the standard UART protocol.

### *28.1.2 Features*

The UNICOMM-UART peripheral mode of the UNICOMM module includes the following features:

- Fully programmable serial interface:
  - 5, 6, 7 or 8 data bits
  - Even, odd, stick, or no-parity bit generation and detection
  - 1 or 2 stop bit generation
  - LSB-first or MSB-first data transmit and receive
  - Line-break detection
  - Glitch filter on the input signal
  - Programmable baud-rate generation with oversampling by 16, 8 or 3
- Separated transmit and receive buffer/FIFO
- Direct Memory Access (DMA) support
- Transmit to receive internal loopback mode operation
- Standard FIFO-level and end-of-transmission interrupts
- Active in all low-power modes (including STOP and STANDBY)
- Support for waking up SYSOSC via an asynchronous fast clock request upon start bit detection when operating in low power modes
- Loopback operation
- Hardware flow control
- 9-bit multi-drop configuration
- Protocols supported:
  - Local Interconnect Network (LIN)
  - DALI
  - IrDA
  - ISO7816 Smart Card
  - RS485
  - Manchester encoding
  - Idle-Line Multiprocessor

| Feature | MIN | MIN_LIN | Basic | Basic + LIN | Advanced |
|---|---|---|---|---|---|
| Receive timeout / Line timeout | No | No | Yes | Yes | Yes |
| Idleline Multiprocessor | No | No | Yes | Yes | Yes |
| RS232 | No | No | Yes | Yes | Yes |
| Multidrop 9-bit | No | No | Yes | Yes | Yes |
| Smartcard | No | No | Yes | No | Yes |
| LIN | No | Yes | No | Yes | Yes |
| DALI & MENC | No | No | No | No | Yes |
| IRDA | No | No | No | No | Yes |
| Low Power Mode Wakeup | Yes if module in PD0 | Yes if module in PD0 | Yes if module in PD0 | Yes if module in PD0 | Yes if module in PD0 |



Copyright © 2026 Texas Instruments Incorporated

| | | | | | |
|---|---|---|---|---|---|
| FIFO | Refer to device specific data sheet | Refer to device specific data sheet | Refer to device specific data sheet | Refer to device specific data sheet | Refer to device specific data sheet |
| DMA | Refer to device specific data sheet | Refer to device specific data sheet | Refer to device specific data sheet | Refer to device specific data sheet | Refer to device specific data sheet |

### *28.1.3 Functional Block Diagram*

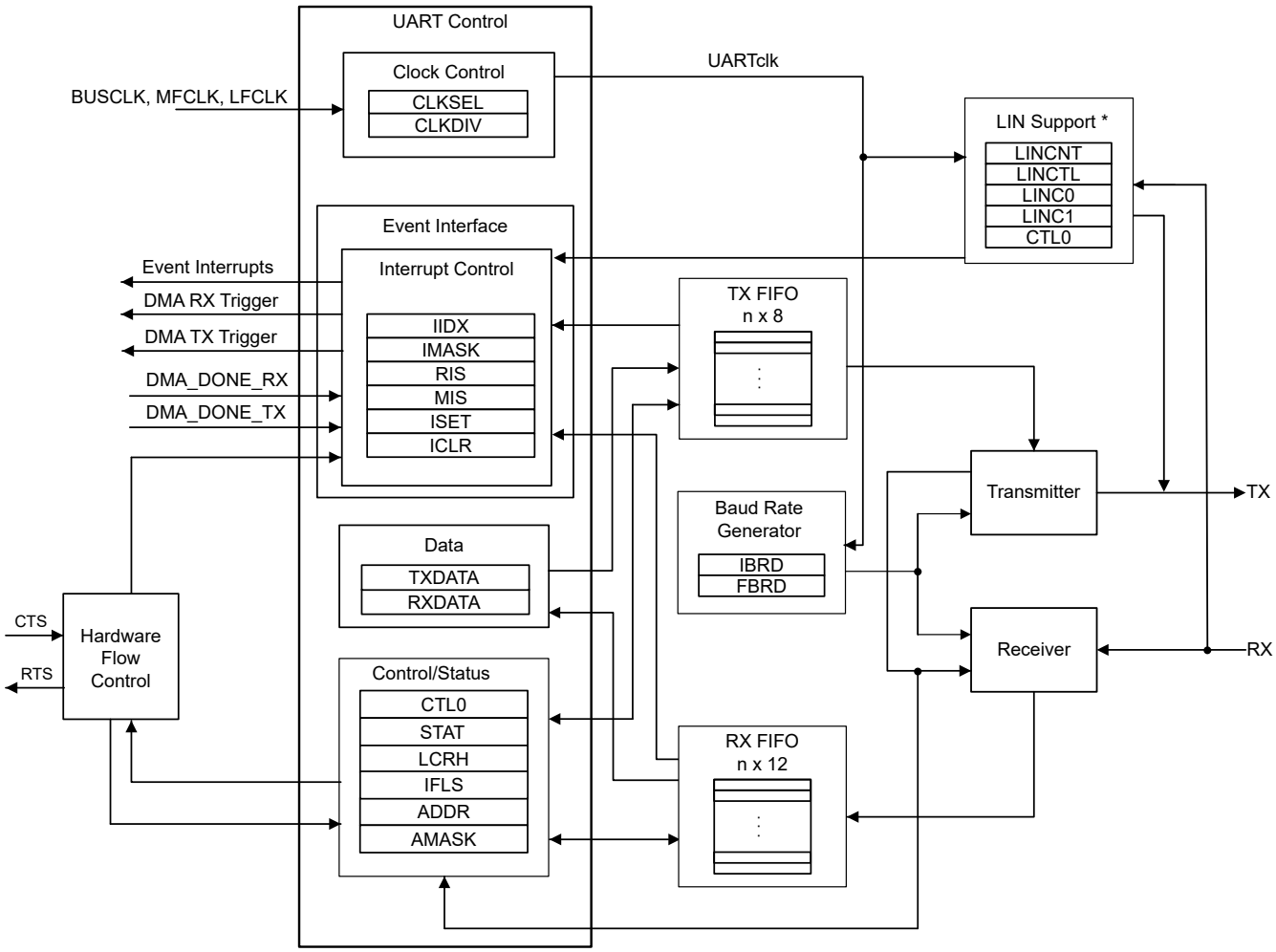



**Figure 28-1. UART Functional Block Diagram**

## 28.2 UART Operation

This section describes the operation of the UART peripheral.

### *28.2.1 Clock Control*

The UART internal functional clock is selected and divided from the functional clock of the IP.

- Use the CLKSEL register to enable and select the functional clock source for the UNICOMM UART module.
- Use the CLKDIV register to select the divide ratio of the UART functional clock.

The selected source clock is always available and the frequency depends on the power mode, for more information, see the *Clock Module (CKM)* section. After enabling the UART module by setting the CTL0.ENABLE bit, the module is ready to start receiving and transmitting data.



Copyright © 2026 Texas Instruments Incorporated

### 28.2.2 General Architecture and Protocol

UART transmits and receives characters at a bit rate that is asynchronous to another device. Timing for each character is based on the selected baud-rate. The transmit and receive functions use the same baud-rate frequency.

Control registers should only be programmed when UART is disabled (ENABLE bit in the CTL0 register is cleared). If UART is operating and gets disabled during transmit or receive operation, the current transaction gets completed before UART stops.

#### 28.2.2.1 Signal Descriptions

UART communications require two pins: Receive Data (RX) and Transmit Data (TX):

- RX (Receive Data): The serial data input. Oversampling techniques are used on the receive signal for accuracy of incoming data.
- TX (Transmit Data): The serial data output. When the transmitter is enabled and no data needs to be transmitted, the TX pin is held high (idle). In some bidirectional protocols like ISO7816 Smart card, this pin is also used to receive data.

In hardware flow control mode, the following pins are also used.

- CTS (Clear To Send): The flow control input. When driven high by an external signal, this signal blocks the data transmission at the end of the current transfer.
- RTS (Request To Send): The flow control output. When held low, this signal indicates that the UART is ready to receive data.

#### 28.2.2.2 Transmit and Receive Logic

Transmit logic performs parallel-to-serial conversion on data read from transmit FIFO. The control logic outputs the serial bit stream beginning with a start bit, followed by data bits (LSB/MSB first based on configuration), parity bit, and stop bits according to programmed configuration

The receive logic performs serial-to-parallel conversion on the received bit stream after a valid start pulse has been detected. Overrun, parity, frame error checking, and line-break detection are also performed, and this status accompanies the data that is written to the receive FIFO. Each RX FIFO element is 12-bits wide to include the data bits and all status information.

- CTL0.ENABLE bit is used to enable and disable the UNICOMM-UART module
- CTL0.TXE and RXE bits are used to enable the transmit and receive mode
- LCRH.WLEN bit is used to configure the number of data bits transmitted or received in a frame
- LCRH.PEN is used to enable parity
- CTL0.MSBFIRST is used to send MSB/LSB first
- LCRH.STP2 is used to send one/ two stop bits (one bit is default)

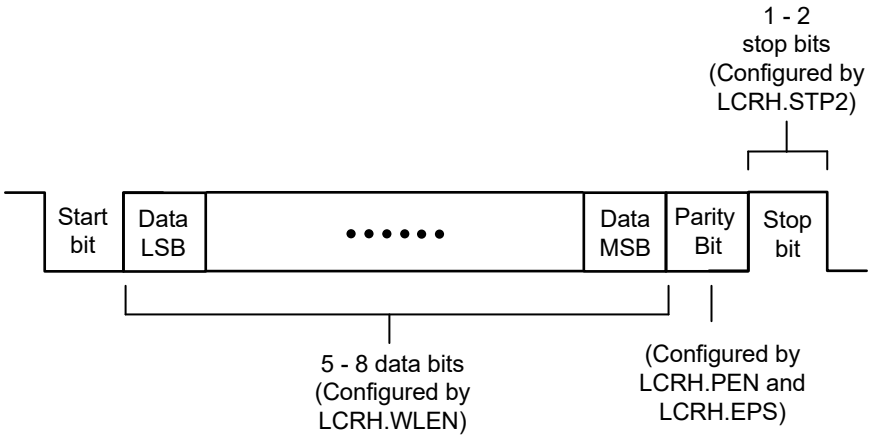



**Figure 28-2. UART Character Frame**



Copyright © 2026 Texas Instruments Incorporated

#### 28.2.2.3 Bit Sampling

UNICOMM UART supports oversampling at 3x, 8x, and 16x the UART bus baud-rate.

- By default, CTL0.HSE is set to 0 (16x oversampling). The center sample (sample 8) is used for sampling data on RX line.
- CTL0.HSE set to ‘1’ selects 8x oversampling. The center sample (sample 4) is used for sampling data on RX line.
- CTL0.HSE set to ‘2’ selects 3x oversampling. The center sample (sample 2) is used for sampling data on RX line.

The previously mentioned scenarios assume IBRD =1 and FBRD =0.

Depending on the application:

- Select oversampling by 3 or 8 to achieve **higher speed** with UARTclk/8 or UARTclk/3. In this case the receiver tolerance to clock deviation is reduced.
- Select oversampling by 16 to increase the **tolerance** of the receiver to clock deviations
  - The maximum speed is limited to UARTclk/16.

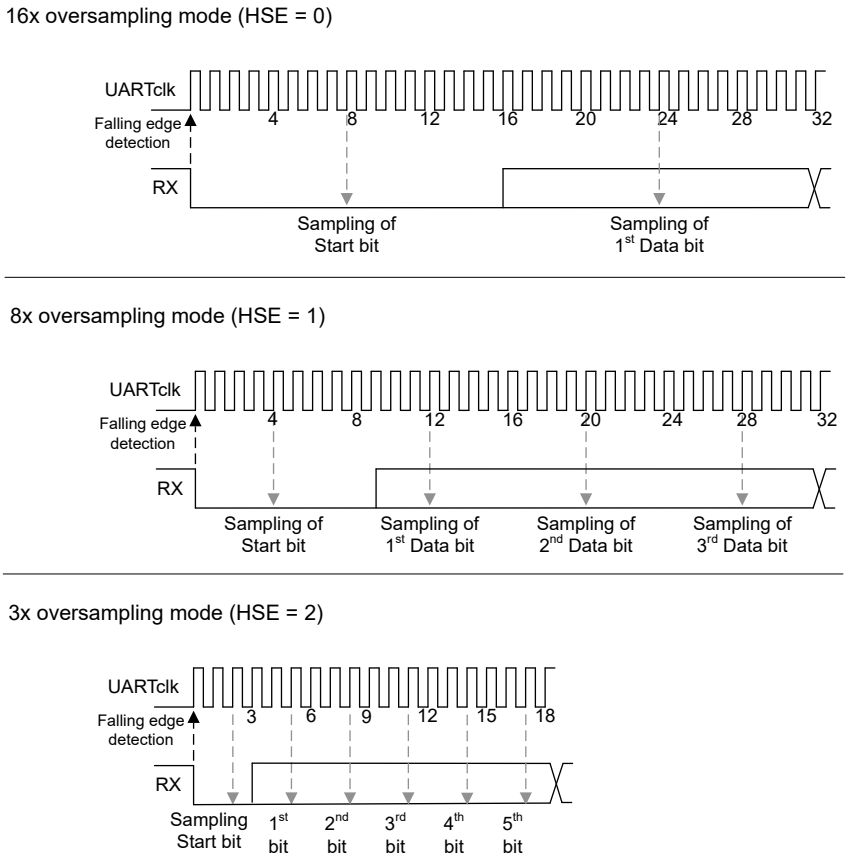



**Figure 28-3. UART Oversampling mode**

#### 28.2.2.4 Majority Voting

Majority voting provides noise immunity by sampling each bit 3 times in the center of the bit period.

For example, when the CTL0.HSE is set to ‘0’ with 16 oversampling; 7th,8th and 9th bit are sampled and majority value is considered as final value to be sampled.

When the CTL0.HSE is set to 1 with 8 oversampling; then 3rd ,4th and 5th bits are sampled and majority value is considered as final value to be sampled. The oversampling is only applicable for 16 and 8 oversampling.



Copyright © 2026 Texas Instruments Incorporated

When the 3 selected samples used in majority vote are not the same; RIS.NERR (noise error bit) is set; received data is transferred into the buffer/fifo in spite of noise error. NERR bit will get appended to received data before storing it in receive buffer/FIFO. Majority voting is supported for all data and parity bits. Majority voting or single sample feature can be selected using MAJVOTE control bit in CTL0 register.

**Note**

Majority voting should not be enabled for IRDA / MENC mode

#### 28.2.2.5 Baud Rate Generation

The baud-rate divisor is a 22-bit number consisting of a 16-bit integer and a 6-bit fractional part. The number formed by these two values is used by the baud-rate generator to determine the bit sample period. Having a fractional baud-rate divisor allows the UART to generate all of the standard baud-rates very accurately. An example calculation is demonstrated below, however tools like the TI System Configurator can perform these calculations automatically.

The 16-bit integer is loaded through the UART Integer Baud-Rate Divisor IBRD register and the 6-bit fractional part is loaded with the UART Fractional Baud-Rate Divisor FBRD register.

The baud-rate divisor can be calculated by using the following formula:

BRD = UART Clock / (Oversampling x Baud rate) (28)

UART Clock (UARTclk) is the clock output of the UART clock control logic, configured by CLKSEL and CLKDIV. Oversampling is selected by the HSE bit in the CTL0 register and can be configured to 16, 8 or 3.

- IBRD = INT(BRD), integer portion of the BRD
- FBRD = BRD % 64 , fractional portion of the BRD

The integer portion of the BRD is loaded into IBRD register. The 6-bit fractional number must be loaded into the FBRD register.

**Note**

When IBRD = 0, FBRD is ignored and no data gets transmitted or received by the UART. Similarly, when IBRD = 65535 (that is 0xFFFF), then FBRD must not be greater than zero. The ongoing transmission or reception is aborted if these requirements are not met.

**Baud Rate Configuration**

The following example shows a simple method to calculate IBRD.DIVINT and FBRD.DIVFRAC for a baud rate of 19200 bit/s:

UARTclk = CLKDIV / CLKDIV.RATIO = 40MHz

CRL0.HSE Oversampling = 16

Baud Rate = 19200bit/s

BRD = UARTclk / (Oversampling * Baud Rate)

BRD = 40MHz / (16 * 19200 bit/s) = 130.2083333

Use digits before the decimal to find IBRD.DIVINT:

**130**.208333 → IBRD.DIVINT = 130d → 82h

Use digits after the decimal to find FBRRD.DIFRAC:

130.**208333** → INT((.2083333 * 64) + 0.5) = INT(13.833333) = 13d → Dh



Copyright © 2026 Texas Instruments Incorporated

**Note**

The +0.5 adder ensures rounding to the closest integer value to keep the rounding error as small as possible

Baud rate settings (IBRD / FBRD) should only be updated when the module is IDLE. Application must ensure STAT.IDLE is ‘0’ prior to updating IBRD/FBRD settings.

#### 28.2.2.6 Data Transmission

Data received or transmitted is stored in two 16-element FIFOs, where the receive FIFO has an extra four bits per character for status information.

**Transmit data**

- Data is written into TXDATA
  - If UART is enabled, the data frame begins transmitting with the parameters indicated in the LCRH register
  - Data continues to be transmitted until there is no data left in the transmit FIFO
- The BUSY bit in the STAT register is asserted when data is written to the transmit buffer/FIFO and remains asserted while data is being transmitted
- BUSY bit is ‘0’ when transmit buffer/FIFO is empty (the last character has been transmitted to the shift register, including the stop bits)
  - BUSY bit is also is set during the generation of a BREAK signal

**Receive data**

- When the receiver is in an idle state and the data input transitions to a low level (START condition detected), the receive counter starts running and data bits are sampled according to the oversampling settings
  - START/STOP bits are not oversampled
- Each START bit is sampled to verify it reads ‘0’
  - If START reads '1', the state machine transitions back to IDLE state
- Each STOP bit is sampled to verify it reads ‘1’
  - If STOP reads '0', a framing error (FRMERR) is set
- When the full word is received, the data is stored in the receive buffer/FIFO along with error bits associated with it
- The BUSY flag is set high when a START condition is detected and is set to low after the full duration of the STOP bit

#### 28.2.2.7 Error and Status

For received data, the data byte and the 4-bit status (break, frame, parity, and overrun) is pushed onto the 12-bit wide receive buffer/FIFO. The error and status can be can be retrieved by reading RXDATA register as showing in Table 28-1. Each of the four below conditions can also set the interrupt flag RXINT if the corresponding interrupt bit is enabled in the IMASK register.

**Table 28-1. UART Error and Conditions**

| Error Condition | Bit Field | Description |
|---|---|---|
| Framing error | FRMERR | A framing error occurs when a low stop bit is detected. When two stop bits are used, both stopbits are checked for framing error. When a framing error is detected, the FRMERR bit is set |
| Parity error | PARERR | A parity error is a mismatch between the number of 1s in a character and the value of the parity bit. When an address bit is included in the character, it is included in the parity calculation. When a parity error is detected, the PARERR bit is set |
| Noise error | NERR | A noise error occurs when the 3 sampled values used for majority voting are not the same. When a noise error occurs, NERR bit is set |
| Break condition | BRKERR | A break is detected when all received data, parity, and stop bits are 0. When a break conditionis detected, the BRKERR bit is set |



Copyright © 2026 Texas Instruments Incorporated

The UART module flag status can also be checked by reading the STAT register.

**Table 28-2. UART Flag Status**

| Bit Field | Description |
|---|---|
| BUSY | This bit is set as soon as the transmit FIFO becomes non-empty (regardless of whether UART is enabled). In IDLELINE mode (when the CTL0.MODE field is configured for IDLELINE), the BUSY signal also stays set during the idle time generation. |
| RXFE | This bit is set when the receive FIFO is empty. |
| RXFF | This bit is set when the receive FIFO is full. |
| TXFE | This bit is set when transmit FIFO is empty. |
| TXFF | This bit is set when transmit FIFO is full. |
| CTS | This bit is set when CTS signal is asserted (low) and cleared when CTS signal is not asserted (high). |
| IDLE | This bit is set when an IDLE line has been detected in idle-line multiprocessor mode. The IDLE bit is used as an address tag for each block of characters. In idle line multiprocessor format, this bit is set when a received character is an address. |

#### 28.2.2.8 DMA Operation

The UNICOMM-UART provides an interface to the DMA module with separate channels for transmit and receive. The DMA operation of the UART is enabled through the UART Event and DMA registers. When DMA operation is enabled, the UART asserts a DMA request on the receive or transmit channel when the associated FIFO can transfer data. The DMA transfer requests are handled automatically by the DMA controller based on how the DMA channel is configured (burst size, transfer size, source address etc.).

- For the receive channel, a DMA transfer request is asserted when the amount of data in the receive FIFO is at or above the FIFO trigger level configured using the RXIFLSEL bit in IFLS register or if the receive timeout has triggered. In the receive timeout trigger case, the amount of data received up to that point is transferred.

**Note**

The DMA moves all data present in the RX FIFO once triggered, not just the amount configured by the RXIFLSEL field.

- For the transmit channel, a DMA transfer request is asserted whenever the transmit FIFO contains fewer characters than the FIFO trigger level configured using the TXIFLSEL bit in IFLS register.

#### 28.2.2.9 Loopback Operation

The UNICOMM-UART can be placed into an internal loopback mode for diagnostic or debug work by setting the LBE bit in the CTL0 register. In loopback mode, the UART operates with the following behavior:

- Data transmitted on the TX output is received on the RX input
- Data transmitted on the TX output is not propagated to the TX IO pin
- Data received on the RX IO pin is ignored



Copyright © 2026 Texas Instruments Incorporated

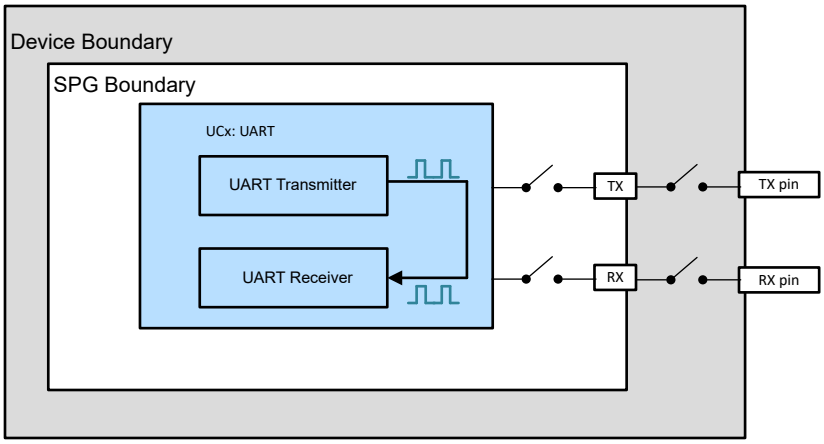



**Figure 28-4. UART Internal Loopback**

#### 28.2.2.10 Resets

The UNICOMM has a general software reset for the UNICOMM module. You can find more information in the UNICOMM Enables & Resets section.

**Hardware Reset Considerations**

A hardware reset also initializes the IO configuration. This sets the IOs to a high-impedance state, allowing the data lines to float. If this is critical for the application or connected devices on the SPI interface, external pull-up or pull-down resistors might be required.

#### 28.2.2.11 Initialization

Follow the high-level UNICOMM configurations in Section 27.3 before executing the below UART-specific configurations.

To enable and initialize the UART, perform the following initialization steps:

1. Clear the ENABLE bit in the CTL0 register before making any of the below configuration changes.
2. Use the baud-rate equation in Section 28.2.2.5 to calculate the IBRD and FBRD registers.
3. Write the integer portion of the BRD to the IBRD register.
4. Write the fractional portion of the BRD to the FBRD register.
5. Write the desired communication mode settings to the CTL0 register (all CTL0 fields other than ENABLE).
6. Write the desired serial parameter configurations to the LCRH register.
7. Configure the desired FIFO trigger levels in the IFLS register.
8. Enable desired interrupts and/or DMA event by using CPU_INT, DMA_TRIG_RX, DMA_TRIG_TX group IMASK registers.
9. (Optionally) Configure the emulation mode for the peripheral in the PDBGCTL register.
10. Enable the UART by setting the CTL0.ENABLE bit.

**Note**

Before UNICOMM-UART is setup or configuration changes, the CTL0.ENABLE bit must be cleared to avoid unpredictable behavior during the updates or for the first data receive or transmitted afterward.

#### 28.2.2.12 FIFO Operation

The UNICOMM-UART has two FIFOs (one each for transmit/receive) with configurable depth. FIFOs are accessed via TXDATA & RXDATA registers.

**FIFO Status Bits**

Each module will support the below FIFO status bits:



Copyright © 2026 Texas Instruments Incorporated

- Receive FIFO Full: when all elements are populated
- Receive FIFO Empty: when there are no elements in the HEAD
- Transmit FIFO Full: when all elements are populated
- Transmit FIFO Empty: when are no elements in the FIFO

In addition to this, FIFO clear status – RXCLR/TXCLR will also be present

**RIS.RXINT Based on FIFO Threshold Settings**

On transmit side, data is written by the CPU/DMA master at top most empty FIFO entry and then transferred to serializer. Thresholds are as per elements that can be written into the FIFO.

1. RX_FIFO full: when all elements are populated
2. RX_FIFO >= ¼ full
3. RX_FIFO >= ½ full
4. RX_FIFO >= ¾ full
5. RX_FIFO almost empty: <= 1 element
6. RX_FIFO almost full: >= (N-1) elements
7. **RX_FIFO not empty: opposite of empty**

**RIS.TXINT based on FIFO threshold settings**

On transmit side, data is written by the CPU/DMA master at topmost empty FIFO entry and then transferred to serializer. Thresholds are as per elements that can be written into the FIFO.

1. TX_FIFO empty: when there no elements in the FIFO
2. TX_FIFO <= ¾ empty
3. TX_FIFO <= ½ empty
4. TX_FIFO <= ¼ empty
5. TX_FIFO almost empty: <=1 element
6. TX_FIFO almost full: >= (N-1) elements
7. **TX_FIFO not full: opposite of full**

**Note**

- Receive FIFO Almost Empty and Transmit FIFO Almost Full options are provided in IFLS setting; however, these should not be used for generating flags based on watermark settings or for DMA usage. Application has to ensure that correct FIFO Thresholds are used.
- Debugger initiated reads should not pop contents out of the FIFO. However, debugger initiated writes into the transmit FIFO should be treated as valid entries.

**Clearing FIFO Contents**

**FIFO clear**: to clear contents of the FIFO, sequence needs to be used:

- Write ‘1’ to CLEAR control bit (TX_CLEAR / RX_CLEAR)
- Wait for clear to complete, i.e. CLEAR status bit reflects ‘1’; poll/check for CLEAR status bit to reflect ‘1’
- Write ‘0’ to CLEAR control bit

**Note**

Randomly changing FIFO levels when FIFO has contents might lead undesired behavior. It is recommended that FIFO is cleared before changing FIFO thresholds

**Hardware monitors empty, full and overrun conditions**

The STAT register contains empty and full flags (TXFE, TXFF, RXFE, and RXFF bits), and the RIS register shows overrun status of the receive FIFO via the OVRERR bit. There is no indicator for a transmit FIFO overrun. A write is just lost, in case it overruns the transmit FIFO. If the FIFOs are not present, the empty and full flags are set according to the status of the 1-byte-deep holding registers.



Copyright © 2026 Texas Instruments Incorporated

**Note**

*Data will be lost if the FIFO is full and overflow flag bit will be set*

Out of reset, both FIFOs are configured to trigger an interrupt at half-level mark.

FIFO integrity is not guaranteed if software disables the module in the middle of a transmission with data in the FIFO, and then re-enables it. When the module gets reset the content of the FIFO are also cleared

#### 28.2.2.13 Interrupt and Events Support

The UART module contains three event publishers and no event subscribers. One event publisher (CPU_INT) manages UART interrupt requests (IRQs) to the CPU subsystem through a static event route. The second and third event publishers (DMA_TRIG_RX, DMA_TRIG_TX) are used to setup the trigger signaling for the DMA through DMA event route.

**Table 28-3. UART Events**

| Event | Type | Source | Destination | Route | Configuration | Functionality |
|---|---|---|---|---|---|---|
| CPU Interrupt | Publisher | UART | CPU Subsystem | Static route | CPU_INT registers | Fixed interrupt route from UART to CPU |
| DMA Trigger | Publisher | UART | DMA | DMA event route | DMA_TRIG_RX registers | Fixed interrupt route from UART to DMA |
| DMA Trigger | Publisher | UART | DMA | DMA event route | DMA_TRIG_TX registers | Fixed interrupt route from UART to DMA |

##### *28.2.2.13.1 CPU Interrupt Event Publisher (CPU_INT)*

The UART module provides 19 interrupt sources which can be configured to source a CPU interrupt event. In order of decreasing interrupt priority, the CPU interrupt events from the UART are:

**Table 28-4. UART CPU Interrupt Event Conditions (CPU_INT)**

| IIDX STAT | Name | Description |
|---|---|---|
| 0x01 | RTOUT | UART receive timeout interrupt. This interrupt is asserted when the RX FIFO is not empty, and no further data is received specified time in the IFLS.RXIFSEL field. |
| 0x02 | FRMERR | UART framing error interrupt. |
| 0x03 | PARERR | UART parity error interrupt. |
| 0x04 | BRKERR | UART break error interrupt. |
| 0x05 | OVRERR | UART receive overrun error interrupt. |
| 0x06 | RXNE | Falling edge on RX interrupt, this interrupt triggers when there is a falling edge on RX line. |
| 0x07 | RXPE | Rising edge on RX interrupt, this interrupt triggers when there is a rising edge on RX line. |
| 0x08 | LINC0 | LIN capture 0 match interrupt, this interrupt triggers when the defined capture 0 value is reached in LIN counter. |
| 0x09 | LINC1 | LIN capture 1 match interrupt, this interrupt triggers when the defined capture 1 value is reached in LIN counter. |
| 0x0A | LINOVF | LIN counter overflow interrupt, this interrupt triggers when the 16bit LIN counter overflows. |
| 0x0B | RXINT | UART receive interrupt. |
| 0x0C | TXINT | UART transmit interrupt. |
| 0x0D | EOT | UART end of transmission interrupt indicates that the last bit of all transmitted data and status has left the serializer and without any further data in the TX FIFO. |
| 0x0E | ADDR_MATCH | Address match interrupt, used in protocols with address to indicate address match happened. |
| 0x0F | CTS | UART clear to send interrupt, indicate the CTS signal status. |



Copyright © 2026 Texas Instruments Incorporated

**Table 28-4. UART CPU Interrupt Event Conditions (CPU_INT) (continued)**

| IIDX STAT | Name | Description |
|---|---|---|
| 0x10 | DMA_DONE_RX | This interrupt is set if the RX DMA channel sends the DONE signal. |
| 0x11 | DMA_DONE_TX | This interrupt is set if the TX DMA channel sends the DONE signal. |
| 0x12 | NERR | The noise error interrupt is set when the 3 sampled values used for majority voting are not the same |
| 0x15 | LTOUT | UART line timeout interrupt, This interrupt is asserted when no further data is received specified time in the IFLS.RXTOSEL bits. |

The CPU interrupt event configuration is managed with the CPU_INT event management registers. See the *Event Registers* chapter.

The receive timeout interrupt is asserted when the RX FIFO is not empty, and no further data is received specified time in the IFLS.RXIFSEL bits. The receive timeout interrupt is cleared either when the FIFO becomes empty through reading all the data (or by reading the holding register), by reading the interrupt index from IIDX or when a 1 is written to the corresponding bit in the ICLR register.

The receive interrupt (RXINT) is set when the RX FIFO progresses through the programmed trigger level. The receive interrupt is cleared by reading data from the RX FIFO until it becomes less than the trigger level, by reading the interrupt index from IIDX, or by writing a '1' to the RXINT bit in ICLR.

The transmit interrupt (TXINT) is set when the TX FIFO progresses through the programmed trigger level. The transmit interrupt is based on a transition through level, therefore the FIFO must be written past the programmed trigger level otherwise no further transmit interrupts will be generated. The transmit interrupt is cleared by writing data to the TX FIFO until it becomes greater than the trigger level, by reading the interrupt index from IIDX, or by writing a '1' to the TXINT bit in ICLR.

#### *28.2.2.13.2 DMA Trigger Publisher (DMA_TRIG_RX, DMA_TRIG_TX)*

The DMA_TRIG_TX and DMA_TRIG_RX group registers are used to setup trigger signaling for the DMA. These registers (IMASK, RIS, MIS, ISET, ICLR) are present for any UCx configured as UART and can be found in the corresponding UNICOMMUART_REGS register structures. See *DMA Trigger* for how the DMA trigger Event works and can be configured. Each DMA channel can be triggered by any of the conditions listed in the *DMA Trigger Source Options* and can send a DMA_DONE signal to the corresponding UCx module.

**Table 28-5. UART DMA Trigger Condition (DMA_TRIG_RX)**

| Name | Description |
|---|---|
| RTOUT | UART receive timeout trigger, This interrupt is asserted when the RX FIFO is not empty, and no further data is received specified time in the IFLS.RXTOSEL bits. |
| RXINT | UART RX FIFO level trigger. |

The receive timeout trigger is asserted when the RX FIFO is not empty, and no further data is received for the amount of time programmed by the IFSEL.RXTOSEL field. The receive timeout interrupt is cleared when the RX FIFO becomes empty through reading all the data, by reading the interrupt index from IIDX, or when a 1 is written to the RTOUT bit in the ICLR register.

The receive interrupt (RXINT) is set when the RX FIFO reaches the programmed trigger level. The receive interrupt is cleared by reading data from the RX FIFO until the level becomes less than the trigger level, by reading the interrupt index from IIDX, or by writing a '1' to the RXINT bit in ICLR.

**Table 28-6. UART DMA Trigger Condition (DMA_TRIG_TX)**

| Name | Description |
|---|---|
| TXINT | UART TX FIFO level trigger.. |

The transmit interrupt (TXINT) is set when the TX FIFO progresses through the programmed trigger level. The transmit interrupt is based on a transition through level, therefore the FIFO must be written past the programmed trigger level otherwise no further transmit interrupts are generated. The transmit interrupt is cleared by writing



Copyright © 2026 Texas Instruments Incorporated

data to the TX FIFO until the level becomes greater than the trigger level, by reading the interrupt index from IIDX, or by writing a '1' to the TXINT bit in ICLR.

#### 28.2.2.14 Emulation Modes

The module behavior while the device is in debug mode is controlled by the FREE and SOFT bits in PDBGCTL register.

When the device is in debug mode and set into halt mode below behavior can be configured.

**Table 28-7. Debug Mode Peripheral Behavior**

| PDBGCTL.FREE | PDBGCTL.SOFT | Function |
|---|---|---|
| **1** | **x** | Modules continues operation |
| **0** | **1** | Module stops after the next transfer has been finished |
| **x** | **0** | SOFT = 0h is **not** supported |

**Graceful Halt**

When emulation mode is set to graceful halt and a halt condition occurs, the IP stops communications after the ongoing frame has finished.

Transmitter

1. IP finishes transmitting the ongoing frame; FSM returns to IDLE and stop
2. External line is driven to IDLE condition
3. Removing halt resumes communication

Receiver

1. IP receives ongoing frame; FSM returns to IDLE and stop
2. Any new messages are not be processed
3. After halt is removed; FSM begins communications
4. FSM can start reception in the middle of a frame

**Note**

Immediate halt is not supported

### *28.2.3 Additional Protocol and Feature Support*

#### 28.2.3.1 Local Interconnect Network (LIN) Support

The Local Interconnect Network (LIN) protocol standard is based on the UART serial data link format. The LIN communication protocol is a single-commander/multiple-responder topology with message identification for multicast transmission between network nodes.



Copyright © 2026 Texas Instruments Incorporated

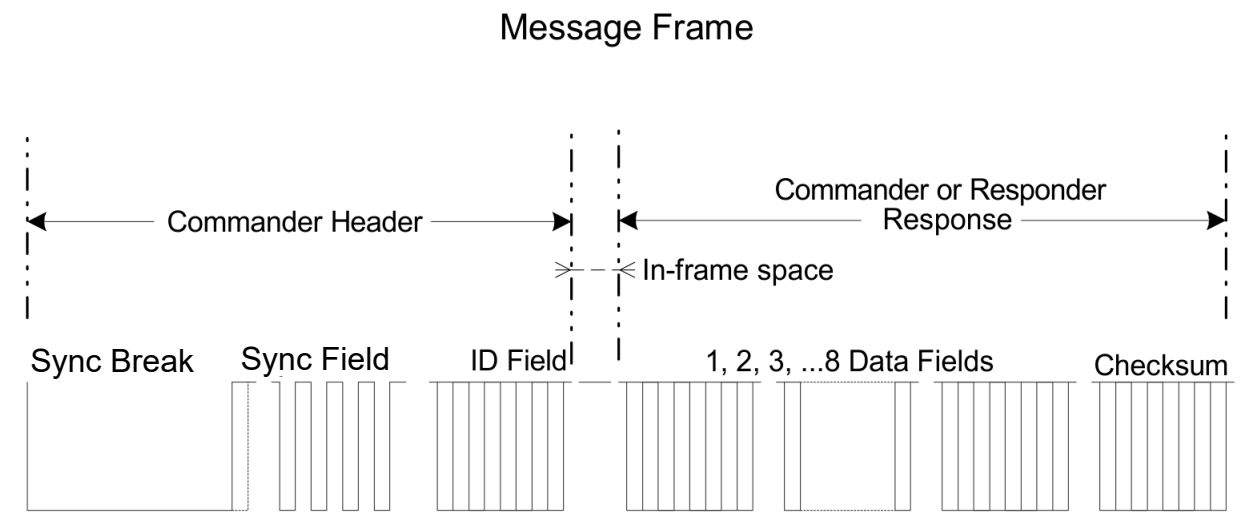



**Figure 28-5. LIN Protocol Message Structure**

The LIN commander issues a BREAK field, SYNC field, and PID (Protected Identifier) field at the start of every frame. The four LIN registers are used so that the LIN responder software driver can reasonably detect BREAK and SYNC fields and measure the necessary timing parameters to adjust the baud rate or detect an error.

See the *LIN Standards and Specifications* page for more information about the LIN protocol standards. The UNICOMM-UART module in LIN mode can operate as a LIN commander or a LIN responder when configured properly. To support the LIN protocol while reducing the number of interrupts needed in software, the following hardware enhancements are included in the UNICOMM-UART module:

- **LIN Control** - LINCTL
  - Configuration of LIN mode behavior.
- **LIN Counter** - LINCNT
  - 16-bit up-counter clocked by the UARTclk.
  - CPU_INT.IMASK.LINOVF flag - Interrupt capability on counter overflow.
- **LIN Capture 0** - LINC0
  - 16-bit Capture 0 with two configurable modes:
    - If LINCTL.LIN0CAP is set - Capture of the LINCNT value on RX pin falling edge. Interrupt capability on capture (CPU_INT.IMASK.LINC0).
    - If LINCTL.LIN0_MATCH is set - Compare of the LINCNT with interrupt capability on match (CPU_INT.IMASK.LINC0).
- **LIN Capture 1** - LINC1: 16-bit Capture 1.
  - Capture LINCNT value on RX pin rising edge. Interrupt capability on capture (CPU_INT.IMASK.LINC0).

Additionally, software can manually control the logic level output of the UNICOMM-UART TXD pin by writing the CTL0.TXD_OUT bit when the CTL0.TXD_OUT_EN is set to 1 with CTL0.TXE set to 0 (transmits disabled). To generate signals such as wake signals, TX pin can be configured to TXD_OUT and TXD_CTL_EN bits in register CTL0 to be software controlled by setting TXD_CTL_EN bit to 1, the TX output pin can be controlled by the TXD_OUT bit if the UART transmit is disabled (CTL0.TXE is cleared).

**Note**

Full hardware support of the LIN protocol is not available in UNICOMM-UART. The above registers are used to aid in a software LIN implementation of the LIN-specific features added on top of the UART protocol.



Copyright © 2026 Texas Instruments Incorporated

#### 28.2.3.1.1 LIN Commander Transmit

The below highlights how a UNICOMM-UART can implement each portion of the LIN Commander transmission using the registers available in the module.

**Transmitting the Break Field**

The break signal can be sent automatically by the UNICOMM-UART by setting the BRK bit in LCRH register for greater than 13*Tbits. This bit needs to be set before any data is written into the transmit data register TXDATA.

**Transmitting the Sync Field**

The sync field can be sent by writing 0x55 to the TXDATA register to be transmitted using the standard UNICOMM-UART transmitter.

**Transmitting the PID Field**

Write the ID byte to the TXDATA register to be transmitted using the standard UNICOMM-UART transmitter. Make sure CTL0.TXE is set.

**Transmitting the Data Fields**

Write the data bytes to TXDATA register to be transmitted using the standard UNICOMM-UART transmitter. Make sure CTL0.TXE is set.

**Transmitting the Checksum**

Calculate and write the checksum byte to the TXDATA register to be transmitted using the standard UNICOMM-UART transmitter. Make sure CTL0.TXE is set.

#### 28.2.3.1.2 LIN Responder Receive

**Receiving the Break Field**

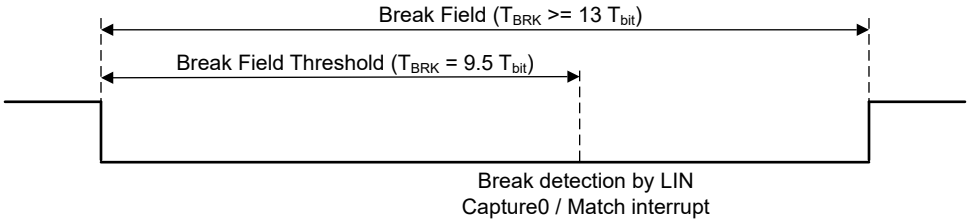



**Figure 28-6. LIN Break Field Detection**

Use the break field detection and compare modes to receive and validate the BREAK field of a LIN message. The following configurations are needed:

1. Initialize the LIN counter register to 0 (LINCNT = 0)
2. Enable counter compare match mode (LINCTL.LINC0_MATCH = 1)
3. Load LINC0 (Counter Capture 0 register) with the counter value corresponding to 9.5 x $T_{bit}$
4. Enable the LINC0 match interrupt (CPU_INT.IMASK.LINC0 = 1)
5. Setup LIN count control (configure the LINCTL register):
   - Enable count while low signal on RXD pin (LINCTL.CNTRXLOW = 1)
   - Enable LIN counter clearing on RXD pin falling edge (LINCTL.ZERONE = 1)
   - Enable LIN counter (LINCTL.CTRENA = 1)

   Optional:
   - Enable the rising edge on UART TX signal interrupt (CPU_INT.IMASK.RXPE = 1), when the RXPE interrupt fires, the software can read the LINCNT directly to see the BREAK field timing.
   - Enable the LIN counter overflow interrupt (CPU_INT.IMASK.LINOVF = 1) to detect the BREAK field is too long and overflows 16-bit counter. The timeout can be calculated as $t_{Timeout}$= $2^{16}$/ UARTclk.



Copyright © 2026 Texas Instruments Incorporated

### Receiving the Sync Field

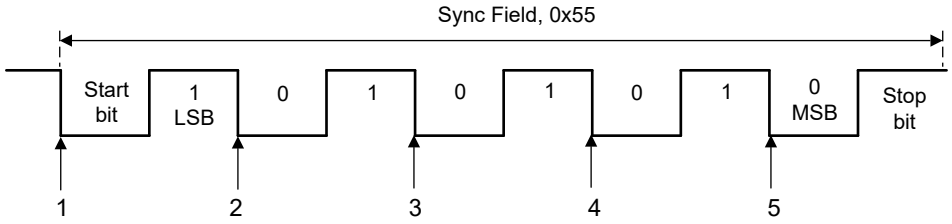



**Figure 28-7. LIN Sync Field Detection**

Sync field validation is required to verify accuracy of the LIN header and to calculate the commander baud rate. The sync field consists of the data 0x55 inside a byte field. After software detects a valid BREAK field, the counter can be set to measure the SYNC field. The LINCNT is set to 0 on the start bit falling edge and counts continuously. The LINC0 capture or RX falling edge interrupt fires at the falling edges of the RX line. During interrupt processing, software can measure the individual HIGH/LOW times of the bits themselves with the values in the LINC0 and LINC1 registers to verify that all of the timings are valid.

The following flow describes a LIN sync field validation procedure:

1. Initialize LIN counter to 0 (LINCNT = 0) after detecting a valid break field.
2. Enable interrupt on RX falling edge (CPU_INT.IMASK.RXNE = 1)
3. Setup LIN count control (LINCTL):
   - Enable LIN counter capture on raising RX edge (LINCTL.LINC1CAP = 1)
   - Enable LIN counter capture on falling RX edge (LINCTL.LINC0CAP = 1)
   - Enable LIN counter clearing on RX falling edge (LINCTL.ZERONE = 1)
   - Enable LIN counter (LINCTL.CTRENA = 1)

Actions at each falling edge of the RX line for the sync field as showing below:

1. LIN counter is set to 0 and start counting on the falling RX edge. (LINCTL.ZERONE = 1)
2. RX falling edge interrupt trigger (RXNE):
   - Read capture register LINC0 (falling edge) and LINC1 (rising edge) values
   - Verify bit times
3. RX falling edge interrupt trigger (RXNE):
   - Read capture register LINC0 (falling edge) and LINC1 (rising edge) values
   - Verify bit times
4. RX falling edge interrupt trigger (RXNE):
   - Read capture register LINC0 (falling edge) and LINC1 (rising edge) values
   - Verify bit times
5. RX falling edge interrupt trigger (RXNE):
   - Read capture register LINC0 (falling edge) and LINC1 (rising edge) values
   - Verify bit times
   - Calculate the proper baud rate to set. Software must set the baud rate before the start bit of the PID field after sync field.

On each interrupt occurrence, the capture registers must be read and the bit times need to be validated by the application software. In case of a bit time verification error, the sync field analysis process must be aborted and the application software must switch back to break detection.

In case of errors like a breaking commander communication during sync field detection, a timeout interrupt can be generated by enabling the LIN counter overflow (IMASK.LINOVF = 1). When the counter overflows, the interrupt handler can abort the sync field analysis and switch back to break detection. The time the counter overflow interrupt occurs can be calculated as $t_{Timeout}$= $2^{16}$/ UART clock.



Copyright © 2026 Texas Instruments Incorporated

A new control bit LINCTL.EN_FRM_ERR is added to enable frame error generation in LIN MODE; i.e. a frame error is detected if LINCTL.EN_FRM_ERR is set and device is in LIN mode. This register bit needs to be enabled after SYNC field detection & should be disabled post checksum detection. In case of EVENT-TRIGGERED frames, if there is a collision, FRAME ERROR will be reported in addition to BREAK FIELD being detected. In these cases, application is expected to ignore the FRAME ERROR.

**Note**

The sync field is automatically stored in the RX FIFO and can be misread as the PID. Therefore, flush the RX FIFO after the sync field is received and before the PID is received.

**Receiving the PID Field**

Read the PID byte from the RXDATA register using the standard UNICOMM-UART receiver. Make sure CTL0.RXE is set.

**Receiving the Data Fields**

Read the data fields from the RXDATA register using the standard UNICOMM-UART receiver. Make sure CTL0.RXE is set.

**Receiving the Checksum Field**

Read the checksum byte from the RXDATA register using the standard UNICOMM-UART receiver and decode the value in software. Make sure CTL0.RXE is set.

#### *28.2.3.1.3 LIN Wakeup*

Per the LIN specification, any node in a sleeping LIN cluster can request a wake up by transmitting a wake up signal. This signal type is shown in Figure 28-8.

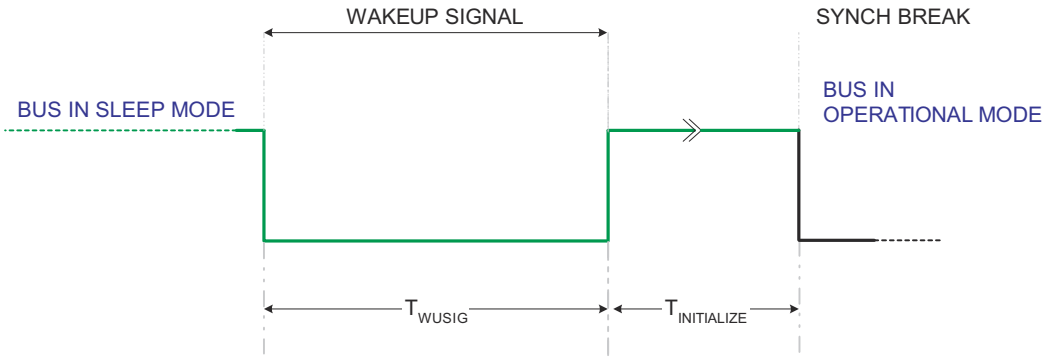



**Figure 28-8. LIN Wakeup**

To generate a wake signal on the TX pin with the UNICOMM-UART as a LIN controller, the application must use the CTL0.TXD_CTL_EN and CTL0.TXD_OUT bits. Set the TXD_CTL_EN bit to enable register control of the TXD pin, then clear the TXD_OUT bit for the duration of $T_{WUSIG}$ .

#### 28.2.3.2 DALI Protocol

Digital Addressable Lighting Interface (DALI) is an International Standard (IEC-62386) lighting control system for providing a single interface for lighting controllers.

The UART module supports sending and receiving forward and backward frames. The timing between any forward and backward frame sequence needs to be handled and checked by software.

**Transmitting a forward or backward frame:**



Copyright © 2026 Texas Instruments Incorporated

When transmitting a forward frame, a second byte must be written to the buffer before first byte has been shifted out (Otherwise, stop bits are sent like a backward frame). The hardware then sends the two bytes without inserting the stop bits.

**Receiving data:**

The UART module in DALI mode checks the 9th bit after the start bit to detect a forward or backward frame. If this bit has a change of phase (= no stop bit) a forward frame is detected and:

- Address compare is done in hardware based on AMASK
- Address and data are transferred into the RX buffer if there is an address match

Otherwise a backward frame is detected and the data is transferred into the RX Buffer without setting the ADDR_MATCH interrupt.

The AMASK register can be used as group assignment used during multicast operation with the MSB to indicate if the device is part of a DALI group. To enable the UART in DALI mode to respond on all ADDRESS the AMASK register needs to be cleared.

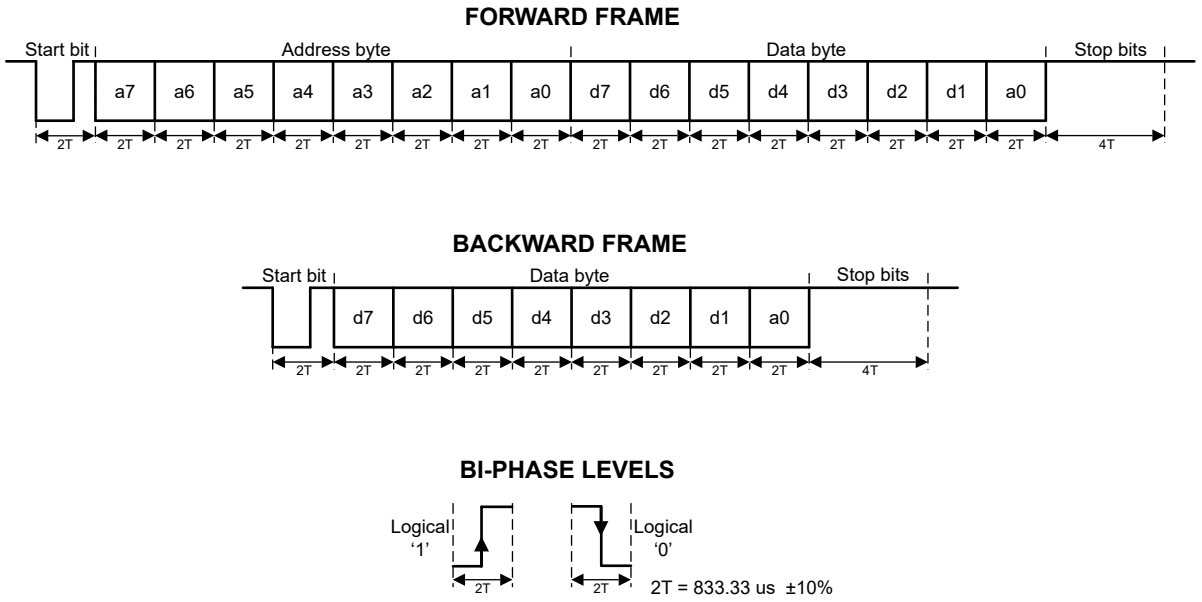



**Figure 28-9. DALI Protocol**

The limits for the times T shall be: 334 µs < T < 500 µs according to standard IEC 62386-102. DALI mode requires Manchester encoding to be enabled. When using DALI mode (CTL0.MODE set to DALI), the CTL0 and LCRH register must be set to:

- 8-bit word length (WLEN bit)
- No parity / 2 Stop Bit
- Manchester encoding enabled
- Baud-rate configuration set to match 2T = 833.33us
- DALI mode requires the FIFO to be enabled

**Note**

When UART is configured as a DALI Control Device: backward frame is stored in the device only if address match is enabled and there is an address match. Upon an address match, both forward frame and backward frame are stored in the receive FIFO and ADDR_MATCH interrupt flag is set.

**Note**

DALI timing support is restricted to +/- 2.0%



Copyright © 2026 Texas Instruments Incorporated

#### 28.2.3.3 IrDA Encoding and Decoding

This section applies to UNICOMM-UART configurations which support the UART-IRDA feature.

When IREN bit in IRCTL register is set, the IrDA encoder and decoder are enabled and provide hardware bit shaping for IrDA communication. IrDA encoding & decoding should only be used with UART mode (CTL0.MODE is 0). Only odd dividers of CLKDIV are support in this mode.

### IrDA Encoding

The encoder sends a pulse for every zero bit in the transmit bit stream coming from the UART (see Figure 28-10). The pulse duration is defined by IRTXPL bits specifying the number of one-half clock periods of the clock selected by IRTXCLK bit.

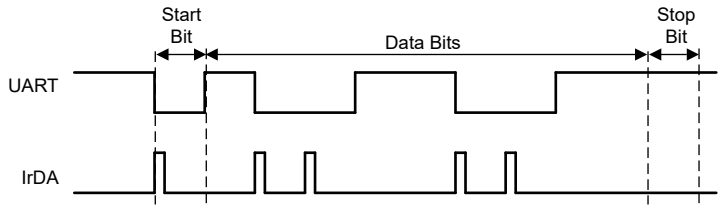



**Figure 28-10. IrDA Protocol**

### IrDA Decoding

The decoder detects high pulses when IRRXPL = 0. Otherwise, it detects low pulses.

A programmable digital filter stage can be enabled by setting UARTx.GFCTL.DGFSEL > 0. When IRCTL.IREN is set, also the digital glitch filter should be set so that only pulses longer than the programmed filter length are passed and shorter pulses are discarded. (See the Glitch Suppression chapter on how to set the filter.)

#### 28.2.3.4 Flow Control

In UART mode (CTL0.MODE set to 0), hardware flow control between two devices is available.

- Connect the Request-to-Send (RTS) output on the transmitter to the Clear-to-Send (CTS) input on the receiver
- Connect the RTS output on the receiver to the CTS input on the transmitter

The RTS output signal on the transmitter is active low, so the CTS input on the reciever expects a low signal on a send request.

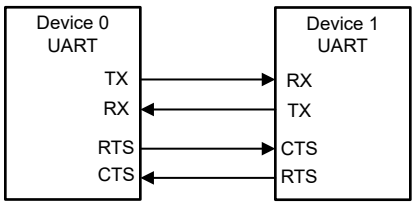



**Figure 28-11. Flow Control**

- The transmitter may only transmit data when the receiver asserts CTS low
- When RTS flow control is enabled, the receiver RTS output signal indicates the state of the receive buffer/ FIFO
- CTS of the transmitting device remains asserted low until, the preprogrammed watermark level is reached in case of FIFO and receive buffer is full in case of a single buffer, indicating that the receiver has no space to store additional characters

The CTSEN and RTSEN bits in the CTL0 register specify the flow control mode.



Copyright © 2026 Texas Instruments Incorporated

**Table 28-8. Flow Control Enable**

| CTSEN | RTSEN | Description |
|---|---|---|
| 1 | 1 | RTS and CTS flow control enabled |
| 1 | 0 | Only CTS flow control enabled |
| 0 | 1 | Only RTS flow control enabled |
| 0 | 0 | Both RTS and CTS flow control disabled |

The RTS bit can be driven by hardware or software. When RTSEN is 1 (i.e. hardware-controlled mode), the value of CTL0.RTS bit is ignored and RTS output signal is generated by hardware trigger levels as described below.

**RTS flow control:**

- The RTS flow control logic is linked to the programmable RX FIFO watermark levels, it can be configured using the IFLS register.
- When RTS flow control is enabled, the RTS is asserted (set low) until the RX FIFO is filled up to the watermark level
  - When the RX FIFO watermark level is reached, the RTS signal is de-asserted (set high), indicating that there is no more room to receive any more data.
  - The transmitter's transmission is expected to cease after the current character has been transmitted
- The RTS signal is reasserted (set low) when data has been read out of the RX FIFO to less than the watermark level.
- If RTS flow control is disabled and the UNICOMM-UART is still enabled, then data is received until the RX FIFO is full, or data is no longer transmitted to it

**Note**

The RTS signal is de-asserted when the FIFO watermark level is reached by putting the last received character into the FIFO. This means that back-to-back transmissions could result in another character transfer after de-asserting RTS. In such cases, the watermark level should be set to one level lower to ensure all data can be received and put into the FIFO. In the case of a single buffer, wait a pre-determined amount of time (or wait for an RTS transition to ensure the receive buffer has been read and a new byte can be sent)

**CTS flow control:**

- If CTS flow control is enabled, transmitter checks CTS signal before transmitting the next byte.
  - If CTS signal is asserted (low), it transmits the byte, otherwise transmission does not occur.
  - The data continues to be transmitted while CTS is asserted (low), and transmit FIFO/buffer is not empty
  - If transmit FIFO/buffer is empty and the CTS signal is asserted (low), no data is transmitted
- If CTS signal is de-asserted (high) and CTS flow control is enabled, then, the current character transmission is completed before stopping
- If CTS flow control is disabled and the UART is enabled, then the data continues to be transmitted until the transmit buffer/FIFO is empty

#### 28.2.3.5 Manchester Encoding and Decoding

This section applies to UNICOMM-UART configurations which support the UART-DALI-MANCHESTER feature.

UART provides options to receive and transmit Manchester encoded data. The function is enabled by the MENC bit in CTL0 register. With the invert function in GPIO control module the output signals can be inverted to generate the G. E. Thomas compliant waveform.

The output signal is generated by XORing the data with UART clock signal. The UART clock needs to be double the speed of the baud-rate. So for the data transmit there is an edge at the beginning and the middle of each data bit. For the receive signal the edge in the middle of the bit is detected to decode the RX data.



Copyright © 2026 Texas Instruments Incorporated

#### 28.2.3.6 RS485 Support

The RS485 drive can control an external RS485 PHY. The RTS I/O is used in this mode as the drive enable. The signal is set automatically to high once a data transmit is started. This signal stays high between back-to-back byte transmits. If a data receive operation is ongoing, a new transmit must be delayed until this data has been received and the drive enable has been set to transmit.

The data exchange sequence is as follows (also see RS485 Data Exchange figure):

- Wait until an ongoing receive has been finished.
- Activate drive enable to transmit on RTS Pin
- Send data (one or more bytes)
- Wait till an ongoing receive has been finished.
- Deactivate drive enable to transmit on RTS Pin

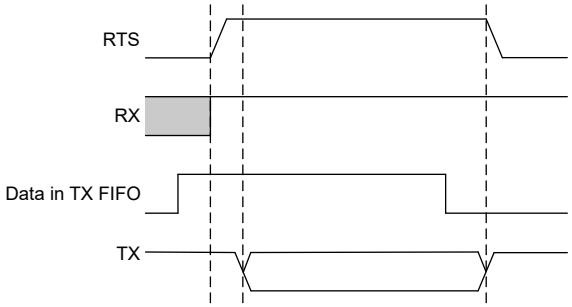



**Figure 28-12. RS485 Data Exchange**

Two bit fields to the LCRH register define the setup and hold time of the external driver direction control:

- EXTDIR_SETUP bits define number of UARTclk between RTS assertion and beginning of START bit
- EXTDIR_HOLD bits define number of UARTclk between STOP (entire bit duration) and RTS de-assertion

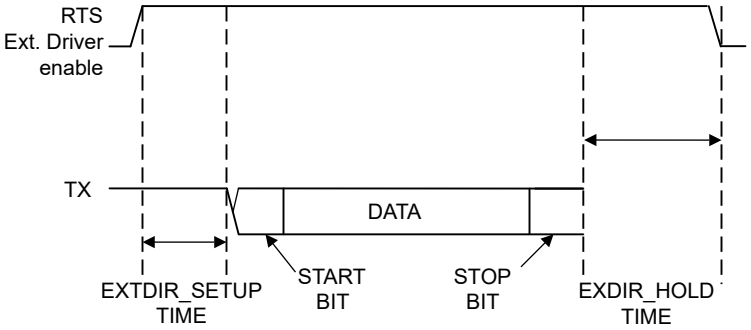



**Figure 28-13. RS485 external driver control**

#### 28.2.3.7 Idle-Line Multiprocessor

This section applies to UNICOMM-UART configurations which support the UART-IDLELINE-MULTIPROC feature.

Idle-line multiprocessor format is selected when IDLELINE is set in the CTL0.MODE register bits.

- Blocks of data are separated by an idle time on the transmit or receive lines.
  - An idle receive line is detected when ten or more continuous ones (marks) are received after the one or two stop bits of a character
- The baud-rate generator is switched off after reception of an idle line until the next start edge is detected. When an idle line is detected, the IDLE bit in UARTx.STAT is set
- In Idle-Line mode the UART receiver operates in no parity mode and the UART word length (UARTx.LCRH.WLEN) must be set to 8bit



Copyright © 2026 Texas Instruments Incorporated

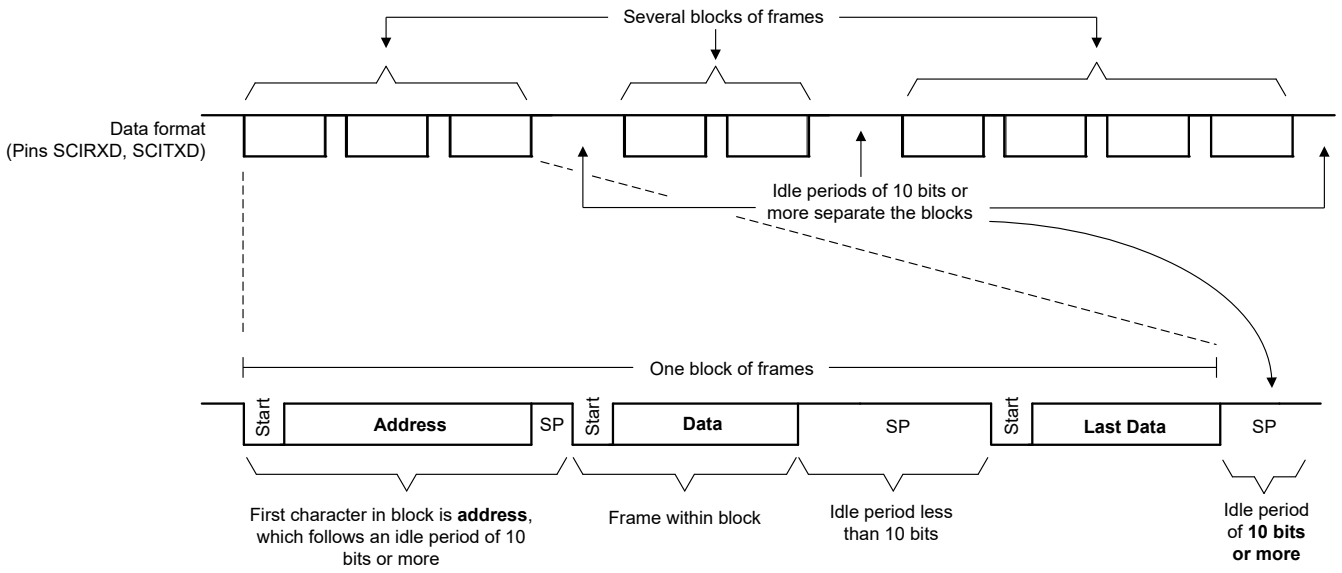



**Figure 28-14. Idle-Line Multiprocessor**

The first character received after an idle period is an address character. The IDLE bit in UARTx.STAT register is used as an address tag for each block of characters.

If an address character is received it is compared against the ADDR register with the AMASK applied. If the received character matches, the address character and all following received characters are transferred into the RXDATA buffer/FIFO and interrupts are generated until the next address without a match is received. The IDLE bit in STAT register is automatically cleared when the address does match; otherwise the IDLE bit is set until the address is matched.

When SENDIDLE control bit is set, the UART inserts an idle period of 11 bit times on the bus. Application has to ensure that there are no active transactions on the transmit line prior to setting SENDIDLE bit.

Once SENDIDLE bit is set, application has to wait for SENDIDLE status bit to be asserted and then clear the SENDIDLE control bit to proceed. The next transfer can then begin with an address character.

The following procedure sends out an idle frame to indicate an address character followed by associated data:

1. Set SENDIDLE then write the address character to TXDATA. TXDATA must be ready for new data (TXINT interrupt must be 1). This generates an idle period of exactly 11 bits followed by the address character. SENDIDLE is reset automatically when the address character has been transferred (all bits are sent out of shift register).
2. Write desired data characters to TXDATA. TXDATA must be ready for new data (TXINT interrupt must be 1). The data written to TXDATA is transferred to the shift register and transmitted as soon as the shift register is ready for new data.

The idle-line time (10 bit periods) must not be exceeded between address and data transmission or between data transmissions. Otherwise, the transmitted data is misinterpreted as an address.

When using IDLELINE mode, the UART Word Length in the LCRH register must be set to:

- 8-bit word length (WLEN bits 6:5 configured to 0x3)

BUSY bit in IDLELINE:

- TX: BUSY is set during the generation of an IDLELINE signal and while the address and data bytes are sent.
- RX: The BUSY signal while receiving data and till first 10 idle bits are received.



Copyright © 2026 Texas Instruments Incorporated

#### 28.2.3.8 9-Bit UART Mode

9-bit mode is enabled by setting the MODE field to ADDR9BIT in the CTL0 register. This feature is useful in a multi-drop configuration where a single controller connected to multiple peripherals can communicate with a particular peripheral through the peripheral's programmed address or a set of addresses, along with a qualifier for an address byte. In 9-bit UART mode, the parity enable/mode bits are ignored and the UART word length (LCRH.WLEN) must be set to 8 bits.

**Receive Transaction:**

In 9-bit mode, a UNICOMM-UART peripheral checks for the address qualifier at the location of the parity bit. If set, the received byte is compared with the preprogrammed address in ADDR register:

- If the address matches, a 9-bit mode address match interrupt (ADDR_MATCH) is generated, if enabled, and further data is received into the RX FIFO.
- If the address does not match, the address byte and the subsequent data bytes are not received into the RX FIFO.

The address is predefined in the ADDR register to match with the received byte. Matching can be extended to a set of addresses using the address mask in the AMASK register. By default, the AMASK is 0xFF, meaning that only the specified address can generate a match.

**Transmit Transaction:**

All the transmits in 9-bit UART mode are interpreted as:

- Address bytes if the 9th bit is set
- Data bytes if the 9th bit is cleared

In 9-bit mode, the 9th bit can be controlled by software. The value set in the EPS bit of the LCRH register determines the value of the 9th bit for transmit transactions. To indicate an address byte, the application must set the EPS bit before the byte transmission. To indicate a data byte, the EPS bit must be cleared before the byte transmission. For a complete transmit transaction, the address byte must be transmitted as a single byte transaction with EPS bit set, followed by a data byte burst where the EPS bit is cleared.

**Table 28-9. 9th Bit Handling**

| PEN | EPS | $9^{th}$ Bit (Transmitted or Verified) |
|---|---|---|
| 0 | X | Not transmitted or verified |
| 1 | 0 | 0 (= Data) |
| 1 | 1 | 1 (= Address) |

#### 28.2.3.9 ISO7816 Smart Card Support

The UNICOMM-UART Smart Card support feature enables basic communication with an ISO7816 smart card. When configuring the CTL0.MODE field, select the smart card option.

The TXD signal is used as a bit clock, and the RXD signal is used as the half-duplex communication line connected to the smart card. Further smart card signals are not supported by UNICOMM-UART.

The clock rate of the UART clock in ISO7816 mode must be in the range of 1 MHz to 5 MHz when using ISO7816 mode

The LCRH register must be set to:

- 8-bit word length (WLEN bits configured to 0x3)
- Even parity (PEN set and EPS set)
- No stick parity (SPS cleared)

In ISO7816 mode, the UART automatically uses 2 stop bits and 16x oversampling

**Note**



Copyright © 2026 Texas Instruments Incorporated

LCRH.STP2 & CTL0.HSE bits are ignored

- If a parity error is detected during a transmission, the RX line is pulled low during the second stop bit
  - The transmission is aborted and the TX FIFO is flushed
  - Additionally, the module raises a parity error interrupt, allowing the software to detect the problem and initiate retransmission of the affected data, as the UART does not support automatic retransmission in this case
- The UART does not support automatic retransmission on parity errors
  - If a parity error is detected on transmission, all further transmit operations are aborted and software must handle retransmission of the affected byte or message
- In Smartcard Mode, the receiver in case of parity error drives the line low and a parity interrupt flag is asserted.
  - The transmitter responds based on the value of this bit

#### 28.2.3.10 Address Detection

This section applies when a UNICOMM-UART module is configured for Idle-line multiprocessor mode or 9-Bit UART mode in the CTL0.MODE register field.

The ADDR register is used to set the specific address that is matched to the received address byte. This register is used in conjunction with AMASK register to form a match for address-byte received. Only bits where the AMASK is set to '1' are considered. So for full address the AMASK register is set to 0xFF.

**Table 28-10. Address Detection**

| Condition | Idle Line Mode | 9-Bit Mode |
|---|---|---|
| **Address match** | Address and Data is moved to RXDATA | Address and Data is moved to RXDATA |
| **Address mismatch** | Address and Data get dropped | Address and Data get dropped |

#### 28.2.3.11 Receive Timeout

When receiving data, if no start edge has been detected for an additional character period (frame period: calculated as per clock and baud rate settings), a receive timeout interrupt (RTOUT) is set even though the FIFO level is not reached. This provides the application a means of reading out the FIFO contents even if the threshold has not been reached. This trigger is also available for DMA transfers.

**Note**

RTOUT is not supported in IDLE LINE mode

#### 28.2.3.12 LINE Timeout

An additional bit, LTOUT (LINE timeout) is provided in UNICOMMUART. When receiving data, if no start edge has been detected for an additional character period (frame period: calculated as per clock and baud rate settings), a line timeout interrupt is set (LTOUT) irrespective of FIFO/buffer contents.

**Note**

At any given point of time, either RTOUT_EN or LTOUT_EN can be set. Both cannot be used together.

#### 28.2.3.13 Low Power Mode

UNICOMM UART supports a low power mode.

When the module is instantiated in PD0 domain (always on); UART receive is supported in STOP/STANDBY.

1. Baud rate supported depends on the configured clock. *Refer to TRM chapter PMCU section Operating Modes*



Copyright © 2026 Texas Instruments Incorporated

2. The module can also request for a fast clock by asserting an asynchronous clock request upon detecting a START condition. Upon receiving an asynchronous clock request, SYSCTL provides SYSOSC at 32MHz in M0+ devices; Refer to device data sheet for SYSOSC wakeup time. The module can also request for a fast clock by asserting an asynchronous clock request upon detecting a START condition.
   a. Asynchronous clock request can be blocked by assert BLOCK_ASYNC_CLK_REQ; in this case, the module does not request a fast clock and UART data reception continues at the clock available in the power mode
   b. Asynchronous clock is provided to the module as long as async_clk_req signal is asserted; typically, the duration of the UART frame and a few extra cycles for loading into FIFO and for FSM transition
   c. Glitches on the line can cause the UART to detect a START condition and request a fast clock. In this case, when the receive line is sampled again and an idle condition is observed, FSM moves back to idle and fast clock request is de-asserted and sysosc is turned off
3. For M0+ devices, power mode transition from low power mode to ACTIVE is only upon receiving an enabled interrupt

#### 28.2.3.14 Suspend Mode

UNICOMM UART supports a suspend mode.

Suspend is an extreme condition of operation of the IP and application should attempt this mode while taking care of the below conditions. Application can suspend external communication by setting *suspend* control bit; when *suspend* is set, ongoing communication will complete, FSM will return to idle and further external communications are stopped. Transmit line/s will be driven to its idle state and further toggle on receive line/s will not be processed. This provides a safe means for application to suspend operations, clear out receive FIFOs prior to disabling the module



Copyright © 2026 Texas Instruments Incorporated

**Note**

The suspend bit needs to be cleared prior to re-enabling the module

Following the steps below to suspend communication and disable UNICOMM UART module:

1. UART bus masters have to ensure that the peripheral can reach idle state
2. Hardware ensures that ongoing transactions complete; in case of transmit operation, all the contents of the FIFO will be sent out, prior to hardware raising an idle state
3. Once SUSPEND bit is written into hardware, CPU has to poll status flags to ensure that device has reached idle
4. Drain out receive FIFO
5. Disable the module

Post suspend/disable module, register configurations are retained. Communication can be resumed by following the below steps on re-enabling the module:

1. Clear SUSPEND bit
2. Re-enable the module

**Suspend Mode Examples**

1. LIN
   a. Transmit side: ensure that checksum has been loaded into transmit buffer
   b. Receive side: ensure that checksum is received
2. Idleline Multiprocessor
   a. Transmit side: ensure that line all address & data bytes for the current frame has been loaded into the transmit FIFO
   b. Receive side: ensure that checksum is received
3. DALI
   a. Transmit side: ensure that all address & data bytes for the current frame has been loaded into the transmit FIFO
   b. Receive side: ensure that we stop transmission after receiving the backward frame
4. Frame Control RTS/CTS
   a. After transferring data from the transmit fifo, transmitting controller has to pull nRTS line high to indicate that there are no further transmissions
   b. Receiving peripheral should raise its nCTS line high to indicate that it can no longer receive data after receiving the current byte

#### 28.2.3.15 Asynchronous Clock Request

UART can be configured to wake up from low power mode due to incoming traffic on receive line. Application has to choose a baud rate that takes clock wakeup time and frequency while specifying baud rate.

#### 28.2.3.16 Module Pseudo Static Bit Configuration

Application has to setup the configuration registers for desired operation before enabling the module (through ENABLE bits). Configuration bits in Table 7 cannot be updated once ENABLE bit is set. If these bits need to be updated, application would have to disable the module, update the configuration registers and then re-enable the module. When the IP is DISABLED (ensure the sequence SUSPEND → Clear FIFO → DISABLE is followed), it will return the FSM to IDLE, FIFO contents will be cleared and the entire functional clock domain returns to its idle state.

**Pseudo Static Configuration Registers**

| Register Name | Fields |
|---|---|



Copyright © 2026 Texas Instruments Incorporated

| | |
|---|---|
| CTL0 | HSE |
| CTL0 | LBE |
| CTL0 | RXE |
| CTL0 | TXE |
| CTL0 | RTSEN |
| CTL0 | CTSEN |
| CTL0 | MENC |
| CTL0 | MODE |
| CTL0 | TXD_OUT |
| CTL0 | TXD_OUT_EN |
| CTL0 | MAJVOTE |
| CTL0 | MSBFIRST |
| LCRH | PEN |
| LCRH | EPS |
| LCRH | STP2 |
| LCRH | WLEN |
| LCRH | EXTDIR_SETUP |
| LCRH | EXTDIR_HOLD |
| IFLS | TXIFLSEL |
| IFLS | RXIFLSEL |
| IFLS | RXTOSEL |
| GFCTL | DGFSEL |
| LINCNT | VALUE |
| LINCTL | CTRENA |
| LINCTL | ZERONE |
| LINCTL | CNTRXLOW |
| LINCTL | LINC0CAP |
| LINCTL | LINC1CAP |
| LINCTL | LINC0_MATCH |
| IRCTL | IRTXPL |
| IRCTL | IRRXPL |
| IRCTL | IREN |
| IRCTL | IRTXCLK |
| AMASK | MSK |
| ADDR | ADDR |
| CLKDIV | RATIO |
| CLKSEL | MFCLK_SEL |
| CLKSEL | BUSCLK_SEL |
| CLKSEL | LFCLK_SEL |
| CLKSEL | ASYNC_LFCLK_SEL |
| CLKSEL | ASYNC_SYSCLK_SEL |
| CLKSEL | ASYNC_HFCLK_SEL |
| CLKSEL | ASYNC_PLL_SEL |
| CPU_INT_IMASK | FRMERR |
| CPU_INT_IMASK | PARERR |
| CPU_INT_IMASK | BRKERR |
| CPU_INT_IMASK | OVRERR |



Copyright © 2026 Texas Instruments Incorporated

| | |
|---|---|
| CPU_INT_IMASK | RXNE |
| CPU_INT_IMASK | RXPE |
| CPU_INT_IMASK | LINOVF |
| CPU_INT_IMASK | RXINT |
| CPU_INT_IMASK | TXINT |
| CPU_INT_IMASK | EOT |
| CPU_INT_IMASK | ADDR_MATCH |
| CPU_INT_IMASK | CTS |
| CPU_INT_IMASK | DMA_DONE_RX |
| CPU_INT_IMASK | DMA_DONE_TX |
| CPU_INT_IMASK | DMA_PREIRQ_RX |
| CPU_INT_IMASK | DMA_PREIRQ_TX |
| CPU_INT_IMASK | LTOUT |
| CPU_INT_IMASK | RTOUT |
| CPU_INT_IMASK | LINC0 |
| CPU_INT_IMASK | LINC1 |
| CPU_INT_IMASK | NERR |
| DMA_TRIG_RX_IMASK | RTOUT |
| DMA_TRIG_RX_IMASK | RXINT |
| DMA_TRIG_TX_IMASK | TXINT |
| SPGDMA_SPGDMARXSZ | SIZE |
| SPGDMA_SPGDMARXDA | ADDR |
| SPGDMA_SPGDMARXCTL | DMAEN |
| SPGDMA_SPGDMARXCTL | DMAMEMINCR |
| SPGDMA_SPGDMARXCTL | DMAPREIRQ |
| SPGDMA_SPGDMARXCTL | DMATM |
| SPGDMA_SPGDMARXCTL | DMADSTWDTH |
| SPGDMA_SPGDMATXSZ | SIZE |
| SPGDMA_SPGDMATXSA | ADDR |
| SPGDMA_SPGDMATXCTL | DMAEN |
| SPGDMA_SPGDMATXCTL | DMAMEMINCR |
| SPGDMA_SPGDMATXCTL | DMAPREIRQ |
| SPGDMA_SPGDMATXCTL | DMATM |
| SPGDMA_SPGDMATXCTL | DMASRCWDTH |



Copyright © 2026 Texas Instruments Incorporated

Table 29-1 lists the memory-mapped registers for the UNICOMMUART registers. All register offset addresses not listed in Table 29-1 should be considered as reserved locations and the register contents should not be modified.

**Table 29-1. UNICOMMUART Registers**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 1000h | CLKDIV | Clock Divider | | Go |
| 1008h | CLKSEL | Clock Select for Ultra Low Power peripherals | | Go |
| 1020h | IIDX | Interrupt index | | Go |
| 1028h | IMASK | Interrupt mask | | Go |
| 1030h | RIS | Raw interrupt status | | Go |
| 1038h | MIS | Masked interrupt status | | Go |
| 1040h | ISET | Interrupt set | | Go |
| 1048h | ICLR | Interrupt clear | | Go |
| 1058h | IMASK | Interrupt mask | DMA_TRIG_RX | Go |
| 1060h | RIS | Raw interrupt status | DMA_TRIG_RX | Go |
| 1068h | MIS | Masked interrupt status | DMA_TRIG_RX | Go |
| 1070h | ISET | Interrupt set | DMA_TRIG_RX | Go |
| 1088h | IMASK | Interrupt mask | DMA_TRIG_TX | Go |
| 1090h | RIS | Raw interrupt status | DMA_TRIG_TX | Go |
| 1098h | MIS | Masked interrupt status | DMA_TRIG_TX | Go |
| 10A0h | ISET | Interrupt set | DMA_TRIG_TX | Go |
| 10E4h | INTCTL | Interrupt control register | | Go |
| 1100h | CTL0 | UART Control Register 0 | | Go |
| 1104h | LCRH | UART Line Control Register | | Go |
| 1108h | STAT | UART Status Register | | Go |
| 110Ch | IFLS | UART Interrupt FIFO Level Select Register | | Go |
| 1110h | IBRD | UART Integer Baud-Rate Divisor Register | | Go |
| 1114h | FBRD | UART Fractional Baud-Rate Divisor Register | | Go |
| 1118h | GFCTL | Glitch Filter Control | | Go |
| 1120h | TXDATA | UART Transmit Data Register | | Go |



Copyright © 2026 Texas Instruments Incorporated

**Table 29-1. UNICOMMUART Registers (continued)**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 1124h | RXDATA | UART Receive Data Register | Go | |
| 1130h | LINCNT | UART LIN Mode Counter Register | Go | |
| 1134h | LINCTL | UART LIN Mode Control Register | Go | |
| 1138h | LINC0 | UART LIN Mode Capture 0 Register | Go | |
| 113Ch | LINC1 | UART LIN Mode Capture 1 Register | Go | |
| 1140h | IRCTL | eUSCI_Ax IrDA Control Word Register | Go | |
| 1148h | AMASK | Self Address Mask Register | Go | |
| 114Ch | ADDR | Self Address Register | Go | |

Complex bit access types are encoded to fit into small table cells. Table 29-2 shows the codes that are used for access types in this section.

**Table 29-2. UNICOMMUART Access Type Codes**

| Access Type | Code | Description |
|---|---|---|
| Read Type | | |
| R | R | Read |
| Write Type | | |
| W | W | Write |
| Reset or Default Value | | |
| *-n* | | Value after reset or the default value |



Copyright © 2026 Texas Instruments Incorporated

### 29.1 CLKDIV (Offset = 1000h) [Reset = 00000000h]

CLKDIV is shown in Figure 29-1 and described in Table 29-3.

Return to the Summary Table.

This register is used to specify module-specific divide ratio of the functional clock

**Figure 29-1. CLKDIV**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | | RATIO | | |
| R-0h | | | | | | | | | | | | | R/W-0h | | |

**Table 29-3. CLKDIV Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | RESERVED | R | 0h | |
| 2-0 | RATIO | R/W | 0h | Selects divide ratio of module clock Division factor 0 : DIV_BY_1 1 : DIV_BY_2 .... 63: DIV_BY_64<br>0h = Do not divide clock source<br>1h = Divide clock source by 2<br>2h = Divide clock source by 3<br>3h = Divide clock source by 4<br>4h = Divide clock source by 5<br>5h = Divide clock source by 6<br>6h = Divide clock source by 7<br>7h = Divide clock source by 8<br>3Ch = Divide clock source by 61<br>3Dh = Divide clock source by 62<br>3Eh = Divide clock source by 63<br>3Fh = Divide clock source by 64 |



Copyright © 2026 Texas Instruments Incorporated

### 29.2 CLKSEL (Offset = 1008h) [Reset = 00000000h]

CLKSEL is shown in Figure 29-2 and described in Table 29-4.

Return to the Summary Table.

Clock source selection for peripherals

**Figure 29-2. CLKSEL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | ASYNC_PLL_SEL | ASYNC_HFCLK_SEL | ASYNC_SYSCLK_SEL | ASYNC_LFCLK_SEL |
| R-0h | | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | BUSCLK_SEL | MFCLK_SEL | LFCLK_SEL | RESERVED |
| R-0h | | | | R/W-0h | R/W-0h | R/W-0h | R-0h |

**Table 29-4. CLKSEL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-12 | RESERVED | R | 0h | |
| 11 | ASYNC_PLL_SEL | R/W | 0h | Asynchronous PLL selected as source<br>0h = Does not select this clock as a source<br>1h = Select this clock as a source |
| 10 | ASYNC_HFCLK_SEL | R/W | 0h | Asynchronous HFCLK selected as source<br>0h = Does not select this clock as a source<br>1h = Select this clock as a source |
| 9 | ASYNC_SYSCLK_SEL | R/W | 0h | Asynchronous sysclk selected as source<br>0h = Does not select this clock as a source<br>1h = Select this clock as a source |
| 8 | ASYNC_LFCLK_SEL | R/W | 0h | Asynchronous lfclk selected as source<br>0h = Does not select this clock as a source<br>1h = Select this clock as a source |
| 7-4 | RESERVED | R | 0h | |
| 3 | BUSCLK_SEL | R/W | 0h | Selects BUS CLK as clock source if enabled<br>0h = Does not select this clock as a source<br>1h = Select this clock as a source |
| 2 | MFCLK_SEL | R/W | 0h | Selects MFCLK as clock source if enabled<br>0h = Does not select this clock as a source<br>1h = Select this clock as a source |
| 1 | LFCLK_SEL | R/W | 0h | Selects LFCLK as clock source if enabled<br>0h = Does not select this clock as a source<br>1h = Select this clock as a source |
| 0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 29.3 IIDX (Offset = 1020h) [Reset = 00000000h]

IIDX is shown in Figure 29-3 and described in Table 29-5.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. Value 0x00 means no event pending. Interrupt 1 is the highest priority, IIDX next highest, 4, 8, … IIDX^31 is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred. On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in [RIS] and [MIS] are cleared as well. After a read from the CPU (not from the debug interface), the register is updated with the next highest priority interrupt, if none are pending, then it should display 0x0.

**Figure 29-3. IIDX**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | STAT | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | |

**Table 29-5. IIDX Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | STAT | R | 0h | UART Module Interrupt Vector Value. This register provides the highest priority interrupt index. A read clears the corresponding interrupt flag in RIS and MIS registers. 15h-1Fh = Reserved<br>00h = No interrupt pending<br>01h = UART receive time-out interrupt; Interrupt Flag: RT; Interrupt Priority: Highest<br>02h = UART framing error interrupt; Interrupt Flag: FE<br>03h = UART parity error interrupt; Interrupt Flag: PE<br>04h = UART break error interrupt; Interrupt Flag: BE<br>05h = UART receive overrun error interrupt; Interrupt Flag: OE<br>06h = Negative edge on UARTxRXD interrupt; Interrupt Flag: RXNE<br>07h = Positive edge on UARTxRXD interrupt; Interrupt Flag: RXPE<br>08h = LIN capture 0 / match interrupt; Interrupt Flag: LINC0<br>09h = LIN capture 1 interrupt; Interrupt Flag: LINC1<br>0Ah = LIN hardware counter overflow interrupt; Interrupt Flag: LINOVF<br>0Bh = UART receive interrupt; Interrupt Flag: RX<br>0Ch = UART transmit interrupt; Interrupt Flag: TX<br>0Dh = UART end of transmission interrupt (transmit serializer empty); Interrupt Flag: EOT<br>0Eh = 9-bit mode address match interrupt; Interrupt Flag: ADDR_MATCH<br>Fh = UART Clear to Send Modem interrupt; Interrupt Flag: CTS<br>10h = DMA DONE on RX<br>11h = DMA DONE on TX<br>12h = Noise Error Event<br>13h = DMA PRE IRQ INTERRUPT<br>14h = DMA PRE IRQ INTERRUPT<br>15h = UART line time-out interrupt; Interrupt Flag |



Copyright © 2026 Texas Instruments Incorporated

### 29.4 IMASK (Offset = 1028h) [Reset = 00000000h]

IMASK is shown in Figure 29-4 and described in Table 29-6.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

**Figure 29-4. IMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | LTOUT | RESERVED | RESERVED | NERR | DMA_DONE_TX |
| R-0h | | | R/W-0h | R-0h | R-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DMA_DONE_RX | CTS | ADDR_MATCH | EOT | TXINT | RXINT | LINOVF | LINC1 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LINC0 | RXPE | RXNE | OVRERR | BRKERR | PARERR | FRMERR | RTOUT |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

**Table 29-6. IMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-21 | RESERVED | R | 0h | |
| 20 | LTOUT | R/W | 0h | Enable UARTOUT Line Time-Out Interrupt.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 19 | RESERVED | R | 0h | Reserved |
| 18 | RESERVED | R | 0h | Reserved |
| 17 | NERR | R/W | 0h | Noise Error on triple voting. Asserted when the 3 samples of majority voting are not equal<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 16 | DMA_DONE_TX | R/W | 0h | Enable DMA Done on TX Event Channel Interrupt<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 15 | DMA_DONE_RX | R/W | 0h | Enable DMA Done on RX Event Channel Interrupt<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 14 | CTS | R/W | 0h | Enable UART Clear to Send Modem Interrupt.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 13 | ADDR_MATCH | R/W | 0h | Enable Address Match Interrupt.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 12 | EOT | R/W | 0h | Enable UART End of Transmission Interrupt Indicates that the last bit of all transmitted data and flags has left the serializer and without any further Data in the TX FIFO or Buffer.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |



Copyright © 2026 Texas Instruments Incorporated

**Table 29-6. IMASK Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 11 | TXINT | R/W | 0h | Enable UART Transmit Interrupt.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 10 | RXINT | R/W | 0h | Enable UART Receive Interrupt.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 9 | LINOVF | R/W | 0h | Enable LIN Hardware Counter Overflow Interrupt.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 8 | LINC1 | R/W | 0h | Enable LIN Capture 1 Interrupt.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 7 | LINC0 | R/W | 0h | Enable LIN Capture 0 / Match Interrupt .<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 6 | RXPE | R/W | 0h | Enable Positive Edge on UARTxRXD Interrupt.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 5 | RXNE | R/W | 0h | Enable Negative Edge on UARTxRXD Interrupt.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 4 | OVRERR | R/W | 0h | Enable UART Receive Overrun Error Interrupt.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 3 | BRKERR | R/W | 0h | Enable UART Break Error Interrupt.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 2 | PARERR | R/W | 0h | Enable UART Parity Error Interrupt.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 1 | FRMERR | R/W | 0h | Enable UART Framing Error Interrupt.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 0 | RTOUT | R/W | 0h | Enable UARTOUT Receive Time-Out Interrupt.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |



Copyright © 2026 Texas Instruments Incorporated

### 29.5 RIS (Offset = 1030h) [Reset = 00000000h]

RIS is shown in Figure 29-5 and described in Table 29-7.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Figure 29-5. RIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | LTOUT | RESERVED | RESERVED | NERR | DMA_DONE_TX |
| R-0h | | | R-0h | R-0h | R-0h | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DMA_DONE_RX | CTS | ADDR_MATCH | EOT | TXINT | RXINT | LINOVF | LINC1 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LINC0 | RXPE | RXNE | OVRERR | BRKERR | PARERR | FRMERR | RTOUT |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 29-7. RIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-21 | RESERVED | R | 0h | |
| 20 | LTOUT | R | 0h | UARTOUT Line Time-Out Interrupt. Set: no start edge has been detected for an additional character period after reception of last character. This is irrespective of FIFO contents Clear: timeout has not occurred<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 19 | RESERVED | R | 0h | Reserved |
| 18 | RESERVED | R | 0h | Reserved |
| 17 | NERR | R | 0h | Noise Error on triple voting. Asserted when the 3 samples of majority voting are not equal. In DALI mode, this is set when there is no toggle in 2*DALI_HI_THRES<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 16 | DMA_DONE_TX | R | 0h | DMA Done on TX Event Channel Interrupt<br>0h = Interrrupt did not occur<br>1h = Interrupt occurred |
| 15 | DMA_DONE_RX | R | 0h | DMA Done on RX Event Channel Interrupt<br>0h = Interrrupt did not occur<br>1h = Interrupt occurred |
| 14 | CTS | R | 0h | UART Clear to Send Modem Interrupt. This interrupt is raised when CTS toggles, either from '0' to '1' or from '1' to '0'.<br>0h = Interrrupt did not occur<br>1h = Interrupt occurred |
| 13 | ADDR_MATCH | R | 0h | Address Match Interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

**Table 29-7. RIS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 12 | EOT | R | 0h | UART End of Transmission Interrupt Indicates that the last bit of all transmitted data and flags has left the serializer and without any further Data in the TX FIFO or Buffer.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 11 | TXINT | R | 0h | UART Transmit Interrupt based on FIFO threshold settings. In case this variant does not have FIFO, then bit is set when BUFFER is empty<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 10 | RXINT | R | 0h | UART Receive Interrupt based on FIFO threshold settings. In case the variant only has a buffer, then, this bit is set when the BUFFER is full.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 9 | LINOVF | R | 0h | LIN Hardware Counter Overflow Interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 8 | LINC1 | R | 0h | LIN Capture 1 Interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 7 | LINC0 | R | 0h | LIN Capture 0 / Match Interrupt .<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 6 | RXPE | R | 0h | Positive Edge on UARTxRXD Interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 5 | RXNE | R | 0h | Negative Edge on UARTxRXD Interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 4 | OVRERR | R | 0h | UART Receive Overrun Error Interrupt. UART Receive Overrun Error Interrupt is raised when the receive FIFO is flow and a new frame is received<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 3 | BRKERR | R | 0h | UART Break Error Interrupt. Break Error Interrupt is raised when all data bits, parity bit and stop bits are '0'<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 2 | PARERR | R | 0h | UART Parity Error Interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 1 | FRMERR | R | 0h | UART Framing Error Interrupt. Framing error interrupt is raised when a "low" stop bit is detected.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 0 | RTOUT | R | 0h | UARTOUT Receive Time-Out Interrupt. Interrupt is raised when a time-out has lapsed and there are still items in the FIFO<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

### 29.6 MIS (Offset = 1038h) [Reset = 00000000h]

MIS is shown in Figure 29-6 and described in Table 29-8.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 29-6. MIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | LTOUT | RESERVED | RESERVED | NERR | DMA_DONE_TX |
| R-0h | | | R-0h | R-0h | R-0h | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DMA_DONE_RX | CTS | ADDR_MATCH | EOT | TXINT | RXINT | LINOVF | LINC1 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LINC0 | RXPE | RXNE | OVRERR | BRKERR | PARERR | FRMERR | RTOUT |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 29-8. MIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-21 | RESERVED | R | 0h | |
| 20 | LTOUT | R | 0h | Masked UARTOUT Line Time-Out Interrupt.<br>0h = Interrupt did not occur or mask was disabled<br>1h = Interrupt occurred |
| 19 | RESERVED | R | 0h | Reserved |
| 18 | RESERVED | R | 0h | Reserved |
| 17 | NERR | R | 0h | Noise Error on triple voting. Asserted when the 3 samples of majority voting are not equal<br>0h = Interrupt did not occur or mask was disabled<br>1h = Interrupt occurred |
| 16 | DMA_DONE_TX | R | 0h | Masked DMA Done on TX Event Channel Interrupt<br>0h = Interrupt did not occur or mask was disabled<br>1h = Interrupt occurred |
| 15 | DMA_DONE_RX | R | 0h | Masked DMA Done on RX Event Channel Interrupt<br>0h = Interrupt did not occur or mask was disabled<br>1h = Interrupt occurred |
| 14 | CTS | R | 0h | Masked UART Clear to Send Modem Interrupt.<br>0h = Interrupt did not occur or mask was disabled<br>1h = Interrupt occurred |
| 13 | ADDR_MATCH | R | 0h | Masked Address Match Interrupt.<br>0h = Interrupt did not occur or mask was disabled<br>1h = Interrupt occurred |
| 12 | EOT | R | 0h | UART End of Transmission Interrupt Indicates that the last bit of all transmitted data and flags has left the serializer and without any further Data in the TX FIFO or Buffer.<br>0h = Interrupt did not occur or mask was disabled<br>1h = Interrupt occurred |
| 11 | TXINT | R | 0h | Masked UART Transmit Interrupt.<br>0h = Interrupt did not occur or mask was disabled<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

**Table 29-8. MIS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 10 | RXINT | R | 0h | Masked UART Receive Interrupt.<br>0h = Interrupt did not occur or mask was disabled<br>1h = Interrupt occurred |
| 9 | LINOVF | R | 0h | Masked LIN Hardware Counter Overflow Interrupt.<br>0h = Interrupt did not occur or mask was disabled<br>1h = Interrupt occurred |
| 8 | LINC1 | R | 0h | Masked LIN Capture 1 Interrupt.<br>0h = Interrupt did not occur or mask was disabled<br>1h = Interrupt occurred |
| 7 | LINC0 | R | 0h | Masked LIN Capture 0 / Match Interrupt .<br>0h = Interrupt did not occur or mask was disabled<br>1h = Interrupt occurred |
| 6 | RXPE | R | 0h | Masked Positive Edge on UARTxRXD Interrupt.<br>0h = Interrupt did not occur or mask was disabled<br>1h = Interrupt occurred |
| 5 | RXNE | R | 0h | Masked Negative Edge on UARTxRXD Interrupt.<br>0h = Interrupt did not occur or mask was disabled<br>1h = Interrupt occurred |
| 4 | OVRERR | R | 0h | Masked UART Receive Overrun Error Interrupt.<br>0h = Interrupt did not occur or mask was disabled<br>1h = Interrupt occurred |
| 3 | BRKERR | R | 0h | Masked UART Break Error Interrupt.<br>0h = Interrupt did not occur or mask was disabled<br>1h = Interrupt occurred |
| 2 | PARERR | R | 0h | Masked UART Parity Error Interrupt.<br>0h = Interrupt did not occur or mask was disabled<br>1h = Interrupt occurred |
| 1 | FRMERR | R | 0h | Masked UART Framing Error Interrupt.<br>0h = Interrupt did not occur or mask was disabled<br>1h = Interrupt occurred |
| 0 | RTOUT | R | 0h | Masked UARTOUT Receive Time-Out Interrupt.<br>0h = Interrupt did not occur or mask was disabled<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

### 29.7 ISET (Offset = 1040h) [Reset = 00000000h]

ISET is shown in Figure 29-7 and described in Table 29-9.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 29-7. ISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | LTOUT | RESERVED | RESERVED | NERR | DMA_DONE_TX |
| R-0h | | | W-0h | R-0h | R-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DMA_DONE_RX | CTS | ADDR_MATCH | EOT | TXINT | RXINT | LINOVF | LINC1 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LINC0 | RXPE | RXNE | OVRERR | BRKERR | PARERR | FRMERR | RTOUT |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

**Table 29-9. ISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-21 | RESERVED | R | 0h | |
| 20 | LTOUT | W | 0h | Set UARTOUT Line Time-Out Interrupt.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 19 | RESERVED | R | 0h | Reserved |
| 18 | RESERVED | R | 0h | Reserved |
| 17 | NERR | W | 0h | Noise Error on triple voting. Asserted when the 3 samples of majority voting are not equal<br>0h = Writing 0 has no effect<br>1h = Set the interrupt |
| 16 | DMA_DONE_TX | W | 0h | Set DMA Done on TX Event Channel Interrupt<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 15 | DMA_DONE_RX | W | 0h | Set DMA Done on RX Event Channel Interrupt<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 14 | CTS | W | 0h | Set UART Clear to Send Modem Interrupt.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 13 | ADDR_MATCH | W | 0h | Set Address Match Interrupt.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 12 | EOT | W | 0h | Set UART End of Transmission Interrupt Indicates that the last bit of all transmitted data and flags has left the serializer and without any further Data in the TX FIFO or Buffer.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |



Copyright © 2026 Texas Instruments Incorporated

**Table 29-9. ISET Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 11 | TXINT | W | 0h | Set UART Transmit Interrupt.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 10 | RXINT | W | 0h | Set UART Receive Interrupt.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 9 | LINOVF | W | 0h | Set LIN Hardware Counter Overflow Interrupt.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 8 | LINC1 | W | 0h | Set LIN Capture 1 Interrupt.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 7 | LINC0 | W | 0h | Set LIN Capture 0 / Match Interrupt .<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 6 | RXPE | W | 0h | Set Positive Edge on UARTxRXD Interrupt.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 5 | RXNE | W | 0h | Set Negative Edge on UARTxRXD Interrupt.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 4 | OVRERR | W | 0h | Set UART Receive Overrun Error Interrupt.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 3 | BRKERR | W | 0h | Set UART Break Error Interrupt.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 2 | PARERR | W | 0h | Set UART Parity Error Interrupt.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 1 | FRMERR | W | 0h | Set UART Framing Error Interrupt.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 0 | RTOUT | W | 0h | Set UARTOUT Receive Time-Out Interrupt.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 29.8 ICLR (Offset = 1048h) [Reset = 00000000h]

ICLR is shown in Figure 29-8 and described in Table 29-10.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

**Figure 29-8. ICLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | LTOUT | RESERVED | RESERVED | NERR | DMA_DONE_TX |
| R-0h | | | W-0h | R-0h | R-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DMA_DONE_RX | CTS | ADDR_MATCH | EOT | TXINT | RXINT | LINOVF | LINC1 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LINC0 | RXPE | RXNE | OVRERR | BRKERR | PARERR | FRMERR | RTOUT |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

**Table 29-10. ICLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-21 | RESERVED | R | 0h | |
| 20 | LTOUT | W | 0h | Clear UARTOUT Line Time-Out Interrupt.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 19 | RESERVED | R | 0h | Reserved |
| 18 | RESERVED | R | 0h | Reserved |
| 17 | NERR | W | 0h | Noise Error on triple voting. Asserted when the 3 samples of majority voting are not equal<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 16 | DMA_DONE_TX | W | 0h | Clear DMA Done on TX Event Channel Interrupt<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 15 | DMA_DONE_RX | W | 0h | Clear DMA Done on RX Event Channel Interrupt<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 14 | CTS | W | 0h | Clear UART Clear to Send Modem Interrupt.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 13 | ADDR_MATCH | W | 0h | Clear Address Match Interrupt.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 12 | EOT | W | 0h | Clear UART End of Transmission Interrupt Indicates that the last bit of all transmitted data and flags has left the serializer and without any further Data in the TX FIFO or Buffer.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 11 | TXINT | W | 0h | Clear UART Transmit Interrupt.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |



Copyright © 2026 Texas Instruments Incorporated

**Table 29-10. ICLR Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 10 | RXINT | W | 0h | Clear UART Receive Interrupt.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 9 | LINOVF | W | 0h | Clear LIN Hardware Counter Overflow Interrupt.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 8 | LINC1 | W | 0h | Clear LIN Capture 1 Interrupt.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 7 | LINC0 | W | 0h | Clear LIN Capture 0 / Match Interrupt .<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 6 | RXPE | W | 0h | Clear Positive Edge on UARTxRXD Interrupt.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 5 | RXNE | W | 0h | Clear Negative Edge on UARTxRXD Interrupt.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 4 | OVRERR | W | 0h | Clear UART Receive Overrun Error Interrupt.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 3 | BRKERR | W | 0h | Clear UART Break Error Interrupt.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 2 | PARERR | W | 0h | Clear UART Parity Error Interrupt.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 1 | FRMERR | W | 0h | Clear UART Framing Error Interrupt.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 0 | RTOUT | W | 0h | Clear UARTOUT Receive Time-Out Interrupt.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 29.9 IMASK (Offset = 1058h) [Reset = 00000000h]

IMASK is shown in Figure 29-9 and described in Table 29-11.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

**Figure 29-9. IMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | RXINT | RESERVED | |
| R-0h | | | | | R/W-0h | R-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | RTOUT |
| R-0h | | | | | | | R/W-0h |

**Table 29-11. IMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-11 | RESERVED | R | 0h | |
| 10 | RXINT | R/W | 0h | Enable UART Receive Interrupt.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 9-1 | RESERVED | R | 0h | |
| 0 | RTOUT | R/W | 0h | Enable UARTOUT Receive Time-Out Interrupt.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |



Copyright © 2026 Texas Instruments Incorporated

### 29.10 RIS (Offset = 1060h) [Reset = 00000000h]

RIS is shown in Figure 29-10 and described in Table 29-12.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme.

**Figure 29-10. RIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | RXINT | RESERVED | |
| R-0h | | | | | R-0h | R-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | RTOUT |
| R-0h | | | | | | | R-0h |

**Table 29-12. RIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-11 | RESERVED | R | 0h | |
| 10 | RXINT | R | 0h | UART Receive Interrupt based on FIFO threshold settings. In case this variant does not have a FIFO, this bit is set when BUFFER is full<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 9-1 | RESERVED | R | 0h | |
| 0 | RTOUT | R | 0h | UARTOUT Receive Time-Out Interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

### 29.11 MIS (Offset = 1068h) [Reset = 00000000h]

MIS is shown in Figure 29-11 and described in Table 29-13.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 29-11. MIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | RXINT | RESERVED | |
| R-0h | | | | | R-0h | R-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | RTOUT |
| R-0h | | | | | | | R-0h |

**Table 29-13. MIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-11 | RESERVED | R | 0h | |
| 10 | RXINT | R | 0h | Masked UART Receive Interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 9-1 | RESERVED | R | 0h | |
| 0 | RTOUT | R | 0h | Masked UARTOUT Receive Time-Out Interrupt.<br>0h = Interrupt did not occur or mask was not enabled<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

### 29.12 ISET (Offset = 1070h) [Reset = 00000000h]

ISET is shown in Figure 29-12 and described in Table 29-14.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 29-12. ISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | RXINT | RESERVED | |
| R-0h | | | | | W-0h | R-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | RTOUT |
| R-0h | | | | | | | W-0h |

**Table 29-14. ISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-11 | RESERVED | R | 0h | |
| 10 | RXINT | W | 0h | Set UART Receive Interrupt.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 9-1 | RESERVED | R | 0h | |
| 0 | RTOUT | W | 0h | Set UARTOUT Receive Time-Out Interrupt.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 29.13 IMASK (Offset = 1088h) [Reset = 00000000h]

IMASK is shown in Figure 29-13 and described in Table 29-15.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

**Figure 29-13. IMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | TXINT | RESERVED | | |
| R-0h | | | | R/W-0h | R-0h | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |

**Table 29-15. IMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-12 | RESERVED | R | 0h | |
| 11 | TXINT | R/W | 0h | Enable UART Transmit Interrupt.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 10-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 29.14 RIS (Offset = 1090h) [Reset = 00000000h]

RIS is shown in Figure 29-14 and described in Table 29-16.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking.

**Figure 29-14. RIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | TXINT | RESERVED | | |
| R-0h | | | | R-0h | R-0h | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |

**Table 29-16. RIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-12 | RESERVED | R | 0h | |
| 11 | TXINT | R | 0h | UART Transmit Interrupt is based on FIFO threshold settings. In case this variant does not have a FIFO, then, this bit is set when the BUFFER is empty<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 10-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 29.15 MIS (Offset = 1098h) [Reset = 00000000h]

MIS is shown in Figure 29-15 and described in Table 29-17.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 29-15. MIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | TXINT | RESERVED | | |
| R-0h | | | | R-0h | R-0h | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |

**Table 29-17. MIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-12 | RESERVED | R | 0h | |
| 11 | TXINT | R | 0h | Masked UART Transmit Interrupt.<br>0h = Interrupt did not occur or mask was not enabled<br>1h = Interrupt occurred |
| 10-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 29.16 ISET (Offset = 10A0h) [Reset = 00000000h]

ISET is shown in Figure 29-16 and described in Table 29-18.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 29-16. ISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | TXINT | RESERVED | | |
| R-0h | | | | W-0h | R-0h | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |

**Table 29-18. ISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-12 | RESERVED | R | 0h | |
| 11 | TXINT | W | 0h | Set UART Transmit Interrupt.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 10-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 29.17 INTCTL (Offset = 10E4h) [Reset = 00000000h]

INTCTL is shown in Figure 29-17 and described in Table 29-19.

Return to the Summary Table.

Interrupt control register

**Figure 29-17. INTCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | INTEVAL |
| R-0h | | | | | | | W-0h |

**Table 29-19. INTCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | INTEVAL | W | 0h | Writing a 1 to this field re-evaluates the interrupt sources.<br>0h = Writing 0 has no effect<br>1h = Interrupt Eval |



Copyright © 2026 Texas Instruments Incorporated

### 29.18 CTL0 (Offset = 1100h) [Reset = 00000038h]

CTL0 is shown in Figure 29-18 and described in Table 29-20.

Return to the Summary Table.

UART Control Register The CTL0 register is the control register. All the bits are cleared on reset except for the Transmit Enable (TXE) and Receive Enable (RXE) bits, which are set. To enable the UART module, the UARTEN bit must be set. If software requires a configuration change in the module, the UARTEN bit must be cleared before the configuration changes are written. If the UART is disabled during a transmit or receive operation, the current transaction is completed prior to the UART stopping. NOTE: The CTL0 register should not be changed while the UART is enabled or else the results are unpredictable. The following sequence is recommended for making changes to the CTL0 register. 1. Disable the UART. 2. Wait for the end of transmission or reception of the current character. 3. Clear transmit FIFO 4. Reprogram the control register. 5. Enable the UART.

**Figure 29-18. CTL0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | MSBFIRST | MAJVOTE | RESERVED | HSE |
| R-0h | | | | R/W-0h | R/W-0h | R-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| HSE | CTSEN | RTSEN | RTS | RESERVED | MODE | | |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | R/W-0h | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MENC | TXD_OUT | TXD_OUT_EN | TXE | RXE | LBE | RESERVED | ENABLE |
| R/W-0h | R/W-0h | R/W-1h | R/W-1h | R/W-1h | R/W-0h | R-0h | R/W-0h |

**Table 29-20. CTL0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | RESERVED | R | 0h | |
| 19 | MSBFIRST | R/W | 0h | Most Significant Bit First This bit has effect both on the way protocol byte is transmitted and received. Notes: User needs to match the protocol to the correct value of this bit to send MSb or LSb first. The hardware engine will send the byte entirely based on this bit.<br>0h = Least significant bit is sent first in the protocol packet<br>1h = Most significant bit is sent first in the protocol packet |
| 18 | MAJVOTE | R/W | 0h | Majority Vote Enable When Majority Voting is enabled, the three center bits are used to determine received sample value. In case of error (i.e. all 3 bits are not the same), noise error is detected and bits RIS.NERR and register RXDATA.NERR are set. Oversampling of 16 : bits 7, 8, 9 are used Oversampling of 8 : bits 3, 4, 5 are used Disabled : Single sample value (center value) used<br>0h = Majority voting is disabled<br>1h = Majority voting is enabled |
| 17 | RESERVED | R | 0h | |
| 16-15 | HSE | R/W | 0h | High-Speed Bit Oversampling Enable<br>0h = 16x oversampling.<br>1h = 8x oversampling.<br>2h = 3x oversampling. IrDA, Manchester, Smartcard and DALI not supported when 3x oversampling is enabled. |



Copyright © 2026 Texas Instruments Incorporated

**Table 29-20. CTL0 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 14 | CTSEN | R/W | 0h | Enable Clear To Send<br>0h = CTS hardware flow control is disabled.<br>1h = CTS hardware flow control is enabled. Data is only transmitted when the UARTxCTS signal is asserted. |
| 13 | RTSEN | R/W | 0h | Enable hardware controlled Request to Send<br>0h = RTS hardware flow control is disabled.<br>1h = RTS hardware flow control is enabled. Data is only requested (by asserting UARTxRTS) when the receive FIFO/buffer has available entries. |
| 12 | RTS | R/W | 0h | Request to Send If RTSEN is set the RTS output signals is controlled by the hardware logic using the FIFO fill level or buffer status. If RTSEN is cleared the RTS output is controlled by the RTS bit. The bit is the complement of the UART request to send, RTS modem status output.<br>0h = Signal not RTS<br>1h = Signal RTS |
| 11 | RESERVED | R | 0h | |
| 10-8 | MODE | R/W | 0h | Set the communication mode and protocol used. (Not defined settings uses the default setting: 0)<br>0h = Normal operation<br>1h = RS485 mode: UART needs to be IDLE with receiving data for the in EXTDIR_HOLD set time. EXTDIR_SETUP defines the time the RTS line is set to high before sending. When the buffer is empty the RTS line is set low again. A transmit will be delayed as long the UART is receiving data.<br>2h = The UART operates in IDLE Line Mode<br>3h = The UART operates in 9 Bit Address mode<br>4h = ISO7816 Smart Card Support The application must ensure that it sets 8-bit word length (WLEN set to 3h) and even parity (PEN set to 1, EPS set to 1, SPS set to 0) in UARTLCRH when using ISO7816 mode. The value of the STP2 bit in UARTLCRH is ignored and the number of stop bits is forced to 2.<br>5h = DALI Mode: |
| 7 | MENC | R/W | 0h | Manchester Encode enable<br>0h = Disable Manchester Encoding<br>1h = Enable Manchester Encoding |
| 6 | TXD_OUT | R/W | 0h | TXD Pin Control Controls the TXD pin when TXD_OUT_EN = 1 and TXE = 0.<br>0h = TXD pin is low<br>1h = TXD pin is high |
| 5 | TXD_OUT_EN | R/W | 1h | TXD Pin Control Enable. When the transmit section of the UART is disabled (TXE = 0), the TXD pin can be controlled by the TXD_OUT bit.<br>0h = TXD pin can not be controlled by TXD_OUT<br>1h = TXD pin can be controlled by TXD_OUT |
| 4 | TXE | R/W | 1h | UART Transmit Enable If the UART is disabled in the middle of a transmission, it completes the current character before stopping. **NOTE:** To enable transmission, the UARTEN bit must be set.<br>0h = The transmit section of the UART is disabled. The UARTxTXD pin of the UART can be controlled by the TXD_CTL bit when enabled.<br>1h = The transmit section of the UART is enabled. |
| 3 | RXE | R/W | 1h | UART Receive Enable If the UART is disabled in the middle of a receive, it completes the current character before stopping. **NOTE:** To enable reception, the UARTEN bit must be set.<br>0h = The receive section of the UART is disabled.<br>1h = The receive section of the UART is enabled. |
| 2 | LBE | R/W | 0h | UART Loop Back Enable<br>0h = Normal operation.<br>1h = The UARTxTX path is fed through the UARTxRX path internally. |



Copyright © 2026 Texas Instruments Incorporated

**Table 29-20. CTL0 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 1 | RESERVED | R | 0h | |
| 0 | ENABLE | R/W | 0h | UART Module Enable. If the UART is disabled in the middle of transmission or reception, it completes the current character before stopping. If the ENABLE bit is not set, all registers can still be accessed and updated. It is recommended to setup and change the UART operation mode with having the ENABLE bit cleared to avoid unpredictable behavior during the setup or update. If disabled the UART module will not send or receive any data and the logic is held in reset state.<br>0h = Disable Module<br>1h = Enable module |



Copyright © 2026 Texas Instruments Incorporated

### 29.19 LCRH (Offset = 1104h) [Reset = 00000000h]

LCRH is shown in Figure 29-19 and described in Table 29-21.

Return to the Summary Table.

UART Line Control Register The LCRH register is the line control register. Serial parameters such as data length, parity, and stop bit selection are implemented in this register.

**Figure 29-19. LCRH**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | SUSPEND | EXTDIR_HOLD | |
| R-0h | | | | | R/W-0h | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| EXTDIR_HOLD | | | EXTDIR_SETUP | | | | |
| R/W-0h | | | R/W-0h | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SENDIDLE | SPS | WLEN | | STP2 | EPS | PEN | BRK |
| R/W-0h | R/W-0h | R/W-0h | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

**Table 29-21. LCRH Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-27 | RESERVED | R | 0h | |
| 26 | SUSPEND | R/W | 0h | Suspend external communication<br>0h = Functional mode resumed<br>1h = External communication suspended |
| 25-21 | EXTDIR_HOLD | R/W | 0h | Defines the number of UARTclk ticks the signal to control the external driver for the RS485 will be reset after the beginning of the stop bit. (If 2 STOP bits are enabled the beginning of the 2nd STOP bit.)<br>0h = Smallest value<br>1Fh = Highest possible value |
| 20-16 | EXTDIR_SETUP | R/W | 0h | Defines the number of UARTclk ticks the signal to control the external driver for the RS485 will be set before the START bit is send<br>0h = Smallest value<br>1Fh = Highest possible value |
| 15-8 | RESERVED | R | 0h | |
| 7 | SENDIDLE | R/W | 0h | UART send IDLE pattern. When this bit is set, SENDIDLE period of 11 bit times will be sent on the TX line. Read STAT.SENDIDLE bit to readback current status of SENDIDLE<br>0h = Normal operation<br>1h = Send Idle Pattern |
| 6 | SPS | R/W | 0h | UART Stick Parity Select The Stick Parity Select (SPS) bit is used to set either a permanent '1' or a permanent '0' as parity when transmitting or receiving data. Its purpose is to typically indicate the first byte of a package or to mark an address byte, for example in a multi-drop RS-485 network. When bits PEN, EPS, and SPS of UARTLCRH are set, the parity bit is transmitted and checked as a 0. When bits PEN and SPS are set and EPS is cleared, the parity bit is transmitted and checked as a 1.<br>0h = Disable Stick Parity<br>1h = Enable Stick Parity |



Copyright © 2026 Texas Instruments Incorporated

**Table 29-21. LCRH Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 5-4 | WLEN | R/W | 0h | UART Word Length The bits indicate the number of data bits transmitted or received in a frame as follows:<br>0h = 5 bits (default)<br>1h = 6 bits<br>2h = 7 bits<br>3h = 8 bits |
| 3 | STP2 | R/W | 0h | UART Two Stop Bits Select When in 7816 smart card mode (the SMART bit is set in the UARTCTL register), the number of stop bits is forced to 2.<br>0h = One stop bit is transmitted at the end of a frame.<br>1h = Two stop bits are transmitted at the end of a frame. The receive logic checks for two stop bits being received and provide Frame Error if either is invalid. |
| 2 | EPS | R/W | 0h | UART Even Parity Select This bit has no effect when parity is disabled by the PEN bit. For 9-Bit UART Mode transmissions, this bit controls the address byte and data byte indication (9th bit). 0 = The transferred byte is a data byte 1 = The transferred byte is an address byte<br>0h = Odd parity is performed, which checks for an odd number of 1s.<br>1h = Even parity generation and checking is performed during transmission and reception, which checks for an even number of 1s in data and parity bits. |
| 1 | PEN | R/W | 0h | UART Parity Enable<br>0h = Parity is disabled and no parity bit is added to the data frame.<br>1h = Parity checking and generation is enabled. |
| 0 | BRK | R/W | 0h | UART Send Break (for LIN Protocol) 1. Break condition is sent on the line for as long as this bit is set<br>0h = Normal operation<br>1h = A low level is continually output on the UARTxTXD signal, after completing transmission of the current character. For the proper execution of the break command, software must set this bit for at least two frames (character periods). |



Copyright © 2026 Texas Instruments Incorporated

### 29.20 STAT (Offset = 1108h) [Reset = 00000000h]

STAT is shown in Figure 29-20 and described in Table 29-22.

Return to the Summary Table.

UART Status Register

**Figure 29-20. STAT**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | SENDIDLE | IDLE | CTS |
| R-0h | | | | | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TXCLR | TXFF | TXFE | RXCLR | RXFF | RXFE | RESERVED | BUSY |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 29-22. STAT Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-11 | RESERVED | R | 0h | |
| 10 | SENDIDLE | R | 0h | TX FIFO Clear Status<br>0h = IDLE condition has not yet been sent on the line<br>1h = IDLE condition has been sent on the line |
| 9 | IDLE | R | 0h | IDLE mode has been detected in Idleline-Multiprocessor-Mode. The IDLE bit is used as an address tag for each block of characters. In idle-line multiprocessor format, this bit is set when a received character is an address.<br>0h = IDLE has not been detected before last received character. (In idle-line multiprocessor mode).<br>1h = IDLE has been detected before last received character. (In idle-line multiprocessor mode). |
| 8 | CTS | R | 1h | Clear To Send<br>0h = The CTS signal is not asserted (high).<br>1h = The CTS signal is asserted (low). |
| 7 | TXCLR | R | 0h | TX FIFO Clear Status<br>0h = FIFO is not cleared<br>1h = FIFO clear is complete |
| 6 | TXFF | R | 0h | UART Transmit FIFO Full The meaning of this bit depends on the state of the FEN bit in the CTL0 register.<br>0h = The transmitter is not full.<br>1h = transmit FIFO is full. |
| 5 | TXFE | R | 1h | UART Transmit FIFO Empty<br>0h = The transmitter has data to transmit.<br>1h = transmit FIFO is empty. |
| 4 | RXCLR | R | 0h | RX FIFO Clear Status<br>0h = FIFO is not cleared<br>1h = FIFO clear is complete |
| 3 | RXFF | R | 0h | UART Receive FIFO Full<br>0h = The receiver can receive data.<br>1h = receive FIFO is full. |



Copyright © 2026 Texas Instruments Incorporated

**Table 29-22. STAT Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | RXFE | R | 1h | UART Receive FIFO Empty<br>0h = The receiver is not empty.<br>1h = Receive FIFO is empty. |
| 1 | RESERVED | R | 0h | |
| 0 | BUSY | R | 0h | UART Busy This bit is set as soon as the transmit FIFO/buffer becomes non-empty or if a receive data is currently ongoing (after the start edge have been detected until a complete byte, including all stop bits, has been received by the shift register). In IDLE_Line mode the Busy signal also stays set during the idle time generation. This bit is set only when the module in enabled.<br>0h = The UART is not busy.<br>1h = The UART is busy transmitting data. This bit remains set until the complete byte, including all stop bits, has been sent/received from/into the shift register. |



Copyright © 2026 Texas Instruments Incorporated

### 29.21 IFLS (Offset = 110Ch) [Reset = 00000022h]

IFLS is shown in Figure 29-21 and described in Table 29-23.

Return to the Summary Table.

The IFLS register is the interrupt FIFO level select register. You can use this register to define the levels at which the TX, RX and timeout interrupt flags are triggered. The interrupts are generated based on a transition through a level rather than being based on the level. That is, the interrupts are generated when the fill level progresses through the trigger level. For example, if the receive trigger level is set to the half-way mark, the interrupt is triggered when the receive FIFO is filled with two or more characters. Out of reset, the TXIFLSEL and RXIFLSEL bits are configured so that the FIFOs trigger an interrupt at the half-way mark.

**Figure 29-21. IFLS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | RXTOSEL | | | |
| R-0h | | | | R/W-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RXCLR | RXIFLSEL | | | TXCLR | TXIFLSEL | | |
| R/W-0h | R/W-2h | | | R/W-0h | R/W-2h | | |

**Table 29-23. IFLS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-12 | RESERVED | R | 0h | |
| 11-8 | RXTOSEL | R/W | 0h | UART Receive Interrupt Timeout Select. When receiving no start edge for an additional character within the set bit times a RX interrupt is set even if the FIFO level is not reached. A value of 0 disables this function.<br>0h = Smallest value<br>Fh = Highest possible value |
| 7 | RXCLR | R/W | 0h | RX FIFO CLEAR. Setting this bit will clear the RX FIFO contents.<br>0h = Disable FIFO clear<br>1h = Enable FIFO Clear |
| 6-4 | RXIFLSEL | R/W | 2h | FIFO Level Select for generating events (interrupt/dma). Note: for undefined settings the default configuration is used.<br>1h = RX FIFO >= 1/4 full<br>2h = RX FIFO >= 1/2 full (default)<br>3h = RX FIFO >= 3/4 full<br>4h = Opposite of empty<br>5h = RX FIFO is full<br>6h = RX_FIFO >= (MAX_FIFO_LEN -1)<br>7h = RX_FIFO <= 1 |
| 3 | TXCLR | R/W | 0h | TX FIFO CLEAR. Setting this bit will clear the TX FIFO contents.<br>0h = Disable FIFO clear<br>1h = Enable FIFO Clear |



Copyright © 2026 Texas Instruments Incorporated

**Table 29-23. IFLS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2-0 | TXIFLSEL | R/W | 2h | FIFO Level Select for generating events (interrupt/dma). Note: for undefined settings the default configuration is used.<br>1h = TX FIFO <= 3/4 empty<br>2h = TX FIFO <= 1/2 empty (default)<br>3h = TX FIFO <= 1/4 empty<br>4h = Opposite of full<br>5h = TX FIFO is empty<br>6h = TX FIFO <= 1<br>7h = TX_FIFO >= (MAX_FIFO_LEN -1) |



Copyright © 2026 Texas Instruments Incorporated

### 29.22 IBRD (Offset = 1110h) [Reset = 00000000h]

IBRD is shown in Figure 29-22 and described in Table 29-24.

Return to the Summary Table.

When changing the IBRD register, the new value does not take effect until transmission/reception of the current character is complete. Any changes to the baud-rate divisor must be followed by a write to the UARTLCRH register. See Baud-Rate Generation chapter for configuration details.

**Figure 29-22. IBRD**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | DIVINT | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | R/W-0h | | | | | | | | | | | | | | | |

**Table 29-24. IBRD Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15-0 | DIVINT | R/W | 0h | Integer Baud-Rate Divisor<br>0h = Smallest value<br>FFFFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 29.23 FBRD (Offset = 1114h) [Reset = 00000000h]

FBRD is shown in Figure 29-23 and described in Table 29-25.

Return to the Summary Table.

UART Fractional Baud-Rate Divisor Register The FBRD register is the fractional part of the baud-rate divisor value. All the bits are cleared on reset. When changing the FBRD register, the new value does not take effect until transmission/reception of the current character is complete. Any changes to the baud-rate divisor must be followed by a write to the UARTLCRH register. See Baud-Rate Generation chapter for configuration details.

**Figure 29-23. FBRD**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | DIVFRAC | | | | | |
| R-0h | | | | | | | | | | R/W-0h | | | | | |

**Table 29-25. FBRD Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | RESERVED | R | 0h | |
| 5-0 | DIVFRAC | R/W | 0h | Fractional Baud-Rate Divisor<br>0h = Smallest value<br>3Fh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 29.24 GFCTL (Offset = 1118h) [Reset = 0000000Xh]

GFCTL is shown in Figure 29-24 and described in Table 29-26.

Return to the Summary Table.

This register control the glitch filter on the RX input.

**Figure 29-24. GFCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | DGFSEL | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | R/W-00Bh | | | | | |

**Table 29-26. GFCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | RESERVED | R | 0h | |
| 5-0 | DGFSEL | R/W | 00Bh | Glitch Suppression Pulse Width This field controls the pulse width select for glitch suppression on the RX line. The value programmed in this field gives the number of cycles of functional clock up to which the glitch has to be suppressed on the RX line.<br>0h = Bypass GF<br>3Fh = Highest Possible Value |



Copyright © 2026 Texas Instruments Incorporated

### 29.25 TXDATA (Offset = 1120h) [Reset = 00000000h]

TXDATA is shown in Figure 29-25 and described in Table 29-27.

Return to the Summary Table.

UART Transmit Data Register. This register is the transmit data register (the interface to the FIFOs). For transmitted data, if the FIFO is enabled, data written to this location is pushed onto the transmit FIFO. If the FIFO is disabled, data is stored in the transmitter holding register (the bottom word of the transmit FIFO). A write to this register initiates a transmission from the UART.

**Figure 29-25. TXDATA**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | DATA | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | W-0h | | | | | | | |

**Table 29-27. TXDATA Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | DATA | W | 0h | Data Transmitted or Received Data that is to be transmitted via the UART is written to this field. When read, this field contains the data that was received by the UART.<br>0h = Smallest value<br>FFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 29.26 RXDATA (Offset = 1124h) [Reset = 00000000h]

RXDATA is shown in Figure 29-26 and described in Table 29-28.

Return to the Summary Table.

UART Receive Data Register. This register is the data receive register (the interface to the FIFOs). For received data, if the FIFO is enabled, the data byte and the 4-bit status (break, frame, parity, and overrun) is pushed onto the 12-bit wide receive FIFO. If the FIFO is disabled, the data byte and status are stored in the receiving holding register (the bottom word of the receive FIFO). The received data can be retrieved by reading this register.

**Figure 29-26. RXDATA**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | NERR | BRKERR | PARERR | FRMERR |
| R-0h | | | | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DATA | | | | | | | |
| R-0h | | | | | | | |

**Table 29-28. RXDATA Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-12 | RESERVED | R | 0h | |
| 11 | NERR | R | 0h | Noise Error. Writing to this bit has no effect. The flag is cleared by writing 1 to the NERR bit in the UART EVENT ICLR register.<br>0h = No noise error occurred<br>1h = Noise error occurred during majority voting |
| 10 | BRKERR | R | 0h | UART Break Error Writing to this bit has no effect. The flag is cleared by writing 1 to the BRKERR bit in the UART EVENT ICLR register. This error is associated with the character at the top of the FIFO. When a break occurs, only one 0 character is loaded into the FIFO. The next character is only enabled after the receive data input goes to a 1 (marking state) and the next valid start bit is received.<br>0h = No break condition has occurred<br>1h = A break condition has been detected, indicating that the receive data input was held low for longer than a full-word transmission time (defined as start, data, parity, and stop bits). |
| 9 | PARERR | R | 0h | UART Parity Error Writing to this bit has no effect. The flag is cleared by writing 1 to the PARERR bit in the UART EVENT ICLR register.<br>0h = No parity error has occurred<br>1h = The parity of the received data character does not match the parity defined by bits 2 and 7 of the UARTLCRH register. |
| 8 | FRMERR | R | 0h | UART Framing Error Writing to this bit has no effect. The flag is cleared by writing 1 to the FRMERR bit in the UART EVENT ICLR register. This error is associated with the character at the top of the FIFO.<br>0h = No framing error has occurred<br>1h = The received character does not have a valid stop bit sequence, which is one or two stop bits depending on the UARTLCRH.STP2 setting (a valid stop bit is 1). |



Copyright © 2026 Texas Instruments Incorporated

**Table 29-28. RXDATA Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7-0 | DATA | R | 0h | Received Data. When read, this field contains the data that was received by the UART.<br>0h = Smallest value<br>FFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 29.27 LINCNT (Offset = 1130h) [Reset = 00000000h]

LINCNT is shown in Figure 29-27 and described in Table 29-29.

Return to the Summary Table.

UART LIN Mode Counter Register

**Figure 29-27. LINCNT**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | VALUE | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | R/W-0h | | | | | | | | | | | | | | | |

**Table 29-29. LINCNT Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15-0 | VALUE | R/W | 0h | 16 bit up counter clocked by the functional clock of the UART.<br>0h = Smallest value<br>FFFFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 29.28 LINCTL (Offset = 1134h) [Reset = 00000000h]

LINCTL is shown in Figure 29-28 and described in Table 29-30.

Return to the Summary Table.

UART LIN Mode Control Register

**Figure 29-28. LINCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| EN_FRM_ERR | LINC0_MATCH | LINC1CAP | LINC0CAP | RESERVED | CNTRXLOW | ZERONE | CTRENA |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | R/W-0h | R/W-0h | R/W-0h |

**Table 29-30. LINCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7 | EN_FRM_ERR | R/W | 0h | Enable FRAME ERROR<br>0h = Disable Frame Error Generation If STOP bit of '0' is received, then, FRAME ERROR is not generated<br>1h = Enable Frame Error Generation If STOP bit of '0' is received, then, FRAME ERROR is generated |
| 6 | LINC0_MATCH | R/W | 0h | Counter Compare Match Mode When this bit is set to 1 a counter compare match with LINC0 register triggers an LINC0 interrupt when enabled.<br>0h = Counter compare match mode disabled (capture mode enabled)<br>1h = Counter compare match enabled (capture mode disabled) |
| 5 | LINC1CAP | R/W | 0h | Capture Counter on positive RXD Edge. When enabled the counter value is captured to LINC1 register on each positive RXD edge. A LINC1 interrupt is triggered when enabled.<br>0h = Capture counter on positive UARTxRXD edge disabled<br>1h = Capture counter on positive UARTxRXD edge enabled |
| 4 | LINC0CAP | R/W | 0h | Capture Counter on negative RXD Edge. When enabled the counter value is captured to LINC0 register on each negative RXD edge. A LINC0 interrupt is triggered when enabled.<br>0h = Capture counter on negative UARTxRXD edge disabled<br>1h = Capture counter on negative UARTxRXD edge enabled |
| 3 | RESERVED | R | 0h | |
| 2 | CNTRXLOW | R/W | 0h | Count while low Signal on RXD When counter is enabled and the signal on RXD is low, the counter increments.<br>0h = Count while low Signal on UARTxRXD disabled<br>1h = Count while low Signal on UARTxRXD enabled |
| 1 | ZERONE | R/W | 0h | Zero on negative Edge of RXD. When enabled the counter is set to 0 and starts counting on a negative edge of RXD<br>0h = Zero on negative edge disabled<br>1h = Zero on negative edge enabled |



Copyright © 2026 Texas Instruments Incorporated

**Table 29-30. LINCTL Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 0 | CTRENA | R/W | 0h | LIN Counter Enable. LIN counter will only count when enabled.<br>0h = Counter disabled<br>1h = Counter enabled |



Copyright © 2026 Texas Instruments Incorporated

### 29.29 LINC0 (Offset = 1138h) [Reset = 00000000h]

LINC0 is shown in Figure 29-29 and described in Table 29-31.

Return to the Summary Table.

UART LIN Mode Capture 0 Register

**Figure 29-29. LINC0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | DATA | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | R/W-0h | | | | | | | | | | | | | | | |

**Table 29-31. LINC0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15-0 | DATA | R/W | 0h | 16 Bit Capture / Compare Register Captures current LINCTR value on RXD falling edge and can generate a LINC0 interrupt when capture is enabled (LINC0CAP = 1). If compare mode is enabled (LINC0_MATCH = 1), a counter match can generate a LINC0 interrupt.<br>0h = Smallest value<br>FFFFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 29.30 LINC1 (Offset = 113Ch) [Reset = 00000000h]

LINC1 is shown in Figure 29-30 and described in Table 29-32.

Return to the Summary Table.

UART LIN Mode Capture 1 Register

**Figure 29-30. LINC1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | DATA | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | R/W-0h | | | | | | | | | | | | | | | |

**Table 29-32. LINC1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15-0 | DATA | R/W | 0h | 16 Bit Capture / Compare Register Captures current LINCTR value on RXD rising edge and can generate a LINC1 interrupt when capture is enabled (LINC1CAP = 1)<br>0h = Smallest value<br>FFFFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 29.31 IRCTL (Offset = 1140h) [Reset = 00000000h]

IRCTL is shown in Figure 29-31 and described in Table 29-33.

Return to the Summary Table.

IrDA Control Register

**Figure 29-31. IRCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | IRRXPL | RESERVED |
| R-0h | | | | | | R/W-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| IRTXPL | | | | | | IRTXCLK | IREN |
| R/W-0h | | | | | | R/W-0h | R/W-0h |

**Table 29-33. IRCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-10 | RESERVED | R | 0h | |
| 9 | IRRXPL | R/W | 0h | IrDA receive input UCAxRXD polarity<br>0h = HIGH : IrDA transceiver delivers a high pulse when a light pulse is seen<br>1h = LOW : IrDA transceiver delivers a low pulse when a light pulse is seen |
| 8 | RESERVED | R | 0h | |
| 7-2 | IRTXPL | R/W | 0h | Transmit pulse length. Pulse length t(PULSE) = (IRTXPLx + 1) / [2 * f(IRTXCLK)] (IRTXCLK = functional clock of the UART)<br>0h = Smallest value<br>3Fh = Highest possible value |
| 1 | IRTXCLK | R/W | 0h | IrDA transmit pulse clock select<br>0h (R/W) = IrDA encode data is based on the functional clock.<br>1h (R/W) = IrDA encode data is based on the Baud Rate clock< when select 8x oversampling, the IRTXPL cycle should less 8; when select 16x oversampling, the IRTXPL cycle should less 16. |
| 0 | IREN | R/W | 0h | IrDA encoder/decoder enable<br>0h (R/W) = IrDA encoder/decoder disabled<br>1h (R/W) = IrDA encoder/decoder enabled |



Copyright © 2026 Texas Instruments Incorporated

### 29.32 AMASK (Offset = 1148h) [Reset = 00000000h]

AMASK is shown in Figure 29-32 and described in Table 29-34.

Return to the Summary Table.

Self Address Mask Register The AMASK register is used to enable the address mask for 9-bit or Idle-Line mode. The address bits are masked to create a set of addresses to be matched with the received address byte. Used in DALI, UART 9-Bit or Idle-Line mode.

**Figure 29-32. AMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | VALUE | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R/W-0h | | | | | | | |

**Table 29-34. AMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | VALUE | R/W | 0h | Self Address Mask for 9-Bit Mode This field contains the address mask that creates a set of addresses that should be matched. A 0 bit in the MSK bitfield configures, that the corresponding bit in the ADDR bitfield of the UARTxADDR register is don't care. A 1 bit in the MSK bitfield configures, that the corresponding bit in the ADDR bitfield of the UARTxADDR register must match.<br>0h = Smallest value<br>FFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 29.33 ADDR (Offset = 114Ch) [Reset = 00000000h]

ADDR is shown in Figure 29-33 and described in Table 29-35.

Return to the Summary Table.

Self Address Register The ADDR register is used to write the specific address that should be matched with the receiving byte when the Address Mask (AMASK) is set to FFh. This register is used in conjunction with AMASK to form a match for address-byte received. Used in DALI, UART 9-Bit or Idle-Line mode.

**Figure 29-33. ADDR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | VALUE | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R/W-0h | | | | | | | |

**Table 29-35. ADDR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | VALUE | R/W | 0h | Self Address for 9-Bit Mode This field contains the address that should be matched when UARTxAMASK is FFh.<br>0h = Smallest value<br>FFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

Chapter 30

# UNICOMM-I2C


**30.1 UNICOMM-I2C Overview** ..... 1571
**30.2 UNICOMM Common Infrastructure** ..... 1574
**30.3 Peripheral Functional Description** ..... 1574
**30.4 UNICOMM I2C Registers** ..... 1608





Copyright © 2026 Texas Instruments Incorporated

## 30.1 UNICOMM-I2C Overview

The UNICOMM-I2CC (Controller) and UNICOMM-I2CT (Target) modules provides a standardized serial interface to transfer data between MSP devices and other external $I^2C$ devices (such as a sensors, memory, or DACs).

This section describes the functionality of a UNICOMM module when configured to operate as an I2C (either I2C controller or I2C target). The protocol mode of a given UNICOMM instance is determined by the programmed SELECT field of the IPMODE register. If a UNICOMM instance is configured for another peripheral mode, the I2CT/I2CC functionality on that UNICOMM instance is disabled and all register reads return a value of zero.

### *30.1.1 Purpose of the Peripheral*

The UNICOMM-I2CC and UNICOMM-I2CT peripherals provides bidirectional data transfer through a two-wire serial bus consisting of a data (SDA) and clock (SCL) line. The $I^2C$ bus is widely used to interface with devices such as battery management ICs, sensors, other MCUs and so on. This $I^2C$ peripheral provides the ability to both transmit to and receive from other $I^2C$ devices on the bus.

### *30.1.2 Features*

The UNICOMM-I2CC and UNICOMM-I2CT support the following features:

- 7-bit and 10-bit addressing
- Supported transmission speeds:
  - Standard-mode (Sm) with a bit rate up to 100 kbps
  - Fast-mode (Fm) with a bit rate up to 400 kbps
  - Fast-mode Plus (Fm+) with a bit rate up to 1 Mbps
- UNICOMM-I2CC supports operation with arbitration, clock synchronization, multiple controller support
- Single buffer or 4-entry FIFO deepth for transmission and reception
- Glitch suppression
- Low power mode support
- Independent controller and target interrupt generation
- Hardware support for DMA with separate channels for transmit and receive

To support I2C protocol, a UNICOMM module must be configured as either UNICOMM-I2CC (Controller) or UNICOMM-I2CT (Target) at the UNICOMM top level. Additionally, each UNICOMM-I2CC and UNICOMM-I2CT may support the additional features listed in the tables below.

**Table 30-1. UNICOMM I2CC Variants and Features**

| Feature | Basic | Advanced |
|---|---|---|
| I2CC-ANALOG-FILTER | No | Yes |
| I2CC-DIGITAL-FILTER | Yes | No |
| I2CC-BURST | No | Yes |
| I2CC-SMBUS/PMBUS | No | Yes |
| I2CC-DMA | Refer to device-specific data sheet | Refer to device-specific data sheet |
| I2CC-FIFO | Refer to device-specific data sheet | Refer to device-specific data sheet |
| I2C-LOW-POWER-MODE | Yes, if module in PD0 | Yes, if module in PD0 |

**Table 30-2. UNICOMM I2CT Variants and Features**

| Feature | Basic | Advanced |
|---|---|---|
| I2CT-ANALOG-FILTER | No | Yes |
| I2CT-DIGITAL-FILTER | Yes | No |
| I2CT-SECOND-TARGET-ADDR | No | Yes |
| I2CT-SMBUS | No | Yes |

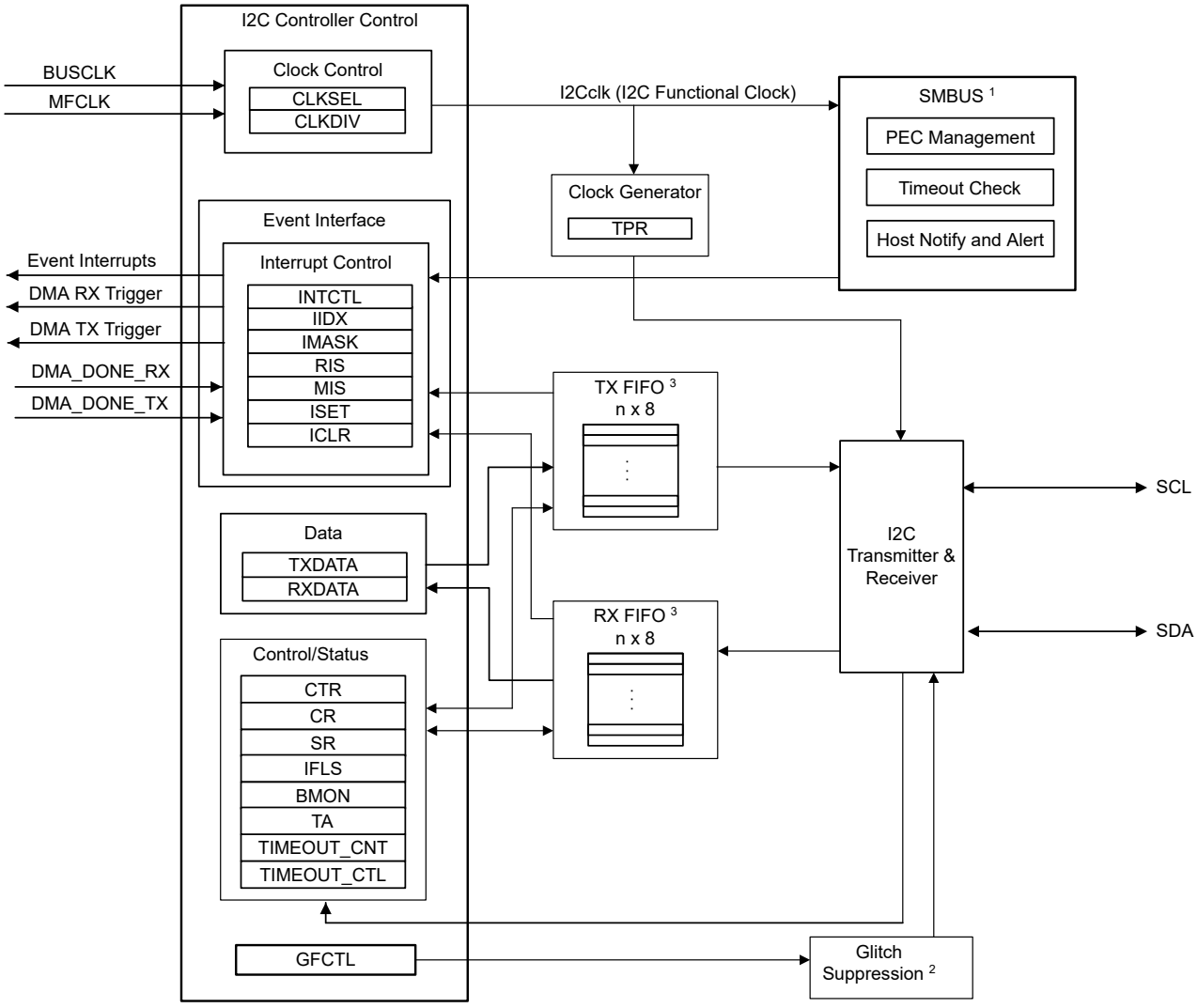


Copyright © 2026 Texas Instruments Incorporated

**Table 30-2. UNICOMM I2CT Variants and Features (continued)**

| Feature | Basic | Advanced |
|---|---|---|
| I2CT-DMA | Refer to device-specific data sheet | Refer to device-specific data sheet |
| I2CT-FIFO | Refer to device-specific data sheet | Refer to device-specific data sheet |
| I2C-LOW-POWER-MODE | Yes, if module in PD0 | Yes, if module in PD0 |

### *30.1.3 Functional Block Diagram*

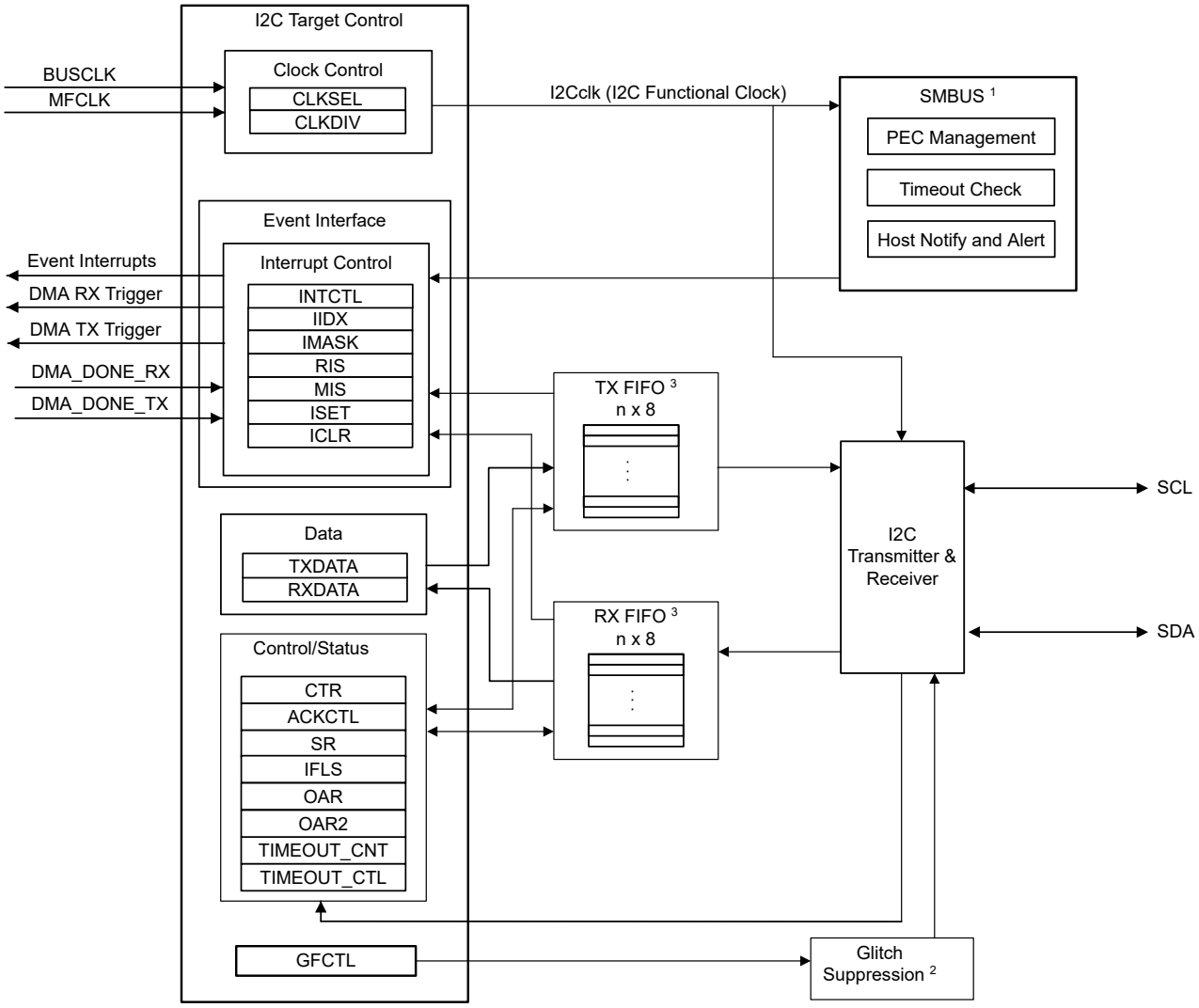



[1] Only Advanced I2CC instances have built-in SMBUS support
[2] Basic I2CC instances have only digital glitch support and Advanced I2CC instances have only analog glitch support
[3] The FIFO depth *n* can be 1 or a multiple of 4

**Figure 30-1. I2C Controller Functional Block Diagram**



Copyright © 2026 Texas Instruments Incorporated




[1] Only Advanced I2CT instances have built-in SMBUS support
[2] Basic I2CT instances have only digital glitch support and Advanced I2CT instances have only analog glitch support
[3] The FIFO depth *n* can be 1 or a multiple of 4

**Figure 30-2. I2C Target Functional Block Diagram**

### *30.1.4 Environment and External Connections*

The $I^2C$ mode supports any target or controller $I^2C$-compatible device. Figure 30-3 shows an example of an $I^2C$ bus. Each UNICOMM-I2C instance is comprised of controller or target functions where the target can be addressed with 2 independent user defined addresses. A device connected to the $I^2C$ bus can be considered as the controller or the target when performing data transfers. A controller initiates a data transfer and generates the clock signal SCL. Any device addressed by a controller is considered a target.

$I^2C$ data is communicated using the serial data (SDA) pin and the serial clock (SCL) pin. Both SDA and SCL are open-drain bidirectional and must be connected to a positive supply voltage using a pullup resistor.



Copyright © 2026 Texas Instruments Incorporated

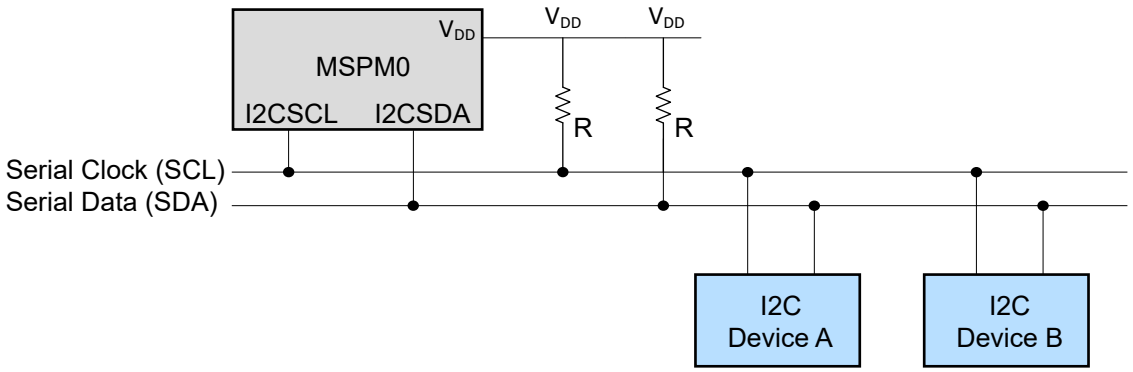



**Figure 30-3. Bus Connection Diagram**

## 30.2 UNICOMM Common Infrastructure

For details on common infrastructure elements refer to SPGSS (Section 27.2).

Sections under common infrastructure are – resets, mode selection, pin overlay, FIFO depth, FIFO operation, Interrupts and Events.

## 30.3 Peripheral Functional Description

### *30.3.1 Clock Control*

#### 30.3.1.1 Clock Select and I²C Speed

Standard, Fast, and Fast Plus modes are selected using a value in the I²C Controller Timer Period (TPR) register that results in a maximum SCL frequency of:

- 100 kbps for Standard mode
- 400 kbps for Fast mode
- 1 Mbps for Fast mode Plus

The I²C frequency I2C_FREQ is determined by the I2Cclk functional clock frequency and the TPR field, as well as the fixed SCL_LP , and SCL_HP values where:

- I2Cclk is the functional clock frequency to the I²C module. Note that the I²C internal functional clock is first divided from the source clock:
  - Use UCx_I2CC/T.CLKSEL register to select the source of the I²C functional clock
    - BUSSCLK: the current bus clock is selected as the source for I²C. The current bus clock depends on power domain. If the I²C instance is in power domain 1 (PD1) refer to MCLK, if the I²C instance is in power domain 0 (PD0) refer to ULPCLK.
    - MFCLK: MFCLK is selected as the source for I²C, refer to MFCLK.
    - The selected source clock is always available and its frequency depends on device power mode, for more information please refers to PMU/Clock section.
  - Use UCx_I2CC/T.CLKDIV register field to select divide ratio of I²C functional clock, options are from divide by 1 to 8. The division range is extensible to 64 by using parameter SYS_EXTEND_CLKDIV if applied.
- SCL_LP is the low phase of SCL: This is fixed at a value of 6
- SCL_HP is the high phase of SCL: This is fixed at a value of 4
- Time Period is the programmed value of the TPR field of the TPR register. This value is is calculated in the equation below and has to be programmed into TPR.

The I²C freqency is calculated as follows:

$$I2C\_FREQ = I2Cclk / ((1+TPR) * (SCL\_LP + SCL\_HP )) \quad (29)$$

For example, if the I²C functional clock frequency is 32 MHz and target SCL frequency is 400 kHz:



Copyright © 2026 Texas Instruments Incorporated

I2Cclk = 32 MHz

I2C_FREQ = 400 kHz

SCL_LP = 6 , SCL_HP = 4

TPR = (I2Cclk / (I2C_FREQ * (4 + 6))) - 1

TPR = 7 (0x07)

**Table 30-3. Examples of Controller Clock Setting for Typical Clock Configurations**

| Functional Clock | TPR Bits Standard Mode 100-kHz SCL | TPR Bits Fast Mode 400-kHz SCL | TPR Bits Fast Mode Pluds 1000-kHz SCL |
|---|---|---|---|
| 4 MHz | 0x03 | - | - |
| 8 MHz | 0x07 | 0x01 | - |
| 20 MHz | 0x13 | 0x04 | 0x01 |
| 32 MHz | 0x1F | 0x07 | 0x02(1) |
| 40 MHz | 0x27 | 0x09 | 0x03 |

(1) With 32-MHz functional clock, TPR = 0x01 generates 1.6-MHz SCL frequency, and TPR = 0x02 generates 1.067-MHz SCL frequency.

The $I^2C$ functional clock must be greater than or equal to 20 times the SCL frequency, I2Cclk ≥ 20 × I2C_FREQ.

The following minimum functional clock frequencies are required when running certain I2C clock speeds:

- I2Cclk ≥ 2 MHz when working with $I^2C$ speed 0 to 100 kHz
- I2Cclk ≥ 8 MHz when working with $I^2C$ speed 100 to 400 kHz
- I2Cclk ≥ 20 MHz when working with $I^2C$ speed 400 kHz to 1 MHz

#### 30.3.1.2 Clock Startup

The selected module clock source is always available and the frequency depends on the power mode, for more information refer to the PMU/Clock section. After enabling the $I^2C$ module by setting the PWREN.ENABLE bit from the UNICOMM top level, the module is ready to start receiving and transmitting data.

### *30.3.2 Signal Descriptions*

The $I^2C$ bus consists of a clock signal and a data signal. The clock signal can be generated either internally (during controller operation) or externally (during target operation).

**Table 30-4. $I^2C$ signal descriptions**

| Device Pin | Function |
|---|---|
| UCx_SCL | $I^2C$ clock signal |
| UCx_SDA | $I^2C$ data signal |

### *30.3.3 General Architecture*

#### 30.3.3.1 $I^2C$ Bus Functional Overview

The $I^2C$ bus uses only two signals: SDA and SCL. SDA is the bidirectional serial data line and SCL is the bidirectional serial clock line. The bus is considered idle when both lines are in high state and no transfer is ongoing.

Every transaction on the $I^2C$ bus is 9-bits long, consisting of 8 data bits and 1 acknowledge bit. A transaction is defined as the time between a valid START and STOP condition—as described in Figure 30-4. The number of bytes per transaction is unrestricted; however, each data byte must be followed by an acknowledge bit and data must be transferred MSB first. When a receiving device cannot receive another complete byte, it can hold the clock line SCL Low and force the transmitting device into a wait state.This process is commonly known as clock stretching. The data transaction continues when the receiver releases the clock SCL.



Copyright © 2026 Texas Instruments Incorporated

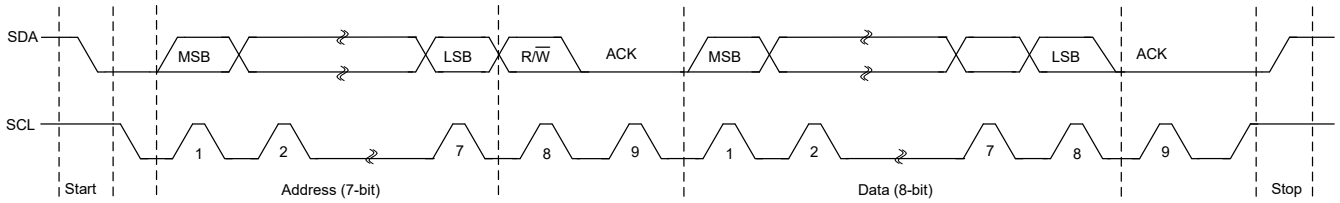



**Figure 30-4. Module Data Transfer**

The data on the SDA line must be stable during the high period of the clock, and the data line can only change when SCL is low (see Figure 30-5), otherwise START or STOP conditions are generated.

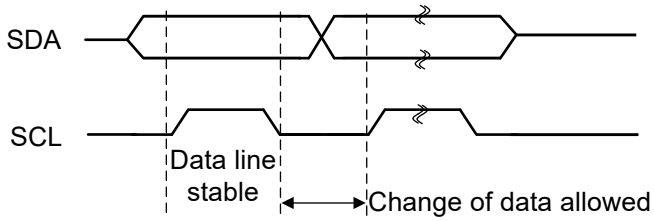



**Figure 30-5. Data Change Period**

#### 30.3.3.2 START and STOP Conditions

The protocol of the $I^2C$ bus defines two states to begin and end a transaction: START and STOP. A high-to-low transition on the SDA line while the SCL is high is defined as a START condition, and a low-to-high transition on the SDA line while SCL is high is defined as a STOP condition. START and STOP conditions are always generated by the controller. Controller mode support for generating START/STOP bits: use CTR.START, CTR.STOP bits. The bus is considered busy after a START condition and free after a STOP condition.

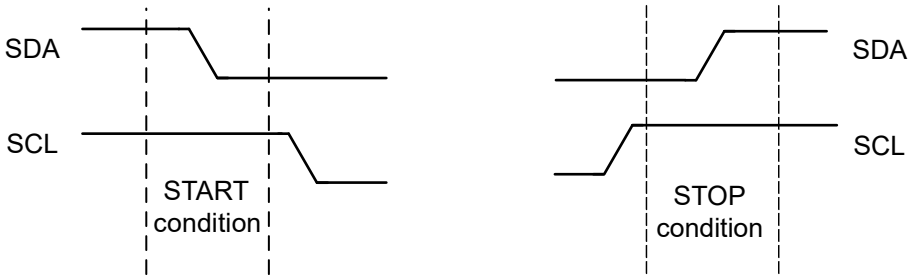



**Figure 30-6. START and STOP Conditions**

The STOP bit determines if the transaction stops at the end of the data cycle or continues on to a repeated START condition.

To generate a single transaction, the $I^2C$ controller target address TA.ADDR register is written with the desired address, the TA.DIR bit should be set to 1 to start a receive operation and 0 to start a transmit operation , and the control register (CTR) is written with ACK = X (0 or 1), STOP = 1, START = 1, and FRM_START = 1 to perform the operation and generate a STOP at the end. When the operation is completed (or aborted due an error) the interrupt flags are set. When the $I^2C$ module operates in Controller receiver mode, the ACK bit is normally (in case of no error) set causing the $I^2C$ bus controller to transmit an acknowledge automatically after each byte. This bit must be cleared when the $I^2C$ bus controller requires no further data to be transmitted from the target transmitter. For more information on $I^2C$ controller mode configuration please refer to Section 30.3.4.1.

When operating in target mode, the START and STOP bits in the CPU_INT.RIS register indicate detection of start and stop conditions on the bus and the CPU_INT.IMASK can be configured to allow START and STOP to be promoted to controller interrupts (when interrupts are enabled). For more information on $I^2C$ target mode configuration please refer to Section 30.3.4.2.



Copyright © 2026 Texas Instruments Incorporated

Bus Status Flags:

- BUSBSY is set when a START is detected on the bus and is sequentially cleared when a STOP condition is detected on the bus. It is also cleared after a clock timeout is set once both SCL and SDA are pulled to high.
- BUSY is also set after a START/Repeated START, and cleared after CTR.BLEN bytes of data is transferred. It is also cleared after data is NACK ‘d or a STOP.
- IDLE is set when the Controller I$^2$C state machine is in the IDLE state indicating there is no ongoing transfer.

#### 30.3.3.3 Dual Address

The I$^2$C target interface supports dual address capability for the Advanced I2CT instances. An additional programmable I$^2$C Target Own Address Register 2 (OAR2) is provided and can be matched if enabled. When dual address is disabled (OAR2.OAR2EN=0), the I$^2$C target provides an ACK on the bus if the address matches the OAR field in the OAR register. When dual address mode is enabled (OAR2.OAR2EN=1), the I$^2$C target provides an ACK on the bus if either the OAR field in the OAR register or the OAR2 field in the OAR2 register is matched.

Matched address is loaded into target status register address match field (TR.ADDRMATCH) and OAR2SEL bit indicates if own address (status bit reads back '0') or own address 2 was matched (status bit reads '1').

#### 30.3.3.4 Address Format

I2C protocol supports 7-bit and 10-bit addressing modes. In both controller and target modes, 7-bit and 10-bit modes are supported.

- In controller mode, target address and mode are loaded into TA.ADDR and TA.MODE fields
- In target mode, two sets of address matching options are provided – own address and own address 2
  - Own address (OAR register) can be used for 7-bit or 10-bit addressing
  - Own address 2 (OAR2 register) supports only 7-bit addressing mode; OAR2 also provides an address masking option

##### *30.3.3.4.1 Data Format with 7-Bit Address*

Data transfers follow the format shown in Figure 30-7. After the START condition, a target address is transmitted. This address is 7-bits long followed by an eighth bit, which is a data direction bit (this bit is only as Controller mode, DIR bit in the TA register). If the TA.DIR bit is 0, it indicates a transmit operation (send), and if it is set to 1, it indicates a request to receive data (receive). A data transfer is always terminated by a STOP condition generated by the controller; however, a controller can initiate communications with another device on the bus by generating a repeated START condition and addressing another target without first generating a STOP condition, see section Repeated Start. Various combinations of receive/transmit formats are then possible within a single transfer. The ninth bit is the Acknowledge bit, which is described in Section 30.3.3.5 .

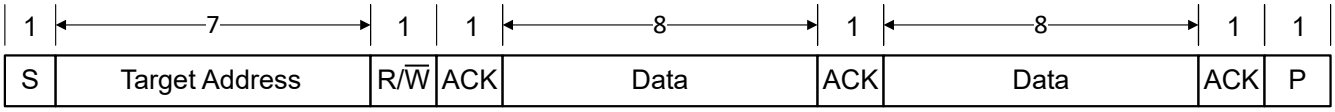



**Figure 30-7. Data Format with 7-Bit Address**

With the CTR.GENCALL bit the I$^2$C module can be enabled to respond on a General Call on the I$^2$C bus. The General Call is identified by address of 0x00 and the R/W bit set to 0. The General Call interrupt can be enabled with the CPU_INT.IMASK.GENCALL bit.

##### *30.3.3.4.2 Data Format with 10-Bit Address*

Data transfers with 10-bit address follow the format shown in below. After the START condition, a leading ***'11110b'*** introduces the 10-bit addressing scheme. The last two address bits of the first byte concatenated with the eight bits of the second byte of the whole 10-bit address. Devices which only use 7-bit addressing simply ignore messages with the leading '11110b'. The data direction bit (this bit is only as Controller mode, DIR bit in the TA register) is located in the first 8-bit data.



Copyright © 2026 Texas Instruments Incorporated

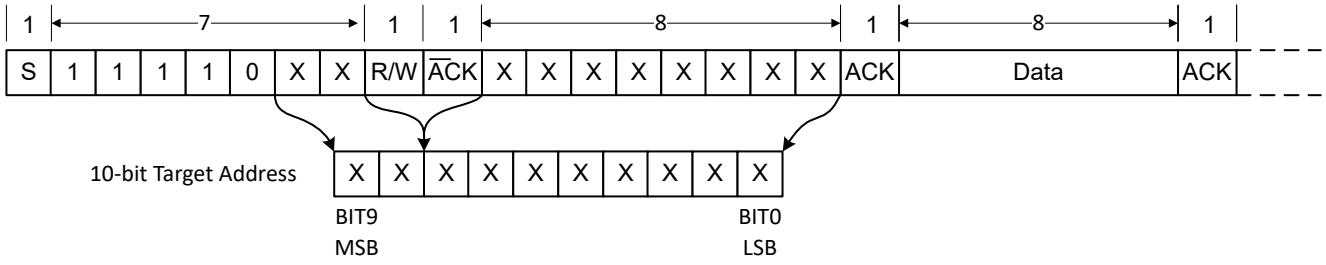



**Figure 30-8. Data Format with 10-Bit Address**

A ***Controller transmitter*** addresses the Target with two address bytes as described below with the RW-Bit='0' followed by data bytes from the Controller.

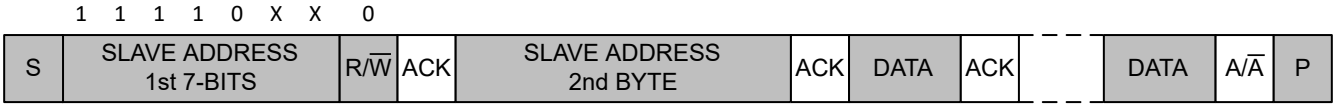



**Figure 30-9. I2C Controller Transmit with 10-Bit Address**

The ***Controller receiver*** transfer is only possible with a Combined Transfer due to the fact that the second address byte can only be transmitted if the RW-Bit of the first address byte is '0'. Hence, the start of a Controller receiver transfer will be the same as a Controller transmitter transfer followed by a repeated start condition and the first byte of address byte with RW-Bit='1' (switching to Target transmitter mode).

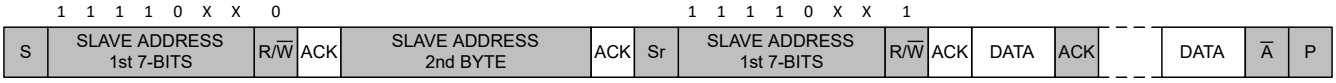



**Figure 30-10. I2C Controller Receive with 10-Bit Address**

#### 30.3.3.5 Acknowledge

All bus transactions have a required acknowledge clock cycle that is generated by the controller. During the acknowledge cycle, the transmitter (which can be the controller or target) releases the SDA line. To acknowledge the transaction, the receiver must pull down SDA during the acknowledge clock cycle. The acknowledge cycle must comply with the data validity requirements.

When a target receiver does not acknowledge the target address, SDA must be left high by the target so that the controller can generate a STOP condition and abort the current transfer or generate a repeated START condition to start a new transfer. If the controller device is acting as a receiver during a transaction, it is responsible for acknowledging each transfer made by the target. Because the controller decides the number of bytes in the transaction, it signals the end of data to the target transmitter by not generating an acknowledge on the last data byte. The target transmitter must then release SDA to allow the controller to generate the STOP or a repeated START condition.

In target receive mode, hardware will always generate an ACK after receiving data. A target can generate an ACK/NACK manually or automatically.

- When ACKCTL.ACKOEN=0, the target will send automatic acknowledgements. The I2CT will receive and ACK all bytes automatically until the RX FIFO is full.
- When ACKCTL.ACKOEN =1, manual acknowledgments are enabled. Manual ACK override can be used to evaluate each received byte or to slow down the communication when automatic FIFO reception is not desired. When manual ACK override operation is enabled, the $I^2C$ target module's clock is pulled low after the last data bit until this ACKCTL.ACKOVAL is written with the indicated response. The reception of new data is indicated by the RXDONE interrupt flag.



Copyright © 2026 Texas Instruments Incorporated

**Note**

Manual acknowledge is not supported when clock stretching is disabled for target receive mode.

If the I²C controller receives a NACK while transmitting data, the NACK and TXDONE bit will be set in the RIS registers. If there is still data in the TX FIFO, the TXEMPTY bit will remain clear to inform software that a TX FIFO flush may be required.

#### 30.3.3.6 Repeated Start

The direction of data flow on SDA can be changed by the controller, without first stopping a transfer, by issuing a repeated START condition. This is called a RESTART, see data format in Figure 30-11. After a RESTART is issued, the target address is again sent out with the new data direction specified by the R/W bit.

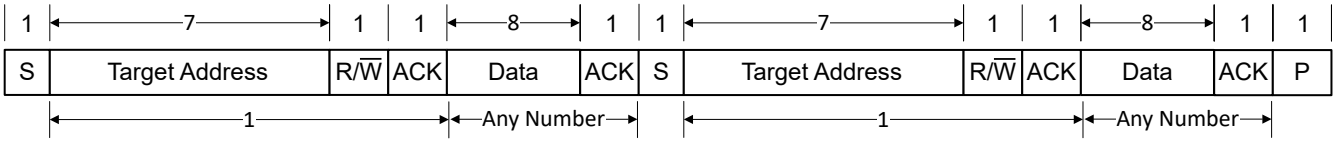



**Figure 30-11. Data Format with Repeated Start**

A repeated start sequence for a controller transmit or receive is as follows:

1. When the device is in the idle state, the Controller writes the target address to the TA register and configures the DIR bit for the desired transfer type.
2. Data is written to the TXDATA register (for transmit) or RXDATA register (for receive).
3. When SR.BUSY status reads back as '0', FRM_START and START bits in the CTR register are set to initiate a transfer.
4. Software waits till BUSY bit in the SR register reads back as '0'.
5. The controller can now change direction of transfer or address another target, by generating a restart condition. Software updates TA register and then, sets FRM_START. This initiates a repeated start on the line.

**Note**

If there is a NACK for address, a STOP condition will be automatically generated. RESTART cannot be sent in that case.

#### 30.3.3.7 Clock Stretching

For a UNICOMM-I2CC, clock stretching can be disabled if none of the targets on the bus support clock stretching, allowing the controller to reach the maximum speed on the bus. Otherwise, the clock can be slowed by a target keeping the clock low or due to the clock status detection delay within the I²C module.

For a UNICOMM-I2CT, clock stretching is activated automatically when either the RX FIFO full or TX FIFO empty is set. Clock stretching support can be enabled or disabled by configuring the CR.CLKSTRETCH bit, where the feature is enabled by default. Clock stretching status is indicated by the TREQ and RREQ bits from the SR register described below.

- TREQ
  - When this bit is set, the I²C target has been addressed as a target transmitter and is using clock stretching to delay the controller until data has been written to TXDATA.
- RREQ
  - When this bit is set, the I²C controller has outstanding receive data from the I2C controller and is using clock stretching to delay the controller until the data has been read from RXDATA.



Copyright © 2026 Texas Instruments Incorporated

**Note**

When device is in low power mode, clock stretching in target mode may be used together with an asynchronous fast clock request to support bringing the device into a suspended low power mode state upon detection of an I2C start bit, enabling the I2C module to support 100kHz (standard mode) out of low power modes where the bus clock speed is below the minimum oversampling speed required by the respective mode. Refer to the clock selection and $I^2C$ speed section for the minimum frequency requirements. When clock stretching is used together with an asynchronous fast clock request, it is possible for the device to wait in STOP or STANDBY mode when the $I^2C$ is idle, and when an $I^2C$ bus edge is seen, the fast clock request will requests SYSOSC at base frequency and the bus clock will switch to SYSOSC at base frequency, allowing the $I^2C$ module to process the bus transaction and wake the processor if an interrupt is generated.

#### 30.3.3.8 Clock Low Timeout

An $I^2C$ target can extend the transaction by pulling the clock low to slow down communication. The UNICOMM-I2C module has TCNTA, a 12-bit programmable counter that tracks how long the clock has been held low. The upper 8 bits of the count value are software programmable through the I2CTIMEOUT_CTL register. The value programmed in the I2CTIMEOUT_CTL.TCNTLA register field has to be greater than 0x01 to enable the timeout feature.

The application can program the eight most significant bits of the counter to reflect the acceptable cumulative low period in a transaction. Each count is equal to a timeout period of (1 + TPR) × 12 I2Cclks' where the TPR is the programmable timer period. The Timeout counter A counts for the entire time SCL is held Low continuously. When SCL is high, the Timeout counter A is reloaded with the value in the I2CTIMEOUT_CTL.TCNTLA register bit, and begins counting down from this value at the falling edge of SCL.

The internal bus clock generated for the timeout counter keeps running at the programmed $I^2C$ speed even if SCL is held low on the bus. The $I^2C$ clock low timeout period is calculated as follows:

- Cumulative clock low period = Timeout counter * One timeout period
- One timeout period = I2Cclk period * (1 + TPR) * 12
- Timeout counter = I2CTIMEOUT_CTL.TCNTLA register (this register contains the upper 8-bits of a 12-bit counter value, the lower 4-bits are set to 0)

As an example, if the functional clock is 20 MHz and the $I^2C$ module was operating at 100-kHz speed, the TPR would be equal to 19. See Section 30.3.1.1 on how TPR is calculated. One timeout period is equal to (1 / 20 MHz) × ( 1 + 19) × 12 or 12 µs. Programming the I2CTIMEOUT_CTL.TCNTLA register to 0xDA would translate to the value 0xDA0, because the lower 4-bits are set to 0x0. This would translate to a decimal value of 3488 clocks or a cumulative clock low period of 3488 × 12 µs, or 41.856 ms at 100 kHz.

The TIMEOUTA bit in the $I^2C$ Controller Raw Interrupt Status CPU_INT.RIS register is set when the clock timeout period is reached, allowing the controller to start corrective action to resolve the remote target state. In addition, the BUSBUSY bit in the $I^2C$ Controller Status SR register is set. This bit is cleared after $I^2C$ goes to idle or during the $I^2C$ controller reset. The status of the raw SDA and SCL signals are readable by software through the SDA and SCL bits in the $I^2C$ Controller Bus Monitor BMON register to help determine the state of the remote target.

In the event of a timeout condition, application software must choose how it intends to attempt bus recovery. If a clock low timeout is detected before the end of a transfer (receive or transmit), software should flush the FIFO before initializing the next transfer. The clock low timeout is needed for SMBus and PMBus implementation.

**Note**

SCL low timeout configuration needs to be set only during initialization and not during active state.

#### 30.3.3.9 Burst Mode

I2C Controller burst mode allows a sequence of data transfers using DMA or software to handle the data in the buffer/FIFO. This mode is only available on Advanced I2CC module instances.Burst mode is enabled by



Copyright © 2026 Texas Instruments Incorporated

setting the CTR.BLEN bits in the I2CC control register to a value 'N', representing the numbers of bytes in the transfer. 'N' must be greater then '1'. This sets the number of bytes transferred by a burst. A copy of this value is automatically written to the SR.BCNT bits in the I2C controller status register to be used as a down-counter during burst transfer.

Interrupt bits RXDONE and TXDONE are set after a burst of 'N' has been transferred.

### NACK in Controller Burst Transmit Sequence

If data is not acknowledged during a controller transmit burst sequence and the CTR.STOP bit ***is*** set, the frame transfer is terminated. Upon receiving a no-acknowledge condition, the controller sets the NACK interrupt bit, sends out a STOP condition on the bus, and terminates the transfer.

If data is not acknowledged during a controller transmit burst sequence and CTR.STOP bit ***is not*** set, hardware waits for software to proceed further. Upon receiving a no-acknowledge condition, the controller NACK interrupt bit is set. Software can read the SR.BCNT field to determine the amount of data that was transferred prior to the burst termination. Then software can send out a STOP condition by setting CTR.STOP or a RESTART condition by setting CTR.FRM_START.

#### 30.3.3.10 Arbitration

A controller can start a transfer only if the bus is in the IDLE state. It is possible for two or more controllers generate a START condition within the minimum hold time of the START condition. In these situcations, an arbitration scheme takes place on the SDA line while SCL is high (see Figure 30-12) . The first controller transmitter that generates a logic high is overruled by the opposing controller generating a logic low and the losing controller releases the bus until the bus is idle again.

Arbitration can take place over several bits. The first stage is a comparison of address bits, and if both controllers are trying to address the same device, arbitration continues on to the comparison of data bits.

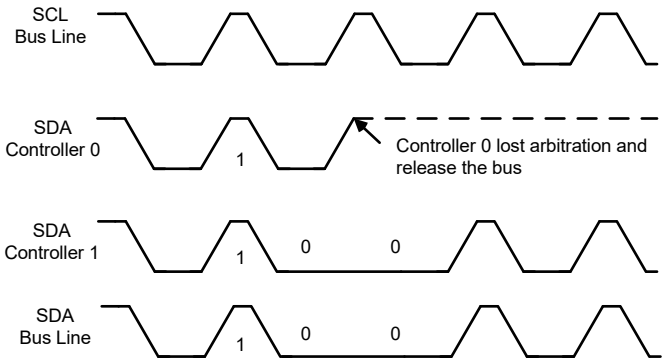



**Figure 30-12. Arbitration Procedure Between Two Controller Transmitters**

When an arbitration loss is detected, the SR.ARBLST flag is set. The flag is reset by the hardware with the next STOP condition detected on the bus. Additionally, the ARBLOST flag in CPU_INT.RIS registers are set.

If arbitration is lost when the $I^2C$ controller has initiated a transfer, the application software needs to execute the following steps to correctly handle the arbitration loss:

- If transmit FIFO is not empty, clear the TX FIFO by setting IFLS.TXCLR bit
- Once the bus is IDLE, the TX FIFO can be filled and enabled, the TXEMPTY bit can be unmasked, and a new transaction can be initiated.

#### 30.3.3.11 Multiple Controller Mode

To enable multiple-controller mode of operation, the CR.MCTL bit can be set. During the arbitration procedure, the clocks from each different controller must be synchronized. A device that first generates a low period on SCL overrules the other devices, which forces the other devices each to start low periods as well. SCL is then



Copyright © 2026 Texas Instruments Incorporated

held low by the device with the longest low period. The other devices must wait until the SCL is released before starting high periods.

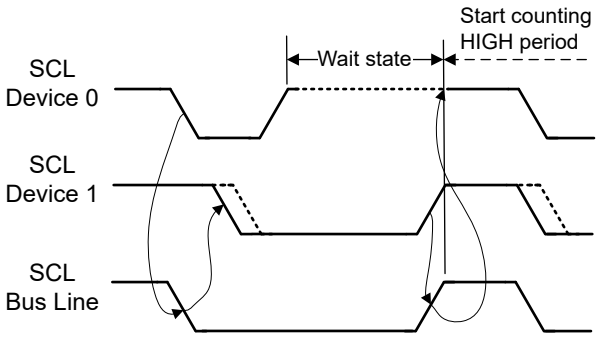



**Figure 30-13. Clock Synchronization**

#### 30.3.3.12 Glitch Suppression

The $I^2C$ module supports glitch suppression on the SCL and SDA lines to meet the 50ns glitch suppression as specified in the $I^2C$ specification.

**Analog glitch Filter**

The analog glitch filter feature is only available for Advanced UNICOMM-I2CC and UNICOMM-I2CT instances.

In Advanced UNICOMM-I2C instances, by default, an analog glitch filter is enabled and configured to suppress spikes with a 50ns pulse width. $I^2C$ spec advises to suppress noise spikes of less than 50ns. The user can disable this filter by clearing the GFCTL.AGFEN bit. The analog glitch filter can only be used to wake up the I2C in low-power mode.

**Digital Glitch Filter**

The digital glitch filter feature is only available for Basic UNICOMM-I2CC and UNICOMM-I2CT instances.

In Basic UNICOMM-I2C instanceks, the DGFSEL bits in the GFCTL register can be programmed to provide glitch suppression on the SCL and SDA lines and maintain proper signal values. The glitch suppression value is in terms of the $I^2C$ functional clock cycles. All signals are delayed internally when glitch suppression is nonzero. For example, if DGFSEL bit is set to 0x7, 31 clocks can be added onto the calculation for the expected transaction time. To disable the digital glitch filter, set the DGFSEL field to zero.

The digital glitch filter can't be used to wake up the I2C from low-power mode.

**Table 30-5. Glitch Suppression Filter**

| | Analog Glitch Filter | Digital Glitch Filter |
|---|---|---|
| **Default** | Default enabled with 50ns | Default bypassed |
| **Pulse width of suppressed spikes** | Fixed with 50ns | Programmable $I^2C$ clock cycle<br>1, 2, 3, 4, 8, 16, 31 |
| **Benefits** | Available without needing clock | • Programmable length to provide extra filtering .<br>• Stable filtering length. |
| **Limitation** | Variation with temperature, voltage, process | • Does not work in low-power mode wakeup when there is no sufficient clock .<br>• Only enabled 3 clock cycles after start of $I^2C$ packet. |



Copyright © 2026 Texas Instruments Incorporated

**Note**

Modifications to the GFCTL register must only be done while the CR.ENABLE bit is cleared.

#### 30.3.3.13 DMA Operation

The UNICOMM $I^2C$ modules, I2CC and I2CT, each provide an interface to the DMA module with seperate channels for transmit and receive. The DMA operation of the $I^2C$ is enabled through the $I^2C$ DMA Event and DMA module registers. When the DMA functionality is enabled, the $I^2C$ asserts a DMA request on the selected channel when the associated FIFO can transmit or receive data. The DMA transfer requests are handled automatically by the DMA controller based on how the DMA channel is configured (burst size, transfer size, source address etc.).

For more information about $I^2C$ event and DMA, please refer to Interrupt and Events Support and DMA Trigger Publisher sections.

- For the receive channel, a DMA transfer request is asserted when the amount of data in the receive FIFO is at or above the FIFO trigger level configured using the RXIFLSEL bit in IFLS register.
- For the transmit channel, a DMA transfer request is asserted whenever the transmit FIFO contains fewer characters than the FIFO trigger level configured using the TXIFLSEL bit in IFLS register.

**Note**

Per DMA channel, only one event source should be enabled by the IMASK register at the same time. The DMA transaction descriptor needs to match for the selected trigger with a correct setup for either Controller or Target and RX or TX configuration. The DMA trigger should only be changed when no $I^2C$ transfer is ongoing and a previous triggered DMA transfer has been finished. If this cannot be ensured the $I^2C$ and DMA channel should be disabled first.

#### 30.3.3.14 FIFO Operation

UNICOMM-I2CC or UNICOMM-I2CT has two dedicated FIFOs, one for receive operations and one for transmit operations. The TXDATA and RXDATA registers are 8-bit and used to interface with the transmit and receive FIFOs respectively. The FIFO level varies by device, and is indicated in the device-specific data sheet.

##### *30.3.3.14.1 FIFO Status Flags*

Each of the common available FIFO status bits for UNICOMM-I2CC/T are explained in the table below. All status bits are included in the Status Bits Register (SR) and can be used by application software to poll the FIFO status as needed.

**Table 30-6. FIFO Status Bits**

| Status Bit | Description |
|---|---|
| RXFE | This bit is set when the RX FIFO is empty. |
| RXFF | This bit is set when the RX FIFO is full. |
| RXCLR | This bit is set when a clear of the RX FIFO, started by setting the IFLS.RXCLR bit, is complete. |
| TXFE | This bit is set when TX FIFO is empty. |
| TXFF | This bit is set when TX FIFO is full. |
| TXCLR | This bit is set when a clear of the TX FIFO, started by setting the IFLS.TXCLR bit, is complete. |

If the FIFOs are not present, the empty and full flags are set according to the status of the 1-byte-deep holding registers.



Copyright © 2026 Texas Instruments Incorporated

**Note**

A write is lost if it overruns the TX FIFO.

**Note**

Data is lost if the RX FIFO is full, and the overflow flag bit is set.

#### *30.3.3.14.2 FIFO Levels*

The interrupt FIFO level for UNICOMM-I2CC/T is configured in the IFLS register.

**Receive FIFO Levels**

When a message gets received, the RX FIFO gets filled immediately. The configurable threshold options for the Receive (RX) FIFO are:

1. RX_FIFO full: when all elements are populated
2. RX_FIFO >= ¼ full
3. RX_FIFO >= ½ full
4. RX_FIFO >= ¾ full
5. RX_FIFO almost empty: <= 1 element
6. RX_FIFO almost full: >= (N-1) elements
7. RX_FIFO not empty: >= 1 element

These various conditions are typically used to generate a CPU_INT or DMA_TRIG_RX event where the CPU/RX subsequently reads a specific number of characters into memory from RXDATA. The number of characters read in each Interrupt Service Routine (ISR) equals how many are ensured present in the FIFO as indicated by the flag.

**Note**

Debugger reads of RXDATA have no effect on the contents in the RX FIFO.

**Transmit FIFO Levels**

On the transmit side, data is written by the CPU/DMA at the top most empty FIFO entry and then transferred to serializer. The configurable threshold options for the Transmit (TX) FIFO are:

1. TX_FIFO empty: when there no elements in the FIFO
2. TX_FIFO <= ¾ empty
3. TX_FIFO <= ½ empty
4. TX_FIFO <= ¼ empty
5. TX_FIFO almost empty: <= 1 element
6. TX_FIFO almost full: >= (N-1) elements
7. TX_FIFO not full: <= (N-1) elements

These various conditions are typically used to generate a CPU_INT or DMA_TRIG_TX event where the CPU/DMA subsequently writes a specific number of characters into the TXDATA register. The number of characters written in each Interrupt Service Routine (ISR) equals how many spots are ensured empty in the FIFO as indicated by the flag.

**Note**

Debugger writes of TXDATA will add elements to the TX FIFO the same as a CPU write.



Copyright © 2026 Texas Instruments Incorporated

**FIFO Levels Reset Status**

Out of reset, both TX and RX FIFOs are configured to trigger an interrupt at half-level mark, and the content of FIFO are also cleared.

**Note**

FIFO integrity is not guaranteed if the software disables the module in the middle of a transmission with data in the FIFO, and then re-enables it.

#### *30.3.3.14.3 Clearing FIFO Contents*

Setting the IFLS.RXCLR and IFLS.TXCLR register bits clears the contents of each of the respective FIFOs.

Follow the below sequence:

1. Write a 1 to the RXCLR or TXCLR bit in the IFLS peripheral-specific register.
2. Wait until the RXCLR or TXCLR status bit in the STAT register is set to 1.
3. Write a 0 to the RXCLR or TXCLR bit in the IFLS peripheral-specific register.

**Note**

Performing the above clear sequence before changing the FIFO level configuration in the RXIFSEL or TXIFSEL fields is recommended.

**Note**

Avoid performing step 1 repeatedly before full sequence is complete.

### 30.3.3.15 Suspend Communication

Suspend is an extreme condition of operation of the I2C module and application should attempt this mode while taking care of the below conditions. Application can suspend external communication by setting CTR.SUSPEND bit.

When SUSPEND is set, ongoing communication will complete, FSM will return to idle and further external communications are stopped. Transmit line/s will be driven to its idle state and further toggle on receive line/s will not be processed. This provides a safe means for application to suspend operations, clear out receive FIFOs prior to disabling the module. SUSPEND bit needs to be cleared prior to re-enabling the module.

Following the steps below to suspend communication and disable I2C module:

1. I2C bus masters have to ensure that the peripheral can reach idle state (detailed in I2C Bus Prerequisites)
2. Hardware ensures that ongoing transactions complete; in case of transmit operation, all the contents of the FIFO will be sent out, prior to hardware raising an idle state
3. Once SUSPEND bit is written into hardware, CPU has to poll status flags to ensure that device has reached idle
4. Drain out receive FIFO
5. Disable the module

Post suspend/disable module, register configurations are retained. Communication can be resumed by following the below steps on re-enabling the module:

1. Clear SUSPEND bit
2. Re-enable the module



Copyright © 2026 Texas Instruments Incorporated

### I2C Bus Prerequisites

To reach IDE state, for I2C module, hardware will ensure that transmit fifo is drained out and will provide an indication that the line is idle– SR.BUSY reads ‘0’. STOP condition may also need to sent / received as per details below:

1. I2C mode
   a. Controller mode: ensure either NACK/STOP is received/transmitted on the line.
      i. Transmitter mode: ensure that receiving target has raised a NACK
      ii. Receiver mode: ‘NACK’ has to be sent on the line
      iii. In mutli-controller systems, ensure that STOP condition has been sent out by the controller.
   b. Target mode: ensure that a NACK has been received on the line
      i. Transmitter mode: ensure that receiving controller has raised a NACK
      ii. Receiver mode: ‘NACK’ has been sent on the line
2. SMBUS mode
   a. Controller mode: ensure that STOP is transmitted by the controller
   b. Target mode: ensure that NACK has been received on the line

#### 30.3.3.16 Low Power Operation

$I^2C$ supports operation in Low Power Modes. In low-power mode, the $I^2C$ interface automatically request the clock selected in the CLKSEL control register after detecting a start condition. The clock request will be released after detection of the STOP condition.

Supported power mode for controller mode:
- Controller mode with 100kHz speed can operate in RUN, SLEEP, STOP power mode
- Controller mode with 400kHz speed can operate in RUN, SLEEP power mode
- Controller mode with 1MHz speed can operate in RUN, SLEEP power mode

Supported power mode and wakeup mode for target mode:
- Target mode with 100kHz speed can operate in RUN, SLEEP, STOP power mode
- Target mode with 100kHz speed can't operate in STANDBY mode because the maximum bus clock in STANDBY mode is 32kHz and to support 100kHz speed we need minimum 400kHz clock. However, in STANDBY mode I2C target can still wakeup from the start bit and perform the (Asynchronous Fast Clock Requests) to temporarily get a 32MHz clock to receive the data until the FIFO or address match interrupt wakes up the CPU.
- Target mode with 400kHz speed in RUN, SLEEP power mode
- Target mode with 1MHz speed in RUN, SLEEP power mode

**Note**

To support STANDBY wakeup with I2C standard mode (100KHz), functional clock should be at 32MHz, i.e. there should not be a further division of 32MHz clock.

#### 30.3.3.17 SMBUS 3.0 Support

The SMBus protocol reuses a number of features of I2C protocol so that the same device can be used to support SMBus as well. This protocol is used in a lot of industrial applications like Battery Management, Memory SSD and several other use cases. See SMI Forum SMBus Specifications for the details of the protocol specification. The additional registers and features supporting the SMBus protocol are only present on UNICOMM instances with Advanced I2CC and I2CT capabilities.



Copyright © 2026 Texas Instruments Incorporated

#### *30.3.3.17.1 Quick Command*

Quick Command is a simple, compact SMBus transaction that sends an address and 1-bit of data in the DIR bit of the I2C header byte to communicate a command to the target, typically a "turn off" or "turn on". The controller has the ability to send a Quick Command by writing the target address and direction value into the TA register, followed by setting the START, STOP, and FRM_START bits in the CTR register and clearing the BLEN field to 0. The target logic also has quick command support built in.

In UNICOMM-I2C Target mode, the TX FIFO must be empty upon receiving a Quick Command and can only be loaded after decoding the received message. Follow the steps below to receive a Quick Command from an I2CT:

1. TX FIFO must be empty
2. Receive and decode a Quick Command message using the SR.QCMDST and SR.QCMDRW bits.
3. If clock stretching is enabled, software must load the TX FIFO with 0xFF to stop clock stretching and allow the I2C Controller to generate a STOP condition.




**Figure 30-14. Quick Command Message**

#### *30.3.3.17.2 SMBUS Enhanced Acknowledge Control*

**I2C Target Enhanced ACK Control**

The below additonal bits in the ACKCTL register provide enhanced acknowledgement functionality. The status bit SR.ACKOEN is updated automatically when the ACKOEN_ON_PECDONE, ACKOEN_ON_PECNEXT, or ACKOEN_ON_START bits in ACKCTL are set. Software has to write ACKCTL.ACKOEN to ‘0’ to clear the acknowledge override condition generated by hardware.

- **ACKOEN_ON_PECDONE**
  - When set, this bit automatically turns on the target override mechanism following the ACK/NACK of the received PEC byte. This allows software to manually handle data bytes received after the PEC byte. The sequence is as follows:
    1. Set ACKCTL.ACKOEN_ON_PECDONE
    2. SR.ACKOEN bit is automatically set after the PEC byte has been ACK/NACK'd
    3. Receive "n" bytes using manual software ACK
    4. Clear SR.ACKOEN after "n" bytes have been received
    5. Receive next "m" bytes using the automatic hardware ACK
- **ACKOEN_ON_PECNEXT**
  - When set, this bit automatically turns on the target override mechanism following the ACK/NACK of the byte received prior to the PEC byte. Software must clear the SR.ACKOEN after ACKing/NACKing the PEC byte. Using this option, software can effectively get the synchronization point by getting an interrupt here (TREQ/RREQ condition with TXEMPTY_ONTREQ/RXFLL_ON_RREQ set), and then manually send the ACK/NACK of the PEC. In other words, the software doesn’t need to rely on byte granularity RXMODE and TXMODE interrupts and get an interrupt after the entire frame has been TX/RX. The sequence is as follows:
    1. Set ACKCTL.ACKOEN_ON_PECNEXT
    2. SR.ACKOEN bit is automatically set before the PEC bytes
    3. Receive the PEC byte and manually ACK/NACK using software
    4. Clear SR.ACKOEN
    5. Receive the entire next frame using automatic hardware ACK until the next PEC byte is received
- **ACKOEN_ON_START**
  - When set, this bit automatically turns on the target override mechanism following a START condition. The sequence is as follows:
    1. Set ACKCTL.ACKOEN_ON_START



Copyright © 2026 Texas Instruments Incorporated

2. SR.ACKOEN bit is automatically set when the START condition is detected
3. Continue receiving bytes with software until remaining bytes are known
4. Clear SR.ACKOEN
5. Receive the rest of the frame using automatic hardware ACK until the next START condition is received

**Note**

The I2CT software must wait a bit duration between generating a NACK and disabling the SR.ACKOEN. These operations cannot happen in the same cycle.

### I2C Controller Enhanced ACK Control

The I2CC, when in receive mode, can ACK/NACK after "n" number of bytes are received. When CTR.ACKOEN is set and the number of bytes configured by CTR.BLEN are received, an RXDONE interrupt is generated and the transaction is either ACK'd or NACK'd depending on the value of CTR.ACK. To stop the transaction, the CTR.STOP can be set. Otherwise, software can update the CTR.BLEN field to a new value to continue with the transaction.

#### *30.3.3.17.3 Clock Low Timeout Detection*

Clock **low** timeout detection is done by setting the TIMEOUT_CTL.TCNTAEN and configuring the timeout period in the TIMEOUT_CNT.TCNTLA (the upper 8-bits of a 12-bit period) register field. The TIMEOUT_CNT.TCNTA register contains the current count value.

Example counter calculations for low SCL timeout detection is shown below.

- For SCL low detect, each count is equal to 520 cycles of the I2Cclk period.
- A pre-multiplier of 520 is used to make the total timeout period greater than 35ms (tTIMEOUT,MAX specified by SMBus 3.0), even at the highest clock frequency of 40MHz.
- At 40MHz, 1 count = 520 * 25ns = 13us
- Max count value = 12'b1111_1111_0000 = 12'd4080
- Total time configurable = 4080 * 13us = 53.04ms

#### *30.3.3.17.4 Clock High Timeout Detection*

Clock **high** timeout detection is done by setting the TIMEOUT_CTL.TCNTBEN and configuring the timeout period in the TIMEOUT_CNT.TCNTLB (the upper 8-bits of a 12-bit period) register field. The TIMEOUT_CNT.TCNTB register contains the current count value.

Example counter calculations for high SCL timeout detection is shown below.

- For SCL high detection, each count is equal to 1 * I2Cclk period.
- A pre-multiplier of 1 is used to make the accuracy of the counter remain high for 50 µs ($t_{HIGH,MAX}$) even for low frequency functional clocks (I2Cclk) .
- At 10MHz, 1 count = 100ns
- Total I2Cclk counts required to count 50 µs = 500
- Value of TIMEOUT_CNT.CNTLB for 50µs bus idle timeout = 500/16 = 32

#### *30.3.3.17.5 Cumulative Clock Low Extended Timeout*

The below two cumulative timeout event specifications come from SMBus 3.0 for targets and controllers.

- $t_{LOW:SEXT}$
  - This value is defined as the cummulative time a given target device is allowed to extend the clock cycles in one message from the initial START to the STOP. Another target device, or the controller, can also extend the clock causing the combined clock low extend time to be greater than $t_{LOW:SEXT}$. Therefore, this parameter is measured with the target device as the sole target of a full-speed controller.
- $t_{LOW:MEXT}$
  - This value is defined as the cumulative time a controller device is allowed to extend the clock cycles within each byte of a message as defined from START-to-ACK, ACK-to-ACK, or ACK-to-STOP. A target device



Copyright © 2026 Texas Instruments Incorporated

or another controller can also extend the clock causing the combined clock low time to be greater than $t_{LOW:MEXT}$ on a given byte. Therefore, this parameter is measured with a full speed target device as the sole target of the controller.

Cumulative timeout events need to be monitored in software for I2CC and I2CT. In most cases, a single clock stretch SCL timeout can act as a sufficient safeguard against cumulative events.

In controller mode, the UNICOMM-I2CC only clock stretches on byte boundaries. This aligns with the definition of $t_{LOW:MEXT}$ and the Counter A low timeout method outlined in Section 30.3.3.17.3 can be used.

In target mode, since clock stretch events can happen at different byte boundaries within a transaction for UNICOMM-I2CT, software must use DMA and FIFO capabilities to avoid exceeding $t_{LOW:SEXT}$. Typically, software takes a longer time to process the first command byte of an SMBUS frame, so the Counter A method can safeguard against software latencies. The rest of the transaction contains byte transfers that must not have software delays.

#### *30.3.3.17.6 Packet Error Checking (PEC)*

PEC is a CRC-8 error checking byte which is calculated using all the message bytes of a transaction, including addresses and read/write bits. The ACK/NACK, START, and STOP bits do not contribute to the PEC calculation. The PEC is appended to the message by the device that supplied the last data byte. PEC is cleared upon STOP condition in controller mode, and is cleared upon NACK in target mode.

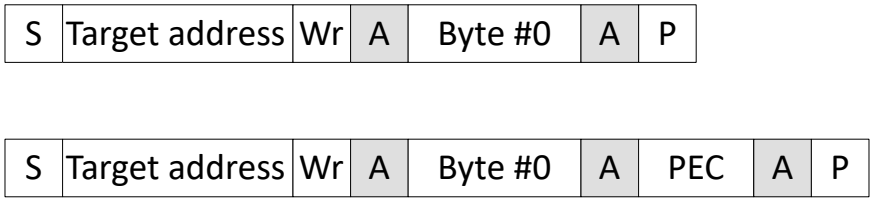



**Figure 30-15. Send Byte Message with and without PEC**

The PEC Polynomial calculation is the below.

x^8 + x^2 + x^1 + 1

The PECCTL register on Advanced UNICOMM-I2CC and UNICOMM-I2CT instances supports PEC transmission and checking for both controller and target modes. Each of the fields from the PECCTL register are described below.

- PECEN
  - This bit enables the SMB Packet Error Checking (PEC) mechanism. When enabled, the PEC is calculated on all bits except the START, STOP, and ACK/NACK. The PEC LFSR and the Byte Counter is set to 0 when the State Machine is in the IDLE state, which occurs following a STOP condition or when a timeout occurs. The Counter is also set to 0 after the PEC byte is sent or received.
- PECCNT
  - When this field is set to a non-zero value, the number of I2C data bytes are counted and compared against this value to keep track of each PEC byte in a transmission/reception.

**Note**

For a I2C Target, dynamic changes made to PECCNT must be done while the SCL is being stretched and only if the previous PECCNT= 0.

**Note**

For a I2C Controller, dynamic changes made to PECCNT must be done only if the previous PECCNT=0. A PECCNT value of 1 is an invalid configuration as this indicates one PEC byte and zero data bytes in the transaction, which is not possible per the SMBus standard.



Copyright © 2026 Texas Instruments Incorporated

For transmitting, an extra byte needs to be inserted into the transmit sequence for each transaction by writing a dummy byte to the TXDATA register. If PECEN is set, I2CC/I2CT calculates the PEC and replaces this dummy byte with the PEC value during transmission when the data byte count reaches the value in PECCNT. For reception, a PEC byte is received the same as a data byte through the RX FIFO. If PECEN is set and the data byte count is equal to the value programmed in PECCNT, the I2CC/I2CT checks the next received byte since that is the PEC. If the PEC byte is non-zero, the PEC_RX_ERR flag in RIS is set and a NACK is transmitted on the line automatically.

The PECSR register is also added to provide status flags relating to PEC operation and functionality for both UNICOMM-I2CC and UNICOMM-I2CT. Each of the fields from the PECSR register are described below.

- PECBYTECNT
  - This is the current PEC Byte Count of the I2C controller or target.
- PECSTS_CHECK
  - This status bit indicates if the PEC was checked in the transaction that occurred before the last STOP. The flag is updated on a STOP condition.
- PECSTS_ERROR
  - This status bit indicates if a PEC check error has occurred in the transaction before the last STOP condition. The flag is updated on a STOP condition.

In a typical I2C Target use case, software can set PECEN=1 and PECCNT=0 and enabled the manual ACK override feature, by setting the ACKOEN bit, until the remaining packet length is known. Software then can set the PECCNT to the remaining packet length (including the PEC byte). Once transmission/reception of the packet is complete, manual ACK override mode can be disabled by clearing ACKOEN.

#### *30.3.3.17.7 Host Notify Protocol*

To support the SMBus host notify protocol specifications, UNICOMM-I2CT Advanced instances contain a default host address. When the CTR.EN_DEFHOSTADR bit is set, the default host address of **7'b000_1000** is always used by the Target address match logic for comparison. When this bit is cleared, the default host address is not used for matching.

#### *30.3.3.17.8 Alert Response Protocol*

A Target-only device can signal the host through GPIO that it wants to talk. The host processes the interrupt and simultaneously accesses all compliant devices through the Alert Response Address. Only the device(s) which generated the request will acknowledge the Alert Response Address.

To support the SMBus host notify protocol specifications, UNICOMM-I2CT Advanced instances contain an alert response address. When the CTR.EN_ALRESPADR bit is set, the alert response address of **7'b000_1100** is always matched by the Target address match logic. When this bit is cleared, the alert response address is not matched. GPIO signaling for alarm situations must be handled in software.

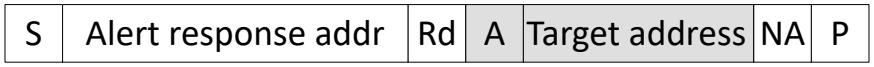



**Figure 30-16. Alert Response Message**

#### *30.3.3.17.9 Address Resolution Protocol*

SMBus target address conflicts can be resolved by dynamically assigning a new unique address to each target device. The Address Resolution Protocol (ARP) outlined below is used by the UNICOMM-I2CC in this event.

1. Prepare to ARP
   a. Controller sends a **Prepare to ARP** command to all the targets on the bus by using the default device address. All targets must ACK this command and clear own addresses.
2. Get UDID
   a. Controller sends the **Get UDID** command to all the targets on the bus. All targets return the unique UDID back to the controller.



Copyright © 2026 Texas Instruments Incorporated

   b. Target arbitration happens since all targets are trying to send a UDID to the controller.
3. Assign address
   a. Controller sends the **Assign Address** command to all the target UDID's to assign each device with a new unique address. The controller needs to keep track of the used address pool and corresponding UDID's in software during this process. All targets need to monitor the UDID bytes and change the own address to the assigned address in the event of a UDID match. For UNICOMM-I2CT instances, this can be done in software. The commander can repeat Step 2 and 3, until there are no ACK's from a target after sending a **Get UDID** command, meaning all ARP-capable targets now have valid target addresses assigned.

Additionally to support the ARP process, UNICOMM-I2CT Advanced instances contain a default device address. When the CTR.EN_DEFDEVADR bit is set, a default device address of **7'b110_0001** is always used by the target address match logic. When this bit is cleared, the default device address is not matched. If arbitration is lost, the RIS.ARBLOST interrupt is set, similar to the arbitration logic of the UNICOMM-I2CC module.

### *30.3.4 Protocol Descriptions & Initialization*

#### 30.3.4.1 I²C Controller Mode

As showing in the function block diagram section, UNICOMM-I2CC instance has a set of specific controller registers to configure the I2C operation.

##### *30.3.4.1.1 I2C Controller Initialization*

Follow the high-level UNICOMM configurations in Section 27.3 before executing the below I2CC-specific configurations.

1. Clear the ENABLE bit in the CR register before making any of the below configuration changes.
2. Set the desired SCL clock speed by writing the TPR bit in TPR register with the correct value. For example, with a 20MHz I2Cclk and 100kbps desired SCL clock, a TPR value of 19 (0x13) needs to be written to TPR. For more information about how to calculate the TPR value, refer to Clock Control.
3. Configure the desired FIFO trigger levels in the IFLS register.
4. Enable/Disable clock stretching in the CR register.
5. Specify the target address, initial direction of transfer, and addressing mode (7-bit or 10-bit ) by writing the TA register.
6. Enable desired interrupts and/or DMA event by using CPU_INT, DMA_TRIG_RX, DMA_TRIG_TX group IMASK registers.
7. (Optionally) Configure the emulation mode for the peripheral in the PDBGCTL register.
8. Enable the I2CC by setting the CR.ENABLE bit.
9. Specify the initial length of transfer, acknowledgement settings, and START/STOP conditions through the CTR register. If operating in controller-transmitter mode, TXDATA can be written with data to be transmitted. Start a frame by writing to the FRM_START bit.

##### *30.3.4.1.2 I2C Controller Status*

The application can read the SR register (detailed in *SR Field Descriptions*) to check the current state of the UNICOMM-I2CC. Status bits that are common across all UNICOMM peripherals and relate to the TX and RX FIFOs are explained in *FIFO Status Flags*.

##### *30.3.4.1.3 I2C Controller Receive Mode*

For an UNICOMM-I2CC to start receiving data out of the idle mode, the user needs to set the START bit in CTR register to generate the START condition. Then the controller automatically sends the START condition followed by the target address as soon as the bus is free.

First, set TA.DIR to 1 to enable receive mode and set CTR.START to 1 to enable the START condition. CTR.BLEN can then be programmed to indicate the number of bytes (n) for the receive operation on Advanced I2CC instances (for Basic UNICOMM-I2CC instances, this length is hardcoded to 1). The CTR.ACK and



Copyright © 2026 Texas Instruments Incorporated

CTR.STOP bit can then be set or cleared based on user configuration. CTR.FRM_START is set to begin the transaction on the line. The packet format is: START + ADDR + R + (DATA*n) +(ACK/NACK) + STOP. The last data ACK/NACK depends on the ACK bit and an additional STOP condition depends on STOP bit.

After last byte is received, the RXDONE interrupt in the CPU_INT.IIDX register is set to indicate that Controller-Receive transaction is complete. User can use the RXTRG interrupt in CPU_INT.IIDX register to read the data from the RX FIFO. This interrupt triggers when the controller RX FIFO contains >= defined bytes, the trigger level can be defined by using the RXIFSEL bit in IFLS register. The flow chart of controller receiver mode is shown in Figure 30-17.



Copyright © 2026 Texas Instruments Incorporated

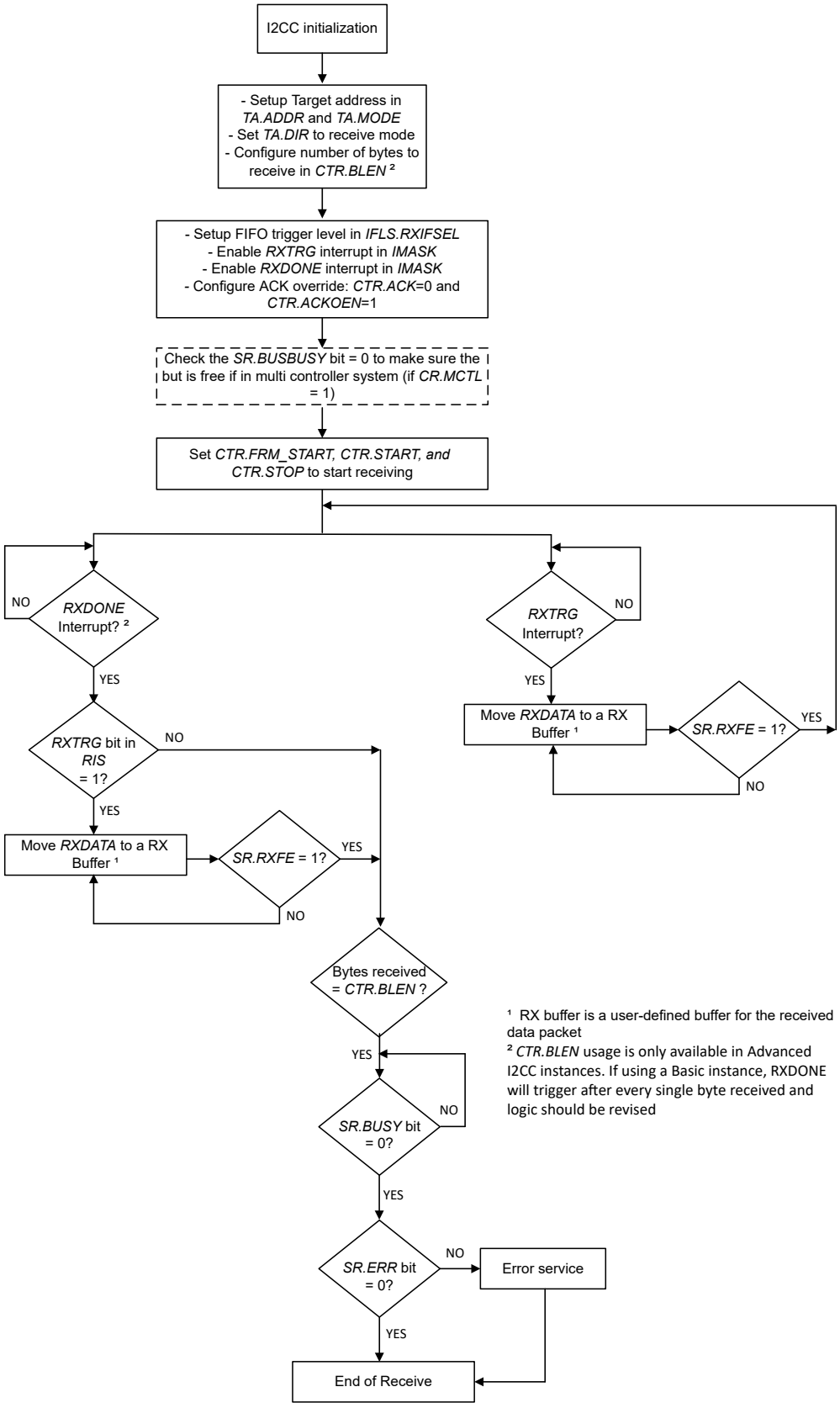



**Figure 30-17. Controller Receive Mode**



Copyright © 2026 Texas Instruments Incorporated

#### *30.3.4.1.4 I2C Controller Transmitter Mode*

For a controller to start transmitting data out of idle mode, the user needs to set the START bit in the CTR register to generate the START condition. Then the controller automatically sends the START condition followed by the target address as soon as the bus is free. The data written into TXDATA is transmitted if arbitration is not lost during transmission of the target address.

First, clear TA.DIR to 0 to enable transmit mode and set CTR.START to 1 to enable the START condition. CTR.BLEN can then be programmed to indicate the number of bytes (n) for the transmit operation on Advanced I2CC instances (for Basic instances this length is hardcoded to 1). The CTR.STOP bit can then be set or cleared based on user configuration. CTR.FRM_START is then set to begin the transaction on the line. The packet format is START + ADDR + W + (DATA*n) + STOP, where transmission of the STOP condition depends on the STOP bit.

After the last byte is transmitted, the TXDONE interrupt in the CPU_INT.IIDX register is set to indicate that the controller transmit transaction is completed. The user can also use the TXEMPTY interrupt from the CPU_INT.IIDX register to see if the TX FIFO is empty and ready to load more data. This interrupt triggers if all data in the transmit FIFO has been shifted out.

The flow chart with the nexessary software steps of using TXDONE and TXTRG interrupt for controller-transmitter mode are shown in Figure 30-18 and Figure 30-19 respectively.



Copyright © 2026 Texas Instruments Incorporated

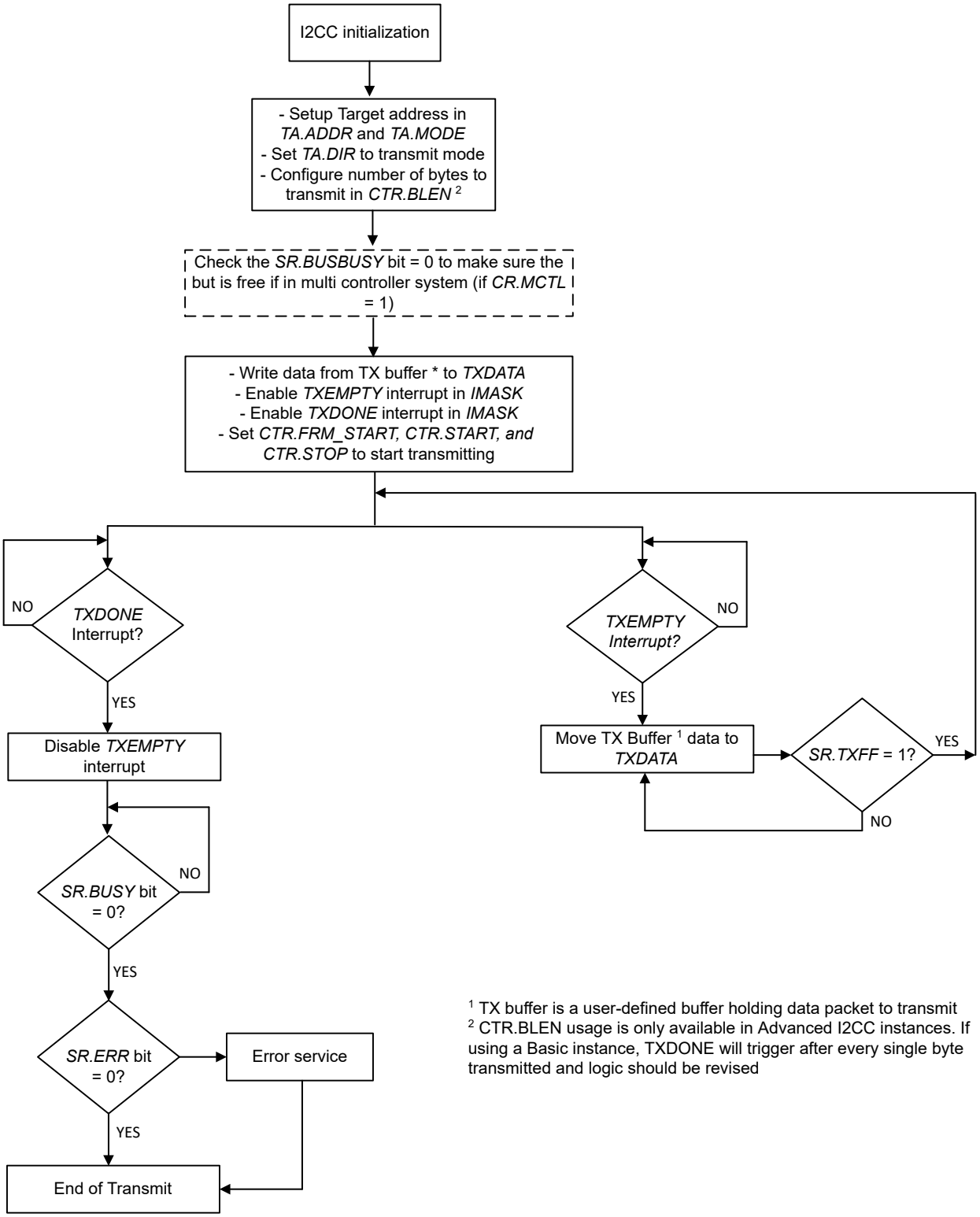



**Figure 30-18. Controller Transmitter Mode using TXDONE**



Copyright © 2026 Texas Instruments Incorporated

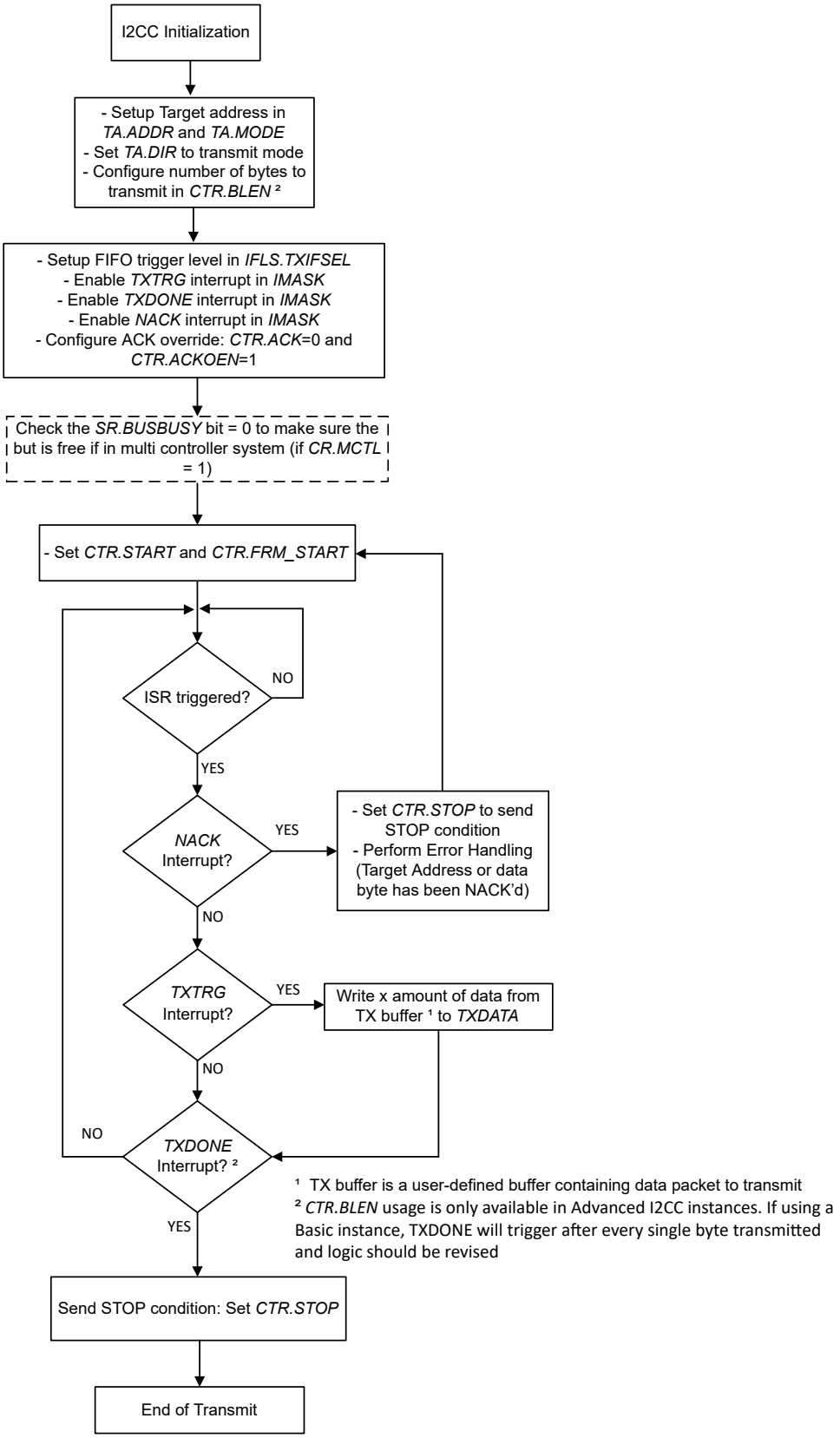



**Figure 30-19. Controller Transmitter Mode using TXTRG**



Copyright © 2026 Texas Instruments Incorporated

#### *30.3.4.1.5 Controller Transaction Configurations*

The UNICOMM-I2CC control register (CTR) and UNICOMM-I2CC target address register (TA) are used during the application run to setup controller transmit and receive transactions. The following settings can be used to control each transaction before software reads/writes TXDATA and RXDATA.

- Length indicates the number of bytes for the transaction and is configured by CTR.BLEN bits (Only available in Advanced UNICOMM I2CC instances. In Basic I2CC instances, transaction length is fixed at one.)
- Direction (transmit or receive) is configured by TA.DIR bit
- ACK generation is configured by CTR.ACK and CTR.ACKOEN bit.
- STOP condition generation is configured by CTR.STOP bit.
- START or Repeated START condition generation is configured by the CTR.START bit.
- Start of transmission is enabled by the CTR.FRM_START bit.
- Transmit/Receive status and Error status can be checked with the SR.BUSY and SR.ERR bits respectively.

**Controller Transmit Data Transactions**

**Table 30-7. Start Transmit From Idle Mode**

| Length | Direction | ACK | STOP | START | FRM_START | Frame Format | Notes |
|---|---|---|---|---|---|---|---|
| n (n>0) | 0 | X | 0 or 1 | 1 | 1 | START+ADDR+R/W+ DATA*n+STOP | Sending of STOP conndition depends on the CTR.STOP bit. |

**Table 30-8. Continue Transmit when Last Transmission Finished without STOP**

| Length | Direction | ACK | STOP | START | FRM_START | Frame Format | Notes |
|---|---|---|---|---|---|---|---|
| n (n>0) | 0 | X | 0 or 1 | 0 | 1 | DATA*n+ ACK/ NACK+STOP | Sending of STOP conndition depends on the CTR.STOP bit. |

If there is a NACK response from the target, the controller automatically sends out a STOP condition to finish the transmit. The Controller is unable to send a RESTART after a ADDR or DATA NACK.

**Controller Receive Data Transactions**

**Table 30-9. Start Receive from Idle Mode**

| Length | Direction | ACK | STOP | START | FRM_START | Frame Format | Notes |
|---|---|---|---|---|---|---|---|
| n (n>0) | 1 | 0 or 1 | 0 or 1 | 1 | 1 | START+ADDR +R/ W+DATA *n +ACK/NACK +STOP | The last data ACK or NACK depends on the CTR.ACK bit; additional sending of STOP condition depends on CTR.STOP bit. |

**Table 30-10. Continue Receive when Last Receive Finished without STOP or NACK**

| Length | Direction | ACK | STOP | START | FRM_START | Frame Format | Notes |
|---|---|---|---|---|---|---|---|
| n (n>0) | 1 | 0 or 1 | 0 or 1 | 0 | 1 | DATA*n+ ACK/NACK +STOP | The last data ACK or NACK depends on the CTR.ACK bit; additional sending of STOP condition depends on CTR.STOP bit. |

This configuration is not allowed if last transaction ended with NACK, since NACK can only be followed by a STOP or RESTART condition. The ACK and STOP bits must not be set to 1 at the same time, as the target needs to be informed to release bus lines before sending out STOP.



Copyright © 2026 Texas Instruments Incorporated

### Controller Repeated START Transactions

If the last transmit or receive finished without a STOP, a Repeated START can be generated to initiate a new transaction

**Table 30-11. Repeated Start Transmit**

| Length | Direction | ACK | STOP | START | FRM_START | Frame Format | Notes |
|---|---|---|---|---|---|---|---|
| n (n>0) | 0 | 0 or 1 | 0 or 1 | 1 | 1 | Restart+ADDR +R/ W+DATA*n +STOP | Additional sending of STOP depends on CTR.STOP bit. |

If there is a NACK response from the target, the controller automatically sends out a STOP condition to finish the transmit. The Controller is unable to send a RESTART after an ADDR or DATA NACK.

**Table 30-12. Repeated Start Receive**

| Length | Direction | ACK | STOP | START | FRM_START | Frame Format | Notes |
|---|---|---|---|---|---|---|---|
| n (n>0) | 1 | 0 or 1 | 0 or 1 | 1 | 1 | Restart+ADDR +R/ W+DATA*n +ACK/ NACK+STOP | The last data followed by ACK or NACK depend on CTR.ACK bit; additional sending of STOP depends on CTR.STOP bit. |

The ACK and STOP bits must not be set to 1 at the same time, as the target needs to be informed to release bus lines before sending out STOP.

### Controller Transmit STOP-only Transaction

**Table 30-13. Send STOP only**

| Length | Direction | ACK | STOP | START | FRM_START | Frame Format | Notes |
|---|---|---|---|---|---|---|---|
| n (n>=0) | X | X | 1 | 0 | 1 | STOP | STOP command |

Only send this transaction after the previous transaction successfully finishes, The STOP condition can't be sent without a NACK to the target if the controller is currently in receive mode.

### Controller Quick Command Transaction

The Quick command can only be sent at the beginning of a transaction, not following other transactions (without a STOP) or repeatted start.

**Table 30-14. Controller Quick Command**

| Length | Direction | ACK | STOP | START | FRM_START | Frame Format | Notes |
|---|---|---|---|---|---|---|---|
| 0 | 0/1 | X | 1 | 1 | 1 | START+ADDR +R/ W+STOP | Quick command |

#### 30.3.4.2 I²C Target Mode

As showing in the function block diagram section, UNICOMM-I2CT instance has a set of specific controller registers to configure the I2C operation.

##### *30.3.4.2.1 I2C Target Initialization*

Follow the high-level UNICOMM configurations in Section 27.3 before executing the below I2CC-specific configurations.

1. Disable the I2CT by clearing the CTR.ENABLE bit.



Copyright © 2026 Texas Instruments Incorporated

2. Set the desired SCL clock speed by writing the TPR bit in TPR register with the correct value. For example, with a 20MHz I2C_CLK and 100kbps desired SCL clock, a TPR value of 19 (0x13) needs to be written to TPR. For more information about how to calculate the TPR value, refer to Clock Control.
3. Configure at least one target address by writing the OAR register. An additional target address can be enabled and configured for Advanced I2CT UNICOMM instances using the OAR2 register.
4. Configure acknowledgement settings in the ACKCTL register.
5. Configure clock stretching, TXTRG/RXTRG interrupt modes, default device addresses, and SMBUS feature settings (only available on Advanced I2CT instances) in the CTR register.
6. Configure the desired FIFO trigger levels in the IFLS register.
7. Enable desired interrupts and/or DMA event by using CPU_INT, DMA_TRIG_RX, DMA_TRIG_TX group IMASK registers.
8. Enable the I2CT by setting the CTR.ENABLE bit.

#### *30.3.4.2.2 I2C Target Status*

The application can read the SR register (detailed in *SR Field Descriptions*) to check the current state of the UNICOMM-I2CT. Status bits that are common across all UNICOMM peripherals and relate to the TX and RX FIFOs are explained in *FIFO Status Flags*.

#### *30.3.4.2.3 I2C Target Receiver Mode*

Target receiver mode is entered when the target address transmitted by the controller matches the I2CT's own address and a **low** R/W bit is received. In target receiver mode, serial data bits received on SDA are shifted in with the clock pulses that are generated by the controller device. The target device does not generate the clock, but can hold SCL low if intervention of the CPU is required after a byte has been received.

After the a data byte is received, the RXDONE interrupt in CPU_INT.IIDX register is set. The $I^2C$ module automatically acknowledges the received data or the user can manually send an acknowledge after each byte received by configuring the ACKCTL register.

When the controller generates a START condition, the START interrupt flag from the CPU_INT.IIDX register is set. When the controller generates a STOP condition, the STOP interrupt flag from the CPU_INT.IIDX register is set.

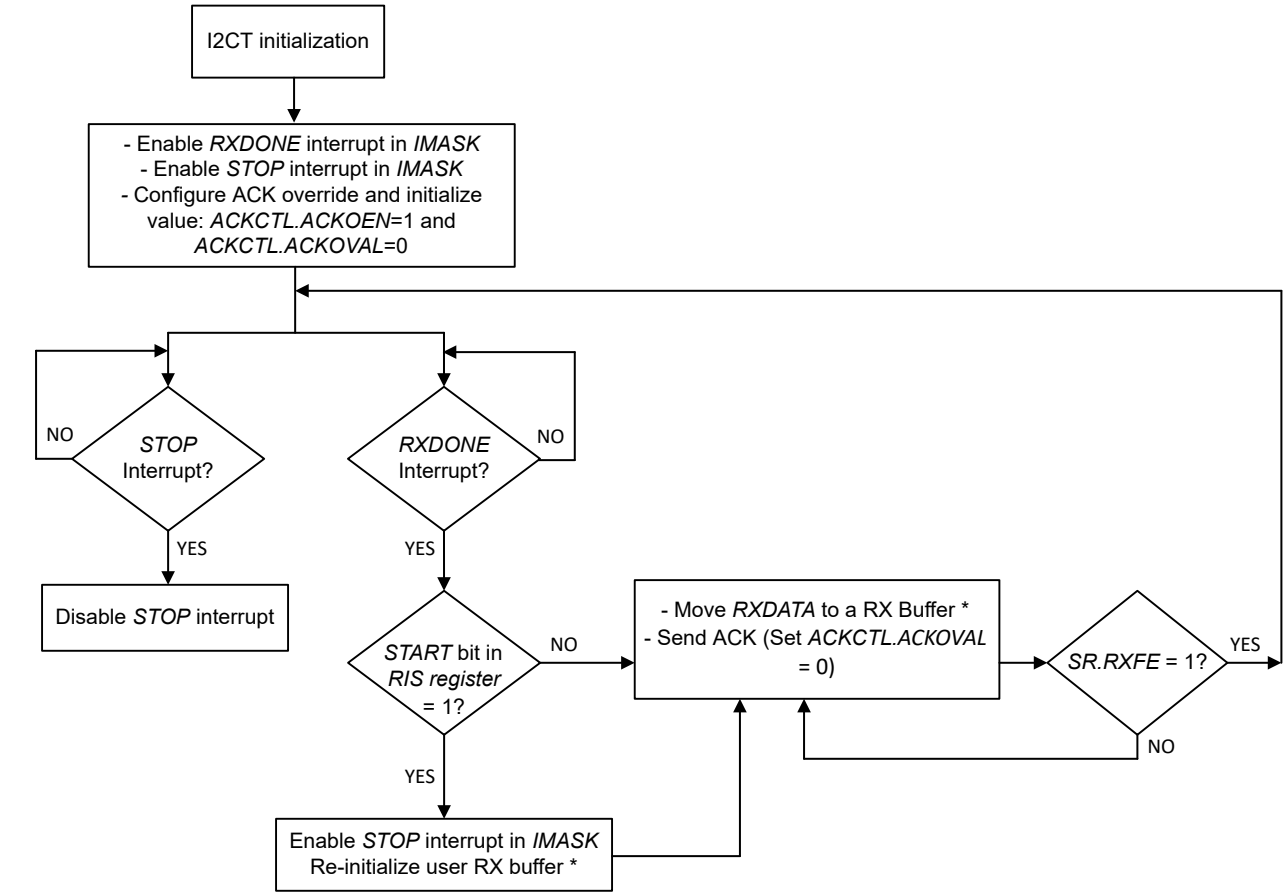
The user can also use the RXTRG interrupt in CPU_INT.IIDX register to read the data from the receive FIFO. This interrupt triggers when the receive FIFO contains >= defined bytes. The FIFO trigger level can be defined in the RXIFSEL bit in the IFLS register.

The RXDONE approach can be used if the target wants to slow down communication to evaluate reception of every byte, while the RXTRG approach is used to maximize throughput and avoid clock stretching.

The flow chart of using RXDONE and RXTRG interrupt to read the receive data are shown in Figure 30-20 and Figure 30-21 respectively.



Copyright © 2026 Texas Instruments Incorporated




* RX buffer is a user defined buffer for the received data packet

**Figure 30-20. Target Receiver Mode using RXDONE and ACK Override**



Copyright © 2026 Texas Instruments Incorporated

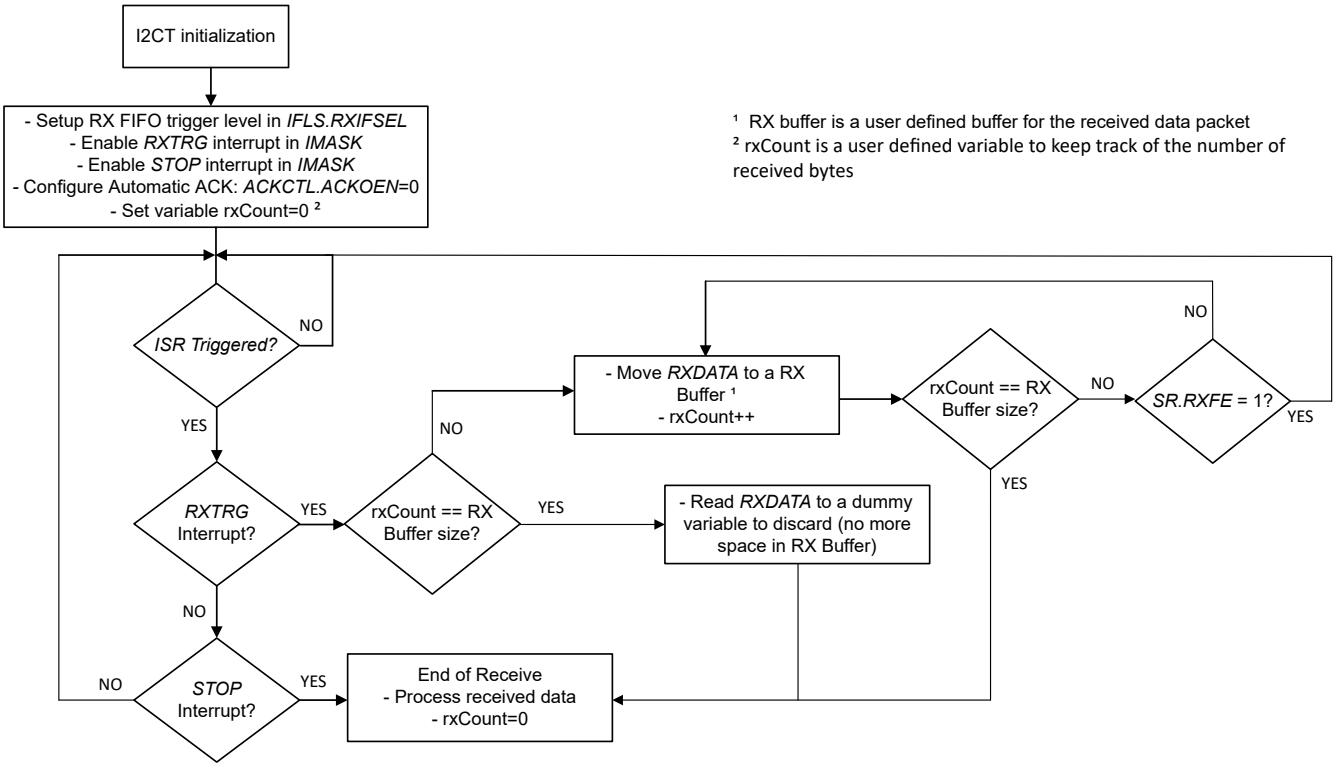



**Figure 30-21. Target Receiver Mode using RXTRG and Automatic ACK**

#### *30.3.4.2.4 I2C Target Transmitter Mode*

Target transmitter mode is entered when the target address transmitted by the controller matches the I2CT's own address with a **high** R/W bit. The target transmitter shifts the serial data out on SDA with the clock pulses that are generated by the controller device. The target device does not generate the clock, but can hold SCL low if intervention of the CPU is required after a byte has been transmitted.

After a data byte is transmitted, the TXDONE interrupt in CPU_INT.IIDX register is set to indicate that a byte has been transmitted.

When the controller generates a START condition, the START interrupt in CPU_INT.IIDX register is set. When the controller generates a STOP condition, the STOP interrupt in CPU_INT.IIDX register is set.

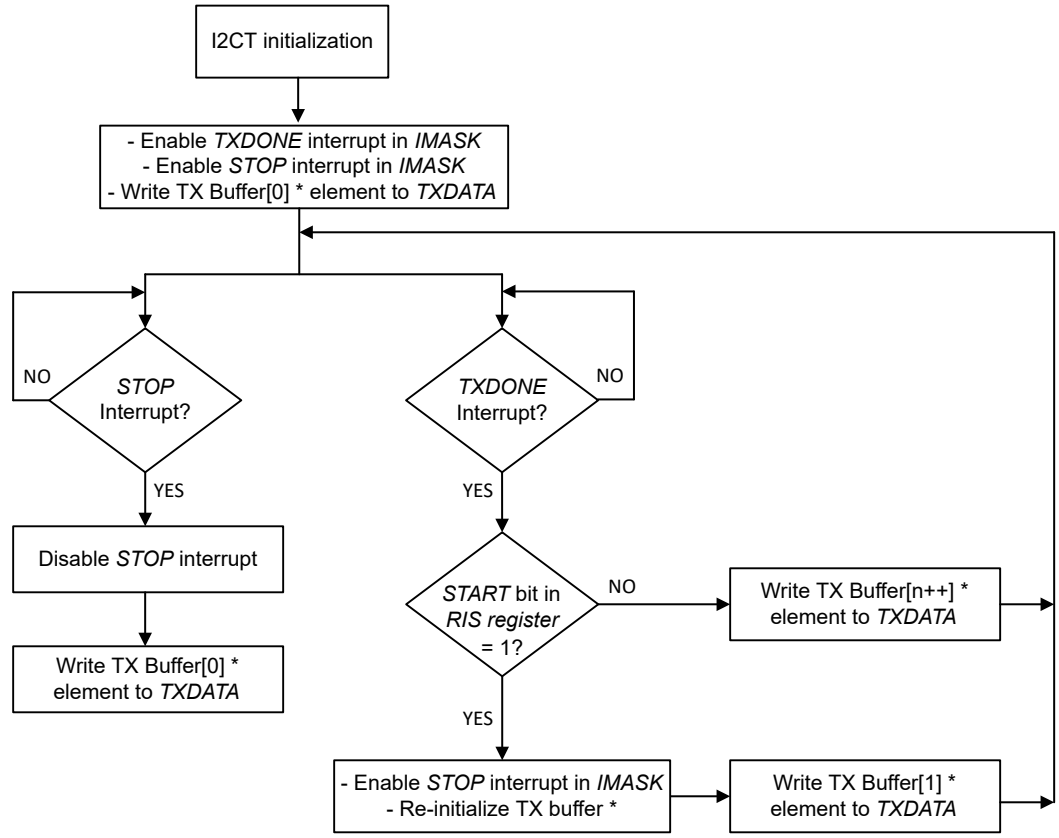
The user can also use the TXTRG interrupt from the CPU_INT.IIDX register to load data to the transmit FIFO. This interrupt triggers when the transmit FIFO contains <= defined bytes. The FIFO trigger level can be defined by using TXIFSEL field in the IFLS register.

The TXDONE approach is used if the target wants to slow down communication to evaluate the transmission of every byte, while the TXTRG approach is used to maximize throughput and avoid clock stretching.

The flow chat of using TXDONE and TXFIFOTRG interrupt to transmit data are shown in Figure 30-22 and Figure 30-23.



Copyright © 2026 Texas Instruments Incorporated




* TX buffer is a user-defined buffer holding data packet to transmit

**Figure 30-22. Target Transmitter Mode using TXDONE**



Copyright © 2026 Texas Instruments Incorporated

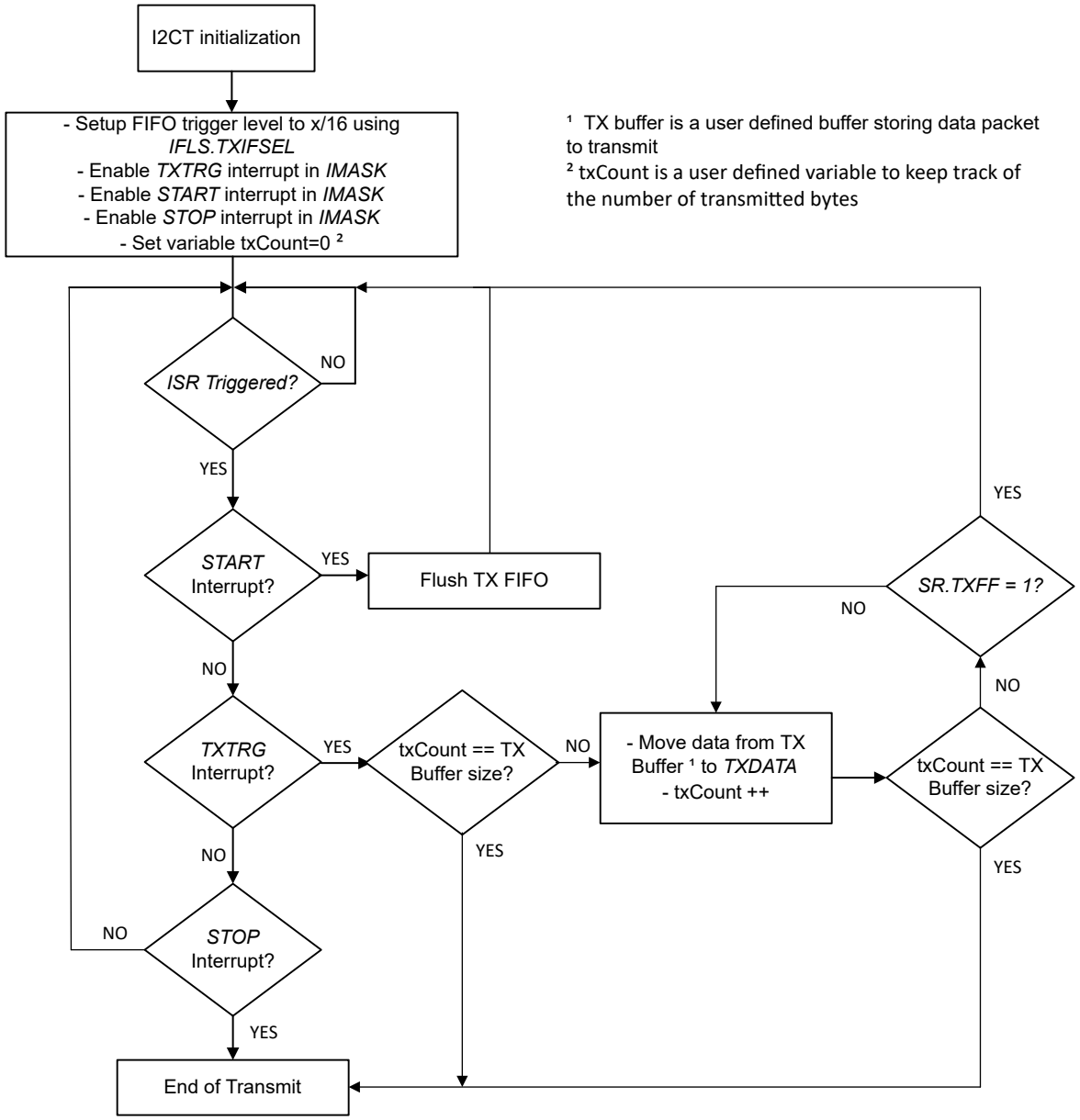



**Figure 30-23. Target Transmitter Mode using TXFIFOTRG**

### *30.3.5 Reset Considerations*

#### Software Reset Considerations

A Software reset can be executed in UNICOMM module by setting the RESETASSERT bit together with the KEY bit in the RSTCTL register (UNICOMM Register). Resets must only be issued after terminating a transaction. For more details, refer to Section 27.2.2.

#### Hardware Reset Considerations

A hardware reset also re-initializes the IO configuration. This sets the IOs to a high-impedance state and with the external pullup resistors for $I^2C$, the lines are pulled high.

Table 30-15 and Table 30-16 show the behavior of status bits when Controller or Target gets disabled.



Copyright © 2026 Texas Instruments Incorporated

**Table 30-15. Status Bits when Controller is Disabled**

| Register | Bit | Behavior when Controller is Disabled | Behavior when Target Disabled | Behavior after Controller/Target is Enabled |
|---|---|---|---|---|
| SR | BUSY | Reset State | Don't care | Updates START condition sending |
| | ERR | Reset State | Don't care | Updates on next event detected |
| | ADRACK | Reset State | Don't care | Updates on next event detected |
| | DATACK | Reset State | Don't care | Updates on next event detected |
| | ARBLST | Reset State | Don't care | Updates on next event detected |
| | IDLE | Reset State | Don't care | Updates on next event detected |
| | BUSBSY | Reset State | Don't care | Updates on next START detected on bus (or SDA or SCL is low) |
| | BCNT | Reset State | Don't care | Updates on next event detected |
| BMON | SCL | Reset State | Don't care | Updates with Controller enable |
| | SDA | Reset State | Don't care | Updates with Controller enable |

**Table 30-16. Status Bits when Target is Disabled**

| Register | Bit | Behavior When Controller Disabled | Behavior When Target Disabled | Behavior After Controller/Target Enabled |
|---|---|---|---|---|
| SR | RREQ | Don't care | Reset State | Updates on next event detected |
| | TREQ | Don't care | Reset State | Updates on next event detected |
| | OAR2SEL | Don't care | Reset State | Updates on next event detected |
| | QCMDST | Don't care | Reset State | Updates on next event detected |
| | QCMDRW | Don't care | Reset State | Updates on next event detected |

### *30.3.6 Initialization*

Please see Section 30.3.4.1.1 for controller initialization and Section 30.3.4.2.1 for target initialization.

### *30.3.7 Interrupt and Events Support*

The $I^2C$ module contains three event publishers and no event subscribers. One event publisher (CPU_INT) manages $I^2C$ interrupt requests (IRQs) to the CPU subsystem through a static event route. The second and third event publishers (DMA_TRIG_RX, DMA_TRIG_TX) are used to setup the trigger signaling for the DMA through DMA event route.

The $I^2C$ events are summarized in Table 30-17.

**Table 30-17. I2C Events**

| Event | Type | Source | Destination | Route | Configuration | Functionality |
|---|---|---|---|---|---|---|
| CPU interrupt | Publisher | $I^2C$ | CPU Subsystem | Static route | CPU_INT registers | Fixed interrupt route from $I^2C$ to CPU |
| DMA trigger | Publisher | $I^2C$ | DMA | DMA event route | DMA_TRIG_RX registers | Fixed interrupt route from $I^2C$ to DMA |
| DMA trigger | Publisher | $I^2C$ | DMA | DMA event route | DMA_TRIG_TX registers | Fixed interrupt route from $I^2C$ to DMA |



Copyright © 2026 Texas Instruments Incorporated

#### 30.3.7.1 CPU Interrupt Event Publisher (CPU_INT)

The I²C module provides 15 interrupt sources for I2C controllers and 17 interrupt sources for I2C targets which can be configured to source a CPU interrupt event. In order of decreasing interrupt priority, the CPU interrupt events from the I²C are:

**Table 30-18. I²C CPU Interrupt Event Conditions for Controller (CPU_INT)**

| IIDX STAT | Name | Description |
|---|---|---|
| 0x01 | RXDONE | Controller receive transaction completed |
| 0x02 | TXDONE | Controller transmit transaction completed |
| 0x03 | RXTRG | Controller receive trigger, occurs when receive buffer has data |
| 0x04 | TXTRG | Controller transmit trigger, occurs when transmit buffer is empty |
| 0x05 | RXFULL | Controller RXFIFO full event. This interrupt is set if an RX FIFO is full. |
| 0x06 | TXEMPTY | Controller transmit FIFO empty interrupt. This interrupt is set if all data in the Transmit FIFO have been shifted out and the transmit goes into idle mode. |
| 0x08 | NACK | Controller Address/Data NACK interrupt |
| 0x09 | START | Controller START detection interrupt |
| 0x0A | STOP | Controller STOP detection interrupt |
| 0x0B | ARBLOST | Controller arbitration lost interrupt |
| 0x0C | PEC_RX_ERR | Controller PEC error occurred |
| 0x0D | TIMEOUTA | Controller Timeout A occurred (clock low timeout) |
| 0x0F | TIMEOUTB | Controller Timeout B occurred (clock high timeout) |
| 0x10 | DMA_DONE_RX | Controller DMA TX done |
| 0x11 | DMA_DONE_TX | Controller DMA RX done |

**Table 30-19. I²C CPU Interrupt Event Conditions for Target (CPU_INT)**

| IIDX STAT | Name | Description |
|---|---|---|
| 0x01 | RXDONE | Target Receive Data Interrupt |
| 0x02 | TXDONE | Target Transmit Transaction completed Interrupt |
| 0x03 | RXTRG | Target Receive Trigger |
| 0x04 | TXTRG | Target Transmit Trigger |
| 0x05 | RXFULL | Target RXFIFO full event. This interrupt is set if an Target RX FIFO is full. |
| 0x06 | TXEMPTY | Target Transmit FIFO Empty interrupt mask. This interrupt is set if all data in the Transmit FIFO have been shifted out and the transmit goes into idle mode. |
| 0x07 | TX_UNFL | Target TX FIFO underflow |
| 0x08 | RX_OVFL | Target RX FIFO overflow |
| 0x09 | GENCALL | Target General Call Interrupt |
| 0x0A | START | Target Start Condition Interrupt |
| 0x0B | STOP | Target Stop Condition Interrupt |
| 0x0C | PEC_RX_ERR | Target RX Pec Error Interrupt |
| 0x0D | TIMEOUTA | Target Timeout A Interrupt |
| 0x0E | TIMEOUTB | Target Timeout B Interrupt |
| 0x10 | DMA_DONE_RX | Target DMA Done on Event Channel RX |
| 0x11 | DMA_DONE_TX | Target DMA Done on Event Channel TX |
| 0x12 | ARBLOST | Target Arbitration Lost Interrupt |

The CPU interrupt event configuration is managed with the CPU_INT event management registers. See *Event Registers* for guidance on configuring the Event registers for CPU interrupts.



Copyright © 2026 Texas Instruments Incorporated

#### 30.3.7.2 DMA Trigger Publisher (DMA_TRIG_RX, DMA_TRIG_TX)

The DMA_TRIG_TX and DMA_TRIG_RX group registers are used to setup trigger signaling for the DMA. These registers (IMASK, RIS, MIS, ISET, ICLR) are present for any UCx configured as an I2C Commander or an I2C Target and can be found in the corresponding UNICOMMI2CC_REGS and UNICOMMI2CT_REGS register structures. See *DMA Trigger* for how the DMA trigger Event works and can be configured. Each DMA channel can be triggered by any of the conditions listed in the *DMA Trigger Source Options* and can send a PRE_IRQ or DMA_DONE signal to the corresponding UCx module.

**Table 30-20. I²C DMA Trigger Condition (DMA_TRIG_RX and DMA_TRIG_TX)**

| Trigger Name | Register Structure | Register Group | Description |
|---|---|---|---|
| RXTRG | UNICOMMI2CC_REGS | DMA_TRIG_RX | Controller receive FIFO event. Trigger when I2CC-configured UCx module's Receive FIFO contains >= IFLS.RXIFSEL defined bytes |
| TXTRG | UNICOMMI2CC_REGS | DMA_TRIG_TX | Controller transmit FIFO event. Trigger when I2CC-configured UCx module's Transmit FIFO contains <= IFLS.TXIFSEL defined bytes |
| RXTRG | UNICOMMI2CT_REGS | DMA_TRIG_RX | Target receive FIFO event. Trigger when I2CT-configured UCx module's Receive FIFO contains >= IFLS.RXIFSEL defined bytes |
| TXTRG | UNICOMMI2CT_REGS | DMA_TRIG_TX | Target transmit FIFO event. Trigger when I2CT-configured UCx module's Transmit FIFO contains <= IFLS.TXIFSEL defined bytes |

The DMA trigger event configuration is managed with the DMA_TRIG_RX and DMA_TRIG_TX event management registers. See Section 7.2.5 for guidance on configuring the Event registers and Section 7.1.3.2 for on how DMA trigger event works. DMA_TRIG_RX and DMA_TRIG_TX are two event management registers that correspond to two DMA channels.

Take an example where UC0 is configured with the I2CC IPMODE and UC1 is configured with the I2CT IPMODE. To use different DMA channels for each of the operations (controller-transmitter, controller-receiver, target-transmitter, target-receiver) four channels are set up as shown in Figure 30-24. For both UC modules, RXTRG is set as the trigger source for DMA_TRIG_RX and TXTRG is set as the trigger source for DMA_TRIG_TX. The DMA_DONE signal is asserted to trigger a CPU interrupt when the DMA operation completes after transferring the preset data length.



Copyright © 2026 Texas Instruments Incorporated

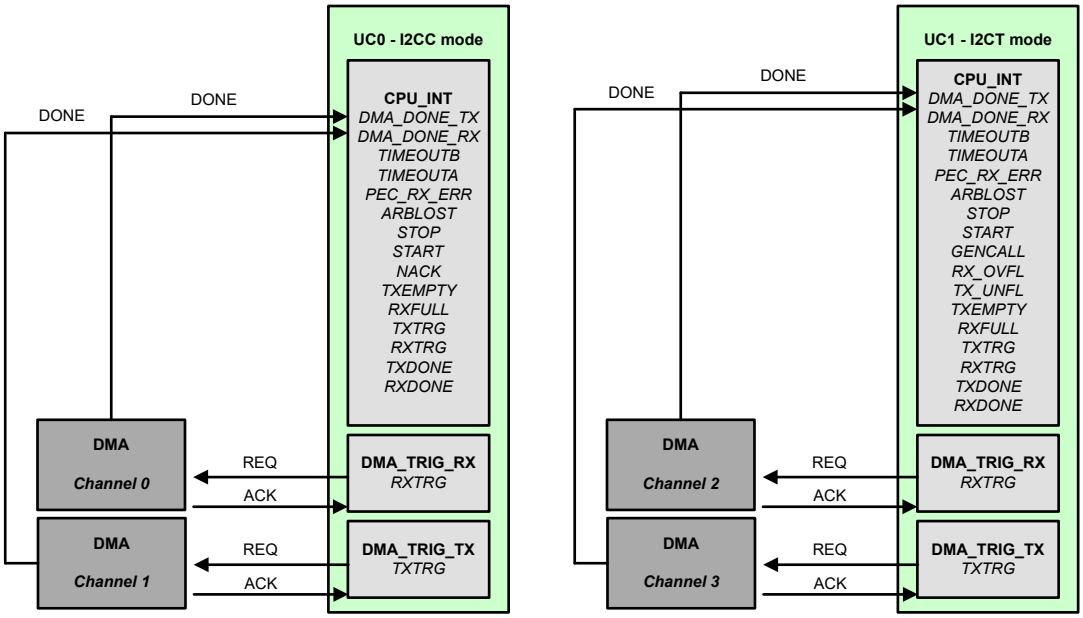



**Figure 30-24. I²C DMA Trigger and Status Example**

### *30.3.8 Emulation Modes*

The UNICOMM I2C module behavior while the device is in debug mode is controlled by the FREE and SOFT bits in PDBGCTL register.

When the device is in debug mode and set into halt mode below behavior can be configured.

**Table 30-21. Debug Mode Peripheral Behavior**

| PDBGCTL.FREE | PDBGCTL.SOFT | Function |
|---|---|---|
| **1** | **x** | Modules continues operation |
| **0** | **1** | Graceful halt<br>Module stops after the next transfer has been finished |
| **0** | **0** | Not supported |

#### Graceful Halt

When emulation mode is set to graceful halt and a halt condition occurs, the I2C instance stops communications after the ongoing frame has finished, shown in Figure 30-25.



Copyright © 2026 Texas Instruments Incorporated

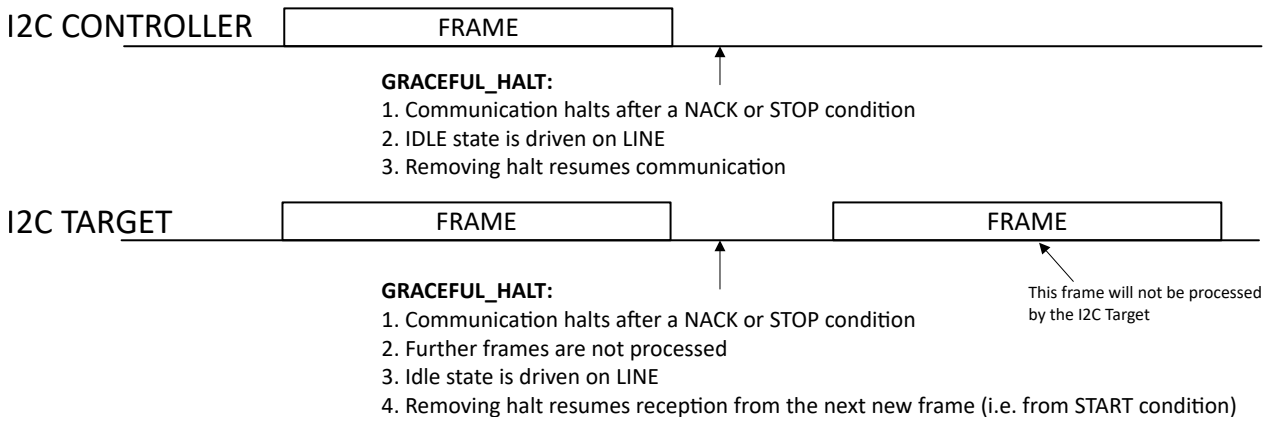



**Figure 30-25. I2C Graceful Halt Emulation Behavior**

Transmitter

- I2C finishes transmitting the ongoing frame and FSM returns to IDLE and stop.
- External lines are driven to IDLE condition.
- Removing halt resumes communication

Receiver

- I2C receives ongoing frame and FSM returns to IDLE and stop
- Any new messages are be processed
- After halt is removed, FSM begins communications
- FSM can start reception in the middle of a frame

## 30.4 UNICOMM I2C Registers

This Section describes the UNICOMM-I2C Registers.



Copyright © 2026 Texas Instruments Incorporated

### 30.4.1 UNICOMMI2CC Registers

Table 30-22 lists the memory-mapped registers for the UNICOMMI2CC registers. All register offset addresses not listed in Table 30-22 should be considered as reserved locations and the register contents should not be modified.

**Table 30-22. UNICOMMI2CC Registers**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 0h | CLKDIV | Clock Divider | | Go |
| 8h | CLKSEL | Clock Source Selection/Enable. | | Go |
| 18h | PDBGCTL | Peripheral Debug Control | | Go |
| 20h | IIDX | Interrupt index | CPU_INT | Go |
| 28h | IMASK | Interrupt mask | CPU_INT | Go |
| 30h | RIS | Raw interrupt status | CPU_INT | Go |
| 38h | MIS | Masked interrupt status | CPU_INT | Go |
| 40h | ISET | Interrupt set | CPU_INT | Go |
| 48h | ICLR | Interrupt clear | CPU_INT | Go |
| 58h | IMASK | Interrupt mask | DMA_TRIG_RX | Go |
| 60h | RIS | Raw interrupt status | DMA_TRIG_RX | Go |
| 68h | MIS | Masked interrupt status | DMA_TRIG_RX | Go |
| 70h | ISET | Interrupt set | DMA_TRIG_RX | Go |
| 88h | IMASK | Interrupt mask | DMA_TRIG_TX | Go |
| 90h | RIS | Raw interrupt status | DMA_TRIG_TX | Go |
| 98h | MIS | Masked interrupt status | DMA_TRIG_TX | Go |
| A0h | ISET | Interrupt set | DMA_TRIG_TX | Go |
| E4h | INTCTL | Interrupt control register | | Go |
| 100h | CTR | Control Register | | Go |
| 104h | CR | Configuration | | Go |
| 108h | SR | Status Register | | Go |
| 10Ch | IFLS | Interrupt FIFO Level Select Register | | Go |
| 110h | TPR | Timer Period | | Go |
| 118h | GFCTL | I2C Glitch Filter Control | | Go |
| 11Ch | BMON | Bus Monitor | | Go |
| 120h | TXDATA | TXData | | Go |
| 124h | RXDATA | RXData | | Go |
| 128h | PECSR | PEC status register | | Go |
| 14Ch | TA | Target Address Register | | Go |
| 150h | TIMEOUT_CNT | I2C Timeout Count Register | | Go |
| 154h | TIMEOUT_CTL | I2C Timeout Count Control Register | | Go |
| 158h | PECCTL | I2C PEC control register | | Go |

Complex bit access types are encoded to fit into small table cells. Table 30-23 shows the codes that are used for access types in this section.



Copyright © 2026 Texas Instruments Incorporated

**Table 30-23. UNICOMMI2CC Access Type Codes**

| Access Type | Code | Description |
|---|---|---|
| Read Type | | |
| R | R | Read |
| Write Type | | |
| W | W | Write |
| Reset or Default Value | | |
| *-n* | | Value after reset or the default value |



Copyright © 2026 Texas Instruments Incorporated

#### 30.4.1.1 CLKDIV (Offset = 0h) [Reset = 00000000h]

CLKDIV is shown in Figure 30-26 and described in Table 30-24.

Return to the Summary Table.

This register is used to specify the module-specific divide ratio of the I2CC functional clock (I2Cclk).

**Figure 30-26. CLKDIV**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | | RATIO | | |
| R-0h | | | | | | | | | | | | | R/W-0h | | |

**Table 30-24. CLKDIV Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | RESERVED | R | 0h | |
| 2-0 | RATIO | R/W | 0h | Selects divide ratio of module clock Division factor 0 : DIV_BY_1 1 : DIV_BY_2 .... 63: DIV_BY_64<br>0h = Do not divide clock source<br>1h = Divide clock source by 2<br>2h = Divide clock source by 3<br>3h = Divide clock source by 4<br>4h = Divide clock source by 5<br>5h = Divide clock source by 6<br>6h = Divide clock source by 7<br>7h = Divide clock source by 8 |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.1.2 CLKSEL (Offset = 8h) [Reset = 00000000h]

CLKSEL is shown in Figure 30-27 and described in Table 30-25.

Return to the Summary Table.

I2CC Clock Source Selection.

**Figure 30-27. CLKSEL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | MCLKDIV2 | RESERVED | | |
| R-0h | | | | R/W-0h | R-0h | | |

**Table 30-25. CLKSEL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | MCLKDIV2 | R/W | 0h | Enables MCLKDIV2 as the UNICOMM I2CC clock source.<br>0h = Does not select this clock as a source<br>1h = Select this clock as a source |
| 2-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.1.3 PDBGCTL (Offset = 18h) [Reset = 00000000h]

PDBGCTL is shown in Figure 30-28 and described in Table 30-26.

Return to the Summary Table.

I2CC Debug Control. This register can be used by the software developer to control the behavior of the peripheral relative to the 'Core Halted' input.

**Figure 30-28. PDBGCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | SOFT | FREE |
| R-0h | | | | | | R/W-0h | R/W-0h |

**Table 30-26. PDBGCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | |
| 1 | SOFT | R/W | 1h | Soft Halt Boundary Control. This function is only available, if [FREE] is set to 'STOP'<br>0h = Not supported<br>1h = The peripheral blocks the debug freeze until it has reached a boundary where it can resume without corruption |
| 0 | FREE | R/W | 1h | Free Run Control.<br>0h = The peripheral freezes functionality while the Core Halted input is asserted and resumes when it is deasserted.<br>1h = The peripheral ignores the state of the Core Halted input |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.1.4 IIDX (Offset = 20h) [Reset = 00000000h] (CPU_INT)

IIDX is shown in Figure 30-29 and described in Table 30-27.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index.

**Figure 30-29. IIDX**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | STAT | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | |

**Table 30-27. IIDX Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | STAT | R | 0h | I2CC Interrupt Vector Value. This field provides the highest priority interrupt index. A read clears the corresponding interrupt flag in RIS and MIS. 15h-1Fh = Reserved.<br>00h = No Interrupt Pending.<br>01h = Receive Done Flag.<br>02h = Transmit Done Flag.<br>03h = Receive FIFO Event.<br>04h = Transmit FIFO Event.<br>5h = RX FIFO Full Event/Interrupt Pending.<br>6h = TX FIFO Empty Event/Interrupt Pending.<br>08h = Address/Data NACK.<br>09h = START Event.<br>0Ah = STOP Event.<br>0Bh = Arbitration Lost.<br>Ch = PEC Receive Error Event. This event is only available on Advanced I2CC instances.<br>Dh = Timeout Counter A Event.<br>Eh = Timeout Counter B Event.<br>10h = DMA DONE on Channel RX.<br>11h = DMA DONE on Channel TX. |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.1.5 IMASK (Offset = 28h) [Reset = 00000000h] (CPU_INT)

IMASK is shown in Figure 30-30 and described in Table 30-28.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

**Figure 30-30. IMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | DMA_DONE_TX |
| R-0h | | | | | | | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DMA_DONE_RX | RESERVED | TIMEOUTB | TIMEOUTA | PEC_RX_ERR | ARBLOST | STOP | START |
| R/W-0h | R-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| NACK | RESERVED | TXEMPTY | RXFULL | TXTRG | RXTRG | TXDONE | RXDONE |
| R/W-0h | R-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

**Table 30-28. IMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-17 | RESERVED | R | 0h | |
| 16 | DMA_DONE_TX | R/W | 0h | I2CC DMA Done on TX Event Channel Interrupt Mask. This interrupt is raised when the DONE signal is sent to the I2CC from a DMA, indicating all the transactions have completed.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 15 | DMA_DONE_RX | R/W | 0h | I2CC DMA Done on RX Event Channel Interrupt Mask. This interrupt is raised when the DONE signal is sent to the I2CC from a DMA indicating all the transactions have completed.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 14 | RESERVED | R | 0h | |
| 13 | TIMEOUTB | R/W | 0h | I2CC Timeout Counter B Interrupt Mask. This interrupt is raised when the SCL line is continously high for longer than the clock timeout period programmed by the TIMEOUT_CTL.TCNTLB register.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 12 | TIMEOUTA | R/W | 0h | I2CC Timeout Counter A Interrupt Mask. This interrupt is raised when the SCL line is continously low for longer than the clock timeout period programmed by the TIMEOUT_CTL.TCNTLA register.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 11 | PEC_RX_ERR | R/W | 0h | I2CC Receive PEC Error Interrupt Mask. This interrupt is raised when the PEC calculated by the I2CC does not match received PEC. This interrupt is only available on Advanced I2CC instances.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |



Copyright © 2026 Texas Instruments Incorporated

**Table 30-28. IMASK Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 10 | ARBLOST | R/W | 0h | I2CC Arbitration Lost Interrupt Mask. This flag indicates that the I2CC has lost an arbitration contest with another Controller in a multi-controller system.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 9 | STOP | R/W | 0h | I2CC STOP Detection Interrupt Mask. This interrupt is raised when a STOP condition is detected on the bus, either generated by the current I2CC or another I2C Controller when in multi-master mode.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 8 | START | R/W | 0h | I2CC START Detection Interrupt Mask. This interrupt is raised when a START condition is detected on the bus, either generated by the current I2CC or another I2C Controller when in multi-master mode.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 7 | NACK | R/W | 0h | I2CC Address/Data NACK Interrupt Mask. This interrupt is raised when an address or data NACK is received by the I2CC.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 6 | RESERVED | R | 0h | |
| 5 | TXEMPTY | R/W | 0h | I2CC TX FIFO Empty Interrupt Mask. This interrupt is raised when all data has been shifted out of the TX FIFO and the I2CC transmitter goes idle.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 4 | RXFULL | R/W | 0h | I2CC RX FIFO Full Interrupt Mask. This interrupt is raised when the RX FIFO is full.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 3 | TXTRG | R/W | 0h | I2CC Transmit Trigger Interrupt Mask. This interrupt is raised when the TX FIFO condition programmed by the IFLS register is met.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 2 | RXTRG | R/W | 0h | I2CC Receive Trigger Interrupt Mask. This interrupt is raised when the RX FIFO condition programmed by the IFLS register is met.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 1 | TXDONE | R/W | 0h | I2CC Transmit Done Interrupt Mask. This interrupt is raised when a burst length of CTR.BLEN bytes is transmitted. For variants where the BLEN register is not available, it is raised after each byte is transmitted. In the case of a quick command, the interrupt is raised when a quick command with R/Wn bit set to '0'.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 0 | RXDONE | R/W | 0h | I2CC Receive Done Interrupt Mask. This interrupt is raised when a burst length of CTR.BLEN bytes is received. For variants where the BLEN register is not available, it is raised after each byte is received. In the case of a quick command, the interrupt is raised when a quick command with R/Wn bit set to '1'.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.1.6 RIS (Offset = 30h) [Reset = 00000000h] (CPU_INT)

RIS is shown in Figure 30-31 and described in Table 30-29.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Figure 30-31. RIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | DMA_DONE_TX |
| R-0h | | | | | | | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DMA_DONE_RX | RESERVED | TIMEOUTB | TIMEOUTA | PEC_RX_ERR | ARBLOST | STOP | START |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| NACK | RESERVED | TXEMPTY | RXFULL | TXTRG | RXTRG | TXDONE | RXDONE |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 30-29. RIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-17 | RESERVED | R | 0h | |
| 16 | DMA_DONE_TX | R | 0h | I2CC DMA Done on TX Event Channel Interrupt Raw Status. This interrupt is raised when the DONE signal is sent to the I2CC from a DMA, indicating all the transactions have completed.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 15 | DMA_DONE_RX | R | 0h | I2CC DMA Done on RX Event Channel Interrupt Raw Status. This interrupt is raised when the DONE signal is sent to the I2CC from a DMA indicating all the transactions have completed.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 14 | RESERVED | R | 0h | |
| 13 | TIMEOUTB | R | 0h | I2CC Timeout Counter B Interrupt Raw Status. This interrupt is raised when the SCL line is continously high for longer than the clock timeout period programmed by the TIMEOUT_CTL.TCNTLB register.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 12 | TIMEOUTA | R | 0h | I2CC Timeout Counter A Interrupt Raw Status. This interrupt is raised when the SCL line is continously low for longer than the clock timeout period programmed by the TIMEOUT_CTL.TCNTLA register.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 11 | PEC_RX_ERR | R | 0h | I2CC Receive PEC Error Interrupt Raw Status. This interrupt is raised when the PEC calculated by the I2CC does not match received PEC. This interrupt is only available on Advanced I2CC instances.<br>0h = Interrupt did not occur<br>1h = Interrupt Occured |



Copyright © 2026 Texas Instruments Incorporated

**Table 30-29. RIS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 10 | ARBLOST | R | 0h | I2CC Arbitration Lost Interrupt Raw Status. This flag indicates that the I2CC has lost an arbitration contest with another Controller in a multi-controller system.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 9 | STOP | R | 0h | I2CC STOP Detection Interrupt Raw Status. This interrupt is raised when a STOP condition is detected on the bus, either generated by the current I2CC or another I2C Controller when in multi-master mode.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 8 | START | R | 0h | I2CC START Detection Interrupt Raw Status. This interrupt is raised when a START condition is detected on the bus, either generated by the current I2CC or another I2C Controller when in multi-master mode.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 7 | NACK | R | 0h | I2CC Address/Data NACK Interrupt Raw Status. This interrupt is raised when an address or data NACK is received by the I2CC.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 6 | RESERVED | R | 0h | |
| 5 | TXEMPTY | R | 0h | I2CC TX FIFO Empty Interrupt Raw Status. This interrupt is raised when all data has been shifted out of the TX FIFO and the I2CC transmitter goes idle.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 4 | RXFULL | R | 0h | I2CC RX FIFO Full Interrupt Raw Status. This interrupt is raised when the RX FIFO is full.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 3 | TXTRG | R | 0h | I2CC Transmit Trigger Interrupt Raw Status. This interrupt is raised when the TX FIFO condition programmed by the IFLS register is met.<br>0h = Clear Interrupt Mask<br>1h = Interrupt occurred |
| 2 | RXTRG | R | 0h | I2CC Receive Trigger Interrupt Raw Status. This interrupt is raised when the RX FIFO condition programmed by the IFLS register is met.<br>0h = Clear Interrupt Mask<br>1h = Interrupt occurred |
| 1 | TXDONE | R | 0h | I2CC Transmit Done Interrupt Raw Status. This interrupt is raised when a burst length of CTR.BLEN bytes is transmitted. For variants where the BLEN register is not available, it is raised after each byte is transmitted. In the case of a quick command, the interrupt is raised when a quick command with R/Wn bit set to '0'.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 0 | RXDONE | R | 0h | I2CC Receive Done Interrupt Raw Status. This interrupt is raised when a burst length of CTR.BLEN bytes is received. For variants where the BLEN register is not available, it is raised after each byte is received. In the case of a quick command, the interrupt is raised when a quick command with R/Wn bit set to '1'.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.1.7 MIS (Offset = 38h) [Reset = 00000000h] (CPU_INT)

MIS is shown in Figure 30-32 and described in Table 30-30.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 30-32. MIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | DMA_DONE_TX |
| R-0h | | | | | | | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DMA_DONE_RX | RESERVED | TIMEOUTB | TIMEOUTA | PEC_RX_ERR | ARBLOST | STOP | START |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| NACK | RESERVED | TXEMPTY | RXFULL | TXTRG | RXTRG | TXDONE | RXDONE |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 30-30. MIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-17 | RESERVED | R | 0h | |
| 16 | DMA_DONE_TX | R | 0h | I2CC DMA Done on TX Event Channel Interrupt Masked Status. This interrupt is raised when the DONE signal is sent to the I2CC from a DMA, indicating all the transactions have completed.<br>0h = Interrupt did not occur or mask was not enabled<br>1h = Interrupt occurred |
| 15 | DMA_DONE_RX | R | 0h | I2CC DMA Done on RX Event Channel Interrupt Masked Status. This interrupt is raised when the DONE signal is sent to the I2CC from a DMA indicating all the transactions have completed.<br>0h = Interrupt did not occur or mask was not enabled<br>1h = Interrupt occurred |
| 14 | RESERVED | R | 0h | |
| 13 | TIMEOUTB | R | 0h | I2CC Timeout Counter B Interrupt Masked Status. This interrupt is raised when the SCL line is continously high for longer than the clock timeout period programmed by the TIMEOUT_CTL.TCNTLB register.<br>0h = Clear interrupt mask<br>1h = Set interrupt mask |
| 12 | TIMEOUTA | R | 0h | I2CC Timeout Counter A Interrupt Masked Status. This interrupt is raised when the SCL line is continously low for longer than the clock timeout period programmed by the TIMEOUT_CTL.TCNTLA register.<br>0h = Interrupt did not occur or mask was not enabled<br>1h = Interrupt occurred |
| 11 | PEC_RX_ERR | R | 0h | I2CC Receive PEC Error Interrupt Masked Status. This interrupt is raised when the PEC calculated by the I2CC does not match received PEC. This interrupt is only available on Advanced I2CC instances.<br>0h = Clear interrupt mask<br>1h = Set interrupt mask |



Copyright © 2026 Texas Instruments Incorporated

**Table 30-30. MIS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 10 | ARBLOST | R | 0h | I2CC Arbitration Lost Interrupt Masked Status. This flag indicates that the I2CC has lost an arbitration contest with another Controller in a multi-controller system.<br>0h = Interrupt did not occur or mask was not enabled<br>1h = Interrupt occurred |
| 9 | STOP | R | 0h | I2CC STOP Detection Interrupt Masked Status. This interrupt is raised when a STOP condition is detected on the bus, either generated by the current I2CC or another I2C Controller when in multi-master mode.<br>0h = Interrupt did not occur or mask was not enabled<br>1h = Interrupt occurred |
| 8 | START | R | 0h | I2CC START Detection Interrupt Masked Status. This interrupt is raised when a START condition is detected on the bus, either generated by the current I2CC or another I2C Controller when in multi-master mode.<br>0h = Interrupt did not occur or mask was not enabled<br>1h = Interrupt occurred |
| 7 | NACK | R | 0h | I2CC Address/Data NACK Interrupt Masked Status. This interrupt is raised when an address or data NACK is received by the I2CC.<br>0h = Interrupt did not occur or mask was not enabled<br>1h = Interrupt occurred |
| 6 | RESERVED | R | 0h | |
| 5 | TXEMPTY | R | 0h | I2CC TX FIFO Empty Interrupt Masked Status. This interrupt is raised when all data has been shifted out of the TX FIFO and the I2CC transmitter goes idle.<br>0h = Interrupt did not occur or mask was not enabled<br>1h = Interrupt occurred |
| 4 | RXFULL | R | 0h | I2CC RX FIFO Full Interrupt Masked Status. This interrupt is raised when the RX FIFO is full.<br>0h = Interrupt did not occur or mask was not enabled<br>1h = Interrupt occurred |
| 3 | TXTRG | R | 0h | I2CC Transmit Trigger Interrupt Masked Status. This interrupt is raised when the TX FIFO condition programmed by the IFLS register is met.<br>0h = Interrupt did not occur or mask was not enabled<br>1h = Interrupt occurred |
| 2 | RXTRG | R | 0h | I2CC Receive Trigger Interrupt Masked Status. This interrupt is raised when the RX FIFO condition programmed by the IFLS register is met.<br>0h = Interrupt did not occur or mask was not enabled<br>1h = Interrupt occurred |
| 1 | TXDONE | R | 0h | I2CC Transmit Done Interrupt Masked Status. This interrupt is raised when a burst length of CTR.BLEN bytes is transmitted. For variants where the BLEN register is not available, it is raised after each byte is transmitted. In the case of a quick command, the interrupt is raised when a quick command with R/Wn bit set to '0'.<br>0h = Interrupt did not occur or mask was not enabled<br>1h = Interrupt occurred |
| 0 | RXDONE | R | 0h | I2CC Receive Done Interrupt Masked Status. This interrupt is raised when a burst length of CTR.BLEN bytes is received. For variants where the BLEN register is not available, it is raised after each byte is received. In the case of a quick command, the interrupt is raised when a quick command with R/Wn bit set to '1'.<br>0h = Interrupt did not occur or mask was not enabled<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.1.8 ISET (Offset = 40h) [Reset = 00000000h] (CPU_INT)

ISET is shown in Figure 30-33 and described in Table 30-31.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 30-33. ISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | DMA_DONE_TX |
| R-0h | | | | | | | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DMA_DONE_RX | RESERVED | TIMEOUTB | TIMEOUTA | PEC_RX_ERR | ARBLOST | STOP | START |
| W-0h | R-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| NACK | RESERVED | TXEMPTY | RXFULL | TXTRG | RXTRG | TXDONE | RXDONE |
| W-0h | R-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

**Table 30-31. ISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-17 | RESERVED | R | 0h | |
| 16 | DMA_DONE_TX | W | 0h | Set I2CC DMA Done on TX Event Channel Interrupt. This interrupt is raised when the DONE signal is sent to the I2CC from a DMA, indicating all the transactions have completed.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt Mask |
| 15 | DMA_DONE_RX | W | 0h | Set I2CC DMA Done on RX Event Channel Interrupt. This interrupt is raised when the DONE signal is sent to the I2CC from a DMA indicating all the transactions have completed.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt Mask |
| 14 | RESERVED | R | 0h | |
| 13 | TIMEOUTB | W | 0h | Set I2CC Timeout Counter B Interrupt. This interrupt is raised when the SCL line is continously high for longer than the clock timeout period programmed by the TIMEOUT_CTL.TCNTLB register.<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 12 | TIMEOUTA | W | 0h | Set I2CC Timeout Counter A Interrupt. This interrupt is raised when the SCL line is continously low for longer than the clock timeout period programmed by the TIMEOUT_CTL.TCNTLA register.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt Mask |
| 11 | PEC_RX_ERR | W | 0h | Set I2CC Receive PEC Error Interrupt. This interrupt is raised when the PEC calculated by the I2CC does not match received PEC. This interrupt is only available on Advanced I2CC instances.<br>0h = Writing 0 has no effect<br>1h = Set interrupt |



Copyright © 2026 Texas Instruments Incorporated

**Table 30-31. ISET Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 10 | ARBLOST | W | 0h | Set I2CC Arbitration Lost Interrupt. This flag indicates that the I2CC has lost an arbitration contest with another Controller in a multi-controller system.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt Mask |
| 9 | STOP | W | 0h | Set I2CC STOP Detection Interrupt. This interrupt is raised when a STOP condition is detected on the bus, either generated by the current I2CC or another I2C Controller when in multi-master mode.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt Mask |
| 8 | START | W | 0h | Set I2CC START Detection Interrupt. This interrupt is raised when a START condition is detected on the bus, either generated by the current I2CC or another I2C Controller when in multi-master mode.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt Mask |
| 7 | NACK | W | 0h | Set I2CC Address/Data NACK Interrupt. This interrupt is raised when an address or data NACK is received by the I2CC.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt Mask |
| 6 | RESERVED | R | 0h | |
| 5 | TXEMPTY | W | 0h | Set I2CC TX FIFO Empty Interrupt. This interrupt is raised when all data has been shifted out of the TX FIFO and the I2CC transmitter goes idle.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 4 | RXFULL | W | 0h | Set I2CC RX FIFO Full Interrupt. This interrupt is raised when the RX FIFO is full.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 3 | TXTRG | W | 0h | Set I2CC Transmit Trigger Interrupt. This interrupt is raised when the TX FIFO condition programmed by the IFLS register is met.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 2 | RXTRG | W | 0h | Set I2CC Receive Trigger Interrupt. This interrupt is raised when the RX FIFO condition programmed by the IFLS register is met.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 1 | TXDONE | W | 0h | Set I2CC Transmit Done Interrupt. This interrupt is raised when a burst length of CTR.BLEN bytes is transmitted. For variants where the BLEN register is not available, it is raised after each byte is transmitted. In the case of a quick command, the interrupt is raised when a quick command with R/Wn bit set to '0'.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt Mask |
| 0 | RXDONE | W | 0h | Set I2CC Receive Done Interrupt. This interrupt is raised when a burst length of CTR.BLEN bytes is received. For variants where the BLEN register is not available, it is raised after each byte is received. In the case of a quick command, the interrupt is raised when a quick command with R/Wn bit set to '1'.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt Mask |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.1.9 ICLR (Offset = 48h) [Reset = 00000000h] (CPU_INT)

ICLR is shown in Figure 30-34 and described in Table 30-32.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

**Figure 30-34. ICLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | DMA_DONE_TX |
| R-0h | | | | | | | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DMA_DONE_RX | RESERVED | TIMEOUTB | TIMEOUTA | PEC_RX_ERR | ARBLOST | STOP | START |
| W-0h | R-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| NACK | RESERVED | TXEMPTY | RXFULL | TXTRG | RXTRG | TXDONE | RXDONE |
| W-0h | R-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

**Table 30-32. ICLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-17 | RESERVED | R | 0h | |
| 16 | DMA_DONE_TX | W | 0h | Clear I2CC DMA Done on TX Event Channel Interrupt. This interrupt is raised when the DONE signal is sent to the I2CC from a DMA, indicating all the transactions have completed.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt Mask |
| 15 | DMA_DONE_RX | W | 0h | Clear I2CC DMA Done on RX Event Channel Interrupt. This interrupt is raised when the DONE signal is sent to the I2CC from a DMA indicating all the transactions have completed.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt Mask |
| 14 | RESERVED | R | 0h | |
| 13 | TIMEOUTB | W | 0h | Clear I2CC Timeout Counter B Interrupt. This interrupt is raised when the SCL line is continously high for longer than the clock timeout period programmed by the TIMEOUT_CTL.TCNTLB register.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 12 | TIMEOUTA | W | 0h | Clear I2CC Timeout Counter A Interrupt. This interrupt is raised when the SCL line is continously low for longer than the clock timeout period programmed by the TIMEOUT_CTL.TCNTLA register.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt Mask |
| 11 | PEC_RX_ERR | W | 0h | Clear I2CC Receive PEC Error Interrupt. This interrupt is raised when the PEC calculated by the I2CC does not match received PEC. This interrupt is only available on Advanced I2CC instances.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |



Copyright © 2026 Texas Instruments Incorporated

**Table 30-32. ICLR Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 10 | ARBLOST | W | 0h | Clear I2CC Arbitration Lost Interrupt. This flag indicates that the I2CC has lost an arbitration contest with another Controller in a multi-controller system.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt Mask |
| 9 | STOP | W | 0h | Clear I2CC STOP Detection Interrupt. This interrupt is raised when a STOP condition is detected on the bus, either generated by the current I2CC or another I2C Controller when in multi-master mode.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt Mask |
| 8 | START | W | 0h | Clear I2CC START Detection Interrupt. This interrupt is raised when a START condition is detected on the bus, either generated by the current I2CC or another I2C Controller when in multi-master mode.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt Mask |
| 7 | NACK | W | 0h | Clear I2CC Address/Data NACK Interrupt. This interrupt is raised when an address or data NACK is received by the I2CC.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt Mask |
| 6 | RESERVED | R | 0h | |
| 5 | TXEMPTY | W | 0h | Clear I2CC TX FIFO Empty Interrupt. This interrupt is raised when all data has been shifted out of the TX FIFO and the I2CC transmitter goes idle.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 4 | RXFULL | W | 0h | Clear I2CC RX FIFO Full Interrupt. This interrupt is raised when the RX FIFO is full.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 3 | TXTRG | W | 0h | Clear I2CC Transmit Trigger Interrupt Mask. This interrupt is raised when the TX FIFO condition programmed by the IFLS register is met.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 2 | RXTRG | W | 0h | Clear I2CC Receive Trigger Interrupt Mask. This interrupt is raised when the RX FIFO condition programmed by the IFLS register is met.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 1 | TXDONE | W | 0h | Clear I2CC Transmit Done Interrupt. This interrupt is raised when a burst length of CTR.BLEN bytes is transmitted. For variants where the BLEN register is not available, it is raised after each byte is transmitted. In the case of a quick command, the interrupt is raised when a quick command with R/Wn bit set to '0'.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt Mask |
| 0 | RXDONE | W | 0h | Clear I2CC Receive Done Interrupt. This interrupt is raised when a burst length of CTR.BLEN bytes is received. For variants where the BLEN register is not available, it is raised after each byte is received. In the case of a quick command, the interrupt is raised when a quick command with R/Wn bit set to '1'.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt Mask |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.1.10 IMASK (Offset = 58h) [Reset = 00000000h] (DMA_TRIG_RX)

IMASK is shown in Figure 30-35 and described in Table 30-33.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

**Figure 30-35. IMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | RXTRG | RESERVED | |
| R-0h | | | | | R/W-0h | R-0h | |

**Table 30-33. IMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | RESERVED | R | 0h | |
| 2 | RXTRG | R/W | 0h | I2CC Receive DMA Trigger Mask. The DMA is triggered when the RX FIFO condition programmed by the IFLS register is met.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 1-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.1.11 RIS (Offset = 60h) [Reset = 00000000h] (DMA_TRIG_RX)

RIS is shown in Figure 30-36 and described in Table 30-34.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Figure 30-36. RIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | RXTRG | RESERVED | |
| R-0h | | | | | R-0h | R-0h | |

**Table 30-34. RIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | RESERVED | R | 0h | |
| 2 | RXTRG | R | 0h | I2CC Receive DMA Trigger Raw Status. The DMA is triggered when the RX FIFO condition programmed by the IFLS register is met.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 1-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.1.12 MIS (Offset = 68h) [Reset = 00000000h] (DMA_TRIG_RX)

MIS is shown in Figure 30-37 and described in Table 30-35.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers. A flag set in this register can be cleared by writing 1 to the ICLR register bit.

**Figure 30-37. MIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | RXTRG | RESERVED | |
| R-0h | | | | | R-0h | R-0h | |

**Table 30-35. MIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | RESERVED | R | 0h | |
| 2 | RXTRG | R | 0h | I2CC Receive DMA Trigger Masked Status. The DMA is triggered when the RX FIFO condition programmed by the IFLS register is met.<br>0h = Interrupt did not occur or mask was not enabled<br>1h = Interrupt occurred |
| 1-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.1.13 ISET (Offset = 70h) [Reset = 00000000h] (DMA_TRIG_RX)

ISET is shown in Figure 30-38 and described in Table 30-36.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 30-38. ISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | RXTRG | RESERVED | |
| R-0h | | | | | W-0h | R-0h | |

**Table 30-36. ISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | RESERVED | R | 0h | |
| 2 | RXTRG | W | 0h | Set I2CC Receive DMA Trigger. The DMA is triggered when the RX FIFO condition programmed by the IFLS register is met.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt Mask |
| 1-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.1.14 IMASK (Offset = 88h) [Reset = 00000000h] (DMA_TRIG_TX)

IMASK is shown in Figure 30-39 and described in Table 30-37.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

**Figure 30-39. IMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | TXTRG | RESERVED | | |
| R-0h | | | | R/W-0h | R-0h | | |

**Table 30-37. IMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | TXTRG | R/W | 0h | I2CC Transmit DMA Trigger Mask. The DMA is triggered when the TX FIFO condition programmed by the IFLS register is met.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 2-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.1.15 RIS (Offset = 90h) [Reset = 00000000h] (DMA_TRIG_TX)

RIS is shown in Figure 30-40 and described in Table 30-38.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Figure 30-40. RIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | TXTRG | RESERVED | | |
| R-0h | | | | R-0h | R-0h | | |

**Table 30-38. RIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | TXTRG | R | 0h | I2CC Transmit DMA Trigger Raw Status. The DMA is triggered when the TX FIFO condition programmed by the IFLS register is met.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 2-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.1.16 MIS (Offset = 98h) [Reset = 00000000h] (DMA_TRIG_TX)

MIS is shown in Figure 30-41 and described in Table 30-39.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 30-41. MIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | TXTRG | RESERVED | | |
| R-0h | | | | R-0h | R-0h | | |

**Table 30-39. MIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | TXTRG | R | 0h | I2CC Transmit DMA Trigger Masked Status. The DMA is triggered when the TX FIFO condition programmed by the IFLS register is met.<br>0h = Interrupt did not occur or mask was not enabled<br>1h = Interrupt occurred |
| 2-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.1.17 ISET (Offset = A0h) [Reset = 00000000h] (DMA_TRIG_TX)

ISET is shown in Figure 30-42 and described in Table 30-40.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 30-42. ISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | TXTRG | RESERVED | | |
| R-0h | | | | W-0h | R-0h | | |

**Table 30-40. ISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | TXTRG | W | 0h | Set I2CC Transmit DMA Trigger. The DMA is triggered when the TX FIFO condition programmed by the IFLS register is met.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 2-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.1.18 INTCTL (Offset = E4h) [Reset = 00000000h]

INTCTL is shown in Figure 30-43 and described in Table 30-41.

Return to the Summary Table.

Interrupt Control Register.

**Figure 30-43. INTCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | INTEVAL |
| R-0h | | | | | | | W-0h |

**Table 30-41. INTCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | INTEVAL | W | 0h | Writing a 1 to this field re-evaluates the interrupt sources.<br>0h = Writing 0 has no effect<br>1h = Interrupt Eval |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.1.19 CTR (Offset = 100h) [Reset = 00000000h]

CTR is shown in Figure 30-44 and described in Table 30-42.

Return to the Summary Table.

I2CC Control Register. Configures the I2CC operation.

**Figure 30-44. CTR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | BLEN | | | |
| R-0h | | | | R/W-0h | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| BLEN | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | SUSPEND | RD_ON_TXEMPTY | ACKOEN | ACK | STOP | START | FRM_START |
| R-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

**Table 30-42. CTR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-28 | RESERVED | R | 0h | |
| 27-16 | BLEN | R/W | 0h | I2CC Transaction Length. This field contains the programmed length of bytes of the next transaction. For variants that don't have the BLEN register, the BLEN is hardcoded to a value of 1.<br>0h = Smallest value<br>FFFh = Highest possible value |
| 15-7 | RESERVED | R | 0h | |
| 6 | SUSPEND | R/W | 0h | Suspend External Communication. When this bit is set, I2CC communication on the external bus is suspended after the I2CC FSM goes idle (SR.BUSY reads '0').<br>0h = Normal communication<br>1h = External communication suspended |
| 5 | RD_ON_TXEMPTY | R/W | 0h | Read on TX EMPTY Enable/Disable. This feature initiates an I2CC write followed by a I2CC read without needing application intervention.<br>0h = No special behavior<br>1h = The I2CC will transmit all the bytes from TX FIFO, issue a RESTART and receive data as per programmed burst length (BLEN). This bit is ignored if DIR bit is not set to READ. |



Copyright © 2026 Texas Instruments Incorporated

**Table 30-42. CTR Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 4 | ACKOEN | R/W | 0h | ACK Override Enable When ACKOEN is enabled: 1. I2CC receives data according to the configured burst length (BLEN). 2. Upon receiving the complete data burst, I2CC generates a RXDONE interrupt. 3. The I2CC then holds SCL line low (clock stretching) during the acknowledge phase. 4. This pause gives the application time to update the ACKOVAL register. At this point, application has two options: Option 1: Terminate transaction 1. Set ACKOVAL to send a NACK. 2. The I2CC will automatically generate a STOP condition after the NACK. Option 2: Continue transaction 1. Clear ACKOVAL to send an ACK. 2. Update any other settings as needed. 3. The transaction will proceed based on the new configuration. This mechanism allows the application to control the transaction flow after each data burst is received. Note: In some variants, when BLEN register is not available, the design implements a hardcoded BLEN value of '1'. Consult your device datasheet.<br>0h = No special behavior<br>1h = Acknowledge Override Feature Enabled |
| 3 | ACK | R/W | 0h | Final Data Acknowledge Value. Determines whether a ACK or NACK is automatically sent by the I2CC after the last byte, of length 'N', are received from the target. This setting is used when CTR.ACKOEN=0.<br>0h = The last received data byte of a transaction is not acknowledged automatically.<br>1h = The last received data byte of a transaction is acknowledged automatically. |
| 2 | STOP | R/W | 0h | Enable STOP Condition Generation. I2CC is enabled to send the STOP condition on the bus at the end of the current transaction, per the I2C protocol.<br>0h = The controller does not generate the STOP condition.<br>1h = The controller generates the STOP condition. |
| 1 | START | R/W | 0h | Enable START Condition Generation. I2CC is enabled to send the START condition on the bus per the I2C protocol, when CTR.FRM_START is also set.<br>0h = The controller does not generate the START condition.<br>1h = The controller generates the START or repeated START condition. |
| 0 | FRM_START | R/W | 0h | Frame Start. Start the transaction on the I2C bus according to the programmed settings.<br>0h = Write of '0' has no effect. Reads back '0' when there is no ongoing transaction<br>1h = When written to '1', a new transaction is started. Has no effect if there is an ongoing transaction. Reads back as '1' if the transaction is ongoing |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.1.20 CR (Offset = 104h) [Reset = 00000000h]

CR is shown in Figure 30-45 and described in Table 30-43.

Return to the Summary Table.

I2CC Configuration Register. Configures and enables the I2CC.

**Figure 30-45. CR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | CLKSTRETCH | MCTL | ENABLE |
| R-0h | | | | | R/W-0h | R/W-0h | R/W-0h |

**Table 30-43. CR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | RESERVED | R | 0h | |
| 2 | CLKSTRETCH | R/W | 0h | Clock Stretching Enable. This bit enables/disables clock stretching detection of the I2CC.<br>0h = Disables the clock stretching detection. This can be disabled if no Target on the bus does support clock stretching, so that the maximum speed on the bus can be reached.<br>1h = Enables the clock stretching detection. Enabling the clock stretching ensures compliance to the I2C standard but could limit the speed due the clock stretching. |
| 1 | MCTL | R/W | 0h | Multi-Controller Mode Enable/Disable. In Multi-Controller mode, the SCL high time begins counting once the SCL line has been detected high. If this bit is not set, the SCL high time begins counting as soon as the SCL line has been set high by the I2CC.<br>0h = Disable MultiController mode.<br>1h = Enable MultiController mode. |
| 0 | ENABLE | R/W | 0h | Enable I2CC module. After this bit has been set, it should not be set again unless it has been cleared by writing a 0 or by a reset, otherwise transfer failures may occur.<br>0h = Disables operation.<br>1h = Enables operation. |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.1.21 SR (Offset = 108h) [Reset = 00000000h]

SR is shown in Figure 30-46 and described in Table 30-44.

Return to the Summary Table.

I2CC Status Register. This status register indicates the current state of the I2CC.

**Figure 30-46. SR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | BCNT | | | |
| R-0h | | | | R-0h | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| BCNT | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | TXFF | TXFE | RXFF | RXFE | TXCLR | RXCLR | RESERVED |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | BUSBSY | IDLE | ARBLST | DATACK | ADRACK | ERR | BUSY |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 30-44. SR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-28 | RESERVED | R | 0h | |
| 27-16 | BCNT | R | 0h | Burst Transaction Count Status. This field contains the current count-down value of the transaction. This field is only available on Advanced I2CC instances.<br>0h = Smallest value<br>FFFh = Highest possible value |
| 15 | RESERVED | R | 0h | |
| 14 | TXFF | R | 0h | TX FIFO Full Status.<br>0h = TX FIFO is not full<br>1h = TX FIFO is full |
| 13 | TXFE | R | 1h | TX FIFO Empty Status.<br>0h = TX FIFO is not empty<br>1h = TX FIFO is empty |
| 12 | RXFF | R | 0h | RX FIFO Full Status.<br>0h = RX FIFO is not full<br>1h = RX FIFO is full |
| 11 | RXFE | R | 1h | RX FIFO Empty Status.<br>0h = RX FIFO is not empty<br>1h = RX FIFO is empty |
| 10 | TXCLR | R | 0h | TX FIFO Clear Status.<br>0h = TX FIFO is not cleared<br>1h = TX FIFO clear is complete |
| 9 | RXCLR | R | 0h | RX FIFO Clear Status.<br>0h = RX FIFO is not cleared<br>1h = RX FIFO clear is complete |
| 8-7 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

**Table 30-44. SR Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 6 | BUSBSY | R | 0h | I2C Bus Busy Status. Indicates when the external line is busy 1. It is set when there is an ongoing transaction on the bus (i.e. if there is a START condition observed) or if SCL toggles. 2. It is cleared upon a STOP condition or when HIGH TIMEOUT occurs & SCL/SDA are both at their IDLE state. This bit reads back '0' when ACTIVE is cleared. The I2CC will wait until this bit is cleared before starting an I2C transaction. When enabling the controller initially, the application should wait for one I2C clock period, poll BUSBSY and start a transaction.<br>0h = The I2C bus is idle.<br>1h = SET |
| 5 | IDLE | R | 1h | I2CC FSM Idle Status. This bit is set when the internal I2CC state machine, rather than when the external bus, is not busy with an ongoing transaction i.e. there is no transmit/receive of data bytes, START, RESTART, Address or STOP signal generation occuring.<br>0h = The I2C controller is not idle.<br>1h = The I2C controller is idle. |
| 4 | ARBLST | R | 0h | Arbitration Lost Status. This status bit indicates whether or not the I2CC won arbitration on the bus.<br>0h = The I2C controller won arbitration.<br>1h = The I2C controller lost arbitration. |
| 3 | DATACK | R | 0h | Data Acknowledge Status. This status bit indicates the acknowledgement status of the previously transmitted data byte.<br>0h = The transmitted data was acknowledged<br>1h = The transmitted data was not acknowledged. |
| 2 | ADRACK | R | 0h | Acknowledge Address<br>0h = The transmitted address was acknowledged<br>1h = The transmitted address was not acknowledged. |
| 1 | ERR | R | 0h | NACK Error Status. This status bit indicates the acknowledgement status of the previously transmitted target address or data byte.<br>0h = No error was detected on the last operation.<br>1h = An error occurred on the last operation. |
| 0 | BUSY | R | 0h | I2CC FSM Busy Status. This bit is set when the internal I2CC state machine, rather than when the external bus, is busy with an ongoing transaction i.e. during transmit/receive of data bytes configured as per BLEN including START, RESTART, Address and STOP signal generation. In variants where the BLEN register is not available, the design implements a hardcoded BLEN value of '1'.<br>0h = The controller is idle.<br>1h = The controller is busy. |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.1.22 IFLS (Offset = 10Ch) [Reset = 00000022h]

IFLS is shown in Figure 30-47 and described in Table 30-45.

Return to the Summary Table.

I2CC Interrupt FIFO Level Select Register. The IFLS register is the interrupt FIFO level select register. Use this register to define the levels at which the TX, RX, and Receive timeout interrupt flags are triggered. The interrupts are generated based on a transition through a level rather than being based on the level itself. That is, the interrupts are generated when the fill level progresses through the trigger level. For example, if the receive trigger level is set to the half-way mark, the interrupt is triggered when the receive FIFO is filled with two or more characters. Out of reset, the TXIFLSEL and RXIFLSEL bits are configured so that the FIFOs trigger an interrupt at the half-way mark.

**Figure 30-47. IFLS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RXCLR | RXIFLSEL | | | TXCLR | TXIFLSEL | | |
| R/W-0h | R/W-2h | | | R/W-0h | R/W-2h | | |

**Table 30-45. IFLS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7 | RXCLR | R/W | 0h | RX FIFO Clear. Setting this bit will clear the RX FIFO contents.<br>0h = Disable FIFO clear<br>1h = Enable FIFO Clear |
| 6-4 | RXIFLSEL | R/W | 2h | RX FIFO Level Select for generating events (interrupt/DMA). Note: for undefined settings the default configuration is used.<br>1h = RX FIFO >= 1/4 full<br>2h = RX FIFO >= 1/2 full (default)<br>3h = RX FIFO >= 3/4 full<br>4h = Opposite of empty<br>5h = RX FIFO is full<br>6h = RX_FIFO >= (MAX_FIFO_LEN -1)<br>7h = RX_FIFO <= 1 |
| 3 | TXCLR | R/W | 0h | TX FIFO Clear. Setting this bit will clear the TX FIFO contents.<br>0h = Disable FIFO clear<br>1h = Enable FIFO Clear |
| 2-0 | TXIFLSEL | R/W | 2h | TX FIFO Level Select for generating events (interrupt/DMA). Note: for undefined settings the default configuration is used.<br>1h = TX FIFO <= 3/4 empty<br>2h = TX FIFO <= 1/2 empty (default)<br>3h = TX FIFO <= 1/4 empty<br>4h = Opposite of full<br>5h = TX FIFO is empty<br>6h = TX FIFO <= 1<br>7h = TX_FIFO >= (MAX_FIFO_LEN -1) |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.1.23 TPR (Offset = 110h) [Reset = 00000001h]

TPR is shown in Figure 30-48 and described in Table 30-46.

Return to the Summary Table.

I2CC Timer Period Register. This register programs the clock period of the SCL line. Transmission speeds of Standard mode (100kbps), Fast mode (400 kbps), or Fast mode plus (1Mbps) are supported.

**Figure 30-48. TPR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | TPR | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | R/W-1h | | | | | | |

**Table 30-46. TPR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-7 | RESERVED | R | 0h | |
| 6-0 | TPR | R/W | 1h | Timer Period. This field is used in the equation to configure SCL_PERIOD SCL_PERIOD = (1 + TPR ) x (SCL_LP + SCL_HP ) x INT_CLK_PRD Where: - SCL_PRD is the SCL line period (I2C clock). - TPR is the Timer Period register value (range of 1 to 127). - SCL_LP is the SCL Low period (fixed at 6). - SCL_HP is the SCL High period (fixed at 4). - CLK_PRD is the functional clock period in ns.<br>0h = Smallest value<br>7Fh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.1.24 GFCTL (Offset = 118h) [Reset = 00000000h]

GFCTL is shown in Figure 30-49 and described in Table 30-47.

Return to the Summary Table.

I2CC Glitch Filter Control Register. This register configures the glitch filter on the SCL and SDA lines.

**Figure 30-49. GFCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | AGFEN |
| R-0h | | | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | DGFSEL | | |
| R-0h | | | | | R/W-0h | | |

**Table 30-47. GFCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-9 | RESERVED | R | 0h | |
| 8 | AGFEN | R/W | 0h | Analog Glitch Suppression Enable.<br>0h = Analog Glitch Filter disable<br>1h = Analog Glitch Filter enable |
| 7-3 | RESERVED | R | 0h | |
| 2-0 | DGFSEL | R/W | 0h | Digital Glitch Suppression Pulse Width. The digital glitch filter is only available on Basic I2CC instances. This field controls the pulse width select for glitch suppression on the SCL and SDA lines. The following glitch suppression values are in terms of I2Cclk cycles.<br>0h = Bypass<br>1h = 1 clock<br>2h = 2 clocks<br>3h = 3 clocks<br>4h = 4 clocks<br>5h = 8 clocks<br>6h = 16 clocks<br>7h = 31 clocks |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.1.25 BMON (Offset = 11Ch) [Reset = 00000003h]

BMON is shown in Figure 30-50 and described in Table 30-48.

Return to the Summary Table.

I2CC Bus Monitor Register. This register is used to read the current SCL and SDA signal logic levels.

**Figure 30-50. BMON**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | | | SDA | SCL |
| R-0h | | | | | | | | | | | | | | R-1h | R-1h |

**Table 30-48. BMON Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | |
| 1 | SDA | R | 1h | I2C SDA Status.<br>0h = The I2CSDA signal is low.<br>1h = The I2CSDA signal is high. Note: During and right after reset, the SDA pin is in GPIO input mode without the internal pull enabled. For proper I2C operation, the user should have the external pull-up resistor in place before starting any I2C operations. |
| 0 | SCL | R | 1h | I2C SCL Status.<br>0h = The I2CSCL signal is low.<br>1h = The I2CSCL signal is high. Note: During and right after reset, the SCL pin is in GPIO input mode without the internal pull enabled. For proper I2C operation, the user should have the external pull-up resistor in place before starting any I2C operations. |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.1.26 TXDATA (Offset = 120h) [Reset = 00000000h]

TXDATA is shown in Figure 30-51 and described in Table 30-49.

Return to the Summary Table.

I2CC Transmit Data Register. This register is the transmit data register (the interface to the TX FIFO). For transmitted data, if the FIFO is enabled, data written to this location is pushed onto the TX FIFO. If the FIFO is disabled, data is stored in the transmitter holding register (the bottom word of the TX FIFO).

**Figure 30-51. TXDATA**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | DATA | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | W-0h | | | | | | | |

**Table 30-49. TXDATA Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | DATA | W | 0h | Transmit Data. This byte contains the data to be transferred during the next transaction.<br>0h = Smallest value<br>FFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.1.27 RXDATA (Offset = 124h) [Reset = 00000000h]

RXDATA is shown in Figure 30-52 and described in Table 30-50.

Return to the Summary Table.

I2CC Receive Data Register. This field contains the current byte being read in the RX FIFO If the FIFO is not present, the data byte and status are stored in the receiving holding register (the bottom word of the receive FIFO). The received data can be retrieved by reading this register.

**Figure 30-52. RXDATA**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | DATA | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | |

**Table 30-50. RXDATA Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | DATA | R | 0h | Received Data. This field contains the last received data.<br>0h = Smallest value<br>FFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.1.28 PECSR (Offset = 128h) [Reset = 00000000h]

PECSR is shown in Figure 30-53 and described in Table 30-51.

Return to the Summary Table.

I2CC PEC Status Register. This status register can be used to read the current status of the packet error checking mechanism.

**Figure 30-53. PECSR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | PECSTS_ERROR | PECSTS_CHECK |
| R-0h | | | | | | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | PECBYTECNT |
| R-0h | | | | | | | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PECBYTECNT | | | | | | | |
| R-0h | | | | | | | |

**Table 30-51. PECSR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-18 | RESERVED | R | 0h | |
| 17 | PECSTS_ERROR | R | 0h | PEC Status Error. This status bit indicates if a PEC check error occurred in the transaction that occurred before the last STOP. This bit is cleared automatically when the next STOP condition is observed on the bus.<br>0h = Indicates PEC check error did not occur in the transaction that occurred before the last STOP<br>1h = Indicates if a PEC check error occurred in the transaction that occurred before the last STOP |
| 16 | PECSTS_CHECK | R | 0h | PEC Status Check. This status bit indicates if the PEC was checked in the transaction that occurred before the last STOP. This bit is cleared automatically when the next STOP condition is observed on the bus.<br>0h = Indicates PEC was not checked in the transaction that occurred before the last STOP<br>1h = Indicates if the PEC was checked in the transaction that occurred before the last STOP |
| 15-9 | RESERVED | R | 0h | |
| 8-0 | PECBYTECNT | R | 0h | PEC Byte Count. This is the current PEC Byte Count of the Controller State Machine.<br>0h = Minimum Value<br>1FFh = Maximum Value |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.1.29 TA (Offset = 14Ch) [Reset = 00000000h]

TA is shown in Figure 30-54 and described in Table 30-52.

Return to the Summary Table.

I2CC Target Address Register. This register configures the target address, addressing mode, and direction (controller-transmit or controller-receive) of the next transaction.

**Figure 30-54. TA**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MODE | RESERVED | | | | ADDR | | |
| R/W-0h | R-0h | | | | R/W-0h | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ADDR | | | | | | | DIR |
| R/W-0h | | | | | | | R/W-0h |

**Table 30-52. TA Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15 | MODE | R/W | 0h | Address Mode Select. This bit selects the addressing mode to be used. When 0, 7-bit addressing is used. When 1, 10-bit addressing is used.<br>0h = 7-bit addressing mode<br>1h = 10-bit addressing mode |
| 14-11 | RESERVED | R | 0h | |
| 10-1 | ADDR | R/W | 0h | I2C Target Address. This field specifies bits A9 through A0 of the Target address. In 7-bit addressing mode, as selected by TA.MODE bit, the top 3 bits are don't care.<br>0h = Smallest value<br>3FFh = Highest possible value |
| 0 | DIR | R/W | 0h | Receive/Transmit Direction. The DIR bit specifies if the next operation is a Receive (High) or Transmit (Low). 0h = Transmit 1h = Receive<br>0h = in transmit mode.<br>1h = is in receive mode. |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.1.30 TIMEOUT_CNT (Offset = 150h) [Reset = 00020002h]

TIMEOUT_CNT is shown in Figure 30-55 and described in Table 30-53.

Return to the Summary Table.

I2CC Timeout Count Register. This register contains the upper 8 bits of a 12-bit current counter values for Timeout Counter A and Timeout Counter B. The lower four bits of the counter are not user visible and are always 0h.

**Figure 30-55. TIMEOUT_CNT**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | TCNTB | | | | | | | | RESERVED | | | | | | | | TCNTA | | | | | | | |
| R-0h | | | | | | | | R-2h | | | | | | | | R-0h | | | | | | | | R-2h | | | | | | | |

**Table 30-53. TIMEOUT_CNT Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | RESERVED | R | 0h | |
| 23-16 | TCNTB | R | 2h | Timeout Count B Current Count. This field contains the upper 8 bits of a 12-bit current counter for Timeout Counter B. This field is only available on Advanced I2CC instances.<br>0h = Smallest Value<br>FFh = Highest possible value |
| 15-8 | RESERVED | R | 0h | |
| 7-0 | TCNTA | R | 2h | Timeout Count A Current Count. This field contains the upper 8 bits of a 12-bit current counter for Timeout Counter A.<br>0h = Smallest Value<br>FFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.1.31 TIMEOUT_CTL (Offset = 154h) [Reset = 00020002h]

TIMEOUT_CTL is shown in Figure 30-56 and described in Table 30-54.

Return to the Summary Table.

I2CC Timeout Control Register. This register configures Timeout Counter A and Timeout Counter B.

**Figure 30-56. TIMEOUT_CTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| TCNTBEN | RESERVED | | | | | | |
| R/W-0h | R-0h | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TCNTLB | | | | | | | |
| R/W-2h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TCNTAEN | RESERVED | | | | | | |
| R/W-0h | R-0h | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TCNTLA | | | | | | | |
| R/W-2h | | | | | | | |

**Table 30-54. TIMEOUT_CTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | TCNTBEN | R/W | 0h | Timeout Counter B Enable. This field is only available on Advanced I2CC instances.<br>0h = Disable Timeout Counter B<br>1h = Enable Timeout Counter B |
| 30-24 | RESERVED | R | 0h | |
| 23-16 | TCNTLB | R/W | 2h | Timeout Count B Load. Counter B is used for SCL High Detection. This field contains the upper 8 bits of a 12-bit pre-load value for the Timeout Counter B. NOTE: The value of CNTLB must be greater than 1h. Each count is equal to 1* clock period. For example, with 10MHz functional clock one timeout period will be equal to1*100ns.<br>0h = Smallest possible value<br>FFh = Highest possible value |
| 15 | TCNTAEN | R/W | 0h | Timeout Counter A Enable.<br>0h = Disable Timeout Counter A<br>1h = Enable Timeout Counter A |
| 14-8 | RESERVED | R | 0h | |
| 7-0 | TCNTLA | R/W | 2h | Timeout Counter A Load Value. Counter A is used for SCL low detection. This field contains the upper 8 bits of a 12-bit pre-load value for the Timeout Counter A. NOTE: The value of CNTLA must be greater than 1h. Each count is equal to 520 times the timeout period of functional clock. For example, with 8MHz functional clock and a 100KHz operating I2C clock, one timeout period will be equal to (1 / 8MHz) * 520 or 65 us.<br>0h = Smallest Value<br>FFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.1.32 PECCTL (Offset = 158h) [Reset = 00000000h]

PECCTL is shown in Figure 30-57 and described in Table 30-55.

Return to the Summary Table.

I2CC PEC Control Register. This register configures the packet error checking (PEC) feature. This register is only available on Advanced I2CC instances.

**Figure 30-57. PECCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | PECEN | RESERVED | | | PECCNT |
| R-0h | | | R/W-0h | R-0h | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PECCNT | | | | | | | |
| R/W-0h | | | | | | | |

**Table 30-55. PECCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-13 | RESERVED | R | 0h | |
| 12 | PECEN | R/W | 0h | PEC Enable. This bit enables the SMB Packet Error Checking (PEC). When enabled, the PEC is calculated on all bits except the START, STOP, and ACK/NACK. The PEC LSFR and the Byte Counter is set to 0 when the State Machine is in the IDLE state, which occur following a STOP or when a timeout occurs. The Counter is also set to 0 after the PEC byte is sent or received. Note that the NACK is automatically send following a PEC byte that results in a PEC error. The PEC Polynomial is x^8 + x^2 + x^1 + 1.<br>0h = PEC is disabled<br>1h = PEC is enabled |
| 11-9 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

**Table 30-55. PECCTL Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 8-0 | PECCNT | R/W | 0h | PEC Count. When this field is non zero, the number of I2C bytes are counted (Note that although the PEC is calculated on the I2C address it is not counted at a byte). When the byte count = PECCNT and the I2CC is transmitting, the contents of the LSFR is loaded into the shift register instead of the byte received from the TX FIFO. When the state machine is receiving, after the last bit of this byte is received the LSFR is checked and if it is non-zero, a PEC RX Error interrupt is generated. The I2C packet must be padded to include the PEC byte for both transmit and receive. In transmit mode, the FIFO must be loaded with a dummy PEC byte. In receive mode, the PEC byte will be passed to the RX FIFO. In the a typical controller use case, the application would set PECEN=1 and PECCNT=SMB packet length (not including the Target Address byte, but including the PEC byte). The I2CC application would then configure the DMA to allow the packet to complete unassisted and write CTR to initiate the transaction. Note that when the byte count = PECCNT, the byte count is reset to 0 and multiple PEC calculations can automatically occur within a single I2C transaction. Note that any write to the PECCTL register will clear the current PEC byte count in the I2CC state machine.<br>0h = Minimum Value<br>1FFh = Maximum Value |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.2 UNICOMMI2CT Registers

Table 30-56 lists the memory-mapped registers for the UNICOMMI2CT registers. All register offset addresses not listed in Table 30-56 should be considered as reserved locations and the register contents should not be modified.

**Table 30-56. UNICOMMI2CT Registers**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 1000h | CLKDIV | Clock Divider | | Go |
| 1008h | CLKSEL | Clock Select for Ultra Low Power peripherals | | Go |
| 1020h | IIDX | Interrupt index | CPU_INT | Go |
| 1028h | IMASK | Interrupt mask | CPU_INT | Go |
| 1030h | RIS | Raw interrupt status | CPU_INT | Go |
| 1038h | MIS | Masked interrupt status | CPU_INT | Go |
| 1040h | ISET | Interrupt set | CPU_INT | Go |
| 1048h | ICLR | Interrupt clear | CPU_INT | Go |
| 1058h | IMASK | Interrupt mask | DMA_TRIG_RX | Go |
| 1060h | RIS | Raw interrupt status | DMA_TRIG_RX | Go |
| 1068h | MIS | Masked interrupt status | DMA_TRIG_RX | Go |
| 1070h | ISET | Interrupt set | DMA_TRIG_RX | Go |
| 1088h | IMASK | Interrupt mask | DMA_TRIG_TX | Go |
| 1090h | RIS | Raw interrupt status | DMA_TRIG_TX | Go |
| 1098h | MIS | Masked interrupt status | DMA_TRIG_TX | Go |
| 10A0h | ISET | Interrupt set | DMA_TRIG_TX | Go |
| 10E4h | INTCTL | Interrupt control register | | Go |
| 1100h | CTR | I2C Target Control Register | | Go |
| 1104h | ACKCTL | I2C Target ACK Control | | Go |
| 1108h | SR | Status Register | | Go |
| 110Ch | IFLS | Interrupt FIFO Level Select Register | | Go |
| 1118h | GFCTL | I2C Glitch Filter Control | | Go |
| 1120h | TXDATA | I2C TXData | | Go |
| 1124h | RXDATA | I2C RXData | | Go |
| 1128h | PECSR | PEC status register | | Go |
| 1148h | OAR2 | Own Address 2 | | Go |
| 114Ch | OAR | I2C Own Address | | Go |
| 1150h | TIMEOUT_CNT | I2C Timeout Count Register | | Go |
| 1154h | TIMEOUT_CTL | I2C Timeout Count Control Register | | Go |
| 1158h | PECCTL | I2C PEC control register | | Go |

Complex bit access types are encoded to fit into small table cells. Table 30-57 shows the codes that are used for access types in this section.



Copyright © 2026 Texas Instruments Incorporated

**Table 30-57. UNICOMMI2CT Access Type Codes**

| Access Type | Code | Description |
|---|---|---|
| Read Type | | |
| R | R | Read |
| Write Type | | |
| W | W | Write |
| Reset or Default Value | | |
| *-n* | | Value after reset or the default value |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.2.1 CLKDIV (Offset = 1000h) [Reset = 00000000h]

CLKDIV is shown in Figure 30-58 and described in Table 30-58.

Return to the Summary Table.

I2CT Clcok Divider. This register is used to specify module-specific divide ratio of the functional clock

**Figure 30-58. CLKDIV**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | | RATIO | | |
| R-0h | | | | | | | | | | | | | R/W-0h | | |

**Table 30-58. CLKDIV Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | RESERVED | R | 0h | |
| 2-0 | RATIO | R/W | 0h | Selects divide ratio of module clock Division factor 0 : DIV_BY_1 1 : DIV_BY_2 .... 63: DIV_BY_64<br>0h = Do not divide clock source<br>1h = Divide clock source by 2<br>2h = Divide clock source by 3<br>3h = Divide clock source by 4<br>4h = Divide clock source by 5<br>5h = Divide clock source by 6<br>6h = Divide clock source by 7<br>7h = Divide clock source by 8<br>3Ch = Divide clock source by 61<br>3Dh = Divide clock source by 62<br>3Eh = Divide clock source by 63<br>3Fh = Divide clock source by 64 |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.2.2 CLKSEL (Offset = 1008h) [Reset = 00000000h]

CLKSEL is shown in Figure 30-59 and described in Table 30-59.

Return to the Summary Table.

I2CT Clock Source Selection.

**Figure 30-59. CLKSEL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | ASYNC_PLL_SEL | ASYNC_HFCLK_SEL | ASYNC_SYSCLK_SEL | RESERVED |
| R-0h | | | | R/W-0h | R/W-0h | R/W-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | BUSCLK_SEL | MFCLK_SEL | RESERVED | |
| R-0h | | | | R/W-0h | R/W-0h | R-0h | |

**Table 30-59. CLKSEL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-12 | RESERVED | R | 0h | |
| 11 | ASYNC_PLL_SEL | R/W | 0h | Asynchronous PLL selected as source<br>0h = Does not select this clock as a source<br>1h = Select this clock as a source |
| 10 | ASYNC_HFCLK_SEL | R/W | 0h | Asynchronous HFCLK selected as source<br>0h = Does not select this clock as a source<br>1h = Select this clock as a source |
| 9 | ASYNC_SYSCLK_SEL | R/W | 0h | Asynchronous sysclk selected as source<br>0h = Does not select this clock as a source<br>1h = Select this clock as a source |
| 8-4 | RESERVED | R | 0h | |
| 3 | BUSCLK_SEL | R/W | 0h | Selects BUS CLK as clock source if enabled<br>0h = Does not select this clock as a source<br>1h = Select this clock as a source |
| 2 | MFCLK_SEL | R/W | 0h | Selects MFCLK as clock source if enabled<br>0h = Does not select this clock as a source<br>1h = Select this clock as a source |
| 1-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.2.3 IIDX (Offset = 1020h) [Reset = 00000000h]

IIDX is shown in Figure 30-60 and described in Table 30-60.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index.

**Figure 30-60. IIDX**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | STAT | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | |

**Table 30-60. IIDX Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | STAT | R | 0h | I2CT Interrupt Vector Value. This field provides the highest priority interrupt index. A read clears the corresponding interrupt flag in RIS and MIS. 15h-1Fh = Reserved.<br>00h = No Interrupt Pending.<br>1h = Receive Done Flag.<br>2h = Transmit Done Flag.<br>3h = RX FIFO Trigger Level.<br>4h = TX FIFO Trigger level<br>5h = RX FIFO Full Event/Interrupt Pending.<br>6h = TX FIFO Empty Event/Interrupt Pending.<br>7h = I2CT TX FIFO Underflow Event.<br>8h = I2CT RX FIFO Overflow Event.<br>9h = General Call Event.<br>Ah = START Event.<br>Bh = STOP Event.<br>Ch = PEC Receive Error Event.<br>Dh = Timeout Counter A Event.<br>Eh = Timeout Counter B Event.<br>10h = DMA DONE on Channel RX.<br>11h = DMA DONE on Channel TX.<br>12h = Arbitration Lost.<br>13h = DMA PRE IRQ INTERRUPT<br>14h = DMA PRE IRQ INTERRUPT |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.2.4 IMASK (Offset = 1028h) [Reset = 00000000h]

IMASK is shown in Figure 30-61 and described in Table 30-61.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

**Figure 30-61. IMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | RESERVED | RESERVED | ARBLOST | DMA_DONE_TX |
| R-0h | | | | R-0h | R-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DMA_DONE_RX | RESERVED | TIMEOUTB | TIMEOUTA | PEC_RX_ERR | STOP | START | GENCALL |
| R/W-0h | R-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RX_OVFL | TX_UNFL | TXEMPTY | RXFULL | TXTRG | RXTRG | TXDONE | RXDONE |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

**Table 30-61. IMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | RESERVED | R | 0h | |
| 19 | RESERVED | R | 0h | Reserved |
| 18 | RESERVED | R | 0h | Reserved |
| 17 | ARBLOST | R/W | 0h | I2CT Arbitration Lost Interrupt Mask. This interrupt is raised when another I2C target on the bus wins arbitration during the "Get UDID" sequence in the SMBus address resolution protocol.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 16 | DMA_DONE_TX | R/W | 0h | I2CT DMA Done on TX Event Channel Interrupt Mask. This interrupt is raised when the DONE signal is sent to the I2CT from a DMA, indicating all the transactions have completed.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 15 | DMA_DONE_RX | R/W | 0h | I2CT DMA Done on RX Event Channel Interrupt Mask. This interrupt is raised when the DONE signal is sent to the I2CT from a DMA, indicating all the transactions have completed.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 14 | RESERVED | R | 0h | |
| 13 | TIMEOUTB | R/W | 0h | I2CT Timeout Counter B Interrupt Mask. This interrupt is raised when the SCL line is continously high for longer than the clock timeout period programmed by the TIMEOUT_CTL.TCNTLB register.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 12 | TIMEOUTA | R/W | 0h | I2CT Timeout Counter A Interrupt Mask. This interrupt is raised when the SCL line is continously low for longer than the clock timeout period programmed by the TIMEOUT_CTL.TCNTLA register.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |



Copyright © 2026 Texas Instruments Incorporated

**Table 30-61. IMASK Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 11 | PEC_RX_ERR | R/W | 0h | I2CT Receive PEC Error Interrupt Mask. This interrupt is raised when the PEC calculated by the I2CT does not match received PEC.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 10 | STOP | R/W | 0h | I2CT STOP Condition Interrupt Mask. This interrupt is raised when a STOP condition is detected on the bus.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 9 | START | R/W | 0h | I2CT START Condition Interrupt Mask. This interrupt is raised when a START or repeated START condition is detected on the bus.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 8 | GENCALL | R/W | 0h | I2CT General Call Interrupt Mask. This interrupt is raised when a general call frame, with an address of 0x00 and R/W bit of '0', is received.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 7 | RX_OVFL | R/W | 0h | I2CT RX FIFO Overflow Interrupt Mask. This interrupt is raised when the I2CT receiver tries to receive data, as a result of receiving an address match with a R/W bit of '0', but the I2CT RX FIFO is full.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 6 | TX_UNFL | R/W | 0h | I2CT TX FIFO Underflow Interrupt Mask. This interrupt is raised when the I2CT transmitter tries to send data, as a result of having received an address match with a R/W bit of '1', but the I2CT TX FIFO is empty.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 5 | TXEMPTY | R/W | 0h | I2CT TX FIFO Empty Interrupt Mask. This interrupt is raised when all data has been shifted out of the TX FIFO and the I2CT transmitter goes idle.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 4 | RXFULL | R/W | 0h | I2CT RX FIFO Full Interrupt Mask. This interrupt is raised when the RX FIFO is full.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 3 | TXTRG | R/W | 0h | I2CT Transmit Trigger Interrupt Mask. This interrupt is raised when the TX FIFO condition, programmed by the IFLS register, is met. when FIFO is not present, this indicates that the buffer is empty.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 2 | RXTRG | R/W | 0h | I2CT Receive Trigger Interrupt Mask. This interrupt is raised when the RX FIFO condition programmed by the IFLS register is met. When FIFO is not present, this indicates that the buffer is full.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 1 | TXDONE | R/W | 0h | I2CT Transmit Done Interrupt Mask. This interrupt is raised after each byte is transmitted.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 0 | RXDONE | R/W | 0h | I2CT Receive Done Interrupt Mask. This interrupt is raised after each byte is received.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.2.5 RIS (Offset = 1030h) [Reset = 00000000h]

RIS is shown in Figure 30-62 and described in Table 30-62.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Figure 30-62. RIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | RESERVED | RESERVED | ARBLOST | DMA_DONE_TX |
| R-0h | | | | R-0h | R-0h | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DMA_DONE_RX | RESERVED | TIMEOUTB | TIMEOUTA | PEC_RX_ERR | STOP | START | GENCALL |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RX_OVFL | TX_UNFL | TXEMPTY | RXFULL | TXTRG | RXTRG | TXDONE | RXDONE |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 30-62. RIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | RESERVED | R | 0h | |
| 19 | RESERVED | R | 0h | Reserved |
| 18 | RESERVED | R | 0h | Reserved |
| 17 | ARBLOST | R | 0h | I2CT Arbitration Lost Interrupt Raw Status. This interrupt is raised when another I2C target on the bus wins arbitration during the "Get UDID" sequence in the SMBus address resolution protocol.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 16 | DMA_DONE_TX | R | 0h | I2CT DMA Done on TX Event Channel Interrupt Raw Status. This interrupt is raised when the DONE signal is sent to the I2CT from a DMA, indicating all the transactions have completed.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 15 | DMA_DONE_RX | R | 0h | I2CT DMA Done on RX Event Channel Interrupt Raw Status. This interrupt is raised when the DONE signal is sent to the I2CT from a DMA, indicating all the transactions have completed.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 14 | RESERVED | R | 0h | |
| 13 | TIMEOUTB | R | 0h | I2CT Timeout Counter B Interrupt Raw Status. This interrupt is raised when the SCL line is continously high for longer than the clock timeout period programmed by the TIMEOUT_CTL.TCNTLB register.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

**Table 30-62. RIS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 12 | TIMEOUTA | R | 0h | I2CT Timeout Counter A Interrupt Raw Status. This interrupt is raised when the SCL line is continously low for longer than the clock timeout period programmed by the TIMEOUT_CTL.TCNTLA register.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 11 | PEC_RX_ERR | R | 0h | I2CT Receive PEC Error Interrupt Raw Status. This interrupt is raised when the PEC calculated by the I2CT does not match received PEC.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 10 | STOP | R | 0h | I2CT STOP Condition Interrupt Raw Status. This interrupt is raised when a STOP condition is detected on the bus.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 9 | START | R | 0h | I2CT START Condition Interrupt Raw Status. This interrupt is raised when a START or repeated START condition is detected on the bus.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 8 | GENCALL | R | 0h | I2CT General Call Interrupt Raw Status. This interrupt is raised when a general call frame, with an address of 0x00 and R/W bit of '0', is received.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 7 | RX_OVFL | R | 0h | I2CT RX FIFO Overflow Interrupt Raw Status. This interrupt is raised when the I2CT receiver tries to receive data, as a result of receiving an address match with a R/W bit of '0', but the I2CT RX FIFO is full.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 6 | TX_UNFL | R | 0h | I2CT TX FIFO Underflow Interrupt Raw Status. This interrupt is raised when the I2CT transmitter tries to send data, as a result of having received an address match with a R/W bit of '1', but the I2CT TX FIFO is empty.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 5 | TXEMPTY | R | 0h | I2CT TX FIFO Empty Interrupt Raw Status. This interrupt is raised when all data has been shifted out of the TX FIFO and the I2CT transmitter goes idle.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 4 | RXFULL | R | 0h | I2CT RX FIFO Full Interrupt Raw Status. This interrupt is raised when the RX FIFO is full.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 3 | TXTRG | R | 0h | I2CT Transmit Trigger Interrupt Raw Status. This interrupt is raised when the TX FIFO condition, programmed by the IFLS register, is met. when FIFO is not present, this indicates that the buffer is empty.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 2 | RXTRG | R | 0h | I2CT Receive Trigger Interrupt Raw Status. This interrupt is raised when the RX FIFO condition programmed by the IFLS register is met. When FIFO is not present, this indicates that the buffer is full.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 1 | TXDONE | R | 0h | I2CT Transmit Done Interrupt Raw Status. This interrupt is raised after each byte is transmitted.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

**Table 30-62. RIS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 0 | RXDONE | R | 0h | I2CT Receive Done Interrupt Raw Status. This interrupt is raised after each byte is received.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.2.6 MIS (Offset = 1038h) [Reset = 00000000h]

MIS is shown in Figure 30-63 and described in Table 30-63.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 30-63. MIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | RESERVED | RESERVED | ARBLOST | DMA_DONE_TX |
| R-0h | | | | R-0h | R-0h | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DMA_DONE_RX | RESERVED | TIMEOUTB | TIMEOUTA | PEC_RX_ERR | STOP | START | GENCALL |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RX_OVFL | TX_UNFL | TXEMPTY | RXFULL | TXTRG | RXTRG | TXDONE | RXDONE |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 30-63. MIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | RESERVED | R | 0h | |
| 19 | RESERVED | R | 0h | Reserved |
| 18 | RESERVED | R | 0h | Reserved |
| 17 | ARBLOST | R | 0h | I2CT Arbitration Lost Interrupt Masked Status. This interrupt is raised when another I2C target on the bus wins arbitration during the "Get UDID" sequence in the SMBus address resolution protocol.<br>0h = Interrupt did not occur or mask was not enabled<br>1h = Interrupt occurred |
| 16 | DMA_DONE_TX | R | 0h | I2CT DMA Done on TX Event Channel Interrupt Masked Status. This interrupt is raised when the DONE signal is sent to the I2CT from a DMA, indicating all the transactions have completed.<br>0h = Interrupt did not occur or mask was not enabled<br>1h = Interrupt occurred |
| 15 | DMA_DONE_RX | R | 0h | I2CT DMA Done on RX Event Channel Interrupt Masked Status. This interrupt is raised when the DONE signal is sent to the I2CT from a DMA, indicating all the transactions have completed.<br>0h = Interrupt did not occur or mask was not enabled<br>1h = Interrupt occurred |
| 14 | RESERVED | R | 0h | |
| 13 | TIMEOUTB | R | 0h | I2CT Timeout Counter B Interrupt Masked Status. This interrupt is raised when the SCL line is continously high for longer than the clock timeout period programmed by the TIMEOUT_CTL.TCNTLB register.<br>0h = Interrupt did not occur or mask was not enabled<br>1h = Interrupt occurred |
| 12 | TIMEOUTA | R | 0h | I2CT Timeout Counter A Interrupt Masked Status. This interrupt is raised when the SCL line is continously low for longer than the clock timeout period programmed by the TIMEOUT_CTL.TCNTLA register.<br>0h = Interrupt did not occur or mask was not enabled<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

**Table 30-63. MIS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 11 | PEC_RX_ERR | R | 0h | I2CT Receive PEC Error Interrupt Masked Status. This interrupt is raised when the PEC calculated by the I2CT does not match received PEC.<br>0h = Interrupt did not occur or mask was not enabled<br>1h = Interrupt occurred |
| 10 | STOP | R | 0h | I2CT STOP Condition Interrupt Masked Status. This interrupt is raised when a STOP condition is detected on the bus.<br>0h = Interrupt did not occur or mask was not enabled<br>1h = Interrupt occurred |
| 9 | START | R | 0h | I2CT START Condition Interrupt Masked Status. This interrupt is raised when a START or repeated START condition is detected on the bus.<br>0h = Interrupt did not occur or mask was not enabled<br>1h = Interrupt occurred |
| 8 | GENCALL | R | 0h | I2CT General Call Interrupt Masked Status. This interrupt is raised when a general call frame, with an address of 0x00 and R/W bit of '0', is received.<br>0h = Interrupt did not occur or mask was not enabled<br>1h = Interrupt occurred |
| 7 | RX_OVFL | R | 0h | I2CT RX FIFO Overflow Interrupt Masked Status. This interrupt is raised when the I2CT receiver tries to receive data, as a result of receiving an address match with a R/W bit of '0', but the I2CT RX FIFO is full.<br>0h = Interrupt did not occur or mask was not enabled<br>1h = Interrupt occurred |
| 6 | TX_UNFL | R | 0h | I2CT TX FIFO Underflow Interrupt Masked Status. This interrupt is raised when the I2CT transmitter tries to send data, as a result of having received an address match with a R/W bit of '1', but the I2CT TX FIFO is empty.<br>0h = Interrupt did not occur or mask was not enabled<br>1h = Interrupt occurred |
| 5 | TXEMPTY | R | 0h | I2CT TX FIFO Empty Interrupt Masked Status. This interrupt is raised when all data has been shifted out of the TX FIFO and the I2CT transmitter goes idle.<br>0h = Interrupt did not occur or mask was not enabled<br>1h = Interrupt occurred |
| 4 | RXFULL | R | 0h | I2CT RX FIFO Full Interrupt Masked Status. This interrupt is raised when the RX FIFO is full.<br>0h = Clear Interrupt Mask<br>1h = Interrupt occurred |
| 3 | TXTRG | R | 0h | I2CT Transmit Trigger Interrupt Masked Status. This interrupt is raised when the TX FIFO condition, programmed by the IFLS register, is met. when FIFO is not present, this indicates that the buffer is empty.<br>0h = Interrupt did not occur or mask was not enabled<br>1h = Interrupt occurred |
| 2 | RXTRG | R | 0h | I2CT Receive Trigger Interrupt Masked Status. This interrupt is raised when the RX FIFO condition programmed by the IFLS register is met. When FIFO is not present, this indicates that the buffer is full.<br>0h = Interrupt did not occur or mask was not enabled<br>1h = Interrupt occurred |
| 1 | TXDONE | R | 0h | I2CT Transmit Done Interrupt Masked Status. This interrupt is raised after each byte is transmitted.<br>0h = Interrupt did not occur or mask was not enabled<br>1h = Interrupt occurred |
| 0 | RXDONE | R | 0h | I2CT Receive Done Interrupt Masked Status. This interrupt is raised after each byte is received.<br>0h = Interrupt did not occur or mask was not enabled<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.2.7 ISET (Offset = 1040h) [Reset = 00000000h]

ISET is shown in Figure 30-64 and described in Table 30-64.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 30-64. ISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | RESERVED | RESERVED | ARBLOST | DMA_DONE_TX |
| R-0h | | | | R-0h | R-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DMA_DONE_RX | RESERVED | TIMEOUTB | TIMEOUTA | PEC_RX_ERR | STOP | START | GENCALL |
| W-0h | R-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RX_OVFL | TX_UNFL | TXEMPTY | RXFULL | TXTRG | RXTRG | TXDONE | RXDONE |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

**Table 30-64. ISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | RESERVED | R | 0h | |
| 19 | RESERVED | R | 0h | Reserved |
| 18 | RESERVED | R | 0h | Reserved |
| 17 | ARBLOST | W | 0h | Set I2CT Arbitration Lost Interrupt. This interrupt is raised when another I2C target on the bus wins arbitration during the "Get UDID" sequence in the SMBus address resolution protocol.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 16 | DMA_DONE_TX | W | 0h | Set I2CT DMA Done on TX Event Channel Interrupt. This interrupt is raised when the DONE signal is sent to the I2CT from a DMA, indicating all the transactions have completed.<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 15 | DMA_DONE_RX | W | 0h | Set I2CT DMA Done on RX Event Channel Interrupt. This interrupt is raised when the DONE signal is sent to the I2CT from a DMA, indicating all the transactions have completed.<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 14 | RESERVED | R | 0h | |
| 13 | TIMEOUTB | W | 0h | Set I2CT Timeout Counter B Interrupt. This interrupt is raised when the SCL line is continously high for longer than the clock timeout period programmed by the TIMEOUT_CTL.TCNTLB register.<br>0h = Writing 0 has no effect<br>1h = Set interrupt |



Copyright © 2026 Texas Instruments Incorporated

**Table 30-64. ISET Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 12 | TIMEOUTA | W | 0h | Set I2CT Timeout Counter A Interrupt. This interrupt is raised when the SCL line is continously low for longer than the clock timeout period programmed by the TIMEOUT_CTL.TCNTLA register.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 11 | PEC_RX_ERR | W | 0h | Set I2CT Receive PEC Error Interrupt. This interrupt is raised when the PEC calculated by the I2CT does not match received PEC.<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 10 | STOP | W | 0h | Set I2CT STOP Condition Interrupt. This interrupt is raised when a STOP condition is detected on the bus.<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 9 | START | W | 0h | Set I2CT START Condition Interrupt. This interrupt is raised when a START or repeated START condition is detected on the bus.<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 8 | GENCALL | W | 0h | Set I2CT General Call Interrupt. This interrupt is raised when a general call frame, with an address of 0x00 and R/W bit of '0', is received.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 7 | RX_OVFL | W | 0h | Set I2CT RX FIFO Overflow Interrupt. This interrupt is raised when the I2CT receiver tries to receive data, as a result of receiving an address match with a R/W bit of '0', but the I2CT RX FIFO is full.<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 6 | TX_UNFL | W | 0h | Set I2CT TX FIFO Underflow Interrupt. This interrupt is raised when the I2CT transmitter tries to send data, as a result of having received an address match with a R/W bit of '1', but the I2CT TX FIFO is empty.<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 5 | TXEMPTY | W | 0h | Set I2CT TX FIFO Empty Interrupt. This interrupt is raised when all data has been shifted out of the TX FIFO and the I2CT transmitter goes idle.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 4 | RXFULL | W | 0h | Set I2CT RX FIFO Full Interrupt. This interrupt is raised when the RX FIFO is full.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt |
| 3 | TXTRG | W | 0h | Set I2CT Transmit Trigger Interrupt. This interrupt is raised when the TX FIFO condition, programmed by the IFLS register, is met. when FIFO is not present, this indicates that the buffer is empty.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt |
| 2 | RXTRG | W | 0h | Set I2CT Receive Trigger Interrupt. This interrupt is raised when the RX FIFO condition programmed by the IFLS register is met. When FIFO is not present, this indicates that the buffer is full.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt |
| 1 | TXDONE | W | 0h | Set I2CT Transmit Done Interrupt. This interrupt is raised after each byte is transmitted.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |



Copyright © 2026 Texas Instruments Incorporated

**Table 30-64. ISET Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 0 | RXDONE | W | 0h | Set I2CT Receive Done Interrupt. This interrupt is raised after each byte is received.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.2.8 ICLR (Offset = 1048h) [Reset = 00000000h]

ICLR is shown in Figure 30-65 and described in Table 30-65.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

**Figure 30-65. ICLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | RESERVED | RESERVED | ARBLOST | DMA_DONE_TX |
| R-0h | | | | R-0h | R-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DMA_DONE_RX | RESERVED | TIMEOUTB | TIMEOUTA | PEC_RX_ERR | STOP | START | GENCALL |
| W-0h | R-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RX_OVFL | TX_UNFL | TXEMPTY | RXFULL | TXTRG | RXTRG | TXDONE | RXDONE |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

**Table 30-65. ICLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | RESERVED | R | 0h | |
| 19 | RESERVED | R | 0h | Reserved |
| 18 | RESERVED | R | 0h | Reserved |
| 17 | ARBLOST | W | 0h | Clear I2CT Arbitration Lost Interrupt. This interrupt is raised when another I2C target on the bus wins arbitration during the "Get UDID" sequence in the SMBus address resolution protocol.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 16 | DMA_DONE_TX | W | 0h | Clear I2CT DMA Done on TX Event Channel Interrupt. This interrupt is raised when the DONE signal is sent to the I2CT from a DMA, indicating all the transactions have completed.<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 15 | DMA_DONE_RX | W | 0h | Clear I2CT DMA Done on RX Event Channel Interrupt. This interrupt is raised when the DONE signal is sent to the I2CT from a DMA, indicating all the transactions have completed.<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 14 | RESERVED | R | 0h | |
| 13 | TIMEOUTB | W | 0h | Clear I2CT Timeout Counter B Interrupt. This interrupt is raised when the SCL line is continously high for longer than the clock timeout period programmed by the TIMEOUT_CTL.TCNTLB register.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 12 | TIMEOUTA | W | 0h | Clear I2CT Timeout Counter A Interrupt. This interrupt is raised when the SCL line is continously low for longer than the clock timeout period programmed by the TIMEOUT_CTL.TCNTLA register.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |



Copyright © 2026 Texas Instruments Incorporated

**Table 30-65. ICLR Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 11 | PEC_RX_ERR | W | 0h | Clear I2CT Receive PEC Error Interrupt. This interrupt is raised when the PEC calculated by the I2CT does not match received PEC.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 10 | STOP | W | 0h | Clear I2CT STOP Condition Interrupt. This interrupt is raised when a STOP condition is detected on the bus.<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 9 | START | W | 0h | Clear I2CT START Condition Interrupt. This interrupt is raised when a START or repeated START condition is detected on the bus.<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 8 | GENCALL | W | 0h | Clear I2CT General Call Interrupt. This interrupt is raised when a general call frame, with an address of 0x00 and R/W bit of '0', is received.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 7 | RX_OVFL | W | 0h | Clear I2CT RX FIFO Overflow Interrupt. This interrupt is raised when the I2CT receiver tries to receive data, as a result of receiving an address match with a R/W bit of '0', but the I2CT RX FIFO is full.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 6 | TX_UNFL | W | 0h | Clear I2CT TX FIFO Underflow Interrupt. This interrupt is raised when the I2CT transmitter tries to send data, as a result of having received an address match with a R/W bit of '1', but the I2CT TX FIFO is empty.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 5 | TXEMPTY | W | 0h | Clear I2CT TX FIFO Empty Interrupt. This interrupt is raised when all data has been shifted out of the TX FIFO and the I2CT transmitter goes idle.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 4 | RXFULL | W | 0h | Clear I2CT RX FIFO Full Interrupt. This interrupt is raised when the RX FIFO is full.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 3 | TXTRG | W | 0h | Clear I2CT Transmit Trigger Interrupt. This interrupt is raised when the TX FIFO condition, programmed by the IFLS register, is met. when FIFO is not present, this indicates that the buffer is empty.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 2 | RXTRG | W | 0h | Clear I2CT Receive Trigger Interrupt. This interrupt is raised when the RX FIFO condition programmed by the IFLS register is met. When FIFO is not present, this indicates that the buffer is full.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 1 | TXDONE | W | 0h | Clear I2CT Transmit Done Interrupt. This interrupt is raised after each byte is transmitted.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 0 | RXDONE | W | 0h | Clear I2CT Receive Done Interrupt. This interrupt is raised after each byte is received.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.2.9 IMASK (Offset = 1058h) [Reset = 00000000h]

IMASK is shown in Figure 30-66 and described in Table 30-66.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

**Figure 30-66. IMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | RXTRG | RESERVED | |
| R-0h | | | | | R/W-0h | R-0h | |

**Table 30-66. IMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | RESERVED | R | 0h | |
| 2 | RXTRG | R/W | 0h | I2CT Receive DMA Trigger Mask. The DMA is triggered when the RX FIFO condition programmed by the IFLS register is met. When FIFO is not present, this indicates that the buffer is full.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 1-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.2.10 RIS (Offset = 1060h) [Reset = 00000000h]

RIS is shown in Figure 30-67 and described in Table 30-67.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Figure 30-67. RIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | RXTRG | RESERVED | |
| R-0h | | | | | R-0h | R-0h | |

**Table 30-67. RIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | RESERVED | R | 0h | |
| 2 | RXTRG | R | 0h | I2CT Receive DMA Trigger Raw Status. The DMA is triggered when the RX FIFO condition programmed by the IFLS register is met. When FIFO is not present, this indicates that the buffer is full.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 1-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.2.11 MIS (Offset = 1068h) [Reset = 00000000h]

MIS is shown in Figure 30-68 and described in Table 30-68.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 30-68. MIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | RXTRG | RESERVED | |
| R-0h | | | | | R-0h | R-0h | |

**Table 30-68. MIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | RESERVED | R | 0h | |
| 2 | RXTRG | R | 0h | I2CT Receive DMA Trigger Masked Status. The DMA is triggered when the RX FIFO condition programmed by the IFLS register is met. When FIFO is not present, this indicates that the buffer is full.<br>0h = Interrupt did not occur or mask was not enabledk<br>1h = Interrupt occurred |
| 1-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.2.12 ISET (Offset = 1070h) [Reset = 00000000h]

ISET is shown in Figure 30-69 and described in Table 30-69.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 30-69. ISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | RXTRG | RESERVED | |
| R-0h | | | | | W-0h | R-0h | |

**Table 30-69. ISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | RESERVED | R | 0h | |
| 2 | RXTRG | W | 0h | Set I2CT Receive DMA Trigger. The DMA is triggered when the RX FIFO condition programmed by the IFLS register is met. When FIFO is not present, this indicates that the buffer is full.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt Mask |
| 1-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.2.13 IMASK (Offset = 1088h) [Reset = 00000000h]

IMASK is shown in Figure 30-70 and described in Table 30-70.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

**Figure 30-70. IMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | TXTRG | RESERVED | | |
| R-0h | | | | R/W-0h | R-0h | | |

**Table 30-70. IMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | TXTRG | R/W | 0h | I2CT Transmit DMA Trigger Mask. The DMA is triggered when the TX FIFO condition programmed by the IFLS register is met. When FIFO is not present, this indicates that the buffer is empty.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 2-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.2.14 RIS (Offset = 1090h) [Reset = 00000000h]

RIS is shown in Figure 30-71 and described in Table 30-71.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Figure 30-71. RIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | TXTRG | RESERVED | | |
| R-0h | | | | R-0h | R-0h | | |

**Table 30-71. RIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | TXTRG | R | 0h | I2CT Transmit DMA Trigger Raw Status. The DMA is triggered when the TX FIFO condition programmed by the IFLS register is met. When FIFO is not present, this indicates that the buffer is empty.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 2-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.2.15 MIS (Offset = 1098h) [Reset = 00000000h]

MIS is shown in Figure 30-72 and described in Table 30-72.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 30-72. MIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | TXTRG | RESERVED | | |
| R-0h | | | | R-0h | R-0h | | |

**Table 30-72. MIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | TXTRG | R | 0h | I2CT Transmit DMA Trigger Masked Status. The DMA is triggered when the TX FIFO condition programmed by the IFLS register is met. When FIFO is not present, this indicates that the buffer is empty.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 2-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.2.16 ISET (Offset = 10A0h) [Reset = 00000000h]

ISET is shown in Figure 30-73 and described in Table 30-73.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 30-73. ISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | TXTRG | RESERVED | | |
| R-0h | | | | W-0h | R-0h | | |

**Table 30-73. ISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | TXTRG | W | 0h | Set I2CT Transmit DMA Trigger. The DMA is triggered when the TX FIFO condition programmed by the IFLS register is met. When FIFO is not present, this indicates that the buffer is empty.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 2-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.2.17 INTCTL (Offset = 10E4h) [Reset = 00000000h]

INTCTL is shown in Figure 30-74 and described in Table 30-74.

Return to the Summary Table.

Interrupt Control Register.

**Figure 30-74. INTCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | INTEVAL |
| R-0h | | | | | | | W-0h |

**Table 30-74. INTCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | INTEVAL | W | 0h | Writing a 1 to this field re-evaluates the interrupt sources.<br>0h = Writing 0 has no effect<br>1h = Interrupt Eval |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.2.18 CTR (Offset = 1100h) [Reset = 00300404h]

CTR is shown in Figure 30-75 and described in Table 30-75.

Return to the Summary Table.

I2CT Control Register.

**Figure 30-75. CTR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | WUEN | CLKSTRETCH | RESERVED | | | |
| R-0h | | R/W-1h | R/W-1h | R-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | SUSPEND | RESERVED | EN_DEFDEVADR | EN_ALRESPADR |
| R-0h | | | | R/W-0h | R-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| EN_DEFHOSTADR | RXFULL_ON_REQ | TXWAIT_STALE_TXFIFO | TXTRIG_TXMODE | TXEMPTY_ON_TREQ | RESERVED | GENCALL | ENABLE |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | R/W-0h | R/W-0h |

**Table 30-75. CTR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-22 | RESERVED | R | 0h | |
| 21 | WUEN | R/W | 1h | Target Wakeup Enable.<br>0h = When 0, the Target is not allowed to clock stretch on START detection<br>1h = When 1, the Target is allowed to clock stretch on START detection and wait for faster clock to be available. This allows clean wake up support for I2C in low power mode use cases |
| 20 | CLKSTRETCH | R/W | 1h | Clock Stretching Enable. The I2CT will automatically clock stretch on the SCL line if the TX FIFO/buffer is empty in transmit mode or the RX FIFO/buffer is full in receive mode.<br>0h = clock stretching is disabled<br>1h = clock stretching is enabled |
| 19-12 | RESERVED | R | 0h | |
| 11 | SUSPEND | R/W | 0h | Suspend External Communication. When this bit is set, I2CT communication on the external bus is suspended after the I2CT FSM goes idle (SR.BUSY reads '0').<br>0h = Functional mode resumed<br>1h = External communication suspended |
| 10 | RESERVED | R | 0h | |
| 9 | EN_DEFDEVADR | R/W | 0h | Enable Default Device Address.<br>0h = When this bit is 0, the default device address is not matched. NOTE: it may still be matched if programmed inside OAR/OAR2.<br>1h = When this bit is 1, default device address of 7'h110_0001 is always matched by the Target address match logic. |
| 8 | EN_ALRESPADR | R/W | 0h | Enable Alert Response Address.<br>0h = When this bit is 0, the alert response address is not matched. NOTE: it may still be matched if programmed inside OAR/OAR2<br>1h = When this bit is 1, alert response address of 7'h000_1100 is always matched by the Target address match logic. |



Copyright © 2026 Texas Instruments Incorporated

**Table 30-75. CTR Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7 | EN_DEFHOSTADR | R/W | 0h | Enable Default Host Address.<br>0h = When this bit is 0, the default host address is not matched<br>NOTE: it may still be matched if programmed inside OAR/OAR2<br>1h = When this bit is 1, default host address of 7'h000_1000 is always matched by the Target address match logic. |
| 6 | RXFULL_ON_RREQ | R/W | 0h | RX Full Interrupt on RREQ. This bit setting determines the behavior of the RXFULL interrupt.<br>0h = When 0, RIS.RXFULL will be set when only the Target RX FIFO is full. This allows the RXFULL interrupt to be used to indicate that the I2C bus is being clock stretched and that the FW must either read the RX FIFO or ACK/NACK the current RX byte.<br>1h = When 1, RIS.RXFULL will be set when the I2CT FSM is in the RX_WAIT or RX_ACK_WAIT states which occurs when the I2C transaction is clock stretched because the RX FIFO is full or the ACKOEN has been set and the FSM is waiting for FW to ACK/NACK the current byte. |
| 5 | TXWAIT_STALE_TXFIFO | R/W | 0h | TX Transfer Wait Configuration. This bit setting determines the behavior of the I2CT transmitter, specifically whether or not the transmitter waits when there is stale data in the TX FIFO. This prevents stale bytes left in the TX FIFO from automatically being sent on the next I2C packet. Note: This setting should only be enabled when TXEMPTY_ON_TREQ is set to prevent the I2CT FSM from waiting for TX FIFO data without an interrupt notification when there is stale FIFO data.<br>0h = When 0, the TX FIFO empty signal to the Target State Machine indicates that the TX FIFO is empty.<br>1h = When 1, the TX FIFO empty signal to the Target State Machine will indicate that the TX FIFO is empty or that the TX FIFO data is stale. The TX FIFO data is determined to be stale when there is data in the TX FIFO when the Target State Machine leaves the TXMODE as defined in the SR register. A stale TX FIFO occurs if a Stop or timeout condition occurs when there are bytes left in the TX FIFO. |
| 4 | TXTRIG_TXMODE | R/W | 0h | TX Trigger when I2CT FSM is in TX Mode. This bit setting determines the behavior of the TXTRG DMA trigger.<br>0h = No special behavior<br>1h = When 1, RIS.TXTRG is set when the I2CT TX FIFO has reached the trigger level AND the I2CT FSM is in the TX mode, as defined in the SR register. When cleared, RIS.TXTRG is set when the I2CT TX FIFO is at or above the trigger level. This setting can be used to hold off the TX DMA until a transaction starts. This allows the DMA to be configured when the I2CT is idle but have it wait until the transaction starts to load the I2CT TX FIFO, so it can load from a memory buffer that might be changing over time. |
| 3 | TXEMPTY_ON_TREQ | R/W | 0h | TX Empty Interrupt on TREQ. This bit setting determines the behavior of the TXEMPTY interrupt.<br>0h = When 0, RIS.TXEMPTY will be set when only the I2CT TX FIFO is empty. This allows the TXEMPTY interrupt to be used to indicate that the I2C bus is being clock stretched and that I2CT TX data is required.<br>1h = When 1, RIS.TXEMPTY will be set when the I2CT FSM is in the TX_WAIT state which occurs when the TX FIFO is empty AND the I2C transaction is being clock stretched waiting for data to be written to the TX FIFO. |
| 2 | RESERVED | R | 0h | |
| 1 | GENCALL | R/W | 0h | General Call/Response Enable. Modify only when UCSWRST = 1.<br>0b = Do not respond to a general call 1b = Respond to a general call<br>0h = Do not respond to a general call<br>1h = Respond to a general call |
| 0 | ENABLE | R/W | 0h | I2CT Enable. Setting this bit enables the module.<br>0h = Disables module operation.<br>1h = Enables module operation. |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.2.19 ACKCTL (Offset = 1104h) [Reset = 00000000h]

ACKCTL is shown in Figure 30-76 and described in Table 30-76.

Return to the Summary Table.

I2CT Acknowledge Control Register. This register enables the I2CT to Not Acknowledge (NACK) for invalid data/commands or Acknowledge (ACK) for valid data/commands. The I2C SCL is pulled low after the last data bit until this register is written.

**Figure 30-76. ACKCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | ACKOEN_ON_PECDONE | ACKOEN_ON_PECNEXT | ACKOEN_ON_START | ACKOVAL | ACKOEN |
| R-0h | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

**Table 30-76. ACKCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-5 | RESERVED | R | 0h | |
| 4 | ACKOEN_ON_PECDONE | R/W | 0h | Acknowledge Override Enable on PEC Done. When set, this bit will automatically turn on target acknowledge enable field following the ACK/NACK of the received PEC byte. Status bit ACKOEN will reflect a value of '1' when automatic override enable is used.<br>0h = No special behavior<br>1h = When set this bit will automatically turn on the Target ACKOEN field following the ACK/NACK of the received PEC byte. |
| 3 | ACKOEN_ON_PECNEXT | R/W | 0h | Acknowledge Override Enable on PEC Next. When set this bit will automatically turn on the I2CT acknowledge override following a ACK/NACK of the byte received just prior to the PEC byte. However, setting ACKCTL.ACKOEN bit will not automatically be ACKed/NACKed by the FSM and firmware must perform this function by writing Target.ACKCTL register. Status bit ACKOEN will reflect a value of '1' when automatic override enable is used.<br>0h = No special behavior<br>1h = When set this bit will automatically turn on the Target ACKOEN field following the ACK/NACK of the byte received just prior to the PEC byte. Note that when ACKOEN is set the PEC byte will not automatically be ACKed/NACKed by the State Machine and FW must perform this function by writing ACKCTL. |
| 2 | ACKOEN_ON_START | R/W | 0h | Acknowledge Override Enable on START. When set this bit will automatically enable target override acknowledge following a Start Condition. Status bit ACKOEN will reflect a value of '1' when automatic override enable is used.<br>0h = No special behavior<br>1h = When set this bit will automatically turn on the Target ACKOEN field following a Start Condition. |



Copyright © 2026 Texas Instruments Incorporated

**Table 30-76. ACKCTL Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 1 | ACKOVAL | R/W | 0h | ACK Override Value. Note: Override control is not applicable to Address frame. Bytes following address frame can be acknowledged using ACKOVAL.<br>0h = An ACK is sent indicating valid data or command.<br>1h = A NACK is sent indicating invalid data or command. |
| 0 | ACKOEN | R/W | 0h | I2CT ACK Override Enable. Determines if the I2CT clock stretches and waits for the application to manually ACK/NACK each received byte using the ACKCTL.ACKOVAL bit, or if the ACK/NACK is done automatically. Read SR.ACKOEN to check current status of this bit.<br>0h = A response in not provided.<br>1h = An ACK or NACK is sent according to the value written to the ACKOVAL bit. |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.2.20 SR (Offset = 1108h) [Reset = 00000000h]

SR is shown in Figure 30-77 and described in Table 30-77.

Return to the Summary Table.

I2CT Status Register. This status register indicates the current state of the I2CT.

**Figure 30-77. SR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | ADDRMATCH | |
| R-0h | | | | | | R-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| ADDRMATCH | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| ACKOEN | TXFF | TXFE | RXFF | RXFE | TXCLR | RXCLR | STALE_TXFIFO |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TXMODE | BUSBSY | QCMDRW | QCMDST | OAR2SEL | RXMODE | TREQ | RREQ |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 30-77. SR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | RESERVED | R | 0h | |
| 25-16 | ADDRMATCH | R | 0h | Address Match Status. Indicates the address for which Target address match happened.<br>0h = Minimum Value<br>3FFh = Maximum Value |
| 15 | ACKOEN | R | 0h | ACK Override Enable Status.<br>0h = ACK override is disabled<br>1h = ACK override was enabled in design |
| 14 | TXFF | R | 0h | Transmit FIFO/Buffer Full Status.<br>0h = Transmit FIFO is not full<br>1h = Transmit FIFO is full |
| 13 | TXFE | R | 1h | Transmit FIFO/Buffer Empty Status.<br>0h = TX FIFO is not empty.<br>1h = TX FIFO is empty |
| 12 | RXFF | R | 0h | RX FIFO Full Status.<br>0h = RX FIFO is not full<br>1h = RX FIFO is full |
| 11 | RXFE | R | 1h | RX FIFO/Buffer Empty Status.<br>0h = RX FIFO is not empty.<br>1h = RX FIFO is empty |
| 10 | TXCLR | R | 0h | TX FIFO Clear Status.<br>0h = FIFO is not cleared<br>1h = FIFO clear is complete |
| 9 | RXCLR | R | 0h | RX FIFO Clear Status.<br>0h = FIFO is not cleared<br>1h = FIFO clear is complete |
| 8 | STALE_TXFIFO | R | 0h | Stale TX FIFO Status.<br>0h = TX FIFO is not stale<br>1h = The TX FIFO is stale. This occurs when the TX FIFO was not emptied during the previous I2C transaction. |



Copyright © 2026 Texas Instruments Incorporated

**Table 30-77. SR Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7 | TXMODE | R | 0h | I2CT FSM is in TX Mode.<br>0h = State Machine is not in TX_DATA, TX_WAIT, TX_ACK or ADDR_ACK state with the bus direction set to read.<br>1h = State Machine is in TX_DATA, TX_WAIT, TX_ACK or ADDR_ACK state with the bus direction set to read. |
| 6 | BUSBSY | R | 0h | I2C Bus Busy Status.<br>0h = The I2C Bus is not busy<br>1h = The I2C Bus is busy. This is cleared on a timeout. |
| 5 | QCMDRW | R | 0h | Quick Command Read / Write Status. This bit only has meaning when the QCMDST bit is set. Value Description: 0: Quick command was a write 1: Quick command was a read<br>0h = Quick command was a write<br>1h = Quick command was a read |
| 4 | QCMDST | R | 0h | Quick Command Status. Value Description: 0: The last transaction was a normal transaction or a transaction has not occurred. 1: The last transaction was a Quick Command transaction<br>0h = The last transaction was a normal transaction or a transaction has not occurred.<br>1h = The last transaction was a Quick Command transaction. |
| 3 | OAR2SEL | R | 0h | OAR2 Address Matched Status. This bit gets re-evaluated after every address comparison.<br>0h = Either the OAR2 address is not matched or the match is in legacy mode.<br>1h = OAR2 address matched and ACKed by the Target. |
| 2 | RXMODE | R | 0h | I2CT FSM is in RX Mode.<br>0h = The Target State Machine is not in the RX_DATA, RX_ACK, RX_WAIT, RX_ACK_WAIT or ADDR_ACK state with the bus direction set to write.<br>1h = The Target State Machine is in the RX_DATA, RX_ACK, RX_WAIT, RX_ACK_WAIT or ADDR_ACK state with the bus direction set to write. |
| 1 | TREQ | R | 0h | Transmit Request Status.<br>0h = No outstanding transmit request.<br>1h = I2C Target is addressed as a transmitter and is using clock stretching to delay the Controller until data has been written to the TXDATA FIFO (TX FIFO is empty). |
| 0 | RREQ | R | 0h | Receive Request Status.<br>0h = No outstanding receive data.<br>1h = Module has outstanding receive data and is using clock stretching to delay the Controller until the data has been read from the RXDATA FIFO (RX FIFO is full). |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.2.21 IFLS (Offset = 110Ch) [Reset = 00000022h]

IFLS is shown in Figure 30-78 and described in Table 30-78.

Return to the Summary Table.

I2CT Interrupt FIFO Level Select Register. The IFLS register is the interrupt FIFO level select register. You can use this register to define the levels at which the TX, RX and timeout interrupt flags are triggered. The interrupts are generated based on a transition through a level rather than being based on the level. That is, the interrupts are generated when the fill level progresses through the trigger level. For example, if the receive trigger level is set to the half-way mark, the interrupt is triggered when the RX FIFO is filled with two or more characters. Out of reset, the TXIFLSEL and RXIFLSEL bits are configured so that the FIFOs trigger an interrupt at the half-way mark.

**Figure 30-78. IFLS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RXCLR | RXIFLSEL | | | TXCLR | TXIFLSEL | | |
| R/W-0h | R/W-2h | | | R/W-0h | R/W-2h | | |

**Table 30-78. IFLS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7 | RXCLR | R/W | 0h | RX FIFO Clear. Setting this bit will clear the RX FIFO contents.<br>0h = Disable FIFO clear<br>1h = Enable FIFO Clear |
| 6-4 | RXIFLSEL | R/W | 2h | RX FIFO Level Select for generating events (interrupt/DMA). Note: for undefined settings the default configuration is used.<br>1h = RX FIFO >= 1/4 full<br>2h = RX FIFO >= 1/2 full (default)<br>3h = RX FIFO >= 3/4 full<br>4h = Opposite of empty<br>5h = RX FIFO is full<br>6h = RX_FIFO >= (MAX_FIFO_LEN -1)<br>7h = RX_FIFO <= 1 |
| 3 | TXCLR | R/W | 0h | TX FIFO Clear. Setting this bit will clear the TX FIFO contents.<br>0h = Disable FIFO clear<br>1h = Enable FIFO Clear |
| 2-0 | TXIFLSEL | R/W | 2h | TX FIFO Level Select for generating events (interrupt/DMA). Note: for undefined settings the default configuration is used.<br>1h = TX FIFO <= 3/4 empty<br>2h = TX FIFO <= 1/2 empty (default)<br>3h = TX FIFO <= 1/4 empty<br>4h = Opposite of full<br>5h = TX FIFO is empty<br>6h = TX FIFO <= 1<br>7h = TX_FIFO >= (MAX_FIFO_LEN -1) |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.2.22 GFCTL (Offset = 1118h) [Reset = 00000000h]

GFCTL is shown in Figure 30-79 and described in Table 30-79.

Return to the Summary Table.

I2CT Glitch Filter Control Register. This register controls the glitch filter on the SCL and SDA lines

**Figure 30-79. GFCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | AGFEN |
| R-0h | | | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | DGFSEL | | |
| R-0h | | | | | R/W-0h | | |

**Table 30-79. GFCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-9 | RESERVED | R | 0h | |
| 8 | AGFEN | R/W | 0h | Analog Glitch Suppression Enable.<br>0h = Analog Glitch Filter disable<br>1h = Analog Glitch Filter enable |
| 7-3 | RESERVED | R | 0h | |
| 2-0 | DGFSEL | R/W | 0h | Glitch Suppression Pulse Width. This field controls the pulse width select for glitch suppression on the SCL and SDA lines. The following values are the glitch suppression values in terms of functional clocks. (Core Domain only)<br>0h = Bypass<br>1h = 1 clock<br>2h = 2 clocks<br>3h = 3 clocks<br>4h = 4 clocks<br>5h = 8 clocks<br>6h = 16 clocks<br>7h = 31 clocks |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.2.23 TXDATA (Offset = 1120h) [Reset = 00000000h]

TXDATA is shown in Figure 30-80 and described in Table 30-80.

Return to the Summary Table.

I2CT Transmit Data Register. This register is the transmit data register (the interface to the FIFOs). For transmitted data, if the FIFO is enabled, data written to this location is pushed onto the TX FIFO. If the FIFO is disabled, data is stored in the transmitter holding register (the bottom word of the TX FIFO).

**Figure 30-80. TXDATA**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | DATA | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | W-0h | | | | | | | |

**Table 30-80. TXDATA Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | DATA | W | 0h | Transmit Data. This byte contains the data to be transferred during the next transaction.<br>0h = Smallest value<br>FFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.2.24 RXDATA (Offset = 1124h) [Reset = 00000000h]

RXDATA is shown in Figure 30-81 and described in Table 30-81.

Return to the Summary Table.

I2CT Receive Data Register. This field contains the current byte being read in the RX FIFO stack. If the FIFO is disabled, the data byte and status are stored in the receiving holding register (the bottom word of the RX FIFO). The received data can be retrieved by reading this register.

**Figure 30-81. RXDATA**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | DATA | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | |

**Table 30-81. RXDATA Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | DATA | R | 0h | Received Data. This field contains the last received data.<br>0h = Smallest value<br>FFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.2.25 PECSR (Offset = 1128h) [Reset = 00000000h]

PECSR is shown in Figure 30-82 and described in Table 30-82.

Return to the Summary Table.

I2CT PEC Status Register.

**Figure 30-82. PECSR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | PECSTS_ERROR | PECSTS_CHECK |
| R-0h | | | | | | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | PECBYTECNT |
| R-0h | | | | | | | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PECBYTECNT | | | | | | | |
| R-0h | | | | | | | |

**Table 30-82. PECSR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-18 | RESERVED | R | 0h | |
| 17 | PECSTS_ERROR | R | 0h | PEC Status Error. This status bit indicates if a PEC check error occurred in the transaction that occurred before the last STOP. This bit is cleared automatically by hardware when the next STOP observed on the line.<br>0h = Indicates PEC check error did not occur in the transaction that occurred before the last Stop<br>1h = Indicates PEC check error occurred in the transaction that occurred before the last Stop |
| 16 | PECSTS_CHECK | R | 0h | PEC Status Check. This status bit indicates if the PEC was checked in the transaction that occurred before the last STOP. This bit is cleared automatically by hardware when the next STOP observed on the line.<br>0h = Indicates PEC was not checked in the transaction that occurred before the last Stop<br>1h = Indicates PEC was checked in the transaction that occurred before the last Stop |
| 15-9 | RESERVED | R | 0h | |
| 8-0 | PECBYTECNT | R | 0h | PEC Byte Count. This is the current PEC Byte Count of the I2CT FSM.<br>0h = Minimum Value<br>1FFh = Maximum Value |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.2.26 OAR2 (Offset = 1148h) [Reset = 00000000h]

OAR2 is shown in Figure 30-83 and described in Table 30-83.

Return to the Summary Table.

I2CT Own Address 2 Register. This register consists of seven address bits that identify the alternate address for the I2C device on the I2C bus.

**Figure 30-83. OAR2**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | OAR2_MASK | | | | | | |
| R-0h | R/W-0h | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| OAR2EN | OAR2 | | | | | | |
| R/W-0h | R/W-0h | | | | | | |

**Table 30-83. OAR2 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-23 | RESERVED | R | 0h | |
| 22-16 | OAR2_MASK | R/W | 0h | Own Address 2 Mask. This field specifies bits A6 through A0 of the Target address. The bits with value ‘1’ in OAR2.OAR2_MASK field will make the corresponding incoming address bits to match by default regardless of the value inside OAR2.OAR2 i.e. corresponding OAR2.OAR2 bit is a don’t care.<br>0h = Minimum Value<br>7Fh = Maximum Value |
| 15-8 | RESERVED | R | 0h | |
| 7 | OAR2EN | R/W | 0h | Own Address 2 Enable.<br>0h = The alternate address is disabled.<br>1h = Enables the use of the alternate address in the OAR2 field. |
| 6-0 | OAR2 | R/W | 0h | Own Address 2. This field specifies the alternate OAR2 address.<br>0h = Smallest value<br>7Fh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.2.27 OAR (Offset = 114Ch) [Reset = 00004000h]

OAR is shown in Figure 30-84 and described in Table 30-84.

Return to the Summary Table.

I2CT Own Address Register. This register consists of seven address bits that identify the I2C device on the I2C bus.

**Figure 30-84. OAR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MODE | OAREN | RESERVED | | | | OAR | |
| R/W-0h | R/W-1h | R-0h | | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| OAR | | | | | | | |
| R/W-0h | | | | | | | |

**Table 30-84. OAR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15 | MODE | R/W | 0h | Addressing Mode Select. This bit selects the addressing mode to be used. When 0, 7-bit addressing is used. When 1, 10-bit addressing is used.<br>0h = Enable 7-bit addressing<br>1h = Enable 10-bit addressing |
| 14 | OAREN | R/W | 1h | Own Address Enable.<br>0h = Disable OAR address<br>1h = Enable OAR address |
| 13-10 | RESERVED | R | 0h | |
| 9-0 | OAR | R/W | 0h | Own Address. This field specifies bits A9 through A0 of the Target address. In 7-bit addressing mode as selected by OAR.MODE bit, the top 3 bits are don't care<br>0h = Smallest value<br>3FFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.2.28 TIMEOUT_CNT (Offset = 1150h) [Reset = 00020002h]

TIMEOUT_CNT is shown in Figure 30-85 and described in Table 30-85.

Return to the Summary Table.

I2CT Timeout Counter. This register contains the upper 8 bits of a 12-bit current counter values for Timeout Counter A and B. The lower four bits of the counter are not user visible and are always 0h.

**Figure 30-85. TIMEOUT_CNT**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | TCNTB | | | | | | | | RESERVED | | | | | | | | TCNTA | | | | | | | |
| R-0h | | | | | | | | R-2h | | | | | | | | R-0h | | | | | | | | R-2h | | | | | | | |

**Table 30-85. TIMEOUT_CNT Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | RESERVED | R | 0h | |
| 23-16 | TCNTB | R | 2h | Timeout Counter B Current Count. This field contains the upper 8 bits of a 12-bit current counter for timeout counter B<br>0h = Smallest Value<br>FFh = Highest possible value |
| 15-8 | RESERVED | R | 0h | |
| 7-0 | TCNTA | R | 2h | Timeout Counter A Current Count. This field contains the upper 8 bits of a 12-bit current counter for timeout counter A<br>0h = Smallest Value<br>FFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.2.29 TIMEOUT_CTL (Offset = 1154h) [Reset = 00020002h]

TIMEOUT_CTL is shown in Figure 30-86 and described in Table 30-86.

Return to the Summary Table.

I2CT Timeout Control. This register contains controls for Timeout Counters A and B.

**Figure 30-86. TIMEOUT_CTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| TCNTBEN | RESERVED | | | | | | |
| R/W-0h | R-0h | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TCNTLB | | | | | | | |
| R/W-2h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TCNTAEN | RESERVED | | | | | | |
| R/W-0h | R-0h | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TCNTLA | | | | | | | |
| R/W-2h | | | | | | | |

**Table 30-86. TIMEOUT_CTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | TCNTBEN | R/W | 0h | Timeout Counter B Enable.<br>0h = Disable Timeout Counter B<br>1h = Enable Timeout Counter B |
| 30-24 | RESERVED | R | 0h | |
| 23-16 | TCNTLB | R/W | 2h | Timeout Count B Load. Counter B is used for SCL High Detection. This field contains the upper 8 bits of a 12-bit pre-load value for the Timeout Counter B count. NOTE: The value of CNTLB must be greater than 1h. Each count is equal to 1* clock period. For example, with 10MHz functional clock one timeout period will be equal to1*100ns.<br>0h = Smallest possible value<br>FFh = Highest possible value |
| 15 | TCNTAEN | R/W | 0h | Timeout Counter A Enable.<br>0h = Disable Timeout Counter B<br>1h = Enable Timeout Counter B |
| 14-8 | RESERVED | R | 0h | |
| 7-0 | TCNTLA | R/W | 2h | Timeout Counter A Load Value. Counter A is used for SCL low detection. This field contains the upper 8 bits of a 12-bit pre-load value for the Timeout A count. NOTE: The value of CNTLA must be greater than 1h. Each count is equal to 520 times the timeout period of functional clock. For example, with 8MHz functional clock and a 100KHz operating I2C clock, one timeout period will be equal to (1 / 8MHz) * 520 or 65 us.<br>0h = Smallest Value<br>FFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 30.4.2.30 PECCTL (Offset = 1158h) [Reset = 00000000h]

PECCTL is shown in Figure 30-87 and described in Table 30-87.

Return to the Summary Table.

I2CT PEC Control Register.

**Figure 30-87. PECCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | PECEN | RESERVED | | | PECCNT |
| R-0h | | | R/W-0h | R-0h | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PECCNT | | | | | | | |
| R/W-0h | | | | | | | |

**Table 30-87. PECCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-13 | RESERVED | R | 0h | |
| 12 | PECEN | R/W | 0h | PEC Enable. This bit enables the SMB Packet Error Checking (PEC). When enabled, the PEC is calculated on all bits except the START, STOP, ACK and NACK. The PEC internal LSFR and the PECCNT are set to 0 when the I2CT FSM is in the IDLE state, which occurs following a STOP or timeout condition. The PECCNT is also set to 0 after the PEC byte is transmitted or received. Note that the NACK is automatically sent following a PEC byte that results in a PEC error. The PEC polynomial is x^8 + x^2 + x^1 + 1.<br>0h = PEC transmission and check is disabled<br>1h = PEC transmission and check is enabled |
| 11-9 | RESERVED | R | 0h | |
| 8-0 | PECCNT | R/W | 0h | PEC Counter. When this field is non zero, the number of I2C data bytes are counted. When the byte count = PECCNT and the I2CT FSM is transmitting, the contents of the internal LSFR is loaded into the shift register instead of the byte received from the TX FIFO. When the FSM is receiving, after the last bit of this byte is received into the LSFR and the FSM checks that it is non-zero, a PEC RX Error interrupt is generated. The I2C packet must be padded to include the PEC byte for both transmit and receive. In transmit mode, the FIFO must be loaded with a dummy PEC byte. In receive mode, the PEC byte will be passed to the RX FIFO. In the normal use case, FW would set PECEN=1 and PECCNT=0 and use the ACKOEN until the remaining SMBus packet length is known. FW would then set the PECCNT to the remaining packet length (including the PEC bye). FW would then configure the DMA to allow the packet to complete unassisted and exit the NACK mode. Note that when the byte count = PECCNT, the byte count is reset to 0 and multiple PEC calculation can automatically occur within a single I2C transaction.<br>0h = Minimum Value<br>1FFh = Maximum Value |



Copyright © 2026 Texas Instruments Incorporated

Chapter 31

# UNICOMM-SPI

**31.1 UNICOMM-SPI Overview** ..... 1694
**31.2 SPI Operation** ..... 1697
**31.3 UNICOMMSPI Registers** ..... 1714



Copyright © 2026 Texas Instruments Incorporated

## 31.1 UNICOMM-SPI Overview

This section describes the functionality of a UNICOMM module when configured to operate as an SPI. The protocol mode of a given UNICOMM instance is determined by the programmed SELECT field of the IPMODE register. If a UNICOMM instance is configured for another peripheral mode, the SPI functionality on that UNICOMM instance is disabled, and all register reads return a value of zero.

The UNICOMM-SPI (Serial Peripheral Interface) module provides a standardized serial interface to transfer data between MSPM0 devices and other external devices with SPI interface. The SPI module provides a standardized serial interface to transfer data between MSPM0/M33 devices and other external devices (such as a Sensors, Memory, ADCs, or DACs) using SPI protocols

### *31.1.1 Purpose of the Peripheral*

The SPI module acts as a controller or peripheral interface for synchronous serial communication with peripheral devices and other controllers. The transmit and receive paths are buffered with internal, independent FIFO memories, allowing up to 4 entries with 16-bit width. A DMA interface is also provided to allow the data exchange with the transmit and receive FIFO.

### *31.1.2 Features*

The UNICOMM-SPI peripheral mode of the UNICOMM module includes the following features:

- Configurable as a controller or a peripheral
- Programmable clock bit rate and prescaler
- Separate transmit (TX) and receive (RX) buffers/first-in first-out buffers (FIFOs)
  - Availability of buffer / FIFO depends on the variant/overlay chosen
- Programmable data frame size from 4-bits to 16-bits (Controller Mode)
- Programmable data frame size from 7-bits to 16-bits (Peripheral Mode)
- Interrupts for transmit and receive buffers/FIFOs, overrun and timeout interrupts, and DMA completion interrupts
- Programmable SPI mode support for Motorola SPI or Texas Instruments frame format
- Single-bit parity is supported in both transmit and receive paths
- Direct memory access controller interface (DMA)
  - Separate channels for transmit and receive
  - Transfer complete interrupt

**Table 31-1. SPI variants and features**

| Feature | Basic | Advanved |
|---|---|---|
| Controller / Peripheral mode | Yes | Yes |
| Controller: 4-16 bits Target: 7-16 bits | Yes | Yes |
| Parity | Yes | Yes |
| Repeat Mode | No | Yes |
| Receive Timeout | No | Yes |
| Command / Data Control | No | Yes |
| Support for four chip selects | No | Yes |
| FIFO | Refer to the device's datasheet | Refer to the device's datasheet |
| DMA | Refer to the device's datasheet | Refer to the device's datasheet |



Copyright © 2026 Texas Instruments Incorporated

### 31.1.3 Functional Block Diagram

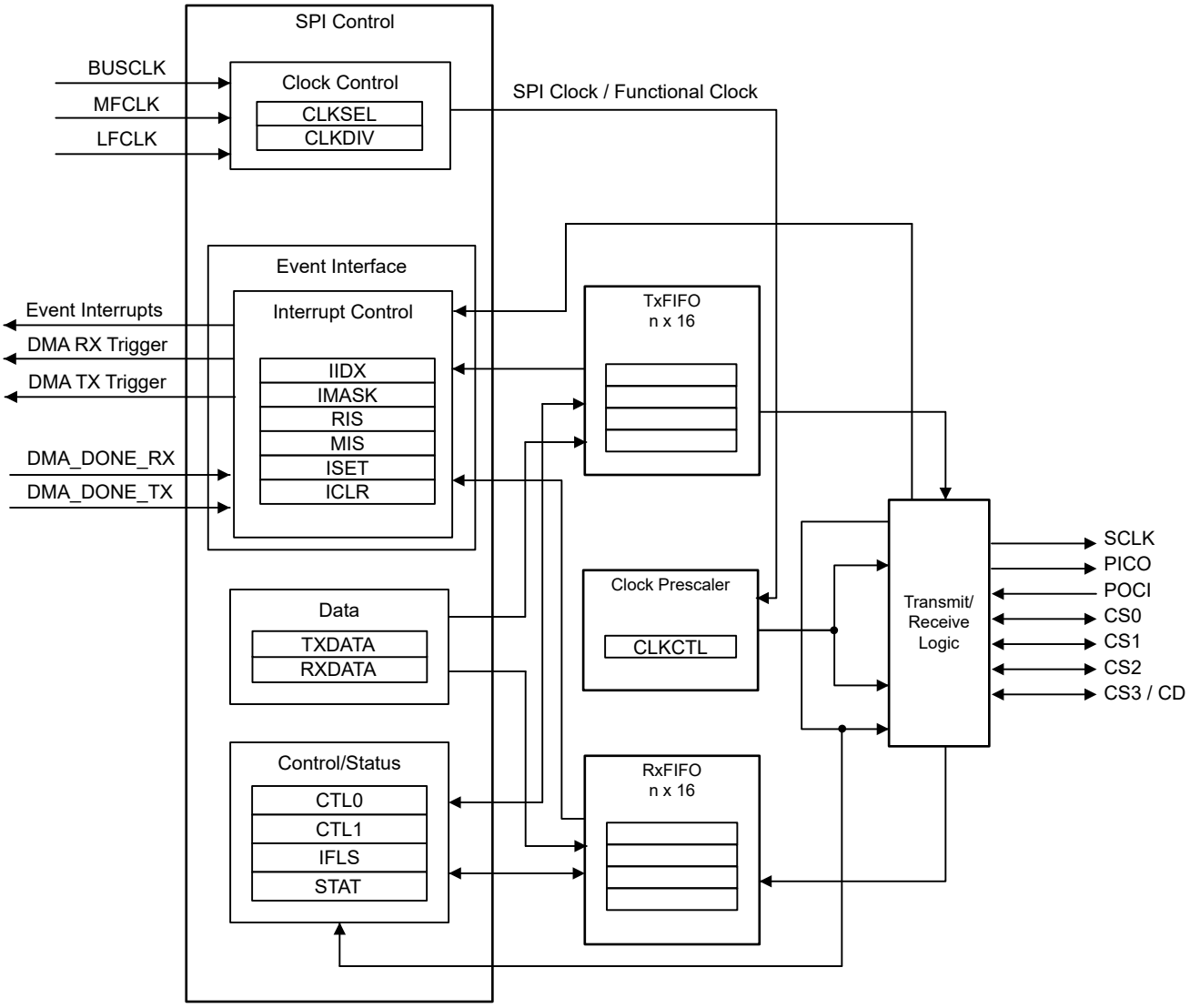



**Figure 31-1. SPI Functional Block Diagram**

### 31.1.4 External Connections and Signal Descriptions

Figure 31-2 and Table 31-2 show an overview of the pin functions for different operation modes of the SPI module.



Copyright © 2026 Texas Instruments Incorporated

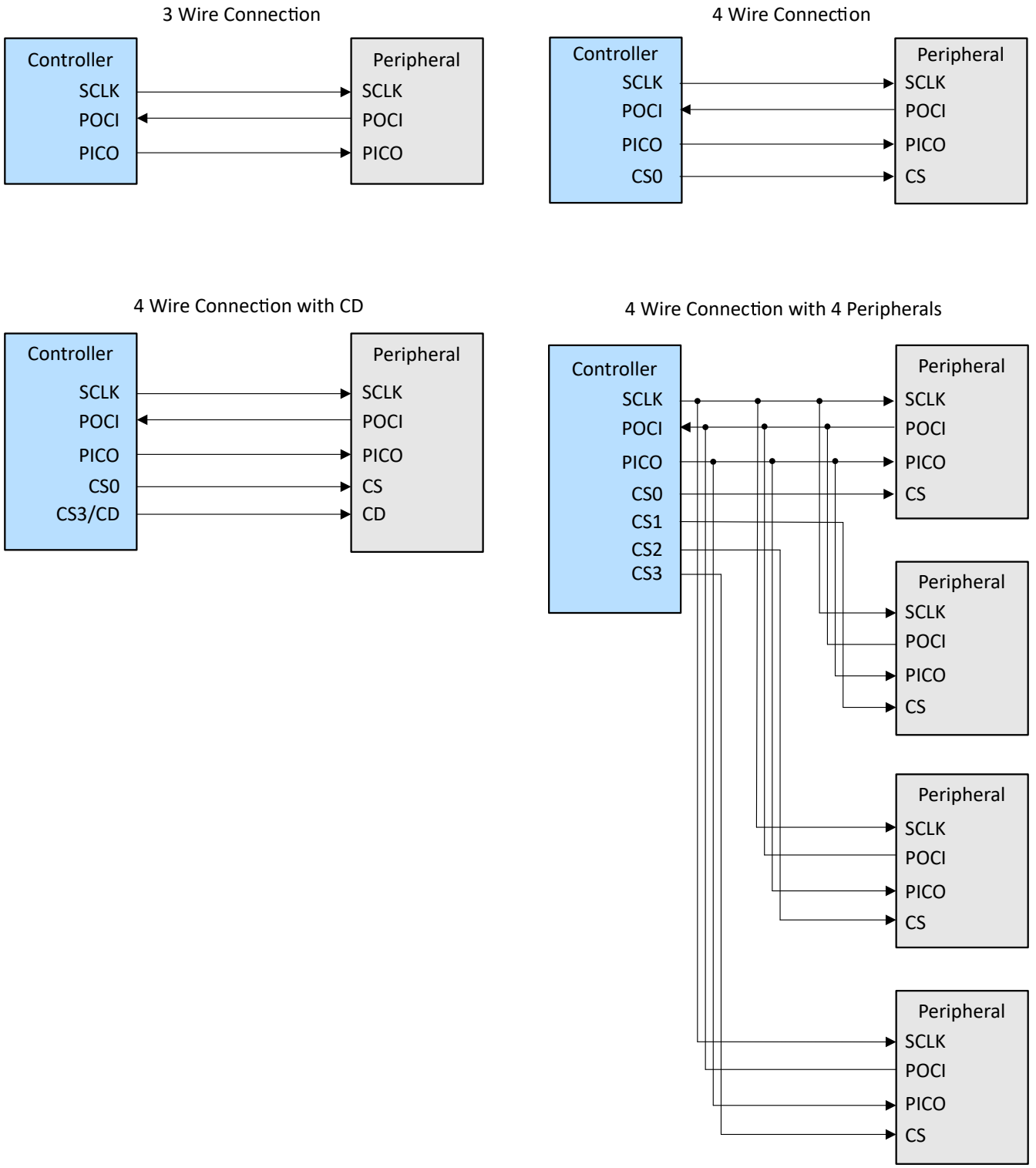



**Figure 31-2. External Connections for Different SPI Configurations**



Copyright © 2026 Texas Instruments Incorporated

**Table 31-2. Pin Function Overview**

| Standard SPI | Feature |
|---|---|
| **SCLK** | **SPI clock**<br>Controller mode: SCLK is an output<br>Peripheral mode: SCLK is an input |
| **PICO** | **Controller out, peripheral in**<br>Controller mode: PICO is the data output line<br>Peripheral mode: PICO is the data input line |
| **POCI** | **Controller in, peripheral out**<br>Controller mode: POCI is the data input line<br>Peripheral mode: POCI is the data output line |
| **CS0** | Chip select signal 0, used in 4-pin mode |
| **CS1** | Chip select signal 1, used in 4-pin mode |
| **CS2** | Chip select signal 2, used in 4-pin mode |
| **CS3/CD** | Chip select signal 3 or Command/Data, used in 4-pin mode |

For UNICOMM, to meet timing requirements in controller mode, chip select and clock should be looped back from IOs. This compensates for the IO delays within the SoC. At the device's IOMUX level, CS and SCLK INENA bits will need to be set.

## 31.2 SPI Operation

### *31.2.1 Clock Frequency Support*

The maximum UNICOMM-SPI frequency supported with controller and peripheral mode depends on the device clock option and IO option. Please refer to specific device data sheet spec for more information.

There is a limitation on SPI clock frequency in both controller and peripheral modes: SPI frequency that can be supported is utmost half of the incoming clock frequency.

- BUSCLK is used for register programming, FIFO, interrupt generation and for SPGDMA; i.e. all elements within SPGSS.
- The UNICOMM SPI has its own clock, as per selected clock source and the FSM and rest of the logic work off this selected clock (functional clock)

The UNICOMM-SPI internal functional clock is selected and divided from the incoming clock to the UNICOMM.

- Use CLKSEL register to select the source of the functional clock – Check each module for supported clocks.
- Use CLKDIV register to select the divide ratio of the function clock.

The selected source clock is always available and frequency depends on selected power mode; for more information, please refers to *Clock Module*.

#### 31.2.1.1 SPI Clock Generation

The SPI includes a programmable bit-rate clock divider and a prescaler to generate the serial output clock (SCLK).

Supported bit rates are up to the input clock divided by 2. The input clock selection depends on the specific device; refer to the device data sheet and *Clock Frequency Support* section.

**SPI Clock** is the output after clock division performed according to ratio selected by the CLKDIV register.

**SPI clock = Selected input clock / (1 + CLKDIV)**

SPI Sampling Clock (SCLK) is the output after dividing the SPI Clock by the Pre-scalar value.

**SCLK= SPI Clock / ((1 + SCR )*2)**

If the factor of two (*2) is set by CLKDIV the input clock must be at least 2 times faster than SPI clock.



Copyright © 2026 Texas Instruments Incorporated

### 31.2.2 General Architecture

#### 31.2.2.1 Chip Select and Command Handling

##### 31.2.2.1.1 Chip Select Control

The UNICOMM-SPI can be configured to be in controller mode by setting the CTL1.CP bit to 1, and in peripheral mode by clearing the CTL1.CP bit.

The CTL0.CSSEL bit selects which connected peripheral is addressed by the up to 4 CS signals. The bits are controlled by the UNICOMM-SPI module in controller or target/peripheral mode. The selected signal is controlled during the transfers.

The chip select signal needs to be provided by the controller in four-wire mode, and the chip select polarity can be inverted by configuring the PINCM.CSx.INV register.

In peripheral mode, the clock is provided by the controller and used by the peripheral to capture the data. The peripheral has the option to operate in 3-wire or 4-wire mode. 4-wire mode only accepts data transfers if the CS is activated.

When the CTL0.CSCLR bit is set, the transmit/receive shift register counter is cleared automatically when the CS goes to the inactive state. When using the Motorola 4-wire, follow these constraints:

- The CS disable period must be longer than 2 functional clock cycles before the CS pin is reasserted
- The CS lead time (CS active to the first bit clock edge) must be at least 2 SPI functional clock cycles.

Following these constraints helps the peripheral synchronize again with the controller in case of a disturbance or glitch on the clock line or during initialization. This bit is relevant only in the peripheral mode.

- CTL0.CSCLR = 0: The transmit/receive bit counter state is retained when the CS signal disables the peripheral.
- CTL0.CSCLR = 1: The transmit/receive bit counter is cleared when the CS signal disables the peripheral.

**Note**

The CSCLR function requires the CS disable pulse to be longer than 2 SPI function clock cycles to properly detect and clear the bit counter in the SPI. The CS lead time (CS active to the first bit clock edge) also needs to be at least 2 SPI function clock cycles. The UNICOMM-SPI module does not support asynchronous clock request generation.

#### 31.2.2.2 Command Data Control

When using the Motorola frame format, the CDMODE bit can be set to use the CS3/CD line as a signal to distinguish between Command and Data information. This is often used for LCD or data storage devices.

- CD level low: command function
- CD level high: data function

The CTL1.CDMODE can be written with a value of 1-14 to specify the number of bytes, and the CD line will go low for the given number of bytes, which are sent by the SPI, starting with the next value to be transmitted. After the number of bytes is transmitted, the CD will automatically go high. If a value of 0xF is set, the C/D stays low permanently; a value of 0 sets the CD line to high immediately after the current character has been transmitted.

This option is only available in controller mode. CTL1.CDENABLE can only be updated when the SPI module is disabled, CTL1.CDMODE can be updated between the different data packages. The counter will be reset with CDENABLE or SPI ENABLE set to disabled before setting a new value in CTL1.CDMODE. The FIFO status should be checked to ensure it is empty.

When writing a new value into CTL1.CDMODE, the internal counter will be reset, and the new value will be used for counting. If the counter counts down to 0 and a new command package needs to be sent, CDMODE needs to be set again; otherwise, the following data is sent with the CD pin signaling data mode.



Copyright © 2026 Texas Instruments Incorporated

CTL1.CDMODE will hold the last written value by the application – this field is not updated by hardware; STAT.CDMODE reflects the internal counter value.

#### 31.2.2.3 Data Format

The control bit CTL1.CP B defines the direction of data input and output using the most-significant bit (MSB) or the least-significant bit (LSB) first. If the parity is enabled, the parity bits are always received as the last bit.

With the control register bits CTL0.DSS, the bit length per transfer will be defined between 4 and 16 bits in Controller mode and 7 and 16 bits in Peripheral mode.

A transfer will be triggered by writing to the TX buffer register. The data write needs to have at least the number of bits of the transfer. For example, if only a byte is written to the TX buffer, but the length of the transfer is > 8, the missing bits will be filled with 0s. On the receive path, received data will be moved to the RX FIFO or the RX buffer after the number of bits defined in the CTL0.DSS register has been received.

The RX and TX buffers shall be accessed with at least the bits covering one transfer.

- 4-8 bits: byte access (Peripheral mode: 7-8 Bits)
- 9-16 bits: 16-bit access

Clock polarity (CTL0.SPO) is used to control the clock polarity when data is not being transferred, and it is only used in the Motorola SPI frame mode.

- 0h = peripheral produces a steady-state LOW value on the CLKOUT pin when data is not being transferred.
- 1h = peripheral produces a steady-state HIGH on the CLKOUT pin when data is not being transferred.

The Clock phase (CTL0.SPH) bit selects the clock edge that captures data and determines whether the data can change state. It has the greatest impact on the first bit transmitted, either by allowing or disallowing a clock transition before the first data capture edge. Please refer to the Motorola SPI frame mode section to check the diagram.

- 0h = Data is captured on the first clock edge transition.
- 1h = Data is captured on the second clock edge transition.

The SPI can be configured to work in Peripheral mode with CTL1.CP  bit = 0. In Peripheral mode, the clock is provided by the controller and is available to the peripheral on the CLK pins, which must be configured as inputs. The Clock Select and divider control bits are not used. The CS input signal selects and enables the peripheral's data receive path in 4-wire mode.

The SPI can be configured to work as a Controller with CTL1.CP  bit = 1. In Controller mode, the clock needs to be generated by selecting an available clock source using the clock select bits. It also needs to control the CS signal depending on the selected protocol.

When setting the CTL1.PEN bit, the last bit will be used as parity to evaluate the integrity of the previous bits—the CTL1.PES bit selects the parity mode as even or odd. When detecting a fault, the interrupt flag RIS.PER is set to mark the data as invalid. Parity checking is a feature that improves the robustness of communication.

#### 31.2.2.4 Delayed data sampling

When the input data arrives at the POCI pin with a delay due to runtime conditions, the previous data is sampled at the sampling clock edge in the following input data sampling stage. To compensate for such a condition, delayed sampling can be enabled via the CLKCTL.DSAMPLE bits. The delayed sampling is only available in controller mode. The delay can be adjusted in steps of SPI input clock steps by setting the control register bits CLKCTL.DSAMPLE. The maximum allowed delay should not exceed the length of one data frame.

Delayed sampling will not be used when CLKCTL.DSAMPLE is set to “zero.”

#### 31.2.2.5 DMA Operation

The section applies to UNICOMM-SPI configurations which support the SPI-DMA feature.



Copyright © 2026 Texas Instruments Incorporated

The UNICOMM-SPI provides an interface to the DMA module, with separate transmit and receive lines. The DMA operation of the SPI is enabled through the SPI DMA Event and DMA module registers. When DMA operation is enabled, the SPI asserts a DMA request on the receive or transmit channel when the associated FIFO can transfer data. The DMA transfer requests are handled automatically by the DMA controller based on the DMA channel configurations (burst size, transfer size, source address, etc).

- For the receive channel, a DMA transfer request is asserted when the receive FIFO contains at least the FIFO trigger level configured by the RXIFLSEL bit in the IFLS register.
- For the transmit channel, a DMA transfer request is asserted whenever the transmit FIFO contains fewer characters than the FIFO trigger level configured using the TXIFLSEL bit in the IFLS register.

See more information about the DMA interrupts and events in Section 31.2.14.2.

#### 31.2.2.6 Low Power Mode

The UNICOMM-SPI module instances may be located in either Power Domain 0 (PD0) or Power Domain 1 (PD1).

For Power Domain 0, the module can remain enabled before entering low-power mode. However, the use of the module during low-power mode is not recommended due to low-power mode limitations in the MCU resources.

For Power Domain 1, the module must be brought to a safe state before entering low-power mode. This is accomplished by completing the suspend/disable process described below.

When operating in Peripheral mode, the UNICOMM-SPI module must reach an idle state (assuming the external SPI controller has completed data transmission). Once idle, perform the following actions in sequence: read all remaining data from the receive FIFO, clear the FIFO, set the SUSPEND bit in the CTL1 register to 1 to suspend the device, and finally, disable the module.

If the module is operating in Controller mode, the UNICOMM-SPI module should transmit all remaining data and clear all transmit FIFOs. Once the module reaches an idle state and all previous steps are complete, suspend the device by setting the SUSPEND bit in the CTL1 register to 1, and disable the module.

The device can then enter a low-power mode state. Upon exiting low-power mode, clear the SUSPEND bit in the CTL1 register to 0, after which the module may be re-enabled.

### *31.2.3 FIFO Operation*

The UNICOMM-SPI module has two FIFOs (one each for transmit/receive) with configurable depth. FIFOs are accessed via the TXDATA & RXDATA registers.

#### 31.2.3.1 FIFO Size

FIFO width will be 16 bits without packing and 32 bits with packing. Refer to the device's datasheet for specific information on the FIFO size.

#### 31.2.3.2 FIFO Status bits

The UNICOMM-SPI module supports the below FIFO status bits:

- Receive FIFO Full: when all elements are populated
- Receive FIFO Empty: when there are no elements in the HEAD
- Transmit FIFO Full: when all elements are populated
- Transmit FIFO Empty: when are no elements in the FIFO

In addition to this, FIFO clear status – RXCLR/TXCLR will also be present (refer to Clearing FIFO Contents).

##### *31.2.3.2.1 RIS.RX based on FIFO threshold settings*

When a message gets received, the RX FIFO gets filled immediately. Below are the available thresholds:

1. RX_FIFO full: when all elements are populated
2. RX_FIFO >= ¼ full
3. RX_FIFO >= ½ full



Copyright © 2026 Texas Instruments Incorporated

4. RX_FIFO >= ¾ full
5. RX_FIFO almost empty: <= 1 element
6. RX_FIFO almost full: >= (N-1) elements
7. **RX_FIFO not empty: opposite of empty**

#### *31.2.3.2.2 RIS.TX based on FIFO threshold settings*

On the transmit side, data is written by the CPU/DMA master at the top most empty FIFO entry and then transferred to serializer. Thresholds are as per elements that can be written into the FIFO.

1. TX_FIFO empty: when there no elements in the FIFO
2. TX_FIFO <= ¾ empty
3. TX_FIFO <= ½ empty
4. TX_FIFO <= ¼ empty
5. TX_FIFO almost empty: <=1 element
6. TX_FIFO almost full: >= (N-1) elements
7. **TX_FIFO not full: opposite of full**

**Note**

Receive FIFO Almost Empty and Transmit FIFO Almost Full options are provided in IFLS setting; however, these should not be used for generating flags based on the watermark settings or for DMA usage. The application has to ensure that correct FIFO Thresholds are used.

**Note**

Debugger initiated reads should not pop contents out of the FIFO. However, debugger initiated writes into the transmit FIFO should be treated as valid entries.

### 31.2.3.3 Clearing FIFO contents

To clear the FIFO contents, the following sequence must be used::

- Write ‘1’ to the CLR control bit (TXCLR/ RXCLR).
- Wait for clear to complete, i.e. CLR status bit reflects ‘1’; poll/check for CLR status bit to reflect ‘1’
- Write ‘0’ to CLR control bit (TXCLR/ RXCLR).

**Note**

When a FIFO clear has been initiated, do not try to re-clear it again.

**Note**

Randomly changing the FIFO levels when the FIFO has contents might lead to undesired behavior. It is recommended that the FIFO is cleared before changing the FIFO thresholds.

### 31.2.3.4 Hardware monitors empty, full and overflow conditions

The STAT register contains empty and full flags (TXFE, TXFF, RXFE, and RXFF bits), and the RIS register shows the overflow status of the receive FIFO via the RXFIFO_OVF bit. There is no indicator for a transmit FIFO overflow. A write is just lost if it overflows the transmit FIFO. If the FIFOs are not present, the empty and full flags are set according to the status of the 1-byte-deep holding registers.

**Note**

Data will be lost if the FIFO is full, and the overflow flag bit will be set.

Out of reset, both FIFOs are configured to trigger an interrupt at half-level mark.



Copyright © 2026 Texas Instruments Incorporated

**Note**

FIFO integrity is not guaranteed if the software disables the module in the middle of a transmission with data in the FIFO, and then re-enables it.

**Note**

When the module gets reset the content of the FIFO are also cleared.

### *31.2.4 Suspend communication*

Suspend is an extreme operating condition for the module, and the application should enter this mode while ensuring the following conditions are met. The application can suspend external communication by setting the suspend control bit.

When suspend is set, ongoing communication will complete, the FSM will return to idle, and further external communication will be stopped. Transmit line/s will be driven to their IDLE state, and further toggle on receive line/s will not be processed. This provides a safe means for the application to suspend operations, clear out receive FIFOs before disabling the module.

**Note**

The *suspend* bit needs to be cleared prior to re-enabling the module.

To suspend/disable the UNICOMM-SPI module, follow the steps below:

1. The SPI-BUS controller have to ensure that the target can reach the IDLE state (refer to the SPI Idle State Requirements)
2. The hardware ensures that ongoing transactions are completed. In the case of a transmit operation, all the contents of the FIFO will be sent out, prior to hardware raising an IDLE state
3. Once the SUSPEND bit is written into hardware; the CPU has to poll the status flags to ensure that device has reached the IDLE state
4. Drain out receive FIFO
5. Disable the UNICOMM-SPI module

After suspending or disabling the module, the register configurations are retained. Communication can be resumed by following the steps below to re-enable the module:

1. Clear the SUSPEND bit
2. Re-enable the UNICOMM-SPI module

#### 31.2.4.1 SPI IDLE State Requirements

To reach the IDLE state in the UNICOMM-SPI module, follow the steps below:

1. Controller side: ensure ‘CS’ is returned to its idle state
2. Peripheral side: ensure ’CS’ is read back as idle before reflecting an IDLE state

The application must ensure that the idle state can be reached before asserting SUSPEND. Example:

1. Controller mode: Ensure the entire frame, i.e., all commands and data, is transmitted
2. Target mode: Ensure that the entire frame has been received/transmitted

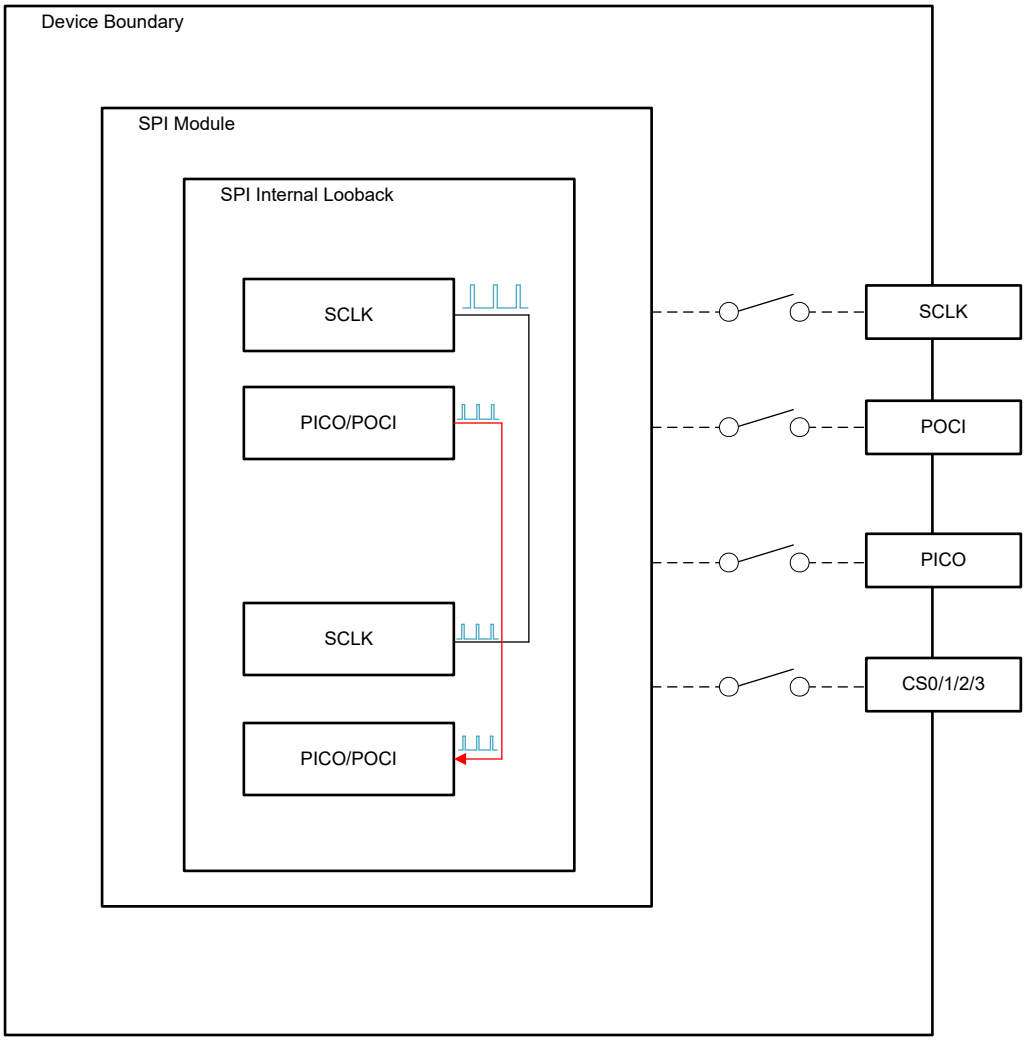
### *31.2.5 Internal Loopback Operation*

The UNICOMM-SPI module can be placed into an internal loopback mode for diagnostic or debug work by setting the LBM bit in the CTL1 register. In loopback mode, the data from the TX FIFO can be serially transmitted into the RX FIFO. Data from the RX FIFO can be read to determine whether transmission has occurred. External toggling of the IOs has no effect when the module is set to internal loopback mode.



Copyright © 2026 Texas Instruments Incorporated

**Internal Loopback Mode**




### *31.2.6 Repeat Transfer mode*

This function is available only in Controller mode by using the CTL1.REPEATTX bits, the last character transmitted will be repeated as defined by the register bits—a value of 0 in CTL1.REPEATTX bits will disable this mode.

The transfer will start by writing data into the TX Buffer. Then the data will be repeatedly sent with the same given value. The behavior is identical to that of if the data were written into the TX Buffer that many times, as defined by the value here. It can be used to clean a transfer or to pull a certain amount of data from a peripheral.

When the REPEATTX is used, it needs to be aligned with the data in the buffer/FIFO. So, the sequence shown below should be used:

- Wait and check till FIFO is empty. Alternatively, clear out the FIFO
- Setup REPEATTX
- Write to TXDATA / FIFO
- Wait till requested data has been received

### *31.2.7 Receive Timeout*

In peripheral mode, the application can detect a receive timeout by programming CTL1.RXTIMEOUT and CLKCTL.SCR. The receive timeout is calculated as the time from the last active clock edge to the programmed



Copyright © 2026 Texas Instruments Incorporated

timeout period. The receive timeout flag is raised if there are contents in the FIFO and the receive timeout period has elapsed.

Receive timeout period calculation

- Clock period for timeout calculation = SPI clock / (2*(1 + SCR))
- Timeout period = CTL1.RXTIMEOUT * clock period for timeout calculation

### *31.2.8 Line Timeout*

An additional bit, LTOUT (LINE timeout), is provided in the UNICOMM-SPI module. In peripheral mode, if no new transaction is detected during the receive timeout period after the last transaction, a line timeout interrupt (LTOUT) is set, irrespective of FIFO/buffer contents.

### *31.2.9 Protocol Descriptions*

The protocol format mode can be selected by using CTL0.FRF register. The supported options include Motorola 3-wire, Motorola 4-wire, and Texas Instruments Synchronous.

#### 31.2.9.1 Motorola SPI Frame Format

The Motorola SPI interface is a 4-wire interface where the CS signal behaves as a peripheral select. In the 3-wire mode the CS signals is not required and the module behaves as if always selected. The main feature of the Motorola SPI format is that the inactive state and phase of the SCLK signal can be programmed through the SPO and SPH bits in the SPIx.CTL0 control register.

**SPO Clock Polarity Bit**

If the CTL0.SPO clock polarity control bit is clear, the bit produces a steady-state low value on the SCLK pin when data is not being transferred. If the CTL0.SPO bit is set, the bit places a steady-state high value on the SCLK pin when data is not being transferred.

**SPH Phase-Control Bit**

The CTL0.SPH phase-control bit selects the clock edge that captures data, and allows it to change state. The state of this bit has the most impact on the first bit transmitted, by either allowing or not allowing a clock transition before the first data capture edge. If the CTL0.SPH phase-control bit is clear, data is captured on the first clock edge transition. If the SPH bit is set, data is captured on the second clock edge transition.

**Motorola SPI Frame Format with SPO = 0 and SPH = 0**

Figure 31-3 shows signal sequences for Motorola SPI format with SPO = 0 and SPH = 0.

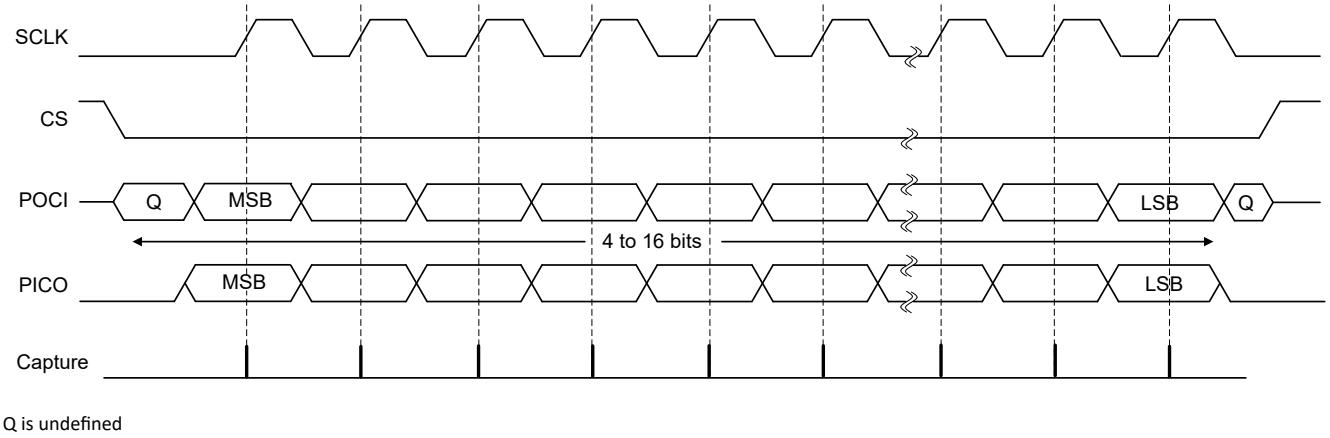



**Figure 31-3. Motorola SPI Format With SPO = 0 and SPH = 0**

In this configuration, the following occurs during idle periods:

- SCLK is forced low
- CS is forced high



Copyright © 2026 Texas Instruments Incorporated

- The transmit data line PICO is forced low
- When the SPI is configured as a controller, it enables the SCLK pin
- When the SPI is configured as a peripheral, it disables the SCLK pin

If the SPI is enabled and the TX FIFO contains valid data, the CS controller signal is driven low at the start of transmission, enabling peripheral data onto the POCI input line of the controller. The controller PICO output pin is enabled.

One-half SCLK period later, valid controller data is transferred to the PICO pin. Once both the controller and peripheral data are set, the SCLK controller clock pin goes high after an additional one-half SCLK period. The data is now captured on the rising edges and propagated on the falling edges of the SCLK signal.

For a single-word transmission, after all bits of the data word are transferred, the CS line is returned to its IDLE high state one SCLK period after the last bit is captured. For continuous back-to-back transmissions, the CS signal must pulse high between each data-word transfer because the peripheral-select pin freezes the data in its serial peripheral register and prevents altering the data if the SPH bit is clear. The controller device must raise the CS pin of the peripheral device between each data transfer to enable the serial peripheral data write. When the continuous transfer completes, the CS pin is returned to its IDLE state one SCLK period after the last bit is captured.

**Motorola SPI Frame Format with SPO = 0 and SPH = 1**

Figure 31-4 shows the signal sequence for Motorola SPI format with SPO = 0 and SPH = 1.

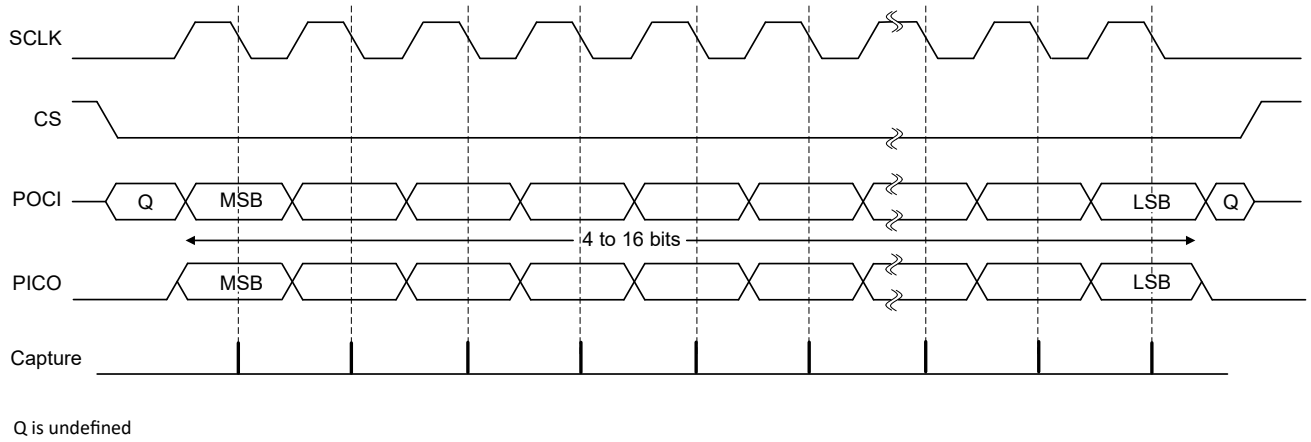



**Figure 31-4. Motorola SPI Frame Format With SPO = 0 and SPH = 1**

In this configuration, the following occurs during idle periods:

- SCLK is forced low
- CS is forced
- The transmit data line PICO is forced low
- When the SPI is configured as a controller, it enables the SCLK pin
- When the SPI is configured as a peripheral, it disables the SCLK pin

If the SPI is enabled and valid data is present in the TX FIFO, CS controller signal goes low at the start of transmission.The controller PICO output is enabled. After an additional one-half SCLK period, both controller and peripheral valid data are enabled onto their respective transmission lines. At the same time, SCLK is enabled with a rising-edge transition. Data is then captured on the falling edges and propagated on the rising edges of the SCLK signal.

Fora single-word transfer, after all bits are transferred, the CS line is returned to its IDLE high state one SCLK period after the last bit is captured. For continuous back-to-back transfers, the CS pin is held low between successive data words and terminates like a single-word transfer.

**Motorola SPI Frame Format with SPO = 1 and SPH = 0**



Copyright © 2026 Texas Instruments Incorporated

Figure 31-5 shows signal sequences for Motorola SPI format with SPO = 1 and SPH = 0.

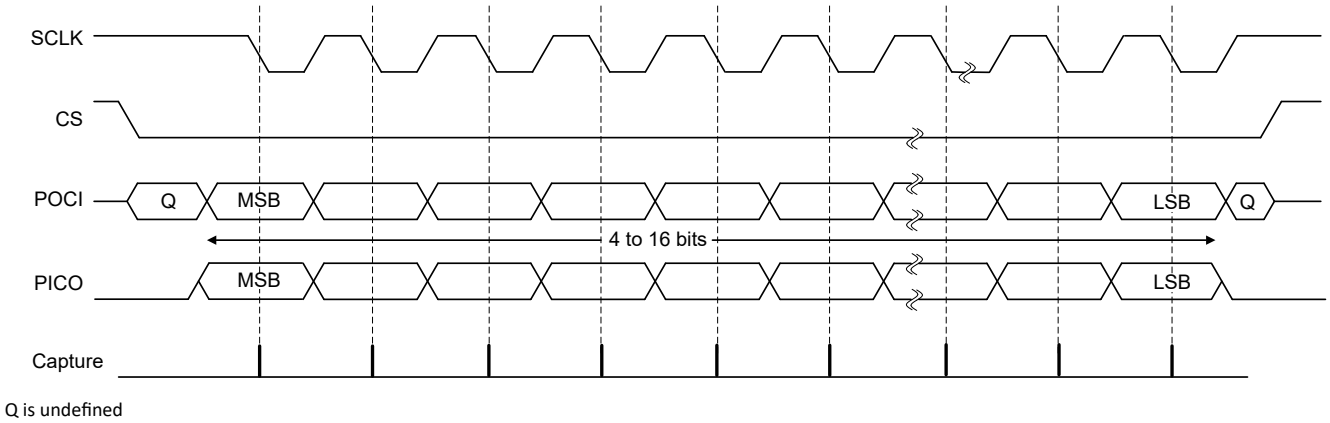



**Figure 31-5. Motorola SPI Frame Format With SPO = 1 and SPH = 0**

In this configuration, the following occurs during idle periods:

- SCLK is forced high
- CS is forced high
- The transmit data line PICO is arbitrarily forced low
- When the SPI is configured as a controller, it enables the SCLK pin
- When the SPI is configured as a peripheral, it disables the SCLK pin

If the SPI is enabled and valid data is in the TX FIFO, the SPI CS controller signal goes low at the start of transmission and transfers the peripheral data to the controller's POCI line immediately. The controller PICO output pin is enabled.

One-half SCLK period later, valid controller data is transferred to the PICO line. When both the controller and peripheral data have been set, the SCLK controller clock pin becomes low after one additional half SCLK period. Data is captured on the falling edges and propagated on the rising edges of the SCLK signal.

For a single-word transmission after all bits of the data word are transferred, the CS line is returned to its IDLE high state one SCLK period after the last bit is captured. For continuous back-to-back transmissions, the CS signal must pulse high between each data-word transfer, as the peripheral-select pin freezes the data in its serial peripheral register and prevents it from being altered when the SPH bit is clear. The controller device must raise the CS pin of the peripheral device between each data transfer to enable the serial peripheral data write. When the continuous transfer completes, the CS pin returns to its IDLE state one SCLK period after the last bit is captured.

**Motorola SPI Frame Format with SPO = 1 and SPH = 1**

Figure 31-6shows the signal sequence for Motorola SPI format with SPO = 1 and SPH = 1.



Copyright © 2026 Texas Instruments Incorporated

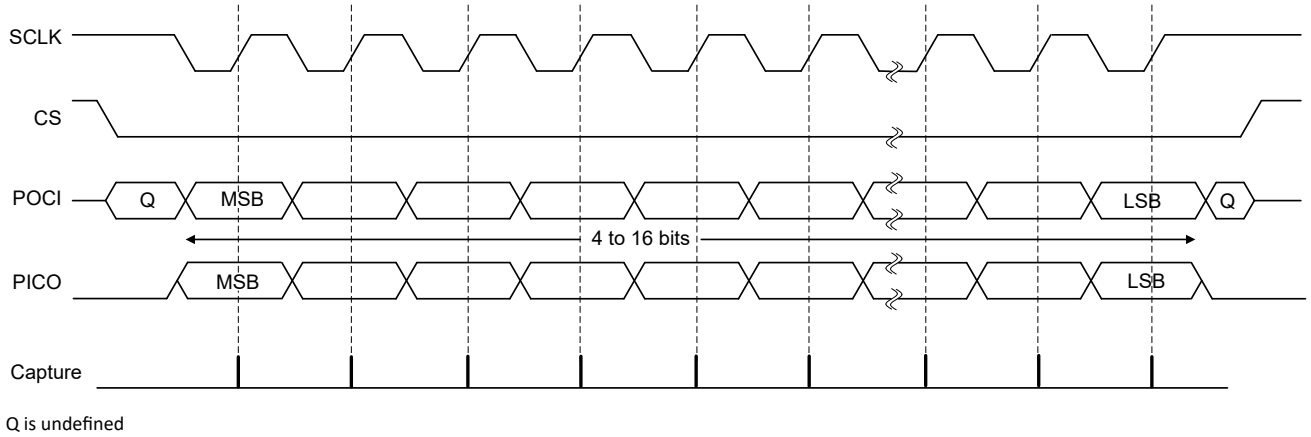



**Figure 31-6. Motorola SPI Frame Format With SPO = 1 and SPH = 1**

In this configuration, the following occurs during idle periods:

- SCLK is forced high
- CS is forced high
- The transmit data line PICO is arbitrarily forced low
- When the SPI is configured as a controller, it enables the SCLK pin
- When the SPI is configured as a peripheral, it disables the SCLK pin

If the SPI is enabled and valid data is in the TX FIFO, the start of transmission is signified by the CS controller signal going low. The controller PICO output pin is enabled. After an additional one-half SCLK period, both controller and peripheral data are enabled onto their respective transmission lines. At the same time, SCLK is enabled with a falling-edge transition. Data is then captured on the rising edges and propagated on the falling edges of the SCLK signal.

For a single word transmission, after all bits are transferred, the CS line returns to its IDLE high state one SCLK period after the last bit is captured. For continuous back-to-back transmissions, the CS pin remains in its active-low state until the final bit of the last word is captured, then returns to its IDLE state. For continuous back-to-back transfers, the CS pin is held low between successive data words and terminates like a single-word transfer.

The serial clock (SCLK) is held inactive while the SPI is idle, and SCLK transitions at the programmed frequency only during data transmission or reception. The IDLE state of SCLK provides a receive timeout indication when the RX FIFO still contains data after the timeout period.

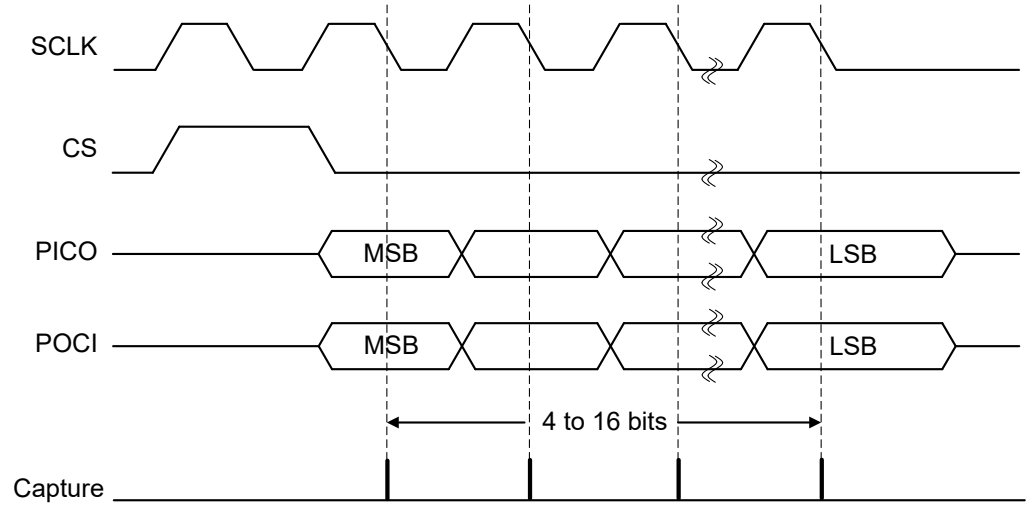
#### 31.2.9.2 Texas Instruments Synchronous Serial Frame Format

The SPI peripheral is compatible with Texas Instruments Synchronous Serial frame format.

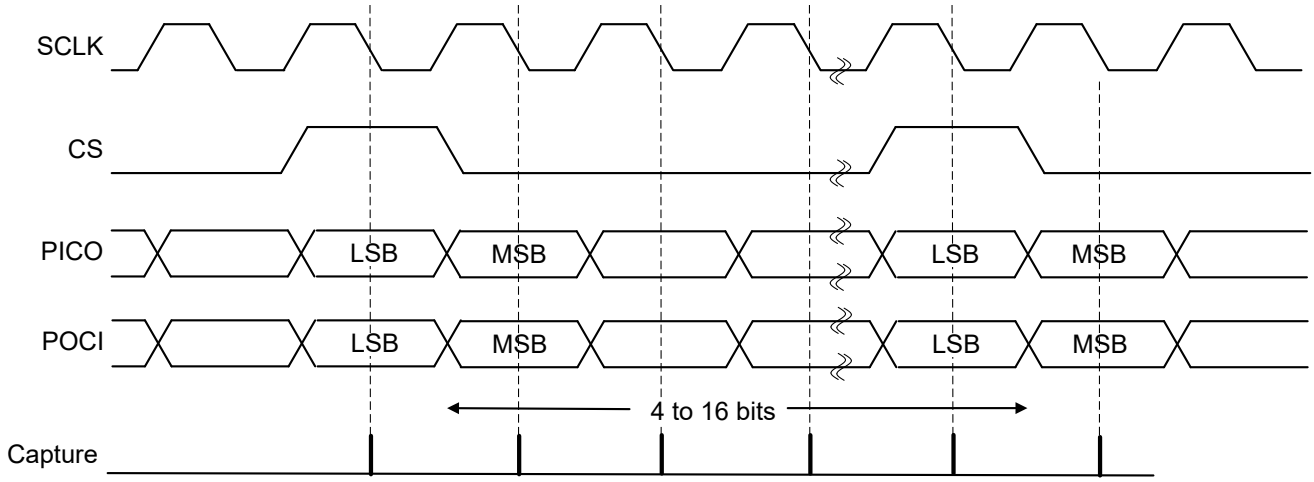
Figure 31-7 shows the TI synchronous serial frame format for a single and continuous transmitted frame.



Copyright © 2026 Texas Instruments Incorporated




**Single Transmission Signal Sequences**




**Continuous Transmission Signal Sequences**

**Figure 31-7. TI Synchronous Serial Frame Format**

SCLK and CS are forced low, and the transmit data line PICO is put in tristate whenever the SPI is idle. When the bottom entry of the TX FIFO contains data, CS is pulsed high for one SCLK period. The transmitted value is also transferred from the TX FIFO to the transmit logic's serial shift register. On the next rising edge of SCLK, the MSB of the 4- to 16-bit data frame is shifted out on the PICO pin. Likewise, the MSB of the received data is shifted onto the POCI pin by the off-chip serial peripheral device. Both the SPI and the off-chip serial peripheral device then clock each data bit into their serial shifter on each falling edge of SCLK. The received data is transferred from the serial shifter to the RX FIFO on the first rising edge of SCLK after the least significant bit (LSB) is latched.

The serial clock (SCLK) is held inactive while the SPI is idle, and SCLK transitions at the programmed frequency only during active transmission or reception of data. The IDLE state of SCLK provides a receive time-out indication when the RX FIFO still contains data after the time-out period.



Copyright © 2026 Texas Instruments Incorporated

### 31.2.10 Status Flags

The table below lists the status and their behavior. FIFO status bits are explained in the common SPGSS chapter.

**Table 31-3. SPI variants and features**

| Status Bit | Description |
|---|---|
| BUSY | **Controler Mode:**<br>1. Busy bit is set (“ACTIVE”) when there is an ongoing transmission or reception<br>2. Busy bit is cleared (“IDLE”) where there is no ongoing transaction (receive or transmit)<br>**Peripheral Mode:**<br>1. BUSY bit is set (“ACTIVE”) when there is an ongoing transmission or reception. When POD is set, i.e. peripheral is only receiving data, in this case BUSY is set, when peripheral is receiving data<br>2. Busy bit is cleared (“IDLE”) when there is no ongoing transaction (receive or transmit) |
| CDMODE | Reflects current counter value and manual status<br>0x0: manual mode – C/D is low (data is being sent)<br>0xf: manual mode – C/D is high (command is being sent)<br>0x0 to 0xe: present byte count |

### 31.2.11 Module configuration

The application has to set the configuration registers for the desired operation before enabling the module (through the ENABLE bits). Configuration bits in Table 4 cannot be updated once the ENABLE bit is set. If these bits need to be updated, the application would have to disable the module, update the configuration registers, and then re-enable the module. When the IP is DISABLED (ensure the sequence SUSPEND → Clear FIFO → DISABLE is followed), the FSM returns to IDLE, the FIFO contents are cleared, and the entire functional clock domain returns to its idle state.

**Note**

In controller mode, CSSEL can be changed dynamically. However, in peripheral mode, CSSEL must be configured before enabling the module.

### 31.2.12 Reset Considerations

#### Software Reset Considerations

The UNICOMM has a general software reset for the UNICOMM module. You can find more information in the UNICOMM Enables & Resets section.

#### Hardware Reset Considerations

A hardware reset also initializes the IO configuration. This sets the IOs to a high-impedance state, allowing the data lines to float. If this is critical for the application or connected devices on the SPI interface, external pull-up or pull-down resistors might be required.

### 31.2.13 Initialization

To enable and initialize the SPI, the following steps are necessary:

1. Configure IOMUX with appropriate GPIO pins for which SPI signals are multiplexed to.



Copyright © 2026 Texas Instruments Incorporated

**Note**

Pull-ups can be used to avoid unnecessary toggles on the SPI pins, which can take the Peripheral to a wrong state. In addition, if the SCLK signal is programmed to steady state High through the SPO bit in the CTL0 register, then software must also configure the GPIO port pin corresponding to the SCLK signal as a pull-up.

2. For each of the frame formats, the SPI is configured using the following steps:

a. If initializing out of reset, ensure that ENABLE bit in the CTL1 register is clear before making any configuration changes.
b. Select and configure the clock pre-scale divisor by writing the CLKSEL and CLKDIV register.
c. Select whether the SPI is a Controller or Peripheral
d. Configure the clock divisor by writing the CLKCTL register.
e. Configure the CTL0 and CTL1 register with based on the protocol, data width and other special configurations.
f. Optionally configure DMA
g. Enable the SPI by setting the ENABLE bit in the CTL1 register

### *31.2.14 Interrupt and Events Support*

The UNICOMM-SPI module contains three event publishers and no event subscribers. One event publisher (CPU_INT) manages the UNICOMM-SPI interrupt requests (IRQs) to the CPU subsystem through a static event route. The second and third event publishers (DMA_TRIG_RX, DMA_TRIG_TX) are used to setup the trigger signaling for the DMA through DMA event route.

The SPI events are summarized in the SPI Events table bellow:

**Table 31-4. SPI Events**

| Event | Type | Source | Destination | Route | Configuration | Functionality |
|---|---|---|---|---|---|---|
| CPU interrupt | Publisher | SPI | CPU Subsystem | Static route | CPU_INT registers | Fixed interrupt route from SPI to CPU |
| DMA trigger | Publisher | SPI | DMA | DMA event route | DMA_TRIG_RX registers | Fixed interrupt route from SPI RX to DMA |
| DMA trigger | Publisher | SPI | DMA | DMA event route | DMA_TRIG_TX registers | Fixed interrupt route from SPI TX to DMA |

#### 31.2.14.1 CPU Interrupt Event Publisher (CPU_INT)

The UNICOMM-SPI module provides 9 interrupt sources that can source a *CPU Interrupt Event*. The SPI CPU_INT Trigger Conditions table lists the CPU interrupt events from the UNICOMM-SPI module in order of decreasing priority.

**Table 31-5. SPI CPU_INT Trigger Conditions**

| IIDX STAT | Name | Description |
|---|---|---|
| 0x01 | PER | Parity error event. This interrupt is set if a Parity error has been detected. |
| 0x02 | RXFIFO_OVF | RXFIFO overflow event. This interrupt is set when an RX FIFO overflow is detected. |
| 0x03 | RX | Receive FIFO event. This interrupt is set when the selected receive FIFO level is reached. |
| 0x04 | RXFULL | RX FIFO full event. This interrupt is set when the RX FIFO is full. |
| 0x05 | TXFIFO_UNF | TX FIFO underflow event. This interrupt is set when a TX FIFO underflow is detected. |
| 0x06 | TX | Transmit FIFO event. This interrupt is set when the selected transmit FIFO level is reached. |
| 0x07 | TXEMPTY | Transmit FIFO empty interrupt. This is set if all data in the transmit FIFO has been shifted out. |



Copyright © 2026 Texas Instruments Incorporated

**Table 31-5. SPI CPU_INT Trigger Conditions (continued)**

| IIDX STAT | Name | Description |
|---|---|---|
| 0x09 | IDLE | SPI IDLE event. SPI has finished transfers and entered the IDLE mode. This interrupt is set when STAT.BUSY bit goes low. |
| 0x0A | RTOUT | SPI Receive Time-Out event. This interrupt is set when the FIFO still contains data after the timeout period. |
| 0x10 | DMA_DONE_RX | This interrupt is set if the RX DMA channel sends the DONE signal. |
| 0x11 | DMA_DONE_TX | This interrupt is set if the TX DMA channel sends the DONE signal. |
| 0x15 | LTOUT | SPI Line Time-Out event. This interrupt is set after the timeout period has lapsed. |

The CPU interrupt event configuration is managed with the CPU_INT event management registers.

The receive interrupt (SPI.RX) changes state when one of the following events occurs:

- FIFO Present: The receive interrupt is set when the receive FIFO is at or above the programmed trigger level. The receive interrupt is cleared by reading data from the receive FIFO until it falls below the trigger threshold, by reading the interrupt index (IIDX read), or by writing a 1 to the receive interrupt bit in ICLR. The receive FIFO should be read until the number of entries in the receive FIFO falls below the programmed threshold in IFLS. This is to ensure that subsequent interrupts are fired correctly.
- Single Buffer Present: The receive interrupt is set when data is received, and the receive buffer is full, thereby filling the location. The receive interrupt is cleared by performing a single read of the receive buffer, by reading the interrupt index from IIDX, or by writing a 1 to the RXINT bit in ICLR.

The transmit interrupt (SPI.TX) changes state when one of the following events occurs

- FIFO Present: The transmit interrupt bit is set when the transmit FIFO is at or below the programmed trigger level. The transmit interrupt is cleared by writing data to the transmit FIFO until it exceeds the trigger level, by reading the interrupt index from IIDX, or by writing a 1 to the transmit interrupt bit in ICLR. The transmit FIFO should be filled till the number of entries in the FIFO crosses the programmed threshold in IFLS. This is to ensure that subsequent interrupts are fired correctly.
- Single Buffer Present: The transmit interrupt bit is set when the buffer is empty, indicating a transmit empty condition. The transmit interrupt is cleared by performing a single write to the transmit buffer, by reading the interrupt index from IIDX, or by writing a 1 to the TXINT bit in ICLR.

#### 31.2.14.2 DMA Trigger Publisher (DMA_TRIG_RX, DMA_TRIG_TX)

The DMA_TRIG_RX and DMA_TRIG_TX registers are used to configure trigger signaling for DMA transfers. These registers can be programmed to initiate DMA operations for receive or transmit events based on trigger conditions defined in the corresponding peripheral registers. (PERIPHERAL.DMA_TRIG_RX.RIS and PERIPHERAL.DMA_TRIG_TX.RIS).

The DMA_TRIG_RX register triggers the DMA to perform receive data transfers, while the DMA_TRIG_TX register triggers the DMA to perform transmit data transfers.

The DMA trigger event configuration is managed with the DMA_TRIG_RX and DMA_TRIG_TX event management registers. See DMA chapter to understand how DMA works.

**DMA configuration for M0+**

For reading received data through DMA:

1. Select the DMA n-channel to be used for transfer – refer to the MSPM0's data sheet for the number of DMA channels available
2. Configure trigger source for the selected DMA channel to be the UNICOMM-SPI's Rx event in the DMA_CH[n].DMATCTL register
3. Configure the UNICOMM-SPI's INT_EVENTx register to generate a DMA request on a UNICOMM-SPI's Rx event
4. Configure the DMA source address as the UNICOMM-SPI's RxFIFO in the DMA_CH[n].DMASA register



Copyright © 2026 Texas Instruments Incorporated

5. Configure the DMA destination address as the SRAM location where data is to be copied in the DMA_CH[n].DMADA register
6. Configure all the triggers for the DMA to be serviced in the DMA_CH[n].DMASZ register.
7. In the DMA_CH[n].DMACTL register:
   a. Configure the transfer's size to be done on each trigger to DMA in the DMADSTWDTH and DMASRCWDTH bit-fields – possible options are 8 or 16 bits
   b. Configure DMASRCINCR to No Change (0h) and DMADSTINCR to 8 or 16 (same as what used for DMADSTWIDTH and DMASRCWIDTH)
   c. Enable the DMA channel by writing 1 to the DMAEN register

For writing data to be transmitted by UNICOMM through DMA:

1. Select the DMA n-channel to be used for transfer – refer to the MSPM0's datasheet for the number of DMA channels available
2. Configure trigger source for the selected DMA channel to be the UNICOMM-SPI's Tx event in the DMA_CH[n].DMATCTL register
3. Configure the UNICOMM-SPI's INT_EVENTx register to generate a DMA request on a UNICOMM-SPI's Tx event
4. Configure the SRAM location where data to be transmitted is present as the DMA source address in the DMA_CH[n].DMASA register
5. Configure the UNICOMM-SPI's Tx FIFO as the DMA destination address in the DMA_CH[n].DMADA register
6. Configure all the triggers for the DMA to be serviced in the DMA_CH[n].DMASZ register.
7. In the DMA_CH[n].DMACTL register:
   a. Configure the transfer's size to be done on each trigger to DMA in the DMADSTWDTH and DMASRCWDTH bit-fields – possible options are 8 or 16 bits
   b. Configure DMADSTINCR to No Change (0h) and DMASRCINCR to 8 or 16 (same as what used for DMADSTWIDTH and DMASRCWIDTH)
   c. Enable the DMA channel by writing 1 to the DMAEN register

### *31.2.15 Emulation Modes*

The UNICOMM-SPI module behavior while the device is in debug mode is controlled by the FREE and SOFT bits in the PDBGCTL register.

When the device is in debug mode and set into halt mode, the following behavior can be configured.

**Table 31-6. Debug Mode Peripheral Behavior**

| PDBGCTL.FREE | PDBGCTL.SOFT | Function |
|---|---|---|
| **1** | **x** | Modules continues operation |
| **0** | **1** | Module stops after the next transfer has been finished |
| **0** | **0** | Not supported |

#### 31.2.15.1 Graceful Halt

When emulation mode is set to graceful halt, and a halt condition occurs, the IP will stop communications after the ongoing frame has finished.

Transmitter

- The UNICOMM-SPI Module will finish transmitting the ongoing frame; FSM will return to IDLE and stop.
- The external line will be driven to the IDLE condition.
- Removing the halt will resume communication.

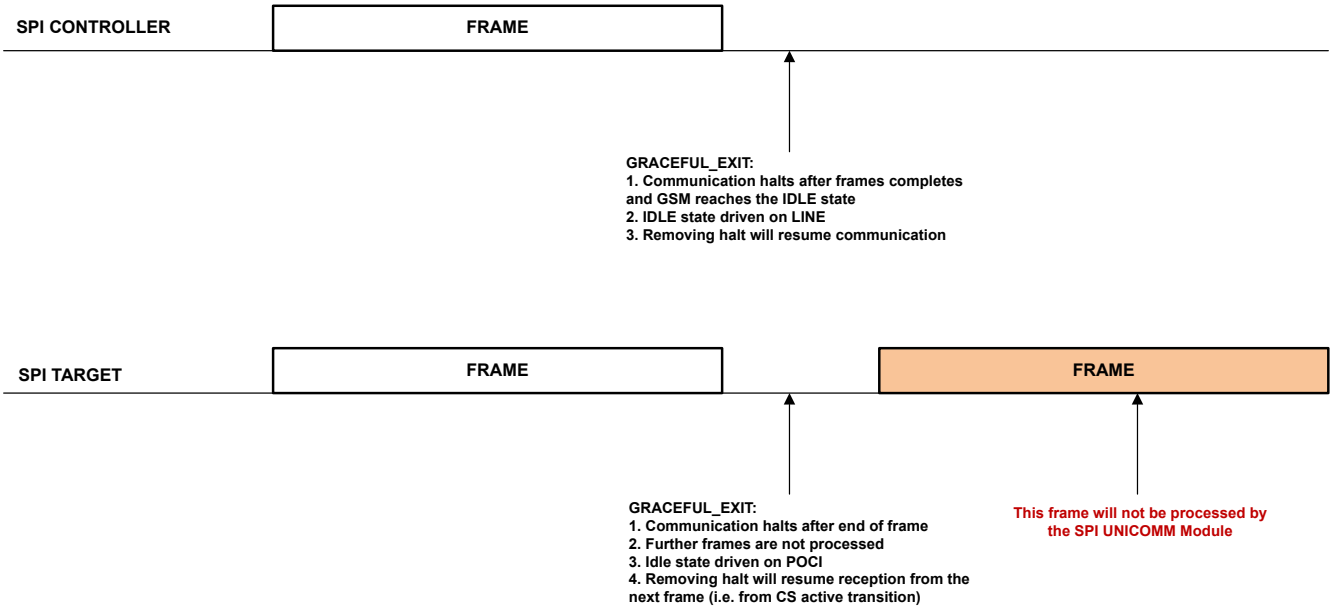
Receiver

- The UNICOMM-SPI Module will receive an ongoing frame; FSM will return to IDLE and stop
- Any new messages will not be processed



Copyright © 2026 Texas Instruments Incorporated

- After the halt is removed, FSM will begin communications
- The FSM may start reception in the middle of a frame






Copyright © 2026 Texas Instruments Incorporated

## 31.3 UNICOMMSPI Registers

Table 31-7 lists the memory-mapped registers for the UNICOMMSPI registers. All register offset addresses not listed in Table 31-7 should be considered as reserved locations and the register contents should not be modified.

**Table 31-7. UNICOMMSPI Registers**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 1000h | CLKDIV | Clock Divider | Section 31.3.1 |
| 1008h | CLKSEL | Clock Select for Ultra Low Power peripherals | Section 31.3.2 |
| 1020h | IIDX | Interrupt Index Register | Section 31.3.3 |
| 1028h | IMASK | Interrupt mask | Section 31.3.4 |
| 1030h | RIS | Raw interrupt status | Section 31.3.5 |
| 1038h | MIS | Masked interrupt status | Section 31.3.6 |
| 1040h | ISET | Interrupt set | Section 31.3.7 |
| 1048h | ICLR | Interrupt clear | Section 31.3.8 |
| 1058h | IMASK | Interrupt mask | Section 31.3.9 |
| 1060h | RIS | Raw interrupt status | Section 31.3.10 |
| 1068h | MIS | Masked interrupt status | Section 31.3.11 |
| 1070h | ISET | Interrupt set | Section 31.3.12 |
| 1088h | IMASK | Interrupt mask | Section 31.3.13 |
| 1090h | RIS | Raw interrupt status | Section 31.3.14 |
| 1098h | MIS | Masked interrupt status | Section 31.3.15 |
| 10A0h | ISET | Interrupt set | Section 31.3.16 |
| 10E4h | INTCTL | Interrupt control register | Section 31.3.17 |
| 1100h | CTL0 | SPI control register 0 | Section 31.3.18 |
| 1108h | STAT | Status Register | Section 31.3.19 |
| 110Ch | IFLS | Interrupt FIFO Level Select Register | Section 31.3.20 |
| 1110h | CLKCTL | Clock prescaler and divider register. | Section 31.3.21 |
| 1120h | TXDATA | TXDATA Register | Section 31.3.22 |
| 1124h | RXDATA | RXDATA Register | Section 31.3.23 |
| 114Ch | CTL1 | SPI control register 1 | Section 31.3.24 |
| 1150h | SPI_CNTR | SPI Counter | Section 31.3.25 |

Complex bit access types are encoded to fit into small table cells. Table 31-8 shows the codes that are used for access types in this section.

**Table 31-8. UNICOMMSPI Access Type Codes**

| Access Type | Code | Description |
|---|---|---|
| Read Type | | |
| R | R | Read |
| Write Type | | |
| W | W | Write |
| Reset or Default Value | | |
| *-n* | | Value after reset or the default value |



Copyright © 2026 Texas Instruments Incorporated

### 31.3.1 CLKDIV Register (Offset = 1000h) [Reset = 00000000h]

CLKDIV is shown in Figure 31-8 and described in Table 31-9.

Return to the Summary Table.

SPI Clock Divider. This register is used to specify module-specific divide ratio of the functional clock

**Figure 31-8. CLKDIV Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | | RATIO | | |
| R-0h | | | | | | | | | | | | | R/W-0h | | |

**Table 31-9. CLKDIV Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | RESERVED | R | 0h | |
| 2-0 | RATIO | R/W | 0h | Selects divide ratio of module clock<br>0h = Do not divide clock source<br>1h = Divide clock source by 2<br>2h = Divide clock source by 3<br>3h = Divide clock source by 4<br>4h = Divide clock source by 5<br>5h = Divide clock source by 6<br>6h = Divide clock source by 7<br>7h = Divide clock source by 8 |



Copyright © 2026 Texas Instruments Incorporated

### 31.3.2 CLKSEL Register (Offset = 1008h) [Reset = 00000000h]

CLKSEL is shown in Figure 31-9 and described in Table 31-10.

Return to the Summary Table.

SPI Source Clock Selection.

**Figure 31-9. CLKSEL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | ASYNC_PLL_SEL | ASYNC_HFCLK_SEL | ASYNC_SYSCLK_SEL | ASYNC_LFCLK_SEL |
| R-0h | | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | BUSCLK_SEL | MFCLK_SEL | LFCLK_SEL | RESERVED |
| R-0h | | | | R/W-0h | R/W-0h | R/W-0h | R-0h |

**Table 31-10. CLKSEL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-12 | RESERVED | R | 0h | |
| 11 | ASYNC_PLL_SEL | R/W | 0h | Asynchronous PLL selected as source<br>0h = Does not select this clock as a source<br>1h = Select this clock as a source |
| 10 | ASYNC_HFCLK_SEL | R/W | 0h | Asynchronous HFCLK selected as source<br>0h = Does not select this clock as a source<br>1h = Select this clock as a source |
| 9 | ASYNC_SYSCLK_SEL | R/W | 0h | Asynchronous sys clock selected as source<br>0h = Does not select this clock as a source<br>1h = Select this clock as a source |
| 8 | ASYNC_LFCLK_SEL | R/W | 0h | Asynchronous lfclk selected as source<br>0h = Does not select this clock as a source<br>1h = Select this clock as a source |
| 7-4 | RESERVED | R | 0h | |
| 3 | BUSCLK_SEL | R/W | 0h | Selects busclk as clock source if enabled<br>0h = Does not select this clock as a source<br>1h = Select this clock as a source |
| 2 | MFCLK_SEL | R/W | 0h | Selects MFCLK as clock source if enabled<br>0h = Does not select this clock as a source<br>1h = Select this clock as a source |
| 1 | LFCLK_SEL | R/W | 0h | Selects LFCLK as clock source if enabled<br>0h = Does not select this clock as a source<br>1h = Select this clock as a source |
| 0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 31.3.3 IIDX Register (Offset = 1020h) [Reset = 00000000h]

IIDX is shown in Figure 31-10 and described in Table 31-11.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. Value 0x00 means no event pending. Interrupt 1 is the highest priority, IIDX next highest, 4, 8, … IIDX^31 is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred. On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in [RIS] and [MIS] are cleared as well. After a read from the CPU (not from the debug interface), the register is updated with the next highest priority interrupt, if none are pending, then it should display 0x0.

**Figure 31-10. IIDX Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | STAT | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | |

**Table 31-11. IIDX Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | STAT | R | 0h | Interrupt index status<br>00h = No interrupt pending<br>1h = Transmit Parity Event/interrupt pending<br>2h = RX FIFO Overflow Event/interrupt pending<br>3h = Receive interrupt<br>4h = RX FIFO Full Interrupt<br>5h = TX FIFO underflow interrupt<br>6h = Transmit Event<br>7h = Transmit Buffer Empty Event/interrupt pending<br>9h = End of Transmit Event/interrupt pending<br>Ah = SPI receive time-out interrupt<br>10h = DMA DONE on RX<br>11h = DMA DONE on TX<br>13h = DMA PRE IRQ RX INTERRUPT<br>14h = DMA PRE IRQ TX INTERRUPT<br>15h = SPI line time-out interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 31.3.4 IMASK Register (Offset = 1028h) [Reset = 00000000h]

IMASK is shown in Figure 31-11 and described in Table 31-12.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

**Figure 31-11. IMASK Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | LTOUT | RESERVED | RESERVED | RESERVED | DMA_DONE_TX |
| R-0h | | | R/W-0h | R-0h | R-0h | R-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DMA_DONE_RX | RESERVED | | | | | RTOUT | IDLE |
| R/W-0h | R-0h | | | | | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | TXEMPTY | TX | TXFIFO_UNF | RXFULL | RX | RXFIFO_OVF | PER |
| R-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

**Table 31-12. IMASK Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-21 | RESERVED | R | 0h | |
| 20 | LTOUT | R/W | 0h | SPI Line Time-Out Interrupt Mask. This interrupt is raised when the UNICOMM-SPI is in peripheral mode, and no new transaction has been detected for the timeout period programmed by the CTL1.RXTIMEOUT field.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 19 | RESERVED | R | 0h | Reserved |
| 18 | RESERVED | R | 0h | Reserved |
| 17 | RESERVED | R | 0h | |
| 16 | DMA_DONE_TX | R/W | 0h | SPI DMA Done on TX Event Channel Interrupt Mask. This interrupt is raised when the DONE signal is sent to the SPI from a DMA indicating all the transactions have completed.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 15 | DMA_DONE_RX | R/W | 0h | SPI DMA Done on RX Event Channel Interrupt Mask. This interrupt is raised when the DONE signal is sent to the SPI from a DMA indicating all the transactions have completed.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 14-10 | RESERVED | R | 0h | |
| 9 | RTOUT | R/W | 0h | SPI Receive Time-Out Interrupt Mask. This interrupt is raised when the UNICOMM-SPI is in peripheral mode receiving data, the SCLK line is inactive while the RX FIFO contains at least one element for the timeout period programmed by the CTL1.RXTIMEOUT field, regardless of if the RX FIFO level configured in the IFLS register has been reached.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |



Copyright © 2026 Texas Instruments Incorporated

**Table 31-12. IMASK Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 8 | IDLE | R/W | 0h | SPI Idle Interrupt Mask. This interrupt is raised when the SPI has finished transfers and gone into an idle state.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 7 | RESERVED | R | 0h | |
| 6 | TXEMPTY | R/W | 0h | SPI TX FIFO Empty Interrupt Mask. This interrupt is raised when all data has been shifted out of the TX FIFO and the SPI transmitter goes idle.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 5 | TX | R/W | 0h | SPI Transmit Interrupt Mask. This interrupt is raised when the TX FIFO level has met the condition configured in the IFLS register. For variants with only a Buffer, indicates that the buffer is empty.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 4 | TXFIFO_UNF | R/W | 0h | SPI TX FIFO Underflow Interrupt Mask. This interrupt is raised when the UNICOMM-SPI (in peripheral mode) gets clock edges on the SCLK line from the SPI controller, but the TX FIFO is empty.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 3 | RXFULL | R/W | 0h | SPI RX FIFO Full Interrupt Mask. This interrupt is raised when the RX FIFO is full.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 2 | RX | R/W | 0h | SPI Receive Interrupt Mask. This interrupt is raised when the RX FIFO level has met the condition configured in the IFLS register. For variants with only a Buffer, indicates that the buffer has received a character.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 1 | RXFIFO_OVF | R/W | 0h | SPI RX FIFO Overflow Interrupt Mask. This interrupt is raised when the SPI receiver tries to receive data, but the SPI RX FIFO is full. In this case, new data will be discarded/lost by the SPI receiver.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 0 | PER | R/W | 0h | SPI Parity Error Interrupt Mask. This interrupt is raised when a character is received with a mismatch between the calculated parity and the received parity.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |



Copyright © 2026 Texas Instruments Incorporated

### 31.3.5 RIS Register (Offset = 1030h) [Reset = 00000000h]

RIS is shown in Figure 31-12 and described in Table 31-13.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Figure 31-12. RIS Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | LTOUT | RESERVED | RESERVED | RESERVED | DMA_DONE_TX |
| R-0h | | | R-0h | R-0h | R-0h | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DMA_DONE_RX | RESERVED | | | | | RTOUT | IDLE |
| R-0h | R-0h | | | | | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | TXEMPTY | TX | TXFIFO_UNF | RXFULL | RX | RXFIFO_OVF | PER |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 31-13. RIS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-21 | RESERVED | R | 0h | |
| 20 | LTOUT | R | 0h | SPI Line Time-Out Interrupt Raw Status. This interrupt is raised when the UNICOMM-SPI is in peripheral mode, and no new transaction has been detected for the timeout period programmed by the CTL1.RXTIMEOUT field.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 19 | RESERVED | R | 0h | Reserved |
| 18 | RESERVED | R | 0h | Reserved |
| 17 | RESERVED | R | 0h | |
| 16 | DMA_DONE_TX | R | 0h | SPI DMA Done on TX Event Channel Interrupt Raw Status. This interrupt is raised when the DONE signal is sent to the SPI from a DMA indicating all the transactions have completed.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 15 | DMA_DONE_RX | R | 0h | SPI DMA Done on RX Event Channel Interrupt Raw Status. This interrupt is raised when the DONE signal is sent to the SPI from a DMA indicating all the transactions have completed.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 14-10 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

**Table 31-13. RIS Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 9 | RTOUT | R | 0h | SPI Receive Time-Out Interrupt Raw Status. This interrupt is raised when the UNICOMM-SPI is in peripheral mode receiving data, the SCLK line is inactive while the RX FIFO contains at least one element for the timeout period programmed by the CTL1.RXTIMEOUT field, regardless of if the RX FIFO level configured in the IFLS register has been reached.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 8 | IDLE | R | 0h | SPI Idle Interrupt Raw Status. This interrupt is raised when the SPI has finished transfers and gone into an idle state.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 7 | RESERVED | R | 0h | |
| 6 | TXEMPTY | R | 0h | SPI TX FIFO Empty Interrupt Raw Status. This interrupt is raised when all data has been shifted out of the TX FIFO and the SPI transmitter goes idle.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 5 | TX | R | 0h | SPI Transmit Interrupt Raw Status. This interrupt is raised when the TX FIFO level has met the condition configured in the IFLS register. For variants with only a Buffer, indicates that the buffer is empty.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 4 | TXFIFO_UNF | R | 0h | SPI TX FIFO Underflow Interrupt Raw Status. This interrupt is raised when the UNICOMM-SPI (in peripheral mode) gets clock edges on the SCLK line from the SPI controller, but the TX FIFO is empty.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 3 | RXFULL | R | 0h | SPI RX FIFO Full Interrupt Raw Status. This interrupt is raised when the RX FIFO is full.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 2 | RX | R | 0h | SPI Receive Interrupt Raw Status. This interrupt is raised when the RX FIFO level has met the condition configured in the IFLS register. For variants with only a Buffer, indicates that the buffer has received a character.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 1 | RXFIFO_OVF | R | 0h | SPI RX FIFO Overflow Interrupt Raw Status. This interrupt is raised when the SPI receiver tries to receive data, but the SPI RX FIFO is full. In this case, new data will be discarded/lost by the SPI receiver.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 0 | PER | R | 0h | SPI Parity Error Interrupt Raw Status. This interrupt is raised when a character is received with a mismatch between the calculated parity and the received parity.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

### 31.3.6 MIS Register (Offset = 1038h) [Reset = 00000000h]

MIS is shown in Figure 31-13 and described in Table 31-14.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 31-13. MIS Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | LTOUT | RESERVED | RESERVED | RESERVED | DMA_DONE_TX |
| R-0h | | | R-0h | R-0h | R-0h | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DMA_DONE_RX | RESERVED | | | | | RTOUT | IDLE |
| R-0h | R-0h | | | | | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | TXEMPTY | TX | TXFIFO_UNF | RXFULL | RX | RXFIFO_OVF | PER |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 31-14. MIS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-21 | RESERVED | R | 0h | |
| 20 | LTOUT | R | 0h | SPI Line Time-Out Interrupt Masked Status. This interrupt is raised when the UNICOMM-SPI is in peripheral mode, and no new transaction has been detected for the timeout period programmed by the CTL1.RXTIMEOUT field.<br>0h = Interrupt did not occur or mask was not enabled<br>1h = Interrupt occured |
| 19 | RESERVED | R | 0h | Reserved |
| 18 | RESERVED | R | 0h | Reserved |
| 17 | RESERVED | R | 0h | |
| 16 | DMA_DONE_TX | R | 0h | SPI DMA Done on TX Event Channel Interrupt Masked Status. This interrupt is raised when the DONE signal is sent to the SPI from a DMA indicating all the transactions have completed.<br>0h = Interrupt did not occur or mask was not enabled<br>1h = Interrupt occurred |
| 15 | DMA_DONE_RX | R | 0h | SPI DMA Done on RX Event Channel Interrupt Masked Status. This interrupt is raised when the DONE signal is sent to the SPI from a DMA indicating all the transactions have completed.<br>0h = Interrupt did not occur or mask was not enabled<br>1h = Interrupt occurred |
| 14-10 | RESERVED | R | 0h | |
| 9 | RTOUT | R | 0h | SPI Receive Time-Out Interrupt Masked Status. This interrupt is raised when the UNICOMM-SPI is in peripheral mode receiving data, the SCLK line is inactive while the RX FIFO contains at least one element for the timeout period programmed by the CTL1.RXTIMEOUT field, regardless of if the RX FIFO level configured in the IFLS register has been reached.<br>0h = Interrupt did not occur or mask was not enabled<br>1h = Interrupt occured |



Copyright © 2026 Texas Instruments Incorporated

**Table 31-14. MIS Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 8 | IDLE | R | 0h | SPI Idle Interrupt Masked Status. This interrupt is raised when the SPI has finished transfers and gone into an idle state.<br>0h = Interrupt did not occur or mask was not enabled<br>1h = Interrupt occurred |
| 7 | RESERVED | R | 0h | |
| 6 | TXEMPTY | R | 0h | SPI TX FIFO Empty Interrupt Masked Status. This interrupt is raised when all data has been shifted out of the TX FIFO and the SPI transmitter goes idle.<br>0h = Interrupt did not occur or mask was not enabled<br>1h = Interrupt occurred |
| 5 | TX | R | 0h | SPI Transmit Interrupt Masked Status. This interrupt is raised when the TX FIFO level has met the condition configured in the IFLS register. For variants with only a Buffer, indicates that the buffer is empty.<br>0h = Interrupt did not occur or mask was not enabled<br>1h = Interrupt occurred |
| 4 | TXFIFO_UNF | R | 0h | SPI TX FIFO Underflow Interrupt Masked Status. This interrupt is raised when the UNICOMM-SPI (in peripheral mode) gets clock edges on the SCLK line from the SPI controller, but the TX FIFO is empty.<br>0h = Interrupt did not occur or mask was not enabled<br>1h = Interrupt occurred |
| 3 | RXFULL | R | 0h | SPI RX FIFO Full Interrupt Masked Status. This interrupt is raised when the RX FIFO is full.<br>0h = Interrupt did not occur or mask was not enabled<br>1h = Interrupt occurred |
| 2 | RX | R | 0h | SPI Receive Interrupt Masked Status. This interrupt is raised when the RX FIFO level has met the condition configured in the IFLS register. For variants with only a Buffer, indicates that the buffer has received a character.<br>0h = Interrupt did not occur or mask was not enabled<br>1h = Interrupt occurred |
| 1 | RXFIFO_OVF | R | 0h | SPI RX FIFO Overflow Interrupt Masked Status. This interrupt is raised when the SPI receiver tries to receive data, but the SPI RX FIFO is full. In this case, new data will be discarded/lost by the SPI receiver.<br>0h = Interrupt did not occur or mask was not enabled<br>1h = Interrupt occurred |
| 0 | PER | R | 0h | SPI Parity Error Interrupt Masked Status. This interrupt is raised when a character is received with a mismatch between the calculated parity and the received parity.<br>0h = Interrupt did not occur or mask was not enabled<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

### 31.3.7 ISET Register (Offset = 1040h) [Reset = 00000000h]

ISET is shown in Figure 31-14 and described in Table 31-15.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 31-14. ISET Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | LTOUT | RESERVED | RESERVED | RESERVED | DMA_DONE_TX |
| R-0h | | | W-0h | R-0h | R-0h | R-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DMA_DONE_RX | RESERVED | | | | | RTOUT | IDLE |
| W-0h | R-0h | | | | | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | TXEMPTY | TX | TXFIFO_UNF | RXFULL | RX | RXFIFO_OVF | PER |
| R-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

**Table 31-15. ISET Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-21 | RESERVED | R | 0h | |
| 20 | LTOUT | W | 0h | Set SPI Line Time-Out Interrupt. This interrupt is raised when the UNICOMM-SPI is in peripheral mode, and no new transaction has been detected for the timeout period programmed by the CTL1.RXTIMEOUT field.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 19 | RESERVED | R | 0h | Reserved |
| 18 | RESERVED | R | 0h | Reserved |
| 17 | RESERVED | R | 0h | |
| 16 | DMA_DONE_TX | W | 0h | Set SPI DMA Done on TX Event Channel Interrupt. This interrupt is raised when the DONE signal is sent to the SPI from a DMA indicating all the transactions have completed.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 15 | DMA_DONE_RX | W | 0h | Set SPI DMA Done on RX Event Channel Interrupt. This interrupt is raised when the DONE signal is sent to the SPI from a DMA indicating all the transactions have completed.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 14-10 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

**Table 31-15. ISET Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 9 | RTOUT | W | 0h | Set SPI Receive Time-Out Interrupt. This interrupt is raised when the UNICOMM-SPI is in peripheral mode receiving data, the SCLK line is inactive while the RX FIFO contains at least one element for the timeout period programmed by the CTL1.RXTIMEOUT field, regardless of if the RX FIFO level configured in the IFLS register has been reached.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 8 | IDLE | W | 0h | Set SPI Idle Interrupt. This interrupt is raised when the SPI has finished transfers and gone into an idle state.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 7 | RESERVED | R | 0h | |
| 6 | TXEMPTY | W | 0h | Set SPI TX FIFO Empty Interrupt. This interrupt is raised when all data has been shifted out of the TX FIFO and the SPI transmitter goes idle.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 5 | TX | W | 0h | Set SPI Transmit Interrupt. This interrupt is raised when the TX FIFO level has met the condition configured in the IFLS register. For variants with only a Buffer, indicates that the buffer is empty.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 4 | TXFIFO_UNF | W | 0h | Set SPI TX FIFO Underflow Interrupt. This interrupt is raised when the UNICOMM-SPI (in peripheral mode) gets clock edges on the SCLK line from the SPI controller, but the TX FIFO is empty.<br>0h = Writing has no effect<br>1h = Set interrupt |
| 3 | RXFULL | W | 0h | Set SPI RX FIFO Full Interrupt. This interrupt is raised when the RX FIFO is full.<br>0h = Writing has no effect<br>1h = Set Interrupt |
| 2 | RX | W | 0h | Set SPI Receive Interrupt. This interrupt is raised when the RX FIFO level has met the condition configured in the IFLS register. For variants with only a Buffer, indicates that the buffer has received a character.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 1 | RXFIFO_OVF | W | 0h | Set SPI RX FIFO Overflow Interrupt. This interrupt is raised when the SPI receiver tries to receive data, but the SPI RX FIFO is full. In this case, new data will be discarded/lost by the SPI receiver.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 0 | PER | W | 0h | Set SPI Parity Error Interrupt. This interrupt is raised when a character is received with a mismatch between the calculated parity and the received parity.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 31.3.8 ICLR Register (Offset = 1048h) [Reset = 00000000h]

ICLR is shown in Figure 31-15 and described in Table 31-16.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

**Figure 31-15. ICLR Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | LTOUT | RESERVED | RESERVED | RESERVED | DMA_DONE_TX |
| R-0h | | | W-0h | R-0h | R-0h | R-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DMA_DONE_RX | RESERVED | | | | | RTOUT | IDLE |
| W-0h | R-0h | | | | | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | TXEMPTY | TX | TXFIFO_UNF | RXFULL | RX | RXFIFO_OVF | PER |
| R-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

**Table 31-16. ICLR Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-21 | RESERVED | R | 0h | |
| 20 | LTOUT | W | 0h | Clear SPI Line Time-Out Interrupt. This interrupt is raised when the UNICOMM-SPI is in peripheral mode, and no new transaction has been detected for the timeout period programmed by the CTL1.RXTIMEOUT field.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 19 | RESERVED | R | 0h | Reserved |
| 18 | RESERVED | R | 0h | Reserved |
| 17 | RESERVED | R | 0h | |
| 16 | DMA_DONE_TX | W | 0h | Clear SPI DMA Done on TX Event Channel Interrupt. This interrupt is raised when the DONE signal is sent to the SPI from a DMA indicating all the transactions have completed.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 15 | DMA_DONE_RX | W | 0h | Clear SPI DMA Done on RX Event Channel Interrupt. This interrupt is raised when the DONE signal is sent to the SPI from a DMA indicating all the transactions have completed.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 14-10 | RESERVED | R | 0h | |
| 9 | RTOUT | W | 0h | Clear SPI Receive Time-Out Interrupt. This interrupt is raised when the UNICOMM-SPI is in peripheral mode receiving data, the SCLK line is inactive while the RX FIFO contains at least one element for the timeout period programmed by the CTL1.RXTIMEOUT field, regardless of if the RX FIFO level configured in the IFLS register has been reached.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |



Copyright © 2026 Texas Instruments Incorporated

**Table 31-16. ICLR Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 8 | IDLE | W | 0h | Clear SPI Idle Interrupt. This interrupt is raised when the SPI has finished transfers and gone into an idle state.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 7 | RESERVED | R | 0h | |
| 6 | TXEMPTY | W | 0h | Clear SPI TX FIFO Empty Interrupt. This interrupt is raised when all data has been shifted out of the TX FIFO and the SPI transmitter goes idle.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 5 | TX | W | 0h | Clear SPI Transmit Interrupt. This interrupt is raised when the TX FIFO level has met the condition configured in the IFLS register. For variants with only a Buffer, indicates that the buffer is empty.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 4 | TXFIFO_UNF | W | 0h | Clear SPI TX FIFO Underflow Interrupt. This interrupt is raised when the UNICOMM-SPI (in peripheral mode) gets clock edges on the SCLK line from the SPI controller, but the TX FIFO is empty.<br>0h = Writing has no effect<br>1h = Clear interrupt |
| 3 | RXFULL | W | 0h | Clear SPI RX FIFO Full Interrupt. This interrupt is raised when the RX FIFO is full.<br>0h = Writing has no effect<br>1h = Clear interrupt |
| 2 | RX | W | 0h | Clear SPI Receive Interrupt. This interrupt is raised when the RX FIFO level has met the condition configured in the IFLS register. For variants with only a Buffer, indicates that the buffer has received a character.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 1 | RXFIFO_OVF | W | 0h | Clear SPI RX FIFO Overflow Interrupt. This interrupt is raised when the SPI receiver tries to receive data, but the SPI RX FIFO is full. In this case, new data will be discarded/lost by the SPI receiver.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 0 | PER | W | 0h | Clear SPI Parity Error Interrupt. This interrupt is raised when a character is received with a mismatch between the calculated parity and the received parity.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 31.3.9 IMASK Register (Offset = 1058h) [Reset = 00000000h]

IMASK is shown in Figure 31-16 and described in Table 31-17.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

**Figure 31-16. IMASK Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | RTOUT | RESERVED |
| R-0h | | | | | | R/W-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | RX | RESERVED | |
| R-0h | | | | | R/W-0h | R-0h | |

**Table 31-17. IMASK Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-10 | RESERVED | R | 0h | |
| 9 | RTOUT | R/W | 0h | SPI Receive Timeout DMA Trigger Mask. The DMA is triggered when the UNICOMM-SPI is in peripheral mode receiving data, the SCLK line is inactive while the RX FIFO contains at least one element for the timeout period programmed by the CTL1.RXTIMEOUT field, regardless of if the RX FIFO level configured in the IFLS register has been reached.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 8-3 | RESERVED | R | 0h | |
| 2 | RX | R/W | 0h | SPI Receive DMA Trigger Mask. The DMA is triggered when the RX FIFO condition programmed by the IFLS register is met. When FIFO is not present, this indicates that the buffer is full.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 1-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 31.3.10 RIS Register (Offset = 1060h) [Reset = 00000000h]

RIS is shown in Figure 31-17 and described in Table 31-18.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Figure 31-17. RIS Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | RTOUT | RESERVED |
| R-0h | | | | | | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | RX | RESERVED | |
| R-0h | | | | | R-0h | R-0h | |

**Table 31-18. RIS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-10 | RESERVED | R | 0h | |
| 9 | RTOUT | R | 0h | SPI Receive Timeout DMA Trigger Raw Status. The DMA is triggered when the UNICOMM-SPI is in peripheral mode receiving data, the SCLK line is inactive while the RX FIFO contains at least one element for the timeout period programmed by the CTL1.RXTIMEOUT field, regardless of if the RX FIFO level configured in the IFLS register has been reached.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 8-3 | RESERVED | R | 0h | |
| 2 | RX | R | 0h | SPI Receive DMA Trigger Raw Status. The DMA is triggered when the RX FIFO condition programmed by the IFLS register is met. When FIFO is not present, this indicates that the buffer is full.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 1-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 31.3.11 MIS Register (Offset = 1068h) [Reset = 00000000h]

MIS is shown in Figure 31-18 and described in Table 31-19.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 31-18. MIS Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | RTOUT | RESERVED |
| R-0h | | | | | | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | RX | RESERVED | |
| R-0h | | | | | R-0h | R-0h | |

**Table 31-19. MIS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-10 | RESERVED | R | 0h | |
| 9 | RTOUT | R | 0h | SPI Receive Timeout DMA Trigger Masked Status. The DMA is triggered when the UNICOMM-SPI is in peripheral mode receiving data, the SCLK line is inactive while the RX FIFO contains at least one element for the timeout period programmed by the CTL1.RXTIMEOUT field, regardless of if the RX FIFO level configured in the IFLS register has been reached.<br>0h = Interrupt did not occur or mask was not enabled.<br>1h = Interrupt Occurred |
| 8-3 | RESERVED | R | 0h | |
| 2 | RX | R | 0h | SPI Receive DMA Trigger Masked Status. The DMA is triggered when the RX FIFO condition programmed by the IFLS register is met. When FIFO is not present, this indicates that the buffer is full.<br>0h = Interrupt did not occur or mask was not enabled<br>1h = Interrupt occurred |
| 1-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 31.3.12 ISET Register (Offset = 1070h) [Reset = 00000000h]

ISET is shown in Figure 31-19 and described in Table 31-20.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 31-19. ISET Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | RTOUT | RESERVED |
| R-0h | | | | | | W-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | RX | RESERVED | |
| R-0h | | | | | W-0h | R-0h | |

**Table 31-20. ISET Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-10 | RESERVED | R | 0h | |
| 9 | RTOUT | W | 0h | Set SPI Receive Timeout DMA Trigger. The DMA is triggered when the UNICOMM-SPI is in peripheral mode receiving data, the SCLK line is inactive while the RX FIFO contains at least one element for the timeout period programmed by the CTL1.RXTIMEOUT field, regardless of if the RX FIFO level configured in the IFLS register has been reached.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt Mask |
| 8-3 | RESERVED | R | 0h | |
| 2 | RX | W | 0h | Set SPI Receive DMA Trigger. The DMA is triggered when the RX FIFO condition programmed by the IFLS register is met. When FIFO is not present, this indicates that the buffer is full.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 1-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 31.3.13 IMASK Register (Offset = 1088h) [Reset = 00000000h]

IMASK is shown in Figure 31-20 and described in Table 31-21.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

**Figure 31-20. IMASK Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | TX | RESERVED | | | | |
| R-0h | | | | | | | | | | R/W-0h | R-0h | | | | |

**Table 31-21. IMASK Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | RESERVED | R | 0h | |
| 5 | TX | R/W | 0h | SPI Transmit DMA Trigger Mask. The DMA is triggered when the TX FIFO condition programmed by the IFLS register is met. When FIFO is not present, this indicates that the buffer is empty.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 4-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 31.3.14 RIS Register (Offset = 1090h) [Reset = 00000000h]

RIS is shown in Figure 31-21 and described in Table 31-22.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Figure 31-21. RIS Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | TX | RESERVED | | | | |
| R-0h | | | | | | | | | | R-0h | R-0h | | | | |

**Table 31-22. RIS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | RESERVED | R | 0h | |
| 5 | TX | R | 0h | SPI Transmit DMA Trigger Raw Status. The DMA is triggered when the TX FIFO condition programmed by the IFLS register is met. When FIFO is not present, this indicates that the buffer is empty.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 4-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 31.3.15 MIS Register (Offset = 1098h) [Reset = 00000000h]

MIS is shown in Figure 31-22 and described in Table 31-23.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 31-22. MIS Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | TX | RESERVED | | | | |
| R-0h | | | | | | | | | | R-0h | R-0h | | | | |

**Table 31-23. MIS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | RESERVED | R | 0h | |
| 5 | TX | R | 0h | SPI Transmit DMA Trigger Masked Status. The DMA is triggered when the TX FIFO condition programmed by the IFLS register is met. When FIFO is not present, this indicates that the buffer is empty.<br>0h = Interrupt did not occur or mask was not enabled.<br>1h = Interrupt occurred |
| 4-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 31.3.16 ISET Register (Offset = 10A0h) [Reset = 00000000h]

ISET is shown in Figure 31-23 and described in Table 31-24.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 31-23. ISET Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | TX | RESERVED | | | | |
| R-0h | | | | | | | | | | W-0h | R-0h | | | | |

**Table 31-24. ISET Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | RESERVED | R | 0h | |
| 5 | TX | W | 0h | Set SPI Transmit DMA Trigger. The DMA is triggered when the TX FIFO condition programmed by the IFLS register is met. When FIFO is not present, this indicates that the buffer is empty.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 4-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 31.3.17 INTCTL Register (Offset = 10E4h) [Reset = 00000000h]

INTCTL is shown in Figure 31-24 and described in Table 31-25.

Return to the Summary Table.

Interrupt Control Register.

**Figure 31-24. INTCTL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | INTEVAL |
| R-0h | | | | | | | W-0h |

**Table 31-25. INTCTL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | INTEVAL | W | 0h | Writing a 1 to this field re-evaluates the interrupt sources.<br>0h = Writing 0 has no effect<br>1h = Interrupt Eval |



Copyright © 2026 Texas Instruments Incorporated

### 31.3.18 CTL0 Register (Offset = 1100h) [Reset = 00000000h]

CTL0 is shown in Figure 31-25 and described in Table 31-26.

Return to the Summary Table.

SPI Control Register 0.

**Figure 31-25. CTL0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | CS_CLK_DLY | | | | | |
| R-0h | | R/W-0h | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CS3_INV | CS2_INV | CS1_INV | CS0_INV | SPI_CNTR_EN | CS_CLK_DLY_EN | CS_ROTATE_EN | DIS_CONT_FRAMES |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | CSCLR | CSSEL | | RESERVED | | SPH | SPO |
| R-0h | R/W-0h | R/W-0h | | R-0h | | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PACKEN | FRF | | DSS | | | | |
| R/W-0h | R/W-0h | | R/W-0h | | | | |

**Table 31-26. CTL0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | RESERVED | R | 0h | |
| 29-24 | CS_CLK_DLY | R/W | 0h | Clock Delay Between CS Assertion and first SCLK. Clock delay is equal to "value of CS_CLK_DLY+1" * FCLK number of cycles. |
| 23 | CS3_INV | R/W | 0h | CS0 Inversion<br>0h = CS is not inverted<br>1h = Invert CS |
| 22 | CS2_INV | R/W | 0h | CS0 Inversion<br>0h = CS is not inverted<br>1h = Invert CS |
| 21 | CS1_INV | R/W | 0h | CS0 Inversion<br>0h = CS is not inverted<br>1h = Invert CS |
| 20 | CS0_INV | R/W | 0h | CS0 Inversion<br>0h = CS is not inverted<br>1h = Invert CS |
| 19 | SPI_CNTR_EN | R/W | 0h | SPI Counter Enable<br>0h = Disable<br>1h = SPI Counter Enable |
| 18 | CS_CLK_DLY_EN | R/W | 0h | CS to SCLK Clock Delay Enable. 1. When enabled, CS to SCLK delay is determined by CS_CLK_DLY configuration 2. When disabled, CS to SCLK delay cannot be controlled through CS_CLK_DLY; it follows default SPI protocol<br>0h = Disable Rotation<br>1h = Enable CS rotation |



Copyright © 2026 Texas Instruments Incorporated

**Table 31-26. CTL0 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 17 | CS_ROTATE_EN | R/W | 0h | Chip select rotate enable Application must provide both Chip Select Number & Data as detailed below. Received data will also be packed as shown below 1. Transmit data : <6'b0, CS for respective frame><zero padding - if necessary, upto 24bits of data as per DSS> 2. Received data will be stored in the FIFO as: <6'b0, CS of frame received><zero padding - if necessary, upto 24bits of data as per DSS> Note: In this mode, packing feature will not be supported. Also, frame sizes upto 24 bits is only supported.<br>0h = Disable Rotation<br>1h = Enable CS rotation |
| 16 | DIS_CONT_FRAMES | R/W | 0h | Disable Continuous Frames 1. This is applicable only when SPH=1 & in Controller mode only. When enabled, CS will toggle after every frame, even if there is data loaded into TX FIFO<br>0h = CS remains active & does not toggle in case of continuous frames<br>1h = CS toggles between frames even if there is data in TX FIFO |
| 15 | RESERVED | R | 0h | |
| 14 | CSCLR | R/W | 0h | Clear shift register counter on CS inactive This bit is relevant only in the peripheral, CTL1.CP=0.<br>0h = Disable automatic clear of shift register when CS goes to disable.<br>1h = Enable automatic clear of shift register when CS goes to disable. |
| 13-12 | CSSEL | R/W | 0h | Select the CS line to control on data transfer This bit is applicable for both controller/target mode<br>0h (R/W) = CS line select: 0<br>1h (R/W) = CS line select: 1<br>2h (R/W) = CS line select: 2<br>3h (R/W) = CS line select: 3 |
| 11-10 | RESERVED | R | 0h | |
| 9 | SPH | R/W | 0h | CLKOUT phase (Motorola SPI frame format only) This bit selects the clock edge that captures data and enables it to change state. It has the most impact on the first bit transmitted by either permitting or not permitting a clock transition before the first data capture edge.<br>0h = Data is captured on the first clock edge transition.<br>1h = Data is captured on the second clock edge transition. |
| 8 | SPO | R/W | 0h | CLKOUT polarity (Motorola SPI frame format only)<br>0h = SPI produces a steady state LOW value on the CLKOUT<br>1h = SPI produces a steady state HIGH value on the CLKOUT |
| 7 | PACKEN | R/W | 0h | Packing Enable. When 1, packing feature is enabled inside the IP When 0, packing feature is disabled inside the IP<br>0h = Packing feature disabled<br>1h = Packing feature enabled |
| 6-5 | FRF | R/W | 0h | Frame format Select<br>0h = Motorola SPI frame format (3 wire mode)<br>1h = Motorola SPI frame format (4 wire mode)<br>2h = TI synchronous serial frame format<br>3h = Reserved - not to be selected |



Copyright © 2026 Texas Instruments Incorporated

**Table 31-26. CTL0 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 4-0 | DSS | R/W | 0h | Data Size Select. Values 0 - 2 are reserved and shall not be used. 3h = 4_BIT : 4-bit data SPI allows only values up to 16 Bit<br>3h (R/W) = Data Size Select bits: 4<br>4h (R/W) = Data Size Select bits: 5<br>5h (R/W) = Data Size Select bits: 6<br>6h (R/W) = Data Size Select bits: 7<br>7h (R/W) = Data Size Select bits: 8<br>8h (R/W) = Data Size Select bits: 9<br>9h (R/W) = Data Size Select bits: 10<br>Ah (R/W) = Data Size Select bits: 11<br>Bh (R/W) = Data Size Select bits: 12<br>Ch (R/W) = Data Size Select bits: 13<br>Dh (R/W) = Data Size Select bits: 14<br>Eh (R/W) = Data Size Select bits: 15<br>Fh (R/W) = Data Size Select bits: 16 |



Copyright © 2026 Texas Instruments Incorporated

### 31.3.19 STAT Register (Offset = 1108h) [Reset = 00000000h]

STAT is shown in Figure 31-26 and described in Table 31-27.

Return to the Summary Table.

SPI Status Register.

**Figure 31-26. STAT Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CDMODE | | | | RESERVED | | | BUSY |
| R-0h | | | | R-0h | | | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TXCLR | TXFF | TXFE | RXCLR | RXFF | RXFE | RESERVED | |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | |

**Table 31-27. STAT Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15-12 | CDMODE | R | 0h | Current CDMODE Status. This reflects the current counter value.<br>0h = Smallest value<br>0h = Manual mode: Data<br>Fh = Manual mode: Command |
| 11-9 | RESERVED | R | 0h | |
| 8 | BUSY | R | 0h | SPI Busy Status.<br>0h = SPI is in idle mode.<br>1h = SPI is currently transmitting and/or receiving data |
| 7 | TXCLR | R | 0h | SPI TX FIFO Clear Status.<br>0h = TX FIFO is not cleared<br>1h = TX FIFO clear is complete |
| 6 | TXFF | R | 0h | SPI TX FIFO Full Status.<br>0h = TX FIFO is not full<br>1h = TX FIFO is full |
| 5 | TXFE | R | 1h | TX FIFO Empty Status.<br>0h = TX FIFO is not empty<br>1h = TX FIFO is empty |
| 4 | RXCLR | R | 0h | SPI RX FIFO Clear Status.<br>0h = FIFO is not cleared<br>1h = FIFO clear is complete |
| 3 | RXFF | R | 0h | SPI RX FIFO Full Status.<br>0h = RX FIFO is not full<br>1h = RX FIFO is full |
| 2 | RXFE | R | 1h | RX FIFO Empty Status.<br>0h = RX FIFO is not empty<br>1h = RX FIFO is empty |
| 1-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 31.3.20 IFLS Register (Offset = 110Ch) [Reset = 00000022h]

IFLS is shown in Figure 31-27 and described in Table 31-28.

Return to the Summary Table.

Interrupt FIFO Level Select. The IFLS register is the interrupt FIFO level select register. You can use this register to define the levels at which the TX, RX and timeout interrupt flags are triggered. The interrupts are generated based on a transition through a level rather than being based on the level. That is, the interrupts are generated when the fill level progresses through the trigger level. For example, if the receive trigger level is set to the half-way mark, the interrupt is triggered when the RX FIFO is filled with two or more characters. Out of reset, the TXIFLSEL and RXIFLSEL bits are configured so that the FIFOs trigger an interrupt at the half-way mark.

**Figure 31-27. IFLS Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RXCLR | RXIFLSEL | | | TXCLR | TXIFLSEL | | |
| R/W-0h | R/W-2h | | | R/W-0h | R/W-2h | | |

**Table 31-28. IFLS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7 | RXCLR | R/W | 0h | RX FIFO Clear. Setting this bit will clear the RX FIFO contents.<br>0h = Disable FIFO clear<br>1h = Enable FIFO Clear |
| 6-4 | RXIFLSEL | R/W | 2h | RX FIFO Level Select for generating events (interrupt/DMA). Note: for undefined settings the default configuration is used.<br>1h = RX FIFO >= 1/4 full<br>2h = RX FIFO >= 1/2 full (default)<br>3h = RX FIFO >= 3/4 full<br>4h = Opposite of empty<br>5h = RX FIFO is full<br>6h = RX_FIFO >= (MAX_FIFO_LEN -1)<br>7h = RX_FIFO <= 1 |
| 3 | TXCLR | R/W | 0h | TX FIFO Clear. Setting this bit will clear the TX FIFO contents.<br>0h = Disable FIFO clear<br>1h = Enable FIFO Clear |
| 2-0 | TXIFLSEL | R/W | 2h | TX FIFO Level Select for generating events (interrupt/DMA). Note: for undefined settings the default configuration is used.<br>1h = TX FIFO <= 3/4 empty<br>2h = TX FIFO <= 1/2 empty (default)<br>3h = TX FIFO <= 1/4 empty<br>4h = Opposite of full<br>5h = TX FIFO is empty<br>6h = TX FIFO <= 1<br>7h = TX_FIFO >= (MAX_FIFO_LEN -1) |



Copyright © 2026 Texas Instruments Incorporated

### 31.3.21 CLKCTL Register (Offset = 1110h) [Reset = 00000000h]

CLKCTL is shown in Figure 31-28 and described in Table 31-29.

Return to the Summary Table.

Clock prescaler and divider register. This register contains the settings for the Clock prescaler and divider settings.

**Figure 31-28. CLKCTL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DSAMPLE | | | | RESERVED | | | | | | | | | | | |
| R/W-0h | | | | R-0h | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | SCR | | | | | | | | | |
| R-0h | | | | | | R/W-0h | | | | | | | | | |

**Table 31-29. CLKCTL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-28 | DSAMPLE | R/W | 0h | Delayed Sampling Value. In controller mode, data on input pin will be delayed sampled by the defined clock cycles of internal functional clock (SPIclk) hence relaxing the setup time of input data. This setting is useful in systems where the board delays and external peripheral delays are more than the input setup time of the controller. Please refer to the datasheet for values of controller input setup time and assess what DSAMPLE value meets the requirement of the system. Note: High values of DSAMPLE can cause HOLD time violations and must be factored in the calculations.<br>0h = Delayed sampling is not used<br>Fh = Highest possible value |
| 27-10 | RESERVED | R | 0h | |
| 9-0 | SCR | R/W | 0h | Serial Clock Divider. This is used to generate the transmit and receive bit rate of the SPI. The SPI bit rate is (SPIclk)/((SCR+1)*2). SCR is a value from 0-1023.<br>0h = Smallest value<br>3FFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 31.3.22 TXDATA Register (Offset = 1120h) [Reset = 00000000h]

TXDATA is shown in Figure 31-29 and described in Table 31-30.

Return to the Summary Table.

TXDATA Register. Writing puts the data into the TX FIFO. Reading this register returns the last written value. When PACKEN=0, only the lower 16-bits of data written into the register is transferred to one 16-bits wide TX FIFO entry When PACKEN=1, upper and lower 16-bits of 32-bit write data are transferred to two16-bits wide TX FIFO entry

**Figure 31-29. TXDATA Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 31-30. TXDATA Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | W | 0h | Transmit Data When read, last written value will be returned. If the last write to this field was a 32-bit write (with PACKEN=1), 32-bits will be returned and if the last write was a 16-bit write (PACKEN=0), those 16-bits will be returned. When written, one or two FIFO entries will be written depending on PACKEN value. Data values are removed from the TX FIFO one value at a time by the transmit logic. It is loaded into the transmit serial shifter, then serially shifted out onto the TXD output pin at the programmed bit rate. When a data size of less than 16 bits is selected, the user must right-justify data written to the TX FIFO. The transmit logic ignores the unused bits.<br>0h = Smallest value<br>FFFFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 31.3.23 RXDATA Register (Offset = 1124h) [Reset = 00000000h]

RXDATA is shown in Figure 31-30 and described in Table 31-31.

Return to the Summary Table.

RXDATA Register. Reading this register returns value(s) from the RX FIFO. If the FIFO is empty the last read value is returned. Writing has not effect and is ignored. When PACKEN=1,two entries of the FIFO are returned as a 32-bit value. When PACKEN=0, 1 entry of FIFO is returned as 16-bit value.

**Figure 31-30. RXDATA Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 31-31. RXDATA Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | R | 0h | Received Data. When PACKEN=1,two entries of the FIFO are returned as a 32-bit value. When PACKEN=0, 1 entry of FIFO is returned as 16-bit value. As data values are removed by the receive logic from the incoming data frame, they are placed into the entry in the RX FIFO, pointed to by the current FIFO write pointer. Received data less than 16 bits is automatically right justified in the receive buffer.<br>0h = Smallest value<br>FFFFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 31.3.24 CTL1 Register (Offset = 114Ch) [Reset = 00000004h]

CTL1 is shown in Figure 31-31 and described in Table 31-32.

Return to the Summary Table.

SPI Control Register 1.

**Figure 31-31. CTL1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | RXTIMEOUT | | | | | |
| R-0h | | R/W-0h | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| REPEATTX | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CDMODE | | | | CDENABLE | RESERVED | SUSPEND | PTEN |
| R/W-0h | | | | R/W-0h | R-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | PES | PREN | MSB | POD | CP | LBM | ENABLE |
| R-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-1h | R/W-0h | R/W-0h |

**Table 31-32. CTL1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | RESERVED | R | 0h | |
| 29-24 | RXTIMEOUT | R/W | 0h | SPI Receive Interrupt Timeout Select (peripheral mode only). Configures the number of bits (at the programmed baud rate), used by the SPI peripheral when there is no SCLK pulse received, to flag/ trigger a timeout condition (RTOUT or LTOUT). A value of 0 disables the timeout functions.<br>0h = Smallest value<br>3Fh = Highest possible value |
| 23-16 | REPEATTX | R/W | 0h | Counter to repeat last transfer. 0: repeat last transfer is disabled. x: repeat the last transfer with the given number. The transfer will be started with writing a data into the TX Buffer. Sending the data will be repeated with the given value, so the data will be transferred X+1 times in total. The behavior is identical as if the data would be written into the TX Buffer that many times as defined by the value here. It can be used to clean a transfer or to pull a certain amount of data by a peripheral.<br>0h = Smallest value<br>FFh = Highest possible value |
| 15-12 | CDMODE | R/W | 0h | Command/Data Mode Value. When CTL1.CDENABLE is 1, CS3 line is used as C/D signal to distinguish between Command (C/D low) and Data (C/D high) information. When a value is written into the CTL1.CDMODE bits, the C/D (CS3) line will go low for the given numbers of byte which are sent by the SPI, starting with the next value to be transmitted after which, C/D line will go high automatically 0: Manual mode with C/D signal as High 1-14: C/D is low while this number of bytes are being sent after which, this field sets to 0 and C/D goes high. Reading this field at any time returns the remaining number of command bytes. 15: Manual mode with C/D signal as Low.<br>0h = Manual mode: Data<br>0h = Smallest value<br>Fh = Manual mode: Command |



Copyright © 2026 Texas Instruments Incorporated

**Table 31-32. CTL1 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 11 | CDENABLE | R/W | 0h | Command/Data Mode Enable.<br>0h = CS3 is used for Chip Select<br>1h = CS3 is used as CD signal |
| 10 | RESERVED | R | 0h | |
| 9 | SUSPEND | R/W | 0h | Suspend External Communication. When this bit is set, SPI communication on the external bus is suspended after the CS has returned to the idle state in controller mode, and after the CS is read as an idle state in peripheral mode.<br>0h = Functional mode resumed<br>1h = External communication suspended |
| 8 | PTEN | R/W | 0h | Parity Transmit Enable. If enabled, parity transmission will be done for both controller and peripheral modes.<br>0h = Parity transmission is disabled<br>1h = Parity transmission is enabled |
| 7 | RESERVED | R | 0h | Reserved |
| 6 | PES | R/W | 0h | Even Parity Select.<br>0h = Odd Parity mode<br>1h = Even Parity mode |
| 5 | PREN | R/W | 0h | Parity Receive Enable. If enabled, parity reception check will be done for both controller and peripheral modes In case of a parity miss-match the parity error flag RIS.PER will be set.<br>0h = Disable Parity receive function<br>1h = Enable Parity receive function |
| 4 | MSB | R/W | 0h | MSB First Select. Controls the direction of the receive and transmit shift register.<br>0h = LSB first<br>1h = MSB first |
| 3 | POD | R/W | 0h | Peripheral-mode: Data output disabled. This bit is relevant only in Peripheral mode. In multiple-peripheral system topologies, SPI controller can broadcast a message to all peripherals, while only one peripheral drives the line. POD can be used by the SPI peripheral to disable driving data on the line.<br>0h = SPI can drive the POCI output in peripheral mode.<br>1h = SPI cannot drive the POCI output in peripheral mode. |
| 2 | CP | R/W | 1h | Controller or Peripheral Mode Select. This bit can be modified only when SPI is disabled, CTL1.ENABLE=0.<br>0h = Select Peripheral mode<br>1h = Select Controller Mode |
| 1 | LBM | R/W | 0h | Loopback Mode Enable.<br>0h = Disable loopback mode<br>1h = Enable loopback mode |
| 0 | ENABLE | R/W | 0h | SPI Enable.<br>0h = Disable module function<br>1h = Enable module function |



Copyright © 2026 Texas Instruments Incorporated

### 31.3.25 SPI_CNTR Register (Offset = 1150h) [Reset = 00000000h]

SPI_CNTR is shown in Figure 31-32 and described in Table 31-33.

Return to the Summary Table.

SPI Counter 1. Application loads count value 2. When enabled, counter counts down to zero 3. Upon reaching "zero", a frame is transmitted; also, counter is reloaded to count value

**Figure 31-32. SPI_CNTR Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | SPICNT | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | R-0h | | | | | | | | | | | | | | | |

**Table 31-33. SPI_CNTR Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15-0 | SPICNT | R | 0h | SPI Count Value. Application loads count value when SPI_CNTR_EN is disabled. Counter is enabled when SPI_CNTR_EN is set.<br>0h = Smallest Value<br>FFFFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

Chapter 32

# CAN-FD

The Modular Controller Area Network (MCAN) peripheral supports both communication through classic CAN and CAN-FD protocols.


**32.1 MCAN Overview** ..... 1749
**32.2 MCAN Environment** ..... 1750
**32.3 CAN Network Basics** ..... 1751
**32.4 MCAN Functional Description** ..... 1753
**32.5 MCAN Integration** ..... 1802
**32.6 Interrupt and Event Support** ..... 1803
**32.7 MCAN Registers** ..... 1805





Copyright © 2026 Texas Instruments Incorporated

## 32.1 MCAN Overview

The Controller Area Network (CAN) is a serial communications protocol that efficiently supports distributed real-time control with a high level of reliability. CAN has high immunity to electrical interference and the ability to detect various type of errors. In CAN, many short messages are broadcast to the entire network, which provides data consistency in every node of the system.

The MCAN module supports both classic CAN and CAN FD (CAN with flexible data-rate) protocols. The CAN FD feature allows higher throughput and increased payload per data frame. Classic CAN and CAN FD devices can coexist on the same network without any conflict provided that partial network transceivers, which can detect and ignore CAN FD without generating bus errors, are used by the classic CAN devices. The MCAN module is compliant to ISO 11898-1:2015.

**Note**

The CAN FD feature is not available on all devices. Refer to the device-specific data sheet for more information.

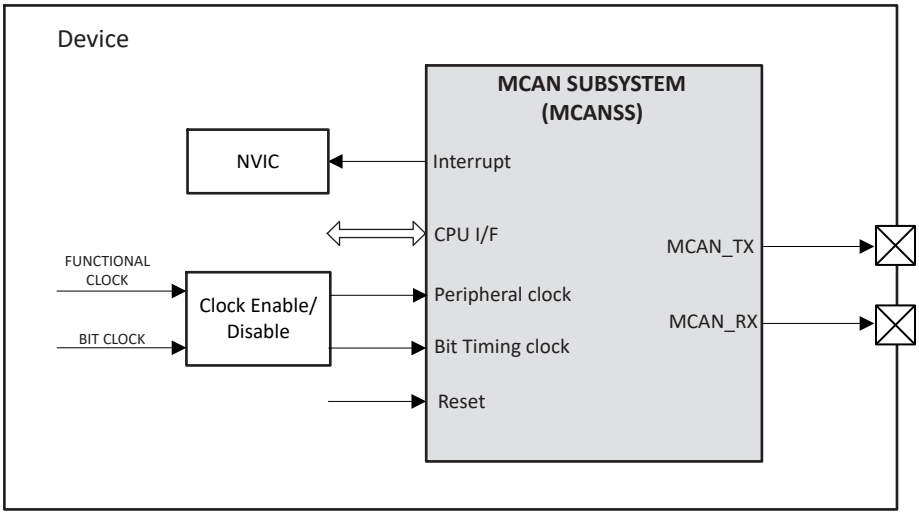



**Figure 32-1. MCAN Block Diagram**



Copyright © 2026 Texas Instruments Incorporated

### 32.1.1 MCAN Features

The MCAN module implements the following features:

- Conforms with CAN Protocol 2.0 A, B and ISO 11898-1:2015
- Full CAN FD support (up to 64 data bytes)
- AUTOSAR and SAE J1939 support
- Up to 32 dedicated transmit buffers
- Configurable transmit FIFO, up to 32 elements
- Configurable transmit queue, up to 32 elements
- Configurable transmit Event FIFO, up to 32 elements
- Up to 64 dedicated receive buffers
- Two configurable receive FIFOs, up to 14 elements each with 1kB message RAM.
- Up to 128 filter elements
- Loop-back mode for self-test
- Maskable interrupt
  - 2 configurable interrupt lines
  - Correctable ECC
  - Counter overflow
  - Clock stop or wakeup
- Non-maskable interrupt (uncorrectable ECC)
- Two clock domains (CAN clock and host clock)
- ECC check for Message RAM
- Clock stop and wakeup support
- Timestamp counter
- Up to 1-Mbps nominal bit rate, up to 8-Mbps data bit rate

Non-supported features:

- Host bus firewall
- Clock calibration
- Debug over CAN

## 32.2 MCAN Environment

The CAN network physical layer consists of a two-wire differential bus, usually twisted pair, and provides a high level of interference immunity. An external CAN transceiver IC is needed to access the bus.

Figure 32-2 shows typical MCAN wiring. Table 32-1 describes the external signals of the MCAN module.



Copyright © 2026 Texas Instruments Incorporated

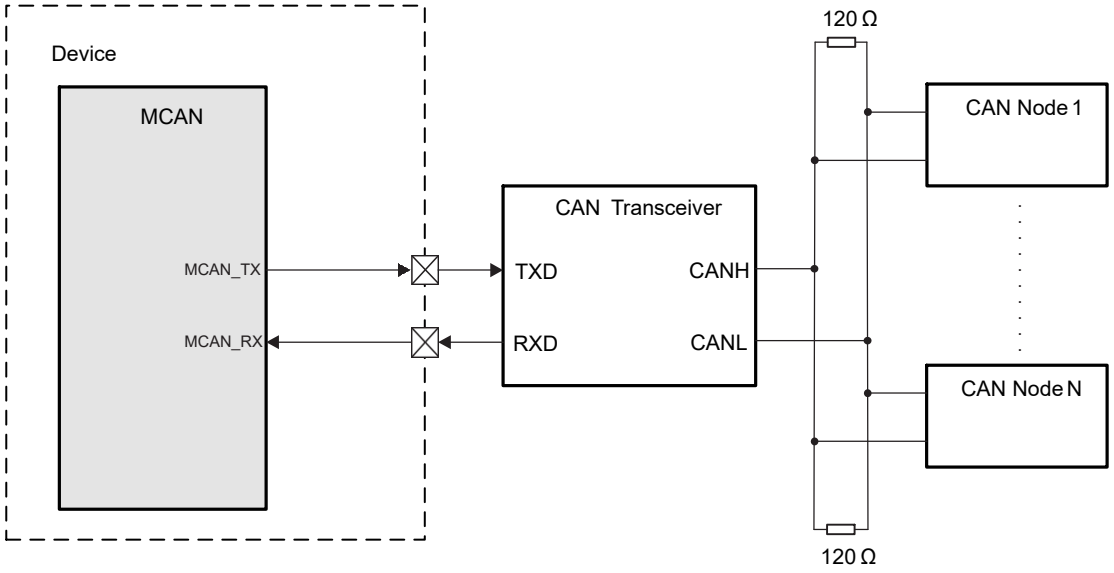



**Figure 32-2. MCAN Typical Bus Wiring**

**Table 32-1. MCAN I/O Description**

| Module Signal | I/O | Description | Value at Reset |
|---|---|---|---|
| MCAN_RX | Input | Serial data input from external CAN transceiver. | HiZ |
| MCAN_TX | Output | Serial data output to external CAN transceiver. | HiZ |

**Note**

See the *Terminal Configurations and Functions* section in the device data sheet and the *General-Purpose Input/Output (GPIO)* chapter to configure this peripheral to be connected to the device pins.

## 32.3 CAN Network Basics

The network basics are:

- The CAN bus is a 2-wire differential bus using non-return-to-zero (NRZ) encoding and has two states:
  - Recessive state (logical 1)
  - Dominant state (logical 0)
- When the bus is idle, any node can initiate a transmission to any other node. When two or more nodes (ECUs) attempt to transmit at the same time, a nondestructive arbitration technique sends messages in order of priority so that no messages are lost.
- The message transmission is multicast. Data messages transmitted are identifier-based, not address-based.
- The content of the message is labeled by the identifier that is unique throughout the network (for example: RPM, temperature, position, pressure, and so forth).
- All nodes on the network receive the message and each performs an acceptance test on the identifier. If the message is relevant, it is processed; otherwise, it is ignored.
- The unique identifier also determines the priority of the message (the lower the numerical value of the identifier, the higher the priority).
- Data is transmitted and received using message frames, consisting of the following basic fields:
  - Arbitration field
  - Control field
  - Data field (up to 8 bytes for classical CAN and up to 64 bytes for CAN FD)



Copyright © 2026 Texas Instruments Incorporated

- CRC field
- ACK field

For more information, see *ISO 11898-1:2015: CAN data link layer and physical signaling*.

MCAN power up and clock sequence :

1. Select CANCLK from the available clock sources: the HFCLK (HFXT or HFCLK_IN) or the SYSPLL (SYSPLLCLK1).
2. Enable the corresponding clock source.
3. Wait until the clock source is stable (SYSCTL_CLKSTATUS[*GOOD] bit = 1).
4. Enable MCAN power by setting PWREN for MCAN.
5. Wait until MCAN is ready (SYSCTL_SYSSTATUS[MCAN0READY] = 1).



Copyright © 2026 Texas Instruments Incorporated

## 32.4 MCAN Functional Description

The MCAN module performs CAN protocol communication according to ISO 11898-1:2015. The bit rate can be programmed to values up to 8Mbit/s. Additional transceiver hardware is required for the connection to the physical layer (CAN bus).

For communication on a CAN network, individual message frames can be configured. The message frames and identifier masks are stored in the Message RAM.

All functions concerning the handling of messages are implemented in the Message Handler.

The register set of the MCAN module can be accessed directly through the module interface. These registers are used to control and configure the CAN core and the Message Handler, and to access the Message RAM.

Figure 32-3 shows the MCAN module block diagram, followed by the description of the MCAN module blocks.

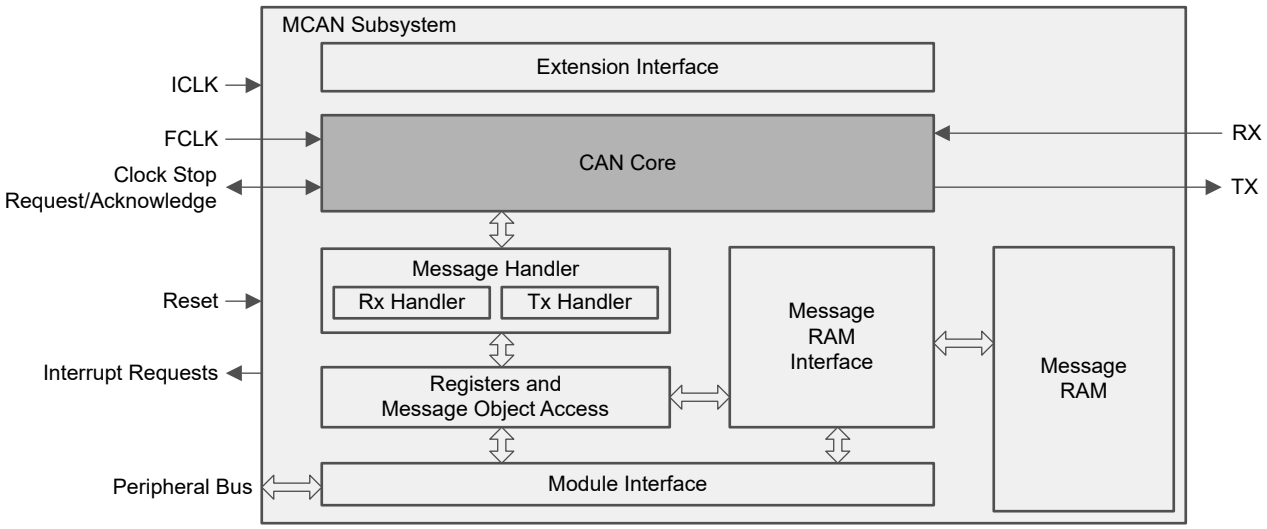



**Figure 32-3. MCAN Block Diagram**

- **CAN Core**: The CAN core consists of the CAN protocol controller and the Rx/Tx shift register. It handles all ISO 11898-1:2015 protocol functions and supports 11-bit and 29-bit identifiers.
- **Message Handler**: the Message Handler (Rx Handler and Tx Handler) is a state machine that controls the data transfer between the single-ported Message RAM and the CAN core's Rx/Tx shift register. It also handles the acceptance filtering and Interrupt generation as programmed in the control registers.
- **Message RAM**: the main purpose of the Message RAM is to store Rx/Tx messages, Tx Event elements, and Message ID Filter elements (for more information, see Section 32.4.19).
- **Message RAM Interface**: enables a connection between the Message RAM and the other blocks in the MCAN module.
- **Registers and Message Object Access**: Data consistency is ensured by indirect accesses to the message objects. During normal operation, all software and DMA accesses to the Message RAM are done through interface registers. The interface registers have the same word-length as the Message RAM.
- **Module Interface**: The MCAN module registers are accessed by the user's software through a 32-bit peripheral bus interface.
- **Clocking**: Two clocks are provided to the MCAN module: the peripheral synchronous clock (interface clock - MCAN_ICLK) and the peripheral asynchronous clock (functional clock - MCAN_FCLK).
- **Extension Interface**: All selected internal status and control signals are routed to this interface (except for the indication signals of configuration change enable bit (MCAN_CCCR.CCE) and Interrupt Register bits (MCAN_IR).



Copyright © 2026 Texas Instruments Incorporated

### 32.4.1 Clock Setup

Peripheral asynchronous clock (MCAN_FCLK) for MCAN can be clocked either through HFXT or SYSPLL. This clock configuration has to be done through SYSCTL REGISTERS. *Refer to CANCLK (CAN-FD Functional Clock ) under the Clocks section.*



Copyright © 2026 Texas Instruments Incorporated

### 32.4.2 Module Clocking Requirements

Two clocks are provided to the MCAN module:

- Host Clock : peripheral synchronous clock (MCAN_ICLK) as the general module clock source.
- CAN Clock: peripheral asynchronous clock (MCAN_FCLK) provided to the CAN core for generating the CAN bit timing.

Within the MCAN module, there is a synchronization mechanism implemented to make sure there is safe data transfer between the two clock domains. There is synchronization between the signals from the Host clock domain to the CAN clock domain and conversely, and between the reset signal (MCAN_RST) to the Host clock domain and to the CAN clock domain.

**Note**

MCAN_ICLK must always be higher or equal to MCAN_FCLK to achieve a stable functionality of the MCAN module: $f_{ICLK} \geq f_{FCLK}$

The CAN-FD supports higher speeds of operation and as such has more stringent timing requirements than the classic CAN. For performance, TI recommends using the lowest N-divider value that maintains a working PLL REF_CLK for the system. Lower N-divider values increase the loop bandwidth of the PLL, which in turn improves timing margins for CAN-FD.

The peripheral asynchronous clock (MCAN_FCLK) can be clocked either through HFXT or SYSPLL. The configuration of this has to be done through the SYSCTL registers. Refer to the *Clocks* module for more information

### 32.4.3 Interrupt Requests

The MCAN module generates interrupt requests that are configured through the Host CPU. Placing the MCAN module in suspend mode prevents the interrupt requests from propagating to the Host CPU. The MCAN core has two interrupt lines and 30 internal interrupt sources. Each source can be configured to drive one of the two interrupt lines. The interrupts are 'level high' interrupts. The MCAN core provides two interrupt requests (MCANSS_INT0 and MCANSS_INT1).

For more information, see the following registers:

- Interrupt Register (MCAN_IR)
- Interrupt Enable (MCAN_IE)
- Interrupt Line Select (MCAN_ILS)
- Interrupt Line Enable (MCAN_ILE)

The MCAN module supports an External Timestamp Counter. The External Timestamp Counter produces an interrupt when it rolls over (see Section 32.4.12.1).

For more information, see the following registers:

- Interrupt Clear Shadow Register (MCANSS_ICS)
- Interrupt Raw Status Register (MCANSS_IRS)
- Interrupt Enable Clear Shadow Register (MCANSS_IECS)
- Interrupt Enable Register (MCANSS_IE)
- Interrupt Enable Status Register (MCANSS_IES)
- End Of Interrupt Register (MCANSS_EOI)
- External Timestamp Prescaler Register (MCANSS_EXT_TS_PRESCALER)
- External Timestamp Unserviced Interrupts Counter Register (MCANSS_EXT_TS_UNSERVICED_INTR_CNTR)

To clear IRQ_INT0, IRQ_INT1 and TS_WAKE interrupts, write to the EOI bitfield for the corresponding interrupt number that is described in the MCANSS_EOI register.



Copyright © 2026 Texas Instruments Incorporated

After servicing an interrupt (external timestamp, interrupt 0/1), write ‘1’ in the corresponding bit in MCANSS_EOI register to clear the interrupt. In case of an ECC interrupt, after clearing the ECC interrupt source, application software must also write a ‘1’ to the EOI registers (MCANERR_SEC_EOI.EOI_WR/ CANERR_DED_EOI.EOI_WR). For more information see Section 32.4.14.2

The IRQ sequence is as follows:

1. Enable one of the interrupt sources by setting corresponding bit in IMASK.
2. Set MCANSS_IE to enable the IRQ, set MCANSS_ILS for line0 or line1.
3. Wait for the interrupt source to be triggered.
4. The interrupt is triggered and the application jumps into IRQ service routine (ISR).
5. In the ISR, check if IRQ is triggered by the expected source by reading IIDX. Reading IIDX clears the IRQ source, so there is no need to write ICLR to clear the IRQ.

   **Due to the design of MCAN, step (6) is required in the ISR to clear IRQ. Check the interrupt source to determine if step 7 is needed:**
6. Write 1 to MCAN_IR to clear the interrupt.
7. If the source is IRQ_INT0, IRQ_INT1, or TS_WAKE, also clear MCANSS_EOI, otherwise the next IRQ will not be recognized.



Copyright © 2026 Texas Instruments Incorporated

### 32.4.4 Operating Modes

The operating modes are discussed in the following sections.

#### 32.4.4.1 Normal Operation

Once the MCAN module is initialized and the MCAN_CCCR.INIT bit is reset to zero, the MCAN module synchronizes itself to the CAN bus and is ready for communication. After passing the acceptance filtering, received messages including Message Identifier (ID) and Data Length Code (DLC) are stored into a dedicated Rx Buffer or into Rx FIFO 0/Rx FIFO 1.

For messages to be transmitted, dedicated Tx buffers, and a Tx FIFO or a Tx queue can be initialized or updated.

**Note**

Automated transmission upon reception of remote frames is not supported.

#### 32.4.4.2 CAN Classic

CAN supports transmission of data up to 8 bytes with standard (11 bit) or (29 bit) identifier.

#### 32.4.4.3 CAN FD Operation

The CAN FD standard allows extended frames to be sent, up to 64 data bytes in a single frame at a higher bit rate for the data phase of a frame, up to 8 Mbps. The CAN FD standard introduces the ability to switch from one bit rate to another. Extended Data Length (EDL), as shown in Figure 32-4 and described in Table 32-2, sets a data length of up to 8 or 64 data bytes. Bit Rate Switching (BRS) indicates whether two bit rates (the data phase is transmitted at a different bit rate compared to the arbitration phase) are enabled.

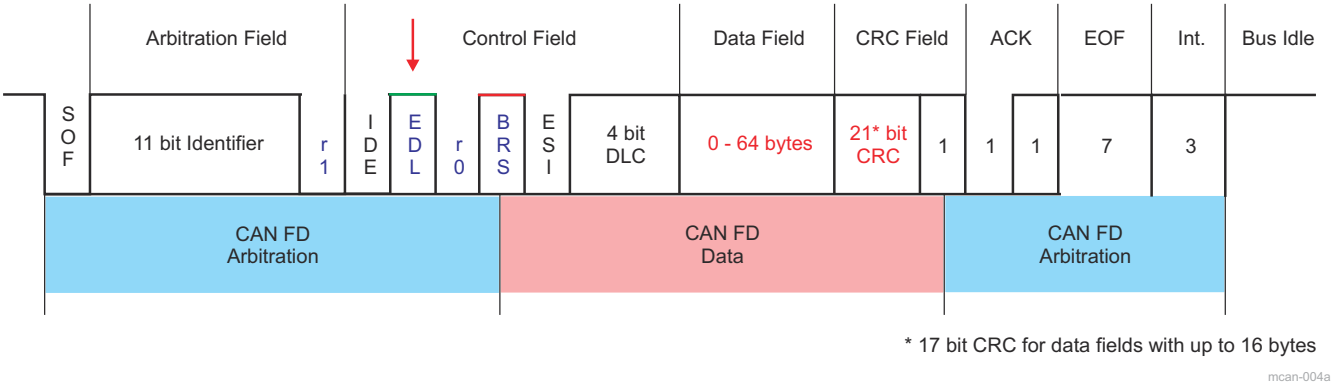



**Figure 32-4. CAN FD Frame**

**Table 32-2. CAN FD Frame Description**

| Bit | Description |
|---|---|
| SOF | Start of Frame |
| IDE | Identifier extension (for 29 bit extended ID) |
| FDF | Flexible Data Format |
| BRS | Bit Rate Switching |
| ESI | Error Status Indicator |
| DLC | Data Length Code |
| CRC | Cyclic Redundancy check |

There are two variants of CAN FD frame transmission:



Copyright © 2026 Texas Instruments Incorporated

- CAN FD frame transmission without bit rate switching
- CAN FD frame transmission where control field, data field, and CRC field are transmitted with a higher bit rate than the beginning and the end of the frame

In the CAN frames, FDF = recessive (logical 1) signifies a CAN FD frame, FDF = dominant (logical 0) signifies a Classic CAN frame. In a CAN FD frame, the two bits following FDF - res and BRS, decide whether the bit rate inside of this CAN FD frame is switched. A CAN FD bit rate switch is signified by res = dominant and BRS = recessive. Note that the coding of res = recessive is reserved for future expansion of the protocol. If the MCAN module receives a frame with FDF = recessive and res = recessive, the MCAN signals a Protocol Exception Event by setting the MCAN_PSR.PXE bit. When Protocol Exception Handling is enabled (MCAN_CCCR.PXHD = 0), this causes the operation state to change from Receiver (MCAN_PSR.ACT = 10) to Integrating (MCAN_PSR.ACT = 00) at the next sample point. In case Protocol Exception Handling is disabled (MCAN_CCCR.PXHD = 1), the MCAN treats a recessive bit as an error and responds with an error frame.

CAN FD operation is enabled by programming the MCAN_CCCR.FDOE bit. If MCAN_CCCR.FDOE = 1, transmission and reception of CAN FD frames is enabled. Transmission and reception of Classic CAN frames is always possible. Whether a CAN FD frame or a Classic CAN frame is transmitted can be configured using the FDF bit in the respective Tx Buffer element.

With MCAN_CCCR.FDOE = 0, received frames are interpreted as Classic CAN frames, which leads to the transmission of an error frame when receiving a CAN FD frame. When CAN FD operation is disabled, no CAN FD frames are transmitted even if the FDF bit of a Tx Buffer element is set. The MCAN_CCCR.FDOE and MCAN_CCCR.BRSE bits can only be changed while the MCAN_CCCR.INIT and MCAN_CCCR.CCE bits are both set. With MCAN_CCCR.FDOE = 0, the setting of bits FDF and BRS is ignored and frames are transmitted in Classic CAN format.

With MCAN_CCCR.FDOE = 1 and MCAN_CCCR.BRSE = 0, only FDF bit of a Tx Buffer element is evaluated. With MCAN_CCCR.FDOE = 1 and MCAN_CCCR.BRSE = 1, transmission of CAN FD frames with bit rate switching is enabled. All Tx Buffer elements with bits FDF and BRS set are transmitted in CAN FD format with bit rate switching.

A mode change during CAN operation is only recommended under the following conditions:

- The failure rate in the CAN FD data phase is significantly higher than in the CAN FD arbitration phase. In this case, disable the CAN FD bit rate switching option for transmissions.
- During system startup, all nodes are transmitting Classic CAN messages until verified that the nodes are able to communicate in CAN FD format. If this is true, all nodes switch to CAN FD operation.
- Wakeup messages in CAN Partial Networking have to be transmitted in Classic CAN format.
- End-of-line programming in case not all nodes are CAN FD capable. Non CAN FD nodes are held in Silent mode until programming has completed. Then all nodes switch back to Classic CAN communication.

The coding of the DLC in the CAN FD format differs from the standard CAN format. The DLC codes 0 to 8 have the same coding as in standard CAN (0 to 8 data bytes), the codes 9 to 15, which in standard CAN all code a data field of 8 bytes, are coded according to Table 32-3.

**Table 32-3. DLC Coding in CAN FD**

| DLC | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|
| Number of Data Bytes | 12 | 16 | 20 | 24 | 32 | 48 | 64 |

### CAN-FD Bit Timing

For CAN FD frames, the bit timing is switched inside the frame after the BRS (Bit Rate Switch) bit in case this bit is recessive. In the CAN FD arbitration phase, before the BRS bit, the nominal CAN bit timing (see Figure 32-5) is used as configured by the Nominal Bit Timing and Prescaler Register (MCAN_NBTP). In the following CAN FD data phase, the data phase bit timing is used as configured by the Data Bit Timing and Prescaler Register (MCAN_DBTP). The bit timing is switched back from the data phase timing at the CRC delimiter or when an error is detected, whichever occurs first.



Copyright © 2026 Texas Instruments Incorporated

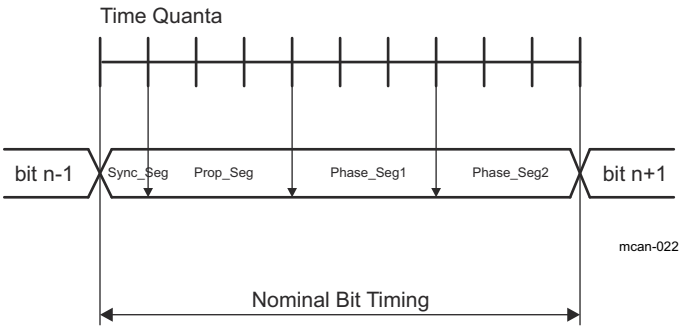



**Figure 32-5. CAN Bit Timing**

The maximum configurable data phase bit timing depends on the CAN clock frequency (MCAN_FCLK). Example: with MCAN_FCLK = 20MHz and the shortest configurable bit time of 4 $t_q$ (time quanta), the bit rate in the data phase is 5 Mbit/s.

In both data frame formats, CAN FD and CAN FD with bit rate switching, the value of the Error Status Indicator (ESI) bit depends on the transmitter error state (see MCAN_PSR.RESI bit) monitored at the start of the transmission. If the transmitter has an error passive flag, the ESI bit is transmitted recessive; else, the ESI bit is transmitted dominant.

### *32.4.5 Software Initialization*

A software initialization begins when the MCAN_CCCR.INIT bit is set to 1. This is done either by software or by a hardware reset, when an uncorrected bit error is detected in the Message RAM, or by going to a Bus_Off state. While the MCAN_CCCR.INIT bit is set, the message transfer is stopped and the status of the output TX pin is recessive (high). The counters of the Error Management Logic (EML) are unchanged. Setting the MCAN_CCCR.INIT bit does not change any configuration register. Resetting the MCAN_CCCR.INIT bit finishes the software initialization. After waiting for the occurrence of a sequence of 11 consecutive recessive bits (indication for Bus_Idle state) the message transfer starts.

Access to the MCAN configuration registers is only enabled when both MCAN_CCCR.INIT and MCAN_CCCR.CCE bits are set (write protection).

The MCAN_CCCR.CCE bit can only be set/reset while the MCAN_CCCR.INIT = 1. The MCAN_CCCR.CCE bit is automatically reset when the MCAN_CCCR.INIT bit is reset.

The following registers are reset when the MCAN_CCCR.CCE bit is set:

- MCAN_HPMS - High Priority Message Status
- MCAN_RXF0S - Rx FIFO 0 Status
- MCAN_RFX1S - Rx FIFO 1 Status
- MCAN_TXFQS - Tx FIFO/Queue Status
- MCAN_TXBRP - Tx Buffer Request Pending
- MCAN_TXBTO - Tx Buffer Transmission Occurred
- MCAN_TXBCF - Tx Buffer Cancellation Finished
- MCAN_TXEFS - Tx Event FIFO Status

The Timeout Counter value MCAN_TOCV.TOC field is preset to the value configured by the MCAN_TOCC.TOP field when the MCAN_CCCR.CCE bit is set.

In addition, the Tx Handler and Rx Handler are held in idle state while MCAN_CCCR.CCE = 1.

The following registers are only writable while MCAN_CCCR.CCE = 0

- MCAN_TXBAR - Tx Buffer Add Request
- MCAN_TXBCR - Tx Buffer Cancellation Request



Copyright © 2026 Texas Instruments Incorporated

MCAN_CCCR.TEST and MCAN_CCCR.MON bits can only be set by the Host CPU while MCAN_CCCR.INIT = 1 and MCAN_CCCR.CCE = 1. Both bits are reset at any time. The MCAN_CCCR.DAR bit can only be set/reset while MCAN_CCCR.INIT = 1 and MCAN_CCCR.CCE = 1.

Table 32-4 shows the steps to configure the MCAN module.

**Table 32-4. Steps to Configure MCAN Module**

| Step | Operation | Description | Pseudo Code |
|---|---|---|---|
| 1 | Initialize MCAN_CCCR | Set MCAN_CCCR.INIT bit<br>and check that the bit has been set | INIT = 1;<br>If INIT ≠ 1, wait until set |
| 2 | Unlock protected registers | Set MCAN_CCCR.CCE bit | CCE = 1; |
| 3 | Configure CAN mode | Set MCAN_CCCR.FDOE bit to CAN FD | FDOE = 1 for CAN FD<br>FDOE = 0 for Classic CAN |
| 4 | Configure Bit Rate Switching | Set MCAN_CCCR.BRSE bit | BRSE = 1 for bit rate switching<br>BRSE = 0 for no bit rate switching |
| 5 | Set nominal bit timing(1) | Set MCAN_NBTP register | |
| 6 | Lock protected registers | Clear MCAN_CCCR.CCE bit | CCE = 0; |
| 7 | Return MCAN module to normal operation | Clear MCAN_CCCR.INIT bit<br>and check that the bit has been cleared | INIT = 0;<br>If INIT ≠ 0, wait until cleared |

(1) See the MCAN_NBTP register on how to program CAN bit timing in the *MCAN_REGS Registers* section.

### *32.4.6 Transmitter Delay Compensation*

#### 32.4.6.1 Description

When only one CAN FD node is transmitting and all other nodes are receivers, the length of the bus line has no impact. When transmitting using the TX pin, the MCAN module receives the transmitted data from its CAN transceiver using the RX pin. The received data is delayed. If the transmitter delay is greater than TSEG1 (time segment before sample point), a bit error is detected.

The MCAN module provides a delay compensation mechanism to compensate for the transmitter delay. The compensation mechanism enables transmission with higher bit rates during the CAN FD data phase independent of the delay of a specific CAN transceiver. Without transmitter delay compensation the bit rate in the data phase is limited by the transmitter delay.

The mechanism enables configurations where the data bit time is shorter than the transmitter delay (it is described in detail in ISO 11898-1:2015). The transmitter delay compensation is enabled by setting the MCAN_DBTP.TDC bit to 1.

The delayed transmit data is compared against the received data at the Secondary Sample Point (SSP) to check for bit errors during the data phase of transmitting nodes. If a bit error is detected, the transmitter reacts on this bit error at the next following regular sample point. During the arbitration phase, the delay compensation is always disabled.

The received bit is compared against the transmitted bit at the SSP. The SSP position is defined as the sum of the measured delay from the MCAN transmit output TX pin through the transceiver to the receive input RX pin plus the transmitter delay compensation offset configured by the MCAN_TDCR.TDCO field (see Figure 32-6). The transmitter delay compensation offset is used to adjust the position of the SSP inside the received bit (example: half of the bit time in the data phase). The position of the SSP is rounded down to the next integer number of mtq.

The actual transmitter delay compensation value can be checked by reading the MCAN_PSR.TDCV field. This field is cleared when the MCAN_CCCR_INIT bit is set and is updated at each transmission of CAN FD frame while the MCAN_DBTP.TDC bit is set.



Copyright © 2026 Texas Instruments Incorporated

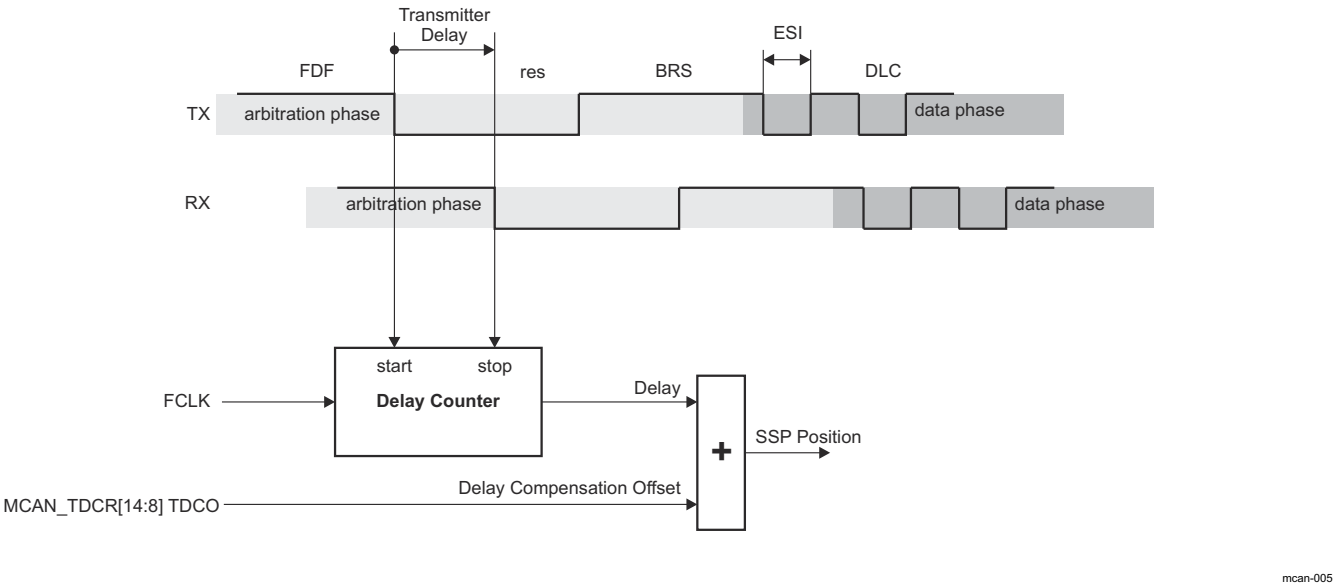



**Figure 32-6. Transmitter Delay Measurement**



Copyright © 2026 Texas Instruments Incorporated

#### 32.4.6.2 Transmitter Delay Compensation Measurement

When transmitter delay compensation is enabled (by programming MCAN_DBTP.TDC = 1), the measurement is started within each transmitted CAN FD frame at the falling edge of FDF bit to bit r0. The measurement is stopped when this edge is seen at the receive input RX pin of the transmitter. The resolution of this measurement is one mtq (see Figure 32-6). The mtq (minimum time quantum) dimension is equal to the CAN clock period (MCAN_FCLK).

The use of a transmitter delay compensation filter window can be enabled by programming the MCAN_TDCR.TDCF field. This filter feature defines a minimum value for the SSP position to avoid the case in which a dominant glitch inside the received FDF bit ends the delay compensation measurement before the falling edge of the received res bit, resulting in an early taken SSP position. Dominant edges on the RX pin that result in an earlier SSP position are ignored for transmitter delay measurement. The measurement is stopped when the SSP position is at least MCAN_TDCR.TDCF field and the RX pin is low.

The following boundary conditions have to be considered:

- The sum of the measured delay from the TX pin to the RX pin and the configured transmitter delay compensation offset (MCAN_TDCR.TDCO field) has to be less than six bit times in the data phase.
- The sum of the measured delay from the TX pin to the RX pin and the configured transmitter delay compensation offset (MCAN_TDCR.TDCO) field has to be less or equal 127 mtq. In case this sum exceeds 127 mtq, the maximum value of 127 mtq is used for transmitter delay compensation.
- The data phase ends at the sample point of the CRC delimiter, that stops checking of receive bits at the SSPs.

### *32.4.7 Restricted Operation Mode*

In restricted operation mode, the CAN node is able to receive data and remote frames and give acknowledgment to valid frames, but the node does not send data frames, remote frames, active error frames, or overload frames. In the case of an error condition or overload condition, the node does not send dominant bits; instead the node waits for the occurrence of bus idle condition to resynchronize itself to the CAN communication. The receive and transmit error counters (MCAN_ECR.REC and MCAN_ECR.TEC) are frozen while CAN error logging (MCAN_ECR.CEL) is active. The Host CPU can set the MCAN module into Restricted Operation Mode by setting the MCAN_CCCR.ASM bit. The bit can only be set by the Host CPU at any time when both MCAN_CCCR.CCE and MCAN_CCCR.INIT bits are set to 1.

The restricted operation mode is automatically entered when the Tx Handler is not able to read data from the Message RAM in time. To leave restricted operation mode, the Host CPU has to reset the MCAN_CCCR.ASM bit. This mode can be used in applications that adapt themselves to different CAN bit rates. In this case, the application tests different bit rates and leaves the restricted operation mode after the node has received a valid frame.

**Note**

The Restricted Operation Mode must not be combined with the Loop Back Mode.



Copyright © 2026 Texas Instruments Incorporated

### 32.4.8 Bus Monitoring Mode

Entering bus monitoring mode is done by setting the MCAN_CCCR.MON bit to 1. In this mode (see ISO 11898-1:2015, *Bus Monitoring* section), the MCAN module is able to receive valid data and remote frames, but cannot start a transmission. The MCAN module sends only recessive bits on the CAN bus. If the MCAN module is required to send a dominant bit (ACK bit, overload flag, active error flag), the bit is rerouted internally so that the MCAN module monitors this dominant bit, although the CAN bus may remain in recessive state. In bus monitoring mode, the MCAN_TXBRP register is held in reset state. The bus monitoring mode can be used to analyze the traffic on a CAN bus without affecting the bus by the transmission of dominant bits. Figure 32-7 shows the connection of the TX and RX signals to the MCAN module in bus monitoring mode.

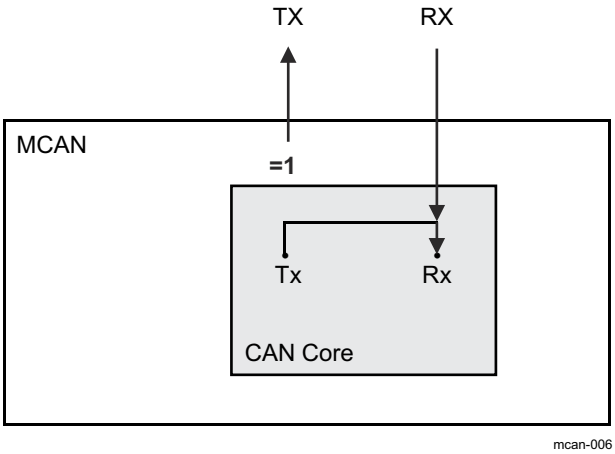



**Figure 32-7. Connection of Signals in Bus Monitoring Mode**



Copyright © 2026 Texas Instruments Incorporated

### 32.4.9 Disabled Automatic Retransmission (DAR) Mode

According to the CAN Specification (see ISO11898-1:2015, *Recovery Management* section), the MCAN module provides means for automatic retransmission of frames that have lost arbitration or that have been disturbed by errors during transmission. By default automatic retransmission is enabled (see the MCAN_CCCR.DAR bit).

#### 32.4.9.1 Frame Transmission in DAR Mode

In DAR mode all transmissions are automatically canceled after they have started on the CAN bus. A Tx buffer's Tx Request Pending (MCAN_TXBRP[xx]) TRPx bit is reset after successful transmission, when a transmission has not yet been started at the point of cancellation, has been aborted due to lost arbitration, or when an error occurred during frame transmission.

Successful transmission:

- Corresponding Tx Buffer Transmission Occurred (MCAN_TXBTO.TOx) TOx bit is set
- Corresponding Tx Buffer Cancellation Finished (MCAN_TXBCF.CFx) CFx bit is not set

Successful transmission in spite of cancellation:

- Corresponding Tx Buffer Transmission Occurred (MCAN_TXBTO.TOx) TOx bit is set
- Corresponding Tx Buffer Cancellation Finished (MCAN_TXBCF.CFx) CFx bit is set

Arbitration lost or frame transmission disturbed:

- Corresponding Tx Buffer Transmission Occurred (MCAN_TXBTO.TOx) TOx bit is not set
- Corresponding Tx Buffer Cancellation Finished (MCAN_TXBCF.CFx) CFx bit is set

In the case of a successful frame transmission, and if storage of Tx events is enabled, a Tx Event FIFO element is written with Event Type ET = 10 (transmission in spite of cancellation).

### 32.4.10 Clock Stop Mode

Entering the clock stop mode can be achieved by programming one of two bits:

- Clock stop request bit in MCAN_IP (MCAN_CCCR.CSR)
- Stop request bit in MCAN_WRAPPER (MCANSS_CLKCTL.STOPREQ)

The register bit within the MCAN_IP (MCAN_CCCR.CSR) reads as 1 as long as the MCAN_WRAPPER bit (MCANSS_CLKCTL.STOPREQ) is asserted.

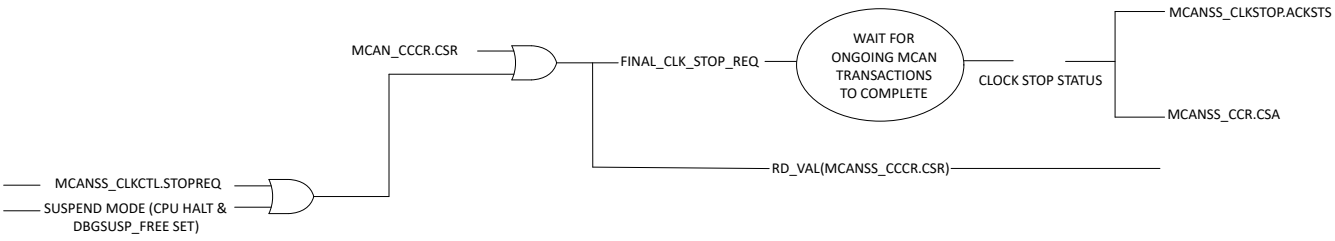



**Figure 32-8. Clock Stop Request**

After all pending transmission requests have completed, the MCAN module waits until the bus idle state is detected and then sets MCAN_CCCR.INIT to 1 to prevent further CAN transfers. The MCAN module then acknowledges the MCAN is ready for power down by setting the clock stop acknowledge bit MCAN_CCCR.CSA to 1 (MCANSS_CLKSTS.CLKSTOP_ACKSTS reflects the same value as well). In this state, before clocks are switched off, further register accesses can be made. However, a write access to the MCAN_CCCR.INIT bit has no effect. Module clock inputs MCAN_ICLK and MCAN_FCLK can now be switched off using MCANSS_CLKEN.CLK_REQEN.

To exit the power-down mode, the application has to turn on module clocks before clearing the clock stop request bit (make sure both MCAN_CCCR.CSR and MCANSS_CLKCTL.STOPREQ are cleared). MCAN acknowledges removal of the clock request by clearing the MCANSS_CLKSTS.CLKSTOP_ACKSTS



Copyright © 2026 Texas Instruments Incorporated

and MCAN_CCCR.CSA bits. The application can now restart CAN communication by clearing the MCAN_CCCR.INIT bit.

Automatic wakeup from the power-down mode (due to activity on MCAN Rx) is supported through the MCANSS_CTRL.AUTOWAKEUP and MCANSS_CTRL.WAKEUPREQEN bits (for more information, see Section 32.4.10.2).

Clock stop and automatic wakeup is illustrated in Figure 32-9.



Copyright © 2026 Texas Instruments Incorporated

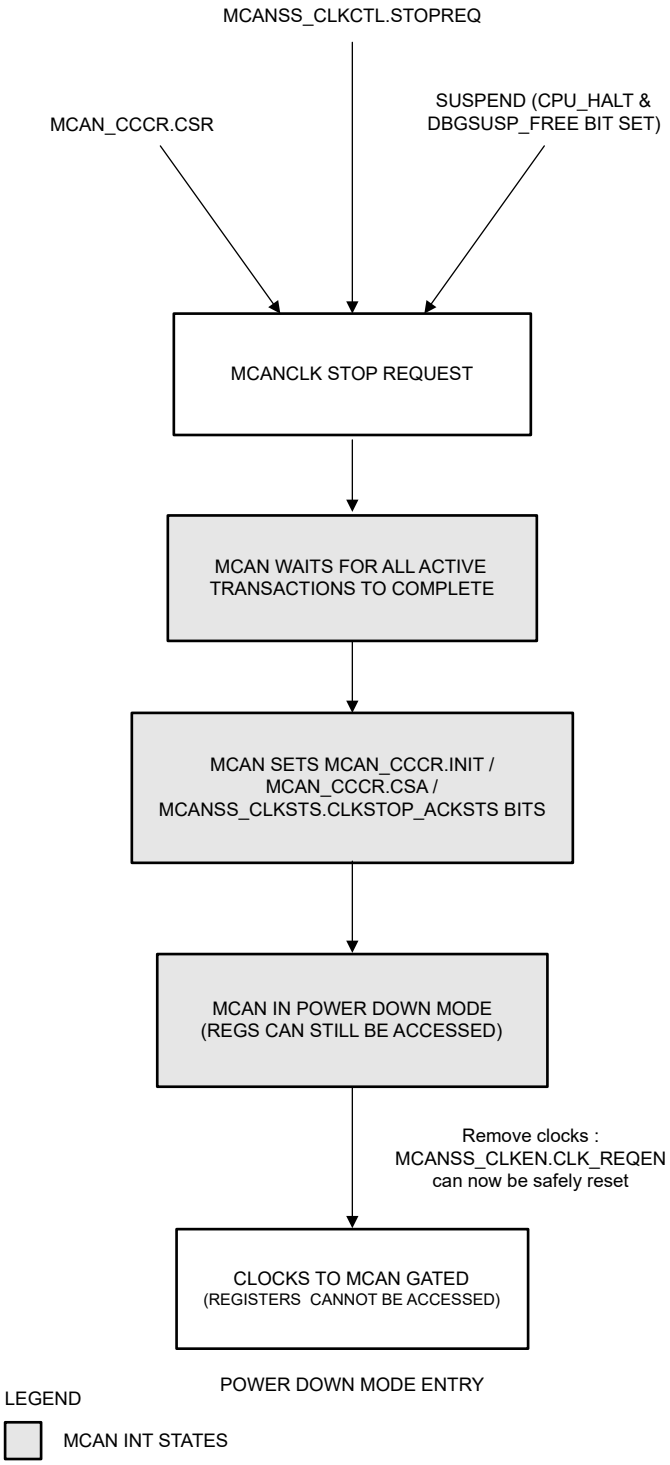



**Figure 32-9. Power Down Entry**

#### 32.4.10.1 Suspend Mode

The MCAN module supports two suspend modes:

- Immediate
- Graceful



Copyright © 2026 Texas Instruments Incorporated

In a graceful suspend mode (see the MCANSS_CTRL.DBGSUSP_FREE bit) when the suspend request is asserted, a clock stop request to the MCAN core is performed. The MCAN core responds with a clock stop acknowledge when all pending Tx messages have been processed and an Idle line had been detected. At that point, the MCAN_CCCR.INIT bit is set and the MCAN core stays Idle. The suspend state can be verified by reading the MCAN_CCCR.INIT bit.

The automatic wakeup feature is enabled by setting the MCANSS_CTRL.AUTOWAKEUP and MCANSS_CTRL.WAKEUPREQEN bits to 1 (for more information, see Section 32.4.10.2). When suspend request is removed, if no external clock stop request is active, a read-modify-write to the MCAN_CCCR.INIT bit is performed to clear the bit.

During suspend mode the auto-clear feature is disabled. The following register fields have an auto-clear feature:

- MCAN_ECR.CEL
- MCAN_PSR.LEC
- MCAN_PSR.DLEC
- MCAN_PSR.RESI
- MCAN_PSR.RBRS
- MCAN_PSR.RFDF
- MCAN_PSR.PXE

#### 32.4.10.2 Wakeup Request

Issuing a clock stop request puts the MCAN module into power-down mode (Sleep Mode). During transition from IDLE to ACTIVE, if the MCANSS_CTRL.AUTOWAKEUP and MCANSS_CTRL.WAKEUPREQEN bits are enabled, after the MCAN Core respond to the removal of the clock stop request with removing the clock stop acknowledge, a read-modify-write is issued to clear the MCAN_CCCR.INIT bit and the MCAN core resumes operation. Please note that after clock stop request has been removed by the hardware the first frame (wakeup frame) is not received. This is because after the clock stop is issued; there are no active clocks running into the IP. Therefore after removing the clock stop request; the wakeup frame that enables clock has to be re-transmitted.

If the MCANSS_CTRL.WAKEUPREQEN bit is set, the MCAN module provides a wakeup request on the following wakeup event:

- The receive RX pin is dominant (logical 0)

The wakeup request is deasserted when any of the following conditions occur:

- Clock stop request is removed and clock stop acknowledge is deasserted
- A reset is applied to the MCAN module



Copyright © 2026 Texas Instruments Incorporated

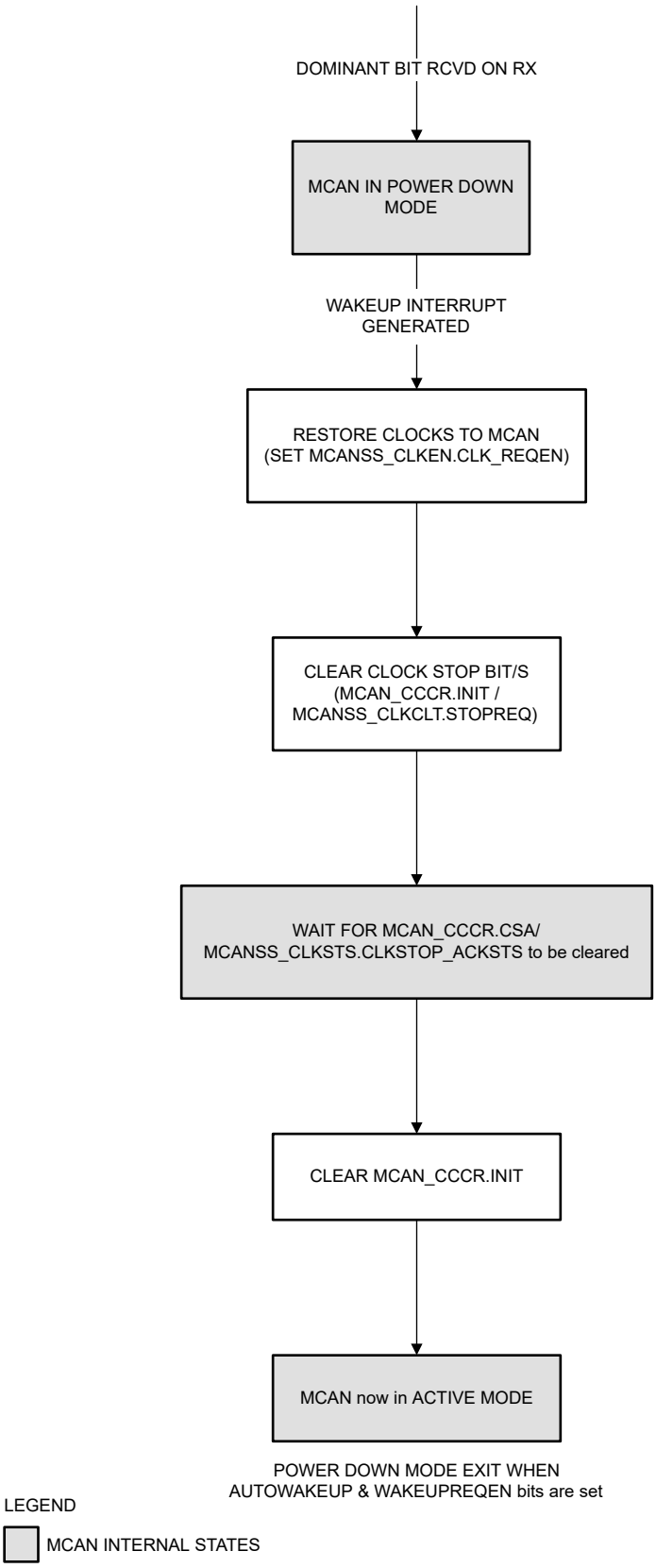



**Figure 32-10. Auto Wakeup Enabled Exit from Power Down**



Copyright © 2026 Texas Instruments Incorporated

Please note the suggested user notes while issuing a wakeup request.

- If there is a message that is being received on the CAN Bus when a Clock Stop Request (CSR) is issued; it will create a wakeup interrupt at the end of the reception resulting in the MCAN module not entering the clock stop mode. To avoid unnecessary wake ups; the user should check the node activity bit (ACT) in the protocol status register(PSR) ensuring an idle state before setting the request.
- The user should also avoid creating a SW polling loop to only exit with a stop clock acknowledge (CSA) since the timing of the software check and wake up interrupt clearing the CSA could cause an infinite software loop
- If there is excessive noise on the Rx pin, it may cause frequent wake up. To avoid this it is recommended that the user waits for a specified time to see if a wake up was received and if not, assert the clock stop state.
- The wakeup logic can be unexpectedly disabled if MCAN is reset while clock stop request is asserted. Therefore, the user should avoid asserting a soft reset to MCAN while CSR is asserted.



Copyright © 2026 Texas Instruments Incorporated

### 32.4.11 Test Modes

The MCAN_TEST register write access is enabled by setting the test mode enable MCAN_CCCR.TEST bit to 1. The MCAN_TEST register allows the configuration of the test modes and test functions.

The transmit (TX) pin has four different output functions which can be selected by programming the MCAN_TEST.TX field. The default function is the serial data output. The pin can also be driven with a constant dominant or recessive value. It is also possible to drive the sample-point signal to monitor the bit-timing.

The actual value of the receive (RX) pin can be monitored from MCAN_TEST.RX bit. Both functions can be used to check the physical layer. Due to the synchronization mechanism between the CAN clock (MCANx_FCLK) and Host clock (MCANx_ICLK) domain, there can be a delay of several Host clock periods between writing to the MCAN_TEST.TX field until the new configuration is visible at the output TX pin. This applies also when reading input RX pin by way of the MCAN_TEST.RX bit.

**Note**

Test modes can be used for self-test only. The software control for TX pin interferes with all CAN protocol functions. It is not recommended to use test modes for an application.

#### 32.4.11.1 External Loop Back Mode

The MCAN module can be set into external loop back mode by programming MCAN_TEST.LBCK to 1. In loop backmode, the MCAN treats its own transmitted messages as received messages and stores them (if they pass acceptance filtering) into an Rx Buffer or an Rx FIFO as shown in below figure. Figure shows the connection of the TX and RX pins to the MCAN module in external loop back mode.

This mode is provided for hardware self-test. To be independent from external stimulation, the MCAN module ignores acknowledge errors (recessive bit sampled in the acknowledge slot of a data/remote frame) in loop back mode. In this mode, the MCAN module performs an internal feedback from the Tx output to the Rx input. The actual value of the RX input pin is disregarded by the MCAN module. The transmitted messages are monitored at the TX pin.

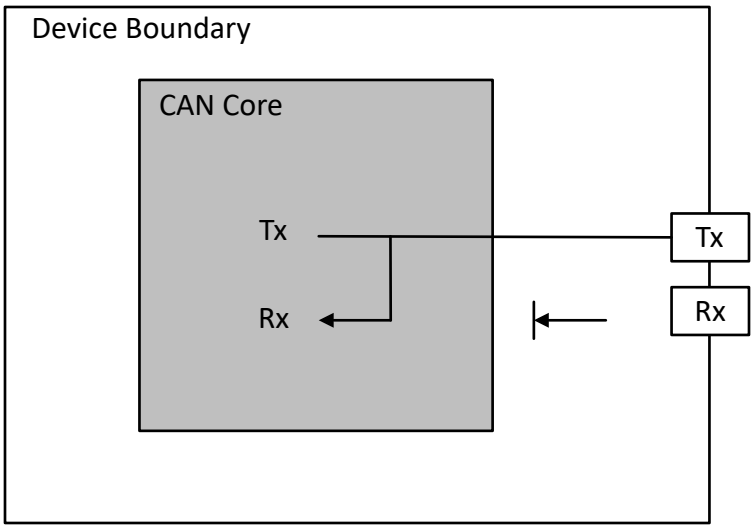



**Figure 32-11. External Loop Back Mode**

#### 32.4.11.2 Internal Loop Back Mode

The MCAN module can be set into internal loop back mode by programming MCAN_TEST.LBCK and MCAN_CCCR.MON bits to 1. The internal loop back mode is used for a Hot Self-test. The Hot Self-test allows



Copyright © 2026 Texas Instruments Incorporated

the MCAN module to be tested without affecting a running CAN system connected to the TX and RX pins. In this mode, the RX pin is disconnected from the MCAN module and the TX pin is held recessive. Figure 32-12 shows the connection of the TX and RX pins to the MCAN module in internal loop back mode.

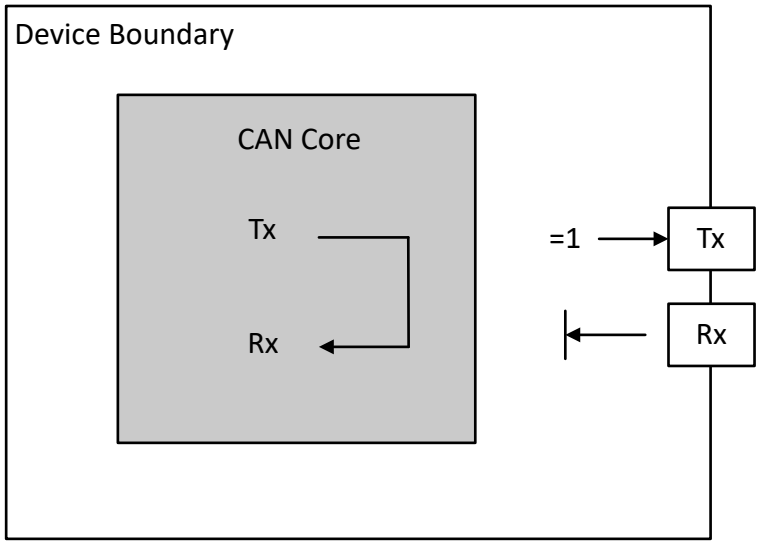



**Figure 32-12. Internal Loop Back Mode**

### *32.4.12 Timestamp Generation*

The MCAN module has an integrated 16-bit wrap-around counter for timestamp generation. The timestamp counter prescaler MCAN_TSCC.TCP field can be configured to clock the counter in multiples of CAN bit times (1-16). The counter is readable by way of the MCAN_TSCV.TSC field. A write access to the MCAN_TSCV register resets the counter to zero. When the timestamp counter wraps around the interrupt MCAN_IR.TSW flag is set. On start of a frame reception/transmission the counter value is captured and stored into the timestamp section of an Rx Buffer/Rx FIFO (RXTS[15:0]) or Tx Event FIFO (TXTS[15:0]) element. For more information regarding FIFO timestamp configuration, see Section 32.4.19.

#### 32.4.12.1 External Timestamp Counter

For CAN FD operation mode, the MCAN core requires an external timestamp counter. An externally generated 16-bit vector can substitute the integrated 16-bit CAN bit time counter (internal timestamp counter) for receive and transmit timestamp generation. An external 16-bit timestamp counter can be used by programming the MCAN_TSCC.TSS field.

The external timestamp counter uses the interface clock (MCANx_ICLK) as a reference clock. The MCAN core accepts a 16-bit timestamp. A 24-bit prescaler provides a programmable resolution for the timestamp (see MCANSS_EXT_TS_PRESCALER.PRESCALER bit field). The external timestamp counter can be enabled or disabled through the MCANSS_CTRL.EXT_TS_CNTR_EN bit. When disabled, the counter is reset back to zero. While enabled, the counter keeps incrementing. When the timestamp rolls over, the MCAN_IRQ_TS interrupt is generated.

When the timestamp rolls over, the MCANSS_IRS register is set. The MCANSS_IE register can be affected by writing to the MCANSS_IESS register to set or to the MCANSS_IECS register to clear. The MCANSS_IESS register is a shadow register mapped to the same address as the MCANSS_IE register. The level interrupt is a reflection of both MCANSS_IRS and MCANSS_IE being set. The MCANSS_IES register reflects the level interrupt. When a rollover event occurs, the interrupt counter is incremented. Writing to the MCANSS_ICS register to clear the MCANSS_IRS register also decrements the interrupt counter. Writing to the MCANSS_EOI register issues another pulse, if the interrupt counter is not zero.



Copyright © 2026 Texas Instruments Incorporated

The rollover event can be artificially simulated by software through writing to the Interrupt Set Shadow register (MCANSS_ISS). The MCANSS_ISS register is a shadow register mapped to the same address as the MCANSS_IRS register.

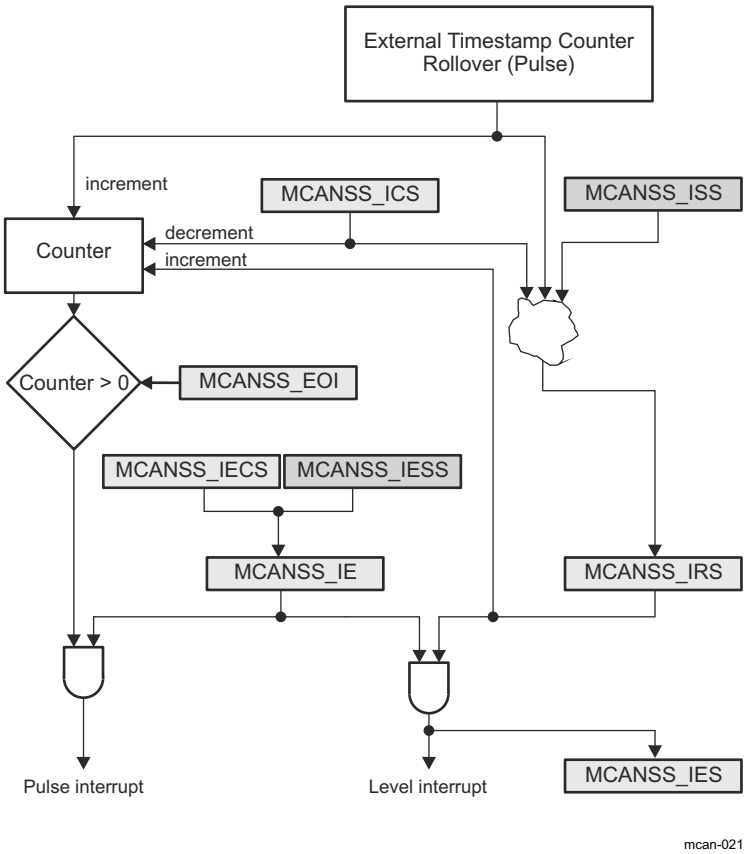



**Figure 32-13. External Timestamp Counter Interrupt**

### *32.4.13 Timeout Counter*

The MCAN module has an integrated 16-bit timeout counter. The timeout counter is used to signal timeout conditions for the Rx FIFO 0, Rx FIFO 1, and Tx Event FIFO Message RAM elements. The timeout counter is configured using the MCAN_TOCC register and is enabled using the MCAN_TOCC.ETOC bit. The timeout counter operates as down-counter and uses the same prescaler programmed by the MCAN_TSCC.TCP field as the timestamp counter. The actual counter value can be monitored from the MCAN_TOCV.TOC field. The timeout counter can be started only when MCAN_CCCR.INIT = 0 and stopped when MCAN_CCCR.INIT = 1 (example: when the MCAN enters Bus_Off state). The operation mode is selected by the MCAN_TOCC.TOS field. When continuous mode is selected, the counter starts when MCAN_CCCR.INIT = 0, a write to the MCAN_TOCV register presets the counter to the value configured by the MCAN_TOCC.TOP field and continues down-counting.

In case the timeout counter is controlled by one of the FIFOs, an empty FIFO presets the counter to the value configured by the MCAN_TOCC.TOP field. Down-counting is started when the first FIFO element is stored. Writing to the MCAN_TOCV register has no effect. When the counter reaches zero, the interrupt MCAN_IR.TOO flag is set.

In continuous mode, the counter is immediately restarted at the value configured by the MCAN_TOCC.TOP field.



Copyright © 2026 Texas Instruments Incorporated

**Note**

The clock signal for the timeout counter is derived from the CAN core sample point signal. Therefore, the point in time where the timeout counter is decremented can vary due to the synchronization/re-synchronization mechanism of the CAN core. If the bit rate switch (BRS) feature in CAN-FD is used, the timeout counter is clocked differently in arbitration and data field.

### *32.4.14 Safety*

The Message RAM (MSG_RAM) is wrapped in an ECC wrapper providing SECDED parity functionality. The MCAN ECC wrapper is controlled by an ECC aggregator.

#### 32.4.14.1 MCAN ECC Wrapper

The ECC wrapper provides single error correction (SEC) and double error detection (DED) parity to the message memory content. The ECC wrapper has side band signals for error notification. The ECC Wrapper implements an error injection test mode.

The error correction is done using a lazy write back. When an error is detected, the error is noted in a FIFO queue that waits for an access gap to write the data back and refresh the memory. If a transaction writes new data to the compromised entry before the lazy write back completes, the write back is discarded.

#### 32.4.14.2 MCAN ECC Aggregator

This section describes the functional details of the MCAN ECC aggregator module.

##### *32.4.14.2.1 MCAN ECC Aggregator Overview*

The MCAN ECC aggregator module supports the following general features:

- Provides a mechanism to control and monitor the ECC RAM in the MCAN module.
- Provides software access to all the ECC related registers.
- Supports software readable status of ECC single/double-bit errors and associated info such as RAM address and data bits that are in error.
- Aggregates level pending status from the ECC RAM into a single interrupt to the Host CPU.

The following feature is not supported:

- Statistics such as tracking the number of single and double-bit errors. If needed, these operations can be handled by software.

##### *32.4.14.2.2 MCAN ECC Aggregator Registers*

There are three groups of registers in the MCAN ECC aggregator module:

- **Global registers:** Aggregator Revision Register (MCANERR_REV), ECC Vector Register (MCANERR_VECTOR), Misc Status Register (MCANERR_STAT), ECC Control Register (MCANERR_CTRL), and ECC Wrapper Revision Register (MCANERR_WRAP_REV).
- **Control and status registers:** ECC Error Control Registers (MCANERR_ERR_CTRL1 and MCANERR_ERR_CTRL2) and ECC Error Status Registers (MCANERR_ERR_STAT1, MCANERR_ERR_STAT2, and MCANERR_ERR_STAT3).
- **Interrupt registers:** interrupt status, interrupt enable set, interrupt enable clear, and EOI (End Of Interrupt) registers that are part of a standard interrupt module. For more information, see the following registers:
  - MCANERR_SEC_EOI
  - MCANERR_SEC_STATUS
  - MCANERR_SEC_ENABLE_SET
  - MCANERR_SEC_ENABLE_CLR
  - MCANERR_DED_EOI
  - MCANERR_DED_STATUS
  - MCANERR_DED_ENABLE_SET
  - MCANERR_DED_ENABLE_CLR



Copyright © 2026 Texas Instruments Incorporated

#### 32.4.14.3 Reads to ECC Control and Status Registers

The reads to the ECC control and status registers are triggered by writing a 'read message' to the ECC Vector Register as:

1. Software writes value (the ECC RAM ID) to the MCANERR_VECTOR.ECC_VECTOR field to select the ECC RAM for control or status.
2. Software writes 1 to the MCANERR_VECTOR.RD_SVBUS bit to trigger a read.
3. Software writes read address to the MCANERR_VECTOR.RD_SVBUS_ADDRESS field.
4. Software then polls the MCANERR_VECTOR.RD_SVBUS_DONE bit to check if the bit is 1. This bit indicates that the read operation has completed.
5. Software reads the data from the ECC control or status register. The following clock cycle (MCAN_ICLK) returns the read data.

#### 32.4.14.4 ECC Interrupts

The ECC aggregator module aggregates the level pending status from the ECC RAM into a single EOI-handshake based interrupt to the Host CPU. Software is expected to follow the sequence described:

1. Software enables the interrupts for the ECC RAM by writing to the MCANERR_SEC_ENABLE_SET/ MCANERR_DED_ENABLE_SET register.
2. Software writes the ECC RAM ID in the MCANERR_VECTOR.ECC_VECTOR.
3. Software writes the MCANERR_VECTOR.RD_SVBUS bit to trigger the read.
4. Software writes the MCANERR_ERR_STAT1 register address to the MCANERR_VECTOR.RD_SVBUS_ADDRESS field. Software needs to load the 'read message' in the rMCANERR_VECTOR register again, if the software needs to read the MCANERR_ERR_STAT2 register.
5. Software polls the MCANERR_VECTOR.RD_SVBUS_DONE bit. When this bit is set, a read of the MCANERR_ERR_STAT1/MCANERR_ERR_STAT2 register is performed.
6. After the interrupt has been serviced, software clears the interrupt status by writing to the MCANERR_ERR_STAT1.CLR_ECC_SEC or MCANERR_ERR_STAT1.CLR_ECC_DED bit depending on the type of the ECC error.
7. Software polls the MCANERR_ERR_STAT1 register to verify that the status bit has been cleared.
8. Software writes to the MCANERR_SEC_EOI/MCANERR_DED_EOI register to clear the interrupt.
9. After clearing the ECC interrupt source, the application software must also write 1 to the MCANERR_SEC_EOI.EOI_WR /MCANERR_DED_EOI.EOI_WR bits.



Copyright © 2026 Texas Instruments Incorporated

### 32.4.15 Tx Handling

The Tx handler is used to handle the Tx requests. It controls the transfer of transmit messages from the dedicated Tx buffers, the Tx FIFO, and the Tx Queue to the CAN Core, the Tx Event FIFO, and the Put and GetIndex operations. The MCAN module supports up to 32 Tx buffers. These Tx buffers can be configured as dedicated Tx buffers, Tx FIFO, or Tx Queue and as combination of dedicated Tx buffers/Tx FIFO or dedicated Tx buffers/Tx Queue. For each Tx Buffer element Classical CAN or CAN FD transmission mode can be configured. Table below shows the possible configurations for message transmission.

**Table 32-5. Message Transmission configurations**

| MCAN_CCCR Register | | Buffer Element | | Frame Transmission |
|---|---|---|---|---|
| BRSE | FDOE | FDF | BRS | |
| ignored | 0 | ignored | ignored | Classic CAN |
| 0 | 1 | 0 | ignored | Classic CAN |
| 0 | 1 | 1 | ignored | CAN FD without bit rate switching |
| 1 | 1 | 0 | ignored | Classic CAN |
| 1 | 1 | 1 | 0 | CAN FD without bit rate switching |
| 1 | 1 | 1 | 1 | CAN FD with bit rate switching |

When the Tx Buffer RequestPending (MCAN_TXBRP) register is updated, or when a transmission has been started the Tx Handler starts scanning to check for the highest priority pending Tx request. The Tx Buffer with the lowest Message ID has highest priority.

**Note**

AUTOSAR requires at least three Tx Queue buffers and support of transmit cancellation



Copyright © 2026 Texas Instruments Incorporated

#### 32.4.15.1 Transmit Pause

The transmit pause feature is intended for use in CAN networks where the CAN Message IDs are specific and cannot easily be changed. These Message IDs may have a higher priority than other defined Message IDs, while in a specific application their relative priority should be inverse. This allows for a case where one ECU sends a burst of CAN messages that cause CAN messages from another ECU to be delayed (paused).

The transmit pause feature is enabled by the MCAN_CCCR.TXP bit. By default this bit is disabled (MCAN_CCCR.TXP= 0). Each time after successfully transmitted message, a pause for two CAN bit times occurs before the start of the next transmission. This allows the other CAN nodes in the network to transmit messages even if their Message IDs have lower priority.



Copyright © 2026 Texas Instruments Incorporated

#### 32.4.15.2 Dedicated Tx Buffers

Dedicated Tx buffers are intended for message transmission under complete control of the Host CPU. There are two options.

- Each dedicated Tx Buffer is configured with a specific Message ID
- Two or more dedicated Tx buffers are configured with the same Message ID. In this case the Tx Buffer with the lowest buffer number is transmitted first

After the data section has been updated, a transmission is requested by an Add Request. This is done using the MCAN_TXBAR[x]ARn bit (where x = 0 to 31). The requested messages arbitrate internally with messages from an optional Tx FIFO or Tx Queue and externally with messages on the CAN bus, and are sent out according to their Message ID.

The following table shows the Tx Buffer, Tx FIFO, and Tx Queue Element Size. A Dedicated Tx Buffer allocates element size 32-bit words in the Message RAM. The start address of a dedicated Tx Buffer in the Message RAM is calculated by adding transmit buffer index from 0 to 31 (MCAN_TXFQS.TFQP) × Element size to the Tx Buffer start address MCAN_TXBC.TBSA field

**Table 32-6. Tx Buffer, Tx FIFO, Tx Queue Element Size**

| MCAN_TXESC.TBDS | DataField [Bytes] | Element Size [RAM Words] |
|---|---|---|
| 000 | 8 | 4 |
| 001 | 12 | 5 |
| 010 | 16 | 6 |
| 011 | 20 | 7 |
| 100 | 24 | 8 |
| 101 | 32 | 10 |
| 110 | 48 | 14 |
| 111 | 64 | 18 |



Copyright © 2026 Texas Instruments Incorporated

#### 32.4.15.3 Tx FIFO

Tx FIFO mode is configured by setting bit MCAN_TXBC.TFQM = 0. The data stored in the Tx FIFO messages are transmitted starting with the message referenced by the Get Index MCAN_TXFQS.TFGI field. After each transmission the Get Index is incremented until the Tx FIFO is empty. The Tx FIFO Free Level MCAN_TXFQS.TFFL field indicates the number of the available free Tx FIFO elements. The Tx FIFO allows transmission of messages with the same Message ID from different Tx buffers in the order these messages have been written to the Tx FIFO.

New transmit messages have to be written to the Tx FIFO starting with the Tx Buffer referenced by the Put Index MCAN_TXFQS.TFQP field. After each Add Request (MCAN_TXBAR[x] ARn = 1) the Put Index is incremented to the next free Tx FIFO element. When the Put Index reaches the Get Index (MCAN_TXFQS.TFQP = MCAN_TXFQS.TFGI), Tx FIFO Full condition is signaled by bit MCAN_TXFQS.TFQF = 1. In this case no further messages should be written to the Tx FIFO until the next message has been transmitted and the Get Index has been incremented.

The number of requested Tx buffers should not exceed the number of free Tx buffers as indicated by the Tx FIFO Free Level MCAN_TXFQS.TFFL field.

In case a transmission request for the Tx Buffer referenced by the Get Index is canceled, the Get Index is incremented to the next Tx Buffer with pending transmission request and the Tx FIFO Free Level MCAN_TXFQS.TFFL field is recalculated. In case transmission cancellation is applied to any other Tx Buffer - the Get Index and the FIFO Free Level remain unchanged.

A Tx FIFO element allocates element size 32-bit words in the Message RAM. The start address of the next available (free) Tx FIFO Buffer is calculated by adding Tx FIFO/Queue Put Index MCAN_TXFQS.TFQP (from 0 to 31) × Element Size to the Tx Buffer Start Address MCAN_TXBC.TBSA field.



Copyright © 2026 Texas Instruments Incorporated

#### 32.4.15.4 Tx Queue

Tx Queue mode is configured by setting bit MCAN_TXBC.TFQM = 1. The data stored in the Tx Queue messages are transmitted starting with the highest priority message (lowest Message ID). In case two or more Queue buffers are configured with the same Message ID, the Queue Buffer with the lowest buffer number is transmitted first.

New transmit messages have to be written to the Tx FIFO starting with the Tx Buffer referenced by the Put Index MCAN_TXFQS.TFQP field. Each Add Request cyclically increments the Put Index to the next free Tx Buffer. In case of Tx Queue Full condition (MCAN_TXFQS.TFQF = 1), the Put Index is not valid and no further message should be written to the Tx Queue until at least one of the requested messages has been sent out or a pending transmission request has been canceled.

The application may use the MCAN_TXBRP register instead of the Put Index and may place messages to any Tx Buffer without pending transmission request.

A Tx Queue Buffer allocates element size 32-bit words in the Message RAM. The start address of the next available (free) Tx Queue Buffer is calculated by adding Tx FIFO/Queue Put Index MCAN_TXFQS.TFQP (from 0 to 31) × Element Size to the Tx Buffer Start Address MCAN_TXBC.TBSA field.



Copyright © 2026 Texas Instruments Incorporated

#### 32.4.15.5 Mixed Dedicated Tx Buffers/Tx FIFO

For this combination the Tx buffers section in the Message RAM is separated in two parts:

- Dedicated Tx buffers: the number of dedicated Tx buffers is configured by MCAN_TXBC.NDTB field.
- Tx FIFO: the number of Tx buffers assigned to the Tx FIFO is configured by the MCAN_TXBC.TFQS field. If the MCAN_TXBC.TFQS field is empty(zero)- only dedicated Tx buffers are used.

**Tx prioritization:**

- Scan Dedicated Tx buffers and oldest pending Tx FIFO Buffer (referenced by the MCAN_TXFQS.TFGI field)
- Buffer with lowest Message ID gets highest priority and is transmitted next
- The following figure shows Mixed Dedicated Tx buffers/Tx FIFO example.

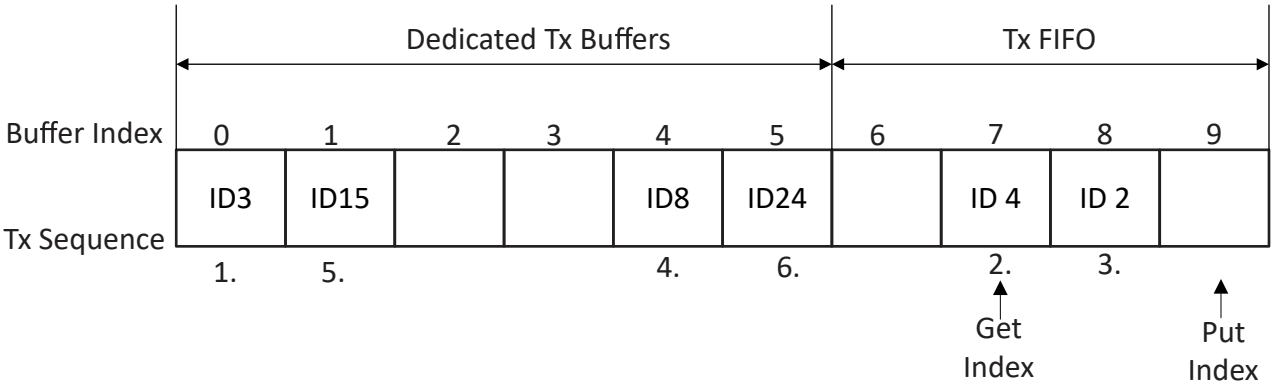



**Figure 32-14. Mixed Dedicated Tx Buffers/ Tx FIFO (example)**



Copyright © 2026 Texas Instruments Incorporated

#### 32.4.15.6 Mixed Dedicated Tx Buffers/Tx Queue

For this combination the Tx buffers section in the Message RAM is separated in two parts:

- Dedicated Tx buffers: the number of Dedicated Tx buffers is configured by the MCAN_TXBC.NDTB field.
- Tx Queue: the number of Tx buffers assigned to the Tx Queue is configured by the MCAN_TXBC.TFQS field If MCAN_TXBC.TFQS field is empty (zero) - only Dedicated Tx buffers are used.

**Tx prioritization:**

- Scan all Tx buffers with activated transmission request
- Tx Buffer with lowest Message ID gets highest priority and is transmitted next
- The following figure shows Mixed Dedicated Tx buffers/Tx Queue example.

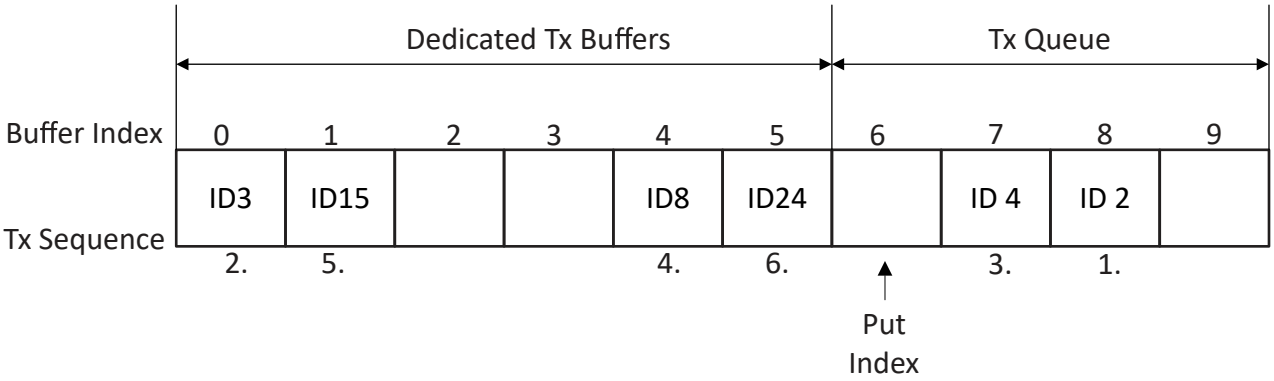



**Figure 32-15. Mixed Dedicated Tx Buffers/ Tx FIFO (Example)**



Copyright © 2026 Texas Instruments Incorporated

#### 32.4.15.7 Transmit Cancellation

This feature is especially intended for gateway and AUTOSAR based applications. The Host CPU can cancel a requested transmission from a dedicated Tx Buffer or a Tx Queue Buffer by setting bit MCAN_TXBCR[n] CRn = 1 (where n = 0 - 31). The corresponding bit position n is equivalent to the number of the Tx buffer.

Transmit cancellation is not intended for Tx FIFO operation.

Successful cancellation is signaled by setting the corresponding bit of the MCAN_TXBCF register (MCAN_TXBCF[n] CFn = 1).

If transmission from a Tx Buffer is already ongoing and a transmit cancellation is requested, the corresponding MCAN_TXBRP[n] TRPn bit remains set as long as the transmission is in progress. If the transmission was successful, the corresponding MCAN_TXBTO[n] TOn and MCAN_TXBCF[n] CFn bits are set. If the transmission was not successful, only the corresponding bit MCAN_TXBCF[n] CFn = 1.

**Note**

If pending transmission is canceled immediately before this transmission could have been started, a short time window occurs where no transmission is started even if another message is also pending in this node. This may enable another node to transmit a message that may have a lower priority than the second message in this node.



Copyright © 2026 Texas Instruments Incorporated

#### 32.4.15.8 Tx Event Handling

To support Tx Event Handling, the Message RAM has implemented a Tx Event FIFO section. Up to 32 Tx Event FIFO elements can be configured. Please see the Tx Event FIFO element chapter. After message transmission on the CAN bus, Message ID and Timestamp are stored in a Tx Event FIFO element. To link a Tx Event to a Tx Event FIFO element, the Message Marker from the transmitted Tx Buffer is copied into the Tx Event FIFO element.

A Tx Event FIFO full condition is signaled by the MCAN_IR.TEFF bit. In this case no further elements are written to the Tx Event FIFO until at least one element has been read out and the Tx Event FIFO Get Index has been incremented (MCAN_TXEFS.EFGI). In case a Tx Event occurs while the Tx Event FIFO is full, this event is rejected and interrupt flag MCAN_IR.TEFL bit is set.

The Tx Event FIFO watermark can be configured to avoid a Tx Event FIFO overflow. When the Tx Event FIFO fill level reaches the Tx Event FIFO watermark configured by the MCAN_TXEFC.EFWM field, interrupt flag MCAN_IR.TEFW is set. When reading from the Tx Event FIFO, two times the Tx Event FIFO Get Index MCAN_TXEFS.EFGI field has to be added to the Tx Event FIFO start address MCAN_TXEFC.EFSA field.



Copyright © 2026 Texas Instruments Incorporated

#### 32.4.15.9 FIFO Acknowledge Handling

The Get Indices of the two Rx FIFOs (Rx FIFO 0 or Rx FIFO 1) and the Tx Event FIFO are controlled by writing to the corresponding FIFO Acknowledge Index (see MCAN_RXF0A, MCAN_RXF1A, and MCAN_TXEFA). Writing to the FIFO Acknowledge Index will set the FIFO Get Index to the FIFO Acknowledge Index plus one and thereby updates the FIFO Fill Level.

There are two use cases:

- A single element has been read from the FIFO: the Get Index value is written to the FIFO Acknowledge Index.
- A sequence of elements has been read from the FIFO: the Get Index value (Index of the last element read) is written to the FIFO Acknowledge Index at the end of that read sequence.

The Host CPU has free access to the Message RAM. Special care has to be taken when reading FIFO elements in an arbitrary order (Get Index not considered). This can be useful when reading a High Priority Message from one of the two Rx FIFOs. In this case the FIFO's Acknowledge Index should not be written because this would set the Get Index to a wrong position and also changes the FIFO's Fill Level. In this case some of the older FIFO elements would be lost.

**Note**

The application has to ensure that a valid value is written to the FIFO Acknowledge Index. The MCAN module does not check for erroneous values.



Copyright © 2026 Texas Instruments Incorporated

### 32.4.16 Rx Handling

The Rx Handler controls the following operations:

- Acceptance filtering
- The transfer of received messages to the Rx buffers or to one of the two Rx FIFOs (Rx FIFO 0 or Rx FIFO 1)
- Rx FIFO Put and Get Index operations

#### 32.4.16.1 Acceptance Filtering

The MCAN module employs two sets of acceptance filters - one set for standard and one set for extended identifiers. These filters can be assigned to an Rx Buffer or to one of the two Rx FIFOs.

The main features of the filter elements are:

- Each filter element can be configured as:
  - Range filter (from - to)
  - Filter for specific IDs (for one or two dedicated IDs)
  - Classic bit mask filter
- Each filter element can be enabled/disabled individually
- Each filter element can be configured for acceptance or rejection filtering
- Filters are checked sequentially and execution (acceptance filtering procedure) stops at the first matching filter element or when the end of the filter list is reached

Related configuration registers are:

- Global Filter Configuration (MCAN_GFC) register
- Standard ID Filter Configuration (MCAN_SIDFC) register
- Extended ID Filter Configuration (MCAN_XIDFC) register
- Extended ID AND Mask (MCAN_XIDAM) register

Depending on the configuration of the filter element (see SFEC/EFEC in Section 32.4.19) if filter matches, one of the following actions is performed:

- Received frame is stored in FIFO 0 or FIFO 1
- Received frame is stored in Rx Buffer
- Received frame is stored in Rx Buffer and generation of pulse at filter event pin is performed. This is high level single MCAN_ICLK pulse.
- Received frame is rejected
- Set High Priority Message interrupt flag MCAN_IR.HPM
- Set High Priority Message interrupt flag MCAN_IR.HPM and store received frame in FIFO 0 or FIFO 1

Acceptance filtering starts when complete Message ID is received. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached. If a filter element matches - the Rx Handler starts writing the received message data in portions of 32-bit to the matching Rx Buffer or Rx FIFO. If an error condition occurs (for example: CRC error), this message is rejected with the following impact on the affected Rx Buffer or Rx FIFO:

- Rx Buffer: New Data flag (MCAN_NDAT1/MCAN_NDAT2) of matching Rx Buffer is not set, but Rx Buffer (partly) overwritten with received data (for error type, see MCAN_PSR.LEC and MCAN_PSR.DLEC fields,respectively).
- Rx FIFO: Put index of matching Rx FIFO is not updated, but related Rx FIFO element (partly) overwritten with received data (for error type, see MCAN_PSR.LEC and MCAN_PSR.DLEC fields, respectively). If matching Rx FIFO is configured to operate in overwrite mode, the boundary conditions described in Section 32.4.17.2 have to be considered.

**Note**

When an accepted message is written to one of the two Rx FIFOs, or into an Rx Buffer, the unmodified received identifier is stored independent of the filters used. The result of the acceptance filter process is strongly depending on the sequence of configured filter elements.



Copyright © 2026 Texas Instruments Incorporated

#### 32.4.16.1.1 Range Filter

Each filter element can be configured to operate as Range Filter (Standard Filter Type SFT = 00/Extended Filter Type EFT = 00). The filter matches for all received message frames with IDs in the range from SFID1 to SFID2 (SFID2 ≥ SFID1) respectively in the range from EFID1 to EFID2 (EFID2 ≥ EFID1). For more information see Section 32.4.19.5 and Section 32.4.19.6.

There are two options for range filtering of extended frames:

- Extended Filter Type EFT = 00: The Extended ID AND Mask (MCAN_XIDAM) is used for Range Filtering. The Message ID of received frames is ANDed with the Extended ID AND Mask (MCAN_XIDAM) before the range filter is applied.
- Extended Filter Type EFT = 11: The Extended ID AND Mask (MCAN_XIDAM) is not used for Range Filtering.

#### 32.4.16.1.2 Filter for Specific IDs

Each filter element can be configured to filter one or two dedicated Message IDs (Standard Filter Type SFT =01/ Extended Filter Type EFT =01). To filter only one specific Message ID, the filter element has to be configured with SFID1 = SFID2 respectively EFID1 = EFID2. For more information, see Section 32.4.19.5 and Section 32.4.19.6.

#### 32.4.16.1.3 Classic Bit Mask Filter

Classic bit mask filtering can filter groups of Message IDs (Standard Filter Type SFT =10/Extended Filter Type EFT =10). This is done by masking single bits of a received Message ID. In this case SFID1/EFID1 element is used as Message ID filter, while the SFID2/EFID2 element is used as filter mask.

A 0 bit at the filter mask (SFID2/EFID2) masks out the corresponding bit position of the configured Message ID filter (SFID1/EFID1) and the value of the received Message ID at that bit position is not relevant for acceptance filtering. Only those bits of the received Message ID where the corresponding mask bits are 1 are relevant for acceptance filtering.

There are two interesting cases:

- All mask bits are 1: a match occurs only when the received Message ID and the configured Message ID filter are identical.
- All mask bits are 0: all Message IDs match.



Copyright © 2026 Texas Instruments Incorporated

#### 32.4.16.1.4 Standard Message ID Filtering

Figure 32-16 shows the standard Message ID (11-bit ID) filtering flow. Section 32.4.19.5 describes the standard Message ID filter element.

The Remote Transmission Request (RTR) and Extended Identifier (XTD) bits of the received frames are compared against the list of configured filter elements. This is controlled by the following registers:

- Global Filter Configuration (MCAN_GFC) register
- Standard ID Filter Configuration (MCAN_SIDFC) register

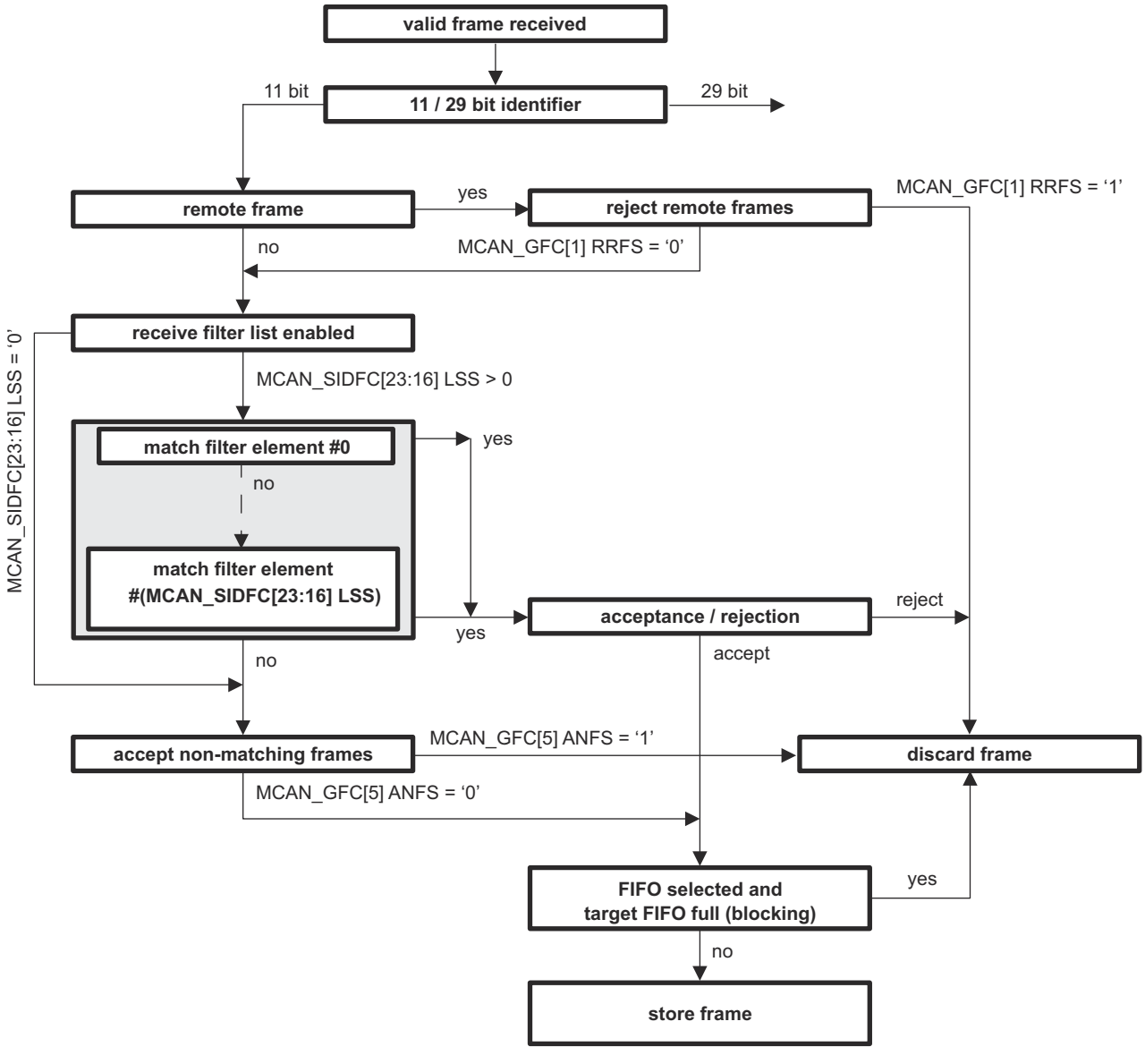



**Figure 32-16. Standard Message ID Filter Path**

**Note**

In MCAN_GFC[1]RRFS and MCAN_GFC[5]ANFS; the [1] and [5] denotes the bit position for the RRFS and ANFS bits respectively in the MCAN_GFC register



Copyright © 2026 Texas Instruments Incorporated

#### *32.4.16.1.5 Extended Message ID Filtering*

Figure 32-17 shows the extended Message ID (29-bit ID) filtering flow. Section 32.4.19.6 describes the extended Message ID filter element.

The Remote Transmission Request (RTR) and Extended Identifier (XTD) bits of the received frames are compared against the list of configured filter elements. This is controlled by the following registers:

- Global Filter Configuration (MCAN_GFC) register
- Extended ID Filter Configuration (MCAN_XIDFC) register

Note that before the filter list is executed, the received identifier is ANDed with the Extended ID AND Mask (MCAN_XIDAM).

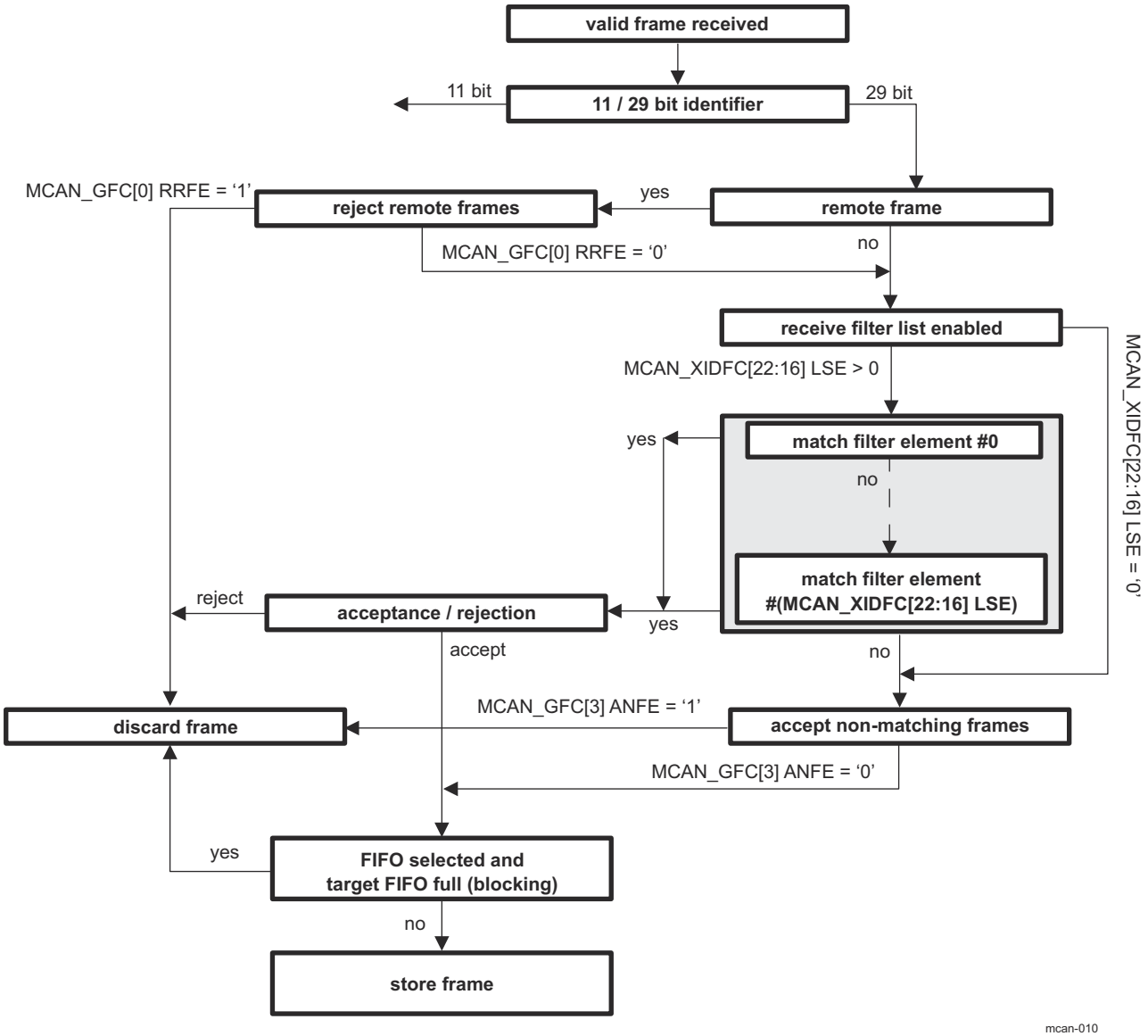



**Figure 32-17. Extended Message ID Filter Path**

In MCAN_GFC[0]RRFS and MCAN_GFC[3]ANFS; the [0] and [3] denotes the bit position for the RRFS and ANFS bits respectively in the MCAN_GFC register



Copyright © 2026 Texas Instruments Incorporated

### 32.4.17 Rx FIFOs

The configuration of the Rx FIFOs (Rx FIFO 0 and Rx FIFO 1) can be done by way of the MCAN_RXF0C and MCAN_RXF1C registers. Each Rx FIFO can be configured to store up to 64 received messages.

After acceptance filtering the received messages that passed are transferred to the Rx FIFO. The filter mechanisms available for the Rx FIFO 0 and Rx FIFO 1 are described in Section 32.4.16.1. The Rx FIFO element is described in Section 32.4.19.2.

The Rx FIFO watermark can be used to prevent an Rx FIFO overflow. If the Rx FIFO fill level reaches the Rx FIFO watermark configured by the MCAN_RXFnC[30:24] FnWM filed (where: n = 0 or 1), an interrupt flag MCAN_IR.RF0W/MCAN_IR.RF1W is set.

When the Rx FIFO Put Index reaches the Rx FIFO Get Index (MCAN_RXFnS[21:16] FnPI = MCAN_RXFnS[13:8] FnGI), an Rx FIFO Full condition is signaled by the MCAN_RXFnS[24] FnF status bit and interrupt flag MCAN_IR.RF0F/MCAN_IR.RF1F is set. Figure 32-18 shows Rx FIFO Status. The FIFOs fill level is presented in the MCAN_RXFnS[6:0] FnFL field (the number of elements stored in Rx FIFO).

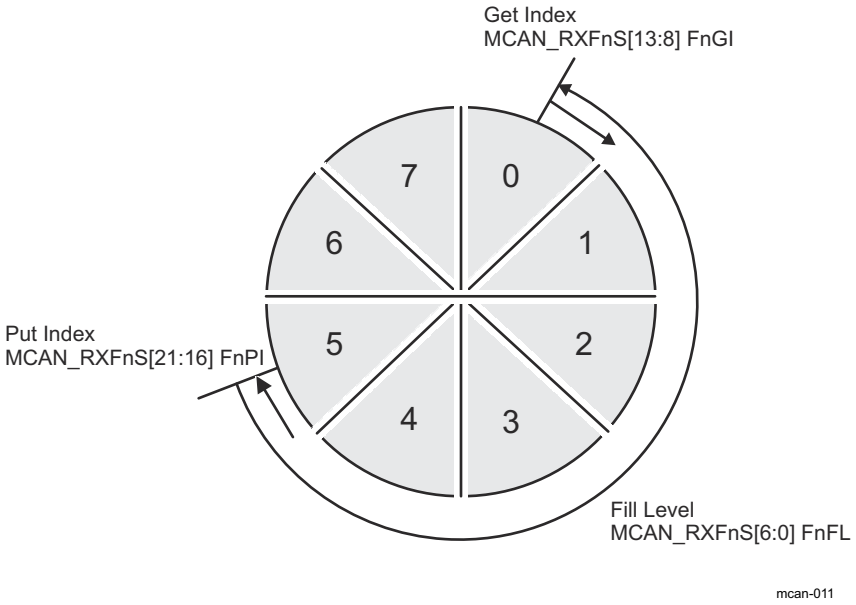



**Figure 32-18. Rx FIFO Status**

Rx FIFOs start address in the Message RAM (MCAN_RXFnC[15:2]FnSA field) has to be configured when reading from an Rx FIFO (Rx FIFO Get Index - MCAN_RXFnS[13:8] FnGI). Table 32-7 presents Rx Buffer/Rx FIFO Element Size for different Rx Buffer / Rx FIFO Data Field Size which is configured by way of the MCAN_RXESC register.

**Table 32-7. Rx Buffer/Rx FIFO Element Size**

| MCAN_RXESC Register RBDS/F0DS/F1DS Bits | Data Field [bytes] | FIFO Element Size [RAM words] |
|---|---|---|
| 000 | 8 | 4 |
| 001 | 12 | 5 |
| 010 | 16 | 6 |
| 011 | 20 | 7 |
| 100 | 24 | 8 |
| 101 | 32 | 10 |
| 110 | 48 | 14 |
| 111 | 64 | 18 |



Copyright © 2026 Texas Instruments Incorporated

#### 32.4.17.1 Rx FIFO Blocking Mode

The Rx FIFO blocking mode is the default operation mode for the Rx FIFOs and is configured by MCAN_RXFnC[31] FnOM = 0.

If an Rx FIFO full condition is reached (MCAN_RXFnS[21:16] FnPI = MCAN_RXFnS[13:8] FnGI), no further messages are written to the corresponding Rx FIFO until at least one message has been read out and the Rx FIFO Get Index has been incremented. An Rx FIFO full condition is signaled by the MCAN_RXFnS[24] FnF = 1 and interrupt flag MCAN_IR.RF0F/MCAN_IR.RF1F is set.

In case a message is received while the corresponding Rx FIFO is full, this message is rejected and the message lost condition is signaled by MCAN_RXFnS[25] RFnL = 1 and interrupt flag MCAN_IR.RF0L/ MCAN_IR.RF1L is set.

#### 32.4.17.2 Rx FIFO Overwrite Mode

The Rx FIFO overwrite mode is configured by MCAN_RXFnC[31] FnOM = 1. When an Rx FIFO full condition is reached (MCAN_RXFnS[21:16] FnPI = MCAN_RXFnS[13:8] FnGI) signaled by MCAN_RXFnS[24] FnF = 1, the next accepted message for the FIFO overwrites the oldest FIFO message. Put index/Get index are both incremented by one.

In overwrite mode if an Rx FIFO full condition is signaled, reading of the Rx FIFO elements starts at least at get index + 1. The reason for this is a received message is written to the Message RAM (Put index) while the Host CPU is reading from the Message RAM (Get index). In this case, inconsistent data can be read from the respective Rx FIFO element. The problem is solved by adding an offset to the Get index when reading from the Rx FIFO. The offset depends on how fast the Host CPU accesses the Rx FIFO. Figure 32-19 shows an offset of two with respect to the Get index when reading the Rx FIFO. In this case, the two messages stored in element 1 and 2 are lost.

After reading from the Rx FIFO, the number of the last element read has to be written to the Rx FIFO Acknowledge Index MCAN_RXFnA[5:0] FnAI. This increments the get index to that element number. In case the Put index has not been incremented to this Rx FIFO element, the Rx FIFO full condition is reset (MCAN_RXFnS[24] FnF = 0).

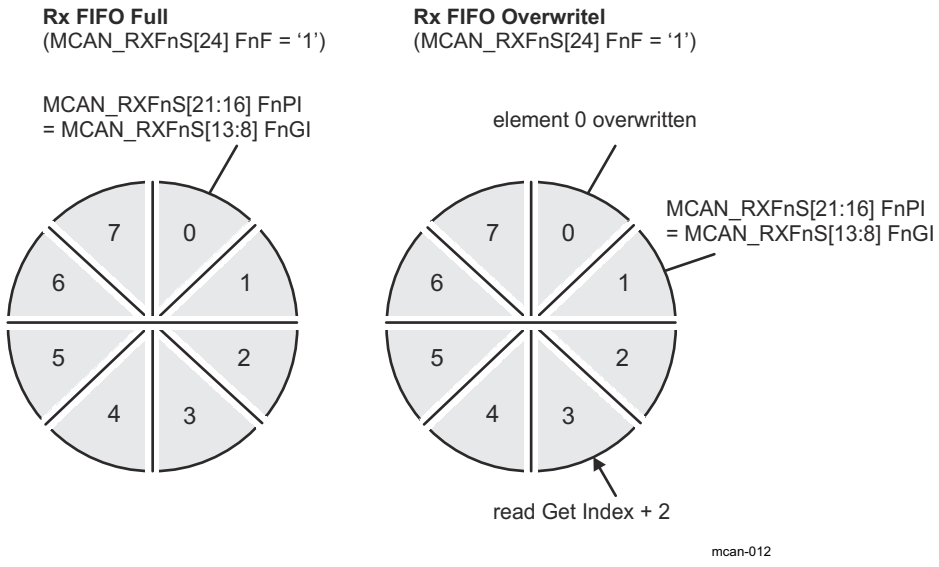



**Figure 32-19. Rx FIFO Overflow Handling**



Copyright © 2026 Texas Instruments Incorporated

### 32.4.18 Dedicated Rx Buffers

The MCAN supports up to 64 dedicated Rx buffers. The start address of the Rx buffers section in the Message RAM is configured by way of the MCAN_RXBC.RBSA field. To store in an Rx Buffer a Standard or Extended Message ID Filter Element with SFEC/EFEC = 111 and SFID2/EFID2[10:9] = 00 has to be configured (see Section 32.4.19.5 and Section 32.4.19.6).

After a received message has been accepted by a filter element, the message is stored into the Rx Buffer in the Message RAM referenced by the filter element (the format is the same as for an Rx FIFO element). In addition, the flag MCAN_IR.DRX (message stored in Dedicated Rx Buffer) is set.

Table 32-8 shows Example Filter Configuration for Rx buffers.

**Table 32-8. Example Filter Configuration for Rx Buffers**

| Filter Element | SFID1[10:0]<br>EFID1[28:0] | SFID2[10:9]<br>EFID2[10:9] | SFID2[5:0]<br>EFID2[5:0] |
|---|---|---|---|
| 0 | ID message 1 | 00 | 00 0000 |
| 1 | ID message 2 | 00 | 00 0001 |
| 2 | ID message 3 | 00 | 00 0010 |

After the last word of a matching received message has been written to the Message RAM, the respective New Data flag in register MCAN_NDAT1/MCAN_NDAT2 is set. As long as the New Data flag is set, the respective Rx Buffer is locked against updates from received matching frames. The New Data flags have to be reset by the Host CPU by writing a 1 to the respective bit position.

While an Rx buffer's New Data flag is set, a Message ID Filter Element referencing this specific Rx Buffer will not match, causing the acceptance filtering to continue. Following Message ID Filter Elements can cause the received message to be stored into another Rx Buffer, or into an Rx FIFO, or the message can be rejected, depending on filter configuration.

#### 32.4.18.1 Rx Buffer Handling

Rx Buffer Handling include the following steps:

- Reset interrupt flag MCAN_IR.DRX
- Read New Data registers
- Read messages from Message RAM
- Reset New Data flags of processed messages

### 32.4.19 Message RAM

The MCAN module has a Message RAM. The main purpose of the Message RAM is to store:

- Received Messages
- Transmit Messages
- Tx Event Elements
- Message ID Filter Elements



Copyright © 2026 Texas Instruments Incorporated

#### 32.4.19.1 Message RAM Configuration

The MCAN module can have different Message RAM sizes. The maximum size of the message RAM is 1kB and is offset 0x3FF from the base address. An example of the MCAN module being configured for 1KB size with a width of 32 bits is desribed here.

The Message RAM can include each of the sections listed in Message RAM Configuration. It is not necessary to configure each of the sections (a section in the Message RAM can be 0) and there is no restriction with respect to the sequence of the sections. For parity checking or ECC, a respective number of bits has to be added to each word. When the MCAN module addresses the Message RAM, it addresses 32-bit words. The start addresses are configurable and they are 32-bit word addresses.

The element size can be configured for:

- Rx FIFO 0, by way of the MCAN_RXESC.F0DS field
- Rx FIFO, 1 by way of the MCAN_RXESC.F1DS field
- Rx buffers, by way of the MCAN_RXESC.RBDS field
- Tx buffers, by way of the MCAN_TXESC.TBDS field

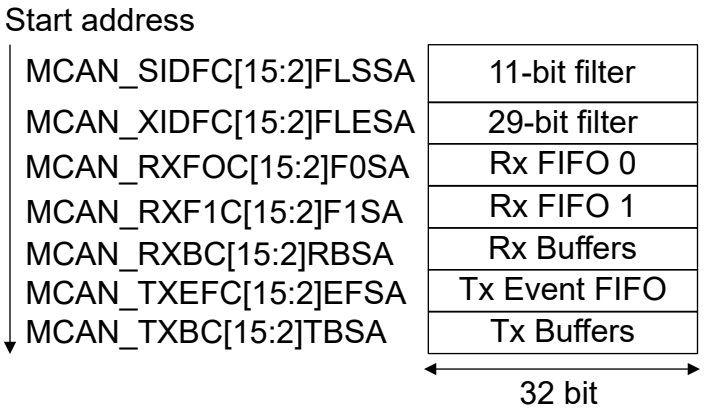



**Figure 32-20. Message RAM Configuration**

The host CPU configures the following information in the message RAM:

- Start addresses of the memory sections
- Number of elements in each section
- The size of the elements in some sections

**Note**

The MCAN module does not check for errors in the Message RAM configuration. The configuration of the start addresses of the different sections and the number of elements of each section has to be done carefully. This prevents falsification or loss of data.

An example of the max elements that can be accommodated for each section based on a RAM size of 1kB for the Rx FIFO is provided in Message Transmission configurations table

**Table 32-9. Message Transmission configurations**

| Frame Size, DLC Code (Bytes) | Element Size (Words) | Maximum Elements |
|---|---|---|
| 8 | 4 | 64 |
| 12 | 5 | 51 |
| 16 | 6 | 43 |
| 20 | 7 | 37 |
| 24 | 8 | 32 |
| 32 | 10 | 26 |
| 48 | 14 | 18 |



Copyright © 2026 Texas Instruments Incorporated

**Table 32-9. Message Transmission configurations (continued)**

| Frame Size, DLC Code (Bytes) | Element Size (Words) | Maximum Elements |
|---|---|---|
| 64 | 18 | 14 |



Copyright © 2026 Texas Instruments Incorporated

#### 32.4.19.2 Rx Buffer and FIFO Element

Up to 64 Rx buffers and two Rx FIFOs can be configured in the Message RAM. Each Rx FIFO section can be configured to store up to 64 received messages. The element size can be configured for storage of CAN FD messages with up to 64 bytes data field by way of the MCAN_RXESC register.

Figure 32-21 shows the Rx Buffer/Rx FIFO element structure. Table 32-10 shows the Rx Buffer/Rx FIFO element field descriptions.

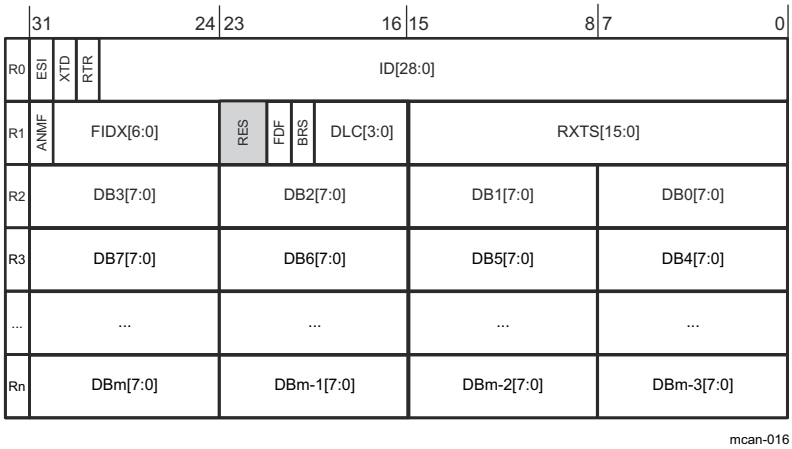



**Figure 32-21. Rx Buffer/Rx FIFO Element Structure**

**Table 32-10. Rx Buffer/Rx FIFO Element Field Descriptions**

| Word | Bits | Field Name | Description |
|---|---|---|---|
| R0 | 31 | ESI | Error State Indicator<br>• 0x0: Transmitting node is error active<br>• 0x1: Transmitting node is error passive |
| | 30 | XTD | Extended Identifier<br>Signals to the Host CPU whether the received frame has a standard or extended identifier.<br>• 0x0: 11-bit standard identifier<br>• 0x1: 29-bit extended identifier |
| | 29 | RTR | Remote Transmission Request<br>Signals to the Host CPU whether the received frame is a data frame or a remote frame.<br>• 0x0: Received frame is a data frame<br>• 0x1: Received frame is a remote frame<br>**Note:** There are no remote frames in CAN FD format. In case a CAN FD frame was received (FDF = 1), RTR bit reflects the state of the reserved r1 bit (RES[23]). In CAN FD frames (FDF=1), the dominant RRS (Remote Request Substitution) bit replaces the RTR (Remote Transmission Request) bit. |
| | 28:0 | ID[28:0] | Identifier<br>Standard or extended identifier depending on XTD bit. A standard identifier is stored into ID[28:18]. |



Copyright © 2026 Texas Instruments Incorporated

**Table 32-10. Rx Buffer/Rx FIFO Element Field Descriptions (continued)**

| Word | Bits | Field Name | Description |
|---|---|---|---|
| R1 | 31 | ANMF | Accepted Non-matching Frame<br>Acceptance of non-matching frames can be enabled using the MCAN_GFC.ANFS and MCAN_GFC.ANFE fields.<br>• 0x0: Received frame matching filter index FIDX field<br>• 0x1: Received frame did not match any Rx filter element |
| | 30:24 | FIDX[6:0] | Filter Index<br>0x0-0x7F (0-127): Index of matching Rx acceptance filter element (invalid if ANMF = 1).<br>Range is 0 to MCAN_SIDFC.LSS - 1 respectively MCAN_XIDFC.LSE - 1. |
| | 23:22 | RES | Reserved |
| | 21 | FDF | FD Format<br>• 0x0: Standard frame format<br>• 0x1: CAN FD frame format (new DLC-coding and CRC) |
| | 20 | BRS | Bit Rate Switch<br>• 0x0: Frame received without bit rate switching<br>• 0x1: Frame received with bit rate switching |
| | 19:16 | DLC[3:0] | Data Length Code<br>• 0x0-0x8 (0-8): CAN + CAN FD: received frame has 0-8 data bytes<br>• 0x9-0xF (9-15): CAN: received frame has 8 data bytes<br>• 0x9-0xF (9-15): CAN FD: received frame has 12/16/20/24/32/48/64 data bytes |
| | 15:0 | RXTS[15:0] | Rx Timestamp<br>Timestamp Counter value captured on start of frame reception.<br>Resolution depending on configuration of the Timestamp Counter Prescaler MCAN_TSCC.TCP. |
| R2 | 31:24 | DB3[7:0] | Data Byte 3 |
| | 23:16 | DB2[7:0] | Data Byte 2 |
| | 15:8 | DB1[7:0] | Data Byte 1 |
| | 7:0 | DB0[7:0] | Data Byte 0 |
| R3 | 31:24 | DB7[7:0] | Data Byte 7 |
| | 23:16 | DB6[7:0] | Data Byte 6 |
| | 15:8 | DB5[7:0] | Data Byte 5 |
| | 7:0 | DB4[7:0] | Data Byte 4 |
| ... | ... | ... | ... |
| Rn | 31:24 | DBm[7:0] | Data Byte m |
| | 23:16 | DBm-1[7:0] | Data Byte m-1 |
| | 15:8 | DBm-2[7:0] | Data Byte m-2 |
| | 7:0 | DBm-3[7:0] | Data Byte m-3 |

**Note:** Depending on the configuration of the element size (MCAN_RXESC), between two and sixteen 32-bit words (Rn = 3-17) are used for storage of a CAN message's data field.



Copyright © 2026 Texas Instruments Incorporated

#### 32.4.19.3 Tx Buffer Element

The Tx buffers section can be configured to hold dedicated Tx buffers as well as a Tx FIFO/Tx Queue. In case that the Tx buffers section is shared by dedicated Tx buffers and a Tx FIFO/Tx Queue, the dedicated Tx buffers start at the beginning of the Tx buffers section followed by the buffers assigned to the Tx FIFO or Tx Queue. The Tx Handler makes difference between dedicated Tx buffers and Tx FIFO/Tx Queue by way of the MCAN_TXBC.TFQS and MCAN_TXBC.NDTB fields. The element size can be configured for storage of CAN FD messages with up to 64 bytes data field by way of the MCAN_TXESC register.

Figure 32-22 shows the Tx Buffer element structure. Table 32-11 shows the Tx Buffer element field descriptions.

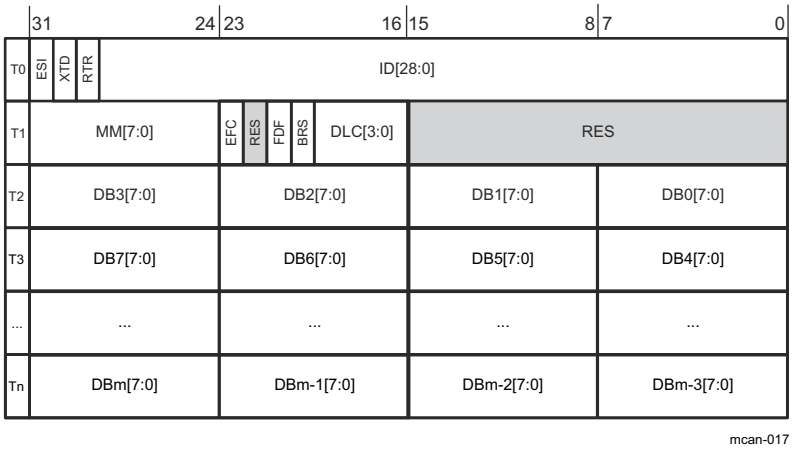



**Figure 32-22. Tx Buffer Element Structure**

**Table 32-11. Tx Buffer Element Field Descriptions**

| Word | Bits | Field Name | Description |
|---|---|---|---|
| T0 | 31 | ESI | Error State Indicator<br>• 0x0: ESI bit in CAN FD format depends only on error passive flag<br>• 0x1: ESI bit in CAN FD format transmitted recessive<br>**Note:** The ESI bit of the transmit buffer is ORed with the error passive flag to decide the value of the ESI bit in the transmitted CAN FD frame. As required by the CAN FD protocol specification, an error active node can optionally transmit the ESI bit recessive, but an error passive node always transmits the ESI bit recessive. |
| | 30 | XTD | Extended Identifier<br>• 0x0: 11-bit standard identifier<br>• 0x1: 29-bit extended identifier |
| | 29 | RTR | Remote Transmission Request<br>• 0x0: Transmit data frame<br>• 0x1: Transmit remote frame<br>**Note:** When RTR = 1, the MCAN module transmits a remote frame according to ISO11898-1:2015, even if the MCAN_CCCR.FDOE bit enables the transmission in CAN FD format. |
| | 28:0 | ID[28:0] | Identifier<br>Standard or extended identifier depending on XTD bit. A standard identifier has to be written to ID[28:18]. |



Copyright © 2026 Texas Instruments Incorporated

**Table 32-11. Tx Buffer Element Field Descriptions (continued)**

| Word | Bits | Field Name | Description |
|---|---|---|---|
| T1 | 31:24 | MM[7:0] | Message Marker<br>Written by Host CPU during Tx Buffer configuration. Copied into Tx Event FIFO element for identification of Tx message status (see also MM[7:0] field in Table 32-12). |
| | 23 | EFC | Event FIFO Control<br>• 0x0: Don't store Tx events<br>• 0x1: Store Tx events |
| | 22 | RES | Reserved |
| | 21 | FDF | FD Format<br>• 0x0: Frame transmitted in Classic CAN format<br>• 0x1: Frame transmitted in CAN FD format |
| | 20 | BRS | Bit Rate Switch<br>• 0x0: CAN FD frames transmitted without bit rate switching<br>• 0x1: CAN FD frames transmitted with bit rate switching<br>**Note:** ESI, FDF, and BRS bits are only evaluated when CAN FD operation is enabled using the MCAN_CCCR.FDOE bit. BRS bit is only evaluated when MCAN_CCCR.BRSE = 1. |
| | 19:16 | DLC[3:0] | Data Length Code<br>• 0x0-0x8 (0-8): CAN + CAN FD: transmit frame has 0-8 data bytes<br>• 0x9-0xF (9-15): CAN: transmit frame has 8 data bytes<br>• 0x9-0xF (9-15): CAN FD: transmit frame has 12/16/20/24/32/48/64 data bytes |
| | 15:0 | RES | Reserved |
| T2 | 31:24 | DB3[7:0] | Data Byte 3 |
| | 23:16 | DB2[7:0] | Data Byte 2 |
| | 15:8 | DB1[7:0] | Data Byte 1 |
| | 7:0 | DB0[7:0] | Data Byte 0 |
| T3 | 31:24 | DB7[7:0] | Data Byte 7 |
| | 23:16 | DB6[7:0] | Data Byte 6 |
| | 15:8 | DB5[7:0] | Data Byte 5 |
| | 7:0 | DB4[7:0] | Data Byte 4 |
| ... | ... | ... | ... |
| Tn | 31:24 | DBm[7:0] | Data Byte m |
| | 23:16 | DBm-1[7:0] | Data Byte m-1 |
| | 15:8 | DBm-2[7:0] | Data Byte m-2 |
| | 7:0 | DBm-3[7:0] | Data Byte m-3 |

**Note**

Depending on the configuration of the element size (MCAN_TXESC), between two and sixteen 32-bit words (Tn = 3-17) are used for storage of a CAN message's data field.



Copyright © 2026 Texas Instruments Incorporated

#### 32.4.19.4 Tx Event FIFO Element

Each element stores information about transmitted messages. By reading the Tx Event FIFO the Host CPU gets this information in the order the messages were transmitted. Status information about the Tx Event FIFO can be obtained from the MCAN_TXEFS register.

Figure 32-23 shows the Tx Event FIFO element structure. Table 32-12 shows the Tx Event FIFO element field descriptions.

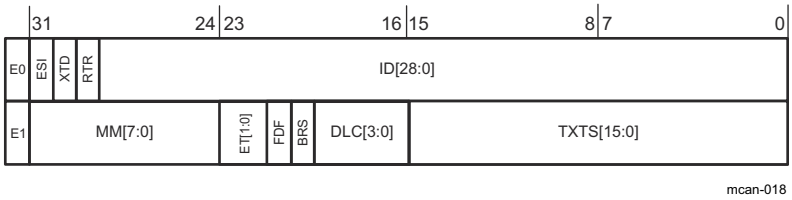



**Figure 32-23. Tx Event FIFO Element Structure**

**Table 32-12. Tx Event FIFO Element Field Descriptions**

| Word | Bits | Field Name | Description |
|---|---|---|---|
| E0 | 31 | ESI | Error State Indicator<br>• 0x0: Transmitting node is error active<br>• 0x1: Transmitting node is error passive |
| | 30 | XTD | Extended Identifier<br>• 0x0: 11-bit standard identifier<br>• 0x1: 29-bit extended identifier |
| | 29 | RTR | Remote Transmission Request<br>• 0x0: Data frame transmitted<br>• 0x1: Remote frame transmitted |
| | 28:0 | ID[28:0] | Identifier<br>Standard or extended identifier depending on XTD bit. A standard identifier has to be written to ID[28:18]. |



Copyright © 2026 Texas Instruments Incorporated

**Table 32-12. Tx Event FIFO Element Field Descriptions (continued)**

| Word | Bits | Field Name | Description |
|---|---|---|---|
| E1 | 31:24 | MM[7:0] | Message Marker<br>Copied from Tx Buffer into Tx Event FIFO element for identification of Tx message status (see also MM[7:0] field in Table 32-11). |
| | 23:22 | ET[1:0] | Event Type<br>• 0x0: Reserved<br>• 0x1: Tx event<br>• 0x2: Transmission in spite of cancellation (always set for transmissions in DAR mode)<br>• 0x3: Reserved |
| | 21 | FDF | FD Format<br>• 0x0: Standard frame format<br>• 0x1: CAN FD frame format (new DLC-coding and CRC) |
| | 20 | BRS | Bit Rate Switch<br>• 0x0: Frame transmitted without bit rate switching<br>• 0x1: Frame transmitted with bit rate switching |
| | 19:16 | DLC[3:0] | Data Length Code<br>• 0x0-0x8 (0-8): CAN + CAN FD: frame with 0-8 data bytes transmitted<br>• 0x9-0xF (9-15): CAN: frame with 8 data bytes transmitted<br>• 0x9-0xF (9-15): CAN FD: frame with 12/16/20/24/32/48/64 data bytes transmitted |
| | 15:0 | TXTS[15:0] | Tx Timestamp<br>Timestamp Counter value captured on start of frame transmission. Resolution depending on configuration of the Timestamp Counter Prescaler MCAN_TSCC.TCP filed. |

#### 32.4.19.5 Standard Message ID Filter Element

Up to 128 filter elements can be configured for 11-bit standard IDs. When accessing a Standard Message ID Filter element, the element address is the Filter List Standard Start Address MCAN_SIDFC.FLSSA field plus the index of the filter element (0-127).

Figure 32-24 shows the Standard Message ID Filter element structure. Table 32-13 shows the Standard Message ID Filter element field descriptions.

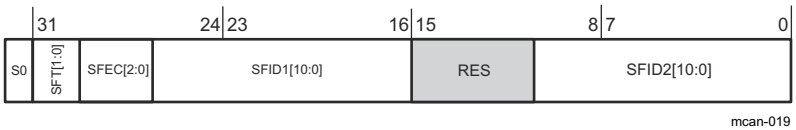



**Figure 32-24. Standard Message ID Filter Element Structure**



Copyright © 2026 Texas Instruments Incorporated

**Table 32-13. Standard Message ID Filter Element Field Descriptions**

| Word | Bits | Field Name | Description |
|---|---|---|---|
| S0 | 31:30 | SFT[1:0] | Standard Filter Type<br>• 0x0: Range filter from SFID1 to SFID2 (SFID2 ≥ SFID1)<br>• 0x1: Dual ID filter for SFID1 or SFID2<br>• 0x2: Classic filter: SFID1 = filter; SFID2 = mask<br>• 0x3: Filter element disabled<br>**Note:** With SFT = 11 the filter element is disabled and the acceptance filtering continues (same behavior as with SFEC = 000) |
| | 29:27 | SFEC[2:0] | Standard Filter Element Configuration<br>All enabled filter elements are used for acceptance filtering of standard frames. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached. If SFEC = 100, 101, or 110 match sets interrupt flag MCAN_IR.HPM and, if enabled, an interrupt is generated. In this case, the MCAN_HPMS register is updated with the status of the priority match.<br>• 0x0: Disable filter element<br>• 0x1: Store in Rx FIFO 0 if filter matches<br>• 0x2: Store in Rx FIFO 1 if filter matches<br>• 0x3: Reject ID if filter matches<br>• 0x4: Set priority if filter matches<br>• 0x5: Set priority and store in FIFO 0 if filter matches<br>• 0x6: Set priority and store in FIFO 1 if filter matches<br>• 0x7: Store into Rx Buffer , configuration of SFT[1:0] ignored |
| | 26:16 | SFID1[10:0] | Standard Filter ID 1<br>When filtering for Rx buffers this field defines the ID of a standard message to be stored. The received identifiers must match exactly, no masking mechanism is used. |
| | 15:11 | RES | Reserved |
| | 10:0 | SFID2[10:0] | Standard Filter ID 2<br>This bit field has a different meaning depending on the configuration of SFEC:<br>• SFEC = 001 - 110: Second ID of standard ID filter element<br>• SFEC = 111: Filter for Rx buffers |
| | | SFID2[10:9] | This filed is decides whether the received message is stored into an Rx Buffer or treated as message A, B, or C of the debug message sequence.<br>• 0x0: Store message into an Rx Buffer<br>• 0x1: Debug Message A<br>• 0x2: Debug Message B<br>• 0x3: Debug Message C<br>**Note:** Debug feature is not supported. |
| | | SFID2[8:6] | This field is used to control the filter event pins at the Extension Interface. A one at the respective bit position enables generation of a pulse at the related filter event pin with the duration of one MCAN_ICKL period in case the filter matches.<br>**Note:** Only two filter event pins are supported. |
| | | SFID2[5:0] | This field defines the offset to the Rx Buffer Start Address MCAN_RXBC.RBSA field for storage of a matching message. |



Copyright © 2026 Texas Instruments Incorporated

#### 32.4.19.6 Extended Message ID Filter Element

Up to 64 filter elements can be configured for 29-bit extended IDs. When accessing an Extended Message ID Filter element, the element address is the Filter List Extended Start Address MCAN_XIDFC.FLESA field plus two times the index of the filter element (0-63).

Figure 32-25 shows the Extended Message ID Filter element structure. Table 32-14 shows the Extended Message ID Filter element field descriptions.

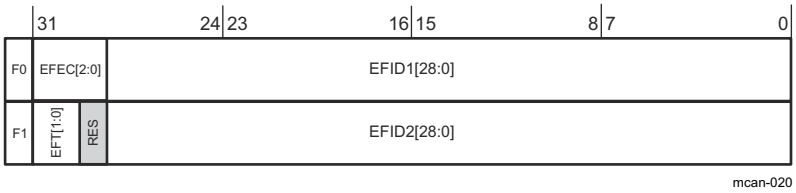



**Figure 32-25. Extended Message ID Filter Element Structure**

**Table 32-14. Extended Message ID Filter Element Field Descriptions**

| Word | Bits | Field Name | Description |
|---|---|---|---|
| F0 | 31:29 | EFEC[2:0] | Extended Filter Element Configuration<br>All enabled filter elements are used for acceptance filtering of extended frames. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached. If EFEC = 100, 101, or 110 match sets interrupt flag MCAN_IR.HPM and, if enabled, an interrupt is generated. In this case, the MCAN_HPMS register is updated with the status of the priority match.<br>• 0x0: Disable filter element<br>• 0x1: Store in Rx FIFO 0 if filter matches<br>• 0x2: Store in Rx FIFO 1 if filter matches<br>• 0x3: Reject ID if filter matches<br>• 0x4: Set priority if filter matches<br>• 0x5: Set priority and store in FIFO 0 if filter matches<br>• 0x6: Set priority and store in FIFO 1 if filter matches<br>• 0x7: Store into Rx Buffer or as debug message, configuration of EFT[1:0] ignored |
| | 28:0 | EFID1[28:0] | Extended Filter ID 1<br>First ID of extended ID filter element.<br>When filtering for Rx buffers this field defines the ID of an extended message to be stored. The received identifiers must match exactly, only XIDAM masking mechanism (see Section 32.4.16.1.5) is used. |



Copyright © 2026 Texas Instruments Incorporated

**Table 32-14. Extended Message ID Filter Element Field Descriptions (continued)**

| Word | Bits | Field Name | Description |
|---|---|---|---|
| F1 | 31:30 | EFT[1:0] | Extended Filter Type<br>• 0x0: Range filter from EFID1 to EFID2 (EFID2 ≥ EFID1)<br>• 0x1: Dual ID filter for EFID1 or EFID2<br>• 0x2: Classic filter: EFID1 = filter, EFID2 = mask<br>• 0x3: Range filter from EFID1 to EFID2 (EFID2 ≥ EFID1), XIDAM mask not applied |
| | 29 | RES | Reserved |
| | 28:0 | EFID2[28:0] | Extended Filter ID 2<br>This bit field has a different meaning depending on the configuration of EFEC:<br>• EFEC = 001 - 110: Second ID of extended ID filter element<br>• EFEC = 111: Filter for Rx buffers |
| | | EFID2[10:9] | This field decides whether the received message is stored into an Rx Buffer or treated as message A, B, or C of the debug message sequence.<br>• 0x0: Store message into an Rx Buffer<br>• 0x1: Debug Message A<br>• 0x2: Debug Message B<br>• 0x3: Debug Message C<br>**Note:** Debug feature is not supported. |
| | | EFID2[8:6] | This field is used to control the filter event pins at the Extension Interface. A one at the respective bit position enables generation of a pulse at the related filter event pin with the duration of one MCAN_ICKL period in case the filter matches.<br>**Note:** Only two filter event pins are supported. |
| | | EFID2[5:0] | This field defines the offset to the Rx Buffer Start Address MCAN_RXBC.RBSA field for storage of a matching message. |

## 32.5 MCAN Integration

Figure 32-26 shows the integration of the MCAN module in the device.



Copyright © 2026 Texas Instruments Incorporated

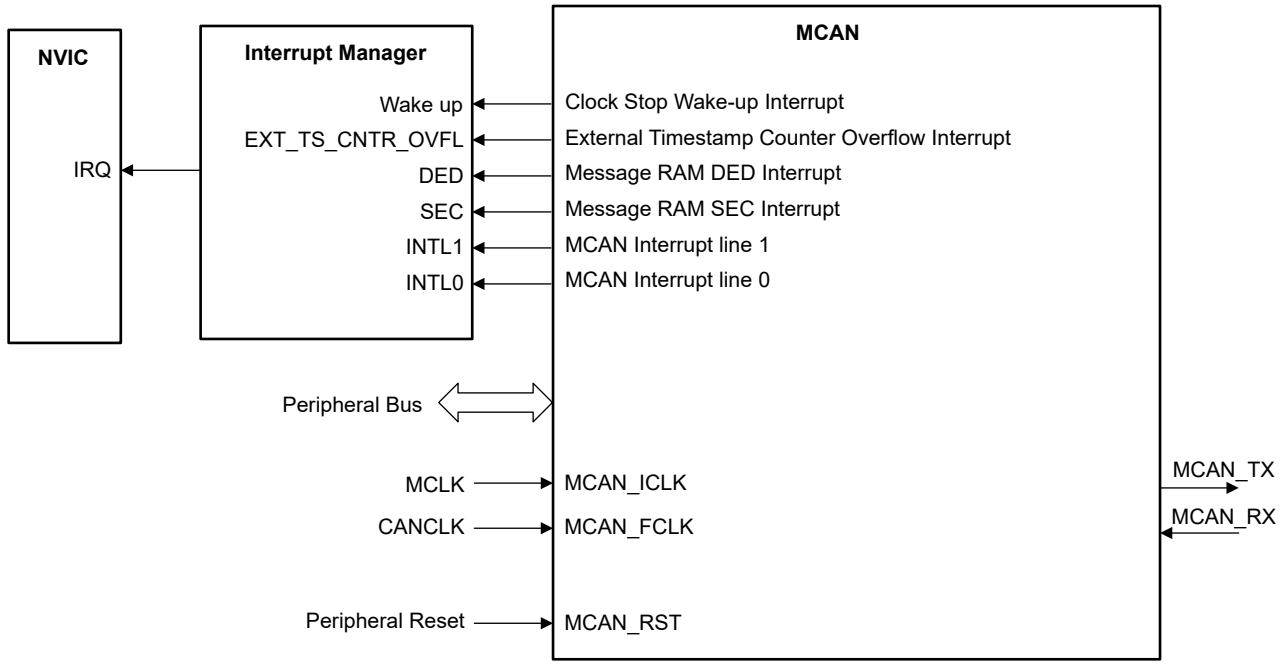



**Figure 32-26. MCAN Integration**

Table 32-15 summarizes the integration of the MCAN module in the device.

**Table 32-15. MCAN Clocks and Resets**

| Destination Signal Name | Source Signal Name | Description |
|---|---|---|
| **Clocks** | | |
| MCAN_ICLK | MCLK | Interface clock for the MCAN module |
| MCAN_FCLK | CANCLK | Bit timing clock for MCAN |
| **Resets** | | |
| MCAN_RST | Peripheral Reset | Asynchronous reset signal to the MCAN module |

## 32.6 Interrupt and Event Support

The MCAN module contains one event publisher (CPU_INT) that manages MCAN interrupt requests (IRQs) to the CPU subsystem via a static event route.

The MCAN events are summarized in Table 32-16.

**Table 32-16. MCAN Events**

| Event | Type | Source | Destination | Route | Configuration | Functionality |
|---|---|---|---|---|---|---|
| CPU interrupt | Publisher | MCAN | CPU Subsystem | Static route | CPU_INT registers | Fixed interrupt route from MCAN to CPU |

### *32.6.1 CPU Interrupt Event Publisher (CPU_INT)*

The MCAN module provides different interrupt sources which can be configured to source a CPU interrupt event. In order of decreasing interrupt priority, the CPU interrupt events from the MCAN are shown in Table 32-17.

**Table 32-17. MCAN Interrupt Event Conditions (CPU_INT)**

| Index (IIDX) | Name | Description |
|---|---|---|
| 0x1 | MCAN_IRQ_INT0 | MCAN interrupt 0 |
| 0x2 | MCAN_IRQ_INT1 | MCAN interrupt 1 |
| 0x3 | MCAN_IRQ_ECC | MCAN ECC SEC (single error correct) interrupt |



Copyright © 2026 Texas Instruments Incorporated

**Table 32-17. MCAN Interrupt Event Conditions (CPU_INT) (continued)**

| Index (IIDX) | Name | Description |
|---|---|---|
| 0x4 | MCAN_IRQ_ECC_UNCORR | MCAN ECC uncorrectable / DED (double error detect) interrupt |
| 0x5 | MCANSS_IRQ_TS_CNTR_OVFL | MCAN timestamp counter overflow interrupt |
| 0x6 | MCANSS_IRQ_TS_WAKE | MCAN clock stop wakeup interrupt |

The CPU interrupt event configuration is managed with the CPU_INT event management registers. Interrupt (RIS) flags are cleared upon software reading the IIDX register or writing to the respective ICLR register bits.

See *Using Event Registers* for guidance on configuring the event registers for CPU interrupts.



Copyright © 2026 Texas Instruments Incorporated

## 32.7 MCAN Registers

Table 32-18 lists the memory-mapped registers for the MCAN registers. All register offset addresses not listed in Table 32-18 should be considered as reserved locations and the register contents should not be modified.

**Table 32-18. MCAN Registers**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 6004h | CANRX | CAN RX IO | | Go |
| 6008h | CANTX | CAN TX IO | | Go |
| 6204h | CANRX | FUPDATE version of CANRX | | Go |
| 6208h | CANTX | FUPDATE version of CANTX | | Go |
| 6480h | CPU_CONNECT_0 | CPU Connect | | Go |
| 6800h | PWREN | Power enable | | Go |
| 6804h | RSTCTL | Reset Control | | Go |
| 6814h | STAT | Status Register | | Go |
| 7000h | MCAN_CREL | MCAN Core Release Register | | Go |
| 7004h | MCAN_ENDN | MCAN Endian Register | | Go |
| 700Ch | MCAN_DBTP | MCAN Data Bit Timing and Prescaler Register | | Go |
| 7010h | MCAN_TEST | MCAN Test Register | | Go |
| 7014h | MCAN_RWD | MCAN RAM Watchdog | | Go |
| 7018h | MCAN_CCCR | MCAN CC Control Register | | Go |
| 701Ch | MCAN_NBTP | MCAN Nominal Bit Timing and Prescaler Register | | Go |
| 7020h | MCAN_TSCC | MCAN Timestamp Counter Configuration | | Go |
| 7024h | MCAN_TSCV | MCAN Timestamp Counter Value | | Go |
| 7028h | MCAN_TOCC | MCAN Timeout Counter Configuration | | Go |
| 702Ch | MCAN_TOCV | MCAN Timeout Counter Value | | Go |
| 7040h | MCAN_ECR | MCAN Error Counter Register | | Go |
| 7044h | MCAN_PSR | MCAN Protocol Status Register | | Go |
| 7048h | MCAN_TDCR | MCAN Transmitter Delay Compensation Register | | Go |
| 7050h | MCAN_IR | MCAN Interrupt Register | | Go |
| 7054h | MCAN_IE | MCAN Interrupt Enable | | Go |
| 7058h | MCAN_ILS | MCAN Interrupt Line Select | | Go |
| 705Ch | MCAN_ILE | MCAN Interrupt Line Enable | | Go |
| 7080h | MCAN_GFC | MCAN Global Filter Configuration | | Go |
| 7084h | MCAN_SIDFC | MCAN Standard ID Filter Configuration | | Go |
| 7088h | MCAN_XIDFC | MCAN Extended ID Filter Configuration | | Go |
| 7090h | MCAN_XIDAM | MCAN Extended ID and Mask | | Go |
| 7094h | MCAN_HPMS | MCAN High Priority Message Status | | Go |
| 7098h | MCAN_NDAT1 | MCAN New Data 1 | | Go |
| 709Ch | MCAN_NDAT2 | MCAN New Data 2 | | Go |
| 70A0h | MCAN_RXF0C | MCAN Rx FIFO 0 Configuration | | Go |
| 70A4h | MCAN_RXF0S | MCAN Rx FIFO 0 Status | | Go |
| 70A8h | MCAN_RXF0A | MCAN Rx FIFO 0 Acknowledge | | Go |
| 70ACh | MCAN_RXBC | MCAN Rx Buffer Configuration | | Go |
| 70B0h | MCAN_RXF1C | MCAN Rx FIFO 1 Configuration | | Go |



Copyright © 2026 Texas Instruments Incorporated

**Table 32-18. MCAN Registers (continued)**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 70B4h | MCAN_RXF1S | MCAN Rx FIFO 1 Status | | Go |
| 70B8h | MCAN_RXF1A | MCAN Rx FIFO 1 Acknowledge | | Go |
| 70BCh | MCAN_RXESC | MCAN Rx Buffer / FIFO Element Size Configuration | | Go |
| 70C0h | MCAN_TXBC | MCAN Tx Buffer Configuration | | Go |
| 70C4h | MCAN_TXFQS | MCAN Tx FIFO / Queue Status | | Go |
| 70C8h | MCAN_TXESC | MCAN Tx Buffer Element Size Configuration | | Go |
| 70CCh | MCAN_TXBRP | MCAN Tx Buffer Request Pending | | Go |
| 70D0h | MCAN_TXBAR | MCAN Tx Buffer Add Request | | Go |
| 70D4h | MCAN_TXBCR | MCAN Tx Buffer Cancellation Request | | Go |
| 70D8h | MCAN_TXBTO | MCAN Tx Buffer Transmission Occurred | | Go |
| 70DCh | MCAN_TXBCF | MCAN Tx Buffer Cancellation Finished | | Go |
| 70E0h | MCAN_TXBTIE | MCAN Tx Buffer Transmission Interrupt Enable | | Go |
| 70E4h | MCAN_TXBCIE | MCAN Tx Buffer Cancellation Finished Interrupt Enable | | Go |
| 70F0h | MCAN_TXEFC | MCAN Tx Event FIFO Configuration | | Go |
| 70F4h | MCAN_TXEFS | MCAN Tx Event FIFO Status | | Go |
| 70F8h | MCAN_TXEFA | MCAN Tx Event FIFO Acknowledge | | Go |
| 7200h | MCANSS_PID | MCAN Subsystem Revision Register | | Go |
| 7204h | MCANSS_CTRL | MCAN Subsystem Control Register | | Go |
| 7208h | MCANSS_STAT | MCAN Subsystem Status Register | | Go |
| 720Ch | MCANSS_ICS | MCAN Subsystem Interrupt Clear Shadow Register | | Go |
| 7210h | MCANSS_IRS | MCAN Subsystem Interrupt Raw Status Register | | Go |
| 7214h | MCANSS_IECS | MCAN Subsystem Interrupt Enable Clear Shadow Register | | Go |
| 7218h | MCANSS_IE | MCAN Subsystem Interrupt Enable Register | | Go |
| 721Ch | MCANSS_IES | MCAN Subsystem Interrupt Enable Status | | Go |
| 7220h | MCANSS_EOI | MCAN Subsystem End of Interrupt | | Go |
| 7224h | MCANSS_EXT_TS_PRESCALER | MCAN Subsystem External Timestamp Prescaler 0 | | Go |
| 7228h | MCANSS_EXT_TS_UNSERVICED_INTR_CNTR | MCAN Subsystem External Timestamp Unserviced Interrupts Counter | | Go |
| 7400h | MCANERR_REV | MCAN Error Aggregator Revision Register | | Go |
| 7408h | MCANERR_VECTOR | MCAN ECC Vector Register | | Go |
| 740Ch | MCANERR_STAT | MCAN Error Misc Status | | Go |
| 7410h | MCANERR_WRAP_REV | MCAN ECC Wrapper Revision Register | | Go |
| 7414h | MCANERR_CTRL | MCAN ECC Control | | Go |
| 7418h | MCANERR_ERR_CTRL1 | MCAN ECC Error Control 1 Register | | Go |
| 741Ch | MCANERR_ERR_CTRL2 | MCAN ECC Error Control 2 Register | | Go |
| 7420h | MCANERR_ERR_STAT1 | MCAN ECC Error Status 1 Register | | Go |
| 7424h | MCANERR_ERR_STAT2 | MCAN ECC Error Status 2 Register | | Go |
| 7428h | MCANERR_ERR_STAT3 | MCAN ECC Error Status 3 Register | | Go |



Copyright © 2026 Texas Instruments Incorporated

**Table 32-18. MCAN Registers (continued)**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 743Ch | MCANERR_SEC_EOI | MCAN Single Error Corrected End of Interrupt Register | | Go |
| 7440h | MCANERR_SEC_STATUS | MCAN Single Error Corrected Interrupt Status Register | | Go |
| 7480h | MCANERR_SEC_ENABLE_SET | MCAN Single Error Corrected Interrupt Enable Set Register | | Go |
| 74C0h | MCANERR_SEC_ENABLE_CLR | MCAN Single Error Corrected Interrupt Enable Clear Register | | Go |
| 753Ch | MCANERR_DED_EOI | MCAN Double Error Detected End of Interrupt Register | | Go |
| 7540h | MCANERR_DED_STATUS | MCAN Double Error Detected Interrupt Status Register | | Go |
| 7580h | MCANERR_DED_ENABLE_SET | MCAN Double Error Detected Interrupt Enable Set Register | | Go |
| 75C0h | MCANERR_DED_ENABLE_CLR | MCAN Double Error Detected Interrupt Enable Clear Register | | Go |
| 7600h | MCANERR_AGGR_ENABLE_SET | MCAN Error Aggregator Enable Set Register | | Go |
| 7604h | MCANERR_AGGR_ENABLE_CLR | MCAN Error Aggregator Enable Clear Register | | Go |
| 7608h | MCANERR_AGGR_STATUS_SET | MCAN Error Aggregator Status Set Register | | Go |
| 760Ch | MCANERR_AGGR_STATUS_CLR | MCAN Error Aggregator Status Clear Register | | Go |
| 7820h | IIDX | Interrupt Index Register | CPU_INT | Go |
| 7828h | IMASK | Interrupt mask | CPU_INT | Go |
| 7830h | RIS | Raw interrupt status | CPU_INT | Go |
| 7838h | MIS | Masked interrupt status | CPU_INT | Go |
| 7840h | ISET | Interrupt set | CPU_INT | Go |
| 7848h | ICLR | Interrupt clear | CPU_INT | Go |
| 78E0h | EVT_MODE | Event Mode | | Go |
| 78FCh | DESC | Module Description | | Go |
| 7900h | MCANSS_CLKEN | MCAN module clock enable | | Go |
| 7904h | MCANSS_CLKDIV | Clock divider | | Go |
| 7908h | MCANSS_CLKCTL | MCAN-SS clock stop control register | | Go |
| 790Ch | MCANSS_CLKSTS | MCANSS clock stop status register | | Go |

Complex bit access types are encoded to fit into small table cells. Table 32-19 shows the codes that are used for access types in this section.

**Table 32-19. MCAN Access Type Codes**

| Access Type | Code | Description |
|---|---|---|
| Read Type | | |
| R | R | Read |
| R-0 | R<br>-0 | Read<br>Returns 0s |
| RC | R<br>C | Read<br>to Clear |
| RS | R<br>S | Read<br>to Set |
| Write Type | | |



Copyright © 2026 Texas Instruments Incorporated

**Table 32-19. MCAN Access Type Codes (continued)**

| Access Type | Code | Description |
|---|---|---|
| W | W | Write |
| W1C | W<br>1C | Write<br>1 to clear |
| W1S | W<br>1S | Write<br>1 to set |
| W1SQ | W<br>1S<br>Q | Write<br>1 to set<br>Qualified. A condition must be met for this operation to occur. |
| WD | W<br>D | Write<br>Decrement. Decrements the specified bit field by the amount written. |
| WI | W<br>I | Write<br>Increment. Increments the specified bit field by the amount written. |
| WK | W<br>K | Write<br>Write protected by a key |
| WQ | W<br>Q | Write<br>Qualified. A condition must be met for this operation to occur. |
| Reset or Default Value | | |
| *-n* | | Value after reset or the default value |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.1 CANRX (Offset = 6004h) [Reset = 00000000h]

CANRX is shown in Figure 32-27 and described in Table 32-20.

Return to the Summary Table.

CAN RX IO

**Figure 32-27. CANRX**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | GFLT | SLEW | WCOMP | WUEN | INV | HIGHZ1 | HIGHZ0 |
| R-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | DRV | | | HYSTEN | INENA | PIPU | PIPD |
| R-0h | R/W-0h | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| GSTATE | | RESERVED | | | | | |
| R/W-0h | | R-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PSTATE | | RESERVED | | | | | |
| R/W-0h | | R-0h | | | | | |

**Table 32-20. CANRX Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | RESERVED | R | 0h | |
| 30 | GFLT | R/W | 0h | Glitch Filter Enable<br>0h = No internal glitch filter<br>1h = Use internal glitch filter |
| 29 | SLEW | R/W | 0h | Reserved Slew Rate Control<br>0h = No Slew Rate Control<br>1h = Use Slew Rate Control |
| 28 | WCOMP | R/W | 0h | Wake up compare value<br>0h = Match 0 will wake<br>1h = Match 1 will wake |
| 27 | WUEN | R/W | 0h | Wake up enable<br>0h = Wake up not enabled<br>1h = Wake up enabled |
| 26 | INV | R/W | 0h | Invert digital input/output relative to peripheral/GPIO<br>0h = Input and output are non-inverted<br>1h = Input and output are inverted |
| 25 | HIGHZ1 | R/W | 0h | High-Z instead of high output<br>0h = Pin can be driven high<br>1h = Pin is tri-stated instead of driven high |
| 24 | HIGHZ0 | R/W | 0h | High-Z instead of low output<br>0h = Pin can be driven low<br>1h = Pin is tri-stated instead of driven low |
| 23 | RESERVED | R | 0h | |
| 22-20 | DRV | R/W | 0h | Drive strength options<br>0h = Lowest drive strength<br>1h = Drive strength 2/8<br>2h = Drive strength 3/8<br>3h = Drive strength 4/8<br>4h = Drive strength 5/8<br>5h = Drive strength 6/8<br>6h = Drive strength 7/8<br>7h = Highest drive strength |



Copyright © 2026 Texas Instruments Incorporated

**Table 32-20. CANRX Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 19 | HYSTEN | R/W | 0h | Hysteresis enable<br>0h = No hysteresis<br>1h = Hysteresis on |
| 18 | INENA | R/W | 0h | Input enable<br>0h = Inputs 0 to connected core<br>1h = Inputs IO pad value to connected core |
| 17 | PIPU | R/W | 0h | Pull up enable<br>0h = No pull up<br>1h = Pull up |
| 16 | PIPD | R/W | 0h | Pull down enable<br>0h = No pull down<br>1h = Pull down |
| 15-14 | GSTATE | R/W | 0h | GPIO Channel State<br>0h = G-Channel is in Unassigned State<br>1h = G-Channel is in Handover State<br>2h = G-Channel is in Connected State and not Locked (That is F field is allowed to change without going back through Unassigned state)<br>3h = G-Channel is in Connected State and Locked (That is F field is not allowed to change to a different non-Zero value until both G and P channels go to Unassigned) |
| 13-8 | RESERVED | R | 0h | |
| 7-6 | PSTATE | R/W | 0h | Peripheral-Analog Channel State<br>0h = P-Channel is in Unassigned State<br>1h = P-Channel is in Handover State<br>2h = P-Channel is in Connected State and not Locked (That is F field is allowed to change without going back through Unassigned state)<br>3h = P-Channel is in Connected State and Locked (That is F field is not allowed to change to a different non-Zero value until both G and P channels go to Unassigned) |
| 5-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.2 CANTX (Offset = 6008h) [Reset = 00000000h]

CANTX is shown in Figure 32-28 and described in Table 32-21.

Return to the Summary Table.

CAN TX IO

**Figure 32-28. CANTX**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | GFLT | SLEW | WCOMP | WUEN | INV | HIGHZ1 | HIGHZ0 |
| R-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | DRV | | | HYSTEN | INENA | PIPU | PIPD |
| R-0h | R/W-0h | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| GSTATE | | RESERVED | | | | | |
| R/W-0h | | R-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PSTATE | | RESERVED | | | | | |
| R/W-0h | | R-0h | | | | | |

**Table 32-21. CANTX Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | RESERVED | R | 0h | |
| 30 | GFLT | R/W | 0h | Glitch Filter Enable<br>0h = No internal glitch filter<br>1h = Use internal glitch filter |
| 29 | SLEW | R/W | 0h | Reserved Slew Rate Control<br>0h = No Slew Rate Control<br>1h = Use Slew Rate Control |
| 28 | WCOMP | R/W | 0h | Wake up compare value<br>0h = Match 0 will wake<br>1h = Match 1 will wake |
| 27 | WUEN | R/W | 0h | Wake up enable<br>0h = Wake up not enabled<br>1h = Wake up enabled |
| 26 | INV | R/W | 0h | Invert digital input/output relative to peripheral/GPIO<br>0h = Input and output are non-inverted<br>1h = Input and output are inverted |
| 25 | HIGHZ1 | R/W | 0h | High-Z instead of high output<br>0h = Pin can be driven high<br>1h = Pin is tri-stated instead of driven high |
| 24 | HIGHZ0 | R/W | 0h | High-Z instead of low output<br>0h = Pin can be driven low<br>1h = Pin is tri-stated instead of driven low |
| 23 | RESERVED | R | 0h | |
| 22-20 | DRV | R/W | 0h | Drive strength options<br>0h = Lowest drive strength<br>1h = Drive strength 2/8<br>2h = Drive strength 3/8<br>3h = Drive strength 4/8<br>4h = Drive strength 5/8<br>5h = Drive strength 6/8<br>6h = Drive strength 7/8<br>7h = Highest drive strength |



Copyright © 2026 Texas Instruments Incorporated

**Table 32-21. CANTX Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 19 | HYSTEN | R/W | 0h | Hysteresis enable<br>0h = No hysteresis<br>1h = Hysteresis on |
| 18 | INENA | R/W | 0h | Input enable<br>0h = Inputs 0 to connected core<br>1h = Inputs IO pad value to connected core |
| 17 | PIPU | R/W | 0h | Pull up enable<br>0h = No pull up<br>1h = Pull up |
| 16 | PIPD | R/W | 0h | Pull down enable<br>0h = No pull down<br>1h = Pull down |
| 15-14 | GSTATE | R/W | 0h | GPIO Channel State<br>0h = G-Channel is in Unassigned State<br>1h = G-Channel is in Handover State<br>2h = G-Channel is in Connected State and not Locked (That is F field is allowed to change without going back through Unassigned state)<br>3h = G-Channel is in Connected State and Locked (That is F field is not allowed to change to a different non-Zero value until both G and P channels go to Unassigned) |
| 13-8 | RESERVED | R | 0h | |
| 7-6 | PSTATE | R/W | 0h | Peripheral-Analog Channel State<br>0h = P-Channel is in Unassigned State<br>1h = P-Channel is in Handover State<br>2h = P-Channel is in Connected State and not Locked (That is F field is allowed to change without going back through Unassigned state)<br>3h = P-Channel is in Connected State and Locked (That is F field is not allowed to change to a different non-Zero value until both G and P channels go to Unassigned) |
| 5-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.3 CANRX (Offset = 6204h) [Reset = 00000000h]

CANRX is shown in Figure 32-29 and described in Table 32-22.

Return to the Summary Table.

FUPDATE version of CANRX

**Figure 32-29. CANRX**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | IOADDR | | | |
| R-0h | | | | W-0h | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| IOADDR | | | | | | | |
| W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| IOADDR | | | | | | | |
| W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| IOADDR | | | | | | LOCK | GSEL |
| W-0h | | | | | | W-0h | W-0h |

**Table 32-22. CANRX Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-28 | RESERVED | R | 0h | |
| 27-2 | IOADDR | W | 0h | IO Address. This is the address that corresponds to the SOC address[27:2] of the module IP instance specific IO signal in the “Full Write” subregion of the pinmux subregion. |
| 1 | LOCK | W | 0h | Sets lock bit<br>0h = Writing this value has no effect<br>1h = Set channel lock bit |
| 0 | GSEL | W | 0h | GPIO channel Select 0: Select the P-Channel for the F update 1: Select the G-Channel for the F update<br>0h = Select the P-Channel for the F update<br>1h = Select the G-Channel for the F update |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.4 CANTX (Offset = 6208h) [Reset = 00000000h]

CANTX is shown in Figure 32-30 and described in Table 32-23.

Return to the Summary Table.

FUPDATE version of CANTX

**Figure 32-30. CANTX**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | IOADDR | | | |
| R-0h | | | | W-0h | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| IOADDR | | | | | | | |
| W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| IOADDR | | | | | | | |
| W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| IOADDR | | | | | | LOCK | GSEL |
| W-0h | | | | | | W-0h | W-0h |

**Table 32-23. CANTX Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-28 | RESERVED | R | 0h | |
| 27-2 | IOADDR | W | 0h | IO Address. This is the address that corresponds to the SOC address[27:2] of the module IP instance specific IO signal in the “Full Write” subregion of the pinmux subregion. |
| 1 | LOCK | W | 0h | Sets lock bit<br>0h = Writing this value has no effect<br>1h = Set channel lock bit |
| 0 | GSEL | W | 0h | GPIO channel Select 0: Select the P-Channel for the F update 1: Select the G-Channel for the F update<br>0h = Select the P-Channel for the F update<br>1h = Select the G-Channel for the F update |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.5 CPU_CONNECT_0 (Offset = 6480h) [Reset = 00000000h]

CPU_CONNECT_0 is shown in Figure 32-31 and described in Table 32-24.

Return to the Summary Table.

Directly connect peripheral publisher port to application processor

**Figure 32-31. CPU_CONNECT_0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | CPUSS0_CONN | RESERVED |
| R-0h | | | | | | R/W-0h | R-0h |

**Table 32-24. CPU_CONNECT_0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | |
| 1 | CPUSS0_CONN | R/W | 0h | CPUSS0 connect bit.<br>0h = The CPU is not connected.<br>1h = The CPU is connected. |
| 0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.6 PWREN (Offset = 6800h) [Reset = 00000000h]

PWREN is shown in Figure 32-32 and described in Table 32-25.

Return to the Summary Table.

Register to control the power state

**Figure 32-32. PWREN**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | ENABLE |
| R-0h | | | | | | | R/WK-0h |

**Table 32-25. PWREN Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | KEY to allow Power State Change<br>26h = KEY to allow write access to this register |
| 23-1 | RESERVED | R | 0h | |
| 0 | ENABLE | R/WK | 0h | Enable the power<br>KEY must be set to 26h to write to this bit.<br>0h = Disable Power<br>1h = Enable Power |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.7 RSTCTL (Offset = 6804h) [Reset = 00000000h]

RSTCTL is shown in Figure 32-33 and described in Table 32-26.

Return to the Summary Table.

Register to control reset assertion and de-assertion

**Figure 32-33. RSTCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | RESETSTKYCLR | RESETASSERT |
| R-0h | | | | | | WK-0h | WK-0h |

**Table 32-26. RSTCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | Unlock key<br>B1h = KEY to allow write access to this register |
| 23-2 | RESERVED | R | 0h | |
| 1 | RESETSTKYCLR | WK | 0h | Clear the RESETSTKY bit in the STAT register<br>KEY must be set to B1h to write to this bit.<br>0h = Writing 0 has no effect<br>1h = Clear reset sticky bit |
| 0 | RESETASSERT | WK | 0h | Assert reset to the peripheral<br>KEY must be set to B1h to write to this bit.<br>0h = Writing 0 has no effect<br>1h = Assert reset |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.8 STAT (Offset = 6814h) [Reset = 00000000h]

STAT is shown in Figure 32-34 and described in Table 32-27.

Return to the Summary Table.

peripheral enable and reset status

**Figure 32-34. STAT**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | RESETSTKY |
| R-0h | | | | | | | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |

**Table 32-27. STAT Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-17 | RESERVED | R | 0h | |
| 16 | RESETSTKY | R | 0x0 | This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register<br>0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register<br>1h = The peripheral was reset since the last bit clear |
| 15-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.9 MCAN_CREL (Offset = 7000h) [Reset = 32380608h]

MCAN_CREL is shown in Figure 32-35 and described in Table 32-28.

Return to the Summary Table.

MCAN Core Release Register

**Figure 32-35. MCAN_CREL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| REL | | | | STEP | | | | SUBSTEP | | | | YEAR | | | |
| R-3h | | | | R-2h | | | | R-3h | | | | R-8h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MON | | | | | | | | DAY | | | | | | | |
| R-6h | | | | | | | | R-8h | | | | | | | |

**Table 32-28. MCAN_CREL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-28 | REL | R | 3h | Core Release. One digit, BCD-coded. |
| 27-24 | STEP | R | 2h | Step of Core Release. One digit, BCD-coded. |
| 23-20 | SUBSTEP | R | 3h | Sub-Step of Core Release. One digit, BCD-coded. |
| 19-16 | YEAR | R | 8h | Time Stamp Year. One digit, BCD-coded. |
| 15-8 | MON | R | 6h | Time Stamp Month. Two digits, BCD-coded. |
| 7-0 | DAY | R | 8h | Time Stamp Day. Two digits, BCD-coded. |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.10 MCAN_ENDN (Offset = 7004h) [Reset = 87654321h]

MCAN_ENDN is shown in Figure 32-36 and described in Table 32-29.

Return to the Summary Table.

MCAN Endian Register

**Figure 32-36. MCAN_ENDN**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ETV | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-87654321h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 32-29. MCAN_ENDN Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | ETV | R | 87654321h | Endianess Test Value. Reading the constant value maintained in this register allows software to determine the endianess of the host CPU. |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.11 MCAN_DBTP (Offset = 700Ch) [Reset = 00000A33h]

MCAN_DBTP is shown in Figure 32-37 and described in Table 32-30.

Return to the Summary Table.

This register is only writable if bits CCCR.CCE and CCCR.INIT are set. The CAN bit time may be programed in the range of 4 to 49 time quanta. The CAN time quantum may be programmed in the range of 1 to 32 m_can_cclk periods. tq = (DBRP + 1) mtq.
DTSEG1 is the sum of Prop_Seg and Phase_Seg1. DTSEG2 is Phase_Seg2.
Therefore the length of the bit time is (programmed values) (DTSEG1 + DTSEG2 + 3) tq or (functional values) (Sync_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2) tq.
The Information Processing Time (IPT) is zero, meaning the data for the next bit is available at the first clock edge after the sample point.

**Figure 32-37. MCAN_DBTP**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TDC | RESERVED | | DBRP | | | | |
| R/WQ-0h | R-0h | | R/WQ-0h | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | DTSEG1 | | | | |
| R-0h | | | R/WQ-Ah | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DTSEG2 | | | | DSJW | | | |
| R/WQ-3h | | | | R/WQ-3h | | | |

**Table 32-30. MCAN_DBTP Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | RESERVED | R | 0h | |
| 23 | TDC | R/WQ | 0h | Transmitter Delay Compensation<br>0 Transmitter Delay Compensation disabled<br>1 Transmitter Delay Compensation enabled |
| 22-21 | RESERVED | R | 0h | |
| 20-16 | DBRP | R/WQ | 0h | Data Bit Rate Prescaler. The value by which the oscillator frequency is divided for generating the bit time quanta. The bit time is built up from a multiple of this quanta. Valid values for the Bit Rate Prescaler are 0 to 31. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.<br>Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. |
| 15-13 | RESERVED | R | 0h | |
| 12-8 | DTSEG1 | R/WQ | Ah | Data Time Segment Before Sample Point. Valid values are 0 to 31. The actual interpretation by the hardware of this value is such that one more than the programmed value is used.<br>Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. |
| 7-4 | DTSEG2 | R/WQ | 3h | Data Time Segment After Sample Point. Valid values are 0 to 15. The actual interpretation by the hardware of this value is such that one more than the programmed value is used.<br>Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. |



Copyright © 2026 Texas Instruments Incorporated

**Table 32-30. MCAN_DBTP Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 3-0 | DSJW | R/WQ | 3h | Data Resynchronization Jump Width. Valid values are 0 to 15. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.<br>Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.12 MCAN_TEST (Offset = 7010h) [Reset = 000000X0h]

MCAN_TEST is shown in Figure 32-38 and described in Table 32-31.

Return to the Summary Table.

Write access to the Test Register has to be enabled by setting bit CCCR.TEST to '1'. All Test Register functions are set to their reset values when bit CCCR.TEST is reset.
Loop Back Mode and software control of the internal CAN TX pin are hardware test modes. Programming of TX ? '00' may disturb the message transfer on the CAN bus.

**Figure 32-38. MCAN_TEST**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RX | TX | | LBCK | RESERVED | | | |
| R-X | R/WQ-0h | | R/WQ-0h | R-0h | | | |

**Table 32-31. MCAN_TEST Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7 | RX | R | X | Receive Pin. Monitors the actual value of the CAN receive pin.<br>0h = DOMINANT : The CAN bus is dominant (CAN RX pin = '0')<br>1h = RECESSIVE : The CAN bus is recessive (CAN RX pin = '1') |
| 6-5 | TX | R/WQ | 0h | Control of Transmit Pin<br>00 CAN TX pin controlled by the CAN Core, updated at the end of the CAN bit time<br>01 Sample Point can be monitored at CAN TX pin<br>10 Dominant ('0') level at CAN TX pin<br>11 Recessive ('1') at CAN TX pin<br>Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. |
| 4 | LBCK | R/WQ | 0h | Loop Back Mode. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.<br>0h = DISABLE : Reset value, Loop Back Mode is disabled<br>1h = ENABLE : Loop Back Mode is enabled |
| 3-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.13 MCAN_RWD (Offset = 7014h) [Reset = 00000000h]

MCAN_RWD is shown in Figure 32-39 and described in Table 32-32.

Return to the Summary Table.

MCAN RAM Watchdog

**Figure 32-39. MCAN_RWD**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | WDV | | | | | | | | WDC | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | R-0h | | | | | | | | R/WQ-0h | | | | | | | |

**Table 32-32. MCAN_RWD Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15-8 | WDV | R | 0h | Watchdog Value. Actual Message RAM Watchdog Counter Value. The RAM Watchdog monitors the READY output of the Message RAM. A Message RAM access via the MCAN's Generic Master Interface starts the Message RAM Watchdog Counter with the value configured by the WDC field. The counter is reloaded with WDC when the Message RAM signals successful completion by activating its READY output. In case there is no response from the Message RAM until the counter has counted down to zero, the counter stops and interrupt flag MCAN_IR.WDI is set. The RAM Watchdog Counter is clocked by the host (system) clock. |
| 7-0 | WDC | R/WQ | 0h | Watchdog Configuration. Start value of the Message RAM Watchdog Counter. With the reset value of '00' the counter is disabled. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.14 MCAN_CCCR (Offset = 7018h) [Reset = 00000001h]

MCAN_CCCR is shown in Figure 32-40 and described in Table 32-33.

Return to the Summary Table.

MCAN CC Control Register

**Figure 32-40. MCAN_CCCR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| NISO | TXP | EFBI | PXHD | RESERVED | | BRSE | FDOE |
| R/WQ-0h | R/WQ-0h | R/WQ-0h | R/WQ-0h | R-0h | | R/WQ-0h | R/WQ-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TEST | DAR | MON | CSR | CSA | ASM | CCE | INIT |
| R/W1SQ-0h | R/WQ-0h | R/W1SQ-0h | R/W-0h | R-0h | R/W1SQ-0h | R/WQ-0h | R/W-1h |

**Table 32-33. MCAN_CCCR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15 | NISO | R/WQ | 0h | Non ISO Operation. If this bit is set, the MCAN uses the CAN FD frame format as specified by the Bosch CAN FD Specification V1.0.<br>0 CAN FD frame format according to ISO 11898-1:2015<br>1 CAN FD frame format according to Bosch CAN FD Specification V1.0 |
| 14 | TXP | R/WQ | 0h | Transmit Pause. If this bit is set, the MCAN pauses for two CAN bit times before starting the next transmission after itself has successfully transmitted a frame.<br>0 Transmit pause disabled<br>1 Transmit pause enabled<br>Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. |
| 13 | EFBI | R/WQ | 0h | Edge Filtering during Bus Integration<br>0 Edge filtering disabled<br>1 Two consecutive dominant tq required to detect an edge for hard synchronization<br>Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. |
| 12 | PXHD | R/WQ | 0h | Protocol Exception Handling Disable<br>0 Protocol exception handling enabled<br>1 Protocol exception handling disabled<br>Note: When protocol exception handling is disabled, the MCAN will transmit an error frame when it detects a protocol exception condition.<br>Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. |
| 11-10 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

**Table 32-33. MCAN_CCCR Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 9 | BRSE | R/WQ | 0h | Bit Rate Switch Enable<br>0 Bit rate switching for transmissions disabled<br>1 Bit rate switching for transmissions enabled<br>Note: When CAN FD operation is disabled FDOE = '0', BRSE is not evaluated.<br>Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. |
| 8 | FDOE | R/WQ | 0h | Flexible Datarate Operation Enable<br>0 FD operation disabled<br>1 FD operation enabled<br>Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. |
| 7 | TEST | R/W1SQ | 0h | Test Mode Enable<br>0 Normal operation, register TEST holds reset values<br>1 Test Mode, write access to register TEST enabled<br>Qualified Write 1 to Set is possible only with CCCR.CCE='1' and CCCR.INIT='1'. |
| 6 | DAR | R/WQ | 0h | Disable Automatic Retransmission<br>0 Automatic retransmission of messages not transmitted successfully enabled<br>1 Automatic retransmission disabled<br>Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. |
| 5 | MON | R/W1SQ | 0h | Bus Monitoring Mode. Bit MON can only be set by SW when both CCE and INIT are set to '1'. The bit can be reset by SW at any time.<br>0 Bus Monitoring Mode is disabled<br>1 Bus Monitoring Mode is enabled<br>Qualified Write 1 to Set is possible only with CCCR.CCE='1' and CCCR.INIT='1'. |
| 4 | CSR | R/W | 0h | Clock Stop Request<br>0 No clock stop is requested<br>1 Clock stop requested. When clock stop is requested, first INIT and then CSA will be set after all pending transfer requests have been completed and the CAN bus reached idle. |
| 3 | CSA | R | 0h | Clock Stop Acknowledge<br>0 No clock stop acknowledged<br>1 MCAN may be set in power down by stopping the Host and CAN clocks |
| 2 | ASM | R/W1SQ | 0h | Restricted Operation Mode. Bit ASM can only be set by SW when both CCE and INIT are set to '1'. The bit can be reset by SW at any time.<br>0 Normal CAN operation<br>1 Restricted Operation Mode active<br>Qualified Write 1 to Set is possible only with CCCR.CCE='1' and CCCR.INIT='1'. |
| 1 | CCE | R/WQ | 0h | Configuration Change Enable<br>0 The CPU has no write access to the protected configuration registers<br>1 The CPU has write access to the protected configuration registers (while CCCR.INIT = '1')<br>Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. |
| 0 | INIT | R/W | 1h | Initialization<br>0 Normal Operation<br>1 Initialization is started<br>Note: Due to the synchronization mechanism between the two clock domains, there may be a delay until the value written to INIT can be read back. Therefore the programmer has to assure that the previous value written to INIT has been accepted by reading INIT before setting INIT to a new value. |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.15 MCAN_NBTP (Offset = 701Ch) [Reset = 06000A03h]

MCAN_NBTP is shown in Figure 32-41 and described in Table 32-34.

Return to the Summary Table.

This register is only writable if bits CCCR.CCE and CCCR.INIT are set. The CAN bit time may be programed in the range of 4 to 385 time quanta. The CAN time quantum may be programmed in the range of 1 to 512 m_can_cclk periods. tq = (NBRP + 1) mtq.
NTSEG1 is the sum of Prop_Seg and Phase_Seg1. NTSEG2 is Phase_Seg2.
Therefore the length of the bit time is (programmed values) (NTSEG1 + NTSEG2 + 3) tq or (functional values) (Sync_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2) tq.
The Information Processing Time (IPT) is zero, meaning the data for the next bit is available at the first clock edge after the sample point.
Note: With a CAN clock of 8MHz, the reset value of 0x06000A03 configures the MCAN for a bit rate of 500 kbit/s.

**Figure 32-41. MCAN_NBTP**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| NSJW | | | | | | | NBRP |
| R/WQ-3h | | | | | | | R/WQ-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| NBRP | | | | | | | |
| R/WQ-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| NTSEG1 | | | | | | | |
| R/WQ-Ah | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | NTSEG2 | | | | | | |
| R-0h | R/WQ-3h | | | | | | |

**Table 32-34. MCAN_NBTP Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-25 | NSJW | R/WQ | 3h | Nominal (Re)Synchronization Jump Width. Valid values are 0 to 127. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.<br>Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. |
| 24-16 | NBRP | R/WQ | 0h | Nominal Bit Rate Prescaler. The value by which the oscillator frequency is divided for generating the bit time quanta. The bit time is built up from a multiple of this quanta. Valid values for the Bit Rate Prescaler are 0 to 511. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.<br>Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. |
| 15-8 | NTSEG1 | R/WQ | Ah | Nominal Time Segment Before Sample Point. Valid values are 1 to 255. The actual interpretation by the hardware of this value is such that one more than the programmed value is used.<br>Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. |
| 7 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

**Table 32-34. MCAN_NBTP Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 6-0 | NTSEG2 | R/WQ | 3h | Nominal Time Segment After Sample Point. Valid values are 1 to 127. The actual interpretation by the hardware of this value is such that one more than the programmed value is used.<br>Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.16 MCAN_TSCC (Offset = 7020h) [Reset = 00000000h]

MCAN_TSCC is shown in Figure 32-42 and described in Table 32-35.

Return to the Summary Table.

MCAN Timestamp Counter Configuration

**Figure 32-42. MCAN_TSCC**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | TCP | | | |
| R-0h | | | | | | | | | | | | R/WQ-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | | | TSS | |
| R-0h | | | | | | | | | | | | | | R/WQ-0h | |

**Table 32-35. MCAN_TSCC Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | RESERVED | R | 0h | |
| 19-16 | TCP | R/WQ | 0h | Timestamp Counter Prescaler. Configures the timestamp and timeout counters time unit in multiples of CAN bit times. Valid values are 0 to 15. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.<br>Note: With CAN FD an external counter is required for timestamp generation (TSS = '10').<br>Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. |
| 15-2 | RESERVED | R | 0h | |
| 1-0 | TSS | R/WQ | 0h | Timestamp Select<br>00 Timestamp counter value always 0x0000<br>01 Timestamp counter value incremented according to TCP<br>10 External timestamp counter value used<br>11 Same as '00'<br>Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.17 MCAN_TSCV (Offset = 7024h) [Reset = 00000000h]

MCAN_TSCV is shown in Figure 32-43 and described in Table 32-36.

Return to the Summary Table.

MCAN Timestamp Counter Value

**Figure 32-43. MCAN_TSCV**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | TSC | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | R/W-0h | | | | | | | | | | | | | | | |

**Table 32-36. MCAN_TSCV Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15-0 | TSC | R/W | 0h | Timestamp Counter. The internal/external Timestamp Counter value is captured on start of frame (both Rx and Tx). When TSCC.TSS = '01', the Timestamp Counter is incremented in multiples of CAN bit times, (1...16), depending on the configuration of TSCC.TCP. A wrap around sets interrupt flag IR.TSW. Write access resets the counter to zero. When TSCC.TSS = '10', TSC reflects the External Timestamp Counter value, and a write access has no impact.<br>Note: A 'wrap around' is a change of the Timestamp Counter value from non-zero to zero not<br>caused by write access to MCAN_TSCV. |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.18 MCAN_TOCC (Offset = 7028h) [Reset = FFFF0000h]

MCAN_TOCC is shown in Figure 32-44 and described in Table 32-37.

Return to the Summary Table.

MCAN Timeout Counter Configuration

**Figure 32-44. MCAN_TOCC**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| TOP | | | | | | | |
| R/WQ-FFFFh | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TOP | | | | | | | |
| R/WQ-FFFFh | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | TOS | | ETOC |
| R-0h | | | | | R/WQ-0h | | R/WQ-0h |

**Table 32-37. MCAN_TOCC Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | TOP | R/WQ | FFFFh | Timeout Period. Start value of the Timeout Counter (down-counter). Configures the Timeout Period.<br>Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. |
| 15-3 | RESERVED | R | 0h | |
| 2-1 | TOS | R/WQ | 0h | Timeout Select. When operating in Continuous mode, a write to TOCV presets the counter to the value configured by TOCC.TOP and continues down-counting. When the Timeout Counter is controlled by one of the FIFOs, an empty FIFO presets the counter to the value configured by TOCC.TOP. Down-counting is started when the first FIFO element is stored.<br>00 Continuous operation<br>01 Timeout controlled by Tx Event FIFO<br>10 Timeout controlled by Rx FIFO 0<br>11 Timeout controlled by Rx FIFO 1<br>Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. |
| 0 | ETOC | R/WQ | 0h | Enable Timeout Counter<br>0 Timeout Counter disabled<br>1 Timeout Counter enabled<br>Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.19 MCAN_TOCV (Offset = 702Ch) [Reset = 0000FFFFh]

MCAN_TOCV is shown in Figure 32-45 and described in Table 32-38.

Return to the Summary Table.

MCAN Timeout Counter Value

**Figure 32-45. MCAN_TOCV**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | TOC | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | R/W-FFFFh | | | | | | | | | | | | | | | |

**Table 32-38. MCAN_TOCV Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15-0 | TOC | R/W | FFFFh | Timeout Counter. The Timeout Counter is decremented in multiples of CAN bit times, (1...16), depending on the configuration of TSCC.TCP. When decremented to zero, interrupt flag IR.TOO is set and the Timeout Counter is stopped. Start and reset/restart conditions are configured via TOCC.TOS. |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.20 MCAN_ECR (Offset = 7040h) [Reset = 00000000h]

MCAN_ECR is shown in Figure 32-46 and described in Table 32-39.

Return to the Summary Table.

MCAN Error Counter Register

**Figure 32-46. MCAN_ECR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | CEL | | | | | | | |
| R-0h | | | | | | | | RC-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RP | REC | | | | | | | TEC | | | | | | | |
| R-0h | R-0h | | | | | | | R-0h | | | | | | | |

**Table 32-39. MCAN_ECR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | RESERVED | R | 0h | |
| 23-16 | CEL | RC | 0h | CAN Error Logging. The counter is incremented each time when a CAN protocol error causes the Transmit Error Counter or the Receive Error Counter to be incremented. It is reset by read access to CEL. The counter stops at 0xFF; the next increment of TEC or REC sets interrupt flag IR.ELO.<br>Note: When CCCR.ASM is set, the CAN protocol controller does not increment TEC and REC when a CAN protocol error is detected, but CEL is still incremented. |
| 15 | RP | R | 0h | Receive Error Passive<br>0 The Receive Error Counter is below the error passive level of 128<br>1 The Receive Error Counter has reached the error passive level of 128 |
| 14-8 | REC | R | 0h | Receive Error Counter. Actual state of the Receive Error Counter, values between 0 and 127.<br>Note: When CCCR.ASM is set, the CAN protocol controller does not increment TEC and REC when a CAN protocol error is detected, but CEL is still incremented. |
| 7-0 | TEC | R | 0h | Transmit Error Counter. Actual state of the Transmit Error Counter, values between 0 and 255.<br>Note: When CCCR.ASM is set, the CAN protocol controller does not increment TEC and REC when a CAN protocol error is detected, but CEL is still incremented. |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.21 MCAN_PSR (Offset = 7044h) [Reset = 00000707h]

MCAN_PSR is shown in Figure 32-47 and described in Table 32-40.

Return to the Summary Table.

MCAN Protocol Status Register

**Figure 32-47. MCAN_PSR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | TDCV | | | | | | |
| R-0h | R-0h | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | PXE | RFDF | RBRS | RESI | DLEC | | |
| R-0h | RC-0h | RC-0h | RC-0h | RC-0h | RS-7h | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BO | EW | EP | ACT | | LEC | | |
| R-0h | R-0h | R-0h | R-0h | | RS-7h | | |

**Table 32-40. MCAN_PSR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-23 | RESERVED | R | 0h | |
| 22-16 | TDCV | R | 0h | Transmitter Delay Compensation Value. Position of the secondary sample point, defined by the sum of the measured delay from the internal CAN TX signal to the internal CAN RX signal and TDCR.TDCO. The SSP position is, in the data phase, the number of mtq between the start of the transmitted bit and the secondary sample point. Valid values are 0 to 127 mtq. |
| 15 | RESERVED | R | 0h | |
| 14 | PXE | RC | 0h | Protocol Exception Event<br>0 No protocol exception event occurred since last read access<br>1 Protocol exception event occurred |
| 13 | RFDF | RC | 0h | Received a CAN FD Message. This bit is set independent of acceptance filtering.<br>0 Since this bit was reset by the CPU, no CAN FD message has been received<br>1 Message in CAN FD format with FDF flag set has been received |
| 12 | RBRS | RC | 0h | BRS Flag of Last Received CAN FD Message. This bit is set together with RFDF, independent of acceptance filtering.<br>0 Last received CAN FD message did not have its BRS flag set<br>1 Last received CAN FD message had its BRS flag set |
| 11 | RESI | RC | 0h | ESI Flag of Last Received CAN FD Message. This bit is set together with RFDF, independent of acceptance filtering.<br>0 Last received CAN FD message did not have its ESI flag set<br>1 Last received CAN FD message had its ESI flag set |
| 10-8 | DLEC | RS | 7h | Data Phase Last Error Code. Type of last error that occurred in the data phase of a CAN FD format frame with its BRS flag set. Coding is the same as for LEC. This field will be cleared to zero when a CAN FD format frame with its BRS flag set has been transferred (reception or transmission) without error. |
| 7 | BO | R | 0h | Bus_Off Status<br>0 The M_CAN is not Bus_Off<br>1 The M_CAN is in Bus_Off state |



Copyright © 2026 Texas Instruments Incorporated

**Table 32-40. MCAN_PSR Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 6 | EW | R | 0h | Warning Status<br>0 Both error counters are below the Error_Warning limit of 96<br>1 At least one of error counter has reached the Error_Warning limit of 96 |
| 5 | EP | R | 0h | Error Passive<br>0 The M_CAN is in the Error_Active state. It normally takes part in bus communication and sends an active error flag when an error has been detected<br>1 The M_CAN is in the Error_Passive state |
| 4-3 | ACT | R | 0h | Node Activity. Monitors the module's CAN communication state.<br>00 Synchronizing - node is synchronizing on CAN communication<br>01 Idle - node is neither receiver nor transmitter<br>10 Receiver - node is operating as receiver<br>11 Transmitter - node is operating as transmitter<br>Note: ACT is set to '00' by a Protocol Exception Event. |
| 2-0 | LEC | RS | 7h | Last Error Code. The LEC indicates the type of the last error to occur on the CAN bus. This field will be cleared to '0' when a message has been transferred (reception or transmission) without error.<br>0 No Error: No error occurred since LEC has been reset by successful reception or transmission.<br>1 Stuff Error: More than 5 equal bits in a sequence have occurred in a part of a received message where this is not allowed.<br>2 Form Error: A fixed format part of a received frame has the wrong format.<br>3 AckError: The message transmitted by the MCAN was not acknowledged by another node.<br>4 Bit1Error: During the transmission of a message (with the exception of the arbitration field), the device wanted to send a recessive level (bit of logical value '1'), but the monitored bus value was dominant.<br>5 Bit0Error: During the transmission of a message (or acknowledge bit, or active error flag, or overload flag), the device wanted to send a dominant level (data or identifier bit logical value '0'), but the monitored bus value was recessive. During Bus_Off recovery this status is set each time a sequence of 11 recessive bits has been monitored. This enables the CPU to monitor the proceeding of the Bus_Off recovery sequence (indicating the bus is not stuck at dominant or continuously disturbed).<br>6 CRCError: The CRC check sum of a received message was incorrect. The CRC of an incoming message does not match with the CRC calculated from the received data.<br>7 NoChange: Any read access to the Protocol Status Register re-initializes the LEC to '7'. When the LEC shows the value '7', no CAN bus event was detected since the last CPU read access to the Protocol Status Register.<br>Note: When a frame in CAN FD format has reached the data phase with BRS flag set, the next CAN event (error or valid frame) will be shown in DLEC instead of LEC. An error in a fixed stuff bit of a CAN FD CRC sequence will be shown as a Form Error, not Stuff Error.<br>Note: The Bus_Off recovery sequence (see ISO 11898-1:2015) cannot be shortened by setting or resetting CCCR.INIT. If the device goes Bus_Off, it will set CCCR.INIT of its own accord, stopping all bus activities. Once CCCR.INIT has been cleared by the CPU, the device will then wait for 129 occurrences of Bus Idle (129 * 11 consecutive recessive bits) before resuming normal operation. At the end of the Bus_Off recovery sequence, the Error Management Counters will be reset. During the waiting time after the resetting of CCCR.INIT, each time a sequence of 11 recessive bits has been monitored, a Bit0Error code is written to PSR.LEC, enabling the CPU to readily check up whether the CAN bus is stuck at dominant or continuously disturbed and to monitor the Bus_Off recovery sequence. ECR.REC is used to count these sequences. |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.22 MCAN_TDCR (Offset = 7048h) [Reset = 00000000h]

MCAN_TDCR is shown in Figure 32-48 and described in Table 32-41.

Return to the Summary Table.

MCAN Transmitter Delay Compensation Register

**Figure 32-48. MCAN_TDCR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | TDCO | | | | | | |
| R-0h | R/WQ-0h | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | TDCF | | | | | | |
| R-0h | R/WQ-0h | | | | | | |

**Table 32-41. MCAN_TDCR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-15 | RESERVED | R | 0h | |
| 14-8 | TDCO | R/WQ | 0h | Transmitter Delay Compensation Offset. Offset value defining the distance between the measured delay from the internal CAN TX signal to the internal CAN RX signal and the secondary sample point. Valid values are 0 to 127 mtq.<br>Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. |
| 7 | RESERVED | R | 0h | |
| 6-0 | TDCF | R/WQ | 0h | Transmitter Delay Compensation Filter Window Length. Defines the minimum value for the SSP position, dominant edges on the internal CAN RX signal that would result in an earlier SSP position are ignored for transmitter delay measurement. The feature is enabled when TDCF is configured to a value greater than TDCO. Valid values are 0 to 127 mtq.<br>Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.23 MCAN_IR (Offset = 7050h) [Reset = 80000000h]

MCAN_IR is shown in Figure 32-49 and described in Table 32-42.

Return to the Summary Table.

The flags are set when one of the listed conditions is detected (edge-sensitive). The flags remain set until the Host clears them. A flag is cleared by writing a '1' to the corresponding bit position. Writing a '0' has no effect. A hard reset will clear the register. The configuration of IE controls whether an interrupt is generated. The configuration of ILS controls on which interrupt line an interrupt is signaled.

**Figure 32-49. MCAN_IR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | ARA | PED | PEA | WDI | BO | EW |
| R-0h | | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| EP | ELO | BEU | RESERVED | DRX | TOO | MRAF | TSW |
| R/W1C-0h | R/W1C-0h | R/W1C-0h | R-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TEFL | TEFF | TEFW | TEFN | TFE | TCF | TC | HPM |
| R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RF1L | RF1F | RF1W | RF1N | RF0L | RF0F | RF0W | RF0N |
| R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h |

**Table 32-42. MCAN_IR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | RESERVED | R | 0h | |
| 29 | ARA | R/W1C | 0h | Access to Reserved Address<br>0 No access to reserved address occurred<br>1 Access to reserved address occurred |
| 28 | PED | R/W1C | 0h | Protocol Error in Data Phase (Data Bit Time is used)<br>0 No protocol error in data phase<br>1 Protocol error in data phase detected (PSR.DLEC ? 0,7) |
| 27 | PEA | R/W1C | 0h | Protocol Error in Arbitration Phase (Nominal Bit Time is used)<br>0 No protocol error in arbitration phase<br>1 Protocol error in arbitration phase detected (PSR.LEC ? 0,7) |
| 26 | WDI | R/W1C | 0h | Watchdog Interrupt<br>0 No Message RAM Watchdog event occurred<br>1 Message RAM Watchdog event due to missing READY |
| 25 | BO | R/W1C | 0h | Bus_Off Status<br>0 Bus_Off status unchanged<br>1 Bus_Off status changed |
| 24 | EW | R/W1C | 0h | Warning Status<br>0 Error_Warning status unchanged<br>1 Error_Warning status changed |
| 23 | EP | R/W1C | 0h | Error Passive<br>0 Error_Passive status unchanged<br>1 Error_Passive status changed |
| 22 | ELO | R/W1C | 0h | Error Logging Overflow<br>0 CAN Error Logging Counter did not overflow<br>1 Overflow of CAN Error Logging Counter occurred |



Copyright © 2026 Texas Instruments Incorporated

**Table 32-42. MCAN_IR Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 21 | BEU | R/W1C | 0h | Bit Error Uncorrected. Message RAM bit error detected, uncorrected. This bit is set when a double bit error is detected by the ECC aggregator attached to the Message RAM. An uncorrected Message RAM bit error sets CCCR.INIT to '1'. This is done to avoid transmission of corrupted data.<br>0 No bit error detected when reading from Message RAM<br>1 Bit error detected, uncorrected (parity logic) |
| 20 | RESERVED | R | 0h | |
| 19 | DRX | R/W1C | 0h | Message Stored to Dedicated Rx Buffer. The flag is set whenever a received message has been stored into a dedicated Rx Buffer.<br>0 No Rx Buffer updated<br>1 At least one received message stored into an Rx Buffer |
| 18 | TOO | R/W1C | 0h | Timeout Occurred<br>0 No timeout<br>1 Timeout reached |
| 17 | MRAF | R/W1C | 0h | Message RAM Access Failure. The flag is set, when the Rx Handler:<br>- has not completed acceptance filtering or storage of an accepted message until the arbitration field of the following message has been received. In this case acceptance filtering or message storage is aborted and the Rx Handler starts processing of the following message.<br>- was not able to write a message to the Message RAM. In this case message storage is aborted.<br>In both cases the FIFO put index is not updated resp. the New Data flag for a dedicated Rx Buffer is not set, a partly stored message is overwritten when the next message is stored to this location.<br>The flag is also set when the Tx Handler was not able to read a message from the Message RAM in time. In this case message transmission is aborted. In case of a Tx Handler access failure the MCAN is switched into Restricted Operation Mode. To leave Restricted Operation Mode, the Host CPU has to reset CCCR.ASM.<br>0 No Message RAM access failure occurred<br>1 Message RAM access failure occurred |
| 16 | TSW | R/W1C | 0h | Timestamp Wraparound<br>0 No timestamp counter wrap-around<br>1 Timestamp counter wrapped around |
| 15 | TEFL | R/W1C | 0h | Tx Event FIFO Element Lost<br>0 No Tx Event FIFO element lost<br>1 Tx Event FIFO element lost, also set after write attempt to Tx Event FIFO of size zero |
| 14 | TEFF | R/W1C | 0h | Tx Event FIFO Full<br>0 Tx Event FIFO not full<br>1 Tx Event FIFO full |
| 13 | TEFW | R/W1C | 0h | Tx Event FIFO Watermark Reached<br>0 Tx Event FIFO fill level below watermark<br>1 Tx Event FIFO fill level reached watermark |
| 12 | TEFN | R/W1C | 0h | Tx Event FIFO New Entry<br>0 Tx Event FIFO unchanged<br>1 Tx Handler wrote Tx Event FIFO element |
| 11 | TFE | R/W1C | 0h | Tx FIFO Empty<br>0 Tx FIFO non-empty<br>1 Tx FIFO empty |
| 10 | TCF | R/W1C | 0h | Transmission Cancellation Finished<br>0 No transmission cancellation finished<br>1 Transmission cancellation finished |
| 9 | TC | R/W1C | 0h | Transmission Completed<br>0 No transmission completed<br>1 Transmission completed |



Copyright © 2026 Texas Instruments Incorporated

**Table 32-42. MCAN_IR Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 8 | HPM | R/W1C | 0h | High Priority Message<br>0 No high priority message received<br>1 High priority message received |
| 7 | RF1L | R/W1C | 0h | Rx FIFO 1 Message Lost<br>0 No Rx FIFO 1 message lost<br>1 Rx FIFO 1 message lost, also set after write attempt to Rx FIFO 1 of size zero |
| 6 | RF1F | R/W1C | 0h | Rx FIFO 1 Full<br>0 Rx FIFO 1 not full<br>1 Rx FIFO 1 full |
| 5 | RF1W | R/W1C | 0h | Rx FIFO 1 Watermark Reached<br>0 Rx FIFO 1 fill level below watermark<br>1 Rx FIFO 1 fill level reached watermark |
| 4 | RF1N | R/W1C | 0h | Rx FIFO 1 New Message<br>0 No new message written to Rx FIFO 1<br>1 New message written to Rx FIFO 1 |
| 3 | RF0L | R/W1C | 0h | Rx FIFO 0 Message Lost<br>0 No Rx FIFO 0 message lost<br>1 Rx FIFO 0 message lost, also set after write attempt to Rx FIFO 0 of size zero |
| 2 | RF0F | R/W1C | 0h | Rx FIFO 0 Full<br>0 Rx FIFO 0 not full<br>1 Rx FIFO 0 full |
| 1 | RF0W | R/W1C | 0h | Rx FIFO 0 Watermark Reached<br>0 Rx FIFO 0 fill level below watermark<br>1 Rx FIFO 0 fill level reached watermark |
| 0 | RF0N | R/W1C | 0h | Rx FIFO 0 New Message<br>0 No new message written to Rx FIFO 0<br>1 New message written to Rx FIFO 0 |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.24 MCAN_IE (Offset = 7054h) [Reset = 00000000h]

MCAN_IE is shown in Figure 32-50 and described in Table 32-43.

Return to the Summary Table.

MCAN Interrupt Enable

**Figure 32-50. MCAN_IE**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | ARAE | PEDE | PEAE | WDIE | BOE | EWE |
| R-0h | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| EPE | ELOE | BEUE | BECE | DRXE | TOOE | MRAFE | TSWE |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TEFLE | TEFFE | TEFWE | TEFNE | TFEE | TCFE | TCE | HPME |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RF1LE | RF1FE | RF1WE | RF1NE | RF0LE | RF0FE | RF0WE | RF0NE |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

**Table 32-43. MCAN_IE Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | RESERVED | R | 0h | |
| 29 | ARAE | R/W | 0h | Access to Reserved Address Enable |
| 28 | PEDE | R/W | 0h | Protocol Error in Data Phase Enable |
| 27 | PEAE | R/W | 0h | Protocol Error in Arbitration Phase Enable |
| 26 | WDIE | R/W | 0h | Watchdog Interrupt Enable |
| 25 | BOE | R/W | 0h | Bus_Off Status Enable |
| 24 | EWE | R/W | 0h | Warning Status Enable |
| 23 | EPE | R/W | 0h | Error Passive Enable |
| 22 | ELOE | R/W | 0h | Error Logging Overflow Enable |
| 21 | BEUE | R/W | 0h | Bit Error Uncorrected Enable |
| 20 | BECE | R/W | 0h | Bit Error Corrected Enable<br>A separate interrupt line reserved for corrected bit errors is provided via the MCAN_ERROR_REGS. It advised for the user to use these registers and leave this bit cleared to '0'. |
| 19 | DRXE | R/W | 0h | Message Stored to Dedicated Rx Buffer Enable |
| 18 | TOOE | R/W | 0h | Timeout Occurred Enable |
| 17 | MRAFE | R/W | 0h | Message RAM Access Failure Enable |
| 16 | TSWE | R/W | 0h | Timestamp Wraparound Enable |
| 15 | TEFLE | R/W | 0h | Tx Event FIFO Element Lost Enable |
| 14 | TEFFE | R/W | 0h | Tx Event FIFO Full Enable |
| 13 | TEFWE | R/W | 0h | Tx Event FIFO Watermark Reached Enable |
| 12 | TEFNE | R/W | 0h | Tx Event FIFO New Entry Enable |
| 11 | TFEE | R/W | 0h | Tx FIFO Empty Enable |
| 10 | TCFE | R/W | 0h | Transmission Cancellation Finished Enable |
| 9 | TCE | R/W | 0h | Transmission Completed Enable |
| 8 | HPME | R/W | 0h | High Priority Message Enable |



Copyright © 2026 Texas Instruments Incorporated

**Table 32-43. MCAN_IE Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7 | RF1LE | R/W | 0h | Rx FIFO 1 Message Lost Enable |
| 6 | RF1FE | R/W | 0h | Rx FIFO 1 Full Enable |
| 5 | RF1WE | R/W | 0h | Rx FIFO 1 Watermark Reached Enable |
| 4 | RF1NE | R/W | 0h | Rx FIFO 1 New Message Enable |
| 3 | RF0LE | R/W | 0h | Rx FIFO 0 Message Lost Enable |
| 2 | RF0FE | R/W | 0h | Rx FIFO 0 Full Enable |
| 1 | RF0WE | R/W | 0h | Rx FIFO 0 Watermark Reached Enable |
| 0 | RF0NE | R/W | 0h | Rx FIFO 0 New Message Enable |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.25 MCAN_ILS (Offset = 7058h) [Reset = 00000000h]

MCAN_ILS is shown in Figure 32-51 and described in Table 32-44.

Return to the Summary Table.

The Interrupt Line Select register assigns an interrupt generated by a specific interrupt flag from the Interrupt Register to one of the two module interrupt lines. For interrupt generation the respective interrupt line has to be enabled via ILE.EINT0 and ILE.EINT1.

**Figure 32-51. MCAN_ILS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | ARAL | PEDL | PEAL | WDIL | BOL | EWL |
| R-0h | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| EPL | ELOL | BEUL | BECL | DRXL | TOOL | MRAFL | TSWL |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TEFLL | TEFFL | TEFWL | TEFNL | TFEL | TCFL | TCL | HPML |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RF1LL | RF1FL | RF1WL | RF1NL | RF0LL | RF0FL | RF0WL | RF0NL |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

**Table 32-44. MCAN_ILS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | RESERVED | R | 0h | |
| 29 | ARAL | R/W | 0h | Access to Reserved Address Line<br>0 Interrupt source is assigned to Interrupt Line 0<br>1 Interrupt source is assigned to Interrupt Line 1 |
| 28 | PEDL | R/W | 0h | Protocol Error in Data Phase Line<br>0 Interrupt source is assigned to Interrupt Line 0<br>1 Interrupt source is assigned to Interrupt Line 1 |
| 27 | PEAL | R/W | 0h | Protocol Error in Arbitration Phase Line<br>0 Interrupt source is assigned to Interrupt Line 0<br>1 Interrupt source is assigned to Interrupt Line 1 |
| 26 | WDIL | R/W | 0h | Watchdog Interrupt Line<br>0 Interrupt source is assigned to Interrupt Line 0<br>1 Interrupt source is assigned to Interrupt Line 1 |
| 25 | BOL | R/W | 0h | Bus_Off Status Line<br>0 Interrupt source is assigned to Interrupt Line 0<br>1 Interrupt source is assigned to Interrupt Line 1 |
| 24 | EWL | R/W | 0h | Warning Status Line<br>0 Interrupt source is assigned to Interrupt Line 0<br>1 Interrupt source is assigned to Interrupt Line 1 |
| 23 | EPL | R/W | 0h | Error Passive Line<br>0 Interrupt source is assigned to Interrupt Line 0<br>1 Interrupt source is assigned to Interrupt Line 1 |
| 22 | ELOL | R/W | 0h | Error Logging Overflow Line<br>0 Interrupt source is assigned to Interrupt Line 0<br>1 Interrupt source is assigned to Interrupt Line 1 |
| 21 | BEUL | R/W | 0h | Bit Error Uncorrected Line<br>0 Interrupt source is assigned to Interrupt Line 0<br>1 Interrupt source is assigned to Interrupt Line 1 |



Copyright © 2026 Texas Instruments Incorporated

**Table 32-44. MCAN_ILS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 20 | BECL | R/W | 0h | Bit Error Corrected Line<br>A separate interrupt line reserved for corrected bit errors is provided via the MCAN_ERROR_REGS. It advised for the user to use these registers and leave the MCAN_IE.BECE bit cleared to '0' (disabled), thereby relegating this bit to not applicable. |
| 19 | DRXL | R/W | 0h | Message Stored to Dedicated Rx Buffer Line<br>0 Interrupt source is assigned to Interrupt Line 0<br>1 Interrupt source is assigned to Interrupt Line 1 |
| 18 | TOOL | R/W | 0h | Timeout Occurred Line<br>0 Interrupt source is assigned to Interrupt Line 0<br>1 Interrupt source is assigned to Interrupt Line 1 |
| 17 | MRAFL | R/W | 0h | Message RAM Access Failure Line<br>0 Interrupt source is assigned to Interrupt Line 0<br>1 Interrupt source is assigned to Interrupt Line 1 |
| 16 | TSWL | R/W | 0h | Timestamp Wraparound Line<br>0 Interrupt source is assigned to Interrupt Line 0<br>1 Interrupt source is assigned to Interrupt Line 1 |
| 15 | TEFLL | R/W | 0h | Tx Event FIFO Element Lost Line<br>0 Interrupt source is assigned to Interrupt Line 0<br>1 Interrupt source is assigned to Interrupt Line 1 |
| 14 | TEFFL | R/W | 0h | Tx Event FIFO Full Line<br>0 Interrupt source is assigned to Interrupt Line 0<br>1 Interrupt source is assigned to Interrupt Line 1 |
| 13 | TEFWL | R/W | 0h | Tx Event FIFO Watermark Reached Line<br>0 Interrupt source is assigned to Interrupt Line 0<br>1 Interrupt source is assigned to Interrupt Line 1 |
| 12 | TEFNL | R/W | 0h | Tx Event FIFO New Entry Line<br>0 Interrupt source is assigned to Interrupt Line 0<br>1 Interrupt source is assigned to Interrupt Line 1 |
| 11 | TFEL | R/W | 0h | Tx FIFO Empty Line<br>0 Interrupt source is assigned to Interrupt Line 0<br>1 Interrupt source is assigned to Interrupt Line 1 |
| 10 | TCFL | R/W | 0h | Transmission Cancellation Finished Line<br>0 Interrupt source is assigned to Interrupt Line 0<br>1 Interrupt source is assigned to Interrupt Line 1 |
| 9 | TCL | R/W | 0h | Transmission Completed Line<br>0 Interrupt source is assigned to Interrupt Line 0<br>1 Interrupt source is assigned to Interrupt Line 1 |
| 8 | HPML | R/W | 0h | High Priority Message Line<br>0 Interrupt source is assigned to Interrupt Line 0<br>1 Interrupt source is assigned to Interrupt Line 1 |
| 7 | RF1LL | R/W | 0h | Rx FIFO 1 Message Lost Line<br>0 Interrupt source is assigned to Interrupt Line 0<br>1 Interrupt source is assigned to Interrupt Line 1 |
| 6 | RF1FL | R/W | 0h | Rx FIFO 1 Full Line<br>0 Interrupt source is assigned to Interrupt Line 0<br>1 Interrupt source is assigned to Interrupt Line 1 |
| 5 | RF1WL | R/W | 0h | Rx FIFO 1 Watermark Reached Line<br>0 Interrupt source is assigned to Interrupt Line 0<br>1 Interrupt source is assigned to Interrupt Line 1 |
| 4 | RF1NL | R/W | 0h | Rx FIFO 1 New Message Line<br>0 Interrupt source is assigned to Interrupt Line 0<br>1 Interrupt source is assigned to Interrupt Line 1 |
| 3 | RF0LL | R/W | 0h | Rx FIFO 0 Message Lost Line<br>0 Interrupt source is assigned to Interrupt Line 0<br>1 Interrupt source is assigned to Interrupt Line 1 |



Copyright © 2026 Texas Instruments Incorporated

**Table 32-44. MCAN_ILS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | RF0FL | R/W | 0h | Rx FIFO 0 Full Line<br>0 Interrupt source is assigned to Interrupt Line 0<br>1 Interrupt source is assigned to Interrupt Line 1 |
| 1 | RF0WL | R/W | 0h | Rx FIFO 0 Watermark Reached Line<br>0 Interrupt source is assigned to Interrupt Line 0<br>1 Interrupt source is assigned to Interrupt Line 1 |
| 0 | RF0NL | R/W | 0h | Rx FIFO 0 New Message Line<br>0 Interrupt source is assigned to Interrupt Line 0<br>1 Interrupt source is assigned to Interrupt Line 1 |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.26 MCAN_ILE (Offset = 705Ch) [Reset = 00000000h]

MCAN_ILE is shown in Figure 32-52 and described in Table 32-45.

Return to the Summary Table.

MCAN Interrupt Line Enable

**Figure 32-52. MCAN_ILE**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | EINT1 | EINT0 |
| R-0h | | | | | | R/W-0h | R/W-0h |

**Table 32-45. MCAN_ILE Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | |
| 1 | EINT1 | R/W | 0h | Enable Interrupt Line 1<br>0 Interrupt Line 1 is disabled<br>1 Interrupt Line 1 is enabled |
| 0 | EINT0 | R/W | 0h | Enable Interrupt Line 0<br>0 Interrupt Line 0 is disabled<br>1 Interrupt Line 0 is enabled |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.27 MCAN_GFC (Offset = 7080h) [Reset = 00000000h]

MCAN_GFC is shown in Figure 32-53 and described in Table 32-46.

Return to the Summary Table.

MCAN Global Filter Configuration

**Figure 32-53. MCAN_GFC**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | ANFS | | ANFE | | RRFS | RRFE |
| R-0h | | R/WQ-0h | | R/WQ-0h | | R/WQ-0h | R/WQ-0h |

**Table 32-46. MCAN_GFC Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | RESERVED | R | 0h | |
| 5-4 | ANFS | R/WQ | 0h | Accept Non-matching Frames Standard. Defines how received messages with 11-bit IDs that do not match any element of the filter list are treated.<br>00 Accept in Rx FIFO 0<br>01 Accept in Rx FIFO 1<br>10 Reject<br>11 Reject<br>Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. |
| 3-2 | ANFE | R/WQ | 0h | Accept Non-matching Frames Extended. Defines how received messages with 29-bit IDs that do not match any element of the filter list are treated.<br>00 Accept in Rx FIFO 0<br>01 Accept in Rx FIFO 1<br>10 Reject<br>11 Reject<br>Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. |
| 1 | RRFS | R/WQ | 0h | Reject Remote Frames Standard<br>0 Filter remote frames with 11-bit standard IDs<br>1 Reject all remote frames with 11-bit standard IDs<br>Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. |
| 0 | RRFE | R/WQ | 0h | Reject Remote Frames Extended<br>0 Filter remote frames with 29-bit extended IDs<br>1 Reject all remote frames with 29-bit extended IDs<br>Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.28 MCAN_SIDFC (Offset = 7084h) [Reset = 00000000h]

MCAN_SIDFC is shown in Figure 32-54 and described in Table 32-47.

Return to the Summary Table.

MCAN Standard ID Filter Configuration

**Figure 32-54. MCAN_SIDFC**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| LSS | | | | | | | |
| R/WQ-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| FLSSA | | | | | | | |
| R/WQ-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| FLSSA | | | | | | RESERVED | |
| R/WQ-0h | | | | | | R-0h | |

**Table 32-47. MCAN_SIDFC Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | RESERVED | R | 0h | |
| 23-16 | LSS | R/WQ | 0h | List Size Standard<br>0 No standard Message ID filter<br>1-128 Number of standard Message ID filter elements<br>>128 Values greater than 128 are interpreted as 128 |
| 15-2 | FLSSA | R/WQ | 0h | Filter List Standard Start Address. Start address of standard Message ID filter list (32-bit word address). |
| 1-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.29 MCAN_XIDFC (Offset = 7088h) [Reset = 00000000h]

MCAN_XIDFC is shown in Figure 32-55 and described in Table 32-48.

Return to the Summary Table.

MCAN Extended ID Filter Configuration

**Figure 32-55. MCAN_XIDFC**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | LSE | | | | | | |
| R-0h | R/WQ-0h | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| FLESA | | | | | | | |
| R/WQ-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| FLESA | | | | | | RESERVED | |
| R/WQ-0h | | | | | | R-0h | |

**Table 32-48. MCAN_XIDFC Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-23 | RESERVED | R | 0h | |
| 22-16 | LSE | R/WQ | 0h | Filter List Extended Start Address. Start address of extended Message ID filter list (32-bit word address).<br>Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. |
| 15-2 | FLESA | R/WQ | 0h | List Size Extended<br>0 No extended Message ID filter<br>1-64 Number of extended Message ID filter elements<br>>64 Values greater than 64 are interpreted as 64<br>Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. |
| 1-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.30 MCAN_XIDAM (Offset = 7090h) [Reset = 1FFFFFFFh]

MCAN_XIDAM is shown in Figure 32-56 and described in Table 32-49.

Return to the Summary Table.

MCAN Extended ID and Mask

**Figure 32-56. MCAN_XIDAM**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | EIDM | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | R/WQ-1FFFFFFFh | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 32-49. MCAN_XIDAM Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | RESERVED | R | 0h | |
| 28-0 | EIDM | R/WQ | 1FFFFFFFh | Extended ID Mask. For acceptance filtering of extended frames the Extended ID AND Mask is ANDed with the Message ID of a received frame. Intended for masking of 29-bit IDs in SAE J1939. With the reset value of all bits set to one the mask is not active.<br>Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.31 MCAN_HPMS (Offset = 7094h) [Reset = 00000000h]

MCAN_HPMS is shown in Figure 32-57 and described in Table 32-50.

Return to the Summary Table.

This register is updated every time a Message ID filter element configured to generate a priority event matches. This can be used to monitor the status of incoming high priority messages and to enable fast access to these messages.

**Figure 32-57. MCAN_HPMS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| FLST | FIDX | | | | | | |
| R-0h | R-0h | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MSI | | BIDX | | | | | |
| R-0h | | R-0h | | | | | |

**Table 32-50. MCAN_HPMS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15 | FLST | R | 0h | Filter List. Indicates the filter list of the matching filter element.<br>0 Standard Filter List<br>1 Extended Filter List |
| 14-8 | FIDX | R | 0h | Filter Index. Index of matching filter element. Range is 0 to SIDFC.LSS - 1 resp. XIDFC.LSE - 1. |
| 7-6 | MSI | R | 0h | Message Storage Indicator<br>00 No FIFO selected<br>01 FIFO message lost<br>10 Message stored in FIFO 0<br>11 Message stored in FIFO 1 |
| 5-0 | BIDX | R | 0h | Buffer Index. Index of Rx FIFO element to which the message was stored. Only valid when MSI(1) = '1'. |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.32 MCAN_NDAT1 (Offset = 7098h) [Reset = 00000000h]

MCAN_NDAT1 is shown in Figure 32-58 and described in Table 32-51.

Return to the Summary Table.

MCAN New Data 1

**Figure 32-58. MCAN_NDAT1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| ND31 | ND30 | ND29 | ND28 | ND27 | ND26 | ND25 | ND24 |
| R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| ND23 | ND22 | ND21 | ND20 | ND19 | ND18 | ND17 | ND16 |
| R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| ND15 | ND14 | ND13 | ND12 | ND11 | ND10 | ND9 | ND8 |
| R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ND7 | ND6 | ND5 | ND4 | ND3 | ND2 | ND1 | ND0 |
| R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h |

**Table 32-51. MCAN_NDAT1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | ND31 | R/W1C | 0h | New Data RX Buffer 31<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 30 | ND30 | R/W1C | 0h | New Data RX Buffer 30<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 29 | ND29 | R/W1C | 0h | New Data RX Buffer 29<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 28 | ND28 | R/W1C | 0h | New Data RX Buffer 28<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 27 | ND27 | R/W1C | 0h | New Data RX Buffer 27<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 26 | ND26 | R/W1C | 0h | New Data RX Buffer 26<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 25 | ND25 | R/W1C | 0h | New Data RX Buffer 25<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 24 | ND24 | R/W1C | 0h | New Data RX Buffer 24<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 23 | ND23 | R/W1C | 0h | New Data RX Buffer 23<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 22 | ND22 | R/W1C | 0h | New Data RX Buffer 22<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |



Copyright © 2026 Texas Instruments Incorporated

**Table 32-51. MCAN_NDAT1 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 21 | ND21 | R/W1C | 0h | New Data RX Buffer 21<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 20 | ND20 | R/W1C | 0h | New Data RX Buffer 20<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 19 | ND19 | R/W1C | 0h | New Data RX Buffer 19<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 18 | ND18 | R/W1C | 0h | New Data RX Buffer 18<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 17 | ND17 | R/W1C | 0h | New Data RX Buffer 17<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 16 | ND16 | R/W1C | 0h | New Data RX Buffer 16<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 15 | ND15 | R/W1C | 0h | New Data RX Buffer 15<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 14 | ND14 | R/W1C | 0h | New Data RX Buffer 14<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 13 | ND13 | R/W1C | 0h | New Data RX Buffer 13<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 12 | ND12 | R/W1C | 0h | New Data RX Buffer 12<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 11 | ND11 | R/W1C | 0h | New Data RX Buffer 11<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 10 | ND10 | R/W1C | 0h | New Data RX Buffer 10<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 9 | ND9 | R/W1C | 0h | New Data RX Buffer 9<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 8 | ND8 | R/W1C | 0h | New Data RX Buffer 8<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 7 | ND7 | R/W1C | 0h | New Data RX Buffer 7<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 6 | ND6 | R/W1C | 0h | New Data RX Buffer 6<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 5 | ND5 | R/W1C | 0h | New Data RX Buffer 5<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 4 | ND4 | R/W1C | 0h | New Data RX Buffer 4<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 3 | ND3 | R/W1C | 0h | New Data RX Buffer 3<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |



Copyright © 2026 Texas Instruments Incorporated

**Table 32-51. MCAN_NDAT1 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | ND2 | R/W1C | 0h | New Data RX Buffer 2<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 1 | ND1 | R/W1C | 0h | New Data RX Buffer 1<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 0 | ND0 | R/W1C | 0h | New Data RX Buffer 0<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.33 MCAN_NDAT2 (Offset = 709Ch) [Reset = 00000000h]

MCAN_NDAT2 is shown in Figure 32-59 and described in Table 32-52.

Return to the Summary Table.

MCAN New Data 2

**Figure 32-59. MCAN_NDAT2**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| ND63 | ND62 | ND61 | ND60 | ND59 | ND58 | ND57 | ND56 |
| R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| ND55 | ND54 | ND53 | ND52 | ND51 | ND50 | ND49 | ND48 |
| R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| ND47 | ND46 | ND45 | ND44 | ND43 | ND42 | ND41 | ND40 |
| R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ND39 | ND38 | ND37 | ND36 | ND35 | ND34 | ND33 | ND32 |
| R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h |

**Table 32-52. MCAN_NDAT2 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | ND63 | R/W1C | 0h | New Data RX Buffer 63<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 30 | ND62 | R/W1C | 0h | New Data RX Buffer 62<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 29 | ND61 | R/W1C | 0h | New Data RX Buffer 61<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 28 | ND60 | R/W1C | 0h | New Data RX Buffer 60<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 27 | ND59 | R/W1C | 0h | New Data RX Buffer 59<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 26 | ND58 | R/W1C | 0h | New Data RX Buffer 58<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 25 | ND57 | R/W1C | 0h | New Data RX Buffer 57<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 24 | ND56 | R/W1C | 0h | New Data RX Buffer 56<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 23 | ND55 | R/W1C | 0h | New Data RX Buffer 55<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 22 | ND54 | R/W1C | 0h | New Data RX Buffer 54<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |



Copyright © 2026 Texas Instruments Incorporated

**Table 32-52. MCAN_NDAT2 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 21 | ND53 | R/W1C | 0h | New Data RX Buffer 53<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 20 | ND52 | R/W1C | 0h | New Data RX Buffer 52<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 19 | ND51 | R/W1C | 0h | New Data RX Buffer 51<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 18 | ND50 | R/W1C | 0h | New Data RX Buffer 50<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 17 | ND49 | R/W1C | 0h | New Data RX Buffer 49<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 16 | ND48 | R/W1C | 0h | New Data RX Buffer 48<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 15 | ND47 | R/W1C | 0h | New Data RX Buffer 47<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 14 | ND46 | R/W1C | 0h | New Data RX Buffer 46<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 13 | ND45 | R/W1C | 0h | New Data RX Buffer 45<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 12 | ND44 | R/W1C | 0h | New Data RX Buffer 44<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 11 | ND43 | R/W1C | 0h | New Data RX Buffer 43<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 10 | ND42 | R/W1C | 0h | New Data RX Buffer 42<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 9 | ND41 | R/W1C | 0h | New Data RX Buffer 41<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 8 | ND40 | R/W1C | 0h | New Data RX Buffer 40<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 7 | ND39 | R/W1C | 0h | New Data RX Buffer 39<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 6 | ND38 | R/W1C | 0h | New Data RX Buffer 38<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 5 | ND37 | R/W1C | 0h | New Data RX Buffer 37<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 4 | ND36 | R/W1C | 0h | New Data RX Buffer 36<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 3 | ND35 | R/W1C | 0h | New Data RX Buffer 35<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |



Copyright © 2026 Texas Instruments Incorporated

**Table 32-52. MCAN_NDAT2 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2 | ND34 | R/W1C | 0h | New Data RX Buffer 34<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 1 | ND33 | R/W1C | 0h | New Data RX Buffer 33<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |
| 0 | ND32 | R/W1C | 0h | New Data RX Buffer 32<br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.34 MCAN_RXF0C (Offset = 70A0h) [Reset = 00000000h]

MCAN_RXF0C is shown in Figure 32-60 and described in Table 32-53.

Return to the Summary Table.

MCAN Rx FIFO 0 Configuration

**Figure 32-60. MCAN_RXF0C**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| F0OM | F0WM | | | | | | |
| R/WQ-0h | R/WQ-0h | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | F0S | | | | | | |
| R-0h | R/WQ-0h | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| F0SA | | | | | | | |
| R/WQ-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| F0SA | | | | | | RESERVED | |
| R/WQ-0h | | | | | | R-0h | |

**Table 32-53. MCAN_RXF0C Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | F0OM | R/WQ | 0h | FIFO 0 Operation Mode. FIFO 0 can be operated in blocking or in overwrite mode.<br>0 FIFO 0 blocking mode<br>1 FIFO 0 overwrite mode<br>Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. |
| 30-24 | F0WM | R/WQ | 0h | Rx FIFO 0 Watermark<br>0 Watermark interrupt disabled<br>1-64 Level for Rx FIFO 0 watermark interrupt (IR.RF0W)<br>>64 Watermark interrupt disabled<br>Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. |
| 23 | RESERVED | R | 0h | |
| 22-16 | F0S | R/WQ | 0h | Rx FIFO 0 Size. The Rx FIFO 0 elements are indexed from 0 to F0S-1.<br>0 No Rx FIFO 0<br>1-64 Number of Rx FIFO 0 elements<br>>64 Values greater than 64 are interpreted as 64<br>Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. |
| 15-2 | F0SA | R/WQ | 0h | Rx FIFO 0 Start Address. Start address of Rx FIFO 0 in Message RAM (32-bit word address).<br>Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. |
| 1-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.35 MCAN_RXF0S (Offset = 70A4h) [Reset = 00000000h]

MCAN_RXF0S is shown in Figure 32-61 and described in Table 32-54.

Return to the Summary Table.

MCAN Rx FIFO 0 Status

**Figure 32-61. MCAN_RXF0S**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | RF0L | F0F |
| R-0h | | | | | | R-0h | R-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | F0PI | | | | | |
| R-0h | | R-0h | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | F0GI | | | | | |
| R-0h | | R-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | F0FL | | | | | | |
| R-0h | R-0h | | | | | | |

**Table 32-54. MCAN_RXF0S Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | RESERVED | R | 0h | |
| 25 | RF0L | R | 0h | Rx FIFO 0 Message Lost. This bit is a copy of interrupt flag IR.RF0L. When IR.RF0L is reset, this bit is also reset.<br>0 No Rx FIFO 0 message lost<br>1 Rx FIFO 0 message lost, also set after write attempt to Rx FIFO 0 of size zero<br>Note: Overwriting the oldest message when RXF0C.F0OM = '1' will not set this flag. |
| 24 | F0F | R | 0h | Rx FIFO 0 Full<br>0 Rx FIFO 0 not full<br>1 Rx FIFO 0 full |
| 23-22 | RESERVED | R | 0h | |
| 21-16 | F0PI | R | 0h | Rx FIFO 0 Put Index. Rx FIFO 0 write index pointer, range 0 to 63. |
| 15-14 | RESERVED | R | 0h | |
| 13-8 | F0GI | R | 0h | Rx FIFO 0 Get Index. Rx FIFO 0 read index pointer, range 0 to 63. |
| 7 | RESERVED | R | 0h | |
| 6-0 | F0FL | R | 0h | Rx FIFO 0 Fill Level. Number of elements stored in Rx FIFO 0, range 0 to 64. |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.36 MCAN_RXF0A (Offset = 70A8h) [Reset = 00000000h]

MCAN_RXF0A is shown in Figure 32-62 and described in Table 32-55.

Return to the Summary Table.

MCAN Rx FIFO 0 Acknowledge

**Figure 32-62. MCAN_RXF0A**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | F0AI | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | R/W-0h | | | | | |

**Table 32-55. MCAN_RXF0A Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | RESERVED | R | 0h | |
| 5-0 | F0AI | R/W | 0h | Rx FIFO 0 Acknowledge Index. After the Host has read a message or a sequence of messages from Rx FIFO 0 it has to write the buffer index of the last element read from Rx FIFO 0 to F0AI. This will set the Rx FIFO 0 Get Index RXF0S.F0GI to F0AI + 1 and update the FIFO 0 Fill Level RXF0S.F0FL. |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.37 MCAN_RXBC (Offset = 70ACh) [Reset = 00000000h]

MCAN_RXBC is shown in Figure 32-63 and described in Table 32-56.

Return to the Summary Table.

MCAN Rx Buffer Configuration

**Figure 32-63. MCAN_RXBC**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | RBSA | | | | | | | | | | | | | | RESERVED | |
| R-0h | | | | | | | | | | | | | | | | R/WQ-0h | | | | | | | | | | | | | | R-0h | |

**Table 32-56. MCAN_RXBC Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15-2 | RBSA | R/WQ | 0h | Rx Buffer Start Address. Configures the start address of the Rx Buffers section in the Message RAM (32-bit word address). +I466 |
| 1-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.38 MCAN_RXF1C (Offset = 70B0h) [Reset = 00000000h]

MCAN_RXF1C is shown in Figure 32-64 and described in Table 32-57.

Return to the Summary Table.

MCAN Rx FIFO 1 Configuration

**Figure 32-64. MCAN_RXF1C**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| F1OM | F1WM | | | | | | |
| R/WQ-0h | R/WQ-0h | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | F1S | | | | | | |
| R-0h | R/WQ-0h | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| F1SA | | | | | | | |
| R/WQ-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| F1SA | | | | | | RESERVED | |
| R/WQ-0h | | | | | | R-0h | |

**Table 32-57. MCAN_RXF1C Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | F1OM | R/WQ | 0h | FIFO 1 Operation Mode. FIFO 1 can be operated in blocking or in overwrite mode.<br>0 FIFO 1 blocking mode<br>1 FIFO 1 overwrite mode<br>Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. |
| 30-24 | F1WM | R/WQ | 0h | Rx FIFO 1 Watermark<br>0 Watermark interrupt disabled<br>1-64 Level for Rx FIFO 1 watermark interrupt (IR.RF1W)<br>>64 Watermark interrupt disabled<br>Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. |
| 23 | RESERVED | R | 0h | |
| 22-16 | F1S | R/WQ | 0h | Rx FIFO 1 Size. The Rx FIFO 1 elements are indexed from 0 to F1S - 1.<br>0 No Rx FIFO 1<br>1-64 Number of Rx FIFO 1 elements<br>>64 Values greater than 64 are interpreted as 64<br>Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. |
| 15-2 | F1SA | R/WQ | 0h | Rx FIFO 1 Start Address Start address of Rx FIFO 1 in Message RAM (32-bit word address). |
| 1-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.39 MCAN_RXF1S (Offset = 70B4h) [Reset = 00000000h]

MCAN_RXF1S is shown in Figure 32-65 and described in Table 32-58.

Return to the Summary Table.

MCAN Rx FIFO 1 Status

**Figure 32-65. MCAN_RXF1S**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DMS | | RESERVED | | | | RF1L | F1F |
| R-0h | | R-0h | | | | R-0h | R-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | F1PI | | | | | |
| R-0h | | R-0h | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | F1GI | | | | | |
| R-0h | | R-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | F1FL | | | | | | |
| R-0h | R-0h | | | | | | |

**Table 32-58. MCAN_RXF1S Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | DMS | R | 0h | Debug Message Status<br>00 Idle state, wait for reception of debug messages<br>01 Debug message A received<br>10 Debug messages A, B received<br>11 Debug messages A, B, C received |
| 29-26 | RESERVED | R | 0h | |
| 25 | RF1L | R | 0h | Rx FIFO 1 Message Lost. This bit is a copy of interrupt flag IR.RF1L. When IR.RF1L is reset, this bit is also reset.<br>0 No Rx FIFO 1 message lost<br>1 Rx FIFO 1 message lost, also set after write attempt to Rx FIFO 1 of size zero<br>Note: Overwriting the oldest message when RXF1C.F1OM = '1' will not set this flag. |
| 24 | F1F | R | 0h | Rx FIFO 1 Full<br>0 Rx FIFO 1 not full<br>1 Rx FIFO 1 full |
| 23-22 | RESERVED | R | 0h | |
| 21-16 | F1PI | R | 0h | Rx FIFO 1 Put Index. Rx FIFO 1 write index pointer, range 0 to 63. |
| 15-14 | RESERVED | R | 0h | |
| 13-8 | F1GI | R | 0h | Rx FIFO 1 Get Index. Rx FIFO 1 read index pointer, range 0 to 63. |
| 7 | RESERVED | R | 0h | |
| 6-0 | F1FL | R | 0h | Rx FIFO 1 Fill Level. Number of elements stored in Rx FIFO 1, range 0 to 64. |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.40 MCAN_RXF1A (Offset = 70B8h) [Reset = 00000000h]

MCAN_RXF1A is shown in Figure 32-66 and described in Table 32-59.

Return to the Summary Table.

MCAN Rx FIFO 1 Acknowledge

**Figure 32-66. MCAN_RXF1A**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | F1AI | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | R/W-0h | | | | | |

**Table 32-59. MCAN_RXF1A Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | RESERVED | R | 0h | |
| 5-0 | F1AI | R/W | 0h | Rx FIFO 1 Acknowledge Index. After the Host has read a message or a sequence of messages from Rx FIFO 1 it has to write the buffer index of the last element read from Rx FIFO 1 to F1AI. This will set the Rx FIFO 1 Get Index RXF1S.F1GI to F1AI + 1 and update the FIFO 1 Fill Level RXF1S.F1FL. |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.41 MCAN_RXESC (Offset = 70BCh) [Reset = 00000000h]

MCAN_RXESC is shown in Figure 32-67 and described in Table 32-60.

Return to the Summary Table.

Configures the number of data bytes belonging to an Rx Buffer / Rx FIFO element. Data field sizes >8 bytes are intended for CAN FD operation only.

**Figure 32-67. MCAN_RXESC**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | RBDS | | |
| R-0h | | | | | R/WQ-0h | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | F1DS | | | RESERVED | F0DS | | |
| R-0h | R/WQ-0h | | | R-0h | R/WQ-0h | | |

**Table 32-60. MCAN_RXESC Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-11 | RESERVED | R | 0h | |
| 10-8 | RBDS | R/WQ | 0h | Rx Buffer Data Field Size<br>000 8 byte data field<br>001 12 byte data field<br>010 16 byte data field<br>011 20 byte data field<br>100 24 byte data field<br>101 32 byte data field<br>110 48 byte data field<br>111 64 byte data field<br>Note: In case the data field size of an accepted CAN frame exceeds the data field size configured for the matching Rx Buffer or Rx FIFO, only the number of bytes as configured by RXESC are stored to the Rx Buffer resp. Rx FIFO element. The rest of the frame's data field is ignored.<br>Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. |
| 7 | RESERVED | R | 0h | |
| 6-4 | F1DS | R/WQ | 0h | Rx FIFO 1 Data Field Size<br>000 8 byte data field<br>001 12 byte data field<br>010 16 byte data field<br>011 20 byte data field<br>100 24 byte data field<br>101 32 byte data field<br>110 48 byte data field<br>111 64 byte data field<br>Note: In case the data field size of an accepted CAN frame exceeds the data field size configured for the matching Rx Buffer or Rx FIFO, only the number of bytes as configured by RXESC are stored to the Rx Buffer resp. Rx FIFO element. The rest of the frame's data field is ignored.<br>Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. |



Copyright © 2026 Texas Instruments Incorporated

**Table 32-60. MCAN_RXESC Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 3 | RESERVED | R | 0h | |
| 2-0 | F0DS | R/WQ | 0h | Rx FIFO 0 Data Field Size<br>000 8 byte data field<br>001 12 byte data field<br>010 16 byte data field<br>011 20 byte data field<br>100 24 byte data field<br>101 32 byte data field<br>110 48 byte data field<br>111 64 byte data field<br>Note: In case the data field size of an accepted CAN frame exceeds the data field size configured for the matching Rx Buffer or Rx FIFO, only the number of bytes as configured by RXESC are stored to the Rx Buffer resp. Rx FIFO element. The rest of the frame's data field is ignored.<br>Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.42 MCAN_TXBC (Offset = 70C0h) [Reset = 00000000h]

MCAN_TXBC is shown in Figure 32-68 and described in Table 32-61.

Return to the Summary Table.

MCAN Tx Buffer Configuration

**Figure 32-68. MCAN_TXBC**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | TFQM | TFQS | | | | | |
| R-0h | R/WQ-0h | R/WQ-0h | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | NDTB | | | | | |
| R-0h | | R/WQ-0h | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TBSA | | | | | | | |
| R/WQ-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TBSA | | | | | | RESERVED | |
| R/WQ-0h | | | | | | R-0h | |

**Table 32-61. MCAN_TXBC Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | RESERVED | R | 0h | |
| 30 | TFQM | R/WQ | 0h | Tx FIFO/Queue Mode<br>0 Tx FIFO operation<br>1 Tx Queue operation<br>Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. |
| 29-24 | TFQS | R/WQ | 0h | Transmit FIFO/Queue Size<br>0 No Tx FIFO/Queue<br>1-32 Number of Tx Buffers used for Tx FIFO/Queue<br>>32 Values greater than 32 are interpreted as 32<br>Note: Be aware that the sum of TFQS and NDTB may be not greater than 32. There is no check<br>for erroneous configurations. The Tx Buffers section in the Message RAM starts with the<br>dedicated Tx Buffers.<br>Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. |
| 23-22 | RESERVED | R | 0h | |
| 21-16 | NDTB | R/WQ | 0h | Number of Dedicated Transmit Buffers<br>0 No Dedicated Tx Buffers<br>1-32 Number of Dedicated Tx Buffers<br>>32 Values greater than 32 are interpreted as 32<br>Note: Be aware that the sum of TFQS and NDTB may be not greater than 32. There is no check<br>for erroneous configurations. The Tx Buffers section in the Message RAM starts with the<br>dedicated Tx Buffers.<br>Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. |
| 15-2 | TBSA | R/WQ | 0h | Tx Buffers Start Address. Start address of Tx Buffers section in Message RAM (32-bit word address).<br>Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. |



Copyright © 2026 Texas Instruments Incorporated

**Table 32-61. MCAN_TXBC Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 1-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.43 MCAN_TXFQS (Offset = 70C4h) [Reset = 00000000h]

MCAN_TXFQS is shown in Figure 32-69 and described in Table 32-62.

Return to the Summary Table.

The Tx FIFO/Queue status is related to the pending Tx requests listed in register TXBRP. Therefore the effect of Add/Cancellation requests may be delayed due to a running Tx scan (TXBRP not yet updated).

**Figure 32-69. MCAN_TXFQS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | TFQF | TFQP | | | | |
| R-0h | | R-0h | R-0h | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | TFGI | | | | |
| R-0h | | | R-0h | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | TFFL | | | | | |
| R-0h | | R-0h | | | | | |

**Table 32-62. MCAN_TXFQS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-22 | RESERVED | R | 0h | |
| 21 | TFQF | R | 0h | Tx FIFO/Queue Full<br>0 Tx FIFO/Queue not full<br>1 Tx FIFO/Queue full |
| 20-16 | TFQP | R | 0h | Tx FIFO/Queue Put Index. Tx FIFO/Queue write index pointer, range 0 to 31.<br>Note: In case of mixed configurations where dedicated Tx Buffers are combined with a Tx FIFO or a Tx Queue, the Put and Get indexes indicate the number of the Tx Buffer starting with the first dedicated Tx Buffers. Example: For a configuration of 12 dedicated Tx Buffers and a Tx FIFO of 20 Buffers a Put Index of 15 points to the fourth buffer of the Tx FIFO. |
| 15-13 | RESERVED | R | 0h | |
| 12-8 | TFGI | R | 0h | Tx FIFO Get Index. Tx FIFO read index pointer, range 0 to 31. Read as zero when Tx Queue operation is configured (TXBC.TFQM = '1').<br>Note: In case of mixed configurations where dedicated Tx Buffers are combined with a Tx FIFO or a Tx Queue, the Put and Get indexes indicate the number of the Tx Buffer starting with the first dedicated Tx Buffers. Example: For a configuration of 12 dedicated Tx Buffers and a Tx FIFO of 20 Buffers a Put Index of 15 points to the fourth buffer of the Tx FIFO. |
| 7-6 | RESERVED | R | 0h | |
| 5-0 | TFFL | R | 0h | Tx FIFO Free Level. Number of consecutive free Tx FIFO elements starting from TFGI, range 0 to 32. Read as zero when Tx Queue operation is configured (TXBC.TFQM = '1'). |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.44 MCAN_TXESC (Offset = 70C8h) [Reset = 00000000h]

MCAN_TXESC is shown in Figure 32-70 and described in Table 32-63.

Return to the Summary Table.

Configures the number of data bytes belonging to a Tx Buffer element. Data field sizes > 8 bytes are intended for CAN FD operation only.

**Figure 32-70. MCAN_TXESC**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | | TBDS | | |
| R-0h | | | | | | | | | | | | | R/WQ-0h | | |

**Table 32-63. MCAN_TXESC Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | RESERVED | R | 0h | |
| 2-0 | TBDS | R/WQ | 0h | Tx Buffer Data Field Size<br>000 8 byte data field<br>001 12 byte data field<br>010 16 byte data field<br>011 20 byte data field<br>100 24 byte data field<br>101 32 byte data field<br>110 48 byte data field<br>111 64 byte data field<br>Note: In case the data length code DLC of a Tx Buffer element is configured to a value higher than the Tx Buffer data field size TXESC.TBDS, the bytes not defined by the Tx Buffer are transmitted as '0xCC' (padding bytes).<br>Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.45 MCAN_TXBRP (Offset = 70CCh) [Reset = 00000000h]

MCAN_TXBRP is shown in Figure 32-71 and described in Table 32-64.

Return to the Summary Table.

MCAN Tx Buffer Request Pending

**Figure 32-71. MCAN_TXBRP**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| TRP31 | TRP30 | TRP29 | TRP28 | TRP27 | TRP26 | TRP25 | TRP24 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TRP23 | TRP22 | TRP21 | TRP20 | TRP19 | TRP18 | TRP17 | TRP16 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TRP15 | TRP14 | TRP13 | TRP12 | TRP11 | TRP10 | TRP9 | TRP8 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TRP7 | TRP6 | TRP5 | TRP4 | TRP3 | TRP2 | TRP1 | TRP0 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 32-64. MCAN_TXBRP Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | TRP31 | R | 0h | Transmission Request Pending 31. See description for bit 0. |
| 30 | TRP30 | R | 0h | Transmission Request Pending 30. See description for bit 0. |
| 29 | TRP29 | R | 0h | Transmission Request Pending 29. See description for bit 0. |
| 28 | TRP28 | R | 0h | Transmission Request Pending 28. See description for bit 0. |
| 27 | TRP27 | R | 0h | Transmission Request Pending 27. See description for bit 0. |
| 26 | TRP26 | R | 0h | Transmission Request Pending 26. See description for bit 0. |
| 25 | TRP25 | R | 0h | Transmission Request Pending 25. See description for bit 0. |
| 24 | TRP24 | R | 0h | Transmission Request Pending 24. See description for bit 0. |
| 23 | TRP23 | R | 0h | Transmission Request Pending 23. See description for bit 0. |
| 22 | TRP22 | R | 0h | Transmission Request Pending 22. See description for bit 0. |
| 21 | TRP21 | R | 0h | Transmission Request Pending 21. See description for bit 0. |
| 20 | TRP20 | R | 0h | Transmission Request Pending 20. See description for bit 0. |
| 19 | TRP19 | R | 0h | Transmission Request Pending 19. See description for bit 0. |
| 18 | TRP18 | R | 0h | Transmission Request Pending 18. See description for bit 0. |
| 17 | TRP17 | R | 0h | Transmission Request Pending 17. See description for bit 0. |
| 16 | TRP16 | R | 0h | Transmission Request Pending 16. See description for bit 0. |
| 15 | TRP15 | R | 0h | Transmission Request Pending 15. See description for bit 0. |
| 14 | TRP14 | R | 0h | Transmission Request Pending 14. See description for bit 0. |
| 13 | TRP13 | R | 0h | Transmission Request Pending 13. See description for bit 0. |
| 12 | TRP12 | R | 0h | Transmission Request Pending 12. See description for bit 0. |
| 11 | TRP11 | R | 0h | Transmission Request Pending 11. See description for bit 0. |
| 10 | TRP10 | R | 0h | Transmission Request Pending 10. See description for bit 0. |
| 9 | TRP9 | R | 0h | Transmission Request Pending 9. See description for bit 0. |
| 8 | TRP8 | R | 0h | Transmission Request Pending 8. See description for bit 0. |
| 7 | TRP7 | R | 0h | Transmission Request Pending 7. See description for bit 0. |



Copyright © 2026 Texas Instruments Incorporated

**Table 32-64. MCAN_TXBRP Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 6 | TRP6 | R | 0h | Transmission Request Pending 6. See description for bit 0. |
| 5 | TRP5 | R | 0h | Transmission Request Pending 5. See description for bit 0. |
| 4 | TRP4 | R | 0h | Transmission Request Pending 4. See description for bit 0. |
| 3 | TRP3 | R | 0h | Transmission Request Pending 3. See description for bit 0. |
| 2 | TRP2 | R | 0h | Transmission Request Pending 2. See description for bit 0. |
| 1 | TRP1 | R | 0h | Transmission Request Pending 1. See description for bit 0. |
| 0 | TRP0 | R | 0h | Transmission Request Pending 0.<br>Each Tx Buffer has its own Transmission Request Pending bit.<br>The bits are set via register TXBAR. The bits are reset after a requested transmission has completed or has been canceled via register TXBCR.<br>TXBRP bits are set only for those Tx Buffers configured via TXBC.<br>After a TXBRP bit has been set, a Tx scan is started to check for the pending Tx request with the highest priority (Tx Buffer with lowest Message ID).<br>A cancellation request resets the corresponding transmission request pending bit of register TXBRP. In case a transmission has already been started when a cancellation is requested, this is done at the end of the transmission, regardless whether the transmission was successful or not. The cancellation request bits are reset directly after the corresponding TXBRP bit has been reset.<br>After a cancellation has been requested, a finished cancellation is signaled via TXBCF<br>- after successful transmission together with the corresponding TXBTO bit<br>- when the transmission has not yet been started at the point of cancellation<br>- when the transmission has been aborted due to lost arbitration<br>- when an error occurred during frame transmission<br>In DAR mode all transmissions are automatically canceled if they are not successful. The corresponding TXBCF bit is set for all unsuccessful transmissions.<br>0 No transmission request pending<br>1 Transmission request pending<br>Note: TXBRP bits which are set while a Tx scan is in progress are not considered during this particular Tx scan. In case a cancellation is requested for such a Tx Buffer, this Add Request is canceled immediately, the corresponding TXBRP bit is reset. |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.46 MCAN_TXBAR (Offset = 70D0h) [Reset = 00000000h]

MCAN_TXBAR is shown in Figure 32-72 and described in Table 32-65.

Return to the Summary Table.

MCAN Tx Buffer Add Request

**Figure 32-72. MCAN_TXBAR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| AR31 | AR30 | AR29 | AR28 | AR27 | AR26 | AR25 | AR24 |
| R/WQ-0h | R/WQ-0h | R/WQ-0h | R/WQ-0h | R/WQ-0h | R/WQ-0h | R/WQ-0h | R/WQ-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| AR23 | AR22 | AR21 | AR20 | AR19 | AR18 | AR17 | AR16 |
| R/WQ-0h | R/WQ-0h | R/WQ-0h | R/WQ-0h | R/WQ-0h | R/WQ-0h | R/WQ-0h | R/WQ-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| AR15 | AR14 | AR13 | AR12 | AR11 | AR10 | AR9 | AR8 |
| R/WQ-0h | R/WQ-0h | R/WQ-0h | R/WQ-0h | R/WQ-0h | R/WQ-0h | R/WQ-0h | R/WQ-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| AR7 | AR6 | AR5 | AR4 | AR3 | AR2 | AR1 | AR0 |
| R/WQ-0h | R/WQ-0h | R/WQ-0h | R/WQ-0h | R/WQ-0h | R/WQ-0h | R/WQ-0h | R/WQ-0h |

**Table 32-65. MCAN_TXBAR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | AR31 | R/WQ | 0h | Add Request 31. See description for bit 0. |
| 30 | AR30 | R/WQ | 0h | Add Request 30. See description for bit 0. |
| 29 | AR29 | R/WQ | 0h | Add Request 29. See description for bit 0. |
| 28 | AR28 | R/WQ | 0h | Add Request 28. See description for bit 0. |
| 27 | AR27 | R/WQ | 0h | Add Request 27. See description for bit 0. |
| 26 | AR26 | R/WQ | 0h | Add Request 26. See description for bit 0. |
| 25 | AR25 | R/WQ | 0h | Add Request 25. See description for bit 0. |
| 24 | AR24 | R/WQ | 0h | Add Request 24. See description for bit 0. |
| 23 | AR23 | R/WQ | 0h | Add Request 23. See description for bit 0. |
| 22 | AR22 | R/WQ | 0h | Add Request 22. See description for bit 0. |
| 21 | AR21 | R/WQ | 0h | Add Request 21. See description for bit 0. |
| 20 | AR20 | R/WQ | 0h | Add Request 20. See description for bit 0. |
| 19 | AR19 | R/WQ | 0h | Add Request 19. See description for bit 0. |
| 18 | AR18 | R/WQ | 0h | Add Request 18. See description for bit 0. |
| 17 | AR17 | R/WQ | 0h | Add Request 17. See description for bit 0. |
| 16 | AR16 | R/WQ | 0h | Add Request 16. See description for bit 0. |
| 15 | AR15 | R/WQ | 0h | Add Request 15. See description for bit 0. |
| 14 | AR14 | R/WQ | 0h | Add Request 14. See description for bit 0. |
| 13 | AR13 | R/WQ | 0h | Add Request 13. See description for bit 0. |
| 12 | AR12 | R/WQ | 0h | Add Request 12. See description for bit 0. |
| 11 | AR11 | R/WQ | 0h | Add Request 11. See description for bit 0. |
| 10 | AR10 | R/WQ | 0h | Add Request 10. See description for bit 0. |
| 9 | AR9 | R/WQ | 0h | Add Request 9. See description for bit 0. |
| 8 | AR8 | R/WQ | 0h | Add Request 8. See description for bit 0. |
| 7 | AR7 | R/WQ | 0h | Add Request 7. See description for bit 0. |



Copyright © 2026 Texas Instruments Incorporated

**Table 32-65. MCAN_TXBAR Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 6 | AR6 | R/WQ | 0h | Add Request 6. See description for bit 0. |
| 5 | AR5 | R/WQ | 0h | Add Request 5. See description for bit 0. |
| 4 | AR4 | R/WQ | 0h | Add Request 4. See description for bit 0. |
| 3 | AR3 | R/WQ | 0h | Add Request 3. See description for bit 0. |
| 2 | AR2 | R/WQ | 0h | Add Request 2. See description for bit 0. |
| 1 | AR1 | R/WQ | 0h | Add Request 1. See description for bit 0. |
| 0 | AR0 | R/WQ | 0h | Add Request 0.<br>Each Tx Buffer has its own Add Request bit. Writing a '1' will set the corresponding Add Request bit; writing a '0' has no impact. This enables the Host to set transmission requests for multiple Tx Buffers with one write to TXBAR. TXBAR bits are set only for those Tx Buffers configured via TXBC. When no Tx scan is running, the bits are reset immediately, else the bits remain set until the Tx scan process has completed.<br>0 No transmission request added<br>1 Transmission requested added<br>Note: If an add request is applied for a Tx Buffer with pending transmission request (corresponding TXBRP bit already set), this add request is ignored.<br>Qualified Write is possible only with CCCR.CCE='0' |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.47 MCAN_TXBCR (Offset = 70D4h) [Reset = 00000000h]

MCAN_TXBCR is shown in Figure 32-73 and described in Table 32-66.

Return to the Summary Table.

MCAN Tx Buffer Cancellation Request

**Figure 32-73. MCAN_TXBCR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CR31 | CR30 | CR29 | CR28 | CR27 | CR26 | CR25 | CR24 |
| R/WQ-0h | R/WQ-0h | R/WQ-0h | R/WQ-0h | R/WQ-0h | R/WQ-0h | R/WQ-0h | R/WQ-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CR23 | CR22 | CR21 | CR20 | CR19 | CR18 | CR17 | CR16 |
| R/WQ-0h | R/WQ-0h | R/WQ-0h | R/WQ-0h | R/WQ-0h | R/WQ-0h | R/WQ-0h | R/WQ-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CR15 | CR14 | CR13 | CR12 | CR11 | CR10 | CR9 | CR8 |
| R/WQ-0h | R/WQ-0h | R/WQ-0h | R/WQ-0h | R/WQ-0h | R/WQ-0h | R/WQ-0h | R/WQ-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CR7 | CR6 | CR5 | CR4 | CR3 | CR2 | CR1 | CR0 |
| R/WQ-0h | R/WQ-0h | R/WQ-0h | R/WQ-0h | R/WQ-0h | R/WQ-0h | R/WQ-0h | R/WQ-0h |

**Table 32-66. MCAN_TXBCR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | CR31 | R/WQ | 0h | Cancellation Request 31. See description for bit 0. |
| 30 | CR30 | R/WQ | 0h | Cancellation Request 30. See description for bit 0. |
| 29 | CR29 | R/WQ | 0h | Cancellation Request 29. See description for bit 0. |
| 28 | CR28 | R/WQ | 0h | Cancellation Request 28. See description for bit 0. |
| 27 | CR27 | R/WQ | 0h | Cancellation Request 27. See description for bit 0. |
| 26 | CR26 | R/WQ | 0h | Cancellation Request 26. See description for bit 0. |
| 25 | CR25 | R/WQ | 0h | Cancellation Request 25. See description for bit 0. |
| 24 | CR24 | R/WQ | 0h | Cancellation Request 24. See description for bit 0. |
| 23 | CR23 | R/WQ | 0h | Cancellation Request 23. See description for bit 0. |
| 22 | CR22 | R/WQ | 0h | Cancellation Request 22. See description for bit 0. |
| 21 | CR21 | R/WQ | 0h | Cancellation Request 21. See description for bit 0. |
| 20 | CR20 | R/WQ | 0h | Cancellation Request 20. See description for bit 0. |
| 19 | CR19 | R/WQ | 0h | Cancellation Request 19. See description for bit 0. |
| 18 | CR18 | R/WQ | 0h | Cancellation Request 18. See description for bit 0. |
| 17 | CR17 | R/WQ | 0h | Cancellation Request 17. See description for bit 0. |
| 16 | CR16 | R/WQ | 0h | Cancellation Request 16. See description for bit 0. |
| 15 | CR15 | R/WQ | 0h | Cancellation Request 15. See description for bit 0. |
| 14 | CR14 | R/WQ | 0h | Cancellation Request 14. See description for bit 0. |
| 13 | CR13 | R/WQ | 0h | Cancellation Request 13. See description for bit 0. |
| 12 | CR12 | R/WQ | 0h | Cancellation Request 12. See description for bit 0. |
| 11 | CR11 | R/WQ | 0h | Cancellation Request 11. See description for bit 0. |
| 10 | CR10 | R/WQ | 0h | Cancellation Request 10. See description for bit 0. |
| 9 | CR9 | R/WQ | 0h | Cancellation Request 9. See description for bit 0. |
| 8 | CR8 | R/WQ | 0h | Cancellation Request 8. See description for bit 0. |
| 7 | CR7 | R/WQ | 0h | Cancellation Request 7. See description for bit 0. |



Copyright © 2026 Texas Instruments Incorporated

**Table 32-66. MCAN_TXBCR Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 6 | CR6 | R/WQ | 0h | Cancellation Request 6. See description for bit 0. |
| 5 | CR5 | R/WQ | 0h | Cancellation Request 5. See description for bit 0. |
| 4 | CR4 | R/WQ | 0h | Cancellation Request 4. See description for bit 0. |
| 3 | CR3 | R/WQ | 0h | Cancellation Request 3. See description for bit 0. |
| 2 | CR2 | R/WQ | 0h | Cancellation Request 2. See description for bit 0. |
| 1 | CR1 | R/WQ | 0h | Cancellation Request 1. See description for bit 0. |
| 0 | CR0 | R/WQ | 0h | Cancellation Request 0.<br>Each Tx Buffer has its own Cancellation Request bit. Writing a '1' will set the corresponding Cancellation Request bit; writing a '0' has no impact. This enables the Host to set cancellation requests for multiple Tx Buffers with one write to TXBCR. TXBCR bits are set only for those Tx Buffers configured via TXBC. The bits remain set until the corresponding bit of TXBRP is reset.<br>0 No cancellation pending<br>1 Cancellation pending<br>Qualified Write is possible only with CCCR.CCE='0' |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.48 MCAN_TXBTO (Offset = 70D8h) [Reset = 00000000h]

MCAN_TXBTO is shown in Figure 32-74 and described in Table 32-67.

Return to the Summary Table.

MCAN Tx Buffer Transmission Occurred

**Figure 32-74. MCAN_TXBTO**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| TO31 | TO30 | TO29 | TO28 | TO27 | TO26 | TO25 | TO24 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TO23 | TO22 | TO21 | TO20 | TO19 | TO18 | TO17 | TO16 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TO15 | TO14 | TO13 | TO12 | TO11 | TO10 | TO9 | TO8 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TO7 | TO6 | TO5 | TO4 | TO3 | TO2 | TO1 | TO0 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 32-67. MCAN_TXBTO Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | TO31 | R | 0h | Transmission Occurred 31. See description for bit 0. |
| 30 | TO30 | R | 0h | Transmission Occurred 30. See description for bit 0. |
| 29 | TO29 | R | 0h | Transmission Occurred 29. See description for bit 0. |
| 28 | TO28 | R | 0h | Transmission Occurred 28. See description for bit 0. |
| 27 | TO27 | R | 0h | Transmission Occurred 27. See description for bit 0. |
| 26 | TO26 | R | 0h | Transmission Occurred 26. See description for bit 0. |
| 25 | TO25 | R | 0h | Transmission Occurred 25. See description for bit 0. |
| 24 | TO24 | R | 0h | Transmission Occurred 24. See description for bit 0. |
| 23 | TO23 | R | 0h | Transmission Occurred 23. See description for bit 0. |
| 22 | TO22 | R | 0h | Transmission Occurred 22. See description for bit 0. |
| 21 | TO21 | R | 0h | Transmission Occurred 21. See description for bit 0. |
| 20 | TO20 | R | 0h | Transmission Occurred 20. See description for bit 0. |
| 19 | TO19 | R | 0h | Transmission Occurred 19. See description for bit 0. |
| 18 | TO18 | R | 0h | Transmission Occurred 18. See description for bit 0. |
| 17 | TO17 | R | 0h | Transmission Occurred 17. See description for bit 0. |
| 16 | TO16 | R | 0h | Transmission Occurred 16. See description for bit 0. |
| 15 | TO15 | R | 0h | Transmission Occurred 15. See description for bit 0. |
| 14 | TO14 | R | 0h | Transmission Occurred 14. See description for bit 0. |
| 13 | TO13 | R | 0h | Transmission Occurred 13. See description for bit 0. |
| 12 | TO12 | R | 0h | Transmission Occurred 12. See description for bit 0. |
| 11 | TO11 | R | 0h | Transmission Occurred 11. See description for bit 0. |
| 10 | TO10 | R | 0h | Transmission Occurred 10. See description for bit 0. |
| 9 | TO9 | R | 0h | Transmission Occurred 9. See description for bit 0. |
| 8 | TO8 | R | 0h | Transmission Occurred 8. See description for bit 0. |
| 7 | TO7 | R | 0h | Transmission Occurred 7. See description for bit 0. |



Copyright © 2026 Texas Instruments Incorporated

**Table 32-67. MCAN_TXBTO Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 6 | TO6 | R | 0h | Transmission Occurred 6. See description for bit 0. |
| 5 | TO5 | R | 0h | Transmission Occurred 5. See description for bit 0. |
| 4 | TO4 | R | 0h | Transmission Occurred 4. See description for bit 0. |
| 3 | TO3 | R | 0h | Transmission Occurred 3. See description for bit 0. |
| 2 | TO2 | R | 0h | Transmission Occurred 2. See description for bit 0. |
| 1 | TO1 | R | 0h | Transmission Occurred 1. See description for bit 0. |
| 0 | TO0 | R | 0h | Transmission Occurred 0.<br>Each Tx Buffer has its own Transmission Occurred bit. The bits are set when the corresponding TXBRP bit is cleared after a successful transmission. The bits are reset when a new transmission is requested by writing a '1' to the corresponding bit of register TXBAR.<br>0 No transmission occurred<br>1 Transmission occurred |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.49 MCAN_TXBCF (Offset = 70DCh) [Reset = 00000000h]

MCAN_TXBCF is shown in Figure 32-75 and described in Table 32-68.

Return to the Summary Table.

MCAN Tx Buffer Cancellation Finished

**Figure 32-75. MCAN_TXBCF**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CF31 | CF30 | CF29 | CF28 | CF27 | CF26 | CF25 | CF24 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CF23 | CF22 | CF21 | CF20 | CF19 | CF18 | CF17 | CF16 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CF15 | CF14 | CF13 | CF12 | CF11 | CF10 | CF9 | CF8 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CF7 | CF6 | CF5 | CF4 | CF3 | CF2 | CF1 | CF0 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 32-68. MCAN_TXBCF Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | CF31 | R | 0h | Cancellation Finished 31. See description for bit 0. |
| 30 | CF30 | R | 0h | Cancellation Finished 30. See description for bit 0. |
| 29 | CF29 | R | 0h | Cancellation Finished 29. See description for bit 0. |
| 28 | CF28 | R | 0h | Cancellation Finished 28. See description for bit 0. |
| 27 | CF27 | R | 0h | Cancellation Finished 27. See description for bit 0. |
| 26 | CF26 | R | 0h | Cancellation Finished 26. See description for bit 0. |
| 25 | CF25 | R | 0h | Cancellation Finished 25. See description for bit 0. |
| 24 | CF24 | R | 0h | Cancellation Finished 24. See description for bit 0. |
| 23 | CF23 | R | 0h | Cancellation Finished 23. See description for bit 0. |
| 22 | CF22 | R | 0h | Cancellation Finished 22. See description for bit 0. |
| 21 | CF21 | R | 0h | Cancellation Finished 21. See description for bit 0. |
| 20 | CF20 | R | 0h | Cancellation Finished 20. See description for bit 0. |
| 19 | CF19 | R | 0h | Cancellation Finished 19. See description for bit 0. |
| 18 | CF18 | R | 0h | Cancellation Finished 18. See description for bit 0. |
| 17 | CF17 | R | 0h | Cancellation Finished 17. See description for bit 0. |
| 16 | CF16 | R | 0h | Cancellation Finished 16. See description for bit 0. |
| 15 | CF15 | R | 0h | Cancellation Finished 15. See description for bit 0. |
| 14 | CF14 | R | 0h | Cancellation Finished 14. See description for bit 0. |
| 13 | CF13 | R | 0h | Cancellation Finished 13. See description for bit 0. |
| 12 | CF12 | R | 0h | Cancellation Finished 12. See description for bit 0. |
| 11 | CF11 | R | 0h | Cancellation Finished 11. See description for bit 0. |
| 10 | CF10 | R | 0h | Cancellation Finished 10. See description for bit 0. |
| 9 | CF9 | R | 0h | Cancellation Finished 9. See description for bit 0. |
| 8 | CF8 | R | 0h | Cancellation Finished 8. See description for bit 0. |
| 7 | CF7 | R | 0h | Cancellation Finished 7. See description for bit 0. |



Copyright © 2026 Texas Instruments Incorporated

**Table 32-68. MCAN_TXBCF Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 6 | CF6 | R | 0h | Cancellation Finished 6. See description for bit 0. |
| 5 | CF5 | R | 0h | Cancellation Finished 5. See description for bit 0. |
| 4 | CF4 | R | 0h | Cancellation Finished 4. See description for bit 0. |
| 3 | CF3 | R | 0h | Cancellation Finished 3. See description for bit 0. |
| 2 | CF2 | R | 0h | Cancellation Finished 2. See description for bit 0. |
| 1 | CF1 | R | 0h | Cancellation Finished 1. See description for bit 0. |
| 0 | CF0 | R | 0h | Cancellation Finished 0.<br>Each Tx Buffer has its own Cancellation Finished bit. The bits are set when the corresponding TXBRP bit is cleared after a cancellation was requested via TXBCR. In case the corresponding TXBRP bit was not set at the point of cancellation, CF is set immediately. The bits are reset when a new transmission is requested by writing a '1' to the corresponding bit of register TXBAR.<br>0 No transmit buffer cancellation<br>1 Transmit buffer cancellation finished |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.50 MCAN_TXBTIE (Offset = 70E0h) [Reset = 00000000h]

MCAN_TXBTIE is shown in Figure 32-76 and described in Table 32-69.

Return to the Summary Table.

MCAN Tx Buffer Transmission Interrupt Enable

**Figure 32-76. MCAN_TXBTIE**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| TIE31 | TIE30 | TIE29 | TIE28 | TIE27 | TIE26 | TIE25 | TIE24 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TIE23 | TIE22 | TIE21 | TIE20 | TIE19 | TIE18 | TIE17 | TIE16 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TIE15 | TIE14 | TIE13 | TIE12 | TIE11 | TIE10 | TIE9 | TIE8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TIE7 | TIE6 | TIE5 | TIE4 | TIE3 | TIE2 | TIE1 | TIE0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

**Table 32-69. MCAN_TXBTIE Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | TIE31 | R/W | 0h | Transmission Interrupt Enable 31. Each Tx Buffer has its own Transmission Interrupt Enable bit.<br>0 Transmission interrupt disabled<br>1 Transmission interrupt enable |
| 30 | TIE30 | R/W | 0h | Transmission Interrupt Enable 30. Each Tx Buffer has its own Transmission Interrupt Enable bit.<br>0 Transmission interrupt disabled<br>1 Transmission interrupt enable |
| 29 | TIE29 | R/W | 0h | Transmission Interrupt Enable 29. Each Tx Buffer has its own Transmission Interrupt Enable bit.<br>0 Transmission interrupt disabled<br>1 Transmission interrupt enable |
| 28 | TIE28 | R/W | 0h | Transmission Interrupt Enable 28. Each Tx Buffer has its own Transmission Interrupt Enable bit.<br>0 Transmission interrupt disabled<br>1 Transmission interrupt enable |
| 27 | TIE27 | R/W | 0h | Transmission Interrupt Enable 27. Each Tx Buffer has its own Transmission Interrupt Enable bit.<br>0 Transmission interrupt disabled<br>1 Transmission interrupt enable |
| 26 | TIE26 | R/W | 0h | Transmission Interrupt Enable 26. Each Tx Buffer has its own Transmission Interrupt Enable bit.<br>0 Transmission interrupt disabled<br>1 Transmission interrupt enable |
| 25 | TIE25 | R/W | 0h | Transmission Interrupt Enable 25. Each Tx Buffer has its own Transmission Interrupt Enable bit.<br>0 Transmission interrupt disabled<br>1 Transmission interrupt enable |
| 24 | TIE24 | R/W | 0h | Transmission Interrupt Enable 24. Each Tx Buffer has its own Transmission Interrupt Enable bit.<br>0 Transmission interrupt disabled<br>1 Transmission interrupt enable |



Copyright © 2026 Texas Instruments Incorporated

**Table 32-69. MCAN_TXBTIE Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 23 | TIE23 | R/W | 0h | Transmission Interrupt Enable 23. Each Tx Buffer has its own Transmission Interrupt Enable bit.<br>0 Transmission interrupt disabled<br>1 Transmission interrupt enable |
| 22 | TIE22 | R/W | 0h | Transmission Interrupt Enable 22. Each Tx Buffer has its own Transmission Interrupt Enable bit.<br>0 Transmission interrupt disabled<br>1 Transmission interrupt enable |
| 21 | TIE21 | R/W | 0h | Transmission Interrupt Enable 21. Each Tx Buffer has its own Transmission Interrupt Enable bit.<br>0 Transmission interrupt disabled<br>1 Transmission interrupt enable |
| 20 | TIE20 | R/W | 0h | Transmission Interrupt Enable 20. Each Tx Buffer has its own Transmission Interrupt Enable bit.<br>0 Transmission interrupt disabled<br>1 Transmission interrupt enable |
| 19 | TIE19 | R/W | 0h | Transmission Interrupt Enable 19. Each Tx Buffer has its own Transmission Interrupt Enable bit.<br>0 Transmission interrupt disabled<br>1 Transmission interrupt enable |
| 18 | TIE18 | R/W | 0h | Transmission Interrupt Enable 18. Each Tx Buffer has its own Transmission Interrupt Enable bit.<br>0 Transmission interrupt disabled<br>1 Transmission interrupt enable |
| 17 | TIE17 | R/W | 0h | Transmission Interrupt Enable 17. Each Tx Buffer has its own Transmission Interrupt Enable bit.<br>0 Transmission interrupt disabled<br>1 Transmission interrupt enable |
| 16 | TIE16 | R/W | 0h | Transmission Interrupt Enable 16. Each Tx Buffer has its own Transmission Interrupt Enable bit.<br>0 Transmission interrupt disabled<br>1 Transmission interrupt enable |
| 15 | TIE15 | R/W | 0h | Transmission Interrupt Enable 15. Each Tx Buffer has its own Transmission Interrupt Enable bit.<br>0 Transmission interrupt disabled<br>1 Transmission interrupt enable |
| 14 | TIE14 | R/W | 0h | Transmission Interrupt Enable 14. Each Tx Buffer has its own Transmission Interrupt Enable bit.<br>0 Transmission interrupt disabled<br>1 Transmission interrupt enable |
| 13 | TIE13 | R/W | 0h | Transmission Interrupt Enable 13. Each Tx Buffer has its own Transmission Interrupt Enable bit.<br>0 Transmission interrupt disabled<br>1 Transmission interrupt enable |
| 12 | TIE12 | R/W | 0h | Transmission Interrupt Enable 12. Each Tx Buffer has its own Transmission Interrupt Enable bit.<br>0 Transmission interrupt disabled<br>1 Transmission interrupt enable |
| 11 | TIE11 | R/W | 0h | Transmission Interrupt Enable 11. Each Tx Buffer has its own Transmission Interrupt Enable bit.<br>0 Transmission interrupt disabled<br>1 Transmission interrupt enable |
| 10 | TIE10 | R/W | 0h | Transmission Interrupt Enable 10. Each Tx Buffer has its own Transmission Interrupt Enable bit.<br>0 Transmission interrupt disabled<br>1 Transmission interrupt enable |



Copyright © 2026 Texas Instruments Incorporated

**Table 32-69. MCAN_TXBTIE Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 9 | TIE9 | R/W | 0h | Transmission Interrupt Enable 9. Each Tx Buffer has its own Transmission Interrupt Enable bit.<br>0 Transmission interrupt disabled<br>1 Transmission interrupt enable |
| 8 | TIE8 | R/W | 0h | Transmission Interrupt Enable 8. Each Tx Buffer has its own Transmission Interrupt Enable bit.<br>0 Transmission interrupt disabled<br>1 Transmission interrupt enable |
| 7 | TIE7 | R/W | 0h | Transmission Interrupt Enable 7. Each Tx Buffer has its own Transmission Interrupt Enable bit.<br>0 Transmission interrupt disabled<br>1 Transmission interrupt enable |
| 6 | TIE6 | R/W | 0h | Transmission Interrupt Enable 6. Each Tx Buffer has its own Transmission Interrupt Enable bit.<br>0 Transmission interrupt disabled<br>1 Transmission interrupt enable |
| 5 | TIE5 | R/W | 0h | Transmission Interrupt Enable 5. Each Tx Buffer has its own Transmission Interrupt Enable bit.<br>0 Transmission interrupt disabled<br>1 Transmission interrupt enable |
| 4 | TIE4 | R/W | 0h | Transmission Interrupt Enable 4. Each Tx Buffer has its own Transmission Interrupt Enable bit.<br>0 Transmission interrupt disabled<br>1 Transmission interrupt enable |
| 3 | TIE3 | R/W | 0h | Transmission Interrupt Enable 3. Each Tx Buffer has its own Transmission Interrupt Enable bit.<br>0 Transmission interrupt disabled<br>1 Transmission interrupt enable |
| 2 | TIE2 | R/W | 0h | Transmission Interrupt Enable 2. Each Tx Buffer has its own Transmission Interrupt Enable bit.<br>0 Transmission interrupt disabled<br>1 Transmission interrupt enable |
| 1 | TIE1 | R/W | 0h | Transmission Interrupt Enable 1. Each Tx Buffer has its own Transmission Interrupt Enable bit.<br>0 Transmission interrupt disabled<br>1 Transmission interrupt enable |
| 0 | TIE0 | R/W | 0h | Transmission Interrupt Enable 0. Each Tx Buffer has its own Transmission Interrupt Enable bit.<br>0 Transmission interrupt disabled<br>1 Transmission interrupt enable |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.51 MCAN_TXBCIE (Offset = 70E4h) [Reset = 00000000h]

MCAN_TXBCIE is shown in Figure 32-77 and described in Table 32-70.

Return to the Summary Table.

MCAN Tx Buffer Cancellation Finished Interrupt Enable

**Figure 32-77. MCAN_TXBCIE**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CFIE31 | CFIE30 | CFIE29 | CFIE28 | CFIE27 | CFIE26 | CFIE25 | CFIE24 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CFIE23 | CFIE22 | CFIE21 | CFIE20 | CFIE19 | CFIE18 | CFIE17 | CFIE16 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CFIE15 | CFIE14 | CFIE13 | CFIE12 | CFIE11 | CFIE10 | CFIE9 | CFIE8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CFIE7 | CFIE6 | CFIE5 | CFIE4 | CFIE3 | CFIE2 | CFIE1 | CFIE0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

**Table 32-70. MCAN_TXBCIE Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | CFIE31 | R/W | 0h | Cancellation Finished Interrupt Enable 31. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit.<br>0 Cancellation finished interrupt disabled<br>1 Cancellation finished interrupt enabled |
| 30 | CFIE30 | R/W | 0h | Cancellation Finished Interrupt Enable 30. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit.<br>0 Cancellation finished interrupt disabled<br>1 Cancellation finished interrupt enabled |
| 29 | CFIE29 | R/W | 0h | Cancellation Finished Interrupt Enable 29. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit.<br>0 Cancellation finished interrupt disabled<br>1 Cancellation finished interrupt enabled |
| 28 | CFIE28 | R/W | 0h | Cancellation Finished Interrupt Enable 28. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit.<br>0 Cancellation finished interrupt disabled<br>1 Cancellation finished interrupt enabled |
| 27 | CFIE27 | R/W | 0h | Cancellation Finished Interrupt Enable 27. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit.<br>0 Cancellation finished interrupt disabled<br>1 Cancellation finished interrupt enabled |
| 26 | CFIE26 | R/W | 0h | Cancellation Finished Interrupt Enable 26. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit.<br>0 Cancellation finished interrupt disabled<br>1 Cancellation finished interrupt enabled |
| 25 | CFIE25 | R/W | 0h | Cancellation Finished Interrupt Enable 25. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit.<br>0 Cancellation finished interrupt disabled<br>1 Cancellation finished interrupt enabled |
| 24 | CFIE24 | R/W | 0h | Cancellation Finished Interrupt Enable 24. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit.<br>0 Cancellation finished interrupt disabled<br>1 Cancellation finished interrupt enabled |



Copyright © 2026 Texas Instruments Incorporated

**Table 32-70. MCAN_TXBCIE Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 23 | CFIE23 | R/W | 0h | Cancellation Finished Interrupt Enable 23. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit.<br>0 Cancellation finished interrupt disabled<br>1 Cancellation finished interrupt enabled |
| 22 | CFIE22 | R/W | 0h | Cancellation Finished Interrupt Enable 22. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit.<br>0 Cancellation finished interrupt disabled<br>1 Cancellation finished interrupt enabled |
| 21 | CFIE21 | R/W | 0h | Cancellation Finished Interrupt Enable 21. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit.<br>0 Cancellation finished interrupt disabled<br>1 Cancellation finished interrupt enabled |
| 20 | CFIE20 | R/W | 0h | Cancellation Finished Interrupt Enable 20. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit.<br>0 Cancellation finished interrupt disabled<br>1 Cancellation finished interrupt enabled |
| 19 | CFIE19 | R/W | 0h | Cancellation Finished Interrupt Enable 19. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit.<br>0 Cancellation finished interrupt disabled<br>1 Cancellation finished interrupt enabled |
| 18 | CFIE18 | R/W | 0h | Cancellation Finished Interrupt Enable 18. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit.<br>0 Cancellation finished interrupt disabled<br>1 Cancellation finished interrupt enabled |
| 17 | CFIE17 | R/W | 0h | Cancellation Finished Interrupt Enable 17. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit.<br>0 Cancellation finished interrupt disabled<br>1 Cancellation finished interrupt enabled |
| 16 | CFIE16 | R/W | 0h | Cancellation Finished Interrupt Enable 16. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit.<br>0 Cancellation finished interrupt disabled<br>1 Cancellation finished interrupt enabled |
| 15 | CFIE15 | R/W | 0h | Cancellation Finished Interrupt Enable 15. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit.<br>0 Cancellation finished interrupt disabled<br>1 Cancellation finished interrupt enabled |
| 14 | CFIE14 | R/W | 0h | Cancellation Finished Interrupt Enable 14. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit.<br>0 Cancellation finished interrupt disabled<br>1 Cancellation finished interrupt enabled |
| 13 | CFIE13 | R/W | 0h | Cancellation Finished Interrupt Enable 13. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit.<br>0 Cancellation finished interrupt disabled<br>1 Cancellation finished interrupt enabled |
| 12 | CFIE12 | R/W | 0h | Cancellation Finished Interrupt Enable 12. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit.<br>0 Cancellation finished interrupt disabled<br>1 Cancellation finished interrupt enabled |
| 11 | CFIE11 | R/W | 0h | Cancellation Finished Interrupt Enable 11. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit.<br>0 Cancellation finished interrupt disabled<br>1 Cancellation finished interrupt enabled |
| 10 | CFIE10 | R/W | 0h | Cancellation Finished Interrupt Enable 10. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit.<br>0 Cancellation finished interrupt disabled<br>1 Cancellation finished interrupt enabled |



Copyright © 2026 Texas Instruments Incorporated

**Table 32-70. MCAN_TXBCIE Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 9 | CFIE9 | R/W | 0h | Cancellation Finished Interrupt Enable 9. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit.<br>0 Cancellation finished interrupt disabled<br>1 Cancellation finished interrupt enabled |
| 8 | CFIE8 | R/W | 0h | Cancellation Finished Interrupt Enable 8. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit.<br>0 Cancellation finished interrupt disabled<br>1 Cancellation finished interrupt enabled |
| 7 | CFIE7 | R/W | 0h | Cancellation Finished Interrupt Enable 7. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit.<br>0 Cancellation finished interrupt disabled<br>1 Cancellation finished interrupt enabled |
| 6 | CFIE6 | R/W | 0h | Cancellation Finished Interrupt Enable 6. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit.<br>0 Cancellation finished interrupt disabled<br>1 Cancellation finished interrupt enabled |
| 5 | CFIE5 | R/W | 0h | Cancellation Finished Interrupt Enable 5. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit.<br>0 Cancellation finished interrupt disabled<br>1 Cancellation finished interrupt enabled |
| 4 | CFIE4 | R/W | 0h | Cancellation Finished Interrupt Enable 4. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit.<br>0 Cancellation finished interrupt disabled<br>1 Cancellation finished interrupt enabled |
| 3 | CFIE3 | R/W | 0h | Cancellation Finished Interrupt Enable 3. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit.<br>0 Cancellation finished interrupt disabled<br>1 Cancellation finished interrupt enabled |
| 2 | CFIE2 | R/W | 0h | Cancellation Finished Interrupt Enable 2. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit.<br>0 Cancellation finished interrupt disabled<br>1 Cancellation finished interrupt enabled |
| 1 | CFIE1 | R/W | 0h | Cancellation Finished Interrupt Enable 1. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit.<br>0 Cancellation finished interrupt disabled<br>1 Cancellation finished interrupt enabled |
| 0 | CFIE0 | R/W | 0h | Cancellation Finished Interrupt Enable 0. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit.<br>0 Cancellation finished interrupt disabled<br>1 Cancellation finished interrupt enabled |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.52 MCAN_TXEFC (Offset = 70F0h) [Reset = 00000000h]

MCAN_TXEFC is shown in Figure 32-78 and described in Table 32-71.

Return to the Summary Table.

MCAN Tx Event FIFO Configuration

**Figure 32-78. MCAN_TXEFC**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | EFWM | | | | | |
| R-0h | | R/WQ-0h | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | EFS | | | | | |
| R-0h | | R/WQ-0h | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| EFSA | | | | | | | |
| R/WQ-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| EFSA | | | | | | RESERVED | |
| R/WQ-0h | | | | | | R-0h | |

**Table 32-71. MCAN_TXEFC Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | RESERVED | R | 0h | |
| 29-24 | EFWM | R/WQ | 0h | Event FIFO Watermark<br>0 Watermark interrupt disabled<br>1-32 Level for Tx Event FIFO watermark interrupt (IR.TEFW)<br>>32 Watermark interrupt disabled |
| 23-22 | RESERVED | R | 0h | |
| 21-16 | EFS | R/WQ | 0h | Event FIFO Size. The Tx Event FIFO elements are indexed from 0 to EFS - 1.<br>0 Tx Event FIFO disabled<br>1-32 Number of Tx Event FIFO elements<br>>32 Values greater than 32 are interpreted as 32 |
| 15-2 | EFSA | R/WQ | 0h | Event FIFO Start Address. Start address of Tx Event FIFO in Message RAM (32-bit word address). |
| 1-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.53 MCAN_TXEFS (Offset = 70F4h) [Reset = 00000000h]

MCAN_TXEFS is shown in Figure 32-79 and described in Table 32-72.

Return to the Summary Table.

MCAN Tx Event FIFO Status

**Figure 32-79. MCAN_TXEFS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | TEFL | EFF |
| R-0h | | | | | | R-0h | R-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | EFPI | | | | |
| R-0h | | | R-0h | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | EFGI | | | | |
| R-0h | | | R-0h | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | EFFL | | | | | |
| R-0h | | R-0h | | | | | |

**Table 32-72. MCAN_TXEFS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | RESERVED | R | 0h | |
| 25 | TEFL | R | 0h | Tx Event FIFO Element Lost. This bit is a copy of interrupt flag IR.TEFL. When IR.TEFL is reset, this bit is also reset.<br>0 No Tx Event FIFO element lost<br>1 Tx Event FIFO element lost, also set after write attempt to Tx Event FIFO of size zero. |
| 24 | EFF | R | 0h | Event FIFO Full<br>0 Tx Event FIFO not full<br>1 Tx Event FIFO full |
| 23-21 | RESERVED | R | 0h | |
| 20-16 | EFPI | R | 0h | Event FIFO Put Index. Tx Event FIFO write index pointer, range 0 to 31. |
| 15-13 | RESERVED | R | 0h | |
| 12-8 | EFGI | R | 0h | Event FIFO Get Index. Tx Event FIFO read index pointer, range 0 to 31. |
| 7-6 | RESERVED | R | 0h | |
| 5-0 | EFFL | R | 0h | Event FIFO Fill Level. Number of elements stored in Tx Event FIFO, range 0 to 32. |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.54 MCAN_TXEFA (Offset = 70F8h) [Reset = 00000000h]

MCAN_TXEFA is shown in Figure 32-80 and described in Table 32-73.

Return to the Summary Table.

MCAN Tx Event FIFO Acknowledge

**Figure 32-80. MCAN_TXEFA**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | EFAI | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | R/W-0h | | | | |

**Table 32-73. MCAN_TXEFA Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-5 | RESERVED | R | 0h | |
| 4-0 | EFAI | R/W | 0h | Event FIFO Acknowledge Index. After the Host has read an element or a sequence of elements from the Tx Event FIFO it has to write the index of the last element read from Tx Event FIFO to EFAI. This will set the Tx Event FIFO Get Index TXEFS.EFGI to EFAI + 1 and update the Event FIFO Fill Level TXEFS.EFFL. |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.55 MCANSS_PID (Offset = 7200h) [Reset = 68E04901h]

MCANSS_PID is shown in Figure 32-81 and described in Table 32-74.

Return to the Summary Table.

MCAN Subsystem Revision Register

**Figure 32-81. MCANSS_PID**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SCHEME | | | | MODULE_ID | | | | | | | | | | | |
| R-1h | | | | R-8E0h | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | | | MAJOR | | | | | MINOR | | | | | |
| | | | | | R-1h | | | | | R-1h | | | | | |

**Table 32-74. MCANSS_PID Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | SCHEME | R | 1h | PID Register Scheme |
| 27-16 | MODULE_ID | R | 8E0h | Module Identification Number |
| 10-8 | MAJOR | R | 1h | Major Revision of the MCAN Subsystem |
| 5-0 | MINOR | R | 1h | Minor Revision of the MCAN Subsystem |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.56 MCANSS_CTRL (Offset = 7204h) [Reset = 00000008h]

MCANSS_CTRL is shown in Figure 32-82 and described in Table 32-75.

Return to the Summary Table.

MCAN Subsystem Control Register

**Figure 32-82. MCANSS_CTRL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | EXT_TS_CNTR_EN | AUTOWAKEUP | WAKEUPREQEN | DBGSUSP_FREE | RESERVED | | |
| R-0h | R/W-0h | R/W-0h | R/W-0h | R/W-1h | R-0h | | |

**Table 32-75. MCANSS_CTRL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-7 | RESERVED | R | 0h | |
| 6 | EXT_TS_CNTR_EN | R/W | 0h | External Timestamp Counter Enable.<br>0 External timestamp counter disabled<br>1 External timestamp counter enabled |
| 5 | AUTOWAKEUP | R/W | 0h | Automatic Wakeup Enable. Enables the MCANSS to automatically clear the MCAN CCCR.INIT bit, fully waking the MCAN up, on an enabled wakeup request.<br>0 Disable the automatic write to CCCR.INIT<br>1 Enable the automatic write to CCCR.INIT |
| 4 | WAKEUPREQEN | R/W | 0h | Wakeup Request Enable. Enables the MCANSS to wakeup on CAN RXD activity.<br>0 Disable wakeup request<br>1 Enables wakeup request |
| 3 | DBGSUSP_FREE | R/W | 1h | Debug Suspend Free Bit. Enables debug suspend.<br>0 Honor debug suspend<br>1 Disregard debug suspend |
| 2-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.57 MCANSS_STAT (Offset = 7208h) [Reset = 0000000Xh]

MCANSS_STAT is shown in Figure 32-83 and described in Table 32-76.

Return to the Summary Table.

MCAN Subsystem Status Register

**Figure 32-83. MCANSS_STAT**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | ENABLE_FDOE | MEM_INIT_DONE | RESET |
| R-0h | | | | | R-X | R-0h | R-0h |

**Table 32-76. MCANSS_STAT Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | RESERVED | R | 0h | |
| 2 | ENABLE_FDOE | R | X | Enable FD (Flexible Data-Rate) Configuration<br>Reflects the value of mcanss_enable_fdoe configuration port,<br>-h = mcanss_enable_fdoe. |
| 1 | MEM_INIT_DONE | R | 0h | Memory Initialization Done.<br>0 Message RAM initialization is in progress<br>1 Message RAM is initialized for use |
| 0 | RESET | R | 0h | Soft Reset Status.<br>0 Not in reset<br>1 Reset is in progress |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.58 MCANSS_ICS (Offset = 720Ch) [Reset = 00000000h]

MCANSS_ICS is shown in Figure 32-84 and described in Table 32-77.

Return to the Summary Table.

MCAN Subsystem Interrupt Clear Shadow Register

**Figure 32-84. MCANSS_ICS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | EXT_TS_CNTR_OVFL |
| R-0h | | | | | | | R-0/W1C-0h |

**Table 32-77. MCANSS_ICS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | EXT_TS_CNTR_OVFL | R-0/W1C | 0h | External Timestamp Counter Overflow Interrupt Status Clear. Reads always return a 0.<br>0 Write of '0' has no effect<br>1 Write of '1' clears the MCANSS_IRS.EXT_TS_CNTR_OVFL bit |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.59 MCANSS_IRS (Offset = 7210h) [Reset = 00000000h]

MCANSS_IRS is shown in Figure 32-85 and described in Table 32-78.

Return to the Summary Table.

MCAN Subsystem Interrupt Raw Status Register

**Figure 32-85. MCANSS_IRS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | EXT_TS_CNTR_OVFL |
| R-0h | | | | | | | R/W1S-0h |

**Table 32-78. MCANSS_IRS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | EXT_TS_CNTR_OVFL | R/W1S | 0h | External Timestamp Counter Overflow Interrupt Status. This bit is set by HW or by a SW write of '1'. To clear, use the MCANSS_ICS.EXT_TS_CNTR_OVFL bit.<br>0 External timestamp counter has not overflowed<br>1 External timestamp counter has overflowed<br>When this bit is set to '1' by HW or SW, the MCANSS_EXT_TS_UNSERVICED_INTR_CNTR.EXT_TS_INTR_CNTR bit field will increment by 1. |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.60 MCANSS_IECS (Offset = 7214h) [Reset = 00000000h]

MCANSS_IECS is shown in Figure 32-86 and described in Table 32-79.

Return to the Summary Table.

MCAN Subsystem Interrupt Enable Clear Shadow Register

**Figure 32-86. MCANSS_IECS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | EXT_TS_CNTR_OVFL |
| R-0h | | | | | | | R-0/W1C-0h |

**Table 32-79. MCANSS_IECS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | EXT_TS_CNTR_OVFL | R-0/W1C | 0h | External Timestamp Counter Overflow Interrupt Enable Clear. Reads always return a 0.<br>0 Write of '0' has no effect<br>1 Write of '1' clears the MCANSS_IES.EXT_TS_CNTR_OVFL bit |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.61 MCANSS_IE (Offset = 7218h) [Reset = 00000000h]

MCANSS_IE is shown in Figure 32-87 and described in Table 32-80.

Return to the Summary Table.

MCAN Subsystem Interrupt Enable Register

**Figure 32-87. MCANSS_IE**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | EXT_TS_CNTR_OVFL |
| R-0h | | | | | | | R/W1S-0h |

**Table 32-80. MCANSS_IE Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | EXT_TS_CNTR_OVFL | R/W1S | 0h | External Timestamp Counter Overflow Interrupt Enable. A write of '0' has no effect. A write of '1' sets the MCANSS_IES.EXT_TS_CNTR_OVFL bit. |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.62 MCANSS_IES (Offset = 721Ch) [Reset = 00000000h]

MCANSS_IES is shown in Figure 32-88 and described in Table 32-81.

Return to the Summary Table.

MCAN Subsystem Interrupt Enable Status

**Figure 32-88. MCANSS_IES**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | EXT_TS_CNTR_OVFL |
| R-0h | | | | | | | R-0h |

**Table 32-81. MCANSS_IES Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | EXT_TS_CNTR_OVFL | R | 0h | External Timestamp Counter Overflow Interrupt Enable Status. To set, use the CANSS_IE.EXT_TS_CNTR_OVFL bit. To clear, use the MCANSS_IECS.EXT_TS_CNTR_OVFL bit.<br>0 External timestamp counter overflow interrupt is not enabled<br>1 External timestamp counter overflow interrupt is enabled |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.63 MCANSS_EOI (Offset = 7220h) [Reset = 00000000h]

MCANSS_EOI is shown in Figure 32-89 and described in Table 32-82.

Return to the Summary Table.

MCAN Subsystem End of Interrupt

**Figure 32-89. MCANSS_EOI**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | EOI | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R-0/W1S-0h | | | | | | | |

**Table 32-82. MCANSS_EOI Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | EOI | R-0/W1S | 0h | End of Interrupt. A write to this register will clear the associated interrupt. If the unserviced interrupt counter is > 1, another interrupt is generated.<br>0x00 External TS Interrupt is cleared<br>0x01 MCAN(0) interrupt is cleared<br>0x02 MCAN(1) interrupt is cleared<br>Other writes are ignored. |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.64 MCANSS_EXT_TS_PRESCALER (Offset = 7224h) [Reset = 00000000h]

MCANSS_EXT_TS_PRESCALER is shown in Figure 32-90 and described in Table 32-83.

Return to the Summary Table.

MCAN Subsystem External Timestamp Prescaler 0

**Figure 32-90. MCANSS_EXT_TS_PRESCALER**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | PRESCALER | | | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | R/W-0h | | | | | | | | | | | | | | | | | | | | | | | |

**Table 32-83. MCANSS_EXT_TS_PRESCALER Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | RESERVED | R | 0h | |
| 23-0 | PRESCALER | R/W | 0h | External Timestamp Prescaler Reload Value. The external timestamp count rate is the host (system) clock rate divided by this value, except in the case of 0. A zero value in this bit field will act identically to a value of 0x000001. |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.65 MCANSS_EXT_TS_UNSERVICED_INTR_CNTR (Offset = 7228h) [Reset = 00000000h]

MCANSS_EXT_TS_UNSERVICED_INTR_CNTR is shown in Figure 32-91 and described in Table 32-84.

Return to the Summary Table.

MCAN Subsystem External Timestamp Unserviced Interrupts Counter

**Figure 32-91. MCANSS_EXT_TS_UNSERVICED_INTR_CNTR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | EXT_TS_INTR_CNTR | | | | |
| R-0h | | | R-0h | | | | |

**Table 32-84. MCANSS_EXT_TS_UNSERVICED_INTR_CNTR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-5 | RESERVED | R | 0h | |
| 4-0 | EXT_TS_INTR_CNTR | R | 0h | External Timestamp Counter Unserviced Rollover Interrupts. If this value is > 1, an MCANSS_EOI write of '1' to bit 0 will issue another interrupt.<br>The status of this bit field is affected by the MCANSS_IRS.EXT_TS_CNTR_OVFL bit field. |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.66 MCANERR_REV (Offset = 7400h) [Reset = 66A0EA00h]

MCANERR_REV is shown in Figure 32-92 and described in Table 32-85.

Return to the Summary Table.

MCAN Error Aggregator Revision Register

**Figure 32-92. MCANERR_REV**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| SCHEME | | | | MODULE_ID | | | |
| R-1h | | | | R-6A0h | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MODULE_ID | | | | | | | |
| R-6A0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| | | | | | REVMAJ | | |
| | | | | | R-2h | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | REVMIN | | | | | |
| | | R-0h | | | | | |

**Table 32-85. MCANERR_REV Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | SCHEME | R | 1h | PID Register Scheme |
| 27-16 | MODULE_ID | R | 6A0h | Module Identification Number |
| 10-8 | REVMAJ | R | 2h | Major Revision of the Error Aggregator |
| 5-0 | REVMIN | R | 0h | Minor Revision of the Error Aggregator |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.67 MCANERR_VECTOR (Offset = 7408h) [Reset = 00000000h]

MCANERR_VECTOR is shown in Figure 32-93 and described in Table 32-86.

Return to the Summary Table.

Each error detection and correction (EDC) controller has a bank of error registers (offsets 0x10 - 0x3B) associated with it. These registers are accessed via an internal serial bus (SVBUS). To access them through the ECC aggregator the controller ID desired must be written to the ECC_VECTOR field, together with the RD_SVBUS trigger and RD_SVBUS_ADDRESS bit field. This initiates the serial read which consummates by setting the RD_SVBUS_DONE bit. At this point the addressed register may be read by a normal CPU read of the appropriate offset address.

**Figure 32-93. MCANERR_VECTOR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | RD_SVBUS_DONE |
| R-0h | | | | | | | R-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RD_SVBUS_ADDRESS | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RD_SVBUS | RESERVED | | | | ECC_VECTOR | | |
| R-0/W1S-0h | R-0h | | | | R/W-0h | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ECC_VECTOR | | | | | | | |
| R/W-0h | | | | | | | |

**Table 32-86. MCANERR_VECTOR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-25 | RESERVED | R | 0h | |
| 24 | RD_SVBUS_DONE | R | 0h | Read Completion Flag |
| 23-16 | RD_SVBUS_ADDRESS | R/W | 0h | Read Address Offset |
| 15 | RD_SVBUS | R-0/W1S | 0h | Read Trigger |
| 14-11 | RESERVED | R | 0h | |
| 10-0 | ECC_VECTOR | R/W | 0h | ECC RAM ID. Each error detection and correction (EDC) controller has a bank of error registers (offsets 0x10 - 0x3B) associated with it. These registers are accessed via an internal serial bus (SVBUS). To access them through the ECC aggregator the controller ID desired must be written to the ECC_VECTOR field, together with the RD_SVBUS trigger and RD_SVBUS_ADDRESS bit field. This initiates the serial read which consummates by setting the RD_SVBUS_DONE bit. At this point the addressed register may be read by a normal CPU read of the appropriate offset address.<br>0x000 Message RAM ECC controller is selected<br>Others Reserved (do not use)<br>Subsequent writes through the SVBUS (offsets 0x10 - 0x3B) have a delayed completion. To avoid conflicts, perform a read back of a register within this range after writing. |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.68 MCANERR_STAT (Offset = 740Ch) [Reset = 00000002h]

MCANERR_STAT is shown in Figure 32-94 and described in Table 32-87.

Return to the Summary Table.

MCAN Error Misc Status

**Figure 32-94. MCANERR_STAT**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | NUM_RAMS | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | R-2h | | | | | | | | | | |

**Table 32-87. MCANERR_STAT Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-11 | RESERVED | R | 0h | |
| 10-0 | NUM_RAMS | R | 2h | Number of RAMs. Number of ECC RAMs serviced by the aggregator. |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.69 MCANERR_WRAP_REV (Offset = 7410h) [Reset = 66A46A02h]

MCANERR_WRAP_REV is shown in Figure 32-95 and described in Table 32-88.

Return to the Summary Table.

This register is accessed through the ECC aggregator via an internal serial bus. To access, the appropriate procedure must be first followed in the MCAN ECC Vector Register.

**Figure 32-95. MCANERR_WRAP_REV**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| SCHEME | | | | MODULE_ID | | | |
| R-1h | | | | R-6A4h | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MODULE_ID | | | | | | | |
| R-6A4h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| | | | | | REVMAJ | | |
| | | | | | R-2h | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | REVMIN | | | | | |
| | | R-2h | | | | | |

**Table 32-88. MCANERR_WRAP_REV Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | SCHEME | R | 1h | PID Register Scheme |
| 27-16 | MODULE_ID | R | 6A4h | Module Identification Number |
| 10-8 | REVMAJ | R | 2h | Major Revision of the Error Aggregator |
| 5-0 | REVMIN | R | 2h | Minor Revision of the Error Aggregator |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.70 MCANERR_CTRL (Offset = 7414h) [Reset = 00000187h]

MCANERR_CTRL is shown in Figure 32-96 and described in Table 32-89.

Return to the Summary Table.

This register is accessed through the ECC aggregator via an internal serial bus. To access, the appropriate procedure must be first followed in the MCAN ECC Vector Register.

**Figure 32-96. MCANERR_CTRL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | CHECK_SVBUS_TIMEOUT |
| R-0h | | | | | | | R/W-1h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | ERROR_ONCE | FORCE_N_ROW | FORCE_DED | FORCE_SEC | ENABLE_RMW | ECC_CHECK | ECC_ENABLE |
| R-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-1h | R/W-1h | R/W-1h |

**Table 32-89. MCANERR_CTRL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-9 | RESERVED | R | 0h | |
| 8 | CHECK_SVBUS_TIMEOUT | R/W | 1h | Enables Serial VBUS timeout mechanism |
| 7 | RESERVED | R | 0h | |
| 6 | ERROR_ONCE | R/W | 0h | If this bit is set, the FORCE_SEC/FORCE_DED will inject an error to the specified row only once. The FORCE_SEC bit will be cleared once a writeback happens. If writeback is not enabled, this error will be cleared the cycle following the read when the data is corrected. For double-bit errors, the FORCE_DED bit will be cleared the cycle following the double-bit error. Any subsequent reads will not force an error. |
| 5 | FORCE_N_ROW | R/W | 0h | Enable single/double-bit error on the next RAM read, regardless of the MCANERR_ERR_CTRL1.ECC_ROW setting. For write through mode, this applies to writes as well as reads. |
| 4 | FORCE_DED | R/W | 0h | Force double-bit error. Cleared the cycle following the error if ERROR_ONCE is asserted. For write through mode, this applies to writes as well as reads. MCANERR_ERR_CTRL1 and MCANERR_ERR_CTRL2 should be configured prior to setting this bit. |
| 3 | FORCE_SEC | R/W | 0h | Force single-bit error. Cleared on a writeback or the cycle following the error if ERROR_ONCE is asserted. For write through mode, this applies to writes as well as reads. MCANERR_ERR_CTRL1 and MCANERR_ERR_CTRL2 should be configured prior to setting this bit. |
| 2 | ENABLE_RMW | R/W | 1h | Enable read-modify-write on partial word writes |
| 1 | ECC_CHECK | R/W | 1h | Enable ECC Check. ECC is completely bypassed if both ECC_ENABLE and ECC_CHECK are '0'. |
| 0 | ECC_ENABLE | R/W | 1h | Enable ECC Generation |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.71 MCANERR_ERR_CTRL1 (Offset = 7418h) [Reset = 00000000h]

MCANERR_ERR_CTRL1 is shown in Figure 32-97 and described in Table 32-90.

Return to the Summary Table.

This register is accessed through the ECC aggregator via an internal serial bus. To access, the appropriate procedure must be first followed in the MCAN ECC Vector Register.

**Figure 32-97. MCANERR_ERR_CTRL1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ECC_ROW | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 32-90. MCANERR_ERR_CTRL1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | ECC_ROW | R/W | 0h | Row address where FORCE_SEC or FORCE_DED needs to be applied. This is ignored if FORCE_N_ROW is set. |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.72 MCANERR_ERR_CTRL2 (Offset = 741Ch) [Reset = 00000000h]

MCANERR_ERR_CTRL2 is shown in Figure 32-98 and described in Table 32-91.

Return to the Summary Table.

This register is accessed through the ECC aggregator via an internal serial bus. To access, the appropriate procedure must be first followed in the MCAN ECC Vector Register.

**Figure 32-98. MCANERR_ERR_CTRL2**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ECC_BIT2 | | | | | | | | | | | | | | | | ECC_BIT1 | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | R/W-0h | | | | | | | | | | | | | | | |

**Table 32-91. MCANERR_ERR_CTRL2 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | ECC_BIT2 | R/W | 0h | Second column/data bit that needs to be flipped when FORCE_DED is set |
| 15-0 | ECC_BIT1 | R/W | 0h | Column/Data bit that needs to be flipped when FORCE_SEC or FORCE_DED is set |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.73 MCANERR_ERR_STAT1 (Offset = 7420h) [Reset = 00000000h]

MCANERR_ERR_STAT1 is shown in Figure 32-99 and described in Table 32-92.

Return to the Summary Table.

This register is accessed through the ECC aggregator via an internal serial bus. To access, the appropriate procedure must be first followed in the MCAN ECC Vector Register.

**Figure 32-99. MCANERR_ERR_STAT1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| ECC_BIT1 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| ECC_BIT1 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CLR_CTRL_REG_ERROR | RESERVED | | CLR_ECC_OTHER | CLR_ECC_DED | | CLR_ECC_SEC | |
| R/W1S-0h | R-0h | | R/W1C-0h | R/WD-0h | | R/WD-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CTRL_REG_ERROR | RESERVED | | ECC_OTHER | ECC_DED | | ECC_SEC | |
| R/W1S-0h | R-0h | | R/W1S-0h | R/WI-0h | | R/WI-0h | |

**Table 32-92. MCANERR_ERR_STAT1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | ECC_BIT1 | R | 0h | ECC Error Bit Position. Indicates the bit position in the RAM data that is in error on an SEC error. Only valid on an SEC error.<br>0 Bit 0 is in error<br>1 Bit 1 is in error<br>2 Bit 2 is in error<br>3 Bit 3 is in error<br>...<br>31 Bit 31 is in error<br>>32 Invalid |
| 15 | CLR_CTRL_REG_ERROR | R/W1S | 0h | Writing a '1' clears the CTRL_REG_ERROR bit |
| 14-13 | RESERVED | R | 0h | |
| 12 | CLR_ECC_OTHER | R/W1C | 0h | Writing a '1' clears the ECC_OTHER bit. |
| 11-10 | CLR_ECC_DED | R/WD | 0h | Clear ECC_DED. A write of a non-zero value to this bit field decrements the ECC_DED bit field by the value provided. |
| 9-8 | CLR_ECC_SEC | R/WD | 0h | Clear ECC_SEC. A write of a non-zero value to this bit field decrements the ECC_SEC bit field by the value provided. |
| 7 | CTRL_REG_ERROR | R/W1S | 0h | Control Register Error. A bit field in the control register is in an ambiguous state. This means that the redundancy registers have detected a state where not all values are the same and has defaulted to the reset state. S/W needs to re-write these registers to a known state. A write of 1 will set this interrupt flag. |
| 6-5 | RESERVED | R | 0h | |
| 4 | ECC_OTHER | R/W1S | 0h | SEC While Writeback Error Status<br>0 No SEC error while writeback pending<br>1 Indicates that successive single-bit errors have occurred while a writeback is still pending |



Copyright © 2026 Texas Instruments Incorporated

**Table 32-92. MCANERR_ERR_STAT1 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 3-2 | ECC_DED | R/WI | 0h | Double Bit Error Detected Status. A 2-bit saturating counter of the number of DED errors that have occurred since last cleared.<br>0 No double-bit error detected<br>1 One double-bit error was detected<br>2 Two double-bit errors were detected<br>3 Three double-bit errors were detected<br>A write of a non-zero value to this bit field increments it by the value provided. |
| 1-0 | ECC_SEC | R/WI | 0h | Single Bit Error Corrected Status. A 2-bit saturating counter of the number of SEC errors that have occurred since last cleared.<br>0 No single-bit error detected<br>1 One single-bit error was detected and corrected<br>2 Two single-bit errors were detected and corrected<br>3 Three single-bit errors were detected and corrected<br>A write of a non-zero value to this bit field increments it by the value provided. |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.74 MCANERR_ERR_STAT2 (Offset = 7424h) [Reset = 00000000h]

MCANERR_ERR_STAT2 is shown in Figure 32-100 and described in Table 32-93.

Return to the Summary Table.

This register is accessed through the ECC aggregator via an internal serial bus. To access, the appropriate procedure must be first followed in the MCAN ECC Vector Register.

**Figure 32-100. MCANERR_ERR_STAT2**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ECC_ROW | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 32-93. MCANERR_ERR_STAT2 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | ECC_ROW | R | 0h | Indicates the row address where the single or double-bit error occurred. This value is address offset/4. |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.75 MCANERR_ERR_STAT3 (Offset = 7428h) [Reset = 00000000h]

MCANERR_ERR_STAT3 is shown in Figure 32-101 and described in Table 32-94.

Return to the Summary Table.

This register is accessed through the ECC aggregator via an internal serial bus. To access, the appropriate procedure must be first followed in the MCAN ECC Vector Register.

**Figure 32-101. MCANERR_ERR_STAT3**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | CLR_SVBUS_TIMEOUT | RESERVED |
| R-0h | | | | | | R-0/W1C-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | SVBUS_TIMEOUT | WB_PEND |
| R-0h | | | | | | R-0/W1S-0h | R-0h |

**Table 32-94. MCANERR_ERR_STAT3 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-10 | RESERVED | R | 0h | |
| 9 | CLR_SVBUS_TIMEOUT | R-0/W1C | 0h | Write 1 to clear the Serial VBUS Timeout Flag |
| 8-2 | RESERVED | R | 0h | |
| 1 | SVBUS_TIMEOUT | R-0/W1S | 0h | Serial VBUS Timeout Flag. Write 1 to set. |
| 0 | WB_PEND | R | 0h | Delayed Write Back Pending Status<br>0 No write back pending<br>1 An ECC data correction write back is pending |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.76 MCANERR_SEC_EOI (Offset = 743Ch) [Reset = 00000000h]

MCANERR_SEC_EOI is shown in Figure 32-102 and described in Table 32-95.

Return to the Summary Table.

MCAN Single Error Corrected End of Interrupt Register

**Figure 32-102. MCANERR_SEC_EOI**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | EOI_WR |
| R-0h | | | | | | | R-0/W1S-0h |

**Table 32-95. MCANERR_SEC_EOI Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | EOI_WR | R-0/W1S | 0h | Write to this register indicates that software has acknowledged the pending interrupt and the next interrupt can be sent to the host. Note that a write to the MCANERR_ERR_STAT1.CLR_ECC_SEC goes through the SVBUS and has a delayed completion. To avoid an additional interrupt, read the MCANERR_ERR_STAT1 register back prior to writing to this bit field. |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.77 MCANERR_SEC_STATUS (Offset = 7440h) [Reset = 00000000h]

MCANERR_SEC_STATUS is shown in Figure 32-103 and described in Table 32-96.

Return to the Summary Table.

MCAN Single Error Corrected Interrupt Status Register

**Figure 32-103. MCANERR_SEC_STATUS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | MSGMEM_PEND |
| R-0h | | | | | | | R-0-0h |

**Table 32-96. MCANERR_SEC_STATUS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | MSGMEM_PEND | R-0 | 0h | Message RAM SEC Interrupt Pending<br>0 No SEC interrupt is pending<br>1 SEC interrupt is pending |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.78 MCANERR_SEC_ENABLE_SET (Offset = 7480h) [Reset = 00000000h]

MCANERR_SEC_ENABLE_SET is shown in Figure 32-104 and described in Table 32-97.

Return to the Summary Table.

MCAN Single Error Corrected Interrupt Enable Set Register

**Figure 32-104. MCANERR_SEC_ENABLE_SET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | MSGMEM_ENABLE_SET |
| R-0h | | | | | | | R/W1S-0h |

**Table 32-97. MCANERR_SEC_ENABLE_SET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | MSGMEM_ENABLE_SET | R/W1S | 0h | Message RAM SEC Interrupt Pending Enable Set. Writing a 1 to this bit enables the Message RAM SEC error interrupts. Writing a 0 has no effect. Reads return the corresponding enable bit's current value. |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.79 MCANERR_SEC_ENABLE_CLR (Offset = 74C0h) [Reset = 00000000h]

MCANERR_SEC_ENABLE_CLR is shown in Figure 32-105 and described in Table 32-98.

Return to the Summary Table.

MCAN Single Error Corrected Interrupt Enable Clear Register

**Figure 32-105. MCANERR_SEC_ENABLE_CLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | MSGMEM_ENABLE_CLR |
| R-0h | | | | | | | R/W1C-0h |

**Table 32-98. MCANERR_SEC_ENABLE_CLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | MSGMEM_ENABLE_CLR | R/W1C | 0h | Message RAM SEC Interrupt Pending Enable Clear. Writing a 1 to this bit disables the Message RAM SEC error interrupts. Writing a 0 has no effect. Reads return the corresponding enable bit's current value. |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.80 MCANERR_DED_EOI (Offset = 753Ch) [Reset = 00000000h]

MCANERR_DED_EOI is shown in Figure 32-106 and described in Table 32-99.

Return to the Summary Table.

MCAN Double Error Detected End of Interrupt Register

**Figure 32-106. MCANERR_DED_EOI**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | EOI_WR |
| R-0h | | | | | | | R-0/W1S-0h |

**Table 32-99. MCANERR_DED_EOI Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | EOI_WR | R-0/W1S | 0h | Write to this register indicates that software has acknowledged the pending interrupt and the next interrupt can be sent to the host. Note that a write to the MCANERR_ERR_STAT1.CLR_ECC_DED goes through the SVBUS and has a delayed completion. To avoid an additional interrupt, read the MCANERR_ERR_STAT1 register back prior to writing to this bit field. |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.81 MCANERR_DED_STATUS (Offset = 7540h) [Reset = 00000000h]

MCANERR_DED_STATUS is shown in Figure 32-107 and described in Table 32-100.

Return to the Summary Table.

MCAN Double Error Detected Interrupt Status Register

**Figure 32-107. MCANERR_DED_STATUS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | MSGMEM_PEND |
| R-0h | | | | | | | R-0-0h |

**Table 32-100. MCANERR_DED_STATUS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | MSGMEM_PEND | R-0 | 0h | Message RAM DED Interrupt Pending<br>0 No DED interrupt is pending<br>1 DED interrupt is pending |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.82 MCANERR_DED_ENABLE_SET (Offset = 7580h) [Reset = 00000000h]

MCANERR_DED_ENABLE_SET is shown in Figure 32-108 and described in Table 32-101.

Return to the Summary Table.

MCAN Double Error Detected Interrupt Enable Set Register

**Figure 32-108. MCANERR_DED_ENABLE_SET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | MSGMEM_ENABLE_SET |
| R-0h | | | | | | | R/W1S-0h |

**Table 32-101. MCANERR_DED_ENABLE_SET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | MSGMEM_ENABLE_SET | R/W1S | 0h | Message RAM DED Interrupt Pending Enable Set. Writing a 1 to this bit enables the Message RAM DED error interrupts. Writing a 0 has no effect. Reads return the corresponding enable bit's current value. |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.83 MCANERR_DED_ENABLE_CLR (Offset = 75C0h) [Reset = 00000000h]

MCANERR_DED_ENABLE_CLR is shown in Figure 32-109 and described in Table 32-102.

Return to the Summary Table.

MCAN Double Error Detected Interrupt Enable Clear Register

**Figure 32-109. MCANERR_DED_ENABLE_CLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | MSGMEM_ENABLE_CLR |
| R-0h | | | | | | | R/W1C-0h |

**Table 32-102. MCANERR_DED_ENABLE_CLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | MSGMEM_ENABLE_CLR | R/W1C | 0h | Message RAM DED Interrupt Pending Enable Clear. Writing a 1 to this bit disables the Message RAM DED error interrupts. Writing a 0 has no effect. Reads return the corresponding enable bit's current value. |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.84 MCANERR_AGGR_ENABLE_SET (Offset = 7600h) [Reset = 00000000h]

MCANERR_AGGR_ENABLE_SET is shown in Figure 32-110 and described in Table 32-103.

Return to the Summary Table.

MCAN Error Aggregator Enable Set Register

**Figure 32-110. MCANERR_AGGR_ENABLE_SET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | ENABLE_TIMEOUT_SET | ENABLE_PARITY_SET |
| R-0h | | | | | | R/W1S-0h | R/W1S-0h |

**Table 32-103. MCANERR_AGGR_ENABLE_SET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | |
| 1 | ENABLE_TIMEOUT_SET | R/W1S | 0h | Write 1 to enable timeout errors. Reads return the corresponding enable bit's current value. |
| 0 | ENABLE_PARITY_SET | R/W1S | 0h | Write 1 to enable parity errors. Reads return the corresponding enable bit's current value. |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.85 MCANERR_AGGR_ENABLE_CLR (Offset = 7604h) [Reset = 00000000h]

MCANERR_AGGR_ENABLE_CLR is shown in Figure 32-111 and described in Table 32-104.

Return to the Summary Table.

MCAN Error Aggregator Enable Clear Register

**Figure 32-111. MCANERR_AGGR_ENABLE_CLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | ENABLE_TIMEOUT_CLR | ENABLE_PARITY_CLR |
| R-0h | | | | | | R/W1C-0h | R/W1C-0h |

**Table 32-104. MCANERR_AGGR_ENABLE_CLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | |
| 1 | ENABLE_TIMEOUT_CLR | R/W1C | 0h | Write 1 to disable timeout errors. Reads return the corresponding enable bit's current value. |
| 0 | ENABLE_PARITY_CLR | R/W1C | 0h | Write 1 to disable parity errors. Reads return the corresponding enable bit's current value. |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.86 MCANERR_AGGR_STATUS_SET (Offset = 7608h) [Reset = 00000000h]

MCANERR_AGGR_STATUS_SET is shown in Figure 32-112 and described in Table 32-105.

Return to the Summary Table.

MCAN Error Aggregator Status Set Register

**Figure 32-112. MCANERR_AGGR_STATUS_SET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | SVBUS_TIMEOUT | | AGGR_PARITY_ERR | |
| R-0h | | | | R/WI-0h | | R/WI-0h | |

**Table 32-105. MCANERR_AGGR_STATUS_SET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3-2 | SVBUS_TIMEOUT | R/WI | 0h | Aggregator Serial VBUS Timeout Error Status<br>2-bit saturating counter of the number of SVBUS timeout errors that have occurred since last cleared.<br>0 No timeout errors have occurred<br>1 One timeout error has occurred<br>2 Two timeout errors have occurred<br>3 Three timeout errors have occurred<br>A write of a non-zero value to this bit field increments it by the value provided. |
| 1-0 | AGGR_PARITY_ERR | R/WI | 0h | Aggregator Parity Error Status<br>2-bit saturating counter of the number of parity errors that have occurred since last cleared.<br>0 No parity errors have occurred<br>1 One parity error has occurred<br>2 Two parity errors have occurred<br>3 Three parity errors have occurred<br>A write of a non-zero value to this bit field increments it by the value provided. |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.87 MCANERR_AGGR_STATUS_CLR (Offset = 760Ch) [Reset = 00000000h]

MCANERR_AGGR_STATUS_CLR is shown in Figure 32-113 and described in Table 32-106.

Return to the Summary Table.

MCAN Error Aggregator Status Clear Register

**Figure 32-113. MCANERR_AGGR_STATUS_CLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | SVBUS_TIMEOUT | | AGGR_PARITY_ERR | |
| R-0h | | | | R/WD-0h | | R/WD-0h | |

**Table 32-106. MCANERR_AGGR_STATUS_CLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3-2 | SVBUS_TIMEOUT | R/WD | 0h | Aggregator Serial VBUS Timeout Error Status<br>2-bit saturating counter of the number of SVBUS timeout errors that have occurred since last cleared.<br>0 No timeout errors have occurred<br>1 One timeout error has occurred<br>2 Two timeout errors have occurred<br>3 Three timeout errors have occurred<br>A write of a non-zero value to this bit field decrements it by the value provided. |
| 1-0 | AGGR_PARITY_ERR | R/WD | 0h | Aggregator Parity Error Status<br>2-bit saturating counter of the number of parity errors that have occurred since last cleared.<br>0 No parity errors have occurred<br>1 One parity error has occurred<br>2 Two parity errors have occurred<br>3 Three parity errors have occurred<br>A write of a non-zero value to this bit field decrements it by the value provided. |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.88 IIDX (Offset = 7820h) [Reset = 00000000h]

IIDX is shown in Figure 32-114 and described in Table 32-107.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. Value 0x00 means no event pending. Interrupt 1 is the highest priority, IIDX next highest, 4, 8, … IIDX^31 is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred. On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in RIS and MIS are cleared as well. After a read from the CPU (not from the debug interface), the register is updated with the next highest priority interrupt, if none are pending, then it should display 0x0.

**Figure 32-114. IIDX**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | STAT | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | |

**Table 32-107. IIDX Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | STAT | R | 0h | Interrupt index status<br>00h = No interrupt pending.<br>1h = MCAN Interrupt Line 0 interrupt pending.<br>2h = MCAN Interrupt Line 1 interrupt pending.<br>3h = Message RAM SEC (Single Error Correction) interrupt pending.<br>4h = Message RAM DED (Double Error Detection) interrupt pending.<br>5h = External Timestamp Counter Overflow interrupt pending.<br>6h = Clock Stop Wake Up interrupt pending. |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.89 IMASK (Offset = 7828h) [Reset = 00000000h]

IMASK is shown in Figure 32-115 and described in Table 32-108.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

**Figure 32-115. IMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | WAKEUP | EXT_TS_CNTR_OVFL | DED | SEC | INTL1 | INTL0 |
| R-0h | | -0 | -0 | -0 | -0 | -0 | -0 |

**Table 32-108. IMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | RESERVED | R | 0h | |
| 5 | WAKEUP | R/W | 0h | Clock Stop Wake Up interrupt mask.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 4 | EXT_TS_CNTR_OVFL | R/W | 0h | External Timestamp Counter Overflow interrupt mask.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 3 | DED | R/W | 0h | Massage RAM DED interrupt mask.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 2 | SEC | R/W | 0h | Message RAM SEC interrupt mask.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 1 | INTL1 | R/W | 0h | MCAN Interrupt Line 1 mask.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 0 | INTL0 | R/W | 0h | MCAN Interrupt Line 0 mask.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.90 RIS (Offset = 7830h) [Reset = 00000000h]

RIS is shown in Figure 32-116 and described in Table 32-109.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Figure 32-116. RIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | WAKEUP | EXT_TS_CNTR_OVFL | DED | SEC | INTL1 | INTL0 |
| R-0h | | -0 | -0 | -0 | -0 | -0 | -0 |

**Table 32-109. RIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | RESERVED | R | 0h | |
| 5 | WAKEUP | R | 0h | Clock Stop Wake Up interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 4 | EXT_TS_CNTR_OVFL | R | 0h | External Timestamp Counter Overflow interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 3 | DED | R | 0h | Message RAM DED interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 2 | SEC | R | 0h | Message RAM SEC interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 1 | INTL1 | R | 0h | MCAN Interrupt Line 1.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 0 | INTL0 | R | 0h | MCAN Interrupt Line 0.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.91 MIS (Offset = 7838h) [Reset = 00000000h]

MIS is shown in Figure 32-117 and described in Table 32-110.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 32-117. MIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | WAKEUP | EXT_TS_CNTR_OVFL | DED | SEC | INTL1 | INTL0 |
| R-0h | | -0 | -0 | -0 | -0 | -0 | -0 |

**Table 32-110. MIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | RESERVED | R | 0h | |
| 5 | WAKEUP | R | 0h | Masked Clock Stop Wake Up interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 4 | EXT_TS_CNTR_OVFL | R | 0h | Masked External Timestamp Counter Overflow interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 3 | DED | R | 0h | Masked Message RAM DED interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 2 | SEC | R | 0h | Masked Message RAM SEC interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 1 | INTL1 | R | 0h | Masked MCAN Interrupt Line 1.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 0 | INTL0 | R | 0h | Masked MCAN Interrupt Line 0.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.92 ISET (Offset = 7840h) [Reset = 00000000h]

ISET is shown in Figure 32-118 and described in Table 32-111.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 32-118. ISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | WAKEUP | EXT_TS_CNTR_OVFL | DED | SEC | INTL1 | INTL0 |
| R-0h | | -0 | -0 | -0 | -0 | -0 | -0 |

**Table 32-111. ISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | RESERVED | R | 0h | |
| 5 | WAKEUP | W | 0h | Set Clock Stop Wake Up interrupt.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 4 | EXT_TS_CNTR_OVFL | W | 0h | Set External Timestamp Counter Overflow interrupt.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 3 | DED | W | 0h | Set Message RAM DED interrupt.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 2 | SEC | W | 0h | Set Message RAM SEC interrupt.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 1 | INTL1 | W | 0h | Set MCAN Interrupt Line 1.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 0 | INTL0 | W | 0h | Set MCAN Interrupt Line 0.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.93 ICLR (Offset = 7848h) [Reset = 00000000h]

ICLR is shown in Figure 32-119 and described in Table 32-112.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

**Figure 32-119. ICLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | WAKEUP | EXT_TS_CNTR_OVFL | DED | SEC | INTL1 | INTL0 |
| R-0h | | -0 | -0 | -0 | -0 | -0 | -0 |

**Table 32-112. ICLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | RESERVED | R | 0h | |
| 5 | WAKEUP | W | 0h | Clear Clock Stop Wake Up interrupt.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 4 | EXT_TS_CNTR_OVFL | W | 0h | Clear External Timestamp Counter Overflow interrupt.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 3 | DED | W | 0h | Clear Message RAM DED interrupt.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 2 | SEC | W | 0h | Clear Message RAM SEC interrupt.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 1 | INTL1 | W | 0h | Clear MCAN Interrupt Line 1.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 0 | INTL0 | W | 0h | Clear MCAN Interrupt Line 0.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.94 EVT_MODE (Offset = 78E0h) [Reset = 00000000h]

EVT_MODE is shown in Figure 32-120 and described in Table 32-113.

Return to the Summary Table.

Event mode register. It is used to select whether each line is disabled, in software mode (software clears the RIS) or in hardware mode (hardware clears the RIS)

**Figure 32-120. EVT_MODE**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | INT0_CFG | |
| R-0h | | | | | | R-0h | |

**Table 32-113. EVT_MODE Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | |
| 1-0 | INT0_CFG | R | 0h | Event line mode select for event corresponding to [IPSTANDARD.CPU_INT]<br>0h = The interrupt or event line is disabled.<br>1h = The interrupt or event line is in software mode. Software must clear the RIS.<br>2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag. |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.95 DESC (Offset = 78FCh) [Reset = 00000000h]

DESC is shown in Figure 32-121 and described in Table 32-114.

Return to the Summary Table.

This register identifies the peripheral and its exact version.

**Figure 32-121. DESC**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MODULEID | | | | | | | | | | | | | | | |
| -0 | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| FEATUREVER | | | | RESERVED | | | | MAJREV | | | | MINREV | | | |
| -0 | | | | R-0h | | | | -0 | | | | -0 | | | |

**Table 32-114. DESC Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | MODULEID | R | 0x0 | Module identification contains a unique peripheral identification number. The assignments are maintained in a central database for all of the platform modules to ensure uniqueness.<br>0h = Smallest value<br>FFFFh = Highest possible value |
| 15-12 | FEATUREVER | R | 0x0 | Feature Set for the module *instance*<br>0h = MCAN module with CAN-FD mode enabled<br>1h = MCAN module with CAN-FD mode disabled |
| 11-8 | RESERVED | R | 0h | |
| 7-4 | MAJREV | R | 0x0 | Major rev of the IP<br>0h = Smallest value<br>Fh = Highest possible value |
| 3-0 | MINREV | R | 0x0 | Minor rev of the IP<br>0h = Smallest value<br>Fh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.96 MCANSS_CLKEN (Offset = 7900h) [Reset = 00000000h]

MCANSS_CLKEN is shown in Figure 32-122 and described in Table 32-115.

Return to the Summary Table.

MCAN module clock (functional clock and Vbusp to access MCAN module MMRs) enable register
<Internal note> This IP-specific MMR itself is not clock gated, as long as IP is enabled by the paper-spin configuration

**Figure 32-122. MCANSS_CLKEN**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | CLK_REQEN |
| R-0h | | | | | | | -0 |

**Table 32-115. MCANSS_CLKEN Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | CLK_REQEN | RW | 0x0 | MCAN functional and MCAN/MCANSS MMR clock request enable bit<br>0h = MCAN module functional clock and Vbusp is not requested. These clocks are gated to the MCAN module.<br>1h = Setting this bit requests MCAN module functional clock and Vbusp.<br>These clocks are not gated to MCAN module. |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.97 MCANSS_CLKDIV (Offset = 7904h) [Reset = 00000000h]

MCANSS_CLKDIV is shown in Figure 32-123 and described in Table 32-116.

Return to the Summary Table.

Needs to go to the Management aperture once available

**Figure 32-123. MCANSS_CLKDIV**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | | | RATIO | |
| R-0h | | | | | | | | | | | | | | R/W-0h | |

**Table 32-116. MCANSS_CLKDIV Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | |
| 1-0 | RATIO | R/W | 0x0 | Clock divide ratio specification. Enables configuring clock divide settings for the MCAN functional clock input to the MCAN-SS.<br>0h (R/W) = Divides input clock by 1<br>1h (R/W) = Divides input clock by 2<br>2h (R/W) = Divides input clock by 4<br>3h (R/W) = Divides input clock by 1 |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.98 MCANSS_CLKCTL (Offset = 7908h) [Reset = 00000000h]

MCANSS_CLKCTL is shown in Figure 32-124 and described in Table 32-117.

Return to the Summary Table.

MCANSS clock stop control MMR.
<Internal note> Bus clock for the wrapper MMRs (including this MMR) is not gated by this register.

**Figure 32-124. MCANSS_CLKCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | WKUP_GLTFLT_EN |
| R-0h | | | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | WAKEUP_INT_EN | RESERVED | | | STOPREQ |
| R-0h | | | R/W-0h | R-0h | | | -0 |

**Table 32-117. MCANSS_CLKCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-9 | RESERVED | R | 0h | |
| 8 | WKUP_GLTFLT_EN | R/W | 0h | Setting this bit enables the glitch filter on MCAN RXD input, which wakes up the MCAN controller to exit clock gating.<br>0h = Disable glitch filter enable on RXD input when MCAN is in clock stop mode (waiting for event on RXD input for clock stop wakeup).<br>1h = Enable glitch filter enable on RXD input when MCAN is in clock stop mode (waiting for event on RXD input for clock stop wakeup). |
| 7-5 | RESERVED | R | 0h | |
| 4 | WAKEUP_INT_EN | R/W | 0h | This bit controls enabling or disabling the MCAN IP clock stop wakeup interrupt (when MCANSS_CTRL.WAKEUPREQEN wakeup request is enabled to wakeup MCAN IP upon CAN RXD activity)<br>0h = Disable MCAN IP clock stop wakeup interrupt<br>1h = Enable MCAN IP clock stop wakeup interrupt |
| 3-1 | RESERVED | R | 0h | |
| 0 | STOPREQ | R/W | 0h | This bit is used to enable/disable MCAN clock (both host clock and functional clock) gating request.<br>Note: This bit can be reset by HW by Clock-Stop Wake-up via CAN RX Activity. See spec for more details.<br>0h = Disable MCAN-SS clock stop request<br>1h = Enable MCAN-SS clock stop request |



Copyright © 2026 Texas Instruments Incorporated

### 32.7.99 MCANSS_CLKSTS (Offset = 790Ch) [Reset = 00000000h]

MCANSS_CLKSTS is shown in Figure 32-125 and described in Table 32-118.

Return to the Summary Table.

MCANSS clock stop status register to indicate status of clock stop mechanism

**Figure 32-125. MCANSS_CLKSTS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | CCLKDONE |
| R-0h | | | | | | | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | STOPREQ_HW_OVR | RESERVED | | | CLKSTOP_ACKSTS |
| R-0h | | | R-0h | R-0h | | | R-0h |

**Table 32-118. MCANSS_CLKSTS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-9 | RESERVED | R | 0h | |
| 8 | CCLKDONE | R | 0h | This bit indicates the status of MCAN controller clock request from GPRCM.<br>0h = MCAN controller clock is not available to the MCAN IP.<br>1h = MCAN controller clock is enabled and available to the MCAN IP. |
| 7-5 | RESERVED | R | 0h | |
| 4 | STOPREQ_HW_OVR | R | 0h | MCANSS clock stop HW override status bit.<br>This bit indicates when the MCANSS_CLKCTL.STOPREQ bit has been cleared by HW when a clock-stop wake-up event via CAN RX activity is triggered.<br>0h = MCANSS_CLKCTL.STOPREQ bit has not been cleared by HW.<br>1h = MCANSS_CLKCTL.STOPREQ bit has been cleared by HW. |
| 3-1 | RESERVED | R | 0h | |
| 0 | CLKSTOP_ACKSTS | R | 0h | Clock stop acknowledge status from MCAN IP<br>0h = No clock stop acknowledged.<br>1h = Clock stop has been acknowledged by MCAN IP; MCAN-SS may be clock gated by stopping both the CAN host and functional clocks. |



Copyright © 2026 Texas Instruments Incorporated

Chapter 33

# USB

**33.1 USB Overview** ..... 1936
**33.2 Functional Description** ..... 1939
**33.3 Initialization and Configuration** ..... 1948
**33.4 Interrupt and Event Support** ..... 1948
**33.5 USB Registers** ..... 1950



Copyright © 2026 Texas Instruments Incorporated

## 33.1 USB Overview

The USB controller operates as a full-speed function controller during point-to-point communications with the USB host. The controller complies with the USB 2.0 standard, which includes SUSPEND and RESUME signaling. The USB controller has sixteen endpoints, one-half of them being for IN transactions and one-half of them being for OUT transactions. One IN and one OUT endpoint are fixed-function endpoints used for control transfers; the others are defined by firmware. A dynamically sizeable FIFO supports queuing multiple packets. Software-controlled connect and disconnect allow flexibility during USB device startup.

### *33.1.1 Features*

The USB module has the following features:

- Complies with USB-IF certification standards
- USB 2.0 full-speed (12 Mbps) operation in host and device modes as well as low-speed (1.5 Mbps) operation in host mode
- Integrated PHY
- Four transfer types: Control, Interrupt, Bulk, and Isochronous
- Sixteen endpoints
  - One dedicated control IN endpoint and one dedicated control OUT endpoint
  - Seven configurable IN endpoints and seven configurable OUT endpoints
- Two kB dedicated endpoint memory
- Dedicated hardware triggers for DMA support
- Support for USB-based device firmware update (DFU) through bootstrap loader (BSL)

### *33.1.2 USB Related Collateral*

**Foundational Materials**

- USB Precision Labs (Video)

**Expert Materials**

- *High-Speed Interface Layout Guidelines Application Report*

### *33.1.3 Block Diagram*

The USB block diagram is shown in Figure 33-1.

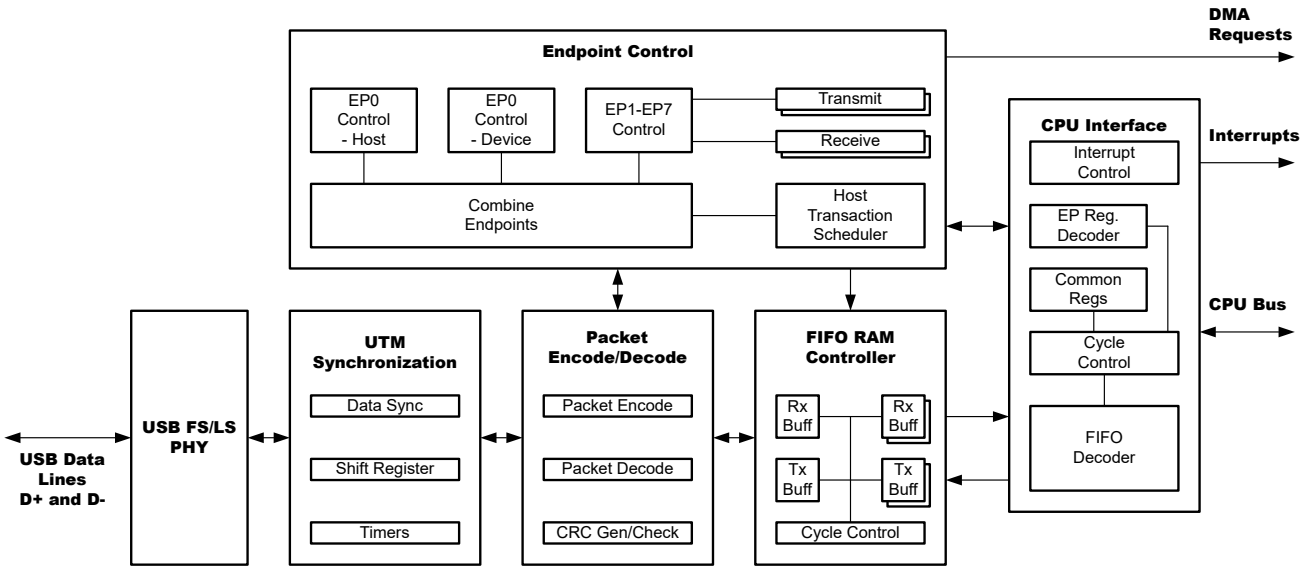



**Figure 33-1. USB Block Diagram**



Copyright © 2026 Texas Instruments Incorporated

#### 33.1.3.1 Signal Description

The USB controller requires a total of three signals (D+, D-, and $V_{Bus}$) to operate in device mode and two signals (D+, D-) to operate in embedded host mode. Because of the differential signaling needed for USB, the pins D+ and D- have special buffers to support USB. As such, their position on the chip is not user-selectable. These pins at reset are, by default, GPIOs. They must be configured before being used as USB function pins. The PHYMODE bit in the USBMODE register should be set to choose the USB function.

### *33.1.4 VBus Recommendations*

Most applications do not need to monitor $V_{Bus}$. Because of this, a dedicated $V_{Bus}$ monitoring pin was not included on this microcontroller. If you are designing a bus-powered device application or an embedded host application, you do not need to monitor $V_{Bus}$. If you are designing a self-powered device, you need to actively monitor the state of the $V_{Bus}$ pin to make sure compliance with the USB specification. In Section 7.1.5 and Section 7.2.1 of the USB Specification Revision 2.0™:

- "The voltage source on the [speed identification] pull-up resistor must be derived from or controlled by the power supplied on the USB cable such that when $V_{BUS}$ is removed, the pull-up resistor does not supply current on the data line to which it is attached.
- When $V_{BUS}$ is removed, the device must remove power from the D+/D- pull-up resistor within 10 seconds.
- Later in the timing tables (Section 7.3.2) of the USB Specification 2.0 it is also stated that the D+/D- pull-up resistor should be applied within 100ms of $V_{Bus}$ reaching a valid level."

Meeting the above specification is easy because of the slow timing requirements. The hardware part of the $V_{Bus}$ monitoring is discussed in this chapter. The corresponding software is discussed briefly, but for examples and an explanation, consult the USB software guide

#### 33.1.4.1 VBUS Status via ODIO

The ODIO of this microcontroller is 5-V tolerant, and because of this, the $V_{Bus}$ can be directly connected. This method does not indicate the absolute voltage of $V_{Bus}$ but will be a logic high when $V_{Bus}$ exceeds VIH. A schematic of this connection is shown in USB VBUS Detection via ODIO.

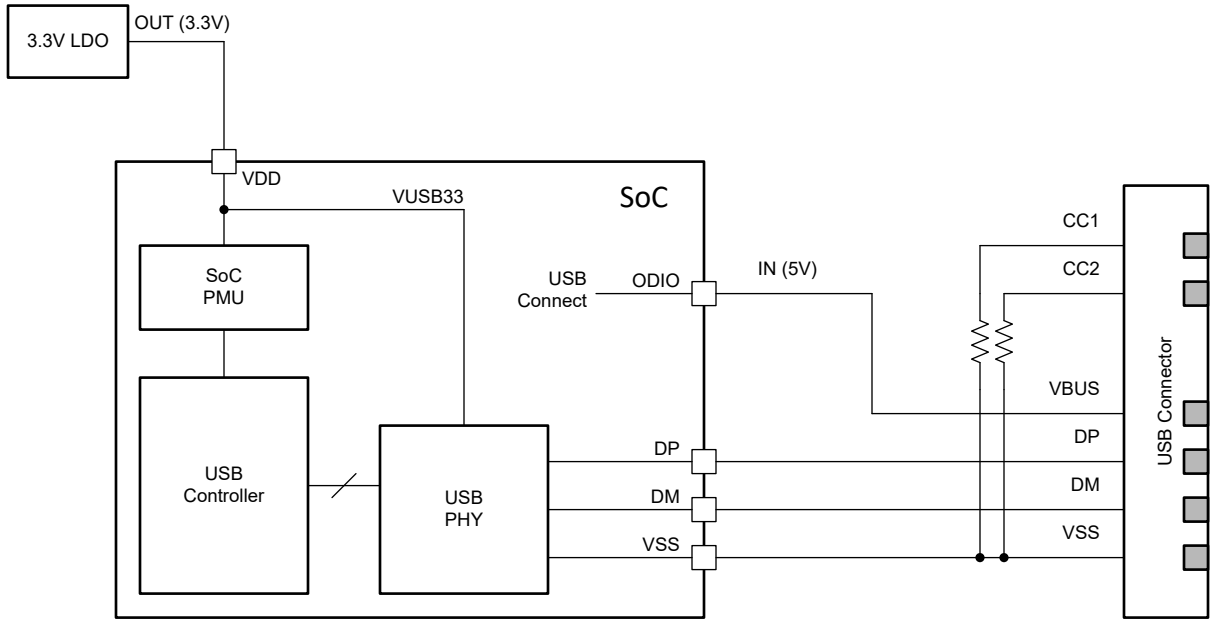



**Figure 33-2. USB VBUS Detection via ODIO**



Copyright © 2026 Texas Instruments Incorporated

#### 33.1.4.2 VBUS Status via COMP

The pins of this microcontroller are not 5-V tolerant, and because of this, the $V_{Bus}$ signal cannot be directly connected to a GPIO pin. Directly connecting 5 V to a pin of the microcontroller destroys the I/O buffer of the pin and possibly more of the chip.

To detect the a sufficiant voltage level of 4.01V on the $V_{Bus}$ signal after a USB cable connect, a resistive divier in combination with the COMP module needs to be used. A schematic of this connection is shown in Figure 33-3. The $V_{Bus}$ signal feeds into a 3:1 resistive divider. The divided output reflects 1/4 of the $V_{Bus}$ signal voltage and will always stay in the maximum operating range of the powered microcontroller. The divided output is then connected to the posetive input of the COMP. The negative input of the COMP is connected to the internal DAC of the COMP. By adjusting the DAC voltage on the negative input to the COMP, the trip voltage can be adjusted. The COMP allows an interrupt event notification to the software, when a USB cable is connected or disconnected to the device. Please refer to the COMP chapter of the TRM for details on how to configer the COMP module and the reference DAC.

**Note**

This method to detect the USB cable connect or disconnect can also be done in STOP and STANDBY mode, since the COMP is operational in these low power modes.

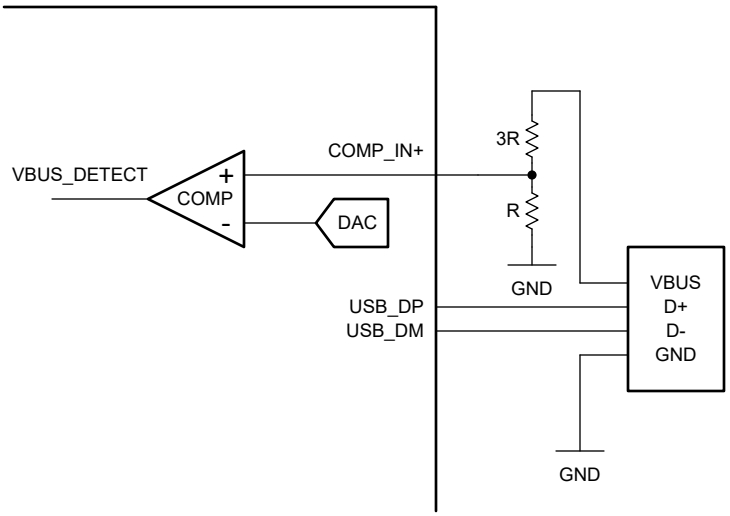



**Figure 33-3. USB VBUS Detection**

#### 33.1.4.3 VUSB33 Status via ADC

Alternatively the connect or disconnect of a USB cable can be achieved, by monitoring the supply voltage on the VUSB33 supply pin. This requires, that the VUSB33 supply pin is connected to the output of a 3.3V LDO, which is directly powered by the $V_{Bus}$ signal off the USB connector. The voltage on the microcontroller VUSB33 supply pin can be monitored by the internal ADC. The assumption is, that the 3.3V LDO will only operate when the USB cable provides a sufficiant supply voltage on the $V_{Bus}$ signal (USB cable connected). Otherwise if the VUSB33 is below 3.0V, this will indicate that the USB cable is not connected. Please refer to the ADC chapter in the TRM for detail regarding the configuration of the ADC, and the device specific datasheet for the ADC input channel mapping.

A schematic of this method to detect the cable connect or disconnect status is shown in USB VUSB33 Detection.

This method of the USB cable connect or disconnect does not require any additional pin, but the ADC operation does limit the use of the STOP and STANDBY low power modes.



Copyright © 2026 Texas Instruments Incorporated

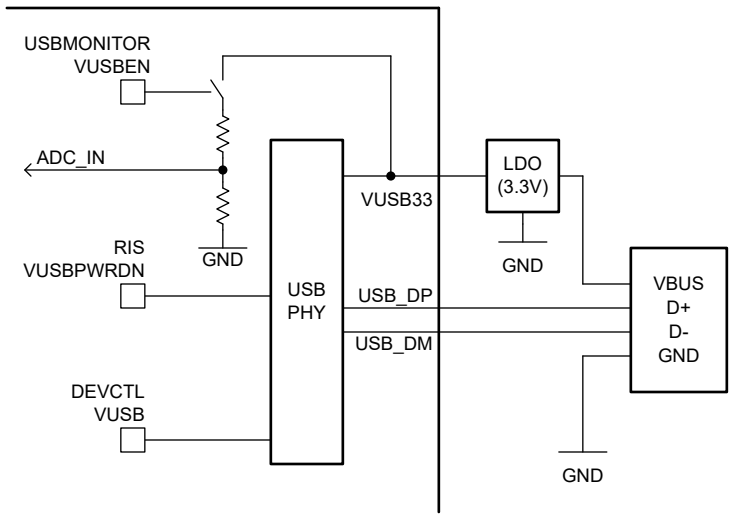



**Figure 33-4. USB VUSB33 Detection**

#### 33.1.4.4 VUSB33 Status via PMU-Monitor

The VUSB33 supply pin is also monitored by the power managment unit (PMU) to control the PHY function in case the VUSB33 supply is unpowered (less than 1.35V). The status of this supply monitor is available as a register bit VUSB in the USB.DEVCTL register. The PMU monitor of the VUSB33 is completely internal to the microcontroller and always available. It does not require the ADC or any other configuration to operate, but it only tells whether VUSB33 is below or above 1.35V.

With the VUSB33 pin connected to the output of a 3.3V LDO, which is directly powered by the $V_{Bus}$ signal off the USB connector like shown in USB VUSB33 Detect, the USB cable connection status can be implied by the VUSB status of the USB.DEVCTL register.

The power loss event on the VUSB33 pin (disconnect of the USB cable) is a notworthy system event and therefore indicated as a VUSBPWRDN event (interrupt) in the USB.RIS/USB.MIS event register. Software can utilize this event to disable the pull-up on the USB_DP/USB_DM pin, without the need to constantly sample the VUSB33 pin via the ADC.

## 33.2 Functional Description

The USB controller can be configured to act as either a dedicated host or device. However, when the USB controller is acting as a self-powered device, a GPIO input or analog comparator input must be connected to $V_{BUS}$ and configured to generate an interrupt when the $V_{BUS}$ level drops. This interrupt is used to disable the pullup resistor on the USB_DP or USB_DM signal.

**Note**

When USB is used in the system, the minimum system frequency is 30MHz.

### *33.2.1 Operation as a Device*

This section describes how the USB controller performs when it is being used as a USB device. IN endpoints, OUT endpoints, entry into and exit from SUSPEND mode, and recognition of start of frame (SOF) are all described.

When in device mode, IN transactions are controlled by the endpoint transmit interface and uses the transmit endpoint registers for the given endpoint. OUT transactions are handled with the endpoints receive interface and use the receive endpoint registers for the given endpoint. When configuring the size of the FIFOs for endpoints, take into account the maximum packet size for an endpoint. Note the following:



Copyright © 2026 Texas Instruments Incorporated

- Bulk endpoints should be the size of the maximum packet (up to 64 bytes) or twice the maximum packet size if double buffering is used (described further in the following section).
- Interrupt endpoints should be the size of the maximum packet (up to 64 bytes) or twice the maximum packet size if double buffering is used.
- It is also possible to specify a separate control endpoint for a USB device. However, in most cases the USB device should use the dedicated control endpoint on the USB controller's endpoint 0.



Copyright © 2026 Texas Instruments Incorporated

#### 33.2.1.1 Control and Configurable Endpoints

When operating as a device, the USB controller provides two dedicated control endpoints (IN and OUT). The remaining available configurable endpoints (one-half IN and one-half OUT) can be used for communications with a host controller. The endpoint number and direction associated with an endpoint is directly related to its register designation. For example, when the Host is transmitting to endpoint 1, all configuration and data is in the endpoint 1 transmit register interface. Endpoint 0 is a dedicated control endpoint used for all control transactions to endpoint 0 during enumeration or when any other control requests are made to endpoint 0. Endpoint 0 uses the first 64 bytes of the USB controller's FIFO RAM as a shared memory for both IN and OUT transactions. The remaining seven endpoints can be configured as control, bulk, isosynchronous, or interrupt endpoints. They should be treated as seven configurable IN and seven configurable OUT endpoints. The endpoint pairs are not required to have the same type for their IN and OUT endpoint configuration. For example, the OUT portion of an endpoint pair could be a bulk endpoint, while the IN portion of that endpoint pair could be an interrupt endpoint. The address and size of the FIFOs attached to each endpoint can be modified to fit the application's needs.

##### *33.2.1.1.1 IN Transactions as a Device*

When operating as a USB device, data for IN transactions is handled through the FIFOs attached to the transmit endpoints. The sizes of the FIFOs for the configurable IN endpoints are determined by the USB Transmit FIFO Start Address (USB.TXFIFOADD) register. The maximum size of a data packet that may be placed in a transmit endpoint’s FIFO for transmission is programmable and is determined by the value written to the USB Maximum Transmit Data Endpoint n (USB.TXMAXPn) register for that endpoint. The endpoint’s FIFO can also be configured to use double-packet or single-packet buffering. When double-packet buffering is enabled, two data packets can be buffered in the FIFO, which also requires that the FIFO is at least two packets in size. When double-packet buffering is disabled, only one packet can be buffered, even if the packet size is less than half the FIFO size.

**Note:** The maximum packet size set for any endpoint must not exceed the FIFO size. The USBTXMAXPn register should not be written to while data is in the FIFO as unexpected results may occur.

**Single-Packet Buffering**

If the size of the transmit endpoint's FIFO is less than twice the maximum packet size for this endpoint (as set in the USB Transmit Dynamic FIFO Sizing (USB.TXFIFOSZ) register), only one packet can be buffered in the FIFO and single-packet buffering is required. When each packet is completely loaded into the transmit FIFO, the TXRDY bit in the USB Transmit Control and Status Endpoint n Low (USB.TXCSRLn) register must be set. If the AUTOSET bit in the USB Transmit Control and Status Endpoint n High (USB.TXCSRHn) register is set, the TXRDY bit is automatically set when a maximum-sized packet is loaded into the FIFO. For packet sizes less than the maximum, the TXRDY bit must be set manually. When the TXRDY bit is set, either manually or automatically, the packet is ready to be sent. When the packet has been successfully sent, both TXRDY and FIFONE are cleared, and the appropriate transmit endpoint interrupt signaled. At this point, the next packet can be loaded into the FIFO.

**Double-Packet Buffering**

If the size of the transmit endpoint's FIFO is at least twice the maximum packet size for this endpoint, two packets can be buffered in the FIFO and double-packet buffering is allowed. As each packet is loaded into the transmit FIFO, the TXRDY bit in the USB.TXCSRLn register must be set. If the AUTOSET bit in the USBTXCSRHn register is set, the TXRDY bit is automatically set when a maximum-sized packet is loaded into the FIFO. For packet sizes less than the maximum, TXRDY must be set manually. When the TXRDY bit is set, either manually or automatically, the packet is ready to be sent. After the first packet is loaded, TXRDY is immediately cleared and an interrupt is generated. A second packet can now be loaded into the transmit FIFO and TXRDY set again (either manually or automatically if the packet is the maximum size). At this point, both packets are ready to be sent. After each packet has been successfully sent, TXRDY is automatically cleared and the appropriate transmit endpoint interrupt signaled to indicate that another packet can now be loaded into the transmit FIFO. The state of the FIFONE bit in the USB.TXCSRLn register at this point indicates how many packets may be loaded. If the FIFONE bit is set, then another packet is in the FIFO and only one more packet can be loaded. If the FIFONE bit is clear, then no packets are in the FIFO and two more packets can be loaded.



Copyright © 2026 Texas Instruments Incorporated

**Note:** Double-packet buffering is disabled if an endpoint's corresponding EPn bit is set in the USB Transmit Double Packet Buffer Disable (USB.TXDPKTBUFDIS) register. This bit is set by default, so it must be cleared to enable double-packet buffering.

#### *33.2.1.1.2 Out Transactions as a Device*

When in device mode, OUT transactions are handled through the USB controller receive FIFOs. The sizes of the receive FIFOs for the configurable OUT endpoints are determined by the USB Receive FIFO Start Address (USB.RXFIFOADD) register. The maximum amount of data received by an endpoint in any packet is determined by the value written to the USB Maximum Receive Data Endpoint n (USB.RXMAXPn) register for that endpoint. When double-packet buffering is enabled, two data packets can be buffered in the FIFO. When double-packet buffering is disabled, only one packet can be buffered even if the packet is less than half the FIFO size.

**Note:** In all cases, the maximum packet size must not exceed the FIFO size.

**Single-Packet Buffering**

If the size of the receive endpoint FIFO is less than twice the maximum packet size for an endpoint, only one data packet can be buffered in the FIFO and single-packet buffering is required. When a packet is received and placed in the receive FIFO, the RXRDY and FULL bits in the USB Receive Control and Status Endpoint n Low (USB.RXCSRL[*n*]) register are set and the appropriate receive endpoint is signaled, indicating that a packet can now be unloaded from the FIFO. After the packet has been unloaded, the RXRDY bit must be cleared in order to allow further packets to be received. This action also generates the acknowledge signaling to the Host controller. If the AUTOCL bit in the USB Receive Control and Status Endpoint n High (USB.RXCSRH[*n*]) register is set and a maximum-sized packet is unloaded from the FIFO, the RXRDY and FULL bits are cleared automatically. For packet sizes less than the maximum, RXRDY must be cleared manually.

**Double-Packet Buffering**

If the size of the receive endpoint FIFO is at least twice the maximum packet size for the endpoint, two data packets can be buffered and double-packet buffering can be used. When the first packet is received and loaded into the receive FIFO, the RXRDY bit in the USB.RXCSRL[*n*] register is set and the appropriate receive endpoint interrupt is signaled to indicate that a packet can now be unloaded from the FIFO.

**Note:** The FULL bit in USB.RXCSRL[*n*] is not set when the first packet is received. It is only set if a second packet is received and loaded into the receive FIFO.

After each packet has been unloaded, the RXRDY bit must be cleared to allow further packets to be received. If the AUTOCL bit in the USB.RXCSRH[*n*] register is set and a maximum-sized packet is unloaded from the FIFO, the RXRDY bit is cleared automatically. For packet sizes less than the maximum, RXRDY must be cleared manually. If the FULL bit is set when RXRDY is cleared, the USB controller first clears the FULL bit, then sets RXRDY again to indicate that there is another packet waiting in the FIFO to be unloaded.

**Note:** Double-packet buffering is disabled if an endpoint's corresponding EPn bit is set in the USB Receive Double Packet Buffer Disable (USB.RXDPKTBUFDIS) register. This bit is set by default, so it must be cleared to enable double-packet buffering.



Copyright © 2026 Texas Instruments Incorporated

#### 33.2.1.1.3 Scheduling

The device has no control over the scheduling of transactions as scheduling is determined by the Host controller. The USB controller can set up a transaction at any time. The USB controller waits for the request from the Host controller and generates an interrupt when the transaction is complete or if it was terminated due to some error. If the Host controller makes a request and the device controller is not ready, the USB controller sends a busy response (NAK) to all requests until it is ready.

#### 33.2.1.1.4 Additional Actions

The USB controller responds automatically to certain conditions on the USB bus or actions by the Host controller such as when the USB controller automatically stalls a control transfer or unexpected zero length OUT data packets.

**Stalled Control Transfer**

The USB controller automatically issues a STALL handshake to a control transfer under the following conditions:

1. The Host sends more data during an OUT data phase of a control transfer than was specified in the device request during the SETUP phase. This condition is detected by the USB controller when the Host sends an OUT token (instead of an IN token) after the last OUT packet has been unloaded and the DATAEND bit in the USB Control and Status Endpoint 0 Low (USB.CSRL0) register has been set.
2. The Host requests more data during an IN data phase of a control transfer than was specified in the device request during the SETUP phase. This condition is detected by the USB controller when the Host sends an IN token (instead of an OUT token) after the CPU has cleared TXRDY and set DATAEND in response to the ACK issued by the Host to what should have been the last packet.
3. The Host sends more than USB.RXMAXPn bytes of data with an OUT data token.
4. The Host sends more than a zero length data packet for the OUT STATUS phase.

**Zero Length OUT Data Packets**

A zero-length OUT data packet is used to indicate the end of a control transfer. In normal operation, such packets should only be received after the entire length of the device request has been transferred. However, if the Host sends a zero-length OUT data packet before the entire length of device request has been transferred, it is signaling the premature end of the transfer. In this case, the USB controller automatically flushes any IN token ready for the data phase from the FIFO and sets the DATAEND bit in the USB.CSRL0 register.

**Setting the Device Address**

When a Host is attempting to enumerate the USB device, it requests that the device change its address from zero to some other value. The address is changed by writing the value that the Host requested to the USB Device Functional Address (USB.FADDR) register. However, care should be taken when writing to USB.FADDR to avoid changing the address before the transaction is complete. This register should only be set after the SET_ADDRESS command is complete. Like all control transactions, the transaction is only complete after the device has left the STATUS phase. In the case of a SET_ADDRESS command, the transaction is completed by responding to the IN request from the Host with a zero-byte packet. Once the device has responded to the IN request, the USB.FADDR register should be programmed to the new value as soon as possible to avoid missing any new commands sent to the new address.

**Note:** If the USB.FADDR register is set to the new value as soon as the device receives the OUT transaction with the SET_ADDRESS command in the packet, it changes the address during the control transfer. In this case, the device does not receive the IN request that allows the USB transaction to exit the STATUS phase of the control transfer because it is sent to the old address. As a result, the Host does not get a response to the IN request, and the Host fails to enumerate the device.



Copyright © 2026 Texas Instruments Incorporated

#### 33.2.1.1.5 Device Mode Suspend

When no activity has occurred on the USB bus for 3 ms, the USB controller automatically enters SUSPEND mode. If the SUSPEND interrupt has been enabled in the USB Interrupt Enable (USBIE) register, an interrupt is generated at this time. When in SUSPEND mode, the PHY also goes into SUSPEND mode. When RESUME signaling is detected, the USB controller exits SUSPEND mode and takes the PHY out of SUSPEND. If the RESUME interrupt is enabled, an interrupt is generated. The USB controller can also be forced to exit SUSPEND mode by setting the RESUME bit in the USB Power (USBPOWER) register. When this bit is set, the USB controller exits SUSPEND mode and drives RESUME signaling onto the bus. The RESUME bit must be cleared after 10 ms (a maximum of 15 ms) to end RESUME signaling. To meet USB power requirements, the controller can be put into Deep Sleep mode which keeps the controller in a static state.

#### 33.2.1.1.6 Start of Frame

When the USB controller is operating in device mode, it receives a Start-Of-Frame (SOF) packet from the Host once every millisecond. When the SOF packet is received, the 11-bit frame number contained in the packet is written into the USB Frame Value (USB.FRAME) register, and an SOF interrupt is also signaled and can be handled by the application. Once the USB controller has started to receive SOF packets, it expects one every millisecond. If no SOF packet is received after 1.00358 ms, the packet is assumed to have been lost, and the USB.FRAME register is not updated. The USB controller continues and resynchronizes these pulses to the received SOF packets when these packets are successfully received again.

#### 33.2.1.1.7 USB Reset

When the USB controller is in device mode and a RESET condition is detected on the USB bus, the USB controller automatically performs the following actions:

- Clears the USBFADDR register
- Clears the USB Endpoint Index (USB.EPINDEX) register
- Flushes all endpoint FIFOs
- Clears all control/status registers
- Enables all endpoint interrupts
- Generates a RESET interrupt

#### 33.2.1.1.8 Connect/Disconnect

The USB controller connection to the USB bus is handled by software. The USB PHY can be switched between normal mode and non-driving mode by setting or clearing the SOFT_CONN bit of the USB.POWER register. When the SOFT_CONN bit is set, the PHY is placed in its normal mode, and the USB_DP/USB_DM lines of the USB bus are enabled. At the same time, the USB controller is placed into a state, in which it does not respond to any USB signaling except a USB RESET. When the SOFTCONN bit is cleared, the PHY is put into non-driving mode, USB_DP and USB_DM are tristated, and the USB controller appears to other devices on the USB bus as if it has been disconnected. The non-driving mode is the default so the USB controller appears disconnected until the SOFT_CONN bit has been set. The application software can then choose when to set the PHY into its normal mode. Systems with a lengthy initialization procedure may use this to ensure that initialization is complete, and the system is ready to perform enumeration before connecting to the USB bus. Once the SOFT_CONN bit has been set, the USB controller can be disconnected by clearing this bit.

**Note**

The USB controller does not generate an interrupt when the device is connected to the Host. However, an interrupt is generated when the Host terminates a session.

### 33.2.2 Operation as a Host

When the USB controller is operating in Host mode, it can either be used for point-to-point communications with another USB device or, when attached to a hub, for communication with multiple devices. Full-speed and low-speed USB devices are supported, both for point-to-point communication and for operation through a hub. The USB controller automatically carries out the necessary transaction translation needed to allow a low-speed



Copyright © 2026 Texas Instruments Incorporated

or full-speed device to be used with a USB 2.0 hub. Control, bulk, and interrupt transactions are supported. This section describes the USB controller's actions when it is being used as a USB Host. Configuration of IN endpoints, OUT endpoints, entry into and exit from SUSPEND mode, and RESET are all described.

When in Host mode, IN transactions are controlled by an endpoint's receive interface. All IN transactions use the receive endpoint registers and all OUT endpoints use the transmit endpoint registers for a given endpoint. As in device mode, the FIFOs for endpoints should take into account the maximum packet size for an endpoint.

- Bulk endpoints should be the size of the maximum packet (up to 64 bytes) or twice the maximum packet size if double buffering is used (described further in the following section).
- Interrupt endpoints should be the size of the maximum packet (up to 64 bytes) or twice the maximum packet size if double buffering is used.
- It is also possible to specify a separate control endpoint to communicate with a device. However, in most cases the USB controller should use the dedicated control endpoint to communicate with a device's endpoint 0.

#### 33.2.2.1 Endpoint Registers

The endpoint registers are used to control the USB endpoint interfaces which communicate with devices that are connected. The endpoints consist of a dedicated control IN endpoint and a dedicated control OUT endpoint. The remaining available endpoints are configurable, with one-half of them being OUT endpoints, and one-half of them being IN endpoints. See Section 33.1.1 for the number of available endpoints on this device.

The dedicated control interface can only be used for control transactions to endpoint 0 of devices. These control transactions are used during enumeration or other control functions that communicate using endpoint 0 of devices. This control endpoint shares the first 64 bytes of the USB controller's FIFO RAM for IN and OUT transactions. The remaining IN and OUT interfaces can be configured to communicate with control, bulk, or interrupt endpoints.

These USB interfaces can be used to simultaneously schedule as many as seven independent OUT and seven independent IN transactions to any endpoints on any device. The IN and OUT controls are paired together in the same set of registers for the respective endpoints. However, the IN and OUT controls can be configured to communicate with different types of endpoints and different endpoints on devices. For example, the first pair of endpoint controls can be split so that the OUT portion is communicating with a device's bulk OUT endpoint 1, while the IN portion is communicating with a device's interrupt IN endpoint 2.

Before accessing any device, whether for point-to-point communications or for communications via a hub, the relevant USB Receive Functional Address Endpoint n (USB.RXFUNCADDRn) or USB Transmit Functional Address Endpoint n (USB.TXFUNCADDRn) registers must be set for each receive or transmit endpoint to record the address of the device being accessed.

The USB controller also supports connections to devices through a USB hub by providing a register that specifies the hub address and port of each USB transfer. The FIFO address and size are customizable and can be specified for each USB IN and OUT transfer. Customization includes allowing one FIFO per transaction, sharing a FIFO across transactions, and allowing for double-buffered FIFOs.

#### 33.2.2.2 IN Transactions as a Host

IN transactions are handled in a similar manner to the way in which OUT transactions are handled when the USB controller is in device mode except that the transaction first must be initiated by setting the REQPKT bit in the USB.CSRL0 register, indicating to the transaction scheduler that there is an active transaction on this endpoint. The transaction scheduler then sends an IN token to the target device. When the packet is received and placed in the receive FIFO, the RXRDY bit in the USBCSRL0 register is set, and the appropriate receive endpoint interrupt is signaled to indicate that a packet can now be unloaded from the FIFO.

When the packet has been unloaded, RXRDY must be cleared. The AUTOCL bit in the USB.RXCSRHn register can be used to have RXRDY automatically cleared when a maximum-sized packet has been unloaded from the FIFO. The AUTORQ bit in USB.RXCSRHn causes the REQPKT bit to be automatically set when the RXRDY bit is cleared. When the RXRDY bit is cleared, the controller sends an acknowledge to the device. When there



Copyright © 2026 Texas Instruments Incorporated

is a known number of packets to be transferred, the USB Request Packet Count in Block Transfer Endpoint n (USB.RQPKTCOUNTn) register associated with the endpoint should be configured to the number of packets to be transferred. The USB controller decrements the value in the USB.RQPKTCOUNTn register following each request. When the USB.RQPKTCOUNTn value decrements to 0, the AUTORQ bit is cleared to prevent any further transactions being attempted. For cases where the size of the transfer is unknown, USB.RQPKTCOUNTn should be cleared. AUTORQ then remains set until cleared by the reception of a short packet (that is, less than the MAXLOAD value in the USB.RXMAXPn register) such as may occur at the end of a bulk transfer.

If the device responds to a bulk or interrupt IN token with a NAK, the USB Host controller keeps retrying the transaction until any NAK Limit that has been set has been reached. If the target device responds with a STALL, however, the USB Host controller does not retry the transaction but sets the STALLED bit in the USB.CSRL0 register. If the target device does not respond to the IN token within the required time, or the packet contained a CRC or bit-stuff error, the USB Host controller retries the transaction. If after three attempts the target device has still not responded, the USB Host controller clears the REQPKT bit and sets the ERROR bit in the USB.CSRL0 register.

#### 33.2.2.3 OUT Transactions as a Host

OUT transactions are handled in a similar manner to the way in which IN transactions are handled when the USB controller is in device mode. The TXRDY bit in the USB.TXCSRLn register must be set as each packet is loaded into the transmit FIFO. Again, setting the AUTOSET bit in the USB.TXCSRHn register automatically sets TXRDY when a maximum-sized packet has been loaded into the FIFO.

If the target device responds to the OUT token with a NAK, the USB Host controller keeps retrying the transaction until the NAK Limit that has been set has been reached. However, if the target device responds with a STALL, the USB controller does not retry the transaction but interrupts the main processor by setting the STALLED bit in the USB.TXCSRLn register. If the target device does not respond to the OUT token within the required time, or the packet contained a CRC or bit-stuff error, the USB Host controller retries the transaction. If after three attempts the target device has still not responded, the USB controller flushes the FIFO and sets the ERROR bit in the USB.TXCSRLn register.

#### 33.2.2.4 Transaction Scheduling

Scheduling of transactions is handled automatically by the USB Host controller. The Host controller allows configuration of the endpoint communication scheduling based on the type of endpoint transaction. Interrupt transactions can be scheduled to occur in the range of every frame to every 255 frames in 1 frame increments. Bulk endpoints do not allow scheduling parameters, but do allow for a NAK timeout in the event an endpoint on a device is not responding.

The USB controller maintains a frame counter. If the target device is a full-speed device, the USB controller automatically sends an SOF packet at the start of each frame and increments the frame counter. If the target device is a low-speed device, a K state is transmitted on the bus to act as a keep-alive to stop the low-speed device from going into SUSPEND mode.

After the SOF packet has been transmitted, the USB Host controller cycles through all the configured endpoints looking for active transactions. An active transaction is defined as a receive endpoint for which the REQPKT bit is set or a transmit endpoint for which the TXRDY bit and/or the FIFONE bit is set.

An interrupt transaction is started if the transaction is found on the first scheduler cycle of a frame and if the interval counter for that endpoint has counted down to zero. As a result, only one interrupt transaction occurs per endpoint every n frames, where n is the interval set via the USB Host Transmit Interval Endpoint n (USB.TXINTERVAL[*n*]) or USB Host Receive Interval Endpoint n (USB.RXINTERVAL[*n*]) register for that endpoint.

An active bulk transaction starts immediately, provided sufficient time is left in the frame to complete the transaction before the next SOF packet is due. If the transaction must be retried (for example, because a NAK was received or the target device did not respond), then the transaction is not retried until the transaction scheduler has first checked all the other endpoints for active transactions. This process ensures that an endpoint



Copyright © 2026 Texas Instruments Incorporated

that is sending a lot of NAKs does not block other transactions on the bus. The controller also allows the user to specify a limit to the length of time for NAKs to be received from a target device before the endpoint times out.

#### 33.2.2.5 USB Hubs

The following setup requirements apply to the USB Host controller only if it is used with a USB hub. When a full- or low-speed device is connected to the USB controller via a USB 2.0 hub, details of the hub address and the hub port also must be recorded in the corresponding USB Receive Hub Address Endpoint n (USB.RXHUBADDRn) and USB Receive Hub Port Endpoint n (USB.RXHUBPORTn) or the USB Transmit Hub Address Endpoint n (USB.TXHUBADDRn) and USB Transmit Hub Port Endpoint n (USB.TXHUBPORTn) registers. In addition, the speed at which the device operates (full or low) must be recorded in the USB Type Endpoint 0 (USBTYPE0) (endpoint 0), USB Host Configure Transmit Type Endpoint n (USB.TXTYPEn), or USB Host Configure Receive Type Endpoint n (USB.RXTYPEn) registers for each endpoint that is accessed by the device.

For hub communications, the settings in these registers record the current allocation of the endpoints to the attached USB devices. To maximize the number of devices supported, the USB Host controller allows this allocation to be changed dynamically by simply updating the address and speed information recorded in these registers. Any changes in the allocation of endpoints to device functions must be made following the completion of any on-going transactions on the endpoints affected.

#### 33.2.2.6 Babble

The USB Host controller does not start a transaction until the bus has been inactive for at least the minimum inter-packet delay. The controller also does not start a transaction unless it can be finished before the end of the frame. If the bus is still active at the end of a frame, then the USB Host controller assumes that the target device to which it is connected has malfunctioned, and the USB controller suspends all transactions and generates a babble interrupt.

#### 33.2.2.7 Host SUSPEND

If the SUSPEND bit in the USBPOWER register is set, the USB Host controller completes the current transaction then stops the transaction scheduler and frame counter. No further transactions are started and no SOF packets are generated.

To exit SUSPEND mode, set the RESUME bit and clear the SUSPEND bit. While the RESUME bit is set, the USB Host controller generates RESUME signaling on the bus. After 20 ms, the RESUME bit must be cleared, at which point the frame counter and transaction scheduler start. The Host supports the detection of a remote wake-up.

#### 33.2.2.8 USB RESET

If the RESET bit in the USB.POWER register is set, the USB Host controller generates USB RESET signaling on the bus. The RESET bit must be set for at least 20 ms to ensure correct resetting of the target device. After the CPU has cleared the bit, the USB Host controller starts its frame counter and transaction scheduler.

#### 33.2.2.9 Connect/Disconnect

A session is started by setting the SESSION bit in the USB device Control (USB.DEVCTL) register, enabling the USB controller to wait for a device to be connected. When a device is detected, a connect interrupt is generated. The speed of the device that has been connected can be determined by reading the USB.DEVCTL register where the FSDEV bit is set for a full-speed device, and the LSDEV bit is set for a low-speed device. The USB controller must generate a RESET to the device, and then the USB Host controller can begin device enumeration. If the device is disconnected while a session is in progress, a disconnect interrupt is generated.



Copyright © 2026 Texas Instruments Incorporated

### 33.2.3 DMA Operation

The USB module supports four RX and four TX triggers (TRIG[A-D]RX and TRIG[A-D]TX) to the DMA. Each trigger can be assigned to a specific Endpoint (EPx)

The DMA controller may be used to read and write the USB FIFOs via software or hardware triggering. See the Direct Memory Access (DMA) chapter for more details about programming the DMA controller.

## 33.3 Initialization and Configuration

To use the USB Controller, the clock for the USB controller must first be configured. USBCLK is a 60-MHz clock to the USB module. This fixed frequency is required for all USB operations.

The USBCLK can either by sourced by the USBFLL or the SYSPLL. When the SYSPLL is used, a crystal needs to used as clock reference for the SYSPLL. The USBCLK can be asynchronous to the system clock. For USB operation the system clock needs to be at least 30MHz.

If the USB Controller is operated in device mode, the clock can come from the USBFLL or the SYSPLL.

If the USB Controller is operated in host mode, the clock needs to come from the SYSPLL with an external crystal as clock reference.

### 33.3.1 Pin Configuration

Before the USB controller can be used, the USB PHY needs to switched into USB mode by setting the PHYMODE bit in the USB.USBMODE register.

If the USB controller operates in host mode, the HOST bit in the USB.USBMODE register needs to be set.

If the USB controller operates in device mode, the DEVICEONLY bit in the USB.USBMODE register needs to be set.

### 33.3.2 Endpoint Configuration

To start communication in Host or device mode, the endpoint registers must first be configured. In Host mode, this configuration establishes a connection between an endpoint register and an endpoint on a device. In device mode, an endpoint must be configured before enumerating to the Host controller.

In both cases, the endpoint 0 configuration is limited because it is a fixed-function, fixed-FIFO-size endpoint. In device and Host modes, the endpoint requires little setup but does require a software-based state machine to progress through the setup, data, and status phases of a standard control transaction. In device mode, the configuration of the remaining endpoints is done once before enumerating and then only changed if an alternate configuration is selected by the Host controller. In Host mode, the endpoints must be configured to operate as control, bulk, or interrupt mode. Once the type of endpoint is configured, a FIFO area must be assigned to each endpoint. In the case of bulk, control and interrupt endpoints, each has a maximum of 64 bytes per transaction. The maximum packet size for the given endpoint must be set prior to sending or receiving data.

Configuring each endpoint's FIFO involves reserving a portion of the overall USB FIFO RAM to each endpoint. The total FIFO RAM available is 2 Kbytes with the first 64 bytes reserved for endpoint 0. The endpoint's FIFO must be at least as large as the maximum packet size. The FIFO can also be configured as a double-buffered FIFO so that interrupts occur at the end of each packet and allow filling the other half of the FIFO.

If operating as a device, the USB device controller's soft connect must be enabled when the device is ready to start communications, indicating to the host controller that the device is ready to start the enumeration process. If operating as a Host controller, the device soft connect must be disabled and power must be provided to $V_{BUS}$.

## 33.4 Interrupt and Event Support

The USB controller contains one event publisher (CPU_INT) which manages USB interrupt requests (IRQs) to the CPU subsystem through a static event rout.



Copyright © 2026 Texas Instruments Incorporated

The CPU_INT event register is a overarching event status, which combines the different event sources in the USB module. The INTRTX, INTRRX and INTRUSB interrupt status in the USB.RIS are set, when a corresponding interrupt in the USB.TXIS, USB.RXIS and USB.USBIS are set.

The VUSBPWRDN is set when the external supply voltage on the VUSB33 pin drops below 1.35V.

The DMADONExRX/DMADONExTX are set when the corresponding DMA channel has completed the transfer.



Copyright © 2026 Texas Instruments Incorporated

## 33.5 USB Registers

Table 33-1 lists the memory-mapped registers for the USB registers. All register offset addresses not listed in Table 33-1 should be considered as reserved locations and the register contents should not be modified.

**Table 33-1. USB Registers**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 800h | PWREN | Power enable | | Go |
| 804h | RSTCTL | Reset Control | | Go |
| 808h | CLKCTL | Clock control | | Go |
| 814h | STAT | Status Register | | Go |
| 1020h | IIDX | Interrupt index | | Go |
| 1028h | IMASK | Interrupt mask | | Go |
| 1030h | RIS | Raw interrupt status | | Go |
| 1038h | MIS | Masked interrupt status | | Go |
| 1040h | ISET | Interrupt set | | Go |
| 1048h | ICLR | Interrupt clear | | Go |
| 10E0h | EVT_MODE | Event Mode | | Go |
| 10FCh | DESC | Module Description | | Go |
| 1100h | USBMODE | USB mode control | | Go |
| 1104h | USBMONITOR | USB Clock Control Register | | Go |
| 1108h | USBDMASEL | USB DMA trigger select | | Go |
| 2000h | FADDR | USB Device Functional Address | | Go |
| 2001h | POWER | USB Power Control | | Go |
| 2002h | TXIS | USB Transmit Interrupt Status | | Go |
| 2004h | RXIS | USB Receive Interrupt Status | | Go |
| 2006h | TXIE | USB Transmit Interrupt Enable | | Go |
| 2008h | RXIE | USB Receive Interrupt Enable | | Go |
| 200Ah | USBIS | USB General Interrupt Status | | Go |
| 200Bh | USBIE | USB General Interrupt Enable | | Go |
| 200Ch | FRAME | USB Last Received Frame Number | | Go |
| 200Eh | EPINDEX | USB Endpoint Index | | Go |
| 200Fh | TESTMODE | USB Test Mode | | *Go* |
| 2010h | IDXTXMAXP | Indexed TXMAP | | Go |
| 2012h | IDXTXCSRL | Indexed TXCSRL | | Go |
| 2013h | IDXTXCSRH | Indexed TXCSRH | | Go |
| 2014h | IDXRXMAXP | Indexed RXMAXP | | Go |
| 2016h | IDXRXCSRL | Indexed RXCSRL | | Go |
| 2017h | IDXRXCSRH | Indexed RXCSRH | | Go |
| 2018h | IDXRXCOUNT | Indexed RXCOUNT | | Go |
| 201Ah | IDXTXTYPE | Indexed TXTYPE | | Go |
| 201Bh | IDXTXINTERVAL | Indexed TXINTERVAL | | Go |
| 201Ch | IDXRXTYPE | Indexed RXTYPE | | Go |
| 201Dh | IDXRXINTERVAL | Indexed RXINTERVAL | | Go |
| 201Fh | IDXFIFOSZ | Indexed FIFOSZ | | Go |
| 2020h + formula | FIFO[y] | USB FIFO Endpoint [n] | | Go |
| 2060h | DEVCTL | USB Device Control | | Go |
| 2061h | MISC | USB Miscellaneous Register | | Go |



Copyright © 2026 Texas Instruments Incorporated

**Table 33-1. USB Registers (continued)**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 2062h | IDXTXFIFOSZ | Indexed USB Transmit Dynamic FIFO Sizing | | Go |
| 2063h | IDXRXFIFOSZ | Indexed USB Receive Dynamic FIFO Sizing | | Go |
| 2064h | IDXTXFIFOADD | Indexed USB Transmit FIFO Start Address | | Go |
| 2066h | IDXRXFIFOADD | Indexed USB Receive FIFO Start Address | | Go |
| 2078h | EPINFO | Endpoint Information | | Go |
| 2079h | RAMINFO | USB FIFO Buffer RAM Information | | Go |
| 207Ah | CONTIM | USB Connect Timing | | Go |
| 207Dh | FSEOF | USB Full-Speed Last Transaction to End of Frame Timing | | Go |
| 207Eh | LSEOF | USB Low-Speed Last Transaction to End of Frame Timing | | Go |
| 2080h + formula | TXFUNCADDR[j] | USB Transmit Functional Address Endpoint [n] | | Go |
| 2082h + formula | TXHUBADDR[j] | USB Transmit Hub Address Endpoint [n] | | Go |
| 2083h + formula | TXHUBPORT[j] | USB Transmit Hub Port Endpoint [n] | | Go |
| 2084h + formula | RXFUNCADDR[j] | USB Receive Functional Address Endpoint [n] | | Go |
| 2086h + formula | RXHUBADDR[j] | USB Receive Hub Address Endpoint [n] | | Go |
| 2087h + formula | RXHUBPORT[j] | USB Receive Hub Port Endpoint [n] | | Go |
| 2102h | CSR0L | USB Control and Status Endpoint 0 Low | | Go |
| 2103h | CSR0H | USB Control and Status Endpoint 0 High | | Go |
| 2108h | COUNT0 | USB Receive Byte Count Endpoint 0 | | Go |
| 210Ah | TYPE0 | USB Type Endpoint 0 | | Go |
| 210Bh | NAKLMT0 | USB NAK Limit Endpoint 0 | | Go |
| 210Fh | CONFIG | USB Config Data | | Go |
| 2110h + formula | TXMAXP[j] | USB Maximum Transmit Data Endpoint [n] | | Go |
| 2112h + formula | TXCSRL[j] | USB Transmit Control and Status Endpoint [n] Low | | Go |
| 2113h + formula | TXCSRH[j] | USB Transmit Control and Status Endpoint [n] High | | Go |
| 2114h + formula | RXMAXP[j] | USB Maximum Receive Data Endpoint [n] | | Go |
| 2116h + formula | RXCSRL[j] | USB Receive Control and Status Endpoint [n] Low | | Go |
| 2117h + formula | RXCSRH[j] | USB Receive Control and Status Endpoint [n] High | | Go |
| 2118h + formula | RXCOUNT[j] | USB Receive Byte Count Endpoint [n] | | Go |
| 211Ah + formula | TXTYPE[j] | USB Host Transmit Configure Type Endpoint [n] | | Go |
| 211Bh + formula | TXINTERVAL[j] | USB Host Transmit Interval Endpoint [n] | | Go |



Copyright © 2026 Texas Instruments Incorporated

**Table 33-1. USB Registers (continued)**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 211Ch + formula | RXTYPE[j] | USB Host Configure Receive Type Endpoint [n] | | Go |
| 211Dh + formula | RXINTERVAL[j] | USB Host Receive Polling Interval Endpoint [n] | | Go |
| 211Fh + formula | FIFOSZ[j] | USB TX and RX FIFO Size Endpoint [n] | | Go |
| 2304h + formula | RQPKTCOUNT[y] | USB Request Packet Count in Block Transfer Endpoint [n] | | Go |
| 2340h | RXDPKTBUFDIS | USB Receive Double Packet Buffer Disable | | Go |
| 2342h | TXDPKTBUFDIS | USB Transmit Double Packet Buffer Disable | | Go |

Complex bit access types are encoded to fit into small table cells. Table 33-2 shows the codes that are used for access types in this section.

**Table 33-2. USB Access Type Codes**

| Access Type | Code | Description |
|---|---|---|
| Read Type | | |
| R | R | Read |
| Write Type | | |
| W | W | Write |
| W1C | W<br>1C | Write<br>1 to clear |
| WK | W<br>K | Write<br>Write protected by a key |
| Reset or Default Value | | |
| *-n* | | Value after reset or the default value |
| Register Array Variables | | |
| i,j,k,l,m,n | | When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula. |
| y | | When this variable is used in a register name, an offset, or an address it refers to the value of a register array. |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.1 PWREN (Offset = 800h) [Reset = 00000000h]

PWREN is shown in Figure 33-5 and described in Table 33-3.

Return to the Summary Table.

Register to control the power state

**Figure 33-5. PWREN**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | ENABLE |
| R-0h | | | | | | | R/WK-0h |

**Table 33-3. PWREN Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | KEY to allow Power State Change<br>26h = KEY to allow write access to this register |
| 23-1 | RESERVED | R | 0h | |
| 0 | ENABLE | R/WK | 0h | Enable the power<br>KEY must be set to 26h to write to this bit.<br>0h = Disable Power<br>1h = Enable Power |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.2 RSTCTL (Offset = 804h) [Reset = 00000000h]

RSTCTL is shown in Figure 33-6 and described in Table 33-4.

Return to the Summary Table.

Register to control reset assertion and de-assertion

**Figure 33-6. RSTCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | RESETSTKYCLR | RESETASSERT |
| R-0h | | | | | | WK-0h | WK-0h |

**Table 33-4. RSTCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | Unlock key<br>B1h = KEY to allow write access to this register |
| 23-2 | RESERVED | R | 0h | |
| 1 | RESETSTKYCLR | WK | 0h | Clear the RESETSTKY bit in the STAT register<br>KEY must be set to B1h to write to this bit.<br>0h = Writing 0 has no effect<br>1h = Clear reset sticky bit |
| 0 | RESETASSERT | WK | 0h | Assert reset to the peripheral<br>KEY must be set to B1h to write to this bit.<br>0h = Writing 0 has no effect<br>1h = Assert reset |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.3 CLKCTL (Offset = 808h) [Reset = 00000000h]

CLKCTL is shown in Figure 33-7 and described in Table 33-5.

Return to the Summary Table.

Clock control register

**Figure 33-7. CLKCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | CLKDIV | | | |
| R-0h | | | | | | | | | | | | R/W-0h | | | |

**Table 33-5. CLKCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3-0 | CLKDIV | R/W | 5h | Divider settings for the functional clock<br>0h = Divide by 1<br>1h = Divide by 2<br>2h = Divide by 3<br>3h = Divide by 4<br>4h = Divide by 5<br>5h = Divide by 6<br>6h = Divide by 10<br>7h = Divide by 12<br>8h = Divide by 15<br>9h = Divide by 20<br>Ah = Divide by 30<br>Bh = Divide by 60 |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.4 STAT (Offset = 814h) [Reset = 00000000h]

STAT is shown in Figure 33-8 and described in Table 33-6.

Return to the Summary Table.

peripheral enable and reset status

**Figure 33-8. STAT**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | RESETSTKY |
| R-0h | | | | | | | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | ENABLE |
| R-0h | | | | | | | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | SWENABLE |
| R-0h | | | | | | | R-0h |

**Table 33-6. STAT Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-17 | RESERVED | R | 0h | |
| 16 | RESETSTKY | R | 0h | This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register<br>0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register<br>1h = The peripheral was reset since the last bit clear |
| 15-9 | RESERVED | R | 0h | |
| 8 | ENABLE | R | 0h | Peripheral enable status<br>0h = Peripheral is disabled<br>1h = Peripheral is enabled |
| 7-1 | RESERVED | R | 0h | |
| 0 | SWENABLE | R | 0h | Software enable status. Shows if enabled via PWREN.ENABLE bit<br>0h = Not enabled via PWREN.ENABLE bit<br>1h = Enabled via PWREN.ENABLE bit |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.5 IIDX (Offset = 1020h) [Reset = 00000000h]

IIDX is shown in Figure 33-9 and described in Table 33-7.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. 0x0 means no event pending. Interrupt 1 is the highest priority, IIDX next highest, 4, 8, … IIDX^31 is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred.
On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in the RIS and MIS are cleared as well. After a read from the CPU (not from the debug interface), the register must be updated with the next highest priority interrupt, if none are pending, then it should display 0x0.

**Figure 33-9. IIDX**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | STAT | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | |

**Table 33-7. IIDX Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | STAT | R | 0h | Interrupt index status<br>0h = No bit is set means there is no pending interrupt request<br>1h = Endpoint 0 and the TX Endpoints interrupt<br>2h = RX Endpoints interrupt<br>3h = USB Interrupts<br>4h = VUSB power down<br>5h = DMA-Done interrupt for Trigger USB-A-RX<br>6h = DMA-Done interrupt for Trigger USB-A-TX<br>7h = DMA-Done interrupt for Trigger USB-B-RX<br>8h = DMA-Done interrupt for Trigger USB-B-TX<br>9h = DMA-Done interrupt for Trigger USB-C-RX<br>Ah = DMA-Done interrupt for Trigger USB-C-TX<br>Bh = DMA-Done interrupt for Trigger USB-D-RX<br>Ch = DMA-Done interrupt for Trigger USB-D-TX<br>Dh = DMA-Pre interrupt for Trigger USB-A-RX<br>Eh = DMA-Pre interrupt for Trigger USB-A-TX<br>Fh = DMA-Pre interrupt for Trigger USB-B-RX<br>10h = DMA-Pre interrupt for Trigger USB-B-TX<br>11h = DMA-Pre interrupt for Trigger USB-C-RX<br>12h = DMA-Pre interrupt for Trigger USB-C-TX<br>13h = DMA-Pre interrupt for Trigger USB-D-RX<br>14h = DMA-Pre interrupt for Trigger USB-D-TX |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.6 IMASK (Offset = 1028h) [Reset = 00000000h]

IMASK is shown in Figure 33-10 and described in Table 33-8.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS."

**Figure 33-10. IMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | DMAPREDTX | DMAPREDRX | DMAPRECTX | DMAPRECRX |
| R-0h | | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DMAPREBTX | DMAPREBRX | DMAPREATX | DMAPREARX | DMADONEDTX | DMADONEDRX | DMADONECTX | DMADONECRX |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DMADONEBTX | DMADONEBRX | DMADONEATX | DMADONEARX | VUSBPWRDN | INTRUSB | INTRRX | INTRTX |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

**Table 33-8. IMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | RESERVED | R | 0h | |
| 19 | DMAPREDTX | R/W | 0h | DMA-Pre interrupt for Trigger USB-D-TX<br>0h = Interrupt Disabled.<br>1h = Interrupt Enabled. |
| 18 | DMAPREDRX | R/W | 0h | DMA-Pre interrupt for Trigger USB-D-RX<br>0h = Interrupt Disabled.<br>1h = Interrupt Enabled. |
| 17 | DMAPRECTX | R/W | 0h | DMA-Pre interrupt for Trigger USB-C-TX<br>0h = Interrupt Disabled.<br>1h = Interrupt Enabled. |
| 16 | DMAPRECRX | R/W | 0h | DMA-Pre interrupt for Trigger USB-C-RX<br>0h = Interrupt Disabled.<br>1h = Interrupt Enabled. |
| 15 | DMAPREBTX | R/W | 0h | DMA-Pre interrupt for Trigger USB-B-TX<br>0h = Interrupt Disabled.<br>1h = Interrupt Enabled. |
| 14 | DMAPREBRX | R/W | 0h | DMA-Pre interrupt for Trigger USB-B-RX<br>0h = Interrupt Disabled.<br>1h = Interrupt Enabled. |
| 13 | DMAPREATX | R/W | 0h | DMA-Pre interrupt for Trigger USB-A-TX<br>0h = Interrupt Disabled.<br>1h = Interrupt Enabled. |
| 12 | DMAPREARX | R/W | 0h | DMA-Pre interrupt for Trigger USB-A-RX<br>0h = Interrupt Disabled.<br>1h = Interrupt Enabled. |
| 11 | DMADONEDTX | R/W | 0h | DMA-Done interrupt for Trigger USB-D-TX<br>0h = Interrupt Disabled.<br>1h = Interrupt Enabled. |
| 10 | DMADONEDRX | R/W | 0h | DMA-Done interrupt for Trigger USB-D-RX<br>0h = Interrupt Disabled.<br>1h = Interrupt Enabled. |



Copyright © 2026 Texas Instruments Incorporated

**Table 33-8. IMASK Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 9 | DMADONECTX | R/W | 0h | DMA-Done interrupt for Trigger USB-C-TX<br>0h = Interrupt Disabled.<br>1h = Interrupt Enabled. |
| 8 | DMADONECRX | R/W | 0h | DMA-Done interrupt for Trigger USB-C-RX<br>0h = Interrupt Disabled.<br>1h = Interrupt Enabled. |
| 7 | DMADONEBTX | R/W | 0h | DMA-Done interrupt for Trigger USB-B-TX<br>0h = Interrupt Disabled.<br>1h = Interrupt Enabled. |
| 6 | DMADONEBRX | R/W | 0h | DMA-Done interrupt for Trigger USB-B-RX<br>0h = Interrupt Disabled.<br>1h = Interrupt Enabled. |
| 5 | DMADONEATX | R/W | 0h | DMA-Done interrupt for Trigger USB-A-TX<br>0h = Interrupt Disabled.<br>1h = Interrupt Enabled. |
| 4 | DMADONEARX | R/W | 0h | DMA-Done interrupt for Trigger USB-A-RX<br>0h = Interrupt Disabled.<br>1h = Interrupt Enabled. |
| 3 | VUSBPWRDN | R/W | 0h | VUSB Power Down<br>0h = Interrupt Disabled.<br>1h = Interrupt Enabled. |
| 2 | INTRUSB | R/W | 0h | USB Interrupts<br>0h = Interrupt Disabled.<br>1h = Interrupt Enabled. |
| 1 | INTRRX | R/W | 0h | RX Endpoints interrupt<br>0h = Interrupt Disabled.<br>1h = Interrupt Enabled. |
| 0 | INTRTX | R/W | 0h | Endpoint 0 and the TX Endpoints interrupt<br>0h = Interrupt Disabled.<br>1h = Interrupt Enabled. |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.7 RIS (Offset = 1030h) [Reset = 00000000h]

RIS is shown in Figure 33-11 and described in Table 33-9.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Figure 33-11. RIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | DMAPREDTX | DMAPREDRX | DMAPRECTX | DMAPRECRX |
| R-0h | | | | R-0h | R-0h | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DMAPREBTX | DMAPREBRX | DMAPREATX | DMAPREARX | DMADONEDTX | DMADONEDRX | DMADONECTX | DMADONECRX |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DMADONEBTX | DMADONEBRX | DMADONEATX | DMADONEARX | VUSBPWRDN | INTRUSB | INTRRX | INTRTX |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 33-9. RIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | RESERVED | R | 0h | |
| 19 | DMAPREDTX | R | 0h | DMA-Pre interrupt for Trigger USB-D-TX<br>0h = Interrupt did not occur<br>1h = Interrupt occured |
| 18 | DMAPREDRX | R | 0h | DMA-Pre interrupt for Trigger USB-D-RX<br>0h = Interrupt did not occur<br>1h = Interrupt occured |
| 17 | DMAPRECTX | R | 0h | DMA-Pre interrupt for Trigger USB-C-TX<br>0h = Interrupt did not occur<br>1h = Interrupt occured |
| 16 | DMAPRECRX | R | 0h | DMA-Pre interrupt for Trigger USB-C-RX<br>0h = Interrupt did not occur<br>1h = Interrupt occured |
| 15 | DMAPREBTX | R | 0h | DMA-Pre interrupt for Trigger USB-B-TX<br>0h = Interrupt did not occur<br>1h = Interrupt occured |
| 14 | DMAPREBRX | R | 0h | DMA-Pre interrupt for Trigger USB-B-RX<br>0h = Interrupt did not occur<br>1h = Interrupt occured |
| 13 | DMAPREATX | R | 0h | DMA-Pre interrupt for Trigger USB-A-TX<br>0h = Interrupt did not occur<br>1h = Interrupt occured |
| 12 | DMAPREARX | R | 0h | DMA-Pre interrupt for Trigger USB-A-RX<br>0h = Interrupt did not occur<br>1h = Interrupt occured |
| 11 | DMADONEDTX | R | 0h | DMA-Done interrupt for Trigger USB-D-TX<br>0h = Interrupt did not occur<br>1h = Interrupt occured |



Copyright © 2026 Texas Instruments Incorporated

**Table 33-9. RIS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 10 | DMADONEDRX | R | 0h | DMA-Done interrupt for Trigger USB-D-RX<br>0h = Interrupt did not occur<br>1h = Interrupt occured |
| 9 | DMADONECTX | R | 0h | DMA-Done interrupt for Trigger USB-C-TX<br>0h = Interrupt did not occur<br>1h = Interrupt occured |
| 8 | DMADONECRX | R | 0h | DMA-Done interrupt for Trigger USB-C-RX<br>0h = Interrupt did not occur<br>1h = Interrupt occured |
| 7 | DMADONEBTX | R | 0h | DMA-Done interrupt for Trigger USB-B-TX<br>0h = Interrupt did not occur<br>1h = Interrupt occured |
| 6 | DMADONEBRX | R | 0h | DMA-Done interrupt for Trigger USB-B-RX<br>0h = Interrupt did not occur<br>1h = Interrupt occured |
| 5 | DMADONEATX | R | 0h | DMA-Done interrupt for Trigger USB-A-TX<br>0h = Interrupt did not occur<br>1h = Interrupt occured |
| 4 | DMADONEARX | R | 0h | DMA-Done interrupt for Trigger USB-A-RX<br>0h = Interrupt did not occur<br>1h = Interrupt occured |
| 3 | VUSBPWRDN | R | 0h | VUSB Power Down<br>0h = Interrupt did not occur<br>1h = Interrupt occured |
| 2 | INTRUSB | R | 0h | USB Interrupts<br>0h = Interrupt did not occur<br>1h = Interrupt occured |
| 1 | INTRRX | R | 0h | RX Endpoints interrupt<br>0h = Interrupt did not occur<br>1h = Interrupt occured |
| 0 | INTRTX | R | 0h | Endpoint 0 and the TX Endpoints interrupt<br>0h = Interrupt did not occur<br>1h = Interrupt occured |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.8 MIS (Offset = 1038h) [Reset = 00000000h]

MIS is shown in Figure 33-12 and described in Table 33-10.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 33-12. MIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | DMAPREDTX | DMAPREDRX | DMAPRECTX | DMAPRECRX |
| R-0h | | | | R-0h | R-0h | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DMAPREBTX | DMAPREBRX | DMAPREATX | DMAPREARX | DMADONEDTX | DMADONEDRX | DMADONECTX | DMADONECRX |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DMADONEBTX | DMADONEBRX | DMADONEATX | DMADONEARX | VUSBPWRDN | INTRUSB | INTRRX | INTRTX |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 33-10. MIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | RESERVED | R | 0h | |
| 19 | DMAPREDTX | R | 0h | DMA-Pre interrupt for Trigger USB-D-TX<br>0h = Interrupt did not occur<br>1h = Interrupt occured |
| 18 | DMAPREDRX | R | 0h | DMA-Pre interrupt for Trigger USB-D-RX<br>0h = Interrupt did not occur<br>1h = Interrupt occured |
| 17 | DMAPRECTX | R | 0h | DMA-Pre interrupt for Trigger USB-C-TX<br>0h = Interrupt did not occur<br>1h = Interrupt occured |
| 16 | DMAPRECRX | R | 0h | DMA-Pre interrupt for Trigger USB-C-RX<br>0h = Interrupt did not occur<br>1h = Interrupt occured |
| 15 | DMAPREBTX | R | 0h | DMA-Pre interrupt for Trigger USB-B-TX<br>0h = Interrupt did not occur<br>1h = Interrupt occured |
| 14 | DMAPREBRX | R | 0h | DMA-Pre interrupt for Trigger USB-B-RX<br>0h = Interrupt did not occur<br>1h = Interrupt occured |
| 13 | DMAPREATX | R | 0h | DMA-Pre interrupt for Trigger USB-A-TX<br>0h = Interrupt did not occur<br>1h = Interrupt occured |
| 12 | DMAPREARX | R | 0h | DMA-Pre interrupt for Trigger USB-A-RX<br>0h = Interrupt did not occur<br>1h = Interrupt occured |
| 11 | DMADONEDTX | R | 0h | DMA-Done interrupt for Trigger USB-D-TX<br>0h = Interrupt did not occur<br>1h = Interrupt occured |
| 10 | DMADONEDRX | R | 0h | DMA-Done interrupt for Trigger USB-D-RX<br>0h = Interrupt did not occur<br>1h = Interrupt occured |



Copyright © 2026 Texas Instruments Incorporated

**Table 33-10. MIS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 9 | DMADONECTX | R | 0h | DMA-Done interrupt for Trigger USB-C-TX<br>0h = Interrupt did not occur<br>1h = Interrupt occured |
| 8 | DMADONECRX | R | 0h | DMA-Done interrupt for Trigger USB-C-RX<br>0h = Interrupt did not occur<br>1h = Interrupt occured |
| 7 | DMADONEBTX | R | 0h | DMA-Done interrupt for Trigger USB-B-TX<br>0h = Interrupt did not occur<br>1h = Interrupt occured |
| 6 | DMADONEBRX | R | 0h | DMA-Done interrupt for Trigger USB-B-RX<br>0h = Interrupt did not occur<br>1h = Interrupt occured |
| 5 | DMADONEATX | R | 0h | DMA-Done interrupt for Trigger USB-A-TX<br>0h = Interrupt did not occur<br>1h = Interrupt occured |
| 4 | DMADONEARX | R | 0h | DMA-Done interrupt for Trigger USB-A-RX<br>0h = Interrupt did not occur<br>1h = Interrupt occured |
| 3 | VUSBPWRDN | R | 0h | VUSB Power Down<br>0h = Interrupt did not occur<br>1h = Interrupt occured |
| 2 | INTRUSB | R | 0h | USB Interrupts<br>0h = Interrupt did not occur<br>1h = Interrupt occured |
| 1 | INTRRX | R | 0h | RX Endpoints interrupt<br>0h = Interrupt did not occur<br>1h = Interrupt occured |
| 0 | INTRTX | R | 0h | Endpoint 0 and the TX Endpoints interrupt<br>0h = Interrupt did not occur<br>1h = Interrupt occured |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.9 ISET (Offset = 1040h) [Reset = 00000000h]

ISET is shown in Figure 33-13 and described in Table 33-11.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 33-13. ISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | DMAPREDTX | DMAPREDRX | DMAPRECTX | DMAPRECRX |
| R-0h | | | | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DMAPREBTX | DMAPREBRX | DMAPREATX | DMAPREARX | DMADONEDTX | DMADONEDRX | DMADONECTX | DMADONECRX |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DMADONEBTX | DMADONEBRX | DMADONEATX | DMADONEARX | VUSBPWRDN | INTRUSB | INTRRX | INTRTX |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

**Table 33-11. ISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | RESERVED | R | 0h | |
| 19 | DMAPREDTX | W | 0h | DMA-Pre interrupt for Trigger USB-D-TX<br>0h = Writing 0 has No Effect<br>1h = Write 1 to set the interrupt |
| 18 | DMAPREDRX | W | 0h | DMA-Pre interrupt for Trigger USB-D-RX<br>0h = Writing 0 has No Effect<br>1h = Write 1 to set the interrupt |
| 17 | DMAPRECTX | W | 0h | DMA-Pre interrupt for Trigger USB-C-TX<br>0h = Writing 0 has No Effect<br>1h = Write 1 to set the interrupt |
| 16 | DMAPRECRX | W | 0h | DMA-Pre interrupt for Trigger USB-C-RX<br>0h = Writing 0 has No Effect<br>1h = Write 1 to set the interrupt |
| 15 | DMAPREBTX | W | 0h | DMA-Pre interrupt for Trigger USB-B-TX<br>0h = Writing 0 has No Effect<br>1h = Write 1 to set the interrupt |
| 14 | DMAPREBRX | W | 0h | DMA-Pre interrupt for Trigger USB-B-RX<br>0h = Writing 0 has No Effect<br>1h = Write 1 to set the interrupt |
| 13 | DMAPREATX | W | 0h | DMA-Pre interrupt for Trigger USB-A-TX<br>0h = Writing 0 has No Effect<br>1h = Write 1 to set the interrupt |
| 12 | DMAPREARX | W | 0h | DMA-Pre interrupt for Trigger USB-A-RX<br>0h = Writing 0 has No Effect<br>1h = Write 1 to set the interrupt |
| 11 | DMADONEDTX | W | 0h | DMA-Done interrupt for Trigger USB-D-TX<br>0h = Writing 0 has No Effect<br>1h = Write 1 to set the interrupt |



Copyright © 2026 Texas Instruments Incorporated

**Table 33-11. ISET Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 10 | DMADONEDRX | W | 0h | DMA-Done interrupt for Trigger USB-D-RX<br>0h = Writing 0 has No Effect<br>1h = Write 1 to set the interrupt |
| 9 | DMADONECTX | W | 0h | DMA-Done interrupt for Trigger USB-C-TX<br>0h = Writing 0 has No Effect<br>1h = Write 1 to set the interrupt |
| 8 | DMADONECRX | W | 0h | DMA-Done interrupt for Trigger USB-C-RX<br>0h = Writing 0 has No Effect<br>1h = Write 1 to set the interrupt |
| 7 | DMADONEBTX | W | 0h | DMA-Done interrupt for Trigger USB-B-TX<br>0h = Writing 0 has No Effect<br>1h = Write 1 to set the interrupt |
| 6 | DMADONEBRX | W | 0h | DMA-Done interrupt for Trigger USB-B-RX<br>0h = Writing 0 has No Effect<br>1h = Write 1 to set the interrupt |
| 5 | DMADONEATX | W | 0h | DMA-Done interrupt for Trigger USB-A-TX<br>0h = Writing 0 has No Effect<br>1h = Write 1 to set the interrupt |
| 4 | DMADONEARX | W | 0h | DMA-Done interrupt for Trigger USB-A-RX<br>0h = Writing 0 has No Effect<br>1h = Write 1 to set the interrupt |
| 3 | VUSBPWRDN | W | 0h | VUSB Power Down<br>0h = Writing 0 has No Effect<br>1h = Write 1 to set the interrupt |
| 2 | INTRUSB | W | 0h | USB Interrupts<br>0h = Writing 0 has No Effect<br>1h = Write 1 to set the interrupt |
| 1 | INTRRX | W | 0h | RX Endpoints interrupt<br>0h = Writing 0 has No Effect<br>1h = Write 1 to set the interrupt |
| 0 | INTRTX | W | 0h | Endpoint 0 and the TX Endpoints interrupt<br>0h = Writing 0 has No Effect<br>1h = Write 1 to set the interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.10 ICLR (Offset = 1048h) [Reset = 00000000h]

ICLR is shown in Figure 33-14 and described in Table 33-12.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

**Figure 33-14. ICLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | DMAPREDTX | DMAPREDRX | DMAPRECTX | DMAPRECRX |
| R-0h | | | | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DMAPREBTX | DMAPREBRX | DMAPREATX | DMAPREARX | DMADONEDTX | DMADONEDRX | DMADONECTX | DMADONECRX |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DMADONEBTX | DMADONEBRX | DMADONEATX | DMADONEARX | VUSBPWRDN | INTRUSB | INTRRX | INTRTX |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

**Table 33-12. ICLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | RESERVED | R | 0h | |
| 19 | DMAPREDTX | W | 0h | DMA-Pre interrupt for Trigger USB-D-TX<br>0h = Writing 0 has No Effect<br>1h = Write 1 to clear the interrupt |
| 18 | DMAPREDRX | W | 0h | DMA-Pre interrupt for Trigger USB-D-RX<br>0h = Writing 0 has No Effect<br>1h = Write 1 to clear the interrupt |
| 17 | DMAPRECTX | W | 0h | DMA-Pre interrupt for Trigger USB-C-TX<br>0h = Writing 0 has No Effect<br>1h = Write 1 to clear the interrupt |
| 16 | DMAPRECRX | W | 0h | DMA-Pre interrupt for Trigger USB-C-RX<br>0h = Writing 0 has No Effect<br>1h = Write 1 to clear the interrupt |
| 15 | DMAPREBTX | W | 0h | DMA-Pre interrupt for Trigger USB-B-TX<br>0h = Writing 0 has No Effect<br>1h = Write 1 to clear the interrupt |
| 14 | DMAPREBRX | W | 0h | DMA-Pre interrupt for Trigger USB-B-RX<br>0h = Writing 0 has No Effect<br>1h = Write 1 to clear the interrupt |
| 13 | DMAPREATX | W | 0h | DMA-Pre interrupt for Trigger USB-A-TX<br>0h = Writing 0 has No Effect<br>1h = Write 1 to clear the interrupt |
| 12 | DMAPREARX | W | 0h | DMA-Pre interrupt for Trigger USB-A-RX<br>0h = Writing 0 has No Effect<br>1h = Write 1 to clear the interrupt |
| 11 | DMADONEDTX | W | 0h | DMA-Done interrupt for Trigger USB-D-TX<br>0h = Writing 0 has No Effect<br>1h = Write 1 to clear the interrupt |
| 10 | DMADONEDRX | W | 0h | DMA-Done interrupt for Trigger USB-D-RX<br>0h = Writing 0 has No Effect<br>1h = Write 1 to clear the interrupt |



Copyright © 2026 Texas Instruments Incorporated

**Table 33-12. ICLR Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 9 | DMADONECTX | W | 0h | DMA-Done interrupt for Trigger USB-C-TX<br>0h = Writing 0 has No Effect<br>1h = Write 1 to clear the interrupt |
| 8 | DMADONECRX | W | 0h | DMA-Done interrupt for Trigger USB-C-RX<br>0h = Writing 0 has No Effect<br>1h = Write 1 to clear the interrupt |
| 7 | DMADONEBTX | W | 0h | DMA-Done interrupt for Trigger USB-B-TX<br>0h = Writing 0 has No Effect<br>1h = Write 1 to clear the interrupt |
| 6 | DMADONEBRX | W | 0h | DMA-Done interrupt for Trigger USB-B-RX<br>0h = Writing 0 has No Effect<br>1h = Write 1 to clear the interrupt |
| 5 | DMADONEATX | W | 0h | DMA-Done interrupt for Trigger USB-A-TX<br>0h = Writing 0 has No Effect<br>1h = Write 1 to clear the interrupt |
| 4 | DMADONEARX | W | 0h | DMA-Done interrupt for Trigger USB-A-RX<br>0h = Writing 0 has No Effect<br>1h = Write 1 to clear the interrupt |
| 3 | VUSBPWRDN | W | 0h | VUSB Power Down<br>0h = Writing 0 has No Effect<br>1h = Write 1 to clear the interrupt |
| 2 | INTRUSB | W | 0h | USB Interrupts<br>0h = Writing 0 has No Effect<br>1h = Write 1 to clear the interrupt |
| 1 | INTRRX | W | 0h | RX Endpoints interrupt<br>0h = Writing 0 has No Effect<br>1h = Write 1 to clear the interrupt |
| 0 | INTRTX | W | 0h | Endpoint 0 and the TX Endpoints interrupt<br>0h = Writing 0 has No Effect<br>1h = Write 1 to clear the interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.11 EVT_MODE (Offset = 10E0h) [Reset = 00000000h]

EVT_MODE is shown in Figure 33-15 and described in Table 33-13.

Return to the Summary Table.

Event mode register. It is used to select whether each line is disabled, in software mode (software clears the RIS) or in hardware mode (hardware clears the RIS)

**Figure 33-15. EVT_MODE**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | EVT0_CFG | |
| R-0h | | | | | | R-0h | |

**Table 33-13. EVT_MODE Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | |
| 1-0 | EVT0_CFG | R | 1h | Event line mode select for event corresponding to (IPSTANDARD.INT_EVENT)(0)<br>0h = The interrupt or event line is disabled.<br>1h = The interrupt or event line is in software mode. Software must clear the RIS.<br>2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag. |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.12 DESC (Offset = 10FCh) [Reset = 00000000h]

DESC is shown in Figure 33-16 and described in Table 33-14.

Return to the Summary Table.

This register identifies the peripheral and its exact version.

**Figure 33-16. DESC**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MODULEID | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| FEATUREVER | | | | RESERVED | | | | MAJREV | | | | MINREV | | | |
| R-0h | | | | R-0h | | | | R-0h | | | | R-0h | | | |

**Table 33-14. DESC Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | MODULEID | R | 1011h | Module identification contains a unique peripheral identification number. The assignments are maintained in a central database for all of the platform modules to ensure uniqueness.<br>0h = Smallest value<br>FFFFh = Highest possible value |
| 15-12 | FEATUREVER | R | 0h | Feature Set for the module *instance*<br>0h = Smallest value<br>Fh = Highest possible value |
| 11-8 | RESERVED | R | 0h | |
| 7-4 | MAJREV | R | 0h | Major rev of the IP<br>0h = Smallest value<br>Fh = Highest possible value |
| 3-0 | MINREV | R | 0h | Minor rev of the IP<br>0h = Smallest value<br>Fh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.13 USBMODE (Offset = 1100h) [Reset = 00000000h]

USBMODE is shown in Figure 33-17 and described in Table 33-15.

Return to the Summary Table.

USB Mode control

**Figure 33-17. USBMODE**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | PHYMODE | RESERVED | | DEVICEONLY | HOST |
| R-0h | | | R/W-0h | R-0h | | R/W-0h | R/W-0h |

**Table 33-15. USBMODE Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-5 | RESERVED | R | 0h | |
| 4 | PHYMODE | R/W | 0h | USB PHY mode. Defines whether DP/DM pins are controled by USB module or IOMUX/GPIO module.<br>0h = The DP/DM pins are assighed to IOMUX/GPIO module,<br>1h = The DP/DM pins are assighed to USB module. |
| 3-2 | RESERVED | R | 0h | |
| 1 | DEVICEONLY | R/W | 0h | USB is configured as DEVICE_ONLY mode<br>0h = USB is configured as DEVICE or HOST.<br>1h = USB is configured as DEVICE only mode. |
| 0 | HOST | R/W | 0h | USB is configured as HOST mode<br>0h = USB is configured as DEVICE or HOST.<br>1h = USB is configured as HOST |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.14 USBMONITOR (Offset = 1104h) [Reset = 00000000h]

USBMONITOR is shown in Figure 33-18 and described in Table 33-16.

Return to the Summary Table.

USB voltage monitor control

**Figure 33-18. USBMONITOR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | VUSBEN |
| R-0h | | | | | | | R/W-0h |

**Table 33-16. USBMONITOR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | VUSBEN | R/W | 0h | Enable the 1/3 resistor divider for VUSB supply. Set this bit to monitor the VUSB supply with the ADC.<br>0h = VUSB monitor disabled<br>1h = VUSB monitor enabled |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.15 USBDMASEL (Offset = 1108h) [Reset = 00000000h]

USBDMASEL is shown in Figure 33-19 and described in Table 33-17.

Return to the Summary Table.

USB DMA Trigger Select

**Figure 33-19. USBDMASEL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | TRIGDTX | | | RESERVED | TRIGDRX | | |
| R-0h | R/W-0h | | | R-0h | R/W-0h | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | TRIGCTX | | | RESERVED | TRIGCRX | | |
| R-0h | R/W-0h | | | R-0h | R/W-0h | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | TRIGBTX | | | RESERVED | TRIGBRX | | |
| R-0h | R/W-0h | | | R-0h | R/W-0h | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | TRIGATX | | | RESERVED | TRIGARX | | |
| R-0h | R/W-0h | | | R-0h | R/W-0h | | |

**Table 33-17. USBDMASEL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | RESERVED | R | 0h | |
| 30-28 | TRIGDTX | R/W | 0h | TRIG D TX select specifies the TX mapping of the USB endpoint on DMA trigger USB-D-TX<br>0h (R/W) = Disabled<br>1h (R/W) = Endpoint 1<br>2h (R/W) = Endpoint 2<br>3h (R/W) = Endpoint 3<br>4h (R/W) = Endpoint 4<br>5h (R/W) = Endpoint 5<br>6h (R/W) = Endpoint 6<br>7h (R/W) = Endpoint 7 |
| 27 | RESERVED | R | 0h | |
| 26-24 | TRIGDRX | R/W | 0h | TRIG D RX select specifies the RX mapping of the USB endpoint on DMA trigger USB-D-RX<br>0h (R/W) = Disabled<br>1h (R/W) = Endpoint 1<br>2h (R/W) = Endpoint 2<br>3h (R/W) = Endpoint 3<br>4h (R/W) = Endpoint 4<br>5h (R/W) = Endpoint 5<br>6h (R/W) = Endpoint 6<br>7h (R/W) = Endpoint 7 |
| 23 | RESERVED | R | 0h | |
| 22-20 | TRIGCTX | R/W | 0h | TRIG C TX select specifies the TX mapping of the USB endpoint on DMA trigger USB-C-TX<br>0h (R/W) = Disabled<br>1h (R/W) = Endpoint 1<br>2h (R/W) = Endpoint 2<br>3h (R/W) = Endpoint 3<br>4h (R/W) = Endpoint 4<br>5h (R/W) = Endpoint 5<br>6h (R/W) = Endpoint 6<br>7h (R/W) = Endpoint 7 |
| 19 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

**Table 33-17. USBDMASEL Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 18-16 | TRIGCRX | R/W | 0h | TRIG C RX select specifies the RX mapping of the USB endpoint on DMA trigger USB-C-RX<br>0h (R/W) = Disabled<br>1h (R/W) = Endpoint 1<br>2h (R/W) = Endpoint 2<br>3h (R/W) = Endpoint 3<br>4h (R/W) = Endpoint 4<br>5h (R/W) = Endpoint 5<br>6h (R/W) = Endpoint 6<br>7h (R/W) = Endpoint 7 |
| 15 | RESERVED | R | 0h | |
| 14-12 | TRIGBTX | R/W | 0h | TRIG B TX select specifies the TX mapping of the USB endpoint on DMA trigger USB-B-TX<br>0h (R/W) = Disabled<br>1h (R/W) = Endpoint 1<br>2h (R/W) = Endpoint 2<br>3h (R/W) = Endpoint 3<br>4h (R/W) = Endpoint 4<br>5h (R/W) = Endpoint 5<br>6h (R/W) = Endpoint 6<br>7h (R/W) = Endpoint 7 |
| 11 | RESERVED | R | 0h | |
| 10-8 | TRIGBRX | R/W | 0h | TRIG B RX select specifies the RX mapping of the USB endpoint on DMA trigger USB-B-RX<br>0h (R/W) = Disabled<br>1h (R/W) = Endpoint 1<br>2h (R/W) = Endpoint 2<br>3h (R/W) = Endpoint 3<br>4h (R/W) = Endpoint 4<br>5h (R/W) = Endpoint 5<br>6h (R/W) = Endpoint 6<br>7h (R/W) = Endpoint 7 |
| 7 | RESERVED | R | 0h | |
| 6-4 | TRIGATX | R/W | 0h | TRIG A TX select specifies the TX mapping of the USB endpoint on DMA trigger USB-A-TX<br>0h (R/W) = Disabled<br>1h (R/W) = Endpoint 1<br>2h (R/W) = Endpoint 2<br>3h (R/W) = Endpoint 3<br>4h (R/W) = Endpoint 4<br>5h (R/W) = Endpoint 5<br>6h (R/W) = Endpoint 6<br>7h (R/W) = Endpoint 7 |
| 3 | RESERVED | R | 0h | |
| 2-0 | TRIGARX | R/W | 0h | TRIG A RX select specifies the RX mapping of the USB endpoint on DMA trigger USB-A-RX<br>0h (R/W) = Disabled<br>1h (R/W) = Endpoint 1<br>2h (R/W) = Endpoint 2<br>3h (R/W) = Endpoint 3<br>4h (R/W) = Endpoint 4<br>5h (R/W) = Endpoint 5<br>6h (R/W) = Endpoint 6<br>7h (R/W) = Endpoint 7 |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.16 FADDR (Offset = 2000h) [Reset = 00h]

FADDR is shown in Figure 33-20 and described in Table 33-18.

Return to the Summary Table.

Functional Address.
FADDR is an 8-bit register that should be written with the 7-bit address of the peripheral part of the transaction. When the USB controller is being used in device mode, this register should be written with the address received through a SET_ADDRESS command, which will then be used for decoding the function address in subsequent token packets.
Note: Device mode only

**Figure 33-20. FADDR**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RESERVED | FUNCADDR | | | | | | |
| R-0h | R/W-0h | | | | | | |

**Table 33-18. FADDR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7 | RESERVED | R | 0h | |
| 6-0 | FUNCADDR | R/W | 0h | FAddr is an 8-bit register that should be written with the 7-bit address of the peripheral part of the transaction.<br>When the USB module is being used in Peripheral mode (DevCtl.D2=0), this register should be written with the address received through a SET_ADDRESS command, which will then be used for decoding the function address in subsequent token packets.<br>Notes: Peripheral Mode Only!! |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.17 POWER (Offset = 2001h) [Reset = 00h]

POWER is shown in Figure 33-21 and described in Table 33-19.

Return to the Summary Table.

USB Power control.
POWER is an 8-bit register that is used for controlling Suspend and Resume signaling, and some basic operational aspects of the
USB controller.

**Figure 33-21. POWER**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| ISOUP | SOFT_CONN | RESERVED | | RESET | RESUME | SUSPEND | ENSUSPENDM |
| R/W-0h | R/W-0h | R-0h | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

**Table 33-19. POWER Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7 | ISOUP | R/W | 0h | Isochronous Update.<br>When set by the CPU, the USB module will wait for an SOF token from the time TXRDY bit is set in the USBTXCSRLn before sending the packet. If an IN token is received before an SOF token, then a zero length data packet will be sent.<br>Note: Only valid in Peripheral Mode. Also, this bit only affects endpoints performing Isochronous transfers.<br>0h (R/W) = No effect<br>1h (R/W) = The USB controller waits for an SOF token. |
| 6 | SOFT_CONN | R/W | 0h | Soft Connect/Disconnect feature.<br>The USB D+/D- lines are enabled when this bit is set by the CPU and tri-stated when this bit is cleared by the CPU.<br>Note: Only valid in Peripheral Mode.<br>0h (R/W) = The USB D+/D- lines are tri-stated.<br>1h (R/W) = The USB D+/D- lines are enabled. |
| 5-4 | RESERVED | R | 0h | |
| 3 | RESET | R/W | 0h | This bit is set when Reset signaling is present on the bus.<br>Note: This bit is Read/Write from the CPU in Host Mode but Read-Only in Peripheral Mode.<br>0h (R/W) = Ends RESET signaling on the bus.<br>1h (R/W) = Enables RESET signaling on the bus. |
| 2 | RESUME | R/W | 0h | Set by the CPU to generate Resume signaling when the device is in Suspend mode.<br>In Peripheral mode, the CPU should clear this bit after 10 ms (a maximum of 15 ms), to end Resume signaling.<br>In Host mode, the CPU should clear this bit after 20 ms.<br>0h (R/W) = Ends RESUME signaling on the bus.<br>1h (R/W) = Enables RESUME signaling when the Device is in SUSPEND mode. |
| 1 | SUSPEND | R/W | 0h | In Host mode, this bit is set by the CPU to enter Suspend mode.<br>In Peripheral mode, this bit is set on entry into Suspend mode. It is cleared when the CPU reads the interrupt register, or sets the Resume bit.<br>0h (R/W) = Ends SUSPEND signaling<br>1h (R/W) = Enables SUSPEND signaling |
| 0 | ENSUSPENDM | R/W | 0h | Set by the CPU to enable the Suspend mode for the USB PHY.<br>0h (R/W) = USB PHY is powered<br>1h (R/W) = Powers down the internal USB PHY. |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.18 TXIS (Offset = 2002h) [Reset = 0000h]

TXIS is shown in Figure 33-22 and described in Table 33-20.

Return to the Summary Table.

Transmit Interrupt Status.
TXIS is a 16-bit read-only register that indicates which interrupts are currently active for Endpoint 0 and the TX Endpoints 1–15.
Note: Bits relating to endpoints that have not been configured will always return 0.
Note: All active interrupts are cleared when this register is read.

**Figure 33-22. TXIS**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| EP7 | EP6 | EP5 | EP4 | EP3 | EP2 | EP1 | EP0 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 33-20. TXIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15-8 | RESERVED | R | 0h | |
| 7 | EP7 | R | 0h | Transmit Endpoint 7 Interrupt<br>0h (R/W) = No interrupt<br>1h (R/W) = The Endpoint 7 transmit interrupt is asserted. |
| 6 | EP6 | R | 0h | Transmit Endpoint 6 Interrupt<br>0h (R/W) = No interrupt<br>1h (R/W) = The Endpoint 6 transmit interrupt is asserted. |
| 5 | EP5 | R | 0h | Transmit Endpoint 5 Interrupt<br>0h (R/W) = No interrupt<br>1h (R/W) = The Endpoint 5 transmit interrupt is asserted. |
| 4 | EP4 | R | 0h | Transmit Endpoint 4 Interrupt<br>0h (R/W) = No interrupt<br>1h (R/W) = The Endpoint 4 transmit interrupt is asserted. |
| 3 | EP3 | R | 0h | Transmit Endpoint 3 Interrupt<br>0h (R/W) = No interrupt<br>1h (R/W) = The Endpoint 3 transmit interrupt is asserted. |
| 2 | EP2 | R | 0h | Transmit Endpoint 2 Interrupt<br>0h (R/W) = No interrupt<br>1h (R/W) = The Endpoint 2 transmit interrupt is asserted. |
| 1 | EP1 | R | 0h | Transmit Endpoint 1 Interrupt<br>0h (R/W) = No interrupt<br>1h (R/W) = The Endpoint 1 transmit interrupt is asserted. |
| 0 | EP0 | R | 0h | Transmit Endpoint 0 Interrupt<br>0h (R/W) = No interrupt<br>1h (R/W) = The Endpoint 0 transmit and receive interrupt is asserted. |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.19 RXIS (Offset = 2004h) [Reset = 0000h]

RXIS is shown in Figure 33-23 and described in Table 33-21.

Return to the Summary Table.

Receive Interrupt Status.
RXIS is a 16-bit read-only register that indicates which of the interrupts for Rx Endpoints 1 – 15 are currently active.
Note: Bits relating to endpoints that have not been configured will always return 0.
Note: All active interrupts are cleared when this register is read.

**Figure 33-23. RXIS**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| EP7 | EP6 | EP5 | EP4 | EP3 | EP2 | EP1 | RESERVED |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 33-21. RXIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15-8 | RESERVED | R | 0h | |
| 7 | EP7 | R | 0h | Recieve Endpoint 7 Interrupt<br>0h (R/W) = No interrupt<br>1h (R/W) = The Endpoint 7 transmit interrupt is asserted. |
| 6 | EP6 | R | 0h | Recieve Endpoint 6 Interrupt<br>0h (R/W) = No interrupt<br>1h (R/W) = The Endpoint 6 transmit interrupt is asserted. |
| 5 | EP5 | R | 0h | Recieve Endpoint 5 Interrupt<br>0h (R/W) = No interrupt<br>1h (R/W) = The Endpoint 5 transmit interrupt is asserted. |
| 4 | EP4 | R | 0h | Recieve Endpoint 4 Interrupt<br>0h (R/W) = No interrupt<br>1h (R/W) = The Endpoint 4 transmit interrupt is asserted. |
| 3 | EP3 | R | 0h | Recieve Endpoint 3 Interrupt<br>0h (R/W) = No interrupt<br>1h (R/W) = The Endpoint 3 transmit interrupt is asserted. |
| 2 | EP2 | R | 0h | Recieve Endpoint 2 Interrupt<br>0h (R/W) = No interrupt<br>1h (R/W) = The Endpoint 2 transmit interrupt is asserted. |
| 1 | EP1 | R | 0h | Recieve Endpoint 1 Interrupt<br>0h (R/W) = No interrupt<br>1h (R/W) = The Endpoint 1 transmit interrupt is asserted. |
| 0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.20 TXIE (Offset = 2006h) [Reset = 00FFh]

TXIE is shown in Figure 33-24 and described in Table 33-22.

Return to the Summary Table.

Transmit Interrupt Enable.
TXIE is a 16-bit register that provides interrupt enable bits for the interrupts in TXIS. Where a bit is set to 1, USB_CPU_INT.RIS.INTRTX will be set on the corresponding interrupt in the TXIS register becoming set. Where a bit is set to 0, the interrupt in TXIE is still set but USB_CPU_INT.RIS.INTRTX is not set.
Note: On reset, the bits corresponding to Endpoint 0 and the TX endpoints included in the USB controller are set to 1, while the remaining bits are set to 0.

**Figure 33-24. TXIE**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| EP7 | EP6 | EP5 | EP4 | EP3 | EP2 | EP1 | EP0 |
| R/W-1h | R/W-1h | R/W-1h | R/W-1h | R/W-1h | R/W-1h | R/W-1h | R/W-1h |

**Table 33-22. TXIE Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15-8 | RESERVED | R | 0h | |
| 7 | EP7 | R/W | 1h | Transmit Endpoint 7 Interrupt Enable<br>0h (R/W) = The EP7 transmit interrupt is suppressed and not sent to the interrupt controller.<br>1h (R/W) = An interrupt is sent to the interrupt controller when the EP7 bit in the USBTXIS register is set. |
| 6 | EP6 | R/W | 1h | Transmit Endpoint 6 Interrupt Enable<br>0h (R/W) = The EP6 transmit interrupt is suppressed and not sent to the interrupt controller.<br>1h (R/W) = An interrupt is sent to the interrupt controller when the EP6 bit in the USBTXIS register is set. |
| 5 | EP5 | R/W | 1h | Transmit Endpoint 5 Interrupt Enable<br>0h (R/W) = The EP5 transmit interrupt is suppressed and not sent to the interrupt controller.<br>1h (R/W) = An interrupt is sent to the interrupt controller when the EP5 bit in the USBTXIS register is set. |
| 4 | EP4 | R/W | 1h | Transmit Endpoint 4 Interrupt Enable<br>0h (R/W) = The EP4 transmit interrupt is suppressed and not sent to the interrupt controller.<br>1h (R/W) = An interrupt is sent to the interrupt controller when the EP4 bit in the USBTXIS register is set. |
| 3 | EP3 | R/W | 1h | Transmit Endpoint 3 Interrupt Enable<br>0h (R/W) = The EP3 transmit interrupt is suppressed and not sent to the interrupt controller.<br>1h (R/W) = An interrupt is sent to the interrupt controller when the EP3 bit in the USBTXIS register is set. |
| 2 | EP2 | R/W | 1h | Transmit Endpoint 2 Interrupt Enable<br>0h (R/W) = The EP2 transmit interrupt is suppressed and not sent to the interrupt controller.<br>1h (R/W) = An interrupt is sent to the interrupt controller when the EP2 bit in the USBTXIS register is set. |
| 1 | EP1 | R/W | 1h | Transmit Endpoint 1 Interrupt Enable<br>0h (R/W) = The EP1 transmit interrupt is suppressed and not sent to the interrupt controller.<br>1h (R/W) = An interrupt is sent to the interrupt controller when the EP1 bit in the USBTXIS register is set. |



Copyright © 2026 Texas Instruments Incorporated

**Table 33-22. TXIE Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 0 | EP0 | R/W | 1h | Transmit Endpoint 0 Interrupt Enable<br>0h (R/W) = The EP0 transmit and receive interrupt is suppressed and not sent to the interupt controller.<br>1h (R/W) = An interrupt is sent to the interrupt controller when the EP0 bit in the USBTXIS register is set. |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.21 RXIE (Offset = 2008h) [Reset = 00FEh]

RXIE is shown in Figure 33-25 and described in Table 33-23.

Return to the Summary Table.

Receive Interrupt Enable.
RXIE is a 16-bit register that provides interrupt enable bits for the interrupts in RXIS. Where a bit is set to 1, USB_CPU_INT.RIS.INTRRX will be set on the corresponding interrupt in the RXIS register becoming set. Where a bit is set to 0, the interrupt in RXIS is still set but USB_CPU_INT.RIS.INTRRX is not set.
Note: On reset, the bits corresponding to the Rx endpoints included in the USB controller are set to 1, while the remaining bits are set to 0.

**Figure 33-25. RXIE**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| EP7 | EP6 | EP5 | EP4 | EP3 | EP2 | EP1 | RESERVED |
| R/W-1h | R/W-1h | R/W-1h | R/W-1h | R/W-1h | R/W-1h | R/W-1h | R-0h |

**Table 33-23. RXIE Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15-8 | RESERVED | R | 0h | |
| 7 | EP7 | R/W | 1h | Recieve Endpoint 7 Interrupt Enable<br>0h (R/W) = No interrupt<br>1h (R/W) = The Endpoint 7 transmit interrupt is asserted. |
| 6 | EP6 | R/W | 1h | Recieve Endpoint 6 Interrupt Enable<br>0h (R/W) = No interrupt<br>1h (R/W) = The Endpoint 6 transmit interrupt is asserted. |
| 5 | EP5 | R/W | 1h | Recieve Endpoint 5 Interrupt Enable<br>0h (R/W) = No interrupt<br>1h (R/W) = The Endpoint 5 transmit interrupt is asserted. |
| 4 | EP4 | R/W | 1h | Recieve Endpoint 4 Interrupt Enable<br>0h (R/W) = No interrupt<br>1h (R/W) = The Endpoint 4 transmit interrupt is asserted. |
| 3 | EP3 | R/W | 1h | Recieve Endpoint 3 Interrupt Enable<br>0h (R/W) = No interrupt<br>1h (R/W) = The Endpoint 3 transmit interrupt is asserted. |
| 2 | EP2 | R/W | 1h | Recieve Endpoint 2 Interrupt Enable<br>0h (R/W) = No interrupt<br>1h (R/W) = The Endpoint 2 transmit interrupt is asserted. |
| 1 | EP1 | R/W | 1h | Recieve Endpoint 1 Interrupt Enable<br>0h (R/W) = No interrupt<br>1h (R/W) = The Endpoint 1 transmit interrupt is asserted. |
| 0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.22 USBIS (Offset = 200Ah) [Reset = 00h]

USBIS is shown in Figure 33-26 and described in Table 33-24.

Return to the Summary Table.

USB Interrupt Status.
USBIS is an 8-bit read-only register that indicates which USB interrupts are currently active.
Note: All active interrupts will be cleared when this register is read.

**Figure 33-26. USBIS**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RESERVED | | DISCON | CONN | SOF | RESETBABBLE | RESUME | SUSPEND |
| R-0h | | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 33-24. USBIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7-6 | RESERVED | R | 0h | |
| 5 | DISCON | R | 0h | Set in Host mode when a device disconnect is detected.<br>Set in Peripheral mode when a session ends. Valid at all transaction speeds.<br>0h (R/W) = No interrupt<br>1h (R/W) = The device has been disconnected from the host. |
| 4 | CONN | R | 0h | Set when a device connection is detected.<br>Note: Only valid in Host mode. Valid at all transaction speeds.<br>0h (R/W) = No interrupt<br>1h (R/W) = The device has detected a connection to the host. |
| 3 | SOF | R | 0h | Start of frame<br>0h (R/W) = No interrupt<br>1h (R/W) = A new frame has started. |
| 2 | RESETBABBLE | R | 0h | Device mode - RESET: Set in Peripheral mode when Reset signaling is detected on the bus.<br>Host mode - BABBLE: Set in Host mode when babble is detected.<br>Note: Only active after first SOF has been sent.<br>0h (R/W) = No interrupt<br>1h (R/W) = RESET / BABBLE signaling has been detected on the bus. |
| 1 | RESUME | R | 0h | Set when Resume signaling is detected on the bus while the USB controller is in Suspend mode.<br>0h (R/W) = No interrupt<br>1h (R/W) = RESUME signaling has been detected on the bus. |
| 0 | SUSPEND | R | 0h | Set when Suspend signaling is detected on the bus.<br>Note: Only valid in Peripheral mode.<br>0h (R/W) = No interrupt<br>1h (R/W) = SUSPEND signaling has been detected on the bus. |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.23 USBIE (Offset = 200Bh) [Reset = 06h]

USBIE is shown in Figure 33-27 and described in Table 33-25.

Return to the Summary Table.

USB Interrupt Enable.
USBIE is an 8-bit register that provides interrupt enable bits for each of the interrupts in USBIS.

**Figure 33-27. USBIE**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RESERVED | | DISCON | CONN | SOF | RESETBABBLE | RESUME | SUSPEND |
| R-0h | | R/W-0h | R/W-0h | R/W-0h | R/W-1h | R/W-1h | R-0h |

**Table 33-25. USBIE Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7-6 | RESERVED | R | 0h | |
| 5 | DISCON | R/W | 0h | Interrupt enable for session disconnect.<br>0h (R/W) = The DISCON interrupt is suppressed and not sent to the interrupt controller.<br>1h (R/W) = An interrupt is sent to the interrupt controller when the DISCON bit in the USBIS register is set. |
| 4 | CONN | R/W | 0h | Interrupt enable for connect detected.<br>Note: Only valid in Host mode. Valid at all transaction speeds.<br>0h (R/W) = The CONN interrupt is suppressed and not sent to the interrupt controller.<br>1h (R/W) = An interrupt is sent to the interrupt controller when the CONN bit in the USBIS register is set. |
| 3 | SOF | R/W | 0h | Interrupt enable for Start of frame.<br>0h (R/W) = The SOF interrupt is suppressed and not sent to the interrupt controller.<br>1h (R/W) = An interrupt is sent to the interrupt controller when the SOF bit in the USBIS register is set. |
| 2 | RESETBABBLE | R/W | 1h | Interrupt enable for RESET/BABBLE signaling detected.<br>0h (R/W) = The RESET interrupt is suppressed and not sent to the interrupt controller.<br>1h (R/W) = An interrupt is sent to the interrupt controller when the RESET bit in the USBIS register is set. |
| 1 | RESUME | R/W | 1h | Interrupt enable for RESUME signaling detected.<br>0h (R/W) = The RESUME interrupt is suppressed and not sent to the interrupt controller.<br>1h (R/W) = An interrupt is sent to the interrupt controller when the RESUME bit in the USBIS register is set. |
| 0 | SUSPEND | R | 0h | Interrupt enable for SUSPEND signaling detected.<br>Note: Only valid in Peripheral mode.<br>0h (R/W) = The SUSPEND interrupt is suppressed and not sent to the interrupt controller.<br>1h (R/W) = An interrupt is sent to the interrupt controller when the DISCON bit in the USBIS register is set. |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.24 FRAME (Offset = 200Ch) [Reset = 0000h]

FRAME is shown in Figure 33-28 and described in Table 33-26.

Return to the Summary Table.

Frame Number.
Frame is a 16-bit read-only register that holds the last received frame number.

**Figure 33-28. FRAME**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | FRAME | | |
| R-0h | | | | | R-0h | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| FRAME | | | | | | | |
| R-0h | | | | | | | |

**Table 33-26. FRAME Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15-11 | RESERVED | R | 0h | |
| 10-0 | FRAME | R | 0h | Frame Number |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.25 EPINDEX (Offset = 200Eh) [Reset = 00h]

EPINDEX is shown in Figure 33-29 and described in Table 33-27.

Return to the Summary Table.

Endpoint Index.
Each TX endpoint and each Rx endpoint have their own set of control/status registers located between 100h – 1FFh. In addition one set of TX control/status and one set of Rx control/status registers appear at 10h – 19h. Index is a 4-bit register that determines which endpoint control/status registers are accessed.
Note: Before accessing an endpoint's control/status registers at 10h – 19h, the endpoint number should be written to the Index register to ensure that the correct control/status registers appear in the memory map.

**Figure 33-29. EPINDEX**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | EPIDX | | | |
| R-0h | | | | R/W-0h | | | |

**Table 33-27. EPINDEX Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7-4 | RESERVED | R | 0h | |
| 3-0 | EPIDX | R/W | 0h | Endpoint Index.<br>This bit field configures which endpoint is accessed when reading or writing to one of the USB controller's indexed registers. A value of 0x0 corresponds to Endpoint 0 and a value of 0xF corresponds to Endpoint 15.<br>0h = Select EP0 as indexed register<br>1h = Select EP1 as indexed register<br>2h = Select EP2 as indexed register<br>3h = Select EP3 as indexed register<br>4h = Select EP4 as indexed register<br>5h = Select EP5 as indexed register<br>6h = Select EP6 as indexed register<br>7h = Select EP7 as indexed register |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.26 IDXTXMAXP (Offset = 2010h) [Reset = X000h]

IDXTXMAXP is shown in Figure 33-30 and described in Table 33-28.

Return to the Summary Table.

Indexed access to register TXMAXP (Maximum Transmit Data).
USB.EPINDEX holds the index pointing to the Endpoint for this register.

**Figure 33-30. IDXTXMAXP**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| DATA | | | | | | | |
| R/W-Xh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DATA | | | | | | | |
| R/W-Xh | | | | | | | |

**Table 33-28. IDXTXMAXP Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15-0 | DATA | R/W | Xh | Control / Status register data. |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.27 IDXTXCSRL (Offset = 2012h) [Reset = X0h]

IDXTXCSRL is shown in Figure 33-31 and described in Table 33-29.

Return to the Summary Table.

Indexed access to register TXCSRL (Transmit Control and Status Low).
USB.EPINDEX holds the index pointing to the Endpoint for this register.

**Figure 33-31. IDXTXCSRL**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| DATA | | | | | | | |
| R/W-Xh | | | | | | | |

**Table 33-29. IDXTXCSRL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7-0 | DATA | R/W | Xh | Control / Status register data. |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.28 IDXTXCSRH (Offset = 2013h) [Reset = X0h]

IDXTXCSRH is shown in Figure 33-32 and described in Table 33-30.

Return to the Summary Table.

Indexed access to register TXCSRH (Transmit Control and Status High).
USB.EPINDEX holds the index pointing to the Endpoint for this register.

**Figure 33-32. IDXTXCSRH**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| DATA | | | | | | | |
| R/W-Xh | | | | | | | |

**Table 33-30. IDXTXCSRH Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7-0 | DATA | R/W | Xh | Control / Status register data. |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.29 IDXRXMAXP (Offset = 2014h) [Reset = X000h]

IDXRXMAXP is shown in Figure 33-33 and described in Table 33-31.

Return to the Summary Table.

Indexed access to register RXMAXP (Maximum Receive Data).
USB.EPINDEX holds the index pointing to the Endpoint for this register.

**Figure 33-33. IDXRXMAXP**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| DATA | | | | | | | |
| R/W-Xh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DATA | | | | | | | |
| R/W-Xh | | | | | | | |

**Table 33-31. IDXRXMAXP Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15-0 | DATA | R/W | Xh | Control / Status register data. |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.30 IDXRXCSRL (Offset = 2016h) [Reset = X0h]

IDXRXCSRL is shown in Figure 33-34 and described in Table 33-32.

Return to the Summary Table.

Indexed access to register RXCSRL (Receive Control and Status Low).
USB.EPINDEX holds the index pointing to the Endpoint for this register.

**Figure 33-34. IDXRXCSRL**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| DATA | | | | | | | |
| R/W-Xh | | | | | | | |

**Table 33-32. IDXRXCSRL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7-0 | DATA | R/W | Xh | Control / Status register data. |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.31 IDXRXCSRH (Offset = 2017h) [Reset = X0h]

IDXRXCSRH is shown in Figure 33-35 and described in Table 33-33.

Return to the Summary Table.

Indexed access to register RXCSRH (Receive Control and Status Low).
USB.EPINDEX holds the index pointing to the Endpoint for this register.

**Figure 33-35. IDXRXCSRH**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| DATA | | | | | | | |
| R/W-Xh | | | | | | | |

**Table 33-33. IDXRXCSRH Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7-0 | DATA | R/W | Xh | Control / Status register data. |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.32 IDXRXCOUNT (Offset = 2018h) [Reset = X000h]

IDXRXCOUNT is shown in Figure 33-36 and described in Table 33-34.

Return to the Summary Table.

Indexed access to register RXCOUNT (Receive Byte Count).
USB.EPINDEX holds the index pointing to the Endpoint for this register.

**Figure 33-36. IDXRXCOUNT**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| DATA | | | | | | | |
| R/W-Xh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DATA | | | | | | | |
| R/W-Xh | | | | | | | |

**Table 33-34. IDXRXCOUNT Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15-0 | DATA | R/W | Xh | Control / Status register data. |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.33 IDXTXTYPE (Offset = 201Ah) [Reset = X0h]

IDXTXTYPE is shown in Figure 33-37 and described in Table 33-35.

Return to the Summary Table.

Indexed access to register TXTYPE (Host Transmit Configure Type).
USB.EPINDEX holds the index pointing to the Endpoint for this register.

**Figure 33-37. IDXTXTYPE**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| DATA | | | | | | | |
| R/W-Xh | | | | | | | |

**Table 33-35. IDXTXTYPE Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7-0 | DATA | R/W | Xh | Control / Status register data. |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.34 IDXTXINTERVAL (Offset = 201Bh) [Reset = X0h]

IDXTXINTERVAL is shown in Figure 33-38 and described in Table 33-36.

Return to the Summary Table.

Indexed access to register TXINTERVAL (Host Transmit Interval).
USB.EPINDEX holds the index pointing to the Endpoint for this register.

**Figure 33-38. IDXTXINTERVAL**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| DATA | | | | | | | |
| R/W-Xh | | | | | | | |

**Table 33-36. IDXTXINTERVAL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7-0 | DATA | R/W | Xh | Control / Status register data. |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.35 IDXRXTYPE (Offset = 201Ch) [Reset = X0h]

IDXRXTYPE is shown in Figure 33-39 and described in Table 33-37.

Return to the Summary Table.

Indexed access to register RXTYPE (Host Configure Receive Type).
USB.EPINDEX holds the index pointing to the Endpoint for this register.

**Figure 33-39. IDXRXTYPE**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| DATA | | | | | | | |
| R/W-Xh | | | | | | | |

**Table 33-37. IDXRXTYPE Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7-0 | DATA | R/W | Xh | Control / Status register data. |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.36 IDXRXINTERVAL (Offset = 201Dh) [Reset = X0h]

IDXRXINTERVAL is shown in Figure 33-40 and described in Table 33-38.

Return to the Summary Table.

Indexed access to register RXINTERVAL (Host Receive Polling Interval).
USB.EPINDEX holds the index pointing to the Endpoint for this register.

**Figure 33-40. IDXRXINTERVAL**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| DATA | | | | | | | |
| R/W-Xh | | | | | | | |

**Table 33-38. IDXRXINTERVAL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7-0 | DATA | R/W | Xh | Control / Status register data. |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.37 IDXFIFOSZ (Offset = 201Fh) [Reset = X0h]

IDXFIFOSZ is shown in Figure 33-41 and described in Table 33-39.

Return to the Summary Table.

Indexed access to register FIFOSZ (FIFO Size).
USB.EPINDEX holds the index pointing to the Endpoint for this register.

**Figure 33-41. IDXFIFOSZ**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| DATA | | | | | | | |
| R/W-Xh | | | | | | | |

**Table 33-39. IDXFIFOSZ Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7-0 | DATA | R/W | Xh | Control / Status register data. |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.38 FIFO[y] (Offset = 2020h + formula) [Reset = 00000000h]

FIFO[y] is shown in Figure 33-42 and described in Table 33-40.

Return to the Summary Table.

FIFO for Endpoint [y].
This address range provides 16 addresses for CPU access to the FIFOs for each endpoint. Writing to these addresses loads data into the TxFIFO for the corresponding endpoint. Reading from these addresses unloads data from the RxFIFO for the corresponding endpoint.
Note: Transfers to and from FIFOs may be 8-bit, 16-bit or 32-bit as required, and any combination of access is allowed provided the data accessed is contiguous. However, all the transfers associated with one packet must be of the same width so that the data is consistently byte-, word- or double-word-aligned. The last transfer may however contain fewer bytes than the previous transfers in order to complete an odd-byte or odd-word transfer.
Note: Depending on the size of the FIFO and the expected maximum packet size, the FIFOs support either single-packet or double-packet buffering. However, burst writing of multiple packets is not supported as flags need to be set after each packet is written.
Note: Following a STALL response or a TX Strike Out error on Endpoint 1 – 15, the associated FIFO is completely flushed.

Offset = 2020h + (y * 4h); where y = 0h to Fh

**Figure 33-42. FIFO[y]**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| EPDATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 33-40. FIFO[y] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | EPDATA | R/W | 0h | Endpoint Data.<br>Writing to this register loads the data into the Transmit FIFO and reading unloads data from the Receive FIFO. |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.39 DEVCTL (Offset = 2060h) [Reset = 80h]

DEVCTL is shown in Figure 33-43 and described in Table 33-41.

Return to the Summary Table.

USB Device Control.
DEVCTL is an 8-bit register that is used to select whether the USB controller is operating in Peripheral mode or in Host mode.
Note: If the PHY is suspended no PHY clock is received.

**Figure 33-43. DEVCTL**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| DEV | FSDEV | LSDEV | VUSB | | HOST | HOSTREQ | SESSION |
| R-1h | R-0h | R-0h | R-0h | | R-0h | R/W-0h | R/W-0h |

**Table 33-41. DEVCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7 | DEV | R | 1h | Device Mode.<br>This Read-only bit indicates whether the USB controller is operating as the host mode or device mode.<br>Note: Only valid while a session is in progress.<br>0h (R/W) = The USB controller is operating in host mode.<br>1h (R/W) = The USB controller is operating in device mode. |
| 6 | FSDEV | R | 0h | Full Speed Device Detected.<br>This Read-only bit is set when a full-speed device has been detected being connected to the port.<br>Note: Only valid in Host mode.<br>0h (R/W) = A full-speed Device has not been detected on the port.<br>1h (R/W) = A full-speed Device has been detected on the port. |
| 5 | LSDEV | R | 0h | Low Speed Device Detected.<br>This Read-only bit is set when a low-speed device has been detected being connected to the port.<br>Note: Only valid in Host mode.<br>0h (R/W) = A low-speed Device has not been detected on the port.<br>1h (R/W) = A low-speed Device has been detected on the port. |
| 4-3 | VUSB | R | 0h | VUSB / USB-PHY is powered.<br>0h (R/W) = VUSB / USB-PHY supply is detected as under 1.35V.<br>3h (R/W) = VUSB / USB-PHY supply is detected as above 1.35V. |
| 2 | HOST | R | 0h | Host Mode.<br>This Read-only bit is set when the USB controller is acting as a Host.<br>Note: Only valid while a session is in progress.<br>0h (R/W) = The USB controller is acting as a Device.<br>1h (R/W) = The USB controller is acting as a Host. |
| 1 | HOSTREQ | R/W | 0h | When set, the USB controller will initiate the Host Negotiation when Suspend mode is entered. It is cleared when Host Negotiation is completed.<br>0h (R/W) = No effect<br>1h (R/W) = Initiates the Host Negotiation when SUSPENDmode is entered. |



Copyright © 2026 Texas Instruments Incorporated

**Table 33-41. DEVCTL Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 0 | SESSION | R/W | 0h | Session Start/End.<br>Host mode: This bit is set or cleared by the CPU to start or end a session.<br>Device mode: This bit is set/cleared by the USB controller when a session starts/ends. It is also set by the CPU to initiate the Session Request Protocol. When the USB controller is in Suspend mode, the bit may be cleared by the CPU to perform a software disconnect.<br>Note: Clearing this bit when the core is not suspended will result in undefined behavior.<br>0h (R/W) = When operating as a Host:<br>When cleared by software, this bit ends a session.<br>When operating as a Device:<br>The USB controller has ended a session. When the USB controller is in SUSPEND mode, this bit may be cleared by software to perform a software disconnect.<br>1h (R/W) = When operating as a Host:<br>When set by software, this bit starts a session.<br>When operating as a Device:<br>The USB controller has started a session. When set by software, the Session Request Protocol is initiated.<br>Clearing this bit when the USB controller is not suspended results in undefined behavior. |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.40 MISC (Offset = 2061h) [Reset = 00h]

MISC is shown in Figure 33-44 and described in Table 33-42.

Return to the Summary Table.

USB Miscellaneous registers
The MISC Register is an 8-bit register that contain various common configuration bits. These bits include the Rx/TX Early DMA enable bits.

**Figure 33-44. MISC**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | TX_EDMA | RX_EDMA |
| R-0h | | | | | | R/W-0h | R/W-0h |

**Table 33-42. MISC Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7-2 | RESERVED | R | 0h | |
| 1 | TX_EDMA | R/W | 0h | Transmit early DMA mode.<br>0h (R/W) = Late mode: DMA_REQ signal for all IN Endpoints will be de-asserted when MAXP bytes have been written to an endpoint. This is late mode.<br>1h (R/W) = Early mode: DMA_REQ signal for all IN Endpoints will be de-asserted when MAXP-8 bytes have been written to an endpoint. This is early mode. |
| 0 | RX_EDMA | R/W | 0h | Receive early DMA mode.<br>0h (R/W) = Late mode: DMA_REQ signal for all OUT Endpoints will be de-asserted when MAXP bytes have been read to an endpoint.<br>1h (R/W) = Early Mode: DMA_REQ signal for all OUT Endpoints will be de-asserted when MAXP-8 bytes have been read to an endpoint. |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.41 IDXTXFIFOSZ (Offset = 2062h) [Reset = 00h]

IDXTXFIFOSZ is shown in Figure 33-45 and described in Table 33-43.

Return to the Summary Table.

Indexed Transmit Dynamic FIFO Sizing.
IDXTXFIFOSZ is a 5-bit register which controls the size of the selected TX endpoint FIFO. There is one set of register per End Point Excluding End Point 0.
Note: These are indexed registers therefore to access them the EPINDEX register must be set to the appropriate end point.

**Figure 33-45. IDXTXFIFOSZ**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | DPB | SIZE | | | |
| R-0h | | | R/W-0h | R/W-0h | | | |

**Table 33-43. IDXTXFIFOSZ Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7-5 | RESERVED | R | 0h | |
| 4 | DPB | R/W | 0h | Double Packet Buffer Support<br>0h (R/W) = Single packet buffering is supported.<br>1h (R/W) = Double packet buffering is enabled. |
| 3-0 | SIZE | R/W | 0h | Max Packet Size<br>0h (R/W) = FIFO size 8-byte<br>1h (R/W) = FIFO size 16-byte<br>2h (R/W) = FIFO size 32-byte<br>3h (R/W) = FIFO size 64-byte<br>4h (R/W) = FIFO size 128-byte<br>5h (R/W) = FIFO size 256-byte<br>6h (R/W) = FIFO size 512-byte<br>7h (R/W) = FIFO size 1024-byte<br>8h (R/W) = FIFO size 2048-byte |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.42 IDXRXFIFOSZ (Offset = 2063h) [Reset = 00h]

IDXRXFIFOSZ is shown in Figure 33-46 and described in Table 33-44.

Return to the Summary Table.

Indexed Receive Dynamic FIFO Sizing.
IDXRXFIFOSZ is a 5-bit register which controls the size of the selected Rx endpoint FIFO. There is one set of register per End Point Excluding End Point 0.
Note: These are indexed registers therefore to access them the EPINDEX register must be set to the appropriate end point.

**Figure 33-46. IDXRXFIFOSZ**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | DPB | SIZE | | | |
| R-0h | | | R/W-0h | R/W-0h | | | |

**Table 33-44. IDXRXFIFOSZ Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7-5 | RESERVED | R | 0h | |
| 4 | DPB | R/W | 0h | Double Packet Buffer Support<br>0h (R/W) = Single packet buffering is supported.<br>1h (R/W) = Double packet buffering is enabled. |
| 3-0 | SIZE | R/W | 0h | Maximum packet size to be allowed. If DPB = 0, the FIFO also is this size; if DPB = 1, the FIFO is twice this size. Packet size in bytes:<br>0h (R/W) = FIFO size 8-byte<br>1h (R/W) = FIFO size 16-byte<br>2h (R/W) = FIFO size 32-byte<br>3h (R/W) = FIFO size 64-byte<br>4h (R/W) = FIFO size 128-byte<br>5h (R/W) = FIFO size 256-byte<br>6h (R/W) = FIFO size 512-byte<br>7h (R/W) = FIFO size 1024-byte<br>8h (R/W) = FIFO size 2048-byte |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.43 IDXTXFIFOADD (Offset = 2064h) [Reset = 0000h]

IDXTXFIFOADD is shown in Figure 33-47 and described in Table 33-45.

Return to the Summary Table.

Indexed Transmit FIFO Start Address.
IDXTXFIFOADD is a 14-bit register which controls the start address of the selected Tx endpoint FIFO. There is one set of register per End Point Excluding End Point 0.
Note: These are indexed registers therefore to access them the EPINDEX register must be set to the appropriate end point.

**Figure 33-47. IDXTXFIFOADD**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| RESERVED | | ADDR | | | | | |
| R-0h | | R/W-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ADDR | | | | | | | |
| R/W-0h | | | | | | | |

**Table 33-45. IDXTXFIFOADD Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15-14 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

**Table 33-45. IDXTXFIFOADD Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 13-0 | ADDR | R/W | 0h | Endpoint Data<br>0h (R/W) = 0.0<br>1h (R/W) = 8.0<br>2h (R/W) = 16.0<br>3h (R/W) = 24.0<br>4h (R/W) = 32.0<br>5h (R/W) = 40.0<br>6h (R/W) = 48.0<br>7h (R/W) = 56.0<br>8h (R/W) = 64.0<br>9h (R/W) = 72.0<br>Ah (R/W) = 80.0<br>Bh (R/W) = 88.0<br>Ch (R/W) = 96.0<br>Dh (R/W) = 104.0<br>Eh (R/W) = 112.0<br>Fh (R/W) = 120.0<br>10h (R/W) = 128.0<br>11h (R/W) = 136.0<br>12h (R/W) = 144.0<br>13h (R/W) = 152.0<br>14h (R/W) = 160.0<br>15h (R/W) = 168.0<br>16h (R/W) = 176.0<br>17h (R/W) = 184.0<br>18h (R/W) = 192.0<br>19h (R/W) = 200.0<br>1Ah (R/W) = 208.0<br>1Bh (R/W) = 216.0<br>1Ch (R/W) = 224.0<br>1Dh (R/W) = 232.0<br>1Eh (R/W) = 240.0<br>1Fh (R/W) = 248.0<br>20h (R/W) = 256.0<br>21h (R/W) = 264.0<br>22h (R/W) = 272.0<br>23h (R/W) = 280.0<br>24h (R/W) = 288.0<br>25h (R/W) = 296.0<br>26h (R/W) = 304.0<br>27h (R/W) = 312.0<br>28h (R/W) = 320.0<br>29h (R/W) = 328.0<br>2Ah (R/W) = 336.0<br>2Bh (R/W) = 344.0<br>2Ch (R/W) = 352.0<br>2Dh (R/W) = 360.0<br>2Eh (R/W) = 368.0<br>2Fh (R/W) = 376.0<br>30h (R/W) = 384.0<br>31h (R/W) = 392.0<br>32h (R/W) = 400.0<br>33h (R/W) = 408.0<br>34h (R/W) = 416.0<br>35h (R/W) = 424.0<br>36h (R/W) = 432.0<br>37h (R/W) = 440.0<br>38h (R/W) = 448.0<br>39h (R/W) = 456.0<br>3Ah (R/W) = 464.0<br>3Bh (R/W) = 472.0<br>3Ch (R/W) = 480.0<br>3Dh (R/W) = 488.0<br>3Eh (R/W) = 496.0<br>3Fh (R/W) = 504.0 |



Copyright © 2026 Texas Instruments Incorporated

**Table 33-45. IDXTXFIFOADD Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| | | | | 40h (R/W) = 512.0<br>41h (R/W) = 520.0<br>42h (R/W) = 528.0<br>43h (R/W) = 536.0<br>44h (R/W) = 544.0<br>45h (R/W) = 552.0<br>46h (R/W) = 560.0<br>47h (R/W) = 568.0<br>48h (R/W) = 576.0<br>49h (R/W) = 584.0<br>4Ah (R/W) = 592.0<br>4Bh (R/W) = 600.0<br>4Ch (R/W) = 608.0<br>4Dh (R/W) = 616.0<br>4Eh (R/W) = 624.0<br>4Fh (R/W) = 632.0<br>50h (R/W) = 640.0<br>51h (R/W) = 648.0<br>52h (R/W) = 656.0<br>53h (R/W) = 664.0<br>54h (R/W) = 672.0<br>55h (R/W) = 680.0<br>56h (R/W) = 688.0<br>57h (R/W) = 696.0<br>58h (R/W) = 704.0<br>59h (R/W) = 712.0<br>5Ah (R/W) = 720.0<br>5Bh (R/W) = 728.0<br>5Ch (R/W) = 736.0<br>5Dh (R/W) = 744.0<br>5Eh (R/W) = 752.0<br>5Fh (R/W) = 760.0<br>60h (R/W) = 768.0<br>61h (R/W) = 776.0<br>62h (R/W) = 784.0<br>63h (R/W) = 792.0<br>64h (R/W) = 800.0<br>65h (R/W) = 808.0<br>66h (R/W) = 816.0<br>67h (R/W) = 824.0<br>68h (R/W) = 832.0<br>69h (R/W) = 840.0<br>6Ah (R/W) = 848.0<br>6Bh (R/W) = 856.0<br>6Ch (R/W) = 864.0<br>6Dh (R/W) = 872.0<br>6Eh (R/W) = 880.0<br>6Fh (R/W) = 888.0<br>70h (R/W) = 896.0<br>71h (R/W) = 904.0<br>72h (R/W) = 912.0<br>73h (R/W) = 920.0<br>74h (R/W) = 928.0<br>75h (R/W) = 936.0<br>76h (R/W) = 944.0<br>77h (R/W) = 952.0<br>78h (R/W) = 960.0<br>79h (R/W) = 968.0<br>7Ah (R/W) = 976.0<br>7Bh (R/W) = 984.0<br>7Ch (R/W) = 992.0<br>7Dh (R/W) = 1000.0<br>7Eh (R/W) = 1008.0<br>7Fh (R/W) = 1016.0<br>80h (R/W) = 1024.0 |



Copyright © 2026 Texas Instruments Incorporated

**Table 33-45. IDXTXFIFOADD Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| | | | | 81h (R/W) = 1032.0<br>82h (R/W) = 1040.0<br>83h (R/W) = 1048.0<br>84h (R/W) = 1056.0<br>85h (R/W) = 1064.0<br>86h (R/W) = 1072.0<br>87h (R/W) = 1080.0<br>88h (R/W) = 1088.0<br>89h (R/W) = 1096.0<br>8Ah (R/W) = 1104.0<br>8Bh (R/W) = 1112.0<br>8Ch (R/W) = 1120.0<br>8Dh (R/W) = 1128.0<br>8Eh (R/W) = 1136.0<br>8Fh (R/W) = 1144.0<br>90h (R/W) = 1152.0<br>91h (R/W) = 1160.0<br>92h (R/W) = 1168.0<br>93h (R/W) = 1176.0<br>94h (R/W) = 1184.0<br>95h (R/W) = 1192.0<br>96h (R/W) = 1200.0<br>97h (R/W) = 1208.0<br>98h (R/W) = 1216.0<br>99h (R/W) = 1224.0<br>9Ah (R/W) = 1232.0<br>9Bh (R/W) = 1240.0<br>9Ch (R/W) = 1248.0<br>9Dh (R/W) = 1256.0<br>9Eh (R/W) = 1264.0<br>9Fh (R/W) = 1272.0<br>A0h (R/W) = 1280.0<br>A1h (R/W) = 1288.0<br>A2h (R/W) = 1296.0<br>A3h (R/W) = 1304.0<br>A4h (R/W) = 1312.0<br>A5h (R/W) = 1320.0<br>A6h (R/W) = 1328.0<br>A7h (R/W) = 1336.0<br>A8h (R/W) = 1344.0<br>A9h (R/W) = 1352.0<br>AAh (R/W) = 1360.0<br>ABh (R/W) = 1368.0<br>ACh (R/W) = 1376.0<br>ADh (R/W) = 1384.0<br>AEh (R/W) = 1392.0<br>AFh (R/W) = 1400.0<br>B0h (R/W) = 1408.0<br>B1h (R/W) = 1416.0<br>B2h (R/W) = 1424.0<br>B3h (R/W) = 1432.0<br>B4h (R/W) = 1440.0<br>B5h (R/W) = 1448.0<br>B6h (R/W) = 1456.0<br>B7h (R/W) = 1464.0<br>B8h (R/W) = 1472.0<br>B9h (R/W) = 1480.0<br>BAh (R/W) = 1488.0<br>BBh (R/W) = 1496.0<br>BCh (R/W) = 1504.0<br>BDh (R/W) = 1512.0<br>BEh (R/W) = 1520.0<br>BFh (R/W) = 1528.0<br>C0h (R/W) = 1536.0<br>C1h (R/W) = 1544.0 |



Copyright © 2026 Texas Instruments Incorporated

**Table 33-45. IDXTXFIFOADD Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| | | | | C2h (R/W) = 1552.0<br>C3h (R/W) = 1560.0<br>C4h (R/W) = 1568.0<br>C5h (R/W) = 1576.0<br>C6h (R/W) = 1584.0<br>C7h (R/W) = 1592.0<br>C8h (R/W) = 1600.0<br>C9h (R/W) = 1608.0<br>CAh (R/W) = 1616.0<br>CBh (R/W) = 1624.0<br>CCh (R/W) = 1632.0<br>CDh (R/W) = 1640.0<br>CEh (R/W) = 1648.0<br>CFh (R/W) = 1656.0<br>D0h (R/W) = 1664.0<br>D1h (R/W) = 1672.0<br>D2h (R/W) = 1680.0<br>D3h (R/W) = 1688.0<br>D4h (R/W) = 1696.0<br>D5h (R/W) = 1704.0<br>D6h (R/W) = 1712.0<br>D7h (R/W) = 1720.0<br>D8h (R/W) = 1728.0<br>D9h (R/W) = 1736.0<br>DAh (R/W) = 1744.0<br>DBh (R/W) = 1752.0<br>DCh (R/W) = 1760.0<br>DDh (R/W) = 1768.0<br>DEh (R/W) = 1776.0<br>DFh (R/W) = 1784.0<br>E0h (R/W) = 1792.0<br>E1h (R/W) = 1800.0<br>E2h (R/W) = 1808.0<br>E3h (R/W) = 1816.0<br>E4h (R/W) = 1824.0<br>E5h (R/W) = 1832.0<br>E6h (R/W) = 1840.0<br>E7h (R/W) = 1848.0<br>E8h (R/W) = 1856.0<br>E9h (R/W) = 1864.0<br>EAh (R/W) = 1872.0<br>EBh (R/W) = 1880.0<br>ECh (R/W) = 1888.0<br>EDh (R/W) = 1896.0<br>EEh (R/W) = 1904.0<br>EFh (R/W) = 1912.0<br>F0h (R/W) = 1920.0<br>F1h (R/W) = 1928.0<br>F2h (R/W) = 1936.0<br>F3h (R/W) = 1944.0<br>F4h (R/W) = 1952.0<br>F5h (R/W) = 1960.0<br>F6h (R/W) = 1968.0<br>F7h (R/W) = 1976.0<br>F8h (R/W) = 1984.0<br>F9h (R/W) = 1992.0<br>FAh (R/W) = 2000.0<br>FBh (R/W) = 2008.0<br>FCh (R/W) = 2016.0<br>FDh (R/W) = 2024.0<br>FEh (R/W) = 2032.0<br>FFh (R/W) = 2040.0 |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.44 IDXRXFIFOADD (Offset = 2066h) [Reset = 0000h]

IDXRXFIFOADD is shown in Figure 33-48 and described in Table 33-46.

Return to the Summary Table.

Indexed Receive FIFO Start Address.
IDXRXFIFOADD is a 14-bit register which controls the start address of the selected Rx endpoint FIFO. There is one set of register per End Point Excluding End Point 0.
Note: These are indexed registers therefore to access them the EPINDEX register must be set to the appropriate end point.

**Figure 33-48. IDXRXFIFOADD**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| RESERVED | | ADDR | | | | | |
| R-0h | | R/W-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ADDR | | | | | | | |
| R/W-0h | | | | | | | |

**Table 33-46. IDXRXFIFOADD Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15-14 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

**Table 33-46. IDXRXFIFOADD Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 13-0 | ADDR | R/W | 0h | Endpoint Data<br>0h (R/W) = 0.0<br>1h (R/W) = 8.0<br>2h (R/W) = 16.0<br>3h (R/W) = 24.0<br>4h (R/W) = 32.0<br>5h (R/W) = 40.0<br>6h (R/W) = 48.0<br>7h (R/W) = 56.0<br>8h (R/W) = 64.0<br>9h (R/W) = 72.0<br>Ah (R/W) = 80.0<br>Bh (R/W) = 88.0<br>Ch (R/W) = 96.0<br>Dh (R/W) = 104.0<br>Eh (R/W) = 112.0<br>Fh (R/W) = 120.0<br>10h (R/W) = 128.0<br>11h (R/W) = 136.0<br>12h (R/W) = 144.0<br>13h (R/W) = 152.0<br>14h (R/W) = 160.0<br>15h (R/W) = 168.0<br>16h (R/W) = 176.0<br>17h (R/W) = 184.0<br>18h (R/W) = 192.0<br>19h (R/W) = 200.0<br>1Ah (R/W) = 208.0<br>1Bh (R/W) = 216.0<br>1Ch (R/W) = 224.0<br>1Dh (R/W) = 232.0<br>1Eh (R/W) = 240.0<br>1Fh (R/W) = 248.0<br>20h (R/W) = 256.0<br>21h (R/W) = 264.0<br>22h (R/W) = 272.0<br>23h (R/W) = 280.0<br>24h (R/W) = 288.0<br>25h (R/W) = 296.0<br>26h (R/W) = 304.0<br>27h (R/W) = 312.0<br>28h (R/W) = 320.0<br>29h (R/W) = 328.0<br>2Ah (R/W) = 336.0<br>2Bh (R/W) = 344.0<br>2Ch (R/W) = 352.0<br>2Dh (R/W) = 360.0<br>2Eh (R/W) = 368.0<br>2Fh (R/W) = 376.0<br>30h (R/W) = 384.0<br>31h (R/W) = 392.0<br>32h (R/W) = 400.0<br>33h (R/W) = 408.0<br>34h (R/W) = 416.0<br>35h (R/W) = 424.0<br>36h (R/W) = 432.0<br>37h (R/W) = 440.0<br>38h (R/W) = 448.0<br>39h (R/W) = 456.0<br>3Ah (R/W) = 464.0<br>3Bh (R/W) = 472.0<br>3Ch (R/W) = 480.0<br>3Dh (R/W) = 488.0<br>3Eh (R/W) = 496.0<br>3Fh (R/W) = 504.0 |



Copyright © 2026 Texas Instruments Incorporated

**Table 33-46. IDXRXFIFOADD Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| | | | | 40h (R/W) = 512.0<br>41h (R/W) = 520.0<br>42h (R/W) = 528.0<br>43h (R/W) = 536.0<br>44h (R/W) = 544.0<br>45h (R/W) = 552.0<br>46h (R/W) = 560.0<br>47h (R/W) = 568.0<br>48h (R/W) = 576.0<br>49h (R/W) = 584.0<br>4Ah (R/W) = 592.0<br>4Bh (R/W) = 600.0<br>4Ch (R/W) = 608.0<br>4Dh (R/W) = 616.0<br>4Eh (R/W) = 624.0<br>4Fh (R/W) = 632.0<br>50h (R/W) = 640.0<br>51h (R/W) = 648.0<br>52h (R/W) = 656.0<br>53h (R/W) = 664.0<br>54h (R/W) = 672.0<br>55h (R/W) = 680.0<br>56h (R/W) = 688.0<br>57h (R/W) = 696.0<br>58h (R/W) = 704.0<br>59h (R/W) = 712.0<br>5Ah (R/W) = 720.0<br>5Bh (R/W) = 728.0<br>5Ch (R/W) = 736.0<br>5Dh (R/W) = 744.0<br>5Eh (R/W) = 752.0<br>5Fh (R/W) = 760.0<br>60h (R/W) = 768.0<br>61h (R/W) = 776.0<br>62h (R/W) = 784.0<br>63h (R/W) = 792.0<br>64h (R/W) = 800.0<br>65h (R/W) = 808.0<br>66h (R/W) = 816.0<br>67h (R/W) = 824.0<br>68h (R/W) = 832.0<br>69h (R/W) = 840.0<br>6Ah (R/W) = 848.0<br>6Bh (R/W) = 856.0<br>6Ch (R/W) = 864.0<br>6Dh (R/W) = 872.0<br>6Eh (R/W) = 880.0<br>6Fh (R/W) = 888.0<br>70h (R/W) = 896.0<br>71h (R/W) = 904.0<br>72h (R/W) = 912.0<br>73h (R/W) = 920.0<br>74h (R/W) = 928.0<br>75h (R/W) = 936.0<br>76h (R/W) = 944.0<br>77h (R/W) = 952.0<br>78h (R/W) = 960.0<br>79h (R/W) = 968.0<br>7Ah (R/W) = 976.0<br>7Bh (R/W) = 984.0<br>7Ch (R/W) = 992.0<br>7Dh (R/W) = 1000.0<br>7Eh (R/W) = 1008.0<br>7Fh (R/W) = 1016.0<br>80h (R/W) = 1024.0 |



Copyright © 2026 Texas Instruments Incorporated

**Table 33-46. IDXRXFIFOADD Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| | | | | 81h (R/W) = 1032.0<br>82h (R/W) = 1040.0<br>83h (R/W) = 1048.0<br>84h (R/W) = 1056.0<br>85h (R/W) = 1064.0<br>86h (R/W) = 1072.0<br>87h (R/W) = 1080.0<br>88h (R/W) = 1088.0<br>89h (R/W) = 1096.0<br>8Ah (R/W) = 1104.0<br>8Bh (R/W) = 1112.0<br>8Ch (R/W) = 1120.0<br>8Dh (R/W) = 1128.0<br>8Eh (R/W) = 1136.0<br>8Fh (R/W) = 1144.0<br>90h (R/W) = 1152.0<br>91h (R/W) = 1160.0<br>92h (R/W) = 1168.0<br>93h (R/W) = 1176.0<br>94h (R/W) = 1184.0<br>95h (R/W) = 1192.0<br>96h (R/W) = 1200.0<br>97h (R/W) = 1208.0<br>98h (R/W) = 1216.0<br>99h (R/W) = 1224.0<br>9Ah (R/W) = 1232.0<br>9Bh (R/W) = 1240.0<br>9Ch (R/W) = 1248.0<br>9Dh (R/W) = 1256.0<br>9Eh (R/W) = 1264.0<br>9Fh (R/W) = 1272.0<br>A0h (R/W) = 1280.0<br>A1h (R/W) = 1288.0<br>A2h (R/W) = 1296.0<br>A3h (R/W) = 1304.0<br>A4h (R/W) = 1312.0<br>A5h (R/W) = 1320.0<br>A6h (R/W) = 1328.0<br>A7h (R/W) = 1336.0<br>A8h (R/W) = 1344.0<br>A9h (R/W) = 1352.0<br>AAh (R/W) = 1360.0<br>ABh (R/W) = 1368.0<br>ACh (R/W) = 1376.0<br>ADh (R/W) = 1384.0<br>AEh (R/W) = 1392.0<br>AFh (R/W) = 1400.0<br>B0h (R/W) = 1408.0<br>B1h (R/W) = 1416.0<br>B2h (R/W) = 1424.0<br>B3h (R/W) = 1432.0<br>B4h (R/W) = 1440.0<br>B5h (R/W) = 1448.0<br>B6h (R/W) = 1456.0<br>B7h (R/W) = 1464.0<br>B8h (R/W) = 1472.0<br>B9h (R/W) = 1480.0<br>BAh (R/W) = 1488.0<br>BBh (R/W) = 1496.0<br>BCh (R/W) = 1504.0<br>BDh (R/W) = 1512.0<br>BEh (R/W) = 1520.0<br>BFh (R/W) = 1528.0<br>C0h (R/W) = 1536.0<br>C1h (R/W) = 1544.0 |



Copyright © 2026 Texas Instruments Incorporated

**Table 33-46. IDXRXFIFOADD Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| | | | | C2h (R/W) = 1552.0<br>C3h (R/W) = 1560.0<br>C4h (R/W) = 1568.0<br>C5h (R/W) = 1576.0<br>C6h (R/W) = 1584.0<br>C7h (R/W) = 1592.0<br>C8h (R/W) = 1600.0<br>C9h (R/W) = 1608.0<br>CAh (R/W) = 1616.0<br>CBh (R/W) = 1624.0<br>CCh (R/W) = 1632.0<br>CDh (R/W) = 1640.0<br>CEh (R/W) = 1648.0<br>CFh (R/W) = 1656.0<br>D0h (R/W) = 1664.0<br>D1h (R/W) = 1672.0<br>D2h (R/W) = 1680.0<br>D3h (R/W) = 1688.0<br>D4h (R/W) = 1696.0<br>D5h (R/W) = 1704.0<br>D6h (R/W) = 1712.0<br>D7h (R/W) = 1720.0<br>D8h (R/W) = 1728.0<br>D9h (R/W) = 1736.0<br>DAh (R/W) = 1744.0<br>DBh (R/W) = 1752.0<br>DCh (R/W) = 1760.0<br>DDh (R/W) = 1768.0<br>DEh (R/W) = 1776.0<br>DFh (R/W) = 1784.0<br>E0h (R/W) = 1792.0<br>E1h (R/W) = 1800.0<br>E2h (R/W) = 1808.0<br>E3h (R/W) = 1816.0<br>E4h (R/W) = 1824.0<br>E5h (R/W) = 1832.0<br>E6h (R/W) = 1840.0<br>E7h (R/W) = 1848.0<br>E8h (R/W) = 1856.0<br>E9h (R/W) = 1864.0<br>EAh (R/W) = 1872.0<br>EBh (R/W) = 1880.0<br>ECh (R/W) = 1888.0<br>EDh (R/W) = 1896.0<br>EEh (R/W) = 1904.0<br>EFh (R/W) = 1912.0<br>F0h (R/W) = 1920.0<br>F1h (R/W) = 1928.0<br>F2h (R/W) = 1936.0<br>F3h (R/W) = 1944.0<br>F4h (R/W) = 1952.0<br>F5h (R/W) = 1960.0<br>F6h (R/W) = 1968.0<br>F7h (R/W) = 1976.0<br>F8h (R/W) = 1984.0<br>F9h (R/W) = 1992.0<br>FAh (R/W) = 2000.0<br>FBh (R/W) = 2008.0<br>FCh (R/W) = 2016.0<br>FDh (R/W) = 2024.0<br>FEh (R/W) = 2032.0<br>FFh (R/W) = 2040.0 |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.45 EPINFO (Offset = 2078h) [Reset = 88h]

EPINFO is shown in Figure 33-49 and described in Table 33-47.

Return to the Summary Table.

Endpoint Information.
This 8-bit read-only register allows read-back of the number of TX and Rx endpoints included in the design.

**Figure 33-49. EPINFO**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RXENDPOINTS | | | | TXENDPOINTS | | | |
| R-8h | | | | R-8h | | | |

**Table 33-47. EPINFO Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7-4 | RXENDPOINTS | R | 8h | The number of Rx endpoints implemented in the design. |
| 3-0 | TXENDPOINTS | R | 8h | The number of TX endpoints implemented in the design. |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.46 RAMINFO (Offset = 2079h) [Reset = 5Ch]

RAMINFO is shown in Figure 33-50 and described in Table 33-48.

Return to the Summary Table.

RAM Buffer Information.
This 8-bit read-only register provides information about the width of the RAM.

**Figure 33-50. RAMINFO**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | RAMBITS | | | |
| R-0h | | | | R-Ch | | | |

**Table 33-48. RAMINFO Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7-4 | RESERVED | R | 0h | |
| 3-0 | RAMBITS | R | Ch | The width of the RAM address bus.<br>8h = USB FIFO RAM size is 1kB<br>9h = USB FIFO RAM size is 2kB<br>Ah = USB FIFO RAM size is 4kB |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.47 CONTIM (Offset = 207Ah) [Reset = 5Ch]

CONTIM is shown in Figure 33-51 and described in Table 33-49.

Return to the Summary Table.

Connect Timing.
This 8-bit register allows some delays to be specified.

**Figure 33-51. CONTIM**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| WTCON | | | | WTID | | | |
| R/W-5h | | | | R/W-Ch | | | |

**Table 33-49. CONTIM Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7-4 | WTCON | R/W | 5h | The connect wait field configures the wait required to allow for the user's connect/disconnect filter, in units of 533.3 ns.<br>Note: The default corresponds to 2.667 us. |
| 3-0 | WTID | R/W | Ch | The wait ID field configures the delay required from the enable of the ID detection to when the ID value is valid, in units of 4.369 ms.<br>Note: The default corresponds to 52.43 ms. |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.48 FSEOF (Offset = 207Dh) [Reset = 77h]

FSEOF is shown in Figure 33-52 and described in Table 33-50.

Return to the Summary Table.

USB-Full-Speed Last Transaction to End of Frame Timing.
This 8-bit register sets the minimum time gap that is to be allowed between the start of the last transaction and the EOF for Full-speed transactions.

**Figure 33-52. FSEOF**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| FSEOFG | | | | | | | |
| R/W-77h | | | | | | | |

**Table 33-50. FSEOF Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7-0 | FSEOFG | R/W | 77h | The full-speed end-of-frame gap field is used during full-speed transactions to configure the gap between the last transaction and the End-of-Frame (EOF), in units of 533.3 ns.<br>Note: The default corresponds to 63.46 us. |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.49 LSEOF (Offset = 207Eh) [Reset = 72h]

LSEOF is shown in Figure 33-53 and described in Table 33-51.

Return to the Summary Table.

USB-Low-Speed Last Transaction to End of Frame Timing.
This 8-bit register sets the minimum time gap that is to be allowed between the start of the last transaction and the EOF for Low-speed transactions.

**Figure 33-53. LSEOF**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| LSEOFG | | | | | | | |
| R-72h | | | | | | | |

**Table 33-51. LSEOF Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7-0 | LSEOFG | R | 72h | The low-speed end-of-frame gap field is used during low-speed transactions to set the gap between the last transaction and the End-of-Frame (EOF), in units of 1.067 us.<br>Note: The default corresponds to 121.6 us. |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.50 TXFUNCADDR[j] (Offset = 2080h + formula) [Reset = 00h]

TXFUNCADDR[j] is shown in Figure 33-54 and described in Table 33-52.

Return to the Summary Table.

Transmit Functional Address Endpoint [j]
TXFUNCADDR is a 7-bit read/write registers that record the address of the target function that is to be accessed through the associated endpoint (EPn). TXFUNCADDR needs to be defined for each TX endpoint that is used.
Note: TXFUNCADDR must be defined for Endpoint 0.
Note: Relevant in host mode only

Offset = 2080h + (j * 8h); where j = 0h to Fh

**Figure 33-54. TXFUNCADDR[j]**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RESERVED | ADDR | | | | | | |
| R-0h | R/W-0h | | | | | | |

**Table 33-52. TXFUNCADDR[j] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7 | RESERVED | R | 0h | |
| 6-0 | ADDR | R/W | 0h | Device Address specifies the USB bus address for the target Device. |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.51 TXHUBADDR[j] (Offset = 2082h + formula) [Reset = 00h]

TXHUBADDR[j] is shown in Figure 33-55 and described in Table 33-53.

Return to the Summary Table.

USB Transmit Hub Address Endpoint [j]
TXHUBADDR is an 8-bit read/write registers which, like TXHUBPORT, only need to be written where a full- or low-speed device is connected to TX/RX Endpoint EPn via a high-speed USB 2.0 hub which carries out the necessary transaction translation to convert between high-speed transmission and full-/low-speed transmission. In such circumstances:
- the lower 7 bits should record the address of this USB 2.0 hub
- the top bit should record whether the hub has multiple transaction translators (set to ‘0’ if single transaction translator; set to ‘1’ if multiple transaction translators).
Note: If Endpoint 0 is connected to a hub, then TXHUBADDR must be defined for EP0.
Note: Relevant in host mode only

Offset = 2082h + (j * 8h); where j = 0h to Fh

**Figure 33-55. TXHUBADDR[j]**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| MULTTRAN | ADDR | | | | | | |
| R/W-0h | R/W-0h | | | | | | |

**Table 33-53. TXHUBADDR[j] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7 | MULTTRAN | R/W | 0h | Record whether the hub has multiple transaction translators.<br>0h (R/W) = Clear to indicate that the hub has a single transaction translator.<br>1h (R/W) = Set to indicate that the hub has multiple transaction translators. |
| 6-0 | ADDR | R/W | 0h | Device Address specifies the USB bus address for the target Device. |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.52 TXHUBPORT[j] (Offset = 2083h + formula) [Reset = 00h]

TXHUBPORT[j] is shown in Figure 33-56 and described in Table 33-54.

Return to the Summary Table.

Transmit Hub Port Endpoint [j]
TXHUBPORT only need to be written where a full- or low-speed device is connected to TX/RX Endpoint EPn via a high-speed USB 2.0 hub which carries out the necessary transaction translation. In such circumstances, these 7-bit read/write registers need to be used to record the port of that USB 2.0 hub through which the target associated with the endpoint is accessed.
Note: If Endpoint 0 is connected to a hub, then TXHUBPORT must be defined.
Note: Relevant in host mode only

Offset = 2083h + (j * 8h); where j = 0h to Fh

**Figure 33-56. TXHUBPORT[j]**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RESERVED | ADDR | | | | | | |
| R-0h | R/W-0h | | | | | | |

**Table 33-54. TXHUBPORT[j] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7 | RESERVED | R | 0h | |
| 6-0 | ADDR | R/W | 0h | Hub Port specifies the USB hub port number. |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.53 RXFUNCADDR[j] (Offset = 2084h + formula) [Reset = 00h]

RXFUNCADDR[j] is shown in Figure 33-57 and described in Table 33-55.

Return to the Summary Table.

Receive Functional Address Endpoint [j]
RXFUNCADDR is a 7-bit read/write registers that record the address of the target function that is to be accessed through the associated endpoint (EPn). RXFUNCADDR needs to be defined for each Rx endpoint that is used.
Note: The RXFUNCADDR register does not exist on EP0.
Note: Relevant in host mode only

Offset = 2084h + (j * 8h); where j = 0h to Fh

**Figure 33-57. RXFUNCADDR[j]**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RESERVED | ADDR | | | | | | |
| R-0h | R/W-0h | | | | | | |

**Table 33-55. RXFUNCADDR[j] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7 | RESERVED | R | 0h | |
| 6-0 | ADDR | R/W | 0h | Device Address specifies the USB bus address for the target Device. |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.54 RXHUBADDR[j] (Offset = 2086h + formula) [Reset = 00h]

RXHUBADDR[j] is shown in Figure 33-58 and described in Table 33-56.

Return to the Summary Table.

Receive Hub Address Endpoint [j]
RXHUBADDR is an 8-bit read/write registers which, like RXHUBPORT, only need to be written where a full- or low-speed device is connected to TX/RX Endpoint EPn via a high-speed USB 2.0 hub which carries out the necessary transaction translation to convert between high-speed transmission and full-/low-speed transmission. In such circumstances:
- the lower 7 bits should record the address of this USB 2.0 hub
- the top bit should record whether the hub has multiple transaction translators (set to '0' if single transaction translator; set to '1' if multiple transaction translators).
Note: The RXHUBADDR register does not exist on EP0.
Note: Relevant in host mode only

Offset = 2086h + (j * 8h); where j = 0h to Fh

**Figure 33-58. RXHUBADDR[j]**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| MULTTRAN | ADDR | | | | | | |
| R/W-0h | R/W-0h | | | | | | |

**Table 33-56. RXHUBADDR[j] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7 | MULTTRAN | R/W | 0h | Record whether the hub has multiple transaction translators.<br>0h (R/W) = Clear to indicate that the hub has a single transaction translator.<br>1h (R/W) = Set to indicate that the hub has multiple transaction translators. |
| 6-0 | ADDR | R/W | 0h | Hub Address |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.55 RXHUBPORT[j] (Offset = 2087h + formula) [Reset = 00h]

RXHUBPORT[j] is shown in Figure 33-59 and described in Table 33-57.

Return to the Summary Table.

USB Receive Hub Port Endpoint [j]
RXHUBPORT only need to be written where a full- or low-speed device is connected to TX/RX Endpoint EPn via a high-speed USB 2.0 hub which carries out the necessary transaction translation. In such circumstances, these 7-bit read/write registers need to be used to record the port of that USB 2.0 hub through which the target associated with the endpoint is accessed.
Note: The HUBPORT register does not exist on EP0.
Note: Relevant in host mode only

Offset = 2087h + (j * 8h); where j = 0h to Fh

**Figure 33-59. RXHUBPORT[j]**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RESERVED | ADDR | | | | | | |
| R-0h | R/W-0h | | | | | | |

**Table 33-57. RXHUBPORT[j] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7 | RESERVED | R | 0h | |
| 6-0 | ADDR | R/W | 0h | Hub Address |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.56 CSR0L (Offset = 2102h) [Reset = 00h]

CSR0L is shown in Figure 33-60 and described in Table 33-58.

Return to the Summary Table.

Control and Status Endpoint 0 Low.
CSR0L is an 8-bit register that provides control and status bits for Endpoint 0.
Note: The interpretation of the register depends on whether the USB controller is acting as a peripheral or as a host.
Note: Users should also be aware that the value returned when the register is read reflects the status attained e.g. as a result of writing to the register.

**Figure 33-60. CSR0L**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| SETENDC_NAKTO | RXRDYC_STATUS | STALL_RQPKT | SETEND_ERROR | DATAEND_SETUP | STALLED | TXRDY | RXRDY |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

**Table 33-58. CSR0L Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7 | SETENDC_NAKTO | R/W | 0h | Setup End Clear / NAK Timeout<br>Device mode: The CPU writes a 1 to this bit to clear the SETEND_ERROR bit. It is cleared automatically.<br>Host mode: This bit will be set when Endpoint 0 is halted following the receipt of NAK responses for longer than the time set by the USBNAKLMT register. The CPU should clear this bit to allow the endpoint to continue.<br>0h (R/W) = No effect<br>1h (R/W) = Clear SETEND_ERROR or SETENDC_KAKTO |
| 6 | RXRDYC_STATUS | R/W | 0h | Receive ready clear / Status packet<br>Device mode: Receive ready clear. The CPU writes a 1 to this bit to clear the RXRDY bit. It is cleared automatically<br>Host mode: Status Packet. The CPU sets this bit at the same time as the TXRDY or STALL_RQPKT bit is set, to perform a status stage transaction. Setting this bit ensures that the DT bit is set in the USBCSRH0 register, so that a DATA1 packet is used for the Status Stage transaction.<br>Setting this bit ensures that the DT bit is set in the USBCSRH0 register so that a DATA1 packet is used for the STATUS stage transaction.<br>0h (R/W) = No transaction<br>1h (R/W) = Clear RXRDY or sent status packet. |
| 5 | STALL_RQPKT | R/W | 0h | Send Stall / Request Packet.<br>Device mode: The CPU writes a 1 to this bit to terminate the current transaction. The STALL handshake will be transmitted and then this bit will be cleared automatically.<br>Host mode: The CPU sets this bit to request an IN transaction. It is cleared when RXRDY bit is set.<br>0h (R/W) = No effect<br>1h (R/W) = Send stall request or request packet is in effect. |
| 4 | SETEND_ERROR | R/W | 0h | Setup end / Error.<br>Device mode: Setup end. This bit will be set when a control transaction ends before the DATAEND_SETUP bit has been set. An interrupt will be generated and the FIFO flushed at this time. The bit is cleared by the CPU writing a 1 to the SETENDC_NAKTO bit.<br>Host mode: Error indication. This bit will be set when three attempts have been made to perform a transaction with no response from the peripheral. The CPU should clear this bit. An interrupt is generated when this bit is set.<br>0h (R/W) = Clear Error.<br>Note: Only effective in host mode.<br>1h (R/W) = Setup end / Error indication. |



Copyright © 2026 Texas Instruments Incorporated

**Table 33-58. CSR0L Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 3 | DATAEND_SETUP | R/W | 0h | Data end / Setup Packet.<br>Device mode: Data end bit. The CPU sets this bit:<br>1. When setting TXRDY for the last data packet.<br>2. When clearing RXRDY after unloading the last data packet.<br>3. When setting TXRDY for a zero length data packet.<br>Host mode: Setup packet. The CPU sets this bit, at the same time as the TXRDY bit is set, to send a SETUP token instead of an OUT token for the transaction.<br>Note: Setting this bit always clears the DT bit in the USBCSRH0 register.<br>0h (R/W) = EFFECT : No effect<br>1h (R/W) = Indicates date end or sends setup token. |
| 2 | STALLED | R/W | 0h | Sent Stall / Receive Stall<br>Device mode: This bit is set when a STALL handshake is transmitted. The CPU should clear this bit.<br>Host mode: This bit is set when a STALL handshake is received. The CPU should clear this bit.<br>0h (R/W) = No handshake has been received.<br>1h (R/W) = A STALL handshake has been received or transmitted |
| 1 | TXRDY | R/W | 0h | Transmit Packet Ready.<br>Device & Host mode: The CPU sets this bit after loading a data packet into the FIFO. It is cleared automatically when a data packet has been transmitted. An interrupt is also generated at this point (if enabled).<br>Host mode: If both the TXRDY and DATAEND_SETUP bits are set, a setup packet is sent. If just TXRDY is set, an OUT packet is sent.<br>0h (R/W) = No transmit packet is ready.<br>1h (R/W) = Software sets this bit after loading a data packet into the TX FIFO. The EP0 bit in the USBTXIS register is also set in this situation. |
| 0 | RXRDY | R/W | 0h | Receive Packet Ready.<br>Device mode: This bit is set when a data packet has been received. An interrupt is generated when this bit is set. The CPU clears this bit by setting the RXRDYC_STATUS bit.<br>Host mode: Software must clear this bit after he packet has been read from the FIFO to acknowledge that the data has been read from the FIFO.<br>0h (R/W) = No receive packet has been received.<br>1h (R/W) = Indicates that a data packet has been received in the RX FIFO. The EP0 bit in the USBTXIS register is also set in this situation. |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.57 CSR0H (Offset = 2103h) [Reset = 00h]

CSR0H is shown in Figure 33-61 and described in Table 33-59.

Return to the Summary Table.

Control and Status Endpoint 0 High.
CSR0H is an 8-bit register that provides control and status bits for Endpoint 0.
Note: The interpretation of the register depends on whether the USB controller is acting as a peripheral or as a host.
Note: Users should also be aware that the value returned when the register is read reflects the status attained e.g. as a result of writing to the register.

**Figure 33-61. CSR0H**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | DISPING | DTWE | DT | FLUSH |
| R-0h | | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

**Table 33-59. CSR0H Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7-4 | RESERVED | R | 0h | |
| 3 | DISPING | R/W | 0h | Diable PING tokens in data and status phases of a high-speed Control transfer.<br>0h (R/W) = Ping is allowed.<br>1h (R/W) = Ping is not allowed. |
| 2 | DTWE | R/W | 0h | Data Toggle Write Enable.<br>This bit is automatically cleared once the new value is written.<br>Note: This bit is only effective in host mode.<br>0h (R/W) = The DT bit cannot be written.<br>1h (R/W) = Enables the current state of the endpoint 0 data toggle to be written (see DT bit). |
| 1 | DT | R/W | 0h | Data Toggle.<br>When read, this bit indicates the current state of the endpoint 0 data toggle. If DTWE is set, this bit may be written with the required setting of the data toggle. If DTWE is Low, this bit cannot be written. Care should be taken when writing to this bit as it should only be changed to RESET USB endpoint 0.<br>Note: This bit is only effective in host mode.<br>0h = Data toggle is low<br>1h = Data toggle is high |
| 0 | FLUSH | R/W | 0h | Flush FIFO.<br>The CPU writes a 1 to this bit to flush the next packet to be transmitted/read from the Endpoint 0 FIFO. The FIFO pointer is reset and the TXRDY/RXRDY bit in the USBCSRL0 register is cleared.<br>Note: FLUSH should only be used when TXRDY/RXRDY is set. At other times, it may cause data to be corrupted.<br>0h (R/W) = No effect<br>1h (R/W) = Flush the FIFO |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.58 COUNT0 (Offset = 2108h) [Reset = 00h]

COUNT0 is shown in Figure 33-62 and described in Table 33-60.

Return to the Summary Table.

Receive Byte Count Endpoint 0
COUNT0 is a 7-bit read-only register that indicates the number of received data bytes in the Endpoint 0 FIFO.
Note: The value returned changes as the contents of the FIFO change and is only valid while RXRDY (CSR0L.RXRDY) is set.

**Figure 33-62. COUNT0**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RESERVED | COUNT | | | | | | |
| R-0h | R-0h | | | | | | |

**Table 33-60. COUNT0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7 | RESERVED | R | 0h | |
| 6-0 | COUNT | R | 0h | FIFO Count.<br>COUNT is a read-only value that indicates the number of received data bytes in the endpoint 0 FIFO.<br>The value returned changes as the contents of the FIFO change and is only valid while RXRDY (USBCSRL.RXRDY) is set. |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.59 TYPE0 (Offset = 210Ah) [Reset = 00h]

TYPE0 is shown in Figure 33-63 and described in Table 33-61.

Return to the Summary Table.

Type Endpoint 0
These bits should be written with the operating speed of the targeted device.
Note: Host mode only.

**Figure 33-63. TYPE0**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| SPEED | | RESERVED | | | | | |
| R/W-0h | | R-0h | | | | | |

**Table 33-61. TYPE0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7-6 | SPEED | R/W | 0h | Operating Speed specifies the operating speed of the target Device. If selected, the target is assumed<br>to have the same connection speed as the USB controller.<br>0h = Unused (Note: If selected, the target will be assumed to be using the same connection speed as the USB controller.)<br>2h (R/W) = Full-speed<br>3h (R/W) = Low-speed |
| 5-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.60 NAKLMT0 (Offset = 210Bh) [Reset = 00h]

NAKLMT0 is shown in Figure 33-64 and described in Table 33-62.

Return to the Summary Table.

NAK Limit Endpoint 0.
NAKLMT is a 5-bit register that sets the number of frames after which Endpoint 0 should timeout on receiving a stream of NAK responses. (Equivalent settings for other endpoints can be made through their TXINTERVAL and RXINTERVAL registers.
Note: A value of 0 or 1 disables the NAK timeout function.

**Figure 33-64. NAKLMT0**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | NAKLMT | | | | |
| R-0h | | | R/W-0h | | | | |

**Table 33-62. NAKLMT0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7-5 | RESERVED | R | 0h | |
| 4-0 | NAKLMT | R/W | 0h | EP0 NAK Limit specifies the number of frames after receiving a stream of NAK responses. |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.61 CONFIG (Offset = 210Fh) [Reset = 06h]

CONFIG is shown in Figure 33-65 and described in Table 33-63.

Return to the Summary Table.

USB Configuration Data
CONFIG is an 8-bit Read-Only register that returns information about the USB controller configuration.

**Figure 33-65. CONFIG**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| MPRXE | MPTXE | BIGENDIAN | HBRXE | HBTXE | DYNFIFO | SOFTCON | UTMIDATWDTH |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-1h | R-1h | R-0h |

**Table 33-63. CONFIG Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7 | MPRXE | R | 0h | When set to ‘1’, automatic amalgamation of bulk packets is selected. |
| 6 | MPTXE | R | 0h | When set to ‘1’, automatic splitting of bulk packets is selected. |
| 5 | BIGENDIAN | R | 0h | Always '0'. Indicates Little Endian ordering. |
| 4 | HBRXE | R | 0h | When set to ‘1’ indicates High-bandwidth Rx ISO Endpoint Support selected. |
| 3 | HBTXE | R | 0h | When set to ‘1’ indicates High-bandwidth TX ISO Endpoint Support selected. |
| 2 | DYNFIFO | R | 1h | When set to ‘1’ indicates Dynamic FIFO Sizing option selected. |
| 1 | SOFTCON | R | 1h | Always ‘1’ . Indicates Soft Connect/Disconnect is enabled. |
| 0 | UTMIDATWDTH | R | 0h | Indicates selected UTMI+ data width. Always 0 indicating 8 bits. |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.62 TXMAXP[j] (Offset = 2110h + formula) [Reset = 0000h]

TXMAXP[j] is shown in Figure 33-66 and described in Table 33-64.

Return to the Summary Table.

Maximum Transmit Data Endpoint [j]
The TXMAXP register defines the maximum amount of data that can be transferred through the selected TX endpoint in a single operation. There is a TXMAXP register for each TX endpoint (except Endpoint 0).

Offset = 2110h + (j * 10h); where j = 0h to Eh

**Figure 33-66. TXMAXP[j]**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | MAXLOAD | | |
| R-0h | | | | | R/W-0h | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MAXLOAD | | | | | | | |
| R/W-0h | | | | | | | |

**Table 33-64. TXMAXP[j] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15-11 | RESERVED | R | 0h | |
| 10-0 | MAXLOAD | R/W | 0h | Maximum Payload specifies the maximum payload in bytes per transaction. |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.63 TXCSRL[j] (Offset = 2112h + formula) [Reset = 00h]

TXCSRL[j] is shown in Figure 33-67 and described in Table 33-65.

Return to the Summary Table.

Transmit Control and Status Endpoint [j] Low
TXCSRL is an 8-bit register that provides control and status bits for transfers for the endpoint. There is a TXCSRL register for each configured TX endpoint (not including Endpoint 0).
Note: The interpretation of the register depends on whether the USB controller is acting as a device or as a host. Users should also be aware that the value returned when the register is read reflects the status attained e.g. as a result of writing to the register.

Offset = 2112h + (j * 10h); where j = 0h to Eh

**Figure 33-67. TXCSRL[j]**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| INCOMPNAKTO | CLRDT | STALLED | STALLSETUP | FLUSH | UNDRNERROR | FIFONE | TXRDY |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

**Table 33-65. TXCSRL[j] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7 | INCOMPNAKTO | R/W | 0h | Incomplete TX / NAK Timeout.<br>Device mode: When the endpoint is being used for high-bandwidth Isochronous, this bit is set to indicate where a large packet has been split into 2 or 3 packets for transmission but insufficient IN tokens have been received to send all the parts. Note: In anything other than isochronous transfers, this bit<br>will always return 0.<br>Host mode - bulk endpoints only: This bit will be set when the TX endpoint is halted following the receipt of NAK responses for longer than the time set as theTXPOLLNAKLMT by the TXINTERVAL register. The CPU should clear this bit to allow the endpoint to continue.<br>Host mode - high-bandwidth Interrupt endpoints only: This bit will be set if no response is received from the device to which the packet is being sent.<br>0h (R/W) = No condition present / clear condition.<br>1h (R/W) = Condition present. |
| 6 | CLRDT | R/W | 0h | Clear DataToggle.<br>The CPU writes a 1 to this bit to reset the endpoint data toggle to 0.<br>Note: Host and device mode.<br>0h (R/W) = No effect<br>1h (R/W) = Writing a 1 to this bit clears the DT bit in the USBTXCSRH[n] register. |
| 5 | STALLED | R/W | 0h | Sent stall / RX stalled.<br>Device mode: This bit is set when a STALL handshake is transmitted. The FIFO is flushed and the TXRDY bit is cleared. The CPU should clear this bit.<br>Host mode: This bit is set when a STALL handshake is received. When this bit is set, any DMA request that is in progress is stopped, the FIFO is completely flushed and the TXRDY bit is cleared. The CPU should clear this bit.<br>0h (R/W) = Clear stall indication.<br>1h (R/W) = Indicates that a STALL handshake has been sent or received. |



Copyright © 2026 Texas Instruments Incorporated

**Table 33-65. TXCSRL[j] Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 4 | STALLSETUP | R/W | 0h | Send stall / Setup packet request.<br>Device mode: The CPU writes a 1 to this bit to issue a STALL handshake to an IN token. The CPU clears this bit to terminate the stall condition. Note: This bit has no effect where the endpoint is being used for Isochronous transfers.<br>Host mode: The CPU sets this bit, at the same time as the TXRDY bit is set, to send a SETUP token instead of an OUT token for the transaction. Note: Setting this bit also clears the USBTXCSRH.DT register.<br>0h (R/W) = No effect<br>1h (R/W) = Send stall or setup packet request. |
| 3 | FLUSH | R/W | 0h | Flush FIFO.<br>The CPU writes a 1 to this bit to flush the latest packet from the endpoint TX FIFO. The FIFO pointer is reset, the TXRDY bit is cleared and an interrupt is generated. May be set simultaneously with TXRDY to abort the packet that is currently being loaded into the FIFO.<br>Note: FLUSH should only be used when TXRDY is set. At other times, it may cause data to be corrupted. Also note that, if the FIFO is double-buffered, FLUSH may need to be set twice to completely clear the FIFO.<br>Note: Host and device mode.<br>0h (R/W) = No effect<br>1h (R/W) = Flushes the latest packet from the endpoint transmit FIFO. |
| 2 | UNDRNERROR | R/W | 0h | Underrun / Error condition.<br>Device mode: The USB controller sets this bit if an IN token is received when TXRDY is not set. The CPU should clear this bit.<br>Host mode: The USB controller sets this bit when 3 attempts have been made to send a packet and no handshake packet has been received. When the bit is set, an interrupt is generated, TXRDY is cleared and the FIFO is completely flushed. The CPU should clear this bit. Valid only when the endpoint is operating in Bulk or Interrupt mode.<br>0h (R/W) = No underrun or error condition<br>1h (R/W) = Underrun or error condition exist. |
| 1 | FIFONE | R/W | 0h | FIFO Not Empty.<br>The USB controller sets this bit when there is at least 1 packet in the TX FIFO.<br>Note: Host and device mode.<br>0h (R/W) = The FIFO is empty<br>1h (R/W) = At least one packet is in the transmit FIFO. |
| 0 | TXRDY | R/W | 0h | Transmit Packet Ready.<br>The CPU sets this bit after loading a data packet into the FIFO. This bit is cleared automatically when a data packet has been transmitted. The EPn bit in the USBTXIS register is also set at this point. TXRDY is also automatically cleared prior to loading a second packet into a double-buffered FIFO.<br>Note: Host and device mode<br>0h (R/W) = No transmit packet is ready.<br>1h (R/W) = Software sets this bit after loading a data packet into the TX FIFO. |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.64 TXCSRH[j] (Offset = 2113h + formula) [Reset = 00h]

TXCSRH[j] is shown in Figure 33-68 and described in Table 33-66.

Return to the Summary Table.

Transmit Control and Status Endpoint [j] High.
TXCSRH is an 8-bit register that provides additional control for transfers through the currently-selected TX endpoint. There is a TXCSRH register for each configured TX endpoint (not including Endpoint 0).
Note: The interpretation of the register depends on whether the USB controller is acting as a peripheral or as a host. Users should also be aware that the value returned when the register is read reflects the status attained e.g. as a result of writing to the register.

Offset = 2113h + (j * 10h); where j = 0h to Eh

**Figure 33-68. TXCSRH[j]**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| AUTOSET | ISO | MODE | DMAEN | FDT | DMAMOD | DTWE | DT |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

**Table 33-66. TXCSRH[j] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7 | AUTOSET | R/W | 0h | Auto Set.<br>If the CPU sets this bit, the TXRDY bit will be be automatically set when data of the maximum packet size (value in USBTXMAXP[n]) is loaded into the transmit FIFO. If a packet of less than the maximum packet size is loaded, then the TXRDY bit must be set manually.<br>Note: Should not be set for either high-bandwidth Isochronous endpoints or high-bandwidth Interrupt endpoints.<br>Note: Device and host mode.<br>0h (R/W) = The TXRDY bit must be set manually.<br>1h (R/W) = The TXRDY bit is automatically set when maximum packet size is loaded into the TX FIFO. |
| 6 | ISO | R/W | 0h | Isochronous Transfers.<br>The CPU sets this bit to enable the TX endpoint for Isochronous transfers, and clears it to enable the TX endpoint for Bulk or Interrupt transfers.<br>Note: This bit only has any effect in device mode. In host mode, it always returns zero.<br>0h (R/W) = Enables the transmit endpoint for bulk or interrupt transfers.<br>1h (R/W) = Enables the transmit endpoint for isochronous transfers. |
| 5 | MODE | R/W | 0h | Mode<br>The CPU sets this bit to enable the endpoint direction as TX, and clears the bit to enable it as Rx.<br>Note: This bit only has an effect when the same endpoint FIFO is used for both transmit and receive transactions.<br>Note: Host and device mode.<br>0h (R/W) = Enables the endpoint direction as RX.<br>1h (R/W) = Enables the endpoint direction as TX. |
| 4 | DMAEN | R/W | 0h | DMA Request Enable<br>The CPU sets this bit to enable the DMA request for the TX endpoint.<br>Note: Four TX and four RX endpoints can be connected to the DMA module. If this bit is set for a particular endpoint, the TRIGATX, TRIGBTX, TRIGCTX, or TRIGDTX field in the USB DMA Select (USBDMASEL) register must be programmed correspondingly.<br>Note: Device and host mode.<br>0h (R/W) = Disables the DMA request for the transmit endpoint.<br>1h (R/W) = Enables the DMA request for the transmit endpoint. |



Copyright © 2026 Texas Instruments Incorporated

**Table 33-66. TXCSRH[j] Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 3 | FDT | R/W | 0h | Force Data Toggle.<br>The CPU sets this bit to force the endpoint DT bit to switch and the data packet to be cleared from the FIFO, regardless of whether an ACK was received. This can be used by Interrupt TX endpoints that are used to communicate rate feedback for Isochronous endpoints.<br>Note: This bit should only be set when the TXRDY bit is set. At other times, it may cause data to be corrupted.<br>Note: Device and host mode.<br>0h (R/W) = No effect<br>1h (R/W) = Forces data toggle |
| 2 | DMAMOD | R/W | 0h | DMA Request Mode.<br>The CPU sets this bit to select DMA Request Mode 1 and clears it to select DMA Request Mode 0.<br>Note: This bit must not be cleared either before or in the same cycle as the DMAEN bit is cleared.<br>Note: Device and host mode.<br>0h (R/W) = An interrupt is generated after every DMA packet transfer.<br>1h (R/W) = An interrupt is generated only after the entire DMA transfer is complete. |
| 1 | DTWE | R/W | 0h | Data Toggle Write Enable.<br>The CPU writes a 1 to this bit to enable the current state of the TX Endpoint data toggle to be<br>written. This bit is automatically cleared once the new value is written.<br>Note: Host mode only.<br>0h (R/W) = The DT bit cannot be written.<br>1h (R/W) = Enables the current state of the transmit endpoint data to be written (see DT bit). |
| 0 | DT | R/W | 0h | Data Toggle.<br>When read, this bit indicates the current state of the transmit endpoint data toggle.<br>If DTWE is High, this bit can be written with the required setting of the data toggle. If DTWE is Low, any value written to this bit is ignored. Care should be taken when writing to this bit as it should only be changed to RESET the transmit endpoint.<br>Note: Host mode only<br>0h = Data toggle is low<br>1h = Data toggle is high |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.65 RXMAXP[j] (Offset = 2114h + formula) [Reset = 0000h]

RXMAXP[j] is shown in Figure 33-69 and described in Table 33-67.

Return to the Summary Table.

Maximum Receive Data Endpoint [j].
The RXMAXP register defines the maximum amount of data that can be transferred through the selected Rx endpoint in a single
operation. There is a RXMAXP register for each RX endpoint (except Endpoint 0).

Offset = 2114h + (j * 10h); where j = 0h to Eh

**Figure 33-69. RXMAXP[j]**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | MAXLOAD | | |
| R-0h | | | | | R/W-0h | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MAXLOAD | | | | | | | |
| R/W-0h | | | | | | | |

**Table 33-67. RXMAXP[j] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15-11 | RESERVED | R | 0h | |
| 10-0 | MAXLOAD | R/W | 0h | Maximum Payload specifies the maximum payload in bytes per transaction. |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.66 RXCSRL[j] (Offset = 2116h + formula) [Reset = 00h]

RXCSRL[j] is shown in Figure 33-70 and described in Table 33-68.

Return to the Summary Table.

Receive Control and Status Endpoint [j] Low.
RXCSRL is an 8-bit register that provides control and status bits for transfers through the currently-selected Rx endpoint. There is a RXCSRL register for each configured Rx endpoint (not including Endpoint 0). Note: The interpretation of the register depends on whether the USB controller is acting as a peripheral or as a host. Users should also be aware that the value returned when the register is read reflects the status attained e.g. as a result of writing to the register.

Offset = 2116h + (j * 10h); where j = 0h to Eh

**Figure 33-70. RXCSRL[j]**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| CLRDT | STALLED | STALLREQPKT | FLUSH | DATAERRNAKTO | OVERERROR | FULL | RXRDY |
| W1C-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | R/W-0h |

**Table 33-68. RXCSRL[j] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7 | CLRDT | W1C | 0h | Clear Data Toggle.<br>The CPU writes a 1 to this bit to reset the endpoint data toggle to 0.<br>Note: Device and host mode.<br>0h (R/W) = No effect<br>1h (R/W) = Writing a 1 to this bit clears the DT bit in the USBRXCSRH[n] register. |
| 6 | STALLED | R/W | 0h | Sent Stall / RX Stall<br>Device mode: This bit is set when a STALL handshake is transmitted. The CPU should clear this bit.<br>Host mode: When a STALL handshake is received, this bit is set and an interrupt is generated. The CPU should clear this bit.<br>0h (R/W) = No handshake has been received.<br>1h (R/W) = A STALL handshake has been received. The EPn bit in the USBRXIS register is also set. |
| 5 | STALLREQPKT | R/W | 0h | Send Stall / Request Packet.<br>Device mode: The CPU writes a 1 to this bit to issue a STALL handshake. The CPU clears this bit to terminate the stall condition.<br>- Note: This bit has no effect where the endpoint is being used for Isochronous transfers.<br>Host mode: The CPU writes a 1 to this bit to request an IN transaction. It is cleared when RXRDY is set.<br>0h (R/W) = No effect<br>1h (R/W) = Send stall or packet request. |
| 4 | FLUSH | R/W | 0h | Flush FIFO.<br>The CPU writes a 1 to this bit to flush the next packet to be read from the endpoint RX FIFO. The FIFO pointer is reset and the RXRDY bit is cleared.<br>Note: Thsi bit should only be set when RXRDY is set. At other times, it may cause data to be corrupted.<br>Note: If the FIFO is double-buffered, FLUSH may need to be set twice to completely clear the FIFO.<br>Note: Device and host mode.<br>0h (R/W) = No effect<br>1h (R/W) = Flushes the next packet to be read from the endpoint receive FIFO. The FIFO pointer is reset and the RXRDY bit is cleared |



Copyright © 2026 Texas Instruments Incorporated

**Table 33-68. RXCSRL[j] Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 3 | DATAERRNAKTO | R/W | 0h | Data Error / NAK Timeout<br>Device mode: Data Error - This bit is set when RXRDY is set, if the data packet has a CRC or bit-stuff error. It is cleared when RXRDY is cleared. - Note: This bit is only valid when the endpoint is operating in ISO mode. In Bulk mode, it always returns zero.<br>Host mode: NAK Timeout - When operating in ISO mode, this bit is set when RXRDY is set, if the data packet has a CRC or bit-stuff error and cleared when RXRDY is cleared. In Bulk mode, this bit will be set when the Rx endpoint is halted following the receipt of NAK responses for longer than the time set as the NAK Limit by the USBRXINTERVAL register. The CPU should clear this bit to allow the endpoint to continue.<br>0h (R/W) = Normal operation<br>1h (R/W) = Data error or NAK Timeout condition exist. |
| 2 | OVERERROR | R/W | 0h | Over run / Error.<br>Device mode: Over run - This bit is set if an OUT packet cannot be loaded into the receive FIFO. The CPU should clear this bit. - Note: This bit is only valid when the endpoint is operating in ISO mode. In Bulk mode, it always returns zero.<br>Host mode: The USB sets this bit when 3 attempts have been made to receive a packet and no data packet has been received. The CPU should clear this bit. An interrupt is generated when the bit is set. - Note: This bit is only valid when the RX endpoint is operating in Bulk or Interrupt mode. In ISO mode, it always returns zero.<br>0h (R/W) = No over run or error condition.<br>1h (R/W) = Under run or error condition exist. |
| 1 | FULL | R | 0h | FIFO Full.<br>This bit is set when no more packets can be loaded into the RX FIFO.<br>Note: Device and host mode.<br>0h (R/W) = The receive FIFO is not full.<br>1h (R/W) = The receive FIFO is full. No more packets can be loaded into the receive FIFO. |
| 0 | RXRDY | R/W | 0h | Receive Packet Ready.<br>This bit is set when a data packet has been received. The CPU should clear this bit when the packet has been unloaded from the Rx FIFO. An interrupt is generated when the bit is set.<br>If the AUTOCLR bit in the USBRXCSRH[n] register is set, then the this bit is automatically cleared when a packet of USBRXMAXP[n] bytes has been unloaded from the receive FIFO. If the AUTOCLR bit is clear, or if packets of less than the maximum packet size are unloaded, then software must clear this bit manually when the packet has been unloaded from the receive FIFO.<br>Note: Device and host mode.<br>0h (R/W) = No data packet has been received.<br>1h (R/W) = Indicates that a data packet has been received. The EP[j] bit in the USBTXIS register is also set in this situation |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.67 RXCSRH[j] (Offset = 2117h + formula) [Reset = 00h]

RXCSRH[j] is shown in Figure 33-71 and described in Table 33-69.

Return to the Summary Table.

Receive Control and Status Endpoint [j] High.
RXCSRH is an 8-bit register that provides additional control and status bits for transfers. There is a RXCSRH register for each configured Rx endpoint (not including Endpoint 0).
Note: The interpretation of the register depends on whether the USB controller is acting as a peripheral or as a host. Users should also be aware that the
value returned when the register is read reflects the status attained e.g. as a result of writing to the register.

Offset = 2117h + (j * 10h); where j = 0h to Eh

**Figure 33-71. RXCSRH[j]**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| AUTOCLR | ISOAUTORQ | DMAEN | PIDERR | DMAMOD | DTWE | DT | INCOMPRX |
| R/W-0h | R/W-0h | R/W-0h | R-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

**Table 33-69. RXCSRH[j] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7 | AUTOCLR | R/W | 0h | Auto Clear.<br>If the CPU sets this bit then the RXRDY bit will be automatically cleared when a packet of USBRXMAXP bytes has been unloaded from the RX FIFO. When packets of less than the maximum packet size are unloaded, RXRDY will have to be cleared manually. When using a DMA to unload the RX-FIFO, data is read from the RX-FIFO in 4 byte chunks regardless of the USBRXMAXP setting.<br>Note: Device and host mode.<br>0h (R/W) = No effect<br>1h (R/W) = Enables the RXRDY bit to be automatically cleared when a packet of USBRXMAXP[n] bytes has been unloaded from the receive FIFO. |
| 6 | ISOAUTORQ | R/W | 0h | Isochronous transfer mode / Auto request.<br>Device mode: The CPU sets this bit to enable the Rx endpoint for Isochronous transfers, and clears it to enable the Rx endpoint for Bulk/Interrupt transfers.<br>Host mode: If the CPU sets this bit, the STALLREQPKT bit will be automatically set when the RXRDY bit is cleared. - Note: This bit is automatically cleared when a short packet is received.<br>0h (R/W) = No effect<br>1h (R/W) = Sets ISO transfer mode or enables the REQPKT bit to be automatically set when the RXRDY bit is cleared. |
| 5 | DMAEN | R/W | 0h | DMA Request Enable<br>The CPU sets this bit to enable the DMA request for the Rx endpoint.<br>Note: Four TX and four RX endpoints can be connected to the DMA module. If this bit is set for a particular endpoint, the TRIGARX, TRIGBRX, TRIGCRX, or TRIGDRX field in the USB DMA Select (USBDMASEL) register must be programmed correspondingly<br>0h (R/W) = Disables the DMA request for the receive endpoint.<br>1h (R/W) = Enables the DMA request for the receive endpoint. |
| 4 | PIDERR | R | 0h | PID error<br>ISO Transactions: The USB controller sets this bit to indicate a PID error in the received packet.<br>Bulk/Interrupt Transactions: The setting of this bit is ignored.<br>Note: Device and host mode.<br>0h (R/W) = No error<br>1h (R/W) = Indicates a PID error in the received packet of an isochronous transaction. |



Copyright © 2026 Texas Instruments Incorporated

**Table 33-69. RXCSRH[j] Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 3 | DMAMOD | R/W | 0h | DMA request mode<br>The CPU sets this bit to select DMA Request Mode 1 and clears it to select DMA Request Mode 0.<br>Note: This bit must not be cleared either before or in the same cycle as the above DMAEN bit is cleared.<br>Note: Device and host mode.<br>0h (R/W) = An interrupt is generated after every DMA packet transfer.<br>1h (R/W) = An interrupt is generated only after the entire DMA transfer is complete. |
| 2 | DTWE | R/W | 0h | Data Toggle Write Enable.<br>The CPU writes a 1 to this bit to enable the current state of the Endpoint 0 data toggle to be written. This bit is automatically cleared once the new value is written.<br>Note: Host mode only.<br>0h (R/W) = The DT bit cannot be written.<br>1h (R/W) = Enables the current state of the receive endpoint data to be written (see DT bit). |
| 1 | DT | R/W | 0h | Data Toggle.<br>When read, this bit indicates the current state of the receive data toggle. If DTWE is High, this bit may be written with the required setting of the data toggle. If DTWE is Low, any value written to this bit is ignored. Care should be taken when writing to this bit as it should only be changed to RESET the receive endpoint.<br>Note: Host mode only.<br>0h = Data toggle is low<br>1h = Data toggle is high |
| 0 | INCOMPRX | R/W | 0h | This bit will be set in a high-bandwidth Isochronous or Interrupt transfer if the packet received is incomplete, because parts of the data were not received. It is cleared when RXRDY is cleared.<br>Note: In anything other than Isochronous transfer, this bit will always return 0.<br>Note: Device and host mode. |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.68 RXCOUNT[j] (Offset = 2118h + formula) [Reset = 0000h]

RXCOUNT[j] is shown in Figure 33-72 and described in Table 33-70.

Return to the Summary Table.

USB Receive Byte Count Endpoint [j].
USBRXCOUNT is a 13-bit read-only register that holds the number of data bytes in the packet currently in line to be read from the RX
FIFO. If the packet was transmitted as multiple bulk packets, the number given will be for the combined packet.
Note: The value returned changes as the FIFO is unloaded and is only valid while RXRDY is set.

Offset = 2118h + (j * 10h); where j = 0h to Eh

**Figure 33-72. RXCOUNT[j]**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | COUNT | | | | |
| R-0h | | | R-0h | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| COUNT | | | | | | | |
| R-0h | | | | | | | |

**Table 33-70. RXCOUNT[j] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15-13 | RESERVED | R | 0h | |
| 12-0 | COUNT | R | 0h | Receive Packet Count indicates the number of bytes in the receive packet. |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.69 TXTYPE[j] (Offset = 211Ah + formula) [Reset = 00h]

TXTYPE[j] is shown in Figure 33-73 and described in Table 33-71.

Return to the Summary Table.

Host Transmit Configure Type Endpoint [j].
TXTYPE is an 8-bit register that should be written with the endpoint number to be targeted by the endpoint, the transaction protocol to use for the currently-selected TX endpoint, and its operating speed. There is a TXRYPE register for each configured TX endpoint (except Endpoint 0, which has its own Type register). SPEED is only valid when the core is configured with the Multipoint option.
Note: Host mode only.

Offset = 211Ah + (j * 10h); where j = 0h to Eh

**Figure 33-73. TXTYPE[j]**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| SPEED | | PROTO | | TEP | | | |
| R/W-0h | | R/W-0h | | R/W-0h | | | |

**Table 33-71. TXTYPE[j] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7-6 | SPEED | R/W | 0h | Operating Speed.<br>This bit field specifies the operating speed of the target Device:<br>0h (R/W) = Default. The target is assumed to be using the same connection speed as the USB controller.<br>2h (R/W) = Full speed<br>3h (R/W) = Low speed |
| 5-4 | PROTO | R/W | 0h | Protocol.<br>Software must configure this bit field to select the required protocol for the transmit endpoint:<br>0h (R/W) = Control<br>1h (R/W) = isochronous<br>2h (R/W) = Bulk<br>3h (R/W) = Interrupt |
| 3-0 | TEP | R/W | 0h | Target Endpoint Number.<br>Software must configure this value to the endpoint number contained in the transmit endpoint descriptor returned to the USB controller during Device enumeration. |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.70 TXINTERVAL[j] (Offset = 211Bh + formula) [Reset = 00h]

TXINTERVAL[j] is shown in Figure 33-74 and described in Table 33-72.

Return to the Summary Table.

Host Transmit Interval Endpoint [j].
TXINTERVAL is an 8-bit register that, for Interrupt and Isochronous transfers, defines the polling interval for the currently-selected TX endpoint. For Bulk endpoints, this register sets the number of frames/microframes after which the endpoint should timeout on receiving a stream of NAK responses.
Note: Host mode only

Offset = 211Bh + (j * 10h); where j = 0h to Eh

**Figure 33-74. TXINTERVAL[j]**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| TXPOLLNAKLMT | | | | | | | |
| R/W-0h | | | | | | | |

**Table 33-72. TXINTERVAL[j] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7-0 | TXPOLLNAKLMT | R/W | 0h | TX Polling / NAK Limit The polling interval for interrupt/isochronous transfers; the NAK limit for bulk transfers. |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.71 RXTYPE[j] (Offset = 211Ch + formula) [Reset = 00h]

RXTYPE[j] is shown in Figure 33-75 and described in Table 33-73.

Return to the Summary Table.

Host Configure Receive Type Endpoint [j].
RXTYPE is an 8-bit register that should be written with the endpoint number to be targeted by the endpoint, the transaction protocol to use for the currently-selected RX endpoint, and its operating speed. There is a RXTYPE register for each configured RX endpoint (except Endpoint 0, which has its own Type register). SPEED are only valid when the core is configured with the Multipoint option.
Note: Host mode only.

Offset = 211Ch + (j * 10h); where j = 0h to Eh

**Figure 33-75. RXTYPE[j]**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| SPEED | | PROTO | | TEP | | | |
| R/W-0h | | R/W-0h | | R/W-0h | | | |

**Table 33-73. RXTYPE[j] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7-6 | SPEED | R/W | 0h | Operating Speed.<br>This bit field specifies the operating speed of the target Device:<br>0h (R/W) = Default. The target is assumed to be using the same connection speed as the USB controller.<br>2h (R/W) = Full speed<br>3h (R/W) = Low speed |
| 5-4 | PROTO | R/W | 0h | Protocol.<br>Software must configure this bit field to select the required protocol for the transmit endpoint:<br>0h (R/W) = Control<br>1h (R/W) = isochronous<br>2h (R/W) = Bulk<br>3h (R/W) = Interrupt |
| 3-0 | TEP | R/W | 0h | Target Endpoint Number.<br>Software must configure this value to the endpoint number contained in the transmit endpoint descriptor returned to the USB controller during Device enumeration. |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.72 RXINTERVAL[j] (Offset = 211Dh + formula) [Reset = 00h]

RXINTERVAL[j] is shown in Figure 33-76 and described in Table 33-74.

Return to the Summary Table.

Host Receive Polling Interval Endpoint [j].
RXINTERVAL is an 8-bit register that, for Interrupt and Isochronous transfers, defines the polling interval for the currently-selected RX endpoint. For Bulk endpoints, this register sets the number of frames/microframes after which the endpoint should timeout on receiving a stream of NAK responses. There is a RXINTERVAL register for each configured RX endpoint (except Endpoint 0). In each case the value that is set defines a number of frames. Note: Host mode only.

Offset = 211Dh + (j * 10h); where j = 0h to Eh

**Figure 33-76. RXINTERVAL[j]**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RXPOLLNAKLMT | | | | | | | |
| R/W-0h | | | | | | | |

**Table 33-74. RXINTERVAL[j] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7-0 | RXPOLLNAKLMT | R/W | 0h | RX Polling / NAK Limit The polling interval for interrupt/isochronous transfers; the NAK limit for bulk transfers. |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.73 FIFOSZ[j] (Offset = 211Fh + formula) [Reset = 00h]

FIFOSZ[j] is shown in Figure 33-77 and described in Table 33-75.

Return to the Summary Table.

FIFO Size Endpoint [j].
FIFOSZ is an 8-bit Read-Only register that returns the sizes of the FIFOs associated with the selected additional TX/Rx endpoints. The lower nibble encodes the size of the selected TX endpoint FIFO; the upper nibble encodes the size of the selected Rx endpoint FIFO. Values of 3 – 13 correspond to a FIFO size of 2n bytes (8 – 8192 bytes). If an endpoint has not been configured, a value of 0 will be displayed. Where the TX and Rx endpoints share the same FIFO, the Rx FIFO size will be encoded as 0xF.
Note: The register only has this interpretation when the Index register is set to select one of Endpoints 1 – 15 and Dynamic Sizing is not selected. It has a special interpretation when the Index register is set to select Endpoint 0, while the result returned is not valid where Dynamic FIFO sizing is used.

Offset = 211Fh + (j * 10h); where j = 0h to Eh

**Figure 33-77. FIFOSZ[j]**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RXSIZE | | | | TXSIZE | | | |
| R-0h | | | | R-0h | | | |

**Table 33-75. FIFOSZ[j] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7-4 | RXSIZE | R | 0h | Returns the configured size of the RX FIFO<br>0h (R/W) = FIFO size 8-byte<br>1h (R/W) = FIFO size 16-byte<br>2h (R/W) = FIFO size 32-byte<br>3h (R/W) = FIFO size 64-byte<br>4h (R/W) = FIFO size 128-byte<br>5h (R/W) = FIFO size 256-byte<br>6h (R/W) = FIFO size 512-byte<br>7h (R/W) = FIFO size 1024-byte<br>8h (R/W) = FIFO size 2048-byte |
| 3-0 | TXSIZE | R | 0h | Returns the configured size of the TX FIFO<br>0h (R/W) = FIFO size 8-byte<br>1h (R/W) = FIFO size 16-byte<br>2h (R/W) = FIFO size 32-byte<br>3h (R/W) = FIFO size 64-byte<br>4h (R/W) = FIFO size 128-byte<br>5h (R/W) = FIFO size 256-byte<br>6h (R/W) = FIFO size 512-byte<br>7h (R/W) = FIFO size 1024-byte<br>8h (R/W) = FIFO size 2048-byte |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.74 RQPKTCOUNT[y] (Offset = 2304h + formula) [Reset = 00000000h]

RQPKTCOUNT[y] is shown in Figure 33-78 and described in Table 33-76.

Return to the Summary Table.

Request Packet Count in Block Transfer Endpoint [y].
For each Rx Endpoint 1 – 15, the USB controller provides a 16-bit RQPKTCOUNT register. This read/write register is used in Host mode to specify the number of packets that are to be transferred in a block transfer of one or more Bulk packets of length MaxP to Rx Endpoint n. The core uses the value recorded in this register to determine the number of requests to issue where the AutoReq option (included in the RXCSR register) has been set.

Offset = 2304h + (y * 4h); where y = 0h to Eh

**Figure 33-78. RQPKTCOUNT[y]**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | COUNT | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | R/W-0h | | | | | | | | | | | | | | | |

**Table 33-76. RQPKTCOUNT[y] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15-0 | COUNT | R/W | 0h | Block Transfer Packet Count sets the number of packets of the size defined by the MAXLOAD bit field that are to be transferred in a block transfer.<br>Note: This is only used in Host mode when AUTORQ is set. The bit has no effect in Device mode or when AUTORQ is not set. |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.75 RXDPKTBUFDIS (Offset = 2340h) [Reset = 0000h]

RXDPKTBUFDIS is shown in Figure 33-79 and described in Table 33-77.

Return to the Summary Table.

USB Receive Double Packet Buffer Disable.
For each Rx Endpoint 1 – 15, the USB controller provides a RXDPKTBUFDIS bit. These read/write bits are used to control the use of double packet buffering on a per endpoint basis. When asserted (RXPKTBUFDIs equals 1'b1), the bit will disable double packet buffering for the corresponding endpoint regardless of the End Point FIFO Size and the INMAXP size relationship. When de-asserted (RXDPKTBUFDIS equals 1'b0), this bit does NOT necessarily enable double packet buffering but rather allows double packet buffering to be determined based upon the End Point FIFO Size and INMAXP size relationship.

**Figure 33-79. RXDPKTBUFDIS**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| EP7 | EP6 | EP5 | EP4 | EP3 | EP2 | EP1 | RESERVED |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 33-77. RXDPKTBUFDIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15-8 | RESERVED | R | 0h | |
| 7 | EP7 | R | 0h | EP7 RX Double Packet Buffer Disable<br>0h (R/W) = Disables double-packet buffering.<br>1h (R/W) = Enables double-packet buffering. |
| 6 | EP6 | R | 0h | EP6 RX Double Packet Buffer Disable<br>0h (R/W) = Disables double-packet buffering.<br>1h (R/W) = Enables double-packet buffering. |
| 5 | EP5 | R | 0h | EP5 RX Double Packet Buffer Disable<br>0h (R/W) = Disables double-packet buffering.<br>1h (R/W) = Enables double-packet buffering. |
| 4 | EP4 | R | 0h | EP4 RX Double Packet Buffer Disable<br>0h (R/W) = Disables double-packet buffering.<br>1h (R/W) = Enables double-packet buffering. |
| 3 | EP3 | R | 0h | EP3 RX Double Packet Buffer Disable<br>0h (R/W) = Disables double-packet buffering.<br>1h (R/W) = Enables double-packet buffering. |
| 2 | EP2 | R | 0h | EP2 RX Double Packet Buffer Disable<br>0h (R/W) = Disables double-packet buffering.<br>1h (R/W) = Enables double-packet buffering. |
| 1 | EP1 | R | 0h | EP1 RX Double Packet Buffer Disable<br>0h (R/W) = Disables double-packet buffering.<br>1h (R/W) = Enables double-packet buffering. |
| 0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 33.5.76 TXDPKTBUFDIS (Offset = 2342h) [Reset = 0000h]

TXDPKTBUFDIS is shown in Figure 33-80 and described in Table 33-78.

Return to the Summary Table.

Transmit Double Packet Buffer Disable.
For each Tx Endpoint 1 – 15, the USB controller provides a TXDPKTBUFDIS bit. These read/write bits are used to control the use of double packet buffering on a per endpoint basis. When asserted (TXPKTBUFDIs equals 1'b1), the bit will disable double packet buffering for the corresponding endpoint regardless of the End Point FIFO Size and the INMAXP size relationship. When de-asserted (TXDPKTBUFDIS equals 1'b0), this bit does NOT necessarily enable double packet buffering but rather allows double packet buffering to be determined based upon the End Point FIFO Size and INMAXP size relationship.

**Figure 33-80. TXDPKTBUFDIS**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| EP7 | EP6 | EP5 | EP4 | EP3 | EP2 | EP1 | RESERVED |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 33-78. TXDPKTBUFDIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 15-8 | RESERVED | R | 0h | |
| 7 | EP7 | R | 0h | EP7 TX Double Packet Buffer Disable<br>0h (R/W) = Disables double-packet buffering.<br>1h (R/W) = Enables double-packet buffering. |
| 6 | EP6 | R | 0h | EP6 TX Double Packet Buffer Disable<br>0h (R/W) = Disables double-packet buffering.<br>1h (R/W) = Enables double-packet buffering. |
| 5 | EP5 | R | 0h | EP5 TX Double Packet Buffer Disable<br>0h (R/W) = Disables double-packet buffering.<br>1h (R/W) = Enables double-packet buffering. |
| 4 | EP4 | R | 0h | EP4 TX Double Packet Buffer Disable<br>0h (R/W) = Disables double-packet buffering.<br>1h (R/W) = Enables double-packet buffering. |
| 3 | EP3 | R | 0h | EP3 TX Double Packet Buffer Disable<br>0h (R/W) = Disables double-packet buffering.<br>1h (R/W) = Enables double-packet buffering. |
| 2 | EP2 | R | 0h | EP2 TX Double Packet Buffer Disable<br>0h (R/W) = Disables double-packet buffering.<br>1h (R/W) = Enables double-packet buffering. |
| 1 | EP1 | R | 0h | EP1 TX Double Packet Buffer Disable<br>0h (R/W) = Disables double-packet buffering.<br>1h (R/W) = Enables double-packet buffering. |
| 0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

Chapter 34

# I2S/TDM

The I2S/TDM module provides a standardized serial interface to transfer audio data. This chapter describes the operation of the I2S/TDM module.


**34.1 I2S/TDM Introduction** ..... 2051
**34.2 I2S/TDM Operation** ..... 2051
**34.3 I2S Registers** ..... 2065





Copyright © 2026 Texas Instruments Incorporated

## 34.1 I2S/TDM Introduction

The I2S/TDM module provides a standardized serial interface to transfer audio data. The module can be configured in different modes including I2S standards, LSB or MSB-justified, PCM/DSP and TDM for example. It can be used in conjunction with the DMA controller.

### *34.1.1 I2S/TDM features*

- Configurable and independent transmitter and receiver functions on data lines
- Configurable controller or target function
- Integrated transmitter and receiver FIFO with packing feature
- Clock generator to target specific audio frequency generation
- Independent channel/slot length and audio word length of 8, 16, 24 or 32-bits
- Audio protocol: I2S, Left- or Right-justified, PCM/DSP, TDM
- Up to 16 slots available in TDM format
- Empty slot configuration for sending 0's, 1's or Hi-Z
- Configurable bit clock sampling edge
- Frame synchronization configurable for offset and bit length
- Independent DMA request for transmitter and receiver functions

## 34.2 I2S/TDM Operation

### *34.2.1 Functional Block Diagram*

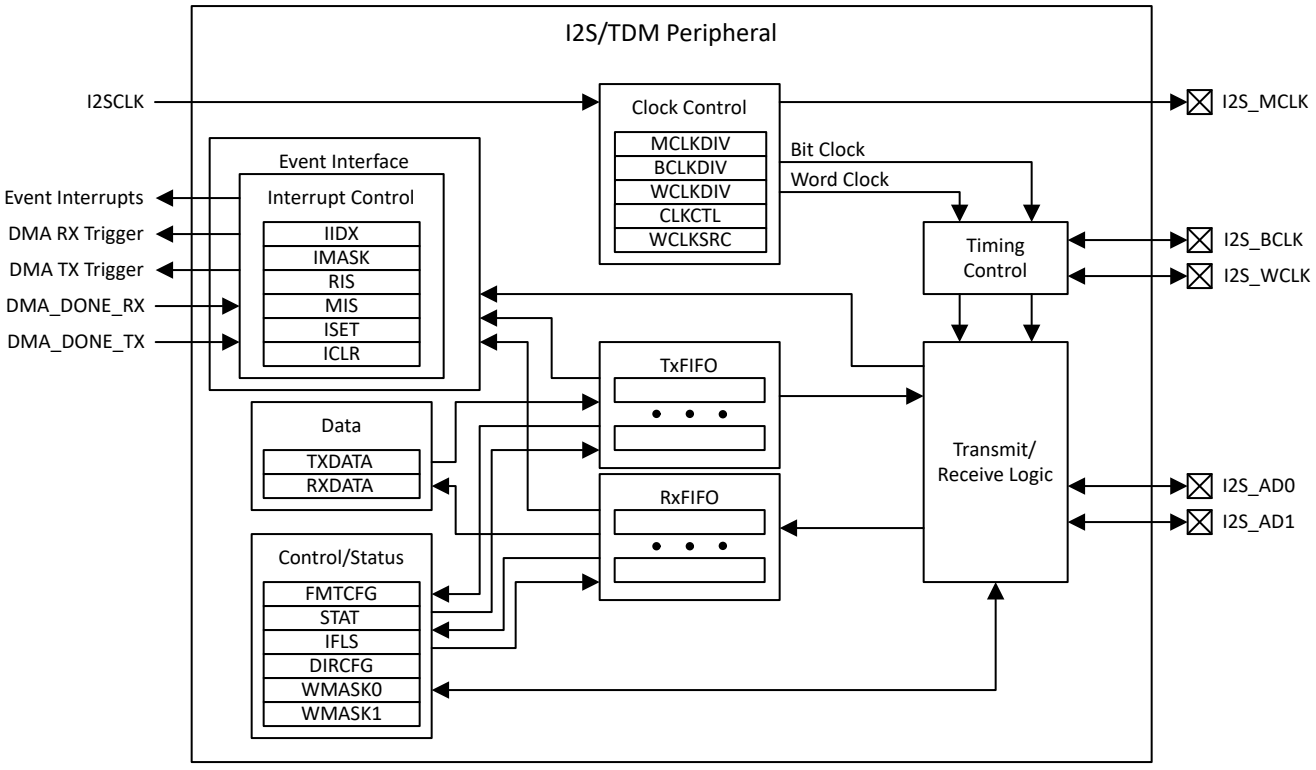



**Figure 34-1. I2S/TDM Functional Block Diagram**

**Note**

Refer to TxFIFO and RxFIFO depth in device specific data sheet.

### *34.2.2 Modes of Operation*

The I2S module can be configured as a controller or target function.



Copyright © 2026 Texas Instruments Incorporated

#### 34.2.2.1 Controller Mode

In controller mode the I2S module generates the bit clock (I2S_BCLK) and word clock (I2S_WCLK). The I2S controller supports a highly configurable data pins I2S_ADx which can be independently configured in transmitter or receiver functions. This allows the controller to support dual transmitter, dual receiver or standard transmit-receive function.

When the module is configured as a controller

- The source clock selection is through registers in system control and can be sourced from multiple sources.
- The controller generates the I2S_BCLK using the I2S.BCLKDIV register.
- The controller generates the I2S_WCLK using the I2S.WCLKDIV register.
- The module can generate an optional master clock (I2S_MCLK) using I2S.MCLKDIV register.

#### 34.2.2.2 Target Mode

In target mode the I2S module recieves the bit clock (I2S_BCLK) and word clock (I2S_WCLK). The I2S target also supports a highly configurable data pins I2S_ADx which can be independently configured in transmitter or receiver functions. This allows the controller to support dual transmitter, dual receiver or standard transmit-receive function.

When the module is configured as a target

- The I2S_BCLK and I2S_WCLK must be sourced through the pad.

### *34.2.3 Clock and Timing Control*

The I2S/TDM internal functional clock is selected from either one of the device clocks (when the module is a controller) or from the pad (when the module is a target).

- Use the I2S.WCLKSRC register to select the source of clock as internal or external.
- If internal clock source is selected, use the SYSCTL.I2S.CLKCFG register to select the source of clock. Refer to the device data sheet for the options available for internal clock sources.
  - SYSOSC
  - HFXT
  - PLL
  - EXTCLK

The clock generator is a highly programmable block in the module which is used to derive the BCLK, WCLK and MCLK. The BCLK and MCLK generation is straightforward and Equation 30 and Equation 31 can be used to derive the clock frequency respectively.

$$BCLK\ Frequency = \frac{Module\ Clock\ Frequency}{BCLKDIV} \tag{30}$$

$$MCLK\ Frequency = \frac{Module\ Clock\ Frequency}{MCLKDIV} \tag{31}$$

The WCLK generation is calculated differently depending on the configuration of CLKCTL.WCLKPHASE register bit.

For frame format where the WCLK is high for one clock (single-phase) the CLKCTL.WCLKPHASE is configured as 0, the Equation 32 defines the equation for calculating the WCLK frequency or frame rate.

$$WCLK\ Frequency\left(Frame\ Rate\right) = \frac{Module\ Clock\ Frequency}{(BCLKDIV \times (WCLKDIV[9:0] + 1))} \tag{32}$$

For frame format where the WCLK is high with a 50% duty cycle (dual-phase) the CLKCTL.WCLKPHASE is configured as 1, Equation 33 defines the equation for calculating the WCLK frequency or frame rate.

$$WCLK\ Frequency\left(Frame\ Rate\right) = \frac{Module\ Clock\ Frequency}{(BCLKDIV \times (2 \times WCLKDIV[9:0]))} \tag{33}$$



Copyright © 2026 Texas Instruments Incorporated

For user defined frame format where WCLK can be arbitrarily high for CLKCTL.WCLKDIV[7:0] and low for CLKCTL.WCLKDIV[15:8], the WCLKPHASE is configured as 2. Equation 34 defines the equation for calculating the WCLK frequency or frame rate.

$$WCLK\ Frequency\left(Frame\ Rate\right) = \frac{Module\ Clock\ Frequency}{(BCLKDIV \times (WCLKDIV[7:0] + WCLKDIV[15:8]))} \quad (34)$$

**Note**

When in controller mode, the bit clock may have stringent requirements for precise clock frequency and jitter specifications. It is preferred that the application configure the internal clock source as HFXT, EXTCLK or PLL (if the clock source for the PLL is HFXT).

### *34.2.4 Frame Synchronization*

The I2S_WCLK signal acts as the frame synchronization signal in the audio frame to denote the start of frame. The signal is completely configurable in order to target the different audio formats. The configurability of the WCLK is managed using CLKCTL, WCLKSRC and WCLKDIV registers.

The I2S_WCLK can be configured based on the serial format. This is configured using the CLKCTL.WCLKPHASE and correspondingly the approproate WCLKDIV.WDIV values

- CLKCTL.WCLKPHASE = 0: Start of frame for data formats such as DSP and PCM short frame format, where I2S_WCLK is high for one and low for WCLKDIV.WDIV[9:0] I2S_BCLK.
- CLKCTL.WCLKPHASE = 1: Start of frame and channel identification for data formats such as I2S, Left-justifed and Right-justified, where I2S_WCLK is high or low for number of bit clocks configured in WCLKDIV.WDIV[9:0].
- CLKCTL.WCLKPHASE = 2: Start of frame for custom formats such as TDM and PCM long frame, where I2S_WCLK is high for WCLKDIV.WDIV[7:0] and low for WCLKDIV.WDIV[15:8] I2S_BCLK

#### 34.2.4.1 Frame and Word Length

The frame length is calculated as the number of I2S_BCLK cycles from a rising or falling I2S_WCLK to the corresponding edge on the next frame. In a frame, the word length or number of data bits per slot/channel is configured through FMTCFG.WORDLEN which has a minimum value of 8 bits and maximum value of 32 bits. The I2S_WCLK is used in target mode to automatically detect the number of I2S_BCLK used for a channel for dual-phase formats:

- If the I2S_WCLK edge occurs before the word length count is over, then the sample word is truncated
- If the I2S_WCLK edge occurs after the word length count is over, then the sample word is zero padded in the least significant bit position.

The data that is transmitted or received on the I2S_ADx comes from the FIFO which can be configured using the FMTCFG.MEMLEN32 as 16-bit or 32-bit data per entry in the FIFO.

**FMTCFG.MEMLEN32 = 0**

When configured as 0, each FIFO entry holds data in the lower 16-bit and the upper 16-bit is unused. This setting must only be used when FMTCFG.WORDLEN ≤ 16-bits. If word length programmed is less than 16-bits then sample words are truncated when transmitting and zero-padded when receving in the least significant bit position as shown in Figure 34-2.



Copyright © 2026 Texas Instruments Incorporated

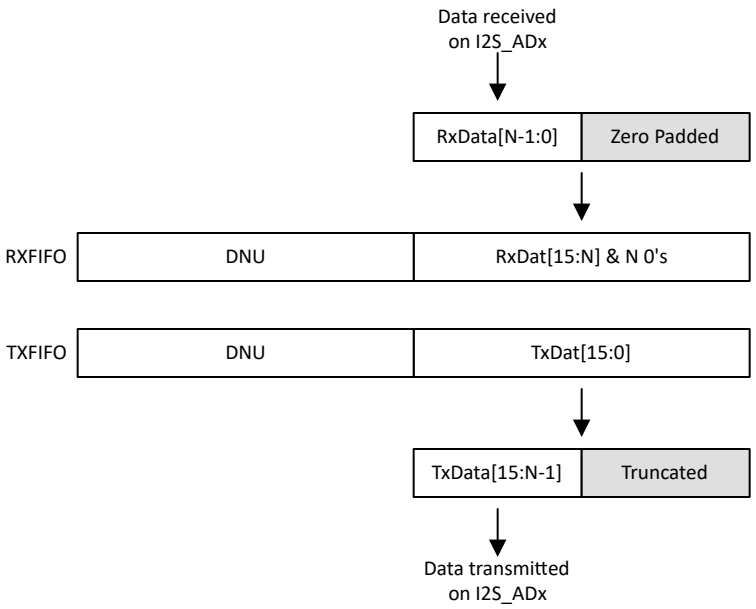



**Figure 34-2. Data Mapping for MEMLEN32=0**

### FMTCFG.MEMLEN32 = 1

When configured as 1, each FIFO entry holds two samples when FMTCFG.WORDLEN ≤ 16-bits or a single entry when FMTCFG.WORDLEN > 16-bits. If the word length programmed results in truncation on transmit or zero-padding on receive, the operation happens in the least significant bit position as shown in Figure 34-3.

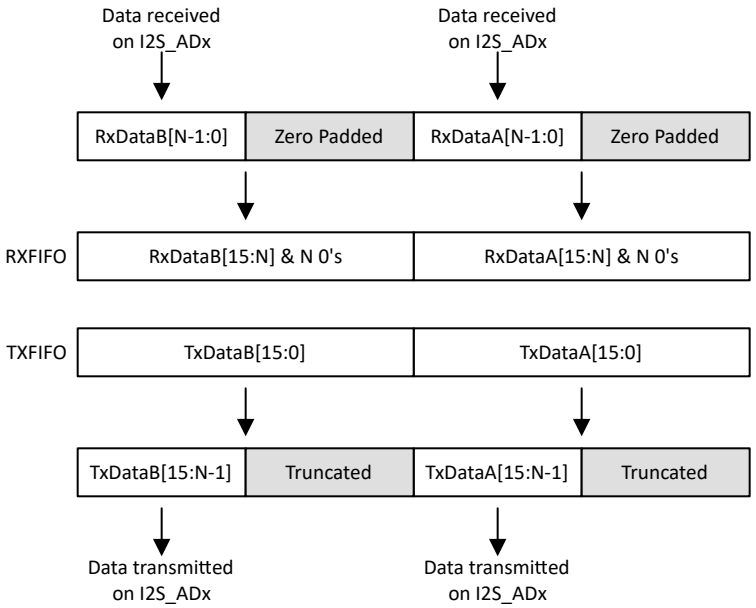



**Figure 34-3. Data Mapping for MEMLEN32=1**

#### 34.2.4.2 Polarity

The I2S_WCLK first active edge is configured using WCLKSRC.WCLKINV field. When the field is programmed as "0", the first I2S_WCLK starts with a rising edge on the first I2S_BCLK. When the field is programmed as "1", the first I2S_WCLK starts with a falling edge on the first I2S_BCLK. Data transmission and receive on I2S_ADx also follows the same phase as the I2S_WCLK for transmit and receive.



Copyright © 2026 Texas Instruments Incorporated

It is important to note that when I2S module is disabled, the I2S_WCLK may be either be "0" or "1" and requires a peripheral reset to ensure that it starts in the correct state.

#### 34.2.4.3 Data Delay (Offset)

Based on the audio frame format, the I2S_ADx can be delayed or not delayed with respect to the I2S_WCLK edge. This configuration is enabled using the FMTCFG.DATADLY field. The valid values that can be programmed are from 0 to 255 I2S_BCLK.

When programmed as non-zero value, the I2S_ADx is transmitted or sampled after FMTCFG.DATADLY number of I2S_BCLK after the I2S_WCLK edge is transmitted.

### *34.2.5 Slot Mapping and Configuration*

The I2S module supports up to 16 slots or channels for different audio formats which can be configured through the DIRCFG and WMASKx register. Each I2S_ADx has a corresponding WMASKx register which is used to configure the channel or slot count. In single-phase format up to 16-bits are available depending on the device capabilities on number of slots supported.

- WMASKx.MASK[n] = 0: The corresponding slot is not used. No data will be transmitted nor will it be received
- WMASKx.MASK[n] = 1: The corresponding slot is used. Data will be transmitted if I2S_ADx is configured in output mode or data will be received if I2S_ADx is configured in input mode.

The slots are configured with least significant bit for the lowest channel. In single-phase formats, channel-0 corresponds to bit-0 in WCMASKx.MASK register, channel-1 to bit-1 and so on. In dual-phase formats only lower 2-bits of the registers are considered for the left and right channels.

Moreover, when a output slot is disabled the behavior of the empty slot can be decided using FMTCFG.EMPTYSLOTOUTPUT

- FMTCFG.EMPTYSLOTOUTPUT = 0: All zeroes are sent during the empty slot
- FMTCFG.EMPTYSLOTOUTPUT = 1: All ones are sent during the empty slot
- FMTCFG.EMPTYSLOTOUTPUT = 2: The output line is tri-stated
- FMTCFG.EMPTYSLOTOUTPUT = 3: Reserved and must not be used

#### Data Direction Configuration

For each I2S_ADx, the direction of data flow (data transmit or receive) is controlled using the DIRCFG.ADx pins.

- DIRCFG.ADx = 0: The corresponding I2S_ADx pin is disabled.
- DIRCFG.ADx = 1: The corresponding I2S_ADx pin is set as input. Data is received on the pin and stored in the RXFIFO.
- DIRCFG.ADx = 2: The corresponding I2S_ADx pin is set as output. Data is transmitted from the TXFIFO to the pin.
- DIRCFG.ADx = 3: Reserved and must not be used.

#### 34.2.5.1 Channel Mapping in Memory

Based on the slot mapping and channel configuration, data is read from the RXFIFO or written to the TXFIFO. For the MCU application, the mapping of the channel in the FIFO is important to understand how data will be received or transmitted to external devices and to maintain appropriate data structure management.

Audio data is always stored in little-endian format in the FIFO. When multiple I2S_ADx pins are used, data is read or stored with I2S_AD0 stored first and I2S_AD1 stored next. For cases where FMTCFG.MEMLEN32 = 1 and data is packed, I2S_AD0 is stored in the lower bytes and I2S_AD1 stored in the upper bytes. However, there are caveats based on configured multi-channels/slots in TDM mode, where individual channel/slot(s) may be disabled. Figure 34-4 to Figure 34-9 show some examples of mapping channel data to memory. Any channel which is grayed out is inactive and no data shall be read or written in memory.



Copyright © 2026 Texas Instruments Incorporated

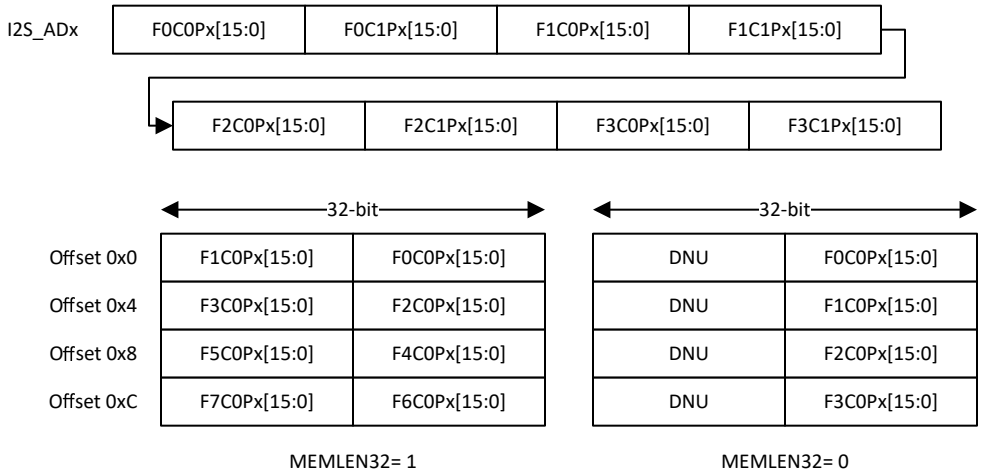



**Figure 34-4. 16-bit Mono Single I2S_ADx to Memory Mapping**

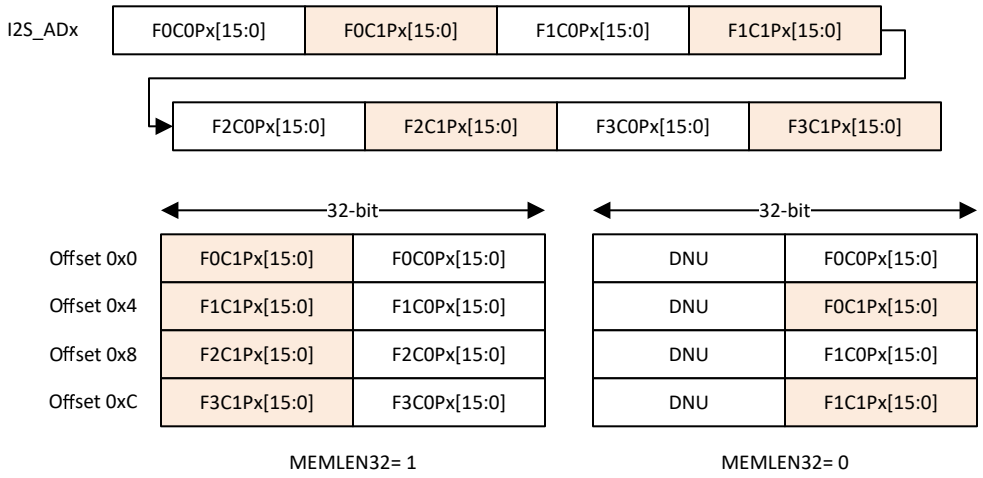
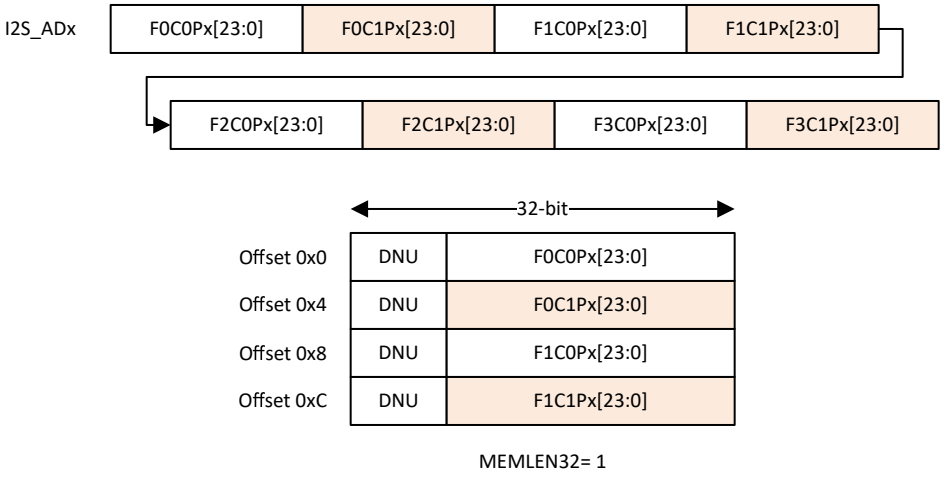



**Figure 34-5. 16-bit Stereo Single I2S_ADx to Memory Mapping**




**Note**

MEMLEN32=0 is not supported

**Figure 34-6. 24-bit Stereo Single I2S_ADx to Memory Mapping**



Copyright © 2026 Texas Instruments Incorporated

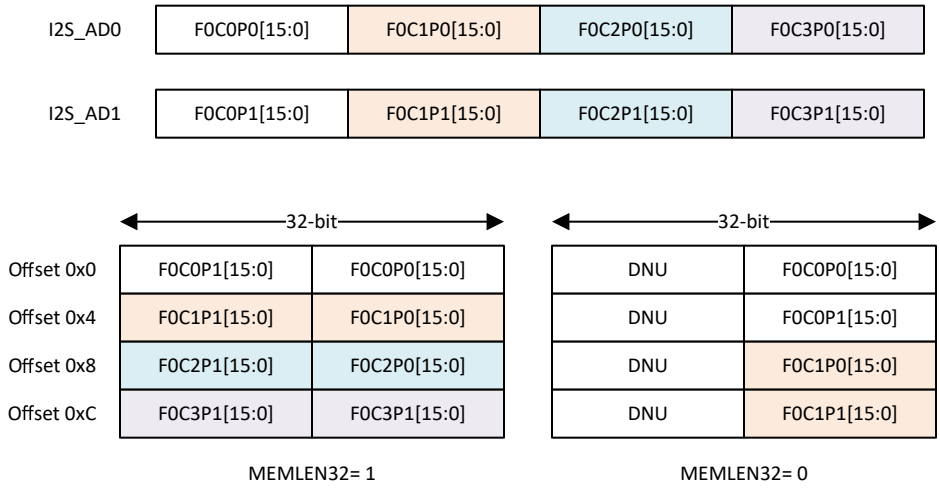



**Figure 34-7. 16-bit TDM Dual I2S_ADx to Memory Mapping**

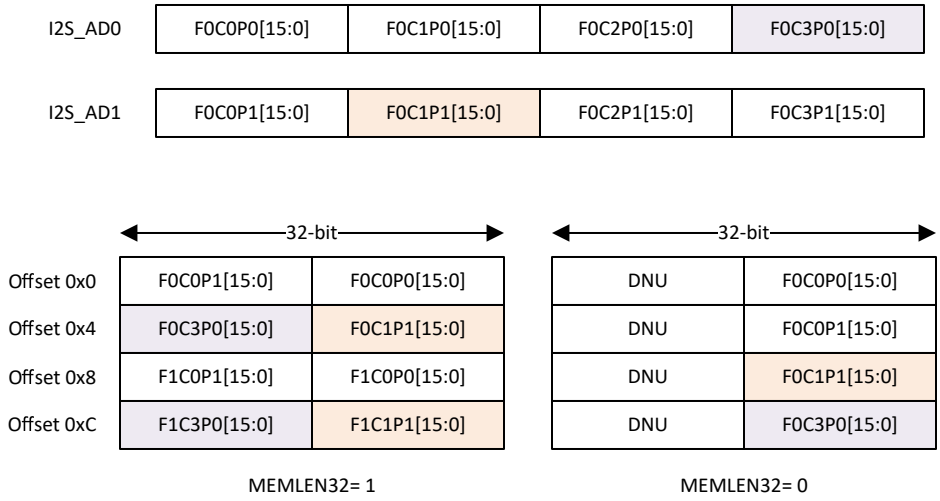



**Figure 34-8. 16-bit TDM Dual I2S_ADx Slots Disabled to Memory Mapping**

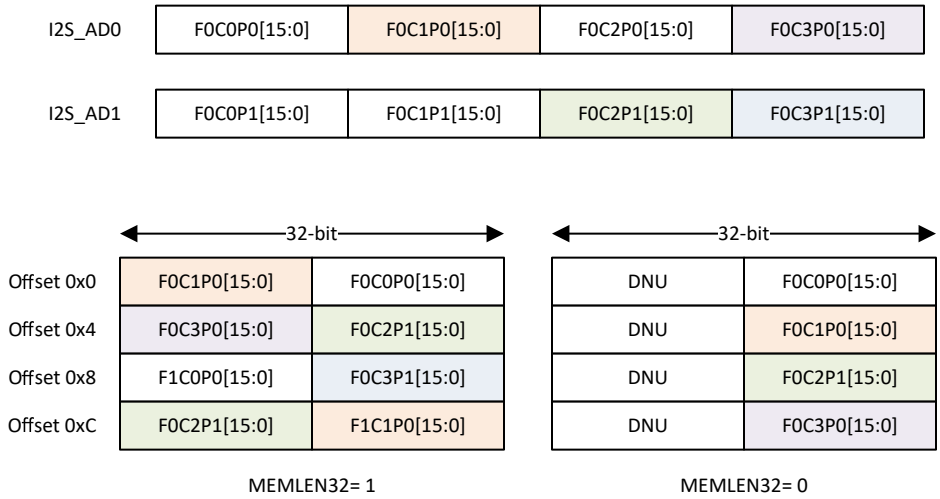



**Figure 34-9. 16-bit TDM Dual I2S_ADx Slots Disabled to Memory Mapping**



Copyright © 2026 Texas Instruments Incorporated

### 34.2.6 Serial Frame Format Examples

The module supports the dual-phase formats I2S, Left-Justfified Format (LJF) and Right-Justified Format (RJF), which supports one or two audio channels per I2S_ADx pin. The module also supports the single-phase format, DSP, PCM and TDM, which supports up to 16 slots.

The I2S_WCLK and I2S_ADx signals are updated on one edge of the I2S_BCLK and sampled on the opposite (trailing edge). All of the serial formats are comprised of *frames*. Each frame starts with the first (rising or falling) I2S_WCLK edge of an audio sample (one or more slots) to the next corresponding (rising or falling) I2S_WCLK edge. The audio sample words transferred on the I2S_ADx are aligned with the I2S_WCLK.

The different serial formats supported by the module require the application to setup the module configuration accordingly as shown in Table 34-1. Subsequent sub-sections illustrate how the application can generate some of the most common audio formats. This is intended to be informational-only.

**Table 34-1. Serial Format Configuration**

| PARAMETER NAME | PARAMETER FIELD | DESCRIPTION |
|---|---|---|
| Data Delay | FMTCFG.DATADLY | I2S_BCLK period between first I2S_WCLK edge and MSB of first audio channel data transferred |
| Word length | FMTCFG.WORDLEN | Number of bits per sample |
| Sample Edge | FMTCFG.SMPLEDGE | The edge at which input data is sampled and output data is clocked out |
| Phase Format | FMTCFG.DUALPHASE | Single phase is used for TDM formats and dual phase is used for I2S formats |
| Word Clock Inversion | WCLKSRC.WCLKINV | Mostly applicable in dual phase format to indicate the left or right channel |
| Word Clock Division Ratio | CLKCTL.WCLKPHASE | Controls how WCLKDIV is used for generating I2S_WCLK |
| Word Clock High Time | WCLKDIV.WDIV | I2S_WCLK high and low time based on CLKCTL.WCLKPHASE settings |
| Word Clock Low Time | WCLKDIV.WDIV | |

#### 34.2.6.1 I2S Format

I2S is a dual-phase format with a 50% I2S_WCLK duty cycle with two channels as shown in Figure 34-10 with the configuration example shown in Table 34-2. The left channel is transferred when I2S_WCLK phase is low followed by the right channel when I2S_WCLK phase is high. The data is delayed by one I2S_BCLK period, I2S_ADx is sampled on the rising edge and launched on the falling edge of the I2S_BCLK.

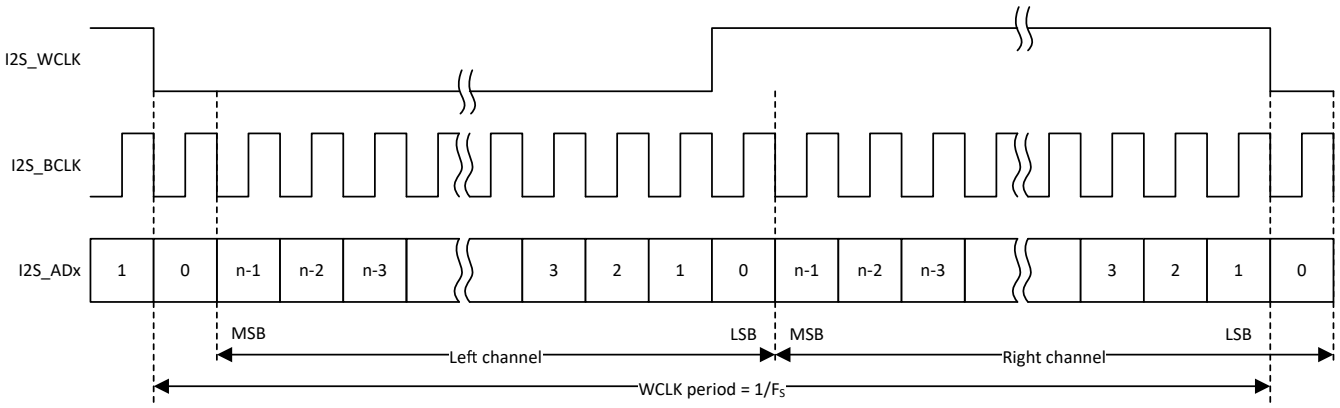



**Figure 34-10. I2S Frame Format**

**Table 34-2. I2S Format Configuration**

| PARAMETER FIELD | VALUE |
|---|---|
| FMTCFG.DATADLY | 1 |
| FMTCFG.WORDLEN | Number of bits per sample in the left or right channel |



Copyright © 2026 Texas Instruments Incorporated

**Table 34-2. I2S Format Configuration (continued)**

| PARAMETER FIELD | VALUE |
|---|---|
| FMTCFG.SMPLEDGE | 1 |
| FMTCFG.DUALPHASE | 1 |
| WCLKSRC.WCLKINV | 1 |
| CLKCTL.WCLKPHASE | 1 |
| WCLKDIV.WDIV[15:10] | Do not program |
| WCLKDIV.WDIV[9:0] | ≥ Number of bits per sample x 2 channels |

#### 34.2.6.2 Right Justified Format

The right-justified format (RJF) is a dual-phase format with a 50% I2S_WCLK duty cycle and two channels as shown in Figure 34-11 with the configuration example shown in Table 34-3. The left channel is transferred when I2S_WCLK phase is high followed by the right channel when I2S_WCLK phase is low.

The data delay (FMTCFG.DATADLY) must be equal to the number of I2S_BCLK periods per I2S_WCLK phase minus the word length. This allows the data to be transmitted or received with the LSB aligned to the end of the phase.

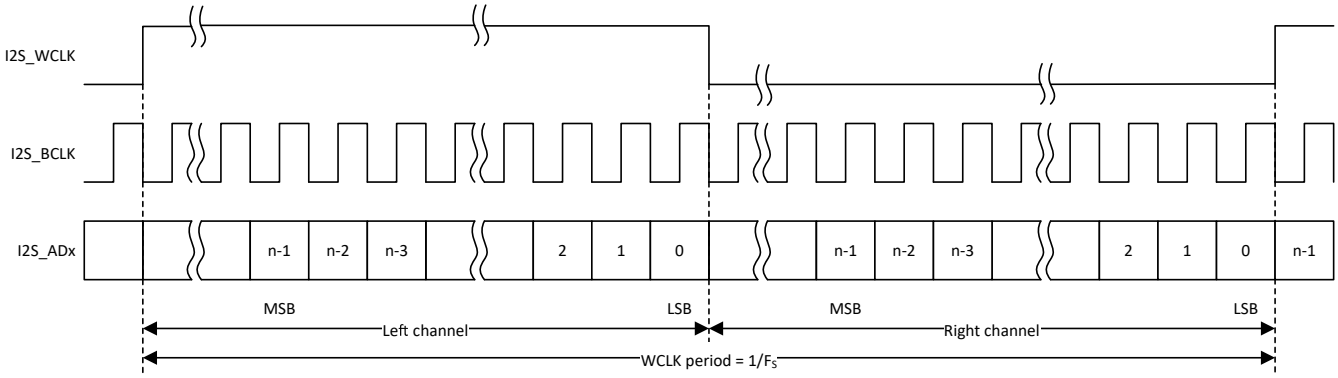



**Figure 34-11. Right Justified Format**

**Table 34-3. Right Justified Format Configuration**

| PARAMETER FIELD | VALUE |
|---|---|
| FMTCFG.DATADLY | WCLKDIV.WDIV[9:0] - FMTCFG.WORDLEN |
| FMTCFG.WORDLEN | Number of bits per sample in the left or right channel |
| FMTCFG.SMPLEDGE | 1 |
| FMTCFG.DUALPHASE | 1 |
| WCLKSRC.WCLKINV | 0 |
| CLKCTL.WCLKPHASE | 1 |
| WCLKDIV.WDIV[15:10] | Do not program |
| WCLKDIV.WDIV[9:0] | ≥ Number of bits per sample x 2 channels |

#### 34.2.6.3 Left Justified Format

The left-justified format (LJF) is a dual-phase format with a 50% I2S_WCLK duty cycle and two channels as shown in Figure 34-12 with the configuration examle shown in Table 34-4. The left channel is transferred when I2S_WCLK phase is high followed by the right channel when I2S_WCLK phase is low.



Copyright © 2026 Texas Instruments Incorporated

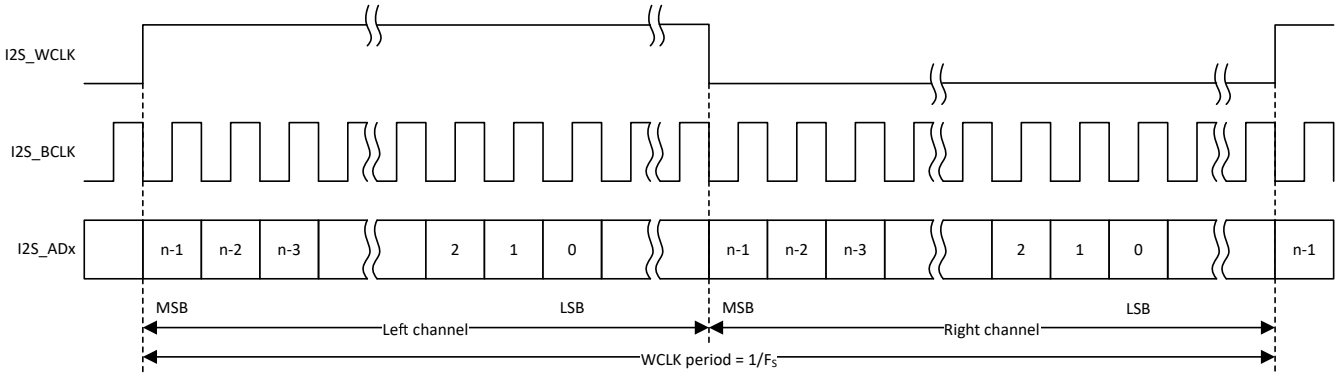



**Figure 34-12. Left Justified Format**

**Table 34-4. Left Justified Format Configuration**

| PARAMETER FIELD | VALUE |
|---|---|
| FMTCFG.DATADLY | 0 |
| FMTCFG.WORDLEN | Number of bits per sample in the left or right channel |
| FMTCFG.SMPLEDGE | 1 |
| FMTCFG.DUALPHASE | 1 |
| WCLKSRC.WCLKINV | 0 |
| CLKCTL.WCLKPHASE | 1 |
| WCLKDIV.WDIV[15:10] | Do not program |
| WCLKDIV.WDIV[9:0] | ≥ Number of bits per sample x 2 channels |

#### 34.2.6.4 DSP Format

The DSP format is a single-phase format with I2S_WCLK high for one I2S_BCLK period as shown in Figure 34-13. Sample words may are transferred one after the other and the MSB is aligned to I2S_WCLK or the subsequent I2S_BCLK, which is programmable as shown in the configuration example in Table 34-5.

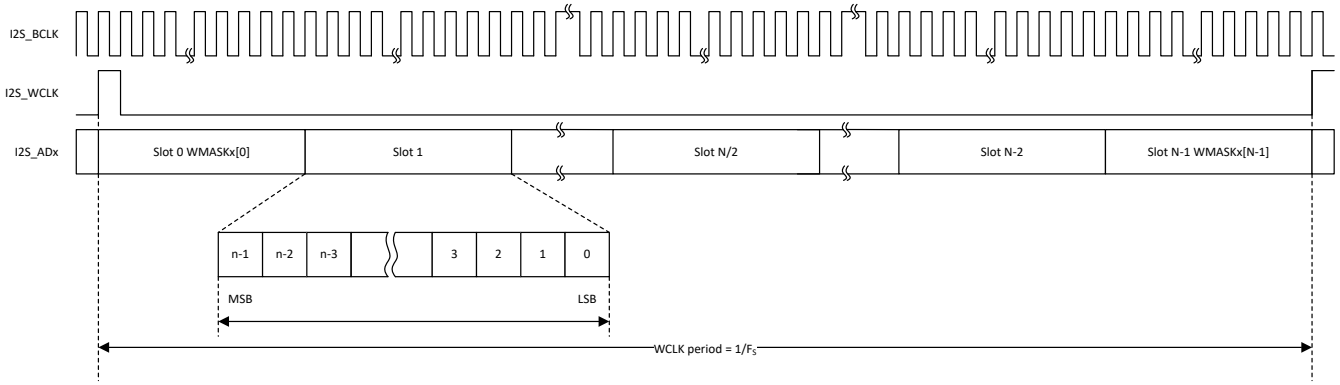



**Figure 34-13. DSP Format**

**Table 34-5. DSP Format Configuration**

| PARAMETER FIELD | VALUE |
|---|---|
| FMTCFG.DATADLY | 0 or 1 |
| FMTCFG.WORDLEN | Number of bits per sample |
| FMTCFG.SMPLEDGE | 0 |
| FMTCFG.DUALPHASE | 0 |
| WCLKSRC.WCLKINV | 0 |



Copyright © 2026 Texas Instruments Incorporated

**Table 34-5. DSP Format Configuration (continued)**

| PARAMETER FIELD | VALUE |
|---|---|
| CLKCTL.WCLKPHASE | 0 |
| WCLKDIV.WDIV[15:10] | Do not program |
| WCLKDIV.WDIV[9:0] | ≥ Number of bits per sample x Number of slots |

#### 34.2.6.5 PCM Long Frame Format

PCM long frame is a single-phase format with I2S_WCLK high for 13 I2S_BCLK period as shown in Figure 34-14. In this frame format, the I2S_WCLK is a pattern which requires setting the CLKCTL.WCLKPHASE for a custom I2S_WCLK creation by programming the WCLKDIV.WDIV accordingly as given in Table 34-6.

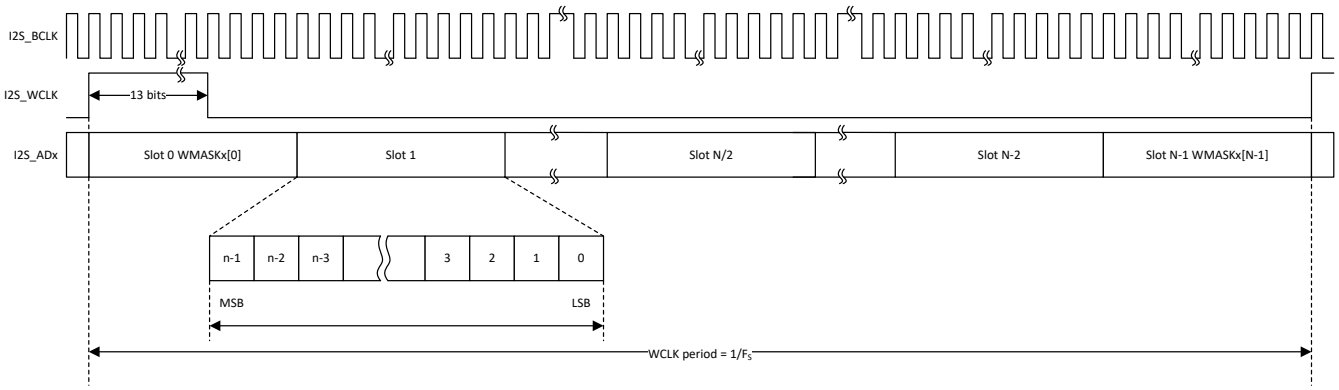



**Figure 34-14. PCM Long Frame Format**

**Table 34-6. PCM Long Frame Format Configuration**

| PARAMETER FIELD | VALUE |
|---|---|
| FMTCFG.DATADLY | 0 |
| FMTCFG.WORDLEN | Number of bits per sample |
| FMTCFG.SMPLEDGE | 0 |
| FMTCFG.DUALPHASE | 0 |
| WCLKSRC.WCLKINV | 0 |
| CLKCTL.WCLKPHASE | 2 |
| WCLKDIV.WDIV[15:8] | 13 - (Number of bits per samples x Number of slots) |
| WCLKDIV.WDIV[7:0] | 13 |

#### 34.2.6.6 TDM Classic Format

TDM frame format is a single-phase format and has multiple variations with I2S_WCLK high and low. One of the format is shown in Figure 34-15, where the I2S_WCLK is a 50% duty cycle over the entire frame rate. This can be achieved by setting the CLKCTL.WCLKPHASE for a custom I2S_WCLK creation and programming the WCLKDIV.WDIV accordingly as given in Table 34-7.



Copyright © 2026 Texas Instruments Incorporated

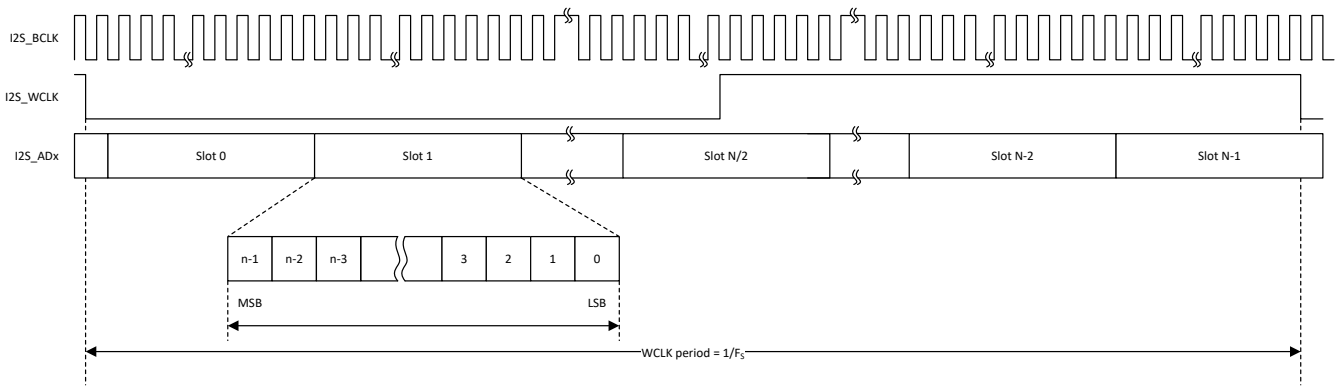



**Figure 34-15. TDM Classic Format**

**Table 34-7. TDM Classic Format Configuration**

| PARAMETER FIELD | VALUE |
|---|---|
| FMTCFG.DATADLY | 1 |
| FMTCFG.WORDLEN | Number of bits per sample |
| FMTCFG.SMPLEDGE | 1 |
| FMTCFG.DUALPHASE | 0 |
| WCLKSRC.WCLKINV | 1 |
| CLKCTL.WCLKPHASE | 2 |
| WCLKDIV.WDIV[15:8] | (Number of bits per sample x Number of slots) ÷ 2 |
| WCLKDIV.WDIV[7:0] | (Number of bits per sample x Number of slots) ÷ 2 |

### *34.2.7 Initialization*

Before the I2S module is setup or configuration changes, the FMTCFG.ENABLE bit must be cleared to avoid unpredictable behavior during the updates or for first data receive or transmitted afterward.

The following steps are common to both controller and target configuration:

1. Configure the I2S pins in the pin's IOMUX register for the I2S_BCLK, I2S_WCLK, I2S_MCLK (optional), I2S_AD0 and I2S_AD1 pin functions.
2. Reset the peripheral using I2Sx.RSTCTL register.
3. Enable the power to I2S peripheral using the I2Sx.PWREN register.

#### Controller mode configuration

1. Select the I2S functional clock source as internal clock by writing I2Sx.WCLKSRC as "2h"
2. Select the I2S_MCLK (optional), I2S_BCLK and I2S_WCLK dividers using the I2Sx.MCLKDIV (optional), I2Sx.BCLKDIV and I2Sx.WCLKDIV, respectively.
3. Setup the configuration for frame format. Example configurations can be found in Section 34.2.6.
4. Setup the configuration for the channel configuration. Example channel configurations can be found in Section 34.2.5.
5. Setup the FIFO thresholds for interrupt or DMA triggers in the I2S_IFLS register.
6. When in the controller mode, enable the frame clock generation by setting I2Sx.CLKCTL.WBEN and I2Sx.CLKCTL.MEN (optional) bit
7. Enable the peripheral operation by setting I2Sx.FMTCFG.ENABLE as '1'.

#### Target mode configuration

1. Select the I2S functional clock source as pad by writing I2Sx.WCLKSRC as "1h"
2. Setup the configuration for frame format. Example configurations can be found in Section 34.2.6.



Copyright © 2026 Texas Instruments Incorporated

3. Setup the configuration for the channel configuration. Example channel configurations can be found in Section 34.2.5.
4. Setup the FIFO thresholds for interrupt or DMA triggers in the I2S_IFLS register.
5. Enable the peripheral operation by setting I2Sx.FMTCFG.ENABLE as '1'.

### 34.2.8 Disabling I2S

I2S is a continuously streaming protocol, and it may be necessary when going into a low power mode like STOP, STANDBY or SHUTDOWN, that the peripheral is disabled. Since there is no IDLE status bit, because there is no such condtition on the bus, the host application must ensure the TXFIFO and RXFIFO are flushed when disabling the peripheral.

Additionally, as the I2S module may be in between frames, it is recommended that the peripheral be reset and an initialization be done when restarting the peripheral to ensure external devices can recognize valid I2S_WCLK frame synchronization boundary.

**Note**

When disabling the I2S_BCLK and I2S_WCLK the host must wait for 4 I2S_BCLK period before enabling it again during a transition.

Remember to write to the PWREN bit to enable writing to the peripherals registers.

### 34.2.9 Interrupts and Events Support

The I2S module contains three event publishers and no event subscribers. One event publisher (CPU_INT) manages I2S interrupt requests (IRQs) to the CPU subsystem through a static event route. The second and third event publishers (DMA_TRIG_RX, DMA_TRIG_TX) are used to setup the trigger signaling for the DMA through DMA event route.

The I2S events are summarized in Table 34-8.

**Table 34-8. I2S Events**

| Event | Type | Source | Destination | Route | Confiuguration | Functionality |
|---|---|---|---|---|---|---|
| CPU interrupt | Publisher | I2S | CPU Subsystem | Static route | CPU_INT registers | Fixed interrupt route from I2S to CPU |
| DMA trigger | Publisher | I2S | DMA | DMA event route | DMA_TRIG_RX registers | Fixed interrupt route from I2S to DMA |
| DMA trigger | Publisher | I2S | DMA | DMA event route | DMA_TRIG_TX registers | Fixed interrupt route from I2S to DMA |

#### 34.2.9.1 CPU Interrupt Event Publisher (CPU_INT)

The I2S module provides 8 interrupt sources which can be configured to source a CPU interrupt event. In order of decreasing interrupt priority, the CPU interrupt events from the I2S are:

**Table 34-9. I2S CPU Interrupt Event Conditions (CPU_INT)**

| IIDX STAT | Name | Description |
|---|---|---|
| 0x01 | WCLKERR | Error interrupt when an unexpected I2Sx_WCLK edge occurs during the data delay period of a phase |
| 0x02 | RXIFG | I2S receive interrupt flag. |
| 0x03 | TXIFG | I2S transmit interrupt flag. |
| 0x04 | RXFIFO_OFV_EVT | RXFIFO overflow interrupt |
| 0x05 | TXFIFO_UNF_EVT | TXFIFO underflow interrupt |
| 0x10 | DMA_DONE_RX | This interrupt is set if the RX DMA channel sends the DONE signal. |



Copyright © 2026 Texas Instruments Incorporated

**Table 34-9. I2S CPU Interrupt Event Conditions (CPU_INT) (continued)**

| IIDX STAT | Name | Description |
|---|---|---|
| 0x11 | DMA_DONE_TX | This interrupt is set if the TX DMA channel sends the DONE signal. |

The CPU interrupt event configuration is managed with the CPU_INT event management registers. See Section 7.2.5 for guidance on configuring the Event registers for CPU interrupts.

The receive interrupt (RXIFG, 0x02) changes state when one of the following events occurs:

- Receive interrupt is set when receive FIFO is greater than or equal to the programmed trigger level.
- Receive interrupt is cleared by reading data from the receive FIFO until it falls below the trigger threshold, by reading interrupt index (IIDX read) or by writing a 1 to the receive interrupt bit in ICLR.

Receive FIFO should be read till the number of entries in the receive FIFO falls below the programmed threshold in IFLS, to ensure that further interrupts are generated correctly.

The transmit interrupt (TXIFG, 0x03) changes state when one of the following events occurs:

- Transmit interrupt is set when transmit FIFO is less than or equal to programmed trigger level.
- Transmit interrupt is cleared by writing data to the transmit FIFO until it becomes greater than the trigger threshold, by reading the interrupt undex (IIDX read) or by writing a 1 to the transmit interrupt bit in ICLR.

Transmit FIFO should be filled till the number of entries in the FIFO crosses the programmed threshold in IFLS, to ensure that further interrupts are generated correctly.

#### 34.2.9.2 DMA Trigger Publisher (DMA_TRIG_RX, DMA_TRIG_TX)

DMA_TRIG_RX and DMA)TRIG_TX registers are used to setup the trigger signaling for the DMA. This can be setup in a flexible way to trigger the DMA for recieve or transmit events with the trigger conditions.

DMA_TRIG_RX is used for triggering the DMA to do a receive data transfer and DMA_TRIG_TX is used for triggering the DMA to do a transmit data transfer. The DMA trigger event configuration is managed with the DMA_TRIG_RX and DMA_TRIG_TX event management registers. Sec Section 7.2.5 for guidance on configuring the Event registers and Section 7.1.3.2 for on how the DMA trigger event works.

### *34.2.10 Emulation Modes*

The module behavior while the device is in debug mode is controlled by the FREE and SOFT bits in PDBGCTL register. When the device is in debug mode and set into halt mode below behavior can be configured.

**Table 34-10. Debug Mode Peripheral Behavior**

| PDBGCTL.FREE | PDBGCTL.SOFT | Function |
|---|---|---|
| 1 | X | Module continues operation |
| 0 | 0 | Reserved. Behavior is unpredictable |
| 0 | 1 | Module stops after the next transfer has finished |



Copyright © 2026 Texas Instruments Incorporated

## 34.3 I2S Registers

Table 34-11 lists the memory-mapped registers for the I2S registers. All register offset addresses not listed in Table 34-11 should be considered as reserved locations and the register contents should not be modified.

**Table 34-11. I2S Registers**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 800h | PWREN | Power enable | | Go |
| 804h | RSTCTL | Reset Control | | Go |
| 808h | CLKCFG | Peripheral Clock Configuration Register | | Go |
| 814h | STAT | Status Register | | Go |
| 1008h | PDBGCTL | Peripheral Debug Control | | Go |
| 1020h | IIDX | Interrupt index | | Go |
| 1028h | IMASK | Interrupt Mask Register Selects mask states of the flags in [IRQFLAGS.*] that contribute to the I2S_IRQ event. | | Go |
| 1030h | RIS | This registers gives the raw interrupt status | | Go |
| 1038h | MIS | This registers gives the raw interrupt status | | Go |
| 1040h | ISET | Interrupt Set Register.<br>This register can be used by software for diagnostics and safety checking purposes. | | Go |
| 1048h | ICLR | Interrupt clear register.<br>This register allows software to clear interrupts. | | Go |
| 1058h | IMASK | Interrupt mask | DMA_TRIG_RX | Go |
| 1060h | RIS | Raw interrupt status | DMA_TRIG_RX | Go |
| 1068h | MIS | Masked interrupt status | DMA_TRIG_RX | Go |
| 1070h | ISET | Interrupt set | DMA_TRIG_RX | Go |
| 1088h | IMASK | Interrupt mask | DMA_TRIG_TX | Go |
| 1090h | RIS | Raw interrupt status | DMA_TRIG_TX | Go |
| 1098h | MIS | Masked interrupt status | DMA_TRIG_TX | Go |
| 10A0h | ISET | Interrupt set | DMA_TRIG_TX | Go |
| 10E4h | INTCTL | Interrupt control register | | Go |
| 1100h | FMTCFG | This register configures the serial interface format | | Go |
| 1104h | CLKCTL | This register controls internal audio clock | | Go |
| 1108h | STAT | Status Register | | Go |
| 110Ch | IFLS | Interrupt FIFO Level Select Register | | Go |
| 1110h | WCLKSRC | This register configures the WCLK Source | | Go |
| 1118h | DIRCFG | This register configures the direction of data pins(AD0/AD1) | | Go |
| 1120h | TXDATA | Transmit Data Register | | Go |



Copyright © 2026 Texas Instruments Incorporated

**Table 34-11. I2S Registers (continued)**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 1124h | RXDATA | Receive Data Register | | Go |
| 1148h | WMASK0 | This register configures the word selection mask for data pin 0(AD0) | | Go |
| 114Ch | WMASK1 | This register configures the word selection mask for data pin 1(AD1) | | Go |
| 1160h | MCLKDIV | This field configures MCLK division ratio | | Go |
| 1164h | WCLKDIV | Configures WCLK division ratio | | Go |
| 1168h | BCLKDIV | This field configures BCLK division ratio | | Go |

Complex bit access types are encoded to fit into small table cells. Table 34-12 shows the codes that are used for access types in this section.

**Table 34-12. I2S Access Type Codes**

| Access Type | Code | Description |
|---|---|---|
| Read Type | | |
| R | R | Read |
| Write Type | | |
| W | W | Write |
| WK | W<br>K | Write<br>Write protected by a key |
| Reset or Default Value | | |
| *-n* | | Value after reset or the default value |



Copyright © 2026 Texas Instruments Incorporated

### 34.3.1 PWREN (Offset = 800h) [Reset = 00000000h]

PWREN is shown in Figure 34-16 and described in Table 34-13.

Return to the Summary Table.

Register to control the power state

**Figure 34-16. PWREN**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | ENABLE |
| R-0h | | | | | | | R/WK-0h |

**Table 34-13. PWREN Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | KEY to allow Power State Change<br>26h = KEY to allow write access to this register |
| 23-1 | RESERVED | R | 0h | |
| 0 | ENABLE | R/WK | 0h | Enable the power<br>KEY must be set to 26h to write to this bit.<br>0h = Disable Power<br>1h = Enable Power |



Copyright © 2026 Texas Instruments Incorporated

### 34.3.2 RSTCTL (Offset = 804h) [Reset = 00000000h]

RSTCTL is shown in Figure 34-17 and described in Table 34-14.

Return to the Summary Table.

Register to control reset assertion and de-assertion

**Figure 34-17. RSTCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | RESETSTKYCLR | RESETASSERT |
| R-0h | | | | | | WK-0h | WK-0h |

**Table 34-14. RSTCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | Unlock key<br>B1h = KEY to allow write access to this register |
| 23-2 | RESERVED | R | 0h | |
| 1 | RESETSTKYCLR | WK | 0h | Clear the RESETSTKY bit in the STAT register<br>KEY must be set to B1h to write to this bit.<br>0h = Writing 0 has no effect<br>1h = Clear reset sticky bit |
| 0 | RESETASSERT | WK | 0h | Assert reset to the peripheral<br>KEY must be set to B1h to write to this bit.<br>0h = Writing 0 has no effect<br>1h = Assert reset |



Copyright © 2026 Texas Instruments Incorporated

### 34.3.3 CLKCFG (Offset = 808h) [Reset = 00000000h]

CLKCFG is shown in Figure 34-18 and described in Table 34-15.

Return to the Summary Table.

Peripheral Clock Configuration Register

**Figure 34-18. CLKCFG**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| KEY | | | | | | | | RESERVED | | | | | | | |
| W-0h | | | | | | | | R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | | | DAICLK | |
| R-0h | | | | | | | | | | | | | | R/W-0h | |

**Table 34-15. CLKCFG Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | KEY to Allow State Change -- 0xA9<br>A9h = key value to allow change field of GPRCM |
| 23-2 | RESERVED | R | 0h | |
| 1-0 | DAICLK | R/W | 0h | Audio Clock<br>0h = SYSOSC<br>1h = HF crystal<br>2h = PLL<br>3h = Reserved |



Copyright © 2026 Texas Instruments Incorporated

### 34.3.4 STAT (Offset = 814h) [Reset = 00000000h]

STAT is shown in Figure 34-19 and described in Table 34-16.

Return to the Summary Table.

peripheral enable and reset status

**Figure 34-19. STAT**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | RESETSTKY |
| R-0h | | | | | | | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |

**Table 34-16. STAT Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-17 | RESERVED | R | 0h | |
| 16 | RESETSTKY | R | 0h | This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register<br>0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register<br>1h = The peripheral was reset since the last bit clear |
| 15-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 34.3.5 PDBGCTL (Offset = 1008h) [Reset = 00000003h]

PDBGCTL is shown in Figure 34-20 and described in Table 34-17.

Return to the Summary Table.

This register can be used by the software developer to control the behavior of the peripheral relative to the 'Core Halted' input

**Figure 34-20. PDBGCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | SOFT | FREE |
| R-0h | | | | | | R/W-1h | R/W-1h |

**Table 34-17. PDBGCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | |
| 1 | SOFT | R/W | 1h | Soft halt boundary control. This function is only available, if FREE is set to 'STOP'<br>0h = The peripheral will halt immediately, even if the resultant state will result in corruption if the system is restarted<br>1h = The peripheral blocks the debug freeze until it has reached a boundary where it can resume without corruption |
| 0 | FREE | R/W | 1h | Free run control<br>0h = The peripheral freezes functionality while the Core Halted input is asserted and resumes when it is de-asserted.<br>1h = The peripheral ignores the state of the Core Halted input |



Copyright © 2026 Texas Instruments Incorporated

### 34.3.6 IIDX (Offset = 1020h) [Reset = 00000000h]

IIDX is shown in Figure 34-21 and described in Table 34-18.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. Value 0x00 means no event pending. Interrupt 1 is the highest priority, IIDX next highest, 4, 8, … IIDX^31 is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred. On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in [RIS] and [MIS] are cleared as well. After a read from the CPU (not from the debug interface), the register is updated with the next highest priority interrupt, if none are pending, then it should display 0x0.

**Figure 34-21. IIDX**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | STAT | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | |

**Table 34-18. IIDX Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | STAT | R | 0h | Module Interrupt Vector Value. This register provides the highest priority interrupt index. A read clears the corresponding interrupt flag in RIS and MIS registers. 15h-1Fh = Reserved<br>00h = No interrupt pending<br>01h = WCLK Error<br>2h = Receive interrupt; Interrupt Flag: RX<br>3h = Transmit interrupt; Interrupt Flag: TX<br>4h = RX FIFO Overflow Event/interrupt pending<br>5h = TX FIFO underflow interrupt<br>10h = DMA DONE on RX<br>11h = DMA DONE on TX<br>13h = DMA PRE IRQ INTERRUPT<br>14h = DMA PRE IRQ INTERRUPT |



Copyright © 2026 Texas Instruments Incorporated

### 34.3.7 IMASK (Offset = 1028h) [Reset = 00000000h]

IMASK is shown in Figure 34-22 and described in Table 34-19.

Return to the Summary Table.

Interrupt Mask Register Selects mask states of the flags in [IRQFLAGS.*] that contribute to the I2S_IRQ event.

**Figure 34-22. IMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | RESERVED | RESERVED | RESERVED | DMA_DONE_TX |
| R-0h | | | | R-0h | R-0h | R-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DMA_DONE_RX | RESERVED | | | | | | |
| R/W-0h | R-0h | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | TXFIFO_UNF | RXFIFO_OVF | TXINT | RXINT | WCLKERR |
| R-0h | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

**Table 34-19. IMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | RESERVED | R | 0h | |
| 19 | RESERVED | R | 0h | Reserved |
| 18 | RESERVED | R | 0h | Reserved |
| 17 | RESERVED | R | 0h | |
| 16 | DMA_DONE_TX | R/W | 0h | Enable DMA Done on TX Event Channel Interrupt<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 15 | DMA_DONE_RX | R/W | 0h | Enable DMA Done on RX Event Channel Interrupt<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 14-5 | RESERVED | R | 0h | |
| 4 | TXFIFO_UNF | R/W | 0h | TX FIFO underflow interrupt mask<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 3 | RXFIFO_OVF | R/W | 0h | RXFIFO overflow event mask.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 2 | TXINT | R/W | 0h | Enable Transmit Interrupt.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 1 | RXINT | R/W | 0h | Enable Receive Interrupt.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 0 | WCLKERR | R/W | 0h | WCLKERR interrupt mask<br>0h = Disable the interrupt mask<br>1h = Enable the interrupt mask |



Copyright © 2026 Texas Instruments Incorporated

### 34.3.8 RIS (Offset = 1030h) [Reset = 00000000h]

RIS is shown in Figure 34-23 and described in Table 34-20.

Return to the Summary Table.

This registers gives the raw interrupt status

**Figure 34-23. RIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | RESERVED | RESERVED | RESERVED | DMA_DONE_TX |
| R-0h | | | | R-0h | R-0h | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DMA_DONE_RX | RESERVED | | | | | | |
| R-0h | R-0h | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | TXFIFO_UNF | RXFIFO_OVF | TXINT | RXINT | WCLKERR |
| R-0h | | | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 34-20. RIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | RESERVED | R | 0h | |
| 19 | RESERVED | R | 0h | Reserved |
| 18 | RESERVED | R | 0h | Reserved |
| 17 | RESERVED | R | 0h | |
| 16 | DMA_DONE_TX | R | 0h | DMA Done on TX Event Channel Interrupt<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 15 | DMA_DONE_RX | R | 0h | DMA Done on RX Event Channel Interrupt<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 14-5 | RESERVED | R | 0h | |
| 4 | TXFIFO_UNF | R | 0h | TX FIFO Underflow Interrupt<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 3 | RXFIFO_OVF | R | 0h | RXFIFO overflow event. This interrupt is set if an RX FIFO overflow has been detected.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 2 | TXINT | R | 0h | Transmit Interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 1 | RXINT | R | 0h | Receive Interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

**Table 34-20. RIS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 0 | WCLKERR | R | 0h | This interrupt is set when: - An unexpected WCLK edge occurs during the data delay period of a phase. Note unexpected WCLK edges during the word and idle periods of the phase are not detected. - In dual-phase mode, when two WCLK edges are less than 4 BCLK cycles apart. - In single-phase mode, when a WCLK pulse occurs before the last channel. This error requires a complete restart since word synchronization has been lost.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

### 34.3.9 MIS (Offset = 1038h) [Reset = 00000000h]

MIS is shown in Figure 34-24 and described in Table 34-21.

Return to the Summary Table.

This registers gives the raw interrupt status

**Figure 34-24. MIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | RESERVED | RESERVED | RESERVED | DMA_DONE_TX |
| R-0h | | | | R-0h | R-0h | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DMA_DONE_RX | RESERVED | | | | | | |
| R-0h | R-0h | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | TXFIFO_UNF | RXFIFO_OVF | TXINT | RXINT | WCLKERR |
| R-0h | | | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 34-21. MIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | RESERVED | R | 0h | |
| 19 | RESERVED | R | 0h | Reserved |
| 18 | RESERVED | R | 0h | Reserved |
| 17 | RESERVED | R | 0h | |
| 16 | DMA_DONE_TX | R | 0h | Masked DMA Done on TX Event Channel Interrupt<br>0h = Interrupt did not occur or mask was disabled.<br>1h = Interrupt occurred |
| 15 | DMA_DONE_RX | R | 0h | Masked DMA Done on RX Event Channel Interrupt<br>0h = Interrupt did not occur or mask was disabled.<br>1h = Interrupt occurred |
| 14-5 | RESERVED | R | 0h | |
| 4 | TXFIFO_UNF | R | 0h | TX FIFO underflow interrupt<br>0h = Interrupt did not occur or mask was disabled.<br>1h = Interrupt occurred |
| 3 | RXFIFO_OVF | R | 0h | Masked RXFIFO overflow event. This interrupt is set if an RX FIFO overflow has been detected.<br>0h = Interrupt did not occur or mask was disabled.<br>1h = Interrupt occurred |
| 2 | TXINT | R | 0h | Masked Transmit Interrupt.<br>0h = Interrupt did not occur or mask was disabled.<br>1h = Interrupt occurred |
| 1 | RXINT | R | 0h | Masked Receive Interrupt.<br>0h = Interrupt did not occur or mask was disabled.<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

**Table 34-21. MIS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 0 | WCLKERR | R | 0h | This interrupt is set when: - An unexpected WCLK edge occurs during the data delay period of a phase. Note unexpected WCLK edges during the word and idle periods of the phase are not detected. - In dual-phase mode, when two WCLK edges are less than 4 BCLK cycles apart. - In single-phase mode, when a WCLK pulse occurs before the last channel. This error requires a complete restart since word synchronization has been lost.<br>0h = Interrupt did not occur or mask was disabled.<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

### 34.3.10 ISET (Offset = 1040h) [Reset = 00000000h]

ISET is shown in Figure 34-25 and described in Table 34-22.

Return to the Summary Table.

Interrupt Set Register. This register can be used by software for diagnostics and safety checking purposes.

**Figure 34-25. ISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | RESERVED | RESERVED | RESERVED | DMA_DONE_TX |
| R-0h | | | | R-0h | R-0h | R-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DMA_DONE_RX | RESERVED | | | | | | |
| W-0h | R-0h | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | TXFIFO_UNF | RXFIFO_OVF | TXINT | RXINT | WCLKERR |
| R-0h | | | W-0h | W-0h | W-0h | W-0h | W-0h |

**Table 34-22. ISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | RESERVED | R | 0h | |
| 19 | RESERVED | R | 0h | Reserved |
| 18 | RESERVED | R | 0h | Reserved |
| 17 | RESERVED | R | 0h | |
| 16 | DMA_DONE_TX | W | 0h | Set DMA Done on TX Event Channel Interrupt<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 15 | DMA_DONE_RX | W | 0h | Set DMA Done on RX Event Channel Interrupt<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 14-5 | RESERVED | R | 0h | |
| 4 | TXFIFO_UNF | W | 0h | Set TX FIFO Underflow Event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 3 | RXFIFO_OVF | W | 0h | Set RXFIFO overflow event.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 2 | TXINT | W | 0h | Set Transmit Interrupt.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 1 | RXINT | W | 0h | Set Receive Interrupt.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 0 | WCLKERR | W | 0h | This field sets the interrupt WCLKERR<br>0h = Writing 0 has no effect<br>1h = Set interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 34.3.11 ICLR (Offset = 1048h) [Reset = 00000000h]

ICLR is shown in Figure 34-26 and described in Table 34-23.

Return to the Summary Table.

Interrupt clear register. This register allows software to clear interrupts.

**Figure 34-26. ICLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | RESERVED | RESERVED | RESERVED | DMA_DONE_TX |
| R-0h | | | | R-0h | R-0h | R-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DMA_DONE_RX | RESERVED | | | | | | |
| W-0h | R-0h | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | TXFIFO_UNF | RXFIFO_OVF | TXINT | RXINT | WCLKERR |
| R-0h | | | W-0h | W-0h | W-0h | W-0h | W-0h |

**Table 34-23. ICLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | RESERVED | R | 0h | |
| 19 | RESERVED | R | 0h | Reserved |
| 18 | RESERVED | R | 0h | Reserved |
| 17 | RESERVED | R | 0h | |
| 16 | DMA_DONE_TX | W | 0h | Clear DMA Done on TX Event Channel Interrupt<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 15 | DMA_DONE_RX | W | 0h | Clear DMA Done on RX Event Channel Interrupt<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 14-5 | RESERVED | R | 0h | |
| 4 | TXFIFO_UNF | W | 0h | Clear TXFIFO underflow event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 3 | RXFIFO_OVF | W | 0h | Clear RXFIFO overflow event.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 2 | TXINT | W | 0h | Clear Transmit Interrupt.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 1 | RXINT | W | 0h | Clear Receive Interrupt.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 0 | WCLKERR | W | 0h | This field clears the interrupt WCLKERR<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 34.3.12 IMASK (Offset = 1058h) [Reset = 00000000h]

IMASK is shown in Figure 34-27 and described in Table 34-24.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

**Figure 34-27. IMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | RXINT | RESERVED |
| R-0h | | | | | | R/W-0h | R-0h |

**Table 34-24. IMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | |
| 1 | RXINT | R/W | 0h | Enable Receive Interrupt.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 34.3.13 RIS (Offset = 1060h) [Reset = 00000000h]

RIS is shown in Figure 34-28 and described in Table 34-25.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Figure 34-28. RIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | RXINT | RESERVED |
| R-0h | | | | | | R-0h | R-0h |

**Table 34-25. RIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | |
| 1 | RXINT | R | 0h | Receive Interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 34.3.14 MIS (Offset = 1068h) [Reset = 00000000h]

MIS is shown in Figure 34-29 and described in Table 34-26.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 34-29. MIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | RXINT | RESERVED |
| R-0h | | | | | | R-0h | R-0h |

**Table 34-26. MIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | |
| 1 | RXINT | R | 0h | Masked Receive Interrupt.<br>0h = Interrupt did not occur or mask was disabled.<br>1h = Interrupt occurred |
| 0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 34.3.15 ISET (Offset = 1070h) [Reset = 00000000h]

ISET is shown in Figure 34-30 and described in Table 34-27.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 34-30. ISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | RXINT | RESERVED |
| R-0h | | | | | | W-0h | R-0h |

**Table 34-27. ISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | |
| 1 | RXINT | W | 0h | Set Receive Interrupt.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 34.3.16 IMASK (Offset = 1088h) [Reset = 00000000h]

IMASK is shown in Figure 34-31 and described in Table 34-28.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

**Figure 34-31. IMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | TXINT | RESERVED | |
| R-0h | | | | | R/W-0h | R-0h | |

**Table 34-28. IMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | RESERVED | R | 0h | |
| 2 | TXINT | R/W | 0h | Enable Transmit Interrupt.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 1-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 34.3.17 RIS (Offset = 1090h) [Reset = 00000000h]

RIS is shown in Figure 34-32 and described in Table 34-29.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Figure 34-32. RIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | TXINT | RESERVED | |
| R-0h | | | | | R-0h | R-0h | |

**Table 34-29. RIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | RESERVED | R | 0h | |
| 2 | TXINT | R | 0h | Transmit Interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 1-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 34.3.18 MIS (Offset = 1098h) [Reset = 00000000h]

MIS is shown in Figure 34-33 and described in Table 34-30.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 34-33. MIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | TXINT | RESERVED | |
| R-0h | | | | | R-0h | R-0h | |

**Table 34-30. MIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | RESERVED | R | 0h | |
| 2 | TXINT | R | 0h | Masked Transmit Interrupt.<br>0h = Interrupt did not occur or mask was disabled.<br>1h = Interrupt occurred |
| 1-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 34.3.19 ISET (Offset = 10A0h) [Reset = 00000000h]

ISET is shown in Figure 34-34 and described in Table 34-31.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 34-34. ISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | TXINT | RESERVED | |
| R-0h | | | | | W-0h | R-0h | |

**Table 34-31. ISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | RESERVED | R | 0h | |
| 2 | TXINT | W | 0h | Set Transmit Interrupt.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 1-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 34.3.20 INTCTL (Offset = 10E4h) [Reset = 00000000h]

INTCTL is shown in Figure 34-35 and described in Table 34-32.

Return to the Summary Table.

Interrupt control register

**Figure 34-35. INTCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | INTEVAL |
| R-0h | | | | | | | W-0h |

**Table 34-32. INTCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | INTEVAL | W | 0h | Writing a 1 to this field re-evaluates the interrupt sources.<br>0h = The interrupt or event line is disabled.<br>1h = The interrupt or event line is in software mode. Software must clear the RIS. |



Copyright © 2026 Texas Instruments Incorporated

### 34.3.21 FMTCFG (Offset = 1100h) [Reset = 00000170h]

FMTCFG is shown in Figure 34-36 and described in Table 34-33.

Return to the Summary Table.

This register configures the serial interface format

**Figure 34-36. FMTCFG**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| ENABLE | RESERVED | | | | | | SUSPEND |
| R/W-0h | R-0h | | | | | | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | EMPTYSLOTOUTPUT |
| R-0h | | | | | | | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DATADLY | | | | | | | |
| R/W-1h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MEMLEN32 | SMPLEDGE | DUALPHASE | WORDLEN | | | | |
| R/W-0h | R/W-1h | R/W-1h | R/W-10h | | | | |

**Table 34-33. FMTCFG Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | ENABLE | R/W | 0h | I2S Module Enable. If the I2S is disabled in the middle of transmission or reception, it completes the current character before stopping. If ENABLE bit is not set, all registers can still be accessed and updated. It is recommended to setup and change I2S operation mode while ENABLE bit is cleared to avoid unpredictable behavior during setup or update. If disabled, I2S module will not send or receive any data.<br>0h = Disable Module<br>1h = Enable module |
| 30-25 | RESERVED | R | 0h | |
| 24 | SUSPEND | R/W | 0h | Suspend external communication<br>0h = Functional mode resumed<br>1h = External communication suspended |
| 23-18 | RESERVED | R | 0h | |
| 17-16 | EMPTYSLOTOUTPUT | R/W | 0h | The field configures the EMPTYSLOTOUTPUT data pin direction<br>0h = Send out zeroes in empty slots<br>1h = send out ones in empty slot<br>2h = Tristate line during empty slot<br>3h = Reserved |
| 15-8 | DATADLY | R/W | 1h | This field configures the number of BCLK periods between a WCLK edge and MSB of the first word in a phase Note: When 0, MSB of the next word will be output in the idle period between LSB of the previous word and the start of the next word. Otherwise logical 0 will be output until the data delay has expired.<br>0h = Zero BCLK periods - LJF and DSP formats<br>1h = One BCLK periods - I2S and DSP formats<br>2h = Two(Min) BCLK periods - RJF format<br>FFh = Max(255) BCLK periods - RJF format |
| 7 | MEMLEN32 | R/W | 0h | This register configures the size of each word stored to or loaded from memory<br>0h = 16BIT : 16-bit (one 16 bit access per sample)<br>1h = 32BIT : 32-bit(one 32-bit access per sample) |



Copyright © 2026 Texas Instruments Incorporated

**Table 34-33. FMTCFG Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 6 | SMPLEDGE | R/W | 1h | This field configures the sample edge/ transfer edge of data (and WCLK) on BCLK<br>0h = Data is sampled on the negative edge and clocked out on the positive edge.<br>1h = Data is sampled on the positive edge and clocked out on the negative edge. |
| 5 | DUALPHASE | R/W | 1h | This field selects between dual-phase or single-phase format<br>0h = Single-phase: DSP format<br>1h = Dual-phase: I2S, LJF and RJF formats |
| 4-0 | WORDLEN | R/W | Fh | Number of bits per word (8-32): In single-phase format, this is the exact number of bits per word. In dual-phase format, this is the maximum number of bits per word. Values below 8 and above 32 give undefined behavior. Data written to memory is always aligned to 16 or 32 bits as defined by MEMLEN32. Bit widths that differ from this alignment will either be truncated or zero padded. |



Copyright © 2026 Texas Instruments Incorporated

### 34.3.22 CLKCTL (Offset = 1104h) [Reset = 00000000h]

CLKCTL is shown in Figure 34-37 and described in Table 34-34.

Return to the Summary Table.

This register controls internal audio clock

**Figure 34-37. CLKCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | MEN | WCLKPHASE | | WBEN |
| R-0h | | | | R/W-0h | R/W-0h | | R/W-0h |

**Table 34-34. CLKCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | MEN | R/W | 0h | This field configures the MCLK generation<br>0h = Disable the generation<br>1h = Enable the generation |
| 2-1 | WCLKPHASE | R/W | 0h | The field configures how the WCLK division ratio is calculated and used to generate different duty cycles (See WDIV) |
| 0 | WBEN | R/W | 0h | This field configures WCLK/BCLK generation<br>0h = Disables the generation<br>1h = Enable the generation |



Copyright © 2026 Texas Instruments Incorporated

### 34.3.23 STAT (Offset = 1108h) [Reset = 00000000h]

STAT is shown in Figure 34-38 and described in Table 34-35.

Return to the Summary Table.

Status Register

**Figure 34-38. STAT**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TXCLR | TXFF | TXFE | RXCLR | RXFF | RXFE | RESERVED | |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | |

**Table 34-35. STAT Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7 | TXCLR | R | 0h | TX FIFO Clear Status<br>0h = FIFO is not cleared<br>1h = FIFO clear is complete |
| 6 | TXFF | R | 0h | Transmit FIFO Full The meaning of this bit depends on the state of the FEN bit in the CTL0 register.<br>0h = The transmitter is not full.<br>1h = If the FIFO is disabled (FEN is 0), the transmit holding register is full. If the FIFO is enabled (FEN is 1), the transmit FIFO is full. |
| 5 | TXFE | R | 1h | Transmit FIFO Empty The meaning of this bit depends on the state of the FEN bit in the CTL0 register.<br>0h = The transmitter has data to transmit.<br>1h = If the FIFO is disabled (FEN is 0), the transmit holding register is empty. If the FIFO is enabled (FEN is 1), the transmit FIFO is empty. |
| 4 | RXCLR | R | 0h | RX FIFO Clear Status<br>0h = FIFO is not cleared<br>1h = FIFO clear is complete |
| 3 | RXFF | R | 0h | Receive FIFO Full The meaning of this bit depends on the state of the FEN bit in the CTL0 register.<br>0h = The receiver can receive data.<br>1h = If the FIFO is disabled (FEN is 0), the receive holding register is full. If the FIFO is enabled (FEN is 1), the receive FIFO is full. |
| 2 | RXFE | R | 1h | Receive FIFO Empty The meaning of this bit depends on the state of the FEN bit in the CTL0 register.<br>0h = The receiver is not empty.<br>1h = If the FIFO is disabled (FEN is 0), the receive holding register is empty. If the FIFO is enabled (FEN is 1), the receive FIFO is empty. |
| 1-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 34.3.24 IFLS (Offset = 110Ch) [Reset = 00000022h]

IFLS is shown in Figure 34-39 and described in Table 34-36.

Return to the Summary Table.

The IFLS register is the interrupt FIFO level select register. You can use this register to define the levels at which the TX, RX and timeout interrupt flags are triggered. The interrupts are generated based on a transition through a level rather than being based on the level. That is, the interrupts are generated when the fill level progresses through the trigger level. For example, if the receive trigger level is set to the half-way mark, the interrupt is triggered when the receive FIFO is filled with two or more characters. Out of reset, the TXIFLSEL and RXIFLSEL bits are configured so that the FIFOs trigger an interrupt at the half-way mark.

**Figure 34-39. IFLS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RXCLR | RXIFLSEL | | | TXCLR | TXIFLSEL | | |
| R/W-0h | R/W-2h | | | R/W-0h | R/W-2h | | |

**Table 34-36. IFLS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7 | RXCLR | R/W | 0h | RX FIFO CLEAR. Setting this bit will clear the RX FIFO contents.<br>0h = Disable FIFO clear<br>1h = Enable FIFO Clear |
| 6-4 | RXIFLSEL | R/W | 2h | Receive Interrupt FIFO Level Select The trigger points for the receive interrupt are as follows: Note: In ULP domain the trigger levels are used for: 0: LVL_1_4 4: LVL_FULL For undefined settings the default configuration is used.<br>1h = RX FIFO >= 1/4 full<br>2h = RX FIFO >= 1/2 full (default)<br>3h = RX FIFO >= 3/4 full<br>5h = RX FIFO is full<br>6h = RX_FIFO >= (MAX_FIFO_LEN -1)<br>7h = RX_FIFO <= 1 |
| 3 | TXCLR | R/W | 0h | TX FIFO CLEAR. Setting this bit will clear the TX FIFO contents.<br>0h = Disable FIFO clear<br>1h = Enable FIFO Clear |
| 2-0 | TXIFLSEL | R/W | 2h | FIFO Level Select for generating events (interrupt/dma). Note: for undefined settings the default configuration is used.<br>1h = TX FIFO <= 3/4 empty<br>2h = TX FIFO <= 1/2 empty (default)<br>3h = TX FIFO <= 1/4 empty<br>5h = TX FIFO is empty<br>6h = TX FIFO <= 1<br>7h = TX_FIFO >= (MAX_FIFO_LEN -1) |



Copyright © 2026 Texas Instruments Incorporated

### 34.3.25 WCLKSRC (Offset = 1110h) [Reset = X0000000h]

WCLKSRC is shown in Figure 34-40 and described in Table 34-37.

Return to the Summary Table.

This register configures the WCLK Source

**Figure 34-40. WCLKSRC**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | WCLKINV | WBCLKSRC | |
| R-0h | | | | | R/W-Xh | R/W-Xh | |

**Table 34-37. WCLKSRC Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | RESERVED | R | 0h | |
| 2 | WCLKINV | R/W | Xh | This field Inverts WCLK source (pad or internal).<br>0h = Source is not inverted<br>1h = Source is inverted |
| 1-0 | WBCLKSRC | R/W | Xh | This field selects WCLK/BCLK source for I2S.<br>0h = None ('0')<br>1h = External WCLK generator, from pad<br>2h = Internal WCLK generator, from module PRCM<br>3h = Not supported. Will give same WCLK as 'NONE' ('00') |



Copyright © 2026 Texas Instruments Incorporated

### 34.3.26 DIRCFG (Offset = 1118h) [Reset = 00000000h]

DIRCFG is shown in Figure 34-41 and described in Table 34-38.

Return to the Summary Table.

This register configures the direction of data pins(AD0/AD1)

**Figure 34-41. DIRCFG**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | AD1 | RESERVED | | | AD0 |
| R-0h | | | R/W-0h | R-0h | | | R/W-0h |

**Table 34-38. DIRCFG Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | RESERVED | R | 0h | |
| 5-4 | AD1 | R/W | 0h | The field configures the AD1 data pin direction<br>0h = Not in use (disabled)<br>1h = Input<br>2h = Output<br>3h = Reserved |
| 3-2 | RESERVED | R | 0h | |
| 1-0 | AD0 | R/W | 0h | The field configures the AD0 data pin direction<br>0h = Not in use (disabled)<br>1h = Input<br>2h = Output<br>3h = Reserved |



Copyright © 2026 Texas Instruments Incorporated

### 34.3.27 TXDATA (Offset = 1120h) [Reset = 00000000h]

TXDATA is shown in Figure 34-42 and described in Table 34-39.

Return to the Summary Table.

Transmit Data Register. This register is the transmit data register (the interface to the FIFOs). For transmitted data, if the FIFO is enabled, data written to this location is pushed onto the transmit FIFO. If the FIFO is disabled, data is stored in the transmitter holding register (the bottom word of the transmit FIFO). A write to this register initiates a transmission from the module.

**Figure 34-42. TXDATA**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 34-39. TXDATA Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | W | 0h | Data Transmitted or Received Data that is to be transmitted via the is written to this field. When read, this field contains the data that was received by the .<br>0h = Smallest value<br>FFFFFFFFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 34.3.28 RXDATA (Offset = 1124h) [Reset = 00000000h]

RXDATA is shown in Figure 34-43 and described in Table 34-40.

Return to the Summary Table.

Receive Data Register. This register is the data receive register (the interface to the FIFOs). For received data, if the FIFO is enabled, the data byte and the 4-bit status (break, frame, parity, and overrun) is pushed onto the 12-bit wide receive FIFO. If the FIFO is disabled, the data byte and status are stored in the receiving holding register (the bottom word of the receive FIFO). The received data can be retrieved by reading this register.

**Figure 34-43. RXDATA**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 34-40. RXDATA Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | DATA | R | 0h | Received Data. When read, this field contains the data that was received by the .<br>0h = Smallest value<br>FFFFFFFFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 34.3.29 WMASK0 (Offset = 1148h) [Reset = 00000000h]

WMASK0 is shown in Figure 34-44 and described in Table 34-41.

Return to the Summary Table.

This register configures the word selection mask for data pin 0(AD0)

**Figure 34-44. WMASK0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | MASK | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | R/W-0h | | | | | | | | | | | | | | | |

**Table 34-41. WMASK0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15-0 | MASK | R/W | 3h | Bit-mask indicating valid channels in a frame on AD0. In single-phase mode, each bit represents one channel, starting with LSB for the first word in the frame. A frame can contain up to 8/16 channels based on SoC configuration. Channels that are not included in the mask will not be sampled nor stored in memory; data clocked out in these slots is as per configured EMPTYSLOTOUTPUT. In dual-phase mode, only the two LSBs are considered. For a stereo configuration, set both bits. For a mono configuration, set bit 0 only. In mono mode, only channel 0 will be sampled and stored to memory, and channel 0 will be repeated when clocked out. In mono mode, only channel 0 will be sampled and stored to memory, and channel 0 will be repeated in the second phase when clocked out. If all bits are zero, no input words will be stored to memory, and the output data lines will be constant '0'. This can be utilized when PWM debug output is desired without any actively used output pins. |



Copyright © 2026 Texas Instruments Incorporated

### 34.3.30 WMASK1 (Offset = 114Ch) [Reset = 00000000h]

WMASK1 is shown in Figure 34-45 and described in Table 34-42.

Return to the Summary Table.

This register configures the word selection mask for data pin 1(AD1)

**Figure 34-45. WMASK1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | MASK | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | R/W-0h | | | | | | | | | | | | | | | |

**Table 34-42. WMASK1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15-0 | MASK | R/W | 3h | Bit-mask indicating valid channels in a frame on AD1. In single-phase mode, each bit represents one channel, starting with LSB for the first word in the frame. A frame can contain up to 8/16 channels based on SoC configuration. Channels that are not included in the mask will not be sampled nor stored in memory; data clocked out in these slots is as per configured EMPTYSLOTOUTPUT. In dual-phase mode, only the two LSBs are considered. For a stereo configuration, set both bits. For a mono configuration, set bit 0 only. In mono mode, only channel 0 will be sampled and stored to memory, and channel 0 will be repeated when clocked out. In mono mode, only channel 0 will be sampled and stored to memory, and channel 0 will be repeated in the second phase when clocked out. If all bits are zero, no input words will be stored to memory, and the output data lines will be constant '0'. This can be utilized when PWM debug output is desired without any actively used output pins. |



Copyright © 2026 Texas Instruments Incorporated

### 34.3.31 MCLKDIV (Offset = 1160h) [Reset = 00000000h]

MCLKDIV is shown in Figure 34-46 and described in Table 34-43.

Return to the Summary Table.

This field configures MCLK division ratio

**Figure 34-46. MCLKDIV**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | MDIV | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | R/W-0h | | | | | | | | | |

**Table 34-43. MCLKDIV Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-10 | RESERVED | R | 0h | |
| 9-0 | MDIV | R/W | 0h | An unsigned factor of the division ratio used to generate MCLK [2-1024]: MCLK = MCUCLK/MDIV[Hz] A value of 0 is interpreted as 1024. A value of 1 is invalid. If MDIV is odd the low phase of the clock is one MCUCLK period longer than the high phase. |



Copyright © 2026 Texas Instruments Incorporated

### 34.3.32 WCLKDIV (Offset = 1164h) [Reset = 00000000h]

WCLKDIV is shown in Figure 34-47 and described in Table 34-44.

Return to the Summary Table.

This field configures WCLK division ratio

**Figure 34-47. WCLKDIV**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | WDIV | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | R/W-0h | | | | | | | | | | | | | | | |

**Table 34-44. WCLKDIV Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15-0 | WDIV | R/W | 0h | If WCLKPHASE = 0, Single phase. WCLK is high one BCLK period and low WDIV[9:0] (unsigned, [1-1023]) BCLK periods. WCLK = MCUCLK / BDIV*(WDIV[9:0] + 1) [Hz] If WCLKPHASE = 1, Dual phase. Each phase on WCLK (50% duty cycle) is WDIV[9:0] (unsigned, [1-1023]) BCLK periods. WCLK = MCUCLK / BDIV*(2*WDIV[9:0]) [Hz] If WCLKPHASE = 2, User defined. WCLK is high WDIV[7:0] (unsigned, [1-255]) BCLK periods and low WDIV[15:8] (unsigned, [1-255]) BCLK periods. WCLK = MCUCLK / (BDIV*(WDIV[7:0] + WDIV[15:8]) [Hz] |



Copyright © 2026 Texas Instruments Incorporated

### 34.3.33 BCLKDIV (Offset = 1168h) [Reset = 00000000h]

BCLKDIV is shown in Figure 34-48 and described in Table 34-45.

Return to the Summary Table.

This field configures BCLK division ratio

**Figure 34-48. BCLKDIV**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | BDIV | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | R/W-0h | | | | | | | | | |

**Table 34-45. BCLKDIV Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-10 | RESERVED | R | 0h | |
| 9-0 | BDIV | R/W | 0h | An unsigned factor of the division ratio used to generate BCLK [2-1024]: BCLK = MCUCLK/BDIV[Hz] A value of 0 is interpreted as 1024. A value of 1 is invalid. If BDIV is odd and SMPLEDGE = 0, the low phase of the clock is one MCUCLK period longer than the high phase. If BDIV is odd and SMPLEDGE = 1 , the high phase of the clock is one MCUCLK period longer than the low phase. |



Copyright © 2026 Texas Instruments Incorporated

Chapter 35

# Timers (TIMx)

The timer module (TIMx) is a timer counting module with multiple compare/capture blocks. Based on the device, two types of timers are available: general-purpose timers (TIMG) and advanced control timers (TIMA). Both timers include many common features that can be used for a variety of functions such as measuring the input signal edge and period (capture mode) or generating output waveforms like PWMs (compare mode output). See the device-specific data sheet to determine which timers and timer instances are available.


**35.1 TIMx Overview** ..... 2104
**35.2 TIMx Operation** ..... 2107
**35.3 TIMx Registers** ..... 2157





Copyright © 2026 Texas Instruments Incorporated

## 35.1 TIMx Overview

The timer module (TIMx) is a timer counting module with multiple compare/capture blocks. Based on the device, two types of timers are available: general-purpose timers (TIMG) and advanced control timers (TIMA). Both timers use a common timer architecture, which allows for easy migration between timer instances with common functions. This minimizes the need to write extra software for timer-based applications and allows for easy porting and maintenance between TIMx instances.

**Note**

See Section 35.1.3 to determine the common features available between TIMG and TIMA instances.

In this section:

- “TIMx” indicates a common feature available on TIMG and TIMA.
- "TIMA" indicates a feature available only on TIMA.
- "TIMG" indicates a feature available only on TIMG.

### *35.1.1 TIMG Overview*

The TIMG module consists of 16-bit and 32-bit auto reload counters driven by a programmable prescaler with up to 4 capture/compare (CC) blocks for multiple capture/compares, PWM outputs, and interval timing. TIMG also has extensive event generation capabilities, including counter overflow, reload, and capture/compare actions for a variety of use cases.

#### 35.1.1.1 TIMG Features

Specific features for TIMG include:

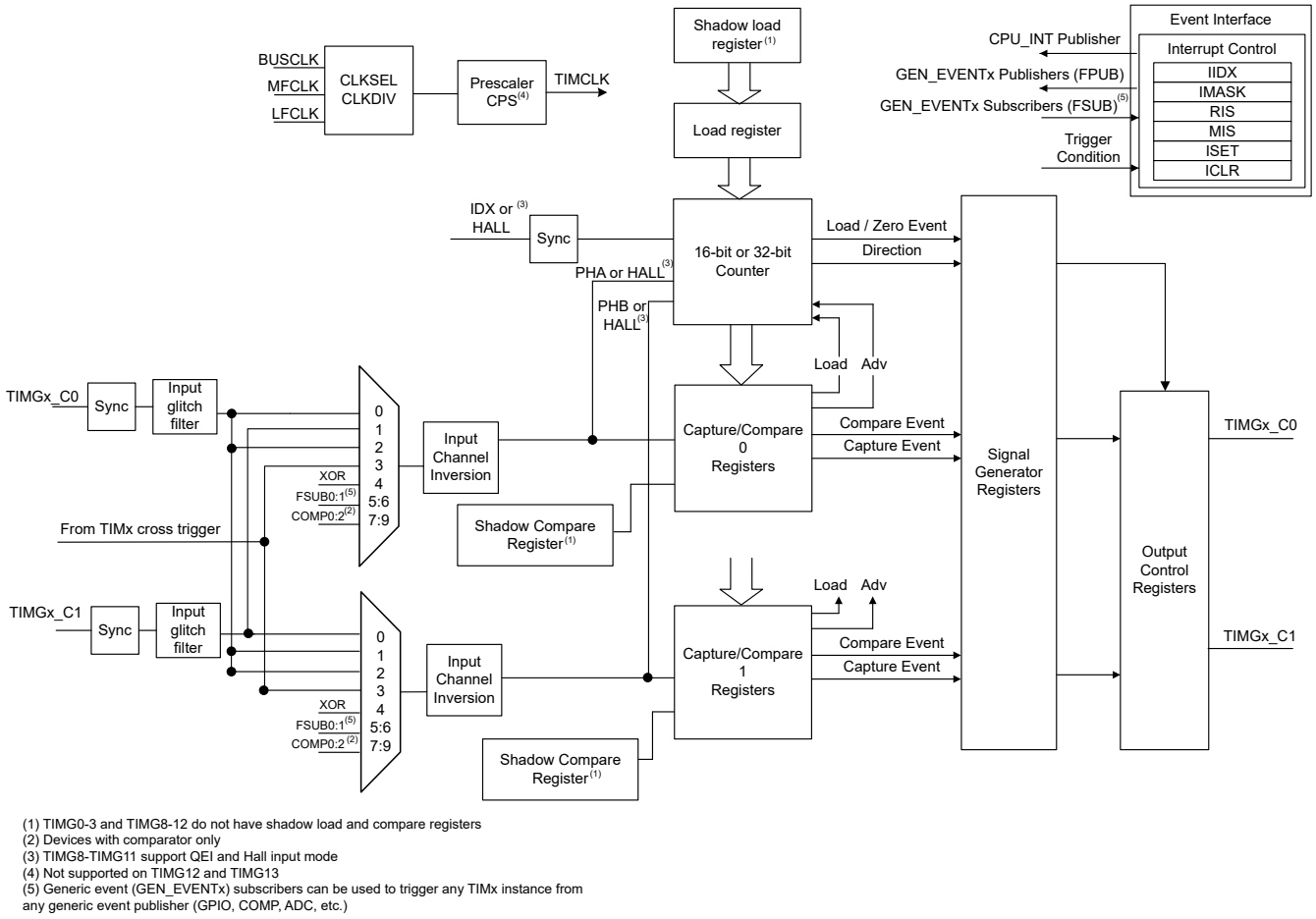
- 16-bit or 32-bit up, down, or up-down counter, with repeat-reload mode
- 8-bit programmable prescaler to divide the counter clock frequency
- Up to four independent channels for
  - Output compare
  - Input capture
  - PWM output (Edge-Aligned and Center-Aligned)
  - One-shot mode
- Selectable and configurable clock source
- Shadow register mode for load and compare values (see Section 35.2.4)
- Support for quadrature encoder interface (QEI) (see Section 35.2.3.1.3)
- 3-input Hall sensor mode for position sensing and speed computation (see Section 35.2.3.1.4)
- Support synchronization and cross trigger among different TIMx instances in the same power domain (see Section 35.2.7)
- Support CPU interrupt generation and cross peripherals (such as ADC, DAC, etc.) using the Event (see Section 35.2.9)

#### 35.1.1.2 Functional Block Diagram

Figure 35-1 shows the TIMG block diagram.



Copyright © 2026 Texas Instruments Incorporated




(1) TIMG0-3 and TIMG8-12 do not have shadow load and compare registers
(2) Devices with comparator only
(3) TIMG8-TIMG11 support QEI and Hall input mode
(4) Not supported on TIMG12 and TIMG13
(5) Generic event (GEN_EVENTx) subscribers can be used to trigger any TIMx instance from any generic event publisher (GPIO, COMP, ADC, etc.)

**Figure 35-1. TIMG Functional Block Diagram**

### *35.1.2 TIMA Overview*

The TIMA module consists of a 16-bit auto reload counter driven by a programmable prescaler with up to four capture/compare (CC) blocks for multiple capture/compares, PWM outputs with deadband insertion, and interval timing. TIMA has extensive event generation capabilities from different counter events such as overflow, reload, and each of the capture/compare events. It also has the hardware design to handle the fault signal generated by internal or external circuitry to indicate a fault in the system.

#### 35.1.2.1 TIMA Features

- 16-bit up, down, or up-down counter, with repeat-reload mode
- Selectable and configurable clock source
- 8-bit programmable prescaler to divide the counter clock frequency
- Repeat counter to generate an interrupt or event only after a given number of cycles of the counter (see Section 35.2.1.2)
- Up to four independent channels for:
  - Output compare
  - Input capture
  - PWM output (Edge-Aligned and Center-Aligned)
  - One-shot mode
- Two additional capture/compare channels for internal events (CC4/CC5)
- Shadow register for load and compare values (see Section 35.2.4)
- Complementary PWM output with programmable deadband insertion (see Section 35.2.5.2.4)
- Asymmetric PWM output (see Section 35.2.2.5)

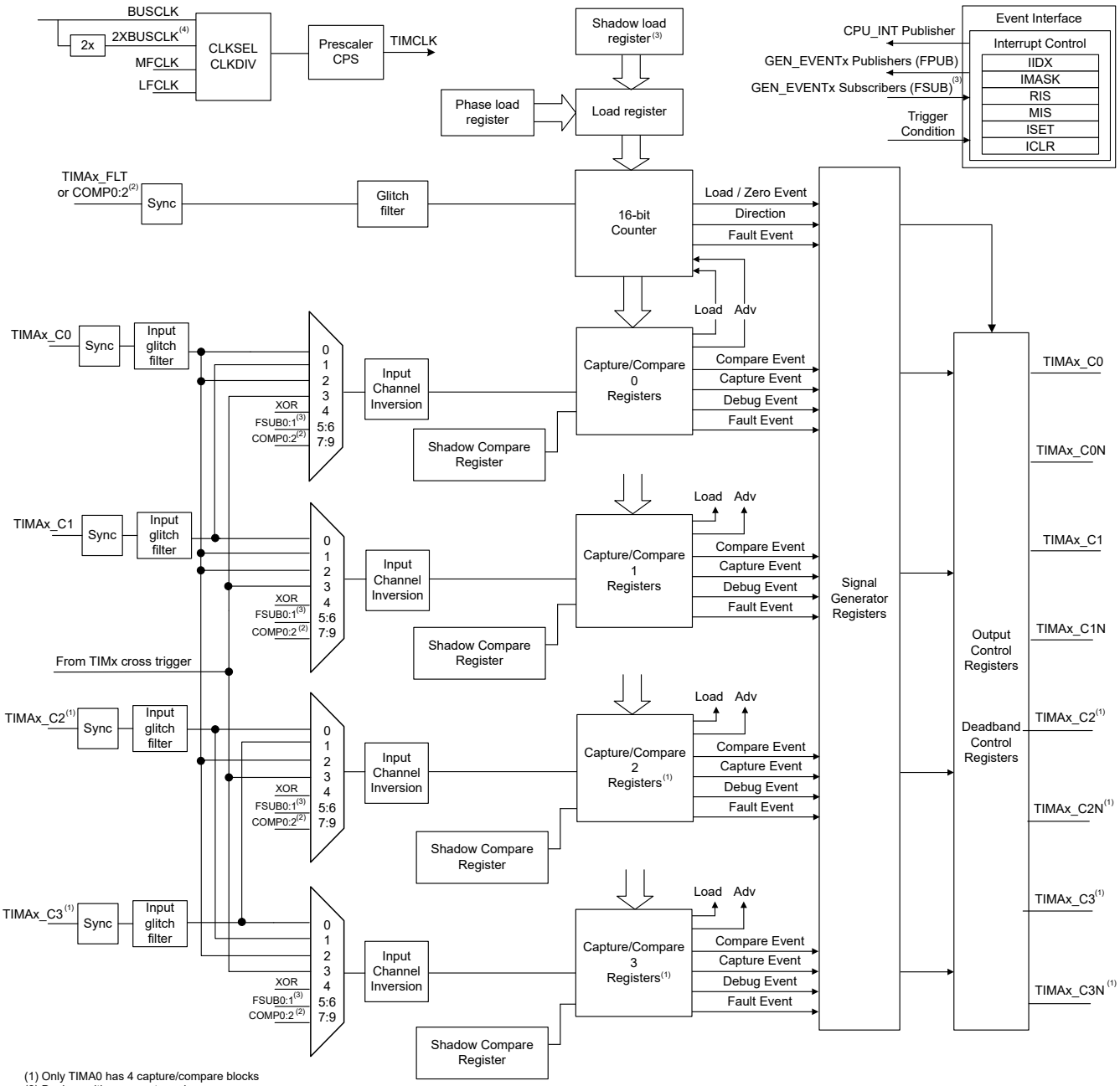


Copyright © 2026 Texas Instruments Incorporated

- Fault handling mechanisms to ensure the output signals are in a safe user-defined state when a fault condition is encountered (see Section 35.2.6)
- Support synchronization and cross trigger among different TIMx instances in the same power domain (see Section 35.2.7)
- Support CPU interrupt generation and cross peripherals (such as ADC, DAC, etc.) using the Event (see Section 35.2.9)

### 35.1.2.2 Functional Block Diagram

Figure 35-2 shows the TIMA block diagram.




(1) Only TIMA0 has 4 capture/compare blocks
(2) Devices with comparator only
(3) Generic event (GEN_EVENTx) subscribers can be used to trigger any TIMx instance from any generic event publisher (GPIO, COMP, ADC, etc.)
(4) Refer to device specific data sheet for 2XBUSCLK functionality

**Figure 35-2. TIMA Block Diagram**



Copyright © 2026 Texas Instruments Incorporated

### 35.1.3 TIMx Instance Configuration

Table 35-1 shows the TIMx instance configuration for TIMA and TIMG instances.

**Table 35-1. TIMx Instance Configuration**

| Instance | Counter Resolution | Prescaler | Repeat Counter | CCP Channels (External/ Internal) | External PWM Channels | Phase Load | Shadow Load | Shadow CCs | Deadband | Fault Handler | QEI / Hall Input Mode |
|---|---|---|---|---|---|---|---|---|---|---|---|
| TIMG0 | 16-bit | 8-bit | - | 2 | 2 | - | - | - | - | - | - |
| TIMG1 | 16-bit | 8-bit | - | 2 | 2 | - | - | - | - | - | - |
| TIMG2 | 16-bit | 8-bit | - | 2 | 2 | - | - | - | - | - | - |
| TIMG3 | 16-bit | 8-bit | - | 2 | 2 | - | - | - | - | - | - |
| TIMG4 | 16-bit | 8-bit | - | 2 | 2 | - | Yes | Yes | - | - | - |
| TIMG5 | 16-bit | 8-bit | - | 2 | 2 | - | Yes | Yes | - | - | - |
| TIMG6 | 16-bit | 8-bit | - | 2 | 2 | - | Yes | Yes | - | - | - |
| TIMG7 | 16-bit | 8-bit | - | 2 | 2 | - | Yes | Yes | - | - | - |
| TIMG8 | 16-bit | 8-bit | - | 2 | 2 | - | - | - | - | - | Yes |
| TIMG9 | 16-bit | 8-bit | - | 2 | 2 | - | - | - | - | - | Yes |
| TIMG10 | 16-bit | 8-bit | - | 2 | 2 | - | - | - | - | - | Yes |
| TIMG11 | 16-bit | 8-bit | - | 2 | 2 | - | - | - | - | - | Yes |
| TIMG12 | 32-bit | - | - | 2 | 2 | - | - | Yes | - | - | - |
| TIMG13 | 32-bit | - | - | 2 | 2 | - | - | Yes | - | - | - |
| TIMG14 | 16-bit | 8-bit | - | 4 | 4 | - | - | - | - | - | - |
| TIMA0 | 16-bit | 8-bit | Yes | 4/2 | 8 | Yes | Yes | Yes | Yes | Yes | - |
| TIMA1 | 16-bit | 8-bit | Yes | 2/2 | 4 | Yes | Yes | Yes | Yes | Yes | - |

**Note**

On TIMA instances, external PWM channels are pairs of complementary PWM output signals with deadband generation with respect to the CC block instance, such as TIMA0_C0 and TIMA0_C0N. For independent PWM output generation, separate noninverting channels must be used, such as TIMA0_C0 and TIMA0_C1.

**Note**

Internal CC channels 4 and 5 (CC_45) can be used for internal compare events and are available in TIMA only.

**Note**

Look at the device-specific data sheet to check which TIMx instances are available on the device and their respective power domains

## 35.2 TIMx Operation

The TIMx module is configured with user software. The setup and operation of TIMx is discussed in the following sections.

**Note**

There are register arrays in the timer module to group registers with same bit fields for different capture/compare ports. For example, TIMx.CC_01[0/1] is a register array that contains the capture/compare registers for both CCP0 and CCP1. Access TIMx.CC_01[0] to read the CC0 capture/compare value, and access TIMx.CC_01[1] to read the CC1 capture/compare value.



Copyright © 2026 Texas Instruments Incorporated

**Note**

TIMx supports event subscribers and event publishers through event register arrays. For more information, please see the Events chapter.

### *35.2.1 Timer Counter*

All TIMx instances have 16-bit counter blocks except for TIMG12 and TIMG13, which have 32-bit counter blocks. The timer counter register (TIMx.CTR) can count down, up-and-down, or up depending on the operation mode. It can also be read or written with software. Each count occurs with each rising edge of the TIMCLK signal or with both edges of external signals.

**Enabling the TIMx Counter**

The counter is clocked by the prescaler output TIMCLK. The counter enable bit TIMx.CTRCTL.EN can be enabled in two ways:

- Set in software manually
- After a load condition (LCOND) or zero condition (ZCOND) is met, and the counter value after enable (CVAE) is changed to the load value or zero value, respectively.

**Note**

The ability to write the counter register while TIMx is running is possible but should be avoided because the user write can collide with a load, zero, or advance event. Depending on the prescaler ratio, the application cannot predict the timing of the write, which can affect the correct timer period.

It is recommended to avoid changing the configuration registers while the timer is running, if the configuration is intended to be changed, stop the counter then adjust the timer settings.

#### 35.2.1.1 Clock Source Select and Prescaler

The TIMx clock (TIMCLK) can be sourced from an internal clock or an external signal trigger to advance the clock.

##### *35.2.1.1.1 Internal Clock and Prescaler*

The TIMx clock (TIMCLK) can be sourced from BUSCLK, MFCLK and LFCLK by setting the TIMx.CLKSEL register. It can also be divided by a ratio by setting the TIMx.CLKDIV register and a prescaler (if present). The selected source clock is always available and the frequency depends on the power mode. For more information, see the *Clock Module (CKM)* section.

The TIMCLK can come from the following sources and refer to the datasheet for the clock availability as per the power domain:

- BUSCLK: the current bus clock is selected as the source for TIMx. The current bus clock depends on power domain.
  - If the TIMx instance is in Power Domain 1 (PD1), refer to MCLK.
  - If the TIMx instance is in Power Domain 0 (PD0), refer to ULPCLK.
- MFCLK: MFCLK is selected as the source for TIMx, refer to MFCLK.
  - When entering a low power mode which disables the MFCLK, any timer sourced with MFCLK will also stop counting until MFCLK operation is restored.
- LFCLK: LFCLK is selected as the source for TIMx, refer to LFCLK.

The selected clock source can be passed directly to TIMx or divided by the 8-bit prescaler. The prescaler setting can be configured with register TIMx.CPS.PCNT bit. The selected TIMCLK source is divided by a value of (PCNT+1). A PCNT value of 0 divides TIMCLK by 1, effectively bypassing the divider. A PCNT value of greater than 0 divides the TIMCLK source to generate a slower clock.



Copyright © 2026 Texas Instruments Incorporated

TIMx also has a software mechanism for disabling the timer clock. Set TIMx.CCLKCTL.CLKEN to 0 to put the timer in an IDLE state.

**TIMCLK Configuration**

To configure the clock source, divider, and prescaler:

1. Select the TIMCLK clock source (BUSCLK, MFCLK, or LFCLK) using the CLKSEL register.
2. Optionally divide the TIMCLK using CLKDIV.RATIO.
3. In TIMx instances with prescalers, optionally set a prescaler using CPS.PCNT.
4. Enable the TIMCLK by setting CCLKCTL.CLKEN = 1.

The frequency of TIMCLK is determined using Equation 35.

$$f_{TIMCLK} = \frac{f_{CLK\_SOURCE}}{((CLKDIV.RATIO + 1)*(CPS.PCNT + 1))} \quad (35)$$

#### *35.2.1.1.2 External Signal Trigger*

The counter can also advance (increment or decrement) by using an external signal on the timer input pin or by triggering from an event from other peripherals in the system. This can be configured by using the advance condition setting (TIMx.CCCTL_xy[0/1].ACOND) to specify what creates the advance event. To specify what event advances the counter, use the TIMx.CTRCTL.CAC setting.

The counter can advance using the internal clock TIMCLK, a different edge of an timer external input, or an internal trigger event from other peripherals.

#### 35.2.1.2 Repeat Counter (TIMA only)

In TIMA only, the repeat counter (RC) is an 8-bit counter that provides the mechanism to suppress unnecessary events and generate real events for interrupt generation. Specifically, the repeat counter can suppress Load, Compare, and Zero events in the case where the timer is generating events that repeat for a known number of cycles, such a periodic PWM output waveform. This prevents generating excessive and unnecessary interrupts every timer period.

When the timer counter (TIMA.CTR) is running, the repeat counter (TIMA.RC) decrements when the timer counter loads or zeroes (depending on counting mode). Software can set the how many timer counter events occur until generating the interrupts/events by setting the TIMA.RCLD register. When TIMA.RC = 0, the repeat counter is reset back to TIMA.RCLD and a Repeat Counter Zero event occurs (REPC) in the Interrupt and Event Status registers.

**Note**

If software configures the counter to stop counting in a debug or fault condition, also stop the repeat counter. See Section 35.2.6 and Section 35.2.10 for more details.

Additionally, the repeat counter provides the ability to suppress generation of Zero, Load, and Compare events when TIMA.RC does not equal zero.

- Zero and Load events are suppressed by setting TIMA.CTRCTL.SLZERCNEZ register bit
- Compare events (see Table 35-15) are suppressed by setting the TIMA.CCCTL_xy[0/1].SCERCNEZ bit

Table 35-2 shows the repeat counter behavior with respect to the timer counter and repeat counter load value.

**Table 35-2. Repeat Counter Behavior**

| TIMA.CTR is counting | Counter value | TIMA.RC = 0 | Repeat Counter Behavior | Suppress Load and Zero Events (SLZERCNEZ = 1) | Suppress Compare Events (SCERCNEZ = 1) |
|---|---|---|---|---|---|
| No | - | - | Does not change | Yes | Yes |
| Yes | TIMA.CTR ≠ 0<br>Or load | - | Does not change | Yes | Yes |



Copyright © 2026 Texas Instruments Incorporated

**Table 35-2. Repeat Counter Behavior (continued)**

| TIMA.CTR is counting | Counter value | TIMA.RC = 0 | Repeat Counter Behavior | Suppress Load and Zero Events (SLZERCNEZ = 1) | Suppress Compare Events (SCERCNEZ = 1) |
|---|---|---|---|---|---|
| Yes | TIMA.CTR = 0 Or load | No | Decrease RC value by 1 | Yes | Yes |
| Yes | TIMA.CTR = 0 Or load | Yes | TIMA.RC = TIMA.RCLD | No | No |

### Repeat counter example

As shown in Figure 35-3, the TIMA.CTR is configured for down-counting mode and zero events are generated once TIMA.CTR = 0. To suppress interrupt generation until 4 timer reloads occur, set TIMA.RCLD = 4 and TIMA.CTRCTL.SLZERCNEZ = 1 to suppress zero and load events until RC = 0 .

**Note**

The use of the repeat counter does not affect the output signal generation. All events are generated regardless of the repeat counter value sent to the signal generator unit.

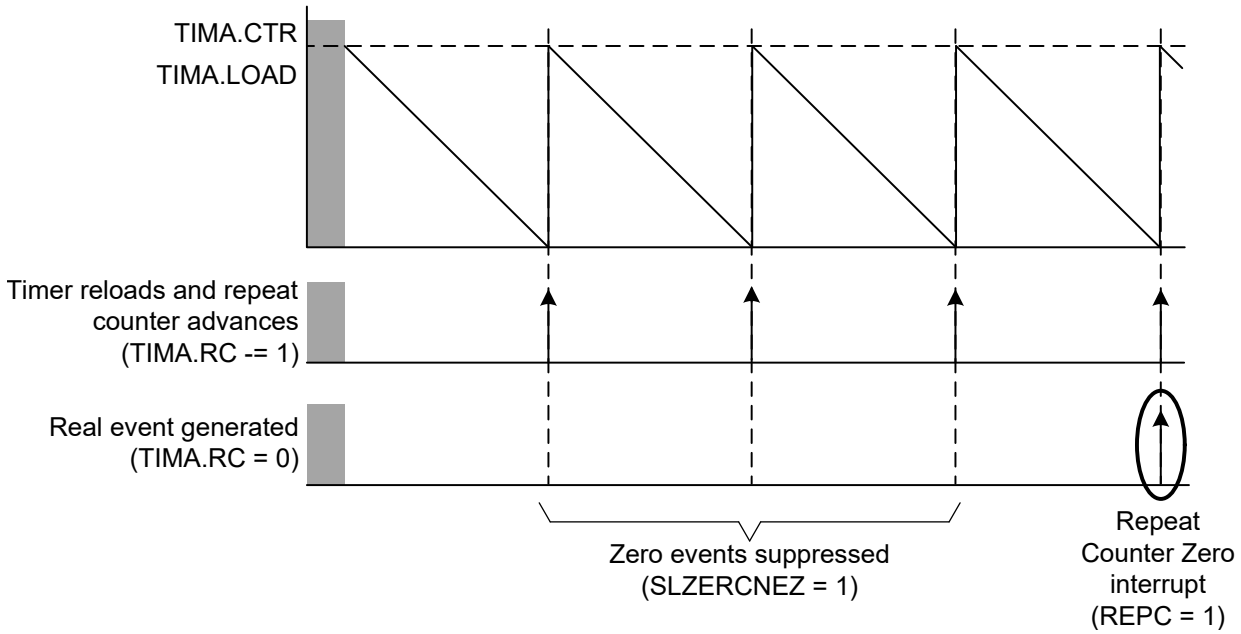



**Figure 35-3. Event Suppressed by Repeat Counter with TIMx.RCLD = 4 in Down Counting Mode**

### *35.2.2 Counting Mode Control*

When the device is out of reset, TIMx is disabled. Writing 1 to the TIMx.CTRCTL.EN bit enables the counter. This bit is automatically cleared if TIMx.CTRCTL.REPEAT=0 (do not automatically reload), and the counter value equals zero.

TIMx has three counting modes when enabled: down, up/down, and up. The operating mode is selected by TIMx.CTRCTL.CM bit (shown in Table 35-3). In up/down mode, after the counter is enabled, the timer will start counting from the TIMx.CTRCTL.CVAE setting.

**Table 35-3. TIMx Counting Modes (CM)**

| TIMx.CTRCTL.CM | Counting Mode |
|---|---|
| 0 | Down |



Copyright © 2026 Texas Instruments Incorporated

**Table 35-3. TIMx Counting Modes (CM) (continued)**

| TIMx.CTRCTL.CM | Counting Mode |
|---|---|
| 1 | Up/Down |
| 2 | Up |

**Table 35-4. TIMx Counter Value after Enable (CVAE)**

| Count Value After Enable (CVAE) | Description |
|---|---|
| 0 | LOAD value |
| 1 | Unchanged from current value |
| 2 | Zero value |

**Note**

Counter value after enable (CVAE) is only applicable for Up/Down mode.

#### 35.2.2.1 One-shot and Periodic Modes

Figure 35-4 shows TIMx working in both one-shot mode and periodic mode.

**One-shot Mode**

When TIMx.CTRCTL.REPEAT bit is set to 0, TIMx does not advance when:

- TIMx.CTR value reaches 0 in either down- or up/down-counting mode
- TIMx.CTR value reaches TIMx.LOAD in up-counting mode

**Periodic Mode (Counter Reload)**

When TIMx.CTRCTL.REPEAT is set to 1h, TIMx automatically repeats once a zero event occurs. This happens when:

- TIMx.CTR reaches 0 in either down- or up/down-counting mode
  - In down-counting mode, a zero event is followed by a load event at the next advance condition
  - In up/down-counting mode, a zero event is followed by an advance event (+1)
- TIMx.CTR reaches TIMx.LOAD in up-counting mode
  - A load event is followed by a zero at the next advance condition

TIMx.CTRCTL.REPEAT can also be set to 3h for TIMx to repeat only when not in a debug condition. If there is a debug condition, TIMx will count to the zero event and repeat only once the debug condition is removed.



Copyright © 2026 Texas Instruments Incorporated

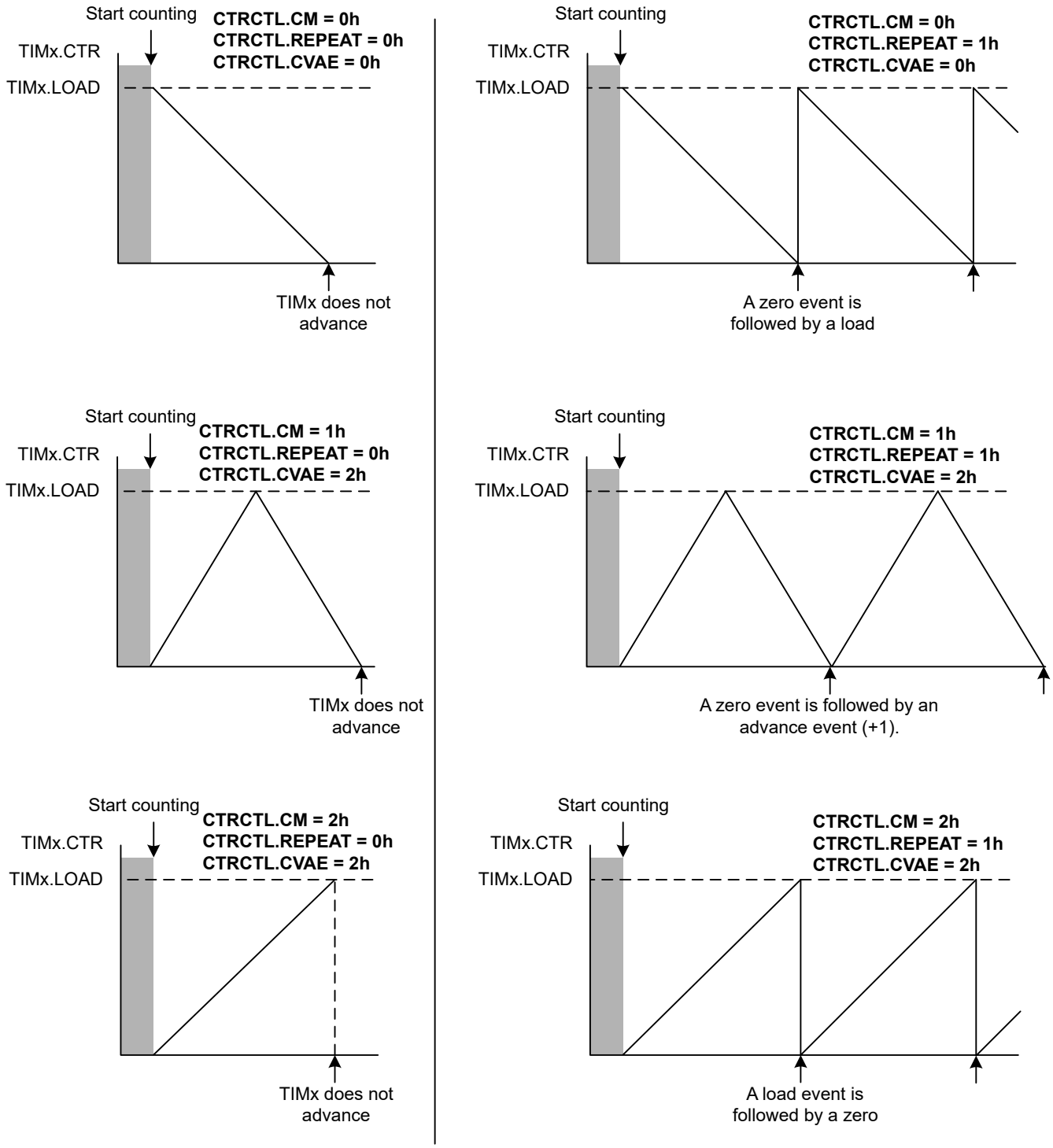



**Figure 35-4. One-shot and Periodic Mode Behavior**

#### 35.2.2.2 Down Counting Mode

In down counting mode (CM = 0), TIMx counts from the value defined in TIMx.LOAD down to zero. When the TIMx.CTR value equals zero and TIMx.CTRCTL.REPEAT is set to 1, the TIMx.LOAD value is loaded into TIMx.CTR and the timer repeats the down counting pattern as shown in Figure 35-5.



Copyright © 2026 Texas Instruments Incorporated

A Zero event is generated when TIMx counts to zero. A Load event is generated when TIMx counts from zero to the TIMx.LOAD value. Figure 35-6 shows the event generating cycle.

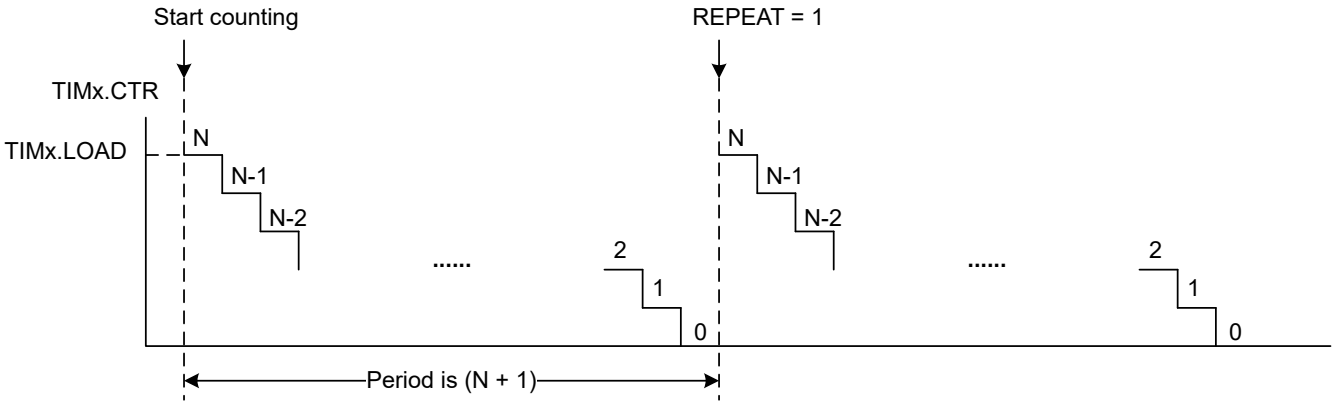



**Figure 35-5. Down Counting Mode**

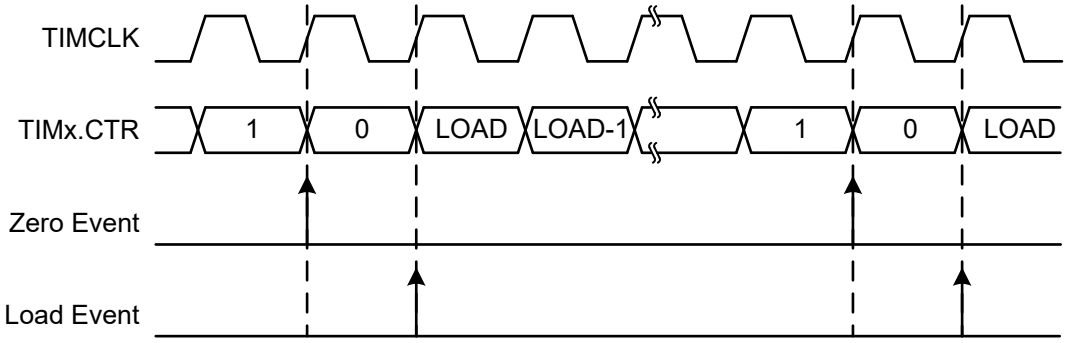



**Figure 35-6. Down Counting Mode Event Generation**

#### 35.2.2.3 Up/Down Counting Mode

The Up/Down counting mode can count in an down-up direction or an up-down direction depending on TIMx.CTRCTL.CVAE value. The TIMx.CTRCTL.CVAE bits specify the initialization condition of the counter.

**Table 35-5. Counter Value after Enable Initialization Conditions**

| TIMx.CTRCTL.CVAE Value | Counter Value After Enable |
|---|---|
| 0x0 | Load Value |
| 0x1 | No Change |
| 0x2 | Zero |

**Counting in down-up direction**

When TIMx.CTRCTL.CVAE = 0, TIMx.CTR is set to TIMx.LOAD register value and TIMx counts in the down direction. When it reaches zero, a Zero event is generated and, if repeat is enabled, TIMx counts back up to TIMx.LOAD value. A Load event is generated when it reaches TIMx.LOAD value.

Figure 35-7 shows TIMx counting in the down-up direction when TIMx.CTRCTL.CVAE = 0.



Copyright © 2026 Texas Instruments Incorporated

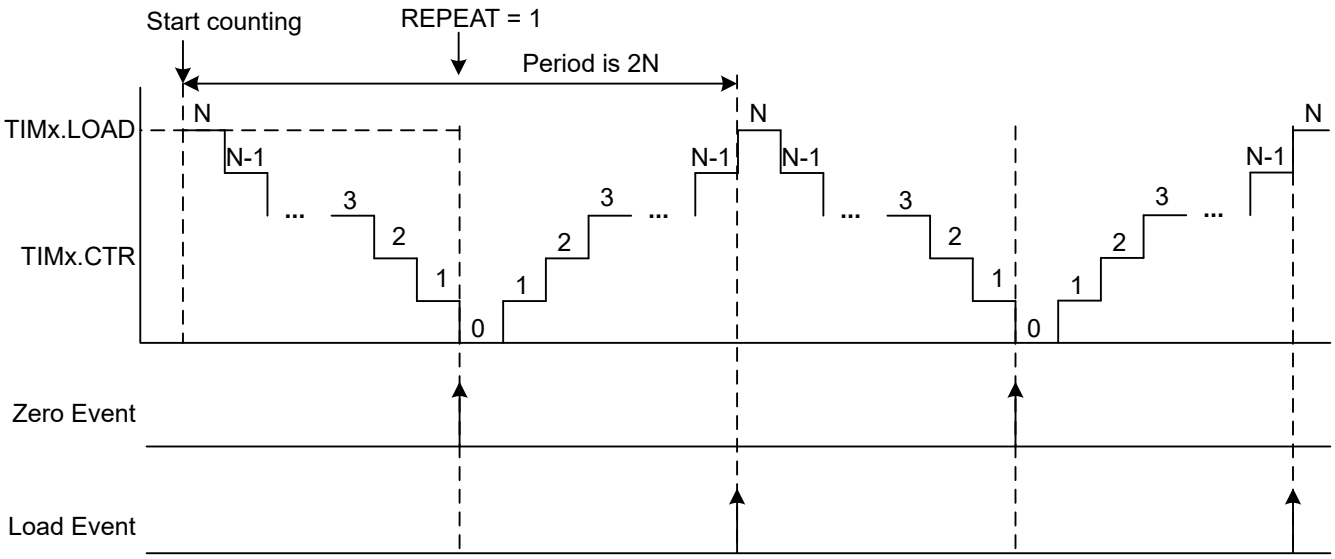



**Figure 35-7. Down-up Counting Mode and Event Generation, CVAE = 0**

### Counting in up-down direction

When TIMx.CTRCTL.CVAE = 2, TIMx.CTR is set to zero and TIMx counts in the up direction. When it reaches TIMx.LOAD, a Load event is generated and TIMx counts back down to zero. A Zero event is generated when it reaches zero.

Figure 35-8 shows TIMx counting in up-down direction when TIMxCTRCTL.CVAE = 2.

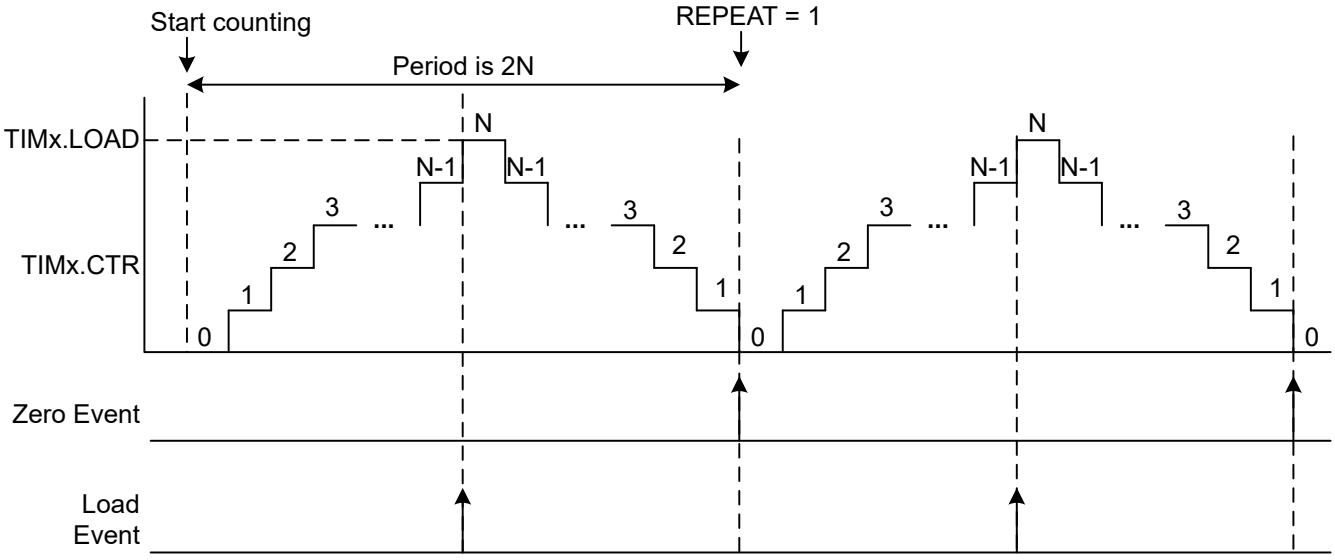



**Figure 35-8. Up-down Counting Mode and Event Generation, CVAE = 2**

#### 35.2.2.4 Up Counting Mode

In up counting mode, TIMx counts from zero up to the value defined in TIMx.LOAD. When the TIMx.CTR value equals TIMx.LOAD and TIMx.CTRCTL.REPEAT is not equal to 0, the zero is loaded into TIMx.CTR and the timer repeats the up counting pattern as shown in Figure 35-5.

A Load event is generated when TIMx counts to TIMx.LOAD. A Zero event is generated when TIMx counts from TIMx.LOAD to the zero value. Figure 35-6 shows the event generating cycle.



Copyright © 2026 Texas Instruments Incorporated

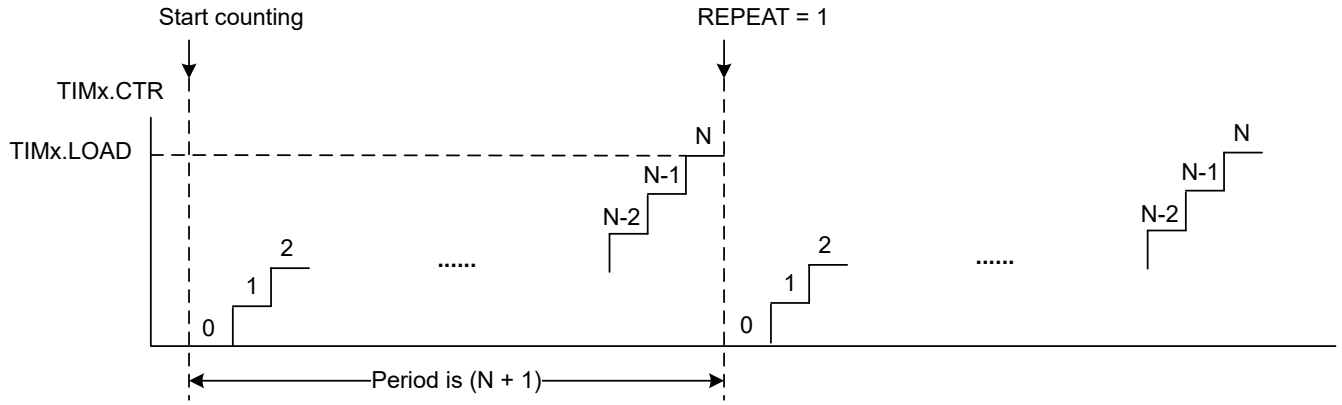



**Figure 35-9. Up Counting Mode**

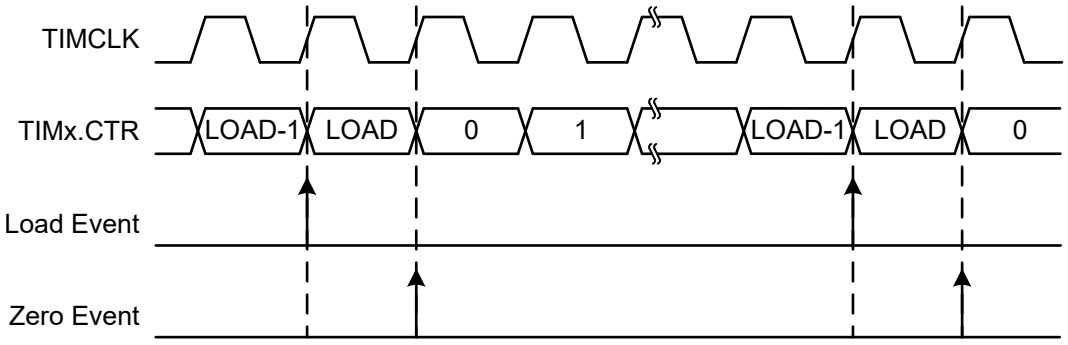



**Figure 35-10. Up Counting Mode Event Generation**

#### 35.2.2.5 Phase Load (TIMA only)

In TIMA only, the phase load register TIMA.PL provides the capability for TIMA.CTR to count from a value other than zero or TIMA.LOAD in Up/Down counting mode. Phase load is used to generate asymmetric center-aligned PWM output signals with a controlled phase shift between different timer instances.

When TIMA.PL is nonzero, phase load is enabled by setting TIMA.CTRCTL.PLEN = 1 and triggered when TIMA.CTTRIG.TRIG = 1. When phase load is triggered while TIMA.CTRCTL.CVAE = 0, the timer counts from the TIMA.PL value in the down direction. When phase load is triggered while TIMA.CTRCTL.CVAE = 0, the TIMx counts from the TIMA.PL value in the up direction.

TIMA.PL is latched when the timer starts, and TIMA.PL is synchronized every time when the counter reaches the previously latched TIMA.PL value. Figure 35-11 shows how the phase load register works when the timer is counting in the up-down direction and the phase load value changes to a new value.

**Note**

If in up or up/down mode and the phase load is larger than the load value, the timer will start at the phase load value and count up to the maximum counter value. It is recommended to have the phase load value less than the load value.



Copyright © 2026 Texas Instruments Incorporated

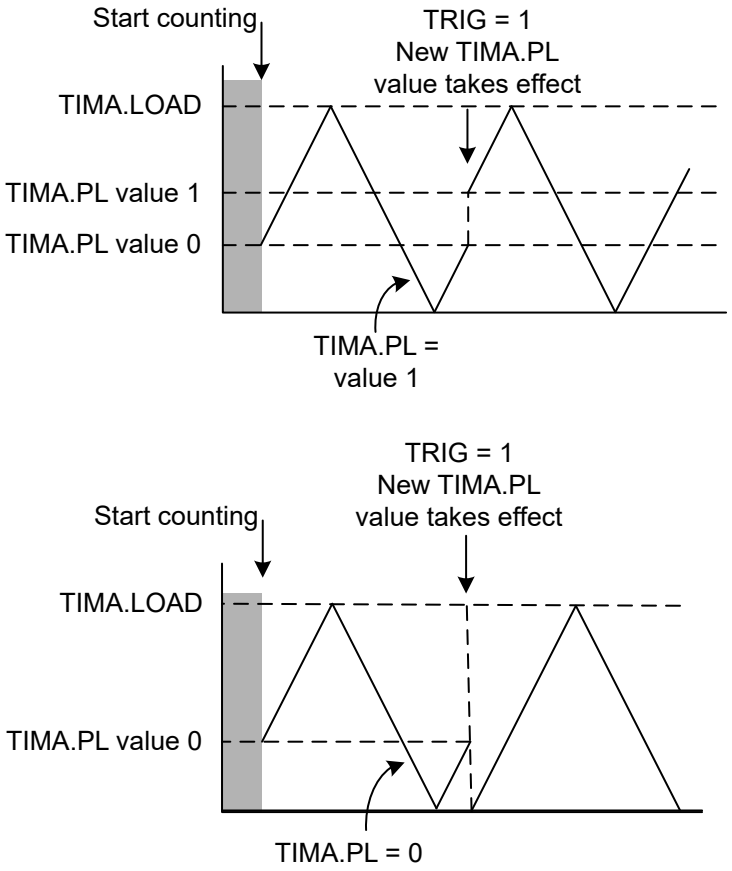



**Figure 35-11. Phase Load Register Synchronization in Up-Down Mode**

### *35.2.3 Capture/Compare Module*

The capture/compare (CC) block is used for capture events or compare events. TIMG has up to 2 identical capture/compare blocks and TIMA has up to 4 identical capture/compare blocks present to support external or internal signals. Additionally, TIMA provides 2 additional CC blocks (CC4 and CC5) for compare events only from internal signals. Any of the TIMx capture/compare blocks may be used to capture timings of an input signal or to generate time intervals.

#### 35.2.3.1 Capture Mode

Capture mode is selected when the TIMx.CCCTL_xy[0/1].COC bit is set to 1. Capture mode is used to generate capture events and record time intervals, which is useful for speed computation or time measurements.

Key registers for configuring capture mode:

- **TIMx.LOAD**: the contents of this register are copied to counter (TIMx.CTR) on any operation designated to do a "load". This value is also used to compare with the counter value for generating a "Load Event" which can be used for interrupt, trigger, or signal generation actions.
- **TIMx.CC_xy[0/1]**: this is a register that can be used as either a capture register to acquire or record the next counter value on an event, or a compare register to the current counter to create an event.
- **TIMx.CCCTL_xy[0/1]**: this register controls the operations of the respective CC (capture/compare) blocks. In capture mode, it can configure whether a rising edge or falling edge generates a load, zero, advance, or capture condition. In compare mode, it controls which sources generate different types of compare events.
- **TIMx.CTRCTL**: this register provides control over the counter operation in different conditions.
- **TIMx.IFCTL_xy[0/1]**: this register controls the input filtering (FE, FP, CPV), selection (ISEL), and inversion (INV) for the associated CC block.



Copyright © 2026 Texas Instruments Incorporated

#### 35.2.3.1.1 Input Selection, Counter Conditions, and Inversion

The TIMx.IFCTL register is used for selecting input source, filtering, and final inversion options for the capture/compare block.

Figure 35-12 shows the block diagram for the TIMx capture block with two CC channels.

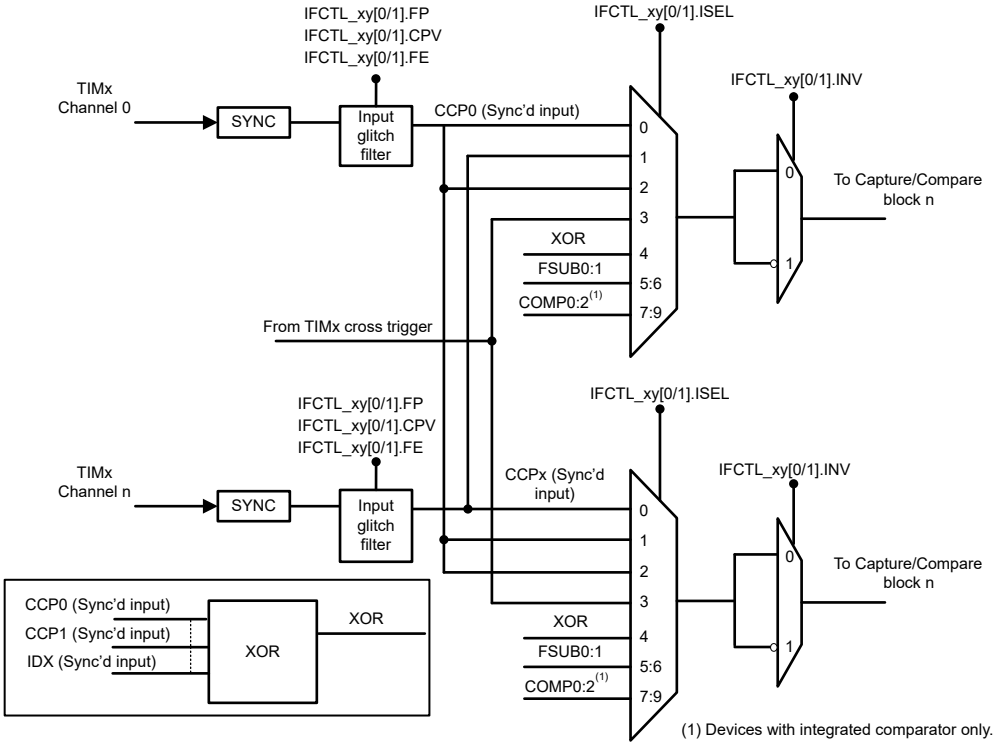



**Figure 35-12. TIMx Capture Block Diagram**

##### 35.2.3.1.1.1 CCP Input Edge Synchronization

When using a capture/compare pin (CCP) as an input, configure the pin control management register (PINCMx) for the TIMx CCP input. Refer to the device data sheet for TIMx CCP pinmux input options, such as TIMG0_C0.

The CCP input signal is always passed through a synchronizer, and the input state (high or low) must be greater than one TIMCLK clock period for the synchronizer to detect the edge. CCP input edge detection requires at least one TIMCLK cycle to synchronize the edge input. Timing in the first TIMCLK cycle is uncertain because the edge detection cannot be predicted in the first TIMCLK period.

When the capture condition occurs, an additional TIMCLK cycle is required to generate the capture event.

##### 35.2.3.1.1.2 CCP Input Pulse Conditions

The TIMx.CCCTL_xy[0/1] register can control whether each timer instance generates a zero, load, capture, or advance pulse based on the edge or polarity of the CCP input signal or trigger edge. The conditions that can be generated are:

- Advance condition (ACOND)
- Load condition (LCOND)
- Zero condition (ZCOND)
- Capture condition (CCOND)

**Advance conditions**

By default, the timer advances based on each TIMCLK. (ACOND = 0h). However, the timer can also advance based off the specified TIMx.CCCTL_xy[0/1].ACOND settings below.



Copyright © 2026 Texas Instruments Incorporated

**Table 35-6. Advance pulse condition settings (ACOND)**

| ACOND | Condition |
|---|---|
| 0h | Each TIMCLK |
| 1h | Rising edge of CCP or trigger assertion edge |
| 2h | Falling edge of CCP or trigger de-assertion edge |
| 3h | Either edge of CCP or trigger |
| 5h | CCP high or trigger assertion |

### Load, zero, and capture conditions

Load, zero, and capture pulses can be generated the LCOND, ZCOND, and CCOND condition settings below in the TIMx.CCCTL_xy[0/1] register.

**Table 35-7. Load, zero, and capture condition settings (LCOND, ZCOND, CCOND)**

| LCOND | ZCOND | CCOND | Condition |
|---|---|---|---|
| N/A | N/A | 0h | None |
| 1h | 1h | 1h | Rising edge of CCP or trigger assertion edge |
| 2h | 2h | 2h | Falling edge of CCP or trigger de-assertion edge |
| 3h | 3h | 3h | Either edge of CCP or trigger |

#### *35.2.3.1.1.3 Counter Control Operation*

To specify what CC instance controls the load, zero, or advance event, the CZC, CAC, and CLC fields are used for configuration in the TIMx.CTRCTL register.

See Table 35-8 for counter zero control settings. For example, if a timer triggers a ZCOND event in Channel 1, then CZC should be set to 1h to register that a ZCOND event in channel 1 triggers the zero event.

**Table 35-8. Counter Zero Control (CZC) settings**

| TIMx.CTRCTL.CZC | Setting |
|---|---|
| 0h | Channel 0 ZCOND event zeroes the TIMx instance |
| 1h | Channel 1 ZCOND event zeroes the TIMx instance |
| 2h | Channel 2 ZCOND event zeroes the TIMx instance (4 CC timer only) |
| 3h | Channel 3 ZCOND event zeroes the TIMx instance (4 CC timer only) |
| 4h | 2-input QEI mode. See Section 35.2.3.1.3 |
| 5h | 3-input QEI mode. SeeSection 35.2.3.1.3 |

See Table 35-9 for counter load control settings. For example, if a timer triggers a LCOND event in Channel 2, then CLC should be set to 2h to register that a LCOND event in channel 2 triggers the load event.

**Table 35-9. Counter Load Control (CLC) settings**

| TIMx.CTRCTL.CLC | Setting |
|---|---|
| 0h | Channel 0 LCOND event loads the TIMx instance |
| 1h | Channel 1 LCOND event loads the TIMx instance |
| 2h | Channel 2 LCOND event loads the TIMx instance (4 CC timer only) |
| 3h | Channel 3 LCOND event loads the TIMx instance (4 CC timer only) |
| 4h | 2-input QEI mode. See Section 35.2.3.1.3 |
| 5h | 3-input QEI mode. SeeSection 35.2.3.1.3 |

See Table 35-10 for counter advance control settings. For example, if a timer triggers a ACOND event in Channel 3, then CAC should be set to 3h to register that a ACOND event in channel 3 triggers the advance event.



Copyright © 2026 Texas Instruments Incorporated

**Table 35-10. Counter Advance Control (CAC) settings**

| TIMx.CTRCTL.CAC | Setting |
|---|---|
| 0h | Channel 0 ACOND event advances the TIMx instance |
| 1h | Channel 1 ACOND event advances the TIMx instance |
| 2h | Channel 2 ACOND event advances the TIMx instance (4 CC timer only) |
| 3h | Channel 3 ACOND event advances the TIMx instance (4 CC timer only) |
| 4h | 2-input QEI mode. See Section 35.2.3.1.3 |
| 5h | 3-input QEI mode. SeeSection 35.2.3.1.3 |

#### *35.2.3.1.1.4 CCP Input Filtering*

The input glitch filter can be enabled by setting the TIMx. IFCTL_01[0/1].FE bit. The filter period is configured by setting the TIMx. IFCTL_01[0/1].FP bit.

A consecutive period or majority voting format selected by the TIMx.IFCTL_xy[0/1].CPV bit is used to select the criteria for a CCP input signal.

- **Consecutive period** - The CCP input signal must be at the specified level for the defined number of FP timer clocks for the CCP input to be processed.
- **Majority voting** - The filter ignores one clock of opposite logic over the filter period. For example, over the number of FP samples of the input, up to 1 sample can be of an opposite logic value (glitch) without affecting the output.

The example shown in Figure 35-13 shows the difference between consecutive period and majority voting formats with a digital filter implemented to capture a CCP input signal of 3 TIMCLK periods.

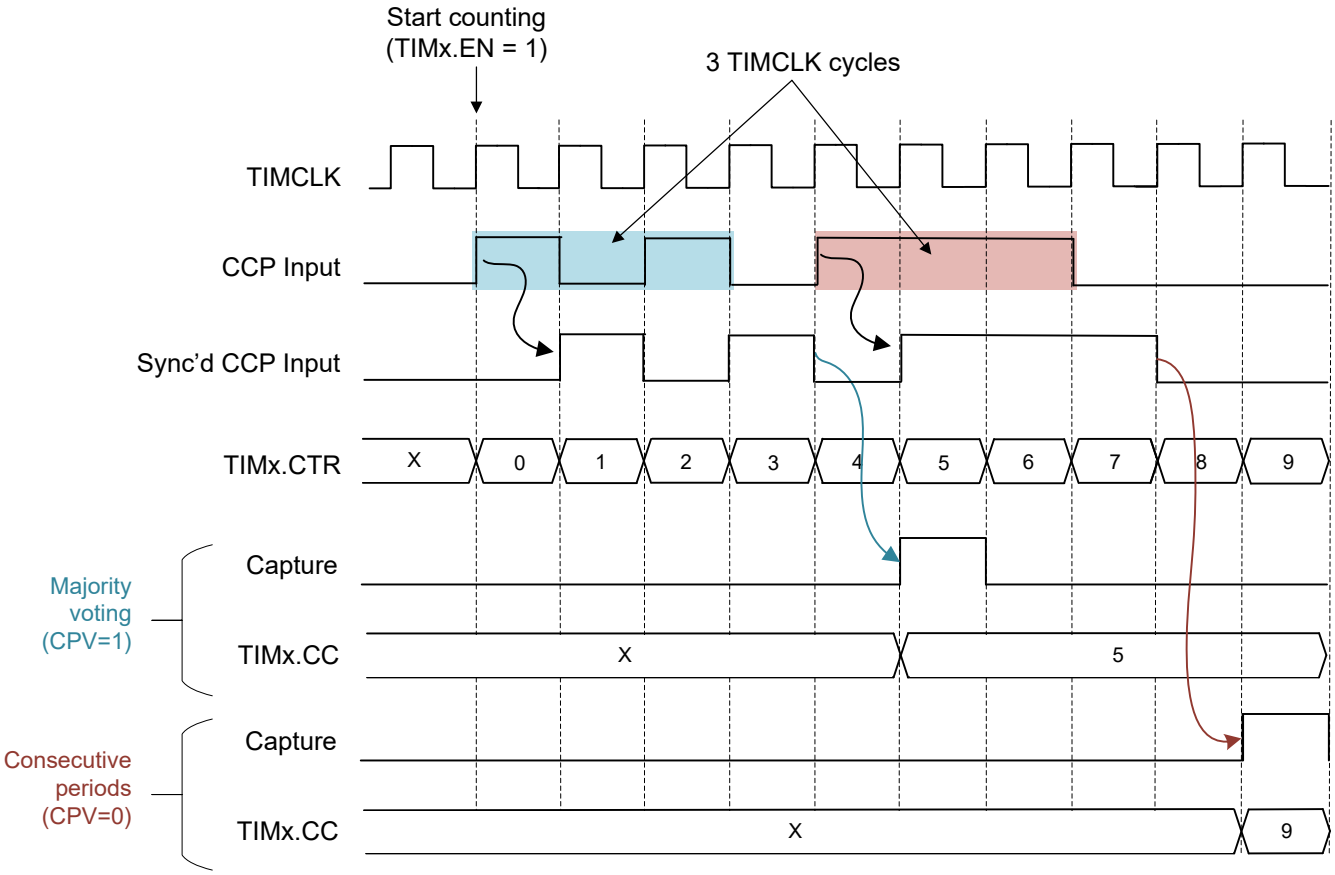



**Figure 35-13. Consecutive Period and Majority Voting for Input Glitch Filtering using FP = 0 (3 TIMCLK cycles)**



Copyright © 2026 Texas Instruments Incorporated

#### 35.2.3.1.1.5 Input Selection

The input select bits TIMx.IFCTL_xy[0/1].ISEL select the input source to the filter input as the corresponding CCP input, CCP0 for cross-triggering across CC blocks, an external trigger, XOR (used in Hall input mode), event subscribers, or comparator inputs.

**Table 35-11. Input Selection Options for TIMx CC Instances**

| Input selection (TIMx.IFCTL_xy[0/1].ISEL) | Source |
|---|---|
| 0h | TIMx CCP of the corresponding capture compare unit |
| 1h | Input pair CCPx of the capture compare unit. [CCP0/CCP1, CCP2/CCP3] |
| 2h | TIMx CCP0 |
| 3h | Cross trigger signal |
| 4h | XOR of CCP inputs. Used in Hall input mode. See Section 35.2.3.1.4 |
| 5h | Subscriber event 0 (FSUB0). See Section 35.2.9.2 |
| 6h | Subscriber event 1 (FSUB1). See Section 35.2.9.2 |
| 7h | COMP0_OUT (devices with comparator only) |
| 8h | COMP1_OUT (devices with comparator only) |
| 9h | COMP2_OUT (devices with comparator only) |

#### 35.2.3.1.2 Use Cases

Several different use cases can be achieved in capture mode and are discussed in the following sections.

#### 35.2.3.1.2.1 Edge Time Capture

Edge time capture measures the time (in TIMCLK cycles) from the start of the capture operation to the signal edge. The counter is loaded when TIMx is enabled and counts with each TIMCLK period until the CCP edge is detected, which triggers the capture of the timer value and generates a capture event. The capture edge time is equivalent to the difference between the starting value of the counter and the capture value in TIMx.CC_xy[0/1] register.

**Edge Time Capture Configuration**

1. Set the TIMx.LOAD value.
2. In the CTRCTL register, set the desired counter control settings for:
   a. Counting mode (CM) and counter value after enable (CVAE) (see as described in Section 35.2.2)
   b. Zero (CZC), advance (CAC), and load control (CLC) to specify what condition controls zeroing, advancing, or loading the counter
   c. Repeat or one-shot mode (REPEAT)
3. Set TIMx.CCCTL_xy[0/1].COC = 1 for capture mode.
4. Configure CCP as an input for the CC block by setting respective bit in the CCPD registers. For instance, if TIMx Channel 0 is an input, set CCPD.C0CCP0 = 0.
5. For the corresponding CC block control register (CCCTL_01[0/1]), set CCOND to the corresponding setting to capture events based off the input signal condition (rising and/or falling edge). Additionally, set ZCOND or LCOND depending on the counting mode used.
6. Configure input capture settings in the TIMx.IFCTL_xy[0/1] register as described in Section 35.2.3.1.1.
7. Enable the counter by setting EN = 1 or waiting for a capture event to occur from the input edge.

**Example using up-counting mode for rising edge capture**

In up-counting mode starting from zero (CM = 2, CVAE = 2), TIMx can be configured to generate a zero pulse and start the counter from the configured capture event (CCOND) by setting ZCOND to 1.

The expected internal timing for a rising or positive edge time capture in up-counting mode is shown in Figure 35-14.



Copyright © 2026 Texas Instruments Incorporated

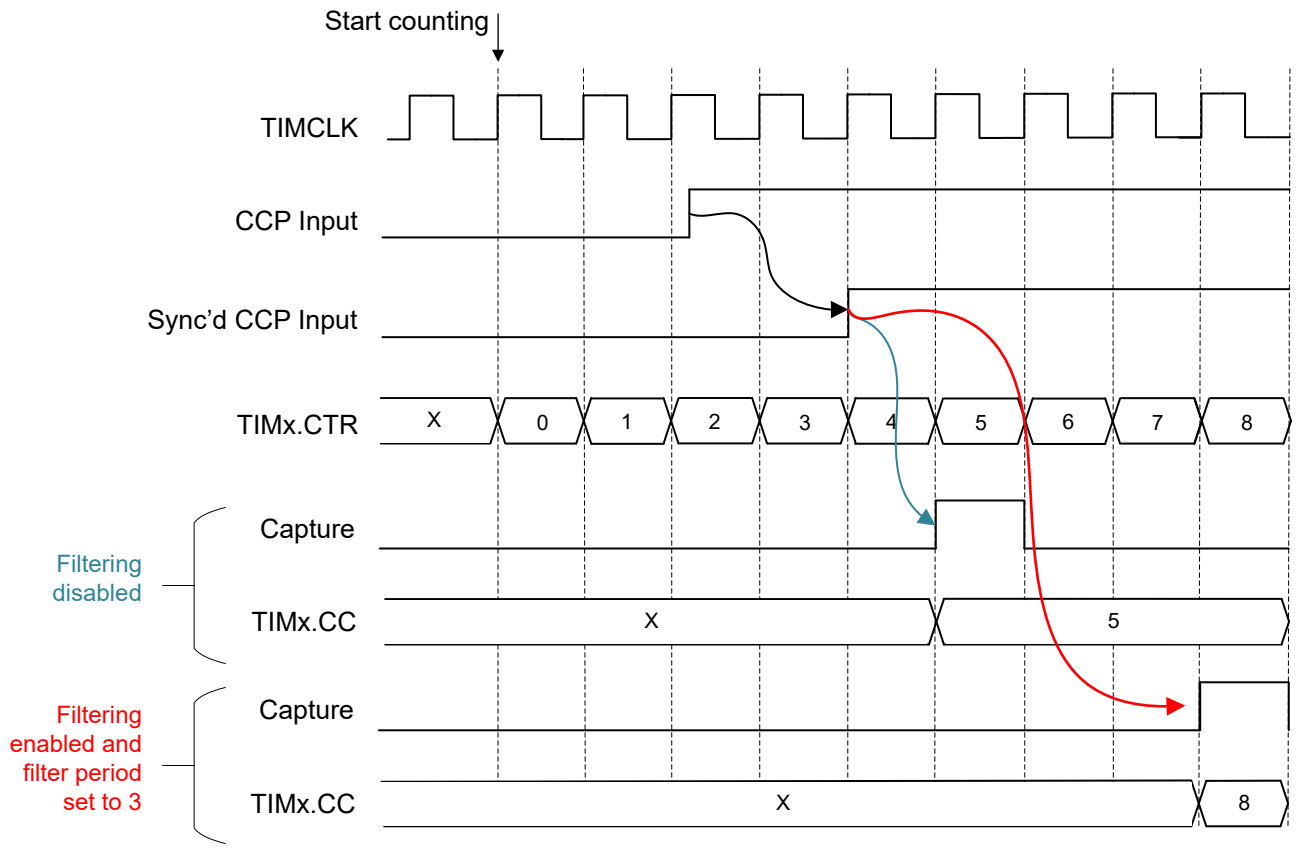



**Figure 35-14. Edge Time Capture Mode in Up-Counting Mode, CVAE = 2**

#### *35.2.3.1.2.2 Period Capture*

Period capture measures the period of a signal on an input CCP in TIMCLK cycles. On each positive (or negative) edge of the CCP input, the TIMx.CTR value is both captured into the TIMx.CC register to generate a capture event. The period capture time is equivalent to the difference between the starting value of the counter generated to the capture event, or time between reoccurring capture events for a periodic input signal.

**Period Capture Configuration**

1. Set the TIMx.LOAD value.
2. In the CTRCTL register, set the desired counter control settings for:
   a. Counting mode (CM) and counter value after enable (CVAE) (see as described in Section 35.2.2)
   b. Advance (CAC) to specify what condition controls advancing the counter
   c. Repeat or one-shot mode (REPEAT)
3. Set TIMx.CCCTL_xy[0/1].COC = 1 for capture mode.
4. Configure CCP as an input for the CC block by setting respective bit in the CCPD registers. For instance, if TIMx Channel 0 is an input, set CCPD.C0CCP0 = 0.
5. For the corresponding CC block control register (CCCTL_01[0/1]),
   a. Set CCOND to the corresponding setting to capture events based off the input signal condition (rising and/or falling edge)
   b. Set ZCOND or LCOND depending on the counting mode used
6. Configure input capture settings in the TIMx.IFCTL_xy[0/1] register as described in Section 35.2.3.1.1.
7. Enable the counter by setting EN = 1 or waiting for a capture event to occur from the input edge.

**Example using up-counting mode for rising-edge period capture**



Copyright © 2026 Texas Instruments Incorporated

In up-counting mode starting from zero (CM = 2, CVAE = 2), TIMx channel 0 can be configured to generate a capture event from a rising edge input by setting CCOND = 1h. After enabling the counter, when a rising edge input is detected, the counter will capture the counter value in TIMx.CC. After the capture event, set the TIMx.LOAD value back to 0 to reset the timer counter for the periodic CCP input signal.

The expected internal timing for a period capture in up-counting mode using two rising edges is shown in Figure 35-14.

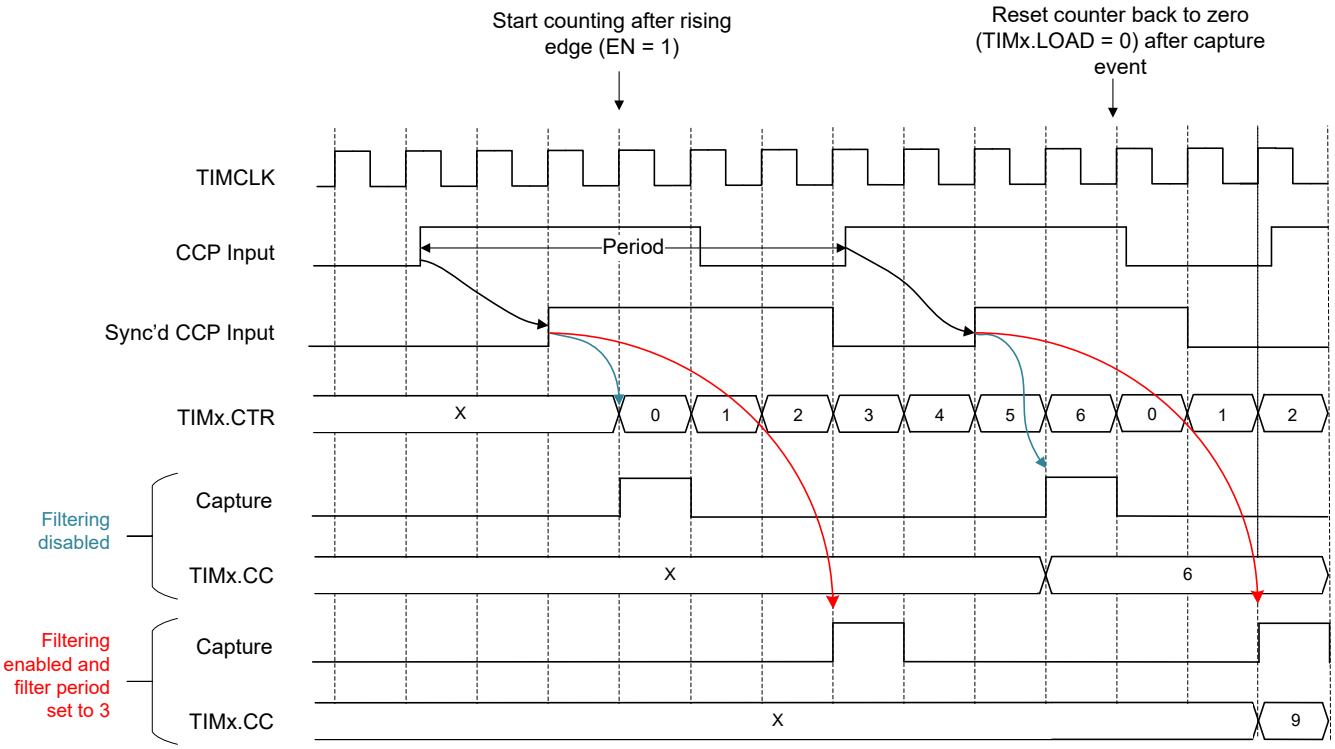



**Figure 35-15. Period Capture Mode in Up-Counting Mode, CVAE = 2**

#### *35.2.3.1.2.3 Pulse Width Capture*

Pulse width capture measures the high-time of a signal on CCP. The high time is the number of TIMCLK periods from rising edge to falling edge of the CCP input, and is useful for applications such as measuring the duty cycle of an PWM input signal. The counter is loaded at the positive edge and captured at the negative edge (capture event is generated).

**Pulse-Width Capture Configuration**

1. Set the TIMx.LOAD value.
2. In the CTRCTL register, set the desired counter control settings for:
   a. Counting mode (CM) and counter value after enable (CVAE) (see as described in Section 35.2.2)
   b. Zero (CZC), advance (CAC), and load control (CLC) to specify what condition controls zeroing, advancing, or loading the counter
   c. Repeat or one-shot mode (REPEAT)
3. Set TIMx.CCCTL_xy[0/1].COC = 1 for capture mode.
4. Configure CCP as an input for the CC block by setting respective bit in the CCPD registers. For instance, if TIMx Channel 0 is an input, set CCPD.C0CCP0 = 0.
5. For the corresponding CC block control register (CCCTL_01[0/1]), set CCOND to the corresponding setting to capture events based off the input signal condition (rising and/or falling edge). Additionally, set ZCOND or LCOND depending on the counting mode used.
6. Configure input capture settings in the TIMx.IFCTL_xy[0/1] register as described in Section 35.2.3.1.1.



Copyright © 2026 Texas Instruments Incorporated

7. Enable the counter by setting EN = 1 or waiting for a capture event to occur from the input edge.

**Example using up-counting mode for pulse width capture**

In up-counting mode starting from zero (CM = 2, CVAE = 2), TIMx channel 0 can be configured to generate a zero pulse and start the counter from the configured capture event (CCOND) by setting ZCOND to 1. To start the counter, a load condition can be triggered from the CCP rising edge input by setting LCOND = 1.

The expected internal timing for a pulse width capture in up-counting mode using a rising and falling edge is shown in Figure 35-14.

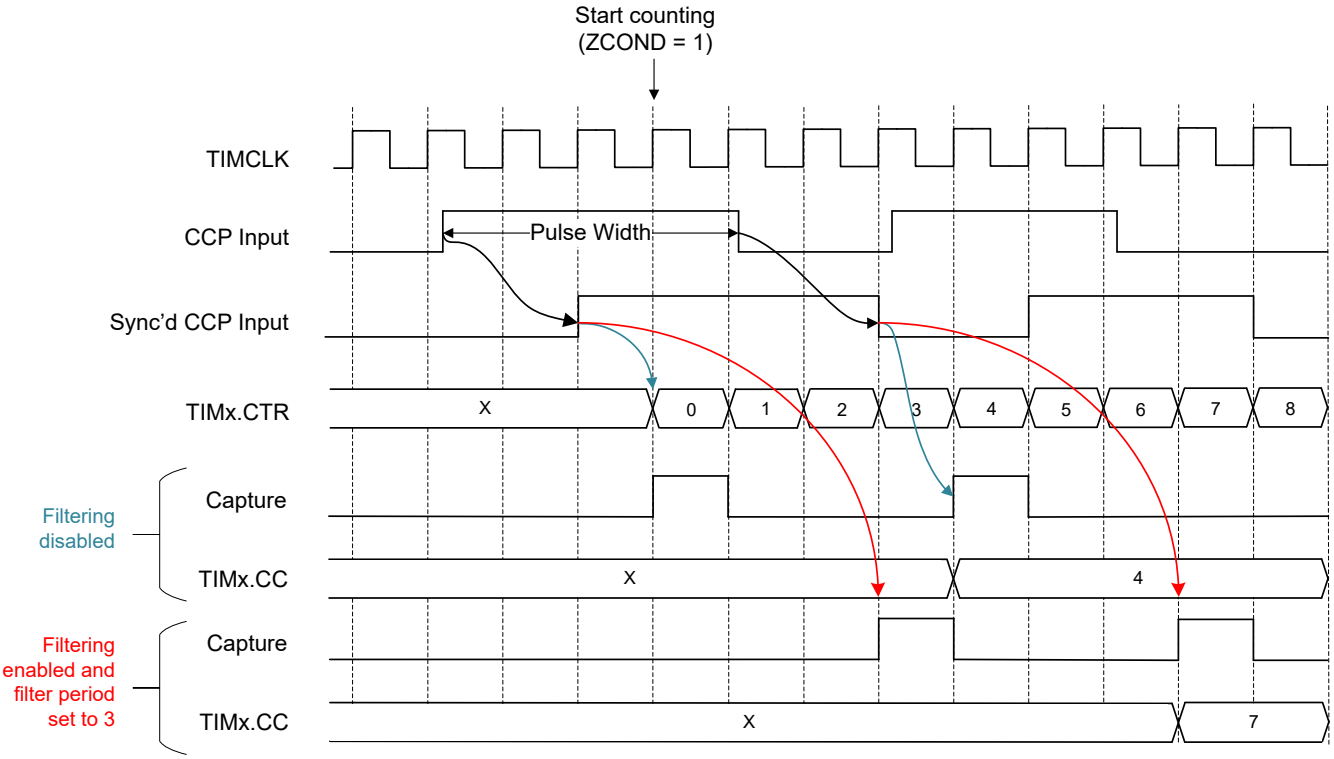



**Figure 35-16. Pulse-Width Capture Mode**

#### *35.2.3.1.2.4 Combined Pulse Width and Period Time*

Using two capture registers can combine pulse-width and period capture of a single input waveform. The input signal can be externally connected to CCP channel 0, and the IFCTL_01[1] register can be configured to have the input connected to CCP channel 1 internally so capture register 0 (TIMx.CC0) captures pulse width and capture register 1 (TIMx.CC1) captures period. The expected internal timing for combined pulse-width and period capture is shown in Figure 35-17.

**Pulse-Width and Period Capture Configuration**

1. Set the TIMx.LOAD value.
2. In the CTRCTL register, set the desired counter control settings for:
   a. Counting mode (CM) and counter value after enable (CVAE) (see as described in Section 35.2.2)
   b. Zero (CZC), advance (CAC), and load control (CLC) to specify what condition controls zeroing, advancing, or loading the counter
   c. Repeat or one-shot mode (REPEAT)
3. Set TIMx.CCCTL_xy[0/1].COC = 1 for capture mode for each CC channel.
4. Configure CCP as an input for each CC block by setting respective bits in the CCPD register. For instance, if TIMx Channel 0 is an input, set CCPD.C0CCP0 = 0.
5. For the corresponding CC block control register (CCCTL_01[0/1]),



Copyright © 2026 Texas Instruments Incorporated

a. Set CCOND to the corresponding setting to capture events based off the input signal condition (rising and/or falling edge)
b. Set ZCOND or LCOND depending on the counting mode used.

6. Configure input capture settings in the TIMx.IFCTL_xy[0/1] register as described in Section 35.2.3.1.1.
7. Enable the counter by setting EN = 1 or waiting for a capture event to occur from the input edge.

**Example using pulse-width and period time capture**

In up counting mode, TIMx can be configured to generate a zero pulse and start the counter from the configured capture event (CCOND) by setting ZCOND to 1.

The expected internal timing for a pulse-width and period capture in up-counting mode using two CC blocks is shown in Figure 35-14.

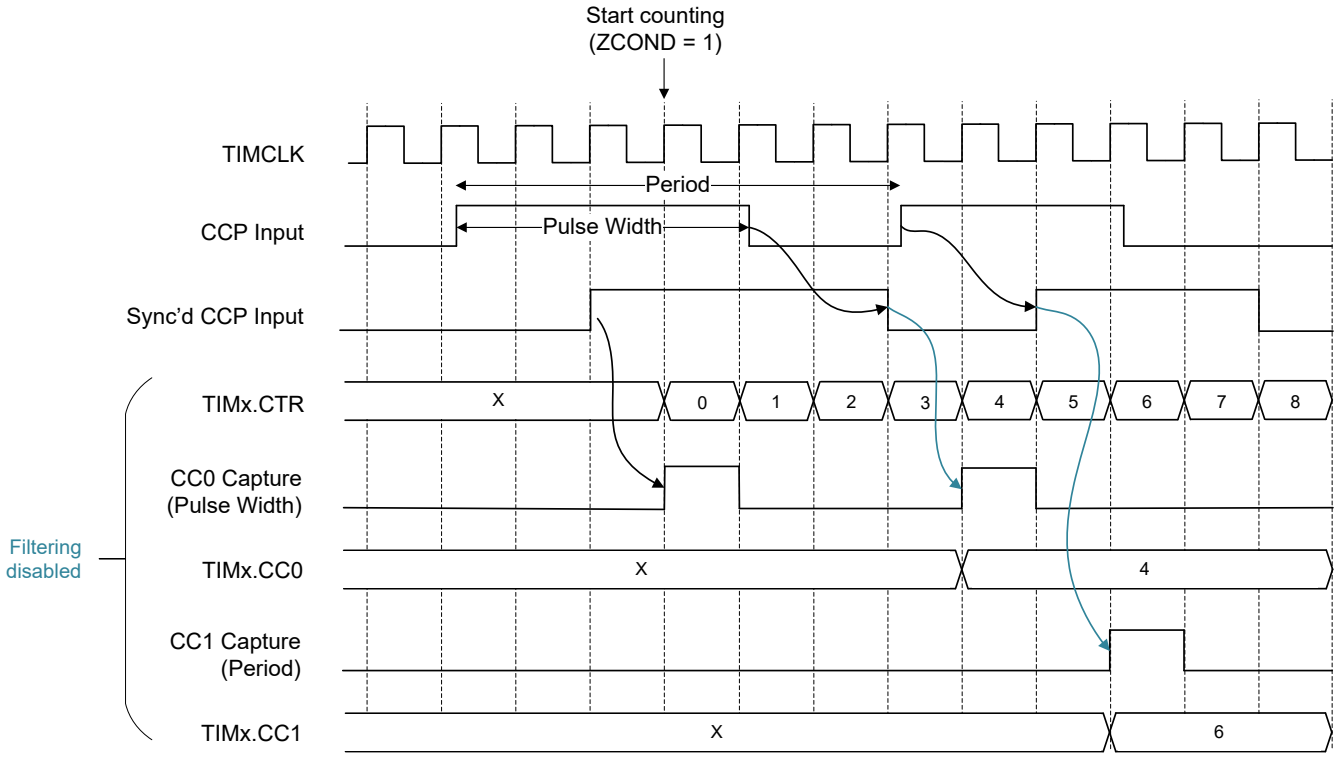



**Figure 35-17. Combined Pulse-Width and Period Capture**

***35.2.3.1.3 QEI Mode (TIMG with QEI support only)***

In TIMGx instances with QEI support, Quadrature Encoder Interface (QEI) mode provides an interface to the output of a quadrature encoder. It decodes the quadrature encoded data to provide the information on the relative positioning and movement of a linear or rotary motion.

The QEI consists of two Gray coded quadrature input signals PHA and PHB, and an index input signal IDX. All input signals go to CCP inputs of a single counter, such that PHA and PHB are mapped to CCP0 and CCP1, and IDX is brought in as a separate input. An error detection mechanism can report erroneous transitions to avoid improper signal decodings.

**Note**

See Section 35.1.3 and the device-specific data sheet for TIMG instances that support QEI / Hall Input Mode.



Copyright © 2026 Texas Instruments Incorporated

#### *35.2.3.1.3.1 QEI With 2-Signal*

QEI is used to decode signals from optical position encoders. Optical position encoders typically output 2 signals (PHA/PHB) which are often used to measure rotary or linear movement of physical items with rotating shafts, such as 3-phase motors. The measurement provided by the interface is incremental, which means as movement occurs, the interface provides the ability to capture the relative change from the previous position.

When operating in QEI mode, a counter accumulates the incremental updates, deriving current position from initial position and the accumulated change. The initial position can be determined by the signal of the IDX input (3-signal QEI mode) or by other software means (software directly setting the initial position). The capture and compare register can be used to store the position value by defining a capture condition.

When a direction change (DC) occurs, a DC interrupt is generated in the RIS register.

Figure 35-18 shows the state machine in the QEI interface to detect the directional rotation from the two CCP input signals, PHA and PHB.

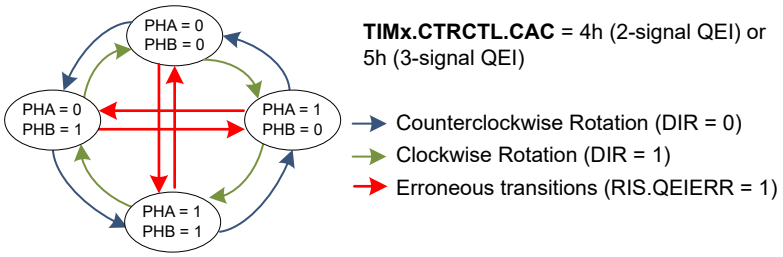



**Figure 35-18. State Machine for 2-signal and 3-signal QEI mode**

### QEI 2-Signal Mode Configuration

1. Configure PINCMx for TIMGx_C0 (PHA) and TIMGx_C1 (PHB).
2. Set TIMG.CCCTL_01[0].COC = 1 and TIMG.CCCTL_01[1].COC = 1 for capture mode for both CCP channels 0 and 1 (PHA and PHB).
3. Configure CCP as an input for each CC block by setting respective bits in the CCPD register. For instance, if TIMx Channel 0 is an input, set CCPD.C0CCP0 = 0.
4. Set the TIMx.LOAD to the encoder resolution, such as 4000.
5. In the CTRCTL register, set the CAC, CZC, and CLC bits to 4h.
6. Configure input capture settings as described in Section 35.2.3.1.1, if desired.
7. Enable the counter to start counting by setting EN = 1.

### Example using QEI mode with 2 input signals

The behavior of PHA/PHB follows Table 35-12 and Figure 35-19.

**Table 35-12. PHA/PHB State Table and Counter Actions**

| Current State (PHA, PHB) | Next State (PHA, PHB) | Direction (DIR) | Counter Action |
|---|---|---|---|
| 00 | 10 | 1 (Up) | +1 (If TIMx.CTR = LOAD, then CTR = 0) |
| 10 | 11 | | |
| 11 | 01 | | |
| 01 | 00 | | |
| 00 | 01 | 0 (Down) | -1 (If TIMx.CTR = 0, then CTR = TIMx.LOAD) |
| 01 | 11 | | |
| 11 | 10 | | |
| 10 | 00 | | |



Copyright © 2026 Texas Instruments Incorporated

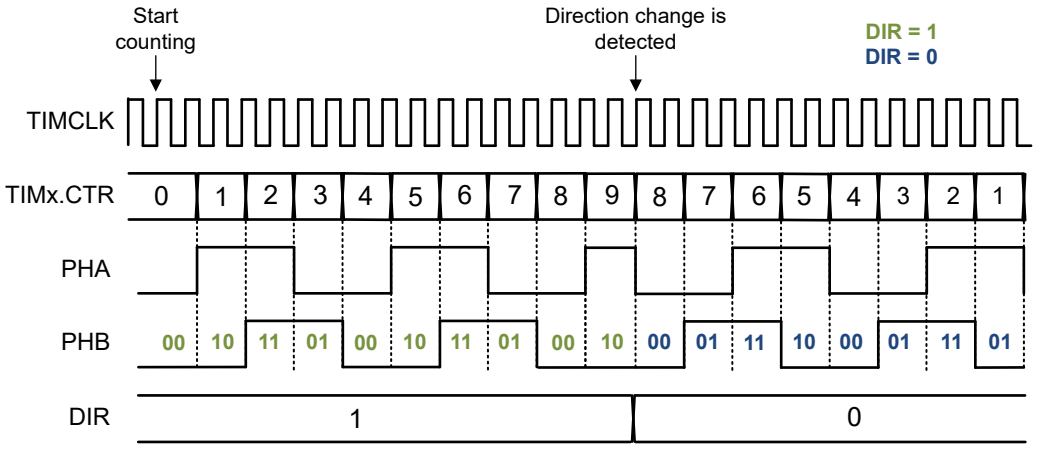



**Figure 35-19. 2-Signal QEI Operation**

#### *35.2.3.1.3.2 QEI With Index Input*

3-signal QEI mode is similar to the 2-signal mode with additional index input signal IDX. Generally, IDX input pulses once per rotation which can be used to reset the counter. It is used in one of two applications:

- One IDX pulse is generated each movement cycle. In this case, the accumulated position represents the fraction of the movement cycle.
- An IDX pulse is generated at a specific position in a noncyclic movement.

**QEI 3-Signal Mode Configuration**

1. Configure PINCMx for TIMGx_C0 (PHA), TIMGx_C1 (PHB), and TIMGx_IDX (IDX).
2. Set TIMG.CCCTL_01[0].COC = 1 and TIMG.CCCTL_01[1].COC = 1 for capture mode for both CCP channels 0 and 1 (PHA and PHB).
3. Configure CCP as an input for each CC block by setting respective bits in the CCPD register. For instance, if TIMx Channel 0 is an input, set CCPD.C0CCP0 = 0.
4. Set the TIMx.LOAD value to the encoder resolution, such as 4000.
5. In the CTRCTL register,set the CAC, CZC, and CLC bits to 5h.
6. Configure input capture settings as described in Section 35.2.3.1.1, if desired.
7. Enable the counter to start counting by setting EN = 1.

The IDX input is sampled when a rising edge is detected. The IDX signal affects the counter value depending on the direction as shown inTable 35-13 and

**Table 35-13. Relation of IDX Input and Counter Value**

| Direction (DIR) | IDX | Counter Action |
|---|---|---|
| 1 | Rising | Zero (TIMx.CTR is set to zero) |
| 0 | Rising | Load (TIMx.CTR is set to TIMx.LOAD) |



Copyright © 2026 Texas Instruments Incorporated

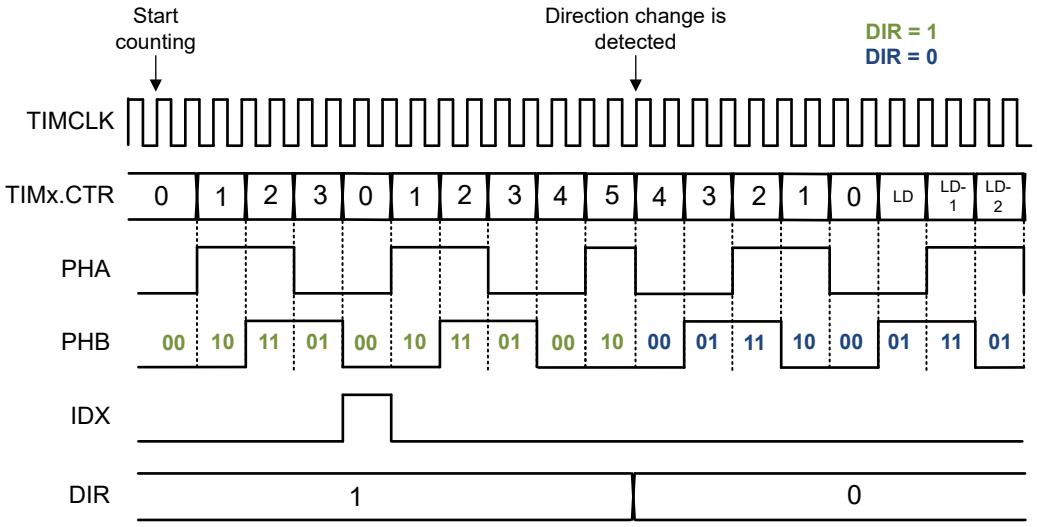



**Figure 35-20. 2-Signal QEI with Index Input Operation**

#### *35.2.3.1.3.3 QEI Error Detection*

The QEI module can detect erroneous transactions or state errors as shown in Figure 35-18. A QEIERR interrupt is generated and the counter or direction signal does not change in the error state.

**Note**

QEIERR can occur if the TIMCLK period is slower than the period of the PHA or PHB signals. In the case of PHA and PHB being asynchronous the QEIERROR can be missed, it is recommended to have the PHA and PHB synchronized.

#### *35.2.3.1.4 Hall Input Mode (TIMG with QEI support only)*

In TIMGx instances with QEI support, three digital Hall signals can be input into CCP channel 0 (CCP0), CCP channel input 1 (CCP1), and IDX for position control of 3-phase Hall-sensored motor applications. Hall signals are used to detect real-time motor position in motor control applications and can be used for speed computation measurements, position control, or motor stall status.

**Note**

See Section 35.1.3 and the device-specific data sheet for TIMG instances that support QEI / Hall Input Mode.

Table 35-14 shows the signal mapping for Hall signals A (U), B (V), and C (W) to TIMG capture/compare input signals.

**Table 35-14. Hall Input and TIMx Input Signal Mapping**

| Hall input signal | TIMx input |
|---|---|
| HALL A / HALL U | CCP0 |
| HALL B / HALL V | CCP1 |
| HALL C / HALL W | IDX |

**Note**

Hall input signals should be digital inputs from Hall sensor ICs with a pullup to VCC.



Copyright © 2026 Texas Instruments Incorporated

As shown in Figure 35-21, the input capture module provides a 3-input XOR of synced CCP0, CCP1, and IDX signals to create a frequency generator (FG) signal. The XOR output signal is selected when IFCTL_xy[0/1].ISEL is set to 4h. See Figure 35-21 for XOR option in the input capture block diagram.

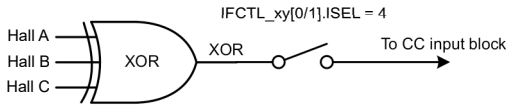



**Figure 35-21. Hall 3-input XOR for Frequency Generator (FG) Signal to CC input block**

The XOR'ed output signal is propagated to the CC block and a period or pulse-width capture can be used to compute the linear motor speed in relation to the calculated period or pulse width in the TIMx.CC register. See Section 35.2.3.1.2.2 and Section 35.2.3.1.2.3 on how to calculate period and pulse-width captures based on the XOR'ed input signal.

Figure 35-22 shows the input signal to the CC block which can be used for speed calculations.

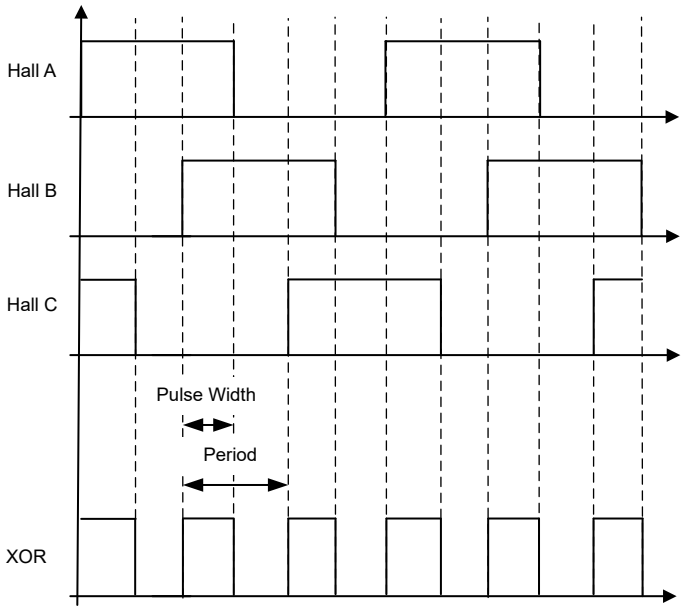



**Figure 35-22. Hall 3-Input XOR Signal used with Pulse Width or Period Capture for Speed Computation**

### Hall Input Mode Configuration

1. Configure PINCMx for TIMGx_C0 (HALLA), TIMGx_C1 (HALLB), and TIMGx_IDX (HALLC).
2. Set TIMG.CCCTL_01[0].COC = 1 and TIMG.CCCTL_01[1].COC = 1 for capture mode for both CCP channels 0 and 1 (HALLA and HALLB).
3. Configure CCP as an input for each CC block by setting respective bits in the CCPD register. For instance, if TIMx Channel 0 is an input, set CCPD.C0CCP0 = 0.
4. Set the TIMx.LOAD value.
5. Set TIMG.IFCTL_xy[0/1].ISEL = 4h to select the XOR option for Hall signals.
6. Enable the counter to start counting by setting EN = 1.

#### 35.2.3.2 Compare Mode

Compare mode is selected when TIMx.CCCTL_xy[0/1].COC = 0. Compare mode is used to generate a compare event to output PWM output signals at specific time intervals. Compare events can be used to generate a timing base internally or generate a PWM output with specific profiles using active, inactive, or toggle action behaviors, and optional deadband insertion.



Copyright © 2026 Texas Instruments Incorporated

Many types of compare mode events can be generated based on the configuration of the CC action register CCACT_xy[0/1]:

- Compare events: occurs for a CC channel when TIMx.CTR counts up (CCU) or down (CCD) to the value in TIMx.CC_xy[0/1]
- Secondary compare events: occurs when a secondary CCx block compare up (CC2U) or down (CC2D) is configured for another CCx block's CCU event or CCD event, respectively. This enables more flexible output generation from other external events such as comparator outputs or fault signals. It can be useful for real-time control applications, like digital power or motor control.

In TIMA only, an additional internal 5th and 6th CC block (TIMA.CC_45[0/1]) can be used for internal secondary compare events while continuing to use CC channels with dedicated output pins for external PWM signal generation.

Table 35-15 shows the types of compare mode events that can be generated and conditions to generate the events.

**Table 35-15. Compare Mode Events**

| Event | Name | Event Condition |
|---|---|---|
| CCDn (n = CC channel) | Capture/compare down event | When timer is counting down, TIMx.CTR = TIMx.CC_xy[0/1] |
| CCUn (n = CC channel) | Capture/compare up event | When timer is counting up, TIMx.CTR = TIMx.CC_xy[0/1] |
| CC2Dn (n = CC channel) | Secondary capture/compare down event | Occurs when CC2SELD is configured for the source of the CCDn event |
| CC2Un (n = CC channel) | Secondary capture/compare up event | Occurs when CC2SELU is configured for the source of the CCUn event |

**Note**

Look at the device specific data sheet to check how many CC channels are available in each TIMx instance on the device.

#### *35.2.3.2.1 Edge Count*

In addition to event or PWM output generation, compare mode can also be used for input signal edge counting to determine when a number of edges has been detected. In edge count operation, a CCP input edge can advance the counter based on the ACOND condition. The counter register is initialized with the starting value, and the number of detected CCP input edges at any time can increment or decrement depending on the counting mode configuration. The user can count rising edges, falling edges, or both edges by configuring the CCOND value.

**Edge Count Configuration**

1. Set TIMx.CCCTL_xy[0/1].COC = 0 for compare mode.
2. Optionally set the corresponding TIMx.CC_xy[0/1] to a compare value to generate a compare interrupt when the counter reaches this value.
3. In the CTRCTL register, set the desired counter control settings for:
   a. Counting mode (CM) and counter value after enable (CVAE) (see as described in Section 35.2.2)
   b. Zero (CZC), advance (CAC), and load control (CLC) to specify what condition controls zeroing, advancing, or loading the counter
   c. Repeat or one-shot mode (REPEAT)
4. Set ACOND to a setting to advance the counter based on the input edge polarity.
5. Configure input capture settings as described in Section 35.2.3.1.1, if desired.
6. Enable the counter by setting EN = 1.

**Example using edge count operation using up-counting mode**



Copyright © 2026 Texas Instruments Incorporated

In up-counting mode starting from zero (CM = 2, CVAE = 2), the expected internal timing for rising edge count operation to increment the counter is shown in Figure 35-23.

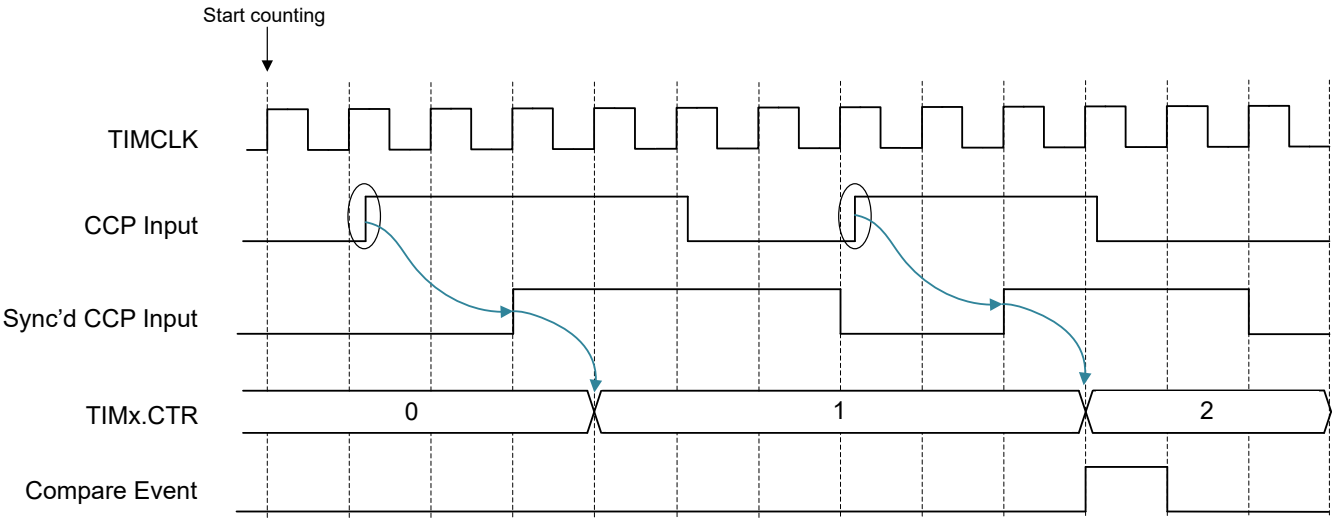



**Figure 35-23. Edge Count Operation to Generate Compare Event (TIMx.CC = 2)**

### *35.2.4 Shadow Load and Shadow Compare*

Some timer modules have a shadow load and shadow compare register feature which gives the user the flexibility of holding the update of load and CC values until a certain event occurs. This is useful in timing-critical applications where PWM control signals need to be updated with correct timings, such as duty cycle updates. Refer to the timer features for specific configurations of the TIMx modules.

**Note**

See Section 35.1.3 and the device-specific data sheet for TIMx instances that support Shadow Load and Shadow Compare.

#### 35.2.4.1 Shadow Load (TIMG4-7, TIMA only)

On shadow-load capable timers, the shadow load feature allows holding the update of load values until a zero event occurs. To enable shadow loading, set the TIMx.GCTL.SHDWLDEN bit while the timer is enabled (EN = 1).

If the TIMx module has a shadow load feature, there is an internal shadow register for the load value (TIMx.LOAD). The shadow register will update the load value at a zero event as shown in Figure 35-24.

**Note**

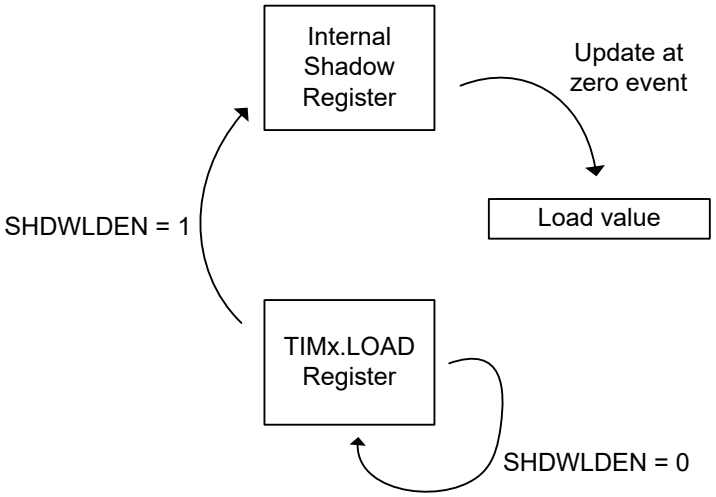
On shadow-load capable timers, SHDWLDEN must be set or else the load value will not update.

If the timer instance does not have shadow load capability (standard timer), the load value will update immediately when TIMx.LOAD is written to.



Copyright © 2026 Texas Instruments Incorporated

Shadow-load capable timers update the load value from the internal shadow register when:
- TIMx.CTRCTL.EN = 1
- GCTL.SHDWLDEN = 1
- Available for TIMG4-7, TIMAx




Standard timers update the load value from TIMx.LOAD immediately when:
- TIMx.CTRCTL.EN = 1
- Available for TIMG0-3, TIMG8-14

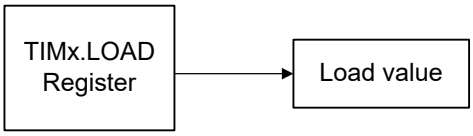



**Figure 35-24. Load value updates for TIMx instances when EN=1**

**Note**

To update the load value for shadow load capable timers, when SHDWLDEN = 0, the timer must be disabled before changing the TIMx.LOAD value.

When TIMx.GCTL.SHDWLDEN = 1, load values update at zero events for all counting modes. Consult the counting mode operations below to determine if a shadow load is needed:

- In down-counting modes, since TIMx.LOAD value is updated when a zero event occurs, a shadow load is not needed in these modes.
- In up/down counting mode, TIMx.LOAD is compared with the counter value to determine if the peak is reached and when to start to counting down. A shadow load is necessary to ensure that TIMx counts up to the load value before the zero event, or else the load value can update immediately and cause incorrect timings.
- In up-counting mode, the timer counts to TIMx.LOAD. A shadow load is necessary to ensure that TIMx counts up to the load value before the zero event, or else the load value can update immediately and cause incorrect timings.

Figure 35-25 shows an example of how shadow load and shadow compare takes effect at the zero event for both the TIMx.LOAD and TIMx.CC value in up/down counting mode.

#### 35.2.4.2 Shadow Compare (TIMG4-7, TIMG12-13, TIMA only)

When shadow compare is enabled for updating the capture/compare register (TIMx.CC), the value written to the respective compare register is first stored into a shadow compare register and then transferred to the compare register at different events configured by setting the TIMx.CCCTL_xy[0/1].CCUPD bits.

Additionally, the capture/compare action register (TIMx.CCACT) has the ability to update the action at different events configured by setting the TIMx.CCCTL_xy[0/1].CCACTUPD bits.

Table 35-16 shows the settings for configuring when shadow compare and actions occur at different events.



Copyright © 2026 Texas Instruments Incorporated

**Note**

For registers with shadow features (LOAD, CC_xy, CCACT_xy) the shadow behavior (GCTL.SHDWLDEN, CCCTL_xy[0/1].CCUPD, CCCTL_xy[0/1].CCACTUPD) should be set before the intended register.

For example, the CCUPD default function is to update the capture compare value immediately, if this is not intended, set the CCUPD before setting the CC_xy[0/1] register.

Setting CCUPD **after** setting capture compare value will cause the CC_xy[0/1] registers to load the default reset value after the CCUPD event, because the capture compare value was written directly to the register instead of to its shadow.

**Table 35-16. Shadow Compare and Action Update Behavior**

| Bit Field | Value | Description/Comment |
|---|---|---|
| CCUPD / CCACTUPD | 0 | The value written to TIMx.CC register take effect immediately. |
| | 1 | The value written to the TIMx.CC register is stored in a shadow compare register and gets transferred to the TIMx.CC register in the TIMCLK cycle following a zero event (TIMx.CTR value equals 0). |
| | 2 | The value written to the TIMx.CC register is stored in a shadow compare register and gets transferred to the TIMx.CC register in the TIMCLK cycle following a compare (down) event (TIMx.CTR value equals TIMx.CC) |
| | 3 | The value written to the TIMx.CC register is stored in a shadow compare register and gets transferred to the TIMx.CC register in the TIMCLK cycle following a compare (up) event (TIMx.CTR value equals TIMx.CC) |
| | 4 | The value written to the TIMx.CC register is stored in a shadow compare register and gets transferred to the TIMx.CC register in the TIMCLK cycle following a zero or load event (TIMx.CTR value equals 0 or TIMx.CTR equals TIMx.LOAD).<br>**Note: this update mechanism is defined for use only in up/down counting mode.** |
| | 5 | The value written to the TIMx.CC register is stored in a shadow compare register and gets transferred to the TIMx.CC register in the TIMCLK cycle following a zero event and the repeat count equaling zero (TIMx.CTR value equals 0 and TIMx.RC equals 0) |
| | 6 | The value written to the TIMx.CC register is stored in a shadow compare register, and gets transferred to the TIMx.CC register in the TIMCLK cycle following a trigger pulse.<br>See Section 35.2.7. |

Figure 35-25 shows an example of how shadow load and shadow compare takes effect at the zero event for both the TIMx.LOAD and TIMx.CC value in up/down counting mode.



Copyright © 2026 Texas Instruments Incorporated

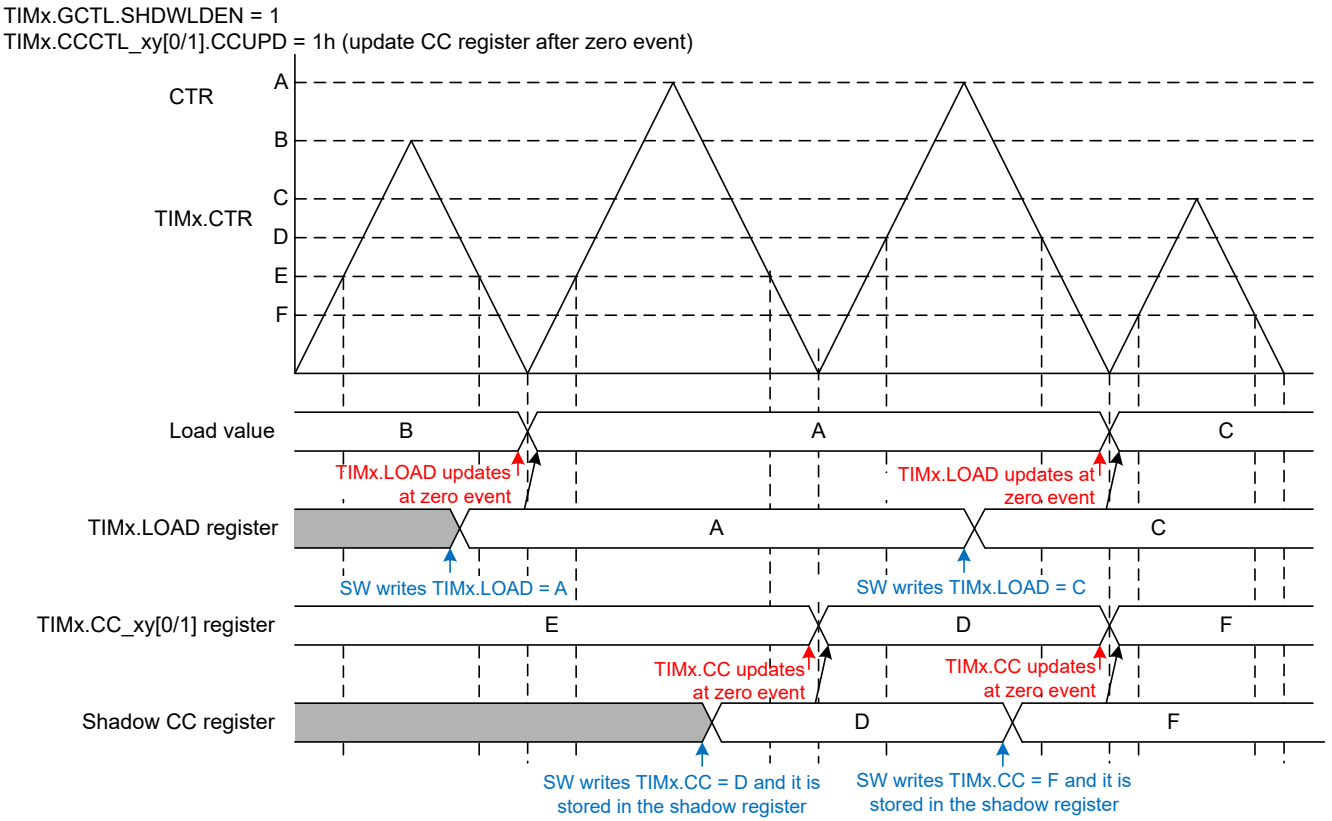



**Figure 35-25. Shadow Load and Shadow Compare Taking Effect at Zero Event in Up/Down Mode**

### *35.2.5 Output Generator*

The output signal generation unit can be used with the counter and capture/compare modules to generate desired pulse-width modulation (PWM) output waveforms, event signals, synchronized capture inputs, or the counter direction. Many output waveforms are generated from counter events (load, zero, counter direction) and the capture/compare block (compare match).

TIMA and TIMG have many common features in the output generation signal unit. Additionally, TIMA has advanced output generation features such as complimentary output signals, deadband insertion, and fault generation.

Figure 35-27 shows the TIMG output block diagram.

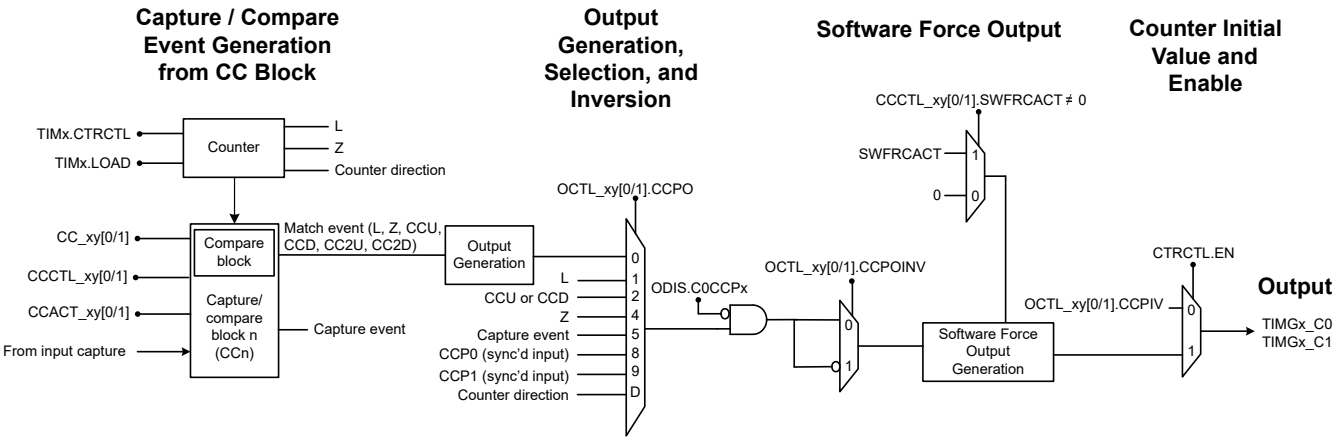



**Figure 35-26. Output Connection for TIMG**



Copyright © 2026 Texas Instruments Incorporated

Figure 35-27 shows the TIMA output block diagram.

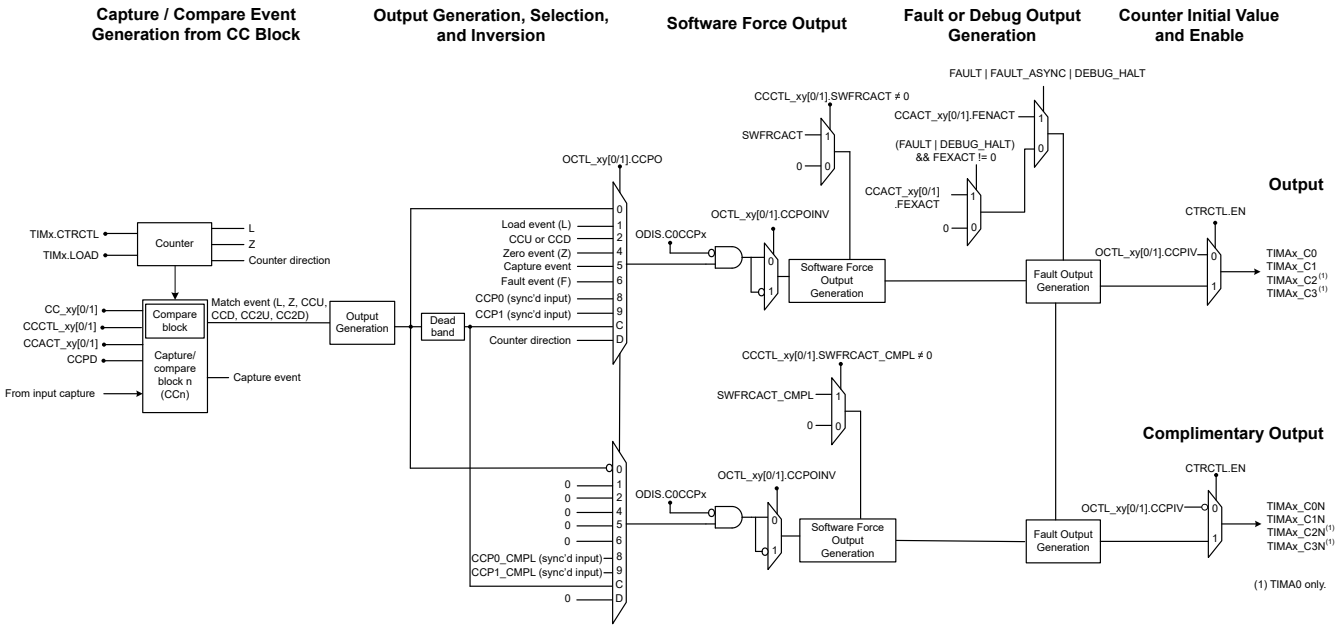



**Figure 35-27. Output Connection for TIMA**

### Signal Generator Actions

Table 35-17 shows the types of signal generator actions capable by the output generator. Signal generator actions are configured in the CCACT_xy[0/1] register for zero, load, and compare events. For types of compare events, see Table 35-15.

**Table 35-17. Signal Generator Actions from Compare Event**

| Value | Action |
|---|---|
| 0h | Event is disabled and a lower priority event is selected if asserting |
| 1h | CCP output value is set high |
| 2h | CCP output value is set low |
| 3h | CCP output value is toggled |

The key registers for generation of output signals are:

- **LOAD**: the contents of this register are copied to the counter (TIMx.CTR) on any operation designated to do a "load". This value is also used to compare with the counter value for generating a "Load Event" that can be used for interrupt, trigger, or signal generator actions.
- **CCPD:** this register configures the direction of the CCP pins as inputs or outputs.
- **CC_xy[0/1]**: this is a register used as a compare value to the current counter to create an match event.
- **CTRCTL**: this register provides control over the counter operation in different conditions.
- **CCCTL_xy[0/1]**: this register controls the operations of the respective CC registers and the counter
- **OCTL_xy[0/1]**: this register controls the output of the capture-compare portion of the counter. This includes the ability to select the source of what is driven out along with initial condition values and final inversion options.
- **CCACT_xy[0/1]**: this register controls the actions of the signal generator of the capture-compare portion based on the events created in the counter block, the capture and compare block, and debug events.
- **ODIS:** this register disables the output signal selected by OCTL.CCPO (before conditional inversion) to allow software the ability to hold the CCP output low during configuration or shutdown.

These are key registers for configuring the compare mode to generate PWM signals:



Copyright © 2026 Texas Instruments Incorporated

#### 35.2.5.1 Configuration

There are five stages to configuring output signal generation in TIMx devices:

- Counter and CC Block Event Generation
- Output Generation, Selection and Inversion
- Software Force Output
- Fault Output Generation (TIMA only)
- Counter Initial Value and Enable

**Counter and CC Block Event Generation**

The counter block contains the counter and produces a load event (L), zero event (Z), and direction of counting based on the counting mode used.

The CC blocks contain the CC register and can generate two types of output signals: compare match events and capture events. Please see Table 35-15 for the compare events that can be generated.

**Output Generation, Selection and Inversion**

The TIMx.CCACT register specifies the waveform generation of a CCP output depending on the counting mode and counter compare actions.

TIMx.OCTL_xy[0/1].CCPO controls the CCP output selection from the output generation unit, output generation unit with deadband (TIMA only), counter events, compare events, capture events, fault events, or signal inputs. The output disable register (ODIS) can optionally disable the CCP output to optionally hold the CCP output low during configuration or shutdown. TIMx.OCTL_xy[0/1].INV controls final inversion options.

**Note**

Mux selections for synchronized inputs are tied to 0 for TIMAx CC2 and CC3 instances. Do not use TIMAx.OCTL_23[0/1].CCPO = 8 or 9.

On TIMA devices only, CCP complimentary output channels can be generated from the output generation unit (denoted by "N" in the signal name). For instance, TIMA0 channel 2 (TIMA0_C2) can also produce a complimentary output (TIMA0_C2N). The CCPO and INV bits also controls the selection and inversion options for the complementary output.

Complimentary outputs with deadband insertion are a common use case for inverter-based applications with half-bridge topologies. For more information, please see Section 35.2.5.2.4.

**Software Force Output**

The output of the signal generator can be overwritten in software by setting CCCTL_xy[0/1].SWFRCACT to a nonzero setting. For TIMA devices only, the complementary output of the signal generator can be overwritten by setting CCCTL_xy[0/1].SWFRCACT_CMPL to a nonzero setting.

For more information, see Section 35.2.5.3.

**Fault / Debug Output Generation (TIMA only)**

On TIMA devices, the CCP output can be overwritten after the software force output block if there is a system fault (FAULT), fault condition upon exit (FEXACT), fault condition upon entry (FENACT), an asynchronous fault (FAULT_ASYNC), or the debugger is halted (DEBUG_HALT).

For more information, see Section 35.2.6 and Section 35.2.10.

**Counter Compare Initial Value and Enable**

To specify an initial value for the CCP output while the counter is disabled, set OCTL_xy[0/1].CCPIV to 0 for a low value or 1 for a high value. This is useful for applications where CCP outputs need to be in a default state before enabling the counter.

To enable the counter, set TIMx.CTRCTL.EN to 1.



Copyright © 2026 Texas Instruments Incorporated

#### 35.2.5.2 Use Cases

Several different use cases can be achieved with the output generator and are discussed in the following sections.

##### *35.2.5.2.1 Edge-Aligned PWM*

To generate edge-aligned PWMs, TIMx can be configured for up- or down-counting mode and the PWM period in TIMCLK cycles is TIMx.LOAD + 1. The waveform uses load, zero, and compare events to drive the CCPx output high or low depending on the configuration settings of the compare/capture block and counter.

**Edge-Aligned PWM Configuration**

To generate edge-aligned PWMs using compare match events from the counter:

1. In the TIMx.CTRCTL register, set the desired counter control settings for:
   a. Up-counting (CM = 2) or down-counting mode (CM = 0) and counter value after enable (CVAE) (see as described in Section 35.2.2)
   b. Zero (CZC), advance (CAC), and load control (CLC) to specify what condition controls zeroing, advancing, or loading the counter
   c. Repeat or one-shot mode (REPEAT)
2. Set the TIMx.LOAD value to configure the PWM period.
3. Set the TIMx.CC_xy[0/1] value to configure the duty cycle.
4. Set TIMx.CCCTL_xy[0/1].COC = 0 for compare mode.
5. Configure CCP as an output for the CC block by setting respective bit in the CCPD registers. For instance, if TIMx Channel 0 is an output, set CCPD.C0CCP0 = 1.
6. In TIMx.CCACT_xy[0/1], set the CCP output action settings for compare events, zero events, load events, software force action, or fault events (TIMA only).
7. In TIMx.OCTL_xy[0/1], set CCPO = 0 to select the signal generator output.
8. Configure polarity of the signal using the CCPOINV bit, and configure CCPIV to specify the CCP output state while disabled.
9. Enable the counter by setting TIMx.CTRCTL.EN = 1.

**Example using edge-aligned PWM in down-counting mode**

A typical 2-channel edge-aligned PWM generation for down-counting mode is shown in Figure 35-28 with the following edge-aligned PWM output waveforms:

- CCP0 output generates:
  - High pulse-width from TIMx.LOAD to TIMx.CC0 value (LACT = 1h)
  - Low pulse-width from TIMx.CC0 value to zero (CDACT = 2h)
- CCP1 output generates:
  - High pulse-width from TIMx.LOAD to TIMx.CC1 value (LACT = 1h)
  - Low pulse-width from TIMx.CC1 value to zero (CDACT = 2h)



Copyright © 2026 Texas Instruments Incorporated

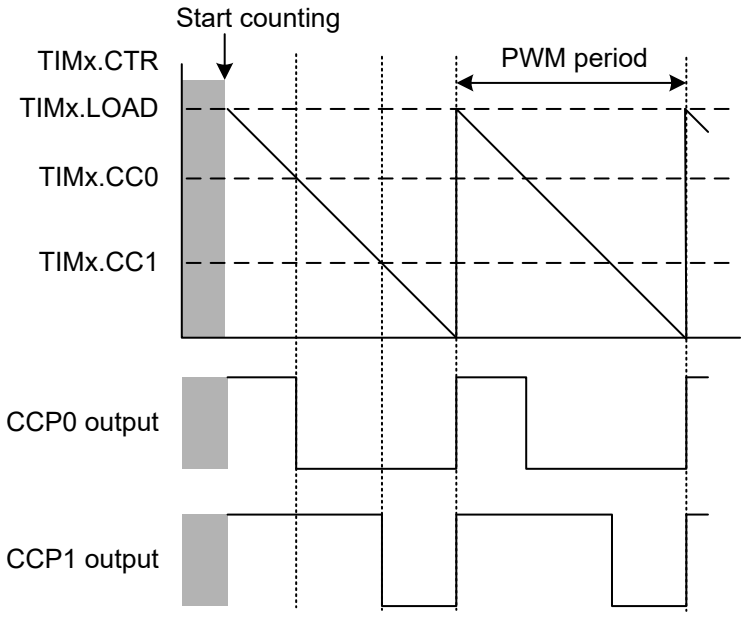



**Figure 35-28. Edge-Aligned PWM Signals in Down-Counting Mode**

### Example using edge-aligned PWM in up-counting mode

A typical 2-channel edge-aligned PWM generation for up-counting mode is shown in Figure 35-28 with the following edge-aligned PWM output waveforms:

- CCP0 output generates:
  - High pulse-width from zero to TIMx.CC0 value (ZACT = 1h)
  - Low pulse-width from TIMx.CC0 value to TIMx.LOAD (CUACT = 2h)
- CCP1 output generates:
  - High pulse-width from zero to TIMx.CC1 value (ZACT = 1h)
  - Low pulse-width from TIMx.CC1 value to TIMx.LOAD (CUACT = 2h)

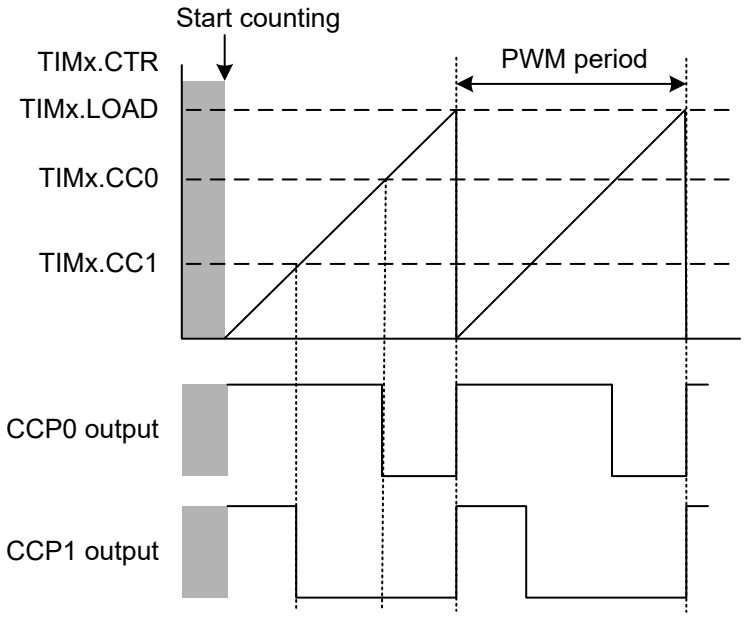



**Figure 35-29. Edge-Aligned PWM Signals in Up-Counting Mode**



Copyright © 2026 Texas Instruments Incorporated

#### 35.2.5.2.2 Center-Aligned PWM

To generate center-aligned PWMs, TIMx is configured for up/down counting mode and the TIMx.LOAD value contains the half-period. The waveform uses up compare events and down compare events to drive the CCPx output high or low depending on the configuration settings of the compare/capture block and counter.

In TIMCLK cycles, the PWM period is (2 * TIMx.LOAD) and the duty cycle is 1 - (TIMx.CC_xy[0/1] / TIMx.LOAD).

**Center-Aligned PWM Configuration**

To generate center-aligned PWMs using compare match events from the counter:

1. In the TIMx.CTRCTL register, set the desired counter control settings for:
   a. Up/down counting mode (CM = 1) and counter value after enable (CVAE) (see as described in Section 35.2.2)
   b. Zero (CZC), advance (CAC), and load control (CLC) to specify what condition controls zeroing, advancing, or loading the counter
   c. Repeat or one-shot mode (REPEAT)
2. Set the TIMx.LOAD value to configure the PWM period.
3. Set the TIMx.CC_xy[0/1] value to configure the duty cycle.
4. Set TIMx.CCCTL_xy[0/1].COC = 0 for compare mode.
5. Configure CCP as an output for the CC block by setting respective bit in the CCPD registers. For instance, if TIMx Channel 0 is an output, set CCPD.C0CCP0 = 1.
6. In TIMx.CCACT_xy[0/1], set the CCP output action settings for compare events, zero events, load events, software force action, or fault events (TIMA only).
7. In TIMx.OCTL_xy[0/1], set CCPO = 0 to select the signal generator output.
8. Configure polarity of the signal using the CCPOINV bit, and configure CCPIV to specify the CCP output state while disabled.
9. Enable the counter by setting TIMx.CTRCTL.EN = 1.

**Example using center-aligned PWM in up/down counting mode**

A typical 2-channel center-aligned PWM generation using up/down counting mode is shown in Figure 35-30 with the following center-aligned PWM output waveforms:

- CCP0 output generates:
  - High pulse-width from TIMx.CC0 compare up event to TIMx.CC0 compare down event (CUACT = 1h)
  - Low pulse-width from TIMx.CC0 compare down event to TIMx.CC0 compare up event (CDACT = 2h)
- CCP1 output generates:
  - High pulse-width from TIMx.CC0 compare up event to TIMx.CC0 compare down event (CUACT = 1h)
  - Low pulse-width from TIMx.CC0 compare down event to TIMx.CC0 compare up event (CDACT = 2h)



Copyright © 2026 Texas Instruments Incorporated

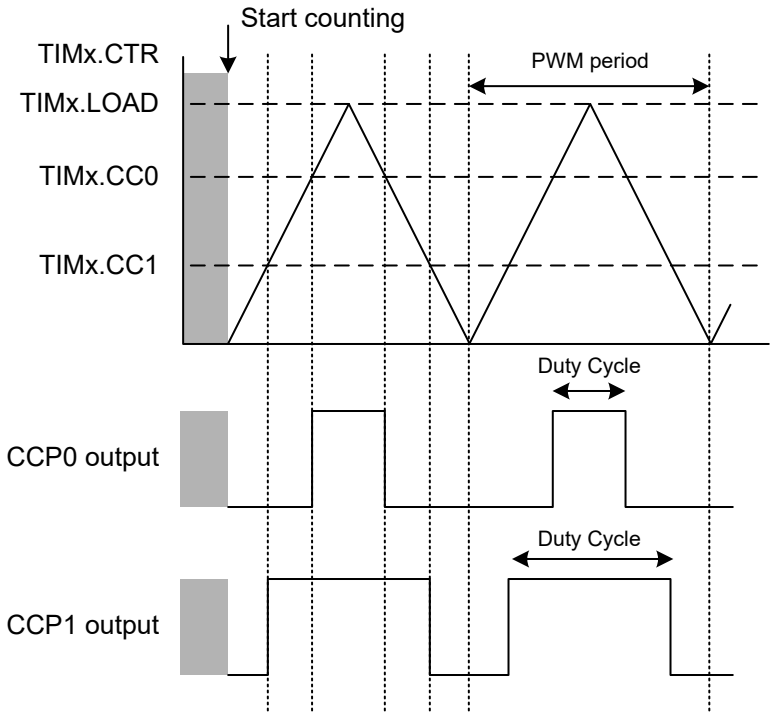



**Figure 35-30. Center-Aligned PWM**

#### *35.2.5.2.3 Asymmetric PWM (TIMA only)*

In TIMA only, asymmetric PWMs can be generated by generating two synchronized center-aligned PWM signals with a controlled phase shift. To generate the asymmetric PWM signals, the phase load feature is used as described in Section 35.2.2.5.

**Asymmetric PWM Configuration**

To generate asymmetric PWMs using compare match events from the counter:

1. Synchronize TIMA0 and TIMA1 using a cross trigger as described in Section 35.2.7.
2. Configure two center-aligned PWMs as described in Section 35.2.5.2.2 using TIMA0 and TIMA1. TIMA0 and TIMA1 should have the same load value (TIMA.LOAD) and compare value (TIMA.CC_xy[0/1]) to generate the same PWM frequency and duty cycle.
3. Add a phase shift value for TIMA0 or TIMA1 by configuring the phase load value TIMA.PL as described in Section 35.2.2.5.
4. Enable the counter by setting TIMA.CTRCTL.EN = 1.

Figure 35-31 shows an example of asymmetric PWM configuration using CCP channel 0 of TIMA0 and TIMA1.



Copyright © 2026 Texas Instruments Incorporated

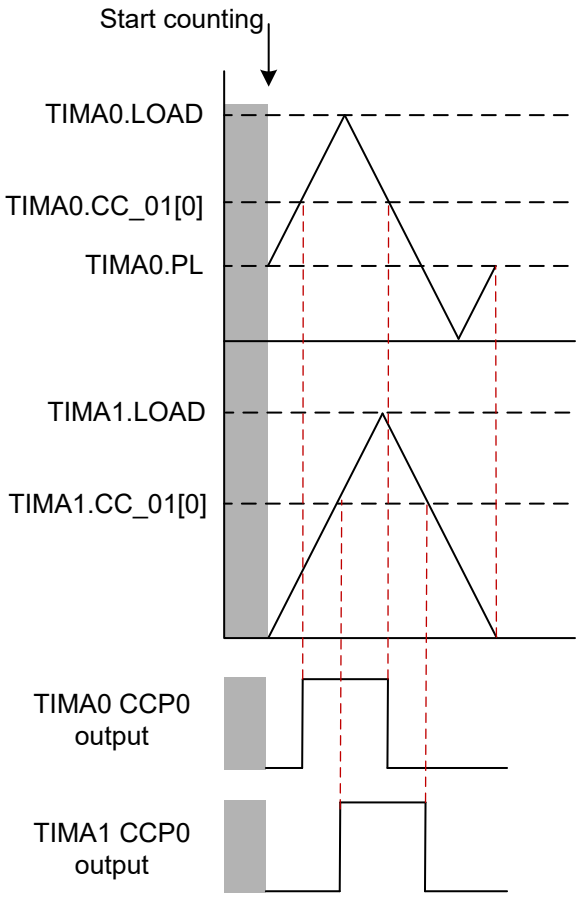



**Figure 35-31. Asymmetric PWM Configuration with Phase Load for CCP channel 0 of TIMA0 and TIMA1**

#### *35.2.5.2.4 Complementary PWM With Deadband Insertion (TIMA only)*

TIMA provides the option of generating complimentary PWM outputs with deadband insertion (nonoverlapping transitions in complimentary PWM signals) from a signal PWM reference signal. Deadband is useful for applications with half-bridge control to avoid shoot-through conditions, such as motor driver or inverter-based applications.

The reference signal is generated on the TIMAx_Cy signal, and the complimentary signal is generated on TIMAx_CyN, where x is the timer instance and y is the CCP output channel. For instances, a reference PWM signal generated on TIMA0 CCP output channel 2 will produce complimentary output signals on TIMA0_C2 and TIMA0_C2N.

The deadband control register (TIMAx.DBCTL) is programmed with the deadband mode and timing information. The deadband mode is selected using the M1_ENABLE bit, and the timing information to control the deadband width in TIMCLK cycles is selected by the RISEDELAY and FALLDELAY bit fields. RISEDELAY and FALLDELAY are a function of the rising or falling edge delay to or from the reference PWM.

See Table 35-18 for the configuration and relationship between deadband mode and deadband width settings.

**Table 35-18. Deadband Modes and Delay Timing Configuration in DBCTL register**

| Deadband Mode | Bitfield | Description | Counting Mode |
|---|---|---|---|
| Mode 0 | M1_ENABLE = 0 | RISEDELAY is applied from **rising edge of reference PWM** to rising edge of TIMAx_Cy signal.<br>FALLDELAY is applied from **falling edge of reference PWM** to rising edge of CxN signal. | Any |



Copyright © 2026 Texas Instruments Incorporated

**Table 35-18. Deadband Modes and Delay Timing Configuration in DBCTL register (continued)**

| Deadband Mode | Bitfield | Description | Counting Mode |
|---|---|---|---|
| Mode 1 | M1_ENABLE = 1 | RISEDELAY is applied from falling edge of TIMAx_CyN signal to **rising edge of reference PWM**. FALLDELAY is applied to **falling edge of reference PWM** to rising edge of TIMAx_CyN signal. | Up/down counting mode only |

### Deadband timing equation and example

The equations for configuring RISEDELAY and FALLDELAY from TIMCLK frequency and deadband timing is shown in Equation 36 and Equation 37.

$$RISEDELAY = f_{TIMCLK} \times t_{dead\_rise} \quad (36)$$

$$FALLDELAY = f_{TIMCLK} \times t_{dead\_fall} \quad (37)$$

For example, if 400ns of deadband is required when using a TIMCLK frequency of 80MHz, and Mode 1 is used with center-aligned PWMs to generate equal deadband every PWM period, then RISEDELAY = FALLDELAY = (80MHz) * (400ns) = 32.

### Complimentary PWM with Deadband Configuration

1. Configure a PWM output for an edge-aligned PWM (Section 35.2.5.2.1) or center-aligned PWM (Section 35.2.5.2.2) for any CCP output channel in TIMA.
2. Configure TIMA.DBCTL with the specified deadband mode (M1_ENABLE) and deadband width RISEDELAY and FALLDELAY, depending on the deadband mode.
3. In TIMx.OCTL_xy[0/1], set CCPO = 0xC to select the signal generator with deadband output.
4. Enable the counter by setting TIMx.CTRCTL.EN = 1.

### Example 1 - Complimentary PWM outputs with deadband using edge-aligned PWM in down-counting mode

For edge-aligned PWM, Mode 0 can only be used for deadband insertion mode. See Figure 35-32 for inserting configurable deadband using down counting mode, TIMA output channel 0, and edge-aligned PWM.

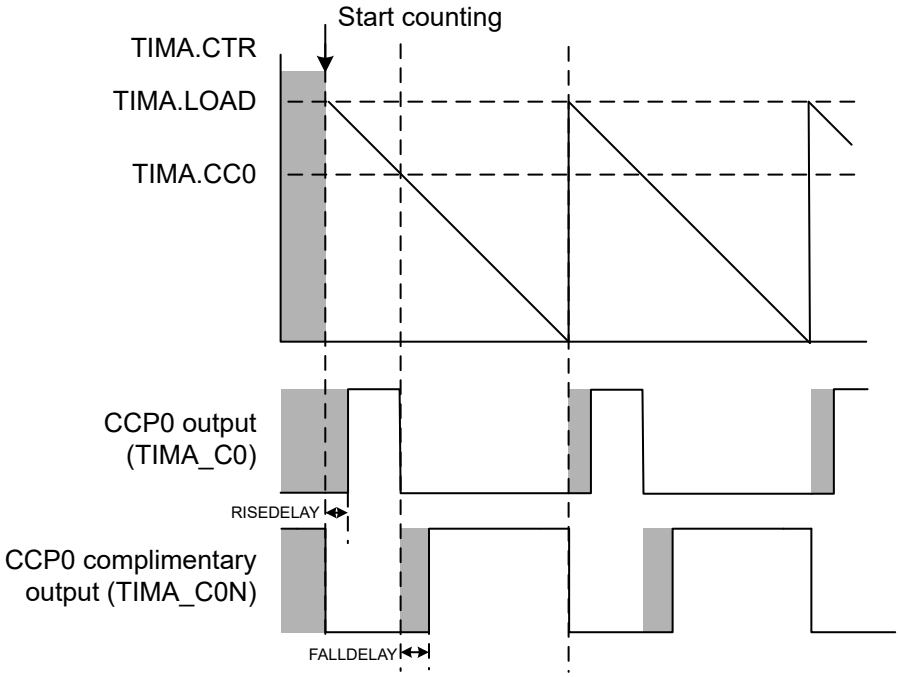



**Figure 35-32. Deadband Insertion (Mode 0) for PWM in Down-Counting Mode**



Copyright © 2026 Texas Instruments Incorporated

**Example 2- Complimentary PWM outputs with deadband using center-aligned PWM**

For center-aligned PWM, Mode 0 or Mode 1 can be used for deadband insertion mode. See Figure 35-33 for inserting configurable deadband using up/down counting mode, TIMA output channel 0, and center-aligned PWMs with deadband.

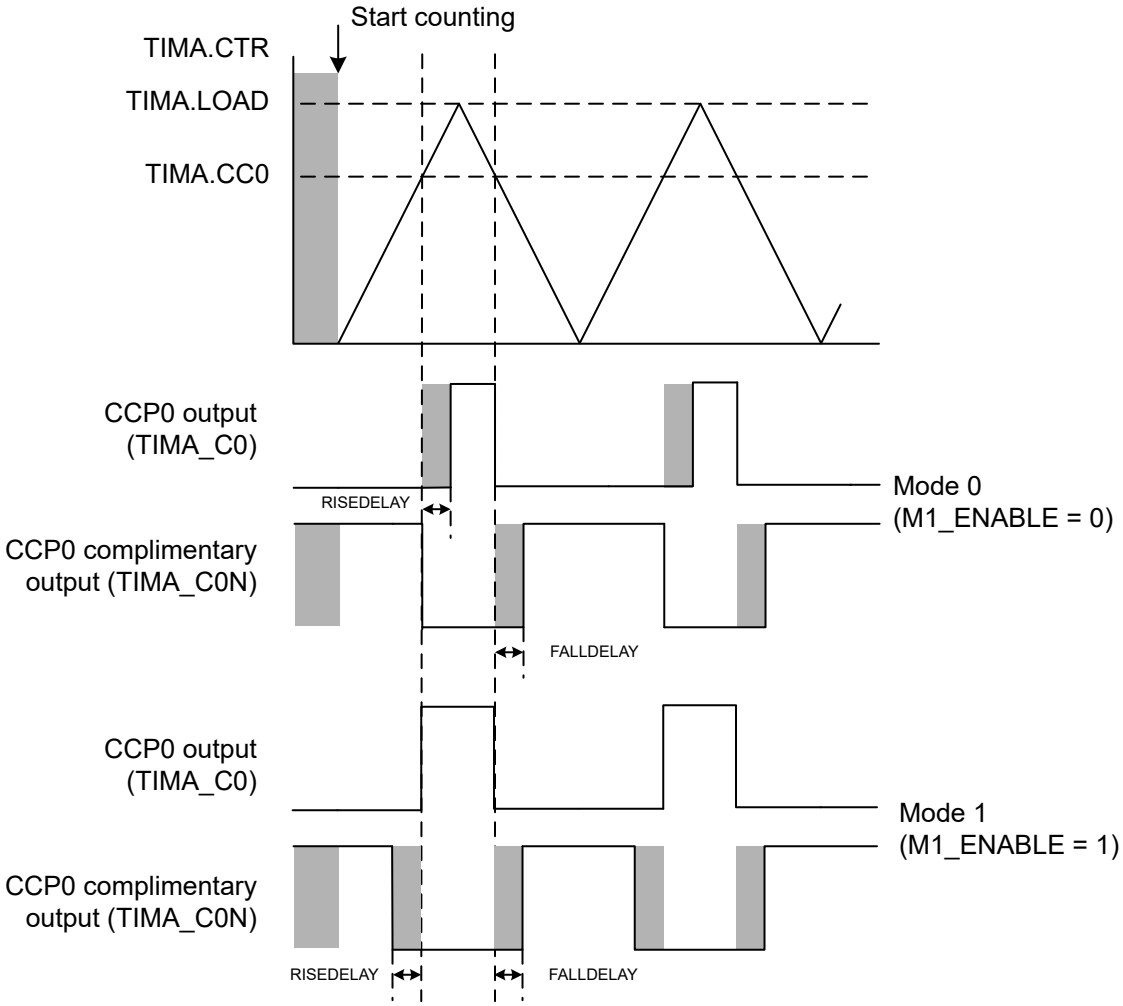



**Figure 35-33. Dead Band Insertion for Center-Aligned PWM (Mode 0 and Mode 1)**

#### 35.2.5.3 Forced Output

Each output channel signal can be forced to a high or low level directly by software, independently of any comparison between the compare register and the counter. A shadow register exists to ensure the forced output action occurs at the end of the timer period.

The output of the CCP channel can be forced to high or low by setting the SWFRCACT bit in the TIMx.CCACT_xy[0/1] register.

In TIMA only, the complimentary output channel can also be forced to high or low by setting the SWFRCACT_CMPL bit in the TIMx.CCACT_xy[0/1] register.

Table 35-19 shows the software force output action configuration options.



Copyright © 2026 Texas Instruments Incorporated

**Table 35-19. Force Output Action Configuration**

| Bit Field | Value | Description/Comment |
|---|---|---|
| SWFRCACT / SWFRCACT_CMPL | 0 | No forced output. Output is directly from the signal generation block. |
| | 1 | Force output high |
| | 2 | Force output low |

### *35.2.6 Fault Handler (TIMA only)*

In TIMA only, there are internal and external fault inputs which can be used to control the generation of PWM signals. The intended use of these inputs is as a mechanism for internal or external circuitry to indicate a fault in the system. This allows the hardware to react quickly to the external fault while optionally signaling an interrupt for software correction and leaving the output signals in a safe state.

It is important to consider the following basic properties of faults in a system, such as:

- Fault input selection (fault signal from external IC, internal signal, etc.)
- How long a fault condition lasts, or the fault condition duration
- How the counter reacts to the entry and exit of a fault condition
- How the output signal reacts to the entry and exit of a fault condition

Fault conditions are synchronously detected using TIMCLK or asynchronously detected. Synchronous faults have a configurable glitch filter and can generate a latched fault event. Asynchronous faults cannot be latched and do not generate a fault event, with a latency of 1-2 TIMCLK cycles to detect the fault and perform a configured action. The CCP output can be configured for either type of fault upon entry and exit conditions.

The fault handler logic diagrams are split into three parts: asynchronous faults, synchronous faults, and fault output generation.

Figure 35-34 shows the asynchronous fault handler logic connections.



Copyright © 2026 Texas Instruments Incorporated

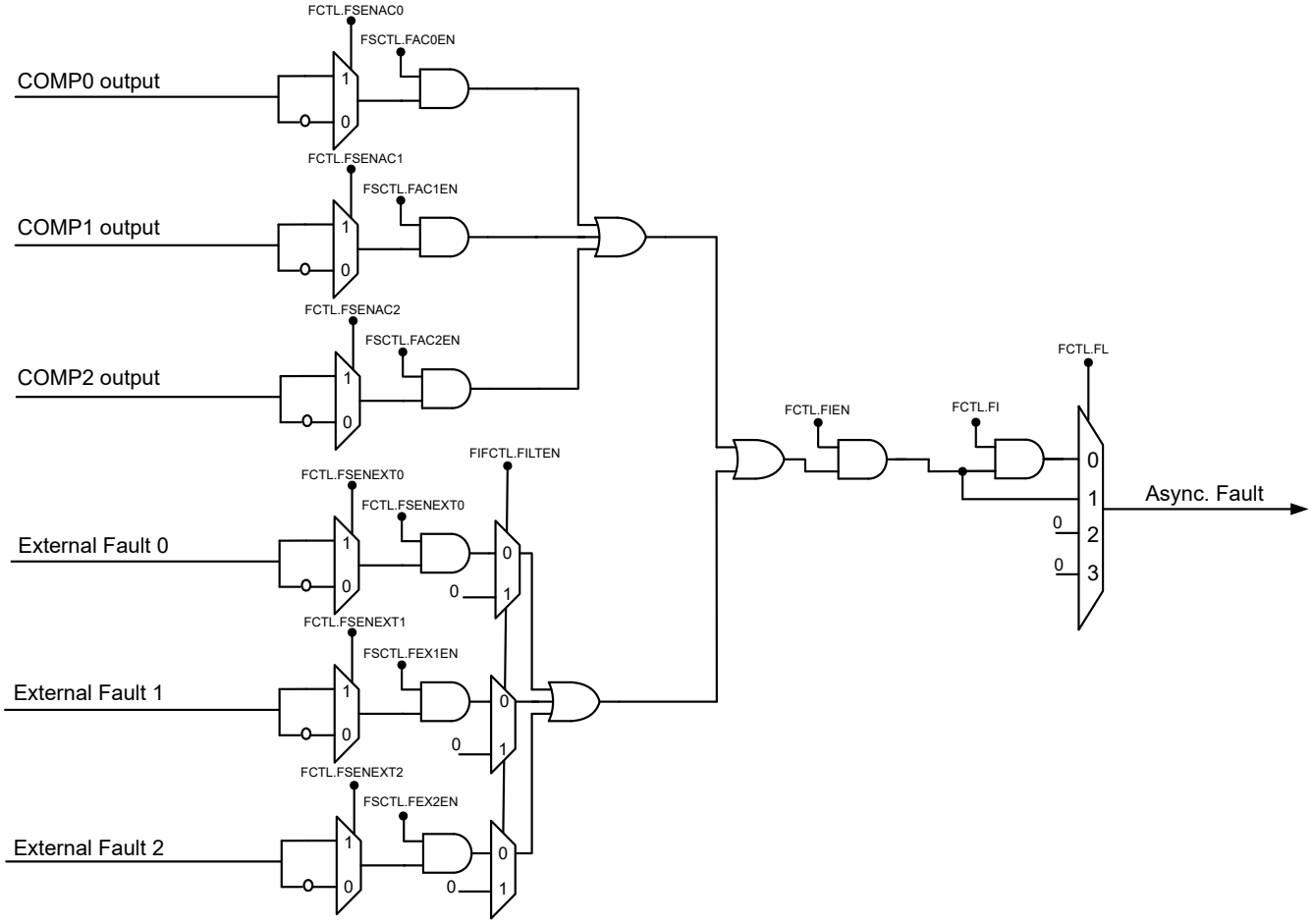



**Figure 35-34. Asynchronous Fault Handler Connections**

Figure 35-35 shows the synchronous fault handler logic connections.



Copyright © 2026 Texas Instruments Incorporated

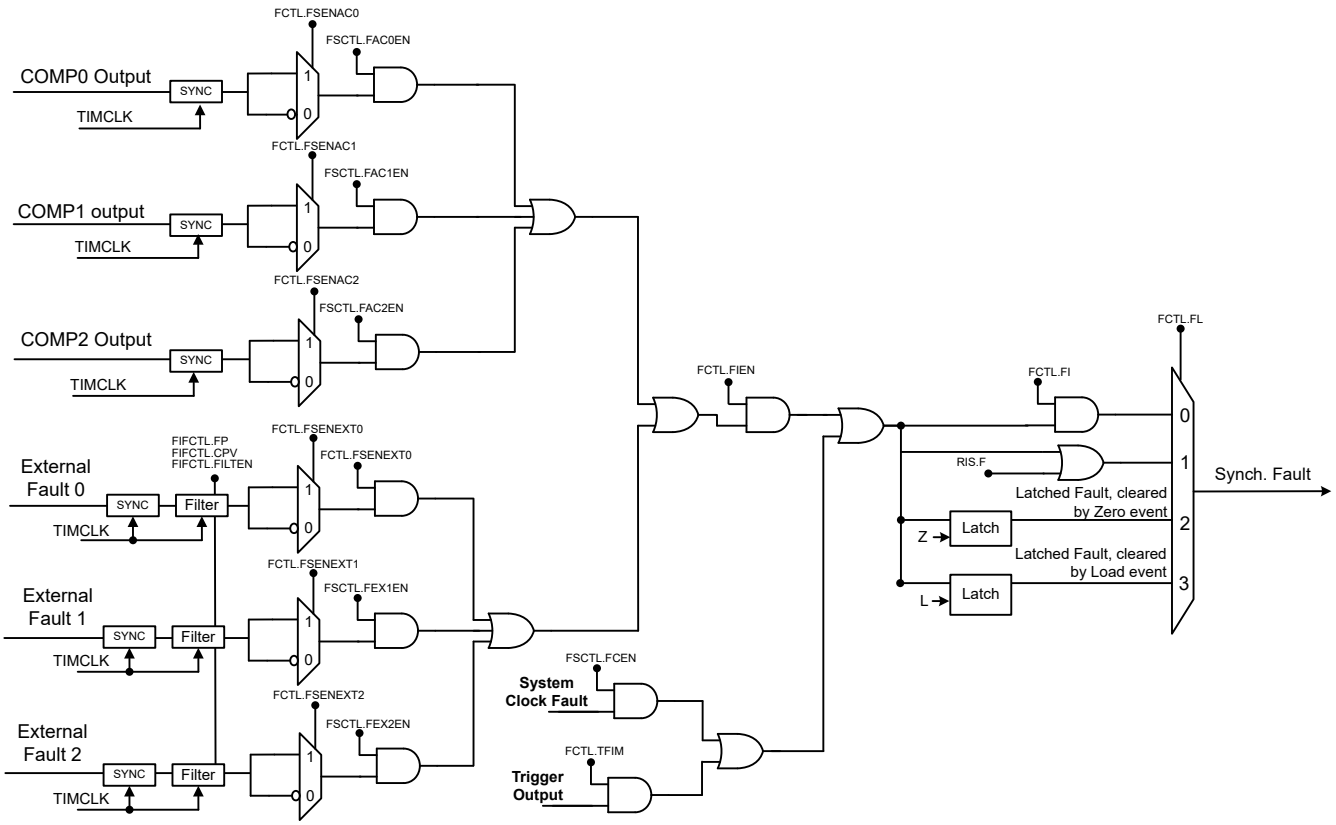



**Figure 35-35. Synchronous Fault Handler Connections**

Figure 35-36 shows the fault output generation logic connections.

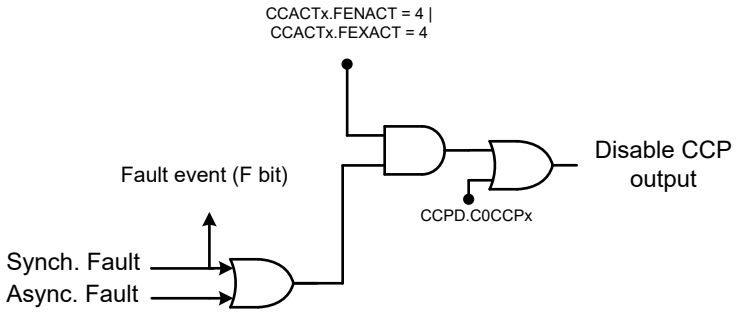



**Figure 35-36. Fault Output Generation Connections**

Key registers for configuring the fault handler are:

- **TIMA.FCTL**: this register controls the fault inputs, fault detection, and error handling behavior.
- **TIMA.FSCTL**: this register controls the fault source selection and enable.
- **TIMA.FIFCTL**: this register controls the input filtering (FILTEN, FP, CPV) for the fault input.
- **TIMA.CCACT_xy[0/1]**: this register controls the actions of the signal generator of the capture-compare portion based on fault events.

#### 35.2.6.1 Fault Input Conditioning

The comparator and external fault pin fault source input passes through a two TIMCLK synchronization stage and the fault input can be filtered through the glitch filter using the fault input filter (TIMA.FIFCTL) register.



Copyright © 2026 Texas Instruments Incorporated

The fault input glitch filter can be enabled by setting the TIMA.FIFCTL.FILTEN bit. The filter period is configured by setting the TIMA.FIFCTL.FP bit.

A consecutive period or majority voting format selected by the TIMA.FIFCTL.CPV bit is used to select the criteria for a CCP input signal.

- **Consecutive period** - The fault input signal must be at the specified level for the defined number of FP timer clocks for the fault input to be processed.
- **Majority voting** - The filter ignores one clock of opposite logic over the filter period. For example, over the number of FP samples of the fault input, up to 1 sample may be of an opposite logic value (glitch) without affecting the output.

The example shown in Figure 35-13 shows the difference between consecutive period and majority voting formats with a digital filter implemented to capture a fault input of 3 TIMCLK periods.

#### 35.2.6.2 Fault Input Sources

The fault control (FCTL) and fault source control (FSCTL) registers are used to select the polarity and enable various fault input sources as shown in Table 35-20.

To enable the final input for fault detection, set TIMA.FTCL.FIEN = 1.

There are four types of fault input sources that are available for synchronous or asynchronous fault detection:

**Comparator (COMP) output**

The comparator output is useful for fault detection when COMPs are used for detecting overcurrent or overvoltage events. To enable the comparator output for fault detection, set the TIMA.FSCTL.FACxEN bit, and configure the polarity to detect the fault using TIMA.FCTL.FSENACx bit (x = 0, 1, or 2 for COMP instance).

**External Fault Pin**

Many IC devices include a fault detection pin (i.e. nFAULT) that an MCU can detect when there is a fault condition in the system. There are 3 fault external signal pins (TIMA_FLTx) connected to every TIMA module, where x = 0, 1, or 2. Each signal pin can be enabled by setting the TIMA.FSCTL.FEXxEN bit, and the polarity of this signal to trigger a fault condition can be configured by using TIMA.FCTL.FSENEXTx bit (where x = 0, 1, or 2 for each TIMA_FLTx pin).

**System Clock Fault**

Any system clock fault can be used to trigger the PWM output(s) to a Hi-Z state. This can be enabled by setting the TIMA.FSCTL.FCEN bit.

**Note**

When a SYSCLK fault occurs, a device reset is generated. Various TIMA fault entry and exit options are invalid while the device is in reset.

**Trigger**

A trigger can be configured to generate a fault condition is detected. This is useful for performing diagnostics or creating fault dependencies from other peripherals in the event fabric. For trigger configuration, please see Section 35.2.7. The fault input mask can be enabled by setting the TIMA.FSCTL.TFIM bit.



Copyright © 2026 Texas Instruments Incorporated

**Table 35-20. Fault Input Sources and Configuration**

| Signal name | Input source | Fault type | Polarity Bit | Enable Bit |
|---|---|---|---|---|
| COMP0_OUT | COMP0 output | Synchronous or Asynchronous | FSENAC0 | FAC0EN |
| COMP1_OUT | COMP1 output | | FSENAC1 | FAC1EN |
| COMP2_OUT | COMP2 output | | FSENAC2 | FAC2EN |
| TIMA_FLT0 | External Fault 0 | | FSENEXT0 | FSENEXT0 |
| TIMA_FLT1 | External Fault 1 | | FSENEXT1 | FSENEXT1 |
| TIMA_FLT2 | External Fault 2 | | FSENEXT2 | FSENEXT2 |
| SYSCLK | System Clock | Synchronous | - | FCEN |
| TRIG | Trigger Output | | - | TFIM |

#### 35.2.6.3 Counter Behavior With Fault Conditions

There are two settings for specifying the counter behavior in fault conditions: TIMA.CTRCTL.FB (during fault behavior) and TIMA.CTRCTL.FRB (fault resume behavior). The counter should continue to be enabled (TIMA.CTRCTL.EN = 1) during the fault handler behavior.

The counter behavior of the fault condition is described in Table 35-21 and Figure 35-37.

**Table 35-21. Counter Behavior in Fault Condition With TIMA.CTRCTL Register**

| Bit Fields | | | | Counter Behavior |
|---|---|---|---|---|
| FB | FRB | CVAE | REPEAT | |
| 0 | X | X | 0 | Ignores fault mode. Counter continues to count during fault and stops when reaches zero. |
| | | | 1/3 | Ignores fault mode. Counter continues to count during fault and repeat. |
| 1 | 0 | X | 0/1/3 | Reacts immediately to fault mode. The counter stops counting immediately and throughout the fault mode. Upon exit of fault mode, the counter continues counting from where it left off. |
| | 1 | 0 | X | Reacts immediately to fault mode. The counter stops counting immediately and throughout the debug mode. Upon exit of fault mode, the counter restarts from LOAD value (restarts a down count). |
| | | 1 | | Reacts immediately to fault mode. The counter stops counting immediately and throughout the debug mode. Upon exit of debug mode, the counter restarts from where it paused at fault entry. |
| | | 2 | | Reacts immediately to fault mode. The counter stops counting immediately and throughout the fault mode. Upon exit of fault mode, it restarts from 0 value (restarts an up/down count). |



Copyright © 2026 Texas Instruments Incorporated

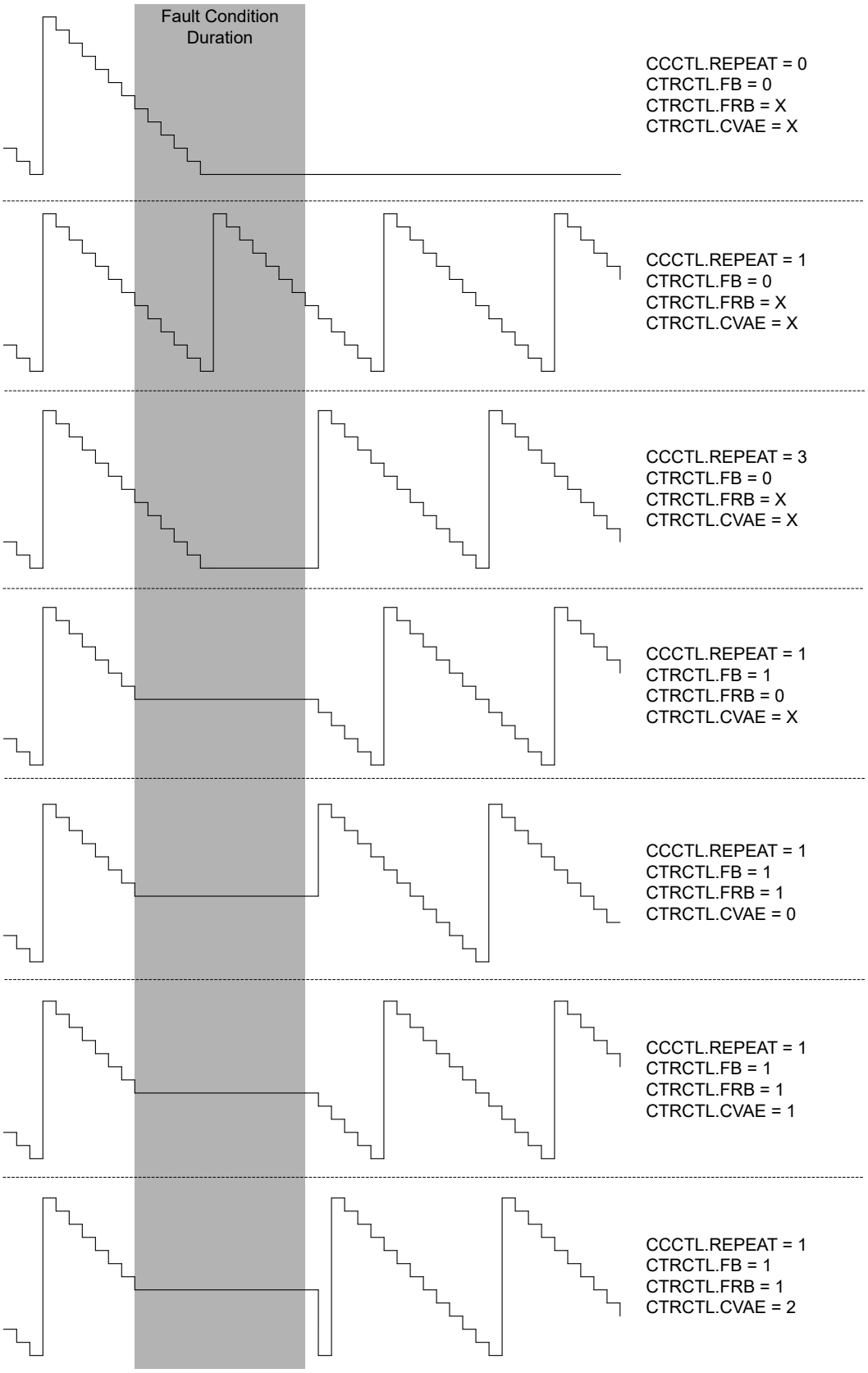



**Figure 35-37. Counter Behavior in Fault Condition with TIMx.CTRCTL Register**



Copyright © 2026 Texas Instruments Incorporated

#### 35.2.6.4 Output Behavior With Fault Conditions

There are two settings for the CCP output channel behavior in a fault condition, TIMA.CCACT_01[0/1].FEXACT (fault exit behavior) and TIMA.CCACT_01[0/1].FENACT (fault entry behavior). The output behavior of fault condition is described in Table 35-22.

**Table 35-22. CCP Output Behavior in Fault Condition With TIMA.CCACT Register**

| Bit Fields | | Output Behavior |
|---|---|---|
| FEXACT | FENACT | |
| 0 | | The CCP output value is unaffected by the fault event |
| 1 | | CCP output value is set high |
| 2 | | CCP output value is set low |
| 3 | | CCP output value toggles |
| 4 | | CCP output is tristated (Hi-Z) |

**Note**

Fault entry and exit behavior is not dependent on the counter being enabled. They will always update the output when in fault mode. Enabling the counter through a software write when in fault mode will generate a load event that will be captured in RIS but the counter will not proceed. Load events will not affect the output when in fault mode. Fault events takes the absolute priority and no events can update the output until you are out of fault mode.

**Note**

If using the TIMA_FALx pin with the CCP capture channel inputs (such as adding deadband to complimentary PWMs) for low latency actions, an external connection should be in hardware between the TIMA_FALx pin and TIMAx_Cy pin (x = timer instance, y = CC channel).

### *35.2.7 Synchronization With Cross Trigger*

When using a main-secondary timer configuration by connecting multiple timers together, the cross-trigger feature can instruct multiple timer modules in the same power domain or across different power domains using the event fabric to start counting simultaneously.

Cross-triggers can be enabled using software, compare events from other timer instances, zero or load events, or generic subscriber events. Some applications may require more than one counter block that can be simultaneously started across the same power domain (such as TIMA0 and TIMA1) or different power domains (such as TIMA0 and TIMG0).

This configuration uses cross triggers from a main timer module as the input trigger condition for the secondary timers. The timer cross trigger is essentially the combined logic of the hardware and software conditions that control the EN bit in the TIMx.CTRCTL register.

The cross triggers that are outputted from the main timer are connected to the external trigger input of other secondary timer modules. As shown in Figure 35-38, TIMGx is the main timer and TIMAx is the secondary timer that will be cross triggered in the configuration example.

**Note**

For power domains and cross trigger selection sources enabled for timer instances, refer to the device-specific data sheet.



Copyright © 2026 Texas Instruments Incorporated

**Note**

The secondary timer must be running at a frequency greater than or equal to the main timer in order to properly receive the cross-trigger. Otherwise, the secondary timer may not observe the cross-trigger pulse, because the cross-trigger would be less than a clock cycle.

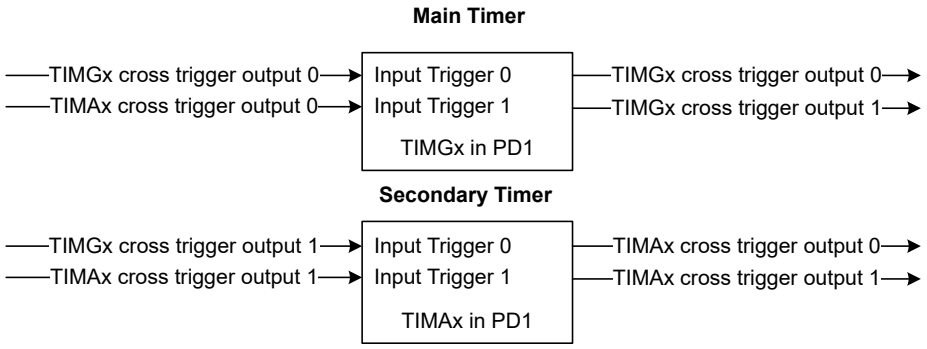



**Figure 35-38. Cross Trigger Connections for Main Timer (TIMGx) and Secondary Timer (TIMAx) in Power Domain 1**

#### 35.2.7.1 Main Timer Cross Trigger Configuration

The following steps are used to configure the main timer cross trigger (which is TIMGx in this example):

1. Configure the main timer (which triggers other secondary timers) for the desired function, such as PWM output generation or using compare mode, to trigger other peripherals. See Section 35.2.5 for how to configure for PWM generation.
2. Enable the cross trigger output function by setting TIMx.CTTRIGCTL.CTEN bit to 1.
3. If the main timer needs to start off the same trigger, required for synchonized timers, **set ETSEL to trigger from the main timer** and set TSEL.TE = 1.
   a. For example, if the main timer is TIMA0 then the ETSEL value needs to equal the TIMA.TRIG0 in order to start the main timer from the cross trigger.
   b. The table of timer cross triggers can be found in the device-specific datasheet under the detailed description section for timers.
4. Choose how to trigger the start of these connected timers, which can be a software trigger or hardware trigger from a subscriber port, zero, load, or compare event.
   a. For a software event trigger, set the TIMx.CTTRIG.TRIG bit.
   b. For a hardware trigger event, select the source for the trigger using TIMx.CTTRIGCTL.EVTCTTRIGSEL and enable the hardware trigger by setting TIMx.CTTRIGCTL.EVTCTEN.

Figure 35-39 shows the connection logic and registers.



Copyright © 2026 Texas Instruments Incorporated

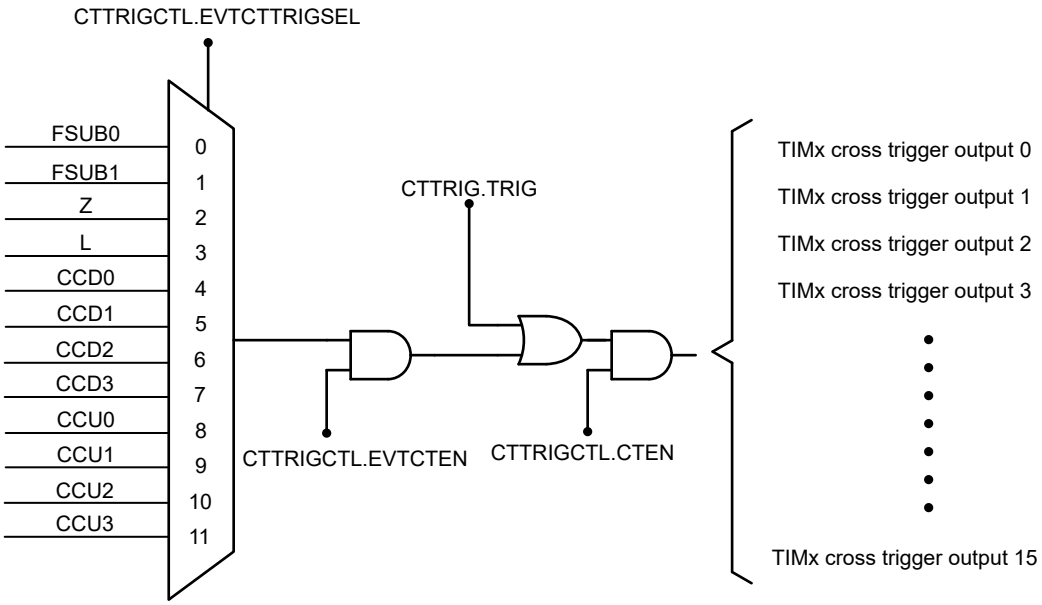



**Figure 35-39. Main Timer Cross Trigger Output Configuration**

#### 35.2.7.2 Secondary Timer Cross Trigger Configuration

The following steps are used to configure the secondary timer cross trigger (which is TIMAx in this example):

1. Configure the secondary timer (triggered by the main timer) for the desired function for this timer, such as PWM output generation or using compare mode, to trigger other peripherals. See Section 35.2.5 for how to configure for PWM generation.
2. Select which input trigger to use according to the device-specific data sheet. Using the example connection in Figure 35-38, TIMAx must be triggered by TIMGx and the cross trigger output 1 of TIMGx is connected to input trigger 0 of TIMAx. Therefore, select input trigger 0 of TIMAx by setting TIMA.TSEL.ETSEL bit to 0.
3. Enable the input trigger function by setting the TIMA.TSEL.TE bit to 1.
4. Set TIMAx.IFCTL_01[0].ISEL = 3 and TIMAx.IFCTL_01[1].ISEL = 3 to select the trigger as the input source.
   a. For a center-aligned PWM, set the TIMA.CCCTL_01[0].ZCOND and TIMA.CCCTL_01[1].ZCOND bits to 1 to use a trigger assertion edge for a zero event.
   b. For an edge-aligned PWM, set the TIMA.CCCTL_01[0].LCOND and TIMA.CCCTL_01[1].LCOND bits to 1 to use the trigger assertion edge for a load event.
5. The TIMx.CTRCTL.EN bit is set as the result of an LCOND or ZCOND condition being met, and the counter value changes to the load value or zero value, respectively.

As the main timer TIMGx must also trigger itself, complete the previous configuration steps for TIMAx to trigger TIMGx itself.

Figure 35-40 shows the logic connection.



Copyright © 2026 Texas Instruments Incorporated

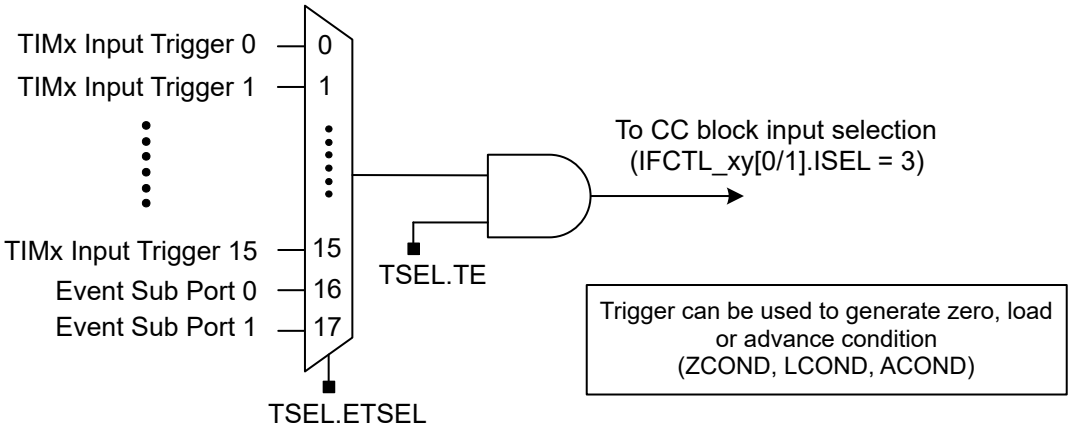



**Figure 35-40. Secondary Timer Cross Trigger Input Configuration**

**Note**

Refer to "TIMx Cross Trigger Map" in the device-specific data sheet for enabled cross trigger mapping using the ETSEL bit. For instance, if the timer instances of a device all use trigger input 0 (TRIG0) to cross trigger other timers, then only TRIG0 can be used to cross trigger other instances.

### *35.2.8 Low Power Operation*

For detailed information on the low power modes in terms of available clock source and behaviors, refer to *Power Domains*.

Timer modules in power domain PD0 can be active and configured to continue counting in all power modes except SHUTDOWN mode. See *Power Domains* for the available clock sources in each low-power mode. The user needs to configure the proper clock to source the timer in low-power mode.

Timer modules in power domain PD1 can only be active in RUN and SLEEP modes. When the system goes to STOP or STANDBY mode, the timer modules will be forced to a disabled state and resume when the systems moves back to RUN or SLEEP modes.

### *35.2.9 Interrupt and Event Support*

TIMx interrupts and events can be configured to any peripheral of the device using the Event Manager. The timer can generate interrupts or events as an **event publisher**, and be triggered from other peripheral events (such as GPIO, comparator, ADC, etc.) as an **event subscriber**. See Section 7.2.5 for guidance on configuring the event registers for CPU interrupts or generic events.

The TIMx module contains three event publishers and two event subscribers.

- One event publisher (CPU_INT) manages TIMx interrupt requests (IRQs) to the CPU subsystem through a static event route.
- The second and third event (GEN_EVENT0 and GEN_EVENT1) are used to setup the generic event publishers and subscribers through Generic route.

TIMx events are summarized in Table 35-23.

**Table 35-23. TIMx Events**

| Event | Type | Source | Destination | Route | Configuration | Functionality |
|---|---|---|---|---|---|---|
| CPU interrupt | Publisher | TIMx | CPU Subsystem | Static route | CPU_INT registers | Fixed interrupt route from TIMx to CPU |
| Generic publisher event | Publisher | TIMx | Other peripherals | Generic route | GEN_EVENT0 and FPUB_0 registers | Configurable interrupt route from TIMx to other peripherals |



Copyright © 2026 Texas Instruments Incorporated

**Table 35-23. TIMx Events (continued)**

| Event | Type | Source | Destination | Route | Configuration | Functionality |
|---|---|---|---|---|---|---|
| Generic publisher event | Publisher | TIMx | Other peripherals | Generic route | GEN_EVENT 1 and FPUB_1 registers | Configurable interrupt route from TIMx to other peripherals |
| Generic subscriber event | Subscriber | Other peripherals | TIMx | Generic route | FSUB_0 | Configurable interrupt route from other peripherals to TIMx |
| Generic subscriber event | Subscriber | Other peripherals | TIMx | Generic route | FSUB_1 | Configurable interrupt route from other peripherals to TIMx |

#### 35.2.9.1 CPU Interrupt Event Publisher (CPU_INT)

The TIMx module provides 18 interrupt sources (depending on the specific TIMx module features) which can be configured to source a CPU interrupt event. The CPU interrupt event configuration is managed with the CPU_INT event management registers. Table 35-24 lists the CPU interrupt events from the TIMx in order of decreasing interrupt priority.

**Table 35-24. TIMx CPU Interrupt Event Conditions (CPU_INT)**

| IIDX STAT | Name | Description | Timer Module |
|---|---|---|---|
| 0x01 | Z | Zero event interrupt. This interrupt is set when there is a zero event. | TIMx |
| 0x02 | L | Load event interrupt. This interrupt is set when there is a load event. | TIMx |
| 0x05 | CCD0 | Capture or compare 0 down event. This interrupt is set when there is a down compare match event at CC0. | TIMx |
| 0x06 | CCD1 | Capture or compare 1 down event. This interrupt is set when there is a down compare match event at CC1. | TIMx |
| 0x07 | CCD2 | Capture or compare 2 down event. This interrupt is set when there is a down compare match event at CC2. This interrupt is only available for TIMA0. | TIMx |
| 0x08 | CCD3 | Capture or compare 3 down event. This interrupt is set when there is a down compare match event at CC3. This interrupt is only available for TIMA0. | TIMx |
| 0x09 | CCU0 | Capture or compare 0 up event. This interrupt is set when there is a up compare match event at CC0. | TIMx |
| 0x0A | CCU1 | Capture or compare 1 up event. This interrupt is set when there is a up compare match event at CC1. | TIMx |
| 0x0B | CCU2 | Capture or compare 2 up event. This interrupt is set when there is a up compare match event at CC2. | TIMx |
| 0x0C | CCU3 | Capture or compare 3 up event. This interrupt is set when there is a up compare match event at CC3. | TIMx |
| 0x0D | CCD4 | Capture or compare 4 down event. This interrupt is set when there is a down compare match event at CC4. This interrupt is only available for TIMA modules. | TIMA |
| 0x0E | CCD5 | Capture or compare 5 down event. This interrupt is set when there is a down compare match event at CC5. This interrupt is only available for TIMA modules. | TIMA |
| 0x0F | CCU4 | Capture or compare 4 up event. This interrupt is set when there is a up compare match event at CC4. This interrupt is only available for TIMA modules. | TIMA |
| 0x10 | CCU5 | Capture or compare 5 up event. This interrupt is set when there is a up compare match event at CC5. This interrupt is only available for TIMA modules. | TIMA |
| 0x19 | F | Fault event interrupt. This interrupt is set when there is a fault condition event. See Section 35.2.6. This interrupt is only available for TIMA modules with fault handler features. | TIMA |
| 0x1A | TOV | Trigger overflow interrupt. This interrupt is set if a trigger event is generated while the associated trigger channel is active. | TIMx |
| 0x1B | REPC | Repeat counter zero interrupt. This bit controls the generation of an interrupt if the repeat counter value transitions from a nonzero value to zero. This interrupt is only available for TIMA modules with a repeat counter feature. | TIMA |
| 0x1C | DC | Direction change interrupt, used in QEI mode. This interrupt is only available for TIMG modules with QEI features. | TIMG |



Copyright © 2026 Texas Instruments Incorporated

**Table 35-24. TIMx CPU Interrupt Event Conditions (CPU_INT) (continued)**

| IIDX STAT | Name | Description | Timer Module |
|---|---|---|---|
| 0X1D | QEIERR | Direction change interrupt, used in QEI mode. This interrupt is only available for TIMG modules with QEI features. | TIMG |

See Section 7.2.5 for guidance on configuring the event registers for CPU interrupts.

#### 35.2.9.2 Generic Event Publisher and Subscriber (GEN_EVENT0 and GEN_EVENT1)

A generic route is a route in which the comparator peripheral publishing the event is configured to use one of several available generic route channels to publish its event to another entity (or entities, in the case of a splitter route), where an entity can be another peripheral, a generic DMA trigger event, or a generic CPU event.

The GEN_EVENT0 and GEN_EVENT1 registers are used to select a peripheral condition (Table 35-25) to use for publishing or subscribing an event. FPUB_0 and FPUB_1 are the publisher port registers and are used to configure which generic route channel to use to broadcast the event. FSUB_0 and FSUB_1 are the subscriber port registers and are used to configure which generic route channel to use to subscribe the event. Other peripherals, the DMA, or the CPU can subscribe to this event by configuring its subscriber port to listen on the same generic route channel which the publishing peripheral is connected to.

For example, through the use of a generic event channel, it is possible to directly start an ADC conversion from a TIMx event by connecting a TIMx FPUB_x and ADC FSUB_0 to the same generic event channel. Refer to Section 7.1.3.3 and Section 7.2.3 for how generic event route works.

**Table 35-25. TIMx Generic Event Conditions (GEN_EVENT0 and GEN_EVENT1)**

| IIDX STAT | Name | Description | Timer Module |
|---|---|---|---|
| 0x01 | Z | Zero event interrupt. This interrupt is set when there is a zero event. | TIMx |
| 0x02 | L | Load event interrupt. This interrupt is set when there is a load event. | TIMx |
| 0x05 | CCD0 | Capture or compare 0 down event. This interrupt is set when there is a down compare match event at CC0. | TIMx |
| 0x06 | CCD1 | Capture or compare 1 down event. This interrupt is set when there is a down compare match event at CC1. | TIMx |
| 0x07 | CCD2 | Capture or compare 2 down event. This interrupt is set when there is a down compare match event at CC2. This interrupt is only available for TIMA0. | TIMx |
| 0x08 | CCD3 | Capture or compare 3 down event. This interrupt is set when there is a down compare match event at CC3. This interrupt is only available for TIMA0. | TIMx |
| 0x09 | CCU0 | Capture or compare 0 up event. This interrupt is set when there is a up compare match event at CC0. | TIMx |
| 0x0A | CCU1 | Capture or compare 1 up event. This interrupt is set when there is a up compare match event at CC1. | TIMx |
| 0x0B | CCU2 | Capture or compare 2 up event. This interrupt is set when there is a up compare match event at CC2. | TIMx |
| 0x0C | CCU3 | Capture or compare 3 up event. This interrupt is set when there is a up compare match event at CC3. | TIMx |
| 0x0D | CCD4 | Capture or compare 4 down event. This interrupt is set when there is a down compare match event at CC4. This interrupt is only available for TIMA modules. | TIMA |
| 0x0E | CCD5 | Capture or compare 5 down event. This interrupt is set when there is a down compare match event at CC5. This interrupt is only available for TIMA modules. | TIMA |
| 0x0F | CCU4 | Capture or compare 4 up event. This interrupt is set when there is a up compare match event at CC4. This interrupt is only available for TIMA modules. | TIMA |
| 0x10 | CCU5 | Capture or compare 5 up event. This interrupt is set when there is a up compare match event at CC5. This interrupt is only available for TIMA modules. | TIMA |
| 0x19 | F | Fault event interrupt. This interrupt is set when there is a fault condition event. See Section 35.2.6. This interrupt is only available for TIMA modules with fault handler features. | TIMA |



Copyright © 2026 Texas Instruments Incorporated

**Table 35-25. TIMx Generic Event Conditions (GEN_EVENT0 and GEN_EVENT1) (continued)**

| IIDX STAT | Name | Description | Timer Module |
|---|---|---|---|
| 0x1A | TOV | Trigger overflow interrupt. This interrupt is set if a trigger event is generated while the associated trigger channel is active. | TIMx |
| 0x1B | REPC | Repeat counter zero interrupt. This bit controls the generation of an interrupt if the repeat counter value transitions from a nonzero value to zero. This interrupt is only available for TIMA modules with a repeat counter feature. | TIMA |
| 0x1C | DC | Direction change interrupt, used in QEI mode. This interrupt is only available for TIMG modules with QEI features. | TIMG |
| 0x1D | QEIERR | Direction change interrupt, used in QEI mode. This interrupt is only available for TIMG modules with QEI features. | TIMG |

See Section 7.2.5 for guidance on configuring the event registers.

#### 35.2.9.3 Generic Subscriber Event Example (COMP to TIMx)

A common use case is to directly trigger a timer action (reset, PWM output, etc.) from the comparator output. The Event Manager can be used to set up the comparator as a generic event publisher (FPUB_1) and the timer as a generic event subscriber (FSUB_0 or FSUB_1) listening for an event from the publisher.

**Comparator configuration (Publisher):**

1. Set COMPIFG bit in GEN_EVENT0 IMASK register to mask the comparator output interrupt. For more info, see the COMP chapter.
2. Configure FPUB_1 register in comparator module to connect to event channel y.
3. Configure and enable the comparator.

**TIMx configuration (Subscriber):**

1. Configure FSUB_0 or FSUB_1 register in the TIMx module to connect to event channel y. Channel y must not be in use by another peripheral.
2. Set TIMx.IFCTL_xy[0/1].ISEL = 5 or 6 to select the CC input source as FSUB0 or FSUB1 for the TIMx module. See Figure 35-12

   and Section 35.2.3.1.1.5.
3. Configure and enable the timer.

**Note**

The comparator output (COMP0:2) has two more options for cross-peripherals:

- Input to the TIMx CC block directly using TIMx.IFCTL_xy[0/1].ISEL = 7h, 8h, or 9h.
  - This is a lower latency path and is useful for applications such as cycle-by-cycle overcurrent limiting. See Figure 35-12.
- Using the timer's cross trigger path to configure the event subscriber port as a trigger source by configuring the TIMx.TSEL.ETSEL = 0x10 for FSUB0 or to 0x11 for FSUB1.
  - Enable the input trigger function by setting the TIMx.TSEL.TE bit to 1.
  - Set TIMx.IFCTL_xy[0/1].ISEL bit to 3 to select the cross trigger as input source for the TIMx module.
  - This is useful for using an event subscriber to trigger multiple timer instances in the same power domain, such as COMP to multiple TIMG instances.

### *35.2.10 Debug Handler (TIMA Only)*

In TIMA only, the counter behavior in CPU halt debug mode and debug resume can also be configured by software using the PDBGCTL and CTRCTL registers.



Copyright © 2026 Texas Instruments Incorporated

TIMA can be configured to stop counting or continue counting when the CPU is halted for debug by the debug subsystem. By default, TIMA stops counting when the CPU is halted for debug and the device is in a debug state. To allow TIMA to continue to free run when the CPU is stopped for debug, set the FREE bit in the PDBGCTL register. See Section 41.2.1.2 for more information.

The CTRCTL register lets the device to resume counting or perform the action specified by the CVAE field following the exit of debug mode using the DRB bit.



Copyright © 2026 Texas Instruments Incorporated

## 35.3 TIMx Registers

Table 35-26 lists the memory-mapped registers for the TIMx registers. All register offset addresses not listed in Table 35-26 should be considered as reserved locations and the register contents should not be modified.

**Table 35-26. TIMX Registers**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 400h | FSUB_0 | Subsciber Port 0 | | Go |
| 404h | FSUB_1 | Subscriber Port 1 | | Go |
| 444h | FPUB_0 | Publisher Port 0 | | Go |
| 448h | FPUB_1 | Publisher Port 1 | | Go |
| 800h | PWREN | Power enable | | Go |
| 804h | RSTCTL | Reset Control | | Go |
| 814h | STAT | Status Register | | Go |
| 1000h | CLKDIV | Clock Divider | | Go |
| 1008h | CLKSEL | Clock Select for Ultra Low Power peripherals | | Go |
| 1018h | PDBGCTL | Peripheral Debug Control | | Go |
| 1020h | IIDX | Interrupt index | CPU_INT | Go |
| 1028h | IMASK | Interrupt mask | CPU_INT | Go |
| 1030h | RIS | Raw interrupt status | CPU_INT | Go |
| 1038h | MIS | Masked interrupt status | CPU_INT | Go |
| 1040h | ISET | Interrupt set | CPU_INT | Go |
| 1048h | ICLR | Interrupt clear | CPU_INT | Go |
| 1050h | IIDX | Interrupt index | GEN_EVENT 0 | Go |
| 1058h | IMASK | Interrupt mask | GEN_EVENT 0 | Go |
| 1060h | RIS | Raw interrupt status | GEN_EVENT 0 | Go |
| 1068h | MIS | Masked interrupt status | GEN_EVENT 0 | Go |
| 1070h | ISET | Interrupt set | GEN_EVENT 0 | Go |
| 1078h | ICLR | Interrupt clear | GEN_EVENT 0 | Go |
| 1080h | IIDX | Interrupt index | GEN_EVENT 1 | Go |
| 1088h | IMASK | Interrupt mask | GEN_EVENT 1 | Go |
| 1090h | RIS | Raw interrupt status | GEN_EVENT 1 | Go |
| 1098h | MIS | Masked interrupt status | GEN_EVENT 1 | Go |
| 10A0h | ISET | Interrupt set | GEN_EVENT 1 | Go |
| 10A8h | ICLR | Interrupt clear | GEN_EVENT 1 | Go |
| 10E0h | EVT_MODE | Event Mode | | Go |
| 10FCh | DESC | Module Description | | Go |
| 1100h | CCPD | CCP Direction | | Go |
| 1104h | ODIS | Output Disable | | Go |



Copyright © 2026 Texas Instruments Incorporated

**Table 35-26. TIMX Registers (continued)**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 1108h | CCLKCTL | Counter Clock Control Register | | Go |
| 110Ch | CPS | Clock Prescale Register | | Go |
| 1110h | CPSV | Clock prescale count status register | | Go |
| 1114h | CTTRIGCTL | Timer Cross Trigger Control Register | | Go |
| 111Ch | CTTRIG | Timer Cross Trigger Register | | Go |
| 1120h | FSCTL | Fault Source Control | | Go |
| 1124h | GCTL | Global control register | | Go |
| 1800h | CTR | Counter Register | | Go |
| 1804h | CTRCTL | Counter Control Register | | Go |
| 1808h | LOAD | Load Register | | Go |
| 1810h + formula | CC_01[y] | Capture or Compare Register 0/1 | | Go |
| 1818h + formula | CC_23[y] | Capture or Compare Register 2/3 | | Go |
| 1820h + formula | CC_45[y] | The CC_45 register are a registers which can be used as compare to the current CTR to create an events CC4U, CC4D, CC5U and CC5D. | | Go |
| 1830h + formula | CCCTL_01[y] | Capture or Compare Control Registers 0/1 | | Go |
| 1838h + formula | CCCTL_23[y] | Capture or Compare Control Registers 2/3 | | Go |
| 1840h + formula | CCCTL_45[y] | Capture or Compare Control Registers 4/5 | | Go |
| 1850h + formula | OCTL_01[y] | CCP Output Control Registers 0/1 | | Go |
| 1858h + formula | OCTL_23[y] | CCP Output Control Registers 2/3 | | Go |
| 1870h + formula | CCACT_01[y] | Capture or Compare Action Registers 0/1 | | Go |
| 1878h + formula | CCACT_23[y] | Capture or Compare Action Registers 2/3 | | Go |
| 1880h + formula | IFCTL_01[y] | Input Filter Control Register 0/1 | | Go |
| 1888h + formula | IFCTL_23[y] | Input Filter Control Register 2/3 | | Go |
| 18A0h | PL | Phase Load Register | | Go |
| 18A4h | DBCTL | Dead Band insertion control register | | Go |
| 18B0h | TSEL | Trigger Select Register | | Go |
| 18B4h | RC | Repeat counter Register | | Go |
| 18B8h | RCLD | Repeat counter load Register | | Go |
| 18BCh | QDIR | QEI Count Direction Register | | Go |
| 18D0h | FCTL | Fault Control Register | | Go |
| 18D4h | FIFCTL | Fault input Filter control register | | Go |

Complex bit access types are encoded to fit into small table cells. Table 35-27 shows the codes that are used for access types in this section.



Copyright © 2026 Texas Instruments Incorporated

**Table 35-27. TIMx Access Type Codes**

| Access Type | Code | Description |
|---|---|---|
| Read Type | | |
| R | R | Read |
| Write Type | | |
| W | W | Write |
| WK | W<br>K | Write<br>Write protected by a key |
| Reset or Default Value | | |
| -*n* | | Value after reset or the default value |
| Register Array Variables | | |
| i,j,k,l,m,n | | When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula. |
| y | | When this variable is used in a register name, an offset, or an address it refers to the value of a register array. |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.1 FSUB_0 (Offset = 400h) [Reset = 00000000h]

FSUB_0 is shown in Figure 35-41 and described in Table 35-28.

Return to the Summary Table.

Subscriber port

**Figure 35-41. FSUB_0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | CHANID | | | |
| R-0h | | | | | | | | | | | | R/W-0h | | | |

**Table 35-28. FSUB_0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3-0 | CHANID | R/W | 0h | 0 = disconnected.<br>1-15 = connected to channelID = CHANID.<br>0h = A value of 0 specifies that the event is not connected<br>Fh = Consult your device datasheet as the actual allowed maximum may be less than 15. |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.2 FSUB_1 (Offset = 404h) [Reset = 00000000h]

FSUB_1 is shown in Figure 35-42 and described in Table 35-29.

Return to the Summary Table.

Subscriber port

**Figure 35-42. FSUB_1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | CHANID | | | |
| R-0h | | | | | | | | | | | | R/W-0h | | | |

**Table 35-29. FSUB_1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3-0 | CHANID | R/W | 0h | 0 = disconnected.<br>1-15 = connected to channelID = CHANID.<br>0h = A value of 0 specifies that the event is not connected<br>Fh = Consult your device datasheet as the actual allowed maximum may be less than 15. |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.3 FPUB_0 (Offset = 444h) [Reset = 00000000h]

FPUB_0 is shown in Figure 35-43 and described in Table 35-30.

Return to the Summary Table.

Publisher port

**Figure 35-43. FPUB_0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | CHANID | | | |
| R-0h | | | | | | | | | | | | R/W-0h | | | |

**Table 35-30. FPUB_0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3-0 | CHANID | R/W | 0h | 0 = disconnected.<br>1-15 = connected to channelID = CHANID.<br>0h = A value of 0 specifies that the event is not connected<br>Fh = Consult your device datasheet as the actual allowed maximum may be less than 15. |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.4 FPUB_1 (Offset = 448h) [Reset = 00000000h]

FPUB_1 is shown in Figure 35-44 and described in Table 35-31.

Return to the Summary Table.

Publisher port

**Figure 35-44. FPUB_1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | CHANID | | | |
| R-0h | | | | | | | | | | | | R/W-0h | | | |

**Table 35-31. FPUB_1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3-0 | CHANID | R/W | 0h | 0 = disconnected.<br>1-15 = connected to channelID = CHANID.<br>0h = A value of 0 specifies that the event is not connected<br>Fh = Consult your device datasheet as the actual allowed maximum may be less than 15. |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.5 PWREN (Offset = 800h) [Reset = 00000000h]

PWREN is shown in Figure 35-45 and described in Table 35-32.

Return to the Summary Table.

Register to control the power state

**Figure 35-45. PWREN**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | ENABLE |
| R-0h | | | | | | | R/WK-0h |

**Table 35-32. PWREN Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | KEY to allow Power State Change<br>26h = KEY to allow write access to this register |
| 23-1 | RESERVED | R | 0h | |
| 0 | ENABLE | R/WK | 0h | Enable the power<br>[EXT_GPRCM.GPRCM.PWREN.KEY] must be set to 26h to write to this bit.<br>0h = Disable Power<br>1h = Enable Power |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.6 RSTCTL (Offset = 804h) [Reset = 00000000h]

RSTCTL is shown in Figure 35-46 and described in Table 35-33.

Return to the Summary Table.

Register to control reset assertion and de-assertion

**Figure 35-46. RSTCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | RESETSTKYCLR | RESETASSERT |
| R-0h | | | | | | WK-0h | WK-0h |

**Table 35-33. RSTCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | Unlock key<br>B1h = KEY to allow write access to this register |
| 23-2 | RESERVED | R | 0h | |
| 1 | RESETSTKYCLR | WK | 0h | Clear the RESETSTKY bit in the STAT register<br>[EXT_GPRCM.GPRCM.RSTCTL.KEY] must be set to B1h to write to this bit.<br>0h = Writing 0 has no effect<br>1h = Clear reset sticky bit |
| 0 | RESETASSERT | WK | 0h | Assert reset to the peripheral<br>[EXT_GPRCM.GPRCM.RSTCTL.KEY] must be set to B1h to write to this bit.<br>0h = Writing 0 has no effect<br>1h = Assert reset |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.7 STAT (Offset = 814h) [Reset = 00000000h]

STAT is shown in Figure 35-47 and described in Table 35-34.

Return to the Summary Table.

peripheral enable and reset status

**Figure 35-47. STAT**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | RESETSTKY |
| R-0h | | | | | | | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |

**Table 35-34. STAT Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-17 | RESERVED | R | 0h | |
| 16 | RESETSTKY | R | 0h | This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register<br>0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register<br>1h = The peripheral was reset since the last bit clear |
| 15-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.8 CLKDIV (Offset = 1000h) [Reset = 00000000h]

CLKDIV is shown in Figure 35-48 and described in Table 35-35.

Return to the Summary Table.

This register is used to specify module-specific divide ratio of the functional clock

**Figure 35-48. CLKDIV**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | | RATIO | | |
| R-0h | | | | | | | | | | | | | R/W-0h | | |

**Table 35-35. CLKDIV Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | RESERVED | R | 0h | |
| 2-0 | RATIO | R/W | 0h | Selects divide ratio of module clock<br>0h = Do not divide clock source<br>1h = Divide clock source by 2<br>2h = Divide clock source by 3<br>3h = Divide clock source by 4<br>4h = Divide clock source by 5<br>5h = Divide clock source by 6<br>6h = Divide clock source by 7<br>7h = Divide clock source by 8 |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.9 CLKSEL (Offset = 1008h) [Reset = 00000000h]

CLKSEL is shown in Figure 35-49 and described in Table 35-36.

Return to the Summary Table.

Clock Source Select Register

**Figure 35-49. CLKSEL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | BUS2XCLK_SEL | BUSCLK_SEL | MFCLK_SEL | LFCLK_SEL | RESERVED |
| R-0h | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h |

**Table 35-36. CLKSEL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-5 | RESERVED | R | 0h | |
| 4 | BUS2XCLK_SEL | R/W | 0h | Selects 2x BUSCLK as clock source if enabled<br>Note: If this bit is set to 1, then LOAD.LD value has to be greater than 1.<br>0h = Does not select this clock as a source<br>1h = Select this clock as a source |
| 3 | BUSCLK_SEL | R/W | 0h | Selects BUSCLK as clock source if enabled<br>0h = Does not select this clock as a source<br>1h = Select this clock as a source |
| 2 | MFCLK_SEL | R/W | 0h | Selects MFCLK as clock source if enabled<br>0h = Does not select this clock as a source<br>1h = Select this clock as a source |
| 1 | LFCLK_SEL | R/W | 0h | Selects LFCLK as clock source if enabled<br>0h = Does not select this clock as a source<br>1h = Select this clock as a source |
| 0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.10 PDBGCTL (Offset = 1018h) [Reset = 00000000h]

PDBGCTL is shown in Figure 35-50 and described in Table 35-37.

Return to the Summary Table.

This register can be used by the software developer to control the behavior of the peripheral relative to the 'Core Halted' input

**Figure 35-50. PDBGCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | SOFT | FREE |
| R-0h | | | | | | R/W-0h | R/W-0h |

**Table 35-37. PDBGCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | |
| 1 | SOFT | R/W | 1h | Soft halt boundary control. This function is only available, if [FREE] is set to 'STOP'<br>0h = The peripheral will halt immediately, even if the resultant state will result in corruption if the system is restarted<br>1h = The peripheral blocks the debug freeze until it has reached a boundary where it can resume without corruption |
| 0 | FREE | R/W | 1h | Free run control<br>0h = The peripheral freezes functionality while the Core Halted input is asserted and resumes when it is deasserted.<br>1h = The peripheral ignores the state of the Core Halted input |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.11 IIDX (Offset = 1020h) [Reset = 00000000h] (CPU_INT)

IIDX is shown in Figure 35-51 and described in Table 35-38.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. Value 0x00 means no event pending. Interrupt 1 is the highest priority, IIDX next highest, 4, 8, … IIDX^31 is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred. On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in [RIS] and [MIS] are cleared as well. After a read from the CPU (not from the debug interface), the register is updated with the next highest priority interrupt, if none are pending, then it should display 0x0.

**Figure 35-51. IIDX**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | STAT | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | |

**Table 35-38. IIDX Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | STAT | R | 0h | Interrupt index status<br>00h = No interrupt pending<br>01h = Interrupt Source: Zero event (Z)<br>02h = nterrupt Source: Load event (L)<br>05h = Interrupt Source: Capture or compare down event (CCD0)<br>06h = Interrupt Source: Capture or compare down event (CCD1)<br>07h = Interrupt Source: Capture or compare down event (CCD2)<br>08h = Interrupt Source: Capture or compare down event (CCD3)<br>09h = Interrupt Source: Capture or compare up event (CCU0)<br>0Ah = Interrupt Source: Capture or compare up event (CCU1)<br>0Bh = Interrupt Source: Capture or compare up event (CCU2)<br>0Ch = Interrupt Source: Capture or compare up event (CCU3)<br>0Dh = Interrupt Source: Compare down event (CCD4)<br>0Eh = Interrupt Source: Compare down event (CCD5)<br>0Fh = Interrupt Source: Compare down event (CCU4)<br>10h = Interrupt Source: Compare down event (CCU5)<br>19h = Interrupt Source: Fault Event generated an interrupt. (F)<br>1Ah = Interrupt Source: Trigger overflow (TOV)<br>1Bh = Interrupt Source: Repeat Counter Zero (REPC)<br>1Ch = Interrupt Source: Direction Change (DC)<br>1Dh = Interrupt Source:QEI Incorrect state transition error (QEIERR) |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.12 IMASK (Offset = 1028h) [Reset = 00000000h] (CPU_INT)

IMASK is shown in Figure 35-52 and described in Table 35-39.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS."

**Figure 35-52. IMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | QEIERR | DC | REPC | TOV | F |
| R-0h | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CCU5 | CCU4 | CCD5 | CCD4 | CCU3 | CCU2 | CCU1 | CCU0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CCD3 | CCD2 | CCD1 | CCD0 | RESERVED | | L | Z |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | | R/W-0h | R/W-0h |

**Table 35-39. IMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | RESERVED | R | 0h | |
| 28 | QEIERR | R/W | 0h | QEIERR Event mask<br>0h = Disable Event<br>1h = Enable Event |
| 27 | DC | R/W | 0h | Direction Change Event mask<br>0h = Disable Event<br>1h = Enable Event |
| 26 | REPC | R/W | 0h | Repeat Counter Zero Event mask<br>0h = Disable Event<br>1h = Enable Event |
| 25 | TOV | R/W | 0h | Trigger Overflow Event mask<br>0h = Disable Event<br>1h = Enable Event |
| 24 | F | R/W | 0h | Fault Event mask<br>0h = Disable Event<br>1h = Enable Event |
| 23-16 | RESERVED | R | 0h | |
| 15 | CCU5 | R/W | 0h | Compare UP event mask CCP5<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 14 | CCU4 | R/W | 0h | Compare UP event mask CCP4<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 13 | CCD5 | R/W | 0h | Compare DN event mask CCP5<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 12 | CCD4 | R/W | 0h | Compare DN event mask CCP4<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |



Copyright © 2026 Texas Instruments Incorporated

**Table 35-39. IMASK Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 11 | CCU3 | R/W | 0h | Capture or Compare UP event mask CCP3<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 10 | CCU2 | R/W | 0h | Capture or Compare UP event mask CCP2<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 9 | CCU1 | R/W | 0h | Capture or Compare UP event mask CCP1<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 8 | CCU0 | R/W | 0h | Capture or Compare UP event mask CCP0<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 7 | CCD3 | R/W | 0h | Capture or Compare DN event mask CCP3<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 6 | CCD2 | R/W | 0h | Capture or Compare DN event mask CCP2<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 5 | CCD1 | R/W | 0h | Capture or Compare DN event mask CCP1<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 4 | CCD0 | R/W | 0h | Capture or Compare DN event mask CCP0<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 3-2 | RESERVED | R | 0h | |
| 1 | L | R/W | 0h | Load Event mask<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 0 | Z | R/W | 0h | Zero Event mask<br>0h = Disable Event<br>1h = Enable Event |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.13 RIS (Offset = 1030h) [Reset = 00000000h] (CPU_INT)

RIS is shown in Figure 35-53 and described in Table 35-40.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Figure 35-53. RIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | QEIERR | DC | REPC | TOV | F |
| R-0h | | | R-0h | R-0h | R-0h | R-0h | R-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CCU5 | CCU4 | CCD5 | CCD4 | CCU3 | CCU2 | CCU1 | CCU0 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CCD3 | CCD2 | CCD1 | CCD0 | RESERVED | | L | Z |
| R-0h | R-0h | R-0h | R-0h | R-0h | | R-0h | R-0h |

**Table 35-40. RIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | RESERVED | R | 0h | |
| 28 | QEIERR | R | 0h | QEIERR, set on an incorrect state transition on the encoder interface.<br>0h = Event Cleared<br>1h = Event Set |
| 27 | DC | R | 0h | Direction Change<br>0h = Event Cleared<br>1h = Event Set |
| 26 | REPC | R | 0h | Repeat Counter Zero<br>0h = Event Cleared<br>1h = Event Set |
| 25 | TOV | R | 0h | Trigger overflow<br>0h = Event Cleared<br>1h = Event Set |
| 24 | F | R | 0h | Fault<br>0h = Event Cleared<br>1h = Event Set |
| 23-16 | RESERVED | R | 0h | |
| 15 | CCU5 | R | 0h | Compare up event generated an interrupt CCP5<br>0h = Event Cleared<br>1h = Event Set |
| 14 | CCU4 | R | 0h | Compare up event generated an interrupt CCU4<br>0h = Event Cleared<br>1h = Event Set |
| 13 | CCD5 | R | 0h | Compare down event generated an interrupt CCD5<br>0h = Event Cleared<br>1h = Event Set |



Copyright © 2026 Texas Instruments Incorporated

**Table 35-40. RIS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 12 | CCD4 | R | 0h | Compare down event generated an interrupt CCD4<br>0h = Event Cleared<br>1h = Event Set |
| 11 | CCU3 | R | 0h | Capture or compare up event generated an interrupt CCP3<br>0h = Event Cleared<br>1h = Event Set |
| 10 | CCU2 | R | 0h | Capture or compare up event generated an interrupt CCP2<br>0h = Event Cleared<br>1h = Event Set |
| 9 | CCU1 | R | 0h | Capture or compare up event generated an interrupt CCP1<br>0h = Event Cleared<br>1h = Event Set |
| 8 | CCU0 | R | 0h | Capture or compare up event generated an interrupt CCP0<br>0h = Event Cleared<br>1h = Event Set |
| 7 | CCD3 | R | 0h | Capture or compare down event generated an interrupt CCP3<br>0h = Event Cleared<br>1h = Event Set |
| 6 | CCD2 | R | 0h | Capture or compare down event generated an interrupt CCP2<br>0h = Event Cleared<br>1h = Event Set |
| 5 | CCD1 | R | 0h | Capture or compare down event generated an interrupt CCP1<br>0h = Event Cleared<br>1h = Event Set |
| 4 | CCD0 | R | 0h | Capture or compare down event generated an interrupt CCP0<br>0h = Event Cleared<br>1h = Event Set |
| 3-2 | RESERVED | R | 0h | |
| 1 | L | R | 0h | Load event generated an interrupt.<br>0h = Event Cleared<br>1h = Event Set |
| 0 | Z | R | 0h | Zero event generated an interrupt.<br>0h = Event Cleared<br>1h = Event Set |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.14 MIS (Offset = 1038h) [Reset = 00000000h] (CPU_INT)

MIS is shown in Figure 35-54 and described in Table 35-41.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 35-54. MIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | QEIERR | DC | REPC | TOV | F |
| R-0h | | | R-0h | R-0h | R-0h | R-0h | R-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CCU5 | CCU4 | CCD5 | CCD4 | CCU3 | CCU2 | CCU1 | CCU0 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CCD3 | CCD2 | CCD1 | CCD0 | RESERVED | | L | Z |
| R-0h | R-0h | R-0h | R-0h | R-0h | | R-0h | R-0h |

**Table 35-41. MIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | RESERVED | R | 0h | |
| 28 | QEIERR | R | 0h | QEIERR<br>0h = Event Cleared<br>1h = Event Set |
| 27 | DC | R | 0h | Direction Change<br>0h = Event Cleared<br>1h = Event Set |
| 26 | REPC | R | 0h | Repeat Counter Zero<br>0h = Event Cleared<br>1h = Event Set |
| 25 | TOV | R | 0h | Trigger overflow<br>0h = Event Cleared<br>1h = Event Set |
| 24 | F | R | 0h | Fault<br>0h = Event Cleared<br>1h = Event Set |
| 23-16 | RESERVED | R | 0h | |
| 15 | CCU5 | R | 0h | Compare up event generated an interrupt CCP5<br>0h = Event Cleared<br>1h = Event Set |
| 14 | CCU4 | R | 0h | Compare up event generated an interrupt CCP4<br>0h = Event Cleared<br>1h = Event Set |
| 13 | CCD5 | R | 0h | Compare down event generated an interrupt CCP5<br>0h = Event Cleared<br>1h = Event Set |
| 12 | CCD4 | R | 0h | Compare down event generated an interrupt CCP4<br>0h = Event Cleared<br>1h = Event Set |



Copyright © 2026 Texas Instruments Incorporated

**Table 35-41. MIS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 11 | CCU3 | R | 0h | Capture or compare up event generated an interrupt CCP3<br>0h = Event Cleared<br>1h = Event Set |
| 10 | CCU2 | R | 0h | Capture or compare up event generated an interrupt CCP2<br>0h = Event Cleared<br>1h = Event Set |
| 9 | CCU1 | R | 0h | Capture or compare up event generated an interrupt CCP1<br>0h = Event Cleared<br>1h = Event Set |
| 8 | CCU0 | R | 0h | Capture or compare up event generated an interrupt CCP0<br>0h = Event Cleared<br>1h = Event Set |
| 7 | CCD3 | R | 0h | Capture or compare down event generated an interrupt CCP3<br>0h = Event Cleared<br>1h = Event Set |
| 6 | CCD2 | R | 0h | Capture or compare down event generated an interrupt CCP2<br>0h = Event Cleared<br>1h = Event Set |
| 5 | CCD1 | R | 0h | Capture or compare down event generated an interrupt CCP1<br>0h = Event Cleared<br>1h = Event Set |
| 4 | CCD0 | R | 0h | Capture or compare down event generated an interrupt CCP0<br>0h = Event Cleared<br>1h = Event Set |
| 3-2 | RESERVED | R | 0h | |
| 1 | L | R | 0h | Load event generated an interrupt.<br>0h = Event Cleared<br>1h = Event Set |
| 0 | Z | R | 0h | Zero event generated an interrupt.<br>0h = Event Cleared<br>1h = Event Set |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.15 ISET (Offset = 1040h) [Reset = 00000000h] (CPU_INT)

ISET is shown in Figure 35-55 and described in Table 35-42.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 35-55. ISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | QEIERR | DC | REPC | TOV | F |
| R-0h | | | W-0h | W-0h | W-0h | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CCU5 | CCU4 | CCD5 | CCD4 | CCU3 | CCU2 | CCU1 | CCU0 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CCD3 | CCD2 | CCD1 | CCD0 | RESERVED | | L | Z |
| W-0h | W-0h | W-0h | W-0h | R-0h | | W-0h | W-0h |

**Table 35-42. ISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | RESERVED | R | 0h | |
| 28 | QEIERR | W | 0h | QEIERR event SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 27 | DC | W | 0h | Direction Change event SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 26 | REPC | W | 0h | Repeat Counter Zero event SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 25 | TOV | W | 0h | Trigger Overflow event SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 24 | F | W | 0h | Fault event SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 23-16 | RESERVED | R | 0h | |
| 15 | CCU5 | W | 0h | Compare up event 5 SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 14 | CCU4 | W | 0h | Compare up event 4 SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 13 | CCD5 | W | 0h | Compare down event 5 SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 12 | CCD4 | W | 0h | Compare down event 4 SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |



Copyright © 2026 Texas Instruments Incorporated

**Table 35-42. ISET Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 11 | CCU3 | W | 0h | Capture or compare up event SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 10 | CCU2 | W | 0h | Capture or compare up event SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 9 | CCU1 | W | 0h | Capture or compare up event SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 8 | CCU0 | W | 0h | Capture or compare up event SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 7 | CCD3 | W | 0h | Capture or compare down event SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 6 | CCD2 | W | 0h | Capture or compare down event SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 5 | CCD1 | W | 0h | Capture or compare down event SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 4 | CCD0 | W | 0h | Capture or compare down event SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 3-2 | RESERVED | R | 0h | |
| 1 | L | W | 0h | Load event SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 0 | Z | W | 0h | Zero event SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.16 ICLR (Offset = 1048h) [Reset = 00000000h] (CPU_INT)

ICLR is shown in Figure 35-56 and described in Table 35-43.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

**Figure 35-56. ICLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | QEIERR | DC | REPC | TOV | F |
| R-0h | | | W-0h | W-0h | W-0h | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CCU5 | CCU4 | CCD5 | CCD4 | CCU3 | CCU2 | CCU1 | CCU0 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CCD3 | CCD2 | CCD1 | CCD0 | RESERVED | | L | Z |
| W-0h | W-0h | W-0h | W-0h | R-0h | | W-0h | W-0h |

**Table 35-43. ICLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | RESERVED | R | 0h | |
| 28 | QEIERR | W | 0h | QEIERR event CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 27 | DC | W | 0h | Direction Change event CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 26 | REPC | W | 0h | Repeat Counter Zero event CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 25 | TOV | W | 0h | Trigger Overflow event CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 24 | F | W | 0h | Fault event CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 23-16 | RESERVED | R | 0h | |
| 15 | CCU5 | W | 0h | Compare up event 5 CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 14 | CCU4 | W | 0h | Compare up event 4 CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 13 | CCD5 | W | 0h | Compare down event 5 CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 12 | CCD4 | W | 0h | Compare down event 4 CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |



Copyright © 2026 Texas Instruments Incorporated

**Table 35-43. ICLR Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 11 | CCU3 | W | 0h | Capture or compare up event CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 10 | CCU2 | W | 0h | Capture or compare up event CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 9 | CCU1 | W | 0h | Capture or compare up event CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 8 | CCU0 | W | 0h | Capture or compare up event CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 7 | CCD3 | W | 0h | Capture or compare down event CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 6 | CCD2 | W | 0h | Capture or compare down event CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 5 | CCD1 | W | 0h | Capture or compare down event CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 4 | CCD0 | W | 0h | Capture or compare down event CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 3-2 | RESERVED | R | 0h | |
| 1 | L | W | 0h | Load event CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 0 | Z | W | 0h | Zero event CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.17 IIDX (Offset = 1050h) [Reset = 00000000h] (GEN_EVENT0)

IIDX is shown in Figure 35-57 and described in Table 35-44.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. Value 0x00 means no event pending. Interrupt 1 is the highest priority, IIDX next highest, 4, 8, … IIDX^31 is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred. On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in [RIS] and [MIS] are cleared as well. After a read from the CPU (not from the debug interface), the register is updated with the next highest priority interrupt, if none are pending, then it should display 0x0.

**Figure 35-57. IIDX**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | STAT | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | |

**Table 35-44. IIDX Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | STAT | R | 0h | Interrupt index status<br>00h = No interrupt pending<br>01h = Interrupt Source: Zero event (Z)<br>02h = nterrupt Source: Load event (L)<br>05h = Interrupt Source: Capture or compare down event (CCD0)<br>06h = Interrupt Source: Capture or compare down event (CCD1)<br>07h = Interrupt Source: Capture or compare down event (CCD2)<br>08h = Interrupt Source: Capture or compare down event (CCD3)<br>09h = Interrupt Source: Capture or compare up event (CCU0)<br>0Ah = Interrupt Source: Capture or compare up event (CCU1)<br>0Bh = Interrupt Source: Capture or compare up event (CCU2)<br>0Ch = Interrupt Source: Capture or compare up event (CCU3)<br>0Dh = Interrupt Source: Compare down event (CCD4)<br>0Eh = Interrupt Source: Compare down event (CCD5)<br>0Fh = Interrupt Source: Compare down event (CCU4)<br>10h = Interrupt Source: Compare down event (CCU5)<br>19h = Interrupt Source: Fault Event generated an interrupt. (F)<br>1Ah = Interrupt Source: Trigger overflow (TOV)<br>1Bh = Interrupt Source: Repeat Counter Zero (REPC)<br>1Ch = Interrupt Source: Direction Change (DC)<br>1Dh = Interrupt Source:QEI Incorrect state transition error (QEIERR) |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.18 IMASK (Offset = 1058h) [Reset = 00000000h] (GEN_EVENT0)

IMASK is shown in Figure 35-58 and described in Table 35-45.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS."

**Figure 35-58. IMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | QEIERR | DC | REPC | TOV | F |
| R-0h | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CCU5 | CCU4 | CCD5 | CCD4 | CCU3 | CCU2 | CCU1 | CCU0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CCD3 | CCD2 | CCD1 | CCD0 | RESERVED | | L | Z |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | | R/W-0h | R/W-0h |

**Table 35-45. IMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | RESERVED | R | 0h | |
| 28 | QEIERR | R/W | 0h | QEIERR Event mask<br>0h = Disable Event<br>1h = Enable Event |
| 27 | DC | R/W | 0h | Direction Change Event mask<br>0h = Disable Event<br>1h = Enable Event |
| 26 | REPC | R/W | 0h | Repeat Counter Zero Event mask<br>0h = Disable Event<br>1h = Enable Event |
| 25 | TOV | R/W | 0h | Trigger Overflow Event mask<br>0h = Disable Event<br>1h = Enable Event |
| 24 | F | R/W | 0h | Fault Event mask<br>0h = Disable Event<br>1h = Enable Event |
| 23-16 | RESERVED | R | 0h | |
| 15 | CCU5 | R/W | 0h | Compare UP event mask CCP5<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 14 | CCU4 | R/W | 0h | Compare UP event mask CCP4<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 13 | CCD5 | R/W | 0h | Compare DN event mask CCP5<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 12 | CCD4 | R/W | 0h | Compare DN event mask CCP4<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |



Copyright © 2026 Texas Instruments Incorporated

**Table 35-45. IMASK Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 11 | CCU3 | R/W | 0h | Capture or Compare UP event mask CCP3<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 10 | CCU2 | R/W | 0h | Capture or Compare UP event mask CCP2<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 9 | CCU1 | R/W | 0h | Capture or Compare UP event mask CCP1<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 8 | CCU0 | R/W | 0h | Capture or Compare UP event mask CCP0<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 7 | CCD3 | R/W | 0h | Capture or Compare DN event mask CCP3<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 6 | CCD2 | R/W | 0h | Capture or Compare DN event mask CCP2<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 5 | CCD1 | R/W | 0h | Capture or Compare DN event mask CCP1<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 4 | CCD0 | R/W | 0h | Capture or Compare DN event mask CCP0<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 3-2 | RESERVED | R | 0h | |
| 1 | L | R/W | 0h | Load Event mask<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 0 | Z | R/W | 0h | Zero Event mask<br>0h = Disable Event<br>1h = Enable Event |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.19 RIS (Offset = 1060h) [Reset = 00000000h] (GEN_EVENT0)

RIS is shown in Figure 35-59 and described in Table 35-46.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Figure 35-59. RIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | QEIERR | DC | REPC | TOV | F |
| R-0h | | | R-0h | R-0h | R-0h | R-0h | R-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CCU5 | CCU4 | CCD5 | CCD4 | CCU3 | CCU2 | CCU1 | CCU0 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CCD3 | CCD2 | CCD1 | CCD0 | RESERVED | | L | Z |
| R-0h | R-0h | R-0h | R-0h | R-0h | | R-0h | R-0h |

**Table 35-46. RIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | RESERVED | R | 0h | |
| 28 | QEIERR | R | 0h | QEIERR, set on an incorrect state transition on the encoder interface.<br>0h = Event Cleared<br>1h = Event Set |
| 27 | DC | R | 0h | Direction Change<br>0h = Event Cleared<br>1h = Event Set |
| 26 | REPC | R | 0h | Repeat Counter Zero<br>0h = Event Cleared<br>1h = Event Set |
| 25 | TOV | R | 0h | Trigger overflow<br>0h = Event Cleared<br>1h = Event Set |
| 24 | F | R | 0h | Fault<br>0h = Event Cleared<br>1h = Event Set |
| 23-16 | RESERVED | R | 0h | |
| 15 | CCU5 | R | 0h | Compare up event generated an interrupt CCP5<br>0h = Event Cleared<br>1h = Event Set |
| 14 | CCU4 | R | 0h | Compare up event generated an interrupt CCU4<br>0h = Event Cleared<br>1h = Event Set |
| 13 | CCD5 | R | 0h | Compare down event generated an interrupt CCD5<br>0h = Event Cleared<br>1h = Event Set |



Copyright © 2026 Texas Instruments Incorporated

**Table 35-46. RIS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 12 | CCD4 | R | 0h | Compare down event generated an interrupt CCD4<br>0h = Event Cleared<br>1h = Event Set |
| 11 | CCU3 | R | 0h | Capture or compare up event generated an interrupt CCP3<br>0h = Event Cleared<br>1h = Event Set |
| 10 | CCU2 | R | 0h | Capture or compare up event generated an interrupt CCP2<br>0h = Event Cleared<br>1h = Event Set |
| 9 | CCU1 | R | 0h | Capture or compare up event generated an interrupt CCP1<br>0h = Event Cleared<br>1h = Event Set |
| 8 | CCU0 | R | 0h | Capture or compare up event generated an interrupt CCP0<br>0h = Event Cleared<br>1h = Event Set |
| 7 | CCD3 | R | 0h | Capture or compare down event generated an interrupt CCP3<br>0h = Event Cleared<br>1h = Event Set |
| 6 | CCD2 | R | 0h | Capture or compare down event generated an interrupt CCP2<br>0h = Event Cleared<br>1h = Event Set |
| 5 | CCD1 | R | 0h | Capture or compare down event generated an interrupt CCP1<br>0h = Event Cleared<br>1h = Event Set |
| 4 | CCD0 | R | 0h | Capture or compare down event generated an interrupt CCP0<br>0h = Event Cleared<br>1h = Event Set |
| 3-2 | RESERVED | R | 0h | |
| 1 | L | R | 0h | Load event generated an interrupt.<br>0h = Event Cleared<br>1h = Event Set |
| 0 | Z | R | 0h | Zero event generated an interrupt.<br>0h = Event Cleared<br>1h = Event Set |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.20 MIS (Offset = 1068h) [Reset = 00000000h] (GEN_EVENT0)

MIS is shown in Figure 35-60 and described in Table 35-47.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 35-60. MIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | QEIERR | DC | REPC | TOV | F |
| R-0h | | | R-0h | R-0h | R-0h | R-0h | R-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CCU5 | CCU4 | CCD5 | CCD4 | CCU3 | CCU2 | CCU1 | CCU0 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CCD3 | CCD2 | CCD1 | CCD0 | RESERVED | | L | Z |
| R-0h | R-0h | R-0h | R-0h | R-0h | | R-0h | R-0h |

**Table 35-47. MIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | RESERVED | R | 0h | |
| 28 | QEIERR | R | 0h | QEIERR<br>0h = Event Cleared<br>1h = Event Set |
| 27 | DC | R | 0h | Direction Change<br>0h = Event Cleared<br>1h = Event Set |
| 26 | REPC | R | 0h | Repeat Counter Zero<br>0h = Event Cleared<br>1h = Event Set |
| 25 | TOV | R | 0h | Trigger overflow<br>0h = Event Cleared<br>1h = Event Set |
| 24 | F | R | 0h | Fault<br>0h = Event Cleared<br>1h = Event Set |
| 23-16 | RESERVED | R | 0h | |
| 15 | CCU5 | R | 0h | Compare up event generated an interrupt CCP5<br>0h = Event Cleared<br>1h = Event Set |
| 14 | CCU4 | R | 0h | Compare up event generated an interrupt CCP4<br>0h = Event Cleared<br>1h = Event Set |
| 13 | CCD5 | R | 0h | Compare down event generated an interrupt CCP5<br>0h = Event Cleared<br>1h = Event Set |
| 12 | CCD4 | R | 0h | Compare down event generated an interrupt CCP4<br>0h = Event Cleared<br>1h = Event Set |



Copyright © 2026 Texas Instruments Incorporated

**Table 35-47. MIS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 11 | CCU3 | R | 0h | Capture or compare up event generated an interrupt CCP3<br>0h = Event Cleared<br>1h = Event Set |
| 10 | CCU2 | R | 0h | Capture or compare up event generated an interrupt CCP2<br>0h = Event Cleared<br>1h = Event Set |
| 9 | CCU1 | R | 0h | Capture or compare up event generated an interrupt CCP1<br>0h = Event Cleared<br>1h = Event Set |
| 8 | CCU0 | R | 0h | Capture or compare up event generated an interrupt CCP0<br>0h = Event Cleared<br>1h = Event Set |
| 7 | CCD3 | R | 0h | Capture or compare down event generated an interrupt CCP3<br>0h = Event Cleared<br>1h = Event Set |
| 6 | CCD2 | R | 0h | Capture or compare down event generated an interrupt CCP2<br>0h = Event Cleared<br>1h = Event Set |
| 5 | CCD1 | R | 0h | Capture or compare down event generated an interrupt CCP1<br>0h = Event Cleared<br>1h = Event Set |
| 4 | CCD0 | R | 0h | Capture or compare down event generated an interrupt CCP0<br>0h = Event Cleared<br>1h = Event Set |
| 3-2 | RESERVED | R | 0h | |
| 1 | L | R | 0h | Load event generated an interrupt.<br>0h = Event Cleared<br>1h = Event Set |
| 0 | Z | R | 0h | Zero event generated an interrupt.<br>0h = Event Cleared<br>1h = Event Set |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.21 ISET (Offset = 1070h) [Reset = 00000000h] (GEN_EVENT0)

ISET is shown in Figure 35-61 and described in Table 35-48.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 35-61. ISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | QEIERR | DC | REPC | TOV | F |
| R-0h | | | W-0h | W-0h | W-0h | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CCU5 | CCU4 | CCD5 | CCD4 | CCU3 | CCU2 | CCU1 | CCU0 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CCD3 | CCD2 | CCD1 | CCD0 | RESERVED | | L | Z |
| W-0h | W-0h | W-0h | W-0h | R-0h | | W-0h | W-0h |

**Table 35-48. ISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | RESERVED | R | 0h | |
| 28 | QEIERR | W | 0h | QEIERR event SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 27 | DC | W | 0h | Direction Change event SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 26 | REPC | W | 0h | Repeat Counter Zero event SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 25 | TOV | W | 0h | Trigger Overflow event SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 24 | F | W | 0h | Fault event SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 23-16 | RESERVED | R | 0h | |
| 15 | CCU5 | W | 0h | Compare up event 5 SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 14 | CCU4 | W | 0h | Compare up event 4 SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 13 | CCD5 | W | 0h | Compare down event 5 SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 12 | CCD4 | W | 0h | Compare down event 4 SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |



Copyright © 2026 Texas Instruments Incorporated

**Table 35-48. ISET Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 11 | CCU3 | W | 0h | Capture or compare up event SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 10 | CCU2 | W | 0h | Capture or compare up event SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 9 | CCU1 | W | 0h | Capture or compare up event SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 8 | CCU0 | W | 0h | Capture or compare up event SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 7 | CCD3 | W | 0h | Capture or compare down event SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 6 | CCD2 | W | 0h | Capture or compare down event SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 5 | CCD1 | W | 0h | Capture or compare down event SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 4 | CCD0 | W | 0h | Capture or compare down event SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 3-2 | RESERVED | R | 0h | |
| 1 | L | W | 0h | Load event SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 0 | Z | W | 0h | Zero event SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.22 ICLR (Offset = 1078h) [Reset = 00000000h] (GEN_EVENT0)

ICLR is shown in Figure 35-62 and described in Table 35-49.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

**Figure 35-62. ICLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | QEIERR | DC | REPC | TOV | F |
| R-0h | | | W-0h | W-0h | W-0h | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CCU5 | CCU4 | CCD5 | CCD4 | CCU3 | CCU2 | CCU1 | CCU0 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CCD3 | CCD2 | CCD1 | CCD0 | RESERVED | | L | Z |
| W-0h | W-0h | W-0h | W-0h | R-0h | | W-0h | W-0h |

**Table 35-49. ICLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | RESERVED | R | 0h | |
| 28 | QEIERR | W | 0h | QEIERR event CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 27 | DC | W | 0h | Direction Change event CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 26 | REPC | W | 0h | Repeat Counter Zero event CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 25 | TOV | W | 0h | Trigger Overflow event CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 24 | F | W | 0h | Fault event CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 23-16 | RESERVED | R | 0h | |
| 15 | CCU5 | W | 0h | Compare up event 5 CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 14 | CCU4 | W | 0h | Compare up event 4 CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 13 | CCD5 | W | 0h | Compare down event 5 CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 12 | CCD4 | W | 0h | Compare down event 4 CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |



Copyright © 2026 Texas Instruments Incorporated

**Table 35-49. ICLR Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 11 | CCU3 | W | 0h | Capture or compare up event CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 10 | CCU2 | W | 0h | Capture or compare up event CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 9 | CCU1 | W | 0h | Capture or compare up event CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 8 | CCU0 | W | 0h | Capture or compare up event CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 7 | CCD3 | W | 0h | Capture or compare down event CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 6 | CCD2 | W | 0h | Capture or compare down event CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 5 | CCD1 | W | 0h | Capture or compare down event CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 4 | CCD0 | W | 0h | Capture or compare down event CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 3-2 | RESERVED | R | 0h | |
| 1 | L | W | 0h | Load event CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 0 | Z | W | 0h | Zero event CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.23 IIDX (Offset = 1080h) [Reset = 00000000h] (GEN_EVENT1)

IIDX is shown in Figure 35-63 and described in Table 35-50.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. Value 0x00 means no event pending. Interrupt 1 is the highest priority, IIDX next highest, 4, 8, … IIDX^31 is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred. On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in [RIS] and [MIS] are cleared as well. After a read from the CPU (not from the debug interface), the register is updated with the next highest priority interrupt, if none are pending, then it should display 0x0.

**Figure 35-63. IIDX**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | STAT | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | |

**Table 35-50. IIDX Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | STAT | R | 0h | Interrupt index status<br>00h = No interrupt pending<br>01h = Interrupt Source: Zero event (Z)<br>02h = nterrupt Source: Load event (L)<br>05h = Interrupt Source: Capture or compare down event (CCD0)<br>06h = Interrupt Source: Capture or compare down event (CCD1)<br>07h = Interrupt Source: Capture or compare down event (CCD2)<br>08h = Interrupt Source: Capture or compare down event (CCD3)<br>09h = Interrupt Source: Capture or compare up event (CCU0)<br>0Ah = Interrupt Source: Capture or compare up event (CCU1)<br>0Bh = Interrupt Source: Capture or compare up event (CCU2)<br>0Ch = Interrupt Source: Capture or compare up event (CCU3)<br>0Dh = Interrupt Source: Compare down event (CCD4)<br>0Eh = Interrupt Source: Compare down event (CCD5)<br>0Fh = Interrupt Source: Compare down event (CCU4)<br>10h = Interrupt Source: Compare down event (CCU5)<br>19h = Interrupt Source: Fault Event generated an interrupt. (F)<br>1Ah = Interrupt Source: Trigger overflow (TOV)<br>1Bh = Interrupt Source: Repeat Counter Zero (REPC)<br>1Ch = Interrupt Source: Direction Change (DC)<br>1Dh = Interrupt Source:QEI Incorrect state transition error (QEIERR) |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.24 IMASK (Offset = 1088h) [Reset = 00000000h] (GEN_EVENT1)

IMASK is shown in Figure 35-64 and described in Table 35-51.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS."

**Figure 35-64. IMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | QEIERR | DC | REPC | TOV | F |
| R-0h | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CCU5 | CCU4 | CCD5 | CCD4 | CCU3 | CCU2 | CCU1 | CCU0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CCD3 | CCD2 | CCD1 | CCD0 | RESERVED | | L | Z |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | | R/W-0h | R/W-0h |

**Table 35-51. IMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | RESERVED | R | 0h | |
| 28 | QEIERR | R/W | 0h | QEIERR Event mask<br>0h = Disable Event<br>1h = Enable Event |
| 27 | DC | R/W | 0h | Direction Change Event mask<br>0h = Disable Event<br>1h = Enable Event |
| 26 | REPC | R/W | 0h | Repeat Counter Zero Event mask<br>0h = Disable Event<br>1h = Enable Event |
| 25 | TOV | R/W | 0h | Trigger Overflow Event mask<br>0h = Disable Event<br>1h = Enable Event |
| 24 | F | R/W | 0h | Fault Event mask<br>0h = Disable Event<br>1h = Enable Event |
| 23-16 | RESERVED | R | 0h | |
| 15 | CCU5 | R/W | 0h | Compare UP event mask CCP5<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 14 | CCU4 | R/W | 0h | Compare UP event mask CCP4<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 13 | CCD5 | R/W | 0h | Compare DN event mask CCP5<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 12 | CCD4 | R/W | 0h | Compare DN event mask CCP4<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |



Copyright © 2026 Texas Instruments Incorporated

**Table 35-51. IMASK Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 11 | CCU3 | R/W | 0h | Capture or Compare UP event mask CCP3<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 10 | CCU2 | R/W | 0h | Capture or Compare UP event mask CCP2<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 9 | CCU1 | R/W | 0h | Capture or Compare UP event mask CCP1<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 8 | CCU0 | R/W | 0h | Capture or Compare UP event mask CCP0<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 7 | CCD3 | R/W | 0h | Capture or Compare DN event mask CCP3<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 6 | CCD2 | R/W | 0h | Capture or Compare DN event mask CCP2<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 5 | CCD1 | R/W | 0h | Capture or Compare DN event mask CCP1<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 4 | CCD0 | R/W | 0h | Capture or Compare DN event mask CCP0<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 3-2 | RESERVED | R | 0h | |
| 1 | L | R/W | 0h | Load Event mask<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 0 | Z | R/W | 0h | Zero Event mask<br>0h = Disable Event<br>1h = Enable Event |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.25 RIS (Offset = 1090h) [Reset = 00000000h] (GEN_EVENT1)

RIS is shown in Figure 35-65 and described in Table 35-52.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Figure 35-65. RIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | QEIERR | DC | REPC | TOV | F |
| R-0h | | | R-0h | R-0h | R-0h | R-0h | R-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CCU5 | CCU4 | CCD5 | CCD4 | CCU3 | CCU2 | CCU1 | CCU0 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CCD3 | CCD2 | CCD1 | CCD0 | RESERVED | | L | Z |
| R-0h | R-0h | R-0h | R-0h | R-0h | | R-0h | R-0h |

**Table 35-52. RIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | RESERVED | R | 0h | |
| 28 | QEIERR | R | 0h | QEIERR, set on an incorrect state transition on the encoder interface.<br>0h = Event Cleared<br>1h = Event Set |
| 27 | DC | R | 0h | Direction Change<br>0h = Event Cleared<br>1h = Event Set |
| 26 | REPC | R | 0h | Repeat Counter Zero<br>0h = Event Cleared<br>1h = Event Set |
| 25 | TOV | R | 0h | Trigger overflow<br>0h = Event Cleared<br>1h = Event Set |
| 24 | F | R | 0h | Fault<br>0h = Event Cleared<br>1h = Event Set |
| 23-16 | RESERVED | R | 0h | |
| 15 | CCU5 | R | 0h | Compare up event generated an interrupt CCP5<br>0h = Event Cleared<br>1h = Event Set |
| 14 | CCU4 | R | 0h | Compare up event generated an interrupt CCU4<br>0h = Event Cleared<br>1h = Event Set |
| 13 | CCD5 | R | 0h | Compare down event generated an interrupt CCD5<br>0h = Event Cleared<br>1h = Event Set |



Copyright © 2026 Texas Instruments Incorporated

**Table 35-52. RIS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 12 | CCD4 | R | 0h | Compare down event generated an interrupt CCD4<br>0h = Event Cleared<br>1h = Event Set |
| 11 | CCU3 | R | 0h | Capture or compare up event generated an interrupt CCP3<br>0h = Event Cleared<br>1h = Event Set |
| 10 | CCU2 | R | 0h | Capture or compare up event generated an interrupt CCP2<br>0h = Event Cleared<br>1h = Event Set |
| 9 | CCU1 | R | 0h | Capture or compare up event generated an interrupt CCP1<br>0h = Event Cleared<br>1h = Event Set |
| 8 | CCU0 | R | 0h | Capture or compare up event generated an interrupt CCP0<br>0h = Event Cleared<br>1h = Event Set |
| 7 | CCD3 | R | 0h | Capture or compare down event generated an interrupt CCP3<br>0h = Event Cleared<br>1h = Event Set |
| 6 | CCD2 | R | 0h | Capture or compare down event generated an interrupt CCP2<br>0h = Event Cleared<br>1h = Event Set |
| 5 | CCD1 | R | 0h | Capture or compare down event generated an interrupt CCP1<br>0h = Event Cleared<br>1h = Event Set |
| 4 | CCD0 | R | 0h | Capture or compare down event generated an interrupt CCP0<br>0h = Event Cleared<br>1h = Event Set |
| 3-2 | RESERVED | R | 0h | |
| 1 | L | R | 0h | Load event generated an interrupt.<br>0h = Event Cleared<br>1h = Event Set |
| 0 | Z | R | 0h | Zero event generated an interrupt.<br>0h = Event Cleared<br>1h = Event Set |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.26 MIS (Offset = 1098h) [Reset = 00000000h] (GEN_EVENT1)

MIS is shown in Figure 35-66 and described in Table 35-53.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 35-66. MIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | QEIERR | DC | REPC | TOV | F |
| R-0h | | | R-0h | R-0h | R-0h | R-0h | R-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CCU5 | CCU4 | CCD5 | CCD4 | CCU3 | CCU2 | CCU1 | CCU0 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CCD3 | CCD2 | CCD1 | CCD0 | RESERVED | | L | Z |
| R-0h | R-0h | R-0h | R-0h | R-0h | | R-0h | R-0h |

**Table 35-53. MIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | RESERVED | R | 0h | |
| 28 | QEIERR | R | 0h | QEIERR<br>0h = Event Cleared<br>1h = Event Set |
| 27 | DC | R | 0h | Direction Change<br>0h = Event Cleared<br>1h = Event Set |
| 26 | REPC | R | 0h | Repeat Counter Zero<br>0h = Event Cleared<br>1h = Event Set |
| 25 | TOV | R | 0h | Trigger overflow<br>0h = Event Cleared<br>1h = Event Set |
| 24 | F | R | 0h | Fault<br>0h = Event Cleared<br>1h = Event Set |
| 23-16 | RESERVED | R | 0h | |
| 15 | CCU5 | R | 0h | Compare up event generated an interrupt CCP5<br>0h = Event Cleared<br>1h = Event Set |
| 14 | CCU4 | R | 0h | Compare up event generated an interrupt CCP4<br>0h = Event Cleared<br>1h = Event Set |
| 13 | CCD5 | R | 0h | Compare down event generated an interrupt CCP5<br>0h = Event Cleared<br>1h = Event Set |
| 12 | CCD4 | R | 0h | Compare down event generated an interrupt CCP4<br>0h = Event Cleared<br>1h = Event Set |



Copyright © 2026 Texas Instruments Incorporated

**Table 35-53. MIS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 11 | CCU3 | R | 0h | Capture or compare up event generated an interrupt CCP3<br>0h = Event Cleared<br>1h = Event Set |
| 10 | CCU2 | R | 0h | Capture or compare up event generated an interrupt CCP2<br>0h = Event Cleared<br>1h = Event Set |
| 9 | CCU1 | R | 0h | Capture or compare up event generated an interrupt CCP1<br>0h = Event Cleared<br>1h = Event Set |
| 8 | CCU0 | R | 0h | Capture or compare up event generated an interrupt CCP0<br>0h = Event Cleared<br>1h = Event Set |
| 7 | CCD3 | R | 0h | Capture or compare down event generated an interrupt CCP3<br>0h = Event Cleared<br>1h = Event Set |
| 6 | CCD2 | R | 0h | Capture or compare down event generated an interrupt CCP2<br>0h = Event Cleared<br>1h = Event Set |
| 5 | CCD1 | R | 0h | Capture or compare down event generated an interrupt CCP1<br>0h = Event Cleared<br>1h = Event Set |
| 4 | CCD0 | R | 0h | Capture or compare down event generated an interrupt CCP0<br>0h = Event Cleared<br>1h = Event Set |
| 3-2 | RESERVED | R | 0h | |
| 1 | L | R | 0h | Load event generated an interrupt.<br>0h = Event Cleared<br>1h = Event Set |
| 0 | Z | R | 0h | Zero event generated an interrupt.<br>0h = Event Cleared<br>1h = Event Set |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.27 ISET (Offset = 10A0h) [Reset = 00000000h] (GEN_EVENT1)

ISET is shown in Figure 35-67 and described in Table 35-54.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 35-67. ISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | QEIERR | DC | REPC | TOV | F |
| R-0h | | | W-0h | W-0h | W-0h | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CCU5 | CCU4 | CCD5 | CCD4 | CCU3 | CCU2 | CCU1 | CCU0 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CCD3 | CCD2 | CCD1 | CCD0 | RESERVED | | L | Z |
| W-0h | W-0h | W-0h | W-0h | R-0h | | W-0h | W-0h |

**Table 35-54. ISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | RESERVED | R | 0h | |
| 28 | QEIERR | W | 0h | QEIERR event SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 27 | DC | W | 0h | Direction Change event SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 26 | REPC | W | 0h | Repeat Counter Zero event SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 25 | TOV | W | 0h | Trigger Overflow event SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 24 | F | W | 0h | Fault event SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 23-16 | RESERVED | R | 0h | |
| 15 | CCU5 | W | 0h | Compare up event 5 SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 14 | CCU4 | W | 0h | Compare up event 4 SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 13 | CCD5 | W | 0h | Compare down event 5 SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 12 | CCD4 | W | 0h | Compare down event 4 SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |



Copyright © 2026 Texas Instruments Incorporated

**Table 35-54. ISET Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 11 | CCU3 | W | 0h | Capture or compare up event SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 10 | CCU2 | W | 0h | Capture or compare up event SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 9 | CCU1 | W | 0h | Capture or compare up event SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 8 | CCU0 | W | 0h | Capture or compare up event SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 7 | CCD3 | W | 0h | Capture or compare down event SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 6 | CCD2 | W | 0h | Capture or compare down event SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 5 | CCD1 | W | 0h | Capture or compare down event SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 4 | CCD0 | W | 0h | Capture or compare down event SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 3-2 | RESERVED | R | 0h | |
| 1 | L | W | 0h | Load event SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |
| 0 | Z | W | 0h | Zero event SET<br>0h = Writing 0 has no effect.<br>1h = Event Set |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.28 ICLR (Offset = 10A8h) [Reset = 00000000h] (GEN_EVENT1)

ICLR is shown in Figure 35-68 and described in Table 35-55.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

**Figure 35-68. ICLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | QEIERR | DC | REPC | TOV | F |
| R-0h | | | W-0h | W-0h | W-0h | W-0h | W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CCU5 | CCU4 | CCD5 | CCD4 | CCU3 | CCU2 | CCU1 | CCU0 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CCD3 | CCD2 | CCD1 | CCD0 | RESERVED | | L | Z |
| W-0h | W-0h | W-0h | W-0h | R-0h | | W-0h | W-0h |

**Table 35-55. ICLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | RESERVED | R | 0h | |
| 28 | QEIERR | W | 0h | QEIERR event CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 27 | DC | W | 0h | Direction Change event CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 26 | REPC | W | 0h | Repeat Counter Zero event CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 25 | TOV | W | 0h | Trigger Overflow event CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 24 | F | W | 0h | Fault event CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 23-16 | RESERVED | R | 0h | |
| 15 | CCU5 | W | 0h | Compare up event 5 CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 14 | CCU4 | W | 0h | Compare up event 4 CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 13 | CCD5 | W | 0h | Compare down event 5 CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 12 | CCD4 | W | 0h | Compare down event 4 CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |



Copyright © 2026 Texas Instruments Incorporated

**Table 35-55. ICLR Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 11 | CCU3 | W | 0h | Capture or compare up event CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 10 | CCU2 | W | 0h | Capture or compare up event CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 9 | CCU1 | W | 0h | Capture or compare up event CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 8 | CCU0 | W | 0h | Capture or compare up event CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 7 | CCD3 | W | 0h | Capture or compare down event CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 6 | CCD2 | W | 0h | Capture or compare down event CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 5 | CCD1 | W | 0h | Capture or compare down event CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 4 | CCD0 | W | 0h | Capture or compare down event CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 3-2 | RESERVED | R | 0h | |
| 1 | L | W | 0h | Load event CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |
| 0 | Z | W | 0h | Zero event CLEAR<br>0h = Writing 0 has no effect.<br>1h = Event Clear |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.29 EVT_MODE (Offset = 10E0h) [Reset = 00000029h]

EVT_MODE is shown in Figure 35-69 and described in Table 35-56.

Return to the Summary Table.

Event mode register. It is used to select whether each line is disabled, in software mode (software clears the RIS) or in hardware mode (hardware clears the RIS)

**Figure 35-69. EVT_MODE**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | EVT2_CFG | | EVT1_CFG | | EVT0_CFG | |
| R-0h | | R-2h | | R-2h | | R-1h | |

**Table 35-56. EVT_MODE Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | RESERVED | R | 0h | |
| 5-4 | EVT2_CFG | R | 2h | Event line mode select for event corresponding to GEN_EVENT1<br>0h = The interrupt or event line is disabled.<br>1h = The interrupt or event line is in software mode. Software must clear the RIS.<br>2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag. |
| 3-2 | EVT1_CFG | R | 2h | Event line mode select for event corresponding to GEN_EVENT0<br>0h = The interrupt or event line is disabled.<br>1h = The interrupt or event line is in software mode. Software must clear the RIS.<br>2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag. |
| 1-0 | EVT0_CFG | R | 1h | Event line mode select for event corresponding to CPU_INT<br>0h = The interrupt or event line is disabled.<br>1h = The interrupt or event line is in software mode. Software must clear the RIS.<br>2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag. |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.30 DESC (Offset = 10FCh) [Reset = 00000000h]

DESC is shown in Figure 35-70 and described in Table 35-57.

Return to the Summary Table.

This register identifies the peripheral and its exact version.

**Figure 35-70. DESC**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MODULEID | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| FEATUREVER | | | | INSTNUM | | | | MAJREV | | | | MINREV | | | |
| R-0h | | | | R-0h | | | | R-0h | | | | R-0h | | | |

**Table 35-57. DESC Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | MODULEID | R | 1111h | Module identification contains a unique peripheral identification number. The assignments are maintained in a central database for all of the platform modules to ensure uniqueness.<br>0h = Smallest value<br>FFFFh = Highest possible value |
| 15-12 | FEATUREVER | R | 0h | Feature Set for the module *instance*<br>0h = Smallest value<br>Fh = Highest possible value |
| 11-8 | INSTNUM | R | 0h | Instance Number within the device. This will be a parameter to the RTL for modules that can have multiple instances<br>0h = Smallest value<br>Fh = Highest possible value |
| 7-4 | MAJREV | R | 0h | Major rev of the IP<br>0h = Smallest value<br>Fh = Highest possible value |
| 3-0 | MINREV | R | 0h | Minor rev of the IP<br>0h = Smallest value<br>Fh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.31 CCPD (Offset = 1100h) [Reset = 00000000h]

CCPD is shown in Figure 35-71 and described in Table 35-58.

Return to the Summary Table.

CCP Direction. Controls whether CCP is used as an input or an output.

**Figure 35-71. CCPD**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | C0CCP3 | C0CCP2 | C0CCP1 | C0CCP0 |
| R-0h | | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

**Table 35-58. CCPD Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | C0CCP3 | R/W | 0h | CCP3 direction<br>0h = Input<br>1h = Output |
| 2 | C0CCP2 | R/W | 0h | CCP2 direction<br>0h = input<br>1h = Output |
| 1 | C0CCP1 | R/W | 0h | CCP1 direction<br>0h = Input<br>1h = Output |
| 0 | C0CCP0 | R/W | 0h | CCP0 direction<br>0h = Input<br>1h = Output |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.32 ODIS (Offset = 1104h) [Reset = 00000000h]

ODIS is shown in Figure 35-72 and described in Table 35-59.

Return to the Summary Table.

The ODIS register output is inverted and then ANDed with the output signal selected by the OCTL register CCPO field (before conditional inversion) to allow software the ability to hold the CCP output low during configuration or shutdown.

**Figure 35-72. ODIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | C0CCP3 | C0CCP2 | C0CCP1 | C0CCP0 |
| R-0h | | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

**Table 35-59. ODIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | C0CCP3 | R/W | 0h | Counter CCP3 Disable Mask<br>Defines whether CCP3 of Counter n is forced low or not<br>0h = Output function as selected by the OCTL register CCPO field are provided to occpout[2].<br>1h = CCP output occpout[3] is forced low. |
| 2 | C0CCP2 | R/W | 0h | Counter CCP2 Disable Mask<br>Defines whether CCP2 of Counter n is forced low or not<br>0h = Output function as selected by the OCTL register CCPO field are provided to output inversion block.<br>1h = CCP output occpout[2] is forced low. |
| 1 | C0CCP1 | R/W | 0h | Counter CCP1 Disable Mask<br>Defines whether CCP0 of Counter n is forced low or not<br>0h = Output function as selected by the OCTL register CCPO field are provided to output inversion block.<br>1h = CCP output occpout[1] is forced low. |
| 0 | C0CCP0 | R/W | 0h | Counter CCP0 Disable Mask<br>Defines whether CCP0 of Counter n is forced low or not<br>0h = Output function as selected by the OCTL register CCPO field are provided to output inversion block.<br>1h = CCP output occpout[0] is forced low. |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.33 CCLKCTL (Offset = 1108h) [Reset = 00000000h]

CCLKCTL is shown in Figure 35-73 and described in Table 35-60.

Return to the Summary Table.

The CCLKCTL register provides a SW mechanism for gating the TIMER clock if the module is expected not to be used but the power domain is alive. This effectively puts the IP in an IDLE state

**Figure 35-73. CCLKCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | CLKEN |
| R-0h | | | | | | | R/W-0h |

**Table 35-60. CCLKCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | CLKEN | R/W | 0h | Clock Enable<br>Disables the clock gating to the module. SW has to explicitly program the value<br>to 0 to gate the clock.<br>0h = Clock is disabled.<br>1h = Clock is enabled |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.34 CPS (Offset = 110Ch) [Reset = 00000000h]

CPS is shown in Figure 35-74 and described in Table 35-61.

Return to the Summary Table.

The CPS register provides the value for the clock pre-scaler.

**Figure 35-74. CPS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | PCNT | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R/W-0h | | | | | | | |

**Table 35-61. CPS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | PCNT | R/W | 0h | Pre-Scale Count<br>This field specifies the pre-scale count value. The selected TIMCLK source is divided by a value of (PCNT+1).<br>A PCNT value of 0 divides TIMCLK by 1, effectively bypassing the divider.<br>A PCNT value of greater than 0 divides the TIMCLK source generating a slower clock<br>0h = Minimum value<br>FFh = Maximum Value |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.35 CPSV (Offset = 1110h) [Reset = 00000000h]

CPSV is shown in Figure 35-75 and described in Table 35-62.

Return to the Summary Table.

The CPSV register provides the ability to read the current clock prescale count value.

**Figure 35-75. CPSV**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | CPSVAL | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | |

**Table 35-62. CPSV Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | CPSVAL | R | 0h | Current Prescale Count Value<br>0h = Minimum value<br>FFh = Maximum Value |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.36 CTTRIGCTL (Offset = 1114h) [Reset = 00000000h]

CTTRIGCTL is shown in Figure 35-76 and described in Table 35-63.

Return to the Summary Table.

Cross Timer Trigger Control Register
This register is used to control the cross trigger connections for enables and faults of different timer instances in the same power domain. Please refer to sections Timer Module Cross Trigger (In/Out) and Fault Cross Triggering for details.

**Figure 35-76. CTTRIGCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | EVTCTTRIGSEL | | | |
| R-0h | | | | R/W-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | EVTCTEN | CTEN |
| R-0h | | | | | | R/W-0h | R/W-0h |

**Table 35-63. CTTRIGCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | RESERVED | R | 0h | |
| 19-16 | EVTCTTRIGSEL | R/W | 0h | Used to Select the subscriber port that should be used for input cross trigger.<br>0h = Use FSUB0 as cross trigger source.<br>1h = Use FSUB1 as cross trigger source.<br>2h = Use Zero event as cross trigger source.<br>3h = Use Load event as cross trigger source.<br>4h = Use CCD0 event as cross trigger source.<br>5h = Use CCD1 event as cross trigger source.<br>6h = Use CCD2 event as cross trigger source.<br>7h = Use CCD3 event as cross trigger source.<br>8h = Use CCU0 event as cross trigger source.<br>9h = Use CCU1 event as cross trigger source.<br>Ah = Use CCU2 event as cross trigger source.<br>Bh = Use CCU3 event as cross trigger source. |
| 15-2 | RESERVED | R | 0h | |
| 1 | EVTCTEN | R/W | 0h | Enable the Input Trigger Conditions to the Timer module as a condition for Cross Triggers.<br>0h = Cross trigger generation disabled.<br>1h = Cross trigger generation enabled |
| 0 | CTEN | R/W | 0h | Timer Cross trigger enable. This field is used to enable whether the SW or HW logic can generate a timer cross trigger event in the system.<br>These cross triggers are connected to the respective timer trigger in of the other timer IPs in the SOC power domain.<br>The timer cross trigger is essentially the combined logic of the HW and SW conditions controlling EN bit in the CTRCTL register.<br>0h = Cross trigger generation disabled.<br>1h = Cross trigger generation enabled |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.37 CTTRIG (Offset = 111Ch) [Reset = 00000000h]

CTTRIG is shown in Figure 35-77 and described in Table 35-64.

Return to the Summary Table.

Cross Timer Trigger Register
This register is used to trigger the timer instances connected and enabled using CTTRIGCTL and CTTRIGMSK registers.

**Figure 35-77. CTTRIG**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | TRIG |
| R-0h | | | | | | | W-0h |

**Table 35-64. CTTRIG Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | TRIG | W | 0h | Generate Cross Trigger<br>This bit when programmed will generate a synchronized trigger condition all the cross trigger enabled Timer instances including current timer instance.<br>0h = Cross trigger generation disabled<br>1h = Generate Cross trigger pulse |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.38 FSCTL (Offset = 1120h) [Reset = 00000000h]

FSCTL is shown in Figure 35-78 and described in Table 35-65.

Return to the Summary Table.

The FSCTL register controls the fault source selection and enable. There are 5 input fault sources either through synchronous path processing or asynchronous path.

**Figure 35-78. FSCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | FEX2EN | FEX1EN | FEX0EN | FAC2EN | FAC1EN | FAC0EN | FCEN |
| R-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

**Table 35-65. FSCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-7 | RESERVED | R | 0h | |
| 6 | FEX2EN | R/W | 0h | This field controls whether the fault is caused by external fault pin 2.<br>0h = Disable<br>1h = Enable |
| 5 | FEX1EN | R/W | 0h | This field controls whether the fault is caused by external fault pin 1.<br>0h = Disable<br>1h = Enable |
| 4 | FEX0EN | R/W | 0h | This field controls whether the fault is caused by external fault pin 0.<br>0h = Disable<br>1h = Enable |
| 3 | FAC2EN | R/W | 0h | This field controls whether the fault is caused by COMP2 output.<br>0h = Disable<br>1h = Enable |
| 2 | FAC1EN | R/W | 0h | This field controls whether the fault is caused by COMP1 output.<br>0h = Disable<br>1h = Enable |
| 1 | FAC0EN | R/W | 0h | This field controls whether the fault signal is caused by COMP0 output.<br>0h = Disable<br>1h = Enable |
| 0 | FCEN | R/W | 0h | This field controls whether the fault is caused by the system clock fault.<br>0h = Disable<br>1h = Enable |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.39 GCTL (Offset = 1124h) [Reset = 00000000h]

GCTL is shown in Figure 35-79 and described in Table 35-66.

Return to the Summary Table.

Global control register

**Figure 35-79. GCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | SHDWLDEN |
| R-0h | | | | | | | R/W-0h |

**Table 35-66. GCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | SHDWLDEN | R/W | 1h | Enables shadow to active load of bufferred registers and register fields.<br>0h = Disable<br>1h = Enable |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.40 CTR (Offset = 1800h) [Reset = 00000000h]

CTR is shown in Figure 35-80 and described in Table 35-67.

Return to the Summary Table.

This is the TIMER counter register.
This can be set by SW. However, the writes will be unpredictable if the software tries to set a value while the counter is running.

**Figure 35-80. CTR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | CCTR | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | R/W-0h | | | | | | | | | | | | | | | |

**Table 35-67. CTR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15-0 | CCTR | R/W | 0h | Current Counter value<br>0h = Minimum value<br>FFFFFFFFh = Maximum Value |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.41 CTRCTL (Offset = 1804h) [Reset = 0000FF80h]

CTRCTL is shown in Figure 35-81 and described in Table 35-68.

Return to the Summary Table.

This register provides control over the counter operation.
The configuration can change as well as setting the EN bit in a single write.
There is no requirement to change the configuration first and then do an additional write to set the EN bit.

**Figure 35-81. CTRCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | CVAE | | RESERVED | | | PLEN |
| R-0h | | R/W-0h | | R-0h | | | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| SLZERCNEZ | RESERVED | | | FRB | FB | DRB | RESERVED |
| R/W-0h | R-0h | | | R/W-0h | R/W-0h | R/W-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CZC | | | CAC | | | CLC | |
| R/W-7h | | | R/W-7h | | | R/W-7h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CLC | RESERVED | CM | | REPEAT | | | EN |
| R/W-7h | R-0h | R/W-0h | | R/W-0h | | | R/W-0h |

**Table 35-68. CTRCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | RESERVED | R | 0h | |
| 29-28 | CVAE | R/W | 0h | Counter Value After Enable. This field specifies the initialization condition of the counter when the EN bit is changed from 0 to 1 by a write to the CTRCTL register. Note that an external event can also cause the EN bit to go active.<br>0h = The counter is set to the LOAD register value<br>1h = The counter value is unchanged from its current value which could have been initialized by software<br>2h = The counter is set to zero |
| 27-25 | RESERVED | R | 0h | |
| 24 | PLEN | R/W | 0h | Phase Load Enable. This bit allows the timer to have phase load feature.<br>0h = Disabled<br>1h = Enabled |
| 23 | SLZERCNEZ | R/W | 0h | Suppress Load and Zero Events if Repeat Counter is Not Equal to Zero.<br>This bit suppresses the generation of the Z (zero) and L (load) events from the counter when the repeat counter (RC) value is not 0.<br>0h = Disabled. Z and L events are always generated from the counter when their conditions are generated.<br>1h = Enabled. Z and L events are generated from the counter when their conditions are generated and the RC register value is 0. |
| 22-20 | RESERVED | R | 0h | |
| 19 | FRB | R/W | 0h | Fault Resume Behavior This bit specifies what the device does following the release/exit of fault condition.<br>0h = Resume counting<br>1h = Perform the action as specified by the CVAE field. |



Copyright © 2026 Texas Instruments Incorporated

**Table 35-68. CTRCTL Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 18 | FB | R/W | 0h | Fault Behavior This bit specifies whether the counter continues running or suspends during a fault mode. There is a separate control under REPEAT to indicate whether counting is to suspend at next Counter==0<br>0h = Continues counting<br>1h = Suspends counting |
| 17 | DRB | R/W | 0h | Debug Resume Behavior This bit specifies what the device does following the release/exit of debug mode.<br>0h = Resume counting<br>1h = Perform the action as specified by the CVAE field. |
| 16 | RESERVED | R | 0h | |
| 15-13 | CZC | R/W | 7h | Counter Zero Control This field specifies what controls the counter operation with respect to zeroing the counter value.<br>Encodings 1-3 are present based on the CCPC parameter value. Bits 4-5 are present based on the HQEI parameter value. Any encodings not provided are documented as reserved.<br>0h = CCCTL_0 ZCOND<br>1h = CCCTL_1 ZCOND<br>2h = CCCTL_2 ZCOND<br>This value exists when there are 4 channels.<br>3h = CCCTL_3 ZCOND<br>This value exists when there are 4 channels.<br>4h = Controlled by 2-input QEI mode<br>This value exists when TIMER support QEI feature.<br>5h = Controlled by 3-input QEI mode<br>This value exists when TIMER support QEI feature. |
| 12-10 | CAC | R/W | 7h | Counter Advance Control. This field specifies what controls the counter operation with respect to advancing (incrementing or decrementing) the counter value.<br>Encodings 1-3 are present based on the CCPC parameter value. Bits 4-5 are present based on the HQEI parameter value. Any encodings not provided are documented as reserved.<br>0h = CCCTL_0 ACOND<br>1h = CCCTL_1 ACOND<br>2h = CCCTL_2 ACOND<br>This value exists when there are 4 channels.<br>3h = CCCTL_3 ACOND<br>This value exists when there are 4 channels.<br>4h = Controlled by 2-input QEI mode<br>This value exists when TIMER support QEI feature.<br>5h = Controlled by 3-input QEI mode<br>This value exists when TIMER support QEI feature. |
| 9-7 | CLC | R/W | 7h | Counter Load Control. This field specifies what controls the counter operation with respect to setting the counter to the LD register value.<br>Encodings 1-3 are present based on the CCPC parameter value. Bits 4-5 are present based on the HQEI parameter value. Any encodings not provided are documented as reserved.<br>0h = CCCTL_0 LCOND<br>1h = CCCTL_1 LCOND<br>2h = CCCTL_2 LCOND<br>This value exists when there are 4 channels.<br>3h = CCCTL_3 LCOND<br>This value exists when there are 4 channels.<br>4h = Controlled by 2 input QEI mode.<br>This value exists when TIMER support QEI feature.<br>5h = Controlled by 3 input QEI mode.<br>This value exists when TIMER support QEI feature. |
| 6 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

**Table 35-68. CTRCTL Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 5-4 | CM | R/W | 0h | Count Mode<br>0h = Down<br>1h = Up/Down<br>2h = Counter counts up. |
| 3-1 | REPEAT | R/W | 0h | Repeat. The repeat bit controls whether the counter continues to advance following a zero event, or the exiting of a debug or fault condition. If counting down, a zero event is followed by a load at the next advance condition. If counting up-down, a zero event is followed by an advance event (+1). The intent of encoding 3 is that if the debug condition is in effect, the generation of the load pulse is deferred until the debug condition is over. This allows the counter to reach zero before counting is suspended.<br>0h = Does not automatically advance following a zero event.<br>1h = Continues to advance following a zero event.<br>2h = Reserved<br>3h = Continues to advance following a zero event if the debug mode is not in effect, or following the release of the debug mode.<br>4h = Reserved |
| 0 | EN | R/W | 0h | Counter Enable. This bit allows the timer to advance This bit is automatically cleared if REPEAT=0 (do not automatically reload) and the counter value equals zero. CPU Write: A register write that sets the EN bit, the counter value is set per the CVAE value. Hardware: This bit may also be set as the result of an LCOND or ZCOND condition being met and the counter value changed to the load value or zero value, respectively.<br>0h = Disabled<br>1h = Enabled |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.42 LOAD (Offset = 1808h) [Reset = 00000000h]

LOAD is shown in Figure 35-82 and described in Table 35-69.

Return to the Summary Table.

The contents of LOAD register are copied to CTR on any operation designated to do a "LOAD". The LOAD is used to compare with the CTR for generating a "Load Event" that can be used for interrupt, trigger, or signal generator actions.

**Figure 35-82. LOAD**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | LD | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | R/W-0h | | | | | | | | | | | | | | | |

**Table 35-69. LOAD Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15-0 | LD | R/W | 0h | Load Value<br>0h = Minimum value<br>FFFFFFFFh = Maximum Value |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.43 CC_01[y] (Offset = 1810h + formula) [Reset = 00000000h]

CC_01[y] is shown in Figure 35-83 and described in Table 35-70.

Return to the Summary Table.

The CC_01 register is a register that can be used as either a capture register, to capture the next CTR value on an event, or a compare to the current CTR to create an event. It cannot operate concurrently as both. There are two Capture-Compare slices of hardware for each counter, hence there are two CC_01 registers per timer. On a capture event, the next value of the CTR is loaded so that CTR and CC_01 (which captured) will be equal on the cycle that an interrupt or trigger is created from the capture action.

Offset = 1810h + (y * 4h); where y = 0h to 1h

**Figure 35-83. CC_01[y]**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | CCVAL | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | R/W-0h | | | | | | | | | | | | | | | |

**Table 35-70. CC_01[y] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15-0 | CCVAL | R/W | 0h | Capture or compare value<br>0h = Minimum value<br>FFFFFFFFh = Maximum Value |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.44 CC_23[y] (Offset = 1818h + formula) [Reset = 00000000h]

CC_23[y] is shown in Figure 35-84 and described in Table 35-71.

Return to the Summary Table.

The CC_23 register is a register that can be used as either a capture register, to capture the next CTR value on an event, or a compare to the current CTR to create an event. It cannot operate concurrently as both. There are two Capture-Compare slices of hardware for each counter, hence there are two CC_01 registers per timer. On a capture event, the next value of the CTR is loaded so that CTR and CC_01 (which captured) will be equal on the cycle that an interrupt or trigger is created from the capture action.

Offset = 1818h + (y * 4h); where y = 0h to 1h

**Figure 35-84. CC_23[y]**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | CCVAL | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | R/W-0h | | | | | | | | | | | | | | | |

**Table 35-71. CC_23[y] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15-0 | CCVAL | R/W | 0h | Capture or compare value<br>0h = Minimum value<br>FFFFFFFFh = Maximum Value |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.45 CC_45[y] (Offset = 1820h + formula) [Reset = 00000000h]

CC_45[y] is shown in Figure 35-85 and described in Table 35-72.

Return to the Summary Table.

The CC_45 register are a registers which can be used as compare to the current CTR to create an events CC4U, CC4D, CC5U and CC5D.

Offset = 1820h + (y * 4h); where y = 0h to 1h

**Figure 35-85. CC_45[y]**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | CCVAL | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | R/W-0h | | | | | | | | | | | | | | | |

**Table 35-72. CC_45[y] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15-0 | CCVAL | R/W | 0h | Capture or compare value<br>0h = Minimum value<br>FFFFFFFFh = Maximum Value |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.46 CCCTL_01[y] (Offset = 1830h + formula) [Reset = 00000000h]

CCCTL_01[y] is shown in Figure 35-86 and described in Table 35-73.

Return to the Summary Table.

The CCCTL_01 registers control the operations of the respective CC registers and the counter.

Offset = 1830h + (y * 4h); where y = 0h to 1h

**Figure 35-86. CCCTL_01[y]**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CC2SELD | | | CCACTUPD | | | SCERCNEZ | CC2SELU |
| R/W-0h | | | R/W-0h | | | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CC2SELU | | RESERVED | CCUPD | | | COC | RESERVED |
| R/W-0h | | R-0h | R/W-0h | | | R/W-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | ZCOND | | | RESERVED | LCOND | | |
| R-0h | R/W-0h | | | R-0h | R/W-0h | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | ACOND | | | RESERVED | CCOND | | |
| R-0h | R/W-0h | | | R-0h | R/W-0h | | |

**Table 35-73. CCCTL_01[y] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | CC2SELD | R/W | 0h | Selects the source second CCD event.<br>0h = Selects CCD from CC0.<br>1h = Selects CCD from CC1.<br>2h = Selects CCD from CC2.<br>3h = Selects CCD from CC3.<br>4h = Selects CCD from CC4.<br>5h = Selects CCD from CC5. |



Copyright © 2026 Texas Instruments Incorporated

**Table 35-73. CCCTL_01[y] Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 28-26 | CCACTUPD | R/W | 0h | CCACT shadow register Update Method<br>This field controls how updates to the CCACT shadow register are performed<br>0h = Value written to the CCACT register has immediate effect.<br>1h = Following a zero event (CTR=0)<br>Writes to the CCACTx_y<br>register are stored in<br>shadow register and transferred to<br>CCACTx_y in the TIMCLK<br>cycle following CTR<br>equals 0.<br>2h = Following a CCD event (CTR=CC_xy)<br>Writes to the CCACTx_y register are stored in<br>shadow register and transferred to<br>CCACTx_y in the TIMCLK<br>cycle following CTR<br>equals the CCx_y register value.<br>3h = Following a CCU event (CTR=CC_xy)<br>Writes to the CCACTx_y<br>register are stored in<br>shadow register and transferred to<br>CCACTx_y in the TIMCLK<br>cycle following CTR<br>equals the CCx_y register<br>value.<br>4h = Following a zero event (CTR=0) or load event (CTR = LOAD)<br>Writes to the CCACTx_y<br>register are stored in<br>shadow register and transferred to<br>CCACTx_y in the TIMCLK<br>cycle following CTR<br>equals 0 or CTR. Equals<br>LDn.<br>Note this update mechanism is<br>defined for use only in<br>configurations using up/down<br>counting. This mode is not intended for use in down count configurations.<br>5h = Following a zero event (CTR=0) with repeat count also zero (RC=0).<br>Writes to the CCACTx_y<br>register are stored in<br>shadow register and transferred to<br>CCACTx_y in the TIMCLK<br>cycle following CTR<br>equals 0 and if RC equal<br>0.<br>6h = On a TRIG pulse, the value stored in CCACT_xy shadow register is loaded into CCACT_xy register. |
| 25 | SCERCNEZ | R/W | 0h | Suppress Compare Event if Repeat Counter is Not Equal to Zero<br>This bit suppresses the generation of the compare (CCD, CCU and RC) events from<br>the counter when the repeat counter (RC) value is not 0.<br>0h = CCD, CCU and<br>RC events are always generated<br>from the counter when their<br>conditions are generated.<br>1h = CCD, CCU and<br>RC events are generated from the<br>counter when their conditions are<br>generated and the RC<br>register value is 0. |



Copyright © 2026 Texas Instruments Incorporated

**Table 35-73. CCCTL_01[y] Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 24-22 | CC2SELU | R/W | 0h | Selects the source second CCU event.<br>0h = Selects CCU from CC0.<br>1h = Selects CCU from CC1.<br>2h = Selects CCU from CC2.<br>3h = Selects CCU from CC3.<br>4h = Selects CCU from CC4.<br>5h = Selects CCU from CC5. |
| 21 | RESERVED | R | 0h | |
| 20-18 | CCUPD | R/W | 0h | Capture and Compare Update Method<br>This field controls how updates to the shadow<br>capture and compare register are performed<br>(when operating in compare mode, COC=0).<br>0h = Writes to the CCx_y register is written to the register directly<br>and has immediate effect.<br>1h = Following a zero event (CTR=0)<br>Writes to the CCx_y<br>register are stored in<br>shadow register and transferred to<br>CCx_y in the TIMCLK<br>cycle following CTR<br>equals 0.<br>2h = Following a CCD event (CTR=CC_xy)<br>Writes to the CCx_y register are stored in<br>shadow register and transferred to<br>CCx_y in the TIMCLK<br>cycle following CTR<br>equals the CCx_y register value.<br>3h = Following a CCU event (CTR=CC_xy)<br>Writes to the CCx_y<br>register are stored in<br>shadow register and transferred to<br>CCx_y in the TIMCLK<br>cycle following CTR<br>equals the CCx_y register<br>value.<br>4h = Following a zero event(CTR=0) or load event (CTR=LOAD)<br>Writes to the CCx_y<br>register are stored in<br>shadow register and transferred to<br>ECCx_y in the TIMCLK<br>cycle following CTR<br>equals 0 or CTR. Equals<br>LD.<br>Note this update mechanism is<br>defined for use only in<br>configurations using up/down<br>counting. This mode is not intended for use in down count<br>configurations.<br>5h = Following a zero event (CTR=0) with repeat count also zero<br>(RC=0).<br>Writes to the CCx_y<br>register are stored in<br>shadow register and transferred to<br>CCx_y in the TIMCLK<br>cycle following CTR<br>equals 0 and if RC equal<br>0.<br>6h = Following a TRIG pulse.<br>Writes to the CCx_y<br>register are stored in<br>shadow register and transferred to<br>CCx_y |



Copyright © 2026 Texas Instruments Incorporated

**Table 35-73. CCCTL_01[y] Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 17 | COC | R/W | 0h | Capture or Compare.<br>Specifies whether the corresponding CC register is used as a capture register or a compare register (never both).<br>0h = Compare<br>1h = Capture |
| 16-15 | RESERVED | R | 0h | |
| 14-12 | ZCOND | R/W | 0h | Zero Condition.<br>This field specifies the condition that generates a zero pulse.<br>0h = CCP edges have no effect<br>1h = Rising edge of CCP or trigger assertion edge<br>2h = Falling edge of CCP or trigger de-assertion edge<br>3h = Either edge of CCP or trigger change (assertion/de-assertion edge) |
| 11 | RESERVED | R | 0h | |
| 10-8 | LCOND | R/W | 0h | Load Condition.<br>Specifies the condition that generates a load pulse.<br>0h = CCP edges have no effect<br>1h = Rising edge of CCP or trigger assertion edge<br>2h = Falling edge of CCP or trigger de-assertion edge<br>3h = Either edge of CCP or trigger change (assertion/de-assertion edge) |
| 7 | RESERVED | R | 0h | |
| 6-4 | ACOND | R/W | 0h | Advance Condition.<br>Specifies the condition that generates an advance pulse.<br>0h = Each TIMCLK<br>1h = Rising edge of CCP or trigger assertion edge<br>2h = Falling edge of CCP or trigger de-assertion edge<br>3h = Either edge of CCP or trigger change (assertion/de-assertion edge)<br>5h = CCP High or Trigger assertion (level) |
| 3 | RESERVED | R | 0h | |
| 2-0 | CCOND | R/W | 0h | Capture Condition.<br>Specifies the condition that generates a capture pulse.<br>0h = None (never captures)<br>1h = Rising edge of CCP or trigger assertion edge<br>2h = Falling edge of CCP or trigger de-assertion edge<br>3h = Either edge of CCP or trigger change (assertion/de-assertion edge) |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.47 CCCTL_23[y] (Offset = 1838h + formula) [Reset = 00000000h]

CCCTL_23[y] is shown in Figure 35-87 and described in Table 35-74.

Return to the Summary Table.

The CCCTL registers control the operations of the respective CC registers and the counter.

Offset = 1838h + (y * 4h); where y = 0h to 1h

**Figure 35-87. CCCTL_23[y]**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CC2SELD | | | CCACTUPD | | | SCERCNEZ | CC2SELU |
| R/W-0h | | | R/W-0h | | | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CC2SELU | | RESERVED | CCUPD | | | COC | RESERVED |
| R/W-0h | | R-0h | R/W-0h | | | R/W-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | ZCOND | | | RESERVED | LCOND | | |
| R-0h | R/W-0h | | | R-0h | R/W-0h | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | ACOND | | | RESERVED | CCOND | | |
| R-0h | R/W-0h | | | R-0h | R/W-0h | | |

**Table 35-74. CCCTL_23[y] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-29 | CC2SELD | R/W | 0h | Selects the source second CCD event.<br>0h = Selects CCD from CC0.<br>1h = Selects CCD from CC1.<br>2h = Selects CCD from CC2.<br>3h = Selects CCD from CC3.<br>4h = Selects CCD from CC4.<br>5h = Selects CCD from CC5. |



Copyright © 2026 Texas Instruments Incorporated

**Table 35-74. CCCTL_23[y] Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 28-26 | CCACTUPD | R/W | 0h | CCACT shadow register Update Method<br>This field controls how updates to the CCCACT shadow register are performed<br>0h = Value written to the CCACTx_y register has immediate effect.<br>1h = Following a zero event (CTR=0)<br>Writes to the CCACTx_y register are stored in shadow register and transferred to CCACTx_y in the TIMCLK cycle following CTR equals 0.<br>2h = Following a CCD event (CTR=CC_xy)<br>Writes to the CCACTx_y register are stored in shadow register and transferred to CCACTx_y in the TIMCLK cycle following CTR equals the CCx_y register value.<br>3h = Following a CCU event (CTR=cc_xy)<br>Writes to the CCACTx_y register are stored in shadow register and transferred to CCACTx_y in the TIMCLK cycle following CTR equals the CCx_y register value.<br>4h = Following a zero event (CTR=0) or load event (CTR=LOAD)<br>Writes to the CCACTx_y register are stored in shadow register and transferred to CCACTx_y in the TIMCLK cycle following CTR equals 0 or CTR. Equals LDn.<br>Note this update mechanism is defined for use only in configurations using up/down counting. This mode is not intended for use in down count configurations.<br>5h = Following a zero event (CTR=0) with repeat count also zero (RC=0).<br>Writes to the CCACTx_y register are stored in shadow register and transferred to CCACTx_y in the TIMCLK cycle following CTR equals 0 and if RC equal 0.<br>6h = On a TRIG pulse, the value stored in CCACTx_y shadow register is loaded into CCACTx_y active register. |
| 25 | SCERCNEZ | R/W | 0h | Suppress Compare Event if Repeat Counter is Not Equal to Zero<br>This bit suppresses the generation of the compare (CCD, CCU and RC) events from the counter when the repeat counter (RCn) value is not 0.<br>0h = CCD, CCU and RC events are always generated from the counter when their conditions are generated.<br>1h = CCD, CCU and RC events are generated from the counter when their conditions are generated and the RC register value is 0. |



Copyright © 2026 Texas Instruments Incorporated

**Table 35-74. CCCTL_23[y] Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 24-22 | CC2SELU | R/W | 0h | Selects the source second CCU event.<br>0h = Selects CCU from CC0.<br>1h = Selects CCU from CC1.<br>2h = Selects CCU from CC2.<br>3h = Selects CCU from CC3.<br>4h = Selects CCU from CC4.<br>5h = Selects CCU from CC5. |
| 21 | RESERVED | R | 0h | |
| 20-18 | CCUPD | R/W | 0h | Capture and Compare Update Method<br>This field controls how updates to the shadow capture and compare register are performed (when operating in compare mode, COC=0).<br>0h = Writes to the CCx_y register is written to the register directly and has immediate effect.<br>1h = Following a zero event (CTR=0) Writes to the CCx_y register are stored in shadow register and transferred to CCx_y in the TIMCLK cycle following CTR equals 0.<br>2h = Following a CCD event (CTR=CC_xy) Writes to the CCx_y register are stored in shadow register and transferred to CCx_y in the TIMCLK cycle following CTR equals the CCx_y register value.<br>3h = Following a CCU event (CTR=CC_xy) Writes to the CCx_y register are stored in shadow register and transferred to CCx_y in the TIMCLK cycle following CTR equals the CCx_y register value.<br>4h = Following a zero or load event Writes to the CCx_y register are stored in shadow register and transferred to CCx_y in the TIMCLK cycle following CTR equals 0 or CTR. Equals LDn.<br>Note this update mechanism is defined for use only in configurations using up/down counting. This mode is not intended for use in down count configurations.<br>5h = Following a zero event (CTR=0) with repeat count also zero (RC=0). Writes to the CCx_y register are stored in shadow register and transferred to CCx_y in the TIMCLK cycle following CTR equals 0 and if RC equal 0.<br>6h = Following a TRIG pulse. Writes to the CCx_y register are stored in shadow register and transferred to CCx_y #xD; 0. |



Copyright © 2026 Texas Instruments Incorporated

**Table 35-74. CCCTL_23[y] Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 17 | COC | R/W | 0h | Capture or Compare.<br>Specifies whether the corresponding CC register is used as a capture register or a compare register (never both).<br>0h = Compare<br>1h = Capture |
| 16-15 | RESERVED | R | 0h | |
| 14-12 | ZCOND | R/W | 0h | Zero Condition.<br>This field specifies the condition that generates a zero pulse. 4h-Fh = Reserved<br>0h = CCP edges have no effect<br>1h = Rising edge of CCP or trigger assertion edge<br>2h = Falling edge of CCP or trigger de-assertion edge<br>3h = Either edge of CCP or trigger change (assertion/de-assertion edge) |
| 11 | RESERVED | R | 0h | |
| 10-8 | LCOND | R/W | 0h | Load Condition.<br>Specifies the condition that generates a load pulse. 4h-Fh = Reserved<br>0h = CCP edges have no effect<br>1h = Rising edge of CCP or trigger assertion edge<br>2h = Falling edge of CCP or trigger de-assertion edge<br>3h = Either edge of CCP or trigger change (assertion/de-assertion edge) |
| 7 | RESERVED | R | 0h | |
| 6-4 | ACOND | R/W | 0h | Advance Condition.<br>Specifies the condition that generates an advance pulse. 6h-Fh = Reserved<br>0h = Each TIMCLK<br>1h = Rising edge of CCP or trigger assertion edge<br>2h = Falling edge of CCP or trigger de-assertion edge<br>3h = Either edge of CCP or trigger change (assertion/de-assertion edge)<br>5h = CCP High or Trigger assertion (level) |
| 3 | RESERVED | R | 0h | |
| 2-0 | CCOND | R/W | 0h | Capture Condition.<br>Specifies the condition that generates a capture pulse. 4h-Fh = Reserved<br>0h = None (never captures)<br>1h = Rising edge of CCP or trigger assertion edge<br>2h = Falling edge of CCP or trigger de-assertion edge<br>3h = Either edge of CCP or trigger change (assertion/de-assertion edge) |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.48 CCCTL_45[y] (Offset = 1840h + formula) [Reset = 00000000h]

CCCTL_45[y] is shown in Figure 35-88 and described in Table 35-75.

Return to the Summary Table.

The CCCTL registers control the operations of the respective CC registers and the counter.

Offset = 1840h + (y * 4h); where y = 0h to 1h

**Figure 35-88. CCCTL_45[y]**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | SCERCNEZ | RESERVED |
| R-0h | | | | | | R/W-0h | R-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | CCUPD | | | RESERVED | |
| R-0h | | | R/W-0h | | | R-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |

**Table 35-75. CCCTL_45[y] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-26 | RESERVED | R | 0h | |
| 25 | SCERCNEZ | R/W | 0h | Suppress Compare Event if Repeat Counter is Not Equal to Zero<br>This bit suppresses the generation of the compare (CCD, CCU and RC) events from<br>the counter when the repeat counter (RC) value is not 0.<br>0h = CCD, CCU and<br>RC events are always generated<br>from the counter when their<br>conditions are generated.<br>1h = CCD, CCU and<br>RC events are generated from the<br>counter when their conditions are<br>generated and the RC<br>register value is 0. |
| 24-21 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

**Table 35-75. CCCTL_45[y] Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 20-18 | CCUPD | R/W | 0h | Capture and Compare Update Method<br>This field controls how updates to the shadow capture and compare register are performed (when operating in compare mode, COC=0).<br>0h = Writes to the CCx_y register is written to the register directly and has immediate effect.<br>1h = Following a zero event (CTR=0)<br>Writes to the CCx_y register are stored in shadow register and transferred to ECCx_y in the TIMCLK cycle following CTR equals 0.<br>2h = Following a CCD event (CTR=CC_xy)<br>Writes to the CCx_y register are stored in shadow register and transferred to CCx_y in the TIMCLK cycle following CTR equals the CCx_y register value.<br>3h = Following a CCU event (CTR=CC_xy)<br>Writes to the CCx_y register are stored in shadow register and transferred to CCx_y in the TIMCLK cycle following CTR equals the CCx_y register value.<br>4h = Following a zero event (CTR=0) or load event (CTR=LOAD)<br>Writes to the CCx_y register are stored in shadow register and transferred to CCx_y in the TIMCLK cycle following CTR equals 0 or CTR. Equals LD.<br>Note this update mechanism is defined for use only in configurations using up/down counting. This mode is not intended for use in down count configurations.<br>5h = Following a zero event (CTR=0) with repeat count also zero (RC=0).<br>Writes to the CCx_y register are stored in shadow register and transferred to CCx_y in the TIMCLK cycle following CTR equals 0 and if RC equal 0.<br>6h = Following a TRIG pulse.<br>Writes to the CCx_y register are stored in shadow register and transferred to CCx_y #xD; 0. |
| 17-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.49 OCTL_01[y] (Offset = 1850h + formula) [Reset = 00000000h]

OCTL_01[y] is shown in Figure 35-89 and described in Table 35-76.

Return to the Summary Table.

The OCTL_01 register controls the CCP output of the Capture-Compare slice of the counter. This includes the ability to select the source of what is driven out along with initial condition values and final inversion options.

Offset = 1850h + (y * 4h); where y = 0h to 1h

**Figure 35-89. OCTL_01[y]**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | CCPIV | CCPOINV | CCPO | | | |
| R-0h | | R/W-0h | R/W-0h | R/W-0h | | | |

**Table 35-76. OCTL_01[y] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | RESERVED | R | 0h | |
| 5 | CCPIV | R/W | 0h | CCP Initial Value This bit specifies the logical value put on the signal generator state while the counter is disabled (CTRCTL.EN == 0).<br>0h = Low<br>1h = High |
| 4 | CCPOINV | R/W | 0h | CCP Output Invert The output as selected by CCPO is conditionally inverted.<br>0h = No inversion<br>1h = Invert |
| 3-0 | CCPO | R/W | 0h | CCP Output Source<br>0h = Signal generator value (for example, PWM, triggered PWM)<br>1h = Load event<br>2h = CCU event or CCD event<br>4h = Zero event<br>5h = Capture event<br>6h = Fault condition<br>8h = Mirror CCP of first capture and compare register to other capture compare blocks<br>9h = Mirror CCP of second capture and compare register in other capture compare blocks<br>Ch = Signal generator output after deadband insertion<br>Dh = Counter direction |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.50 OCTL_23[y] (Offset = 1858h + formula) [Reset = 00000000h]

OCTL_23[y] is shown in Figure 35-90 and described in Table 35-77.

Return to the Summary Table.

The OCTL register controls the CCP output of the Capture-Compare slice of the counter. This includes the ability to select the source of what is driven out along with initial condition values and final inversion options.

Offset = 1858h + (y * 4h); where y = 0h to 1h

**Figure 35-90. OCTL_23[y]**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | CCPIV | CCPOINV | CCPO | | | |
| R-0h | | R/W-0h | R/W-0h | R/W-0h | | | |

**Table 35-77. OCTL_23[y] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-6 | RESERVED | R | 0h | |
| 5 | CCPIV | R/W | 0h | CCP Initial Value This bit specifies the logical value put on the signal generator state while the counter is disabled (CTRCTL.EN == 0).<br>0h = Low<br>1h = High |
| 4 | CCPOINV | R/W | 0h | CCP Output Invert The output as selected by CCPO is conditionally inverted.<br>0h = No inversion<br>1h = Invert |
| 3-0 | CCPO | R/W | 0h | CCP Output Source<br>0h = Signal generator value (for example, PWM, triggered PWM)<br>1h = Load condition<br>2h = CCU event or CCD event<br>4h = Zero event<br>5h = Capture event<br>6h = Fault Condition<br>8h = Mirror CCP of first capture and compare register in other capture compare blocks<br>9h = Mirror CCP of second capture and compare register in other capture compare blocksi /bn,.<br>Ch = Deadband Inserted Output<br>Dh = Counter direction |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.51 CCACT_01[y] (Offset = 1870h + formula) [Reset = 00000000h]

CCACT_01[y] is shown in Figure 35-91 and described in Table 35-78.

Return to the Summary Table.

The CCACT_01 register controls the actions of the signal generator of the capture-compare slice based on the events created in the counter block, the capture and compare block and debug events.

Offset = 1870h + (y * 4h); where y = 0h to 1h

**Figure 35-91. CCACT_01[y]**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| SWFRCACT_CMPL | | SWFRCACT | | FEXACT | | | FENACT |
| R/W-0h | | R/W-0h | | R/W-0h | | | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| FENACT | | RESERVED | | | | | CC2UACT |
| R/W-0h | | R-0h | | | | | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CC2UACT | RESERVED | CC2DACT | | RESERVED | CUACT | | RESERVED |
| R/W-0h | R-0h | R/W-0h | | R-0h | R/W-0h | | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CDACT | | RESERVED | LACT | | RESERVED | ZACT | |
| R/W-0h | | R-0h | R/W-0h | | R-0h | R/W-0h | |

**Table 35-78. CCACT_01[y] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | SWFRCACT_CMPL | R/W | 0h | CCP Complimentary output Action on Software Force Output<br>This field describes the resulting action of software force.<br>This action has a shadow register, which will be updated under specific condition.<br>So that this register cannot take into effect immediately.<br>0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event.<br>1h = CCP Complimentary output value is set high<br>2h = CCP Complimentary output value is set low |
| 29-28 | SWFRCACT | R/W | 0h | CCP Output Action on Software Force Output<br>This field describes the resulting action of software force.<br>This action has a shadow register, which will be updated under specific condition.<br>So that this register cannot take into effect immediately.<br>0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event.<br>1h = CCP output value is set high<br>2h = CCP output value is set low |
| 27-25 | FEXACT | R/W | 0h | CCP Output Action on Fault Exit<br>This field describes the resulting action of the signal generator upon exiting the fault condition.<br>0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event.<br>1h = CCP output value is set high<br>2h = CCP output value is set low<br>3h = CCP output value is toggled<br>4h = CCP output value is tristated |



Copyright © 2026 Texas Instruments Incorporated

**Table 35-78. CCACT_01[y] Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 24-22 | FENACT | R/W | 0h | CCP Output Action on Fault Entry<br>This field describes the resulting action of the signal generator upon detecting a fault.<br>0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event.<br>1h = CCP output value is set high<br>2h = CCP output value is set low<br>3h = CCP output value is toggled<br>4h = CCP output value is tristated |
| 21-17 | RESERVED | R | 0h | |
| 16-15 | CC2UACT | R/W | 0h | CCP Output Action on CC2U event.<br>0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event.<br>1h = CCP output value is set high<br>2h = CCP output value is set low<br>3h = CCP output value is toggled |
| 14 | RESERVED | R | 0h | |
| 13-12 | CC2DACT | R/W | 0h | CCP Output Action on CC2D event.<br>0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event.<br>1h = CCP output value is set high<br>2h = CCP output value is set low<br>3h = CCP output value is toggled |
| 11 | RESERVED | R | 0h | |
| 10-9 | CUACT | R/W | 0h | CCP Output Action on Compare (Up) This field describes the resulting action of the signal generator upon detecting a compare event while counting up.<br>0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event.<br>1h = CCP output value is set high<br>2h = CCP output value is set low<br>3h = CCP output value is toggled |
| 8 | RESERVED | R | 0h | |
| 7-6 | CDACT | R/W | 0h | CCP Output Action on Compare (Down) This field describes the resulting action of the signal generator upon detecting a compare event while counting down.<br>0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event.<br>1h = CCP output value is set high<br>2h = CCP output value is set low<br>3h = CCP output value is toggled |
| 5 | RESERVED | R | 0h | |
| 4-3 | LACT | R/W | 0h | CCP Output Action on Load Specifies what changes occur to CCP output as the result of a load event.<br>0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event.<br>1h = CCP output value is set high<br>2h = CCP output value is set low<br>3h = CCP output value is toggled |
| 2 | RESERVED | R | 0h | |
| 1-0 | ZACT | R/W | 0h | CCP Output Action on Zero Specifies what changes occur to CCP output as the result of a zero event.<br>0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event.<br>1h = CCP output value is set high<br>2h = CCP output value is set low<br>3h = CCP output value is toggled |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.52 CCACT_23[y] (Offset = 1878h + formula) [Reset = 00000000h]

CCACT_23[y] is shown in Figure 35-92 and described in Table 35-79.

Return to the Summary Table.

The CCACT register controls the actions of the signal generator of the capture-compare slice based on the events created in the counter block, the capture and compare block and debug events.

Offset = 1878h + (y * 4h); where y = 0h to 1h

**Figure 35-92. CCACT_23[y]**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| SWFRCACT_CMPL | | SWFRCACT | | FEXACT | | | FENACT |
| R/W-0h | | R/W-0h | | R/W-0h | | | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| FENACT | | RESERVED | | | | | CC2UACT |
| R/W-0h | | R-0h | | | | | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CC2UACT | RESERVED | CC2DACT | | RESERVED | CUACT | | RESERVED |
| R/W-0h | R-0h | R/W-0h | | R-0h | R/W-0h | | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CDACT | | RESERVED | LACT | | RESERVED | ZACT | |
| R/W-0h | | R-0h | R/W-0h | | R-0h | R/W-0h | |

**Table 35-79. CCACT_23[y] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-30 | SWFRCACT_CMPL | R/W | 0h | CCP Complimentary Output Action on Software Force Output<br>This field describes the resulting action of software force.<br>This action has a shadow register, which will be updated under specific condition.<br>So that this register cannot take into effect immediately.<br>0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event.<br>1h = CCP Complimentary output value is set high<br>2h = CCP Complimentary output value is set low |
| 29-28 | SWFRCACT | R/W | 0h | CCP Output Action on Software Force Output<br>This field describes the resulting action of software force.<br>This action has a shadow register, which will be updated under specific condition.<br>So that this register cannot take into effect immediately.<br>0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event.<br>1h = CCP output value is set high<br>2h = CCP output value is set low |
| 27-25 | FEXACT | R/W | 0h | CCP Output Action on Fault Exit<br>This field describes the resulting action of the signal generator upon exiting the fault condition.<br>0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event.<br>1h = CCP output value is set high<br>2h = CCP output value is set low<br>3h = CCP output value is toggled<br>4h = CCP output value is tristated |



Copyright © 2026 Texas Instruments Incorporated

**Table 35-79. CCACT_23[y] Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 24-22 | FENACT | R/W | 0h | CCP Output Action on Fault Entry<br>This field describes the resulting action of the signal generator upon detecting a fault.<br>0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event.<br>1h = CCP output value is set high<br>2h = CCP output value is set low<br>3h = CCP output value is toggled<br>4h = CCP output value is tristated |
| 21-17 | RESERVED | R | 0h | |
| 16-15 | CC2UACT | R/W | 0h | CCP Output Action on CC2U event.<br>0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event.<br>1h = CCP output value is set high<br>2h = CCP output value is set low<br>3h = CCP output value is toggled |
| 14 | RESERVED | R | 0h | |
| 13-12 | CC2DACT | R/W | 0h | CCP Output Action on CC2D event.<br>0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event.<br>1h = CCP output value is set high<br>2h = CCP output value is set low<br>3h = CCP output value is toggled |
| 11 | RESERVED | R | 0h | |
| 10-9 | CUACT | R/W | 0h | CCP Output Action on Compare (Up) This field describes the resulting action of the signal generator upon detecting a compare event while counting up.<br>0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event.<br>1h = CCP output value is set high<br>2h = CCP output value is set low<br>3h = CCP output value is toggled |
| 8 | RESERVED | R | 0h | |
| 7-6 | CDACT | R/W | 0h | CCP Output Action on Compare (Down) This field describes the resulting action of the signal generator upon detecting a compare event while counting down.<br>0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event.<br>1h = CCP output value is set high<br>2h = CCP output value is set low<br>3h = CCP output value is toggled |
| 5 | RESERVED | R | 0h | |
| 4-3 | LACT | R/W | 0h | CCP Output Action on Load Specifies what changes occur to CCP output as the result of a load event.<br>0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event.<br>1h = CCP output value is set high<br>2h = CCP output value is set low<br>3h = CCP output value is toggled |
| 2 | RESERVED | R | 0h | |
| 1-0 | ZACT | R/W | 0h | CCP Output Action on Zero Specifies what changes occur to CCP output as the result of a zero event.<br>0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event.<br>1h = CCP output value is set high<br>2h = CCP output value is set low<br>3h = CCP output value is toggled |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.53 IFCTL_01[y] (Offset = 1880h + formula) [Reset = 00000000h]

IFCTL_01[y] is shown in Figure 35-93 and described in Table 35-80.

Return to the Summary Table.

The IFCTL_01 register controls the input selection and inversion for the associated Capture-Compare slice.

Offset = 1880h + (y * 4h); where y = 0h to 1h

**Figure 35-93. IFCTL_01[y]**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | FE | CPV | RESERVED | FP | |
| R-0h | | | R/W-0h | R/W-0h | R-0h | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INV | RESERVED | | | ISEL | | | |
| R/W-0h | R-0h | | | R/W-0h | | | |

**Table 35-80. IFCTL_01[y] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-13 | RESERVED | R | 0h | |
| 12 | FE | R/W | 0h | Filter Enable<br>This bit controls whether the input is filtered by the input filter or bypasses to the edge detect.<br>0h = Bypass.<br>1h = Filtered. |
| 11 | CPV | R/W | 0h | Consecutive Period/Voting Select<br>This bit controls whether the input filter uses a stricter consecutive period count or majority voting.<br>0h = Consecutive Periods<br>The input must be at a specific logic level for the period defined by FP before it is passed to the filter output.<br>1h = Voting<br>The filter ignores one clock of opposite logic over the filter period.<br>I.e. Over FP samples of the input, up to 1 sample may be of an opposite logic value (glitch) without affecting the output. |
| 10 | RESERVED | R | 0h | |
| 9-8 | FP | R/W | 0h | Filter Period. This field specifies the sample period for the input filter. I.e. The input is sampled for FP timer clocks during filtering.<br>0h = The division factor is 3<br>1h = The division factor is 5<br>2h = The division factor is 8 |



Copyright © 2026 Texas Instruments Incorporated

**Table 35-80. IFCTL_01[y] Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7 | INV | R/W | 0h | Input Inversion This bit controls whether the selected input is inverted.<br>0h = Noninverted<br>1h = Inverted |
| 6-4 | RESERVED | R | 0h | |
| 3-0 | ISEL | R/W | 0h | Input Select (CCP0) This field selects the input source to the filter input. 4h-7h = Reserved<br>0h = CCP of the corresponding capture compare unit<br>1h = Input pair CCPX of the capture compare unit. For CCP0 input pair is CCP1 and for CCP1 input pair is CCP0.<br>2h = CCP0 of the counter<br>3h = Trigger<br>4h = XOR of CCP inputs as input source (Used in Hall input mode).<br>5h = subscriber 0 event as input source.<br>6h = subscriber 1 event as input source.<br>7h = Comparator 0 output.<br>8h = Comparator 1 output.<br>9h = Comparator 2 output. |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.54 IFCTL_23[y] (Offset = 1888h + formula) [Reset = 00000000h]

IFCTL_23[y] is shown in Figure 35-94 and described in Table 35-81.

Return to the Summary Table.

The IFCTL register controls the input selection and inversion for the associated Capture-Compare slice.

Offset = 1888h + (y * 4h); where y = 0h to 1h

**Figure 35-94. IFCTL_23[y]**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | FE | CPV | RESERVED | FP | |
| R-0h | | | R/W-0h | R/W-0h | R-0h | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INV | RESERVED | | | ISEL | | | |
| R/W-0h | R-0h | | | R/W-0h | | | |

**Table 35-81. IFCTL_23[y] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-13 | RESERVED | R | 0h | |
| 12 | FE | R/W | 0h | Filter Enable<br>This bit controls whether the input is filtered by the input filter or bypasses to the edge detect.<br>0h = Bypass.<br>1h = Filtered. |
| 11 | CPV | R/W | 0h | Consecutive Period/Voting Select<br>This bit controls whether the input filter uses a stricter consecutive period count or majority voting.<br>0h = Consecutive Periods<br>The input must be at a specific logic level for the period defined by FP before it is passed to the filter output.<br>1h = Voting<br>The filter ignores one clock of opposite logic over the filter period.<br>I.e. Over FP samples of the input, up to 1 sample may be of an opposite logic value (glitch) without affecting the output. |
| 10 | RESERVED | R | 0h | |
| 9-8 | FP | R/W | 0h | Filter Period. This field specifies the sample period for the input filter. I.e. The input is sampled for FP timer clocks during filtering.<br>0h = The division factor is 3<br>1h = The division factor is 5<br>2h = The division factor is 8 |



Copyright © 2026 Texas Instruments Incorporated

**Table 35-81. IFCTL_23[y] Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7 | INV | R/W | 0h | Input Inversion This bit controls whether the selected input is inverted.<br>0h = Noninverted<br>1h = Inverted |
| 6-4 | RESERVED | R | 0h | |
| 3-0 | ISEL | R/W | 0h | Input Select (CCP0) This field selects the input source to the filter input. 4h-7h = Reserved<br>0h = CCP of the corresponding capture compare unit<br>1h = Input pair CCPX of the capture compare unit. For CCP0 input pair is CCP1 and for CCP1 input pair is CCP0.<br>2h = CCP0 of the counter<br>3h = Trigger<br>4h = XOR of CCP inputs as input source (Used in Hall input mode).<br>5h = subscriber 0 event as input source.<br>6h = subscriber 1 event as input source.<br>7h = Comparator 0 output.<br>8h = Comparator 1 output.<br>9h = Comparator 2 output. |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.55 PL (Offset = 18A0h) [Reset = 00000000h]

PL is shown in Figure 35-95 and described in Table 35-82.

Return to the Summary Table.

This is the phase load register.

**Figure 35-95. PL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | PHASE | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | R/W-0h | | | | | | | | | | | | | | | |

**Table 35-82. PL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15-0 | PHASE | R/W | 0h | Phase Load value<br>0h = Minimum value<br>FFFFFFFFh = Maximum Value |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.56 DBCTL (Offset = 18A4h) [Reset = 00000000h]

DBCTL is shown in Figure 35-96 and described in Table 35-83.

Return to the Summary Table.

The DBCTL register controls the dead band insertion of the pulse width modulated output.

**Figure 35-96. DBCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | FALLDELAY | | | |
| R-0h | | | | R/W-0h | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| FALLDELAY | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | M1_ENABLE | RISEDELAY | | | |
| R-0h | | | R/W-0h | R/W-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RISEDELAY | | | | | | | |
| R/W-0h | | | | | | | |

**Table 35-83. DBCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-28 | RESERVED | R | 0h | |
| 27-16 | FALLDELAY | R/W | 0h | Fall Delay<br>The number of TIMCLK periods inserted between the fall edge of CCP signal and the rise edge of CCP complimentary signal.<br>0h = Minimum value<br>FFFh = Maximum Value |
| 15-13 | RESERVED | R | 0h | |
| 12 | M1_ENABLE | R/W | 0h | Dead Band Mode 1 Enable.<br>0h = Disabled<br>1h = Enabled |
| 11-0 | RISEDELAY | R/W | 0h | Rise Delay<br>The number of TIMCLK periods inserted between the fall edge of CCP signal and the rise edge of CCP complimentary signal.<br>0h = Minimum value<br>FFFh = Maximum Value |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.57 TSEL (Offset = 18B0h) [Reset = 00000000h]

TSEL is shown in Figure 35-97 and described in Table 35-84.

Return to the Summary Table.

The TSEL register controls the input trigger enable and selection of the trigger source. Trigger sources are generated by other SoC elements through their respective publisher ports (subscribed in by the timer's subscriber port).

**Figure 35-97. TSEL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | TE | RESERVED | | | | ETSEL | | | | |
| R-0h | | | | | | R/W-0h | R-0h | | | | R/W-0h | | | | |

**Table 35-84. TSEL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-10 | RESERVED | R | 0h | |
| 9 | TE | R/W | 0h | Trigger Enable.<br>This selects whether a trigger is enabled or not for this counter<br>0x0 = Triggers are not used<br>0x1 = Triggers are used as selected by the ETSEL field<br>0h = Triggers are not used.<br>1h = Triggers are used as selected by the IE, ITSEL and ETSEL fields. |
| 8-5 | RESERVED | R | 0h | |
| 4-0 | ETSEL | R/W | 0h | External Trigger Select.<br>This selects which System Event is used if the input filter selects trigger.<br>Triggers 0-15 are used to connect triggers generated by other timer modules. Refer to the SoC datasheet for details related to timer trigger sources.<br>Triggers 16 and 17 are connected to event manager subscriber ports.<br>Event lines 18-31 are reserved for future use.<br>0h = TRIGx = External trigger input from TIM x.<br>1h = TRIGx = External trigger input from TIM x.<br>2h = TRIGx = External trigger input from TIM x.<br>3h = TRIGx = External trigger input from TIM x.<br>4h = TRIGx = External trigger input from TIM x.<br>5h = TRIGx = External trigger input from TIM x.<br>6h = TRIGx = External trigger input from TIM x.<br>7h = TRIGx = External trigger input from TIM x.<br>8h = TRIGx = External trigger input from TIM x.<br>9h = TRIGx = External trigger input from TIM x.<br>Ah = TRIGx = External trigger input from TIM x.<br>Bh = TRIGx = External trigger input from TIM x.<br>Ch = TRIGx = External trigger input from TIM x.<br>Dh = TRIGx = External trigger input from TIM x.<br>Eh = TRIGx = External trigger input from TIM x.<br>Fh = TRIGx = External trigger input from TIM x.<br>10h = TRIG_SUBx = External trigger input from subscriber port x.<br>11h = TRIG_SUBx = External trigger input from subscriber port x. |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.58 RC (Offset = 18B4h) [Reset = 00000000h]

RC is shown in Figure 35-98 and described in Table 35-85.

Return to the Summary Table.

Repeat counter is to reduce interrupt overhead. The repeat counter provides the mechanism to suppress un-necessary interrupts;
reducing the number of interrupts generated by each event type to 1 for the program number of periods. Specifically, the repeat timer may suppress Load, Compare (up/down, normal/shadow), and Zero events.

**Figure 35-98. RC**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | RC | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | |

**Table 35-85. RC Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | RC | R | 0h | Repeat Counter Value<br>0h = Minimum value<br>FFh = Maximum Value |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.59 RCLD (Offset = 18B8h) [Reset = 00000000h]

RCLD is shown in Figure 35-99 and described in Table 35-86.

Return to the Summary Table.

The load register value is transferred to the counter when the counter load input is asserted.

**Figure 35-99. RCLD**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | RCLD | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R/W-0h | | | | | | | |

**Table 35-86. RCLD Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | RCLD | R/W | 0h | Repeat Counter Load Value<br>This field provides the value loaded into the repeat counter at a load event following the repeat counter value equaling 0.<br>0h = Minimum value<br>FFh = Maximum Value |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.60 QDIR (Offset = 18BCh) [Reset = 00000000h]

QDIR is shown in Figure 35-100 and described in Table 35-87.

Return to the Summary Table.

The QDIR register provides the direction of count which is intended for use when operating the counter in QEI.

**Figure 35-100. QDIR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | | | | DIR |
| R-0h | | | | | | | | | | | | | | | R-0h |

**Table 35-87. QDIR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | DIR | R | 0h | Direction of count<br>0h = Down (Phase B leads Phase A)<br>1h = Up (Phase A leads Phase B) |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.61 FCTL (Offset = 18D0h) [Reset = 00000000h]

FCTL is shown in Figure 35-101 and described in Table 35-88.

Return to the Summary Table.

The FCTL register controls the fault inputs, fault detection and error handling behavior.

**Figure 35-101. FCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | FSENEXT2 | FSENEXT1 | FSENEXT0 | FSENAC2 | FSENAC1 | FSENAC0 |
| R-0h | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TFIM | RESERVED | | FL | | FI | RESERVED | FIEN |
| R/W-0h | R-0h | | R/W-0h | | R/W-0h | R-0h | R/W-0h |

**Table 35-88. FCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-14 | RESERVED | R | 0h | |
| 13 | FSENEXT2 | R/W | 0h | Specifies whether the external fault pin2 high/low is treated as fault condition.<br>0h = Fault Input is active low.<br>1h = Fault Input is active high. |
| 12 | FSENEXT1 | R/W | 0h | Specifies whether the external fault pin1 high/low is treated as fault condition.<br>0h = Fault Input is active low.<br>1h = Fault Input is active high. |
| 11 | FSENEXT0 | R/W | 0h | Specifies whether the external fault pin0 high/low is treated as fault condition.<br>0h = Fault Input is active low.<br>1h = Fault Input is active high. |
| 10 | FSENAC2 | R/W | 0h | Specifies whether the COMP2 output high/low is treated as fault condition.<br>0h = Fault Input is active low.<br>1h = Fault Input is active high. |
| 9 | FSENAC1 | R/W | 0h | Specifies whether the COMP1 output high/low is treated as fault condition.<br>0h = Fault Input is active low.<br>1h = Fault Input is active high. |
| 8 | FSENAC0 | R/W | 0h | Specifies whether the COMP0 output high/low is treated as fault condition.<br>0h = Fault Input is active low.<br>1h = Fault Input is active high. |
| 7 | TFIM | R/W | 0h | Trigger Fault Input Mask<br>Specifies whether the selected trigger participates as a fault input. If enabled and the trigger asserts, the trigger is treated as a fault.<br>0h = Selected trigger does not participate in fault condition generation<br>1h = Selected trigger participates in fault condition generation |
| 6-5 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

**Table 35-88. FCTL Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 4-3 | FL | R/W | 0h | Fault Latch mode<br>Specifies whether the fault condition is latched and configures the latch clear conditions.<br>0h = Overall fault condition is not dependent on the F bit in RIS<br>1h = Overall fault condition is dependent on the F bit in RIS<br>2h = Fault condition is latched. Fault condition is cleared on a zero event if the fault input is 0.<br>3h = Fault condition is latched. Fault condition is cleared on a load event if the fault input is 0. |
| 2 | FI | R/W | 0h | Fault Input<br>Specifies whether the overall fault condition is dependent on the sensed fault pin.<br>0h = Overall Fault condition is not dependent on sensed input.<br>1h = Overall Fault condition is dependent on sensed input. |
| 1 | RESERVED | R | 0h | |
| 0 | FIEN | R/W | 0h | Fault Input Enable<br>This bit enables the input for fault detection.<br>0h = Fault Input Disabled<br>1h = Fault Input Enabled |



Copyright © 2026 Texas Instruments Incorporated

### 35.3.62 FIFCTL (Offset = 18D4h) [Reset = 00000000h]

FIFCTL is shown in Figure 35-102 and described in Table 35-89.

Return to the Summary Table.

The FIFCTL register controls the filtering for the fault input.

**Figure 35-102. FIFCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | FILTEN | CPV | RESERVED | FP | |
| R-0h | | | R/W-0h | R/W-0h | R-0h | R/W-0h | |

**Table 35-89. FIFCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-5 | RESERVED | R | 0h | |
| 4 | FILTEN | R/W | 0h | Filter Enable<br>This bit controls whether the input is filtered by the input filter or bypasses to go directly to the optional pre-scale filter and then to the edge detect.<br>0h = Bypass<br>1h = Filtered. |
| 3 | CPV | R/W | 0h | Consecutive Period/Voting Select<br>This bit controls whether the input filter uses a stricter consecutive period count or majority voting.<br>0h = Consecutive Periods.<br>The input must be at a specific logic level for the period defined by FP before it is passed to the filter output.<br>1h = Voting.<br>The filter ignores one clock of opposite logic over the filter period. I.e. Over FP samples of the input, up to 1 sample may be of an opposite logic value (glitch) without affecting the output |
| 2 | RESERVED | R | 0h | |
| 1-0 | FP | R/W | 0h | Filter Period<br>This field specifies the sample period for the input filter. I.e. The input is sampled for FP timer clocks during filtering.<br>0h = Filter Period 3<br>1h = Filter Period 5<br>2h = Filter Period 8 |



Copyright © 2026 Texas Instruments Incorporated

Chapter 36

# TIMB

The Basic Timer (TIMB) delivers fundamental timing functionality through configurable counters that support periodic interrupts for various timing applications.


**36.1 TIMB Overview** ..... 2252
**36.2 TIMB Operation** ..... 2253
**36.3 TIMB Example Applications** ..... 2258
**36.4 TIMB Registers** ..... 2264





Copyright © 2026 Texas Instruments Incorporated

## 36.1 TIMB Overview

The Basic Timer (TIMB) is designed to provide fundamental timing functionality for system operations and event management. TIMB offers configurable periodic interrupt generation through an array of simple counters. The module can be used in various timing applications, from basic periodic interrupts to more complex event timing measurements. Each counter in the array operates independently or can be chained together to create longer time bases. The event management system allows the timer to measure timing properties of external events, including duration measurement and event counting.

### *36.1.1 Features*

TIMB features include:

- Configurable array of 2-8 counters
  - 16-bit resolution
  - Bus clock source with prescaler
  - Up counting mode
  - Chaining 16-bit counters for longer time base
  - One-shot or continuous operation
  - Event-based control (start/stop/reset)
- Event generation capabilities
  - Counter overflow events
  - Start/stop events
  - Capture events
- Event Measurement features
  - Event counting
  - Duration measurement
  - Event sequence detection
- Interrupt / event generation support
- Hardware register locking during operation

### *36.1.2 TIMB Block Diagram*

Figure 36-1 depicts the top level functionality of a TIMB instance. TIMB consists of an array of counters (CNTR0 to CNTRN). Each counter generates three events, counter overflow (OVF), counter start (START), and counter stop (STOP). The events from each counter can be enabled selectively to generate the timer interrupt (BTIMINT). A generic publisher event is also available for triggering other IPs based on timer events.

Each counter can START, STOP, or RESET from external events such as an OVF, ADC, timer, GPIO or another peripheral. All counters are clocked from the Bus Clock.

**Note**

Refer to the device-specific data sheet to determine the number of counters and the external event which are supported.



Copyright © 2026 Texas Instruments Incorporated

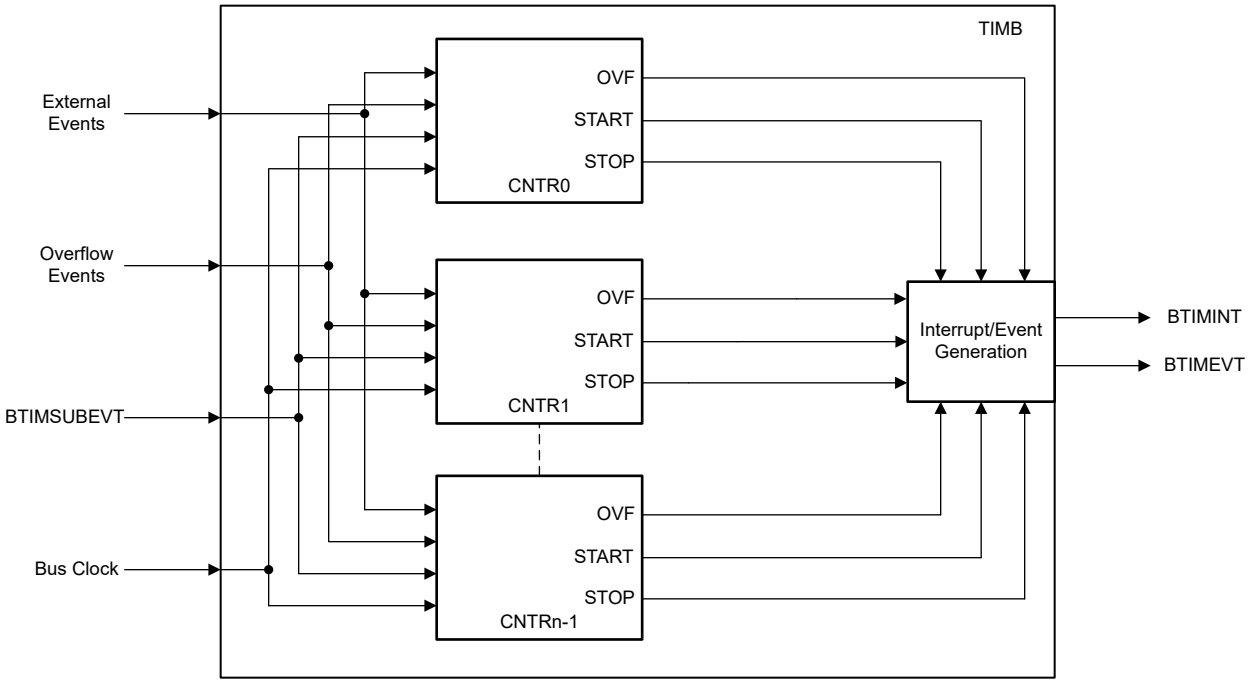



**Figure 36-1. TIMB Block Diagram**

## 36.2 TIMB Operation

### *36.2.1 Counter Block Operation*

The functions of the counter block are outlined in the Figure 36-2. The counter operation can be controlled from hardware events (clock, reset, and enable) or through software configuration. Once enabled (TIMB.CTL0[j].EN = 1), the counter (TIMB.CNT[j]) increments by one on each positive edge of the counter clock. The counter continues incrementing until reaching the value stored in the respective load register (TIMB.LD[j]) generating an overflow (OVF). On the next positive clock edge the counter is reset to zero. The counter enable and reset behavior is summarized in Table 36-1.



Copyright © 2026 Texas Instruments Incorporated

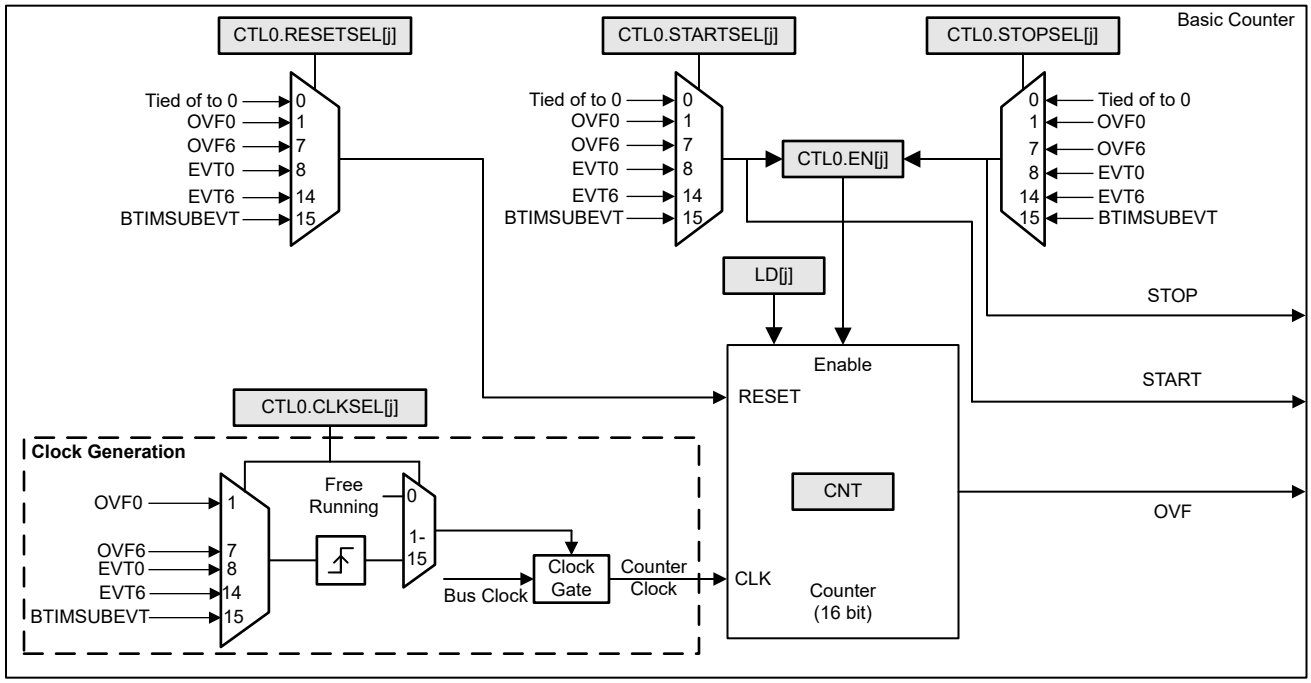



**Figure 36-2. TIMB Counter Block**

**Table 36-1. Counter Operation**

| CTL0.EN[j] | Counter reset | Effect on counter |
|---|---|---|
| 0 | x | Counter remains in the current state |
| 1 | 0 | If the current value of the counter equals the LD value then, the counter overflows to 0, else counter = counter + 1 |
| 1 | 1 | Counter is cleared to 0 |

#### 36.2.1.1 Clock Source Selection

The Basic Timer provides flexible clock source selection for each counter through the TIMB.CTL0[j].CLKSEL field. Each counter is driven by the BUSCLK which can be gated by external event edges or overflow events from other counters. By default the BUSCLK is the clock source to the counter.

**Note**

Counter n can only use the OVF from counters with indices less than (n-1) as a clock source selection.

The clock path includes:

1. Event source multiplexer controlled by TIMB.CTL0[j].CLKSEL
2. Edge detection logic for event-based clocking
3. BUSCLK clock gating

The Counter Clock Source Generation figure indicates the relationship between BUSCLK, TIMB.CTL0[j].CLKSEL, and the counter clock. In this example, EVT1 is selected to gate the BUSCLK using the TIMB.CTL0[j].CLKSEL field. The counter clock is pulsed on the next positive edge of EVT1.



Copyright © 2026 Texas Instruments Incorporated

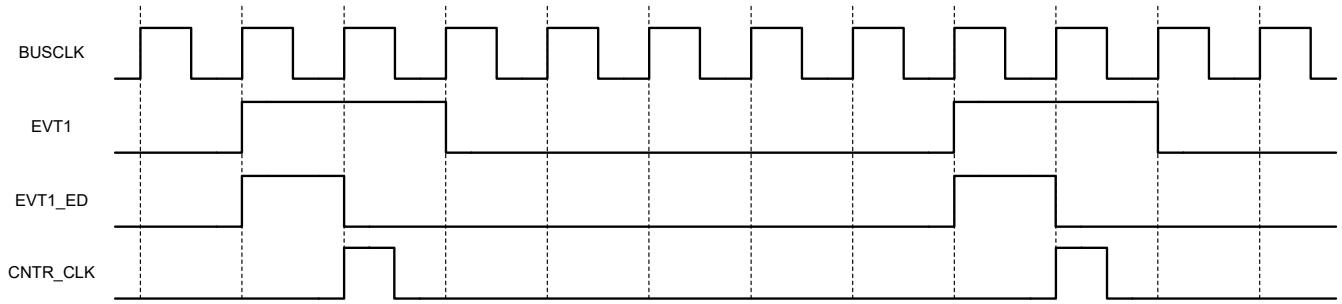



**Figure 36-3. Counter Clock Source Generation**

#### 36.2.1.2 Counter Reset Generation

The counter can be reset to zero based on the selected reset sources configured by TIMB.CTL0[j].RESETSEL. When the reset source is pulsed high the counter is cleared to zero.

**Note**

Counter n can only use the OVF from counters with indices less than (n-1) as a reset source selection.

Figure 36-4 shows the counters reset generation functionality. EVT2 is selected as the reset source for the counter. On the next positive edge of EVT2 the counter is cleared to zero.

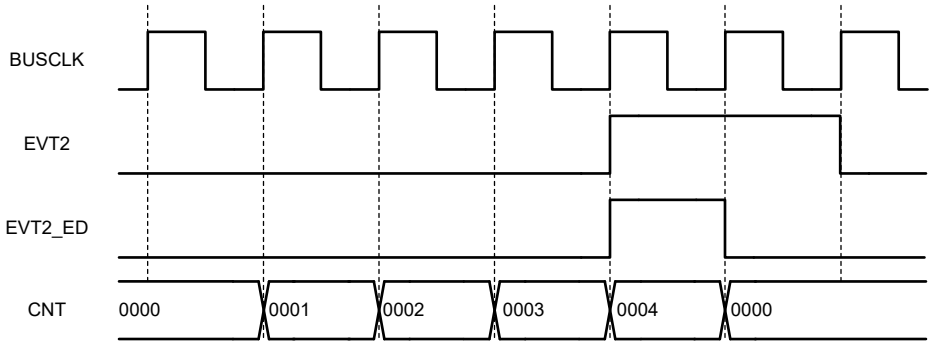



**Figure 36-4. Counter Reset Example**

#### 36.2.1.3 Event Based Enable and Disable

The TIMB.CTL0[j].EN field enables the counter based on software or hardware events. The start event can be configured to be one of the events using TIMB.CTL0[j].STARTSEL field. In this case, TIMB.CTL0[j].EN is set based on the selected start event, provided the TIMB.MIS.CNTxSTOP bit is 0. Similarly, the source of stop event can be configured by using the TIMB.CTL0[j].STOPSEL field. If a stop event occurs, TIMB.CTL0[j].EN is cleared.

**Note**

In cases where the same event is chosen as both the start and stop sourcesm, clearing TIMB.CTL0[j].EN has a higher priority than setting TIMB.CTL0[j].EN.

Effect of START and STOP events on CTL0[j].EN demonstrates the effect of start and stop events on TIMB.CTL0[j].EN. TIMB.CTL[j].EN is set on the next positive clock edge after the EVT1_ED and is cleared on the next positive edge after EVT2_ED.

- TIMB.CTL0[j].EN is not enabled in software
- TIMB.CTL0[j].STARTSEL is configured to EVT1
- TIMB.CTL0[j].STOPSEL is configured to EVT2



Copyright © 2026 Texas Instruments Incorporated

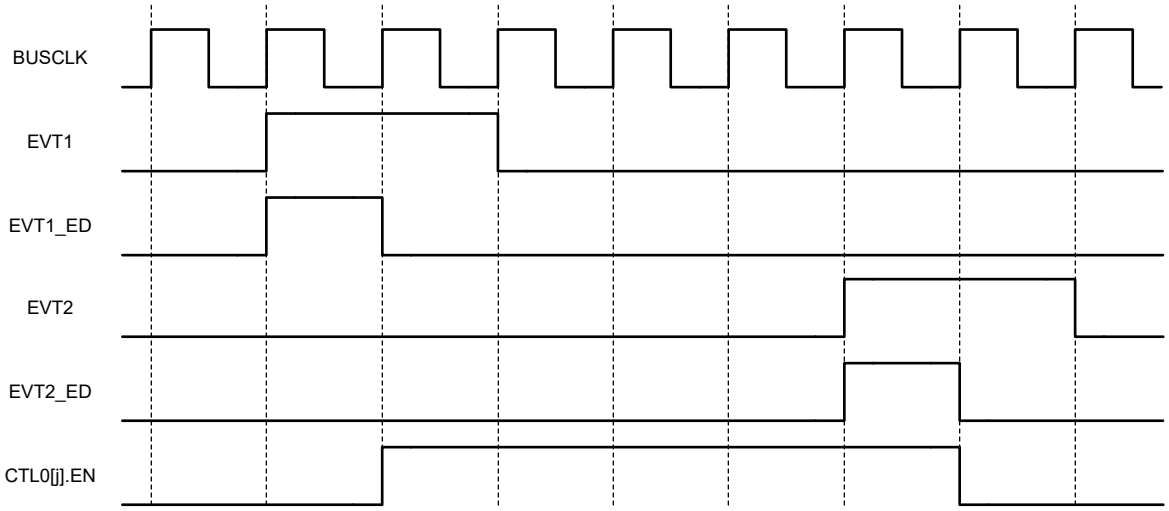



**Figure 36-5. Effect of START and STOP events on CTL0[j].EN**

#### 36.2.1.4 Event Generation

Each counter generates 3 events, OVF, START and STOP. Figure 36-6 depicts example waveforms to illustrate the generation of the OVF event. OVF is pulsed for 1 bus clock cycle when the counter value equals the TIMB.LD[j] value.

START and STOP event generation is shown in START & STOP Event Generation. In this scenario the counter generates a START event on EVT1 and a STOP event on EVT2.

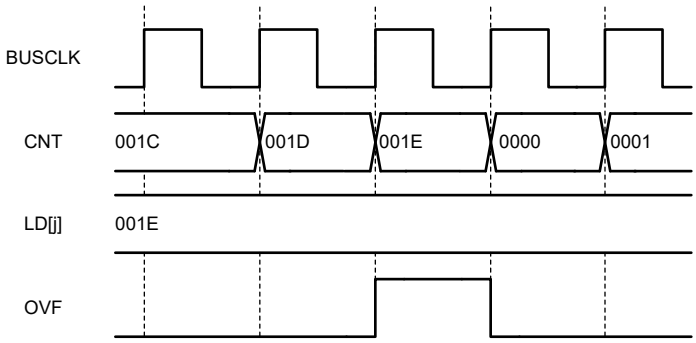



**Figure 36-6. OVF Generation**



Copyright © 2026 Texas Instruments Incorporated

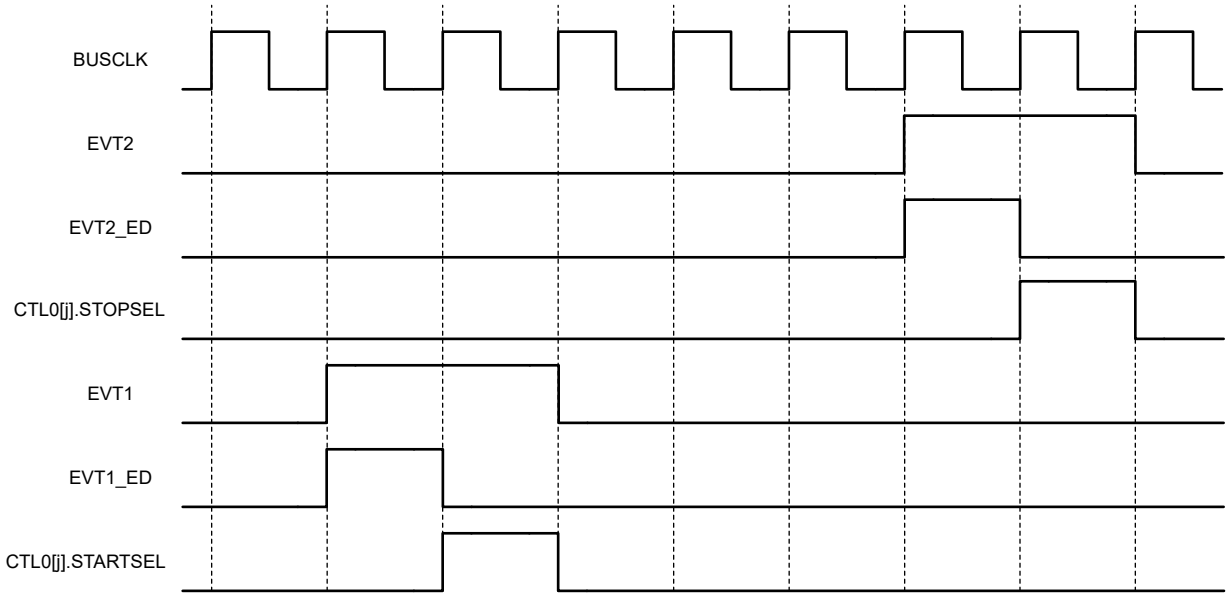



**Figure 36-7. START & STOP Event Generation**

#### 36.2.1.5 Interrupt Generation

One interrupt is generated for all of the counters with the TIMB array. The sources to the interrupt are the OVF, START, & STOP events from each timer. Refer to Table 36-2 for a summary of the interrupt sources.

The TIMB.IIDX register is a priority enabled interrupt index. The interrupts are set from the Raw Interrupt Status (TIMB.RIS) and Interrupt Mask (TIMB.IMASK) registers. Setting an IMASK bit un-masks the corresponding interrupt source and exposes the interrupt in the TIMB.IIDX register. On each read of TIMB.IIDX the highest priority interrupt is indicated. After it's read, the pending interrupt is cleared.

**Table 36-2. TIMB Interrupt Sources**

| Counter | Name | Description |
|---|---|---|
| CNTRx | CNTxOVF | CNTRx.OVF event occurred. |
| | CNTxSTRT | CNTRx.START event occurred. |
| | CNTxSTOP | CNTRx.STOP event occurred. |

#### 36.2.1.6 Counter Behavior on a Debug Halt

The behavior of the counters on a CPU halting execution due to a debug halt event, can by the TIMB.PDBGCTL.FREE and TIMB.PDBGCTL.SOFT bits. Table 36-3 describes the effect of debug halt on the counter behavior.

When the CPU is halted during debug operations, the TIMB counter register (CNT) is write-protected and cannot be modified by the debugger. This behavior occurs when the FREE bit is cleared to 0, which is the expected operation to maintain timer integrity during debug sessions. All other BTIMER memory-mapped registers (MMRs) retain normal read/write access during debug operations.

**Table 36-3. Effects of Debug Halt on Counter**

| TIMB.PDBGCTL.FREE | TIMB.PDBGCTL.SOFT | Effect of debug halt on the counter. |
|---|---|---|
| 1 | x | No effect |
| 0 | 0 | Counter stops immediately. When the debug halt condition is removed, the counter starts from the current value. |



Copyright © 2026 Texas Instruments Incorporated

**Table 36-3. Effects of Debug Halt on Counter (continued)**

| TIMB.PDBGCTL.FREE | TIMB.PDBGCTL.SOFT | Effect of debug halt on the counter. |
|---|---|---|
| 0 | 1 | Counter stops on reaching 0. When the debug halt condition is removed, the counter starts from the current value. |

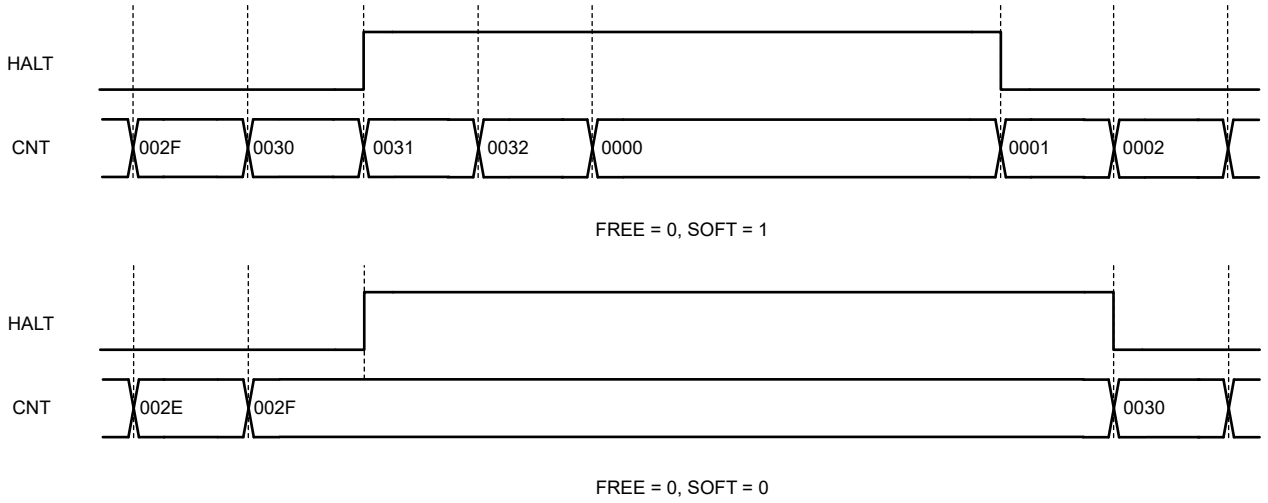



**Figure 36-8. Effects of Debug Halt on Counter Timing Diagram**

#### 36.2.1.7 Hardware Locking of Configuration Registers

When TIMB.CTL0[j].EN = 1, writes to the following register fields are ignored in hardware. This prevents unintential writes to the configuration registers when the counter is operational.

- TIMB.CTL0[j].STARTSEL
- TIMB.CTL0[j].STOPSEL
- TIMB.CTL0[j].RESETSEL
- TIMB.CTL0[j].CLOCKSEL
- TIMB.LD[j].VAL
- TIMB.CNT[j].VALUE

## 36.3 TIMB Example Applications

### *36.3.1 Periodic Interrupt Generation*

This is a basic use case, wherein a periodic interrupt is generated with one counter. The register configuration for the example waveform shown is tabulated in Table 36-4. In this example, an interrupt is generated when CNTR0 equals a value of 3.

**Table 36-4. Periodic Interrupt Configuration**

| Register | Value | Description |
|---|---|---|
| TIMB.CTL0[0].STARTSEL | 0 | N/A |
| TIMB.CTL0[0].STOPSEL | 0 | N/A |
| TIMB.CTL0[0].RESETSEL | 0 | N/A |
| TIMB.CTL0[0].CLKSEL | 0 | Bus clock clocks the counter |
| TIMB[0].LD | 3 | LD value |
| TIMB.IMASK | 0x001 | OVF0 interrupt is enabled |
| TIMB.CTL0[0].EN | 1 | Enabled by software |



Copyright © 2026 Texas Instruments Incorporated

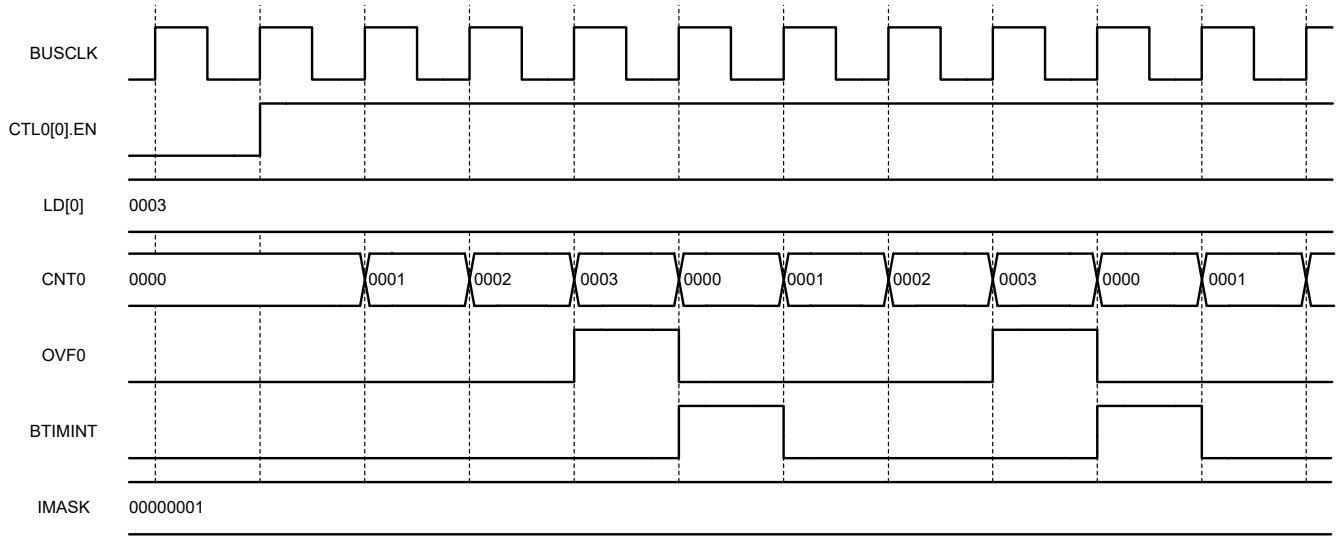



**Figure 36-9. Periodic Interrupt Generation Example**

### *36.3.2 Counter Chaining*

In the Figure 36-10, two counters are cascaded to create a larger time duration. CNTR0 OVF0 is used as the clock source of CNTR1. The interrupt is generated when the CNTR1 OVF1 occurs. This results in interrupt being generated every 12 bus clock cycles.

**Table 36-5. Counter Chaining Register Configuration**

| Register | Value | Description |
|---|---|---|
| **CNTR0** | | |
| TIMB.CTL0[0].STARTSEL | 0 | N/A |
| TIMB.CTL0[0].STOPSEL | 0 | N/A |
| TIMB.CTL0[0].RESETSEL | 0 | N/A |
| TIMB.CTL0[0].CLKSEL | 0 | N/A |
| TIMB.LD[0] | 2 | LD value |
| TIMB.CTL0[0].EN | 1 | Enabled by software |
| **CNTR1** | | |
| TIMB.CTL0[1].STARTSEL | 0 | N/A |
| TIMB.CTL0[1].STOPSEL | 0 | N/A |
| TIMB.CTL0[1].RESETSEL | 0 | N/A |
| TIMB.CTL0[1].CLKSEL | 1 | OVF0 is the CLK source |
| TIMB.LD[1] | 3 | LD Value |
| TIMB.CTL0[1].EN | 1 | Enabled by software |
| IMASK | 0x008 | OVF interrupr of CNTR1 enabled |



Copyright © 2026 Texas Instruments Incorporated

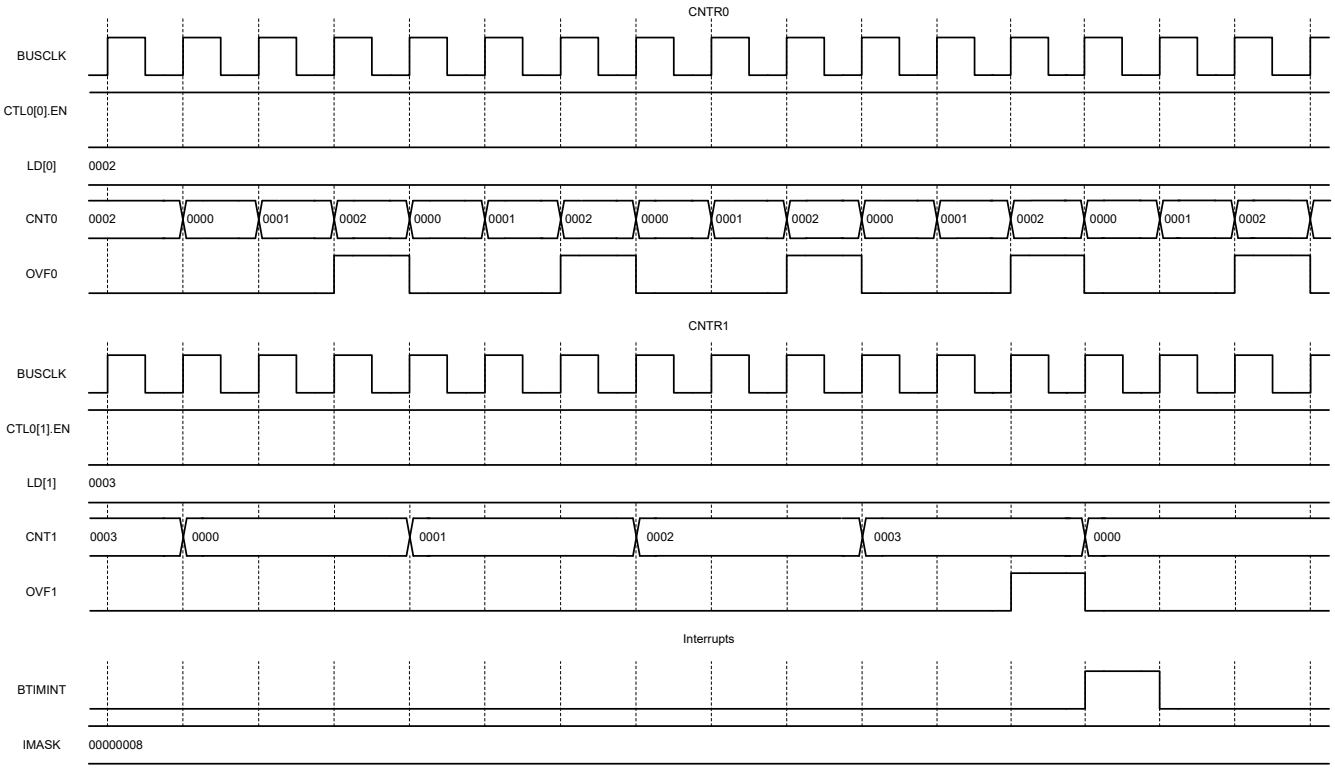



**Figure 36-10. Counter Chaining Example**

### *36.3.3 Event Counting*

The Figure 36-11 implements a counter to generate an interrupt after counting a certain number of events. In this specific example, as the LD value is 2, an interrupt is generated after 3 events on EVT2.

**Table 36-6. Event Counting Register Configuration**

| Register | Value | Description |
|---|---|---|
| TIMB.CTL0[j].STARTSEL | 9 | Enabled by EVT2 |
| TIMB.CTL0.STOPSEL | 0 | N/A |
| TIMB.CTL0[j].RESETSEL | 0 | N/A |
| TIMB.CTL0[j].CLKSEL | 9 | EVT2 is selected |
| TIMB.LD[j] | 2 | LD Value |
| TIMB.IMASK | 0x001 | OVF interrupt enabled |
| TIMB.CTL0[j].EN | 0 | Enabled by hardware event |



Copyright © 2026 Texas Instruments Incorporated

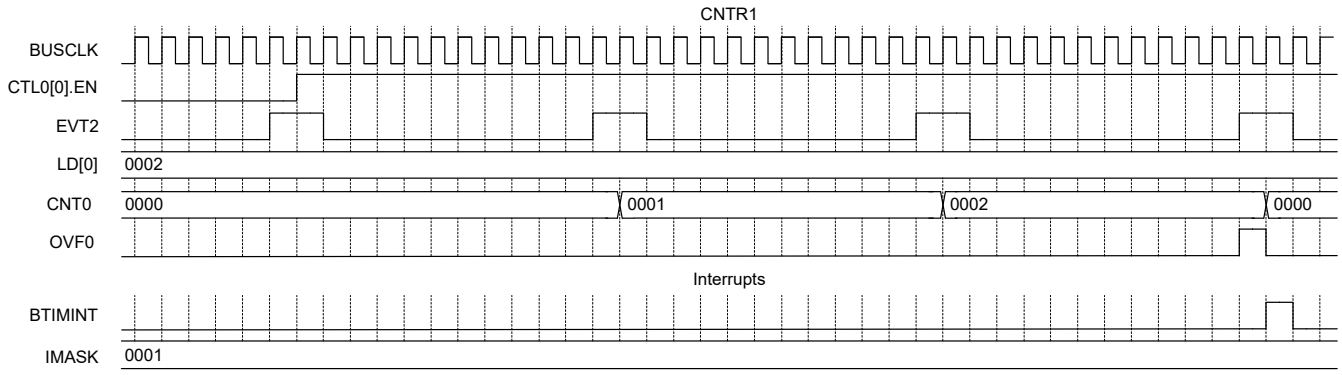



**Figure 36-11. Event Counting Example**

### *36.3.4 Event Duration Measurement*

In the Figure 36-12, the average period of EVT2 is being measured. The following configuration is used to achieve this function.

- EVT2 is used as the start source for CNTR0 and CNTR1
- EVT2 is also used as the clock source of CNTR1
- CNTR0 operates off the BUSCLK
- CNTR1's OVF is the stop source for both counters
- The LD value of CNTR0 needs to be large enough so that CNTR0 does not overflow before the STOP event
- The LD value of CNTR1 is set to 2

Both the counters start on the first EVT2 event. After 2 more EVT2 events, CNTR1 will assert OVF1. This will stop both the counters. Using the value of CNTR0, we can compute the average duration of EVT2. Note that if a new EVT2 event occurs before the application reads the CNTR0 value, both counters will restart, causing the CNTR0 value to be overwritten and resulting in an incorrect average period measurement.

EVT2-duration = (CNTR0 value when it stopped)/(CNTR1 LD value + 1) = 36 (0x24)/3 = 12 cycles.

**Table 36-7. Event Duration Example Configuration**

| Register | Value | Description |
|---|---|---|
| **CNTR0** | | |
| TIMB.CTL0[0].EN | 0 | Enabled by hardware event |
| TIMB.CTL0[0].STARTSEL | 9 | EVT2 is used as the start source |
| TIMB.CTL0[0].STOPSEL | 2 | CNTR1 OVF is selected |
| TIMB.CTL0[0].RESETSEL | 0 | N/A |
| TIMB.CTL0[0].CLKSEL | 0 | Bus clock is the clock source |
| TIMB.LD[0] | Large value | CNTR0 should not overflow before the stop event |
| TIMB.CTL0[0].EN | 0 | Enabled by hardware event |
| **CNTR1** | | |
| TIMB.CTL0[1].STARTSEL | 9 | EVT2 is used as the start source |
| TIMB.CTL0[1].STOPSEL | 2 | CNTR1 OVF is selected |
| TIMB.CTL0[1].RESETSEL | 0 | N/A |
| TIMB.CTL0[1].CLKSEL | 9 | EVT2 is used as the clock source |
| TIMB[1].LD | 2 | LD value |
| TIMB[1].IMASK | 0x008 | OVF1 is enabled |
| TIMB.CTL0[1].EN | 0 | Enabled by hardware event |



Copyright © 2026 Texas Instruments Incorporated

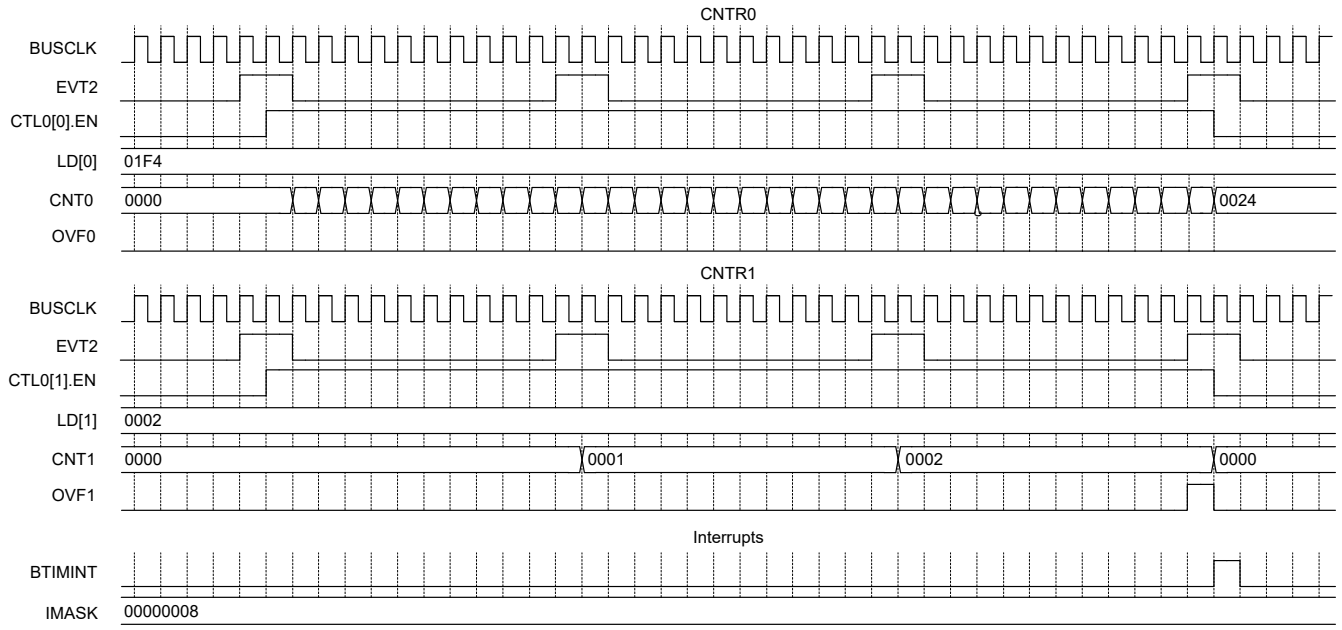



**Figure 36-12. Event Duration Measurement Example**

### *36.3.5 Event Sequence Checking*

The Figure 36-13 implements a single counter to detect the occurrence of EVT3 after EVT2. If EVT3 occurs within a specified window after EVT2 then the counter is stopped and reset. The window interval is determined by the LD[j] value. If EVT3 does not occur in this window, the counter overflows and generates an interrupt.

**Table 36-8. Event Squence Configuration**

| Register | Value | Description |
|---|---|---|
| TIMB.CTL0.STARTSEL[1] | 9 | EVT2 is used as the start source |
| TIMB.CTL0.STOPSEL[1] | 10 | EVT3 is used as the stop source |
| TIMB.CTL0.RESETSEL[1] | 10 | EVT3 is used as the reset source |
| TIMB.CTL0.CLKSEL[1] | 0 | BUSCLK is used as the counter clock |
| TIMB.LD[1] | 0x06 | LD defines the window |
| TIMB.IMASK | 0x08 | OVF1 is enabled |
| TIMB.CTL0.EN[1] | 0 | Enabled by hardware |

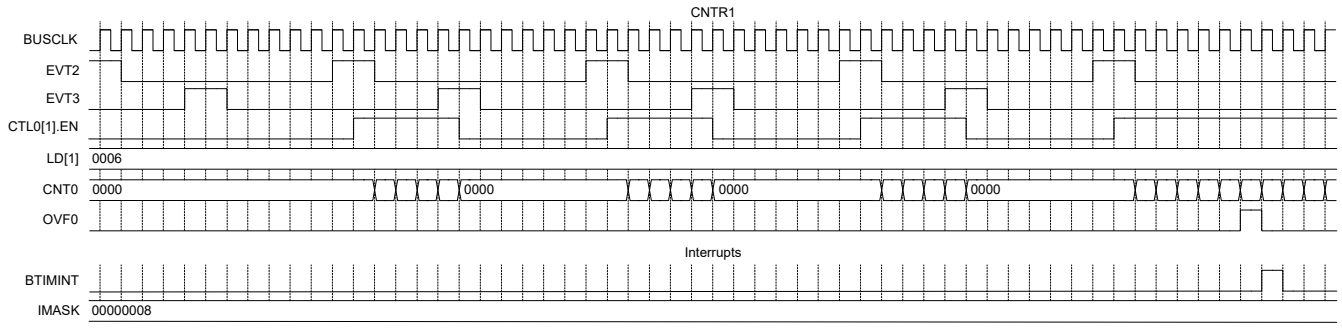



**Figure 36-13. Event Sequence Checker Example**

### *36.3.6 PWM Generation*

This example generates a pulse width modulated (PWM) signal using GPIO and two TIMB counters. One counter defines the period of the PWM signal and the other counter defines the pulse width of the PWM signal.



Copyright © 2026 Texas Instruments Incorporated

The register configuration of the counters, IMASK configuration of timer publisher are described in the table below.

- CNTR0 is setup to be a free running counter.
- CNTR1 is started on CNTR0 overflow.
- CNTR1 is stopped and reset on OVF1. This ensures that CNTR1 runs once on every CNTR0 overflow
- A publisher event is generated on OVF0 and OVF1
- GPIO pin is programmed to toggle on the publisher event of the timer

Table 36-9 illustrates an example PWM generation method. The relative position of OVF1 with respect to OVF0 varies as the LD value of CNTR1 is varied. This modulates the pulse width of the GPIO pin as seen in the Figure 36-14.

**Table 36-9. PWM Register Configuration**

| Register | Value | Description |
|---|---|---|
| **CNTR0** | | |
| TIMB.CNT[0] | LD -1 | EVT2 is used as the start source. |
| TIMB.CTL0[0].CLKSEL | 0 | Bus clock is used to clock the counter |
| TIMB.LD[0] | 8 | LD value defines PWM signal period. Set to 8 in this example |
| TIMB.CTL[0].EN | 1 | Enable counter 0 after configuring all the registers. |
| **CNTR1** | | |
| TIMB.CTL0[1].STARTSEL | 1 | Use CNTR0's OVF as the start condition. |
| TIMB.CTL0[1].STOPSEL | 2 | Use CNTR1's OVF as the stop condition. |
| TIMB.CTL0[1].RESETSEL | 2 | Use CNTR1's OVF as the reset condition. |
| TIMB.LD[1] | Vary between 2 and CNTR0.LD -2 | LD value defines the high pulse width of the PWM. |
| TIMB.CNT0[1].CTL0.EN | 1 | Enable counter 1 after configuring all the registers. |
| **Publisher configuration** | | |
| TIMB.PUB0.IMASK | 0x9 | Generate a publisher event on CNTR0.OVF and CNTR1.OVF |

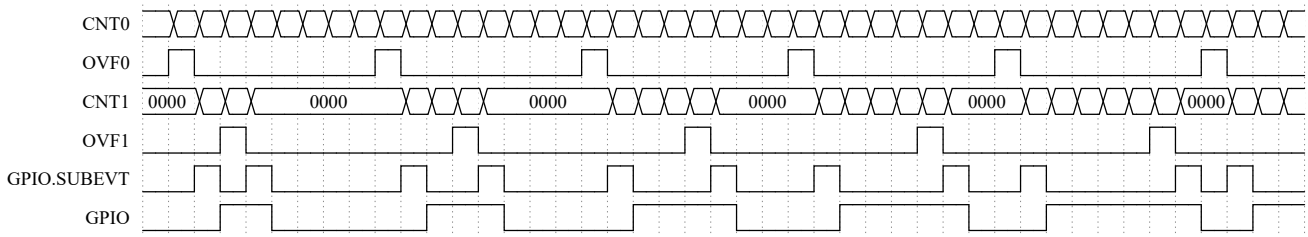



**Figure 36-14. PWM Example**



Copyright © 2026 Texas Instruments Incorporated

## 36.4 TIMB Registers

Table 36-10 lists the memory-mapped registers for the TIMB registers. All register offset addresses not listed in Table 36-10 should be considered as reserved locations and the register contents should not be modified.

**Table 36-10. TIMB Registers**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 400h | FSUB_0 | Subscriber Configuration Register. | Section 36.4.1 |
| 444h | FPUB_0 | Publisher Port 0 | Section 36.4.2 |
| 800h | PWREN | Power enable | Section 36.4.3 |
| 804h | RSTCTL | Reset Control | Section 36.4.4 |
| 814h | STAT | Status Register | Section 36.4.5 |
| 1018h | PDBGCTL | Peripheral Debug Control | Section 36.4.6 |
| 1020h | IIDX | Interrupt index | Section 36.4.7 |
| 1028h | IMASK | Interrupt Mask.<br>If a bit is set, then corresponding interrupt is un-masked.<br>Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS. | Section 36.4.8 |
| 1030h | RIS | Raw interrupt status.<br>Reflects all pending interrupts, regardless of masking.<br>The RIS register allows the user to implement a poll scheme.<br>A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled. | Section 36.4.9 |
| 1038h | MIS | Masked interrupt status.<br>This is an AND of the IMASK and RIS registers. | Section 36.4.10 |
| 1040h | ISET | Interrupt set.<br>Allows interrupts to be set by software (useful in diagnostics and safety checks).<br>Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set.<br>If the interrupt is enabled through the mask, then the corresponding MIS bit is also set. | Section 36.4.11 |
| 1048h | ICLR | Interrupt clear.<br>Write a 1 to clear corresponding Interrupt. | Section 36.4.12 |
| 1050h | IIDX | Interrupt index | Section 36.4.13 |
| 1058h | IMASK | Interrupt Mask.<br>If a bit is set, then corresponding interrupt is un-masked.<br>Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS. | Section 36.4.14 |
| 1060h | RIS | Raw interrupt status.<br>Reflects all pending interrupts, regardless of masking.<br>The RIS register allows the user to implement a poll scheme.<br>A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled. | Section 36.4.15 |
| 1068h | MIS | Masked interrupt status.<br>This is an AND of the IMASK and RIS registers. | Section 36.4.16 |
| 1070h | ISET | Interrupt set.<br>Allows interrupts to be set by software (useful in diagnostics and safety checks).<br>Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set.<br>If the interrupt is enabled through the mask, then the corresponding MIS bit is also set. | Section 36.4.17 |
| 1078h | ICLR | Interrupt clear.<br>Write a 1 to clear corresponding Interrupt. | Section 36.4.18 |



Copyright © 2026 Texas Instruments Incorporated

**Table 36-10. TIMB Registers (continued)**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 10E0h | EVT_MODE | Event Mode | Section 36.4.19 |
| 1100h + formula | CTL0_j | CTL0 register | Section 36.4.20 |
| 1104h + formula | LD_j | Counter period register. | Section 36.4.21 |
| 1108h + formula | CNT_j | Counter register. | Section 36.4.22 |

Complex bit access types are encoded to fit into small table cells. Table 36-11 shows the codes that are used for access types in this section.

**Table 36-11. TIMB Access Type Codes**

| Access Type | Code | Description |
|---|---|---|
| Read Type | | |
| R | R | Read |
| RH | R<br>H | Read<br>Set or cleared by hardware |
| Write Type | | |
| W | W | Write |
| WK | W<br>K | Write<br>Write protected by a key |
| Reset or Default Value | | |
| -*n* | | Value after reset or the default value |
| Register Array Variables | | |
| i,j,k,l,m,n | | When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula. |
| y | | When this variable is used in a register name, an offset, or an address it refers to the value of a register array. |



Copyright © 2026 Texas Instruments Incorporated

### 36.4.1 FSUB_0 Register (Offset = 400h) [Reset = 00000000h]

FSUB_0 is shown in Table 36-12.

Return to the Summary Table.

Subscriber port

**Table 36-12. FSUB_0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R/W | 0h | |
| 3-0 | CHANID | R/W | 0h | 0 = disconnected.<br>1-15 = connected to channelID = CHANID.<br>0h = A value of 0 specifies that the event is not connected<br>Fh = Consult your device datasheet as the actual allowed maximum may be less than 255. |



Copyright © 2026 Texas Instruments Incorporated

### 36.4.2 FPUB_0 Register (Offset = 444h) [Reset = 00000000h]

FPUB_0 is shown in Table 36-13.

Return to the Summary Table.

Publisher port

**Table 36-13. FPUB_0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R/W | 0h | |
| 3-0 | CHANID | R/W | 0h | 0 = disconnected.<br>1-15 = connected to channelID = CHANID.<br>0h = A value of 0 specifies that the event is not connected<br>Fh = Consult your device datasheet as the actual allowed maximum may be less than 15. |



Copyright © 2026 Texas Instruments Incorporated

### 36.4.3 PWREN Register (Offset = 800h) [Reset = 00000000h]

PWREN is shown in Table 36-14.

Return to the Summary Table.

Register to control the power state

**Table 36-14. PWREN Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | KEY to allow Power State Change<br>26h = KEY to allow write access to this register |
| 23-1 | RESERVED | R/W | 0h | |
| 0 | ENABLE | R/WK | 0h | Enable the power<br>KEY must be set to 26h to write to this bit.<br>0h = Disable Power<br>1h = Enable Power |



Copyright © 2026 Texas Instruments Incorporated

### 36.4.4 RSTCTL Register (Offset = 804h) [Reset = 00000000h]

RSTCTL is shown in Table 36-15.

Return to the Summary Table.

Register to control reset assertion and de-assertion

**Table 36-15. RSTCTL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | Unlock key<br>B1h = KEY to allow write access to this register |
| 23-2 | RESERVED | W | 0h | |
| 1 | RESETSTKYCLR | WK | 0h | Clear the RESETSTKY bit in the STAT register<br>KEY must be set to B1h to write to this bit.<br>0h = Writing 0 has no effect<br>1h = Clear reset sticky bit |
| 0 | RESETASSERT | WK | 0h | Assert reset to the peripheral<br>KEY must be set to B1h to write to this bit.<br>0h = Writing 0 has no effect<br>1h = Assert reset |



Copyright © 2026 Texas Instruments Incorporated

### 36.4.5 STAT Register (Offset = 814h) [Reset = 00000000h]

STAT is shown in Table 36-16.

Return to the Summary Table.

peripheral enable and reset status

**Table 36-16. STAT Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-17 | RESERVED | R | 0h | |
| 16 | RESETSTKY | R | 0h | This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register<br>0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register<br>1h = The peripheral was reset since the last bit clear |
| 15-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 36.4.6 PDBGCTL Register (Offset = 1018h) [Reset = 00000000h]

PDBGCTL is shown in Table 36-17.

Return to the Summary Table.

This register can be used by the software developer to control the behavior of the peripheral relative to the 'Core Halted' input

**Table 36-17. PDBGCTL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | R/W | 0h | |
| 1 | SOFT | R/W | 0h | Soft halt boundary control. This function is only available, if FREE is set to 'STOP'<br>0h = The peripheral will halt immediately, even if the resultant state will result in corruption if the system is restarted<br>1h = The peripheral blocks the debug freeze until it has reached a boundary where it can resume without corruption |
| 0 | FREE | R/W | 0h | Free run control<br>0h = The peripheral freezes functionality while the Core Halted input is asserted and resumes when it is deasserted.<br>1h = The peripheral ignores the state of the Core Halted input |



Copyright © 2026 Texas Instruments Incorporated

### 36.4.7 IIDX Register (Offset = 1020h) [Reset = 00000000h]

IIDX is shown in Table 36-18.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. 0x0 means no event pending. Interrupt 1 is the highest priority, 2 next highest priority etc. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred.
On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in the RIS and MIS are cleared as well. After a read from the CPU (not from the debug interface), the register must be updated with the next highest priority interrupt, if none are pending, then it should display 0x0.

**Table 36-18. IIDX Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-10 | RESERVED | R | 0h | |
| 9-0 | STAT | R | 0h | Interrupt index status<br>00h = No bit is set means there is no pending interrupt request<br>01h = Counter0 period match interrupt<br>02h = Counter 0 start<br>03h = Counter0 stop<br>04h = Counter1 period match interrupt<br>05h = Counter 1 start<br>06h = Counter1 stop<br>07h = Counter2 period match interrupt<br>08h = Counter 2 start<br>09h = Counter2 stop<br>0Ah = Counter3 period match interrupt<br>0Bh = Counter 3 start<br>0Ch = Counter3 stop<br>Dh = Counter4 period match interrupt<br>Eh = Counter 4 start<br>Fh = Counter4 stop<br>10h = Counter5 period match interrupt<br>11h = Counter 5 start<br>12h = Counter5 stop<br>13h = Counter6 period match interrupt<br>14h = Counter 6 start<br>15h = Counter6 stop<br>16h = Counter7 period match interrupt<br>17h = Counter 7 start<br>18h = Counter7 stop |



Copyright © 2026 Texas Instruments Incorporated

### 36.4.8 IMASK Register (Offset = 1028h) [Reset = 00000000h]

IMASK is shown in Table 36-19.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

**Table 36-19. IMASK Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | RESERVED | R/W | 0h | |
| 23 | CNT7STOP | R/W | 0h | CNT7STOP Interrupt enable bit<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 22 | CNT7STRT | R/W | 0h | CNT7STRT Interrupt enable bit<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 21 | CNT7OVF | R/W | 0h | CNT7OVF Interrupt enable bit<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 20 | CNT6STOP | R/W | 0h | CNT6STOP Interrupt enable bit<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 19 | CNT6STRT | R/W | 0h | CNT6STRT Interrupt enable bit<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 18 | CNT6OVF | R/W | 0h | CNT6OVF Interrupt enable bit<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 17 | CNT5STOP | R/W | 0h | CNT5STOP Interrupt enable bit<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 16 | CNT5STRT | R/W | 0h | CNT5STRT Interrupt enable bit<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 15 | CNT5OVF | R/W | 0h | CNT5OVF Interrupt enable bit<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 14 | CNT4STOP | R/W | 0h | CNT4STOP Interrupt enable bit<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 13 | CNT4STRT | R/W | 0h | CNT4STRT Interrupt enable bit<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 12 | CNT4OVF | R/W | 0h | CNT4OVF Interrupt enable bit<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 11 | CNT3STOP | R/W | 0h | CNT3STOP Interrupt enable bit<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 10 | CNT3STRT | R/W | 0h | CNT3STRT Interrupt enable bit<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 9 | CNT3OVF | R/W | 0h | CNT3OVF Interrupt enable bit<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 8 | CNT2STOP | R/W | 0h | CNT2STOP Interrupt enable bit<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |



Copyright © 2026 Texas Instruments Incorporated

**Table 36-19. IMASK Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7 | CNT2STRT | R/W | 0h | CNT2STRT Interrupt enable bit<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 6 | CNT2OVF | R/W | 0h | CNT2OVF Interrupt enable bit<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 5 | CNT1STOP | R/W | 0h | CNT1STOP Interrupt enable bit<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 4 | CNT1STRT | R/W | 0h | CNT1STRT Interrupt enable bit<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 3 | CNT1OVF | R/W | 0h | CNT1OVF Interrupt enable bit<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 2 | CNT0STOP | R/W | 0h | CNT0STOP Interrupt enable bit<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 1 | CNT0STRT | R/W | 0h | CNT0STRT Interrupt enable bit<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 0 | CNT0OVF | R/W | 0h | CNT0OVF Interrupt enable bit<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |



Copyright © 2026 Texas Instruments Incorporated

### 36.4.9 RIS Register (Offset = 1030h) [Reset = 00000000h]

RIS is shown in Table 36-20.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Table 36-20. RIS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | RESERVED | R | 0h | |
| 23 | CNT7STOP | RH | 0h | Counter 7 stop event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 22 | CNT7STRT | RH | 0h | Counter 7 start event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 21 | CNT7OVF | RH | 0h | Counter 7 Period match flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 20 | CNT6STOP | RH | 0h | Counter 6 stop event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 19 | CNT6STRT | RH | 0h | Counter 6 start event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 18 | CNT6OVF | RH | 0h | Counter 6 Period match flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 17 | CNT5STOP | RH | 0h | Counter 5 stop event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 16 | CNT5STRT | RH | 0h | Counter 5 start event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 15 | CNT5OVF | RH | 0h | Counter 5 Period match flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 14 | CNT4STOP | RH | 0h | Counter 4 stop event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 13 | CNT4STRT | RH | 0h | Counter 4 start event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 12 | CNT4OVF | RH | 0h | Counter 4 Period match flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 11 | CNT3STOP | RH | 0h | Counter 3 stop event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 10 | CNT3STRT | RH | 0h | Counter 3 start event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 9 | CNT3OVF | RH | 0h | Counter 3 Period match flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |



Copyright © 2026 Texas Instruments Incorporated

**Table 36-20. RIS Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 8 | CNT2STOP | RH | 0h | Counter 2 stop event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 7 | CNT2STRT | RH | 0h | Counter 2 start event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 6 | CNT2OVF | RH | 0h | Counter 2 Period match flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 5 | CNT1STOP | RH | 0h | Counter 1 stop event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 4 | CNT1STRT | RH | 0h | Counter 1 start event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 3 | CNT1OVF | RH | 0h | Counter 1 Period match flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 2 | CNT0STOP | RH | 0h | Counter 0 stop event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 1 | CNT0STRT | RH | 0h | Counter 0 start event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 0 | CNT0OVF | RH | 0h | Counter 0 Period match flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |



Copyright © 2026 Texas Instruments Incorporated

### 36.4.10 MIS Register (Offset = 1038h) [Reset = 00000000h]

MIS is shown in Table 36-21.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Table 36-21. MIS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | RESERVED | R | 0h | |
| 23 | CNT7STOP | R | 0h | Counter 7 stop event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 22 | CNT7STRT | R | 0h | Counter 7 start event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 21 | CNT7OVF | R | 0h | Counter 7 period match flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 20 | CNT6STOP | R | 0h | Counter 6 stop event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 19 | CNT6STRT | R | 0h | Counter 6 start event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 18 | CNT6OVF | R | 0h | Counter 6 period match flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 17 | CNT5STOP | R | 0h | Counter 5 stop event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 16 | CNT5STRT | R | 0h | Counter 5 start event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 15 | CNT5OVF | R | 0h | Counter 5 period match flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 14 | CNT4STOP | R | 0h | Counter 4 stop event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 13 | CNT4STRT | R | 0h | Counter 4 start event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 12 | CNT4OVF | R | 0h | Counter 4 period match flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 11 | CNT3STOP | R | 0h | Counter 3 stop event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 10 | CNT3STRT | R | 0h | Counter 3 start event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 9 | CNT3OVF | R | 0h | Counter 3 period match flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 8 | CNT2STOP | R | 0h | Counter 2 stop event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |



Copyright © 2026 Texas Instruments Incorporated

**Table 36-21. MIS Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7 | CNT2STRT | R | 0h | Counter 2 start event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 6 | CNT2OVF | R | 0h | Counter 2 period match flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 5 | CNT1STOP | R | 0h | Counter 1 stop event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 4 | CNT1STRT | R | 0h | Counter 1 start event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 3 | CNT1OVF | R | 0h | Counter 1 period match flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 2 | CNT0STOP | R | 0h | Counter 0 stop event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 1 | CNT0STRT | R | 0h | Counter 0 start event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 0 | CNT0OVF | R | 0h | Counter 0 period match flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |



Copyright © 2026 Texas Instruments Incorporated

### 36.4.11 ISET Register (Offset = 1040h) [Reset = 00000000h]

ISET is shown in Table 36-22.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Table 36-22. ISET Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | RESERVED | W | 0h | |
| 23 | CNT7STOP | W | 0h | Set Counter 7 stop event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 22 | CNT7STRT | W | 0h | Set Counter 7 start event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 21 | CNT7OVF | W | 0h | Set Counter 7 period match flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 20 | CNT6STOP | W | 0h | Set Counter 6 stop event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 19 | CNT6STRT | W | 0h | Set Counter 6 start event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 18 | CNT6OVF | W | 0h | Set Counter 6 period match flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 17 | CNT5STOP | W | 0h | Set Counter 5 stop event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 16 | CNT5STRT | W | 0h | Set Counter 5 start event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 15 | CNT5OVF | W | 0h | Set Counter 5 period match flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 14 | CNT4STOP | W | 0h | Set Counter 4 stop event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 13 | CNT4STRT | W | 0h | Set Counter 4 start event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 12 | CNT4OVF | W | 0h | Set Counter 4 period match flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 11 | CNT3STOP | W | 0h | Set Counter 3 stop event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 10 | CNT3STRT | W | 0h | Set Counter 3 start event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 9 | CNT3OVF | W | 0h | Set Counter 3 period match flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |



Copyright © 2026 Texas Instruments Incorporated

**Table 36-22. ISET Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 8 | CNT2STOP | W | 0h | Set Counter 2 stop event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 7 | CNT2STRT | W | 0h | Set Counter 2 start event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 6 | CNT2OVF | W | 0h | Set Counter 2 period match flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 5 | CNT1STOP | W | 0h | Set Counter 1 stop event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 4 | CNT1STRT | W | 0h | Set Counter 1 start event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 3 | CNT1OVF | W | 0h | Set Counter 1 period match flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 2 | CNT0STOP | W | 0h | Set Counter 0 stop event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 1 | CNT0STRT | W | 0h | Set Counter 0 start event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 0 | CNT0OVF | W | 0h | Set Counter 0 period match flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |



Copyright © 2026 Texas Instruments Incorporated

### 36.4.12 ICLR Register (Offset = 1048h) [Reset = 00000000h]

ICLR is shown in Table 36-23.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

**Table 36-23. ICLR Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | RESERVED | W | 0h | |
| 23 | CNT7STOP | W | 0h | Clear Counter 7 stop event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 22 | CNT7STRT | W | 0h | Clear Counter 7 start event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 21 | CNT7OVF | W | 0h | Clear Counter 7 period match flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 20 | CNT6STOP | W | 0h | Clear Counter 6 stop event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 19 | CNT6STRT | W | 0h | Clear Counter 6 start event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 18 | CNT6OVF | W | 0h | Clear Counter 6 period match flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 17 | CNT5STOP | W | 0h | Clear Counter 5 stop event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 16 | CNT5STRT | W | 0h | Clear Counter 5 start event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 15 | CNT5OVF | W | 0h | Clear Counter 5 period match flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 14 | CNT4STOP | W | 0h | Clear Counter 4 stop event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 13 | CNT4STRT | W | 0h | Clear Counter 4 start event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 12 | CNT4OVF | W | 0h | Clear Counter 4 period match flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 11 | CNT3STOP | W | 0h | Clear Counter 3 stop event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 10 | CNT3STRT | W | 0h | Clear Counter 3 start event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 9 | CNT3OVF | W | 0h | Clear Counter 3 period match flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 8 | CNT2STOP | W | 0h | Clear Counter 2 stop event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |



Copyright © 2026 Texas Instruments Incorporated

**Table 36-23. ICLR Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7 | CNT2STRT | W | 0h | Clear Counter 2 start event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 6 | CNT2OVF | W | 0h | Clear Counter 2 period match flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 5 | CNT1STOP | W | 0h | Clear Counter 1 stop event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 4 | CNT1STRT | W | 0h | Clear Counter 1 start event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 3 | CNT1OVF | W | 0h | Clear Counter 1 period match flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 2 | CNT0STOP | W | 0h | Clear Counter 0 stop event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 1 | CNT0STRT | W | 0h | Clear Counter 0 start event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 0 | CNT0OVF | W | 0h | Clear Counter 0 period match flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |



Copyright © 2026 Texas Instruments Incorporated

### 36.4.13 IIDX Register (Offset = 1050h) [Reset = 00000000h]

IIDX is shown in Table 36-24.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. 0x0 means no event pending. Interrupt 1 is the highest priority, 2 next highest priority etc. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred.
On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in the RIS and MIS are cleared as well. After a read from the CPU (not from the debug interface), the register must be updated with the next highest priority interrupt, if none are pending, then it should display 0x0.

**Table 36-24. IIDX Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-10 | RESERVED | R | 0h | |
| 9-0 | STAT | R | 0h | Interrupt index status<br>00h = No bit is set means there is no pending interrupt request<br>01h = Counter0 period match interrupt<br>02h = Counter 0 start<br>03h = Counter0 stop<br>04h = Counter1 period match interrupt<br>05h = Counter 1 start<br>06h = Counter1 stop<br>07h = Counter2 period match interrupt<br>08h = Counter 2 start<br>09h = Counter2 stop<br>0Ah = Counter3 period match interrupt<br>0Bh = Counter 3 start<br>0Ch = Counter3 stop<br>Dh = Counter4 period match interrupt<br>Eh = Counter 4 start<br>Fh = Counter4 stop<br>10h = Counter5 period match interrupt<br>11h = Counter 5 start<br>12h = Counter5 stop<br>13h = Counter6 period match interrupt<br>14h = Counter 6 start<br>15h = Counter6 stop<br>16h = Counter7 period match interrupt<br>17h = Counter 7 start<br>18h = Counter7 stop |



Copyright © 2026 Texas Instruments Incorporated

### 36.4.14 IMASK Register (Offset = 1058h) [Reset = 00000000h]

IMASK is shown in Table 36-25.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

**Table 36-25. IMASK Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | RESERVED | R/W | 0h | |
| 23 | CNT7STOP | R/W | 0h | CNT7STOP Interrupt enable bit<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 22 | CNT7STRT | R/W | 0h | CNT7STRT Interrupt enable bit<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 21 | CNT7OVF | R/W | 0h | CNT7OVF Interrupt enable bit<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 20 | CNT6STOP | R/W | 0h | CNT6STOP Interrupt enable bit<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 19 | CNT6STRT | R/W | 0h | CNT6STRT Interrupt enable bit<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 18 | CNT6OVF | R/W | 0h | CNT6OVF Interrupt enable bit<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 17 | CNT5STOP | R/W | 0h | CNT5STOP Interrupt enable bit<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 16 | CNT5STRT | R/W | 0h | CNT5STRT Interrupt enable bit<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 15 | CNT5OVF | R/W | 0h | CNT5OVF Interrupt enable bit<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 14 | CNT4STOP | R/W | 0h | CNT4STOP Interrupt enable bit<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 13 | CNT4STRT | R/W | 0h | CNT4STRT Interrupt enable bit<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 12 | CNT4OVF | R/W | 0h | CNT4OVF Interrupt enable bit<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 11 | CNT3STOP | R/W | 0h | CNT3STOP Interrupt enable bit<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 10 | CNT3STRT | R/W | 0h | CNT3STRT Interrupt enable bit<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 9 | CNT3OVF | R/W | 0h | CNT3OVF Interrupt enable bit<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 8 | CNT2STOP | R/W | 0h | CNT2STOP Interrupt enable bit<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |



Copyright © 2026 Texas Instruments Incorporated

**Table 36-25. IMASK Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7 | CNT2STRT | R/W | 0h | CNT2STRT Interrupt enable bit<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 6 | CNT2OVF | R/W | 0h | CNT2OVF Interrupt enable bit<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 5 | CNT1STOP | R/W | 0h | CNT1STOP Interrupt enable bit<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 4 | CNT1STRT | R/W | 0h | CNT1STRT Interrupt enable bit<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 3 | CNT1OVF | R/W | 0h | CNT1OVF Interrupt enable bit<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 2 | CNT0STOP | R/W | 0h | CNT0STOP Interrupt enable bit<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 1 | CNT0STRT | R/W | 0h | CNT0STRT Interrupt enable bit<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 0 | CNT0OVF | R/W | 0h | CNT0OVF Interrupt enable bit<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |



Copyright © 2026 Texas Instruments Incorporated

### 36.4.15 RIS Register (Offset = 1060h) [Reset = 00000000h]

RIS is shown in Table 36-26.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Table 36-26. RIS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | RESERVED | R | 0h | |
| 23 | CNT7STOP | RH | 0h | Counter 7 stop event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 22 | CNT7STRT | RH | 0h | Counter 7 start event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 21 | CNT7OVF | RH | 0h | Counter 7 Period match flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 20 | CNT6STOP | RH | 0h | Counter 6 stop event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 19 | CNT6STRT | RH | 0h | Counter 6 start event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 18 | CNT6OVF | RH | 0h | Counter 6 Period match flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 17 | CNT5STOP | RH | 0h | Counter 5 stop event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 16 | CNT5STRT | RH | 0h | Counter 5 start event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 15 | CNT5OVF | RH | 0h | Counter 5 Period match flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 14 | CNT4STOP | RH | 0h | Counter 4 stop event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 13 | CNT4STRT | RH | 0h | Counter 4 start event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 12 | CNT4OVF | RH | 0h | Counter 4 Period match flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 11 | CNT3STOP | RH | 0h | Counter 3 stop event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 10 | CNT3STRT | RH | 0h | Counter 3 start event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 9 | CNT3OVF | RH | 0h | Counter 3 Period match flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |



Copyright © 2026 Texas Instruments Incorporated

**Table 36-26. RIS Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 8 | CNT2STOP | RH | 0h | Counter 2 stop event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 7 | CNT2STRT | RH | 0h | Counter 2 start event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 6 | CNT2OVF | RH | 0h | Counter 2 Period match flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 5 | CNT1STOP | RH | 0h | Counter 1 stop event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 4 | CNT1STRT | RH | 0h | Counter 1 start event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 3 | CNT1OVF | RH | 0h | Counter 1 Period match flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 2 | CNT0STOP | RH | 0h | Counter 0 stop event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 1 | CNT0STRT | RH | 0h | Counter 0 start event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 0 | CNT0OVF | RH | 0h | Counter 0 Period match flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |



Copyright © 2026 Texas Instruments Incorporated

### 36.4.16 MIS Register (Offset = 1068h) [Reset = 00000000h]

MIS is shown in Table 36-27.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Table 36-27. MIS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | RESERVED | R | 0h | |
| 23 | CNT7STOP | R | 0h | Counter 7 stop event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 22 | CNT7STRT | R | 0h | Counter 7 start event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 21 | CNT7OVF | R | 0h | Counter 7 period match flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 20 | CNT6STOP | R | 0h | Counter 6 stop event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 19 | CNT6STRT | R | 0h | Counter 6 start event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 18 | CNT6OVF | R | 0h | Counter 6 period match flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 17 | CNT5STOP | R | 0h | Counter 5 stop event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 16 | CNT5STRT | R | 0h | Counter 5 start event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 15 | CNT5OVF | R | 0h | Counter 5 period match flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 14 | CNT4STOP | R | 0h | Counter 4 stop event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 13 | CNT4STRT | R | 0h | Counter 4 start event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 12 | CNT4OVF | R | 0h | Counter 4 period match flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 11 | CNT3STOP | R | 0h | Counter 3 stop event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 10 | CNT3STRT | R | 0h | Counter 3 start event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 9 | CNT3OVF | R | 0h | Counter 3 period match flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 8 | CNT2STOP | R | 0h | Counter 2 stop event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |



Copyright © 2026 Texas Instruments Incorporated

**Table 36-27. MIS Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7 | CNT2STRT | R | 0h | Counter 2 start event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 6 | CNT2OVF | R | 0h | Counter 2 period match flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 5 | CNT1STOP | R | 0h | Counter 1 stop event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 4 | CNT1STRT | R | 0h | Counter 1 start event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 3 | CNT1OVF | R | 0h | Counter 1 period match flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 2 | CNT0STOP | R | 0h | Counter 0 stop event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 1 | CNT0STRT | R | 0h | Counter 0 start event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 0 | CNT0OVF | R | 0h | Counter 0 period match flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |



Copyright © 2026 Texas Instruments Incorporated

### 36.4.17 ISET Register (Offset = 1070h) [Reset = 00000000h]

ISET is shown in Table 36-28.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Table 36-28. ISET Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | RESERVED | W | 0h | |
| 23 | CNT7STOP | W | 0h | Set Counter 7 stop event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 22 | CNT7STRT | W | 0h | Set Counter 7 start event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 21 | CNT7OVF | W | 0h | Set Counter 7 period match flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 20 | CNT6STOP | W | 0h | Set Counter 6 stop event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 19 | CNT6STRT | W | 0h | Set Counter 6 start event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 18 | CNT6OVF | W | 0h | Set Counter 6 period match flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 17 | CNT5STOP | W | 0h | Set Counter 5 stop event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 16 | CNT5STRT | W | 0h | Set Counter 5 start event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 15 | CNT5OVF | W | 0h | Set Counter 5 period match flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 14 | CNT4STOP | W | 0h | Set Counter 4 stop event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 13 | CNT4STRT | W | 0h | Set Counter 4 start event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 12 | CNT4OVF | W | 0h | Set Counter 4 period match flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 11 | CNT3STOP | W | 0h | Set Counter 3 stop event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 10 | CNT3STRT | W | 0h | Set Counter 3 start event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 9 | CNT3OVF | W | 0h | Set Counter 3 period match flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |



Copyright © 2026 Texas Instruments Incorporated

**Table 36-28. ISET Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 8 | CNT2STOP | W | 0h | Set Counter 2 stop event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 7 | CNT2STRT | W | 0h | Set Counter 2 start event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 6 | CNT2OVF | W | 0h | Set Counter 2 period match flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 5 | CNT1STOP | W | 0h | Set Counter 1 stop event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 4 | CNT1STRT | W | 0h | Set Counter 1 start event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 3 | CNT1OVF | W | 0h | Set Counter 1 period match flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 2 | CNT0STOP | W | 0h | Set Counter 0 stop event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 1 | CNT0STRT | W | 0h | Set Counter 0 start event flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 0 | CNT0OVF | W | 0h | Set Counter 0 period match flag<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |



Copyright © 2026 Texas Instruments Incorporated

### 36.4.18 ICLR Register (Offset = 1078h) [Reset = 00000000h]

ICLR is shown in Table 36-29.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

**Table 36-29. ICLR Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | RESERVED | W | 0h | |
| 23 | CNT7STOP | W | 0h | Clear Counter 7 stop event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 22 | CNT7STRT | W | 0h | Clear Counter 7 start event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 21 | CNT7OVF | W | 0h | Clear Counter 7 period match flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 20 | CNT6STOP | W | 0h | Clear Counter 6 stop event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 19 | CNT6STRT | W | 0h | Clear Counter 6 start event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 18 | CNT6OVF | W | 0h | Clear Counter 6 period match flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 17 | CNT5STOP | W | 0h | Clear Counter 5 stop event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 16 | CNT5STRT | W | 0h | Clear Counter 5 start event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 15 | CNT5OVF | W | 0h | Clear Counter 5 period match flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 14 | CNT4STOP | W | 0h | Clear Counter 4 stop event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 13 | CNT4STRT | W | 0h | Clear Counter 4 start event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 12 | CNT4OVF | W | 0h | Clear Counter 4 period match flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 11 | CNT3STOP | W | 0h | Clear Counter 3 stop event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 10 | CNT3STRT | W | 0h | Clear Counter 3 start event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 9 | CNT3OVF | W | 0h | Clear Counter 3 period match flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 8 | CNT2STOP | W | 0h | Clear Counter 2 stop event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |



Copyright © 2026 Texas Instruments Incorporated

**Table 36-29. ICLR Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 7 | CNT2STRT | W | 0h | Clear Counter 2 start event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 6 | CNT2OVF | W | 0h | Clear Counter 2 period match flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 5 | CNT1STOP | W | 0h | Clear Counter 1 stop event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 4 | CNT1STRT | W | 0h | Clear Counter 1 start event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 3 | CNT1OVF | W | 0h | Clear Counter 1 period match flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 2 | CNT0STOP | W | 0h | Clear Counter 0 stop event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 1 | CNT0STRT | W | 0h | Clear Counter 0 start event flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |
| 0 | CNT0OVF | W | 0h | Clear Counter 0 period match flag.<br>0h = Interrupt is not pending.<br>1h = Interrupt is pending. |



Copyright © 2026 Texas Instruments Incorporated

### 36.4.19 EVT_MODE Register (Offset = 10E0h) [Reset = 00000000h]

EVT_MODE is shown in Table 36-30.

Return to the Summary Table.

Event mode register. It is used to select whether each line is disabled, in software mode (software clears the RIS) or in hardware mode (hardware clears the RIS)

**Table 36-30. EVT_MODE Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3-2 | GEN_EVENT_CFG | R | 0h | Event line mode select for event corresponding to CPU_INT<br>0h = The interrupt or event line is disabled.<br>1h = The interrupt or event line is in software mode. Software must clear the RIS.<br>2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag. |
| 1-0 | CPU_INT_CFG | R | 1h | Event line mode select for event corresponding to CPU_INT<br>0h = The interrupt or event line is disabled.<br>1h = The interrupt or event line is in software mode. Software must clear the RIS.<br>2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag. |



Copyright © 2026 Texas Instruments Incorporated

### 36.4.20 CTL0_j Register (Offset = 1100h + formula) [Reset = 00000000h]

CTL0_j is shown in Table 36-31.

Return to the Summary Table.

CTL0 register

Offset = 1100h + (j * 100h); where j = 0h to 7h

**Table 36-31. CTL0_j Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | RESERVED | R/W | 0h | |
| 19-16 | CLKSEL | R/W | 0h | Clock sources |
| 15-12 | RESETSEL | R/W | 0h | Selects the source used to reset the counter. |
| 11-8 | STOPSEL | R/W | 0h | Selects the source used to stop the counter. |
| 7-4 | STARTSEL | R/W | 0h | Selects the source used to start the counter. |
| 3-1 | RESERVED | R/W | 0h | |
| 0 | EN | RH/W | 0h | Counter Enable |



Copyright © 2026 Texas Instruments Incorporated

### 36.4.21 LD_j Register (Offset = 1104h + formula) [Reset = 00000000h]

LD_j is shown in Table 36-32.

Return to the Summary Table.

Period register.

Offset = 1104h + (j * 100h); where j = 0h to 7h

**Table 36-32. LD_j Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R/W | 0h | |
| 15-0 | VAL | R/W | 0h | Period value<br>0001h (R/W) = Minimum counter value<br>FFFFh (R/W) = Maximum counter value |



Copyright © 2026 Texas Instruments Incorporated

### 36.4.22 CNT_j Register (Offset = 1108h + formula) [Reset = 00000000h]

CNT_j is shown in Table 36-33.

Return to the Summary Table.

Counter register.

Offset = 1108h + (j * 100h); where j = 0h to 7h

**Table 36-33. CNT_j Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15-0 | VALUE | RH/W | 0h | Counter Value |



Copyright © 2026 Texas Instruments Incorporated

Chapter 37

# Low Frequency Subsystem (LFSS_B)


**37.1 Overview** ..... 2299
**37.2 Clock System** ..... 2299
**37.3 LFSS Reset** ..... 2300
**37.4 Real Time Counter (RTC_x)** ..... 2300
**37.5 Independent Watchdog Timer (IWDT)** ..... 2301
**37.6 Lock Function of RTC and IWDT** ..... 2301
**37.7 LFSS Registers** ..... 2302





Copyright © 2026 Texas Instruments Incorporated

## 37.1 Overview

The Low-Frequency Subsystem (LFSS) combines several functional peripherals under one shared subsystem. These peripherals are clocked by the low-frequency clock (LFCLK) or need to be active during low-power modes. The low-frequency clock has a typical frequency of 32kHz and is mainly intended for long-term timekeeping. In some devices, the LFSS is powered by a separate battery backup power domain called PDB, and the device will have a physical VBAT power pin present.

**Note**

Not all the features described in this chapter are present in all devices. Refer to the device-specific data sheet to know which features of LFSS are present in the specific device.

The superset version of the LFSS contains following components:

- A dedicated battery backup power domain and a dedicated supply pin (VBAT)
- Real time clock (RTC) with additional prescalar extension and timestamp captures
- An independent watchdog timer (IWDT)
- Tamper detection input/output (TIO) module
- A small scratchpad memory storage (SPM)

## 37.2 Clock System

The low-frequency clock system is part of LFSS, but the control bits are located in the SYSCTL module. Figure 37-1 illustrates the components of the clock system.

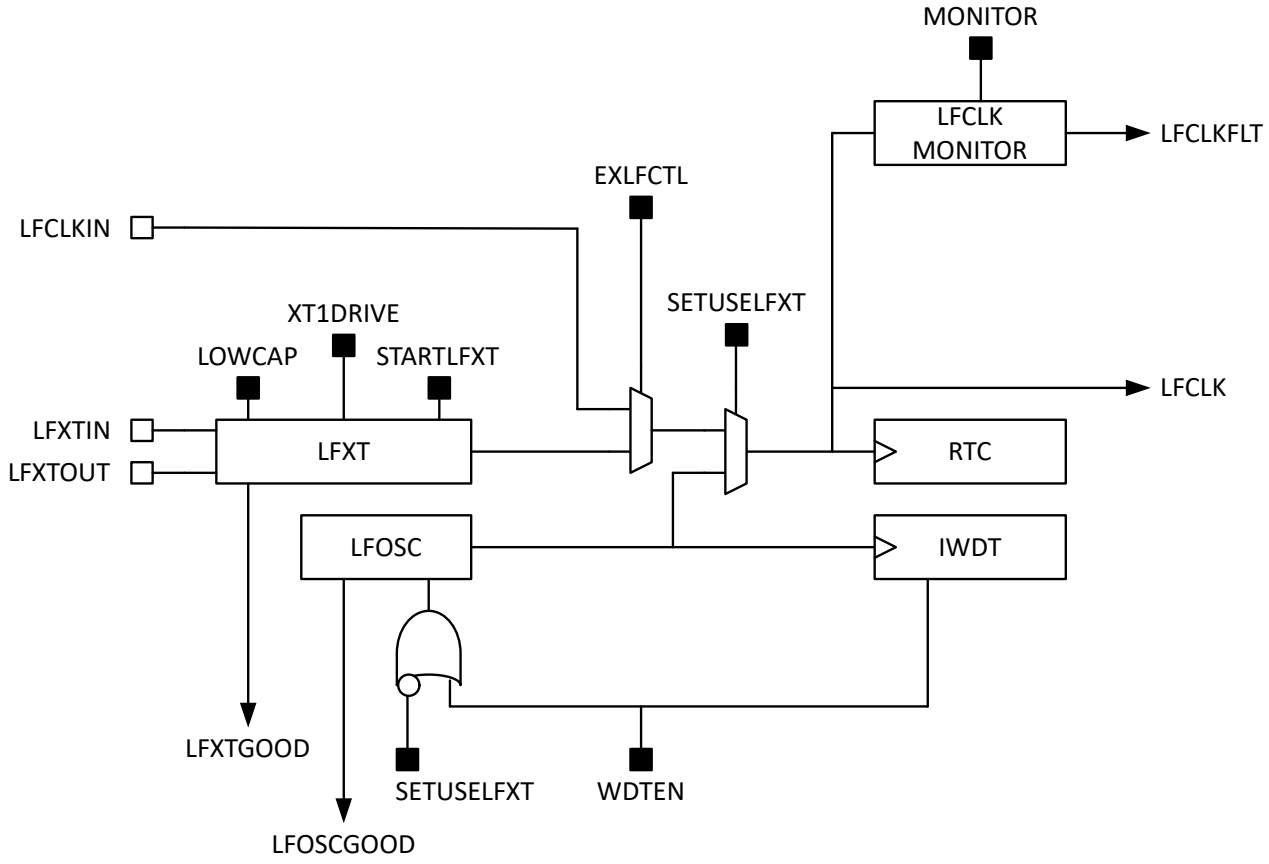



**Figure 37-1. LFSS Low-Frequency Clock System**

The control bits for LFCLK control is located inside the SYSCTL registers. In case of a power loss of the VDD / VCORE domain, the contents of the registers are lost.



Copyright © 2026 Texas Instruments Incorporated

In devices with a dedicated backup power domain (PDB) which remains powered by VBAT:

- The shadow latches will keep the LFCLK configuration active until the power returns
- The shadow latches are not reset on return of power and keep the last configuration status until the device clock configuration is restored by software

In devices without a dedicated backup power domain:

- The LFSS will not be operational without VDD / VCORE present

In both scenarios, the user must reconfigure the LFCLK registers after restoring power on VDD / VCORE to the device.

## 37.3 LFSS Reset

In devices with a dedicated battery backup power domain (PDB) with supply pin (VBAT), the LFSS has its VBAT power management unit (VBAT PMU) and its own reset generation circuit related to the VBAT supply pin.

**Note**

For devices without the PDB and VBAT supply pin, refer to the respective TRM chapters on the supported peripherals (RTC, IWDT)

Two subcircuits are related to the reset of the LFSS.

- The VBAT power-on-reset (POR) circuit is a $V_{th}$ based supply voltage monitor. The POR circuit is used to reset the VBAT PMU and initiate the startup of the cold boot sequence.
- The VBAT brown-out-reset (BOR) circuit is a bandgap reference-based voltage monitor and enables the VBAT PMU LDO when the VBAT supply is large enough to operate the LDO safely. The battery backup power domain (PDB) becomes available after this sequence, and will reset all logic in the PDB.

The battery backup power domain (PDB) does not experience a reset unless the power supply on the VBAT pin drops below the VBAT BOR level. The VDD BOR reset from the Main PMU does not reset the PDB domain while VBAT remains sufficiently powered.

In addition, the LFSS supports an VBAT PMU POR level software-reset request. This reset supports software development and from the LFSS point of view it looks like the VBAT supply was temporary disconnected and reconnected. This also means the PDB domain is brought down and back up and all flops have a full reset.

**Note**

The software POR request is mainly for the software-development to emulate the removal of the external battery from VBAT. After this POR request event, the software needs to re-enable the LFXTAL, switch the LFCLK to the LFXTAL, and finally re-initialize the peripherals within LFSS (RTC, IWDT, TIO, SPM).

After the LFXT and LFSS peripherals are configured and initialized by software, a persistent status bit in the LFSS domain indicates the “running / RTC active” status to the software. On device power-up, the software reads the reset cause table and initializes the device accordingly (for example, initial power up or wake from SHUTDOWN mode). Before initializing the RTC, the software should check the LFXT and RTC good/active status and skip the initialization when already running and active.

## 37.4 Real Time Counter (RTC_x)

The low-frequency subsystem (LFSS) contains a real-time counter (RTC).

In devices for which the LFSS is powered by an independent VBAT supply pin to support the backup-battery power domain (PDB), the RTC variant has extended features and is referred to as RTC_A.

In devices for which the LFSS is powered internally by the main power supply pin VDD, the RTC variant is referred as RTC_B.

The design of the RTC_x instances are intended to be backward compatible with RTC.



Copyright © 2026 Texas Instruments Incorporated

For more information on the LFSS real-time counter and various RTC instances, see the RTC chapter.

## 37.5 Independent Watchdog Timer (IWDT)

The low-frequency subsystem (LFSS) contains an independent watchdog timer (IWDT).

- In devices for which the LFSS is powered by an independent VBAT supply pin to support the backup-battery power domain (PDB), the IWDT variant is independent in the sense of having both an independent power supply and an independent, fail-safe clock source.
- In devices for which the LFSS is powered internally by the main power supply pin VDD, the IWDT variant is only independent in the sense of having an independent, fail-safe clock source.

The independent watchdog timer (IWDT) is a device-independent supervisor which monitors code execution and overall hang up scenarios of the device. If the application software does not successfully reset the watchdog within the programmed time, the watchdog generates a POR reset to the device.

For more information on the LFSS independent watchdog timer, see the IWDT chapter.

## 37.6 Lock Function of RTC and IWDT

Safety or secure applications must make sure that the RTC value and IWDT configurations cannot be changed after initial configuration on accident by software or during an address transient error during an intentional write access. Therefore, the RTC counter, LFXT clock configuration, and IWDT configuration registers are protected by a lock bit.

When the lock bit is set, the protected registers cannot be changed even if the correct key is applied. The lock bit is writable and can be unset in case the RTC time needs to be updated or some other configuration needs to be changed. The RTC alarm, RTC timestamp, and IWDT restart function are not affected by the lock bit.



Copyright © 2026 Texas Instruments Incorporated

## 37.7 LFSS Registers

Table 37-1 lists the memory-mapped registers for the LFSS registers. All register offset addresses not listed in Table 37-1 should be considered as reserved locations and the register contents should not be modified.

**Table 37-1. LFSS Registers**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 400h | FSUB_0 | Subsciber Port 0 | | Go |
| 444h | FPUB_0 | Publisher Port 0 | | Go |
| 1004h | CLKSEL | Clock Select for Ultra Low Power peripherals | | Go |
| 1020h | IIDX | Interrupt Index Register | | Go |
| 1028h | IMASK | Interrupt mask | | Go |
| 1030h | RIS | Raw interrupt status | | Go |
| 1038h | MIS | Masked interrupt status | | Go |
| 1040h | ISET | Interrupt set | | Go |
| 1048h | ICLR | Interrupt clear | | Go |
| 1050h | IIDX | Interrupt Index Register | | Go |
| 1058h | IMASK | Interrupt mask | | Go |
| 1060h | RIS | Raw interrupt status | | Go |
| 1068h | MIS | Masked interrupt status | | Go |
| 1070h | ISET | Interrupt set | | Go |
| 1078h | ICLR | Interrupt clear | | Go |
| 10E0h | EVT_MODE | Event Mode | | Go |
| 10FCh | DESC | LFSS Descriptor Register | | Go |
| 1100h | CLKCTL | RTC Clock Control Register | | Go |
| 1104h | DBGCTL | RTC Module Debug Control Register | | Go |
| 1108h | CTL | RTC Control Register | | Go |
| 110Ch | STA | RTC Status Register | | Go |
| 1110h | CAL | RTC Clock Offset Calibration Register | | Go |
| 1114h | TCMP | RTC Temperature Compensation Register | | Go |
| 1118h | SEC | RTC Seconds Register - Calendar Mode With Binary / BCD Format | | Go |
| 111Ch | MIN | RTC Minutes Register - Calendar Mode With Binary / BCD Format | | Go |
| 1120h | HOUR | RTC Hours Register - Calendar Mode With Binary / BCD Format | | Go |
| 1124h | DAY | RTC Day Of Week / Month Register - Calendar Mode With Binary / BCD Format | | Go |
| 1128h | MON | RTC Month Register - Calendar Mode With Binary / BCD Format | | Go |
| 112Ch | YEAR | RTC Year Register - Calendar Mode With Binary / BCD Format | | Go |
| 1130h | A1MIN | RTC Minute Alarm Register - Calendar Mode With Binary / BCD Format | | Go |
| 1134h | A1HOUR | RTC Hours Alarm Register - Calendar Mode With Binary / BCD Format | | Go |
| 1138h | A1DAY | RTC Alarm Day Of Week / Month Register - Calendar Mode With Binary / BCD Format | | Go |



Copyright © 2026 Texas Instruments Incorporated

**Table 37-1. LFSS Registers (continued)**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 113Ch | A2MIN | RTC Minute Alarm Register - Calendar Mode With Binary / BCD Format | | Go |
| 1140h | A2HOUR | RTC Hours Alarm Register - Calendar Mode With Binary / BCD Format | | Go |
| 1144h | A2DAY | RTC Alarm Day Of Week / Month Register - Calendar Mode With Binary / BCD Format | | Go |
| 1148h | PSCTL | RTC Prescale Timer 0/1 Control Register | | Go |
| 114Ch | EXTPSCTL | RTC Prescale Timer 2 Control Register | | Go |
| 1150h | TSSEC | Time Stamp Seconds Register - Calendar Mode With Binary / BCD Format | | Go |
| 1154h | TSMIN | Time Stamp Minutes Register - Calendar Mode With Binary / BCD Format | | Go |
| 1158h | TSHOUR | Time Stamp Hours Register - Calendar Mode With Binary / BCD Format | | Go |
| 115Ch | TSDAY | Time Stamp Day Of Week / Month Register - Calendar Mode With Binary / BCD Format | | Go |
| 1160h | TSMON | Time Stamp Month Register - Calendar Mode With Binary / BCD Format | | Go |
| 1164h | TSYEAR | Time Stamp Years Register - Calendar Mode With Binary / BCD Format | | Go |
| 1168h | TSSTAT | Time Stamp Status Register | | Go |
| 116Ch | TSCTL | Time Stamp Control Register | | Go |
| 1170h | TSCLR | Time Stamp Clear Register | | Go |
| 11F0h | LFSSRST | Low frequency subsystem reset request | | Go |
| 11FCh | RTCLOCK | Real time clock lock register | | Go |
| 1200h + formula | TIOCTL[y] | Tamper I/O Control Register | | Go |
| 1280h | TOUT3_0 | Tamper Output 3 to 0 | | Go |
| 1284h | TOUT7_4 | Tamper Output 7 to 4 | | Go |
| 1288h | TOUT11_8 | Tamper Output 11 to 8 | | Go |
| 128Ch | TOUT15_12 | Tamper Output 15 to 12 | | Go |
| 1290h | TOE3_0 | Tamper Output Enable 3 to 0 | | Go |
| 1294h | TOE7_4 | Tamper Output Enable 7 to 4 | | Go |
| 1298h | TOE11_8 | Tamper Output Enable 7 to 4 | | Go |
| 129Ch | TOE15_12 | Tamper Output Enable 7 to 4 | | Go |
| 12A0h | TIN3_0 | Tamper Input Register | | Go |
| 12A4h | TIN7_4 | Tamper Input Register | | Go |
| 12A8h | TIN11_8 | Tamper Input Register | | Go |
| 12ACh | TIN15_12 | Tamper Input Register | | Go |
| 12C0h | HEARTBEAT | Heartbeat Register | | Go |
| 12FCh | TIOLOCK | Tamper I/O lock register | | Go |
| 1300h | WDTEN | Watchdog Timer Enable Register | | Go |
| 1304h | WDTDBGCTL | Watchdog Timer Debug Control Register | | Go |
| 1308h | WDTCTL | Watchdog Timer Control Register | | Go |



Copyright © 2026 Texas Instruments Incorporated

**Table 37-1. LFSS Registers (continued)**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 130Ch | WDTCNTRST | Watchdog Timer Counter Reset Register | | Go |
| 1310h | WDTSTAT | Watchdog Timer Status Register | | Go |
| 13FCh | WDTLOCK | Watchdog timer lock register | | Go |
| 1400h + formula | SPMEM[y] | Scratch Pad Memory Data Register | | Go |
| 1500h | SPMWPROT0 | Scratch Pad Memory Write Protect Register 0 | | Go |
| 1504h | SPMWPROT1 | Scratch Pad Memory Write Protect Register 1 | | Go |
| 1508h | SPMWPROT2 | Scratch Pad Memory Write Protect Register 2 | | Go |
| 150Ch | SPMWPROT3 | Scratch Pad Memory Write Protect Register 3 | | Go |
| 1510h | SPMWPROT4 | Scratch Pad Memory Write Protect Register 4 | | Go |
| 1514h | SPMWPROT5 | Scratch Pad Memory Write Protect Register 5 | | Go |
| 1518h | SPMWPROT6 | Scratch Pad Memory Write Protect Register 6 | | Go |
| 151Ch | SPMWPROT7 | Scratch Pad Memory Write Protect Register 7 | | Go |
| 1540h | SPMTERASE0 | Scratch Pad Memory Tamper Erase Register 0 | | Go |
| 1544h | SPMTERASE1 | Scratch Pad Memory Tamper Erase Register 1 | | Go |
| 1548h | SPMTERASE2 | Scratch Pad Memory Tamper Erase Register 2 | | Go |
| 154Ch | SPMTERASE3 | Scratch Pad Memory Tamper Erase Register 3 | | Go |
| 1550h | SPMTERASE4 | Scratch Pad Memory Tamper Erase Register 4 | | Go |
| 1554h | SPMTERASE5 | Scratch Pad Memory Tamper Erase Register 5 | | Go |
| 1558h | SPMTERASE6 | Scratch Pad Memory Tamper Erase Register 6 | | Go |
| 155Ch | SPMTERASE7 | Scratch Pad Memory Tamper Erase Register 7 | | Go |

Complex bit access types are encoded to fit into small table cells. Table 37-2 shows the codes that are used for access types in this section.

**Table 37-2. LFSS Access Type Codes**

| Access Type | Code | Description |
|---|---|---|
| Read Type | | |
| R | R | Read |
| R-0 | R<br>-0 | Read<br>Returns 0s |
| Rmodify | R<br>modify | Read |
| Write Type | | |
| W | W | Write |



Copyright © 2026 Texas Instruments Incorporated

**Table 37-2. LFSS Access Type Codes (continued)**

| Access Type | Code | Description |
|---|---|---|
| WK | W<br>K | Write<br>Write protected by a key |
| Reset or Default Value | | |
| *-n* | | Value after reset or the default value |
| Register Array Variables | | |
| i,j,k,l,m,n | | When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula. |
| y | | When this variable is used in a register name, an offset, or an address it refers to the value of a register array. |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.1 FSUB_0 (Offset = 400h) [Reset = 00000000h]

FSUB_0 is shown in Figure 37-2 and described in Table 37-3.

Return to the Summary Table.

Subscriber port

**Figure 37-2. FSUB_0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | CHANID | | | |
| R-0h | | | | | | | | | | | | R/W-0h | | | |

**Table 37-3. FSUB_0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3-0 | CHANID | R/W | 0h | 0 = disconnected.<br>1-15 = connected to channelID = CHANID.<br>0h = A value of 0 specifies that the event is not connected<br>Fh = Consult your device data sheet as the actual allowed maximum may be less than 15. |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.2 FPUB_0 (Offset = 444h) [Reset = 00000000h]

FPUB_0 is shown in Figure 37-3 and described in Table 37-4.

Return to the Summary Table.

Publisher port

**Figure 37-3. FPUB_0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | CHANID | | | |
| R-0h | | | | | | | | | | | | R/W-0h | | | |

**Table 37-4. FPUB_0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3-0 | CHANID | R/W | 0h | 0 = disconnected.<br>1-15 = connected to channelID = CHANID.<br>0h = A value of 0 specifies that the event is not connected<br>Fh = Consult your device data sheet as the actual allowed maximum may be less than 15. |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.3 CLKSEL (Offset = 1004h) [Reset = 00000000h]

CLKSEL is shown in Figure 37-4 and described in Table 37-5.

Return to the Summary Table.

Clock source selection for ULP peripherals

**Figure 37-4. CLKSEL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | LFCLK_SEL | RESERVED |
| R-0h | | | | | | R-0h | R-0h |

**Table 37-5. CLKSEL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | |
| 1 | LFCLK_SEL | R | 0h | Selects LFCLK as clock source if enabled<br>0h = LFCLK is disabled<br>1h = LFCLK is enabled |
| 0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.4 IIDX (Offset = 1020h) [Reset = 00000000h]

IIDX is shown in Figure 37-5 and described in Table 37-6.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. Value 0x00 means no event pending. Interrupt 1 is the highest priority, IIDX next highest, 4, 8, … IIDX^31 is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred. On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in [RIS] and [MIS] are cleared as well. After a read from the CPU (not from the debug interface), the register is updated with the next highest priority interrupt, if none are pending, then it should display 0x0.

**Figure 37-5. IIDX**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | STAT | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | |

**Table 37-6. IIDX Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | STAT | R | 0h | Interrupt index status<br>00h = No interrupt pending<br>01h = RTC ready<br>02h = RTC time event<br>03h = RTC alarm 1<br>04h = RTC alarm 2<br>05h = RTC prescale timer 0<br>06h = RTC prescale timer 1<br>07h = RTC prescale timer 2<br>08h = Time stamp event<br>09h = Tamper I/O 0 event<br>0Ah = Tamper I/O 1 event<br>0Bh = Tamper I/O 2 event<br>0Ch = Tamper I/O 3 event<br>0Dh = Tamper I/O 4 event<br>0Eh = Tamper I/O 5 event<br>0Fh = Tamper I/O 6 event<br>10h = Tamper I/O 7 event<br>11h = Tamper I/O 8 event<br>12h = Tamper I/O 9 event<br>13h = Tamper I/O 10 event<br>14h = Tamper I/O 11 event<br>15h = Tamper I/O 12 event<br>16h = Tamper I/O 13 event<br>17h = Tamper I/O 14 event<br>18h = Tamper I/O 15 event |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.5 IMASK (Offset = 1028h) [Reset = 00000000h]

IMASK is shown in Figure 37-6 and described in Table 37-7.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is unmasked. Unmasking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

**Figure 37-6. IMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TIO15 | TIO14 | TIO13 | TIO12 | TIO11 | TIO10 | TIO9 | TIO8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TIO7 | TIO6 | TIO5 | TIO4 | TIO3 | TIO2 | TIO1 | TIO0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TSEVT | RT2PS | RT1PS | RT0PS | RTCA2 | RTCA1 | RTCTEV | RTCRDY |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

**Table 37-7. IMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | RESERVED | R | 0h | |
| 23 | TIO15 | R/W | 0h | Tamper I/O 15 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 22 | TIO14 | R/W | 0h | Tamper I/O 14 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 21 | TIO13 | R/W | 0h | Tamper I/O 13 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 20 | TIO12 | R/W | 0h | Tamper I/O 12 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 19 | TIO11 | R/W | 0h | Tamper I/O 11 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 18 | TIO10 | R/W | 0h | Tamper I/O 10 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 17 | TIO9 | R/W | 0h | Tamper I/O 9 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 16 | TIO8 | R/W | 0h | Tamper I/O 8 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 15 | TIO7 | R/W | 0h | Tamper I/O 7 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 14 | TIO6 | R/W | 0h | Tamper I/O 6 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |



Copyright © 2026 Texas Instruments Incorporated

**Table 37-7. IMASK Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 13 | TIO5 | R/W | 0h | Tamper I/O 5 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 12 | TIO4 | R/W | 0h | Tamper I/O 4 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 11 | TIO3 | R/W | 0h | Tamper I/O 3 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 10 | TIO2 | R/W | 0h | Tamper I/O 2 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 9 | TIO1 | R/W | 0h | Tamper I/O 1 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 8 | TIO0 | R/W | 0h | Tamper I/O 0 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 7 | TSEVT | R/W | 0h | Time stamp event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 6 | RT2PS | R/W | 0h | RTC prescale timer 2<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 5 | RT1PS | R/W | 0h | RTC prescale timer 1<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 4 | RT0PS | R/W | 0h | RTC prescale timer 0<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 3 | RTCA2 | R/W | 0h | RTC alarm 2<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 2 | RTCA1 | R/W | 0h | RTC alarm 1<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 1 | RTCTEV | R/W | 0h | RTC time event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 0 | RTCRDY | R/W | 0h | RTC ready<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.6 RIS (Offset = 1030h) [Reset = 00000000h]

RIS is shown in Figure 37-7 and described in Table 37-8.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Figure 37-7. RIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TIO15 | TIO14 | TIO13 | TIO12 | TIO11 | TIO10 | TIO9 | TIO8 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TIO7 | TIO6 | TIO5 | TIO4 | TIO3 | TIO2 | TIO1 | TIO0 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TSEVT | RT2PS | RT1PS | RT0PS | RTCA2 | RTCA1 | RTCTEV | RTCRDY |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 37-8. RIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | RESERVED | R | 0h | |
| 23 | TIO15 | R | 0h | Tamper I/O 15 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 22 | TIO14 | R | 0h | Tamper I/O 14 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 21 | TIO13 | R | 0h | Tamper I/O 13 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 20 | TIO12 | R | 0h | Tamper I/O 12 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 19 | TIO11 | R | 0h | Tamper I/O 11 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 18 | TIO10 | R | 0h | Tamper I/O 10 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 17 | TIO9 | R | 0h | Tamper I/O 9 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 16 | TIO8 | R | 0h | Tamper I/O 8 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 15 | TIO7 | R | 0h | Tamper I/O 7 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

**Table 37-8. RIS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 14 | TIO6 | R | 0h | Tamper I/O 6 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 13 | TIO5 | R | 0h | Tamper I/O 5 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 12 | TIO4 | R | 0h | Tamper I/O 4 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 11 | TIO3 | R | 0h | Tamper I/O 3 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 10 | TIO2 | R | 0h | Tamper I/O 2 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 9 | TIO1 | R | 0h | Tamper I/O 1 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 8 | TIO0 | R | 0h | Tamper I/O 0 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 7 | TSEVT | R | 0h | Time stamp event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 6 | RT2PS | R | 0h | RTC prescale timer 2<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 5 | RT1PS | R | 0h | RTC prescale timer 1<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 4 | RT0PS | R | 0h | RTC prescale timer 0<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 3 | RTCA2 | R | 0h | RTC alarm 2<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 2 | RTCA1 | R | 0h | RTC alarm 1<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 1 | RTCTEV | R | 0h | RTC time event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 0 | RTCRDY | R | 0h | RTC ready<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.7 MIS (Offset = 1038h) [Reset = 00000000h]

MIS is shown in Figure 37-8 and described in Table 37-9.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 37-8. MIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TIO15 | TIO14 | TIO13 | TIO12 | TIO11 | TIO10 | TIO9 | TIO8 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TIO7 | TIO6 | TIO5 | TIO4 | TIO3 | TIO2 | TIO1 | TIO0 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TSEVT | RT2PS | RT1PS | RT0PS | RTCA2 | RTCA1 | RTCTEV | RTCRDY |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 37-9. MIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | RESERVED | R | 0h | |
| 23 | TIO15 | R | 0h | Tamper I/O 15 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 22 | TIO14 | R | 0h | Tamper I/O 14 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 21 | TIO13 | R | 0h | Tamper I/O 13 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 20 | TIO12 | R | 0h | Tamper I/O 12 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 19 | TIO11 | R | 0h | Tamper I/O 11 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 18 | TIO10 | R | 0h | Tamper I/O 10 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 17 | TIO9 | R | 0h | Tamper I/O 9 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 16 | TIO8 | R | 0h | Tamper I/O 8 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 15 | TIO7 | R | 0h | Tamper I/O 7 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 14 | TIO6 | R | 0h | Tamper I/O 6 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

**Table 37-9. MIS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 13 | TIO5 | R | 0h | Tamper I/O 5 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 12 | TIO4 | R | 0h | Tamper I/O 4 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 11 | TIO3 | R | 0h | Tamper I/O 3 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 10 | TIO2 | R | 0h | Tamper I/O 2 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 9 | TIO1 | R | 0h | Tamper I/O 1 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 8 | TIO0 | R | 0h | Tamper I/O 0 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 7 | TSEVT | R | 0h | Time stamp event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 6 | RT2PS | R | 0h | RTC prescale timer 2<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 5 | RT1PS | R | 0h | RTC prescale timer 1<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 4 | RT0PS | R | 0h | RTC prescale timer 0<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 3 | RTCA2 | R | 0h | RTC alarm 2<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 2 | RTCA1 | R | 0h | RTC alarm 1<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 1 | RTCTEV | R | 0h | RTC time event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 0 | RTCRDY | R | 0h | RTC ready<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.8 ISET (Offset = 1040h) [Reset = 00000000h]

ISET is shown in Figure 37-9 and described in Table 37-10.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 37-9. ISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TIO15 | TIO14 | TIO13 | TIO12 | TIO11 | TIO10 | TIO9 | TIO8 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TIO7 | TIO6 | TIO5 | TIO4 | TIO3 | TIO2 | TIO1 | TIO0 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TSEVT | RT2PS | RT1PS | RT0PS | RTCA2 | RTCA1 | RTCTEV | RTCRDY |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

**Table 37-10. ISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | RESERVED | R | 0h | |
| 23 | TIO15 | W | 0h | Tamper I/O 15 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 22 | TIO14 | W | 0h | Tamper I/O 14 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 21 | TIO13 | W | 0h | Tamper I/O 13 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 20 | TIO12 | W | 0h | Tamper I/O 12 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 19 | TIO11 | W | 0h | Tamper I/O 11 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 18 | TIO10 | W | 0h | Tamper I/O 10 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 17 | TIO9 | W | 0h | Tamper I/O 9 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 16 | TIO8 | W | 0h | Tamper I/O 8 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 15 | TIO7 | W | 0h | Tamper I/O 7 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |



Copyright © 2026 Texas Instruments Incorporated

**Table 37-10. ISET Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 14 | TIO6 | W | 0h | Tamper I/O 6 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 13 | TIO5 | W | 0h | Tamper I/O 5 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 12 | TIO4 | W | 0h | Tamper I/O 4 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 11 | TIO3 | W | 0h | Tamper I/O 3 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 10 | TIO2 | W | 0h | Tamper I/O 2 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 9 | TIO1 | W | 0h | Tamper I/O 1 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 8 | TIO0 | W | 0h | Tamper I/O 0 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 7 | TSEVT | W | 0h | Time stamp event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 6 | RT2PS | W | 0h | RTC prescale timer 2<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 5 | RT1PS | W | 0h | RTC prescale timer 1<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 4 | RT0PS | W | 0h | RTC prescale timer 0<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 3 | RTCA2 | W | 0h | RTC alarm 2<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 2 | RTCA1 | W | 0h | RTC alarm 1<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 1 | RTCTEV | W | 0h | RTC time event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 0 | RTCRDY | W | 0h | RTC ready<br>0h = Writing 0 has no effect<br>1h = Set interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.9 ICLR (Offset = 1048h) [Reset = 00000000h]

ICLR is shown in Figure 37-10 and described in Table 37-11.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

**Figure 37-10. ICLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TIO15 | TIO14 | TIO13 | TIO12 | TIO11 | TIO10 | TIO9 | TIO8 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TIO7 | TIO6 | TIO5 | TIO4 | TIO3 | TIO2 | TIO1 | TIO0 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TSEVT | RT2PS | RT1PS | RT0PS | RTCA2 | RTCA1 | RTCTEV | RTCRDY |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

**Table 37-11. ICLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | RESERVED | R | 0h | |
| 23 | TIO15 | W | 0h | Tamper I/O 15 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 22 | TIO14 | W | 0h | Tamper I/O 14 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 21 | TIO13 | W | 0h | Tamper I/O 13 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 20 | TIO12 | W | 0h | Tamper I/O 12 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 19 | TIO11 | W | 0h | Tamper I/O 11 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 18 | TIO10 | W | 0h | Tamper I/O 10 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 17 | TIO9 | W | 0h | Tamper I/O 9 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 16 | TIO8 | W | 0h | Tamper I/O 8 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 15 | TIO7 | W | 0h | Tamper I/O 7 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 14 | TIO6 | W | 0h | Tamper I/O 6 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |



Copyright © 2026 Texas Instruments Incorporated

**Table 37-11. ICLR Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 13 | TIO5 | W | 0h | Tamper I/O 5 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 12 | TIO4 | W | 0h | Tamper I/O 4 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 11 | TIO3 | W | 0h | Tamper I/O 3 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 10 | TIO2 | W | 0h | Tamper I/O 2 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 9 | TIO1 | W | 0h | Tamper I/O 1 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 8 | TIO0 | W | 0h | Tamper I/O 0 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 7 | TSEVT | W | 0h | Time stamp event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 6 | RT2PS | W | 0h | RTC prescale timer 2<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 5 | RT1PS | W | 0h | RTC prescale timer 1<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 4 | RT0PS | W | 0h | RTC prescale timer 0<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 3 | RTCA2 | W | 0h | RTC alarm 2<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 2 | RTCA1 | W | 0h | RTC alarm 1<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 1 | RTCTEV | W | 0h | RTC time event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 0 | RTCRDY | W | 0h | RTC ready<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.10 IIDX (Offset = 1050h) [Reset = 00000000h]

IIDX is shown in Figure 37-11 and described in Table 37-12.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. Value 0x00 means no event pending. Interrupt 1 is the highest priority, IIDX next highest, 4, 8, … IIDX^31 is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred. On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in [RIS] and [MIS] are cleared as well. After a read from the CPU (not from the debug interface), the register is updated with the next highest priority interrupt, if none are pending, then it should display 0x0.

**Figure 37-11. IIDX**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | STAT | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | |

**Table 37-12. IIDX Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | STAT | R | 0h | Interrupt index status<br>00h = No interrupt pending<br>01h = RTC ready<br>02h = RTC time event<br>03h = RTC alarm 1<br>04h = RTC alarm 2<br>05h = RTC prescale timer 0<br>06h = RTC prescale timer 1<br>07h = RTC prescale timer 2<br>08h = Time stamp event<br>09h = Tamper I/O 0 event<br>0Ah = Tamper I/O 1 event<br>0Bh = Tamper I/O 2 event<br>0Ch = Tamper I/O 3 event<br>0Dh = Tamper I/O 4 event<br>0Eh = Tamper I/O 5 event<br>0Fh = Tamper I/O 6 event<br>10h = Tamper I/O 7 event<br>11h = Tamper I/O 8 event<br>12h = Tamper I/O 9 event<br>13h = Tamper I/O 10 event<br>14h = Tamper I/O 11 event<br>15h = Tamper I/O 12 event<br>16h = Tamper I/O 13 event<br>17h = Tamper I/O 14 event<br>18h = Tamper I/O 15 event |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.11 IMASK (Offset = 1058h) [Reset = 00000000h]

IMASK is shown in Figure 37-12 and described in Table 37-13.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is unmasked. Unmasking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

**Figure 37-12. IMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TIO15 | TIO14 | TIO13 | TIO12 | TIO11 | TIO10 | TIO9 | TIO8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TIO7 | TIO6 | TIO5 | TIO4 | TIO3 | TIO2 | TIO1 | TIO0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TSEVT | RT2PS | RT1PS | RT0PS | RTCA2 | RTCA1 | RTCTEV | RTCRDY |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

**Table 37-13. IMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | RESERVED | R | 0h | |
| 23 | TIO15 | R/W | 0h | Tamper I/O 15 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 22 | TIO14 | R/W | 0h | Tamper I/O 14 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 21 | TIO13 | R/W | 0h | Tamper I/O 13 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 20 | TIO12 | R/W | 0h | Tamper I/O 12 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 19 | TIO11 | R/W | 0h | Tamper I/O 11 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 18 | TIO10 | R/W | 0h | Tamper I/O 10 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 17 | TIO9 | R/W | 0h | Tamper I/O 9 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 16 | TIO8 | R/W | 0h | Tamper I/O 8 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 15 | TIO7 | R/W | 0h | Tamper I/O 7 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 14 | TIO6 | R/W | 0h | Tamper I/O 6 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |



Copyright © 2026 Texas Instruments Incorporated

**Table 37-13. IMASK Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 13 | TIO5 | R/W | 0h | Tamper I/O 5 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 12 | TIO4 | R/W | 0h | Tamper I/O 4 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 11 | TIO3 | R/W | 0h | Tamper I/O 3 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 10 | TIO2 | R/W | 0h | Tamper I/O 2 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 9 | TIO1 | R/W | 0h | Tamper I/O 1 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 8 | TIO0 | R/W | 0h | Tamper I/O 0 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 7 | TSEVT | R/W | 0h | Time stamp event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 6 | RT2PS | R/W | 0h | RTC prescale timer 2<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 5 | RT1PS | R/W | 0h | RTC prescale timer 1<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 4 | RT0PS | R/W | 0h | RTC prescale timer 0<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 3 | RTCA2 | R/W | 0h | RTC alarm 2<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 2 | RTCA1 | R/W | 0h | RTC alarm 1<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 1 | RTCTEV | R/W | 0h | RTC time event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 0 | RTCRDY | R/W | 0h | RTC ready<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.12 RIS (Offset = 1060h) [Reset = 00000000h]

RIS is shown in Figure 37-13 and described in Table 37-14.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Figure 37-13. RIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TIO15 | TIO14 | TIO13 | TIO12 | TIO11 | TIO10 | TIO9 | TIO8 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TIO7 | TIO6 | TIO5 | TIO4 | TIO3 | TIO2 | TIO1 | TIO0 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TSEVT | RT2PS | RT1PS | RT0PS | RTCA2 | RTCA1 | RTCTEV | RTCRDY |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 37-14. RIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | RESERVED | R | 0h | |
| 23 | TIO15 | R | 0h | Tamper I/O 15 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 22 | TIO14 | R | 0h | Tamper I/O 14 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 21 | TIO13 | R | 0h | Tamper I/O 13 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 20 | TIO12 | R | 0h | Tamper I/O 12 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 19 | TIO11 | R | 0h | Tamper I/O 11 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 18 | TIO10 | R | 0h | Tamper I/O 10 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 17 | TIO9 | R | 0h | Tamper I/O 9 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 16 | TIO8 | R | 0h | Tamper I/O 8 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 15 | TIO7 | R | 0h | Tamper I/O 7 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

**Table 37-14. RIS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 14 | TIO6 | R | 0h | Tamper I/O 6 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 13 | TIO5 | R | 0h | Tamper I/O 5 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 12 | TIO4 | R | 0h | Tamper I/O 4 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 11 | TIO3 | R | 0h | Tamper I/O 3 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 10 | TIO2 | R | 0h | Tamper I/O 2 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 9 | TIO1 | R | 0h | Tamper I/O 1 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 8 | TIO0 | R | 0h | Tamper I/O 0 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 7 | TSEVT | R | 0h | Time stamp event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 6 | RT2PS | R | 0h | RTC prescale timer 2<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 5 | RT1PS | R | 0h | RTC prescale timer 1<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 4 | RT0PS | R | 0h | RTC prescale timer 0<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 3 | RTCA2 | R | 0h | RTC alarm 2<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 2 | RTCA1 | R | 0h | RTC alarm 1<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 1 | RTCTEV | R | 0h | RTC time event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 0 | RTCRDY | R | 0h | RTC ready<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.13 MIS (Offset = 1068h) [Reset = 00000000h]

MIS is shown in Figure 37-14 and described in Table 37-15.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 37-14. MIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TIO15 | TIO14 | TIO13 | TIO12 | TIO11 | TIO10 | TIO9 | TIO8 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TIO7 | TIO6 | TIO5 | TIO4 | TIO3 | TIO2 | TIO1 | TIO0 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TSEVT | RT2PS | RT1PS | RT0PS | RTCA2 | RTCA1 | RTCTEV | RTCRDY |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 37-15. MIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | RESERVED | R | 0h | |
| 23 | TIO15 | R | 0h | Tamper I/O 15 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 22 | TIO14 | R | 0h | Tamper I/O 14 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 21 | TIO13 | R | 0h | Tamper I/O 13 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 20 | TIO12 | R | 0h | Tamper I/O 12 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 19 | TIO11 | R | 0h | Tamper I/O 11 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 18 | TIO10 | R | 0h | Tamper I/O 10 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 17 | TIO9 | R | 0h | Tamper I/O 9 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 16 | TIO8 | R | 0h | Tamper I/O 8 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 15 | TIO7 | R | 0h | Tamper I/O 7 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 14 | TIO6 | R | 0h | Tamper I/O 6 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

**Table 37-15. MIS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 13 | TIO5 | R | 0h | Tamper I/O 5 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 12 | TIO4 | R | 0h | Tamper I/O 4 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 11 | TIO3 | R | 0h | Tamper I/O 3 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 10 | TIO2 | R | 0h | Tamper I/O 2 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 9 | TIO1 | R | 0h | Tamper I/O 1 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 8 | TIO0 | R | 0h | Tamper I/O 0 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 7 | TSEVT | R | 0h | Time stamp event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 6 | RT2PS | R | 0h | RTC prescale timer 2<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 5 | RT1PS | R | 0h | RTC prescale timer 1<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 4 | RT0PS | R | 0h | RTC prescale timer 0<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 3 | RTCA2 | R | 0h | RTC alarm 2<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 2 | RTCA1 | R | 0h | RTC alarm 1<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 1 | RTCTEV | R | 0h | RTC time event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 0 | RTCRDY | R | 0h | RTC ready<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.14 ISET (Offset = 1070h) [Reset = 00000000h]

ISET is shown in Figure 37-15 and described in Table 37-16.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 37-15. ISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TIO15 | TIO14 | TIO13 | TIO12 | TIO11 | TIO10 | TIO9 | TIO8 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TIO7 | TIO6 | TIO5 | TIO4 | TIO3 | TIO2 | TIO1 | TIO0 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TSEVT | RT2PS | RT1PS | RT0PS | RTCA2 | RTCA1 | RTCTEV | RTCRDY |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

**Table 37-16. ISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | RESERVED | R | 0h | |
| 23 | TIO15 | W | 0h | Tamper I/O 15 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 22 | TIO14 | W | 0h | Tamper I/O 14 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 21 | TIO13 | W | 0h | Tamper I/O 13 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 20 | TIO12 | W | 0h | Tamper I/O 12 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 19 | TIO11 | W | 0h | Tamper I/O 11 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 18 | TIO10 | W | 0h | Tamper I/O 10 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 17 | TIO9 | W | 0h | Tamper I/O 9 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 16 | TIO8 | W | 0h | Tamper I/O 8 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 15 | TIO7 | W | 0h | Tamper I/O 7 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |



Copyright © 2026 Texas Instruments Incorporated

**Table 37-16. ISET Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 14 | TIO6 | W | 0h | Tamper I/O 6 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 13 | TIO5 | W | 0h | Tamper I/O 5 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 12 | TIO4 | W | 0h | Tamper I/O 4 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 11 | TIO3 | W | 0h | Tamper I/O 3 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 10 | TIO2 | W | 0h | Tamper I/O 2 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 9 | TIO1 | W | 0h | Tamper I/O 1 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 8 | TIO0 | W | 0h | Tamper I/O 0 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 7 | TSEVT | W | 0h | Time stamp event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 6 | RT2PS | W | 0h | RTC prescale timer 2<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 5 | RT1PS | W | 0h | RTC prescale timer 1<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 4 | RT0PS | W | 0h | RTC prescale timer 0<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 3 | RTCA2 | W | 0h | RTC alarm 2<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 2 | RTCA1 | W | 0h | RTC alarm 1<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 1 | RTCTEV | W | 0h | RTC time event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 0 | RTCRDY | W | 0h | RTC ready<br>0h = Writing 0 has no effect<br>1h = Set interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.15 ICLR (Offset = 1078h) [Reset = 00000000h]

ICLR is shown in Figure 37-16 and described in Table 37-17.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

**Figure 37-16. ICLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TIO15 | TIO14 | TIO13 | TIO12 | TIO11 | TIO10 | TIO9 | TIO8 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TIO7 | TIO6 | TIO5 | TIO4 | TIO3 | TIO2 | TIO1 | TIO0 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TSEVT | RT2PS | RT1PS | RT0PS | RTCA2 | RTCA1 | RTCTEV | RTCRDY |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

**Table 37-17. ICLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | RESERVED | R | 0h | |
| 23 | TIO15 | W | 0h | Tamper I/O 15 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 22 | TIO14 | W | 0h | Tamper I/O 14 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 21 | TIO13 | W | 0h | Tamper I/O 13 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 20 | TIO12 | W | 0h | Tamper I/O 12 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 19 | TIO11 | W | 0h | Tamper I/O 11 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 18 | TIO10 | W | 0h | Tamper I/O 10 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 17 | TIO9 | W | 0h | Tamper I/O 9 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 16 | TIO8 | W | 0h | Tamper I/O 8 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 15 | TIO7 | W | 0h | Tamper I/O 7 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 14 | TIO6 | W | 0h | Tamper I/O 6 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |



Copyright © 2026 Texas Instruments Incorporated

**Table 37-17. ICLR Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 13 | TIO5 | W | 0h | Tamper I/O 5 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 12 | TIO4 | W | 0h | Tamper I/O 4 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 11 | TIO3 | W | 0h | Tamper I/O 3 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 10 | TIO2 | W | 0h | Tamper I/O 2 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 9 | TIO1 | W | 0h | Tamper I/O 1 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 8 | TIO0 | W | 0h | Tamper I/O 0 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 7 | TSEVT | W | 0h | Time stamp event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 6 | RT2PS | W | 0h | RTC prescale timer 2<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 5 | RT1PS | W | 0h | RTC prescale timer 1<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 4 | RT0PS | W | 0h | RTC prescale timer 0<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 3 | RTCA2 | W | 0h | RTC alarm 2<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 2 | RTCA1 | W | 0h | RTC alarm 1<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 1 | RTCTEV | W | 0h | RTC time event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 0 | RTCRDY | W | 0h | RTC ready<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.16 EVT_MODE (Offset = 10E0h) [Reset = 00000000h]

EVT_MODE is shown in Figure 37-17 and described in Table 37-18.

Return to the Summary Table.

Event mode register. It is used to select whether each line is disabled, in software mode (software clears the RIS) or in hardware mode (hardware clears the RIS)

**Figure 37-17. EVT_MODE**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | EVT1_CFG | | EVT0_CFG | |
| R-0h | | | | R-0h | | R-0h | |

**Table 37-18. EVT_MODE Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3-2 | EVT1_CFG | R | 0h | Event line mode 1 select<br>2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag. |
| 1-0 | EVT0_CFG | R | 0h | Event line mode 0 select<br>1h = The interrupt or event line is in software mode. The software ISR clears the associated RIS flag. |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.17 DESC (Offset = 10FCh) [Reset = 00000000h]

DESC is shown in Figure 37-18 and described in Table 37-19.

Return to the Summary Table.

RTC Descriptor Register

**Figure 37-18. DESC**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MODULEID | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| FEATUREVER | | | | INSTNUM | | | | MAJREV | | | | MINREV | | | |
| R-0h | | | | R-0h | | | | R-0h | | | | R-0h | | | |

**Table 37-19. DESC Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | MODULEID | R | 0h | Module identifier. This ID is unique for each module. 0x2911 = Module ID of the LFSS Module<br>0h = Smallest value<br>FFFFh = Highest possible value |
| 15-12 | FEATUREVER | R | 0h | Feature set of this module. Differentiates the complexity of the actually instantiated module if there are differences.<br>0h = Smallest value<br>Fh = Highest possible value |
| 11-8 | INSTNUM | R | 0h | Instantiated version. Describes which instance of the module accessed.<br>0h = This is the default, if there is only one instance - like for SSIM |
| 7-4 | MAJREV | R | 0h | Major revision. This number holds the module revision and is incremented by the module developers. n = Major version (see device-specific data sheet)<br>0h = Smallest value<br>Fh = Highest possible value |
| 3-0 | MINREV | R | 0h | Minor revision. This number holds the module revision and is incremented by the module developers. n = Minor module revision (see device-specific data sheet)<br>0h = Smallest value<br>Fh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.18 CLKCTL (Offset = 1100h) [Reset = 00000000h]

CLKCTL is shown in Figure 37-19 and described in Table 37-20.

Return to the Summary Table.

RTC Clock Control Register.
This register can be made read only access by the RTCLOCK register.

**Figure 37-19. CLKCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MODCLKEN | RESERVED | | | | | | |
| R/W-0h | R-0h | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |

**Table 37-20. CLKCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | MODCLKEN | R/W | 0h | This bit enables the supply of the 32kHz clock to the RTC. It will not power-up the 32kHz crystal oscillator this needs to be done in the Clock System Module.<br>0h = 32kHz clock is not supplied to the RTC.<br>1h = 32kHz clock is supplied to the RTC. |
| 30-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.19 DBGCTL (Offset = 1104h) [Reset = 00000000h]

DBGCTL is shown in Figure 37-20 and described in Table 37-21.

Return to the Summary Table.

RTC Module Debug Control Register

**Figure 37-20. DBGCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | DBGINT | DBGRUN |
| R-0h | | | | | | R/W-0h | R/W-0h |

**Table 37-21. DBGCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | |
| 1 | DBGINT | R/W | 0h | Debug Interrupt Enable.<br>0h = Interrupts of the module will not be captured anymore if CPU is in debug state. Which means no update to the RTCRIS, RTCMISC and RTCIIDX register.<br>1h = Interrupts are enabled in debug mode. Interrupt requests are signaled to the interrupt controller. If the flags are required by software (polling mode) the DGBINT bit need to be set to 1. |
| 0 | DBGRUN | R/W | 0h | Debug Run.<br>0h = Counter is halted if CPU is in debug state.<br>1h = Continue to operate normally ignoring the debug state of the CPU. |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.20 CTL (Offset = 1108h) [Reset = 00000000h]

CTL is shown in Figure 37-21 and described in Table 37-22.

Return to the Summary Table.

RTC Control Register

**Figure 37-21. CTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RTCBCD | RESERVED | | | | | RTCTEVTX | |
| R/W-0h | R-0h | | | | | R/W-0h | |

**Table 37-22. CTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7 | RTCBCD | R/W | 0h | Real-time clock BCD select. Selects BCD counting for real-time clock.<br>0h = Binary code selected<br>1h = Binary coded decimal (BCD) code selected |
| 6-2 | RESERVED | R | 0h | |
| 1-0 | RTCTEVTX | R/W | 0h | Real-time clock time event 0x0 = Minute changed 0x1 = Hour changed 0x2 = Every day at midnight (00:00) 0x3 = Every day at noon (12:00)<br>0h = Generate RTC event every minute.<br>1h = Generate RTC event every hour.<br>2h = Generate RTC event at midnight.<br>3h = Generate RTC event at noon. |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.21 STA (Offset = 110Ch) [Reset = 00000000h]

STA is shown in Figure 37-22 and described in Table 37-23.

Return to the Summary Table.

RTC Status Register

**Figure 37-22. STA**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | RTCTCOK | RTCTCRDY | RTCRDY |
| R-0h | | | | | R-0h | R-0h | R-0h |

**Table 37-23. STA Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | RESERVED | R | 0h | |
| 2 | RTCTCOK | R | 0h | Real-time clock temperature compensation write OK. This is a read-only bit that indicates if the write to RTCTCMP is successful or not.<br>0h = Write to RTCTCMPx is unsuccessful<br>1h = Write to RTCTCMPx is successful |
| 1 | RTCTCRDY | R | 0h | Real-time clock temperature compensation ready. This is a read only bit that indicates when the RTCTCMPx can be written. Write to RTCTCMPx should be avoided when RTCTCRDY is reset.<br>0h = RTC temperature compensation in transition<br>1h = RTC temperature compensation ready |
| 0 | RTCRDY | R | 0h | Real-time clock ready. This bit indicates when the real-time clock time values are safe for reading.<br>0h = RTC time values in transition<br>1h = RTC time values safe for reading. |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.22 CAL (Offset = 1110h) [Reset = 00000000h]

CAL is shown in Figure 37-23 and described in Table 37-24.

Return to the Summary Table.

RTC Clock Offset Calibration Register

**Figure 37-23. CAL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | RTCCALFX | |
| R-0h | | | | | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RTCOCALS | RESERVED | | | | | | |
| R/W-0h | R-0h | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RTCOCALX | | | | | | | |
| R/W-0h | | | | | | | |

**Table 37-24. CAL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-18 | RESERVED | R | 0h | |
| 17-16 | RTCCALFX | R/W | 0h | Real-time clock calibration frequency. Selects frequency output to RTCCLK pin for calibration measurement. The corresponding port must be configured for the peripheral module function.<br>0h = 32kHz<br>1h = 512Hz<br>2h = 256Hz<br>3h = 1Hz |
| 15 | RTCOCALS | R/W | 0h | Real-time clock offset error calibration sign. This bit decides the sign of offset error calibration.<br>0h = Down calibration. Frequency adjusted down.<br>1h = Up calibration. Frequency adjusted up. |
| 14-8 | RESERVED | R | 0h | |
| 7-0 | RTCOCALX | R/W | 0h | Real-time clock offset error calibration. Each LSB represents approximately +1ppm (RTCOCALXS = 1) or -1ppm (RTCOCALXS = 0) adjustment in frequency. Maximum effective calibration value is +/-240ppm. Excess values written above +/-240ppm will be ignored by hardware.<br>0h = Smallest value<br>FFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.23 TCMP (Offset = 1114h) [Reset = 00000000h]

TCMP is shown in Figure 37-24 and described in Table 37-25.

Return to the Summary Table.

RTC Temperature Compensation Register

**Figure 37-24. TCMP**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RTCTCMPS | RESERVED | | | | | | |
| R/W-0h | R-0h | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RTCTCMPX | | | | | | | |
| R/W-0h | | | | | | | |

**Table 37-25. TCMP Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15 | RTCTCMPS | R/W | 0h | Real-time clock temperature compensation sign. This bit decides the sign of temperature compensation.<br>0h = Down calibration. Frequency adjusted down.<br>1h = Up calibration. Frequency adjusted up. |
| 14-8 | RESERVED | R | 0h | |
| 7-0 | RTCTCMPX | R/W | 0h | Real-time clock temperature compensation. Value written into this register is used for temperature compensation of RTC. Each LSB represents approximately +1ppm (RTCTCMPS = 1) or -1ppm (RTCTCMPS = 0) adjustment in frequency. Maximum effective calibration value is +/-240ppm. Excess values written above +/-240ppm are ignored by hardware. Reading from RTCTCMP register at any time returns the cumulative value which is the signed addition of RTCOCALx and RTCTCMPX values, and the updated sign bit (RTCTCMPS) of the addition result.<br>00h = Smallest value<br>FFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.24 SEC (Offset = 1118h) [Reset = 0000XXXXh]

SEC is shown in Figure 37-25 and described in Table 37-26.

Return to the Summary Table.

RTC Seconds Register - Calendar Mode With Binary / BCD Format.
This register can be made read only access by the RTCLOCK register.

**Figure 37-25. SEC**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | SECHIGHBCD | | | SECLOWBCD | | | |
| R-0h | R/W-X | | | R/W-X | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | SECBIN | | | | | |
| R-0h | | R/W-X | | | | | |

**Table 37-26. SEC Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-15 | RESERVED | R | 0h | |
| 14-12 | SECHIGHBCD | R/W | X | Seconds BCD – high digit (0 to 5). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>5h = Highest possible value |
| 11-8 | SECLOWBCD | R/W | X | Seconds BCD – low digit (0 to 9). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>9h = Highest possible value |
| 7-6 | RESERVED | R | 0h | |
| 5-0 | SECBIN | R/W | X | Seconds Binary (0 to 59). If RTCBCD=1 write to these bits will be ignored and read give the value 0.<br>00h = Smallest value<br>3Bh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.25 MIN (Offset = 111Ch) [Reset = 0000XXXXh]

MIN is shown in Figure 37-26 and described in Table 37-27.

Return to the Summary Table.

RTC Minutes Register - Calendar Mode With Binary / BCD Format.
This register can be made read only access by the RTCLOCK register.

**Figure 37-26. MIN**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | MINHIGHBCD | | | MINLOWBCD | | | |
| R-0h | R/W-X | | | R/W-X | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | MINBIN | | | | | |
| R-0h | | R/W-X | | | | | |

**Table 37-27. MIN Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-15 | RESERVED | R | 0h | |
| 14-12 | MINHIGHBCD | R/W | X | Minutes BCD – high digit (0 to 5). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>5h = Highest possible value |
| 11-8 | MINLOWBCD | R/W | X | Minutes BCD – low digit (0 to 9). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>9h = Highest possible value |
| 7-6 | RESERVED | R | 0h | |
| 5-0 | MINBIN | R/W | X | Minutes Binary (0 to 59). If RTCBCD=1 write to these bits will be ignored and read give the value 0.<br>00h = Smallest value<br>3Bh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.26 HOUR (Offset = 1120h) [Reset = 0000XXXXh]

HOUR is shown in Figure 37-27 and described in Table 37-28.

Return to the Summary Table.

RTC Hours Register - Calendar Mode With Binary / BCD Format.
This register can be made read only access by the RTCLOCK register.

**Figure 37-27. HOUR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | HOURHIGHBCD | | HOURLOWBCD | | |
| R-0h | | | R/W-X | | R/W-X | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | HOURBIN | | | | |
| R-0h | | | R/W-X | | | | |

**Table 37-28. HOUR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-14 | RESERVED | R | 0h | |
| 13-12 | HOURHIGHBCD | R/W | X | Hours BCD – high digit (0 to 2). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>2h = Highest possible value |
| 11-8 | HOURLOWBCD | R/W | X | Hours BCD – low digit (0 to 9). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>9h = Highest possible value |
| 7-5 | RESERVED | R | 0h | |
| 4-0 | HOURBIN | R/W | X | Hours Binary (0 to 23). If RTCBCD=1 write to these bits will be ignored and read give the value 0.<br>00h = Smallest value<br>17h = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.27 DAY (Offset = 1124h) [Reset = 00XXXX0Xh]

DAY is shown in Figure 37-28 and described in Table 37-29.

Return to the Summary Table.

RTC Day of Week/Month Register - Calendar Mode With Binary / BCD Format.
This register can be made read only access by the RTCLOCK register.

**Figure 37-28. DAY**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | DOMHIGHBCD | | DOMLOWBCD | | | |
| R-0h | | R/W-X | | R/W-X | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | DOMBIN | | | | |
| R-0h | | | R/W-X | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | DOW | | |
| R-0h | | | | | R/W-X | | |

**Table 37-29. DAY Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-22 | RESERVED | R | 0h | |
| 21-20 | DOMHIGHBCD | R/W | X | Day of month BCD – high digit (0 to 3). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>3h = Highest possible value |
| 19-16 | DOMLOWBCD | R/W | X | Day of month BCD – low digit (0 to 9). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>9h = Highest possible value |
| 15-13 | RESERVED | R | 0h | |
| 12-8 | DOMBIN | R/W | X | Day of month Binary (1 to 28, 29, 30, 31). If RTCBCD=1 write to these bits will be ignored and read give the value 0.<br>00h = Smallest value<br>1Fh = Highest possible value |
| 7-3 | RESERVED | R | 0h | |
| 2-0 | DOW | R/W | X | Day of week (0 to 6). These bits are valid if RTCBCD=1 or RTCBCD=0.<br>0h = Smallest value<br>6h = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.28 MON (Offset = 1128h) [Reset = 0000XX0Xh]

MON is shown in Figure 37-29 and described in Table 37-30.

Return to the Summary Table.

RTC Month Register - Calendar Mode With Binary / BCD Format.
This register can be made read only access by the RTCLOCK register.

**Figure 37-29. MON**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | MONHIGHBCD | MONLOWBCD | | | |
| R-0h | | | R/W-X | R/W-X | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | MONBIN | | | |
| R-0h | | | | R/W-X | | | |

**Table 37-30. MON Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-13 | RESERVED | R | 0h | |
| 12 | MONHIGHBCD | R/W | X | Month BCD – high digit (0 or 1). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>1h = Highest possible value |
| 11-8 | MONLOWBCD | R/W | X | Month BCD – low digit (0 to 9). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>9h = Highest possible value |
| 7-4 | RESERVED | R | 0h | |
| 3-0 | MONBIN | R/W | X | Month Binary (1 to 12). If RTCBCD=1 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>Ch = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.29 YEAR (Offset = 112Ch) [Reset = XXXX0XXXh]

YEAR is shown in Figure 37-30 and described in Table 37-31.

Return to the Summary Table.

RTC Year Register - Calendar Mode With Binary / BCD Format.
This register can be made read only access by the RTCLOCK register.

**Figure 37-30. YEAR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | CENTHIGHBCD | | | CENTLOWBCD | | | |
| R-0h | R/W-X | | | R/W-X | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DECADEBCD | | | | YEARLOWESTBCD | | | |
| R/W-X | | | | R/W-X | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | YEARHIGHBIN | | | |
| R-0h | | | | R/W-X | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| YEARLOWBIN | | | | | | | |
| R/W-X | | | | | | | |

**Table 37-31. YEAR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | RESERVED | R | 0h | |
| 30-28 | CENTHIGHBCD | R/W | X | Century BCD – high digit (0 to 4). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>4h = Highest possible value |
| 27-24 | CENTLOWBCD | R/W | X | Century BCD – low digit (0 to 9). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>9h = Highest possible value |
| 23-20 | DECADEBCD | R/W | X | Decade BCD (0 to 9). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>9h = Highest possible value |
| 19-16 | YEARLOWESTBCD | R/W | X | Year BCD – lowest digit (0 to 9). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>9h = Highest possible value |
| 15-12 | RESERVED | R | 0h | |
| 11-8 | YEARHIGHBIN | R/W | X | Year Binary – high byte. Valid values for Year are 0 to 4095. If RTCBCD=1 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>Fh = Highest possible value |
| 7-0 | YEARLOWBIN | R/W | X | Year Binary – low byte. Valid values for Year are 0 to 4095. If RTCBCD=1 write to these bits will be ignored and read give the value 0.<br>00h = Smallest value<br>FFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.30 A1MIN (Offset = 1130h) [Reset = 0000XXXXh]

A1MIN is shown in Figure 37-31 and described in Table 37-32.

Return to the Summary Table.

RTC Minutes Alarm Register - Calendar Mode With Binary / BCD Format

**Figure 37-31. A1MIN**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| AMINAEBCD | AMINHIGHBCD | | | AMINLOWBCD | | | |
| R/W-0h | R/W-X | | | R/W-X | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| AMINAEBIN | RESERVED | AMINBIN | | | | | |
| R/W-0h | R-0h | R/W-X | | | | | |

**Table 37-32. A1MIN Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15 | AMINAEBCD | R/W | 0h | Alarm Minutes BCD enable. If RTCBCD=0 this bit is always 0. Write to this bit will be ignored. 0x0= Alarm disabled. 0x1= Alarm enabled.<br>0h = No alarm<br>1h = Alarm enabled |
| 14-12 | AMINHIGHBCD | R/W | X | Alarm Minutes BCD – high digit (0 to 5). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>5h = Highest possible value |
| 11-8 | AMINLOWBCD | R/W | X | Alarm Minutes BCD – low digit (0 to 9). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>9h = Highest possible value |
| 7 | AMINAEBIN | R/W | 0h | Alarm Minutes Binary enable. If RTCBCD=1 this bit is always 0. Write to this bit will be ignored. 0x0= Alarm disabled. 0x1= Alarm enabled.<br>0h = No alarm<br>1h = Alarm enabled |
| 6 | RESERVED | R | 0h | |
| 5-0 | AMINBIN | R/W | X | Alarm Minutes Binary (0 to 59). If RTCBCD=1 write to these bits will be ignored and read give the value 0.<br>00h = Smallest value<br>3Bh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.31 A1HOUR (Offset = 1134h) [Reset = 0000XX0Xh]

A1HOUR is shown in Figure 37-32 and described in Table 37-33.

Return to the Summary Table.

RTC Hours Alarm Register - Calendar Mode With Binary / BCD Format

**Figure 37-32. A1HOUR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| AHOURAEBCD | RESERVED | AHOURHIGHBCD | | AHOURLOWBCD | | | |
| R/W-0h | R-0h | R/W-X | | R/W-X | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| AHOURAEBIN | RESERVED | | AHOURBIN | | | | |
| R/W-0h | R-0h | | R/W-X | | | | |

**Table 37-33. A1HOUR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15 | AHOURAEBCD | R/W | 0h | Alarm Hours BCD enable. If RTCBCD=0 this bit is always 0. Write to this bit will be ignored. 0x0= Alarm disabled. 0x1= Alarm enabled.<br>0h = No alarm<br>1h = Alarm enabled |
| 14 | RESERVED | R | 0h | |
| 13-12 | AHOURHIGHBCD | R/W | X | Alarm Hours BCD – high digit (0 to 2). If RTCBCD=0 write to these bits will be ignored and read give the value 0..<br>0h = Smallest value<br>2h = Highest possible value |
| 11-8 | AHOURLOWBCD | R/W | X | Alarm Hours BCD – low digit (0 to 9). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>9h = Highest possible value |
| 7 | AHOURAEBIN | R/W | 0h | Alarm Hours Binary enable. If RTCBCD=1 this bit is always 0. Write to this bit will be ignored. 0x0= Alarm disabled. 0x1= Alarm enabled.<br>0h = No alarm<br>1h = Alarm enabled |
| 6-5 | RESERVED | R | 0h | |
| 4-0 | AHOURBIN | R/W | X | Alarm Hours Binary (0 to 23). If RTCBCD=1 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>17h = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.32 A1DAY (Offset = 1138h) [Reset = 00XXXX0Xh]

A1DAY is shown in Figure 37-33 and described in Table 37-34.

Return to the Summary Table.

RTC Alarm Day Of Week / Month Register - Calendar Mode With Binary / BCD Format

**Figure 37-33. A1DAY**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| ADOMAEBCD | RESERVED | ADOMHIGHBCD | | ADOMLOWBCD | | | |
| R/W-0h | R-0h | R/W-X | | R/W-X | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| ADOMAEBIN | RESERVED | | ADOMBIN | | | | |
| R/W-0h | R-0h | | R/W-X | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ADOWAE | RESERVED | | | | ADOW | | |
| R/W-0h | R-0h | | | | R/W-X | | |

**Table 37-34. A1DAY Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | RESERVED | R | 0h | |
| 23 | ADOMAEBCD | R/W | 0h | Alarm Day of month BCD enable. If RTCBCD=0 this bit is always 0. Write to this bit will be ignored. 0x0= Alarm disabled. 0x1= Alarm enabled.<br>0h = No alarm<br>1h = Alarm enabled |
| 22 | RESERVED | R | 0h | |
| 21-20 | ADOMHIGHBCD | R/W | X | Alarm Day of month BCD – high digit (0 to 3). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>3h = Highest possible value |
| 19-16 | ADOMLOWBCD | R/W | X | Alarm Day of month BCD – low digit (0 to 9). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>9h = Highest possible value |
| 15 | ADOMAEBIN | R/W | 0h | Alarm Day of month Binary enable. If RTCBCD=1 this bit is always 0. Write to this bit will be ignored. 0x0= Alarm disabled. 0x1= Alarm enabled.<br>0h = No alarm<br>1h = Alarm enabled |
| 14-13 | RESERVED | R | 0h | |
| 12-8 | ADOMBIN | R/W | X | Alarm Day of month Binary (1 to 28, 29, 30, 31) If RTCBCD=1 write to these bits will be ignored and read give the value 0.<br>00h = Smallest value<br>1Fh = Highest possible value |
| 7 | ADOWAE | R/W | 0h | Alarm Day of week enable. This bit are valid if RTCBCD=1 or RTCBCD=0.<br>0h = No alarm<br>1h = Alarm enabled |
| 6-3 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

**Table 37-34. A1DAY Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2-0 | ADOW | R/W | X | Alarm Day of week (0 to 6). These bits are valid if RTCBCD=1 or RTCBCD=0.<br>0h = Smallest value<br>6h = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.33 A2MIN (Offset = 113Ch) [Reset = 0000XXXXh]

A2MIN is shown in Figure 37-34 and described in Table 37-35.

Return to the Summary Table.

RTC Minutes Alarm Register - Calendar Mode With Binary / BCD Format

**Figure 37-34. A2MIN**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| AMINAEBCD | AMINHIGHBCD | | | AMINLOWBCD | | | |
| R/W-0h | R/W-X | | | R/W-X | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| AMINAEBIN | RESERVED | AMINBIN | | | | | |
| R/W-0h | R-0h | R/W-X | | | | | |

**Table 37-35. A2MIN Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15 | AMINAEBCD | R/W | 0h | Alarm Minutes BCD enable. If RTCBCD=0 this bit is always 0. Write to this bit will be ignored. 0x0= Alarm disabled. 0x1= Alarm enabled.<br>0h = No alarm<br>1h = Alarm enabled |
| 14-12 | AMINHIGHBCD | R/W | X | Alarm Minutes BCD – high digit (0 to 5). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>5h = Highest possible value |
| 11-8 | AMINLOWBCD | R/W | X | Alarm Minutes BCD – low digit (0 to 9). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>9h = Highest possible value |
| 7 | AMINAEBIN | R/W | 0h | Alarm Minutes Binary enable. If RTCBCD=1 this bit is always 0. Write to this bit will be ignored. 0x0= Alarm disabled. 0x1= Alarm enabled.<br>0h = No alarm<br>1h = Alarm enabled |
| 6 | RESERVED | R | 0h | |
| 5-0 | AMINBIN | R/W | X | Alarm Minutes Binary (0 to 59). If RTCBCD=1 write to these bits will be ignored and read give the value 0.<br>00h = Smallest value<br>3Bh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.34 A2HOUR (Offset = 1140h) [Reset = 0000XX0Xh]

A2HOUR is shown in Figure 37-35 and described in Table 37-36.

Return to the Summary Table.

RTC Hours Alarm Register - Calendar Mode With Binary / BCD Format

**Figure 37-35. A2HOUR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| AHOURAEBCD | RESERVED | AHOURHIGHBCD | | AHOURLOWBCD | | | |
| R/W-0h | R-0h | R/W-X | | R/W-X | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| AHOURAEBIN | RESERVED | | AHOURBIN | | | | |
| R/W-0h | R-0h | | R/W-X | | | | |

**Table 37-36. A2HOUR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15 | AHOURAEBCD | R/W | 0h | Alarm Hours BCD enable. If RTCBCD=0 this bit is always 0. Write to this bit will be ignored. 0x0= Alarm disabled. 0x1= Alarm enabled.<br>0h = No alarm<br>1h = Alarm enabled |
| 14 | RESERVED | R | 0h | |
| 13-12 | AHOURHIGHBCD | R/W | X | Alarm Hours BCD – high digit (0 to 2). If RTCBCD=0 write to these bits will be ignored and read give the value 0..<br>0h = Smallest value<br>2h = Highest possible value |
| 11-8 | AHOURLOWBCD | R/W | X | Alarm Hours BCD – low digit (0 to 9). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>9h = Highest possible value |
| 7 | AHOURAEBIN | R/W | 0h | Alarm Hours Binary enable. If RTCBCD=1 this bit is always 0. Write to this bit will be ignored. 0x0= Alarm disabled. 0x1= Alarm enabled.<br>0h = No alarm<br>1h = Alarm enabled |
| 6-5 | RESERVED | R | 0h | |
| 4-0 | AHOURBIN | R/W | X | Alarm Hours Binary (0 to 23). If RTCBCD=1 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>17h = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.35 A2DAY (Offset = 1144h) [Reset = 00XXXX0Xh]

A2DAY is shown in Figure 37-36 and described in Table 37-37.

Return to the Summary Table.

RTC Alarm Day Of Week / Month Register - Calendar Mode With Binary / BCD Format

**Figure 37-36. A2DAY**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| ADOMAEBCD | RESERVED | ADOMHIGHBCD | | ADOMLOWBCD | | | |
| R/W-0h | R-0h | R/W-X | | R/W-X | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| ADOMAEBIN | RESERVED | | ADOMBIN | | | | |
| R/W-0h | R-0h | | R/W-X | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ADOWAE | RESERVED | | | | ADOW | | |
| R/W-0h | R-0h | | | | R/W-X | | |

**Table 37-37. A2DAY Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | RESERVED | R | 0h | |
| 23 | ADOMAEBCD | R/W | 0h | Alarm Day of month BCD enable. If RTCBCD=0 this bit is always 0. Write to this bit will be ignored. 0x0= Alarm disabled. 0x1= Alarm enabled.<br>0h = No alarm<br>1h = Alarm enabled |
| 22 | RESERVED | R | 0h | |
| 21-20 | ADOMHIGHBCD | R/W | X | Alarm Day of month BCD – high digit (0 to 3). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>3h = Highest possible value |
| 19-16 | ADOMLOWBCD | R/W | X | Alarm Day of month BCD – low digit (0 to 9). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>9h = Highest possible value |
| 15 | ADOMAEBIN | R/W | 0h | Alarm Day of month Binary enable. If RTCBCD=1 this bit is always 0. Write to this bit will be ignored. 0x0= Alarm disabled. 0x1= Alarm enabled.<br>0h = No alarm<br>1h = Alarm enabled |
| 14-13 | RESERVED | R | 0h | |
| 12-8 | ADOMBIN | R/W | X | Alarm Day of month Binary (1 to 28, 29, 30, 31) If RTCBCD=1 write to these bits will be ignored and read give the value 0.<br>00h = Smallest value<br>1Fh = Highest possible value |
| 7 | ADOWAE | R/W | 0h | Alarm Day of week enable. This bit are valid if RTCBCD=1 or RTCBCD=0.<br>0h = No alarm<br>1h = Alarm enabled |
| 6-3 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

**Table 37-37. A2DAY Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2-0 | ADOW | R/W | X | Alarm Day of week (0 to 6). These bits are valid if RTCBCD=1 or RTCBCD=0.<br>0h = Smallest value<br>6h = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.36 PSCTL (Offset = 1148h) [Reset = 00000008h]

PSCTL is shown in Figure 37-37 and described in Table 37-38.

Return to the Summary Table.

RTC Prescale Timer Control Register

**Figure 37-37. PSCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | RT1IP | | | RESERVED | |
| R-0h | | | R/W-0h | | | R-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | RT0IP | | | RESERVED | |
| R-0h | | | R/W-2h | | | R-0h | |

**Table 37-38. PSCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-21 | RESERVED | R | 0h | |
| 20-18 | RT1IP | R/W | 0h | Prescale timer 1 interrupt interval<br>0h = Interval every 15.6 millisecond<br>1h = Interval every 31.3 millisecond<br>2h = Interval every 62.5 millisecond<br>3h = Interval every 125 millisecond<br>4h = Interval every 250 millisecond<br>5h = Interval every 500 millisecond<br>6h = Interval every 1 second<br>7h = Interval every 2 second |
| 17-5 | RESERVED | R | 0h | |
| 4-2 | RT0IP | R/W | 2h | Prescale timer 0 interrupt interval<br>2h = Interval every 244 microsecond<br>3h = Interval every 488 microsecond<br>4h = Interval every 0.98 millisecond<br>5h = Interval every 1.95 millisecond<br>6h = Interval every 3.91 millisecond<br>7h = Interval every 7.81 millisecond |
| 1-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.37 EXTPSCTL (Offset = 114Ch) [Reset = 00000000h]

EXTPSCTL is shown in Figure 37-38 and described in Table 37-39.

Return to the Summary Table.

Extended Prescale Timer Control Register

**Figure 37-38. EXTPSCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | RT2PS | | RESERVED | |
| R-0h | | | | | | | | | | | | R/W-0h | | R-0h | |

**Table 37-39. EXTPSCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3-2 | RT2PS | R/W | 0h | Prescale timer 2 interrupt interval<br>0h = Interval every 4 second<br>1h = Interval every 8 second<br>2h = Interval every 16 second |
| 1-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.38 TSSEC (Offset = 1150h) [Reset = 00000000h]

TSSEC is shown in Figure 37-39 and described in Table 37-40.

Return to the Summary Table.

RTC Second Time Stamp Capture - Calendar Mode With Binary / BCD Format

**Figure 37-39. TSSEC**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | SECHIGHBCD | | | SECLOWBCD | | | |
| R-0h | R-0h | | | R-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | SECBIN | | | | | |
| R-0h | | R-0h | | | | | |

**Table 37-40. TSSEC Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-15 | RESERVED | R | 0h | |
| 14-12 | SECHIGHBCD | R | 0h | Time Stamp Seconds BCD – high digit (0 to 5). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>5h = Highest possible value |
| 11-8 | SECLOWBCD | R | 0h | Time Stamp Seconds BCD – low digit (0 to 9). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>9h = Highest possible value |
| 7-6 | RESERVED | R | 0h | |
| 5-0 | SECBIN | R | 0h | Time Stamp Second Binary (0 to 59). If RTCBCD=1 write to these bits will be ignored and read give the value 0.<br>00h = Smallest value<br>3Bh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.39 TSMIN (Offset = 1154h) [Reset = 00000000h]

TSMIN is shown in Figure 37-40 and described in Table 37-41.

Return to the Summary Table.

RTC Minutes Time Stamp Capture - Calendar Mode With Binary / BCD Format

**Figure 37-40. TSMIN**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | MINHIGHBCD | | | MINLOWBCD | | | |
| R-0h | R-0h | | | R-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | MINBIN | | | | | |
| R-0h | | R-0h | | | | | |

**Table 37-41. TSMIN Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-15 | RESERVED | R | 0h | |
| 14-12 | MINHIGHBCD | R | 0h | Time Stamp Minutes BCD – high digit (0 to 5). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>5h = Highest possible value |
| 11-8 | MINLOWBCD | R | 0h | Time Stamp Minutes BCD – low digit (0 to 9). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>9h = Highest possible value |
| 7-6 | RESERVED | R | 0h | |
| 5-0 | MINBIN | R | 0h | Time Stamp Minutes Binary (0 to 59). If RTCBCD=1 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>3Bh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.40 TSHOUR (Offset = 1158h) [Reset = 00000000h]

TSHOUR is shown in Figure 37-41 and described in Table 37-42.

Return to the Summary Table.

RTC Hours Time Stamp Capture - Calendar Mode With Binary / BCD Format

**Figure 37-41. TSHOUR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | HOURHIGHBCD | | HOURLOWBCD | | |
| R-0h | | | R-0h | | R-0h | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | HOURBIN | | | | |
| R-0h | | | R-0h | | | | |

**Table 37-42. TSHOUR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-14 | RESERVED | R | 0h | |
| 13-12 | HOURHIGHBCD | R | 0h | Time Stamp Hours BCD – high digit (0 to 2). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>00h = Smallest value<br>02h = Highest possible value |
| 11-8 | HOURLOWBCD | R | 0h | Time Stamp Hours BCD – low digit (0 to 9). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>00h = Smallest value<br>09h = Highest possible value |
| 7-5 | RESERVED | R | 0h | |
| 4-0 | HOURBIN | R | 0h | Time Stamp Hours Binary (0 to 23). If RTCBCD=1 write to these bits will be ignored and read give the value 0.<br>00h = Smallest value<br>17h = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.41 TSDAY (Offset = 115Ch) [Reset = 00000000h]

TSDAY is shown in Figure 37-42 and described in Table 37-43.

Return to the Summary Table.

RTC Day Of Week / Month Time Stamp Capture - Calendar Mode With Binary / BCD Format

**Figure 37-42. TSDAY**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | DOMHIGHBCD | | DOMLOWBCD | | | |
| R-0h | | R-0h | | R-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | DOMBIN | | | | |
| R-0h | | | R-0h | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | DOW | | |
| R-0h | | | | | R-0h | | |

**Table 37-43. TSDAY Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-22 | RESERVED | R | 0h | |
| 21-20 | DOMHIGHBCD | R | 0h | Time Stamp Day of month BCD – high digit (0 to 3). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>3h = Highest possible value |
| 19-16 | DOMLOWBCD | R | 0h | Time Stamp Day of month BCD – low digit (0 to 9). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>9h = Highest possible value |
| 15-13 | RESERVED | R | 0h | |
| 12-8 | DOMBIN | R | 0h | Time Stamp Day of month Binary (1 to 28, 29, 30, 31) If RTCBCD=1 write to these bits will be ignored and read give the value 0.<br>00h = Smallest value<br>1Fh = Highest possible value |
| 7-3 | RESERVED | R | 0h | |
| 2-0 | DOW | R | 0h | Time Stamp Day of week (0 to 6). These bits are valid if RTCBCD=1 or RTCBCD=0.<br>0h = Smallest value<br>6h = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.42 TSMON (Offset = 1160h) [Reset = 00000000h]

TSMON is shown in Figure 37-43 and described in Table 37-44.

Return to the Summary Table.

RTC Month Time Stamp Capture - Calendar Mode With Binary / BCD Format

**Figure 37-43. TSMON**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | MONHIGHBCD | MONLOWBCD | | | |
| R-0h | | | R-0h | R-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | MONBIN | | | |
| R-0h | | | | R-0h | | | |

**Table 37-44. TSMON Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-13 | RESERVED | R | 0h | |
| 12 | MONHIGHBCD | R | 0h | Time Stamp Month BCD – high digit (0 or 1). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>1h = Highest possible value |
| 11-8 | MONLOWBCD | R | 0h | Time Stamp Month BCD – low digit (0 to 9). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>9h = Highest possible value |
| 7-4 | RESERVED | R | 0h | |
| 3-0 | MONBIN | R | 0h | Time Stamp Month Binary (1 to 12). If RTCBCD=1 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>Ch = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.43 TSYEAR (Offset = 1164h) [Reset = 00000000h]

TSYEAR is shown in Figure 37-44 and described in Table 37-45.

Return to the Summary Table.

RTC Years Time Stamp Capture - Calendar Mode With Binary / BCD Format

**Figure 37-44. TSYEAR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | CENTHIGHBCD | | | CENTLOWBCD | | | |
| R-0h | R-0h | | | R-0h | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DECADEBCD | | | | YERARLOWESTBCD | | | |
| R-0h | | | | R-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | YEARHIGHBIN | | | |
| R-0h | | | | R-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| YEARLOWBIN | | | | | | | |
| R-0h | | | | | | | |

**Table 37-45. TSYEAR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | RESERVED | R | 0h | |
| 30-28 | CENTHIGHBCD | R | 0h | Time Stamp Century BCD – high digit (0 to 4). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>4h = Highest possible value |
| 27-24 | CENTLOWBCD | R | 0h | Time Stamp Century BCD – low digit (0 to 9). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>9h = Highest possible value |
| 23-20 | DECADEBCD | R | 0h | Time Stamp Decade BCD (0 to 9). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>9h = Highest possible value |
| 19-16 | YERARLOWESTBCD | R | 0h | Time Stamp Year BCD – lowest digit (0 to 9). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>9h = Highest possible value |
| 15-12 | RESERVED | R | 0h | |
| 11-8 | YEARHIGHBIN | R | 0h | Time Stamp Year Binary – high byte. Valid values for Year are 0 to 4095. If RTCBCD=1 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>Fh = Highest possible value |
| 7-0 | YEARLOWBIN | R | 0h | Time Stamp Year Binary – low byte. Valid values for Year are 0 to 4095. If RTCBCD=1 write to these bits will be ignored and read give the value 0.<br>00h = Smallest value<br>FFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.44 TSSTAT (Offset = 1168h) [Reset = 00000000h]

TSSTAT is shown in Figure 37-45 and described in Table 37-46.

Return to the Summary Table.

Time Stamp Status

**Figure 37-45. TSSTAT**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | TSVDDEVT |
| R-0h | | | | | | | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TSTIOEVT15 | TSTIOEVT14 | TSTIOEVT13 | TSTIOEVT12 | TSTIOEVT11 | TSTIOEVT10 | TSTIOEVT9 | TSTIOEVT8 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TSTIOEVT7 | TSTIOEVT6 | TSTIOEVT5 | TSTIOEVT4 | TSTIOEVT3 | TSTIOEVT2 | TSTIOEVT1 | TSTIOEVT0 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 37-46. TSSTAT Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-17 | RESERVED | R | 0h | |
| 16 | TSVDDEVT | R | 0h | Loss of VDD caused time stamp event<br>0h = no event detected<br>1h = event detected |
| 15 | TSTIOEVT15 | R | 0h | Tamper I/O 15 caused time stamp event<br>0h = no event detected<br>1h = event detected |
| 14 | TSTIOEVT14 | R | 0h | Tamper I/O 14 caused time stamp event<br>0h = no event detected<br>1h = event detected |
| 13 | TSTIOEVT13 | R | 0h | Tamper I/O 13 caused time stamp event<br>0h = no event detected<br>1h = event detected |
| 12 | TSTIOEVT12 | R | 0h | Tamper I/O 12 caused time stamp event<br>0h = no event detected<br>1h = event detected |
| 11 | TSTIOEVT11 | R | 0h | Tamper I/O 11 caused time stamp event<br>0h = no event detected<br>1h = event detected |
| 10 | TSTIOEVT10 | R | 0h | Tamper I/O 10 caused time stamp event<br>0h = no event detected<br>1h = event detected |
| 9 | TSTIOEVT9 | R | 0h | Tamper I/O 9 caused time stamp event<br>0h = no event detected<br>1h = event detected |
| 8 | TSTIOEVT8 | R | 0h | Tamper I/O 8 caused time stamp event<br>0h = no event detected<br>1h = event detected |
| 7 | TSTIOEVT7 | R | 0h | Tamper I/O 7 caused time stamp event<br>0h = no event detected<br>1h = event detected |



Copyright © 2026 Texas Instruments Incorporated

**Table 37-46. TSSTAT Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 6 | TSTIOEVT6 | R | 0h | Tamper I/O 6 caused time stamp event<br>0h = no event detected<br>1h = event detected |
| 5 | TSTIOEVT5 | R | 0h | Tamper I/O 5 caused time stamp event<br>0h = no event detected<br>1h = event detected |
| 4 | TSTIOEVT4 | R | 0h | Tamper I/O 4 caused time stamp event<br>0h = no event detected<br>1h = event detected |
| 3 | TSTIOEVT3 | R | 0h | Tamper I/O 3 caused time stamp event<br>0h = no event detected<br>1h = event detected |
| 2 | TSTIOEVT2 | R | 0h | Tamper I/O 2 caused time stamp event<br>0h = no event detected<br>1h = event detected |
| 1 | TSTIOEVT1 | R | 0h | Tamper I/O 1 caused time stamp event<br>0h = no event detected<br>1h = event detected |
| 0 | TSTIOEVT0 | R | 0h | Tamper I/O 0 caused time stamp event<br>0h = no event detected<br>1h = event detected |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.45 TSCTL (Offset = 116Ch) [Reset = 00000000h]

TSCTL is shown in Figure 37-46 and described in Table 37-47.

Return to the Summary Table.

time stamp control register

**Figure 37-46. TSCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| R-0/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | TSCAPTURE | RESERVED | | | TSVDDEN |
| R-0h | | | R/WK-0h | R-0h | | | R/WK-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TSTIOEN15 | TSTIOEN14 | TSTIOEN13 | TSTIOEN12 | TSTIOEN11 | TSTIOEN10 | TSTIOEN9 | TSTIOEN8 |
| R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TSTIOEN7 | TSTIOEN6 | TSTIOEN5 | TSTIOEN4 | TSTIOEN3 | TSTIOEN2 | TSTIOEN1 | TSTIOEN0 |
| R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h |

**Table 37-47. TSCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | R-0/W | 0h | need to write (KEY=0xC5) to update this register<br>C5h = This field must be written with 0xC5 to be able to clear any of the enable bits |
| 23-21 | RESERVED | R | 0h | |
| 20 | TSCAPTURE | R/WK | 0h | Defines the capture method of the RTC timestamp when a time stamp event occurs.<br>KEY must be set to C5h to write to this bit.<br>0h = Time stamp holds RTC capture at first occurrence of time stamp event.<br>1h = Time stamp holds RTC capture at last occurrence of time stamp event. |
| 19-17 | RESERVED | R | 0h | |
| 16 | TSVDDEN | R/WK | 0h | Time Stamp by VDD Loss detection enable<br>KEY must be set to C5h to write to this bit.<br>0h = function disabled<br>1h = function enabled |
| 15 | TSTIOEN15 | R/WK | 0h | Time Stamp by Tamper I/O 15 enable<br>KEY must be set to C5h to write to this bit.<br>0h = function disabled<br>1h = function enabled |
| 14 | TSTIOEN14 | R/WK | 0h | Time Stamp by Tamper I/O 14 enable<br>KEY must be set to C5h to write to this bit.<br>0h = function disabled<br>1h = function enabled |
| 13 | TSTIOEN13 | R/WK | 0h | Time Stamp by Tamper I/O 13 enable<br>KEY must be set to C5h to write to this bit.<br>0h = function disabled<br>1h = function enabled |



Copyright © 2026 Texas Instruments Incorporated

**Table 37-47. TSCTL Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 12 | TSTIOEN12 | R/WK | 0h | Time Stamp by Tamper I/O 12 enable<br>KEY must be set to C5h to write to this bit.<br>0h = function disabled<br>1h = function enabled |
| 11 | TSTIOEN11 | R/WK | 0h | Time Stamp by Tamper I/O 11 enable<br>KEY must be set to C5h to write to this bit.<br>0h = function disabled<br>1h = function enabled |
| 10 | TSTIOEN10 | R/WK | 0h | Time Stamp by Tamper I/O 10 enable<br>KEY must be set to C5h to write to this bit.<br>0h = function disabled<br>1h = function enabled |
| 9 | TSTIOEN9 | R/WK | 0h | Time Stamp by Tamper I/O 9 enable<br>KEY must be set to C5h to write to this bit.<br>0h = function disabled<br>1h = function enabled |
| 8 | TSTIOEN8 | R/WK | 0h | Time Stamp by Tamper I/O 8 enable<br>KEY must be set to C5h to write to this bit.<br>0h = function disabled<br>1h = function enabled |
| 7 | TSTIOEN7 | R/WK | 0h | Time Stamp by Tamper I/O 7 enable<br>KEY must be set to C5h to write to this bit.<br>0h = function disabled<br>1h = function enabled |
| 6 | TSTIOEN6 | R/WK | 0h | Time Stamp by Tamper I/O 6 enable<br>KEY must be set to C5h to write to this bit.<br>0h = function disabled<br>1h = function enabled |
| 5 | TSTIOEN5 | R/WK | 0h | Time Stamp by Tamper I/O 5 enable<br>KEY must be set to C5h to write to this bit.<br>0h = function disabled<br>1h = function enabled |
| 4 | TSTIOEN4 | R/WK | 0h | Time Stamp by Tamper I/O 4 enable<br>KEY must be set to C5h to write to this bit.<br>0h = function disabled<br>1h = function enabled |
| 3 | TSTIOEN3 | R/WK | 0h | Time Stamp by Tamper I/O 3 enable<br>KEY must be set to C5h to write to this bit.<br>0h = function disabled<br>1h = function enabled |
| 2 | TSTIOEN2 | R/WK | 0h | Time Stamp by Tamper I/O 2 enable<br>KEY must be set to C5h to write to this bit.<br>0h = function disabled<br>1h = function enabled |
| 1 | TSTIOEN1 | R/WK | 0h | Time Stamp by Tamper I/O 1 enable<br>KEY must be set to C5h to write to this bit.<br>0h = function disabled<br>1h = function enabled |
| 0 | TSTIOEN0 | R/WK | 0h | Time Stamp by Tamper I/O 0 enable<br>KEY must be set to C5h to write to this bit.<br>0h = function disabled<br>1h = function enabled |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.46 TSCLR (Offset = 1170h) [Reset = 00000000h]

TSCLR is shown in Figure 37-47 and described in Table 37-48.

Return to the Summary Table.

time stamp clear register

**Figure 37-47. TSCLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| KEY | | | | | | | | RESERVED | | | | | | | |
| R-0/W-0h | | | | | | | | R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | | | | CLR |
| R-0h | | | | | | | | | | | | | | | WK-0h |

**Table 37-48. TSCLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | R-0/W | 0h | need to write (KEY=0xE2) to update this register<br>E2h = This field must be written with 0xE2 to be able to clear any of the enable bits |
| 23-1 | RESERVED | R | 0h | |
| 0 | CLR | WK | 0h | Clear time stamp and status register.<br>KEY must be set to E2h to write to this bit.<br>0h = Writing 0 has no effect<br>1h = clear time stamp event |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.47 LFSSRST (Offset = 11F0h) [Reset = 00000000h]

LFSSRST is shown in Figure 37-48 and described in Table 37-49.

Return to the Summary Table.

Low frequency subsystem reset request. Asserting the VBATPOR bit in this register will issue a power cycle on the battery backup domain. This reset has the same effect as removing and reconnecting the power supply to the VBAT power pin.
This register can be write protected by the RTCLOCK register.

**Figure 37-48. LFSSRST**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| R-0/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | VBATPOR |
| R-0h | | | | | | | R/WK-0h |

**Table 37-49. LFSSRST Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | R-0/W | 0h | need to write (KEY=0x12) to update this register<br>12h = This field must be written with 0x12 to be able to request the power on reset. |
| 23-1 | RESERVED | R | 0h | |
| 0 | VBATPOR | R/WK | 0h | If set, the register bit will request a power on reset to the PMU of the LFSS.<br>KEY must be set to 12h to write to this bit.<br>0h = Writing this value has no effect.<br>1h = Request power on reset to the LFSS. |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.48 RTCLOCK (Offset = 11FCh) [Reset = 00000000h]

RTCLOCK is shown in Figure 37-49 and described in Table 37-50.

Return to the Summary Table.

The RTC lock bit protects the CLKCTL, SEC, MIN, HOUR, DAY, MON, YEAR and LFSSRST registers from accidental updates.

**Figure 37-49. RTCLOCK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| R-0/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | PROTECT |
| R-0h | | | | | | | R/WK-0h |

**Table 37-50. RTCLOCK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | R-0/W | 0h | need to write (KEY=0x22) to update this register<br>22h = This field must be written with 0x22 to be able to update any of the bits. |
| 23-1 | RESERVED | R | 0h | |
| 0 | PROTECT | R/WK | 0h | If set, the register bit will protect the CLKCTL, SEC, MIN, HOUR, DAY, MON, YEAR and LFSSRST from accidental writes.<br>KEY must be set to 22h to write to this bit.<br>0h = RTC counter is writable.<br>1h = RTC counter is read only access. |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.49 TIOCTL[y] (Offset = 1200h + formula) [Reset = 00000000h]

TIOCTL[y] is shown in Figure 37-50 and described in Table 37-51.

Return to the Summary Table.

tamper I/O control register

Offset = 1200h + (y * 4h); where y = 0h to Fh

**Figure 37-50. TIOCTL[y]**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | OUTINV | INENA | PIPU | PIPD |
| R-0h | | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | FILTEREN | | RESERVED | | POLARITY | |
| R-0h | | R/W-0h | | R-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | TOUTSEL | | RESERVED | | | IOMUX |
| R-0h | | Rmodify/W-0h | | R-0h | | | R/W-0h |

**Table 37-51. TIOCTL[y] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-20 | RESERVED | R | 0h | |
| 19 | OUTINV | R/W | 0h | Output inversion enable<br>0h = The output inversion is disabled.<br>1h = The output inversion is enabled. |
| 18 | INENA | R/W | 0h | input enable<br>0h = The input path is disabled.<br>1h = The input path is enabled. |
| 17 | PIPU | R/W | 0h | pull up enable<br>0h = The pull-up function is disabled.<br>1h = The pull-up function is enabled. |
| 16 | PIPD | R/W | 0h | pull down enable<br>0h = The pull-down function is disabled.<br>1h = The pull-down function is enabled. |
| 15-14 | RESERVED | R | 0h | |
| 13-12 | FILTEREN | R/W | 0h | Programmable counter length of digital glitch filter for TIO0<br>0h = no filter on the tamper I/O beyond CDC synchronization sample<br>1h = 1 FLCLK minimum sample (30us)<br>2h = 3 LFCLK minimum sample (100us)<br>3h = 6 LFCLK minimum sample (200us) |
| 11-10 | RESERVED | R | 0h | |
| 9-8 | POLARITY | R/W | 0h | Enables and configures edge detection polarity for TIO<br>0h = Edge detection disabled<br>1h = Detects rising edge of input event<br>2h = Detects falling edge of input event<br>3h = Detects both rising and falling edge of input event |
| 7-6 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

**Table 37-51. TIOCTL[y] Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 5-4 | TOUTSEL | Rmodify/W | 0h | Selects the source for TOUT control<br>0h = The TOUT register is the source for TOUT<br>1h = The LFCLK is the source for TOUT<br>2h = The heart beat generator is the source for TOUT<br>3h = The time stamp event status is the source for TOUT |
| 3-1 | RESERVED | R | 0h | |
| 0 | IOMUX | R/W | 0h | tamper I/O is controlled by SoC IOMUX module<br>0h = The tamper I/O is controlled by the IOMUX of the SoC and does allow assignment to a peripheral function. In the case the main supply (VDD) is lost, this I/O will go into a Hi-Z state.<br>1h = The tamper I/O is controlled by the TIOCTL register and stays functional during loss of the main supply (VDD). |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.50 TOUT3_0 (Offset = 1280h) [Reset = 00000000h]

TOUT3_0 is shown in Figure 37-51 and described in Table 37-52.

Return to the Summary Table.

Tamper I/O output for pins configured as TIO3 to TIO0.

**Figure 37-51. TOUT3_0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | TIO3 |
| R-0h | | | | | | | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | TIO2 |
| R-0h | | | | | | | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | TIO1 |
| R-0h | | | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | TIO0 |
| R-0h | | | | | | | R/W-0h |

**Table 37-52. TOUT3_0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-25 | RESERVED | R | 0h | |
| 24 | TIO3 | R/W | 0h | This bit sets the value of the pin tamper I/O 3 (TIO3) when the output is enabled through TOE3 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 23-17 | RESERVED | R | 0h | |
| 16 | TIO2 | R/W | 0h | This bit sets the value of the pin tamper I/O 2 (TIO0) when the output is enabled through TOE2 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 15-9 | RESERVED | R | 0h | |
| 8 | TIO1 | R/W | 0h | This bit sets the value of the pin tamper I/O 1 (TIO1) when the output is enabled through TOE1 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 7-1 | RESERVED | R | 0h | |
| 0 | TIO0 | R/W | 0h | This bit sets the value of the pin tamper I/O 0 (TIO0) when the output is enabled through TOE0 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.51 TOUT7_4 (Offset = 1284h) [Reset = 00000000h]

TOUT7_4 is shown in Figure 37-52 and described in Table 37-53.

Return to the Summary Table.

Tamper I/O output for pins configured as TIO7 to TIO4.

**Figure 37-52. TOUT7_4**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | TIO7 |
| R-0h | | | | | | | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | TIO6 |
| R-0h | | | | | | | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | TIO5 |
| R-0h | | | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | TIO4 |
| R-0h | | | | | | | R/W-0h |

**Table 37-53. TOUT7_4 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-25 | RESERVED | R | 0h | |
| 24 | TIO7 | R/W | 0h | This bit sets the value of the pin tamper I/O 7 (TIO7) when the output is enabled through TOE7 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 23-17 | RESERVED | R | 0h | |
| 16 | TIO6 | R/W | 0h | This bit sets the value of the pin tamper I/O 2 (TIO6) when the output is enabled through TOE6 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 15-9 | RESERVED | R | 0h | |
| 8 | TIO5 | R/W | 0h | This bit sets the value of the pin tamper I/O 5 (TIO5) when the output is enabled through TOE5 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 7-1 | RESERVED | R | 0h | |
| 0 | TIO4 | R/W | 0h | This bit sets the value of the pin tamper I/O 4 (TIO4) when the output is enabled through TOE4 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.52 TOUT11_8 (Offset = 1288h) [Reset = 00000000h]

TOUT11_8 is shown in Figure 37-53 and described in Table 37-54.

Return to the Summary Table.

Tamper I/O output for pins configured as TIO11 to TIO8.

**Figure 37-53. TOUT11_8**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | TIO11 |
| R-0h | | | | | | | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | TIO10 |
| R-0h | | | | | | | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | TIO9 |
| R-0h | | | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | TIO8 |
| R-0h | | | | | | | R/W-0h |

**Table 37-54. TOUT11_8 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-25 | RESERVED | R | 0h | |
| 24 | TIO11 | R/W | 0h | This bit sets the value of the pin tamper I/O 11 (TIO11) when the output is enabled through TOE11 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 23-17 | RESERVED | R | 0h | |
| 16 | TIO10 | R/W | 0h | This bit sets the value of the pin tamper I/O 10 (TIO10) when the output is enabled through TOE10 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 15-9 | RESERVED | R | 0h | |
| 8 | TIO9 | R/W | 0h | This bit sets the value of the pin tamper I/O 9 (TIO9) when the output is enabled through TOE9 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 7-1 | RESERVED | R | 0h | |
| 0 | TIO8 | R/W | 0h | This bit sets the value of the pin tamper I/O 8 (TIO8) when the output is enabled through TOE8 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.53 TOUT15_12 (Offset = 128Ch) [Reset = 00000000h]

TOUT15_12 is shown in Figure 37-54 and described in Table 37-55.

Return to the Summary Table.

Tamper I/O output for pins configured as TIO15 to TIO12.

**Figure 37-54. TOUT15_12**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | TIO15 |
| R-0h | | | | | | | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | TIO14 |
| R-0h | | | | | | | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | TIO13 |
| R-0h | | | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | TIO12 |
| R-0h | | | | | | | R/W-0h |

**Table 37-55. TOUT15_12 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-25 | RESERVED | R | 0h | |
| 24 | TIO15 | R/W | 0h | This bit sets the value of the pin tamper I/O 15 (TIO15) when the output is enabled through TOE15 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 23-17 | RESERVED | R | 0h | |
| 16 | TIO14 | R/W | 0h | This bit sets the value of the pin tamper I/O 14 (TIO14) when the output is enabled through TOE14 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 15-9 | RESERVED | R | 0h | |
| 8 | TIO13 | R/W | 0h | This bit sets the value of the pin tamper I/O 13 (TIO13) when the output is enabled through TOE13 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |
| 7-1 | RESERVED | R | 0h | |
| 0 | TIO12 | R/W | 0h | This bit sets the value of the pin tamper I/O 12 (TIO12) when the output is enabled through TOE12 register.<br>0h = Output is set to 0<br>1h = Output is set to 1 |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.54 TOE3_0 (Offset = 1290h) [Reset = 00000000h]

TOE3_0 is shown in Figure 37-55 and described in Table 37-56.

Return to the Summary Table.

Tamper I/O output enable for pins configured as TIO3 to TIO0.

**Figure 37-55. TOE3_0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | TIO3 |
| R-0h | | | | | | | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | TIO2 |
| R-0h | | | | | | | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | TIO1 |
| R-0h | | | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | TIO0 |
| R-0h | | | | | | | R/W-0h |

**Table 37-56. TOE3_0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-25 | RESERVED | R | 0h | |
| 24 | TIO3 | R/W | 0h | Enables data output for tamper I/O 3<br>0h = output disabled<br>1h = output enabled |
| 23-17 | RESERVED | R | 0h | |
| 16 | TIO2 | R/W | 0h | Enables data output for tamper I/O 2<br>0h = output disabled<br>1h = output enabled |
| 15-9 | RESERVED | R | 0h | |
| 8 | TIO1 | R/W | 0h | Enables data output for tamper I/O 1<br>0h = output disabled<br>1h = output enabled |
| 7-1 | RESERVED | R | 0h | |
| 0 | TIO0 | R/W | 0h | Enables data output for tamper I/O 0<br>0h = output disabled<br>1h = output enabled |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.55 TOE7_4 (Offset = 1294h) [Reset = 00000000h]

TOE7_4 is shown in Figure 37-56 and described in Table 37-57.

Return to the Summary Table.

Tamper I/O output enable for pins configured as TIO7 to TIO4.

**Figure 37-56. TOE7_4**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | TIO7 |
| R-0h | | | | | | | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | TIO6 |
| R-0h | | | | | | | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | TIO5 |
| R-0h | | | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | TIO4 |
| R-0h | | | | | | | R/W-0h |

**Table 37-57. TOE7_4 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-25 | RESERVED | R | 0h | |
| 24 | TIO7 | R/W | 0h | Enables data output for tamper I/O 7<br>0h = output disabled<br>1h = output enabled |
| 23-17 | RESERVED | R | 0h | |
| 16 | TIO6 | R/W | 0h | Enables data output for tamper I/O 6<br>0h = output disabled<br>1h = output enabled |
| 15-9 | RESERVED | R | 0h | |
| 8 | TIO5 | R/W | 0h | Enables data output for tamper I/O 5<br>0h = output disabled<br>1h = output enabled |
| 7-1 | RESERVED | R | 0h | |
| 0 | TIO4 | R/W | 0h | Enables data output for tamper I/O 4<br>0h = output disabled<br>1h = output enabled |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.56 TOE11_8 (Offset = 1298h) [Reset = 00000000h]

TOE11_8 is shown in Figure 37-57 and described in Table 37-58.

Return to the Summary Table.

Tamper I/O output enable for pins configured as TIO11 to TIO8.

**Figure 37-57. TOE11_8**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | TIO11 |
| R-0h | | | | | | | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | TIO10 |
| R-0h | | | | | | | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | TIO9 |
| R-0h | | | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | TIO8 |
| R-0h | | | | | | | R/W-0h |

**Table 37-58. TOE11_8 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-25 | RESERVED | R | 0h | |
| 24 | TIO11 | R/W | 0h | Enables data output for tamper I/O 11<br>0h = output disabled<br>1h = output enabled |
| 23-17 | RESERVED | R | 0h | |
| 16 | TIO10 | R/W | 0h | Enables data output for tamper I/O 10<br>0h = output disabled<br>1h = output enabled |
| 15-9 | RESERVED | R | 0h | |
| 8 | TIO9 | R/W | 0h | Enables data output for tamper I/O 9<br>0h = output disabled<br>1h = output enabled |
| 7-1 | RESERVED | R | 0h | |
| 0 | TIO8 | R/W | 0h | Enables data output for tamper I/O 8<br>0h = output disabled<br>1h = output enabled |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.57 TOE15_12 (Offset = 129Ch) [Reset = 00000000h]

TOE15_12 is shown in Figure 37-58 and described in Table 37-59.

Return to the Summary Table.

Tamper I/O output enable for pins configured as TIO15 to TIO12.

**Figure 37-58. TOE15_12**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | TIO15 |
| R-0h | | | | | | | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | TIO14 |
| R-0h | | | | | | | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | TIO13 |
| R-0h | | | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | TIO12 |
| R-0h | | | | | | | R/W-0h |

**Table 37-59. TOE15_12 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-25 | RESERVED | R | 0h | |
| 24 | TIO15 | R/W | 0h | Enables data output for tamper I/O 15<br>0h = output disabled<br>1h = output enabled |
| 23-17 | RESERVED | R | 0h | |
| 16 | TIO14 | R/W | 0h | Enables data output for tamper I/O 14<br>0h = output disabled<br>1h = output enabled |
| 15-9 | RESERVED | R | 0h | |
| 8 | TIO13 | R/W | 0h | Enables data output for tamper I/O 13<br>0h = output disabled<br>1h = output enabled |
| 7-1 | RESERVED | R | 0h | |
| 0 | TIO12 | R/W | 0h | Enables data output for tamper I/O 12<br>0h = output disabled<br>1h = output enabled |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.58 TIN3_0 (Offset = 12A0h) [Reset = 00000000h]

TIN3_0 is shown in Figure 37-59 and described in Table 37-60.

Return to the Summary Table.

Tamper I/O inputs for pins configured as TIO3 to TIO0.

**Figure 37-59. TIN3_0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | TIO3 |
| R-0h | | | | | | | R-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | TIO2 |
| R-0h | | | | | | | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | TIO1 |
| R-0h | | | | | | | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | TIO0 |
| R-0h | | | | | | | R-0h |

**Table 37-60. TIN3_0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-25 | RESERVED | R | 0h | |
| 24 | TIO3 | R | 0h | This bit reads the data input value of tamper I/O 3.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 23-17 | RESERVED | R | 0h | |
| 16 | TIO2 | R | 0h | This bit reads the data input value of tamper I/O 2.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 15-9 | RESERVED | R | 0h | |
| 8 | TIO1 | R | 0h | This bit reads the data input value of tamper I/O 1.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 7-1 | RESERVED | R | 0h | |
| 0 | TIO0 | R | 0h | This bit reads the data input value of tamper I/O 0.<br>0h = Input value is 0<br>1h = Input value is 1 |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.59 TIN7_4 (Offset = 12A4h) [Reset = 00000000h]

TIN7_4 is shown in Figure 37-60 and described in Table 37-61.

Return to the Summary Table.

Tamper I/O inputs for pins configured as TIO7 to TIO4.

**Figure 37-60. TIN7_4**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | TIO7 |
| R-0h | | | | | | | R-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | TIO6 |
| R-0h | | | | | | | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | TIO5 |
| R-0h | | | | | | | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | TIO4 |
| R-0h | | | | | | | R-0h |

**Table 37-61. TIN7_4 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-25 | RESERVED | R | 0h | |
| 24 | TIO7 | R | 0h | This bit reads the data input value of tamper I/O 7.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 23-17 | RESERVED | R | 0h | |
| 16 | TIO6 | R | 0h | This bit reads the data input value of tamper I/O 6.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 15-9 | RESERVED | R | 0h | |
| 8 | TIO5 | R | 0h | This bit reads the data input value of tamper I/O 5.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 7-1 | RESERVED | R | 0h | |
| 0 | TIO4 | R | 0h | This bit reads the data input value of tamper I/O 4.<br>0h = Input value is 0<br>1h = Input value is 1 |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.60 TIN11_8 (Offset = 12A8h) [Reset = 00000000h]

TIN11_8 is shown in Figure 37-61 and described in Table 37-62.

Return to the Summary Table.

Tamper I/O inputs for pins configured as TIO11 to TIO8.

**Figure 37-61. TIN11_8**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | TIO11 |
| R-0h | | | | | | | R-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | TIO10 |
| R-0h | | | | | | | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | TIO9 |
| R-0h | | | | | | | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | TIO8 |
| R-0h | | | | | | | R-0h |

**Table 37-62. TIN11_8 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-25 | RESERVED | R | 0h | |
| 24 | TIO11 | R | 0h | This bit reads the data input value of tamper I/O 11.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 23-17 | RESERVED | R | 0h | |
| 16 | TIO10 | R | 0h | This bit reads the data input value of tamper I/O 10.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 15-9 | RESERVED | R | 0h | |
| 8 | TIO9 | R | 0h | This bit reads the data input value of tamper I/O 9.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 7-1 | RESERVED | R | 0h | |
| 0 | TIO8 | R | 0h | This bit reads the data input value of tamper I/O 8.<br>0h = Input value is 0<br>1h = Input value is 1 |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.61 TIN15_12 (Offset = 12ACh) [Reset = 00000000h]

TIN15_12 is shown in Figure 37-62 and described in Table 37-63.

Return to the Summary Table.

Tamper I/O inputs for pins configured as TIO15 to TIO12.

**Figure 37-62. TIN15_12**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | TIO15 |
| R-0h | | | | | | | R-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | TIO14 |
| R-0h | | | | | | | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | TIO13 |
| R-0h | | | | | | | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | TIO12 |
| R-0h | | | | | | | R-0h |

**Table 37-63. TIN15_12 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-25 | RESERVED | R | 0h | |
| 24 | TIO15 | R | 0h | This bit reads the data input value of tamper I/O 15.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 23-17 | RESERVED | R | 0h | |
| 16 | TIO14 | R | 0h | This bit reads the data input value of tamper I/O 14.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 15-9 | RESERVED | R | 0h | |
| 8 | TIO13 | R | 0h | This bit reads the data input value of tamper I/O 13.<br>0h = Input value is 0<br>1h = Input value is 1 |
| 7-1 | RESERVED | R | 0h | |
| 0 | TIO12 | R | 0h | This bit reads the data input value of tamper I/O 12.<br>0h = Input value is 0<br>1h = Input value is 1 |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.62 HEARTBEAT (Offset = 12C0h) [Reset = 00000000h]

HEARTBEAT is shown in Figure 37-63 and described in Table 37-64.

Return to the Summary Table.

The configuration register for the heart beat generator

**Figure 37-63. HEARTBEAT**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | HBMODE | |
| R-0h | | | | | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | HBWIDTH | | |
| R-0h | | | | | R/W-0h | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | HBINTERVAL | | |
| R-0h | | | | | R/W-0h | | |

**Table 37-64. HEARTBEAT Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-18 | RESERVED | R | 0h | |
| 17-16 | HBMODE | R/W | 0h | Heart beat mode<br>0h = Heart beat disabled<br>1h = Heart beat always enabled<br>2h = Heart beat enabled when time stamp event detected<br>3h = Heart beat when VDD has fail condition |
| 15-11 | RESERVED | R | 0h | |
| 10-8 | HBWIDTH | R/W | 0h | Heart beat interval width<br>0h = Heart beat pulse width 1msec<br>1h = Heart beat pulse width 2msec<br>2h = Heart beat pulse width 4msec<br>3h = Heart beat pulse width 8msec<br>4h = Heart beat pulse width 16msec<br>5h = Heart beat pulse width 32msec<br>6h = Heart beat pulse width 64msec<br>7h = Heart beat pulse width 128msec |
| 7-3 | RESERVED | R | 0h | |
| 2-0 | HBINTERVAL | R/W | 0h | Heart beat interval<br>0h = Heart beat interval 0.125 sec<br>1h = Heart beat interval 0.25 sec<br>2h = Heart beat interval 0.5 sec<br>3h = Heart beat interval 1 sec<br>4h = Heart beat interval 2 sec<br>5h = Heart beat interval 4 sec<br>6h = Heart beat interval 8 sec<br>7h = Heart beat interval 16 sec |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.63 TIOLOCK (Offset = 12FCh) [Reset = 00000000h]

TIOLOCK is shown in Figure 37-64 and described in Table 37-65.

Return to the Summary Table.

The TIO lock bit protects the TIOCTL and HEARTBEAT registers from accidental updates.

**Figure 37-64. TIOLOCK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| R-0/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | PROTECT |
| R-0h | | | | | | | R/WK-0h |

**Table 37-65. TIOLOCK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | R-0/W | 0h | need to write (KEY=0x18) to update this register<br>18h = This field must be written with 0x18 to be able to clear any of the enable bits |
| 23-1 | RESERVED | R | 0h | |
| 0 | PROTECT | R/WK | 0h | If set, the register bit will protect the TIOCTL and HEARTBEAT from accidental writes.<br>KEY must be set to 18h to write to this bit.<br>0h = Tamper I/O control is writable.<br>1h = Tamper I/O control is read only access. |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.64 WDTEN (Offset = 1300h) [Reset = 00000000h]

WDTEN is shown in Figure 37-65 and described in Table 37-66.

Return to the Summary Table.

Watchdog Timer Enable Register.
This register can be made read only access by the WDTLOCK register.

**Figure 37-65. WDTEN**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| R-0/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | ENABLE |
| R-0h | | | | | | | R/WK-0h |

**Table 37-66. WDTEN Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | R-0/W | 0h | KEY to allow write access to this register. Writing to this register with an incorrect key causes a POR level reset. Read as 0.<br>EEh = This field must be written with 0xEE to be update the enable bit. |
| 23-1 | RESERVED | R | 0h | |
| 0 | ENABLE | R/WK | 0h | Enable bit for the WDT.<br>KEY must be set to EEh to write to this bit.<br>0h = Disable WDT<br>1h = Enable WDT |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.65 WDTDBGCTL (Offset = 1304h) [Reset = 00000000h]

WDTDBGCTL is shown in Figure 37-66 and described in Table 37-67.

Return to the Summary Table.

Watchdog Timer Debug Control

**Figure 37-66. WDTDBGCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | FREE |
| R-0h | | | | | | | R/W-0h |

**Table 37-67. WDTDBGCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | FREE | R/W | 0h | Free run control<br>0h = The WDT freezes functionality while the CPU is Halted during debug and resumes when the CPU is active.<br>1h = The WDT ignores the state of the CPU Halted debug state. |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.66 WDTCTL (Offset = 1308h) [Reset = 00000043h]

WDTCTL is shown in Figure 37-67 and described in Table 37-68.

Return to the Summary Table.

Watchdog Timer Control Register.
This register can be made read only access by the WDTLOCK register.

**Figure 37-67. WDTCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| R-0/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | PER | | | RESERVED | CLKDIV | | |
| R-0h | R/WK-4h | | | R-0h | R/WK-3h | | |

**Table 37-68. WDTCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | R-0/W | 0h | KEY to allow write access to this register. Writing to this register with an incorrect key causes a POR level reset. Read as 0.<br>C6h = This field must be written with 0xC6 to be able to write any of the enable bits |
| 23-7 | RESERVED | R | 0h | |
| 6-4 | PER | R/WK | 4h | Timer Period of the WDT. These bits select the total watchdog timer count.<br>KEY must be set to C6h to write to this bit.<br>0h = Total timer count is $2^{25}$<br>1h = Total timer count is $2^{21}$<br>2h = Total timer count is $2^{18}$<br>3h = Total timer count is $2^{15}$<br>4h = Total timer count is $2^{12}$ (default)<br>5h = Total timer count is $2^{10}$<br>6h = Total timer count is $2^{8}$<br>7h = Total timer count is $2^{6}$ |
| 3 | RESERVED | R | 0h | |
| 2-0 | CLKDIV | R/WK | 3h | Module Clock Divider, Divide the clock source by CLKDIV+1. Divider values from /1 to /8 are possible.<br>KEY must be set to C6h to write to this bit.<br>0h = Minimum value<br>7h = Maximum value |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.67 WDTCNTRST (Offset = 130Ch) [Reset = 00000000h]

WDTCNTRST is shown in Figure 37-68 and described in Table 37-69.

Return to the Summary Table.

Watchdog Timer Counter Reset Register

**Figure 37-68. WDTCNTRST**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESTART | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-0/W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 37-69. WDTCNTRST Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | RESTART | R-0/W | 0h | Writing 03A7h to this register restarts the WDT Counter. Writing any other value causes a POR level reset. Read as 0x0h.<br>0h = Minimum value<br>000003A7h = VALUE to restart the WDT counter<br>FFFFFFFFh = Maximum value |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.68 WDTSTAT (Offset = 1310h) [Reset = 00000000h]

WDTSTAT is shown in Figure 37-69 and described in Table 37-70.

Return to the Summary Table.

Watchdog Timer Status Register

**Figure 37-69. WDTSTAT**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | | | | RUN |
| R-0h | | | | | | | | | | | | | | | R-0h |

**Table 37-70. WDTSTAT Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | RUN | R | 0h | Watchdog running status flag.<br>0h = Watchdog counter stopped.<br>1h = Watchdog running. |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.69 WDTLOCK (Offset = 13FCh) [Reset = 00000000h]

WDTLOCK is shown in Figure 37-70 and described in Table 37-71.

Return to the Summary Table.

The WDT lock bit protects the WDTEN and WDTCTL registers from accidental updates.

**Figure 37-70. WDTLOCK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| R-0/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | PROTECT |
| R-0h | | | | | | | R/WK-0h |

**Table 37-71. WDTLOCK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | R-0/W | 0h | need to write (KEY=0xBD) to update this register<br>BDh = This field must be written with 0xBD to be able to clear any of the enable bits |
| 23-1 | RESERVED | R | 0h | |
| 0 | PROTECT | R/WK | 0h | If set, the register bit will protect the WDTEN and WDTCTL from accidental writes.<br>KEY must be set to BDh to write to this bit.<br>0h = Watchdog timer control is writable.<br>1h = Watchdog timer control is read only access. |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.70 SPMEM[y] (Offset = 1400h + formula) [Reset = 00000000h]

SPMEM[y] is shown in Figure 37-71 and described in Table 37-72.

Return to the Summary Table.

Scratch pad memory

Offset = 1400h + (y * 4h); where y = 0h to 1Fh

**Figure 37-71. SPMEM[y]**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA3 | | | | | | | | DATA2 | | | | | | | | DATA1 | | | | | | | | DATA0 | | | | | | | |
| R/W-0h | | | | | | | | R/W-0h | | | | | | | | R/W-0h | | | | | | | | R/W-0h | | | | | | | |

**Table 37-72. SPMEM[y] Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | DATA3 | R/W | 0h | memory data byte 3<br>0h = Smallest value<br>FFh = Highest possible value |
| 23-16 | DATA2 | R/W | 0h | memory data byte 2<br>0h = Smallest value<br>FFh = Highest possible value |
| 15-8 | DATA1 | R/W | 0h | memory data byte 1<br>0h = Smallest value<br>FFh = Highest possible value |
| 7-0 | DATA0 | R/W | 0h | memory data byte 0<br>0h = Smallest value<br>FFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.71 SPMWPROT0 (Offset = 1500h) [Reset = 00000000h]

SPMWPROT0 is shown in Figure 37-72 and described in Table 37-73.

Return to the Summary Table.

Scratch pad memory write protect 0

**Figure 37-72. SPMWPROT0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| R-0/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| WP_3_3 | WP_3_2 | WP_3_1 | WP_3_0 | WP_2_3 | WP_2_2 | WP_2_1 | WP_2_0 |
| R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| WP_1_3 | WP_1_2 | WP_1_1 | WP_1_0 | WP_0_3 | WP_0_2 | WP_0_1 | WP_0_0 |
| R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h |

**Table 37-73. SPMWPROT0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | R-0/W | 0h | need to write (KEY=0xE8) to update this register<br>E8h = This field must be written with 0xE8 to be update the write protect bits. |
| 23-16 | RESERVED | R | 0h | |
| 15 | WP_3_3 | R/WK | 0h | write protect SPMEM3 - DATA3<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 14 | WP_3_2 | R/WK | 0h | write protect SPMEM3 - DATA2<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 13 | WP_3_1 | R/WK | 0h | write protect SPMEM3 - DATA1<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 12 | WP_3_0 | R/WK | 0h | write protect SPMEM3 - DATA0<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 11 | WP_2_3 | R/WK | 0h | write protect SPMEM2 - DATA3<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 10 | WP_2_2 | R/WK | 0h | write protect SPMEM2 - DATA2<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |



Copyright © 2026 Texas Instruments Incorporated

**Table 37-73. SPMWPROT0 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 9 | WP_2_1 | R/WK | 0h | write protect SPMEM2 - DATA1<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 8 | WP_2_0 | R/WK | 0h | write protect SPMEM2 - DATA0<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 7 | WP_1_3 | R/WK | 0h | write protect SPMEM1 - DATA3<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 6 | WP_1_2 | R/WK | 0h | write protect SPMEM1 - DATA2<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 5 | WP_1_1 | R/WK | 0h | write protect SPMEM1 - DATA1<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 4 | WP_1_0 | R/WK | 0h | write protect SPMEM1 - DATA0<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 3 | WP_0_3 | R/WK | 0h | write protect SPMEM0 - DATA3<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 2 | WP_0_2 | R/WK | 0h | write protect SPMEM0 - DATA2<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 1 | WP_0_1 | R/WK | 0h | write protect SPMEM0 - DATA1<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 0 | WP_0_0 | R/WK | 0h | write protect SPMEM0 - DATA0<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.72 SPMWPROT1 (Offset = 1504h) [Reset = 00000000h]

SPMWPROT1 is shown in Figure 37-73 and described in Table 37-74.

Return to the Summary Table.

Scratch pad memory write protect 1

**Figure 37-73. SPMWPROT1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| R-0/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| WP_7_3 | WP_7_2 | WP_7_1 | WP_7_0 | WP_6_3 | WP_6_2 | WP_6_1 | WP_6_0 |
| R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| WP_5_3 | WP_5_2 | WP_5_1 | WP_5_0 | WP_4_3 | WP_4_2 | WP_4_1 | WP_4_0 |
| R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h |

**Table 37-74. SPMWPROT1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | R-0/W | 0h | need to write (KEY=0xE8) to update this register<br>E8h = This field must be written with 0xE8 to be update the write protect bits. |
| 23-16 | RESERVED | R | 0h | |
| 15 | WP_7_3 | R/WK | 0h | write protect SPMEM7 - DATA3<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 14 | WP_7_2 | R/WK | 0h | write protect SPMEM7 - DATA2<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 13 | WP_7_1 | R/WK | 0h | write protect SPMEM7 - DATA1<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 12 | WP_7_0 | R/WK | 0h | write protect SPMEM7 - DATA0<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 11 | WP_6_3 | R/WK | 0h | write protect SPMEM6 - DATA3<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 10 | WP_6_2 | R/WK | 0h | write protect SPMEM6 - DATA2<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |



Copyright © 2026 Texas Instruments Incorporated

**Table 37-74. SPMWPROT1 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 9 | WP_6_1 | R/WK | 0h | write protect SPMEM6 - DATA1<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 8 | WP_6_0 | R/WK | 0h | write protect SPMEM6 - DATA0<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 7 | WP_5_3 | R/WK | 0h | write protect SPMEM5 - DATA3<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 6 | WP_5_2 | R/WK | 0h | write protect SPMEM5 - DATA2<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 5 | WP_5_1 | R/WK | 0h | write protect SPMEM5 - DATA1<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 4 | WP_5_0 | R/WK | 0h | write protect SPMEM5 - DATA0<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 3 | WP_4_3 | R/WK | 0h | write protect SPMEM4 - DATA3<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 2 | WP_4_2 | R/WK | 0h | write protect SPMEM4 - DATA2<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 1 | WP_4_1 | R/WK | 0h | write protect SPMEM4 - DATA1<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 0 | WP_4_0 | R/WK | 0h | write protect SPMEM4 - DATA0<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.73 SPMWPROT2 (Offset = 1508h) [Reset = 00000000h]

SPMWPROT2 is shown in Figure 37-74 and described in Table 37-75.

Return to the Summary Table.

Scratch pad memory write protect 2

**Figure 37-74. SPMWPROT2**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| R-0/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| WP_11_3 | WP_11_2 | WP_11_1 | WP_11_0 | WP_10_3 | WP_10_2 | WP_10_1 | WP_10_0 |
| R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| WP_9_3 | WP_9_2 | WP_9_1 | WP_9_0 | WP_8_3 | WP_8_2 | WP_8_1 | WP_8_0 |
| R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h |

**Table 37-75. SPMWPROT2 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | R-0/W | 0h | need to write (KEY=0xE8) to update this register<br>E8h = This field must be written with 0xE8 to be update the write protect bits. |
| 23-16 | RESERVED | R | 0h | |
| 15 | WP_11_3 | R/WK | 0h | write protect SPMEM11 - DATA3<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 14 | WP_11_2 | R/WK | 0h | write protect SPMEM11 - DATA2<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 13 | WP_11_1 | R/WK | 0h | write protect SPMEM11 - DATA1<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 12 | WP_11_0 | R/WK | 0h | write protect SPMEM11 - DATA0<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 11 | WP_10_3 | R/WK | 0h | write protect SPMEM10 - DATA3<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 10 | WP_10_2 | R/WK | 0h | write protect SPMEM610- DATA2<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |



Copyright © 2026 Texas Instruments Incorporated

**Table 37-75. SPMWPROT2 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 9 | WP_10_1 | R/WK | 0h | write protect SPMEM10 - DATA1<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 8 | WP_10_0 | R/WK | 0h | write protect SPMEM10 - DATA0<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 7 | WP_9_3 | R/WK | 0h | write protect SPMEM9 - DATA3<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 6 | WP_9_2 | R/WK | 0h | write protect SPMEM9 - DATA2<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 5 | WP_9_1 | R/WK | 0h | write protect SPMEM9 - DATA1<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 4 | WP_9_0 | R/WK | 0h | write protect SPMEM9 - DATA0<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 3 | WP_8_3 | R/WK | 0h | write protect SPMEM8 - DATA3<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 2 | WP_8_2 | R/WK | 0h | write protect SPMEM8 - DATA2<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 1 | WP_8_1 | R/WK | 0h | write protect SPMEM8 - DATA1<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 0 | WP_8_0 | R/WK | 0h | write protect SPMEM8 - DATA0<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.74 SPMWPROT3 (Offset = 150Ch) [Reset = 00000000h]

SPMWPROT3 is shown in Figure 37-75 and described in Table 37-76.

Return to the Summary Table.

Scratch pad memory write protect 3

**Figure 37-75. SPMWPROT3**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| R-0/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| WP_15_3 | WP_15_2 | WP_15_1 | WP_15_0 | WP_14_3 | WP_14_2 | WP_14_1 | WP_14_0 |
| R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| WP_13_3 | WP_13_2 | WP_13_1 | WP_13_0 | WP_12_3 | WP_12_2 | WP_12_1 | WP_12_0 |
| R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h |

**Table 37-76. SPMWPROT3 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | R-0/W | 0h | need to write (KEY=0xE8) to update this register<br>E8h = This field must be written with 0xE8 to be update the write protect bits. |
| 23-16 | RESERVED | R | 0h | |
| 15 | WP_15_3 | R/WK | 0h | write protect SPMEM15 - DATA3<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 14 | WP_15_2 | R/WK | 0h | write protect SPMEM15 - DATA2<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 13 | WP_15_1 | R/WK | 0h | write protect SPMEM15 - DATA1<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 12 | WP_15_0 | R/WK | 0h | write protect SPMEM15 - DATA0<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 11 | WP_14_3 | R/WK | 0h | write protect SPMEM14 - DATA3<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 10 | WP_14_2 | R/WK | 0h | write protect SPMEM14- DATA2<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |



Copyright © 2026 Texas Instruments Incorporated

**Table 37-76. SPMWPROT3 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 9 | WP_14_1 | R/WK | 0h | write protect SPMEM14 - DATA1<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 8 | WP_14_0 | R/WK | 0h | write protect SPMEM14 - DATA0<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 7 | WP_13_3 | R/WK | 0h | write protect SPMEM13 - DATA3<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 6 | WP_13_2 | R/WK | 0h | write protect SPMEM13 - DATA2<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 5 | WP_13_1 | R/WK | 0h | write protect SPMEM13 - DATA1<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 4 | WP_13_0 | R/WK | 0h | write protect SPMEM13 - DATA0<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 3 | WP_12_3 | R/WK | 0h | write protect SPMEM12 - DATA3<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 2 | WP_12_2 | R/WK | 0h | write protect SPMEM12 - DATA2<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 1 | WP_12_1 | R/WK | 0h | write protect SPMEM12 - DATA1<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 0 | WP_12_0 | R/WK | 0h | write protect SPMEM12 - DATA0<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.75 SPMWPROT4 (Offset = 1510h) [Reset = 00000000h]

SPMWPROT4 is shown in Figure 37-76 and described in Table 37-77.

Return to the Summary Table.

Scratch pad memory write protect 4

**Figure 37-76. SPMWPROT4**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| R-0/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| WP_19_3 | WP_19_2 | WP_19_1 | WP_19_0 | WP_18_3 | WP_18_2 | WP_18_1 | WP_18_0 |
| R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| WP_17_3 | WP_17_2 | WP_17_1 | WP_17_0 | WP_16_3 | WP_16_2 | WP_16_1 | WP_16_0 |
| R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h |

**Table 37-77. SPMWPROT4 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | R-0/W | 0h | need to write (KEY=0xE8) to update this register<br>E8h = This field must be written with 0xE8 to be update the write protect bits. |
| 23-16 | RESERVED | R | 0h | |
| 15 | WP_19_3 | R/WK | 0h | write protect SPMEM19 - DATA3<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 14 | WP_19_2 | R/WK | 0h | write protect SPMEM19 - DATA2<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 13 | WP_19_1 | R/WK | 0h | write protect SPMEM19 - DATA1<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 12 | WP_19_0 | R/WK | 0h | write protect SPMEM19 - DATA0<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 11 | WP_18_3 | R/WK | 0h | write protect SPMEM18 - DATA3<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 10 | WP_18_2 | R/WK | 0h | write protect SPMEM18- DATA2<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |



Copyright © 2026 Texas Instruments Incorporated

**Table 37-77. SPMWPROT4 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 9 | WP_18_1 | R/WK | 0h | write protect SPMEM18 - DATA1<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 8 | WP_18_0 | R/WK | 0h | write protect SPMEM18 - DATA0<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 7 | WP_17_3 | R/WK | 0h | write protect SPMEM17 - DATA3<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 6 | WP_17_2 | R/WK | 0h | write protect SPMEM17 - DATA2<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 5 | WP_17_1 | R/WK | 0h | write protect SPMEM17 - DATA1<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 4 | WP_17_0 | R/WK | 0h | write protect SPMEM17 - DATA0<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 3 | WP_16_3 | R/WK | 0h | write protect SPMEM16 - DATA3<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 2 | WP_16_2 | R/WK | 0h | write protect SPMEM16 - DATA2<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 1 | WP_16_1 | R/WK | 0h | write protect SPMEM16 - DATA1<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 0 | WP_16_0 | R/WK | 0h | write protect SPMEM16 - DATA0<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.76 SPMWPROT5 (Offset = 1514h) [Reset = 00000000h]

SPMWPROT5 is shown in Figure 37-77 and described in Table 37-78.

Return to the Summary Table.

Scratch pad memory write protect 5

**Figure 37-77. SPMWPROT5**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| R-0/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| WP_23_3 | WP_23_2 | WP_23_1 | WP_23_0 | WP_22_3 | WP_22_2 | WP_22_1 | WP_22_0 |
| R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| WP_21_3 | WP_21_2 | WP_21_1 | WP_21_0 | WP_20_3 | WP_20_2 | WP_20_1 | WP_20_0 |
| R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h |

**Table 37-78. SPMWPROT5 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | R-0/W | 0h | need to write (KEY=0xE8) to update this register<br>E8h = This field must be written with 0xE8 to be update the write protect bits. |
| 23-16 | RESERVED | R | 0h | |
| 15 | WP_23_3 | R/WK | 0h | write protect SPMEM23 - DATA3<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 14 | WP_23_2 | R/WK | 0h | write protect SPMEM23 - DATA2<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 13 | WP_23_1 | R/WK | 0h | write protect SPMEM23 - DATA1<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 12 | WP_23_0 | R/WK | 0h | write protect SPMEM23 - DATA0<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 11 | WP_22_3 | R/WK | 0h | write protect SPMEM22 - DATA3<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 10 | WP_22_2 | R/WK | 0h | write protect SPMEM22- DATA2<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |



Copyright © 2026 Texas Instruments Incorporated

**Table 37-78. SPMWPROT5 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 9 | WP_22_1 | R/WK | 0h | write protect SPMEM22 - DATA1<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 8 | WP_22_0 | R/WK | 0h | write protect SPMEM22 - DATA0<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 7 | WP_21_3 | R/WK | 0h | write protect SPMEM21 - DATA3<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 6 | WP_21_2 | R/WK | 0h | write protect SPMEM21 - DATA2<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 5 | WP_21_1 | R/WK | 0h | write protect SPMEM21 - DATA1<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 4 | WP_21_0 | R/WK | 0h | write protect SPMEM21 - DATA0<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 3 | WP_20_3 | R/WK | 0h | write protect SPMEM20 - DATA3<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 2 | WP_20_2 | R/WK | 0h | write protect SPMEM20 - DATA2<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 1 | WP_20_1 | R/WK | 0h | write protect SPMEM20 - DATA1<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 0 | WP_20_0 | R/WK | 0h | write protect SPMEM20 - DATA0<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.77 SPMWPROT6 (Offset = 1518h) [Reset = 00000000h]

SPMWPROT6 is shown in Figure 37-78 and described in Table 37-79.

Return to the Summary Table.

Scratch pad memory write protect 6

**Figure 37-78. SPMWPROT6**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| R-0/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| WP_27_3 | WP_27_2 | WP_27_1 | WP_27_0 | WP_26_3 | WP_26_2 | WP_26_1 | WP_26_0 |
| R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| WP_25_3 | WP_25_2 | WP_25_1 | WP_25_0 | WP_24_3 | WP_24_2 | WP_24_1 | WP_24_0 |
| R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h |

**Table 37-79. SPMWPROT6 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | R-0/W | 0h | need to write (KEY=0xE8) to update this register<br>E8h = This field must be written with 0xE8 to be update the write protect bits. |
| 23-16 | RESERVED | R | 0h | |
| 15 | WP_27_3 | R/WK | 0h | write protect SPMEM27 - DATA3<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 14 | WP_27_2 | R/WK | 0h | write protect SPMEM27 - DATA2<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 13 | WP_27_1 | R/WK | 0h | write protect SPMEM27 - DATA1<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 12 | WP_27_0 | R/WK | 0h | write protect SPMEM27 - DATA0<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 11 | WP_26_3 | R/WK | 0h | write protect SPMEM26 - DATA3<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 10 | WP_26_2 | R/WK | 0h | write protect SPMEM26- DATA2<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |



Copyright © 2026 Texas Instruments Incorporated

**Table 37-79. SPMWPROT6 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 9 | WP_26_1 | R/WK | 0h | write protect SPMEM26 - DATA1<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 8 | WP_26_0 | R/WK | 0h | write protect SPMEM26 - DATA0<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 7 | WP_25_3 | R/WK | 0h | write protect SPMEM25 - DATA3<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 6 | WP_25_2 | R/WK | 0h | write protect SPMEM25 - DATA2<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 5 | WP_25_1 | R/WK | 0h | write protect SPMEM25 - DATA1<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 4 | WP_25_0 | R/WK | 0h | write protect SPMEM25 - DATA0<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 3 | WP_24_3 | R/WK | 0h | write protect SPMEM24 - DATA3<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 2 | WP_24_2 | R/WK | 0h | write protect SPMEM24 - DATA2<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 1 | WP_24_1 | R/WK | 0h | write protect SPMEM24 - DATA1<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 0 | WP_24_0 | R/WK | 0h | write protect SPMEM24 - DATA0<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.78 SPMWPROT7 (Offset = 151Ch) [Reset = 00000000h]

SPMWPROT7 is shown in Figure 37-79 and described in Table 37-80.

Return to the Summary Table.

Scratch pad memory write protect 7

**Figure 37-79. SPMWPROT7**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| R-0/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| WP_31_3 | WP_31_2 | WP_31_1 | WP_31_0 | WP_30_3 | WP_30_2 | WP_30_1 | WP_30_0 |
| R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| WP_29_3 | WP_29_2 | WP_29_1 | WP_29_0 | WP_28_3 | WP_28_2 | WP_28_1 | WP_28_0 |
| R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h |

**Table 37-80. SPMWPROT7 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | R-0/W | 0h | need to write (KEY=0xE8) to update this register<br>E8h = This field must be written with 0xE8 to be update the write protect bits. |
| 23-16 | RESERVED | R | 0h | |
| 15 | WP_31_3 | R/WK | 0h | write protect SPMEM31 - DATA3<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 14 | WP_31_2 | R/WK | 0h | write protect SPMEM31 - DATA2<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 13 | WP_31_1 | R/WK | 0h | write protect SPMEM31 - DATA1<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 12 | WP_31_0 | R/WK | 0h | write protect SPMEM31 - DATA0<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 11 | WP_30_3 | R/WK | 0h | write protect SPMEM30 - DATA3<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 10 | WP_30_2 | R/WK | 0h | write protect SPMEM30- DATA2<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |



Copyright © 2026 Texas Instruments Incorporated

**Table 37-80. SPMWPROT7 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 9 | WP_30_1 | R/WK | 0h | write protect SPMEM30 - DATA1<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 8 | WP_30_0 | R/WK | 0h | write protect SPMEM30 - DATA0<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 7 | WP_29_3 | R/WK | 0h | write protect SPMEM29 - DATA3<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 6 | WP_29_2 | R/WK | 0h | write protect SPMEM29 - DATA2<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 5 | WP_29_1 | R/WK | 0h | write protect SPMEM29 - DATA1<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 4 | WP_29_0 | R/WK | 0h | write protect SPMEM29 - DATA0<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 3 | WP_28_3 | R/WK | 0h | write protect SPMEM28 - DATA3<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 2 | WP_28_2 | R/WK | 0h | write protect SPMEM28 - DATA2<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 1 | WP_28_1 | R/WK | 0h | write protect SPMEM28 - DATA1<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |
| 0 | WP_28_0 | R/WK | 0h | write protect SPMEM28 - DATA0<br>KEY must be set to E8h to write to this bit.<br>0h = SPMEM is read and write access<br>1h = SPMEM is read only access |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.79 SPMTERASE0 (Offset = 1540h) [Reset = 00000000h]

SPMTERASE0 is shown in Figure 37-80 and described in Table 37-81.

Return to the Summary Table.

Scratch pad memory tamper erase enable 0

**Figure 37-80. SPMTERASE0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| R-0/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TE_3_3 | TE_3_2 | TE_3_1 | TE_3_0 | TE_2_3 | TE_2_2 | TE_2_1 | TE_2_0 |
| R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TE_1_3 | TE_1_2 | TE_1_1 | TE_1_0 | TE_0_3 | TE_0_2 | TE_0_1 | TE_0_0 |
| R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h |

**Table 37-81. SPMTERASE0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | R-0/W | 0h | need to write (KEY=0xA3) to update this register<br>A3h = This field must be written with 0xA3 to be update the tamper erase enable bit. |
| 23-16 | RESERVED | R | 0h | |
| 15 | TE_3_3 | R/WK | 0h | tamper erase enable SPMEM3 - DATA3<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 14 | TE_3_2 | R/WK | 0h | tamper erase enable SPMEM3 - DATA2<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 13 | TE_3_1 | R/WK | 0h | tamper erase enable SPMEM3 - DATA1<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 12 | TE_3_0 | R/WK | 0h | tamper erase enable SPMEM3 - DATA0<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 11 | TE_2_3 | R/WK | 0h | tamper erase enable SPMEM2 - DATA3<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 10 | TE_2_2 | R/WK | 0h | tamper erase enable SPMEM2 - DATA2<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |



Copyright © 2026 Texas Instruments Incorporated

**Table 37-81. SPMTERASE0 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 9 | TE_2_1 | R/WK | 0h | tamper erase enable SPMEM2 - DATA1<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 8 | TE_2_0 | R/WK | 0h | tamper erase enable SPMEM2 - DATA0<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 7 | TE_1_3 | R/WK | 0h | tamper erase enable SPMEM1 - DATA3<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 6 | TE_1_2 | R/WK | 0h | tamper erase enable SPMEM1 - DATA2<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 5 | TE_1_1 | R/WK | 0h | tamper erase enable SPMEM1 - DATA1<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 4 | TE_1_0 | R/WK | 0h | tamper erase enable SPMEM1 - DATA0<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 3 | TE_0_3 | R/WK | 0h | tamper erase enable SPMEM0 - DATA3<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 2 | TE_0_2 | R/WK | 0h | tamper erase enable SPMEM0 - DATA2<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 1 | TE_0_1 | R/WK | 0h | tamper erase enable SPMEM0 - DATA1<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 0 | TE_0_0 | R/WK | 0h | tamper erase enable SPMEM0 - DATA0<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.80 SPMTERASE1 (Offset = 1544h) [Reset = 00000000h]

SPMTERASE1 is shown in Figure 37-81 and described in Table 37-82.

Return to the Summary Table.

Scratch pad memory tamper erase enable 1

**Figure 37-81. SPMTERASE1**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| R-0/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TE_7_3 | TE_7_2 | TE_7_1 | TE_7_0 | TE_6_3 | TE_6_2 | TE_6_1 | TE_6_0 |
| R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TE_5_3 | TE_5_2 | TE_5_1 | TE_5_0 | TE_4_3 | TE_4_2 | TE_4_1 | TE_4_0 |
| R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h |

**Table 37-82. SPMTERASE1 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | R-0/W | 0h | need to write (KEY=0xA3) to update this register<br>A3h = This field must be written with 0xA3 to be update the tamper erase enable bit. |
| 23-16 | RESERVED | R | 0h | |
| 15 | TE_7_3 | R/WK | 0h | tamper erase enable SPMEM7 - DATA3<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 14 | TE_7_2 | R/WK | 0h | tamper erase enable SPMEM7 - DATA2<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 13 | TE_7_1 | R/WK | 0h | tamper erase enable SPMEM7 - DATA1<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 12 | TE_7_0 | R/WK | 0h | tamper erase enable SPMEM7 - DATA0<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 11 | TE_6_3 | R/WK | 0h | tamper erase enable SPMEM6 - DATA3<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 10 | TE_6_2 | R/WK | 0h | tamper erase enable SPMEM6 - DATA2<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |



Copyright © 2026 Texas Instruments Incorporated

**Table 37-82. SPMTERASE1 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 9 | TE_6_1 | R/WK | 0h | tamper erase enable SPMEM6 - DATA1<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 8 | TE_6_0 | R/WK | 0h | tamper erase enable SPMEM6 - DATA0<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 7 | TE_5_3 | R/WK | 0h | tamper erase enable SPMEM5 - DATA3<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 6 | TE_5_2 | R/WK | 0h | tamper erase enable SPMEM5 - DATA2<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 5 | TE_5_1 | R/WK | 0h | tamper erase enable SPMEM5 - DATA1<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 4 | TE_5_0 | R/WK | 0h | tamper erase enable SPMEM5 - DATA0<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 3 | TE_4_3 | R/WK | 0h | tamper erase enable SPMEM4 - DATA3<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 2 | TE_4_2 | R/WK | 0h | tamper erase enable SPMEM4 - DATA2<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 1 | TE_4_1 | R/WK | 0h | tamper erase enable SPMEM4 - DATA1<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 0 | TE_4_0 | R/WK | 0h | tamper erase enable SPMEM4 - DATA0<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.81 SPMTERASE2 (Offset = 1548h) [Reset = 00000000h]

SPMTERASE2 is shown in Figure 37-82 and described in Table 37-83.

Return to the Summary Table.

Scratch pad memory tamper erase enable 2

**Figure 37-82. SPMTERASE2**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| R-0/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TE_11_3 | TE_11_2 | TE_11_1 | TE_11_0 | TE_10_3 | TE_10_2 | TE_10_1 | TE_10_0 |
| R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TE_9_3 | TE_9_2 | TE_9_1 | TE_9_0 | TE_8_3 | TE_8_2 | TE_8_1 | TE_8_0 |
| R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h |

**Table 37-83. SPMTERASE2 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | R-0/W | 0h | need to write (KEY=0xA3) to update this register<br>A3h = This field must be written with 0xA3 to be update the tamper erase enable bit. |
| 23-16 | RESERVED | R | 0h | |
| 15 | TE_11_3 | R/WK | 0h | tamper erase enable SPMEM11 - DATA3<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 14 | TE_11_2 | R/WK | 0h | tamper erase enable SPMEM11 - DATA2<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 13 | TE_11_1 | R/WK | 0h | tamper erase enable SPMEM11 - DATA1<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 12 | TE_11_0 | R/WK | 0h | tamper erase enable SPMEM11 - DATA0<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 11 | TE_10_3 | R/WK | 0h | tamper erase enable SPMEM10 - DATA3<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 10 | TE_10_2 | R/WK | 0h | tamper erase enable SPMEM10 - DATA2<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |



Copyright © 2026 Texas Instruments Incorporated

**Table 37-83. SPMTERASE2 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 9 | TE_10_1 | R/WK | 0h | tamper erase enable SPMEM10 - DATA1<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 8 | TE_10_0 | R/WK | 0h | tamper erase enable SPMEM10 - DATA0<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 7 | TE_9_3 | R/WK | 0h | tamper erase enable SPMEM9 - DATA3<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 6 | TE_9_2 | R/WK | 0h | tamper erase enable SPMEM9 - DATA2<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 5 | TE_9_1 | R/WK | 0h | tamper erase enable SPMEM9 - DATA1<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 4 | TE_9_0 | R/WK | 0h | tamper erase enable SPMEM9 - DATA0<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 3 | TE_8_3 | R/WK | 0h | tamper erase enable SPMEM8 - DATA3<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 2 | TE_8_2 | R/WK | 0h | tamper erase enable SPMEM8 - DATA2<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 1 | TE_8_1 | R/WK | 0h | tamper erase enable SPMEM8 - DATA1<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 0 | TE_8_0 | R/WK | 0h | tamper erase enable SPMEM8 - DATA0<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.82 SPMTERASE3 (Offset = 154Ch) [Reset = 00000000h]

SPMTERASE3 is shown in Figure 37-83 and described in Table 37-84.

Return to the Summary Table.

Scratch pad memory tamper erase enable 3

**Figure 37-83. SPMTERASE3**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| R-0/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TE_15_3 | TE_15_2 | TE_15_1 | TE_15_0 | TE_14_3 | TE_14_2 | TE_14_1 | TE_14_0 |
| R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TE_13_3 | TE_13_2 | TE_13_1 | TE_13_0 | TE_12_3 | TE_12_2 | TE_12_1 | TE_12_0 |
| R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h |

**Table 37-84. SPMTERASE3 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | R-0/W | 0h | need to write (KEY=0xA3) to update this register<br>A3h = This field must be written with 0xA3 to be update the tamper erase enable bit. |
| 23-16 | RESERVED | R | 0h | |
| 15 | TE_15_3 | R/WK | 0h | tamper erase enable SPMEM15 - DATA3<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 14 | TE_15_2 | R/WK | 0h | tamper erase enable SPMEM15 - DATA2<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 13 | TE_15_1 | R/WK | 0h | tamper erase enable SPMEM15 - DATA1<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 12 | TE_15_0 | R/WK | 0h | tamper erase enable SPMEM15 - DATA0<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 11 | TE_14_3 | R/WK | 0h | tamper erase enable SPMEM14 - DATA3<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 10 | TE_14_2 | R/WK | 0h | tamper erase enable SPMEM14 - DATA2<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |



Copyright © 2026 Texas Instruments Incorporated

**Table 37-84. SPMTERASE3 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 9 | TE_14_1 | R/WK | 0h | tamper erase enable SPMEM14 - DATA1<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 8 | TE_14_0 | R/WK | 0h | tamper erase enable SPMEM14 - DATA0<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 7 | TE_13_3 | R/WK | 0h | tamper erase enable SPMEM13 - DATA3<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 6 | TE_13_2 | R/WK | 0h | tamper erase enable SPMEM13 - DATA2<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 5 | TE_13_1 | R/WK | 0h | tamper erase enable SPMEM13 - DATA1<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 4 | TE_13_0 | R/WK | 0h | tamper erase enable SPMEM13 - DATA0<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 3 | TE_12_3 | R/WK | 0h | tamper erase enable SPMEM12 - DATA3<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 2 | TE_12_2 | R/WK | 0h | tamper erase enable SPMEM12 - DATA2<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 1 | TE_12_1 | R/WK | 0h | tamper erase enable SPMEM12 - DATA1<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 0 | TE_12_0 | R/WK | 0h | tamper erase enable SPMEM12 - DATA0<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.83 SPMTERASE4 (Offset = 1550h) [Reset = 00000000h]

SPMTERASE4 is shown in Figure 37-84 and described in Table 37-85.

Return to the Summary Table.

Scratch pad memory tamper erase enable 4

**Figure 37-84. SPMTERASE4**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| R-0/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TE_19_3 | TE_19_2 | TE_19_1 | TE_19_0 | TE_18_3 | TE_18_2 | TE_18_1 | TE_18_0 |
| R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TE_17_3 | TE_17_2 | TE_17_1 | TE_17_0 | TE_16_3 | TE_16_2 | TE_16_1 | TE_16_0 |
| R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h |

**Table 37-85. SPMTERASE4 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | R-0/W | 0h | need to write (KEY=0xA3) to update this register<br>A3h = This field must be written with 0xA3 to be update the tamper erase enable bit. |
| 23-16 | RESERVED | R | 0h | |
| 15 | TE_19_3 | R/WK | 0h | tamper erase enable SPMEM19 - DATA3<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 14 | TE_19_2 | R/WK | 0h | tamper erase enable SPMEM19 - DATA2<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 13 | TE_19_1 | R/WK | 0h | tamper erase enable SPMEM19 - DATA1<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 12 | TE_19_0 | R/WK | 0h | tamper erase enable SPMEM19 - DATA0<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 11 | TE_18_3 | R/WK | 0h | tamper erase enable SPMEM18 - DATA3<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 10 | TE_18_2 | R/WK | 0h | tamper erase enable SPMEM18 - DATA2<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |



Copyright © 2026 Texas Instruments Incorporated

**Table 37-85. SPMTERASE4 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 9 | TE_18_1 | R/WK | 0h | tamper erase enable SPMEM18 - DATA1<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 8 | TE_18_0 | R/WK | 0h | tamper erase enable SPMEM18 - DATA0<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 7 | TE_17_3 | R/WK | 0h | tamper erase enable SPMEM17 - DATA3<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 6 | TE_17_2 | R/WK | 0h | tamper erase enable SPMEM17 - DATA2<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 5 | TE_17_1 | R/WK | 0h | tamper erase enable SPMEM17 - DATA1<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 4 | TE_17_0 | R/WK | 0h | tamper erase enable SPMEM17 - DATA0<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 3 | TE_16_3 | R/WK | 0h | tamper erase enable SPMEM16 - DATA3<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 2 | TE_16_2 | R/WK | 0h | tamper erase enable SPMEM16 - DATA2<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 1 | TE_16_1 | R/WK | 0h | tamper erase enable SPMEM16 - DATA1<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 0 | TE_16_0 | R/WK | 0h | tamper erase enable SPMEM16 - DATA0<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.84 SPMTERASE5 (Offset = 1554h) [Reset = 00000000h]

SPMTERASE5 is shown in Figure 37-85 and described in Table 37-86.

Return to the Summary Table.

Scratch pad memory tamper erase enable 5

**Figure 37-85. SPMTERASE5**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| R-0/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TE_23_3 | TE_23_2 | TE_23_1 | TE_23_0 | TE_22_3 | TE_22_2 | TE_22_1 | TE_22_0 |
| R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TE_21_3 | TE_21_2 | TE_21_1 | TE_21_0 | TE_20_3 | TE_20_2 | TE_20_1 | TE_20_0 |
| R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h |

**Table 37-86. SPMTERASE5 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | R-0/W | 0h | need to write (KEY=0xA3) to update this register<br>A3h = This field must be written with 0xA3 to be update the tamper erase enable bit. |
| 23-16 | RESERVED | R | 0h | |
| 15 | TE_23_3 | R/WK | 0h | tamper erase enable SPMEM23 - DATA3<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 14 | TE_23_2 | R/WK | 0h | tamper erase enable SPMEM23 - DATA2<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 13 | TE_23_1 | R/WK | 0h | tamper erase enable SPMEM23 - DATA1<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 12 | TE_23_0 | R/WK | 0h | tamper erase enable SPMEM23 - DATA0<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 11 | TE_22_3 | R/WK | 0h | tamper erase enable SPMEM22 - DATA3<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 10 | TE_22_2 | R/WK | 0h | tamper erase enable SPMEM22 - DATA2<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |



Copyright © 2026 Texas Instruments Incorporated

**Table 37-86. SPMTERASE5 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 9 | TE_22_1 | R/WK | 0h | tamper erase enable SPMEM22 - DATA1<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 8 | TE_22_0 | R/WK | 0h | tamper erase enable SPMEM22 - DATA0<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 7 | TE_21_3 | R/WK | 0h | tamper erase enable SPMEM21 - DATA3<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 6 | TE_21_2 | R/WK | 0h | tamper erase enable SPMEM21 - DATA2<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 5 | TE_21_1 | R/WK | 0h | tamper erase enable SPMEM21 - DATA1<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 4 | TE_21_0 | R/WK | 0h | tamper erase enable SPMEM21 - DATA0<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 3 | TE_20_3 | R/WK | 0h | tamper erase enable SPMEM20 - DATA3<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 2 | TE_20_2 | R/WK | 0h | tamper erase enable SPMEM20 - DATA2<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 1 | TE_20_1 | R/WK | 0h | tamper erase enable SPMEM20 - DATA1<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 0 | TE_20_0 | R/WK | 0h | tamper erase enable SPMEM20 - DATA0<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.85 SPMTERASE6 (Offset = 1558h) [Reset = 00000000h]

SPMTERASE6 is shown in Figure 37-86 and described in Table 37-87.

Return to the Summary Table.

Scratch pad memory tamper erase enable 6

**Figure 37-86. SPMTERASE6**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| R-0/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TE_27_3 | TE_27_2 | TE_27_1 | TE_27_0 | TE_26_3 | TE_26_2 | TE_26_1 | TE_26_0 |
| R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TE_25_3 | TE_25_2 | TE_25_1 | TE_25_0 | TE_24_3 | TE_24_2 | TE_24_1 | TE_24_0 |
| R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h |

**Table 37-87. SPMTERASE6 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | R-0/W | 0h | need to write (KEY=0xA3) to update this register<br>A3h = This field must be written with 0xA3 to be update the tamper erase enable bit. |
| 23-16 | RESERVED | R | 0h | |
| 15 | TE_27_3 | R/WK | 0h | tamper erase enable SPMEM27 - DATA3<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 14 | TE_27_2 | R/WK | 0h | tamper erase enable SPMEM27 - DATA2<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 13 | TE_27_1 | R/WK | 0h | tamper erase enable SPMEM27 - DATA1<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 12 | TE_27_0 | R/WK | 0h | tamper erase enable SPMEM27 - DATA0<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 11 | TE_26_3 | R/WK | 0h | tamper erase enable SPMEM26 - DATA3<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 10 | TE_26_2 | R/WK | 0h | tamper erase enable SPMEM26 - DATA2<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |



Copyright © 2026 Texas Instruments Incorporated

**Table 37-87. SPMTERASE6 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 9 | TE_26_1 | R/WK | 0h | tamper erase enable SPMEM26 - DATA1<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 8 | TE_26_0 | R/WK | 0h | tamper erase enable SPMEM26 - DATA0<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 7 | TE_25_3 | R/WK | 0h | tamper erase enable SPMEM25 - DATA3<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 6 | TE_25_2 | R/WK | 0h | tamper erase enable SPMEM25 - DATA2<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 5 | TE_25_1 | R/WK | 0h | tamper erase enable SPMEM25 - DATA1<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 4 | TE_25_0 | R/WK | 0h | tamper erase enable SPMEM25 - DATA0<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 3 | TE_24_3 | R/WK | 0h | tamper erase enable SPMEM24 - DATA3<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 2 | TE_24_2 | R/WK | 0h | tamper erase enable SPMEM24 - DATA2<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 1 | TE_24_1 | R/WK | 0h | tamper erase enable SPMEM24 - DATA1<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 0 | TE_24_0 | R/WK | 0h | tamper erase enable SPMEM24 - DATA0<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |



Copyright © 2026 Texas Instruments Incorporated

### 37.7.86 SPMTERASE7 (Offset = 155Ch) [Reset = 00000000h]

SPMTERASE7 is shown in Figure 37-87 and described in Table 37-88.

Return to the Summary Table.

Scratch pad memory tamper erase enable 7

**Figure 37-87. SPMTERASE7**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| R-0/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TE_31_3 | TE_31_2 | TE_31_1 | TE_31_0 | TE_30_3 | TE_30_2 | TE_30_1 | TE_30_0 |
| R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TE_29_3 | TE_29_2 | TE_29_1 | TE_29_0 | TE_28_3 | TE_28_2 | TE_28_1 | TE_28_0 |
| R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h |

**Table 37-88. SPMTERASE7 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | R-0/W | 0h | need to write (KEY=0xA3) to update this register<br>A3h = This field must be written with 0xA3 to be update the tamper erase enable bit. |
| 23-16 | RESERVED | R | 0h | |
| 15 | TE_31_3 | R/WK | 0h | tamper erase enable SPMEM31 - DATA3<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 14 | TE_31_2 | R/WK | 0h | tamper erase enable SPMEM31 - DATA2<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 13 | TE_31_1 | R/WK | 0h | tamper erase enable SPMEM31 - DATA1<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 12 | TE_31_0 | R/WK | 0h | tamper erase enable SPMEM31 - DATA0<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 11 | TE_30_3 | R/WK | 0h | tamper erase enable SPMEM30 - DATA3<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 10 | TE_30_2 | R/WK | 0h | tamper erase enable SPMEM30 - DATA2<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |



Copyright © 2026 Texas Instruments Incorporated

**Table 37-88. SPMTERASE7 Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 9 | TE_30_1 | R/WK | 0h | tamper erase enable SPMEM30 - DATA1<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 8 | TE_30_0 | R/WK | 0h | tamper erase enable SPMEM30 - DATA0<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 7 | TE_29_3 | R/WK | 0h | tamper erase enable SPMEM29 - DATA3<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 6 | TE_29_2 | R/WK | 0h | tamper erase enable SPMEM29 - DATA2<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 5 | TE_29_1 | R/WK | 0h | tamper erase enable SPMEM29 - DATA1<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 4 | TE_29_0 | R/WK | 0h | tamper erase enable SPMEM29 - DATA0<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 3 | TE_28_3 | R/WK | 0h | tamper erase enable SPMEM28 - DATA3<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 2 | TE_28_2 | R/WK | 0h | tamper erase enable SPMEM28 - DATA2<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 1 | TE_28_1 | R/WK | 0h | tamper erase enable SPMEM28 - DATA1<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |
| 0 | TE_28_0 | R/WK | 0h | tamper erase enable SPMEM28 - DATA0<br>KEY must be set to A3h to write to this bit.<br>0h = SPMEM is unmodified during tamper event<br>1h = SPMEM will be erased with tamper event |



Copyright © 2026 Texas Instruments Incorporated

Chapter 38

# RTC

The real-time clock (RTC) module provides clock counters with calendar mode, a flexible programmable alarm, offset calibration, and a provision for temperature compensation.


**38.1 Overview** ..... 2424
**38.2 Basic Operation** ..... 2425
**38.3 Configuration** ..... 2427
**38.4 RTC Registers** ..... 2438





Copyright © 2026 Texas Instruments Incorporated

## 38.1 Overview

The real-time clock module provides time-keeping for the application, with counters for seconds, minutes, hours, day of the week, day of the month, and year, in selectable binary or binary-coded decimal (BCD) format. A variety of programmable interrupt alarms are provided. There are RTC instances (**RTC**, **RTC_x**) that provide common key features, and additional instances (**RTC_x**) that provide specific key features.

Common key features of across all instances of the real-time clock (**RTC**, **RTC_x**) include:

- Real-time clock and calendar mode providing seconds, minutes, hours, day of week, day of month, and year
- Selectable binary or binary-coded decimal (BCD) format
- Leap-year correction (valid for year 1901 through 2099)
- Two customizable calendar alarm interrupts based on minute, hour, day of the week, and day of the month
- Interval alarm interrupt to wake every minute, every hour, at midnight, or at noon
- Periodic interrupt to wake at 4096, 2048, 1024, 512, 256, or 128Hz
- Periodic interrupt to wake at 64, 32, 16, 8, 4, 2, 1, and 0.5Hz
- Interrupt capability down to STANDBY mode with STOPCLKSTBY
- Calibration for crystal offset error and crystal temperature drift (up to ±240 ppm total)
- RTC clock output to pin for calibration

**RTC_x** instances include features such as:

- Three bit prescaler for heartbeat function with interrupt generation
- RTC external clock selection of untrimmed 32kHz, trimmed 512Hz, 256Hz or 1Hz
- RTC time stamp capture upon detection of a timer stamp event, including tamper (TIO) event and VDD fail event
- RTC counter lock function

### *38.1.1 RTC Instances*

Table 38-1 shows a comparison table of RTC instances and supported features.

**Table 38-1. RTC Instances and Key Features**

| RTC Feature | RTC | RTC_A | RTC_B |
|---|---|---|---|
| Power enable register | Yes | – | – |
| Real-time clock and calendar mode providing seconds, minutes, hours, day of week, day of month, and year | Yes | Yes | Yes |
| Selectable binary or binary-coded decimal (BCD) format | Yes | Yes | Yes |
| Leap-year correction (valid for year 1901 through 2099) | Yes | Yes | Yes |
| Two customizable calendar alarm interrupts based on minute, hour, day of the week, and day of the month | Yes | Yes | Yes |
| Interval alarm interrupt to wake every minute, every hour, at midnight, or at noon | Yes | Yes | Yes |
| Periodic interrupt to wake at 4096, 2048, 1024, 512, 256, or 128Hz | Yes | Yes | Yes |
| Periodic interrupt to wake at 64, 32, 16, 8, 4, 2, 1, and 0.5Hz | Yes | Yes | Yes |
| Periodic interrupt to wake at 16, 8, 4 seconds | – | – | Yes |
| Interrupt capability down to STANDBY mode with STOPCLKSTBY | Yes | Yes | Yes |
| Calibration for crystal offset error and crystal temperature drift (up to ±240 ppm total) | Yes | Yes | Yes |
| RTC clock output to pin for calibration (GPIO) | Yes | Yes | Yes |
| RTC clock output to pin for calibration (TIO) | – | Yes | – |
| Three bit prescaler for heartbeat function with interrupt generation | – | Yes | – |
| RTC external clock selection of untrimmed 32kHz, trimmed 512Hz, 256Hz, or 1Hz | – | Yes | – |



Copyright © 2026 Texas Instruments Incorporated

**Table 38-1. RTC Instances and Key Features (continued)**

| RTC Feature | RTC | RTC_A | RTC_B |
|---|---|---|---|
| RTC time stamp capture upon detection of a timer stamp event, including:<br>• TIO event<br>• VDD fail event | – | Yes | – |
| RTC counter lock function | – | Yes | – |

**Note**

To see which RTC instances and features are supported in a device, please see the device-specific data sheet.

## 38.2 Basic Operation

The RTC utilizes two prescaler blocks (RT0PS, RT1PS) to generate a 128Hz and 1Hz clock, respectively, from the source 32.768kHz RTCCLK. The prescaler blocks each support generating a periodic interrupt alarm at a variety of rates from 4096Hz down to 0.5Hz. The 1Hz clock output from RT1PS ticks the counter block, which tracks seconds, minutes, hours, and day of the week. The counter block also sources an interval alarm interrupt once per minute, per hour, at midnight, or at noon. The counter midnight output sources the calendar block, which provides day of the month, month, and year, with leap year correction. Two independent calendar alarm interrupts are provided to wake the application at specific times.



Copyright © 2026 Texas Instruments Incorporated

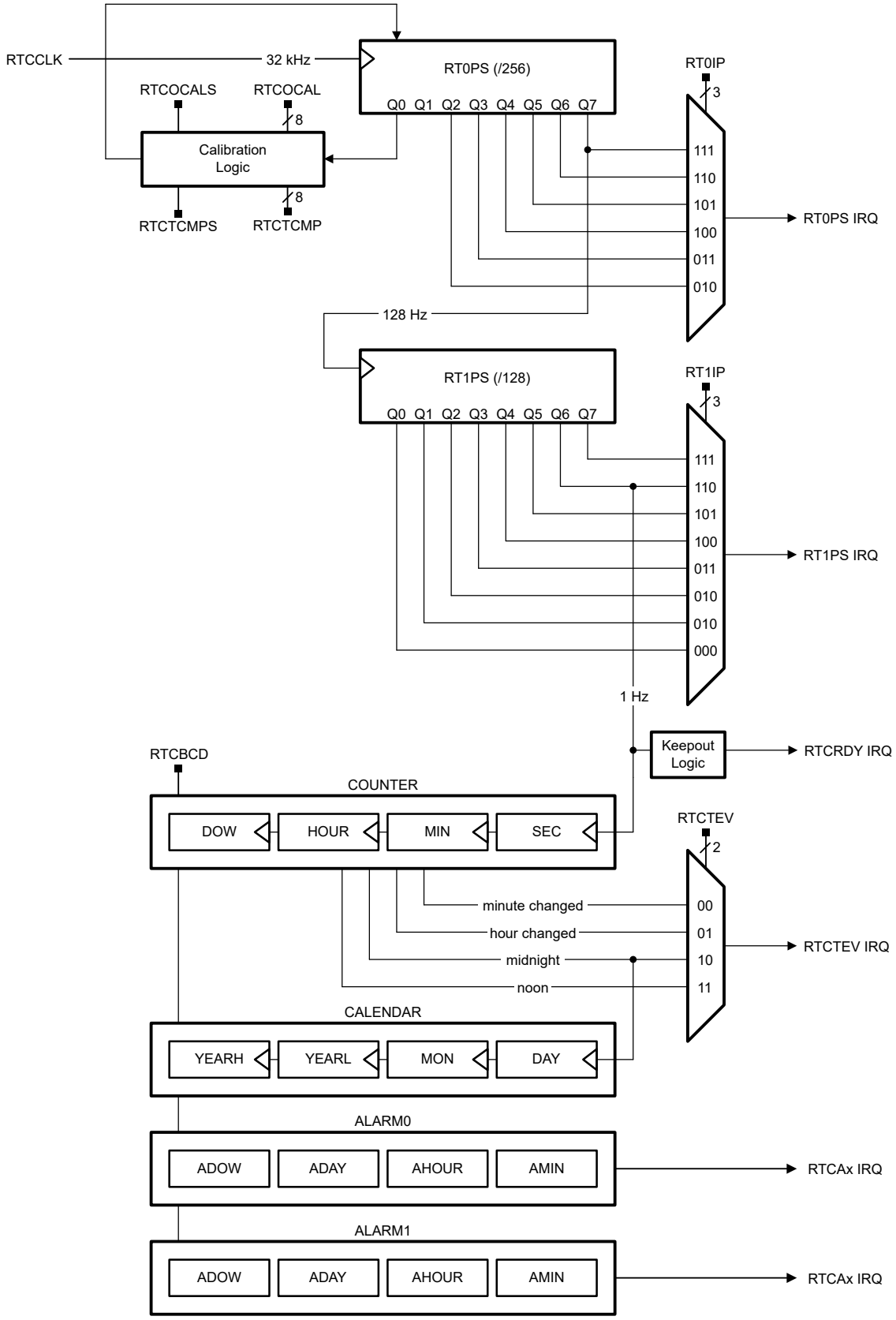



**Figure 38-1. RTC Block Diagram**



Copyright © 2026 Texas Instruments Incorporated

## 38.3 Configuration

The RTC is configured with the RTC peripheral registers.

The RTC state can be reset at any time by asserting the VBATPOR bit in the LFSSRST register.

Application software may reset the RTC state at any time by setting the RESETASSERT bit in the RSTCTL register of the RTC together with the matching RSTCTL KEY value. The RESETSTKY bit in the STAT register of the RTC will be set following a reset of the RTC. This flag may be cleared by setting the RESETSTKYCLR bit in the RSTCTL register of the RTC. As such, software may determine if the RTC has been reset since the sticky reset bit (RESETSTKY) was last cleared. Many RTC registers have no initial conditions. These registers must be configured by application software before use.

To start the RTC counters, the MODCLKEN bit in the CLKCTL register must be set by application software. The MODCLKEN bit is initially cleared following a reset of the RTC.

When enabled and configured, the RTC runs in all power modes except SHUTDOWN. The RTC is not reset by a CPURST, SYSRST, or NRST pin triggered BOOTRST (see the reset control section for a detailed overview of device reset levels). As such, the RTC continues to operate even through software invocation of the bootstrap loader (BSL) or a hold of <1s on the external NRST pin.

### *38.3.1 Clocking*

The RTC is clocked directly from RTCCLK, which is sourced either from the internal LFOSC or the external low frequency crystal oscillator (LFXT). The LFOSC and LFXT both require startup time. Ensure that the oscillator sourcing LFCLK in the application is configured and that LFCLK is operational before enabling power to the RTC module.

Two prescale dividers, RT0PS and RT1PS, are automatically configured to provide a 1-second clock interval for the RTC. The LFCLK must be running at 32kHz clock frequency for proper RTC operation. The RT0PS divider is sourced directly from LFCLK at 32kHz. The output of the RT0PS divider block is 32kHz/256, or 128Hz, and is used to source the input of the RT1PS divider block. The RT1PS output is 128Hz/128, or 1Hz, and is used to source the real-time clock counter registers with the required 1-second time interval.

Clearing the MODCLKEN bit in the CLKCTL register of the RTC halts the propagation of LFCLK to the real-time counters and the prescale counters (RT0PS and RT1PS) in the RTC. It does not have any effect on the LFCLK itself.

### *38.3.2 Reading and Writing to RTC Peripheral Registers*

The peripheral bus clock (ULPCLK) is asynchronous to the RTC clock source (RTCCLK). As a result, special care must be taken when reading and writing certain RTC peripheral registers.

#### Counter/Calendar Registers

The RTC counter/calendar registers are updated once per second. To prevent reading any counter/calendar register at the time of an update (which could result in an invalid time being read), a keep-out window is provided. The keep-out window is approximately 128/32768 seconds before the counters update. The read-only RTCRDY bit in the STA register is reset during the keep-out window, and set outside the keep-out window period. Any read of the counter/calendar registers while RTCRDY bit in the STA register is reset can potentially be invalid, and the time read should be ignored.

The RTCRDY interrupt mechanism can also be used to safely read the RTC counter/calendar registers. When the RTCRDY interrupt is enabled, an interrupt is generated based on the rising edge of the RTCRDY bit, causing the RTCRDY interrupt flag to be set. At this point, the application has nearly a complete second to safely read any or all of the RTC registers. To ensure a safe read, the RTCRDY bit in the STA register can be read again in the interrupt service routine and other interrupts can be disabled. This synchronization process prevents reading the time value during a counter/calendar transition. The RTCRDY interrupt flag is reset automatically when the interrupt is serviced, or can be reset by software.



Copyright © 2026 Texas Instruments Incorporated

The RTC counter/calendar registers can be written to at any time. Writes to the calendar registers take 2 to 3 RTCCLK cycles to take effect. If a back-to-back write is done on counter/calendar registers, then it is possible that for 2-3 RTCCLK cycles, the register is set to an undefined value. Therefore, back-to-back writes to calendar registers need to be avoided. Note that due to the synchronization, if a read immediately follows a write to the calendar registers, the read back value is always the actual counter value, which can be different than the written value due to the 2-3 RTCCLK cycles required to synchronize the newly written value.

The following registers are subject to the restrictions above: SEC, MIN, HOUR, DAY, MON, YEAR.

**Control Register**

The RTC control register (CTL) can be read at any time. Writes to the CTL register should only be done when the TEV (interval timer) interrupt is disabled and after the RTCRDY interrupt is set.

The following register is subject to the restrictions above: CTL.

**Alarm Registers**

The RTC alarm configuration registers can be read at any time. Writes to the alarm registers must be done when the corresponding alarm interrupt is disabled, and after the RTCRDY interrupt is set.

The following registers are subject to the restrictions above: A1MIN, A1HOUR, A1DAY, A2MIN, A2HOUR, A2DAY.

**Offset Correction and Temperature Compensation Registers**

The offset correction register (CAL) and temperature compensation register (TCMP) must only be written when the RTCTCRDY bit in the STA register is set. RTCTCRDY is a read-only bit which is set when the hardware is ready to take in new correction or compensation values. Writes applied when RTCTCRDY is cleared have no effect. The RTCTCOK status bit is provided in the STA register. If a write to CAL or TCMP was successful, RTCTCOK will be set, else it will be reset. RTCTCOK will hold its status until the next write attempt. If a write is unsuccessful, software needs to re-attempt the write when RTCTCRDY is set.

The following registers are subject to the restrictions above: TCMP, CAL.

**Prescaler Control Registers**

The prescaler interval control registers can be read at any time. Writes to the prescaler control registers should only be done when the PS0 and PS1 interrupts are disabled.

The following registers are subject to the restrictions above: PSCTL.

### *38.3.3 Binary vs. BCD*

The RTC provides seconds, minutes, hours, day of week, day of month, month, and year in selectable binary or binary-coded decimal (BCD) format. When the RTCBCD bit in the CTL register of the RTC is set, BCD format is selected. When BCD format is selected, only the BCD bits are valid in the calendar registers. The format must be selected before the time is set.

### *38.3.4 Leap Year Handling*

The calendar includes a leap-year algorithm which considers all years evenly divisible by four as leap years. This algorithm is accurate from the year 1901 through 2099.

### *38.3.5 Calendar Alarm Configuration*

The RTC provides a flexible alarm system. There are two user-programmable alarms (A0 and A1) which can be programmed based on the settings contained in the alarm registers for minutes, hours, day of week, and day of month.



Copyright © 2026 Texas Instruments Incorporated

Each alarm register contains an alarm enable (AE) bit that can be used to enable the respective alarm register. By setting the AE bits of the various alarm registers, a variety of alarm events can be generated.

- **Example 1:** A user wants to set an alarm every hour at 15 minutes past the hour (for example, at 00:15:00, 01:15:00, 02:15:00, and so on). This is possible by setting AxMIN to 15. By setting the AE bit of the AxMIN and clearing all other AE bits of the alarm registers, the alarm is enabled. When enabled, the RTCAx interrupt flag is set when the count transitions from 00:14:59 to 00:15:00, 01:14:59 to 01:15:00, 02:14:59 to 02:15:00, and so on.
- **Example 2:** A user wants to set an alarm every day at 04:00:00. This is possible by setting AxHOUR to 4. By setting the AE bit of the AxHOUR and clearing all other AE bits of the alarm registers, the alarm is enabled. When enabled, the RTCAx interrupt flag is set when the count transitions from 03:59:59 to 04:00:00.
- **Example 3:** A user wants to set an alarm for 06:30:00. AxHOUR would be set to 6 and AxMIN would be set to 30. By setting the AE bits of AxHOUR and AxMIN, the alarm is enabled. When enabled, the RTCAx interrupt flag is set when the time count transitions from 06:29:59 to 06:30:00. In this case, the alarm event occurs every day at 06:30:00.
- **Example 4:** A user wants to set an alarm every Tuesday at 06:30:00. AxDOW in the AxDAY register would be set to 2, AxHOUR would be set to 6 and AxMIN would be set to 30. By setting the AE bits of AxDOW, AxHOUR and AxMIN, the alarm is enabled. When enabled, the RTCAx interrupt flag is set when the time count transitions from 06:29:59 to 06:30:00 and the DOW transitions from 1 to 2.
- **Example 5:** A user wants to set an alarm the fifth day of each month at 06:30:00. AxDAY would be set to 5, AxHOUR would be set to 6 and AxMIN would be set to 30. By setting the AE bits of AxDAY, AxHOUR and AxMIN, the alarm is enabled. When enabled, the RTCAx interrupt flag is set when the time count transitions from 06:29:59 to 06:30:00 and the RTCDAY equals 5.

**Note**

**Invalid alarm settings**

Invalid alarm settings are not checked through hardware. It is the user's responsibility that valid alarm settings are entered.

**Note**

**Invalid time and date values**

Writing of invalid date and/or time information or data values outside the legal ranges specified in the calendar and alarm registers can result in unpredictable behavior.

**Note**

**Setting the alarm**

Before setting an initial alarm, all alarm registers including the AE bits should be cleared.

To prevent potential erroneous alarm conditions from occurring, the alarms should be disabled by clearing the RTCAx interrupt enable, RTCAx interrupt flag, and AE bits before writing initial or new time values to the RTC time registers.

### *38.3.6 Interval Alarm Configuration*

The interval alarm is provided in addition to the two calendar alarms, and can be configured to generate an interval alarm event when the following events occur:

- MIN changes
- HOUR changes
- Every day at midnight
- Every day at noon



Copyright © 2026 Texas Instruments Incorporated

The interval is selected by programming the RTCTEVTX field in the CTL register of the RTC. The interval alarm sources the RTCTEV interrupt.

### 38.3.7 Periodic Alarm Configuration

Two periodic alarms are provided, and can be configured to generate two periodic time bases for the application.

The RT0PS periodic alarm can be configured to generate a periodic interrupt at one of the following rates: 4096Hz, 2048Hz, 1024Hz, 512Hz, 256Hz, or 128Hz. The RT0PS period is set by the RT0IP bits in the PSCTL register. The RT0PS event sources the RT0PS interrupt.

The RT1PS periodic alarm can be configured to generate a periodic interrupt at one of the following rates: 64Hz, 32Hz, 16Hz, 8Hz, 4Hz, 2Hz, 1Hz, or 0.5Hz. The RT1PS period is set by the RT1IP bits in the PSCTL register. The RT1PS event sources the RT1PS interrupt.

**Note**

Changing the settings of the periodic alarm configuration (RT0IP, RT1IP) while the corresponding prescalar is running can result in setting the corresponding interrupt flags. To avoid this, the RT0PS and RT1PS interrupts should be masked when changing RT0IP or RT1IP while the prescaler is running.

### 38.3.8 Calibration

The RTC module provides calibration mechanisms for reducing timekeeping error due to crystal offset error and crystal temperature drift. Crystal offset error can be corrected by measuring the crystal frequency through the RTC_OUT output and adjusting calibration registers in the RTC. Crystal temperature drift can be corrected by measuring the system temperature with software, computing a correction factor in ppm based on the measured temperature, and applying the correction factor to the RTC.

Up to ±240 ppm of offset correction or up to ±240 ppm of temperature compensation correction can be applied individually, but the total correction allowed when both offset correction and temperature compensation are used together is ±240 ppm. This means that the signed addition of offset error and temperature compensation should not exceed the maximum of ±240 ppm, else the values in excess of ±240 ppm will be ignored by hardware.



Copyright © 2026 Texas Instruments Incorporated

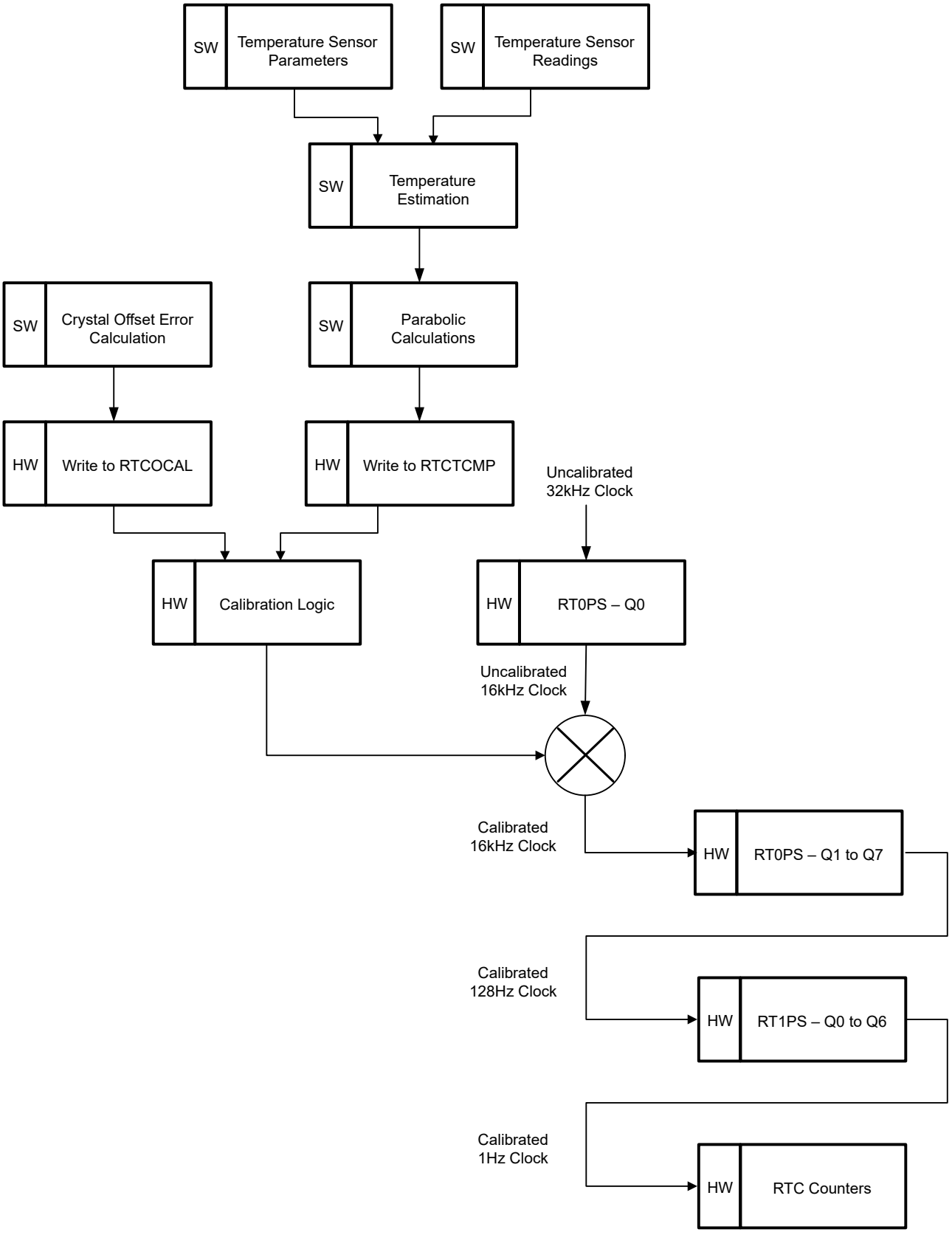



**Figure 38-2. RTC Calibration**



Copyright © 2026 Texas Instruments Incorporated

#### 38.3.8.1 Crystal Offset Error

The RTC module can be calibrated for crystal manufacturing tolerance or offset error for higher timekeeping accuracy. A crystal frequency error of up to ±240 ppm can be calibrated smoothly over a period of 60 seconds.

The CAL register is used to adjust the frequency. The calibration value is written into the RTCOCALX field in the CAL register, and each LSB in this field represents approximately ±1 ppm correction (the sign is based on the RTCOCALS bit in the CAL register). When RTCOCALS is set, up calibration is selected and each LSB in RTCOCALX represents +1 ppm adjustment. When RTOCALS is cleared, down calibration is selected and each LSB in RTCOCALX represents -1 ppm adjustment. The RTCOCALX field is 8 bits. Software can write a value of up to 256ppm into this register, but the maximum frequency error that can be corrected is only 240ppm. Software is responsible for writing legal values into RTCOCALX. Offset error calibration is inactive when the RTC is not enabled (MODCLKEN is cleared) or when RTCOCALX is zero. Writing to the RTOCAL register resets the temperature compensation to zero.

**Note**

The CAL register must only be accessed via half-word (16-bit) or word (32-bit) wide operations to ensure that the sign bit is set together with the calibration value.

To calibrate the RTC frequency, the RTC_OUT output signal is available on a pin. The RTCCALFX field in the CAL register can be used to select the frequency rate of the output signal to the pin. The other settings of the RTCCALFX select one of the three frequency options to output: 512Hz, 256Hz, or 1Hz. RTC_OUT output can be measured, and the result of this measurement can be applied to the RTCOCALS and RTCOCALX bits to effectively reduce the initial offset of the RTCCLK.

##### *38.3.8.1.1 Offset Error Correction Mechanism*

In the RTC, offset error calibration takes place over a period of 60 seconds. To achieve approximately ±1ppm correction, the 16kHz clock (Q0 output of RT0PS) is adjusted to add or subtract 1 clock pulse. This correction happens once every quarter second until the programmed ppm error is compensated.

All three possible output frequencies 512Hz, 256Hz, and 1Hz at the RTC_OUT pin are affected by calibration settings. RT0PS interrupt triggered by RT0PS – Q2 to Q7 is based on calibrated clock. RT1PS interrupt (RT1PS) and RTC clock time event interrupt (RTCTEV) are also based on the calibrated clock.

The following can be used as a guide for setting RTCOCALS and RTCOCALX, given $f_{RTCCLK} = f_{RTCCLK, meas}$ x divider factor (set by RTCCALFX):

**Slow Crystal ($f_{RTCCLK}$ < 32768Hz )**

- Set RTCOCALS = 1
- Set RTCOCALX to Round (60 x 16384 x (1-$f_{RTCCLK}$/32768))

As an example for up calibration, when the measured frequency is 511.9658Hz against the reference frequency of 512Hz, the frequency error is approximately 67ppm low. In order to increase the frequency by 67ppm, RTCOCALS should be set, and RTCOCALx should be set to Round (60 x 16384 x (1-511.9658x64/32768)) = 66.

**Fast Crystal ($f_{ACLK}$ ≥ 32768Hz)**

- Set RTCOCALS = 0
- Set RTCOCALX to Round (60 x 16384 x (1-$f_{RTCCLK}$/32768))

As an example for down calibration, when the measured frequency is 512.0241Hz against the reference frequency of 512Hz, the frequency error is approximately 47ppm high. In order to decrease the frequency by 47ppm, RTCOCALS should be cleared, and RTCOCALx should be set to Round (60 x 16384 x (1-512.0241x64/32768)) = 46.

#### 38.3.8.2 Crystal Temperature Error

The output frequency of a crystal can vary significantly due to drift in temperature. A hybrid software and hardware approach can be applied to achieve temperature compensation for the RTC. The RTC supports temperature compensation up to ±240 ppm.



Copyright © 2026 Texas Instruments Incorporated

Application software can make use of the on-chip temperature sensor to measure the device temperature at desired intervals (for example, every few seconds or minutes) to approximate the ambient temperature of the circuit. Then, software can be used to do parabolic calculations to determine the corresponding frequency error in ppm. This frequency error can then be written into the TCMP register for temperature compensation.

The RTCTCMPX field contains 8 bits which provide frequency correction up to ±240 ppm. The LSB represents ±1 ppm based on the RTCTCMPS bit in the TCMP register. When RTCTCMPS is set, each step in RTCTCMPX represents a +1ppm adjustment (up calibration). When the RTCTCMPS bit is cleared, each step in RTCTCMPX represents a -1ppm adjustment (down calibration).

#### *38.3.8.2.1 Temperature Drift Correction Mechanism*

When the temperature compensation value is written into RTCTCMPX, it is added together with the offset error calibration value an the resulting value is taken into account from the next calibration cycle onwards. The ongoing calibration cycle will not be affected by writes into the TCMP register.

Reading the TCMPX at any time returns the cumulative correction value (offset + temperature) which is the signed addition of RTCOCALX and RTCTCMPX, and the updated sign bit (RTCTCMPS) of the addition result. Note that writing to the RTCOCAL register will reset the temperature compensation value to zero.

For example, when RTCOCALX is +150 ppm, and RTCTCMP is written with +200 ppm, the effective value taken in for the next calibration cycle is +240 ppm (saturated). If the RTCOCALX is +150 ppm, and RTCTCMP is written with +50 ppm, the effective value taken for the next calibration cycle is +200 ppm (40 ppm of compensation headroom remains).

To achieve effective temperature compensation, software is responsible to:

1. Measure the temperature as often as required by the operating conditions, crystal performance, and targeted accuracy
2. Calculate the frequency error due to temperature
3. Write the temperature compensation value into RTCTCMPX while not exceeding the maximum combined correction limit of ±240 ppm

Writing to the TCMP register for temperature compensation requires 60 seconds (one minute) to take effect as a part of the next calibration cycle. Therefore, if temperature must be measured more frequently than once per minute (for example, once every 5 seconds) then it needs to average the error over one minute and update the TCMP register once per minute.

**Note**

The TCMP register must only be accessed via half-word (16-bit) or word (32-bit) wide operations to ensure that the sign bit is set together with the calibration value.

### *38.3.9 RTC Prescaler Extension*

In RTC_x instances only, a three-bit prescaler extension is included to support a longer interval time for a heartbeat function. The RT2PS prescaler allows interrupt generation with a 4s, 8s, or 16s interval.

The RTCCLK is used to provide RTC clock to the outside (512Hz, 256Hz, or 1Hz). The clock source is after the calibration circuit and allows to trim the LFXT offset and temperature compensation register.

For some applications, the raw uncalibrated 32kHz clock can be used as source for other components in the system. For LFSS, LFCLK is a mux option of RTCCLK.

See Figure 38-3 for configuration of the prescalers, internal timer events, and RTC output options.



Copyright © 2026 Texas Instruments Incorporated

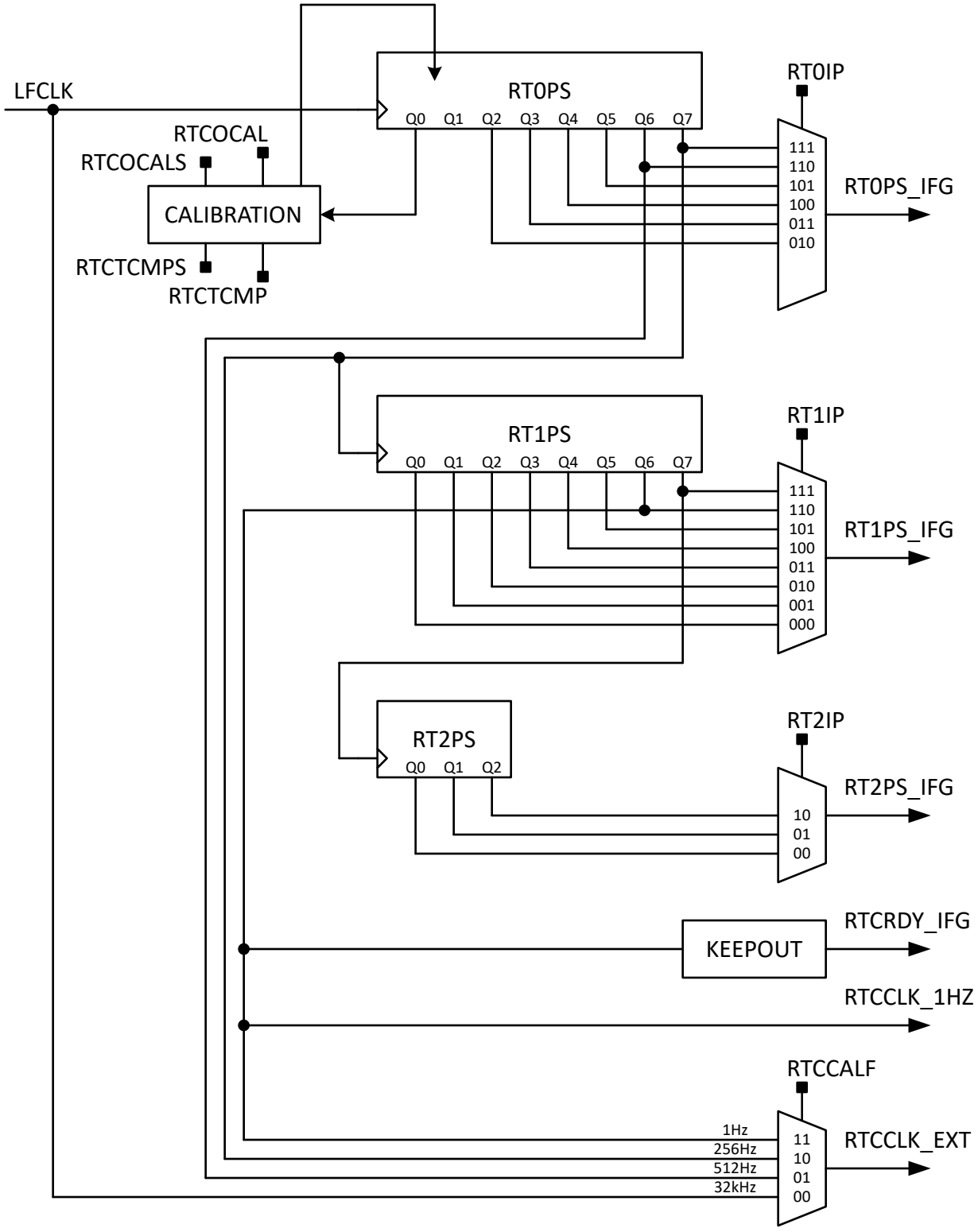



**Figure 38-3. RTC Prescaler and Interval Timer Event**

### *38.3.10 RTC Timestamp Capture*

In RTC_x instances only, an RTC timestamp feature allows the capture of an RTC status on the detection of a timestamp event. When a timestamp event happens, the whole RTC register from seconds to years is copied



Copyright © 2026 Texas Instruments Incorporated

to the timestamp capture registers. In case of multiple timestamp events, the TSCAPTURE register defines whether only the first or every (last) RTC status should be captured. The timestamp event status is sticky and needs to be cleared by software with the TSCLR register bit.

See Figure 38-4 for the RTC block diagram and timestamping.

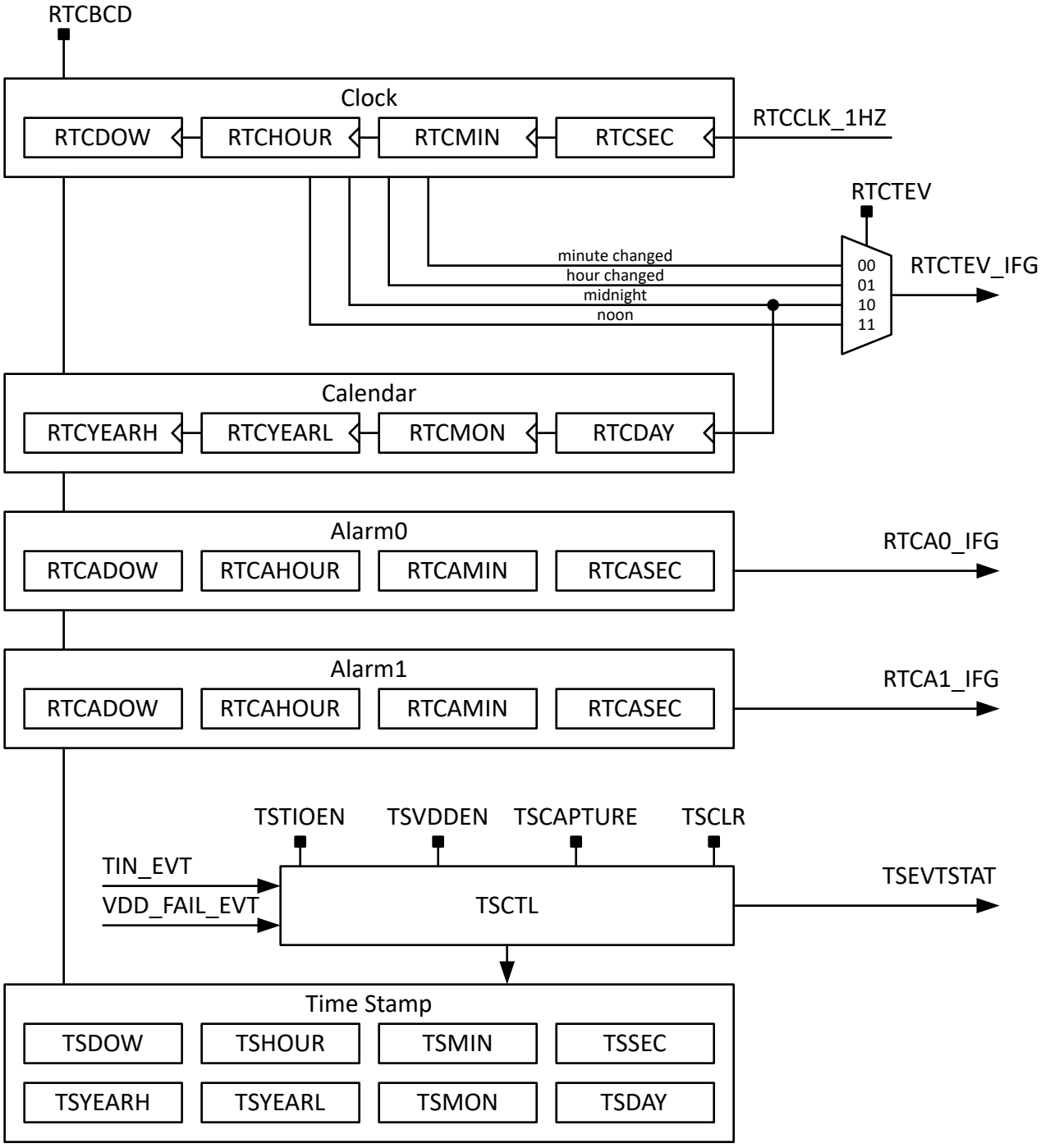



**Figure 38-4. RTC Block Diagram and Timestamping**

The following timestamp events are supported:

- **Tamper I/O input event after filtering and edge detection**



Copyright © 2026 Texas Instruments Incorporated

  - Depending on the configuration of the polarity (rising, falling or both edges), the tamper I/O input will trigger a timestamp event capture when the TIO changes its state.
  - This feature is used in a typical tamper detect application when the TIO input is connected to a switch which detects the opening of an enclosure.
- **VDD fail event**
  - The VDD fail event allows the capture of a timestamp when the main supply (VDD) is lost. Once the device's main supply regains power, the timestamp status information indicates that a power low timestamp event happened.
  - This feature is used in applications where it is important to know how long the main power was lost.

### *38.3.11 RTC Events*

The RTC module contains two event publishers and no event subscribers. One event publisher (CPU_INT) manages RTC interrupt requests (IRQs) to the CPU subsystem through a static event route. The second event publisher (GEN_EVENT) can be used to publish RTC events to a subscriber through a generic event route channel.

The RTC events are summarized in Table 38-2.

**Table 38-2. RTC Events**

| Event | Type | Source | Destination | Route | Configuration | Functionality |
|---|---|---|---|---|---|---|
| CPU interrupt event | Publisher | RTC | CPU Subsystem | Static route | CPU_INT registers | Fixed interrupt route from RTC to CPU |
| Generic event | Publisher | RTC | Generic event channel | Generic route (FPUB_0) | GEN_EVENT registers, FPUB_0 register | Trigger generic event channel from RTC |

#### 38.3.11.1 CPU Interrupt Event Publisher (CPU_INT)

The RTC module provides six interrupt sources which can be configured to source a CPU interrupt event. In order of decreasing interrupt priority, the CPU interrupt events from the RTC are given in Table 38-3.

**Table 38-3. RTC CPU Interrupt Event Conditions (CPU_INT)**

| Index (IIDX) | Name | Description |
|---|---|---|
| 0 | RTCRDY | Indicates that the RTC counter/calendar registers are safe to be read for approximately one second |
| 1 | RTCTEV | Interval interrupt, configurable as once per minute, per hour, at midnight, or at noon |
| 2 | RTCA1 | Calendar alarm 1 interrupt |
| 3 | RTCA2 | Calendar alarm 2 interrupt |
| 4 | RTC0PS | Prescaler 0 periodic alarm interrupt |
| 5 | RTC1PS | Prescaler 1 periodic alarm interrupt |

The CPU interrupt event configuration is managed with the CPU_INT event management registers. See Section 7.2.5 for guidance on configuring the event registers for CPU interrupts.

#### 38.3.11.2 Generic Event Publisher (GEN_EVENT)

The RTC module provides six interrupt sources, of which one can be configured to publish an event to a generic event route channel are given in Table 38-4.

**Table 38-4. RTC Generic Event Publisher Conditions (GEN_EVENT)**

| Index | Name | Description |
|---|---|---|
| 0 | RTCRDY | Indicates that the RTC counter/calendar registers are safe to be read for approximately one second |



Copyright © 2026 Texas Instruments Incorporated

**Table 38-4. RTC Generic Event Publisher Conditions (GEN_EVENT) (continued)**

| Index | Name | Description |
|---|---|---|
| 1 | RTCTEV | Interval interrupt, configurable as once per minute, per hour, at midnight, or at noon |
| 2 | RTCA1 | Calendar alarm 1 interrupt |
| 3 | RTCA2 | Calendar alarm 2 interrupt |
| 4 | RTC0PS | Prescaler 0 periodic alarm interrupt |
| 5 | RTC1PS | Prescaler 1 periodic alarm interrupt |

The generic event publisher configuration is managed with the GEN_EVENT event management registers. See Section 7.2.5 for guidance on configuring the event registers for generic event publishers.

The generic event channel which generic event is to publish to must be selected by writing the target generic channel ID to the **FPUB_0** register in the RTC. See Section 7.1.3.3 for guidance on configuring generic event routes.

If this publisher is not used in an application, the FPUB_0 register can be left in a disconnected state (set equal to zero) and no events should be unmasked through the MIS register in the RTC GEN_EVENT register set.



Copyright © 2026 Texas Instruments Incorporated

## 38.4 RTC Registers

Table 38-5 lists the memory-mapped registers for the RTC registers. All register offset addresses not listed in Table 38-5 should be considered as reserved locations and the register contents should not be modified.

**Table 38-5. RTC Registers**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 444h | FPUB_0 | Publisher Port 0 | | Go |
| 800h | PWREN | Power enable | | Go |
| 804h | RSTCTL | Reset Control | | Go |
| 808h | CLKCFG | Peripheral Clock Configuration Register | | Go |
| 814h | STAT | Status Register | | Go |
| 1004h | CLKSEL | Clock Select for Ultra Low Power peripherals | | Go |
| 1020h | IIDX | Interrupt Index Register | CPU_INT | Go |
| 1028h | IMASK | Interrupt mask | CPU_INT | Go |
| 1030h | RIS | Raw interrupt status | CPU_INT | Go |
| 1038h | MIS | Masked interrupt status | CPU_INT | Go |
| 1040h | ISET | Interrupt set | CPU_INT | Go |
| 1048h | ICLR | Interrupt clear | CPU_INT | Go |
| 1050h | IIDX | Interrupt Index Register | GEN_EVENT | Go |
| 1054h | IMASK | Interrupt mask | GEN_EVENT | Go |
| 1058h | RIS | Raw interrupt status | GEN_EVENT | Go |
| 105Ch | MIS | Masked interrupt status | GEN_EVENT | Go |
| 1060h | ISET | Interrupt set | GEN_EVENT | Go |
| 1064h | ICLR | Interrupt clear | GEN_EVENT | Go |
| 10E0h | EVT_MODE | Event Mode | | Go |
| 10FCh | DESC | RTC Descriptor Register | | Go |
| 1100h | CLKCTL | RTC Clock Control Register | | Go |
| 1104h | DBGCTL | RTC Module Debug Control Register | | Go |
| 1108h | CTL | RTC Control Register | | Go |
| 110Ch | STA | RTC Status Register | | Go |
| 1110h | CAL | RTC Clock Offset Calibration Register | | Go |
| 1114h | TCMP | RTC Temperature Compensation Register | | Go |
| 1118h | SEC | RTC Seconds Register - Calendar Mode With Binary / BCD Format | | Go |
| 111Ch | MIN | RTC Minutes Register - Calendar Mode With Binary / BCD Format | | Go |
| 1120h | HOUR | RTC Hours Register - Calendar Mode With Binary / BCD Format | | Go |
| 1124h | DAY | RTC Day Of Week / Month Register - Calendar Mode With Binary / BCD Format | | Go |
| 1128h | MON | RTC Month Register - Calendar Mode With Binary / BCD Format | | Go |
| 112Ch | YEAR | RTC Year Register - Calendar Mode With Binary / BCD Format | | Go |
| 1130h | A1MIN | RTC Minute Alarm Register - Calendar Mode With Binary / BCD Format | | Go |
| 1134h | A1HOUR | RTC Hours Alarm Register - Calendar Mode With Binary / BCD Format | | Go |



Copyright © 2026 Texas Instruments Incorporated

**Table 38-5. RTC Registers (continued)**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 1138h | A1DAY | RTC Alarm Day Of Week / Month Register - Calendar Mode With Binary / BCD Format | | Go |
| 113Ch | A2MIN | RTC Minute Alarm Register - Calendar Mode With Binary / BCD Format | | Go |
| 1140h | A2HOUR | RTC Hours Alarm Register - Calendar Mode With Binary / BCD Format | | Go |
| 1144h | A2DAY | RTC Alarm Day Of Week / Month Register - Calendar Mode With Binary / BCD Format | | Go |
| 1148h | PSCTL | RTC Prescale Timer 0/1 Control Register | | Go |
| 114Ch | EXTPSCTL | RTC Prescale Timer 2 Control Register | | Go |
| 1150h | TSSEC | Time Stamp Seconds Register - Calendar Mode With Binary / BCD Format | | Go |
| 1154h | TSMIN | Time Stamp Minutes Register - Calendar Mode With Binary / BCD Format | | Go |
| 1158h | TSHOUR | Time Stamp Hours Register - Calendar Mode With Binary / BCD Format | | Go |
| 115Ch | TSDAY | Time Stamp Day Of Week / Month Register - Calendar Mode With Binary / BCD Format | | Go |
| 1160h | TSMON | Time Stamp Month Register - Calendar Mode With Binary / BCD Format | | Go |
| 1164h | TSYEAR | Time Stamp Years Register - Calendar Mode With Binary / BCD Format | | Go |
| 1168h | TSSTAT | Time Stamp Status Register | | Go |
| 116Ch | TSCTL | Time Stamp Control Register | | Go |
| 1170h | TSCLR | Time Stamp Clear Register | | Go |
| 1174h | LFSSRST | Low frequency subsystem reset request | | Go |
| 1178h | RTCLOCK | Real time clock lock register | | Go |

Complex bit access types are encoded to fit into small table cells. Table 38-6 shows the codes that are used for access types in this section.

**Table 38-6. RTC Access Type Codes**

| Access Type | Code | Description |
|---|---|---|
| Read Type | | |
| R | R | Read |
| R-0 | R<br>-0 | Read<br>Returns 0s |
| Write Type | | |
| W | W | Write |
| WK | W<br>K | Write<br>Write protected by a key |
| Reset or Default Value | | |
| *-n* | | Value after reset or the default value |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.1 FPUB_0 (Offset = 444h) [Reset = 00000000h]

FPUB_0 is shown in Figure 38-5 and described in Table 38-7.

Return to the Summary Table.

Publisher port

**Figure 38-5. FPUB_0**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | CHANID | | | |
| R-0h | | | | | | | | | | | | R/W-0h | | | |

**Table 38-7. FPUB_0 Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3-0 | CHANID | R/W | 0h | 0 = disconnected.<br>1-15 = connected to channelID = CHANID.<br>0h = A value of 0 specifies that the event is not connected<br>Fh = Consult your device data sheet as the actual allowed maximum may be less than 15. |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.2 PWREN (Offset = 800h) [Reset = 00000000h]

PWREN is shown in Figure 38-6 and described in Table 38-8.

Return to the Summary Table.

Register to control the power state

**Figure 38-6. PWREN**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| R-0/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | ENABLE |
| R-0h | | | | | | | R/WK-0h |

**Table 38-8. PWREN Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | R-0/W | 0h | KEY to allow Power State Change<br>26h = KEY to allow write access to this register |
| 23-1 | RESERVED | R | 0h | |
| 0 | ENABLE | R/WK | 0h | Enable the power<br>KEY must be set to 26h to write to this bit.<br>0h = Disable Power<br>1h = Enable Power |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.3 RSTCTL (Offset = 804h) [Reset = 00000000h]

RSTCTL is shown in Figure 38-7 and described in Table 38-9.

Return to the Summary Table.

Register to control reset assertion and de-assertion

**Figure 38-7. RSTCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | RESETSTKYCLR | RESETASSERT |
| R-0h | | | | | | WK-0h | WK-0h |

**Table 38-9. RSTCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | Unlock key<br>B1h = KEY to allow write access to this register |
| 23-2 | RESERVED | R | 0h | |
| 1 | RESETSTKYCLR | WK | 0h | Clear the RESETSTKY bit in the STAT register<br>KEY must be set to B1h to write to this bit.<br>0h = Writing 0 has no effect<br>1h = Clear reset sticky bit |
| 0 | RESETASSERT | WK | 0h | Assert reset to the peripheral<br>KEY must be set to B1h to write to this bit.<br>0h = Writing 0 has no effect<br>1h = Assert reset |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.4 CLKCFG (Offset = 808h) [Reset = 00000000h]

CLKCFG is shown in Figure 38-8 and described in Table 38-10.

Return to the Summary Table.

Peripheral Clock Configuration Register

**Figure 38-8. CLKCFG**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| R-0/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | BLOCKASYNC |
| R-0h | | | | | | | R/WK-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |

**Table 38-10. CLKCFG Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | R-0/W | 0h | KEY to Allow State Change -- 0xA9<br>A9h (W) = key value to allow change field of GPRCM |
| 23-9 | RESERVED | R | 0h | |
| 8 | BLOCKASYNC | R/WK | 0h | Async Clock Request is blocked from starting SYSOSC or forcing bus clock to 32MHz<br>KEY must be set to A9h to write to this bit.<br>0h = Not block async clock request<br>1h = Block async clock request |
| 7-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.5 STAT (Offset = 814h) [Reset = 00000000h]

STAT is shown in Figure 38-9 and described in Table 38-11.

Return to the Summary Table.

peripheral enable and reset status

**Figure 38-9. STAT**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | RESETSTKY |
| R-0h | | | | | | | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |

**Table 38-11. STAT Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-17 | RESERVED | R | 0h | |
| 16 | RESETSTKY | R | 0h | This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register<br>0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register<br>1h = The peripheral was reset since the last bit clear |
| 15-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.6 CLKSEL (Offset = 1004h) [Reset = 00000000h]

CLKSEL is shown in Figure 38-10 and described in Table 38-12.

Return to the Summary Table.

Clock source selection for ULP peripherals

**Figure 38-10. CLKSEL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | LFCLK_SEL | RESERVED |
| R-0h | | | | | | R-0h | R-0h |

**Table 38-12. CLKSEL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | |
| 1 | LFCLK_SEL | R | 0h | Selects LFCLK as clock source if enabled<br>0h = LFCLK is disabled as clock source<br>1h = LFCLK is enabled as clock source |
| 0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.7 IIDX (Offset = 1020h) [Reset = 00000000h]

IIDX is shown in Figure 38-11 and described in Table 38-13.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. Value 0x00 means no event pending. Interrupt 1 is the highest priority, IIDX next highest, 4, 8, … IIDX^31 is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred. On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in [RIS] and [MIS] are cleared as well. After a read from the CPU (not from the debug interface), the register is updated with the next highest priority interrupt, if none are pending, then it should display 0x0.

**Figure 38-11. IIDX**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | STAT | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | |

**Table 38-13. IIDX Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | STAT | R | 0h | Interrupt index status<br>00h = No interrupt pending<br>1h = RTC-Ready interrupt; Interrupt flag: RTCRDY<br>2h = Time-Event interrupt; Interrupt flag: RTCTEV<br>3h = Alarm-1 interrupt; Interrupt flag: RTCA1<br>4h = Alarm-2 interrupt; Interrupt flag: RTCA2<br>5h = Prescaler-0 interrupt; Interrupt flag: RT0PS<br>6h = Prescaler-1 interrupt; Interrupt flag: RT1PS<br>07h = RTC prescale timer 2<br>08h = Time stamp event<br>09h = Tamper I/O 0 event<br>0Ah = Tamper I/O 1 event<br>0Bh = Tamper I/O 2 event<br>0Ch = Tamper I/O 3 event<br>0Dh = Tamper I/O 4 event<br>0Eh = Tamper I/O 5 event<br>0Fh = Tamper I/O 6 event<br>10h = Tamper I/O 7 event<br>11h = Tamper I/O 8 event<br>12h = Tamper I/O 9 event<br>13h = Tamper I/O 10 event<br>14h = Tamper I/O 11 event<br>15h = Tamper I/O 12 event<br>16h = Tamper I/O 13 event<br>17h = Tamper I/O 14 event<br>18h = Tamper I/O 15 event |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.8 IMASK (Offset = 1028h) [Reset = 00000000h]

IMASK is shown in Figure 38-12 and described in Table 38-14.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is unmasked. Unmasking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

**Figure 38-12. IMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TIO15 | TIO14 | TIO13 | TIO12 | TIO11 | TIO10 | TIO9 | TIO8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TIO7 | TIO6 | TIO5 | TIO4 | TIO3 | TIO2 | TIO1 | TIO0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TSEVT | RT2PS | RT1PS | RT0PS | RTCA2 | RTCA1 | RTCTEV | RTCRDY |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

**Table 38-14. IMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | RESERVED | R | 0h | |
| 23 | TIO15 | R/W | 0h | Tamper I/O 15 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 22 | TIO14 | R/W | 0h | Tamper I/O 14 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 21 | TIO13 | R/W | 0h | Tamper I/O 13 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 20 | TIO12 | R/W | 0h | Tamper I/O 12 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 19 | TIO11 | R/W | 0h | Tamper I/O 11 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 18 | TIO10 | R/W | 0h | Tamper I/O 10 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 17 | TIO9 | R/W | 0h | Tamper I/O 9 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 16 | TIO8 | R/W | 0h | Tamper I/O 8 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 15 | TIO7 | R/W | 0h | Tamper I/O 7 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 14 | TIO6 | R/W | 0h | Tamper I/O 6 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |



Copyright © 2026 Texas Instruments Incorporated

**Table 38-14. IMASK Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 13 | TIO5 | R/W | 0h | Tamper I/O 5 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 12 | TIO4 | R/W | 0h | Tamper I/O 4 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 11 | TIO3 | R/W | 0h | Tamper I/O 3 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 10 | TIO2 | R/W | 0h | Tamper I/O 2 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 9 | TIO1 | R/W | 0h | Tamper I/O 1 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 8 | TIO0 | R/W | 0h | Tamper I/O 0 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 7 | TSEVT | R/W | 0h | Time stamp event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 6 | RT2PS | R/W | 0h | RTC prescale timer 2<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 5 | RT1PS | R/W | 0h | Enable Prescaler-1 interrupt<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 4 | RT0PS | R/W | 0h | Enable Prescaler-0 interrupt<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 3 | RTCA2 | R/W | 0h | Enable Alarm-2 interrupt<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 2 | RTCA1 | R/W | 0h | Enable Alarm-1 interrupt<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 1 | RTCTEV | R/W | 0h | Enable Time-Event interrupt<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 0 | RTCRDY | R/W | 0h | Enable RTC-Ready interrupt<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.9 RIS (Offset = 1030h) [Reset = 00000000h]

RIS is shown in Figure 38-13 and described in Table 38-15.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Figure 38-13. RIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TIO15 | TIO14 | TIO13 | TIO12 | TIO11 | TIO10 | TIO9 | TIO8 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TIO7 | TIO6 | TIO5 | TIO4 | TIO3 | TIO2 | TIO1 | TIO0 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TSEVT | RT2PS | RT1PS | RT0PS | RTCA2 | RTCA1 | RTCTEV | RTCRDY |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 38-15. RIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | RESERVED | R | 0h | |
| 23 | TIO15 | R | 0h | Tamper I/O 15 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 22 | TIO14 | R | 0h | Tamper I/O 14 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 21 | TIO13 | R | 0h | Tamper I/O 13 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 20 | TIO12 | R | 0h | Tamper I/O 12 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 19 | TIO11 | R | 0h | Tamper I/O 11 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 18 | TIO10 | R | 0h | Tamper I/O 10 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 17 | TIO9 | R | 0h | Tamper I/O 9 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 16 | TIO8 | R | 0h | Tamper I/O 8 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 15 | TIO7 | R | 0h | Tamper I/O 7 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

**Table 38-15. RIS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 14 | TIO6 | R | 0h | Tamper I/O 6 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 13 | TIO5 | R | 0h | Tamper I/O 5 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 12 | TIO4 | R | 0h | Tamper I/O 4 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 11 | TIO3 | R | 0h | Tamper I/O 3 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 10 | TIO2 | R | 0h | Tamper I/O 2 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 9 | TIO1 | R | 0h | Tamper I/O 1 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 8 | TIO0 | R | 0h | Tamper I/O 0 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 7 | TSEVT | R | 0h | Time stamp event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 6 | RT2PS | R | 0h | RTC prescale timer 2<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 5 | RT1PS | R | 0h | Raw Prescaler-1 interrupt status<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 4 | RT0PS | R | 0h | Raw Prescaler-0 interrupt status<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 3 | RTCA2 | R | 0h | Raw Alarm-2 interrupts status<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 2 | RTCA1 | R | 0h | Raw Alarm-1 interrupt status<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 1 | RTCTEV | R | 0h | Raw Time-Event interrupt status<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 0 | RTCRDY | R | 0h | Raw RTC-Ready interrupts status<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.10 MIS (Offset = 1038h) [Reset = 00000000h]

MIS is shown in Figure 38-14 and described in Table 38-16.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 38-14. MIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TIO15 | TIO14 | TIO13 | TIO12 | TIO11 | TIO10 | TIO9 | TIO8 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TIO7 | TIO6 | TIO5 | TIO4 | TIO3 | TIO2 | TIO1 | TIO0 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TSEVT | RT2PS | RT1PS | RT0PS | RTCA2 | RTCA1 | RTCTEV | RTCRDY |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 38-16. MIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | RESERVED | R | 0h | |
| 23 | TIO15 | R | 0h | Tamper I/O 15 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 22 | TIO14 | R | 0h | Tamper I/O 14 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 21 | TIO13 | R | 0h | Tamper I/O 13 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 20 | TIO12 | R | 0h | Tamper I/O 12 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 19 | TIO11 | R | 0h | Tamper I/O 11 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 18 | TIO10 | R | 0h | Tamper I/O 10 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 17 | TIO9 | R | 0h | Tamper I/O 9 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 16 | TIO8 | R | 0h | Tamper I/O 8 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 15 | TIO7 | R | 0h | Tamper I/O 7 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 14 | TIO6 | R | 0h | Tamper I/O 6 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

**Table 38-16. MIS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 13 | TIO5 | R | 0h | Tamper I/O 5 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 12 | TIO4 | R | 0h | Tamper I/O 4 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 11 | TIO3 | R | 0h | Tamper I/O 3 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 10 | TIO2 | R | 0h | Tamper I/O 2 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 9 | TIO1 | R | 0h | Tamper I/O 1 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 8 | TIO0 | R | 0h | Tamper I/O 0 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 7 | TSEVT | R | 0h | Time stamp event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 6 | RT2PS | R | 0h | RTC prescale timer 2<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 5 | RT1PS | R | 0h | Masked Prescaler-1 interrupt status<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 4 | RT0PS | R | 0h | Masked Prescaler-0 interrupt status<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 3 | RTCA2 | R | 0h | Masked Alarm-2 interrupt status<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 2 | RTCA1 | R | 0h | Masked Alarm-1 interrupt status<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 1 | RTCTEV | R | 0h | Masked Time-Event interrupt status<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 0 | RTCRDY | R | 0h | Masked RTC-Ready interrupt status<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.11 ISET (Offset = 1040h) [Reset = 00000000h]

ISET is shown in Figure 38-15 and described in Table 38-17.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 38-15. ISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TIO15 | TIO14 | TIO13 | TIO12 | TIO11 | TIO10 | TIO9 | TIO8 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TIO7 | TIO6 | TIO5 | TIO4 | TIO3 | TIO2 | TIO1 | TIO0 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TSEVT | RT2PS | RT1PS | RT0PS | RTCA2 | RTCA1 | RTCTEV | RTCRDY |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

**Table 38-17. ISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | RESERVED | R | 0h | |
| 23 | TIO15 | W | 0h | Tamper I/O 15 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 22 | TIO14 | W | 0h | Tamper I/O 14 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 21 | TIO13 | W | 0h | Tamper I/O 13 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 20 | TIO12 | W | 0h | Tamper I/O 12 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 19 | TIO11 | W | 0h | Tamper I/O 11 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 18 | TIO10 | W | 0h | Tamper I/O 10 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 17 | TIO9 | W | 0h | Tamper I/O 9 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 16 | TIO8 | W | 0h | Tamper I/O 8 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 15 | TIO7 | W | 0h | Tamper I/O 7 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |



Copyright © 2026 Texas Instruments Incorporated

**Table 38-17. ISET Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 14 | TIO6 | W | 0h | Tamper I/O 6 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 13 | TIO5 | W | 0h | Tamper I/O 5 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 12 | TIO4 | W | 0h | Tamper I/O 4 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 11 | TIO3 | W | 0h | Tamper I/O 3 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 10 | TIO2 | W | 0h | Tamper I/O 2 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 9 | TIO1 | W | 0h | Tamper I/O 1 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 8 | TIO0 | W | 0h | Tamper I/O 0 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 7 | TSEVT | W | 0h | Time stamp event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 6 | RT2PS | W | 0h | RTC prescale timer 2<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 5 | RT1PS | W | 0h | Set Prescaler-1 interrupt<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 4 | RT0PS | W | 0h | Set Prescaler-0 interrupt<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 3 | RTCA2 | W | 0h | Set Alarm-2 interrupt<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 2 | RTCA1 | W | 0h | Set Alarm-1 interrupt<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 1 | RTCTEV | W | 0h | Set Time-Event interrupt<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |
| 0 | RTCRDY | W | 0h | Set RTC-Ready interrupt<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.12 ICLR (Offset = 1048h) [Reset = 00000000h]

ICLR is shown in Figure 38-16 and described in Table 38-18.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

**Figure 38-16. ICLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TIO15 | TIO14 | TIO13 | TIO12 | TIO11 | TIO10 | TIO9 | TIO8 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TIO7 | TIO6 | TIO5 | TIO4 | TIO3 | TIO2 | TIO1 | TIO0 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TSEVT | RT2PS | RT1PS | RT0PS | RTCA2 | RTCA1 | RTCTEV | RTCRDY |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

**Table 38-18. ICLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | RESERVED | R | 0h | |
| 23 | TIO15 | W | 0h | Tamper I/O 15 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 22 | TIO14 | W | 0h | Tamper I/O 14 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 21 | TIO13 | W | 0h | Tamper I/O 13 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 20 | TIO12 | W | 0h | Tamper I/O 12 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 19 | TIO11 | W | 0h | Tamper I/O 11 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 18 | TIO10 | W | 0h | Tamper I/O 10 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 17 | TIO9 | W | 0h | Tamper I/O 9 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 16 | TIO8 | W | 0h | Tamper I/O 8 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 15 | TIO7 | W | 0h | Tamper I/O 7 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 14 | TIO6 | W | 0h | Tamper I/O 6 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |



Copyright © 2026 Texas Instruments Incorporated

**Table 38-18. ICLR Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 13 | TIO5 | W | 0h | Tamper I/O 5 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 12 | TIO4 | W | 0h | Tamper I/O 4 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 11 | TIO3 | W | 0h | Tamper I/O 3 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 10 | TIO2 | W | 0h | Tamper I/O 2 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 9 | TIO1 | W | 0h | Tamper I/O 1 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 8 | TIO0 | W | 0h | Tamper I/O 0 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 7 | TSEVT | W | 0h | Time stamp event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 6 | RT2PS | W | 0h | RTC prescale timer 2<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 5 | RT1PS | W | 0h | Clear Prescaler-1 interrupt<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 4 | RT0PS | W | 0h | Clear Prescaler-0 interrupt<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 3 | RTCA2 | W | 0h | Clear Alarm-2 interrupt<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 2 | RTCA1 | W | 0h | Clear Alarm-1 interrupt<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 1 | RTCTEV | W | 0h | Clear Time-Event interrupt<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |
| 0 | RTCRDY | W | 0h | Clear RTC-Ready interrupt<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.13 IIDX (Offset = 1050h) [Reset = 00000000h]

IIDX is shown in Figure 38-17 and described in Table 38-19.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. Value 0x00 means no event pending. Interrupt 1 is the highest priority, IIDX next highest, 4, 8, … IIDX^31 is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred. On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in [RIS] and [MIS] are cleared as well. After a read from the CPU (not from the debug interface), the register is updated with the next highest priority interrupt, if none are pending, then it should display 0x0.

**Figure 38-17. IIDX**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | STAT | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | |

**Table 38-19. IIDX Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | STAT | R | 0h | Interrupt index status<br>00h = No interrupt pending<br>01h = RTC ready<br>02h = RTC time event<br>03h = RTC alarm 1<br>04h = RTC alarm 2<br>05h = RTC prescale timer 0<br>06h = RTC prescale timer 1<br>07h = RTC prescale timer 2<br>08h = Time stamp event<br>09h = Tamper I/O 0 event<br>0Ah = Tamper I/O 1 event<br>0Bh = Tamper I/O 2 event<br>0Ch = Tamper I/O 3 event<br>0Dh = Tamper I/O 4 event<br>0Eh = Tamper I/O 5 event<br>0Fh = Tamper I/O 6 event<br>10h = Tamper I/O 7 event<br>11h = Tamper I/O 8 event<br>12h = Tamper I/O 9 event<br>13h = Tamper I/O 10 event<br>14h = Tamper I/O 11 event<br>15h = Tamper I/O 12 event<br>16h = Tamper I/O 13 event<br>17h = Tamper I/O 14 event<br>18h = Tamper I/O 15 event |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.14 IMASK (Offset = 1054h) [Reset = 00000000h]

IMASK is shown in Figure 38-18 and described in Table 38-20.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is unmasked. Unmasking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

**Figure 38-18. IMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TIO15 | TIO14 | TIO13 | TIO12 | TIO11 | TIO10 | TIO9 | TIO8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TIO7 | TIO6 | TIO5 | TIO4 | TIO3 | TIO2 | TIO1 | TIO0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TSEVT | RT2PS | RT1PS | RT0PS | RTCA2 | RTCA1 | RTCTEV | RTCRDY |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

**Table 38-20. IMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | RESERVED | R | 0h | |
| 23 | TIO15 | R/W | 0h | Tamper I/O 15 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 22 | TIO14 | R/W | 0h | Tamper I/O 14 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 21 | TIO13 | R/W | 0h | Tamper I/O 13 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 20 | TIO12 | R/W | 0h | Tamper I/O 12 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 19 | TIO11 | R/W | 0h | Tamper I/O 11 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 18 | TIO10 | R/W | 0h | Tamper I/O 10 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 17 | TIO9 | R/W | 0h | Tamper I/O 9 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 16 | TIO8 | R/W | 0h | Tamper I/O 8 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 15 | TIO7 | R/W | 0h | Tamper I/O 7 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 14 | TIO6 | R/W | 0h | Tamper I/O 6 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |



Copyright © 2026 Texas Instruments Incorporated

**Table 38-20. IMASK Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 13 | TIO5 | R/W | 0h | Tamper I/O 5 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 12 | TIO4 | R/W | 0h | Tamper I/O 4 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 11 | TIO3 | R/W | 0h | Tamper I/O 3 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 10 | TIO2 | R/W | 0h | Tamper I/O 2 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 9 | TIO1 | R/W | 0h | Tamper I/O 1 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 8 | TIO0 | R/W | 0h | Tamper I/O 0 event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 7 | TSEVT | R/W | 0h | Time stamp event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 6 | RT2PS | R/W | 0h | RTC prescale timer 2<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 5 | RT1PS | R/W | 0h | RTC prescale timer 1<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 4 | RT0PS | R/W | 0h | RTC prescale timer 0<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 3 | RTCA2 | R/W | 0h | RTC alarm 2<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 2 | RTCA1 | R/W | 0h | RTC alarm 1<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 1 | RTCTEV | R/W | 0h | RTC time event<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |
| 0 | RTCRDY | R/W | 0h | RTC ready<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.15 RIS (Offset = 1058h) [Reset = 00000000h]

RIS is shown in Figure 38-19 and described in Table 38-21.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Figure 38-19. RIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TIO15 | TIO14 | TIO13 | TIO12 | TIO11 | TIO10 | TIO9 | TIO8 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TIO7 | TIO6 | TIO5 | TIO4 | TIO3 | TIO2 | TIO1 | TIO0 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TSEVT | RT2PS | RT1PS | RT0PS | RTCA2 | RTCA1 | RTCTEV | RTCRDY |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 38-21. RIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | RESERVED | R | 0h | |
| 23 | TIO15 | R | 0h | Tamper I/O 15 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 22 | TIO14 | R | 0h | Tamper I/O 14 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 21 | TIO13 | R | 0h | Tamper I/O 13 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 20 | TIO12 | R | 0h | Tamper I/O 12 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 19 | TIO11 | R | 0h | Tamper I/O 11 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 18 | TIO10 | R | 0h | Tamper I/O 10 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 17 | TIO9 | R | 0h | Tamper I/O 9 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 16 | TIO8 | R | 0h | Tamper I/O 8 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 15 | TIO7 | R | 0h | Tamper I/O 7 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

**Table 38-21. RIS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 14 | TIO6 | R | 0h | Tamper I/O 6 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 13 | TIO5 | R | 0h | Tamper I/O 5 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 12 | TIO4 | R | 0h | Tamper I/O 4 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 11 | TIO3 | R | 0h | Tamper I/O 3 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 10 | TIO2 | R | 0h | Tamper I/O 2 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 9 | TIO1 | R | 0h | Tamper I/O 1 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 8 | TIO0 | R | 0h | Tamper I/O 0 event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 7 | TSEVT | R | 0h | Time stamp event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 6 | RT2PS | R | 0h | RTC prescale timer 2<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 5 | RT1PS | R | 0h | RTC prescale timer 1<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 4 | RT0PS | R | 0h | RTC prescale timer 0<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 3 | RTCA2 | R | 0h | RTC alarm 2<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 2 | RTCA1 | R | 0h | RTC alarm 1<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 1 | RTCTEV | R | 0h | RTC time event<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |
| 0 | RTCRDY | R | 0h | RTC ready<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.16 MIS (Offset = 105Ch) [Reset = 00000000h]

MIS is shown in Figure 38-20 and described in Table 38-22.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 38-20. MIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TIO15 | TIO14 | TIO13 | TIO12 | TIO11 | TIO10 | TIO9 | TIO8 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TIO7 | TIO6 | TIO5 | TIO4 | TIO3 | TIO2 | TIO1 | TIO0 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TSEVT | RT2PS | RT1PS | RT0PS | RTCA2 | RTCA1 | RTCTEV | RTCRDY |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 38-22. MIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | RESERVED | R | 0h | |
| 23 | TIO15 | R | 0h | Tamper I/O 15 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 22 | TIO14 | R | 0h | Tamper I/O 14 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 21 | TIO13 | R | 0h | Tamper I/O 13 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 20 | TIO12 | R | 0h | Tamper I/O 12 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 19 | TIO11 | R | 0h | Tamper I/O 11 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 18 | TIO10 | R | 0h | Tamper I/O 10 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 17 | TIO9 | R | 0h | Tamper I/O 9 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 16 | TIO8 | R | 0h | Tamper I/O 8 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 15 | TIO7 | R | 0h | Tamper I/O 7 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 14 | TIO6 | R | 0h | Tamper I/O 6 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

**Table 38-22. MIS Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 13 | TIO5 | R | 0h | Tamper I/O 5 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 12 | TIO4 | R | 0h | Tamper I/O 4 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 11 | TIO3 | R | 0h | Tamper I/O 3 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 10 | TIO2 | R | 0h | Tamper I/O 2 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 9 | TIO1 | R | 0h | Tamper I/O 1 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 8 | TIO0 | R | 0h | Tamper I/O 0 event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 7 | TSEVT | R | 0h | Time stamp event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 6 | RT2PS | R | 0h | RTC prescale timer 2<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 5 | RT1PS | R | 0h | RTC prescale timer 1<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 4 | RT0PS | R | 0h | RTC prescale timer 0<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 3 | RTCA2 | R | 0h | RTC alarm 2<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 2 | RTCA1 | R | 0h | RTC alarm 1<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 1 | RTCTEV | R | 0h | RTC time event<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |
| 0 | RTCRDY | R | 0h | RTC ready<br>0h = Interrupt did not occur or is masked out<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.17 ISET (Offset = 1060h) [Reset = 00000000h]

ISET is shown in Figure 38-21 and described in Table 38-23.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 38-21. ISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TIO15 | TIO14 | TIO13 | TIO12 | TIO11 | TIO10 | TIO9 | TIO8 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TIO7 | TIO6 | TIO5 | TIO4 | TIO3 | TIO2 | TIO1 | TIO0 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TSEVT | RT2PS | RT1PS | RT0PS | RTCA2 | RTCA1 | RTCTEV | RTCRDY |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

**Table 38-23. ISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | RESERVED | R | 0h | |
| 23 | TIO15 | W | 0h | Tamper I/O 15 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 22 | TIO14 | W | 0h | Tamper I/O 14 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 21 | TIO13 | W | 0h | Tamper I/O 13 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 20 | TIO12 | W | 0h | Tamper I/O 12 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 19 | TIO11 | W | 0h | Tamper I/O 11 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 18 | TIO10 | W | 0h | Tamper I/O 10 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 17 | TIO9 | W | 0h | Tamper I/O 9 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 16 | TIO8 | W | 0h | Tamper I/O 8 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 15 | TIO7 | W | 0h | Tamper I/O 7 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |



Copyright © 2026 Texas Instruments Incorporated

**Table 38-23. ISET Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 14 | TIO6 | W | 0h | Tamper I/O 6 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 13 | TIO5 | W | 0h | Tamper I/O 5 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 12 | TIO4 | W | 0h | Tamper I/O 4 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 11 | TIO3 | W | 0h | Tamper I/O 3 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 10 | TIO2 | W | 0h | Tamper I/O 2 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 9 | TIO1 | W | 0h | Tamper I/O 1 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 8 | TIO0 | W | 0h | Tamper I/O 0 event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 7 | TSEVT | W | 0h | Time stamp event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 6 | RT2PS | W | 0h | RTC prescale timer 2<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 5 | RT1PS | W | 0h | RTC prescale timer 1<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 4 | RT0PS | W | 0h | RTC prescale timer 0<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 3 | RTCA2 | W | 0h | RTC alarm 2<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 2 | RTCA1 | W | 0h | RTC alarm 1<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 1 | RTCTEV | W | 0h | RTC time event<br>0h = Writing 0 has no effect<br>1h = Set interrupt |
| 0 | RTCRDY | W | 0h | RTC ready<br>0h = Writing 0 has no effect<br>1h = Set interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.18 ICLR (Offset = 1064h) [Reset = 00000000h]

ICLR is shown in Figure 38-22 and described in Table 38-24.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

**Figure 38-22. ICLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TIO15 | TIO14 | TIO13 | TIO12 | TIO11 | TIO10 | TIO9 | TIO8 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TIO7 | TIO6 | TIO5 | TIO4 | TIO3 | TIO2 | TIO1 | TIO0 |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TSEVT | RT2PS | RT1PS | RT0PS | RTCA2 | RTCA1 | RTCTEV | RTCRDY |
| W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h | W-0h |

**Table 38-24. ICLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | RESERVED | R | 0h | |
| 23 | TIO15 | W | 0h | Tamper I/O 15 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 22 | TIO14 | W | 0h | Tamper I/O 14 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 21 | TIO13 | W | 0h | Tamper I/O 13 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 20 | TIO12 | W | 0h | Tamper I/O 12 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 19 | TIO11 | W | 0h | Tamper I/O 11 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 18 | TIO10 | W | 0h | Tamper I/O 10 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 17 | TIO9 | W | 0h | Tamper I/O 9 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 16 | TIO8 | W | 0h | Tamper I/O 8 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 15 | TIO7 | W | 0h | Tamper I/O 7 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 14 | TIO6 | W | 0h | Tamper I/O 6 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |



Copyright © 2026 Texas Instruments Incorporated

**Table 38-24. ICLR Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 13 | TIO5 | W | 0h | Tamper I/O 5 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 12 | TIO4 | W | 0h | Tamper I/O 4 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 11 | TIO3 | W | 0h | Tamper I/O 3 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 10 | TIO2 | W | 0h | Tamper I/O 2 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 9 | TIO1 | W | 0h | Tamper I/O 1 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 8 | TIO0 | W | 0h | Tamper I/O 0 event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 7 | TSEVT | W | 0h | Time stamp event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 6 | RT2PS | W | 0h | RTC prescale timer 2<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 5 | RT1PS | W | 0h | RTC prescale timer 1<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 4 | RT0PS | W | 0h | RTC prescale timer 0<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 3 | RTCA2 | W | 0h | RTC alarm 2<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 2 | RTCA1 | W | 0h | RTC alarm 1<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 1 | RTCTEV | W | 0h | RTC time event<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |
| 0 | RTCRDY | W | 0h | RTC ready<br>0h = Writing 0 has no effect<br>1h = Clear interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.19 EVT_MODE (Offset = 10E0h) [Reset = 00000000h]

EVT_MODE is shown in Figure 38-23 and described in Table 38-25.

Return to the Summary Table.

Event mode register. It is used to select whether each line is disabled, in software mode (software clears the RIS) or in hardware mode (hardware clears the RIS)

**Figure 38-23. EVT_MODE**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | EVT1_CFG | | EVT0_CFG | |
| R-0h | | | | R-0h | | R-0h | |

**Table 38-25. EVT_MODE Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3-2 | EVT1_CFG | R | 0h | Event line mode 1 select<br>0h = The interrupt or event line is disabled.<br>1h = The interrupt or event line is in software mode. The software ISR clears the associated RIS flag.<br>2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag. |
| 1-0 | EVT0_CFG | R | 0h | Event line mode 0 select<br>0h = The interrupt or event line is disabled.<br>1h = The interrupt or event line is in software mode. The software ISR clears the associated RIS flag.<br>2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag. |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.20 DESC (Offset = 10FCh) [Reset = 09118010h]

DESC is shown in Figure 38-24 and described in Table 38-26.

Return to the Summary Table.

RTC Descriptor Register

**Figure 38-24. DESC**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MODULEID | | | | | | | | | | | | | | | |
| R-911h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| FEATUREVER | | | | INSTNUM | | | | MAJREV | | | | MINREV | | | |
| R-8h | | | | R-0h | | | | R-1h | | | | R-0h | | | |

**Table 38-26. DESC Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | MODULEID | R | 911h | Module identifier. This ID is unique for each module. 0x0911 = Module ID of the RTC Module<br>0000h = Minimum value<br>FFFFh = Maximum value |
| 15-12 | FEATUREVER | R | 8h | Feature set of this module. Differentiates the complexity of the actually instantiated module if there are differences.<br>0h = Minimum value<br>Fh = Maximum value |
| 11-8 | INSTNUM | R | 0h | Instantiated version. Describes which instance of the module accessed.<br>0h = This is the default, if there is only one instance - like for SSIM |
| 7-4 | MAJREV | R | 1h | Major revision. This number holds the module revision and is incremented by the module developers. n = Major version (see device-specific data sheet)<br>0h = Minimum value<br>Fh = Maximum value |
| 3-0 | MINREV | R | 0h | Minor revision. This number holds the module revision and is incremented by the module developers. n = Minor module revision (see device-specific data sheet)<br>0h = Minimum value<br>Fh = Maximum value |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.21 CLKCTL (Offset = 1100h) [Reset = 00000000h]

CLKCTL is shown in Figure 38-25 and described in Table 38-27.

Return to the Summary Table.

RTC Clock Control Register

**Figure 38-25. CLKCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MODCLKEN | RESERVED | | | | | | |
| R/W-0h | R-0h | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |

**Table 38-27. CLKCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | MODCLKEN | R/W | 0h | This bit enables the supply of the 32kHz clock to the RTC. It will not power-up the 32kHz crystal oscillator this needs to be done in the Clock System Module.<br>0h = 32kHz clock is not supplied to the RTC.<br>1h = 32kHz clock is supplied to the RTC. |
| 30-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.22 DBGCTL (Offset = 1104h) [Reset = 00000000h]

DBGCTL is shown in Figure 38-26 and described in Table 38-28.

Return to the Summary Table.

RTC Module Debug Control Register

**Figure 38-26. DBGCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | DBGINT | DBGRUN |
| R-0h | | | | | | R/W-0h | R/W-0h |

**Table 38-28. DBGCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | |
| 1 | DBGINT | R/W | 0h | Debug Interrupt Enable.<br>0h = Interrupts of the module will not be captured anymore if CPU is in debug state. Which means no update to the RTCRIS, RTCMISC and RTCIIDX register.<br>1h = Interrupts are enabled in debug mode. Interrupt requests are signaled to the interrupt controller. If the flags are required by software (polling mode) the DGBINT bit need to be set to 1. |
| 0 | DBGRUN | R/W | 0h | Debug Run.<br>0h = Counter is halted if CPU is in debug state.<br>1h = Continue to operate normally ignoring the debug state of the CPU. |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.23 CTL (Offset = 1108h) [Reset = 00000000h]

CTL is shown in Figure 38-27 and described in Table 38-29.

Return to the Summary Table.

RTC Control Register

**Figure 38-27. CTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RTCBCD | RESERVED | | | | | RTCTEVTX | |
| R/W-0h | R-0h | | | | | R/W-0h | |

**Table 38-29. CTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7 | RTCBCD | R/W | 0h | Real-time clock BCD select. Selects BCD counting for real-time clock.<br>0h = Binary code selected<br>1h = Binary coded decimal (BCD) code selected |
| 6-2 | RESERVED | R | 0h | |
| 1-0 | RTCTEVTX | R/W | 0h | Real-time clock time event.<br>0h = Minute changed.<br>1h = Hour changed.<br>2h = Every day at midnight (00:00).<br>3h = Every day at noon (12:00). |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.24 STA (Offset = 110Ch) [Reset = 00000000h]

STA is shown in Figure 38-28 and described in Table 38-30.

Return to the Summary Table.

RTC Status Register

**Figure 38-28. STA**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | RTCTCOK | RTCTCRDY | RTCRDY |
| R-0h | | | | | R-0h | R-0h | R-0h |

**Table 38-30. STA Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-3 | RESERVED | R | 0h | |
| 2 | RTCTCOK | R | 0h | Real-time clock temperature compensation write OK. This is a read-only bit that indicates if the write to RTCTCMP is successful or not.<br>0h = Write to RTCTCMPx is unsuccessful<br>1h = Write to RTCTCMPx is successful |
| 1 | RTCTCRDY | R | 0h | Real-time clock temperature compensation ready. This is a read only bit that indicates when the RTCTCMPx can be written. Write to RTCTCMPx should be avoided when RTCTCRDY is reset.<br>0h = Real-time clock temperature compensation not ready<br>1h = Real-time clock temperature compensation ready |
| 0 | RTCRDY | R | 0h | Real-time clock ready. This bit indicates when the real-time clock time values are safe for reading.<br>0h = RTC time values in transition<br>1h = RTC time values safe for reading |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.25 CAL (Offset = 1110h) [Reset = 00000000h]

CAL is shown in Figure 38-29 and described in Table 38-31.

Return to the Summary Table.

RTC Clock Offset Calibration Register

**Figure 38-29. CAL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | RTCCALFX | |
| R-0h | | | | | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RTCOCALS | RESERVED | | | | | | |
| R/W-0h | R-0h | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RTCOCALX | | | | | | | |
| R/W-0h | | | | | | | |

**Table 38-31. CAL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-18 | RESERVED | R | 0h | |
| 17-16 | RTCCALFX | R/W | 0h | Real-time clock calibration frequency. Selects frequency output to RTC_OUT pin for calibration measurement. The corresponding port must be configured for the peripheral module function.<br>0h = No frequency output to RTC_OUT pin<br>1h = 512Hz<br>2h = 256Hz<br>3h = 1Hz |
| 15 | RTCOCALS | R/W | 0h | Real-time clock offset error calibration sign. This bit decides the sign of offset error calibration.<br>0h = Down calibration. Frequency adjusted down.<br>1h = Up calibration. Frequency adjusted up. |
| 14-8 | RESERVED | R | 0h | |
| 7-0 | RTCOCALX | R/W | 0h | Real-time clock offset error calibration. Each LSB represents approximately +1ppm (RTCOCALXS = 1) or -1ppm (RTCOCALXS = 0) adjustment in frequency. Maximum effective calibration value is +/-240ppm. Excess values written above +/-240ppm will be ignored by hardware.<br>0h = Minimum effective calibration value.<br>FFh = Maximum effective calibration value is +/-240ppm. Excess values written above +/-240ppm will be ignored by hardware. |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.26 TCMP (Offset = 1114h) [Reset = 00000000h]

TCMP is shown in Figure 38-30 and described in Table 38-32.

Return to the Summary Table.

RTC Temperature Compensation Register

**Figure 38-30. TCMP**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RTCTCMPS | RESERVED | | | | | | |
| R/W-0h | R-0h | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RTCTCMPX | | | | | | | |
| R/W-0h | | | | | | | |

**Table 38-32. TCMP Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15 | RTCTCMPS | R/W | 0h | Real-time clock temperature compensation sign. This bit decides the sign of temperature compensation.<br>0h = Down calibration. Frequency adjusted down.<br>1h = Up calibration. Frequency adjusted up. |
| 14-8 | RESERVED | R | 0h | |
| 7-0 | RTCTCMPX | R/W | 0h | Real-time clock temperature compensation. Value written into this register is used for temperature compensation of RTC. Each LSB represents approximately +1ppm (RTCTCMPS = 1) or -1ppm (RTCTCMPS = 0) adjustment in frequency. Maximum effective calibration value is +/-240ppm. Excess values written above +/-240ppm are ignored by hardware. Reading from RTCTCMP register at any time returns the cumulative value which is the signed addition of RTCOCALx and RTCTCMPX values, and the updated sign bit (RTCTCMPS) of the addition result.<br>00h = Minimum value<br>FFh = Maximum value |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.27 SEC (Offset = 1118h) [Reset = 0000XXXXh]

SEC is shown in Figure 38-31 and described in Table 38-33.

Return to the Summary Table.

RTC Seconds Register - Calendar Mode With Binary / BCD Format

**Figure 38-31. SEC**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | SECHIGHBCD | | | SECLOWBCD | | | |
| R-0h | R/W-X | | | R/W-X | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | SECBIN | | | | | |
| R-0h | | R/W-X | | | | | |

**Table 38-33. SEC Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-15 | RESERVED | R | 0h | |
| 14-12 | SECHIGHBCD | R/W | X | Seconds BCD – high digit (0 to 5). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Minimum value<br>5h = Maximum value |
| 11-8 | SECLOWBCD | R/W | X | Seconds BCD – low digit (0 to 9). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Minimum value<br>9h = Maximum value |
| 7-6 | RESERVED | R | 0h | |
| 5-0 | SECBIN | R/W | X | Seconds Binary (0 to 59). If RTCBCD=1 write to these bits will be ignored and read give the value 0.<br>00h = Minimum value<br>3Bh = Maximum value |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.28 MIN (Offset = 111Ch) [Reset = 0000XXXXh]

MIN is shown in Figure 38-32 and described in Table 38-34.

Return to the Summary Table.

RTC Minutes Register - Calendar Mode With Binary / BCD Format

**Figure 38-32. MIN**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | MINHIGHBCD | | | MINLOWBCD | | | |
| R-0h | R/W-X | | | R/W-X | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | MINBIN | | | | | |
| R-0h | | R/W-X | | | | | |

**Table 38-34. MIN Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-15 | RESERVED | R | 0h | |
| 14-12 | MINHIGHBCD | R/W | X | Minutes BCD – high digit (0 to 5). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Minimum value<br>5h = Maximum value |
| 11-8 | MINLOWBCD | R/W | X | Minutes BCD – low digit (0 to 9). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Minimum value<br>9h = Maximum value |
| 7-6 | RESERVED | R | 0h | |
| 5-0 | MINBIN | R/W | X | Minutes Binary (0 to 59). If RTCBCD=1 write to these bits will be ignored and read give the value 0.<br>00h = Minimum value<br>3Bh = Maximum value |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.29 HOUR (Offset = 1120h) [Reset = 0000XXXXh]

HOUR is shown in Figure 38-33 and described in Table 38-35.

Return to the Summary Table.

RTC Hours Register - Calendar Mode With Binary / BCD Format

**Figure 38-33. HOUR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | HOURHIGHBCD | | HOURLOWBCD | | |
| R-0h | | | R/W-X | | R/W-X | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | HOURBIN | | | | |
| R-0h | | | R/W-X | | | | |

**Table 38-35. HOUR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-14 | RESERVED | R | 0h | |
| 13-12 | HOURHIGHBCD | R/W | X | Hours BCD – high digit (0 to 2). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Minimum value.<br>2h = Maximum value. |
| 11-8 | HOURLOWBCD | R/W | X | Hours BCD – low digit (0 to 9). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Minimum value.<br>9h = Maximum value. |
| 7-5 | RESERVED | R | 0h | |
| 4-0 | HOURBIN | R/W | X | Hours Binary (0 to 23). If RTCBCD=1 write to these bits will be ignored and read give the value 0.<br>00h = Minimum value.<br>17h = Maximum value. |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.30 DAY (Offset = 1124h) [Reset = 00XXXX0Xh]

DAY is shown in Figure 38-34 and described in Table 38-36.

Return to the Summary Table.

RTC Day of Week/Month Register - Calendar Mode With Binary / BCD Format

**Figure 38-34. DAY**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | DOMHIGHBCD | | DOMLOWBCD | | | |
| R-0h | | R/W-X | | R/W-X | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | DOMBIN | | | | |
| R-0h | | | R/W-X | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | DOW | | |
| R-0h | | | | | R/W-X | | |

**Table 38-36. DAY Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-22 | RESERVED | R | 0h | |
| 21-20 | DOMHIGHBCD | R/W | X | Day of month BCD – high digit (0 to 3). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Minimum value<br>3h = Maximum value |
| 19-16 | DOMLOWBCD | R/W | X | Day of month BCD – low digit (0 to 9). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Minimum value<br>9h = Maximum value |
| 15-13 | RESERVED | R | 0h | |
| 12-8 | DOMBIN | R/W | X | Day of month Binary (1 to 28, 29, 30, 31). If RTCBCD=1 write to these bits will be ignored and read give the value 0.<br>00h = Minimum value<br>1Fh = Maximum value |
| 7-3 | RESERVED | R | 0h | |
| 2-0 | DOW | R/W | X | Day of week (0 to 6). These bits are valid if RTCBCD=1 or RTCBCD=0.<br>0h = Minimum value<br>6h = Maximum value |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.31 MON (Offset = 1128h) [Reset = 0000XX0Xh]

MON is shown in Figure 38-35 and described in Table 38-37.

Return to the Summary Table.

RTC Month Register - Calendar Mode With Binary / BCD Format

**Figure 38-35. MON**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | MONHIGHBCD | MONLOWBCD | | | |
| R-0h | | | R/W-X | R/W-X | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | MONBIN | | | |
| R-0h | | | | R/W-X | | | |

**Table 38-37. MON Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-13 | RESERVED | R | 0h | |
| 12 | MONHIGHBCD | R/W | X | Month BCD – high digit (0 or 1). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Minimum value<br>1h = Maximum value |
| 11-8 | MONLOWBCD | R/W | X | Month BCD – low digit (0 to 9). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Minimum value<br>9h = Maximum value |
| 7-4 | RESERVED | R | 0h | |
| 3-0 | MONBIN | R/W | X | Month Binary (1 to 12). If RTCBCD=1 write to these bits will be ignored and read give the value 0.<br>0h = Minimum value<br>Ch = Maximum value |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.32 YEAR (Offset = 112Ch) [Reset = XXXX0XXXh]

YEAR is shown in Figure 38-36 and described in Table 38-38.

Return to the Summary Table.

RTC Year Register - Calendar Mode With Binary / BCD Format

**Figure 38-36. YEAR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | CENTHIGHBCD | | | CENTLOWBCD | | | |
| R-0h | R/W-X | | | R/W-X | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DECADEBCD | | | | YEARLOWESTBCD | | | |
| R/W-X | | | | R/W-X | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | YEARHIGHBIN | | | |
| R-0h | | | | R/W-X | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| YEARLOWBIN | | | | | | | |
| R/W-X | | | | | | | |

**Table 38-38. YEAR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | RESERVED | R | 0h | |
| 30-28 | CENTHIGHBCD | R/W | X | Century BCD – high digit (0 to 4). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Minimum value<br>4h = Maximum value |
| 27-24 | CENTLOWBCD | R/W | X | Century BCD – low digit (0 to 9). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Minimum value<br>9h = Maximum value |
| 23-20 | DECADEBCD | R/W | X | Decade BCD (0 to 9). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Minimum value<br>9h = Maximum value |
| 19-16 | YEARLOWESTBCD | R/W | X | Year BCD – lowest digit (0 to 9). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Minimum value<br>9h = Maximum value |
| 15-12 | RESERVED | R | 0h | |
| 11-8 | YEARHIGHBIN | R/W | X | Year Binary – high byte. Valid values for Year are 0 to 4095. If RTCBCD=1 write to these bits will be ignored and read give the value 0.<br>0h = Minimum value<br>Fh = Maximum value |
| 7-0 | YEARLOWBIN | R/W | X | Year Binary – low byte. Valid values for Year are 0 to 4095. If RTCBCD=1 write to these bits will be ignored and read give the value 0.<br>00h = Minimum value<br>FFh = Maximum value |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.33 A1MIN (Offset = 1130h) [Reset = 00000000h]

A1MIN is shown in Figure 38-37 and described in Table 38-39.

Return to the Summary Table.

RTC Minutes Alarm Register - Calendar Mode With Binary / BCD Format

**Figure 38-37. A1MIN**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| AMINAEBCD | AMINHIGHBCD | | | AMINLOWBCD | | | |
| R/W-0h | R/W-0h | | | R/W-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| AMINAEBIN | RESERVED | AMINBIN | | | | | |
| R/W-0h | R-0h | R/W-0h | | | | | |

**Table 38-39. A1MIN Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15 | AMINAEBCD | R/W | 0h | Alarm Minutes BCD enable. If RTCBCD=0 this bit is always 0. Write to this bit will be ignored.<br>0h = Alarm disabled<br>1h = Alarm enabled |
| 14-12 | AMINHIGHBCD | R/W | 0h | Alarm Minutes BCD – high digit (0 to 5). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Minimum value<br>5h = Maximum value |
| 11-8 | AMINLOWBCD | R/W | 0h | Alarm Minutes BCD – low digit (0 to 9). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Minimum value<br>9h = Maximum value |
| 7 | AMINAEBIN | R/W | 0h | Alarm Minutes Binary enable. If RTCBCD=1 this bit is always 0. Write to this bit will be ignored.<br>0h = Alarm disabled<br>1h = Alarm enabled |
| 6 | RESERVED | R | 0h | |
| 5-0 | AMINBIN | R/W | 0h | Alarm Minutes Binary (0 to 59). If RTCBCD=1 write to these bits will be ignored and read give the value 0.<br>00h = Minimum value<br>3Bh = Maximum value |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.34 A1HOUR (Offset = 1134h) [Reset = 00000000h]

A1HOUR is shown in Figure 38-38 and described in Table 38-40.

Return to the Summary Table.

RTC Hours Alarm Register - Calendar Mode With Binary / BCD Format

**Figure 38-38. A1HOUR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| AHOURAEBCD | RESERVED | AHOURHIGHBCD | | AHOURLOWBCD | | | |
| R/W-0h | R-0h | R/W-0h | | R/W-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| AHOURAEBIN | RESERVED | | AHOURBIN | | | | |
| R/W-0h | R-0h | | R/W-0h | | | | |

**Table 38-40. A1HOUR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15 | AHOURAEBCD | R/W | 0h | Alarm Hours BCD enable. If RTCBCD=0 this bit is always 0. Write to this bit will be ignored.<br>0h = Alarm disabled<br>1h = Alarm enabled |
| 14 | RESERVED | R | 0h | |
| 13-12 | AHOURHIGHBCD | R/W | 0h | Alarm Hours BCD – high digit (0 to 2). If RTCBCD=0 write to these bits will be ignored and read give the value 0..<br>0h = Minimum value<br>2h = Maximum value |
| 11-8 | AHOURLOWBCD | R/W | 0h | Alarm Hours BCD – low digit (0 to 9). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Minimum value<br>9h = Maximum value |
| 7 | AHOURAEBIN | R/W | 0h | Alarm Hours Binary enable. If RTCBCD=1 this bit is always 0. Write to this bit will be ignored.<br>0h = Alarm disabled<br>1h = Alarm enabled |
| 6-5 | RESERVED | R | 0h | |
| 4-0 | AHOURBIN | R/W | 0h | Alarm Hours Binary (0 to 23). If RTCBCD=1 write to these bits will be ignored and read give the value 0.<br>00h = Minimum value<br>17h = Maximum value |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.35 A1DAY (Offset = 1138h) [Reset = 00000000h]

A1DAY is shown in Figure 38-39 and described in Table 38-41.

Return to the Summary Table.

RTC Alarm Day Of Week / Month Register - Calendar Mode With Binary / BCD Format

**Figure 38-39. A1DAY**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| ADOMAEBCD | RESERVED | ADOMHIGHBCD | | ADOMLOWBCD | | | |
| R/W-0h | R-0h | R/W-0h | | R/W-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| ADOMAEBIN | RESERVED | | ADOMBIN | | | | |
| R/W-0h | R-0h | | R/W-0h | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ADOWAE | RESERVED | | | | ADOW | | |
| R/W-0h | R-0h | | | | R/W-0h | | |

**Table 38-41. A1DAY Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | RESERVED | R | 0h | |
| 23 | ADOMAEBCD | R/W | 0h | Alarm Day of month BCD enable. If RTCBCD=0 this bit is always 0. Write to this bit will be ignored.<br>0h = Alarm disabled<br>1h = Alarm enabled |
| 22 | RESERVED | R | 0h | |
| 21-20 | ADOMHIGHBCD | R/W | 0h | Alarm Day of month BCD – high digit (0 to 3). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Minimum value<br>3h = Maximum value |
| 19-16 | ADOMLOWBCD | R/W | 0h | Alarm Day of month BCD – low digit (0 to 9). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Minimum value<br>9h = Maximum value |
| 15 | ADOMAEBIN | R/W | 0h | Alarm Day of month Binary enable. If RTCBCD=1 this bit is always 0. Write to this bit will be ignored.<br>0h = Alarm disabled<br>1h = Alarm enabled |
| 14-13 | RESERVED | R | 0h | |
| 12-8 | ADOMBIN | R/W | 0h | Alarm Day of month Binary (1 to 28, 29, 30, 31) If RTCBCD=1 write to these bits will be ignored and read give the value 0.<br>00h = Minimum value<br>1Fh = Maximum value |
| 7 | ADOWAE | R/W | 0h | Alarm Day of week enable. This bit are valid if RTCBCD=1 or RTCBCD=0.<br>0h = Alarm disabled<br>1h = Alarm enabled |
| 6-3 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

**Table 38-41. A1DAY Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2-0 | ADOW | R/W | 0h | Alarm Day of week (0 to 6). These bits are valid if RTCBCD=1 or RTCBCD=0.<br>0h = Minimum value<br>6h = Maximum value |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.36 A2MIN (Offset = 113Ch) [Reset = 00000000h]

A2MIN is shown in Figure 38-40 and described in Table 38-42.

Return to the Summary Table.

RTC Minutes Alarm Register - Calendar Mode With Binary / BCD Format

**Figure 38-40. A2MIN**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| AMINAEBCD | AMINHIGHBCD | | | AMINLOWBCD | | | |
| R/W-0h | R/W-0h | | | R/W-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| AMINAEBIN | RESERVED | AMINBIN | | | | | |
| R/W-0h | R-0h | R/W-0h | | | | | |

**Table 38-42. A2MIN Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15 | AMINAEBCD | R/W | 0h | Alarm Minutes BCD enable. If RTCBCD=0 this bit is always 0. Write to this bit will be ignored.<br>0h = Alarm disabled<br>1h = Alarm enabled |
| 14-12 | AMINHIGHBCD | R/W | 0h | Alarm Minutes BCD – high digit (0 to 5). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Minimum value<br>5h = Maximum value |
| 11-8 | AMINLOWBCD | R/W | 0h | Alarm Minutes BCD – low digit (0 to 9). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Minimum value<br>9h = Maximum value |
| 7 | AMINAEBIN | R/W | 0h | Alarm Minutes Binary enable. If RTCBCD=1 this bit is always 0. Write to this bit will be ignored.<br>0h = Alarm disabled<br>1h = Alarm enabled |
| 6 | RESERVED | R | 0h | |
| 5-0 | AMINBIN | R/W | 0h | Alarm Minutes Binary (0 to 59). If RTCBCD=1 write to these bits will be ignored and read give the value 0.<br>00h = Minimum value<br>3Bh = Maximum value |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.37 A2HOUR (Offset = 1140h) [Reset = 00000000h]

A2HOUR is shown in Figure 38-41 and described in Table 38-43.

Return to the Summary Table.

RTC Hours Alarm Register - Calendar Mode With Binary / BCD Format

**Figure 38-41. A2HOUR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| AHOURAEBCD | RESERVED | AHOURHIGHBCD | | AHOURLOWBCD | | | |
| R/W-0h | R-0h | R/W-0h | | R/W-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| AHOURAEBIN | RESERVED | | AHOURBIN | | | | |
| R/W-0h | R-0h | | R/W-0h | | | | |

**Table 38-43. A2HOUR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | |
| 15 | AHOURAEBCD | R/W | 0h | Alarm Hours BCD enable. If RTCBCD=0 this bit is always 0. Write to this bit will be ignored.<br>0h = Alarm disabled<br>1h = Alarm enabled |
| 14 | RESERVED | R | 0h | |
| 13-12 | AHOURHIGHBCD | R/W | 0h | Alarm Hours BCD – high digit (0 to 2). If RTCBCD=0 write to these bits will be ignored and read give the value 0..<br>0h = Minimum value<br>2h = Maximum value |
| 11-8 | AHOURLOWBCD | R/W | 0h | Alarm Hours BCD – low digit (0 to 9). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Minimum value<br>9h = Maximum value |
| 7 | AHOURAEBIN | R/W | 0h | Alarm Hours Binary enable. If RTCBCD=1 this bit is always 0. Write to this bit will be ignored.<br>0h = Alarm disabled<br>1h = Alarm enabled |
| 6-5 | RESERVED | R | 0h | |
| 4-0 | AHOURBIN | R/W | 0h | Alarm Hours Binary (0 to 23). If RTCBCD=1 write to these bits will be ignored and read give the value 0.<br>00h = Minimum value<br>17h = Maximum value |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.38 A2DAY (Offset = 1144h) [Reset = 00000000h]

A2DAY is shown in Figure 38-42 and described in Table 38-44.

Return to the Summary Table.

RTC Alarm Day Of Week / Month Register - Calendar Mode With Binary / BCD Format

**Figure 38-42. A2DAY**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| ADOMAEBCD | RESERVED | ADOMHIGHBCD | | ADOMLOWBCD | | | |
| R/W-0h | R-0h | R/W-0h | | R/W-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| ADOMAEBIN | RESERVED | | ADOMBIN | | | | |
| R/W-0h | R-0h | | R/W-0h | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ADOWAE | RESERVED | | | | ADOW | | |
| R/W-0h | R-0h | | | | R/W-0h | | |

**Table 38-44. A2DAY Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | RESERVED | R | 0h | |
| 23 | ADOMAEBCD | R/W | 0h | Alarm Day of month BCD enable. If RTCBCD=0 this bit is always 0. Write to this bit will be ignored.<br>0h = Alarm disabled<br>1h = Alarm enabled |
| 22 | RESERVED | R | 0h | |
| 21-20 | ADOMHIGHBCD | R/W | 0h | Alarm Day of month BCD – high digit (0 to 3). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Minimum value<br>3h = Maximum value |
| 19-16 | ADOMLOWBCD | R/W | 0h | Alarm Day of month BCD – low digit (0 to 9). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Minimum value<br>9h = Maximum value |
| 15 | ADOMAEBIN | R/W | 0h | Alarm Day of month Binary enable. If RTCBCD=1 this bit is always 0. Write to this bit will be ignored.<br>0h = Alarm disabled<br>1h = Alarm enabled |
| 14-13 | RESERVED | R | 0h | |
| 12-8 | ADOMBIN | R/W | 0h | Alarm Day of month Binary (1 to 28, 29, 30, 31) If RTCBCD=1 write to these bits will be ignored and read give the value 0.<br>00h = Minimum value<br>1Fh = Maximum value |
| 7 | ADOWAE | R/W | 0h | Alarm Day of week enable. This bit are valid if RTCBCD=1 or RTCBCD=0.<br>0h = Alarm disabled<br>1h = Alarm enabled |
| 6-3 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

**Table 38-44. A2DAY Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 2-0 | ADOW | R/W | 0h | Alarm Day of week (0 to 6). These bits are valid if RTCBCD=1 or RTCBCD=0.<br>0h = Minimum value<br>6h = Maximum value |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.39 PSCTL (Offset = 1148h) [Reset = 00000008h]

PSCTL is shown in Figure 38-43 and described in Table 38-45.

Return to the Summary Table.

RTC Prescale Timer 0/1 Control Register

**Figure 38-43. PSCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | RT1IP | | | RESERVED | |
| R-0h | | | R/W-0h | | | R-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | RT0IP | | | RESERVED | |
| R-0h | | | R/W-2h | | | R-0h | |

**Table 38-45. PSCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-21 | RESERVED | R | 0h | |
| 20-18 | RT1IP | R/W | 0h | Prescale timer 1 interrupt interval<br>0h = Divide by 2 - 15.6 millisecond interval<br>1h = Divide by 4 - 31.2 millisecond interval<br>2h = Divide by 8 - 62.5 millisecond interval<br>3h = Divide by 16 - 125 millisecond interval<br>4h = Divide by 32 - 250 millisecond interval<br>5h = Divide by 64 - 500 millisecond interval<br>6h = Divide by 128 - 1 second interval<br>7h = Divide by 256 - 2 second interval |
| 17-5 | RESERVED | R | 0h | |
| 4-2 | RT0IP | R/W | 2h | Prescale timer 0 interrupt interval<br>2h = Divide by 8 - 244 microsecond interval<br>3h = Divide by 16 - 488 microsecond interval<br>4h = Divide by 32 - 976 microsecond interval<br>5h = Divide by 64 - 1.95 millisecond interval<br>6h = Divide by 128 - 3.90 millisecond interval<br>7h = Divide by 256 - 7.81 millisecond interval |
| 1-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.40 EXTPSCTL (Offset = 114Ch) [Reset = 00000000h]

EXTPSCTL is shown in Figure 38-44 and described in Table 38-46.

Return to the Summary Table.

Extended Prescale Timer Control Register

**Figure 38-44. EXTPSCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | RT2PS | | RESERVED | |
| R-0h | | | | | | | | | | | | R/W-0h | | R-0h | |

**Table 38-46. EXTPSCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3-2 | RT2PS | R/W | 0h | Prescale timer 2 interrupt interval<br>0h = Interval every 4 second<br>1h = Interval every 8 second<br>2h = Interval every 16 second |
| 1-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.41 TSSEC (Offset = 1150h) [Reset = 00000000h]

TSSEC is shown in Figure 38-45 and described in Table 38-47.

Return to the Summary Table.

RTC Second Time Stamp Capture - Calendar Mode With Binary / BCD Format

**Figure 38-45. TSSEC**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | SECHIGHBCD | | | SECLOWBCD | | | |
| R-0h | R-0h | | | R-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | SECBIN | | | | | |
| R-0h | | R-0h | | | | | |

**Table 38-47. TSSEC Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-15 | RESERVED | R | 0h | |
| 14-12 | SECHIGHBCD | R | 0h | Time Stamp Seconds BCD – high digit (0 to 5). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>5h = Highest possible value |
| 11-8 | SECLOWBCD | R | 0h | Time Stamp Seconds BCD – low digit (0 to 9). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>9h = Highest possible value |
| 7-6 | RESERVED | R | 0h | |
| 5-0 | SECBIN | R | 0h | Time Stamp Second Binary (0 to 59). If RTCBCD=1 write to these bits will be ignored and read give the value 0.<br>00h = Smallest value<br>3Bh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.42 TSMIN (Offset = 1154h) [Reset = 00000000h]

TSMIN is shown in Figure 38-46 and described in Table 38-48.

Return to the Summary Table.

RTC Minutes Time Stamp Capture - Calendar Mode With Binary / BCD Format

**Figure 38-46. TSMIN**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | MINHIGHBCD | | | MINLOWBCD | | | |
| R-0h | R-0h | | | R-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | MINBIN | | | | | |
| R-0h | | R-0h | | | | | |

**Table 38-48. TSMIN Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-15 | RESERVED | R | 0h | |
| 14-12 | MINHIGHBCD | R | 0h | Time Stamp Minutes BCD – high digit (0 to 5). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>5h = Highest possible value |
| 11-8 | MINLOWBCD | R | 0h | Time Stamp Minutes BCD – low digit (0 to 9). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>9h = Highest possible value |
| 7-6 | RESERVED | R | 0h | |
| 5-0 | MINBIN | R | 0h | Time Stamp Minutes Binary (0 to 59). If RTCBCD=1 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>3Bh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.43 TSHOUR (Offset = 1158h) [Reset = 00000000h]

TSHOUR is shown in Figure 38-47 and described in Table 38-49.

Return to the Summary Table.

RTC Hours Time Stamp Capture - Calendar Mode With Binary / BCD Format

**Figure 38-47. TSHOUR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | HOURHIGHBCD | | HOURLOWBCD | | |
| R-0h | | | R-0h | | R-0h | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | HOURBIN | | | | |
| R-0h | | | R-0h | | | | |

**Table 38-49. TSHOUR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-14 | RESERVED | R | 0h | |
| 13-12 | HOURHIGHBCD | R | 0h | Time Stamp Hours BCD – high digit (0 to 2). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>00h = Smallest value<br>02h = Highest possible value |
| 11-8 | HOURLOWBCD | R | 0h | Time Stamp Hours BCD – low digit (0 to 9). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>00h = Smallest value<br>09h = Highest possible value |
| 7-5 | RESERVED | R | 0h | |
| 4-0 | HOURBIN | R | 0h | Time Stamp Hours Binary (0 to 23). If RTCBCD=1 write to these bits will be ignored and read give the value 0.<br>00h = Smallest value<br>17h = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.44 TSDAY (Offset = 115Ch) [Reset = 00000000h]

TSDAY is shown in Figure 38-48 and described in Table 38-50.

Return to the Summary Table.

RTC Day Of Week / Month Time Stamp Capture - Calendar Mode With Binary / BCD Format

**Figure 38-48. TSDAY**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | DOMHIGHBCD | | DOMLOWBCD | | | |
| R-0h | | R-0h | | R-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | DOMBIN | | | | |
| R-0h | | | R-0h | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | DOW | | |
| R-0h | | | | | R-0h | | |

**Table 38-50. TSDAY Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-22 | RESERVED | R | 0h | |
| 21-20 | DOMHIGHBCD | R | 0h | Time Stamp Day of month BCD – high digit (0 to 3). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>3h = Highest possible value |
| 19-16 | DOMLOWBCD | R | 0h | Time Stamp Day of month BCD – low digit (0 to 9). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>9h = Highest possible value |
| 15-13 | RESERVED | R | 0h | |
| 12-8 | DOMBIN | R | 0h | Time Stamp Day of month Binary (1 to 28, 29, 30, 31) If RTCBCD=1 write to these bits will be ignored and read give the value 0.<br>00h = Smallest value<br>1Fh = Highest possible value |
| 7-3 | RESERVED | R | 0h | |
| 2-0 | DOW | R | 0h | Time Stamp Day of week (0 to 6). These bits are valid if RTCBCD=1 or RTCBCD=0.<br>0h = Smallest value<br>6h = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.45 TSMON (Offset = 1160h) [Reset = 00000000h]

TSMON is shown in Figure 38-49 and described in Table 38-51.

Return to the Summary Table.

RTC Month Time Stamp Capture - Calendar Mode With Binary / BCD Format

**Figure 38-49. TSMON**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | MONHIGHBCD | MONLOWBCD | | | |
| R-0h | | | R-0h | R-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | MONBIN | | | |
| R-0h | | | | R-0h | | | |

**Table 38-51. TSMON Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-13 | RESERVED | R | 0h | |
| 12 | MONHIGHBCD | R | 0h | Time Stamp Month BCD – high digit (0 or 1). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>1h = Highest possible value |
| 11-8 | MONLOWBCD | R | 0h | Time Stamp Month BCD – low digit (0 to 9). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>9h = Highest possible value |
| 7-4 | RESERVED | R | 0h | |
| 3-0 | MONBIN | R | 0h | Time Stamp Month Binary (1 to 12). If RTCBCD=1 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>Ch = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.46 TSYEAR (Offset = 1164h) [Reset = 00000000h]

TSYEAR is shown in Figure 38-50 and described in Table 38-52.

Return to the Summary Table.

RTC Years Time Stamp Capture - Calendar Mode With Binary / BCD Format

**Figure 38-50. TSYEAR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | CENTHIGHBCD | | | CENTLOWBCD | | | |
| R-0h | R-0h | | | R-0h | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DECADEBCD | | | | YERARLOWESTBCD | | | |
| R-0h | | | | R-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | YEARHIGHBIN | | | |
| R-0h | | | | R-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| YEARLOWBIN | | | | | | | |
| R-0h | | | | | | | |

**Table 38-52. TSYEAR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31 | RESERVED | R | 0h | |
| 30-28 | CENTHIGHBCD | R | 0h | Time Stamp Century BCD – high digit (0 to 4). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>4h = Highest possible value |
| 27-24 | CENTLOWBCD | R | 0h | Time Stamp Century BCD – low digit (0 to 9). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>9h = Highest possible value |
| 23-20 | DECADEBCD | R | 0h | Time Stamp Decade BCD (0 to 9). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>9h = Highest possible value |
| 19-16 | YERARLOWESTBCD | R | 0h | Time Stamp Year BCD – lowest digit (0 to 9). If RTCBCD=0 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>9h = Highest possible value |
| 15-12 | RESERVED | R | 0h | |
| 11-8 | YEARHIGHBIN | R | 0h | Time Stamp Year Binary – high byte. Valid values for Year are 0 to 4095. If RTCBCD=1 write to these bits will be ignored and read give the value 0.<br>0h = Smallest value<br>Fh = Highest possible value |
| 7-0 | YEARLOWBIN | R | 0h | Time Stamp Year Binary – low byte. Valid values for Year are 0 to 4095. If RTCBCD=1 write to these bits will be ignored and read give the value 0.<br>00h = Smallest value<br>FFh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.47 TSSTAT (Offset = 1168h) [Reset = 00000000h]

TSSTAT is shown in Figure 38-51 and described in Table 38-53.

Return to the Summary Table.

Time Stamp Status

**Figure 38-51. TSSTAT**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | TSVDDEVT |
| R-0h | | | | | | | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TSTIOEVT15 | TSTIOEVT14 | TSTIOEVT13 | TSTIOEVT12 | TSTIOEVT11 | TSTIOEVT10 | TSTIOEVT9 | TSTIOEVT8 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TSTIOEVT7 | TSTIOEVT6 | TSTIOEVT5 | TSTIOEVT4 | TSTIOEVT3 | TSTIOEVT2 | TSTIOEVT1 | TSTIOEVT0 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

**Table 38-53. TSSTAT Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-17 | RESERVED | R | 0h | |
| 16 | TSVDDEVT | R | 0h | Loss of VDD caused time stamp event<br>0h = no event detected<br>1h = event detected |
| 15 | TSTIOEVT15 | R | 0h | Tamper I/O 15 caused time stamp event<br>0h = no event detected<br>1h = event detected |
| 14 | TSTIOEVT14 | R | 0h | Tamper I/O 14 caused time stamp event<br>0h = no event detected<br>1h = event detected |
| 13 | TSTIOEVT13 | R | 0h | Tamper I/O 13 caused time stamp event<br>0h = no event detected<br>1h = event detected |
| 12 | TSTIOEVT12 | R | 0h | Tamper I/O 12 caused time stamp event<br>0h = no event detected<br>1h = event detected |
| 11 | TSTIOEVT11 | R | 0h | Tamper I/O 11 caused time stamp event<br>0h = no event detected<br>1h = event detected |
| 10 | TSTIOEVT10 | R | 0h | Tamper I/O 10 caused time stamp event<br>0h = no event detected<br>1h = event detected |
| 9 | TSTIOEVT9 | R | 0h | Tamper I/O 9 caused time stamp event<br>0h = no event detected<br>1h = event detected |
| 8 | TSTIOEVT8 | R | 0h | Tamper I/O 8 caused time stamp event<br>0h = no event detected<br>1h = event detected |
| 7 | TSTIOEVT7 | R | 0h | Tamper I/O 7 caused time stamp event<br>0h = no event detected<br>1h = event detected |



Copyright © 2026 Texas Instruments Incorporated

**Table 38-53. TSSTAT Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 6 | TSTIOEVT6 | R | 0h | Tamper I/O 6 caused time stamp event<br>0h = no event detected<br>1h = event detected |
| 5 | TSTIOEVT5 | R | 0h | Tamper I/O 5 caused time stamp event<br>0h = no event detected<br>1h = event detected |
| 4 | TSTIOEVT4 | R | 0h | Tamper I/O 4 caused time stamp event<br>0h = no event detected<br>1h = event detected |
| 3 | TSTIOEVT3 | R | 0h | Tamper I/O 3 caused time stamp event<br>0h = no event detected<br>1h = event detected |
| 2 | TSTIOEVT2 | R | 0h | Tamper I/O 2 caused time stamp event<br>0h = no event detected<br>1h = event detected |
| 1 | TSTIOEVT1 | R | 0h | Tamper I/O 1 caused time stamp event<br>0h = no event detected<br>1h = event detected |
| 0 | TSTIOEVT0 | R | 0h | Tamper I/O 0 caused time stamp event<br>0h = no event detected<br>1h = event detected |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.48 TSCTL (Offset = 116Ch) [Reset = 00000000h]

TSCTL is shown in Figure 38-52 and described in Table 38-54.

Return to the Summary Table.

time stamp control register

**Figure 38-52. TSCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| R-0/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | TSCAPTURE | RESERVED | | | TSVDDEN |
| R-0h | | | R/WK-0h | R-0h | | | R/WK-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TSTIOEN15 | TSTIOEN14 | TSTIOEN13 | TSTIOEN12 | TSTIOEN11 | TSTIOEN10 | TSTIOEN9 | TSTIOEN8 |
| R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TSTIOEN7 | TSTIOEN6 | TSTIOEN5 | TSTIOEN4 | TSTIOEN3 | TSTIOEN2 | TSTIOEN1 | TSTIOEN0 |
| R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h | R/WK-0h |

**Table 38-54. TSCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | R-0/W | 0h | need to write (KEY=0xC5) to update this register<br>C5h = This field must be written with 0xC5 to be able to clear any of the enable bits |
| 23-21 | RESERVED | R | 0h | |
| 20 | TSCAPTURE | R/WK | 0h | Defines the capture method of the RTC timestamp when a time stamp event occurs.<br>[IPSPECIFIC_RTC.TSCTL.KEY] must be written to access this bit.<br>0h = Time stamp holds RTC capture at first occurrence of time stamp event.<br>1h = Time stamp holds RTC capture at last occurrence of time stamp event. |
| 19-17 | RESERVED | R | 0h | |
| 16 | TSVDDEN | R/WK | 0h | Time Stamp by VDD Loss detection enable<br>[IPSPECIFIC_RTC.TSCTL.KEY] must be written to access this bit.<br>0h = function disabled<br>1h = function enabled |
| 15 | TSTIOEN15 | R/WK | 0h | Time Stamp by Tamper I/O 15 enable<br>[IPSPECIFIC_RTC.TSCTL.KEY] must be written to access this bit.<br>0h = function disabled<br>1h = function enabled |
| 14 | TSTIOEN14 | R/WK | 0h | Time Stamp by Tamper I/O 14 enable<br>[IPSPECIFIC_RTC.TSCTL.KEY] must be written to access this bit.<br>0h = function disabled<br>1h = function enabled |
| 13 | TSTIOEN13 | R/WK | 0h | Time Stamp by Tamper I/O 13 enable<br>[IPSPECIFIC_RTC.TSCTL.KEY] must be written to access this bit.<br>0h = function disabled<br>1h = function enabled |



Copyright © 2026 Texas Instruments Incorporated

**Table 38-54. TSCTL Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 12 | TSTIOEN12 | R/WK | 0h | Time Stamp by Tamper I/O 12 enable<br>[IPSPECIFIC_RTC.TSCTL.KEY] must be written to access this bit.<br>0h = function disabled<br>1h = function enabled |
| 11 | TSTIOEN11 | R/WK | 0h | Time Stamp by Tamper I/O 11 enable<br>[IPSPECIFIC_RTC.TSCTL.KEY] must be written to access this bit.<br>0h = function disabled<br>1h = function enabled |
| 10 | TSTIOEN10 | R/WK | 0h | Time Stamp by Tamper I/O 10 enable<br>[IPSPECIFIC_RTC.TSCTL.KEY] must be written to access this bit.<br>0h = function disabled<br>1h = function enabled |
| 9 | TSTIOEN9 | R/WK | 0h | Time Stamp by Tamper I/O 9 enable<br>[IPSPECIFIC_RTC.TSCTL.KEY] must be written to access this bit.<br>0h = function disabled<br>1h = function enabled |
| 8 | TSTIOEN8 | R/WK | 0h | Time Stamp by Tamper I/O 8 enable<br>[IPSPECIFIC_RTC.TSCTL.KEY] must be written to access this bit.<br>0h = function disabled<br>1h = function enabled |
| 7 | TSTIOEN7 | R/WK | 0h | Time Stamp by Tamper I/O 7 enable<br>[IPSPECIFIC_RTC.TSCTL.KEY] must be written to access this bit.<br>0h = function disabled<br>1h = function enabled |
| 6 | TSTIOEN6 | R/WK | 0h | Time Stamp by Tamper I/O 6 enable<br>[IPSPECIFIC_RTC.TSCTL.KEY] must be written to access this bit.<br>0h = function disabled<br>1h = function enabled |
| 5 | TSTIOEN5 | R/WK | 0h | Time Stamp by Tamper I/O 5 enable<br>[IPSPECIFIC_RTC.TSCTL.KEY] must be written to access this bit.<br>0h = function disabled<br>1h = function enabled |
| 4 | TSTIOEN4 | R/WK | 0h | Time Stamp by Tamper I/O 4 enable<br>[IPSPECIFIC_RTC.TSCTL.KEY] must be written to access this bit.<br>0h = function disabled<br>1h = function enabled |
| 3 | TSTIOEN3 | R/WK | 0h | Time Stamp by Tamper I/O 3 enable<br>[IPSPECIFIC_RTC.TSCTL.KEY] must be written to access this bit.<br>0h = function disabled<br>1h = function enabled |
| 2 | TSTIOEN2 | R/WK | 0h | Time Stamp by Tamper I/O 2 enable<br>[IPSPECIFIC_RTC.TSCTL.KEY] must be written to access this bit.<br>0h = function disabled<br>1h = function enabled |
| 1 | TSTIOEN1 | R/WK | 0h | Time Stamp by Tamper I/O 1 enable<br>[IPSPECIFIC_RTC.TSCTL.KEY] must be written to access this bit.<br>0h = function disabled<br>1h = function enabled |
| 0 | TSTIOEN0 | R/WK | 0h | Time Stamp by Tamper I/O 0 enable<br>[IPSPECIFIC_RTC.TSCTL.KEY] must be written to access this bit.<br>0h = function disabled<br>1h = function enabled |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.49 TSCLR (Offset = 1170h) [Reset = 00000000h]

TSCLR is shown in Figure 38-53 and described in Table 38-55.

Return to the Summary Table.

time stamp clear register

**Figure 38-53. TSCLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| KEY | | | | | | | | RESERVED | | | | | | | |
| R-0/W-0h | | | | | | | | R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | | | | CLR |
| R-0h | | | | | | | | | | | | | | | WK-0h |

**Table 38-55. TSCLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | R-0/W | 0h | need to write (KEY=0xE2) to update this register<br>E2h = This field must be written with 0xE2 to be able to clear any of the enable bits |
| 23-1 | RESERVED | R | 0h | |
| 0 | CLR | WK | 0h | Clear time stamp and status register.<br>[IPSPECIFIC_RTC.TSCLR.KEY] must be written to access this bit.<br>0h = Writing 0 has no effect<br>1h = clear time stamp event |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.50 LFSSRST (Offset = 1174h) [Reset = 00000000h]

LFSSRST is shown in Figure 38-54 and described in Table 38-56.

Return to the Summary Table.

Low frequency subsystem reset request. Asserting the VBATPOR bit in this register will issue a power cycle on the battery backup domain. This reset has the same effect as removing and reconnecting the power supply to the VBAT power pin.
This register can be write protected by the RTCLOCK register.

**Figure 38-54. LFSSRST**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| R-0/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | VBATPOR |
| R-0h | | | | | | | R/WK-0h |

**Table 38-56. LFSSRST Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | R-0/W | 0h | need to write (KEY=0x12) to update this register<br>12h = This field must be written with 0x12 to be able to request the power on reset. |
| 23-1 | RESERVED | R | 0h | |
| 0 | VBATPOR | R/WK | 0h | If set, the register bit will request a power on reset to the PMU of the LFSS.<br>[IPSPECIFIC_RTC.LFSSRST.KEY] must be written to access this bit.<br>0h = Writing this value has no effect.<br>1h = Request power on reset to the LFSS. |



Copyright © 2026 Texas Instruments Incorporated

### 38.4.51 RTCLOCK (Offset = 1178h) [Reset = 00000000h]

RTCLOCK is shown in Figure 38-55 and described in Table 38-57.

Return to the Summary Table.

The RTC lock bit protects the CLKCTL, SEC, MIN, HOUR, DAY, MON, YEAR and LFSSRST registers from accidental updates.

**Figure 38-55. RTCLOCK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| R-0/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | PROTECT |
| R-0h | | | | | | | R/WK-0h |

**Table 38-57. RTCLOCK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | R-0/W | 0h | need to write (KEY=0x22) to update this register<br>22h = This field must be written with 0x22 to be able to update any of the bits. |
| 23-1 | RESERVED | R | 0h | |
| 0 | PROTECT | R/WK | 0h | If set, the register bit will protect the CLKCTL, SEC, MIN, HOUR, DAY, MON, YEAR and LFSSRST from accidental writes. [IPSPECIFIC_RTC.RTCLOCK.KEY] must be written to access this bit.<br>0h = RTC counter is writable.<br>1h = RTC counter is read only access. |



Copyright © 2026 Texas Instruments Incorporated

Chapter 39
# IWDT

The independent watchdog timer (IWDT) is a device-independent supervisor which monitors code execution and overall hang up scenarios of the device. This IWDT has its own system independent clock source. If the application software does not successfully reset the watchdog within the programmed time, the watchdog generates a POR reset to the device.


**39.1 IWDT Overview** ..... 2506
**39.2 IWDT Clock Configuration** ..... 2506
**39.3 IWDT Period Selection** ..... 2506
**39.4 Debug Behavior of the IWDT** ..... 2507
**39.5 IWDT Registers** ..... 2507





Copyright © 2026 Texas Instruments Incorporated

## 39.1 IWDT Overview

Key features of the IWDT include:

- A 25-bit counter
- Counter driven from LFOSC (fixed 32kHz clock path) with a programmable clock divider
- Eight selectable watchdog timer periods

Figure 39-1 shows the block diagram of the IWDT.

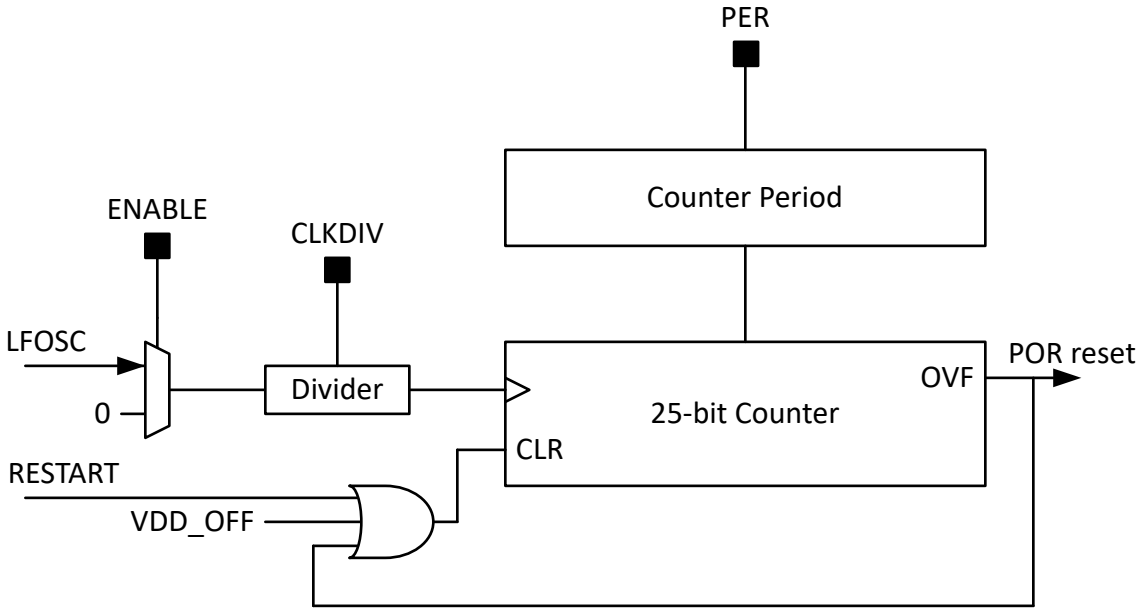



**Figure 39-1. Independent Watchdog Timer (IWDT) Block Diagram**

The typical use case of the IWDT is the replacement of an external watchdog timer in the system. Some safety applications require the monitoring of a device by an independent watchdog timer. This means the IWDT requires a system independent clock source, which is achieved through LFOSC.

The primary function of the IWDT is to initiate a full power on reset (POR) of the device when correct operation of the device has failed due to an unexpected software or system delay. The IWDT can be programmed with a predefined time within the application software.

The application software must restart the timer, indicating that application execution is proceeding normally. If application software fails to restart the timer within the specified time, the IWDT will issue a POR reset request to PMU of the device to generate a POR, restarting the device in a cold boot process.

## 39.2 IWDT Clock Configuration

The IWDT runs from the 32kHz low-frequency oscillator (LFOSC). A clock divider supports dividing the input clock from /1 (no divide) to /8 (divide-by-8) using the CLKDIV field in the WDTCTL register. The default CLKDIV setting is 0x03 (32kHz divided by 4, or 8kHz).

## 39.3 IWDT Period Selection

The IWDT has a 25-bit counter which is initially stopped after a power up of the VBAT power domain. The IWDT period (total time interval) is set by the PER field in the WDTCTL register. The total IWDT period is calculated as follows:

TWDT = (CLKDIV + 1) * PERCOUNT / 32768Hz

The total timer count PERCOUNT is selected to be one of 8 possible period count values, with the encoding given in Table 39-1.



Copyright © 2026 Texas Instruments Incorporated

**Table 39-1. IWDT Period and Period Count Settings**

| Period (PER) | 0x0 | 0x1 | 0x2 | 0x3 | 0x4 | 0x5 | 0x6 | 0x7 |
|---|---|---|---|---|---|---|---|---|
| Period Count (PERCOUNT) | 225 | 221 | 218 | 215 | 212 | 210 | 28 | 26 |

The combination of the period selection (PER) and clock divider (CLKDIV) enable a wide range of IWDT periods, from as short as 1.95ms to as long as 136.53 minutes.

Table 39-2 gives all possible IWDT periods for a given combination of PER and CLKDIV.

**Table 39-2. IWDT Period Timing Options**

| CLKDIV | PER | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 0x0 | 0x1 | 0x2 | 0x3 | 0x4 | 0x5 | 0x6 | 0x7 |
| | min | min | s | s | s | ms | ms | ms |
| 0x0 (/1) | 17.07 | 1.07 | 8.00 | 1.00 | 0.13 | 31.25 | 7.81 | 1.95 |
| 0x1 (/2) | 34.13 | 2.13 | 16.00 | 2.00 | 0.25 | 62.50 | 15.63 | 3.91 |
| 0x2 (/3) | 51.20 | 3.20 | 24.00 | 3.00 | 0.38 | 93.75 | 23.44 | 5.86 |
| 0x3 (/4) | 68.27 | 4.27 | 32.00 | 4.00 | 0.50 | 125.00 | 32.25 | 7.81 |
| 0x4 (/5) | 85.33 | 5.33 | 40.00 | 5.00 | 0.63 | 156.25 | 39.06 | 9.77 |
| 0x5 (/6) | 102.40 | 6.40 | 48.00 | 6.00 | 0.75 | 187.50 | 46.88 | 11.72 |
| 0x6 (/7) | 119.47 | 7.47 | 56.00 | 7.00 | 0.88 | 218.75 | 54.69 | 14.67 |
| 0x7 (/8) | 136.53 | 8.53 | 64.00 | 8.00 | 1.00 | 250.00 | 62.50 | 15.63 |

## 39.4 Debug Behavior of the IWDT

The IWDT can be configured to stop counting or continue counting when the CPU is halted for debug by the debug subsystem. By default, the IWDT stops counting when the CPU is halted for debug and the device is in a debug state. To allow the IWDT to continue to free run when the CPU is stopped for debug, set the FREE bit in the WDTDBGCTL register.

## 39.5 IWDT Registers

For IWDT registers, refer to the Section 37.7 related to the watchdog timer.



Copyright © 2026 Texas Instruments Incorporated

Chapter 40

# Window Watchdog Timer (WWDT)

The window watchdog timer (WWDT) supervises code execution. If the application software does not successfully reset the window watchdog within the programmed open time window, the window watchdog generates a reset.


**40.1 WWDT Overview** ..... 2509
**40.2 WWDT Operation** ..... 2510
**40.3 WWDT Registers** ..... 2514





Copyright © 2026 Texas Instruments Incorporated

## 40.1 WWDT Overview

The primary function of the window watchdog timer (WWDT) is to initiate a reset when correct operation of the device has failed due to an unexpected software or system delay. The WWDT can be programmed with a predefined time window within which the application software must restart the timer, indicating that application execution is proceeding normally. If application software fails to restart the timer within the specified window, the WWDT will issue a WWDT violation signal to SYSCTL to generate a reset.

If watchdog functionality is not required in an application, the WWDT can also be configured as a basic system interval timer which is capable of generating periodic maskable interrupts to the CPU.

Key features of the WWDT include:

- A 26-bit counter with closed and open window
- Counter driven from LFCLK (fixed 32kHz clock path) with a programmable clock divider
- Eight selectable watchdog timer periods
- Optional automatic suspension of counter when operating in low power modes
- Support for standard window watchdog mode or interval timer (nonwatchdog) mode

By default, WWDT0 initiates a BOOTRST, and WWDT1 initiates an NMI. The user is able to change the WWDT configuration such that a WWDT0 violation will generate a NMI, and a WWDT1 will generate a SYSRST.

### *40.1.1 Watchdog Mode*

In watchdog mode, the WWDT is configured to count up to the specified WWDT period. The WWDT counter must be restarted with the configured open window of the WWDT period, or the WWDT will assert a WWDT violation to SYSCTL and a reset will be generated.

The window watchdog timer supports detecting both a "too late" response as well as a "too early" response through the use of an optional closed window, as shown in Figure 40-1. The WWDT period consists of a closed window period and an open window period. The closed window period begins first, followed by the open window period. The WWDT can only be restarted during the open window period. An attempt to restart the WWDT during the closed window period results in a violation. Following the closed window, if the WWDT is not restarted before the end of the open window, the WWDT period expires and a violation is also generated.

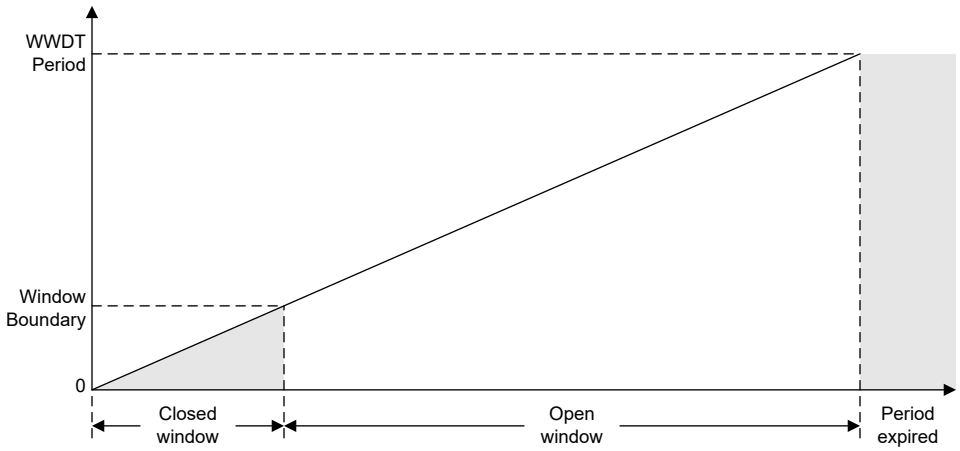



**Figure 40-1. WWDT Functionality**

If the closed window functionality is not desired, it can be disabled (set to 0%), giving traditional watchdog timer functionality where the WWDT can be reset any time before the WWDT period expires.

### *40.1.2 Interval Timer Mode*

The WWDT can be used in interval timer mode to generate periodic interrupts to the CPU when not using the watchdog functionality. When used in interval timer mode, a WWDT interrupt is generated when the WWDT period expires, or when an incorrect password is applied to the WWDT control registers.



Copyright © 2026 Texas Instruments Incorporated

## 40.2 WWDT Operation

The WWDT must be enabled before being configured for use through the PWREN register (see peripheral power enable).

The WWDT is configured through the WWDTCTL0 and WWDTCTL1 registers. The registers are password protected. Any register access (read or write) must be a 32-bit access. Write access must also include the corresponding password in the most significant byte (0xC9 for WWDTCTL0, and 0xBE for WWDTCTL1). Attempting a register write without the correct password, or attempting a write with an access other than a 32-bit access generates a WWDT violation to SYSCTL. The password byte always reads as 0x00.

The WWDT is disabled and cleared after a SYSRST. The WWDTCTL0 register sets the static configuration of the WWDT, including: the clock divider, the timer period, the two closed window percentages, the timer mode (WWDT or interval), and the stop-in-sleep status. The first write (with a key match) to the WWDTCTL0 register enables the WWDT. Once the WWDT is enabled, the WWDTCTL0 register becomes write protected. Any attempt to write to the WWDTCTL0 register after the WWDT is enabled generates a WWDT violation to SYSCTL. The RUN bit in the WWDTSTAT register indicates that the WWDT is running.

Figure 40-2 shows the WWDT functional block diagram.

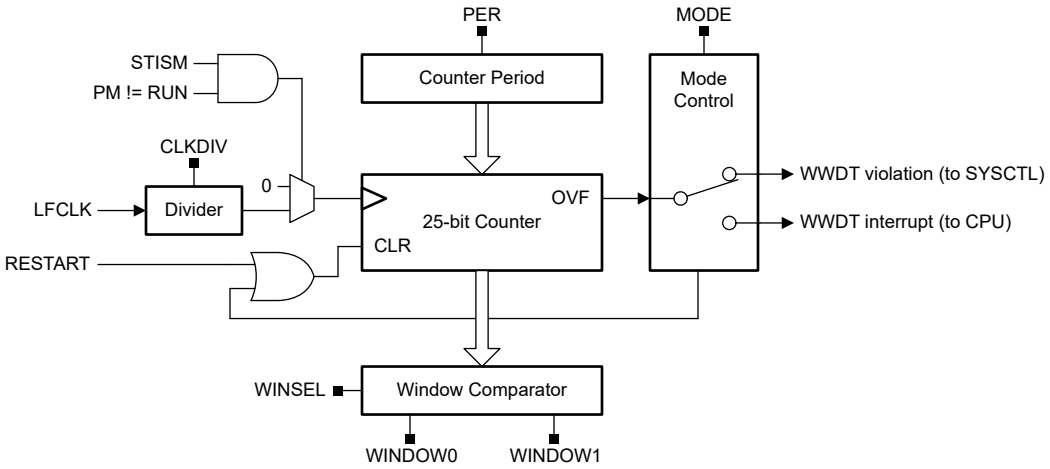



**Figure 40-2. WWDT Diagram**

### *40.2.1 Mode Selection*

The WWDT operating mode (watchdog mode or interval timer mode) is selected by the MODE bit in the WWDTCTL0 register. Watchdog mode is the default mode (MODE cleared). Setting the MODE bit configures the WWDT for interval mode.

When the WWDT is in watchdog mode, the WWDT counter must be restarted within the open window period by writing the RESTART value (0x000000A7) to the WWDTCNTRST register. After a reset or restart, the WWDT counter will restart from zero. A failure to restart the WWDT within the open window or an attempt to restart the WWDT counter during the closed window will generate a WWDT violation to SYSCTL. Writing any value other than the RESTART value to the WWDTCNTRST register also generates a WWDT violation.

When the WWDT is in interval mode, the timer acts as an interval timer, generating WWDT interrupts to the CPU as specified by the WWDT period. As soon as the WWDT is enabled in interval mode, the WWDT interval timer interrupt will be asserted after the expiration of the timer. It is not necessary to restart the WWDT in interval timer mode.

### *40.2.2 Clock Configuration*

The WWDT runs from the 32kHz low-frequency clock (LFCLK). A clock divider supports dividing the input clock from /1 (no divide) to /8 (divide-by-8) using the CLKDIV field in the WWDTCTL0 register. The default CLKDIV setting is 0x03 (32kHz divided by 4, or 8kHz).



Copyright © 2026 Texas Instruments Incorporated

By running from the LFCLK, the WWDT time base is independent of the main clock (MCLK) and CPU clock (CPUCLK) time base, provided that these clocks are not also configured to be running from the LFCLK. While the time base may be considered as independent and derived from a separate oscillator source, LFCLK edges are synchronized to the MCLK before sourcing the WWDT to enable simple access to the memory-mapped registers from application software. A simplified view of the clock scheme is given in Figure 40-3. In Figure 40-3, the internal LFOSC is configured to set the LFCLK rate, and the internal SYSOSC sets the MCLK/CPUCLK rate. Clock selection muxes and dividers are excluded from the figure to simplify the view; the complete clock tree is provided in *Clock Tree*.

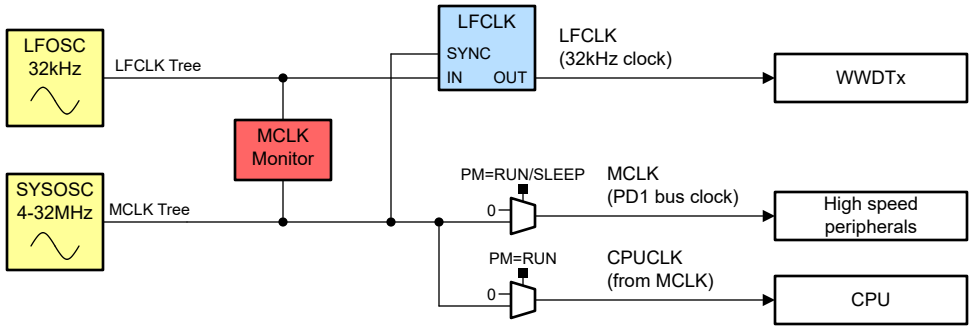



**Figure 40-3. WWDT Simplified Clock Source Tree**

In the event that the MCLK fails and synchronization of the LFCLK to the MCLK is lost, this failure may be detected by enabling the continuous MCLK monitor. When the MCLK monitor is enabled, a loss of MCLK is always a catastrophic failure which generates a BOOTRST within 12 LFCLK cycles. As a result, a loss of the MCLK tree, and corresponding loss of synchronization, does not prevent a BOOTRST from being generated.

### Period Selection

The WWDT has a 26-bit counter which is initially stopped after a SYSRST. The WWDT period (total time interval) is set by the PER field in the WWDTCTL0 register. The total WWDT period is calculated as follows:

$$T_{WWDT} = (CLKDIV + 1) * PER_{COUNT} / 32768Hz \quad (38)$$

The total timer count $PER_{COUNT}$ is selected to be one of 8 possible period count values, with the encoding given in Table 40-1.

**Table 40-1. WWDT Period Total Timer Count**

| PER | 0x0 | 0x1 | 0x2 | 0x3 | 0x4 | 0x5 | 0x6 | 0x7 |
|---|---|---|---|---|---|---|---|---|
| $PER_{COUNT}$ | $2^{26}$ | $2^{21}$ | $2^{18}$ | $2^{15}$ | $2^{12}$ | $2^{10}$ | $2^{8}$ | $2^{6}$ |

The combination of the period selection (PER) and clock divider (CLKDIV) enable a wide range of WWDT periods, from as short as 1.95ms to as long as 273.07 minutes. Table 40-2 gives all possible WWDT periods for a given combination of PER and CLKDIV.

**Table 40-2. WWDT Period Timing Options**

| CLKDIV | PER | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 0x0 | 0x1 | 0x2 | 0x3 | 0x4 | 0x5 | 0x6 | 0x7 |
| | *min* | *min* | *s* | *s* | *s* | *ms* | *ms* | *ms* |
| 0x0 (/1) | 34.13 | 1.07 | 8.00 | 1.00 | 0.13 | 31.25 | 7.81 | 1.95 |
| 0x1 (/2) | 68.27 | 2.13 | 16.00 | 2.00 | 0.25 | 62.50 | 15.63 | 3.91 |
| 0x2 (/3) | 102.40 | 3.20 | 24.00 | 3.00 | 0.38 | 93.75 | 23.44 | 5.86 |
| 0x3 (/4) | 136.53 | 4.27 | 32.00 | 4.00 | 0.50 | 125.00 | 32.25 | 7.81 |
| 0x4 (/5) | 170.67 | 5.33 | 40.00 | 5.00 | 0.63 | 156.25 | 39.06 | 9.77 |



Copyright © 2026 Texas Instruments Incorporated

**Table 40-2. WWDT Period Timing Options (continued)**

| CLKDIV | PER | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 0x0 | 0x1 | 0x2 | 0x3 | 0x4 | 0x5 | 0x6 | 0x7 |
| | *min* | *min* | *s* | *s* | *s* | *ms* | *ms* | *ms* |
| 0x5 (/6) | 204.80 | 6.40 | 48.00 | 6.00 | 0.75 | 187.50 | 46.88 | 11.72 |
| 0x6 (/7) | 238.93 | 7.47 | 56.00 | 7.00 | 0.88 | 218.75 | 54.69 | 13.67 |
| 0x7 (/8) | 273.07 | 8.53 | 64.00 | 8.00 | 1.00 | 250.00 | 62.50 | 15.63 |

### Synchronization Delay

When starting or re-starting the WWDT counter, a maximum synchronization delay of one 32kHz clock cycle (30.5µs) can occur before the WWDT counter begins counting from zero. The periods given in Table 40-2 do not include this synchronization delay.

### Closed Window Selection

Configuration of two closed window periods is supported by setting the WINDOW0 and WINDOW1 fields in the WWDTCTL0 register. The WINSEL bit in the WWDTCTL1 register determines the active window (either WINDOW0 or WINDOW1). Either window can be set to one of 8 possible window settings.

**Table 40-3. WWDT Window Options**

| WINDOW | 0x0 | 0x1 | 0x2 | 0x3 | 0x4 | 0x5 | 0x6 | 0x7 |
|---|---|---|---|---|---|---|---|---|
| Closed window | 0% | 12.5% | 18.75% | 25% | 50% | 75% | 81.25% | 87.5% |

Setting a WINDOWx value to 0x0 (0% closed, 100% open) is equivalent to disabling the window function of the WWDT. In this configuration, the WWDT can be restarted at any point during the WWDT period.

The active window selection can be changed after the WWDT has been enabled. When the WWDT is restarted by writing to the WWDTCNTRST register, the closed window selection (WINSEL) must not be changed for at least four 32kHz clock cycles (≈122µs).

### *40.2.3 Low-Power Mode Behavior*

The WWDT counter can be configured to continue counting when the device is in a low-power mode (CPU is disabled) or to continue to run when the device is in a low-power mode.

The STISM bit in the WWDTCTL0 register controls if the WWDT counter stops counting in low power modes. By default, the STISM bit is cleared, indicating that the WWDT continues to count in low-power modes. To stop the WWDT from counting in low-power modes, set the STISM bit when loading the WWDTCTL0 configuration to start the WWDT. In this case, when the low-power mode is exited and the CPU returns to operation, the WWDT counter resumes counting from the same value it held before entering the low-power mode.

### *40.2.4 Debug Behavior*

The WWDT can be configured to stop counting or continue counting when the CPU is halted for debug by the debug subsystem. By default, the WWDT stops counting when the CPU is halted for debug and the device is in a debug state. To allow the WWDT to continue to free run when the CPU is stopped for debug, set the FREE bit in the PDBGCTL register.

### *40.2.5 WWDT Events*

The WWDT module contains one event publisher and no event subscribers. One event publisher (CPU_INT) manages WWDT interrupt requests (IRQs) to the CPU subsystem through a static event route.

Table 40-4 lists the WWDT events.



Copyright © 2026 Texas Instruments Incorporated

**Table 40-4. WWDT Events**

| Event | Type | Source | Destination | Route | Configuration | Functionality |
|---|---|---|---|---|---|---|
| CPU Interrupt Event | Publisher | WWDT | CPU Subsystem | Static route | CPU_INT registers | Fixed interrupt route from WWDT to CPU |

#### 40.2.5.1 CPU Interrupt Event Publisher (CPU_INT)

The WWDT module provides one interrupt source which can be configured to source a CPU interrupt event. The WWDT interrupt conditions are given in Table 40-5.

**Table 40-5. WWDT CPU Interrupt Conditions (CPU_INT)**

| Index (IIDX) | Name | Description |
|---|---|---|
| 0 | INTTIM | Indicates that the WWDT interval timer period has expired |

The CPU interrupt event configuration is managed with the CPU_INT event management registers. See Section 7.2.5 for guidance on configuring the Event registers for CPU interrupts.



Copyright © 2026 Texas Instruments Incorporated

## 40.3 WWDT Registers

Table 40-6 lists the memory-mapped registers for the WWDT registers. All register offset addresses not listed in Table 40-6 should be considered as reserved locations and the register contents should not be modified.

**Table 40-6. WWDT Registers**

| Offset | Acronym | Register Name | Section |
|---|---|---|---|
| 800h | PWREN | Power enable | Section 40.3.1 |
| 804h | RSTCTL | Reset Control | Section 40.3.2 |
| 814h | STAT | Status Register | Section 40.3.3 |
| 1018h | PDBGCTL | Peripheral Debug Control | Section 40.3.4 |
| 1020h | IIDX | Interrupt index | Section 40.3.5 |
| 1028h | IMASK | Interrupt mask | Section 40.3.6 |
| 1030h | RIS | Raw interrupt status | Section 40.3.7 |
| 1038h | MIS | Masked interrupt status | Section 40.3.8 |
| 1040h | ISET | Interrupt set | Section 40.3.9 |
| 1048h | ICLR | Interrupt clear | Section 40.3.10 |
| 10E0h | EVT_MODE | Event Mode | Section 40.3.11 |
| 10FCh | DESC | Module Description | Section 40.3.12 |
| 1100h | WWDTCTL0 | Window Watchdog Timer Control Register 0 | Section 40.3.13 |
| 1104h | WWDTCTL1 | Window Watchdog Timer Control Register 0 | Section 40.3.14 |
| 1108h | WWDTCNTRST | Window Watchdog Timer Counter Reset Register | Section 40.3.15 |
| 110Ch | WWDTSTAT | Window Watchdog Timer Status Register | Section 40.3.16 |

Complex bit access types are encoded to fit into small table cells. Table 40-7 shows the codes that are used for access types in this section.

**Table 40-7. WWDT Access Type Codes**

| Access Type | Code | Description |
|---|---|---|
| **Read Type** | | |
| R | R | Read |
| **Write Type** | | |
| K | K | Write protected by a key |
| W | W | Write |
| WK | W<br>K | Write<br>Write protected by a key |
| **Reset or Default Value** | | |
| *-n* | | Value after reset or the default value |



Copyright © 2026 Texas Instruments Incorporated

### 40.3.1 PWREN Register (Offset = 800h) [Reset = 00000000h]

PWREN is shown in Figure 40-4 and described in Table 40-8.

Return to the Summary Table.

Register to control the power state

**Figure 40-4. PWREN Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R/W- | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R/W- | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | ENABLE |
| R/W- | | | | | | | K-0h |

**Table 40-8. PWREN Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | KEY to allow Power State Change<br>26h = KEY to allow write access to this register |
| 23-1 | RESERVED | R/W | 0h | |
| 0 | ENABLE | K | 0h | Enable the power<br>Note: For safety devices the power cannot be disabled once enabled.<br>KEY must be set to 26h to write to this bit.<br>0h = Disable Power<br>1h = Enable Power |



Copyright © 2026 Texas Instruments Incorporated

### 40.3.2 RSTCTL Register (Offset = 804h) [Reset = 00000000h]

RSTCTL is shown in Figure 40-5 and described in Table 40-9.

Return to the Summary Table.

Register to control reset assertion and de-assertion

**Figure 40-5. RSTCTL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | RESETSTKYCLR | RESETASSERT |
| W-0h | | | | | | WK-0h | WK-0h |

**Table 40-9. RSTCTL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | Unlock key<br>B1h = KEY to allow write access to this register |
| 23-2 | RESERVED | W | 0h | |
| 1 | RESETSTKYCLR | WK | 0h | Clear RESETSTKY<br>KEY must be set to B1h to write to this bit.<br>0h = Writing 0 has no effect<br>1h = Clear reset sticky bit |
| 0 | RESETASSERT | WK | 0h | Assert reset to the peripheral<br>Note: For safety devices a watchdog reset by software is not possible.<br>KEY must be set to B1h to write to this bit.<br>0h = Writing 0 has no effect<br>1h = Assert reset |



Copyright © 2026 Texas Instruments Incorporated

### 40.3.3 STAT Register (Offset = 814h) [Reset = 00000000h]

STAT is shown in Figure 40-6 and described in Table 40-10.

Return to the Summary Table.

peripheral enable and reset status

**Figure 40-6. STAT Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R- | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | RESETSTKY |
| R- | | | | | | | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R- | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | |
| R- | | | | | | | |

**Table 40-10. STAT Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-17 | RESERVED | R | 0h | |
| 16 | RESETSTKY | R | 0h | This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register<br>0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register<br>1h = The peripheral was reset since the last bit clear |
| 15-0 | RESERVED | R | 0h | |



Copyright © 2026 Texas Instruments Incorporated

### 40.3.4 PDBGCTL Register (Offset = 1018h) [Reset = 00000000h]

PDBGCTL is shown in Figure 40-7 and described in Table 40-11.

Return to the Summary Table.

This register can be used by the software developer to control the behavior of the peripheral relative to the 'Core Halted' input

**Figure 40-7. PDBGCTL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | FREE |
| R/W-0h | | | | | | | R/W-0h |

**Table 40-11. PDBGCTL Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R/W | 0h | |
| 0 | FREE | R/W | 0h | Free run control<br>0h = The peripheral freezes functionality while the Core Halted input is asserted and resumes when it is deasserted.<br>1h = The peripheral ignores the state of the Core Halted input |



Copyright © 2026 Texas Instruments Incorporated

### 40.3.5 IIDX Register (Offset = 1020h) [Reset = 00000000h]

IIDX is shown in Figure 40-8 and described in Table 40-12.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index.

**Figure 40-8. IIDX Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | | | | STAT | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | | | |

**Table 40-12. IIDX Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-5 | RESERVED | R | 0h | |
| 4-0 | STAT | R | 0h | Module Interrupt Vector Value. This register provides the highest priority interrupt index. A read clears the corresponding interrupt flag in RIS and MISC.<br>0h = No interrupt pending<br>1h = Interval Timer Interrupt; Interrupt Flag: INTTIM; Interrupt Priority: Highest |



Copyright © 2026 Texas Instruments Incorporated

### 40.3.6 IMASK Register (Offset = 1028h) [Reset = 00000000h]

IMASK is shown in Figure 40-9 and described in Table 40-13.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is unmasked. Unmasking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS."

**Figure 40-9. IMASK Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | INTTIM |
| R/W-0h | | | | | | | R/W-0h |

**Table 40-13. IMASK Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R/W | 0h | |
| 0 | INTTIM | R/W | 0h | Interval Timer Interrupt.<br>0h = Clear Interrupt Mask<br>1h = Set Interrupt Mask |



Copyright © 2026 Texas Instruments Incorporated

### 40.3.7 RIS Register (Offset = 1030h) [Reset = 00000000h]

RIS is shown in Figure 40-10 and described in Table 40-14.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Figure 40-10. RIS Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | INTTIM |
| R-0h | | | | | | | R-0h |

**Table 40-14. RIS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | INTTIM | R | 0h | Interval Timer Interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

### 40.3.8 MIS Register (Offset = 1038h) [Reset = 00000000h]

MIS is shown in Figure 40-11 and described in Table 40-15.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 40-11. MIS Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | INTTIM |
| R-0h | | | | | | | R-0h |

**Table 40-15. MIS Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | INTTIM | R | 0h | Interval Timer Interrupt.<br>0h = Interrupt did not occur<br>1h = Interrupt occurred |



Copyright © 2026 Texas Instruments Incorporated

### 40.3.9 ISET Register (Offset = 1040h) [Reset = 00000000h]

ISET is shown in Figure 40-12 and described in Table 40-16.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 40-12. ISET Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | INTTIM |
| W-0h | | | | | | | W-0h |

**Table 40-16. ISET Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | W | 0h | |
| 0 | INTTIM | W | 0h | Interval Timer Interrupt.<br>0h = Writing 0 has no effect<br>1h = Set Interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 40.3.10 ICLR Register (Offset = 1048h) [Reset = 00000000h]

ICLR is shown in Figure 40-13 and described in Table 40-17.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

**Figure 40-13. ICLR Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | INTTIM |
| W-0h | | | | | | | W-0h |

**Table 40-17. ICLR Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | W | 0h | |
| 0 | INTTIM | W | 0h | Interval Timer Interrupt.<br>0h = Writing 0 has no effect<br>1h = Clear Interrupt |



Copyright © 2026 Texas Instruments Incorporated

### 40.3.11 EVT_MODE Register (Offset = 10E0h) [Reset = 00000001h]

EVT_MODE is shown in Figure 40-14 and described in Table 40-18.

Return to the Summary Table.

Event mode register. It is used to select whether each line is disabled, in software mode (software clears the RIS) or in hardware mode (hardware clears the RIS)

**Figure 40-14. EVT_MODE Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R/W- | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R/W- | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R/W- | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | INT0_CFG | |
| R/W- | | | | | | R-1h | |

**Table 40-18. EVT_MODE Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | R/W | 0h | |
| 1-0 | INT0_CFG | R | 1h | Event line mode select for event corresponding to none.INT_EVENT[0]<br>0h = The interrupt or event line is disabled.<br>1h = The interrupt or event line is in software mode. Software must clear the RIS.<br>2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag. |



Copyright © 2026 Texas Instruments Incorporated

### 40.3.12 DESC Register (Offset = 10FCh) [Reset = 1F117010h]

DESC is shown in Figure 40-15 and described in Table 40-19.

Return to the Summary Table.

This register identifies the peripheral and its exact version.

**Figure 40-15. DESC Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MODULEID | | | | | | | | | | | | | | | |
| R-1F11h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| FEATUREVER | | | | INSTNUM | | | | MAJREV | | | | MINREV | | | |
| R-7h | | | | R-0h | | | | R-1h | | | | R-0h | | | |

**Table 40-19. DESC Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | MODULEID | R | 1F11h | Module identification contains a unique peripheral identification number. The assignments are maintained in a central database for all of the platform modules to ensure uniqueness.<br>0h = Smallest value<br>FFFFh = Highest possible value |
| 15-12 | FEATUREVER | R | 7h | Feature Set for the module *instance*<br>0h = Smallest value<br>Fh = Highest possible value |
| 11-8 | INSTNUM | R | 0h | Instance Number within the device. This will be a parameter to the RTL for modules that can have multiple instances<br>0h = Smallest value<br>Fh = Highest possible value |
| 7-4 | MAJREV | R | 1h | Major rev of the IP<br>0h = Smallest value<br>Fh = Highest possible value |
| 3-0 | MINREV | R | 0h | Minor rev of the IP<br>0h = Smallest value<br>Fh = Highest possible value |



Copyright © 2026 Texas Instruments Incorporated

### 40.3.13 WWDTCTL0 Register (Offset = 1100h) [Reset = 00000043h]

WWDTCTL0 is shown in Figure 40-16 and described in Table 40-20.

Return to the Summary Table.

Window Watchdog Timer Control 0 Register
NOTE: Write to this register is enabled after System Reset. The first successful write (key match) enables the Watchdog. When the watchdog is enabled all subsequent writes to this register activate the WWDT error signal to the ESM.

**Figure 40-16. WWDTCTL0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | STISM | MODE |
| R/W-0h | | | | | | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | WINDOW1 | | | RESERVED | WINDOW0 | | |
| R/W-0h | R/W-0h | | | R/W-0h | R/W-0h | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | PER | | | RESERVED | CLKDIV | | |
| R/W-0h | R/W-4h | | | R/W-0h | R/W-3h | | |

**Table 40-20. WWDTCTL0 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | KEY to allow write access to this register.<br>Writing to this register with an incorrect key activates the WWDT error signal to the ESM.<br>Read as 0.<br>C9h (W) = KEY to allow write access to this register |
| 23-18 | RESERVED | R/W | 0h | |
| 17 | STISM | R/W | 0h | Stop In Sleep Mode.<br>The functionality of this bit requires that POLICY.HWCEN = 0. If POLICY.HWCEN = 1 the WWDT resets during sleep and needs re-configuration.<br>Note: This bit has no effect for the global Window Watchdog as Sleep Mode is not supported.<br>0h = The WWDT continues to function in Sleep mode.<br>1h = The WWDT stops in Sleep mode and resumes where it was stopped after wakeup. |
| 16 | MODE | R/W | 0h | Window Watchdog Timer Mode<br>0h = Window Watchdog Timer Mode. The WWDT will generate a error signal to the ESM when following conditions occur: - Timer Expiration (Timeout) - Reset WWDT during the active window closed period - Keyword violation<br>1h = Interval Timer Mode. The WWDT acts as an interval timer. It generates an interrupt on timeout. |
| 15 | RESERVED | R/W | 0h | |



Copyright © 2026 Texas Instruments Incorporated

**Table 40-20. WWDTCTL0 Register Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 14-12 | WINDOW1 | R/W | 0h | Closed window period in percentage of the timer interval. WWDTCTL1.WINSEL determines the active window setting (WWDTCTL0.WINDOW0 or WWDTCTL0.WINDOW1).<br>0h = 0% (No closed Window)<br>1h = 12.50% of the total timer period is closed window<br>2h = 18.75% of the total timer period is closed window<br>3h = 25% of the total timer period is closed window<br>4h = 50% of the total timer period is closed window<br>5h = 75% of the total timer period is closed window<br>6h = 81.25% of the total timer period is closed window<br>7h = 87.50% of the total timer period is closed window |
| 11 | RESERVED | R/W | 0h | |
| 10-8 | WINDOW0 | R/W | 0h | Closed window period in percentage of the timer interval. WWDTCTL1.WINSEL determines the active window setting (WWDTCTL0.WINDOW0 or WWDTCTL0.WINDOW1).<br>0h = 0% (No closed Window)<br>1h = 12.50% of the total timer period is closed window<br>2h = 18.75% of the total timer period is closed window<br>3h = 25% of the total timer period is closed window<br>4h = 50% of the total timer period is closed window<br>5h = 75% of the total timer period is closed window<br>6h = 81.25% of the total timer period is closed window<br>7h = 87.50% of the total timer period is closed window |
| 7 | RESERVED | R/W | 0h | |
| 6-4 | PER | R/W | 4h | Timer Period of the WWDT. These bits select the total watchdog timer count.<br>0h = Total timer count is $2^{25}$<br>1h = Total timer count is $2^{21}$<br>2h = Total timer count is $2^{18}$<br>3h = Total timer count is $2^{15}$<br>4h = Total timer count is $2^{12}$ (default)<br>5h = Total timer count is $2^{10}$<br>6h = Total timer count is $2^{8}$<br>7h = Total timer count is $2^{6}$ |
| 3 | RESERVED | R/W | 0h | |
| 2-0 | CLKDIV | R/W | 3h | Module Clock Divider, Divide the clock source by CLKDIV+1. Divider values from /1 to /8 are possible.<br>The clock divider is currently 4 bits. Bit 4 has no effect and should always be written with 0.<br>0h = Minimum value<br>7h = Maximum value |



Copyright © 2026 Texas Instruments Incorporated

### 40.3.14 WWDTCTL1 Register (Offset = 1104h) [Reset = 00000000h]

WWDTCTL1 is shown in Figure 40-17 and described in Table 40-21.

Return to the Summary Table.

Window Watchdog Timer Control 1 Register

**Figure 40-17. WWDTCTL1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | WINSEL |
| R/W-0h | | | | | | | R/W-0h |

**Table 40-21. WWDTCTL1 Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-24 | KEY | W | 0h | KEY to allow write access to this register.<br>Writing to this register with an incorrect key activates the WWDT error signal to the ESM.<br>Read as 0.<br>BEh (W) = KEY to allow write access to this register |
| 23-1 | RESERVED | R/W | 0h | |
| 0 | WINSEL | R/W | 0h | Close Window Select<br>0h = In window mode field WINDOW0 of WDDTCTL0 defines the closed window size.<br>1h = In window mode field WINDOW1 of WDDTCTL0 defines the closed window size. |



Copyright © 2026 Texas Instruments Incorporated

### 40.3.15 WWDTCNTRST Register (Offset = 1108h) [Reset = 00000000h]

WWDTCNTRST is shown in Figure 40-18 and described in Table 40-22.

Return to the Summary Table.

Window Watchdog Timer Counter Restart Register

**Figure 40-18. WWDTCNTRST Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESTART | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 40-22. WWDTCNTRST Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | RESTART | R/W | 0h | Window Watchdog Timer Counter Restart Writing 00A7h to this register restarts the WWDT Counter.<br>Writing any other value causes an error generation to the ESM.<br>Read as 0.<br>0h = Minimum value<br>FFFFFFFFh = Maximum value |



Copyright © 2026 Texas Instruments Incorporated

### 40.3.16 WWDTSTAT Register (Offset = 110Ch) [Reset = 00000000h]

WWDTSTAT is shown in Figure 40-19 and described in Table 40-23.

Return to the Summary Table.

Window Watchdog Timer Status Register
A write to this register has no effect.

**Figure 40-19. WWDTSTAT Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | | | | | | | | | | RUN |
| R-0h | | | | | | | | | | | | | | | R-0h |

**Table 40-23. WWDTSTAT Register Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | |
| 0 | RUN | R | 0h | Watchdog running status flag.<br>0h = Watchdog counter stopped.<br>1h = Watchdog running. |



Copyright © 2026 Texas Instruments Incorporated

Chapter 41

# Debug

The debug subsystem (DEBUGSS) is implemented in all MSP devices. The DEBUGSS enables comprehensive debug of application software running on the processor during development by interfacing an external debug probe to the device systems through a serial wire debug (SWD) interface.


**41.1 DEBUGSS Overview** ..... 2533
**41.2 DEBUGSS Operation** ..... 2536
**41.3 DEBUGSS Registers** ..... 2544





Copyright © 2026 Texas Instruments Incorporated

## 41.1 DEBUGSS Overview

The debug subsystem (DEBUGSS) interfaces the ARM Serial Wire Debug (SWD) two-wire physical interface to multiple debug functions within the device. MSP devices support debugging of processor execution, the device state, and the power state (through EnergyTrace technology). The DEBUGSS also provides a mailbox system for communicating with software through SWD.

Key features provided by the debug subsystem include:

- The ARM Serial Wire Debug (SWD) two-wire (SWDIO, SWCLK) debug interface, compatible with both TI and 3rd party debug probes
  - On-chip pullup and pulldown resistors for SWDIO and SWCLK, respectively, enabled by default
  - Support for disabling SWD functions to use SWD pins as general-purpose input/output pins
  - Support for debug on all low power modes
- Debug of the processor
  - Run, halt, and step debug support
  - Up to 4 hardware breakpoints (BPU)
  - Up to 2 hardware watchpoints (DWT)
  - Instruction trace of up to 4 branches through the ARM micro trace buffer (MTB)
  - Supporting software breakpoints
- Software-configurable peripheral behavior during processor debug
  - Ability to free run select peripherals through debug halt
  - Ability to halt select peripherals on a debug halt
  - Ability to request reset and mode changes to the PMCU
- Monitoring of CPU state through EnergyTrace technology
- Mailbox (DSSM) for passing data and control signals between the SWD interface and boot ROM (as well as application software)
- Support for various security features, including SWD lockout and password authenticated debugging
- **Note**

  Refer to the datasheet for device-specific features

### *41.1.1 Debug Interconnect*

The DEBUGSS architecture is given in Figure 41-1.



Copyright © 2026 Texas Instruments Incorporated

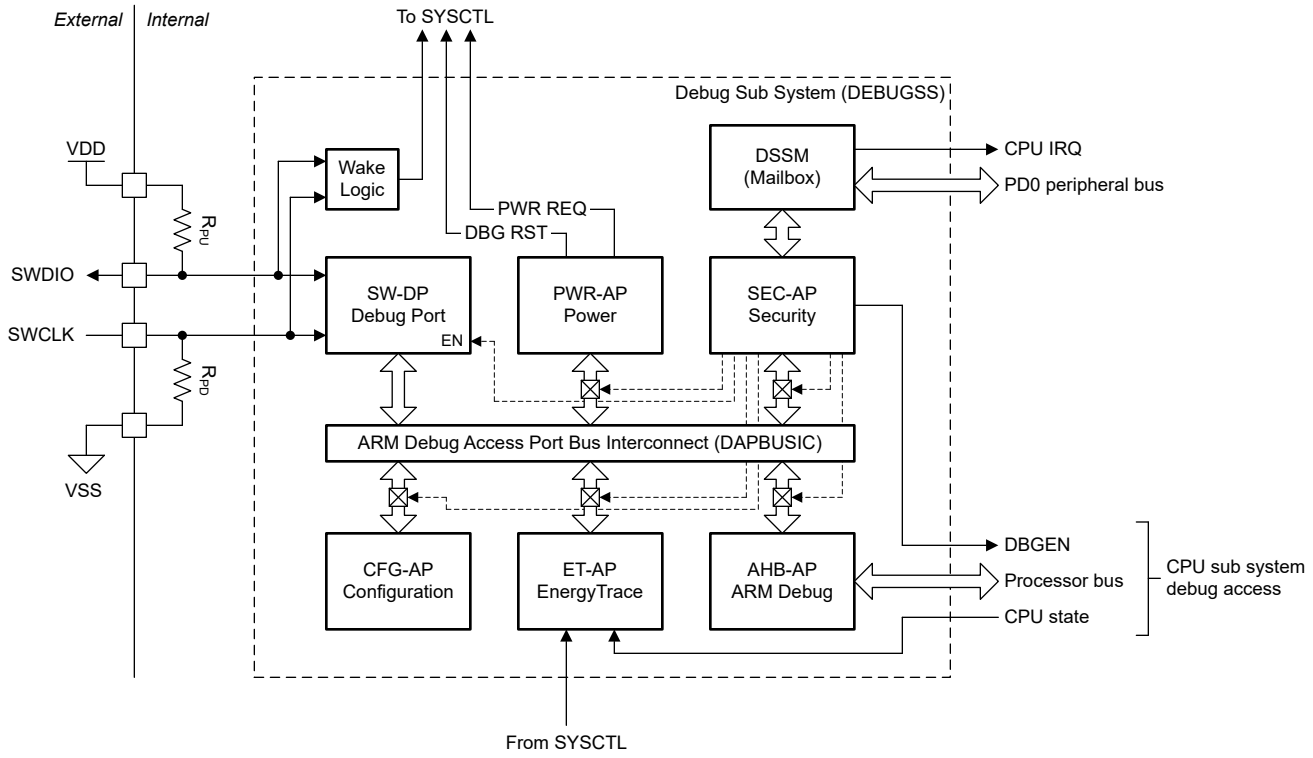



**Figure 41-1. Debug Sub System Block Diagram**

The SWD physical interface interacts with the Arm serial wire debug port (SW-DP) to gain access to the debug access port bus interconnect (DAPBUSIC) when the SW-DP is enabled. From TI, devices ship with the SW-DP enabled to allow SWD access to the device for development and production programming, but the SW-DP can be configured to be permanently disabled through the boot configuration policy (see Section 41.2.3).

The DAPBUSIC enables a debug probe to access one or more debug access ports. For a debug probe to be able to communicate with an access port, the SW-DP debug port must not be disabled by the device boot configuration policy, and the target access port must also not be disabled by the boot configuration policy. The available access ports are given in Section 41.1.3.

The SWD and SW-DP also contain signaling to the PMCU module to support debug-generated resets and operating mode changes (see Section 41.2.2).

### *41.1.2 Physical Interface*

Debug connections to the device are supported through an Arm serial wire debug (SWD) compliant interface. The SWD interface requires two connections:

- A bidirectional data line (SWDIO) used to send data to, and receive data from, the device
- A unidirectional clock line (SWCLK) driven by the debug probe connecting to the device

The SWD interface uses the standard logic levels of the device for SWD communication. See the device-specific data sheet for input and output logic levels for a given supply voltage (VDD). A SWCLK frequency of up to 10MHz is supported by the DEBUGSS.

During SWD operation, the SWDIO line can be driven high or driven low by either the target device or the debug probe. As either device can drive the line, when ownership of the shared SWDIO line is switched between the device and the debug probe, undriven time slots are inserted as a part of the SWD protocol. The primary purpose of the pullup resistor on the SWDIO line, and the pulldown resistor on the SWCLK line, is to place the SWD pins into a known state when no debug probe is attached. A minimum resistance of 100kΩ is recommended by Arm. The internal pullup/pulldown resistors fulfill this requirement and external resistors are not required for correct operation of the SWD interface.



Copyright © 2026 Texas Instruments Incorporated

After a power-on reset (POR), MSP devices configure the SWD pins in SWD mode with an internal pullup resistor enabled on the SWDIO line and an internal pulldown resistor enabled on the SWCLK line. If the device configuration has not permanently disabled all SWD access, then the SWD interface is enabled during the boot process and a debug probe can be connected to the DEBUGSS.

In the event that a device was configured by software to enter SHUTDOWN mode, and a debug probe is then connected to the SWD pins with SWCLK active, wakeup logic will trigger an exit from SHUTDOWN mode and cause a BOR. A debug connection can then be established to the DEBUGSS after the BOR completes.

Upon physical connection of a debug probe, a valid JTAG-to-SWD sequence must be sent from the debug probe to the target device to initiate a SWD connection with the SW-DP. Any other sequence will not wake the device from SHUTDOWN mode. Once the sequence is applied and the SWD connection is established, communication with enabled debug access points is possible.

For secure applications the sowftware can be made aware of a connect by utilizing the DEBUGSS PWRUPIFG interrupt. When the debug probe is disconnected and the SWD connection is lost, the PWRDWNIFG interrupt is asserted.

It is possible for application software to disable the SWD interface in SYSCTL, freeing the IO to be used for general purpose IO functionality. Review *SWD Pins* in SYSCTL for using the SWD pins for functionality other than SWD. Once software disables SWD functionality, it is not possible to re-enable it other than by triggering a POR. A POR will automatically re-enable the SWD functionality and put the SWD pins into SWD mode with pullup/pulldown resistors enabled. To re-gain debug access to a device which contains software that disables the SWD pins at startup, it is necessary to hold the device in a reset state with the NRST pin during a POR. This will prevent the application software from starting and will allow the debug probe to gain access to the device, at which point a mass erase DSSM command can be sent from the integrated development environment to the device via the debug probe to remove the application software which is disabling the SWD pins.

**Note**

BOR, BOOTRST, and SYSRST levels do reset the IOMUX logic, which will re-enable the pullup/pulldown resistors on the SWDIO/SWCLK pin. However, the SWD functionality remains disabled until the next POR. Because the device always powers up with the SWDIO pullup and SWCLK pulldown resistors enabled, the hardware design must accommodate this when using the SWD pins for functions other than SWD after startup. After reset, application software may disable the pullup/pulldown resistors in the IOMUX to free the SWD pins for other purposes.

### *41.1.3 Debug Access Ports*

The debug access ports in the DEBUGSS are given in Table 41-1.

**Table 41-1. DEBUGSS Access Port Listing**

| APSEL | AP | Port Description | Purpose |
|---|---|---|---|
| 0x0 | AHB-AP | MCPUSS debug access port | Debug of the processor and peripherals |
| 0x1 | CFG-AP | Configuration access port | Access device type information |
| 0x2 | SEC-AP | Security access port | Access the debug mailbox (DSSM) |
| 0x3 | ET-AP | EnergyTrace™ technology access port | Read the power state data from EnergyTrace technology for power aware debug |
| 0x4 | PWR-AP | Power access port | Configure the device power states (interfaces with PMCU/SYSCTL) |

The AHB-AP, PWR-AP, and ET-AP provide the complete device debug functionality (processor debug, peripheral and memory bus access, power state control, and processor state). See Section 41.2.1 for more information.

The CFG-AP provides device information to the debug probe so that the debug probe can identify device characteristics, including the device part number and the device revision.

The SEC-AP provides access to the mailbox for communicating with software running on the device through SWD. See Section 41.2.4 for more information.



Copyright © 2026 Texas Instruments Incorporated

## 41.2 DEBUGSS Operation

The features and behaviors of the DEBUGSS module is discussed in the following sections.

### *41.2.1 Debug Features*

The DEBUGSS supports processor debug, processor trace, peripheral debug, and energy state debug.

#### 41.2.1.1 Processor Debug

The Arm Cortex-M0+ processor supports a wide range of features to simplify debugging of application software during development. Key features supported by MSP MCUs include:

- Ability to halt the processor through a assertion of a halt signal, a configured debug event (such as a hard fault entry or reset), or a BKPT instruction (for software breakpoints)
- Ability to step through instructions (with or without peripheral interrupts enabled)
- Ability to run through instructions (with or without peripheral interrupts enabled)
- Ability to read and write CPU registers when halted
- Ability to read exception information through the Cortex-M0+ system control space (SCS)
- Support for up to 4 hardware breakpoints
- Support for up to 2 hardware watchpoints
- Support for accessing the device memory map
- Support for Microtrace Buffer (MTB) up to 4 jump locations

##### *41.2.1.1.1 Breakpoint Unit (BPU)*

The breakpoint unit (BPU) provides up to 4 comparators which can be used to generate a debug event when the address of an instruction fetch matches the address programmed into the respective BPU comparator.

The BPU does not generate a debug event upon an address match for a data read or data write access.

Address matching is possible for half-word (16-bit) instructions and word (32-bit) instructions fetched from the CODE region (0x0000.0000 to 0x1FFF.FFFF).

If a debug scenario requires more than the supported hardware breakpoints, software breakpoints can be used together with hardware breakpoints using the BKPT instruction. If debugging of code in the SRAM region is desired, hardware breakpoints are not available and software breakpoints must be inserted by the debug probe instead.

```
// Example of a breakpoint function in C (TI Arm CLANG compiler)
__BKPT(0);
```

##### *41.2.1.1.2 Data Watchpoint and Trace Unit (DWT)*

The data watchpoint and trace unit (DWT) provides up to 2 comparators which both support generating an event upon a data address match (watchpoint event) or an instruction address match (PC watchpoint event).

The DWT comparators support masking of the address, enabling an event to be generated when the processor attempts to access an address within a specified address range.

##### *41.2.1.1.3 Processor Trace (MTB)*

Some devices support basic instruction execution trace to obtain context of the sequence of execution which led to a certain state of the processor. The processor trace engine is based on the Arm CoreSight MTB-M0+ micro trace buffer.

The MTB captures the processor's program counter (PC) state when the PC changes in a nonsequential way due to a branch instruction or an exception. Load and store activity is not captured by the MTB. When nonsequential execution is detected by the MTB, the change is captured and stored into a small buffer memory (described in Table 41-2) which can be read out later by application software or the debug probe.



Copyright © 2026 Texas Instruments Incorporated

**Table 41-2. MTB Buffer Memory**

| Start Address | End Address | Length |
|---|---|---|
| 0x4040.3000 | 0x4040.3020 | 32B (4 trace packets) |

#### 41.2.1.2 Peripheral Debug

In addition to processor debug, the DEBUGSS can be used to access the device memory map from the perspective of the processor. Thus, a connected debug probe can be used to read and write memory-mapped peripheral registers, the system SRAM, and the flash memory.

Certain peripherals support advanced debug configuration options. These options are configured by application software (or optionally, the debug probe) by setting/clearing various debug control bits in the memory map of a given peripheral. In general, the debug behavior of a particular peripheral is specified in the PDBGCTL register of each peripheral. Many peripherals offer the option of halting the functional clock to the peripheral when the processor is halted for debug, thus pausing the peripheral together with the processor (default configuration), or letting the peripheral run even when the processor is halted for debug.

For example, the WWDT peripheral supports the FREE bit in the PDBGCTL register. Setting the FREE bit in PDBGCTL for a WWDT causes the WWDT counter to run even if the processor is halted for debug. Which will result in the watchdog not being serviced (causing an interrupt or reset).

#### 41.2.1.3 EnergyTrace Technology

The DEBUGSS in MSPM0 devices supports EnergyTrace technology. EnergyTrace technology enables power profiling of MCU devices running application code. This is very useful when developing an application which must be optimized for low-power operation.

Development tools from Texas Instruments, including the MSPM0 LaunchPad development tools, support hardware energy measurement of the target MSPM0 over time through EnergyTrace charge counting. This mechanism enables a developer to obtain an energy usage profile for an application, based on real current measurements with a wide dynamic range.

To give context to the energy measurements made by the hardware development tools supporting EnergyTrace technology, MSPM0 MCUs also enable EnergyTrace+. EnergyTrace+ is a component of the DEBUGSS that lets the debug probe log the state of the processor (RUN, SLEEP) and the current program counter value while the device is running. This state information can be then overlaid with energy measurements to determine if the cause of high current is the processor running or some other activity on the device.

TI's Code Composer Studio integrated development environment provides out-of-the-box support for EnergyTrace energy measurement and EnergyTrace+ processor state logging with MSPM0 devices.

**Note**

EnergyTrace+ state logging is not supported in SHUTDOWN mode, however the energy profiling function of EnergyTrace is still functional

### *41.2.2 Behavior in Low Power Modes*

The DEBUGSS supports maintaining a debug connection through SWD in all operating modes except SHUTDOWN. While in SHUTDOWN mode the DEBUGSS supports wakeup of the device to regain full access over the SWD connection.

By default access to device memory and peripherals is possible in RUN mode and SLEEP mode, in which a debug probe can be actively connected to the AHB-AP access port to interface with the processor. In STOP and STANDBY modes, a debug connection can be established and/or maintained with the DEBUGSS, but not with the CPU debug access port. The PWR-AP of the DEBUGSS allows an overwrite of the default behavior to keep access to the processor interface established while being in STOP or STANDBY mode.

To enter SHUTDOWN mode while actively debugging, the debugger must be disconnected from the device so that the device can properly enter the low power mode. The debug logic will then be powered down with the device VCORE. While a debug connection to the DEBUGSS is not possible while the device is in SHUTDOWN



Copyright © 2026 Texas Instruments Incorporated

mode, a debug probe can cause the device to exit SHUTDOWN mode by attempting to communicate with the SWD pins. The device will detect attempted SWD communication even when the device is in SHUTDOWN. If activity is detected, a SHUTDOWN exit is initiated and the device will transition through a BOR state, after which a debug connection can be made to the DEBUGSS through SWD.

For more details regarding the access points of the DEBUGSS please refer to the following application note Hardware Programming and Debugger Guide for MSP.

The DEBUGSS functionality by operating mode is given in Table 41-3.

**Table 41-3. DEBUGSS Functionality by Operating Mode**

| Capability | RUN | SLEEP | STOP | STANDBY | SHUTDOWN | NRST HOLD |
|---|---|---|---|---|---|---|
| Processor debug | Y | Y | N | N | N | N |
| Memory map access | Y | Y | N | N | N | N |
| Debug status through SW-DP | Y | Y | Y | Y | N | Y |
| Debug state maintained | Y | Y | Y | Y | N | N |
| Wake from SWD | - | - | - | - | Y | - |

### *41.2.3 Restricting Debug Access*

The debug subsystem supports several methods for restricting access to the device through the SWD interface. The debug access policy is determined by the user configuration specified in the NONMAIN flash region.

There are 4 levels of access control, given in Table 41-4. By default, products shipped from TI arrive in a "debug enabled" state where the device is fully open. This state is not recommended for production. For production, TI recommends changing the debug configuration to password protected, debug disabled, or SWD disabled.

**Table 41-4. Debug Access Control**

| DEBUGSS Function | Debug Configuration | | | |
|---|---|---|---|---|
| | Debug Enabled (default) | Debug Enabled with Password | Debug Disabled | SWD Disabled |
| SW-DP (debug port) | EN | EN | EN | DIS |
| CFG-AP | EN | EN | EN | DIS |
| SEC-AP | EN | EN | EN | DIS |
| ET-AP | EN | EN w/ PW | DIS | DIS |
| AHB-AP (CPU Debug) | EN | EN w/ PW | DIS | DIS |
| PWR-AP | EN | EN w/ PW | DIS | DIS |

The debug access control can be configured by writing predefined values into the NONMAIN BOOTCFG0.

**Table 41-5. Debug Access Control Settings for BOOTCFG0**

| Debug Configuration | SWDP_MODE | DEBUGACCESS |
|---|---|---|
| Debug Enabled (default) | AABBh | AABBh |
| Debug Enabled with Password | AABBh | CCDDh |
| Debug Disabled | AABBh | 5566h |
| SWD Disabled | 5566h | 5566h |

When debug is set to enabled with password, the debug access command together with the user-specified debug access password must be provided to the DEBUGSS mailbox, and a BOOTRST must be issued. The password for access control is stored in NONMAIN utilizing the PWDDEBUGLOCK registers. Depending on the SOC implentation the password is either 128-bit plain text or 256-bit SHA-256 hash value. Refer to the NONMAIN Layout Types table for SOC password implementation.

When debug is disabled, the SW-DP is disabled during the boot process and any commands previously sent to the mailbox are ignored during boot. Following boot, any attempt to connect to the SW-DP is ignored.



Copyright © 2026 Texas Instruments Incorporated

Permanently lock debug access to the device by configuring the NONMAIN flash region to disable debug access while also configuring the NONMAIN flash region as statically write protected (locked). Locking the NONMAIN configuration has the added security of preventing the bootstrap loader (BSL) and application code from changing the debug security policy.

**Table 41-6. Debug related password registers**

| Debug Function | Debug Access Control | Factory Reset | Mass Erase |
|---|---|---|---|
| Register Name | PWDDEBUGLOCK | PWDFACTORYRESET | PWDMASSERASE |

### 128-bit plain text password

To enable a password on a debug function write a 128-bit hex value split into four 32-bit words into the respective password register.

Example: Factory Reset Password Configuration

1. Create 128-bit hex value
   a. 0xCAFECAFE12345678A5A5C3C30000FFFF
2. Split the 128-bit value into four 32-bit words and write the value into the respective registers
   a. PWDFACTORYRESET[0]←0x0000FFFF
   b. PWDFACTORYRESET[1]←0xA5A5C3C3
   c. PWDFACTORYRESET[2]←0x12345678
   d. PWDFACTORYRESET[3]←0xCAFECAFE
3. Repeat the same steps for any function that needs password protection

### 256-bit SHA-256 hash value

To enable a password on a debug function write the SHA-256 digest of the 128-bit plain text password into the respective registers.

Example: Factory Reset Password Configuration

1. Determine the desired 128-bit plain text password

   0xCAFECAFE12345678A5A5C3C30000FFFF
2. Split into four 32-bit words, reverse the password endianess, and then combine into one value

   a. 0xFECAFECA
   b. 0x78563412
   c. 0xC3C3A5A5
   d. 0xFFFF0000
   e. 0xFECAFECA78563412C3C3A5A5FFFF0000
3. Calculate the SHA of the value and break up the output into eight 32-bit words
   a. 0xE6226BD43FB9B539E364ABAC313E51244355F1275C1D5BE010793A0E01DA0D17
   b. 0xE6226BD4
   c. 0x3FB9B539
   d. 0xE364ABAC
   e. 0x313E5124
   f. 0x4355F127
   g. 0x5C1D5BE0
   h. 0x10793A0E
   i. 0x01DA0D17
4. Reverse the output of the eight 32-bit words



Copyright © 2026 Texas Instruments Incorporated

a. 0xE6226BD4 → 0xD46B22E6
b. 0x3FB9B539 → 0x39B5B93F
c. 0xE364ABAC → 0xACAB64E3
d. 0x313E5124 → 0x24513E31
e. 0x4355F127 → 0x27F15543
f. 0x5C1D5BE0 → 0xE05B1D5C
g. 0x10793A0E → 0x0E3A7910
h. 0x01DA0D17 → 0x170DDA01

5. Split the 128-bit value into four 32-bit words and write the value into the respective registers
   a. PWDFACTORYRESET[0]←0xD46B22E6
   b. PWDFACTORYRESET[1]←0x39B5B93F
   c. PWDFACTORYRESET[2]←0xACAB64E3
   d. PWDFACTORYRESET[3]←0x24513E31
   e. PWDFACTORYRESET[4]←0x27F15543
   f. PWDFACTORYRESET[5]←0xE05B1D5C
   g. PWDFACTORYRESET[6]←0x0E3A7910
   h. PWDFACTORYRESET[7]←0x170DDA01
6. Repeat the same steps for any function that needs password protection

### *41.2.4 Mailbox (DSSM)*

The debug subsystem mailbox (DSSM) enables a debug probe to pass messages to the target device through the SWD interface, as well as making it possible for the target device to return data to the debug probe.

The DSSM supports the following functions:

- Transmission of commands to the device during boot, including authenticating the debug probe for password-protected debug, mass erase, and factory reset operations
- Communicating with application software running on the target device when no other communication interface is present

Two 32-bit word data buffers are provided for TX data (debug probe to target device) and RX data (target device to debug probe). These data buffers are implemented as 32-bit memory-mapped registers in the DEBUGSS. In addition, TXCTL and RXCTL registers are provided for enabling flow control and indicating status of the mailbox.

**Table 41-7. DSSM Register Functions**

| DSSM Register | Description | Debug Probe | Target Device | Actions |
|---|---|---|---|---|
| TX_DATA | Data buffer | RW | R | TXCTL.TRANSMIT is set on write by the debug probe, and cleared on a read by the target device; TXIFG is also set on a write by the debug probe |
| TXCTL | Flow control and status | RW | R | None |
| RX_DATA | Data buffer | R | RW | RXCTL.RECEIVE is set on write by the target device, and cleared on a read by the debug probe; RXIFG is also set on a write by the target device |
| RXCTL | Flow control and status | R | RW | None |

The TXCTL and RXCTL registers provide generic transmit and receive flags. BIT0 of the TXCTL and RXCTL registers provide a specific TRANSMIT and RECEIVE flag which indicate the status of the TX_DATA and RX_DATA. The TRANSMIT bit is set in the TXCTL register when a debug probe writes data to the TX_DATA buffer register. The TRANSMIT flag will then remain set until the CPU in the target device reads TX_DATA or a POR occurs. The RECEIVE flag is set in the RXCTL register when the CPU in the target device writes data to



Copyright © 2026 Texas Instruments Incorporated

the RX_DATA buffer register. The RECEIVE flag will then remain set until the debug probe reads the data from RX_DATA.

It is not possible for software running on the target device to write to TX_DATA, and it is also not possible for target software to clear the TRANSMIT flag other than by reading TX_DATA.

The upper 31 bits of the TXCTL register, TRANSMIT_FLAGS, contain generic flag bits which can be set or cleared by the debug probe to implement a protocol if desired. Only the debug probe can write to the TRANSMIT_FLAGS field in TXCTL. The DSSM Commands is an example of a protocol to communicate with the target device. Also refer to *Boot Configuration* for security level control related to the DSSM commands.

In a similar way, only the target device software can write to RX_DATA and RXCTL. The debug probe cannot write to RX_DATA and it can only clear the RECEIVE flag in RXCTL by reading RX_DATA. BIT1 through BIT7 (0xFE) of RXCTL contains the RECEIVE_FLAGS field. Software on the target device can set or clear bits in the RECEIVE_FLAGS field to implement a protocol if desired. These flags can be read by the debug probe but can not be modified by the debug probe.

**Table 41-8. DEBUGSS SEC-AP Register Definition**

| AP SEL | ADDR | BANK | INDEX | REGISTER NAME | BIT 31 - 8 | BIT 7 - 1 | BIT 0 |
|---|---|---|---|---|---|---|---|
| 2 | 0x00 | 0 | 0 | TXDATA | TX DATA | | |
| | 0x04 | 0 | 1 | TXCTL | TRANSMIT_FLAGS | | TRANSMIT |
| | 0x08 | 0 | 2 | RXDATA | RX Data | | |
| | 0x0C | 0 | 3 | RXCTL | Reserved | RECEIVE_FLAGS | RECEIVE |
| | 0x0FC | 15 | 3 | IDR | Access point ID | | |

#### 41.2.4.1 DSSM Events

The DSSM contains one event publisher and no event subscribers. One event publisher (CPU_INT) manages DSSM interrupt requests (IRQs) to the CPU subsystem through a static event route.

The DSSM events are summarized in Table 41-9.

**Table 41-9. DSSM Events**

| Event | Type | Source | Destination | Route | Configuration | Functionality |
|---|---|---|---|---|---|---|
| CPU Interrupt Event | Publisher | DEBUGSS | CPU Subsystem | Static route | CPU_INT registers | Fixed interrupt route from DEBUGSS to CPU |

##### *41.2.4.1.1 CPU Interrupt Event (CPU_INT)*

The DSSM provides 4 interrupt sources which can be configured to source a CPU interrupt event. In order of decreasing interrupt priority, the CPU interrupt events from the DSSM are given in Table 41-10.

**Table 41-10. DSSM CPU Interrupt Event Conditions (CPU_INT)**

| Index (IIDX) | Name | Description |
|---|---|---|
| 0 | TXIFG | Indicates that the TX_DATA buffer in the DSSM has received data. |
| 1 | RXIFG | Indicates that the data in RX_DATA buffer in the DSSM was read. |
| 2 | PWRUPIFG | Indicates that the DEBUGSS was started due to a debug probe attaching to the device. |
| 3 | PWRDWNIFG | Indicates that the DEBUGSS was stopped due to a debug probe disconnecting from the device. |

The CPU interrupt event configuration is managed with the CPU_INT event management registers. See Section 7.2.5 for guidance on configuring the Event registers for CPU interrupts.



Copyright © 2026 Texas Instruments Incorporated

#### 41.2.4.2 Reference

DSSM commands are serviced by the bootcode upon a BOOTRST. If passwords are enabled for DSSM commands, the commands do not execute until the correct password sequence is entered. Refer to Table 41-11 for possible DSSM commands and values.

**Table 41-11. DSSM Commands**

| DSSM Command | DSSM Value |
|---|---|
| Factory Reset | 0x020Ah |
| Mass Erase | 0x020Ch |
| Password Authentication | 0x030Eh |
| Data Exchange | 0x00EEh |
| Wait for debug | 0x0206h |

**Factory Reset**

The "Factory Reset" command erases all contents within main and nonmain memory, then nonmain is re-populated with default contents. This command is useful in the cases of:

- Nonmain misconfiguration
- Disabled debug access
- Peripheral/device misconfiguration

The factory reset has its own configuration and password refer to for security level details.

**Note**

The factory reset restores the NONMAIN configuration memory and therefore all previous security level settings are lost.

**Mass Erase**

The "Mass Erase" command erases all contents within the main memory but nonmain remains untouched. This command is useful in the case of:

- Peripheral/device misconfiguration

**Password Authentication**

The "Password Authentication" command unlocks debug access after the correct password has been processed.

**Data Exchange**

Data exchange is the only DSSM command that does not require a BOOTRST to process the command. This command is used in combination with factory reset, mass erase, or password authentication which do require a password. After sending the initial command to the mailbox and performing a reset, the user must begin sending the password to the TXDATA register. After each word 0x00EEh must be written to the TXCTL register.

**Wait for Debug**

The "Wait for Debug" command resets the peripherals defined by the reset level and then forces the device into the reset handler.



Copyright © 2026 Texas Instruments Incorporated

### Custom DSSM Command

The user can also create a custom DSSM command and perform actions defined by the user. This is done by having the debugger communicate to the CPU core through the TXDATA and TXCTL registers. The core can then receive messages from the debugger and send responses back by using the RXDATA and RXCTL registers for the debugger to read. CPU interrupt events can be configured for activity seen in the TX_DATA buffer, RX_DATA buffer, and DAP connection.



Copyright © 2026 Texas Instruments Incorporated

## 41.3 DEBUGSS Registers

Table 41-12 lists the memory-mapped registers for the DEBUGSS registers. All register offset addresses not listed in Table 41-12 should be considered as reserved locations and the register contents should not be modified.

**Table 41-12. DEBUGSS Registers**

| Offset | Acronym | Register Name | Group | Section |
|---|---|---|---|---|
| 1020h | IIDX | Interrupt index | CPU_INT | Go |
| 1028h | IMASK | Interrupt mask | CPU_INT | Go |
| 1030h | RIS | Raw interrupt status | CPU_INT | Go |
| 1038h | MIS | Masked interrupt status | CPU_INT | Go |
| 1040h | ISET | Interrupt set | CPU_INT | Go |
| 1048h | ICLR | Interrupt clear | CPU_INT | Go |
| 10E0h | EVT_MODE | Event Mode | | Go |
| 10FCh | DESC | Module Description | | Go |
| 1100h | TXD | Transmit data register | | Go |
| 1104h | TXCTL | Transmit control register | | Go |
| 1108h | RXD | Receive data register | | Go |
| 110Ch | RXCTL | Receive control register | | Go |
| 1200h | SPECIAL_AUTH | Special enable authorization register | | Go |
| 1210h | APP_AUTH | Application CPU0 authorization register | | Go |

Complex bit access types are encoded to fit into small table cells. Table 41-13 shows the codes that are used for access types in this section.

**Table 41-13. DEBUGSS Access Type Codes**

| Access Type | Code | Description |
|---|---|---|
| **Read Type** | | |
| R | R | Read |
| R-0 | R<br>-0 | Read<br>Returns 0s |
| **Write Type** | | |
| W | W | Write |
| WK | W<br>K | Write<br>Write protected by a key |
| **Reset or Default Value** | | |
| *-n* | | Value after reset or the default value |



Copyright © 2026 Texas Instruments Incorporated

### 41.3.1 IIDX (Offset = 1020h) [Reset = 00000000h]

IIDX is shown in Figure 41-2 and described in Table 41-14.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. 0xFF means no event pending. Interrupt 0x0 is the highest priority, 0x1 next highest, and 0xFE is the least priority. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred.
On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in the RIS and MIS are cleared as well. After a read from the CPU (not from the debug interface), the register must be updated with the next highest priority interrupt, if none are pending, then it displays 0xFF.

**Figure 41-2. IIDX**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | | | | | | | | | | | | | | | | | | STAT | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | R-0h | | | | | | | |

**Table 41-14. IIDX Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | |
| 7-0 | STAT | R | 0h | Interrupt index status<br>0h = No pending interrupt request<br>1h = TX interrupt<br>2h = RX interrupt<br>3h = Power-up interrupt. A debug session has started.<br>4h = Power-up interrupt. A debug session has started. |



Copyright © 2026 Texas Instruments Incorporated

### 41.3.2 IMASK (Offset = 1028h) [Reset = 00000000h]

IMASK is shown in Figure 41-3 and described in Table 41-15.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is unmasked. Unmasking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

**Figure 41-3. IMASK**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | PWRDWNIFG | PWRUPIFG | RXIFG | TXIFG |
| R/W-0h | | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

**Table 41-15. IMASK Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R/W | 0h | |
| 3 | PWRDWNIFG | R/W | 0h | Masks PWRDWNIFG in MIS register<br>0h = Interrupt is masked out<br>1h = Interrupt will request an interrupt service routine and corresponding bit in MIS will be set |
| 2 | PWRUPIFG | R/W | 0h | Masks PWRUPIFG in MIS register<br>0h = Interrupt is masked out<br>1h = Interrupt will request an interrupt service routine and corresponding bit in MIS will be set |
| 1 | RXIFG | R/W | 0h | Masks RXIFG in MIS register<br>0h = Interrupt is masked out<br>1h = Interrupt will request an interrupt service routine and corresponding bit in MIS will be set |
| 0 | TXIFG | R/W | 0h | Masks TXIFG in MIS register<br>0h = Interrupt is masked out<br>1h = Interrupt will request an interrupt service routine and corresponding bit in MIS will be set |



Copyright © 2026 Texas Instruments Incorporated

### 41.3.3 RIS (Offset = 1030h) [Reset = 00000000h]

RIS is shown in Figure 41-4 and described in Table 41-16.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

**Figure 41-4. RIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | PWRDWNIFG | PWRUPIFG | RXIFG | TXIFG |
| R-0h | | | | R-0h | R-0h | R-0h | R-0h |

**Table 41-16. RIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | PWRDWNIFG | R | 0h | Raw interrupt status for PWRDWNIFG<br>0h = PWRUPIFG did not occur<br>1h = PWRUPIFG occurred |
| 2 | PWRUPIFG | R | 0h | Raw interrupt status for PWRUPIFG<br>0h = PWRUPIFG did not occur<br>1h = PWRUPIFG occurred |
| 1 | RXIFG | R | 0h | Raw interrupt status for RXIFG<br>0h = RXIFG did not occur<br>1h = RXIFG occurred |
| 0 | TXIFG | R | 0h | Raw interrupt status for TXIFG<br>0h = TXIFG did not occur<br>1h = TXIFG occurred |



Copyright © 2026 Texas Instruments Incorporated

### 41.3.4 MIS (Offset = 1038h) [Reset = 00000000h]

MIS is shown in Figure 41-5 and described in Table 41-17.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

**Figure 41-5. MIS**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | PWRDWNIFG | PWRUPIFG | RXIFG | TXIFG |
| R-0h | | | | R-0h | R-0h | R-0h | R-0h |

**Table 41-17. MIS Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | PWRDWNIFG | R | 0h | Masked interrupt status for PWRDWNIFG<br>0h = PWRUPIFG did not request an interrupt service routine<br>1h = PWRUPIFG requests an interrupt service routine |
| 2 | PWRUPIFG | R | 0h | Masked interrupt status for PWRUPIFG<br>0h = PWRUPIFG did not request an interrupt service routine<br>1h = PWRUPIFG requests an interrupt service routine |
| 1 | RXIFG | R | 0h | Masked interrupt status for RXIFG<br>0h = RXIFG did not request an interrupt service routine<br>1h = RXIFG requests an interrupt service routine |
| 0 | TXIFG | R | 0h | Masked interrupt status for TXIFG<br>0h = TXIFG did not request an interrupt service routine<br>1h = TXIFG requests an interrupt service routine |



Copyright © 2026 Texas Instruments Incorporated

### 41.3.5 ISET (Offset = 1040h) [Reset = 00000000h]

ISET is shown in Figure 41-6 and described in Table 41-18.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

**Figure 41-6. ISET**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | PWRDWNIFG | PWRUPIFG | RXIFG | TXIFG |
| W-0h | | | | W-0h | W-0h | W-0h | W-0h |

**Table 41-18. ISET Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | W | 0h | |
| 3 | PWRDWNIFG | W | 0h | Sets PWRDWNIFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to PWRUPIFG is set |
| 2 | PWRUPIFG | W | 0h | Sets PWRUPIFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to PWRUPIFG is set |
| 1 | RXIFG | W | 0h | Sets RXIFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to RXIFG is set |
| 0 | TXIFG | W | 0h | Sets TXIFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to TXIFG is set |



Copyright © 2026 Texas Instruments Incorporated

### 41.3.6 ICLR (Offset = 1048h) [Reset = 00000000h]

ICLR is shown in Figure 41-7 and described in Table 41-19.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

**Figure 41-7. ICLR**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | PWRDWNIFG | PWRUPIFG | RXIFG | TXIFG |
| W-0h | | | | W-0h | W-0h | W-0h | W-0h |

**Table 41-19. ICLR Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | W | 0h | |
| 3 | PWRDWNIFG | W | 0h | Clears PWRDWNIFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to PWRUPIFG is cleared |
| 2 | PWRUPIFG | W | 0h | Clears PWRUPIFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to PWRUPIFG is cleared |
| 1 | RXIFG | W | 0h | Clears RXIFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to RXIFG is cleared |
| 0 | TXIFG | W | 0h | Clears TXIFG in RIS register<br>0h = Writing a 0 has no effect<br>1h = RIS bit corresponding to TXIFG is cleared |



Copyright © 2026 Texas Instruments Incorporated

### 41.3.7 EVT_MODE (Offset = 10E0h) [Reset = 00000001h]

EVT_MODE is shown in Figure 41-8 and described in Table 41-20.

Return to the Summary Table.

Event mode register. It is used to select whether each line is disabled, in software mode (software clears the RIS) or in hardware mode (hardware clears the RIS)

**Figure 41-8. EVT_MODE**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R- | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R- | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R- | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | | | INT0_CFG | |
| R- | | | | | | R-1h | |

**Table 41-20. EVT_MODE Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | |
| 1-0 | INT0_CFG | R | 1h | Event line mode select for peripheral events<br>0h = The interrupt or event line is disabled.<br>1h = The interrupt or event line is in software mode. Software must clear the RIS.<br>2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag. |



Copyright © 2026 Texas Instruments Incorporated

### 41.3.8 DESC (Offset = 10FCh) [Reset = 03400000h]

DESC is shown in Figure 41-9 and described in Table 41-21.

Return to the Summary Table.

This register identifies the peripheral and its exact version.

**Figure 41-9. DESC**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MODULEID | | | | | | | | | | | | | | | |
| R-340h | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| FEATUREVER | | | | INSTNUM | | | | MAJREV | | | | MINREV | | | |
| R-0h | | | | R-0h | | | | R-0h | | | | R-0h | | | |

**Table 41-21. DESC Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-16 | MODULEID | R | 340h | Module identification contains a unique peripheral identification number. The assignments are maintained in a central database for all of the platform modules to ensure uniqueness. |
| 15-12 | FEATUREVER | R | 0h | Feature Set for the module *instance* |
| 11-8 | INSTNUM | R | 0h | Instance Number within the device. This will be a parameter to the RTL for modules that can have multiple instances |
| 7-4 | MAJREV | R | 0h | Major rev of the IP |
| 3-0 | MINREV | R | 0h | Minor rev of the IP |



Copyright © 2026 Texas Instruments Incorporated

### 41.3.9 TXD (Offset = 1100h) [Reset = 00000000h]

TXD is shown in Figure 41-10 and described in Table 41-22.

Return to the Summary Table.

This register is used for data transfers from external debug tools to the DSSM module. The register is written by the debug tool and read by the CPU.

**Figure 41-10. TXD**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TX_DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 41-22. TXD Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | TX_DATA | R | 0h | Contains data written by an external debug tool to the SEC-AP TXDATA register |



Copyright © 2026 Texas Instruments Incorporated

### 41.3.10 TXCTL (Offset = 1104h) [Reset = 00000000h]

TXCTL is shown in Figure 41-11 and described in Table 41-23.

Return to the Summary Table.

Transmit control register

**Figure 41-11. TXCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| TRANSMIT_FLAGS | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TRANSMIT_FLAGS | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TRANSMIT_FLAGS | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TRANSMIT_FLAGS | | | | | | | TRANSMIT |
| R-0h | | | | | | | R-0h |

**Table 41-23. TXCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-1 | TRANSMIT_FLAGS | R | 0h | Generic TX flags that can be set by external debug tool. Functionality is defined by SW. |
| 0 | TRANSMIT | R | 0h | Indicates data request in DSSM.TXD, set on write via Debug AP to DSSM.TXD.<br>A read of the DSSM.TXD register by SW will clear the TX field. The tool can check that TXD is empty by reading this field.<br>0h = TXD is empty<br>1h = TXD is full |



Copyright © 2026 Texas Instruments Incorporated

### 41.3.11 RXD (Offset = 1108h) [Reset = 00000000h]

RXD is shown in Figure 41-12 and described in Table 41-24.

Return to the Summary Table.

Receive data register. This register contains the data written by the CPU.
This data is read by external debug tool.

**Figure 41-12. RXD**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RX_DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R/W-0h | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 41-24. RXD Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-0 | RX_DATA | R/W | 0h | Contains data written by SM/OW. |



Copyright © 2026 Texas Instruments Incorporated

### 41.3.12 RXCTL (Offset = 110Ch) [Reset = 00000000h]

RXCTL is shown in Figure 41-13 and described in Table 41-25.

Return to the Summary Table.

Receive control register

**Figure 41-13. RXCTL**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RECEIVE_FLAGS | | | | | | | RECEIVE |
| R/W-0h | | | | | | | R-0h |

**Table 41-25. RXCTL Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-8 | RESERVED | R/W | 0h | |
| 7-1 | RECEIVE_FLAGS | R/W | 0h | Generic RX flags that can be set by SW and read by external debug tool. Functionality is defined by SW. |
| 0 | RECEIVE | R | 0h | Indicates SW write to the DSSM.RXD register.<br>A read of the DSSM.RXD register by SWD Access Port will clear the RX field.<br>0h = RXD empty<br>1h = RXD full |



Copyright © 2026 Texas Instruments Incorporated

### 41.3.13 SPECIAL_AUTH (Offset = 1200h) [Reset = 00000013h]

SPECIAL_AUTH is shown in Figure 41-14 and described in Table 41-26.

Return to the Summary Table.

This register is used to control ET-AP, DFT-TAP, SWD, CFG-AP and SEC-AP.

**Figure 41-14. SPECIAL_AUTH**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | PWRAPEN | AHBAPEN | CFGAPEN | ETAPEN | DFTAPEN | SWDPORTEN | SECAPEN |
| R-0h | R-0h | R-0h | R-1h | R-0h | R-0h | R-1h | R-1h |

**Table 41-26. SPECIAL_AUTH Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-7 | RESERVED | R | 0h | |
| 6 | PWRAPEN | R | 0h | An active high input. When asserted (and SWD access is also permitted), the debug tools can then access the PWR-AP to power and reset state of the CPU. When deasserted, a DAPBUS firewall will isolate the AP and prevent access.<br>0h = Disable PWR-AP<br>1h = Enable PWR-AP |
| 5 | AHBAPEN | R | 0h | Disabling / enabling debug access to the M0+ Core via the AHB-AP DAP bus isolation.<br>0h = Disable AHB-AP<br>1h = Enable AHB-AP |
| 4 | CFGAPEN | R | 1h | An active high input. When asserted (and SWD access is also permitted), the debug tools can use the Config-AP to read device configuration information. When deasserted, a DAPBUS firewall will isolate the AP and prevent access to the Config-AP.<br>0h = Disable CFG-AP<br>1h = Enable CFG-AP |
| 3 | ETAPEN | R | 0h | An active high input. When asserted (and SWD access is also permitted), the debug tools can then access an ET-AP external to the DebugSS lite. When deasserted, a DAPBUS firewall will isolate the AP and prevent access.<br>0h = Disable ET+ -AP<br>1h = Enable ET+ -AP |
| 2 | DFTAPEN | R | 0h | An active high input. When asserted (and SWD access is also permitted), the debug tools can then access the DFT-AP external to the DebugSS lite. When deasserted, a DAPBUS firewall will isolate the AP and prevent access.<br>0h = Disable DFT-TAP<br>1h = Enable DFT-TAP |
| 1 | SWDPORTEN | R | 1h | When asserted, the SW-DP functions normally.<br>When deasserted, the SW-DP effectively disables all external debug access.<br>0h = Disable SWD port<br>1h = Enable SWD port |



Copyright © 2026 Texas Instruments Incorporated

**Table 41-26. SPECIAL_AUTH Field Descriptions (continued)**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 0 | SECAPEN | R | 1h | An active high input. When asserted (and SWD access is also permitted), the debug tools can use the Security-AP to communicate with security control logic. When deasserted, a DAPBUS firewall will isolate the AP and prevent access to the Security-AP.<br>0h = Disable SEC-AP<br>1h = Enable SEC-AP |



Copyright © 2026 Texas Instruments Incorporated

### 41.3.14 APP_AUTH (Offset = 1210h) [Reset = 00000000h]

APP_AUTH is shown in Figure 41-15 and described in Table 41-27.

Return to the Summary Table.

This register is used to control DBGEN, NIDEN, SPIDEN, and SPNIDEN of Application CPU0. DBGEN, NIDEN are further processed by DSW based on Active and Debug IPF ID.

**Figure 41-15. APP_AUTH**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RESERVED | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESERVED | | | | SPNIDEN | SPIDEN | NIDEN | DBGEN |
| R-0h | | | | R-0h | R-0h | R-0h | R-0h |

**Table 41-27. APP_AUTH Field Descriptions**

| Bit | Field | Type | Reset | Description |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | |
| 3 | SPNIDEN | R | 0h | Secure noninvasive debug enable.<br>0h = Invasive debug disabled<br>1h = Invasive debug enabled |
| 2 | SPIDEN | R | 0h | Secure invasive debug enable.<br>0h = Invasive debug disabled<br>1h = Invasive debug enabled |
| 1 | NIDEN | R | 0h | Controls noninvasive debug enable.<br>0h = Non-invasive debug disabled<br>1h = Non-invasive debug enabled |
| 0 | DBGEN | R | 0h | Controls invasive debug enable.<br>0h = Invasive debug disabled<br>1h = Invasive debug enabled |



Copyright © 2026 Texas Instruments Incorporated

# Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

**Changes from March 1, 2026 to July 30, 2026 (from Revision D (March 2026) to Revision E (July 2026))** **Page**

- Added table that provides an overview of NONMAIN memory types....30
- Added NONMAIN memory types register maps....30
- Added a note to clarify peripheral enable requirement during a DMA transfer....533
- Added description on memory areas which are exempt from DMA address or CPU hard fault erros....534
- Updated FLASHCTL registers....585
- Added a note for saved context ready interrupt usage....870
- Clarified reading from DATA_CAPTURE after IRQ....989
- Added clarification on the dependency of MFPCLK for sample time generator operation....1163
- Added description on the MMR read/write behavior during a debug halt....2257
- Added clarifying note to TIMB event duration measurement example....2261
- Updated number of BPU and DWTs available for M33C. Clarified Debug abilities and access....2532
- Updated where to find SOC password implementation....2538



Copyright © 2026 Texas Instruments Incorporated

# IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you fully indemnify TI and its representatives against any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale, TI’s General Quality Guidelines, or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products. Unless TI explicitly designates a product as custom or customer-specified, TI products are standard, catalog, general purpose devices.

TI objects to and rejects any additional or different terms you may propose.

Copyright © 2026, Texas Instruments Incorporated

Last updated 10/2025