MSPM33 C3-Series 160MHz Microcontrollers

Technical Reference Manual

Literature Number: SLAU962
DECEMBER 2025

Table of Contents

Read This First 13
About This Manual 13
Notational Conventions 13
Glossary 13
Related Documentation 13
Support Resources 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 18
1.3.4 System PPB Region 18
1.4 Boot Configuration 18
1.4.1 Configuration Memory 18
1.4.2 NON_MAIN_Configuration Registers 20
1.5 Factory Constants 115
1.5.1 FACTORYREGION Registers 116
1.6 Memory configuration 135
1.6.1 memcfg Registers 136
2 PMCU 142
2.1 PMCU Overview 143
2.1.1 Power Domains 144
2.1.2 Operating Modes 144
2.2 Power Management (PMU) 146
2.2.1 Power Supply 147
2.2.2 Core Regulator 147
2.2.3 Supply Supervisors 147
2.2.4 Bandgap Reference 149
2.2.5 VBOOST for Analog Muxes 149
2.2.6 Peripheral Enable 151
2.3 Clock Module (CKM) 152
2.3.1 Oscillators 152
2.3.2 Clocks 161
2.3.3 Clock Tree 166
2.4 Clock Monitors 168
2.4.1 LFCLK Monitor 168
2.4.2 MCLK Monitor 170
2.4.3 Startup Monitors 170
2.5 Frequency Clock Counter (FCC) 171
2.5.1 Using the FCC 172
2.5.2 FCC Frequency Computation and Accuracy 174
2.6 System Controller (SYSCTL) 174
2.6.1 Resets and Device Initialization 174
2.6.2 Operating Mode Selection 182
2.6.3 Asynchronous Fast Clock Requests 183
2.6.4 SRAM Write Protection 185
2.6.5 Flash Wait States 186

2.6.6 Flash Bank Address Swap ... 186
2.6.7 Shutdown Mode Handling (if present) ... 186
2.6.8 Configuration Lockout ... 187
2.6.9 System Status ... 187
2.6.10 Error Handling ... 188
2.6.11 SYSCTL Events ... 189
2.7 Quick Start Reference ... 191
2.7.1 Default Device Configuration ... 191
2.7.2 Leveraging MFCLK ... 191
2.7.3 Optimizing Power Consumption in STOP Mode ... 192
2.7.4 Optimizing Power Consumption in STANDBY Mode ... 192
2.7.5 Increasing MCLK and ULPCLK Precision ... 192
2.7.6 Configuring MCLK for Maximum Speed ... 192
2.7.7 High Speed Clock (SYSPLL, HFCLK) Handling in Low-Power Modes ... 192
2.7.8 Optimizing for Lowest Wakeup Latency ... 193
2.8 SYSCTL Registers ... 194
3 CPU ... 270
3.1 Overview ... 271
3.2 CPU ... 271
3.2.1 Arm Cortex-M33 CPU ... 271
3.2.2 CPU Register File ... 272
3.2.3 Stack Behavior ... 275
3.2.4 Execution Modes and Privilege Levels ... 275
3.2.5 Address Space and Supported Data Sizes ... 275
3.2.6 Secure memory partitioning ... 276
3.3 Interrupts and Exceptions ... 276
3.3.1 Peripheral Interrupts (IRQs) ... 277
3.3.2 Interrupt and Exception Table ... 280
3.3.3 Processor Lockup Scenario ... 280
3.4 CPU Peripherals ... 281
3.4.1 System Control Block (SCB) ... 281
3.4.2 System Tick Timer (SysTick) ... 281
3.4.3 Memory Protection Unit (MPU) ... 282
3.4.4 Security Attribute Unit ... 284
3.4.5 Implementation Defined Attribution Unit (IDAU) ... 285
3.4.6 Floating Point Unit (FPU) ... 285
3.4.7 Digital Signal Processing Extension ... 285
3.5 Read-Only Memory (ROM) ... 286
4 Secure ROM ... 287
4.1 ROM Overview ... 288
4.2 Memory Map ... 288
4.3 Boot Configuration Routine (BCR) ... 288
4.3.1 SWD Mass Erase and Factory Reset Commands ... 288
4.3.2 Application HASH Verification ... 289
4.3.3 Fast Boot ... 289
4.4 Bootstrap Loader (BSL) ... 289
4.4.1 Application Version ... 290
4.4.2 GPIO Invoke ... 290
4.4.3 BSL Triggered Mass Erase and Factory Reset ... 290
4.5 Lifecycle Management ... 291
4.5.1 Device Sub-Type ... 291
4.5.2 Lifecycle Transitions ... 292
4.6 Boot and Startup Sequence ... 292
4.6.1 Secure Boot ... 294
4.6.2 Customer Secure Code (CSC) ... 295
5 NVM (Flash) ... 296
5.1 NVM Overview ... 297
5.1.1 Key Features ... 297
5.1.2 System Components ... 297
5.1.3 Terminology ... 297
5.2 Flash Memory Bank Organization ... 298

5.2.1 Banks....298
5.2.2 Flash Memory Regions....298
5.2.3 Addressing....298
5.3 Flash Controller....299
5.3.1 Overview of Flash Controller Commands....299
5.3.2 NOOP Command....300
5.3.3 PROGRAM Command....300
5.3.4 ERASE Command....304
5.3.5 READVERIFY Command....305
5.3.6 Command Diagnostics....306
5.3.7 Overriding the System Address With a Bank ID, Region ID, and Bank Address....306
5.4 Flash Programming Interface....307
5.4.1 Flash Resource Ownership Check....307
5.4.2 Authorization Check....307
5.4.3 FPI SEC Error Handling....313
5.4.4 Bank Erase Protection....313
5.5 Flash Read Interface....314
5.5.1 Bank Address Swapping....314
5.5.2 ECC Error Handling....314
5.5.3 GSC SEC Error Handling....315
5.6 FLASHCTL Registers....316
5.7 FRI Registers....321
6 EAM....327
6.1 EAM Introduction....328
6.2 EAM Operation....328
6.2.1 Security Error Aggregator....328
6.2.2 Safety Error Aggregator....329
6.3 EAM Registers....330
7 Direct Memory Access (DMA)....352
7.1 DMA Overview....353
7.2 DMA Operation....354
7.2.1 Addressing Modes....354
7.2.2 Channel Types....356
7.2.3 Transfer Modes....357
7.2.4 Extended Modes....359
7.2.5 Initiating DMA Transfers....361
7.2.6 Stopping DMA Transfers....362
7.2.7 Channel Priorities....362
7.2.8 Burst Block Mode....362
7.2.9 Using DMA with System Interrupts....362
7.2.10 DMA Controller Interrupts....363
7.2.11 DMA Trigger Event Status....363
7.2.12 DMA Operating Mode Support....363
7.2.13 DMA Address and Data Errors....364
7.2.14 Interrupt and Event Support....364
7.3 DMA Registers....365
8 Events....413
8.1 Events Overview....414
8.1.1 Event Publisher....414
8.1.2 Event Subscriber....414
8.1.3 Event Fabric Routing....414
8.1.4 Event Routing Map....416
8.1.5 Event Propagation Latency....417
8.2 Events Operation....418
8.2.1 CPU Interrupt....418
8.2.2 DMA Trigger....418
8.2.3 Peripheral to Peripheral Event....419
8.2.4 Extended Module Description Register....420
8.2.5 Using Event Registers....420
9 IOMUX....423
9.1 IOMUX Overview....424

9.1.1 IO Types and Analog Sharing....424
9.2 IOMUX Operation....427
9.2.1 Peripheral Function (PF) Assignment....427
9.2.2 Logic High to Hi-Z Conversion....427
9.2.3 Logic Inversion....428
9.2.4 SHUTDOWN Mode Wakeup Logic....428
9.2.5 Pullup/Pulldown Resistors....429
9.2.6 Drive Strength Control....429
9.2.7 Hysteresis and Logic Level Control....429
9.3 IOMUX Registers....431
10 General-Purpose Input/Output (GPIO)....434
10.1 GPIO Overview....435
10.2 GPIO Operation....435
10.2.1 GPIO Ports....435
10.2.2 GPIO Read/Write Interface....436
10.2.3 GPIO Input Glitch Filtering and Synchronization....436
10.2.4 GPIO Fast Wake....437
10.2.5 Event Publishers and Subscribers....438
10.3 GPIO Registers....439
11 Global Security Controller....547
11.1 GSC Introduction....548
11.1.1 GSC Features....548
11.2 GSC Operation....548
11.2.1 Functional Block Diagram....548
11.2.2 Peripheral Protection Controller....548
11.2.3 SRAM Protection Controller....549
11.2.4 Flash Protection Controller....551
11.2.5 Strict Secure and Privilege Context Protection....555
11.2.6 GSC Configuration Lock....556
11.3 GSC Registers....557
12 PKA....659
12.1 PKA Introduction....660
12.1.1 PKA features....660
12.2 PKA Operation....660
12.2.1 Functional Block Diagram....660
12.2.2 Theory of Operations....660
12.2.3 Complex Commands....663
12.2.4 Command Execution and Status....664
12.2.5 Initialization....664
12.2.6 Interrupts support....665
12.3 PKA Registers....666
13 AES....702
13.1 AES Overview....703
13.1.1 AESADV Performance....703
13.2 AESADV Operation....704
13.2.1 Loading the Key....704
13.2.2 Writing Input Data....704
13.2.3 Reading Output Data....704
13.2.4 Operation Descriptions....705
13.2.5 AES Events....722
13.3 AESADV Registers....724
14 SHA2....794
14.1 SHA Introduction....795
14.1.1 SHA features....795
14.2 SHA Operation....795
14.2.1 Functional Block Diagram....795
14.2.2 HMAC Controller....795
14.2.3 HASH/HMAC Engine....796
14.3 SHA Auto-Feed Mode....798
14.4 SHA Registers....799
15 CRC....864

15.1 CRC Overview ... 865
15.1.1 CRC16-CCITT ... 865
15.1.2 CRC32-ISO3309 ... 865
15.2 CRC Operation ... 865
15.2.1 CRC Generator Implementation ... 866
15.2.2 Configuration ... 866
15.3 CRCP0 Registers ... 868
16 Keystore ... 881
16.1 Overview ... 882
16.2 Detailed Description ... 882
16.3 KEYSTORECTL Registers ... 883
17 TRNG ... 890
17.1 TRNG Overview ... 891
17.2 TRNG Operation ... 891
17.2.1 TRNG Generation Data Path ... 891
17.2.2 Clock Configuration and Output Rate ... 892
17.2.3 Behavior in Low Power Modes ... 892
17.2.4 Health Tests ... 892
17.2.5 Configuration ... 894
17.3 TRNG Registers ... 898
18 HSADC ... 915
18.1 Introduction ... 916
18.1.1 Features ... 916
18.1.2 Block Diagram ... 917
18.2 HSADC Operation ... 917
18.2.1 ADC Configurability ... 918
18.2.2 SOC Principle of Operation ... 920
18.2.3 EOC and Interrupt Operation ... 926
18.2.4 Post-Processing Blocks ... 928
18.2.5 Power-Up Sequence ... 931
18.2.6 ADC Timings ... 931
18.3 ADC_LITE_REGS Registers ... 935
18.4 ADC_LITE_RESULT_REGS Registers ... 1044
19 VREF ... 1082
19.1 VREF Overview ... 1083
19.2 VREF Operation ... 1084
19.2.1 Internal Reference Generation ... 1085
19.2.2 External Reference Input ... 1085
19.2.3 Analog Peripheral Interface ... 1085
19.3 VREF Registers ... 1087
20 COMP ... 1097
20.1 Comparator Overview ... 1098
20.2 Comparator Operation ... 1100
20.2.1 Comparator Configuration ... 1100
20.2.2 Comparator Channels Selection ... 1100
20.2.3 Comparator Output ... 1101
20.2.4 Output Filter ... 1101
20.2.5 Sampled Output Mode ... 1102
20.2.6 Blanking Mode ... 1102
20.2.7 Reference Voltage Generator ... 1102
20.2.8 Comparator Hysteresis ... 1103
20.2.9 Input SHORT Switch ... 1103
20.2.10 Analog Comparison Feature ... 1104
20.2.11 Interrupt and Events Support ... 1106
20.3 COMP Registers ... 1108
21 UNICOMM ... 1137
21.1 Overview ... 1138
21.1.1 Block Diagram ... 1140
21.2 Unicomm Architecture ... 1140
21.2.1 Serial Peripheral Group (SPG) Configurations ... 1140
21.2.2 FIFO Operation ... 1141

21.2.3 Enables & Resets ... 1144
21.2.4 Interrupts ... 1144
21.2.5 DMA Operation ... 1145
21.3 High-Level Initialization ... 1145
21.4 UNICOMM/SPGSS Registers ... 1146
21.4.1 UNICOMM Registers ... 1146
21.4.2 SPG Registers ... 1152
22 UNICOMM UART ... 1162
22.1 UART Overview ... 1163
22.1.1 Purpose of the Peripheral ... 1163
22.1.2 Features ... 1163
22.1.3 Functional Block Diagram ... 1164
22.2 UART Operation ... 1164
22.2.1 Clock Control ... 1164
22.2.2 General Architecture and Protocol ... 1165
22.2.3 Additional Protocol and Feature Support ... 1170
22.2.4 Low Power Operation ... 1181
22.2.5 Reset Considerations ... 1181
22.2.6 UART Initialization ... 1182
22.2.7 Interrupt and Events Support ... 1182
22.2.8 Emulation Modes ... 1184
22.3 UNICOMMUART Registers ... 1186
23 UNICOMM I2C ... 1228
23.1 Overview ... 1229
23.1.1 Purpose of the Peripheral ... 1229
23.1.2 Features ... 1229
23.1.3 Functional Block Diagram ... 1230
23.2 Peripheral Functional Description ... 1230
23.2.1 Clock Control ... 1230
23.2.2 Signal Descriptions ... 1231
23.2.3 General Architecture ... 1232
23.2.4 Protocol Descriptions & Initialization ... 1239
23.2.5 Reset Considerations ... 1249
23.2.6 Initialization ... 1250
23.2.7 Interrupt and Events Support ... 1250
23.2.8 Emulation Modes ... 1252
23.3 UNICOMMI2CC Registers ... 1253
23.4 UNICOMMI2CT Registers ... 1293
24 UNICOMM SPI ... 1332
24.1 Overview ... 1333
24.1.1 Purpose of the Peripheral ... 1333
24.1.2 Features ... 1333
24.1.3 **Features supported in each variant** ... 1333
24.1.4 Functional Block Diagram ... 1334
24.2 UNICOMM common infrastructure ... 1334
24.3 External Connections and Signal Descriptions ... 1334
24.4 SPI Operation ... 1336
24.4.1 Clock Frequency Support ... 1336
24.4.2 General Architecture ... 1336
24.4.3 FIFO Size ... 1339
24.4.4 Internal Loopback Operation ... 1339
24.4.5 Repeat Transfer mode ... 1340
24.4.6 Receive Timeout ... 1340
24.4.7 Line Timeout ... 1341
24.4.8 Protocol Descriptions ... 1341
24.4.9 Status Flags ... 1346
24.4.10 Module pseudo static bit configuration ... 1346
24.4.11 Reset Considerations ... 1346
24.4.12 Initialization ... 1346
24.4.13 Interrupt and Events Support ... 1347
24.4.14 Emulation Modes ... 1347

24.5 UNICOMMSPI Registers....1348
25 QSPI....1376
25.1 QSPI Overview....1377
25.1.1 Purpose of the Peripheral....1377
25.1.2 Features....1377
25.1.3 Functional Block Diagram....1378
25.1.4 External Connections and Signal Descriptions....1378
25.2 QSPI Operation....1380
25.2.1 Clock Control....1380
25.2.2 General Architecture....1380
25.2.3 Reset Considerations....1382
25.2.4 Initialization....1382
25.2.5 QSPI Controller Description....1383
25.2.6 Interrupt and Events Support....1389
25.2.7 Emulation Modes....1390
25.3 QSPI Registers....1391
26 CAN-FD....1435
26.1 MCAN Overview....1436
26.1.1 MCAN Features....1437
26.2 MCAN Environment....1437
26.3 CAN Network Basics....1438
26.4 MCAN Functional Description....1440
26.4.1 Clock Setup....1441
26.4.2 Module Clocking Requirements....1442
26.4.3 Interrupt Requests....1442
26.4.4 Operating Modes....1444
26.4.5 Software Initialization....1446
26.4.6 Transmitter Delay Compensation....1447
26.4.7 Restricted Operation Mode....1449
26.4.8 Bus Monitoring Mode....1450
26.4.9 Disabled Automatic Retransmission (DAR) Mode....1451
26.4.10 Clock Stop Mode....1451
26.4.11 Test Modes....1457
26.4.12 Timestamp Generation....1458
26.4.13 Timeout Counter....1459
26.4.14 Safety....1460
26.4.15 Tx Handling....1462
26.4.16 Rx Handling....1472
26.4.17 Rx FIFOs....1476
26.4.18 Dedicated Rx Buffers....1478
26.4.19 Message RAM....1478
26.5 MCAN Integration....1489
26.6 Interrupt and Event Support....1490
26.6.1 CPU Interrupt Event Publisher (CPU_INT)....1490
26.7 MCAN Registers....1492
27 I2S/TDM....1622
27.1 I2S/TDM Introduction....1623
27.1.1 I2S/TDM features....1623
27.2 I2S/TDM Operation....1623
27.2.1 Functional Block Diagram....1623
27.2.2 Modes of Operation....1623
27.2.3 Clock and Timing Control....1624
27.2.4 Frame Synchronization....1625
27.2.5 Slot Mapping and Configuration....1627
27.2.6 Serial Frame Format Examples....1630
27.2.7 Initialization....1634
27.2.8 Disabling I2S....1635
27.2.9 Interrupts and Events Support....1635
27.2.10 Emulation Modes....1636
27.3 I2S Registers....1637
28 Timers (TIMx)....1675

28.1 TIMx Overview ... 1676
28.1.1 TIMG Overview ... 1676
28.1.2 TIMA Overview ... 1677
28.1.3 TIMx Instance Configuration ... 1679
28.2 TIMx Operation ... 1679
28.2.1 Timer Counter ... 1679
28.2.2 Counting Mode Control ... 1682
28.2.3 Capture/Compare Module ... 1688
28.2.4 Shadow Load and Shadow Compare ... 1702
28.2.5 Output Generator ... 1704
28.2.6 Fault Handler (TIMA only) ... 1714
28.2.7 Synchronization With Cross Trigger ... 1720
28.2.8 Low Power Operation ... 1722
28.2.9 Interrupt and Event Support ... 1723
28.2.10 Debug Handler (TIMA Only) ... 1726
28.3 TIMx Registers ... 1727
29 Low Frequency Subsystem (LFSS) ... 1821
29.1 Overview ... 1822
29.2 Clock System ... 1823
29.3 LFSS Reset Using VBAT ... 1824
29.4 Power Domains and Supply Detection ... 1824
29.4.1 Startup When VBAT Powers on First ... 1825
29.4.2 Startup when VDD powers on first ... 1826
29.4.3 Behavior When VDD is Lost ... 1826
29.4.4 Behavior when VBAT is lost ... 1826
29.4.5 Behavior when the device goes into SHUTDOWN mode ... 1826
29.4.6 Supercapacitor Charging Circuit ... 1826
29.5 Real Time Counter (RTC_x) ... 1827
29.6 Independent Watchdog Timer (IWDT) ... 1827
29.7 Tamper Input and Output ... 1828
29.7.1 IOMUX Mode ... 1829
29.7.2 Tamper Mode ... 1829
29.8 Scratchpad Memory ... 1831
29.9 Lock Function of RTC, TIO, and IWDT ... 1831
29.10 LFSS Registers ... 1833
30 IWDT ... 1954
30.2 IWDT Clock Configuration ... 1955
30.3 IWDT Period Selection ... 1955
30.4 Debug Behavior of the IWDT ... 1955
30.5 IWDT Registers ... 1955
31 Window Watchdog Timer (WWDT) ... 1956
31.1 WWDT Overview ... 1957
31.1.1 Watchdog Mode ... 1957
31.1.2 Interval Timer Mode ... 1958
31.2 WWDT Operation ... 1958
31.2.1 Mode Selection ... 1958
31.2.2 Clock Configuration ... 1959
31.2.3 Low-Power Mode Behavior ... 1960
31.2.4 Debug Behavior ... 1960
31.2.5 WWDT Events ... 1961
31.3 WWDT Registers ... 1962
32 Debug ... 1980
32.1 DEBUGSS Overview ... 1981
32.1.1 Debug Interconnect ... 1981
32.1.2 Physical Interface ... 1982
32.1.3 Debug Access Ports ... 1983
32.2 DEBUGSS Operation ... 1984
32.2.1 Debug Features ... 1984
32.2.2 Behavior in Low Power Modes ... 1985
32.2.3 Restricting Debug Access ... 1986
32.2.4 Mailbox (DSSM) ... 1988

32.3 DEBUGSS Registers 1991
33 Revision History 2007

This page intentionally left blank.

About This Manual

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.

Related Documentation

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.
All trademarks are the property of their respective owners.

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 18
1.5 Factory Constants 115
1.6 Memory configuration 135

1.1 Architecture Overview

MSPM33 C-series MCUs (MSPM33Cx) 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.

MSPM33 C-series devices also offer enhanced robustness with ECC-protected flash memory, ECC-protected SRAM, available dual window-watchdog timers, and support for up to 125°C ambient temperature and AEC-Q100 Grade 1 qualification (see device datasheet).

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 MSPM33Cx 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, and is 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 MSPM33Cx 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/4
- The PD1 (power domain 1) peripheral buses, clocked from MCLK, MCLK/2 or MCLK/4
- The PD0 (power domain 0) peripheral bus, clocked from ULPCLK

The CPU and the DMA controllers are the bus initators in the device. Arbitration between the CPU and the DMAs for shared peripherals happens between the PD1 amd PD0 peripheral bus. The DMA does not have access to peripherals on the CPU-only PD1 peripheral bus or the CPU sub-system. 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.

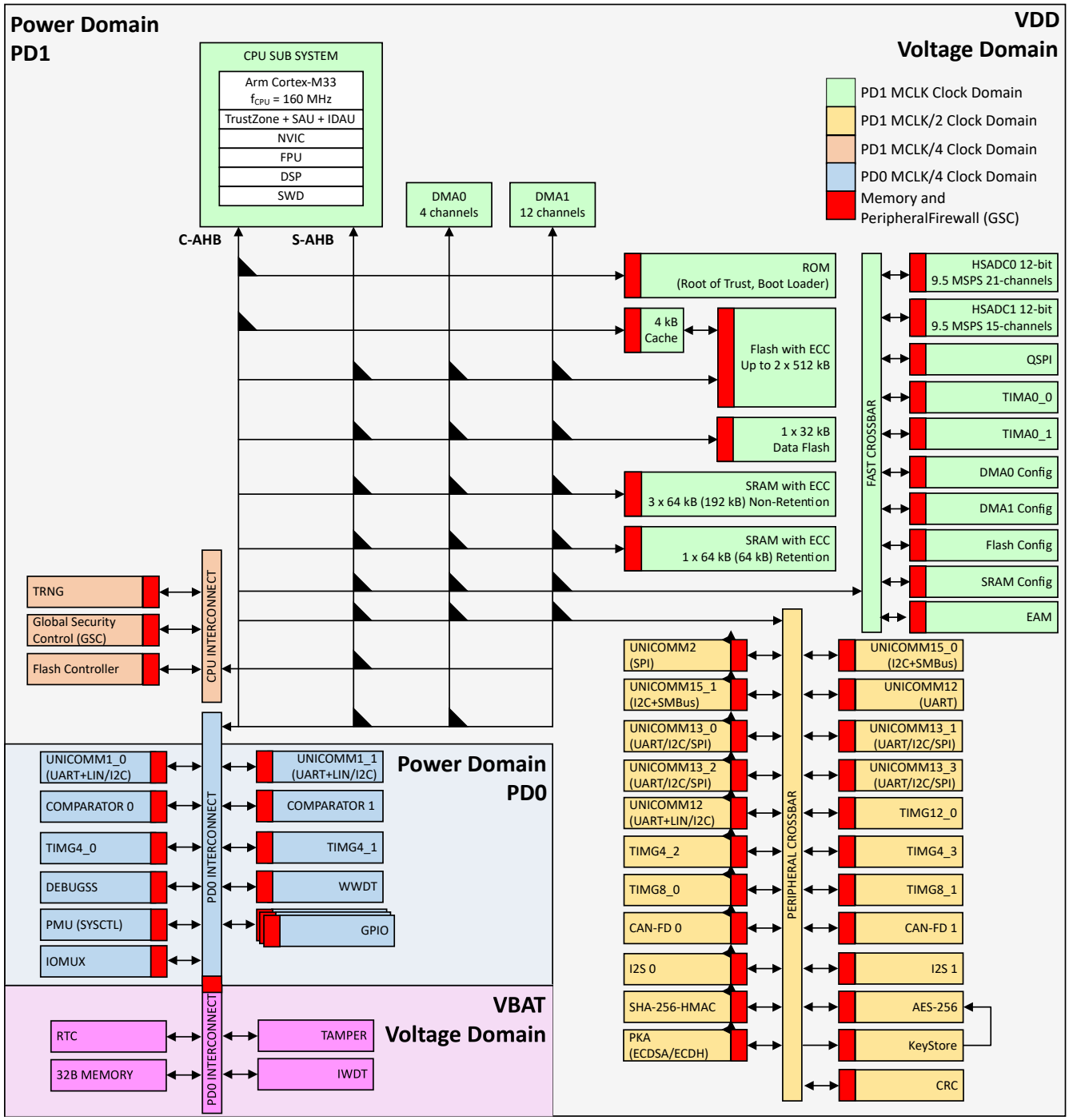


Figure 1-1. MSPM33Cx Bus Organization

Note

This is a generic diagram of the typical peripherals on a MSPM33Cx 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.

1.3 Platform Memory Map

All MSPM33Cx 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	Sub Region	Start Address	End Address	Description
Code	Non-Secure	0x0000.0000	0x0FFF.FFFF	Flash memory and ROM
	Secure	0x1000.0000	0x1FFF.FFFF	
SRAM	Non-Secure	0x2000.0000	0x2FFF.FFFF	SRAM
	Secure	0x3000.0000	0x3FFF.FFFF	
Peripheral	Non-Secure	0x4000.0000	0x4FFF.FFFF	Global peripheral memory-mapped registers and global nonexecutable data memory
	Secure	0x5000.0000	0x5FFF.FFFF	
Subsystem	Non-Secure	0x6000.0000	0x6FFF.FFFF	Local CPU subsystem memory-mapped registers
	Secure	0x7000.0000	0x7FFF.FFFF	
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 in secure address map and the other in non-secure address 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 up to certain MCLK frequency, as documented in the device specific datasheet. 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.

Devices optionally support ECC checking of the SRAM. Refer to the device-specific data sheet to determine if a device supports ECC on a specific SRAM bank. For information on how ECC errors are handled by the device, see EAM module.

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 two address subregions in the overall SRAM region. One is the secure address subregion (0x3000.0000) and the other in the non-secure address subregion (0x2000.0000).

Note

In some devices, the SRAM is organized into two or more banks. Not all the banks may have have ECC. Typically, the lowest number bank will include ECC while additional banks may or may not include ECC.

1.3.3 Peripheral Region

The peripheral region contains the memory-mapped peripherals on the peripheral buses. Just like the code and SRAM region, the peripheral is also aliased in secure and non-secure subregion.

1.3.4 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, security attribute unit (SAU) 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 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 for non-secure boot devices. To enable secure boot, a signed application image contains the SP and reset vector in the header which is interpreted by the ROM and updated in the GSC VTOR location.

1.4.1 Configuration Memory

The configuration memory is a dedicated region of flash memory (not the same as the code region) 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 configuration flash region.

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

The configuration data in this 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 configuration memory 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 configuration memory following the erase. As such, it is the responsibility of the host which is connected to the microcontroller target (via the BSL interface) to re-program the configuration memory with a valid configuration before terminating the BSL session.

Note

If a factory reset command is executed through the BSL, and a valid 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 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.

1.4.2 NON_MAIN_Configuration Registers

Table 1-2 lists the memory-mapped registers for the NON_MAIN_Configuration registers. All register offset addresses not listed in Table 1-2 should be considered as reserved locations and the register contents should not be modified.

Table 1-2. NON_MAIN_CONFIGURATION Registers

Offset	Acronym	Register Name	Section
80101800h	BCR_CONFIG_ID	Predetermined Bootcode config signature ID	Section 1.4.2.1
80101804h	BOOTCFG0	Controls enable/disable of AHB-AP, ET-AP, PWR-AP. Values: DEBUG_EN, DEBUG_EN_PW, DEBUG_NS_EN, DEBUG_NS_EN_PW, DEBUG_DIS; Serial Wire Debug Port mode. Disabling this will disable all Debug ports including CFG_AP, SEC_AP. When disabled no DSSM commands are serviced	Section 1.4.2.2
80101808h	BOOTCFG1	Controls BSL pin invocation capability. Values: BSL_PIN_INVOKE_EN, BSL_PIN_INVOKE_DIS; Controls debug access release until INITDONE is issued	Section 1.4.2.3
8010180Ch	BOOTCFG2	Controls CSC policy in SYSCTL. YES enables CSC policy checking; Controls flash bank swap policy in SYSCTL	Section 1.4.2.4
80101810h	BOOTCFG3	Fast boot mode configuration - Skips certain boot time checks when enabled; Bootloader mode enable/disable control. Must be enabled for BSL functionality	Section 1.4.2.5
80101814h	BOOTCFG4	Mass erase mode configuration. Values: BC_CFG_MASS_ERASE_EN(0xAABB), BC_CFG_MASS_ERASE_EN_PW(0xCCDD), BC_CFG_MASS_ERASE_DIS(0x5522); Factory reset mode configuration. Values: BC_CFG_FACTORY_RESET_EN(0xAABB), BC_CFG_FACTORY_RESET_EN_PW(0xCCDD), BC_CFG_FACTORY_RESET_DIS(0x5522)	Section 1.4.2.6
80101818h	MASS_ERASE_0	Password hash for DSSM_BC_MASS_ERASE_REQUEST command - Word 0	Section 1.4.2.7
8010181Ch	MASS_ERASE_1	Password hash for DSSM_BC_MASS_ERASE_REQUEST command - Word 1	Section 1.4.2.8
80101820h	MASS_ERASE_2	Password hash for DSSM_BC_MASS_ERASE_REQUEST command - Word 2	Section 1.4.2.9
80101824h	MASS_ERASE_3	Password hash for DSSM_BC_MASS_ERASE_REQUEST command - Word 3	Section 1.4.2.10
80101828h	MASS_ERASE_4	Password hash for DSSM_BC_MASS_ERASE_REQUEST command - Word 4	Section 1.4.2.11
8010182Ch	MASS_ERASE_5	Password hash for DSSM_BC_MASS_ERASE_REQUEST command - Word 5	Section 1.4.2.12
80101830h	MASS_ERASE_6	Password hash for DSSM_BC_MASS_ERASE_REQUEST command - Word 6	Section 1.4.2.13
80101834h	MASS_ERASE_7	Password hash for DSSM_BC_MASS_ERASE_REQUEST command - Word 7	Section 1.4.2.14
80101838h	FACTORY_RESET_0	Password hash for factory reset command - Word 0	Section 1.4.2.15
8010183Ch	FACTORY_RESET_1	Password hash for factory reset command - Word 1	Section 1.4.2.16

Table 1-2. NON_MAIN_CONFIGURATION Registers (continued)

Offset	Acronym	Register Name	Section
80101840h	FACTORY_RESET_2	Password hash for factory reset command - Word 2	Section 1.4.2.17
80101844h	FACTORY_RESET_3	Password hash for factory reset command - Word 3	Section 1.4.2.18
80101848h	FACTORY_RESET_4	Password hash for factory reset command - Word 4	Section 1.4.2.19
8010184Ch	FACTORY_RESET_5	Password hash for factory reset command - Word 5	Section 1.4.2.20
80101850h	FACTORY_RESET_6	Password hash for factory reset command - Word 6	Section 1.4.2.21
80101854h	FACTORY_RESET_7	Password hash for factory reset command - Word 7	Section 1.4.2.22
80101858h	DEBUG_LOCK_0	Password hash for debug access authentication - Word 0	Section 1.4.2.23
8010185Ch	DEBUG_LOCK_1	Password hash for debug access authentication - Word 1	Section 1.4.2.24
80101860h	DEBUG_LOCK_2	Password hash for debug access authentication - Word 2	Section 1.4.2.25
80101864h	DEBUG_LOCK_3	Password hash for debug access authentication - Word 3	Section 1.4.2.26
80101868h	DEBUG_LOCK_4	Password hash for debug access authentication - Word 4	Section 1.4.2.27
8010186Ch	DEBUG_LOCK_5	Password hash for debug access authentication - Word 5	Section 1.4.2.28
80101870h	DEBUG_LOCK_6	Password hash for debug access authentication - Word 6	Section 1.4.2.29
80101874h	DEBUG_LOCK_7	Password hash for debug access authentication - Word 7	Section 1.4.2.30
80101878h	DEBUG_NS_LOCK_0	Password hash for non-secure debug access - Word 0	Section 1.4.2.31
8010187Ch	DEBUG_NS_LOCK_1	Password hash for non-secure debug access - Word 1	Section 1.4.2.32
80101880h	DEBUG_NS_LOCK_2	Password hash for non-secure debug access - Word 2	Section 1.4.2.33
80101884h	DEBUG_NS_LOCK_3	Password hash for non-secure debug access - Word 3	Section 1.4.2.34
80101888h	DEBUG_NS_LOCK_4	Password hash for non-secure debug access - Word 4	Section 1.4.2.35
8010188Ch	DEBUG_NS_LOCK_5	Password hash for non-secure debug access - Word 5	Section 1.4.2.36
80101890h	DEBUG_NS_LOCK_6	Password hash for non-secure debug access - Word 6	Section 1.4.2.37
80101894h	DEBUG_NS_LOCK_7	Password hash for non-secure debug access - Word 7	Section 1.4.2.38
80101898h	BAD_CONV_0	This field stores the SHA-256 hash (32 bytes) of the 16-byte password used for bad device conversion. The hash is stored as an array of 8 32-bit words.	Section 1.4.2.39
8010189Ch	BAD_CONV_1	Bad conversion password hash - Word 1	Section 1.4.2.40
801018A0h	BAD_CONV_2	Bad conversion password hash - Word 2	Section 1.4.2.41
801018A4h	BAD_CONV_3	Bad conversion password hash - Word 3	Section 1.4.2.42
801018A8h	BAD_CONV_4	Bad conversion password hash - Word 4	Section 1.4.2.43
801018ACh	BAD_CONV_5	Bad conversion password hash - Word 5	Section 1.4.2.44
801018B0h	BAD_CONV_6	Bad conversion password hash - Word 6	Section 1.4.2.45
801018B4h	BAD_CONV_7	Bad conversion password hash - Word 7	Section 1.4.2.46
801018B8h	SECURE_BOOT_MODE	Controls application authentication; Reserved field	
801018BCh	USER_SECURE_APP_START_ADDR	Starting address for secure application	Section 1.4.2.48
801018C0h	USER_SECURE_APP_LENGTH	Length of secure application	Section 1.4.2.49
801018C4h	USER_SECURE_APP_HASH_0	Secure application hash - Word 0	Section 1.4.2.50
801018C8h	USER_SECURE_APP_HASH_1	Secure application hash - Word 1	Section 1.4.2.51
801018CCh	USER_SECURE_APP_HASH_2	Secure application hash - Word 2	Section 1.4.2.52
801018D0h	USER_SECURE_APP_HASH_3	Secure application hash - Word 3	Section 1.4.2.53
801018D4h	USER_SECURE_APP_HASH_4	Secure application hash - Word 4	Section 1.4.2.54
801018D8h	USER_SECURE_APP_HASH_5	Secure application hash - Word 5	Section 1.4.2.55
801018DCh	USER_SECURE_APP_HASH_6	Secure application hash - Word 6	Section 1.4.2.56
801018E0h	USER_SECURE_APP_HASH_7	Secure application hash - Word 7	Section 1.4.2.57
801018E4h	BANK0_NM_USER_CONFIG	User Config Non Main Flash Static Write Protection. WEP_DISABLE=0xAABB; Reserved field	

Table 1-2. NON_MAIN_CONFIGURATION Registers (continued)

Offset	Acronym	Register Name	Section
801018E8h	BANK0_WRITE_ERASE_PROTECTION_A	Write protection for first 32 sectors	Section 1.4.2.59
801018ECh	BANK0_WRITE_ERASE_PROTECTION_B	Write Protection for 512KB-64KB	Section 1.4.2.60
801018F0h	BANK0_SECURITY_PROTECTION_A	Security protection for first 32 sectors	Section 1.4.2.61
801018F4h	BANK0_SECURITY_PROTECTION_B	Security protection for 512KB-64KB	Section 1.4.2.62
801018F8h	BANK0_PRIVILEGE_PROTECTION_A	Privilege protection for first 32 sectors	Section 1.4.2.63
801018FCh	BANK0_PRIVILEGE_PROTECTION_B	Privilege protection for 512KB-64KB	Section 1.4.2.64
80101900h	BANK0_RESERVED	Reserved field for BANK0 alignment	
80101904h	BANK1_WRITE_ERASE_PROTECTION_A	Write protection for first 32 sectors	Section 1.4.2.65
80101908h	BANK1_WRITE_ERASE_PROTECTION_B	Write Protection for 512KB-64KB	Section 1.4.2.66
8010190Ch	BANK1_SECURITY_PROTECTION_A	Security protection for first 32 sectors	Section 1.4.2.67
80101910h	BANK1_SECURITY_PROTECTION_B	Security protection for 512KB-64KB	Section 1.4.2.68
80101914h	BANK1_PRIVILEGE_PROTECTION_A	Privilege protection for first 32 sectors	Section 1.4.2.69
80101918h	BANK1_PRIVILEGE_PROTECTION_B	Privilege protection for 512KB-64KB	Section 1.4.2.70
8010191Ch	BANK1_RESERVED	Reserved field for BANK1	
80101920h	DBANK_WRITE_ERASE_PROTECTION	Write/Erase protection for Data Bank	Section 1.4.2.71
80101924h	DBANK_SECURITY_PROTECTION	Security protection for Data Bank	Section 1.4.2.72
80101928h	DBANK_PRIVILEGE_PROTECTION	Privilege protection for Data Bank	Section 1.4.2.73
8010192Ch	DBANK_RESERVED	Reserved field for DBANK	
80101930h	BOOTCLK0	Clock configuration; PLL multiplier value; PLL divider value; PLL clock source configuration	Section 1.4.2.74
80101934h	BOOTCLK1	CPU delay cycles for PLL settling time; CPU delay cycles for HFXT monitoring; Reserved configuration field 0	
80101938h	RESERVED_REG_0	Reserved configuration field 0; Reserved configuration field 1	
8010193Ch	RESERVED_REG_1	Reserved configuration field 2; Reserved configuration field 3	
80101940h	RESERVED_REG_2	Reserved configuration field 4; Reserved configuration field 5	
80101944h	RESERVED_REG_3	Reserved configuration field 6; Reserved configuration field 7	
80101948h	RESERVED_REG_4	Reserved configuration field 8; Reserved configuration field 9	
8010194Ch	CRC	CRC-32 of BCR config structure	Section 1.4.2.76
80101C00h	BSL_CONFIG_ID	Predetermined Bootloader config signature ID	Section 1.4.2.77
80101C04h	BSLPINCFG0	UART receive pin number configuration; UART receive pin multiplexer selection; UART transmit pin number configuration; UART transmit pin multiplexer selection	Section 1.4.2.78
80101C08h	BSLPINCFG1	I2C data pin number configuration; I2C data pin multiplexer selection; I2C clock pin number configuration; I2C clock pin multiplexer selection	Section 1.4.2.79
80101C0Ch	BSLPINCFG2	MCAN receive pin number configuration; MCAN receive pin multiplexer selection; MCAN transmit pin number configuration; MCAN transmit pin multiplexer selection	Section 1.4.2.80
80101C10h	BSLCONFIG0	BSL invoke pin configuration data 0; BSL invoke pin configuration data 1; Memory readout control. ENABLE allows memory read operations	Section 1.4.2.81
80101C14h	PASSWORD_0	BSL access password - Word 0	Section 1.4.2.82

Table 1-2. NON_MAIN_CONFIGURATION Registers (continued)

Offset	Acronym	Register Name	Section
80101C18h	PASSWORD_1	BSL access password - Word 1	Section 1.4.2.83
80101C1Ch	PASSWORD_2	BSL access password - Word 2	Section 1.4.2.84
80101C20h	PASSWORD_3	BSL access password - Word 3	Section 1.4.2.85
80101C24h	PASSWORD_4	BSL access password - Word 4	Section 1.4.2.86
80101C28h	PASSWORD_5	BSL access password - Word 5	Section 1.4.2.87
80101C2Ch	PASSWORD_6	BSL access password - Word 6	Section 1.4.2.88
80101C30h	PASSWORD_7	BSL access password - Word 7	Section 1.4.2.89
80101C34h	APP_REV_POINTER	Pointer to application version information in MAIN flash	Section 1.4.2.90
80101C38h	BSLCONFIG1	Security alert response: Factory Reset/Disable BSL/ Ignore; UART communication speed selection for ROM BSL	Section 1.4.2.91
80101C3Ch	I2C_SLAVE_ADDR	I2C slave address for ROM BSL I2C interface; Reserved field; Reserved field	
80101C40h	RESERVED_REG_0	Reserved field; Reserved field; Reserved field; Reserved field	
80101C44h	RESERVED_REG_1	Reserved field; Reserved field; Reserved field; Reserved field	
80101C48h	RESERVED_REG_2	Reserved field; Reserved field; Reserved field; Reserved field	
80101C4Ch	CRC	CRC of BSL configuration structure	

Complex bit access types are encoded to fit into small table cells. Table 1-3 shows the codes that are used for access types in this section.

Table 1-3. NON_MAIN_Configuration 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

1.4.2.1 BCR_CONFIG_ID Register (Offset = 80101800h) [Reset = 00000000h]

BCR_CONFIG_ID is shown in Table 1-4.

Return to the Summary Table.

Predetermined Bootcode config signature ID

Table 1-4. BCR_CONFIG_ID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCR_CONFIG_ID	R/W	0h	Predetermined Bootcode config signature ID

1.4.2.2 BOOTCFG0 Register (Offset = 80101804h) [Reset = 00000000h]

BOOTCFG0 is shown in Table 1-5.

Return to the Summary Table.

Controls enable/disable of AHB-AP, ET-AP, PWR-AP. Values: DEBUG_EN, DEBUG_EN_PW, DEBUG_NS_EN, DEBUG_NS_EN_PW, DEBUG_DIS; Serial Wire Debug Port mode. Disabling this will disable all Debug ports including CFG_AP, SEC_AP. When disabled no DSSM commands are serviced

Table 1-5. BOOTCFG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	SWDP_MODE	R/W	0h	Serial Wire Debug Port mode. Disabling this will disable all Debug ports including CFG_AP, SEC_AP. When disabled no DSSM commands are serviced 5522h = DISABLED configuration AABBh = ENABLED configuration
15-0	DEBUG_ACCESS	R/W	0h	Controls enable/disable of AHB-AP, ET-AP, PWR-AP. Values: DEBUG_EN, DEBUG_EN_PW, DEBUG_NS_EN, DEBUG_NS_EN_PW, DEBUG_DIS 5522h = DISABLED configuration AABBh = ENABLED configuration

1.4.2.3 BOOTCFG1 Register (Offset = 80101808h) [Reset = 00000000h]

BOOTCFG1 is shown in Table 1-6.

Return to the Summary Table.

Controls BSL pin invocation capability. Values: BSL_PIN_INVOKE_EN, BSL_PIN_INVOKE_DIS; Controls debug access release until INITDONE is issued

Table 1-6. BOOTCFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	DEBUG_HOLD	R/W	0h	Controls debug access release until INITDONE is issued
15-0	BSL_PIN_INVOKE	R/W	0h	Controls BSL pin invocation capability. Values: BSL_PIN_INVOKE_EN, BSL_PIN_INVOKE_DIS 5522h = DISABLED configuration AABBh = ENABLED configuration

1.4.2.4 BOOTCFG2 Register (Offset = 8010180Ch) [Reset = 00000000h]

BOOTCFG2 is shown in Table 1-7.

Return to the Summary Table.

Controls CSC policy in SYSCTL. YES enables CSC policy checking; Controls flash bank swap policy in SYSCTL

Table 1-7. BOOTCFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	FLASH_BANK_SWAP_POLICY	R/W	0h	Controls flash bank swap policy in SYSCTL
15-0	CSC_EXISTS	R/W	0h	Controls CSC policy in SYSCTL. YES enables CSC policy checking 5522h = DISABLED configuration AABBh = ENABLED configuration

1.4.2.5 BOOTCFG3 Register (Offset = 80101810h) [Reset = 00000000h]

BOOTCFG3 is shown in Table 1-8.

Return to the Summary Table.

Fast boot mode configuration - Skips certain boot time checks when enabled; Bootloader mode enable/disable control. Must be enabled for BSL functionality

Table 1-8. BOOTCFG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	BOOTLOADER_MODE	R/W	0h	Bootloader mode enable/disable control. Must be enabled for BSL functionality
15-0	FAST_BOOT_MODE	R/W	0h	Fast boot mode configuration - Skips certain boot time checks when enabled

1.4.2.6 BOOTCFG4 Register (Offset = 80101814h) [Reset = 00000000h]

BOOTCFG4 is shown in Table 1-9.

Return to the Summary Table.

Mass erase mode configuration. Values: BC_CFG_MASS_ERASE_EN(0xAABB), BC_CFG_MASS_ERASE_EN_PW(0xCCDD), BC_CFG_MASS_ERASE_DIS(0x5522); Factory reset mode configuration. Values: BC_CFG_FACTORY_RESET_EN(0xAABB), BC_CFG_FACTORY_RESET_EN_PW(0xCCDD), BC_CFG_FACTORY_RESET_DIS(0x5522)

Table 1-9. BOOTCFG4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	FACTORY_RESET_MODE	R/W	0h	Factory reset mode configuration. Values: BC_CFG_FACTORY_RESET_EN(0xAABB), BC_CFG_FACTORY_RESET_EN_PW(0xCCDD), BC_CFG_FACTORY_RESET_DIS(0x5522) 5522h = DISABLE configuration AABBh = ENABLE configuration CCDDh = ENABLE_WITH_PASSWORD configuration
15-0	MASS_ERASE_MODE	R/W	0h	Mass erase mode configuration. Values: BC_CFG_MASS_ERASE_EN(0xAABB), BC_CFG_MASS_ERASE_EN_PW(0xCCDD), BC_CFG_MASS_ERASE_DIS(0x5522) 5522h = DISABLE configuration AABBh = ENABLE configuration CCDDh = ENABLE_WITH_PASSWORD configuration

1.4.2.7 MASS_ERASE_0 Register (Offset = 80101818h) [Reset = 00000000h]

MASS_ERASE_0 is shown in Table 1-10.

Return to the Summary Table.

Password hash for DSSM_BC_MASS_ERASE_REQUEST command - Word 0

Table 1-10. MASS_ERASE_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MASS_ERASE_0	R/W	0h	Password hash for DSSM_BC_MASS_ERASE_REQUEST command - Word 0

1.4.2.8 MASS_ERASE_1 Register (Offset = 8010181Ch) [Reset = 00000000h]

MASS_ERASE_1 is shown in Table 1-11.

Return to the Summary Table.

Password hash for DSSM_BC_MASS_ERASE_REQUEST command - Word 1

Table 1-11. MASS_ERASE_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MASS_ERASE_1	R/W	0h	Password hash for DSSM_BC_MASS_ERASE_REQUEST command - Word 1

1.4.2.9 MASS_ERASE_2 Register (Offset = 80101820h) [Reset = 00000000h]

MASS_ERASE_2 is shown in Table 1-12.

Return to the Summary Table.

Password hash for DSSM_BC_MASS_ERASE_REQUEST command - Word 2

Table 1-12. MASS_ERASE_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MASS_ERASE_2	R/W	0h	Password hash for DSSM_BC_MASS_ERASE_REQUEST command - Word 2

1.4.2.10 MASS_ERASE_3 Register (Offset = 80101824h) [Reset = 00000000h]

MASS_ERASE_3 is shown in Table 1-13.

Return to the Summary Table.

Password hash for DSSM_BC_MASS_ERASE_REQUEST command - Word 3

Table 1-13. MASS_ERASE_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MASS_ERASE_3	R/W	0h	Password hash for DSSM_BC_MASS_ERASE_REQUEST command - Word 3

1.4.2.11 MASS_ERASE_4 Register (Offset = 80101828h) [Reset = 00000000h]

MASS_ERASE_4 is shown in Table 1-14.

Return to the Summary Table.

Password hash for DSSM_BC_MASS_ERASE_REQUEST command - Word 4

Table 1-14. MASS_ERASE_4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MASS_ERASE_4	R/W	0h	Password hash for DSSM_BC_MASS_ERASE_REQUEST command - Word 4

1.4.2.12 MASS_ERASE_5 Register (Offset = 8010182Ch) [Reset = 00000000h]

MASS_ERASE_5 is shown in Table 1-15.

Return to the Summary Table.

Password hash for DSSM_BC_MASS_ERASE_REQUEST command - Word 5

Table 1-15. MASS_ERASE_5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MASS_ERASE_5	R/W	0h	Password hash for DSSM_BC_MASS_ERASE_REQUEST command - Word 5

1.4.2.13 MASS_ERASE_6 Register (Offset = 80101830h) [Reset = 00000000h]

MASS_ERASE_6 is shown in Table 1-16.

Return to the Summary Table.

Password hash for DSSM_BC_MASS_ERASE_REQUEST command - Word 6

Table 1-16. MASS_ERASE_6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MASS_ERASE_6	R/W	0h	Password hash for DSSM_BC_MASS_ERASE_REQUEST command - Word 6

1.4.2.14 MASS_ERASE_7 Register (Offset = 80101834h) [Reset = 00000000h]

MASS_ERASE_7 is shown in Table 1-17.

Return to the Summary Table.

Password hash for DSSM_BC_MASS_ERASE_REQUEST command - Word 7

Table 1-17. MASS_ERASE_7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MASS_ERASE_7	R/W	0h	Password hash for DSSM_BC_MASS_ERASE_REQUEST command - Word 7

1.4.2.15 FACTORY_RESET_0 Register (Offset = 80101838h) [Reset = 00000000h]

FACTORY_RESET_0 is shown in Table 1-18.

Return to the Summary Table.

Password hash for factory reset command - Word 0

Table 1-18. FACTORY_RESET_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FACTORY_RESET_0	R/W	0h	Password hash for factory reset command - Word 0

1.4.2.16 FACTORY_RESET_1 Register (Offset = 8010183Ch) [Reset = 00000000h]

FACTORY_RESET_1 is shown in Table 1-19.

Return to the Summary Table.

Password hash for factory reset command - Word 1

Table 1-19. FACTORY_RESET_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FACTORY_RESET_1	R/W	0h	Password hash for factory reset command - Word 1

1.4.2.17 FACTORY_RESET_2 Register (Offset = 80101840h) [Reset = 00000000h]

FACTORY_RESET_2 is shown in Table 1-20.

Return to the Summary Table.

Password hash for factory reset command - Word 2

Table 1-20. FACTORY_RESET_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FACTORY_RESET_2	R/W	0h	Password hash for factory reset command - Word 2

1.4.2.18 FACTORY_RESET_3 Register (Offset = 80101844h) [Reset = 00000000h]

FACTORY_RESET_3 is shown in Table 1-21.

Return to the Summary Table.

Password hash for factory reset command - Word 3

Table 1-21. FACTORY_RESET_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FACTORY_RESET_3	R/W	0h	Password hash for factory reset command - Word 3

1.4.2.19 FACTORY_RESET_4 Register (Offset = 80101848h) [Reset = 00000000h]

FACTORY_RESET_4 is shown in Table 1-22.

Return to the Summary Table.

Password hash for factory reset command - Word 4

Table 1-22. FACTORY_RESET_4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FACTORY_RESET_4	R/W	0h	Password hash for factory reset command - Word 4

1.4.2.20 FACTORY_RESET_5 Register (Offset = 8010184Ch) [Reset = 00000000h]

FACTORY_RESET_5 is shown in Table 1-23.

Return to the Summary Table.

Password hash for factory reset command - Word 5

Table 1-23. FACTORY_RESET_5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FACTORY_RESET_5	R/W	0h	Password hash for factory reset command - Word 5

1.4.2.21 FACTORY_RESET_6 Register (Offset = 80101850h) [Reset = 00000000h]

FACTORY_RESET_6 is shown in Table 1-24.

Return to the Summary Table.

Password hash for factory reset command - Word 6

Table 1-24. FACTORY_RESET_6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FACTORY_RESET_6	R/W	0h	Password hash for factory reset command - Word 6

1.4.2.22 FACTORY_RESET_7 Register (Offset = 80101854h) [Reset = 00000000h]

FACTORY_RESET_7 is shown in Table 1-25.

Return to the Summary Table.

Password hash for factory reset command - Word 7

Table 1-25. FACTORY_RESET_7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FACTORY_RESET_7	R/W	0h	Password hash for factory reset command - Word 7

1.4.2.23 DEBUG_LOCK_0 Register (Offset = 80101858h) [Reset = 00000000h]

DEBUG_LOCK_0 is shown in Table 1-26.

Return to the Summary Table.

Password hash for debug access authentication - Word 0

Table 1-26. DEBUG_LOCK_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DEBUG_LOCK_0	R/W	0h	Password hash for debug access authentication - Word 0

1.4.2.24 DEBUG_LOCK_1 Register (Offset = 8010185Ch) [Reset = 00000000h]

DEBUG_LOCK_1 is shown in Table 1-27.

Return to the Summary Table.

Password hash for debug access authentication - Word 1

Table 1-27. DEBUG_LOCK_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DEBUG_LOCK_1	R/W	0h	Password hash for debug access authentication - Word 1

1.4.2.25 DEBUG_LOCK_2 Register (Offset = 80101860h) [Reset = 00000000h]

DEBUG_LOCK_2 is shown in Table 1-28.

Return to the Summary Table.

Password hash for debug access authentication - Word 2

Table 1-28. DEBUG_LOCK_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DEBUG_LOCK_2	R/W	0h	Password hash for debug access authentication - Word 2

1.4.2.26 DEBUG_LOCK_3 Register (Offset = 80101864h) [Reset = 00000000h]

DEBUG_LOCK_3 is shown in Table 1-29.

Return to the Summary Table.

Password hash for debug access authentication - Word 3

Table 1-29. DEBUG_LOCK_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DEBUG_LOCK_3	R/W	0h	Password hash for debug access authentication - Word 3

1.4.2.27 DEBUG_LOCK_4 Register (Offset = 80101868h) [Reset = 00000000h]

DEBUG_LOCK_4 is shown in Table 1-30.

Return to the Summary Table.

Password hash for debug access authentication - Word 4

Table 1-30. DEBUG_LOCK_4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DEBUG_LOCK_4	R/W	0h	Password hash for debug access authentication - Word 4

1.4.2.28 DEBUG_LOCK_5 Register (Offset = 8010186Ch) [Reset = 00000000h]

DEBUG_LOCK_5 is shown in Table 1-31.

Return to the Summary Table.

Password hash for debug access authentication - Word 5

Table 1-31. DEBUG_LOCK_5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DEBUG_LOCK_5	R/W	0h	Password hash for debug access authentication - Word 5

1.4.2.29 DEBUG_LOCK_6 Register (Offset = 80101870h) [Reset = 00000000h]

DEBUG_LOCK_6 is shown in Table 1-32.

Return to the Summary Table.

Password hash for debug access authentication - Word 6

Table 1-32. DEBUG_LOCK_6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DEBUG_LOCK_6	R/W	0h	Password hash for debug access authentication - Word 6

1.4.2.30 DEBUG_LOCK_7 Register (Offset = 80101874h) [Reset = 00000000h]

DEBUG_LOCK_7 is shown in Table 1-33.

Return to the Summary Table.

Password hash for debug access authentication - Word 7

Table 1-33. DEBUG_LOCK_7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DEBUG_LOCK_7	R/W	0h	Password hash for debug access authentication - Word 7

1.4.2.31 DEBUG_NS_LOCK_0 Register (Offset = 80101878h) [Reset = 00000000h]

DEBUG_NS_LOCK_0 is shown in Table 1-34.

Return to the Summary Table.

Password hash for non-secure debug access - Word 0

Table 1-34. DEBUG_NS_LOCK_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DEBUG_NS_LOCK_0	R/W	0h	Password hash for non-secure debug access - Word 0

1.4.2.32 DEBUG_NS_LOCK_1 Register (Offset = 8010187Ch) [Reset = 00000000h]

DEBUG_NS_LOCK_1 is shown in Table 1-35.

Return to the Summary Table.

Password hash for non-secure debug access - Word 1

Table 1-35. DEBUG_NS_LOCK_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DEBUG_NS_LOCK_1	R/W	0h	Password hash for non-secure debug access - Word 1

1.4.2.33 DEBUG_NS_LOCK_2 Register (Offset = 80101880h) [Reset = 00000000h]

DEBUG_NS_LOCK_2 is shown in Table 1-36.

Return to the Summary Table.

Password hash for non-secure debug access - Word 2

Table 1-36. DEBUG_NS_LOCK_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DEBUG_NS_LOCK_2	R/W	0h	Password hash for non-secure debug access - Word 2

1.4.2.34 DEBUG_NS_LOCK_3 Register (Offset = 80101884h) [Reset = 00000000h]

DEBUG_NS_LOCK_3 is shown in Table 1-37.

Return to the Summary Table.

Password hash for non-secure debug access - Word 3

Table 1-37. DEBUG_NS_LOCK_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DEBUG_NS_LOCK_3	R/W	0h	Password hash for non-secure debug access - Word 3

1.4.2.35 DEBUG_NS_LOCK_4 Register (Offset = 80101888h) [Reset = 00000000h]

DEBUG_NS_LOCK_4 is shown in Table 1-38.

Return to the Summary Table.

Password hash for non-secure debug access - Word 4

Table 1-38. DEBUG_NS_LOCK_4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DEBUG_NS_LOCK_4	R/W	0h	Password hash for non-secure debug access - Word 4

1.4.2.36 DEBUG_NS_LOCK_5 Register (Offset = 8010188Ch) [Reset = 00000000h]

DEBUG_NS_LOCK_5 is shown in Table 1-39.

Return to the Summary Table.

Password hash for non-secure debug access - Word 5

Table 1-39. DEBUG_NS_LOCK_5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DEBUG_NS_LOCK_5	R/W	0h	Password hash for non-secure debug access - Word 5

1.4.2.37 DEBUG_NS_LOCK_6 Register (Offset = 80101890h) [Reset = 00000000h]

DEBUG_NS_LOCK_6 is shown in Table 1-40.

Return to the Summary Table.

Password hash for non-secure debug access - Word 6

Table 1-40. DEBUG_NS_LOCK_6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DEBUG_NS_LOCK_6	R/W	0h	Password hash for non-secure debug access - Word 6

1.4.2.38 DEBUG_NS_LOCK_7 Register (Offset = 80101894h) [Reset = 00000000h]

DEBUG_NS_LOCK_7 is shown in Table 1-41.

Return to the Summary Table.

Password hash for non-secure debug access - Word 7

Table 1-41. DEBUG_NS_LOCK_7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DEBUG_NS_LOCK_7	R/W	0h	Password hash for non-secure debug access - Word 7

1.4.2.39 BAD_CONV_0 Register (Offset = 80101898h) [Reset = 00000000h]

BAD_CONV_0 is shown in Table 1-42.

Return to the Summary Table.

This field stores the SHA-256 hash (32 bytes) of the 16-byte password used for bad device conversion. The hash is stored as an array of 8 32-bit words.

Table 1-42. BAD_CONV_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BAD_CONV_0	R/W	0h	Bad conversion password hash - Word 0

1.4.2.40 BAD_CONV_1 Register (Offset = 8010189Ch) [Reset = 00000000h]

BAD_CONV_1 is shown in Table 1-43.

Return to the Summary Table.

Bad conversion password hash - Word 1

Table 1-43. BAD_CONV_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BAD_CONV_1	R/W	0h	Bad conversion password hash - Word 1

1.4.2.41 BAD_CONV_2 Register (Offset = 801018A0h) [Reset = 00000000h]

BAD_CONV_2 is shown in Table 1-44.

Return to the Summary Table.

Bad conversion password hash - Word 2

Table 1-44. BAD_CONV_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BAD_CONV_2	R/W	0h	Bad conversion password hash - Word 2

1.4.2.42 BAD_CONV_3 Register (Offset = 801018A4h) [Reset = 00000000h]

BAD_CONV_3 is shown in Table 1-45.

Return to the Summary Table.

Bad conversion password hash - Word 3

Table 1-45. BAD_CONV_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BAD_CONV_3	R/W	0h	Bad conversion password hash - Word 3

1.4.2.43 BAD_CONV_4 Register (Offset = 801018A8h) [Reset = 00000000h]

BAD_CONV_4 is shown in Table 1-46.

Return to the Summary Table.

Bad conversion password hash - Word 4

Table 1-46. BAD_CONV_4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BAD_CONV_4	R/W	0h	Bad conversion password hash - Word 4

1.4.2.44 BAD_CONV_5 Register (Offset = 801018ACh) [Reset = 00000000h]

BAD_CONV_5 is shown in Table 1-47.

Return to the Summary Table.

Bad conversion password hash - Word 5

Table 1-47. BAD_CONV_5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BAD_CONV_5	R/W	0h	Bad conversion password hash - Word 5

1.4.2.45 BAD_CONV_6 Register (Offset = 801018B0h) [Reset = 00000000h]

BAD_CONV_6 is shown in Table 1-48.

Return to the Summary Table.

Bad conversion password hash - Word 6

Table 1-48. BAD_CONV_6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BAD_CONV_6	R/W	0h	Bad conversion password hash - Word 6

1.4.2.46 BAD_CONV_7 Register (Offset = 801018B4h) [Reset = 00000000h]

BAD_CONV_7 is shown in Table 1-49.

Return to the Summary Table.

Bad conversion password hash - Word 7

Table 1-49. BAD_CONV_7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BAD_CONV_7	R/W	0h	Bad conversion password hash - Word 7

1.4.2.47 SECURE_BOOT_MODE Register (Offset = 801018B8h) [Reset = 00000000h]

SECURE_BOOT_MODE is shown in Table 1-50.

Return to the Summary Table.

Controls application authentication; Reserved field

Table 1-50. SECURE_BOOT_MODE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	Reserved field
15-0	SECURE_BOOT_MODE	R/W	0h	Controls application authentication AABBh = BC_CFG_SECURE_BOOT_CRC_EN configuration CCDDh = BC_CFG_SECURE_BOOT_HASH_EN configuration FFFFh = DISABLED configuration

1.4.2.48 USER_SECURE_APP_START_ADDR Register (Offset = 801018BCh) [Reset = 00000000h]

USER_SECURE_APP_START_ADDR is shown in Table 1-51.

Return to the Summary Table.

Starting address for secure application

Table 1-51. USER_SECURE_APP_START_ADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	USER_SECURE_APP_START_ADDR	R/W	0h	Starting address for secure application

1.4.2.49 USER_SECURE_APP_LENGTH Register (Offset = 801018C0h) [Reset = 00000000h]

USER_SECURE_APP_LENGTH is shown in Table 1-52.

Return to the Summary Table.

Length of secure application

Table 1-52. USER_SECURE_APP_LENGTH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	USER_SECURE_APP_LENGTH	R/W	0h	Length of secure application

1.4.2.50 USER_SECURE_APP_HASH_0 Register (Offset = 801018C4h) [Reset = 00000000h]

USER_SECURE_APP_HASH_0 is shown in Table 1-53.

Return to the Summary Table.

Secure application hash - Word 0

Table 1-53. USER_SECURE_APP_HASH_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	USER_SECURE_APP_HASH_0	R/W	0h	Secure application hash - Word 0

1.4.2.51 USER_SECURE_APP_HASH_1 Register (Offset = 801018C8h) [Reset = 00000000h]

USER_SECURE_APP_HASH_1 is shown in Table 1-54.

Return to the Summary Table.

Secure application hash - Word 1

Table 1-54. USER_SECURE_APP_HASH_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	USER_SECURE_APP_HASH_1	R/W	0h	Secure application hash - Word 1

1.4.2.52 USER_SECURE_APP_HASH_2 Register (Offset = 801018CCh) [Reset = 00000000h]

USER_SECURE_APP_HASH_2 is shown in Table 1-55.

Return to the Summary Table.

Secure application hash - Word 2

Table 1-55. USER_SECURE_APP_HASH_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	USER_SECURE_APP_HASH_2	R/W	0h	Secure application hash - Word 2

1.4.2.53 USER_SECURE_APP_HASH_3 Register (Offset = 801018D0h) [Reset = 00000000h]

USER_SECURE_APP_HASH_3 is shown in Table 1-56.

Return to the Summary Table.

Secure application hash - Word 3

Table 1-56. USER_SECURE_APP_HASH_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	USER_SECURE_APP_HASH_3	R/W	0h	Secure application hash - Word 3

1.4.2.54 USER_SECURE_APP_HASH_4 Register (Offset = 801018D4h) [Reset = 00000000h]

USER_SECURE_APP_HASH_4 is shown in Table 1-57.

Return to the Summary Table.

Secure application hash - Word 4

Table 1-57. USER_SECURE_APP_HASH_4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	USER_SECURE_APP_HASH_4	R/W	0h	Secure application hash - Word 4

1.4.2.55 USER_SECURE_APP_HASH_5 Register (Offset = 801018D8h) [Reset = 00000000h]

USER_SECURE_APP_HASH_5 is shown in Table 1-58.

Return to the Summary Table.

Secure application hash - Word 5

Table 1-58. USER_SECURE_APP_HASH_5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	USER_SECURE_APP_HASH_5	R/W	0h	Secure application hash - Word 5

1.4.2.56 USER_SECURE_APP_HASH_6 Register (Offset = 801018DCh) [Reset = 00000000h]

USER_SECURE_APP_HASH_6 is shown in Table 1-59.

Return to the Summary Table.

Secure application hash - Word 6

Table 1-59. USER_SECURE_APP_HASH_6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	USER_SECURE_APP_HASH_6	R/W	0h	Secure application hash - Word 6

1.4.2.57 USER_SECURE_APP_HASH_7 Register (Offset = 801018E0h) [Reset = 00000000h]

USER_SECURE_APP_HASH_7 is shown in Table 1-60.

Return to the Summary Table.

Secure application hash - Word 7

Table 1-60. USER_SECURE_APP_HASH_7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	USER_SECURE_APP_HASH_7	R/W	0h	Secure application hash - Word 7

1.4.2.58 BANK0_NM_USER_CONFIG Register (Offset = 801018E4h) [Reset = 00000000h]

BANK0_NM_USER_CONFIG is shown in Table 1-61.

Return to the Summary Table.

User Config Non Main Flash Static Write Protection. WEP_DISABLE=0xAABB; Reserved field

Table 1-61. BANK0_NM_USER_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	Reserved field
15-0	BANK0_NM_USER_CONFIG	R/W	0h	User Config Non Main Flash Static Write Protection. WEP_DISABLE=0xAABB

1.4.2.59 BANK0_WRITE_ERASE_PROTECTION_A Register (Offset = 801018E8h) [Reset = 00000000h]

BANK0_WRITE_ERASE_PROTECTION_A is shown in Table 1-62.

Return to the Summary Table.

Write protection for first 32 sectors

Table 1-62. BANK0_WRITE_ERASE_PROTECTION_A Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BANK0_WRITE_ERASE_PROTECTION_A	R/W	0h	Write protection for first 32 sectors

1.4.2.60 BANK0_WRITE_ERASE_PROTECTION_B Register (Offset = 801018ECh) [Reset = 00000000h]

BANK0_WRITE_ERASE_PROTECTION_B is shown in Table 1-63.

Return to the Summary Table.

Write Protection for 512KB-64KB

Table 1-63. BANK0_WRITE_ERASE_PROTECTION_B Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BANK0_WRITE_ERASE_PROTECTION_B	R/W	0h	Write Protection for 512KB-64KB

1.4.2.61 BANK0_SECURITY_PROTECTION_A Register (Offset = 801018F0h) [Reset = 00000000h]

BANK0_SECURITY_PROTECTION_A is shown in Table 1-64.

Return to the Summary Table.

Security protection for first 32 sectors

Table 1-64. BANK0_SECURITY_PROTECTION_A Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BANK0_SECURITY_PROTECTION_A	R/W	0h	Security protection for first 32 sectors

1.4.2.62 BANK0_SECURITY_PROTECTION_B Register (Offset = 801018F4h) [Reset = 00000000h]

BANK0_SECURITY_PROTECTION_B is shown in Table 1-65.

Return to the Summary Table.

Security protection for 512KB-64KB

Table 1-65. BANK0_SECURITY_PROTECTION_B Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BANK0_SECURITY_PROTECTION_B	R/W	0h	Security protection for 512KB-64KB

1.4.2.63 BANK0_PRIVILEGE_PROTECTION_A Register (Offset = 801018F8h) [Reset = 00000000h]

BANK0_PRIVILEGE_PROTECTION_A is shown in Table 1-66.

Return to the Summary Table.

Privilege protection for first 32 sectors

Table 1-66. BANK0_PRIVILEGE_PROTECTION_A Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BANK0_PRIVILEGE_PROTECTION_A	R/W	0h	Privilege protection for first 32 sectors

1.4.2.64 BANK0_PRIVILEGE_PROTECTION_B Register (Offset = 801018FCh) [Reset = 00000000h]

BANK0_PRIVILEGE_PROTECTION_B is shown in Table 1-67.

Return to the Summary Table.

Privilege protection for 512KB-64KB

Table 1-67. BANK0_PRIVILEGE_PROTECTION_B Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BANK0_PRIVILEGE_PROTECTION_B	R/W	0h	Privilege protection for 512KB-64KB

1.4.2.65 BANK1_WRITE_ERASE_PROTECTION_A Register (Offset = 80101904h) [Reset = 00000000h]

BANK1_WRITE_ERASE_PROTECTION_A is shown in Table 1-68.

Return to the Summary Table.

Write protection for first 32 sectors

Table 1-68. BANK1_WRITE_ERASE_PROTECTION_A Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BANK1_WRITE_ERASE_PROTECTION_A	R/W	0h	Write protection for first 32 sectors

1.4.2.66 BANK1_WRITE_ERASE_PROTECTION_B Register (Offset = 80101908h) [Reset = 00000000h]

BANK1_WRITE_ERASE_PROTECTION_B is shown in Table 1-69.

Return to the Summary Table.

Write Protection for 512KB-64KB

Table 1-69. BANK1_WRITE_ERASE_PROTECTION_B Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BANK1_WRITE_ERASE_PROTECTION_B	R/W	0h	Write Protection for 512KB-64KB

1.4.2.67 BANK1_SECURITY_PROTECTION_A Register (Offset = 8010190Ch) [Reset = 00000000h]

BANK1_SECURITY_PROTECTION_A is shown in Table 1-70.

Return to the Summary Table.

Security protection for first 32 sectors

Table 1-70. BANK1_SECURITY_PROTECTION_A Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BANK1_SECURITY_PROTECTION_A	R/W	0h	Security protection for first 32 sectors

1.4.2.68 BANK1_SECURITY_PROTECTION_B Register (Offset = 80101910h) [Reset = 00000000h]

BANK1_SECURITY_PROTECTION_B is shown in Table 1-71.

Return to the Summary Table.

Security protection for 512KB-64KB

Table 1-71. BANK1_SECURITY_PROTECTION_B Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BANK1_SECURITY_PROTECTION_B	R/W	0h	Security protection for 512KB-64KB

1.4.2.69 BANK1_PRIVILEGE_PROTECTION_A Register (Offset = 80101914h) [Reset = 00000000h]

BANK1_PRIVILEGE_PROTECTION_A is shown in Table 1-72.

Return to the Summary Table.

Privilege protection for first 32 sectors

Table 1-72. BANK1_PRIVILEGE_PROTECTION_A Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BANK1_PRIVILEGE_PROTECTION_A	R/W	0h	Privilege protection for first 32 sectors

1.4.2.70 BANK1_PRIVILEGE_PROTECTION_B Register (Offset = 80101918h) [Reset = 00000000h]

BANK1_PRIVILEGE_PROTECTION_B is shown in Table 1-73.

Return to the Summary Table.

Privilege protection for 512KB-64KB

Table 1-73. BANK1_PRIVILEGE_PROTECTION_B Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BANK1_PRIVILEGE_PROTECTION_B	R/W	0h	Privilege protection for 512KB-64KB

1.4.2.71 DBANK_WRITE_ERASE_PROTECTION Register (Offset = 80101920h) [Reset = 00000000h]

DBANK_WRITE_ERASE_PROTECTION is shown in Table 1-74.

Return to the Summary Table.

Write/Erase protection for Data Bank

Table 1-74. DBANK_WRITE_ERASE_PROTECTION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DBANK_WRITE_ERASE_PROTECTION	R/W	0h	Write/Erase protection for Data Bank

1.4.2.72 DBANK_SECURITY_PROTECTION Register (Offset = 80101924h) [Reset = 00000000h]

DBANK_SECURITY_PROTECTION is shown in Table 1-75.

Return to the Summary Table.

Security protection for Data Bank

Table 1-75. DBANK_SECURITY_PROTECTION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DBANK_SECURITY_PROTECTION	R/W	0h	Security protection for Data Bank

1.4.2.73 DBANK_PRIVILEGE_PROTECTION Register (Offset = 80101928h) [Reset = 00000000h]

DBANK_PRIVILEGE_PROTECTION is shown in Table 1-76.

Return to the Summary Table.

Privilege protection for Data Bank

Table 1-76. DBANK_PRIVILEGE_PROTECTION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DBANK_PRIVILEGE_PROTECTION	R/W	0h	Privilege protection for Data Bank

1.4.2.74 BOOTCLK0 Register (Offset = 80101930h) [Reset = 00000000h]

BOOTCLK0 is shown in Table 1-77.

Return to the Summary Table.

Clock configuration; PLL multiplier value; PLL divider value; PLL clock source configuration

Table 1-77. BOOTCLK0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SYSPLL_SOURCE	R/W	0h	PLL clock source configuration AAh = BC_CFG_PLL_SOURCE_HFXT_20MHZ configuration BBh = BC_CFG_PLL_SOURCE_HFXT_40MHZ configuration FFh = DEFAULT configuration
23-16	SYSPLL_CONFIG_RDIV	R/W	0h	PLL divider value
15-8	SYSPLL_CONFIG_QDIV	R/W	0h	PLL multiplier value
7-0	SYSPLL_CONFIG	R/W	0h	Clock configuration AAh = PLL_80 configuration BBh = PLL_CUSTOM configuration FFh = NOPLL configuration

1.4.2.75 BOOTCLK1 Register (Offset = 80101934h) [Reset = 00000000h]

BOOTCLK1 is shown in Table 1-78.

Return to the Summary Table.

CPU delay cycles for PLL settling time; CPU delay cycles for HFXT monitoring; Reserved configuration field 0

Table 1-78. BOOTCLK1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESVD_0	R/W	0h	Reserved configuration field 0
15-8	HFXT_STARTUP_MONITOR_TIME	R/W	0h	CPU delay cycles for HFXT monitoring
7-0	SYSPLL_SETTLING_TIME	R/W	0h	CPU delay cycles for PLL settling time

1.4.2.76 CRC Register (Offset = 8010194Ch) [Reset = 00000000h]

CRC is shown in Table 1-79.

Return to the Summary Table.

CRC-32 of BCR config structure

Table 1-79. CRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CRC	R/W	0h	CRC-32 of BCR config structure

1.4.2.77 BSL_CONFIG_ID Register (Offset = 80101C00h) [Reset = 00000000h]

BSL_CONFIG_ID is shown in Table 1-80.

Return to the Summary Table.

Predetermined Bootloader config signature ID

Table 1-80. BSL_CONFIG_ID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BSL_CONFIG_ID	R/W	0h	Predetermined Bootloader config signature ID

1.4.2.78 BSLPINCFG0 Register (Offset = 80101C04h) [Reset = 00000000h]

BSLPINCFG0 is shown in Table 1-81.

Return to the Summary Table.

UART receive pin number configuration; UART receive pin multiplexer selection; UART transmit pin number configuration; UART transmit pin multiplexer selection

Table 1-81. BSLPINCFG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	UART_TXD_PF_MUX_SEL	R/W	0h	UART transmit pin multiplexer selection
23-16	UART_TXD_PAD_NUM	R/W	0h	UART transmit pin number configuration
15-8	UART_RXD_PF_MUX_SEL	R/W	0h	UART receive pin multiplexer selection
7-0	UART_RXD_PAD_NUM	R/W	0h	UART receive pin number configuration

1.4.2.79 BSLPINCFG1 Register (Offset = 80101C08h) [Reset = 00000000h]

BSLPINCFG1 is shown in Table 1-82.

Return to the Summary Table.

I2C data pin number configuration; I2C data pin multiplexer selection; I2C clock pin number configuration; I2C clock pin multiplexer selection

Table 1-82. BSLPINCFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	I2C_SCL_PF_MUX_SEL	R/W	0h	I2C clock pin multiplexer selection
23-16	I2C_SCL_PAD_NUM	R/W	0h	I2C clock pin number configuration
15-8	I2C_SDA_PF_MUX_SEL	R/W	0h	I2C data pin multiplexer selection
7-0	I2C_SDA_PAD_NUM	R/W	0h	I2C data pin number configuration

1.4.2.80 BSLPINCFG2 Register (Offset = 80101C0Ch) [Reset = 00000000h]

BSLPINCFG2 is shown in Table 1-83.

Return to the Summary Table.

MCAN receive pin number configuration; MCAN receive pin multiplexer selection; MCAN transmit pin number configuration; MCAN transmit pin multiplexer selection

Table 1-83. BSLPINCFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	MCAN_TX_PF_MUX_SEL	R/W	0h	MCAN transmit pin multiplexer selection
23-16	MCAN_TX_PAD_NUM	R/W	0h	MCAN transmit pin number configuration
15-8	MCAN_RX_PF_MUX_SEL	R/W	0h	MCAN receive pin multiplexer selection
7-0	MCAN_RX_PAD_NUM	R/W	0h	MCAN receive pin number configuration

1.4.2.81 BSLCONFIG0 Register (Offset = 80101C10h) [Reset = 00000000h]

BSLCONFIG0 is shown in Table 1-84.

Return to the Summary Table.

BSL invoke pin configuration data 0; BSL invoke pin configuration data 1; Memory readout control. ENABLE allows memory read operations

Table 1-84. BSLCONFIG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	READOUT	R/W	0h	Memory readout control. ENABLE allows memory read operations
15-8	PIN_DATA_1	R/W	0h	BSL invoke pin configuration data 1
7-0	PIN_DATA_0	R/W	0h	BSL invoke pin configuration data 0

1.4.2.82 PASSWORD_0 Register (Offset = 80101C14h) [Reset = 00000000h]

PASSWORD_0 is shown in Table 1-85.

Return to the Summary Table.

BSL access password - Word 0

Table 1-85. PASSWORD_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PASSWORD_0	R/W	0h	BSL access password - Word 0

1.4.2.83 PASSWORD_1 Register (Offset = 80101C18h) [Reset = 00000000h]

PASSWORD_1 is shown in Table 1-86.

Return to the Summary Table.

BSL access password - Word 1

Table 1-86. PASSWORD_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PASSWORD_1	R/W	0h	BSL access password - Word 1

1.4.2.84 PASSWORD_2 Register (Offset = 80101C1Ch) [Reset = 00000000h]

PASSWORD_2 is shown in Table 1-87.

Return to the Summary Table.

BSL access password - Word 2

Table 1-87. PASSWORD_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PASSWORD_2	R/W	0h	BSL access password - Word 2

1.4.2.85 PASSWORD_3 Register (Offset = 80101C20h) [Reset = 00000000h]

PASSWORD_3 is shown in Table 1-88.

Return to the Summary Table.

BSL access password - Word 3

Table 1-88. PASSWORD_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PASSWORD_3	R/W	0h	BSL access password - Word 3

1.4.2.86 PASSWORD_4 Register (Offset = 80101C24h) [Reset = 00000000h]

PASSWORD_4 is shown in Table 1-89.

Return to the Summary Table.

BSL access password - Word 4

Table 1-89. PASSWORD_4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PASSWORD_4	R/W	0h	BSL access password - Word 4

1.4.2.87 PASSWORD_5 Register (Offset = 80101C28h) [Reset = 00000000h]

PASSWORD_5 is shown in Table 1-90.

Return to the Summary Table.

BSL access password - Word 5

Table 1-90. PASSWORD_5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PASSWORD_5	R/W	0h	BSL access password - Word 5

1.4.2.88 PASSWORD_6 Register (Offset = 80101C2Ch) [Reset = 00000000h]

PASSWORD_6 is shown in Table 1-91.

Return to the Summary Table.

BSL access password - Word 6

Table 1-91. PASSWORD_6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PASSWORD_6	R/W	0h	BSL access password - Word 6

1.4.2.89 PASSWORD_7 Register (Offset = 80101C30h) [Reset = 00000000h]

PASSWORD_7 is shown in Table 1-92.

Return to the Summary Table.

BSL access password - Word 7

Table 1-92. PASSWORD_7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PASSWORD_7	R/W	0h	BSL access password - Word 7

1.4.2.90 APP_REV_POINTER Register (Offset = 80101C34h) [Reset = 00000000h]

APP_REV_POINTER is shown in Table 1-93.

Return to the Summary Table.

Pointer to application version information in MAIN flash

Table 1-93. APP_REV_POINTER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_REV_POINTER	R/W	0h	Pointer to application version information in MAIN flash

1.4.2.91 BSLCONFIG1 Register (Offset = 80101C38h) [Reset = 00000000h]

BSLCONFIG1 is shown in Table 1-94.

Return to the Summary Table.

Security alert response: Factory Reset/Disable BSL/Ignore; UART communication speed selection for ROM BSL

Table 1-94. BSLCONFIG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	UART_BAUD_RATE	R/W	0h	UART communication speed selection for ROM BSL 1h = BAUDRATE_4800 configuration 2h = BAUDRATE_9600 configuration 3h = BAUDRATE_19200 configuration 4h = BAUDRATE_38400 configuration 5h = BAUDRATE_57600 configuration 6h = BAUDRATE_115200 configuration 7h = BAUDRATE_1000000 configuration 8h = BAUDRATE_2000000 configuration 9h = BAUDRATE_3000000 configuration
15-0	SECURITY_ALERT_LEVEL	R/W	0h	Security alert response: Factory Reset/Disable BSL/Ignore AABBh = FACTORY_RESET configuration CCDDh = DISABLE_BSL configuration FFFFh = DO_NOTHING configuration

1.4.2.92 I2C_SLAVE_ADDR Register (Offset = 80101C3Ch) [Reset = 00000000h]

I2C_SLAVE_ADDR is shown in Table 1-95.

Return to the Summary Table.

I2C slave address for ROM BSL I2C interface; Reserved field; Reserved field

Table 1-95. I2C_SLAVE_ADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	0h	Reserved field
23-16	RESERVED	R/W	0h	Reserved field
15-0	I2C_SLAVE_ADDR	R/W	0h	I2C slave address for ROM BSL I2C interface

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 Registers

Table 1-96 lists the memory-mapped registers for the FACTORYREGION registers. All register offset addresses not listed in Table 1-96 should be considered as reserved locations and the register contents should not be modified.

Table 1-96. FACTORYREGION Registers

Offset	Acronym	Register Name	Section
80111000h	TRACEID	Defined by TI, during ATE, based on wafer	Section 1.5.1.1
80111004h	DEVICEID	This is the JTAGIDCODE that comes from the Ramp system	Section 1.5.1.2
80111008h	USERID	Defined by TI, depending on device spin	Section 1.5.1.3
8011100Ch	BSLPIN_UART	BSL UART Pin Configuration	Section 1.5.1.4
80111010h	BSLPIN_I2C	BSL I2C Pin Configuration	Section 1.5.1.5
80111014h	BSLPIN_CAN	BSL CAN Pin Configuration	Section 1.5.1.6
80111018h	BSLPIN_INVOKE	BSL Pin Invocation Configuration	Section 1.5.1.7
8011101Ch	SRAMFLASH		Section 1.5.1.8
80111020h	PLLSTARTUP0_4_8MHZ		Section 1.5.1.9
80111024h	PLLSTARTUP1_4_8MHZ	System PLL Paramater 1 MMR --- Data from Flash Table Lookup	Section 1.5.1.10
80111028h	PLLSTARTUP0_8_16MHZ		Section 1.5.1.11
8011102Ch	PLLSTARTUP1_8_16MHZ	System PLL Paramater 1 MMR --- Data from Flash Table Lookup	Section 1.5.1.12
80111030h	PLLSTARTUP0_16_32MHZ		Section 1.5.1.13
80111034h	PLLSTARTUP1_16_32MHZ	System PLL Paramater 1 MMR --- Data from Flash Table Lookup	Section 1.5.1.14
80111038h	PLLSTARTUP0_32_48MHZ		Section 1.5.1.15
8011103Ch	PLLSTARTUP1_32_48MHZ	System PLL Paramater 1 MMR --- Data from Flash Table Lookup	Section 1.5.1.16
80111040h	TEMP_SENSE0	Temperature sensor room temperature calibration code. This is ADC conversion results of temperature sensor output voltage. Included in BOOTCRC calculation.	Section 1.5.1.17
80111044h	RESERVED00		
80111048h	RESERVED01		
8011104Ch	RESERVED02		
80111050h	RESERVED03		
80111054h	RESERVED04		
80111058h	RESERVED05		
8011105Ch	RESERVED06		
80111060h	RESERVED07		
80111064h	RESERVED08		
80111068h	RESERVED09		
8011106Ch	RESERVED10		
80111070h	RESERVED11		
80111074h	RESERVED12		
80111078h	RESERVED13		
8011107Ch	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 1-97 shows the codes that are used for access types in this section.

Table 1-97. FACTORYREGION Access Type Codes

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

1.5.1.1 TRACEID Register (Offset = 80111000h) [Reset = 00000000h]

TRACEID is shown in Table 1-98.

Return to the Summary Table.

unique per part shipped, done per established TI process

Table 1-98. TRACEID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R	0h	

1.5.1.2 DEVICEID Register (Offset = 80111004h) [Reset = 1BBB702Fh]

DEVICEID is shown in Table 1-99.

Return to the Summary Table.

per Connectivity format, provisioned from RAMP and is die rev specific

Table 1-99. DEVICEID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	VERSION	R	1h	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	BBB7h	Part number of the device.
11-1	MANUFACTURER	R	17h	TI's JEDEC bank and company code, which is: 00000010111b
0	ALWAYS_1	R	1h	This is always 1

1.5.1.3 USERID Register (Offset = 80111008h) [Reset = 00000000h]

USERID is shown in Table 1-100.

Return to the Summary Table.

per Connectivity format, defines the variant feature set

Table 1-100. USERID Register Field Descriptions

Bit	Field	Type	Reset	Description
31	START	R	1h	
30-28	MAJORREV	R	0h	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
27-24	MINORREV	R	0h	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.
23-16	VARIANT	R	0h	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.
15-0	PART	R	0h	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.

1.5.1.4 BSLPIN_UART Register (Offset = 8011100Ch) [Reset = 00000000h]

BSLPIN_UART is shown in Table 1-101.

Return to the Summary Table.

BSL UART Pin Configuration

Table 1-101. BSLPIN_UART Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	UART_TXD_PF	R	07h	UART TXD Pin Function Selection Value
23-16	UART_TXD_PAD	R	15h	UART TXD Pin used by BSL
15-8	UART_RXD_PF	R	07h	UART RXD Pin Function Selection Value
7-0	UART_RXD_PAD	R	16h	UART RXD Pad used by BSL

1.5.1.5 BSLPIN_I2C Register (Offset = 80111010h) [Reset = 00000000h]

BSLPIN_I2C is shown in Table 1-102.

Return to the Summary Table.

BSL I2C Pin Configuration

Table 1-102. BSLPIN_I2C Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	I2C_SCL_PF	R	6h	I2C SCL Pin Function Selection Value
23-16	I2C_SCL_PAD	R	2h	I2C SCL Pin used by BSL
15-8	I2C_SDA_PF	R	6h	I2C SDA Pin Function Selection Value
7-0	I2C_SDA_PAD	R	1h	I2C SDA Pin used by BSL

1.5.1.6 BSLPIN_CAN Register (Offset = 80111014h) [Reset = 00000000h]

BSLPIN_CAN is shown in Table 1-103.

Return to the Summary Table.

BSL CAN Pin Configuration

Table 1-103. BSLPIN_CAN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	CAN_TX_PF	R	04h	MCAN TX Pin Function Selection Value
23-16	CAN_TX_PAD	R	3Bh	MCAN TX Pin used by BSL
15-8	CAN_RX_PF	R	04h	MCAN RX Pin Function Selection Value
7-0	CAN_RX_PAD	R	3Ch	MCAN RX Pin used by BSL

1.5.1.7 BSLPIN_INVOKE Register (Offset = 80111018h) [Reset = 00000000h]

BSLPIN_INVOKE is shown in Table 1-104.

Return to the Summary Table.

BSL Pin Invocation Configuration

Table 1-104. BSLPIN_INVOKE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R	0h	
14-13	GPIO_REG_SEL	R	0h	GPIO Module Selection
12-8	GPIO_PIN_SEL	R	12h	GPIO Pin Number in GPIO Module
7	GPIO_LEVEL	R	1h	GPIO Level Configuration for BSL Invocation
6-0	BSL_PAD	R	28h	BSL Invocation Pin Number

1.5.1.8 SRAMFLASH Register (Offset = 8011101Ch) [Reset = 00000000h]

SRAMFLASH is shown in Table 1-105.

Return to the Summary Table.

Table 1-105. SRAMFLASH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	DATAFLASH_SZ	R	20h	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.
25-16	SRAM_SZ	R	100h	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	2h	Defines the number of main flash banks
11-0	MAINFLASH_SZ	R	400h	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.

1.5.1.9 PLLSTARTUP0_4_8MHZ Register (Offset = 80111020h) [Reset = 00000000h]

PLLSTARTUP0_4_8MHZ is shown in Table 1-106.

Return to the Summary Table.

Table 1-106. PLLSTARTUP0_4_8MHZ Register Field Descriptions

Bit	Field	Type	Reset	Description
31	CAPBOVERRIDE	R	1h	Override Enable For Cap B 0h = 0 1h = 1
30-29	RESERVED	R	0h	
28-24	CAPBVAL	R	1h	Override Value for Cap B
23-22	RESERVED	R	0h	
21-16	CPCURRENT	R	Ah	Charge Pump Current
15-14	RESERVED	R	0h	
13-8	STARTTIMELP	R	16h	Startup time from Low Power Exit to Locked Clock in resolution of 1usec
7-6	RESERVED	R	0h	
5-0	STARTTIME	R	16h	Startup time from Enable to Locked Clock in resolution of 1usec

1.5.1.10 PLLSTARTUP1_4_8MHZ Register (Offset = 80111024h) [Reset = 00000000h]

PLLSTARTUP1_4_8MHZ is shown in Table 1-107.

Return to the Summary Table.

Table 1-107. PLLSTARTUP1_4_8MHZ Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	LPFRESC	R	FFh	Loop Filter Res C
23-18	RESERVED	R	0h	
17-8	LPFRESA	R	4h	Loop Filter Res A
7-5	RESERVED	R	0h	
4-0	LPFCAPA	R	1Fh	Loop Filter Cap A

1.5.1.11 PLLSTARTUP0_8_16MHZ Register (Offset = 80111028h) [Reset = 00000000h]

PLLSTARTUP0_8_16MHZ is shown in Table 1-108.

Return to the Summary Table.

Table 1-108. PLLSTARTUP0_8_16MHZ Register Field Descriptions

Bit	Field	Type	Reset	Description
31	CAPBOVERRIDE	R	1h	Override Enable For Cap B 0h = 0 1h = 1
30-29	RESERVED	R	0h	
28-24	CAPBVAL	R	1h	Override Value for Cap B
23-22	RESERVED	R	0h	
21-16	CPCURRENT	R	Ah	Charge Pump Current
15-14	RESERVED	R	0h	
13-8	STARTTIMELP	R	16h	Startup time from Low Power Exit to Locked Clock in resolution of 1usec
7-6	RESERVED	R	0h	
5-0	STARTTIME	R	16h	Startup time from Enable to Locked Clock in resolution of 1usec

1.5.1.12 PLLSTARTUP1_8_16MHZ Register (Offset = 8011102Ch) [Reset = 00000000h]

PLLSTARTUP1_8_16MHZ is shown in Table 1-109.

Return to the Summary Table.

Table 1-109. PLLSTARTUP1_8_16MHZ Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	LPFRESC	R	FFh	Loop Filter Res C
23-18	RESERVED	R	0h	
17-8	LPFRESA	R	4h	Loop Filter Res A
7-5	RESERVED	R	0h	
4-0	LPFCAPA	R	Fh	Loop Filter Cap A

1.5.1.13 PLLSTARTUP0_16_32MHZ Register (Offset = 80111030h) [Reset = 00000000h]

PLLSTARTUP0_16_32MHZ is shown in Table 1-110.

Return to the Summary Table.

Table 1-110. PLLSTARTUP0_16_32MHZ Register Field Descriptions

Bit	Field	Type	Reset	Description
31	CAPBOVERRIDE	R	1h	Override Enable For Cap B 0h = 0 1h = 1
30-29	RESERVED	R	0h	
28-24	CAPBVAL	R	1h	Override Value for Cap B
23-22	RESERVED	R	0h	
21-16	CPCURRENT	R	Ah	Charge Pump Current
15-14	RESERVED	R	0h	
13-8	STARTTIMELP	R	16h	Startup time from Low Power Exit to Locked Clock in resolution of 1usec
7-6	RESERVED	R	0h	
5-0	STARTTIME	R	16h	Startup time from Enable to Locked Clock in resolution of 1usec

1.5.1.14 PLLSTARTUP1_16_32MHZ Register (Offset = 80111034h) [Reset = 00000000h]

PLLSTARTUP1_16_32MHZ is shown in Table 1-111.

Return to the Summary Table.

Table 1-111. PLLSTARTUP1_16_32MHZ Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	LPFRESC	R	FFh	Loop Filter Res C
23-18	RESERVED	R	0h	
17-8	LPFRESA	R	4h	Loop Filter Res A
7-5	RESERVED	R	0h	
4-0	LPFCAPA	R	Fh	Loop Filter Cap A

1.5.1.15 PLLSTARTUP0_32_48MHZ Register (Offset = 80111038h) [Reset = 00000000h]

PLLSTARTUP0_32_48MHZ is shown in Table 1-112.

Return to the Summary Table.

Table 1-112. PLLSTARTUP0_32_48MHZ Register Field Descriptions

Bit	Field	Type	Reset	Description
31	CAPBOVERRIDE	R	1h	Override Enable For Cap B 0h = 0 1h = 1
30-29	RESERVED	R	0h	
28-24	CAPBVAL	R	1h	Override Value for Cap B
23-22	RESERVED	R	0h	
21-16	CPCURRENT	R	Ah	Charge Pump Current
15-14	RESERVED	R	0h	
13-8	STARTTIMELP	R	16h	Startup time from Low Power Exit to Locked Clock in resolution of 1usec
7-6	RESERVED	R	0h	
5-0	STARTTIME	R	16h	Startup time from Enable to Locked Clock in resolution of 1usec

1.5.1.16 PLLSTARTUP1_32_48MHZ Register (Offset = 8011103Ch) [Reset = 00000000h]

PLLSTARTUP1_32_48MHZ is shown in Table 1-113.

Return to the Summary Table.

Table 1-113. PLLSTARTUP1_32_48MHZ Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	LPFRESC	R	FFh	Loop Filter Res C
23-18	RESERVED	R	0h	
17-8	LPFRESA	R	4h	Loop Filter Res A
7-5	RESERVED	R	0h	
4-0	LPFCAPA	R	Fh	Loop Filter Cap A

1.5.1.17 TEMP_SENSE0 Register (Offset = 80111040h) [Reset = 00000000h]

TEMP_SENSE0 is shown in Table 1-114.

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.

Table 1-114. TEMP_SENSE0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R	0h	

1.5.1.18 BOOTCRC Register (Offset = 8011107Ch) [Reset = 00000000h]

BOOTCRC is shown in Table 1-115.

Return to the Summary Table.

BOOTCRC records the 32-bit CRC of all locations in OPEN including reserved locations.

Table 1-115. BOOTCRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R	0h	

1.6 Memory configuration

On the MSPM33C3 devices the MEMCFG registers controlling configurations on RAM and ROM. Some of the key features of the MEMCFG registers are

1. Configuring SRAM for write access based on ECC or Data bits
2. Configuring Wait-states on SRAM based on SRAM bank.
3. Configuring Wait-states on ROM

For more details please see Section 1.6.1.

1.6.1 memcfg Registers

Table 1-116 lists the memory-mapped registers for the memcfg 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. MEMCFG Registers

Offset	Acronym	Register Name	Section
1000h	TEST	RAM TEST Register	Section 1.6.1.1
1004h	RAM_CACHE_CONFIG	RAM CACHE configuration	Section 1.6.1.2
1008h	RAM_WS_CONFIG	RAM wait state configuration	Section 1.6.1.3
100Ch	RAM_CACHE_CLEAR	RAM cache clear	Section 1.6.1.4
1010h	ROM_WS_CONFIG	ROM wait state configuration	Section 1.6.1.5

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. memcfg Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value

1.6.1.1 TEST Register (Offset = 1000h) [Reset = 00000000h]

TEST is shown in Table 1-118.

Return to the Summary Table.

RAM TEST Register

Table 1-118. TEST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7-6	TEST_GLXMP_2	R/W	0h	Selects the defferent modes for GLXMP_2 RAM: 00: Functional Mode. 01: Writes are allowed to data bits only. No write to ECC bits. 10: Writes are allowed to ECC bits only. No write to data bits. 11: Same as functional mode Reset type: SYSRSn
5-4	TEST_GLXMP_1	R/W	0h	Selects the defferent modes for GLXMP_1 RAM: 00: Functional Mode. 01: Writes are allowed to data bits only. No write to ECC bits. 10: Writes are allowed to ECC bits only. No write to data bits. 11: Same as functional mode Reset type: SYSRSn
3-2	TEST_GLXMP_0	R/W	0h	Selects the defferent modes for GLXMP_0 RAM: 00: Functional Mode. 01: Writes are allowed to data bits only. No write to ECC bits. 10: Writes are allowed to ECC bits only. No write to data bits. 11: Same as functional mode Reset type: SYSRSn
1-0	TEST_ULL	R/W	0h	Selects the defferent modes for ULL RAM: 00: Functional Mode. 01: Writes are allowed to data bits only. No write to ECC bits. 10: Writes are allowed to ECC bits only. No write to data bits. 11: Same as functional mode Reset type: SYSRSn

1.6.1.2 RAM_CACHE_CONFIG Register (Offset = 1004h) [Reset = 00000000h]

RAM_CACHE_CONFIG is shown in Table 1-119.

Return to the Summary Table.

RAM CACHE configuration

Table 1-119. RAM_CACHE_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R-0	0h	Reserved
3	GLXMP_2_CACHE_ENABLE	R/W	0h	Cache configuration for GLXMP_2 RAM: 1: Cache enable. 0: Cache support disable. Reset type: SYSRSn
2	GLXMP_1_CACHE_ENABLE	R/W	0h	Cache configuration for GLXMP_1 RAM: 1: Cache enable. 0: Cache support disable. Reset type: SYSRSn
1	GLXMP_0_CACHE_ENABLE	R/W	0h	Cache configuration for GLXMP_0 RAM: 1: Cache enable. 0: Cache support disable. Reset type: SYSRSn
0	ULL_CACHE_ENABLE	R/W	0h	Cache configuration for ULL RAM: 1: Cache enable. 0: Cache support disable. Reset type: SYSRSn

1.6.1.3 RAM_WS_CONFIG Register (Offset = 1008h) [Reset = 00000000h]

RAM_WS_CONFIG is shown in Table 1-120.

Return to the Summary Table.

RAM wait state configuration

Table 1-120. RAM_WS_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R-0	0h	Reserved
3	GLXMP_2_WS_ENABLE	R/W	0h	Wait state configuration for GLXMP_2 RAM: 0: 1wait state disable. 1: 1 wait enable. Reset type: SYSRSn
2	GLXMP_1_WS_ENABLE	R/W	0h	Wait state configuration for GLXMP_1 RAM: 0: 1wait state disable. 1: 1 wait enable. Reset type: SYSRSn
1	GLXMP_0_WS_ENABLE	R/W	0h	Wait state configuration for GLXMP_0 RAM: 0: 1wait state disable. 1: 1 wait enable. Reset type: SYSRSn
0	ULL_WS_ENABLE	R/W	0h	Wait state configuration for ULL RAM: 0: 1wait state disable. 1: 1 wait enable. Reset type: SYSRSn

1.6.1.4 RAM_CACHE_CLEAR Register (Offset = 100Ch) [Reset = 00000000h]

RAM_CACHE_CLEAR is shown in Table 1-121.

Return to the Summary Table.

RAM cache clear

Table 1-121. RAM_CACHE_CLEAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R-0	0h	Reserved
3	GLXMP_2_WS_CACHE_CLEAR	W	0h	Cache clear configuration for GLXMP_2 RAM: 0: wont clear the cache valid slots. 1: Clear the all cache valid slots and write back the write buffer data to memory macro. Reset type: SYSRSn
2	GLXMP_1_WS_CACHE_CLEAR	W	0h	Cache clear configuration for GLXMP_1 RAM: 0: wont clear the cache valid slots. 1: Clear the all cache valid slots and write back the write buffer data to memory macro. Reset type: SYSRSn
1	GLXMP_0_CACHE_CLEAR	W	0h	Cache clear configuration for GLAMP_0 RAM: 0: wont clear the cache valid slots. 1: Clear the all cache valid slots and write back the write buffer data to memory macro. Reset type: SYSRSn
0	ULL_CACHE_CLEAR	W	0h	Cache clear configuration for ULL RAM: 0: wont clear the cache valid slots. 1: Clear the all cache valid slots and write back the write buffer data to memory macro. Reset type: SYSRSn

1.6.1.5 ROM_WS_CONFIG Register (Offset = 1010h) [Reset = 00000000h]

ROM_WS_CONFIG is shown in Table 1-122.

Return to the Summary Table.

ROM wait state configuration

Table 1-122. ROM_WS_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R-0	0h	Reserved
0	ROM_WS_ENABLE	R/W	0h	Wait state configuration ROM: 0: 1wait state disable. 1: 1 wait enable. Reset type: SYSRSn

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 143
2.2 Power Management (PMU) 146
2.3 Clock Module (CKM) 152
2.4 Clock Monitors 168
2.5 Frequency Clock Counter (FCC) 171
2.6 System Controller (SYSCTL) 174
2.7 Quick Start Reference 191
2.8 SYSCTL Registers 194

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.

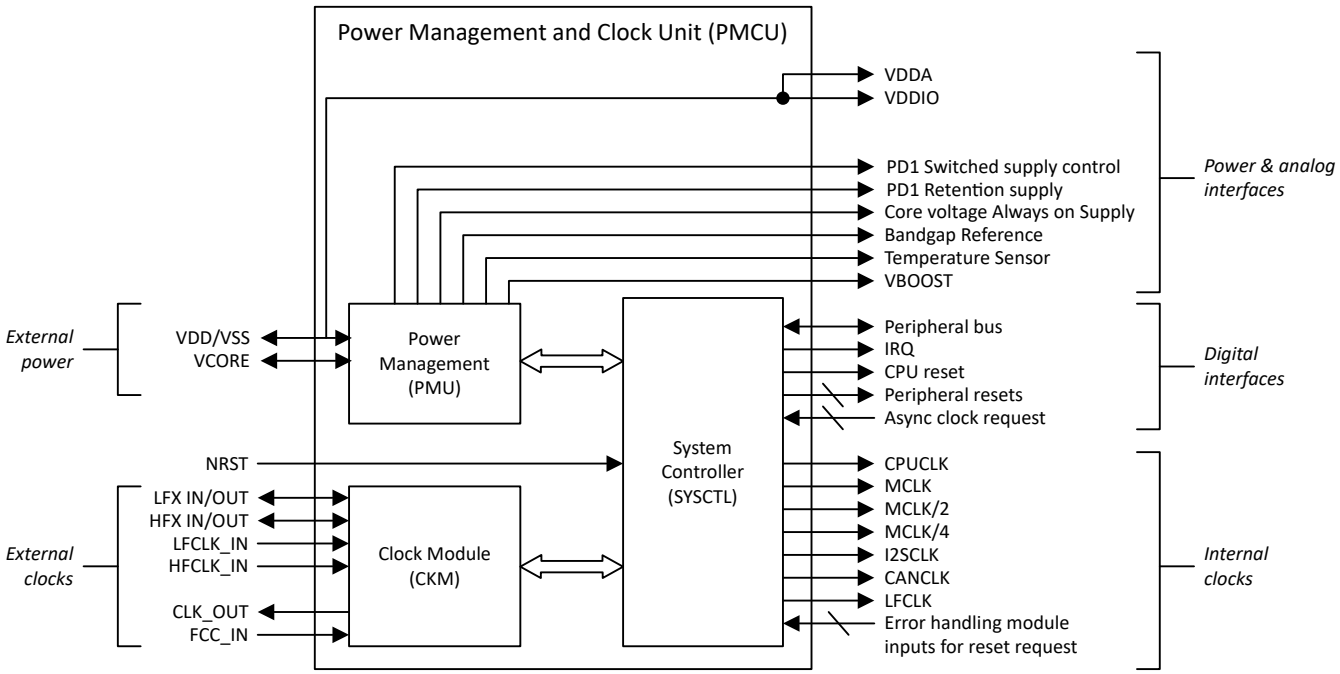


Figure 2-1. PMCU Top-Level Diagram

Note

Not all devices have all of the PMCU features shown in Figure 2-1. See the device-specific data sheet to understand the features present on a given device.

Using this Guide

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. While PD1 is disabled in STOP and STANDBY mode, the CPU registers, SRAM bank 0, and some 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.

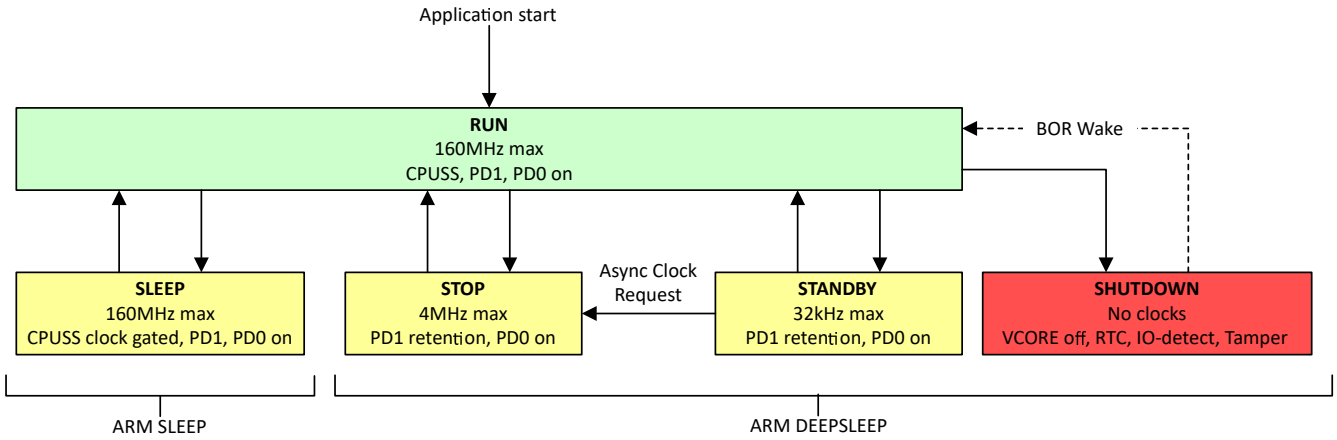
The device-specific data sheet describes which peripherals on a device are in PD1 and which are in PD0.

The device also has a primary external supply (VDD) domain that provides power to the IO and analog peripherals as well as a supplementary backup supply (VBAT) domain that provides power to the low-frequency subsystem.

2.1.2 Operating Modes

MSPM33C3 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.



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. See the *Operating Mode Selection* section for information on how to configure the device for a particular operating mode.

Operating Mode Concept

MSPM33 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.

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.

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 in STOP mode. All other wakeups except from STANDBY mode brings the device to RUN mode.

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, 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. In RUN mode, the MCLK and the CPUCLK are derived from a fast clock source like SYSOSC, HFCLK or SYSPLL. The VCORE is supplied by the on-chip regulator. All functions are available at the highest frequency as specified in the device-specific datasheet.

Software if needed can pick lower PLL frequency or lower bus clock frequency to save dynamic power by appropriate sequencing of PLL dividers and locak wait. The flash and SRAM wait state for the specific lower RUN mode clock frequency must be set as provided in the device-specific datasheet

2.1.2.2 SLEEP Mode

In SLEEP mode, the CPU is disabled (clock gated). Otherwise, the device configuration is the same as RUN. The SLEEP policy is determined by the current RUN policy when SLEEP mode is entered.

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. For details on peripheral availability, please refer to the *supported functionality by operating mode* table within the device-specific datasheet.

During STOP mode operation, the SYSPLL//HFCLKIN are switched off.

2.1.2.4 STANDBY Mode

In STANDBY mode, the CPU, SRAM, and PD1 peripherals are disabled and in retention. PD0 peripherals, 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.

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. can synchronously wakeup the device. Comparator wakeup is not supported to avoid high power consumption.
- **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 GPIO interrupt triggers an asynchronous fast clock request to wake the system. Other PD0 peripherals (such as UART, I2C, GPIO) 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, core clocks are not 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. LFSS register contents are maintained in shutdown mode while VBAT is powered. The BOR and bandgap circuit are disabled.

Specific IO configuration settings are retained in the SHUTDOWN operating mode.

- Drive strength
- Pullup enable
- Pulldown enable

- Input enable
- Output enable
- Output state

The device can wake through a wake-up capable IO, a debug connection, or NRST. On devices which have VBAT supply, LFSS peripherals are able to wake the device from shutdown mode.

SHUTDOWN mode has the lowest current consumption of any operating mode. Exiting SHUTDOWN mode triggers a BOR.

2.1.2.6 Suspended Low-Power Mode Operation

Some peripherals can be configured to temporarily suspend STANDBY mode operation to handle a temporary activity or process an event. An asynchronous fast clock request allows STANDBY mode to be suspended.

Suspended 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 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

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
- 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
  - The BOR monitor allows for configuring of threshold voltages to report an NMI
- Bandgap voltage reference supporting the BOR, core regulator, and analog peripherals
- Analog mux VBOOST unit for increasing analog mux performance

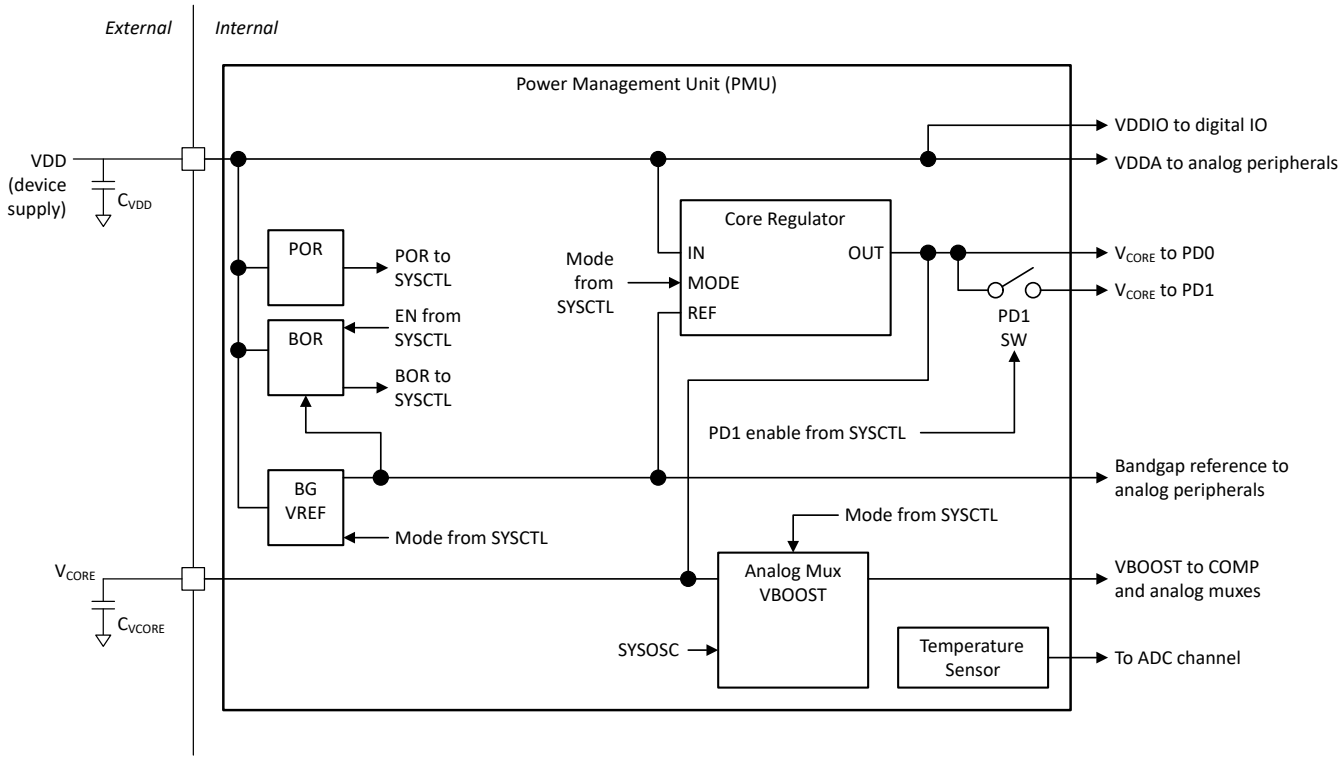


Figure 2-2. MSPM33C3 PMU Block Diagram

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 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).

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.

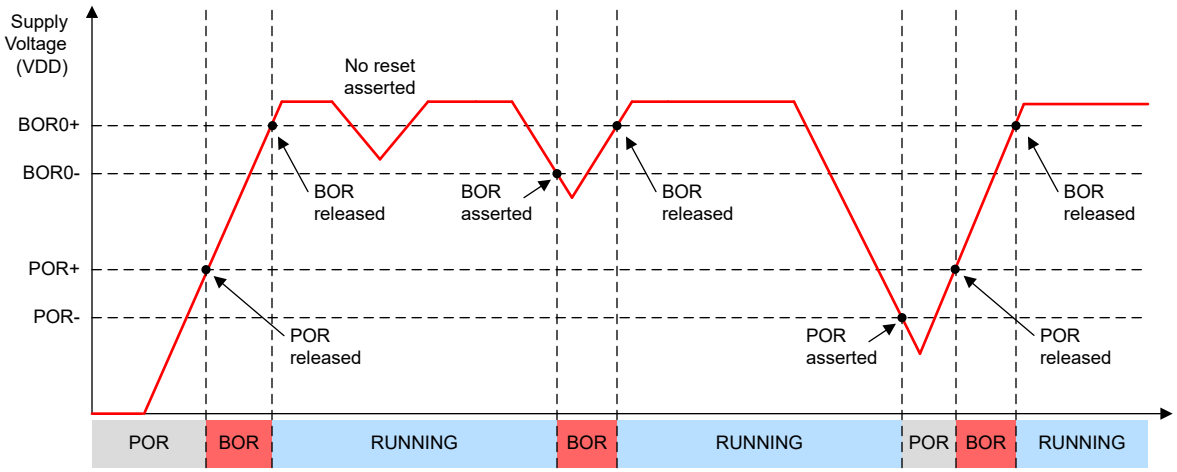


Figure 2-3. POR/BOR vs. Supply Voltage (VDD)

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
- 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 ANACPUMPCFG control bits in the GENCLKCFG register in SYSCTL (please refer to the table below)
- The enabling of HFXT as a HFCLK clock source

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.

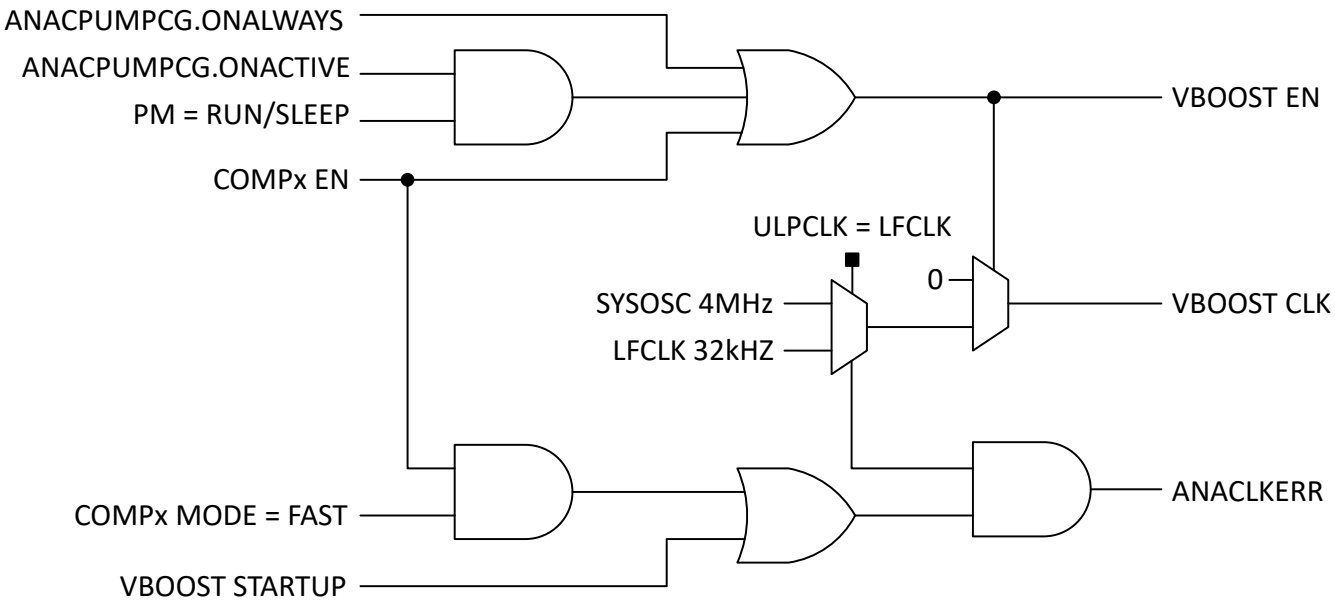


Figure 2-4. VBOOST Request Logic

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 ULPCLK tree source. The VBOOST clock is selected by automatically by SYSCTL according to the logic shown in *VBOOST Request Logic* and described here:

- When the 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:

- Fast mode COMP operation
- VBOOST is starting up (transitioning from disabled to enabled)

SYSCTL does not change the current system clock configuration for a VBOOST request. If the ULPCLK tree is sourced from LFCLK in one of these scenarios, SYSCTL asserts an ANACLKERR status in the SYSSTATUS register in SYSCTL to indicate to application software that there is a mismatch between the VBOOST clock requirement and the current 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	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	RUN, SLEEP, STOP
COMP enabled	Fast mode (FAST)	4MHz	Not LFCLK	RUN, SLEEP, STOP
	Ultra-low-power mode (ULP) (2)	4MHz or 32kHz	Don't care	RUN, SLEEP, STOP, STANDBY0
VBOOST running with no peripheral request active (due to ANACPUMPCFG!=0x0)(2)		4MHz or 32kHz	Don't care	RUN, SLEEP, STOP, STANDBY0
HFXT		4MHz or 32kHz	Don't care	RUN, SLEEP, STOP,

(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 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 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 COMP is enabled in FAST mode. When the VBOOST circuit has started and is ready, the ANACPUMPGOOD bit is set in the SYSSTATUS register in SYSCTL.

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.

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)

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 supports correction with an internal resistor, depending on the device capabilities.
- 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
  - Refer to the device-specific data sheet to determine if multiple frequency settings for SYSOSC are supported
- A secondary output with a constant 4MHz frequency for use by MFCLK.
  - When f_{SYSOSC} = 4MHz, the MFCLK output is derived from SYSOSC directly.
  - When f_{SYSOSC} is not 4MHz, the MFCLK 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.

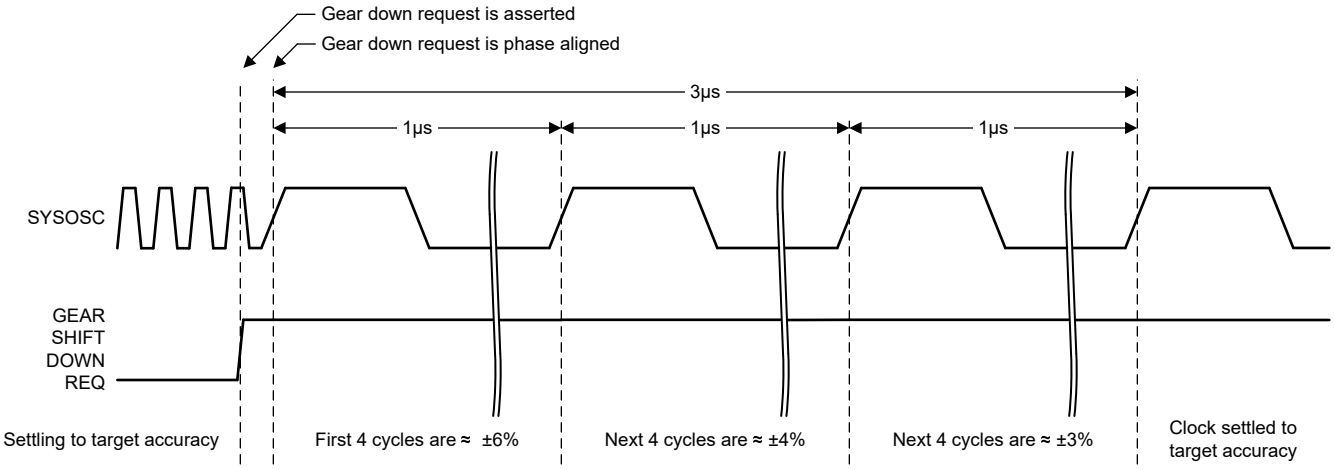


Figure 2-5. SYSOSC Gearshift Down Timing

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.

2.3.1.2.2 SYSOSC Frequency and User Trims

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 HSCLK.

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 an internal resistor to stabilize the SYSOSC frequency by providing a precise reference current for the SYSOSC.

2.3.1.2.3.1 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.

2.3.1.3 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.

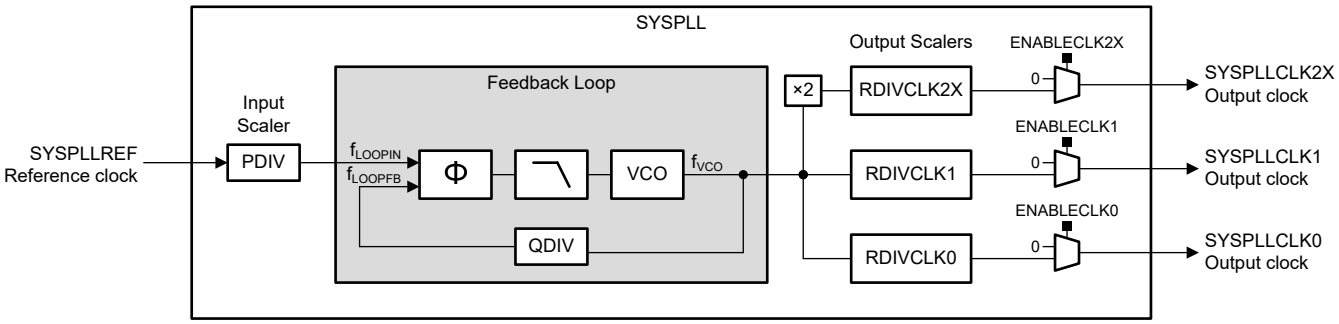


Figure 2-6. SYSPLL Diagram

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 below.

$$SYSPLLREF_{DIV} = 2^{PDIV} \quad (1)$$

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 below.

$$SYSPLLFB_{DIV} = QDIV + 1 \quad (2)$$

The output frequency of the VCO f_{VCO} is given below.

$$f_{VCO} = f_{SYSPLLREF} \times SYSPLLFB_{DIV} / SYSPLLREF_{DIV} \quad (3)$$

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. The following 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. The following shows how to compute the effective SYSPLLCLK0 divider based on a given RDIVCLK0 setting, and the equation below 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 160MHz to source MCLK and 40MHz to source CANCLK

To achieve this, the VCO can be configured for 320MHz through the use of PDIV and QDIV. Then, SYSPLLCLK0 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)
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 20 (set SYSPLLCFG1.QDIV to 19), giving f_{VCO} =320MHz (16MHz multiplied by 20)
7. Set the SYSPLL output dividers for SYSPLLCLK0 to /2 (set SYSPLLCFG0.RDIVCLK0 to 0x0) to get 160MHz at SYSPLLCLK0
8. Enable SYSPLLCLK0 outputs by setting the ENABLECLK0 bit in the SYSPLLCFG0 register

9. With SYSOSC enabled and running at base frequency (32 MHz, this is the default state out of reset), enable the SYSPLL by setting SYSPLLEN in the HSCLKEN register
10. Wait for the SYSPLLGOOD indication by testing SYSPLLGOOD in the CLKSTATUS register
11. Select SYSPLLCLK0 as the PLL output to the HSCLK mux by resetting MCLK2XVCO in the SYSPLLCFG0 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 SYSPLLCLK0 at 160 MHz
14. To configure the CANCLK enable CANEXTDIVEN and configure GENCLKEN.EXTDIVCAN to divide by 4 before sourcing CANCLK then, set the CANCLKSRC bit in the GENCLKCFG register

The SYSPLL divider values used in this example are summarized below for reference.

Table 2-3. 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	0x13	/20
Output clock 0 divider	SYSPLLCFG0	RDIVCLK0	0x0	/2
CAN clock divider	GENCLKEN	EXTDIVCAN	0x3	/4

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 160MHz output frequency is desired with f_{VCO} =160MHz, f_{VCO} = 160MHz can be derived from a SYSPLLREF clock 32MHz by setting PDIV to /8 and QDIV to /40. However, this gives slower startup because f_{LOOPIN} is <8MHz. The same result can be achieved by setting PDIV to /2 and QDIV to /10, but because f_{LOOPIN} is 16MHz 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}).

SYSPLL supports four f_{LOOPIN} frequency ranges, given in Table 2-4. 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 2-4. SYSPLL Parameter Lookup

f_{LOOPIN}	Lookup Address (PARAM0)	Lookup Address (PARAM1)	SDK Symbol
32MHz ≤ FREQ ≤ 48MHz	0x8011.1038	0x8011.103C	DL_SYSCTL_SYSPLL_INPUT_FREQ_32_48_MHZ
16MHz ≤ FREQ < 32MHz	0x8011.1030	0x8011.1034	DL_SYSCTL_SYSPLL_INPUT_FREQ_16_32_MHZ
8MHz ≤ FREQ < 16MHz	0x8011.1028	0x8011.102C	DL_SYSCTL_SYSPLL_INPUT_FREQ_8_16_MHZ
4MHz ≤ FREQ < 8MHz	0x8011.1020	0x8011.1024	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 2-5 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 2-5. 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).

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. Once STARTLFXT is set by the application software, the internal LFOSC is disabled to conserve power while the LFXT starts in anticipation of LFCLK moving to the LFXT.

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. LFOSC cannot be re-enabled and LFCLK thus cannot be switched back to LFOSC other than by a software-triggered POR reset.

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. A fault will be generated if the LFCLK monitor for LFXT or LFCLK_IN is enabled. After the fault LFCLK will enable LFOSC and switch to LFOSC without the need of BOOTRST

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

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.

2.3.1.6 High Frequency Crystal Oscillator (HFXT)

The high frequency crystal oscillator (HFXT) can be used with standard crystals and resonators (4 - 48 MHz) 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 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 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
  - LFCLK: Fixed 32kHz clock, synchronized to MCLK/ULPCLK
  - HFCLK: High frequency external clock
- Peripheral Specific Clocks
  - CANCLK: CAN-FD module functional clock
  - RTCCLK: Fixed 32kHz clock direct to RTC
  - I2SCLK: I2S clock with divider options of 4, 8, 12, 16
- 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). MCLK and its divided version of MCLK/2 and MCLK/4 cover all the peripherals in PD1. It is typically the highest speed clock in the system and supports operation up to 160MHz 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. MCLK/4 also works as the source for PD0 peripherals.

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) up to 160MHz which can be sourced by:
  - **SYSPLLCLK0** or **SYSPLLCLK2X** (used to reach the max MCLK speed of 160MHz)
  - **HFCLK** for applications where the main clock needs to be as accurate as possible
- **SYSOSC** at 4, or 32MHz (the default clock source after reset)
- **SYSOSC** at 4MHz in STOP mode

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. The following table gives the proper register bit configurations for selecting different clocks for MCLK in RUN and SLEEP modes.

Table 2-6. MSPM33C3 MCLK Source Selection in RUN and SLEEP Mode

Desired Source	MCLKCFG.USEHSCLK	HSCLKCFG.HSCLKSEL	SYSPLLCFG0.MCLK2XVCO
SYSOSC	0	Don't care	Don't care
SYSPLLCLK0	1	0	0
SYSPLLCLK2X	1	0	1
HFCLK	1	1	Don't care

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 or HFXT)

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.

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 (MCLK/4) 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, and the selected power mode.

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, SYSCTL disables MDIV/4 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 shall switch back to MCLK/4 divider. In this case it is the responsibility of the application software to ensure that ULPCLK is ≤40MHz in RUN and SLEEP modes by configuring MCLK/4 divider appropriately.

ULPCLK Behavior in STOP and STANDBY Modes

- In STOP mode, the MCLK tree runs from SYSOSC with a 4MHz rate.
- 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 2-7. MSPM33C3 ULPCLK by Operating Mode

Selected Power Mode	Configuration	Register Settings	ULPCLK Frequency
RUN or SLEEP (40MHz maximum)	MCLK source is SYSOSC (RUN, SLEEP)	MCLKCFG.USEHSCLK=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) (RUN, SLEEP)	MCLKCFG.USEHSCLK=0x1	ULPCLK is sourced from MCLK according to the MCLK configuration with f_{ULPCLK} = f_{MCLK}/4
STOP (4MHz maximum)	STOP with SYSOSC enabled	-	ULPCLK is sourced from SYSOSC with f_{ULPCLK} = 4MHz
STANDBY (32kHz maximum)	STANDBY with ULPCLK and LFCLK enabled (STANDBY0)	MCLKCFG.STOPCLKSTBY= 0x0	ULPCLK is sourced from LFCLK with f_{ULPCLK} = f_{LFCLK} = 32kHz
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 can either be 32, or 4 MHz, SYSCTL automatically applies a divider to SYSOSC to keep MFCLK at a constant 4 MHz. 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.

Requirements for Using MFCLK

1. 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.
2. 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.
3. 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.

2.3.2.5 LFCLK (Low-Frequency Clock)

LFCLK provides a continuous 32kHz clock to a variety of peripherals on the device. After a BOOTRST, LFCLK is initially sourced 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.

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 (STANDBY). To do so, set the STOPCLKSTBY bit in the MCLKCFG register in SYSCTL before entering STANDBY. In this state, the RTC, TIMG0 and TIMG1 are the only clocked peripherals. In case IWDT in VBAT is enabled, then LFOSC remains ON regardless of the LFCLK source selection

2.3.2.6 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:

- MCLK clock source
- SYSPLL reference clock
- CANCLK clock source
- I2S clock source

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).

2.3.2.7 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 muxed output from SYSPLL (SYSPLLCLK0 or SYSPLLCLK2X) followed by dedicated clock divider EXTDIVCAN. 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).

2.3.2.8 I2SCLK (I2S Functional Clock)

The I2SCLK is a functional clock provided directly to the I2S module from either the HFCLK (HFXT or HFCLK_IN) or the SYSOSC or SYSPLL clock divided output. I2SCLK is asynchronous from the main clock

(MCLK) for the highest possible accuracy. The I2SCLK clock configuration is provided within the I2S peripheral itself (review the I2S chapter for more detail). Corresponding selections are as below:

Table 2-8. I2SCLK Source Selection in RUN and Sleep Mode

CLKCFG[1:0]	Source Selection
00	SYSOSC selected
01	HFCLK selected
10	SYSPLL clock selected
11	Do not use

The SYSPLL clock source is used to synthesize the audio clock crystal to higher multiple so as to allow oversampling if needed. The divider in the clock source of SYSPLL is introduced to allow SYSPLL output frequency division to the generate the right audio clock while maintaining highest MCLK for the performance from the device.

Table 2-9. I2SCLK Divider from SYSPLL

GENCLKEN[17:16]	Divider Setting
00	SYSPLL divider by 4
01	SYSPLL divider by 8
10	SYSPLL divider by 12
11	SYSPLL divider by 16

2.3.2.9 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.

2.3.2.10 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
- 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.
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, the clock divider must be enabled (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.

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.

2.3.2.11 Direct Clock Connections for Infrastructure

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, ADCs, OPAs, and GPAMP (see device specific datasheet for availability of peripherals). 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, ADC, 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 MSPM33C3 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.

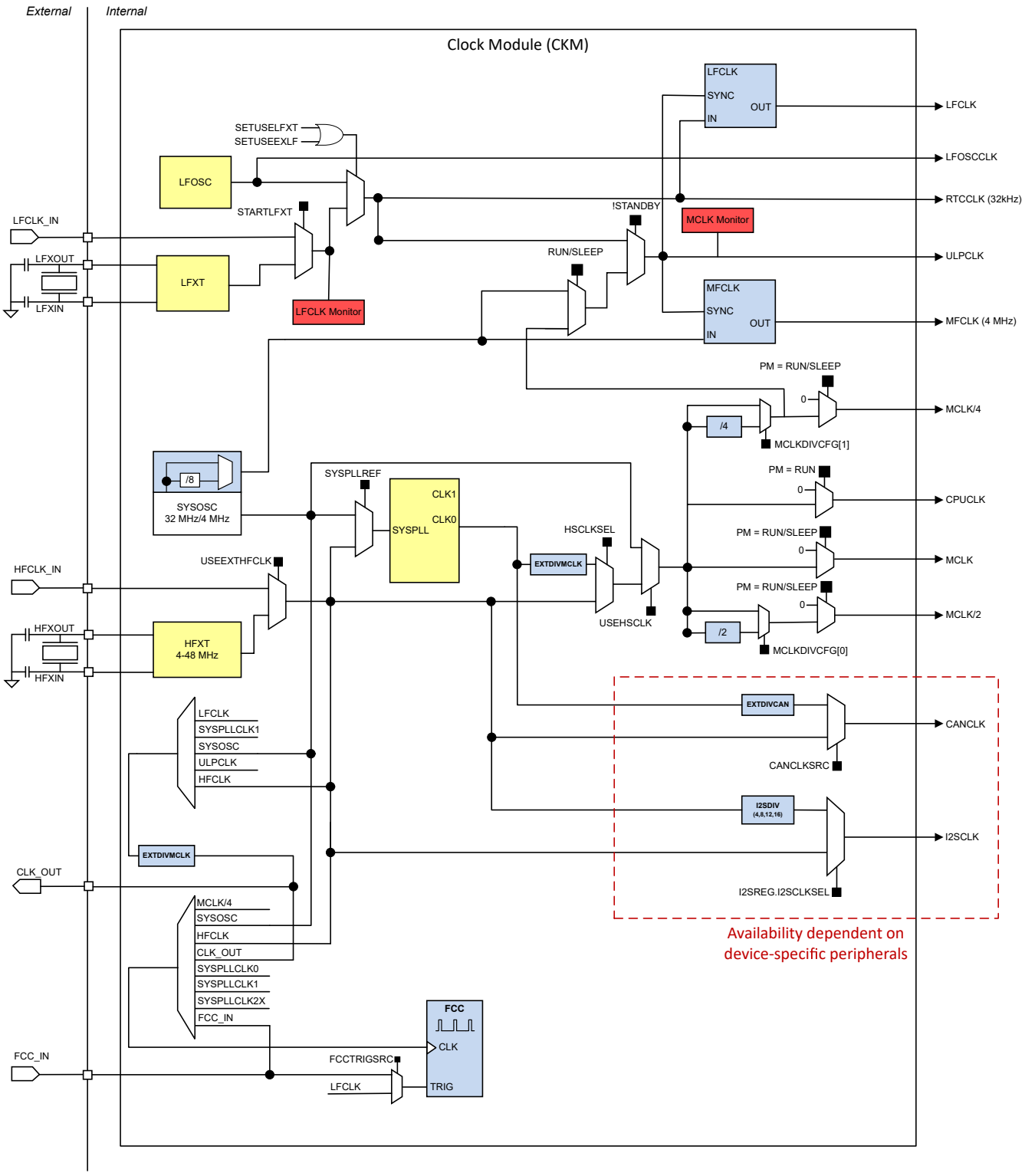


Figure 2-7. MSPM33C3 Top Level Clocking Tree

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. LFCK 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.

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, SHA 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.

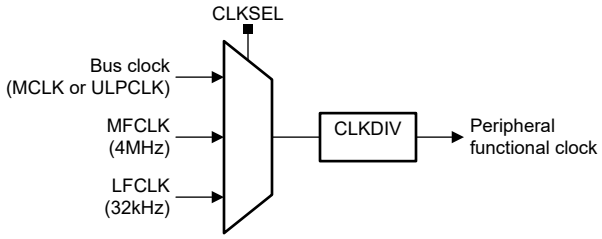


Figure 2-8. Peripheral Clock Selection Mux and Divider

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)

2.4 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.

2.4.1 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.

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 ULPCLK, 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 ULPCLK 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 ULPCLK is sourced from LFCLK, the failure will generate a BOOTRST automatically.
   b. If ULPCLK is not sourced from LFCLK, an NMI will be generated. In the NMI handling, application software can switch ULPCLK to LFCLK when the LFCLK fault is detected. This action will trigger the hardware to generate a BOOTRST automatically.
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.
   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.

2.4.2 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.

2.4.3 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.

2.4.3.1 LFOSC Startup Monitor

The LFOSC is started automatically after a BOOTRST. The LFOSC takes some time to start. 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. See the device-specific data sheet for the LFOSC startup time.

2.4.3.2 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.

2.4.3.3 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 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.

2.4.3.4 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.

2.4.3.5 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.

2.5 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/4
- SYSOSC
- HFCLK
- CLK_OUT
- SYSPLL (any of the three SYSPLL outputs)
- The external FCC input (FCC_IN)

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 will cause the FCC to capture the number of source clock pulses which occurred within 1 to 32 32.768kHz clock periods of the LFXT (30.5 µs).

The FCC counter is 22 bits and supports counting from 0 up to $2^{22} – 1$ or 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, it cannot be used for both functions during the same FCC capture. It must be configured as either the FCC clock source or the FCC trigger.

2.5.1 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 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.

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.

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.

2.5.2 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 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%
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%

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.

2.6 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.

All power, clock, and reset configuration is done through the SYSCTL memory-mapped register interface.

2.6.1 Resets and Device Initialization

The SYSCTL manages device reset levels and device initialization.

2.6.1.1 Reset Levels

The device has five reset levels:

1. Power-on reset (POR)
2. Brownout reset (BOR)

3. Boot reset (BOOTRST)
4. System reset (SYSRST)
5. CPU reset (CPURST)

The relationships between reset levels are given in Figure 2-9.

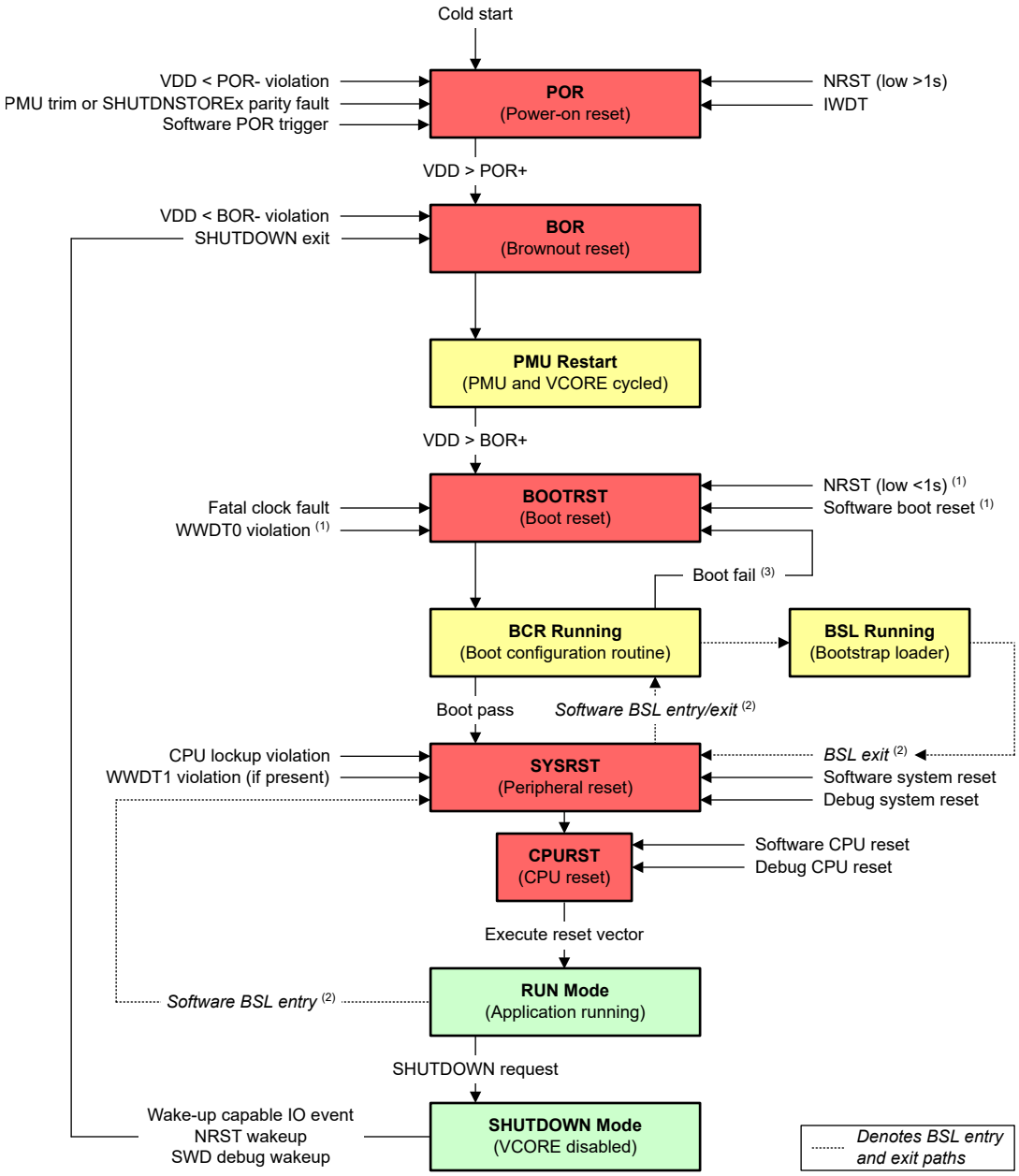


Figure 2-9. Device Reset Levels

(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, 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 *Boot Fail Handling*.

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.

2.6.1.1.1 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
- VBAT Domain WDT or Independent Watchdog Timer (IWDT)

A POR always resets the shutdown memory, re-enables the NRST/SWD pin functions (if disabled), and triggers a BOR.

2.6.1.1.2 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).

2.6.1.1.3 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).

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 *Boot Fail Handling*).

2.6.1.1.4 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 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.
- 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.

2.6.1.1.5 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.

2.6.1.2 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), unless specifically noted in the device datasheet.
- 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
- SYSPLL is disabled
- HFXT is disabled
- Peripherals are disabled
- Any flash sectors configured to be write protected at boot are write protected

2.6.1.3 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.

2.6.1.4 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

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.

2.6.1.5 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.

2.6.1.6 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.

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										

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
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

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 themselves retain their 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 device needs to be reconfigured
- **RESETCAUSE<0x04:** The SHUTDNSTOREx memory (if present) was reset and device needs 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 constents 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
```

```
    }
    if (cause<0x0C)
    {
        // RTC/LFXT/LFCLK state was lost
    }
    if (cause<0x1C)
    {
        // The peripherals were reset
    }
}
```

2.6.1.7 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.

2.6.1.8 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.

2.6.2 Operating Mode Selection

The device operating mode is configured through the use of the following:

1. Policy bits in the MCLKCFG registers in SYSCTL (to control the behavior of SYSOSC in STANDBY mode)
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))

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 (1)	STOP	STANDBY		SHUTDOWN (if present)
Register	Bit				STANDBY0	STANDBY1	
MCLKCFG	STOPCLKSTBY	-	-	-	0	1	-
PMODECFG	DSLEEP	-	-	00	01	01	10
SCR	SLEEPDEEP	0	0	1	1	1	1

(1) 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.

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.

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.

2.6.3 Asynchronous Fast Clock Requests

Peripherals are configured to asynchronously assert a hardware request to the SYSCTL for a fast clock source when the device is operating in STANDBY mode. This mechanism for applications where the ULPCLK tree is

sourced from LFCLK (at 32kHz), but a faster SYSOSC (at 4 MHz) 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 GPIO toggle validation thriugh the glitch filter).

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 TIMG8_x, leaving TIMG8_x and the RTC as the only clocked peripherals. To wake up the device from this state where the bus clock (ULPCLK) is disabled, a TIMG8_x 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 described.

Asynchronous fast clock requests temporarily provide peripherals with bus clock (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 STANDBY mode, the low power state is temporarily suspended to support running the bus clock (ULPCLK) at the SYSOSC in 4 MHz mode, SYSOSC start-up in 4 MHz will be forced in this state.
2. Device in STANDBY mode and temporarily operates in STOP mode. If there is an interrupt or request for the CPU or DMA to be woken up, then device transitions to RUN mode.
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 switches 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
- Asynchronous fast clock requests are globally blocked by setting the BLOCKASYNCALL bit in the SYSOSCCFG register in SYSCTL

Peripheral Support

The RTC, TIMG8_x, GPIO, I^2C, and UART peripherals all provide support for generating an asynchronous fast clock request. The purpose, request source, and configuration requirements for these peripherals are given in Table 2-14.

Table 2-14. Peripheral Support for Asynchronous Fast Clock Requests

Peripheral	Purpose	Request Source	Configuration
RTC (if present)	Fast CPU wake from RTC event	RTC IRQ to CPU	An RTC IRQ event always generates an asynchronous fast clock request when the device is in STANDBY1 mode; this is needed to wake the device as the main ULPCLK is disabled to reduce power consumption. The RTC can also be configured to generate an asynchronous fast clock request in any operating mode by clearing the BLOCKASYNC bit in the RTC CLKCFG register; this provides the lowest latency RTC event response.

Table 2-14. Peripheral Support for Asynchronous Fast Clock Requests (continued)

Peripheral	Purpose	Request Source	Configuration
TIMG8_x	Fast CPU wake from TIMGx event	TIMGx, IRQ to CPU	An IRQ event from TIMG8_x generates an asynchronous fast clock request when the device is in STANDBY1 mode and the corresponding IMASK interrupt is set in the TIMG8_x registers. This is needed to wake the device as the ULPCLK is disabled to reduce power consumption.
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 STANDBY mode 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.
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.

Asynchronous Fast Clock Request Logic

The logic for asserting a fast clock request is given in the following figure.

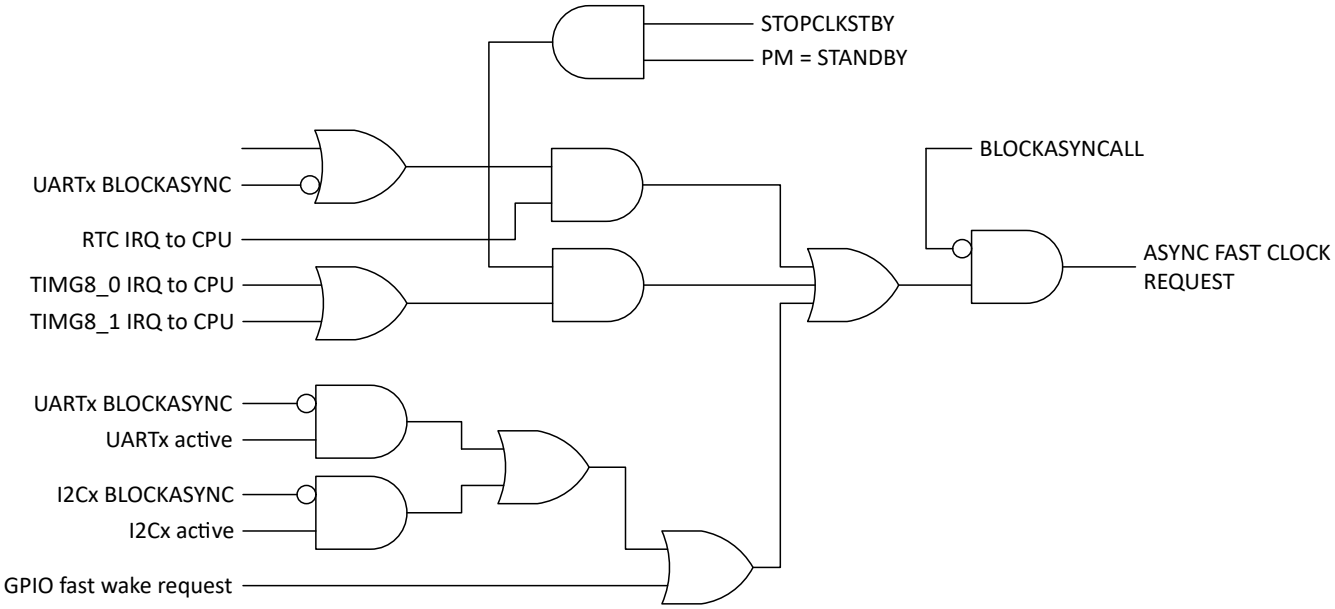


Figure 2-10. Asynchronous Fast Clock Request Logic

Note

Not all the devices support RTC or comparator, check the device specific data sheet to determine if these peripherals are present or not.

2.6.4 SRAM Write Protection

Certain applications need to place read-only data into SRAM. This can occur if code is placed into SRAM (for zero wait state execution) or if critical lookup tables are placed in SRAM (for zero wait state reads). In these cases, especially when code is to be executed from the SRAM, it is desirable to prevent unintentional writes to SRAM addresses that can corrupt executable code in the event of a buffer overrun or a stack overflow. Likewise,

it is desirable to prevent execution from nonwrite-protected SRAM addresses. To improve robustness of data in stored in SRAM, SYSCTL provides a write-exclusive-execute boundary mechanism.

To use this feature, first load the read-only data into the desired SRAM address, then configure the SRAM address range to be write protected. SRAM contents which are to be read-execute (no writes) should be placed into the upper portion of SRAM. SRAM contents which are to be read-write (no execute) should be placed into the lower portion of the SRAM. Then, the SRAMBOUNDARY register may be written with the desired boundary to partition the SRAM into two regions, with the lower region being RW and the upper region being RX.

2.6.5 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 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 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.

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.

2.6.6 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 itself does not need to have knowledge of the bank it resides in 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 considerations apply when swapping the flash bank address space. See the nonvolatile memory system chapter of this guide for considerations when using flash bank address swapping.

2.6.7 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 serial wire debug (SWD) pins also remain locked until application software sets the RELEASE bit**. As a result, a debug connection cannot establish when waking up from SHUTDOWN mode until the IO are released by application software.

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.

2.6.8 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
- PMODECFG
- FCC
- RSTCAUSE (read-to-clear), RESETLEVEL, RESETCMD
- BORTHRESHOLD, BORCLRCMD
- SHDNIOREL (if present)
- SHUTDNSTOREx (if present)
- MLDOPENCFG

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.

2.6.9 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.

2.6.10 Error Handling

MSPM33 devices include several diagnostic mechanism to detect errors at runtime. Table 2-15 lists error sources and their corresponding handling mechanism.

Note

Not all MSPM33 devices support all diagnostic features. For example, some devices have partial ECC coverage on memories. 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.

Table 2-15. Error Sources and Handling Mechanisms

Error Source	Error	Handling Mechanism
Flash	Non-correctable ECC error	• 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 • The FLASHDED sticky bit is set in the SYSSTATUS register in SYSCTL
	Correctable ECC error	• A FLASHSEC interrupt is also generated in SYSCTL • The FLASHSEC sticky bit is set in the SYSSTATUS register in SYSCTL
SRAM	Non-correctable ECC error	• An SRAMDED nonmaskable interrupt is generated to the processor
	Correctable ECC error	• A SYSCTL SRAMSED interrupt is generated to the processor
	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
SHUTDNSTORExMemory (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 • 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

Table 2-15. Error Sources and Handling Mechanisms (continued)

Error Source	Error	Handling Mechanism
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

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.

2.6.11 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 Table 2-16.

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

2.6.11.1 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.

Table 2-17. SYSCTL CPU Interrupt Event Sources (continued)

Index (IIDX)	Name	Description
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.
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 *Using Event Registers* for guidance on configuring these registers for CPU interrupts.

2.6.11.2 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 2-18.

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	Security Error	Access violation to the secure resources.
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 was detected.
7	VBATDN	VBAT LDO output is not in valid range

Table 2-18. Nonmaskable Interrupt Event Table (continued)

Index (IIDX)	Name	Description
8	VBATUP	VBAT LDO output is valid

The CPU nonmaskable interrupt event configuration is managed with the NMIIIDX, NMIRIS, NMIISET, and NMIICLR event management registers. See Section 8.2.5 for guidance on configuring the interrupt management registers.

2.7 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.

2.7.1 Default Device Configuration

The default operating configuration of the device provides basic functionality which can be suitable for many applications without modification.

MSPM33C3 devices power up and release reset for execution of application code when the external supply (VDD and VSS) reaches $V_{DD(min)}$. 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.

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. 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.
- In STOP mode, PD1 peripheras are power gated and are kept in retention (unavailable for use). The default STOP mode is designed for scenarios where power optimization is important but the clock faster than 32 kHz is required. MCLK tree will run at 4 MHz and peripherals in PD0 see the bus clock (ULPCLK) at 4 MHz..
  - PD0 peripherals configured to run from LFCLK continue to run at 32kHz.
- In STANDBY mode, 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.

2.7.2 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.

2.7.3 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. The device can run at lower clock frequency hence the divider from SYSOSC output automatically changes to provide a clock output of 4 MHz.

2.7.4 Optimizing Power Consumption in STANDBY Mode

In STANDBY mode, if only RTC, TIMG8_x, 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, TIMG8_x running (STANDBY1). See the LFCLK section. In this state, RTC, TIMG8_x, or asynchronous activity/event will trigger an asynchronous fast clock request to wake the system.

2.7.5 Increasing MCLK and ULPCLK 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 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/4 (12MHz).

If precision clocking is needed for the CAN-FD controller or the I2S instances, and high CPU performance (high MCLK) is also required, it is also possible to source the CAN-FD controller (CANCLK) and the I2S sampling clock (I2SCLK) from HFCLK directly, asynchronous to MCLK, while running MCLK at maximum frequency using the PLL for best compute performance.

2.7.6 Configuring MCLK for Maximum Speed

The best CPU compute performance is obtained by using the SYSPLL to generate an 160MHz clock from either the SYSOSC reference or HFXT. To configure the SYSPLL to generate an 816MHz output sourcing MCLK, see the SYSPLL configuration section.

Running MCLK at 160MHz also provides the best possible timer resolution for TIMA and TIMG peripherals in the PD1 domain (6.25ns).

2.7.7 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.

2.7.8 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.

2.8 SYSCTL Registers

Table 2-19 lists the memory-mapped registers for the SYSCTL 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 Registers

Offset	Acronym	Register Name	Section
1020h	IIDX	SYSCTL interrupt index	Section 2.8.1
1028h	IMASK	SYSCTL interrupt mask	Section 2.8.2
1030h	RIS	SYSCTL raw interrupt status	Section 2.8.3
1038h	MIS	SYSCTL masked interrupt status	Section 2.8.4
1040h	ISET	SYSCTL interrupt set	Section 2.8.5
1048h	ICLR	SYSCTL interrupt clear	Section 2.8.6
1050h	NMIIIDX	NMI interrupt index	Section 2.8.7
1060h	NMIRIS	NMI raw interrupt status	Section 2.8.8
1070h	NMIISET	NMI interrupt set	Section 2.8.9
1078h	NMIICLR	NMI interrupt clear	Section 2.8.10
1100h	SYSOSCCFG	SYSOSC configuration	Section 2.8.11
1104h	MCLKCFG	Main clock (MCLK) configuration	Section 2.8.12
1108h	HSCLKEN	High-speed clock (HSCLK) source enable/disable	Section 2.8.13
110Ch	HSCLKCFG	High-speed clock (HSCLK) source selection	Section 2.8.14
1110h	HFCLKCLKCFG	High-frequency clock (HFCLK) configuration	Section 2.8.15
1114h	LFCLKCFG	Low frequency crystal oscillator (LFXT) configuration	Section 2.8.16
1120h	SYSPLLCFG0	SYSPLL reference and output configuration	Section 2.8.17
1124h	SYSPLLCFG1	SYSPLL reference and feedback divider	Section 2.8.18
1128h	SYSPLLPARAM0	SYSPLL PARAM0 (load from FACTORY region)	Section 2.8.19
112Ch	SYSPLLPARAM1	SYSPLL PARAM1 (load from FACTORY region)	Section 2.8.20
1130h	SYSPLLPARAM2	SYSPLL PARAM2 (load from FACTORY region)	Section 2.8.21
1134h	SYSPLLLDOCTL	SYSPLL LDO CTL (load from FACTORY region)	Section 2.8.22
1138h	SYSPLLLDOPROG	SYSPLL LDO VOUT PROG (load from FACTORY region)	Section 2.8.23
113Ch	GENCLKEN	General clock enable control	Section 2.8.24
1140h	GENCLKCFG	General clock configuration	Section 2.8.25
1144h	PMODECFG	Power mode configuration	Section 2.8.26
1148h	MLDOLPENCFG	LDO Configuration Control	Section 2.8.27
1150h	FCC	Frequency clock counter (FCC) count	Section 2.8.28
1154h	PMULDOSPARECTL	LDO Spare Control	Section 2.8.29
1158h	SYSCTL_ECO_REG1	Sysctl ECO Reg 1	Section 2.8.30
115Ch	SYSCTL_ECO_REG2	Sysctl ECO Reg 2	Section 2.8.31
1180h	SYSTEMCFG	System configuration	Section 2.8.32
1184h	SRAMCFG	System SRAM configuration	Section 2.8.33
1200h	WRITELOCK	SYSCTL register write lockout	Section 2.8.34
1204h	CLKSTATUS	Clock module (CKM) status	Section 2.8.35
1208h	SYSSTATUS	System status information	Section 2.8.36
1220h	RSTCAUSE	Reset cause	Section 2.8.37
1300h	RESETLEVEL	Reset level for application-triggered reset command	Section 2.8.38
1304h	RESETCMD	Execute an application-triggered reset command	Section 2.8.39
1308h	BORTHRESHOLD	BOR threshold selection	Section 2.8.40
130Ch	BORCLRCMD	Set the BOR threshold	Section 2.8.41

Table 2-19. SYSCTL Registers (continued)

Offset	Acronym	Register Name	Section
1310h	SYSOSCFCLCTL	SYSOSC frequency correction loop (FCL) ROSC enable	Section 2.8.42
1314h	LFXTCTL	LFXT and LFCLK control	Section 2.8.43
1318h	EXLFCTL	LFCLK_IN and LFCLK control	Section 2.8.44
131Ch	SHDNIOREL	SHUTDOWN IO release control	Section 2.8.45
1320h	EXRSTPIN	Disable the reset function of the NRST pin	Section 2.8.46
1324h	SYSSTATUSCLR	Clear sticky bits of SYSSTATUS	Section 2.8.47
1328h	SWDCFG	Disable the SWD function on the SWD pins	Section 2.8.48
132Ch	FCCCMD	Frequency clock counter start capture	Section 2.8.49
1400h	SHUTDNSTORE0	Shutdown storage memory (byte 0)	Section 2.8.50
1404h	SHUTDNSTORE1	Shutdown storage memory (byte 1)	Section 2.8.51
1408h	SHUTDNSTORE2	Shutdown storage memory (byte 2)	Section 2.8.52
140Ch	SHUTDNSTORE3	Shutdown storage memory (byte 3)	Section 2.8.53
1410h	ADCSEQFRCGB	ADC Global Sequence Force	Section 2.8.54
1414h	ADCSEQFRCGBSEL	ADC Global Sequence Force Select	Section 2.8.55
1418h	M33SPARESOCLOCK1	M33C1 Spare SOC LOCK Reg 1	Section 2.8.56
141Ch	M33SPARESOCLOCK2	M33C1 Spare SOC LOCK Reg 2	Section 2.8.57
1420h	SYSCTL_READ_REG	Sysctl read only Reg	Section 2.8.58
3000h	FWEPROTMAIN	1 Sector Write-Erase per bit starting at address 0x0 of flash	Section 2.8.59
3014h	FWPROTMAINDATA	Read-Write Protection for first 4 Sectors of Data Bank	Section 2.8.60
3018h	FRXPROTMAINSTART	Flash RX Protection Start Address	Section 2.8.61
301Ch	FRXPROTMAINEND	Flash RX Protection End Address	Section 2.8.62
3020h	FIPPROTMAINSTART	Flash IP Protection Start Address	Section 2.8.63
3024h	FIPPROTMAINEND	Flash IP Protection End Address	Section 2.8.64
3038h	FLBANKSWPPOLICY	Flash Bank Swap Policy	Section 2.8.65
303Ch	FLBANKSWP	Flash MAIN bank address swap	Section 2.8.66
3044h	FWENABLE	Security Firewall Enable Register	Section 2.8.67
3048h	SECSTATUS	Security Configuration status	Section 2.8.68
3060h	INITDONE	INITCODE PASS	Section 2.8.69

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 Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
RC	R C	Read to Clear
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set
Reset or Default Value		

Table 2-20. SYSCTL Access Type Codes (continued)

Access Type	Code	Description
-n		Value after reset or the default value

2.8.1 IIDX Register (Offset = 1020h) [Reset = 00000000h]

IIDX is shown in Table 2-21.

Return to the Summary Table.

SYSCTL interrupt index

Table 2-21. IIDX Register 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. 0h = No interrupt pending 1h = LFOSCGOOD interrupt pending 2h = 2 3h = 3 4h = 4 5h = 5 6h = 6 7h = 7 8h = 8

2.8.2 IMASK Register (Offset = 1028h) [Reset = 00000000h]

IMASK is shown in Table 2-22.

Return to the Summary Table.

SYSCTL interrupt mask

Table 2-22. IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	
7	HSCLKGOOD	R/W	0h	HSCLK GOOD 0h = 0 1h = 1
6	SYSPLLGOOD	R/W	0h	SYSPLL GOOD 0h = 0 1h = 1
5	HFCLKGOOD	R/W	0h	HFCLK GOOD 0h = 0 1h = 1
4	LFXTGOOD	R/W	0h	LFXT GOOD 0h = 0 1h = 1
3	SRAMSEC	R/W	0h	SRAM Single Error Correct 0h = 0 1h = 1
2	FLASHSEC	R/W	0h	Flash Single Error Correct 0h = 0 1h = 1
1	ANACLKERR	R/W	0h	Analog Clocking Consistency Error 0h = 0 1h = 1
0	LFOSCGOOD	R/W	0h	Enable or disable the LFOSCGOOD interrupt. LFOSCGOOD indicates that the LFOSC has started successfully. 0h = Interrupt disabled 1h = Interrupt enabled

2.8.3 RIS Register (Offset = 1030h) [Reset = 00000000h]

RIS is shown in Table 2-23.

Return to the Summary Table.

SYSCTL raw interrupt status

Table 2-23. RIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7	HSCLKGOOD	R	0h	HSCLK GOOD 0h = 0 1h = 1
6	SYSPLLGOOD	R	0h	SYSPLL GOOD 0h = 0 1h = 1
5	HFCLKGOOD	R	0h	HFCLK GOOD 0h = 0 1h = 1
4	LFXTGOOD	R	0h	LFXT GOOD 0h = 0 1h = 1
3	SRAMSEC	R	0h	SRAM Single Error Correct 0h = 0 1h = 1
2	FLASHSEC	R	0h	Flash Single Error Correct 0h = 0 1h = 1
1	ANACLKERR	R	0h	Analog Clocking Consistency Error 0h = 0 1h = 1
0	LFOSCGOOD	R	0h	Raw status of the LFOSCGOOD interrupt. 0h = No interrupt pending 1h = Interrupt pending

2.8.4 MIS Register (Offset = 1038h) [Reset = 00000000h]

MIS is shown in Table 2-24.

Return to the Summary Table.

SYSCTL masked interrupt status

Table 2-24. MIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7	HSCLKGOOD	R	0h	HSCLK GOOD 0h = 0 1h = 1
6	SYSPLLGOOD	R	0h	SYSPLL GOOD 0h = 0 1h = 1
5	HFCLKGOOD	R	0h	HFCLK GOOD 0h = 0 1h = 1
4	LFXTGOOD	R	0h	LFXT GOOD 0h = 0 1h = 1
3	SRAMSEC	R	0h	SRAM Single Error Correct 0h = 0 1h = 1
2	FLASHSEC	R	0h	Flash Single Error Correct 0h = 0 1h = 1
1	ANACLKERR	R	0h	Analog Clocking Consistency Error 0h = 0 1h = 1
0	LFOSCGOOD	R	0h	Masked status of the LFOSCGOOD interrupt. 0h = No interrupt pending 1h = Interrupt pending

2.8.5 ISET Register (Offset = 1040h) [Reset = 00000000h]

ISET is shown in Table 2-25.

Return to the Summary Table.

SYSCTL interrupt set

Table 2-25. ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	W	0h	
7	HSCLKGOOD	W1S	0h	HSCLK GOOD 0h = 0 1h = 1
6	SYSPLLGOOD	W1S	0h	SYSPLL GOOD 0h = 0 1h = 1
5	HFCLKGOOD	W1S	0h	HFCLK GOOD 0h = 0 1h = 1
4	LFXTGOOD	W1S	0h	LFXT GOOD 0h = 0 1h = 1
3	SRAMSEC	W1S	0h	SRAM Single Error Correct 0h = 0 1h = 1
2	FLASHSEC	W1S	0h	Flash Single Error Correct 0h = 0 1h = 1
1	ANACLKERR	W1S	0h	Analog Clocking Consistency Error 0h = 0 1h = 1
0	LFOSCGOOD	W1S	0h	Set the LFOSCGOOD interrupt. 0h = Writing 0h hs no effect 1h = Set interrupt

2.8.6 ICLR Register (Offset = 1048h) [Reset = 00000000h]

ICLR is shown in Table 2-26.

Return to the Summary Table.

SYSCTL interrupt clear

Table 2-26. ICLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	W	0h	
7	HSCLKGOOD	W1C	0h	HSCLK GOOD 0h = 0 1h = 1
6	SYSPLLGOOD	W1C	0h	SYSPLL GOOD 0h = 0 1h = 1
5	HFCLKGOOD	W1C	0h	HFCLK GOOD 0h = 0 1h = 1
4	LFXTGOOD	W1C	0h	LFXT GOOD 0h = 0 1h = 1
3	SRAMSEC	W1C	0h	SRAM Single Error Correct 0h = 0 1h = 1
2	FLASHSEC	W1C	0h	Flash Single Error Correct 0h = 0 1h = 1
1	ANACLKERR	W1C	0h	Analog Clocking Consistency Error 0h = 0 1h = 1
0	LFOSCGOOD	W1C	0h	Clear the LFOSCGOOD interrupt. 0h = Writing 0h has no effect 1h = Clear interrupt

2.8.7 NMIIIDX Register (Offset = 1050h) [Reset = 00000000h]

NMIIIDX is shown in Table 2-27.

Return to the Summary Table.

NMI interrupt index

Table 2-27. NMIIIDX Register 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. 0h = No NMI pending 1h = BOR Threshold NMI pending 2h = 2 3h = 3 4h = 4 5h = 5 6h = 6 7h = 7 8h = 8

2.8.8 NMIRIS Register (Offset = 1060h) [Reset = 00000000h]

NMIRIS is shown in Table 2-28.

Return to the Summary Table.

NMI raw interrupt status

Table 2-28. NMIRIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7	VBATUP	R	0h	VBAT Power On 0h = 0 1h = 1
6	VBATDN	R	0h	VBAT Power Off 0h = 0 1h = 1
5	SRAMDED	R	0h	SRAM Double Error Detect 0h = 0 1h = 1
4	FLASHDED	R	0h	Flash Double Error Detect 0h = 0 1h = 1
3	LFCLKFAIL	R	0h	LFXT-EXLF Monitor Fail 0h = 0 1h = 1
2	SECURITY	R	0h	Security Fault 0h = 0 1h = 1
1	WWDT0	R	0h	Watch Dog 0 Fault 0h = 0 1h = 1
0	BORLVL	R	0h	Raw status of the BORLVL NMI 0h = No interrupt pending 1h = Interrupt pending

2.8.9 NMIISET Register (Offset = 1070h) [Reset = 00000000h]

NMIISET is shown in Table 2-29.

Return to the Summary Table.

NMI interrupt set

Table 2-29. NMIISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	W	0h	
7	VBATUP	W1S	0h	VBAT Power On 0h = 0 1h = 1
6	VBATDN	W1S	0h	VBAT Power Off 0h = 0 1h = 1
5	SRAMDED	W1S	0h	SRAM Double Error Detect 0h = 0 1h = 1
4	FLASHDED	W1S	0h	Flash Double Error Detect 0h = 0 1h = 1
3	LFCLKFAIL	W1S	0h	LFXT-EXLF Monitor Fail 0h = 0 1h = 1
2	SECURITY	W1S	0h	Security Fault 0h = 0 1h = 1
1	WWDT0	W1S	0h	Watch Dog 0 Fault 0h = 0 1h = 1
0	BORLVL	W1S	0h	Set the BORLVL NMI 0h = Writing 0h hs no effect 1h = Set interrupt

2.8.10 NMIICLR Register (Offset = 1078h) [Reset = 00000000h]

NMIICLR is shown in Table 2-30.

Return to the Summary Table.

NMI interrupt clear

Table 2-30. NMIICLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	W	0h	
7	VBATUP	W1C	0h	VBAT Power On 0h = 0 1h = 1
6	VBATDN	W1C	0h	VBAT Power Off 0h = 0 1h = 1
5	SRAMDED	W1C	0h	SRAM Double Error Detect 0h = 0 1h = 1
4	FLASHDED	W1C	0h	Flash Double Error Detect 0h = 0 1h = 1
3	LFCLKFAIL	W1C	0h	LFXT-EXLF Monitor Fail 0h = 0 1h = 1
2	SECURITY	W1C	0h	Security Fault 0h = 0 1h = 1
1	WWDT0	W1C	0h	Watch Dog 0 Fault 0h = 0 1h = 1
0	BORLVL	W1C	0h	Clr the BORLVL NMI 0h = Writing 0h hs no effect 1h = Clear interrupt

2.8.11 SYSOSCCFG Register (Offset = 1100h) [Reset = 0002XXXXh]

SYSOSCCFG is shown in Table 2-31.

Return to the Summary Table.

SYSOSC configuration

Table 2-31. SYSOSCCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	0h	
17	FASTCPUEVENT	R/W	1h	if disabled CPU will not wakeup and continue in STANDBY 0h = An interrupt to the CPU will not assert a fast clock request 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. 0h = Asynchronous fast clock requests are controlled by the requesting peripheral 1h = All asynchronous fast clock requests are blocked
15-2	RESERVED	R/W	0h	
1-0	FREQ	R/W	0h	Target operating frequency for the system oscillator (SYSOSC) 0h = Base frequency (32MHz) 1h = Low frequency (4MHz)

2.8.12 MCLKCFG Register (Offset = 1104h) [Reset = 07XXX2X0h]

MCLKCFG is shown in Table 2-32.

Return to the Summary Table.

Main clock (MCLK) configuration

Table 2-32. MCLKCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R/W	0h	
26-24	MCLKDIVCFG	R/W	7h	MCLK Divider Configuration bits [1:0] are defined as MCLK4 is Bypass, MCLK2 is Bypass 0h = MCLK=No Divide MCLK2=No Divide MCLK4=No Divide 1h = MCLK=No Divide MCLK2=No Divide MCLK4=Divide MCLK by 2 3h = MCLK=No Divide MCLK2=No Divide MCLK4=Divide MCLK by 4 5h = MCLK=No Divide MCLK2=Divide MCLK by 2 MCLK4=Divide MCLK by 2 7h = MCLK=No Divide MCLK2=Divide MCLK by 2 MCLK4=Divide MCLK by 4
23	RESERVED	R/W	0h	
22	MCLKDEADCHK	R/W	0h	MCLKDEADCHK enables or disables the continuous MCLK dead check monitor. LFCLK must be running before MCLKDEADCHK is enabled. 0h = The MCLK dead check monitor is disabled 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. 0h = ULPCLK/LFCLK runs to all PD0 peripherals in STANDBY mode 1h = ULPCLK/LFCLK is disabled to all peripherals in STANDBY mode except TIMG0 and TIMG1
20-17	RESERVED	R/W	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. 0h = MCLK will not use the high speed clock (HSCLK) 1h = MCLK will use the high speed clock (HSCLK) in RUN and SLEEP mode
15-13	RESERVED	R/W	0h	
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). 0h = The 4MHz rate MFCLK to peripherals is enabled 1h = The 4MHz rate MFCLK to peripherals is enabled.
11-0	RESERVED	R/W	0h	

2.8.13 HSCLKEN Register (Offset = 1108h) [Reset = 0000XXXXh]

HSCLKEN is shown in Table 2-33.

Return to the Summary Table.

High-speed clock (HSCLK) source enable/disable

Table 2-33. HSCLKEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R/W	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. 0h = Use HFXT as the HFCLK source 1h = Use the HFCLK_IN digital clock input as the HFCLK source
15-9	RESERVED	R/W	0h	
8	SYSPLLEN	R/W	0h	SYSPLLEN enables or disables the system phase-lock loop (SYSPLL). 0h = Disable the SYSPLL 1h = Enable the SYSPLL
7-1	RESERVED	R/W	0h	
0	HFXTEN	R/W	0h	HFXTEN enables or disables the high frequency crystal oscillator (HFXT). 0h = Disable the HFXT 1h = Enable the HFXT

2.8.14 HSCLKCFG Register (Offset = 110Ch) [Reset = 00000000h]

HSCLKCFG is shown in Table 2-34.

Return to the Summary Table.

High-speed clock (HSCLK) source selection

Table 2-34. HSCLKCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	HSCLKSEL	R/W	0h	HSCLKSEL selects the HSCLK source (SYSPLL or HFCLK). 0h = HSCLK is sourced from the SYSPLL 1h = HSCLK is sourced from the HFCLK

2.8.15 HFCLKCLKCFG Register (Offset = 1110h) [Reset = 1XXXXX00h]

HFCLKCLKCFG is shown in Table 2-35.

Return to the Summary Table.

High-frequency clock (HFCLK) configuration

Table 2-35. HFCLKCLKCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	HFCLKFLTCHK	R/W	1h	HFCLKFLTCHK enables or disables the HFCLK startup monitor. 0h = HFCLK startup is not checked 1h = HFCLK startup is checked
27-14	RESERVED	R/W	0h	
13-12	HFXTRSEL	R/W	0h	HFXT Range Select 0h = 4MHz <= HFXT frequency <= 8MHz 1h = 8MHz < HFXT frequency <= 16MHz 2h = 16MHz < HFXT frequency <= 32MHz 3h = 32MHz < HFXT frequency <= 48MHz
11-8	RESERVED	R/W	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. 0h = Minimum startup time (approximatly zero) FFh = Maximum startup time (approximatly 16.32ms)

2.8.16 LFCLKCFG Register (Offset = 1114h) [Reset = 000000XXh]

LFCLKCFG is shown in Table 2-36.

Return to the Summary Table.

Low frequency crystal oscillator (LFXT) configuration

Table 2-36. LFCLKCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R/W	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. 0h = LFXT low capacitance mode is disabled 1h = LFXT low capacitance mode is enabled
7-5	RESERVED	R/W	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. 0h = Clock monitor is disabled 1h = Clock monitor is enabled
3-2	RESERVED	R/W	0h	
1-0	XT1DRIVE	R/W	3h	XT1DRIVE selects the low frequency crystal oscillator (LFXT) drive strength. 0h = Lowest drive and current 1h = Lower drive and current 2h = Higher drive and current 3h = Highest drive and current

2.8.17 SYSPLLCFG0 Register (Offset = 1120h) [Reset = 000000XXh]

SYSPLLCFG0 is shown in Table 2-37.

Return to the Summary Table.

SYSPLL reference and output configuration

Table 2-37. SYSPLLCFG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R/W	0h	
19-16	RDIVCLK0	R/W	0h	RDIVCLK0 sets the final divider for the SYSPLLCLK0 output. 0h = SYSPLLCLK0 is divided by 2 1h = SYSPLLCLK0 is divided by 4 2h = SYSPLLCLK0 is divided by 6 3h = SYSPLLCLK0 is divided by 8 4h = SYSPLLCLK0 is divided by 10 5h = SYSPLLCLK0 is divided by 12 6h = SYSPLLCLK0 is divided by 14 7h = SYSPLLCLK0 is divided by 16 8h = SYSPLLCLK0 is divided by 18 9h = SYSPLLCLK0 is divided by 20 Ah = SYSPLLCLK0 is divided by 22 Bh = SYSPLLCLK0 is divided by 24 Ch = SYSPLLCLK0 is divided by 26 Dh = SYSPLLCLK0 is divided by 28 Eh = SYSPLLCLK0 is divided by 30 Fh = SYSPLLCLK0 is divided by 32
15-12	RDIVCLK1	R/W	0h	RDIVCLK1 sets the final divider for the SYSPLLCLK1 output. 0h = SYSPLLCLK1 is divided by 2 1h = SYSPLLCLK1 is divided by 4 2h = SYSPLLCLK1 is divided by 6 3h = SYSPLLCLK1 is divided by 8 4h = SYSPLLCLK1 is divided by 10 5h = SYSPLLCLK1 is divided by 12 6h = SYSPLLCLK1 is divided by 14 7h = SYSPLLCLK1 is divided by 16 8h = SYSPLLCLK1 is divided by 18 9h = SYSPLLCLK1 is divided by 20 Ah = SYSPLLCLK1 is divided by 22 Bh = SYSPLLCLK1 is divided by 24 Ch = SYSPLLCLK1 is divided by 26 Dh = SYSPLLCLK1 is divided by 28 Eh = SYSPLLCLK1 is divided by 30 Fh = SYSPLLCLK1 is divided by 32
11-8	RDIVCLK2X	R/W	0h	RDIVCLK2X sets the final divider for the SYSPLLCLK2X output. 0h = SYSPLLCLK1 is divided by 1 1h = SYSPLLCLK1 is divided by 2 2h = SYSPLLCLK1 is divided by 3 3h = SYSPLLCLK1 is divided by 4 4h = SYSPLLCLK1 is divided by 5 5h = SYSPLLCLK1 is divided by 6 6h = SYSPLLCLK1 is divided by 7 7h = SYSPLLCLK1 is divided by 8 8h = SYSPLLCLK1 is divided by 9 9h = SYSPLLCLK1 is divided by 10 Ah = SYSPLLCLK1 is divided by 11 Bh = SYSPLLCLK1 is divided by 12 Ch = SYSPLLCLK1 is divided by 13 Dh = SYSPLLCLK1 is divided by 14 Eh = SYSPLLCLK1 is divided by 15 Fh = SYSPLLCLK1 is divided by 16
7	RESERVED	R/W	0h	

Table 2-37. SYSPLLCFG0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	ENABLECLK2X	R/W	0h	ENABLECLK2X enables or disables the SYSPLLCLK2X output. 0h = SYSPLLCLK2X is disabled 1h = SYSPLLCLK2X is enabled
5	ENABLECLK1	R/W	0h	ENABLECLK1 enables or disables the SYSPLLCLK1 output. 0h = SYSPLLCLK1 is disabled 1h = SYSPLLCLK1 is enabled
4	ENABLECLK0	R/W	0h	ENABLECLK0 enables or disables the SYSPLLCLK0 output. 0h = SYSPLLCLK0 is disabled 1h = SYSPLLCLK0 is enabled
3-2	RESERVED	R/W	0h	
1	MCLK2XVCO	R/W	0h	MCLK2XVCO selects the SYSPLL output which is sent to the HSCLK mux for use by MCLK. 0h = The SYSPLLCLK0 output is sent to the HSCLK mux 1h = The SYSPLLCLK2X output is sent to the HSCLK mux
0	SYSPLLREF	R/W	0h	SYSPLLREF selects the system PLL (SYSPLL) reference clock source. 0h = SYSPLL reference is SYSOSC 1h = SYSPLL reference is HFCLK

2.8.18 SYSPLLCFG1 Register (Offset = 1124h) [Reset = 000023XXh]

SYSPLLCFG1 is shown in Table 2-38.

Return to the Summary Table.

SYSPLL reference and feedback divider

Table 2-38. SYSPLLCFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R/W	0h	
14-8	QDIV	R/W	23h	QDIV selects the SYSPLL feedback path divider. 0h = Divide-by-one is not a valid QDIV option 1h = Feedback path is divided by 2 7Eh = Feedback path is divided by 127 (0x7E)
7-2	RESERVED	R/W	0h	
1-0	PDIV	R/W	0h	PDIV selects the SYSPLL reference clock prescale divider. 0h = SYSPLLREF is not divided 1h = SYSPLLREF is divided by 2 2h = SYSPLLREF is divided by 4 3h = SYSPLLREF is divided by 8

2.8.19 SYSPLLPARAM0 Register (Offset = 1128h) [Reset = X1XAX0X0h]

SYSPLLPARAM0 is shown in Table 2-39.

Return to the Summary Table.

SYSPLL PARAM0 (load from FACTORY region)

Table 2-39. SYSPLLPARAM0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	CAPBOVERRIDE	R/W	1h	CAPBOVERRIDE controls the override for Cap B 0h = Cap B override disabled 1h = Cap B override enabled
30-29	RESERVED	R/W	0h	
28-24	CAPBVAL	R/W	1h	Override value for Cap B
23-22	RESERVED	R/W	0h	
21-16	CPCURRENT	R/W	Ah	Charge pump current
15	RESERVED	R/W	0h	
14-8	STARTTIMELP	R/W	0h	Startup time from low power mode exit to locked clock, in 1us resolution
7	RESERVED	R/W	0h	
6-0	STARTTIME	R/W	0h	Startup time from enable to locked clock, in 1us resolution

2.8.20 SYSPLLPARAM1 Register (Offset = 112Ch) [Reset = 0FXX01XFh]

SYSPLLPARAM1 is shown in Table 2-40.

Return to the Summary Table.

SYSPLL PARAM1 (load from FACTORY region)

Table 2-40. SYSPLLPARAM1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	LPFRESC	R/W	Fh	Loop filter Res C
23-18	RESERVED	R/W	0h	
17-8	LPFRESA	R/W	1h	Loop filter Res A
7-5	RESERVED	R/W	0h	
4-0	LPFCAPA	R/W	Fh	Loop filter Cap A

2.8.21 SYSPLLPARAM2 Register (Offset = 1130h) [Reset = 0000000Xh]

SYSPLLPARAM2 is shown in Table 2-41.

Return to the Summary Table.

SYSPLL PARAM2 (load from FACTORY region)

Table 2-41. SYSPLLPARAM2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	RNGFIXVCOIBIASCFG	R/W	1h	0 value for Temperature Compensation R addition
2	RESERVED	R/W	0h	
1-0	LPFCAPC	R/W	0h	Loop filter Cap C

2.8.22 SYSPLLLDOCTL Register (Offset = 1134h) [Reset = 00000000h]

SYSPLLLDOCTL is shown in Table 2-42.

Return to the Summary Table.

SYSPLL LDO CTL (load from FACTORY region)

Table 2-42. SYSPLLLDOCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	
15-0	LDOCTLLOWV	R/W	0h	LDO Configurability

2.8.23 SYSPLLLDOPROG Register (Offset = 1138h) [Reset = 00000004h]

SYSPLLLDOPROG is shown in Table 2-43.

Return to the Summary Table.

SYSPLL LDO VOUT PROG (load from FACTORY region)

Table 2-43. SYSPLLLDOPROG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	LDOVOUTPROGLOWV	R/W	4h	HPLL LDO Vout Prog

2.8.24 GENCLKEN Register (Offset = 113Ch) [Reset = 000000XXh]

GENCLKEN is shown in Table 2-44.

Return to the Summary Table.

General clock enable control

Table 2-44. GENCLKEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R/W	0h	
18	I2SPLLCLKDIVEN	R/W	0h	I2SPLLCLKDIVEN enables or disables the divider function of the PLL Source to I2S. 0h = CLock divider is disabled (passthrough, I2SPLLCLKDIV is not applied) 1h = Clock divider is enabled (I2SPLLCLKDIV is applied)
17-16	I2SPLLCLKDIVCFG	R/W	0h	I2SPLLCLKDIVCFG selects the divider value for the divider for the PLL Source to CAN. 0h = CLK_OUT source is divided by 4 1h = CLK_OUT source is divided by 8 2h = CLK_OUT source is divided by 12 3h = CLK_OUT source is divided by 16
15	CANEXTDIVEN	R/W	0h	CANEXTDIVEN enables or disables the divider function of the PLL Source to CAN. 0h = CLock divider is disabled (passthrough, EXTDIVCAN is not applied) 1h = Clock divider is enabled (EXTDIVCAN is applied)
14-12	EXTDIVCAN	R/W	0h	EXTDIVCAN selects the divider value for the divider for the PLL Source to CAN. 0h = CLK_OUT source is divided by 2 1h = CLK_OUT source is divided by 4 2h = CLK_OUT source is divided by 6 3h = CLK_OUT source is divided by 8 4h = CLK_OUT source is divided by 10 5h = CLK_OUT source is divided by 12 6h = CLK_OUT source is divided by 14 7h = CLK_OUT source is divided by 16
11	MCLKEXTDIVEN	R/W	0h	MCLKEXTDIVEN enables or disables the divider function of the PLL Source to MCLK. 0h = CLock divider is disabled (passthrough, EXTDIVMCLK is not applied) 1h = Clock divider is enabled (EXTDIVMCLK is applied)
10-8	EXTDIVMCLK	R/W	0h	EXTDIVMCLK selects the divider value for the divider for the PLL Source MCLK. 0h = CLK_OUT source is divided by 2 1h = CLK_OUT source is divided by 4 2h = CLK_OUT source is divided by 6 3h = CLK_OUT source is divided by 8 4h = CLK_OUT source is divided by 10 5h = CLK_OUT source is divided by 12 6h = CLK_OUT source is divided by 14 7h = CLK_OUT source is divided by 16
7-1	RESERVED	R/W	0h	
0	EXCLKEN	R/W	0h	EXCLKEN enables the CLK_OUT external clock output block. 0h = CLK_OUT block is disabled 1h = CLK_OUT block is enabled

2.8.25 GENCLKCFG Register (Offset = 1140h) [Reset = 00000X0Xh]

GENCLKCFG is shown in Table 2-45.

Return to the Summary Table.

General clock configuration

Table 2-45. GENCLKCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	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. 0h = VBOOST is enabled on request from a COMP, GPAMP, or OPA 1h = VBOOST is enabled when the device is in RUN or SLEEP mode, or when a COMP/GPAMP/OPA is enabled 2h = VBOOST is always enabled
21	FCCLVLTRIG	R/W	0h	FCCLVLTRIG selects the frequency clock counter (FCC) trigger mode. 0h = Rising edge to rising edge triggered 1h = Level triggered
20	FCCTRIGSRC	R/W	0h	FCCTRIGSRC selects the frequency clock counter (FCC) trigger source. 0h = FCC trigger is the external pin 1h = FCC trigger is the LFCLK
19-16	FCCSELCLK	R/W	0h	FCCSELCLK selectes the frequency clock counter (FCC) clock source. 0h = FCC clock is MCLK/4 1h = FCC clock is SYSOSC 2h = FCC clock is HFCLK 3h = FCC clock is the CLK_OUT selection 4h = FCC clock is SYSPLLCLK0 5h = FCC clock is SYSPLLCLK1 6h = FCC clock is SYSPLLCLK2X 7h = FCC clock is the FCCIN external input
15-9	RESERVED	R/W	0h	
8	CANCLKSRC	R/W	0h	CANCLKSRC selects the CANCLK source. 0h = CANCLK source is HFCLK 1h = CANCLK source is SYSPLLOUT0 or SYSPLLOUT2x
7	EXCLKDIVEN	R/W	0h	EXCLKDIVEN enables or disables the divider function of the CLK_OUT external clock output block. 0h = CLock divider is disabled (passthrough, EXCLKDIVVAL is not applied) 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. 0h = CLK_OUT source is divided by 2 1h = CLK_OUT source is divided by 4 2h = CLK_OUT source is divided by 6 3h = CLK_OUT source is divided by 8 4h = CLK_OUT source is divided by 10 5h = CLK_OUT source is divided by 12 6h = CLK_OUT source is divided by 14 7h = CLK_OUT source is divided by 16
3	RESERVED	R/W	0h	

Table 2-45. GENCLKCFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2-0	EXCLKSRC	R/W	0h	EXCLKSRC selects the source for the CLK_OUT external clock output block. ULPCLK requires the CLK_OUT divider (EXCLKDIVEN) to be enabled 0h = CLK_OUT is SYSOSC 1h = CLK_OUT is ULPCLK (EXCLKDIVEN must be enabled) 2h = CLK_OUT is LFCLK 3h = Reserved 4h = CLK_OUT is HFCLK 5h = CLK_OUT is SYSPLLCLK1 (SYSPLLCLK1 must be <=48MHz)

2.8.26 PMODECFG Register (Offset = 1144h) [Reset = 00000000h]

PMODECFG is shown in Table 2-46.

Return to the Summary Table.

Power mode configuration

Table 2-46. PMODECFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	0h	
1-0	DSLEEP	R/W	0h	DSLEEP selects the operating mode to enter upon a DEEPSLEEP request from the CPU. 0h = STOP mode is entered 1h = STANDBY mode is entered 2h = SHUTDOWN mode is entered 3h = Reserved

2.8.27 MLDOLPENCFG Register (Offset = 1148h) [Reset = 000000XXh]

MLDOLPENCFG is shown in Table 2-47.

Return to the Summary Table.

LDO Configuration Control

Table 2-47. MLDOLPENCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R/W	0h	
8	SVBPRETDIS	R/W	0h	Control to disable the disconnect of core LDO from retention for STOP and STANDBY 0h = Disconnect the core LDO from retention domain during STOP and STANDBY 1h = Do Not Disconnect the core LDO from retention domain during STOP and STANDBY
7-1	RESERVED	R/W	0h	
0	CVLODIS	R/W	0h	Control to disable lowering the core voltage for STOP and STANDBY 0h = Lower Core Voltage for STOP and STANDBY mode 1h = Do Not Lower Core Voltage for STOP and STANDBY mode to provide faster wakeup

2.8.28 FCC Register (Offset = 1150h) [Reset = 00000000h]

FCC is shown in Table 2-48.

Return to the Summary Table.

Frequency clock counter (FCC) count

Table 2-48. FCC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	DATA	R	0h	Frequency clock counter (FCC) count value.

2.8.29 PMULDOSPARECTL Register (Offset = 1154h) [Reset = 00000000h]

PMULDOSPARECTL is shown in Table 2-49.

Return to the Summary Table.

LDO Spare Control

Table 2-49. PMULDOSPARECTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3-0	sparectrl	R/W	0h	Spare PMU LDO control for M33

2.8.30 SYSCTL_ECO_REG1 Register (Offset = 1158h) [Reset = 00000000h]

SYSCTL_ECO_REG1 is shown in Table 2-50.

Return to the Summary Table.

Sysctl ECO Reg 1

Table 2-50. SYSCTL_ECO_REG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ecoreg	R/W	0h	ECO Reg 1 for M33

2.8.31 SYSCTL_ECO_REG2 Register (Offset = 115Ch) [Reset = 00000000h]

SYSCTL_ECO_REG2 is shown in Table 2-51.

Return to the Summary Table.

Sysctl ECO Reg 2

Table 2-51. SYSCTL_ECO_REG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ecoreg	R/W	0h	ECO Reg 2 for M33

2.8.32 SYSTEMCFG Register (Offset = 1180h) [Reset = 00XXXXXXh]

SYSTEMCFG is shown in Table 2-52.

Return to the Summary Table.

System configuration

Table 2-52. SYSTEMCFG Register 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 1Bh = Issue write
23-9	RESERVED	R/W	0h	
8	SUPERCAPEN	R/W	0h	SUPERCAP specifies whether the battery backup system can be powered by a SUPERCAP 0h = SUPERCAP Function is not enabled 1h = SUPERCAP Function is not enabled
7-3	RESERVED	R/W	0h	
2	FLASHECCRSTDIS	R/W	1h	FLASHECCRSTDIS specifies whether a flash ECC double error detect (DED) will trigger a SYSRST or an NMI. 0h = Flash ECC DED will trigger a SYSRST 1h = Flash ECC DED will trigger a NMI
1	RESERVED	R/W	0h	
0	WWDTLP0RSTDIS	R/W	0h	WWDTLP0RSTDIS specifies whether a WWDT Error Event will trigger a BOOTRST or an NMI. 0h = WWDTLP0 Error Event will trigger a BOOTRST 1h = WWDTLP0 Error Event will trigger an NMI

2.8.33 SRAMCFG Register (Offset = 1184h) [Reset = 00XXXXXXh]

SRAMCFG is shown in Table 2-53.

Return to the Summary Table.

System SRAM configuration

Table 2-53. SRAMCFG Register 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 B5h = Issue write
23-20	RESERVED	R/W	0h	
19	BANKINITDIS3	R/W	0h	SRAM BANK3 Initialization 0h = SRAM BANK3 will Initialize when transitioning from OFF to ON 1h = SRAM BANK3 will NOT Initialize when transitioning from OFF to ON
18	BANKINITDIS2	R/W	0h	SRAM BANK2 Initialization 0h = SRAM BANK2 will Initialize when transitioning from OFF to ON 1h = SRAM BANK2 will NOT Initialize when transitioning from OFF to ON
17	BANKINITDIS1	R/W	0h	SRAM BANK1 Initialization 0h = SRAM BANK1 will Initialize when transitioning from OFF to ON 1h = SRAM BANK1 will NOT Initialize when transitioning from OFF to ON
16-0	RESERVED	R/W	0h	

2.8.34 WRITELOCK Register (Offset = 1200h) [Reset = 00000000h]

WRITELOCK is shown in Table 2-54.

Return to the Summary Table.

SYSCTL register write lockout

Table 2-54. WRITELOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	ACTIVE	R/W	0h	ACTIVE controls whether critical SYSCTL registers are write protected or not. 0h = Allow writes to lockable registers 1h = Disallow writes to lockable registers

2.8.35 CLKSTATUS Register (Offset = 1204h) [Reset = XXXXX0XXh]

CLKSTATUS is shown in Table 2-55.

Return to the Summary Table.

Clock module (CKM) status

Table 2-55. CLKSTATUS Register 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. 0h = No analog clock errors detected 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. 0h = writes to SYSPLLCFG0/1 and SYSPLLPARAM0/1 are allowed 1h = writes to SYSPLLCFG0/1 and SYSPLLPARAM0/1 are blocked
28	HFCLKBLKUPD	R	0h	HFCLKBLKUPD indicates when writes to the HFCLKCLKCFG register are blocked. 0h = Writes to HFCLKCLKCFG are allowed 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. 0h = FCC capture is not done 1h = FCC capture is done
24	FCLMODE	R	0h	FCLMODE indicates if the SYSOSC frequency correction loop (FCL) is enabled. 0h = SYSOSC FCL is disabled 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. 0h = No LFCLK fault detected 1h = LFCLK stuck fault detected
22	RESERVED	R	0h	
21	HSCLKGOOD	R	0h	HSCLKGOOD is set by hardware if the selected clock source for HSCLK started successfully. 0h = The HSCLK source did not start correctly 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. 0h = The HSCLK source was not started or started correctly 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. 0h = MCLK is not sourced from LFCLK 1h = MCLK is sourced from LFCLK
16	CURHSCLKSEL	R	0h	CURHSCLKSEL indicates the current clock source for HSCLK. 0h = HSCLK is currently sourced from the SYSPLL 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. 0h = SYSPLL started correctly and is enabled 1h = SYSPLL is disabled or was dead startup

Table 2-55. CLKSTATUS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
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. 0h = HFCLK started correctly and is enabled 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. 0h = SYSPLL, HFCLK, or both were started correctly and remain enabled 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. 0h = LFOSC is not ready 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. 0h = LFXT did not start correctly 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. 0h = SYSPLL did not start correctly 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. 0h = HFCLK did not start correctly 1h = HFCLK started correctly
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. 0h = LFCLK is sourced from the internal LFOSC 1h = LFCLK is sourced from the LFXT (crystal) 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). 0h = MCLK is not sourced from HSCLK 1h = MCLK is sourced from HSCLK
3-2	RESERVED	R	0h	
1-0	SYSOSCFREQ	R	0h	SYSOSCFREQ indicates the current SYSOSC operating frequency. 0h = SYSOSC is at base frequency (32MHz) 1h = SYSOSC is at low frequency (4MHz)

2.8.36 SYSSTATUS Register (Offset = 1208h) [Reset = XXX00X00h]

SYSSTATUS is shown in Table 2-56.

Return to the Summary Table.

System status information

Table 2-56. SYSSTATUS Register 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.
29-20	RESERVED	R	0h	
19	SRAMBANK3READY	R	0h	SRAM BANK3 READY STATE 0h = SRAM BANK3 is NOT READY for access 1h = SRAM BANK3 is READY for access
18	SRAMBANK2READY	R	0h	SRAM BANK2 READY STATE 0h = SRAM BANK2 is NOT READY for access 1h = SRAM BANK2 is READY for access
17	SRAMBANK1READY	R	0h	SRAM BANK1 READY STATE 0h = SRAM BANK1 is NOT READY for access 1h = SRAM BANK1 is READY for access
16	PKAREADY	R	0h	PKAREADY indicates when the PKA peripheral is ready. 0h = PKA is not ready 1h = PKA is ready
15	RESERVED	R	0h	
14	SHDNIOLOCK	R	0h	SHDNIOLOCK indicates when IO is locked due to SHUTDOWN 0h = IO IS NOT Locked due to SHUTDOWN 1h = IO IS Locked due to SHUTDOWN
13	SWDCFGDIS	R	0h	SWDCFGDIS indicates when user has disabled the use of SWD Port 0h = SWD Port Enabled 1h = SWD Port Disabled
12	EXTRSTPINDIS	R	0h	EXTRSTPINDIS indicates when user has disabled the use of external reset pin 0h = External Reset Pin Enabled 1h = External Reset Pin Disabled
11-10	RESERVED	R	0h	
9	MCAN1READY	R	0h	MCAN1READY indicates when the MCAN1 peripheral is ready. 0h = MCAN1 is not ready 1h = MCAN1 is ready
8	MCAN0READY	R	0h	MCAN0READY indicates when the MCAN0 peripheral is ready. 0h = MCAN0 is not ready 1h = MCAN0 is ready
7	VBATGOOD	R	0h	VBATGOOD is set by hardware when the VBAT Power Domain is valid. 0h = VBAT Power Domain is not valid 1h = VBAT Power Domain is valid
6	PMUIREFGOOD	R	0h	PMUIREFGOOD is set by hardware when the PMU current reference is ready. 0h = IREF is not ready 1h = IREF is ready
5	ANACPUMPGOOD	R	0h	ANACPUMPGOOD is set by hardware when the VBOOST analog mux charge pump is ready. 0h = VBOOST is not ready 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. 0h = No BOR violation occured 1h = A BOR violation occured and the BOR threshold was switched to BOR0

Table 2-56. SYSSTATUS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	BORCURTHRESHOLD	R	0h	BORCURTHRESHOLD indicates the active brown-out reset supply monitor configuration. 0h = Default minimum threshold; a BOR0- violation triggers a BOR 1h = A BOR1- violation generates a BORLVL interrupt 2h = A BOR2- violation generates a BORLVL interrupt 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). 0h = No flash ECC single bit error detected 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). 0h = No flash ECC double bit error detected 1h = Flash ECC double bit error detected

2.8.37 RSTCAUSE Register (Offset = 1220h) [Reset = 00000000h]

RSTCAUSE is shown in Table 2-57.

Return to the Summary Table.

Reset cause

Table 2-57. RSTCAUSE Register 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. 0h = No reset since last read 1h = POR- violation, SHUTDNSTOREx or PMU trim parity fault 2h = NRST triggered POR (>1s hold) 3h = Software triggered POR 4h = BOR0- violation 5h = SHUTDOWN mode exit 8h = Non-PMU trim parity fault 9h = Fatal clock failure Ch = NRST triggered BOOTRST (<1s hold) Dh = Software triggered BOOTRST Eh = WWDT0 violation 10h = BSL exit 11h = BSL entry 14h = Flash uncorrectable ECC error 15h = CPULOCK violation 1Ah = Debug triggered SYSRST 1Bh = Software triggered SYSRST 1Ch = Debug triggered CPURST 1Dh = Software triggered CPURST

2.8.38 RESETLEVEL Register (Offset = 1300h) [Reset = 00000000h]

RESETLEVEL is shown in Table 2-58.

Return to the Summary Table.

Reset level for application-triggered reset command

Table 2-58. RESETLEVEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	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. 0h = Issue a SYSRST (CPU plus peripherals only) 1h = Issue a BOOTRST (CPU, peripherals, and boot configuration routine) 2h = Issue a SYSRST and enter the boot strap loader (BSL) 3h = Issue a power-on reset (POR) 4h = Issue a SYSRST and exit the boot strap loader (BSL)

2.8.39 RESETCMD Register (Offset = 1304h) [Reset = 00XXXXXXh]

RESETCMD is shown in Table 2-59.

Return to the Summary Table.

Execute an application-triggered reset command

Table 2-59. RESETCMD Register 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. E4h = Issue reset
23-1	RESERVED	W	0h	
0	GO	W	0h	Execute the reset specified in RESETLEVEL.LEVEL. Must be written together with the KEY. 1h = Issue reset

2.8.40 BORTHRESHOLD Register (Offset = 1308h) [Reset = 00000000h]

BORTHRESHOLD is shown in Table 2-60.

Return to the Summary Table.

BOR threshold selection

Table 2-60. BORTHRESHOLD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	0h	
1-0	LEVEL	R/W	0h	LEVEL specifies the desired BOR threshold and BOR mode. 0h = Default minimum threshold; a BOR0- violation triggers a BOR 1h = A BOR1- violation generates a BORLVL interrupt 2h = A BOR2- violation generates a BORLVL interrupt 3h = A BOR3- violation generates a BORLVL interrupt

2.8.41 BORCLRCMD Register (Offset = 130Ch) [Reset = 00XXXXXXh]

BORCLRCMD is shown in Table 2-61.

Return to the Summary Table.

Set the BOR threshold

Table 2-61. BORCLRCMD Register 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. C7h = Issue clear
23-1	RESERVED	W	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. 1h = Issue clear

2.8.42 SYSOSCFCLCTL Register (Offset = 1310h) [Reset = 00XXXXXXh]

SYSOSCFCLCTL is shown in Table 2-62.

Return to the Summary Table.

SYSOSC frequency correction loop (FCL) ROSC enable

Table 2-62. SYSOSCFCLCTL Register 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. 2Ah = Issue Command
23-1	RESERVED	W	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. 1h = Enable the SYSOSC FCL

2.8.43 LFXTCTL Register (Offset = 1314h) [Reset = 00XXXXXXh]

LFXTCTL is shown in Table 2-63.

Return to the Summary Table.

LFXT and LFCLK control

Table 2-63. LFXTCTL Register 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. 91h = Issue command
23-2	RESERVED	W	0h	
1	SETUSELFXT	W	0h	Set SETUSELFXT to switch LFCLK to LFXT. Once set, SETUSELFXT remains set until the next BOOTRST. 0h = 0 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. 0h = LFXT not started 1h = Start LFXT

2.8.44 EXLFCTL Register (Offset = 1318h) [Reset = 00XXXXXXh]

EXLFCTL is shown in Table 2-64.

Return to the Summary Table.

LFCLK_IN and LFCLK control

Table 2-64. EXLFCTL Register 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. 36h = Issue command
23-1	RESERVED	W	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. 1h = Use LFCLK_IN as the LFCLK source

2.8.45 SHDNIOREL Register (Offset = 131Ch) [Reset = 00XXXXXXh]

SHDNIOREL is shown in Table 2-65.

Return to the Summary Table.

SHUTDOWN IO release control

Table 2-65. SHDNIOREL Register 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. 91h = Issue command
23-1	RESERVED	W	0h	
0	RELEASE	W	0h	Set RELEASE to release the IO after a SHUTDOWN mode exit. 1h = Release IO

2.8.46 EXRSTPIN Register (Offset = 1320h) [Reset = 00XXXXXXh]

EXRSTPIN is shown in Table 2-66.

Return to the Summary Table.

Disable the reset function of the NRST pin

Table 2-66. EXRSTPIN Register 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. 1Eh = Issue command
23-1	RESERVED	W	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. 0h = Reset function of NRST pin is enabled 1h = Reset function of NRST pin is disabled

2.8.47 SYSSTATUSCLR Register (Offset = 1324h) [Reset = 00XXXXXXh]

SYSSTATUSCLR is shown in Table 2-67.

Return to the Summary Table.

Clear sticky bits of SYSSTATUS

Table 2-67. SYSSTATUSCLR Register 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. CEh = Issue command
23-1	RESERVED	W	0h	
0	ALLECC	W	0h	Set ALLECC to clear all ECC related SYSSTATUS indicators. 1h = Clear ECC error state

2.8.48 SWDCFG Register (Offset = 1328h) [Reset = 00XXXXXXh]

SWDCFG is shown in Table 2-68.

Return to the Summary Table.

Disable the SWD function on the SWD pins

Table 2-68. SWDCFG Register 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. 62h = Issue command
23-1	RESERVED	W	0h	
0	DISABLE	W	0h	Set DISABLE to disable the SWD function on SWD pins, allowing the SWD pins to be used as GPIO. 1h = Disable SWD function on SWD pins

2.8.49 FCCCMD Register (Offset = 132Ch) [Reset = 00XXXXXXh]

FCCCMD is shown in Table 2-69.

Return to the Summary Table.

Frequency clock counter start capture

Table 2-69. FCCCMD Register 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. 0Eh = Issue command
23-1	RESERVED	W	0h	
0	GO	W	0h	Set GO to start a capture with the frequency clock counter (FCC). 1h = 1

2.8.50 SHUTDNSTORE0 Register (Offset = 1400h) [Reset = 00000000h]

SHUTDNSTORE0 is shown in Table 2-70.

Return to the Summary Table.

Shutdown storage memory (byte 0)

Table 2-70. SHUTDNSTORE0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	
7-0	DATA	R/W	0h	Shutdown storage byte 0

2.8.51 SHUTDNSTORE1 Register (Offset = 1404h) [Reset = 00000000h]

SHUTDNSTORE1 is shown in Table 2-71.

Return to the Summary Table.

Shutdown storage memory (byte 1)

Table 2-71. SHUTDNSTORE1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	
7-0	DATA	R/W	0h	Shutdown storage byte 1

2.8.52 SHUTDNSTORE2 Register (Offset = 1408h) [Reset = 00000000h]

SHUTDNSTORE2 is shown in Table 2-72.

Return to the Summary Table.

Shutdown storage memory (byte 2)

Table 2-72. SHUTDNSTORE2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	
7-0	DATA	R/W	0h	Shutdown storage byte 2

2.8.53 SHUTDNSTORE3 Register (Offset = 140Ch) [Reset = 00000000h]

SHUTDNSTORE3 is shown in Table 2-73.

Return to the Summary Table.

Shutdown storage memory (byte 3)

Table 2-73. SHUTDNSTORE3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	
7-0	DATA	R/W	0h	Shutdown storage byte 3

2.8.54 ADCSEQFRCGB Register (Offset = 1410h) [Reset = 00000000h]

ADCSEQFRCGB is shown in Table 2-74.

Return to the Summary Table.

ADC Global Sequence Force

Table 2-74. ADCSEQFRCGB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	SEQ3	R/W	0h	Generate synchronous SW trigger for SEQ3
2	SEQ2	R/W	0h	Generate synchronous SW trigger for SEQ2
1	SEQ1	R/W	0h	Generate synchronous SW trigger for SEQ1
0	SEQ0	R/W	0h	Generate synchronous SW trigger for SEQ0

2.8.55 ADCSEQFRCGBSEL Register (Offset = 1414h) [Reset = 00000000h]

ADCSEQFRCGBSEL is shown in Table 2-75.

Return to the Summary Table.

ADC Global Sequence Force Select

Table 2-75. ADCSEQFRCGBSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	0h	
1	ADCB	R/W	0h	Generate synchronous SW trigger for ADCB
0	ADCA	R/W	0h	Generate synchronous SW trigger for ADCA

2.8.56 M33SPARESOCLOCK1 Register (Offset = 1418h) [Reset = 00000000h]

M33SPARESOCLOCK1 is shown in Table 2-76.

Return to the Summary Table.

M33C1 Spare SOC LOCK Reg 1

Table 2-76. M33SPARESOCLOCK1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SPARE	R/W	0h	Spare SOC LOCK Register 1

2.8.57 M33SPARESOCLOCK2 Register (Offset = 141Ch) [Reset = 00000000h]

M33SPARESOCLOCK2 is shown in Table 2-77.

Return to the Summary Table.

M33C1 Spare SOC LOCK Reg 2

Table 2-77. M33SPARESOCLOCK2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SPARE	R/W	0h	Spare SOC LOCK Register 2

2.8.58 SYSCTL_READ_REG Register (Offset = 1420h) [Reset = 00000000h]

SYSCTL_READ_REG is shown in Table 2-78.

Return to the Summary Table.

Sysctl read only Reg

Table 2-78. SYSCTL_READ_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ecoreg	R/W	0h	Read only register

2.8.59 FWEPROTMAIN Register (Offset = 3000h) [Reset = 00000000h]

FWEPROTMAIN is shown in Table 2-79.

Return to the Summary Table.

1 Sector Write-Erase per bit starting at address 0x0 of flash

Table 2-79. FWEPROTMAIN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R/W	0h	Not used

2.8.60 FWPROTMAINDATA Register (Offset = 3014h) [Reset = 00000000h]

FWPROTMAINDATA is shown in Table 2-80.

Return to the Summary Table.

Read-Write Protection for first 4 Sectors of Data Bank

Table 2-80. FWPROTMAINDATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	
7-0	DATA	R/W	0h	Not used

2.8.61 FRXPROTMAINSTART Register (Offset = 3018h) [Reset = 000000XXh]

FRXPROTMAINSTART is shown in Table 2-81.

Return to the Summary Table.

Flash RX Protection Start Address

Table 2-81. FRXPROTMAINSTART Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	RESERVED	R/W	0h	
21-6	ADDR	R/W	0h	Not used
5-0	RESERVED	R/W	0h	

2.8.62 FRXPROTMAINEND Register (Offset = 301Ch) [Reset = 000000XXh]

FRXPROTMAINEND is shown in Table 2-82.

Return to the Summary Table.

Flash RX Protection End Address

Table 2-82. FRXPROTMAINEND Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	RESERVED	R/W	0h	
21-6	ADDR	R/W	0h	Not used
5-0	RESERVED	R/W	0h	

2.8.63 FIPPROTMAINSTART Register (Offset = 3020h) [Reset = 000000XXh]

FIPPROTMAINSTART is shown in Table 2-83.

Return to the Summary Table.

Flash IP Protection Start Address

Table 2-83. FIPPROTMAINSTART Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	RESERVED	R/W	0h	
21-6	ADDR	R/W	0h	Not used
5-0	RESERVED	R/W	0h	

2.8.64 FIPPROTMAINEND Register (Offset = 3024h) [Reset = 000000XXh]

FIPPROTMAINEND is shown in Table 2-84.

Return to the Summary Table.

Flash IP Protection End Address

Table 2-84. FIPPROTMAINEND Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	RESERVED	R/W	0h	
21-6	ADDR	R/W	0h	Not used
5-0	RESERVED	R/W	0h	

2.8.65 FLBANKSWPPOLICY Register (Offset = 3038h) [Reset = 00XXXXXXh]

FLBANKSWPPOLICY is shown in Table 2-85.

Return to the Summary Table.

Flash Bank Swap Policy

Table 2-85. FLBANKSWPPOLICY Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	Must have KEY==0xCA(202) for write CAh = Write Key
23-1	RESERVED	W	0h	
0	DISABLE	W	0h	1: Disables Policy To Allow Flash Bank Swapping 1h = Disallow Bank Swap

2.8.66 FLBANKSWP Register (Offset = 303Ch) [Reset = 00XXXXXXh]

FLBANKSWP is shown in Table 2-86.

Return to the Summary Table.

Flash MAIN bank address swap

Table 2-86. FLBANKSWP Register 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. 58h = Issue write
23-1	RESERVED	W	0h	
0	USEUPPER	W	0h	1: Use Upper Bank as Logical 0 0h = Normal (default) memory map addressing scheme 1h = Flash upper region address space swapped with lower region

2.8.67 FWENABLE Register (Offset = 3044h) [Reset = 00XXXXXXh]

FWENABLE is shown in Table 2-87.

Return to the Summary Table.

Security Firewall Enable Register

Table 2-87. FWENABLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	Not used 76h = Write Key
23-9	RESERVED	W	0h	
8	SRAMBOUNDARYLOCK	W	0h	Not used 1h = SRAMBOUNDARY MMR Locked
7	RESERVED	W	0h	
6	FLIPPROT	W	0h	Not used 1h = Turn On Flash IP Protection
5	RESERVED	W	0h	
4	FLRXPROT	W	0h	Not used 1h = Turn On Flash Read-eXecute Protection
3-0	RESERVED	W	0h	

2.8.68 SECSTATUS Register (Offset = 3048h) [Reset = 00000XXXh]

SECSTATUS is shown in Table 2-88.

Return to the Summary Table.

Security Configuration status

Table 2-88. SECSTATUS Register 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 0h = 0 1h = 1
9	RESERVED	R	0h	
8	SRAMBOUNDARYLOCK	R	0h	Not used 0h = 0 1h = 1
7	RESERVED	R	0h	
6	FLIPPROT	R	0h	Not used 0h = 0 1h = 1
5	RESERVED	R	0h	
4	FLRXPROT	R	0h	Not used 0h = 0 1h = 1
3	RESERVED	R	0h	
2	CSCEXISTS	R	0h	1: CSC Exists in the system 0h = System does not have a CSC 1h = System does have a CSC
1	RESERVED	R	0h	
0	INITDONE	R	0h	1: CSC has been completed 0h = INIT is not yet done 1h = INIT is done

2.8.69 INITDONE Register (Offset = 3060h) [Reset = 00XXXXXXh]

INITDONE is shown in Table 2-89.

Return to the Summary Table.

INITCODE PASS

Table 2-89. INITDONE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	Must have KEY==0x9D(157) for write 9Dh = Issue Reset
23-1	RESERVED	W	0h	
0	PASS	W	0h	INITCODE writes 1 for PASS, left unwritten a timeout will occur if not blocked 1h = INITCODE PASS

Chapter 3

CPU

The CPU subsystem (MCPUSS) includes the Arm Cortex-M33 processor and the interrupt logic.

3.1 Overview 271
3.2 CPU 271
3.3 Interrupts and Exceptions 276
3.4 CPU Peripherals 281
3.5 Read-Only Memory (ROM) 286

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-M33 32-bit CPU and internal peripherals
- The CPU bus splitter and router
- The interrupt management logic and DEEPSLEEP entry and exit logic
- The CPU debug interface to the debug subsystem
- The read-only memory (ROM) used for the BCR and BSL

The top level architecture of the MCPUSS is shown in Figure 3-1.

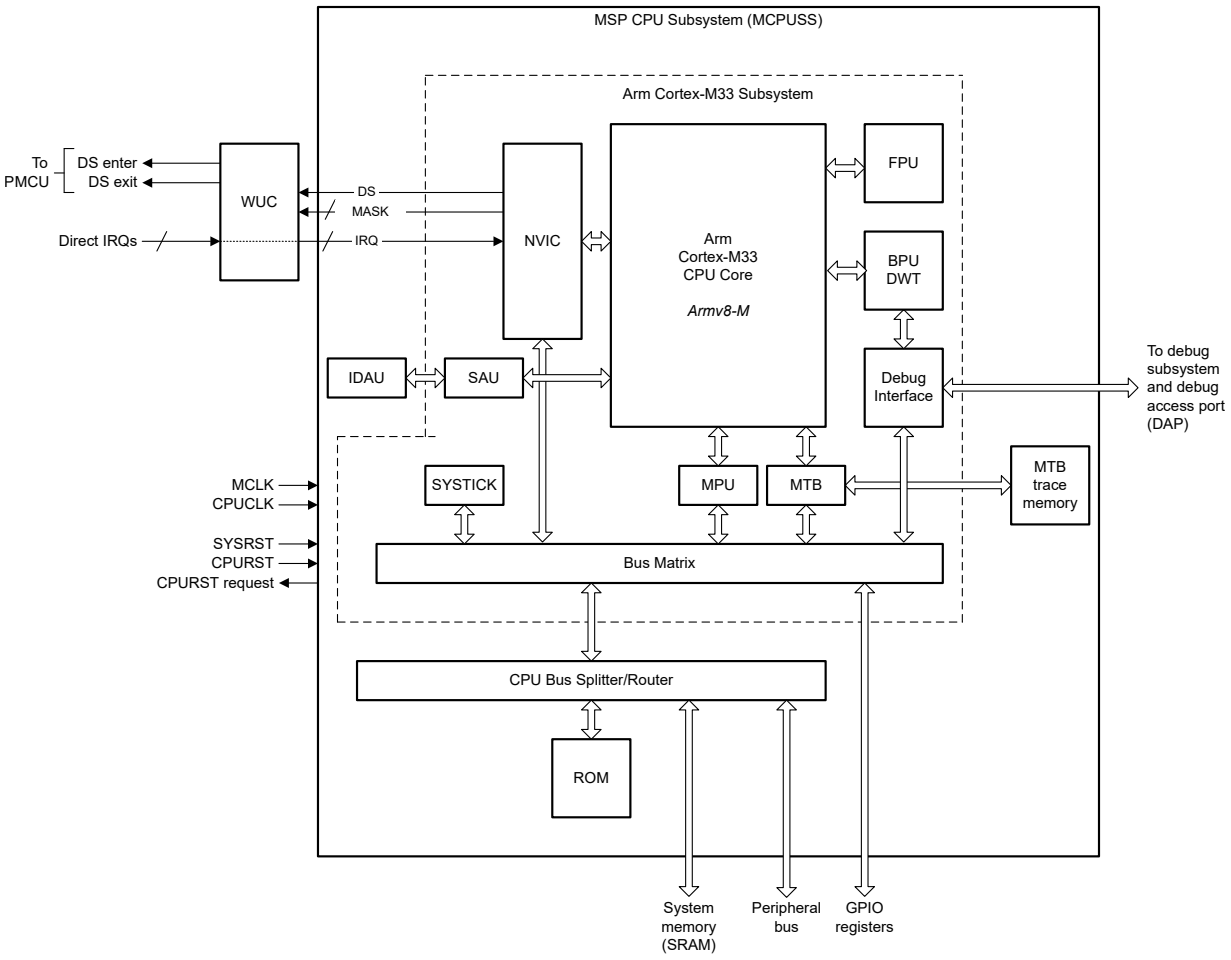


Figure 3-1. MSPM33Cxx MCPUSS Top Level Diagram

3.2 CPU

The CPU subsystem (MCPUSS) includes the Arm Cortex-M33 processor and the interrupt logic.

3.2.1 Arm Cortex-M33 CPU

The MCPUSS contains an energy-efficient Arm Cortex-M33 CPU implementing the Armv8-M instruction set architecture with support for CPU clock speeds up to 160MHz. The Cortex-M33 is a Harvard architecture style 32-bit processor with a 3-stage ultra-low power pipeline.

The Cortex-M33 implementation on MSPM33Cxx devices has the following features:

- Up to 160MHz execution frequency
- Little-endian (least significant byte at lowest byte address location)
- Support for 32-bit word instruction fetches
- Single cycle 32×32 multiply instruction
- Integrated memory protection unit (MPU)
- Integrated security attribution unit (SAU)
- Integrated implementation defined attribution unit (IDAU)
- Integrated floating point unit (FPU)
- DSP extensions
- User and privileged execution modes
- Integrated 24-bit system tick timer (SYSTICK)
- Two hardware breakpoints and eight hardware watchpoints for debug
- Micro Trace Buffer (MTB)
- Reset-all-registers support
- Vector table offset support

The Cortex-M33 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-M33 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-M33 processor, refer to the *Arm Cortex-M33 Devices Generic User's Guide*.

3.2.2 CPU Register File

The Arm Cortex-M33 processor instructions operate on registers in the CPU register file. The processor contains a register file consisting of 16 32-bit wide standard registers and 3 special registers as shown in CPU Registers.

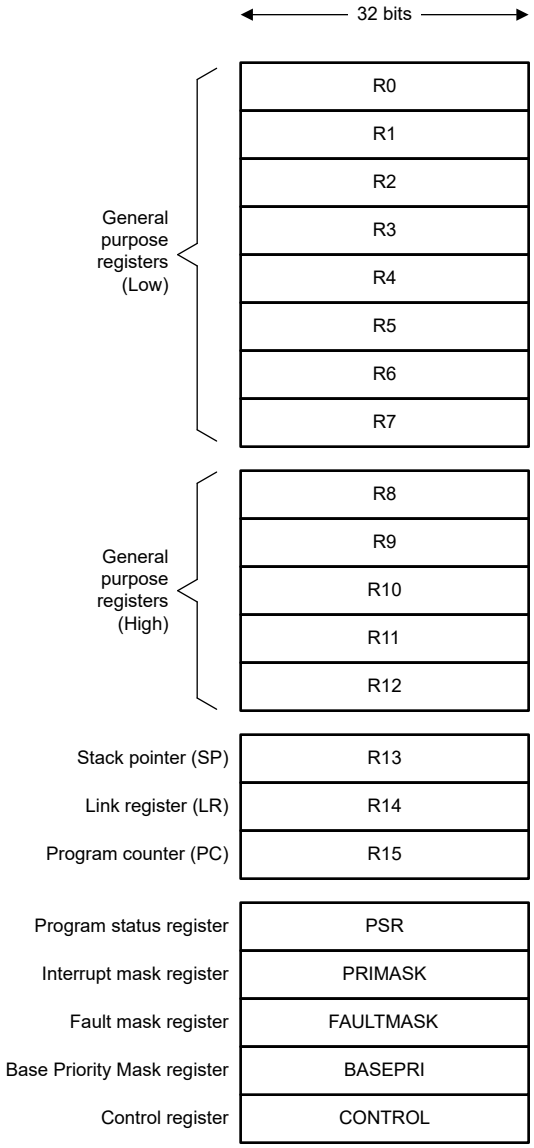


Figure 3-2. CPU Registers

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 MSP is always used. When the processor is running in thread mode, the MSP or 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).

Both the MSP and PSP have two seperate register's for the secure and non-secure state. For more information on which state the processor is in please see Section 3.4.4

The MSPLIM and PSPLIM are registers that limit the extent to which the MSP and PSP registers can descend respectively. Similar to the MSP and PSP registers, they also both have a secure and non-secure register.

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.
- **Fault Mask Register (FAULTMASK):** The Fault mask register prevents activation of all exceptions with configurable priority and some exceptions with fixed priority. The priority it is boosted to changes based on the value of BIT14 (PRIS) and BIT13 (BFHFNMINS) bit of the AIRCR register in the SCB. For more information please see the Fault Mask register description in the Arm Cortex-M33 Devices Generic User Guide.
- **Base Priority Mask Register (BASEPRI)** The base priority mask register can be used to change the priority level that is required for exception preemption. BIT0 to BIT7 are used for configure what the boosted priority will be. When boosting a non-secure exception only the bottom half bits are used.
- **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
IEPSR	IPSR and EPSR

For more details on the register file, please see Arm Cortex-M33 Devices Generic User Guide.

3.2.3 Stack Behavior

The Arm Cortex-M33 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-M33 supports managing two independent stacks with two pointers: the main stack (MSP) and the process stack (PSP).

3.2.4 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 handles 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 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, and SCB).

3.2.5 Address Space and Supported Data Sizes

MSPM33 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 bytes, 16-bit half-words, and 32-bit words data sizes. Signed and unsigned data is supported, and signed data is stored in CPU registers in 32-bit two's complement format. The Armv8-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.

Stores from CPU registers to memory are sign agnostic.

All instruction and data accesses use little endian byte order.

3.2.6 Secure memory partitioning

The Arm Cortex-M33 CPU can define regions as secure, non-secure callable, non-secure. These regions can limit what the processor can access to make sure a bad actor cannot read or access memory limited by the programmer.

Secure Memory

Secure addresses are only accessible by secure software or secure masters. Attempting to access secure memory from a non-secure location will result in a secure fault being thrown by the processor.

For more information on which regions are secure and non-secure please see the Section 3.4.5

Non-secure callable (NSC) memory

In the non-secure callable region of memory the programmer can use the secure gateway command (SG) to program will run a secure state while in this region. Any code inside this area after excecuting the SG command will be able to access secure regions of memory. If the SG command is never executed any access to secure regions of memory will result in a secure fault.

Non-secure (NS) memory

This memory location is accessable by all software running on the device. If the program counter is in this location the program will be restricted to only accessing other code in this region.

For detailed information on the memory protect, see the Security Attribuition and Memory Protection section of the Arm Cortex-M33 Devices Generic User Guide.

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 Non-Maskable Interrupt (NMI, software trigger or hardware error signal from SYSCTL)
- A Secure Fault
- 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 States

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 but has not completed)
- **Active and pending** (actively being serviced by the processor and there is a pending exception from the same source)

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, HardFault) 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)

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)
- The Wake-Up Controller (WUC)

MSPM33 devices include an Arm Nested Vectored Interrupt Controller (NVIC) with the Cortex-M33 CPU for managing peripheral interrupts. The NVIC operation is tightly integrated with the processor and supports up to 480 native peripheral interrupt sources.

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 480 native peripheral interrupt sources. For information on which peripheral is being mapped please see the corresponding data sheet.

The NVIC is configured through memory-mapped registers in the system private peripheral bus (PPB) region. See *Arm Cortex-M33 NVIC Registers* 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 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. Each ICER and ISER register supports 32 different interrupts. The 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. For example, SETENA[m] in NVIC_ISERn, allows interrupt 32n + m to be accessed.

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 or ICER registers 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 clear 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. Each ISPR and ICPR register supports 32 interrupts, with interrupt

zero in the BIT0 position (LSB) and interrupt 31 in the BIT31 position (MSB) of each register. For example, SETPEND[m] in NVIC_ISPRn, allows interrupt 32n + m to be accessed.

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 IPR registers in the NVIC. Each priority field is 8 bits in length, and the priority for 4 interrupts is configured per 32-bit IPR register. The Arm Cortex-M33 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.

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-M33 NVIC Registers

Address	Register	CMSIS	Description
0xE000.E100 - 0xE000.E13C	NVIC_ISER0 - NVIC_ISER15	NVIC->ISER[0] - NVIC->ISER[15]	Interrupt set-enable register
0xE000.E180 - 0xE000.E1BC	NVIC_ICER0 - NVIC_ICER15	NVIC->ICER[0] - NVIC->ICER[15]	Interrupt clear-enable register
0xE000.E200 - 0xE000.E23C	NVIC_ISPR0 - NVIC_ISPR15	NVIC->ISPR[0] - NVIC->ISPR[15]	Interrupt set-pending register
0xE000.E280 - 0xE000.E2BC	NVIC_ICPR0 - NVIC_ICPR15	NVIC->ICPR[0] - NVIC->ICPR[15]	Interrupt clear-pending register
0xE000.E300 - 0xE000.E33C	NVIC_IABR0 - NVIC_IABR15	NVIC->IABR[0] - NVIC->IABR[15]	Interrupt active bit register
0xE000.E400 - 0xE000.E5DC	NVIC_IPR0 - NVIC_IPR119	NVIC->IPR[0] - NVIC->IPR[119]	Interrupt priority register (0 - 3) - Interrupt priority register (115 - 119)

3.3.1.2 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-M33 interrupt vector table is 500 words long (2000 bytes). The complete platform interrupt and exception table with vector table addresses is given in Table 3-3.

See the device-specific data sheet for a complete list of which interrupts a particular device supports along with a more detail interrupt vector table describing which NVIC number is used for what peripheral.

Table 3-3. MSPM33 Platform Processor Interrupt and Exception Table

Exception Number	NVIC Number(1)	Priority Group	Secure Vector	Vector Table Address	Vector Description(2)
-	-	-	-	0x0000.0000	Stack pointer
1	-	-4	Reset	0x0000.0004	Reset vector
2	-	-2	NMI	0x0000.0008	NMI handler
3	-	-1	Hard fault	0x0000.000C	Hard fault handler
4	-	Selectable	MemManage	0x0000.0010	Memory Protection Violation
5	-	Selectable	BusFault	0x0000.0014	Memory-related fault
6	-	Selectable	UsageFault	0x0000.0018	UsageFault
7	-	Selectable	SecureFault	0x0000.001C	SecureFault
8	-	-	Reserved	0x0000.0020	-
9	-	-	Reserved	0x0000.0024	-
10	-	-	Reserved	0x0000.0028	-
11	-	-5	SVCall	0x0000.002C	Supervisor call handler
12	-	-4	DebugMonitor	0x0000.0030	-
13	-	-	Reserved	0x0000.0034	-
14	-	-2	PendSV	0x0000.0038	Pended supervisor handler
15	-	-1	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
...	...	...	...	...	...
499	479	Selectable	Device Interrupt 479	0x0000.0050	Handler for Device Interrupt 479

(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).
(2) For more information on the exceptions, please see Arm Cortex-M33 Generic User Guide

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.

3.3.3 Processor Lockup Scenario

There are several exception conditions which can cause the processor to enter a lockup state. On MSPM33 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-M33 includes tightly coupled peripherals for system timing, memory protection, floating point computation, and security

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 3-4 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 3-4. Arm Cortex-M33 System Control Block Registers

Address	Register	CMSIS	Description
0xE000.ED00	CPUID	SCB->CPUID	Read-only register indicating the CPU type and revision
0xE002.ED00	CPUID	SCB->CPUID_NS	Non secure 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.ED18	SHPR1	SCB->SHP[1]	Used to configure priority of Secure fault, Usagefault, BusFault, and Memmanage
0xE000.ED1C	SHPR2	SCB->SHP[2]	Used to configure the priority of the SVCall system handler
0xE000.ED20	SHPR3	SCB->SHP[3]	Used to configure the priority of the SysTick and PendSV system handlers

For detailed information on the SCB register configuration, see the SCB section of the Arm Cortex-M33 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 *SysTick Registers* 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

There are two different SysTick timers: SysTick Secure and SysTick Non-Secure. Both SysTicks are sourced by the same clock the M33 core uses. The main difference between the two registers is where you can access them.

The SysTick timer is a simple 24-bit down counter, which counts down from its reload value to zero. Upon reaching zero, SysTick will reload the value programmed into the reload value register (SYST_RVR) on the next clock cycle, and then again begin counting down to zero. Writing a value of zero to the SYST_RVR disables the counter on the next wrap.

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 3-5. SysTick Registers

Security	Address	Register	CMSIS	Description
Secure	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
Nonsecure	0xE002.E014		SysTick_NS->CTRL	
Secure	0xE000.E014	SYST_RVR	SysTick->LOAD	Reload register used to program the counter reload value to set the SysTick interval in MCLK cycles
Nonsecure	0xE002.E014		SysTick_NS->LOAD	
Secure	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
Nonsecure	0xE002.E018		SysTick_NS->VAL	
Secure	0xE000.E01C	SYST_CALIB	SysTick->CALIB	Not implemented
Nonsecure	0xE002.E01C		SysTick_NS->CALIB	

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.

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-M33, 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 HardFault is generated if MemMange fault is not enabled in the processor. It is also possible to restrict access to both privileged and unprivileged code (no access possible through the processor).

The Arm Cortex-M33 has two memory protection units one controls the secure memory and the other controls the non secure memory. The MPU that is used by the program will change based on what location the program is in. Please see the SAU section for more details.

The MPU is configured through memory-mapped registers in the system private peripheral bus (PPB) region. See *MPU Registers* 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.

Since 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-M33 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.

Table 3-6. MPU Registers

Security	Address	Register	CMSIS	Description
Secure	0xE000.ED90	MPU_TYPE	MPU->TYPE	Type register indicating that the MPU is present.
Nonsecure	0xE002.ED90		MPU_NS->TYPE	

Table 3-6. MPU Registers (continued)

Security	Address	Register	CMSIS	Description
Secure	0xE000.ED94	MPU_CTRL	MPU->CTRL	MPU control register for enabling and configuring the MPU.
Nonsecure	0xE002.ED94		MPU_NS->CTRL	
Secure	0xE000.ED98	MPU_RNR	MPU->RNR	Region select register for using MPU_RBAR and MPU_RASR.
Nonsecure	0xE002.ED98		MPU_NS->RNR	
Secure	0xE000.ED9C	MPU_RBAR	MPU->RBAR	Region base address configuration register.
Nonsecure	0xE002.ED9C		MPU_NS->RBAR	
Secure	0xE000.EDA0	MPU_RASR	MPU->RASR	Region size and memory attributes register.
Nonsecure	0xE002.EDA0		MPU_NS->RASR	

For detailed information on the MPU register configuration, see the MPU section of the Arm Cortex-M33 Devices Generic User Guide.

3.4.4 Security Attribute Unit

The Security Attribute Unit (SAU) tells the CPU if the region being selected is secure or non-secure and is able to configure regions to be secure or non-secure. The processor can use this information to throw a Security fault.

In this implementation of the Arm Cortex-M33 there are 8 regions of memory to configure as secure using the SAU. The SAU register is only readable and configurable when the processor is in a secure region. The programmer can use this register to define regions that are non-secure callable to secure. By doing this the programmer can enter secure regions through NSC regions using the SG function from non secure callable regions of code. By default regions that are not configured by the SAU will be left as secure. The SAU works with the IDAU to determine the final security level of a region please see Security attribution and memory protection section of the Arm Cortex-M33 Processor Technical Reference Manual. To see more information on the IDAU configuration please see the Section 3.4.5.

Table 3-7. Security Attribute Unit Register

Address	Register	CMSIS	Description
0xE000.EDD0	SAU_CTRL	SAU->CTRL	Allows enabling of the Security Attribution Unit
0xE000.EDD4	SAU_TYPE	SAU->TYPE	Indicates the number of regions implemented by the Security Attribution Unit
0xE000.EDD8	SAU_RNR	SAU->RNR	Selects the region currently accessed by SAU_RBAR and SAU_RLAR
0xE000.EDDC	SAU_RBAR	SAU->RBAR	Provides indirect read and write access to the base address of the currently selected SAU region
0xE000.EDE0	SAU_RLAR	SAU->RLAR	Allows the user to define the currently selected SAU region as secure
0xE000.EDE4	SFSR	SAU->SFSR	Provides information about any security related faults

Table 3-7. Security Attribute Unit Register (continued)

Address	Register	CMSIS	Description
0xE000.EDE8	SFAR	SAU->SFAR	Shows the address of the memory location that caused the security violation

For more information on the SAU registers please see the SAU section of the Arm Cortex-M33 Devices Generic User Guide

3.4.5 Implementation Defined Attribution Unit (IDAU)

By default the Arm Cortex-M33 defines all memory as secure through the SAU. The SAU can then define certain regions as non-secure or secure. To define regions of memory as non-secure callable the IDAU is used which configures set regions of memory as non-secure callable.

The IDAU defines addresses with non-secure or non-secure callable. The IDAU for the MSPM33C uses bit 28 of the address to determine if memory is secure or non-secure. If bit 28 of the address is 1 the address is attributed with Non-secure callable and if bit 28 of the address is 0 the memory is attributed with non-secure. For example register 0x3000.0000 is attributed with non-secure callable.

By default the IDAU defines the memory as just non-secure callable and non-secure. The programmer must define the regions of memory as secure if they want to use the secure memory features of the Arm Cortex-M33 section. To define which regions they would like the non-secure callable to be secure they must use the SAU. Please see the deviceSection 3.2.6 for information on the different security levels of memory. See also the Section 3.4.4 for information on configuring secure regions of memory.

For detailed information on the IDAU, see the IDAU section of the Arm Cortex-M33 Devices Generic User Guide.

3.4.6 Floating Point Unit (FPU)

The floating point unit (FPU) supports single-precision add, subtract, multiply, accumulate, and square root operations. This floating point unit follows the IEEE 754 standard.

The FPU is used to help with floating point computation by reformating the users data into floating point format. The unit has thirty-two 32-bit single word registers as well as sixteen 64-bit doubleword registers.

Note

The FPU is not connected to the NVIC. This requires the programmer to check the FPSCR register to see the status of the FPU.

Table 3-8. FPU Registers

Security	Address	Register	CMSIS	Description
Secure	0xE000.EF34	FPCCR	FPU->FPCCR	Used to control different attributes of the FPU
Nonsecure	0xE002.EF34		FPU_NS->FPCCR	
Secure	0xE000.EF38	FPCAR	FPU->FPCAR	Holds the location of upopulated floating-point register space
Nonsecure	0xE002.EF38		FPU_NS->FPCAR	
Secure	0xE000.EF3C	FPDSCR	FPU->FPDSCR	holds values for reading the status of the FPU
Nonsecure	0xE002.EF3C		FPU_NS->FPDSCR	

For detailed information on the FPU , see the FPU section of the Arm Cortex-M33 Devices Generic User Guide.

3.4.7 Digital Signal Processing Extension

The Arm Cortex-M33 allows specific instructions to be used for digital signal processing (DSP). This gives programmers the ability to use the DSP instructions given in the Armv8-m instruction set. For more information on these instructions please see Armv8-M Architecture Reference Manual.

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.

Chapter 4

Secure ROM

The read-only memory (ROM) is used to configure the device before starting the user application.

4.1 ROM Overview 288
4.2 Memory Map 288
4.3 Boot Configuration Routine (BCR) 288
4.4 Bootstrap Loader (BSL) 289
4.5 Lifecycle Management 291
4.6 Boot and Startup Sequence 292

4.1 ROM Overview

The ROM contains the executable code for the boot configuration routine (BCR), bootstrap loader (BSL) and Crypto. The ROM feature includes:

- Configuring the SoC Security as per the NON-Main flash data
- Loads TI trim data
- Authentication of the Application in Flash
- ROM Services
  - Authentication of service requests through DSSM
  - Regression Services
  - Debug Services
  - RMA Services
  - BCR Existing Services

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.

MSPM33 C-series devices provide a 32kB ROM that includes the implementation for BCR and BSL.

4.2 Memory Map

High Level View of Different Flash, Data, Engineering, TRIM Sectors and their associated Protections.

Table 4-1. ROM Memory Map

Memory Usage	Memory Size	Start Address	Permission	Purpose
ROM	32KB	TI reserved	Immutable	BCR&BSL code
ROM Extension	12KB	TI reserved	Immutable	Authentication code
BANK0 NM1	2KB	0x80100800	Immutable and BCR has Write/Read permission.	Customer keys
BANK0 NM2	2KB	0x80101000	OTP	Customer application config data, FOTA / Bank Management, PLL tables, Flash Latencies
BANK0 NM3	2KB	0x80101800	Reprogrammable	BCR & BSL config
BANK1 NM1	2KB	0x80102800	OTP	Keys for Customer application authentication
BANK1 NM2	2KB	0x80103000	Reprogrammable	ROM configurations (Sector handling)
BANK1 NM3	2KB	0x80103800	Reprogrammable	BCR & BSL config(Duplicate)

4.3 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)

4.3.1 SWD Mass Erase and Factory 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 factory reset can be individually configured to be enabled, enabled with a unique 128-bit password, or disabled. By default, both commands are enabled.

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

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 mode, configure the BOOTCFG3.FACTORYRESETCMDACCESS fieldin the NONMAIN memory.

4.3.2 Application HASH Verification

The BCR supports executing a complete HASH 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 HASH integrity check at boot, the following information must be programmed into the BCR configuration in configuration memory:

- 32-bit starting address of the CRC check (USER_SECURE_APP_START_ADDR.ADDRESS field)
- The length of the application (specified in bytes) for which the CRC check is applied (USER_SECURE_APP_LENGTH.LENGTH field)
- The precalculated 256-bit HASH value to test against (USER_SECURE_APP_HASH[x].DIGEST field)
- The enabled key value for the HASH check (SECURE_BOOT_MODE.APPMODE field)

In the event that an application HASH 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.

4.3.3 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 or HASH check (even if the application CRC or HASH check is enabled).

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

4.4 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
- Configurable UART and I2C pins

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.

4.4.1 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 APP_REV_POINTER field in configuration memory. The version data is only returned if the address specified in APP_REV_POINTER corresponds to a valid flash memory address.

4.4.2 GPIO Invoke

The bootloader supports hardware invoking after a BOOTRST through the use of a GPIO. The BSL configuration in the configuration 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 configuration memory.

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

To specify the device pin to be used for BSL_invoke, configure the following fields in the configuration 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.

4.4.3 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 BANK0_WRITE_ERASE_PROTECTION_A, BANK0_WRITE_ERASE_PROTECTION_B, BANK1_WRITE_ERASE_PROTECTION_A and BANK1_WRITE_ERASE_PROTECTION_B fields in the configuration memory) will not be erased. The 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 device configuration memory.

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

1. The configuration memory itself is not currently configured to be statically write protected (BANK0_NM_USER_CONFIG field in the configuration memory is set to not protected)
2. The factory reset command is not configured to be disabled (BOOTCFG4.FACTORY_RESET_MODE field in the configuration memory is not set to disabled)

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

Note

If the configuration memory 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 configuration memory following a BSL factory reset.

4.5 Lifecycle Management

MSPM33 C-series devices support lifecycle management to manage the device process from TI factory to development to production to end-of-life. Lifecycle controls which features of the device are enabled during the various stages of its life.

Lifecycle management allows customers to increase the level of security from all open to all closed in a controlled manner, and allows customers to prevent information from getting leaked by converting the device to specific state when device is out of life or in failure analysis.

The bootcode supports SoC lifecycle with two states that represent the operational state of the device.

- HS (High Security) - operational unit
  - The HS lifecycle state is divided into multiple **Device Sub-Type** states that represent the specific configuration the device is operational under.
- DCOM/BAD - decommissioned or broken unit

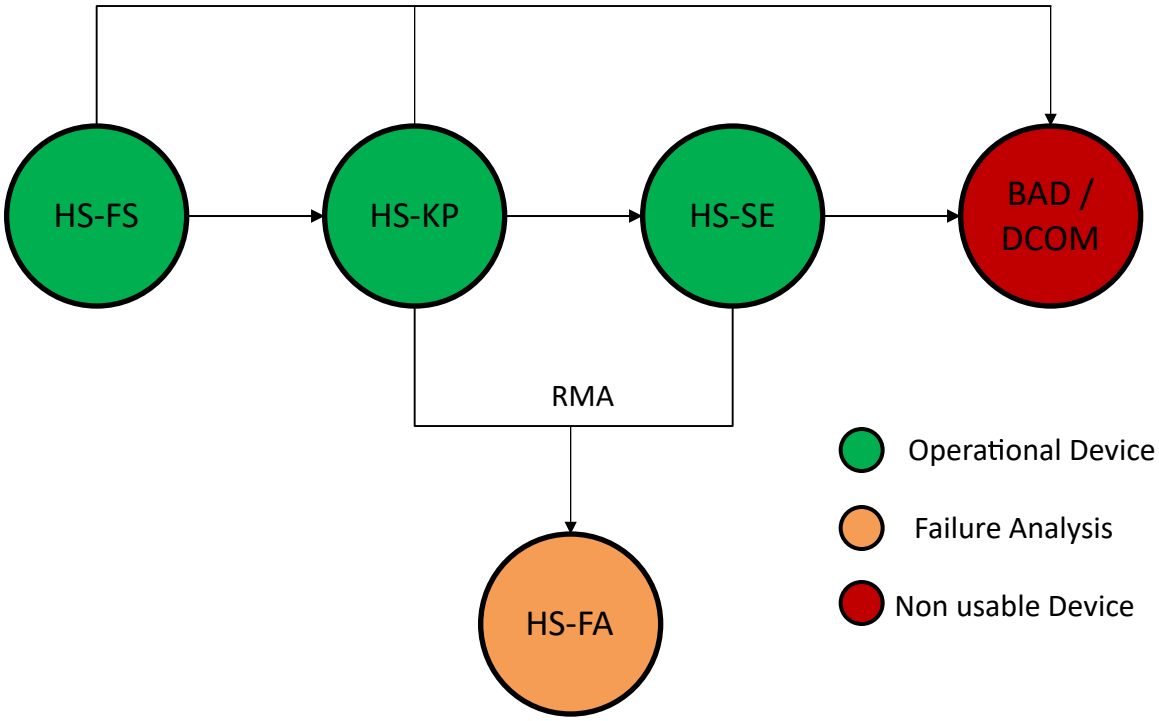


Figure 4-1. Lifecycle Transitions

4.5.1 Device Sub-Type

HS lifecycle state is divided into the following **Device Sub-Type** states. The bootcode will read the corresponding security sector to determine the device sub-types for High Security.

- HS-FS (Field Securable) - The devices shipped from TI will be in this state. In this state the device can be used with or without security features. If security features are not required, then the device will be in this state

for the rest of the device's lifecycle (unless the device enters BAD state or a failure analysis is performed). If security features are required then the device is ready for key provisioning in this state.
- HS-KP (Key Provision) - The device is set to this state after keys (requried for Secure-Boot) are provisioned. In this state the device is ready for programming either the signed Customer Secure Code (CSC) or signed application.
- HS-SE (Security Enforced) - This is the state a device moves to once the signed CSC or application has been programmed and verified by Secure Boot.

In addition to the above mentioned states, the device enters HS-FA (Failure Analysis) state when failure analysis is enabled on the device.

4.5.2 Lifecycle Transitions

During the life of the device, the state changes based on the current state of the device and the security settings that are applied.

The following is sequence of steps for transitioning from HS-FS to HS-KP.
- DSSM command (DSSM_COMMAND_SECURE_KEY_PROVISIONING) shall be given by user to device.
- 8 RoT public Keys (6 for secure boot and 2 for secure debug) shall be sent to the device.
- ROM marks the device as HS-KP.

The following is sequence of steps for transition from HS-KP to HS-SE.
- If the debug port is not locked, then application/CSC can be programmed via debug port.
- Alternatively invoke BSL to program the application
- Program the signed CSC or application over BSL
- Once ROM validates the CSC or application, the device is marked as HS-SE

The following is sequence of steps for transition from HS-KP/HS-SE to HS-FA.
- DSSM command (DSSM_COMMAND_RMA) shall be given by user to device
- If Combined certificate is received and validated then ROM Marks the device as HS-FA
- Device will go to HS-FA Authenticated Flow

To convert the device to DCOM/BAD, user can send DSSM command (DSSM_COMMAND_CONVERT_BAD_DEVICE).

Note

Customer return flow

Customers need to transition the device to HS-FA state from HS-KP / HS-SE before sending the device back to TI for failure analysis. Customer has the option to secure the different assets (e.g. code, keys, etc.) by programming / erasing them. Customer should be aware that erasing the content will restrict / limit the debug of some of the failures (e.g. flash data retention failure cannot be debugged once the content is erased).

4.6 Boot and Startup Sequence

Figure 4-2 illustrates the boot and startup sequence in security enabled applications. At BOOTRST, TI boot-code execution commences. According to the lifecycle status, device will execute corresponding process as secure boot, regular boot, retest flow or other processes.

The bootloader(BSL) can also be invoked by bootcode when a valid bootloader invocation condition is detected. The Bootloader provides a method to program and verify the device Flash memory through a standard serial interface.

During the boot code, 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. In Bootcode DSSM helps in transmission of commands to the device during boot, including

lifecycle state change request, authenticating the debug probe for password-protected debug, mass erase, and factory reset operations.

After successful boot, bootcode 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(Customer Secure Code) 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 MSPM33 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.

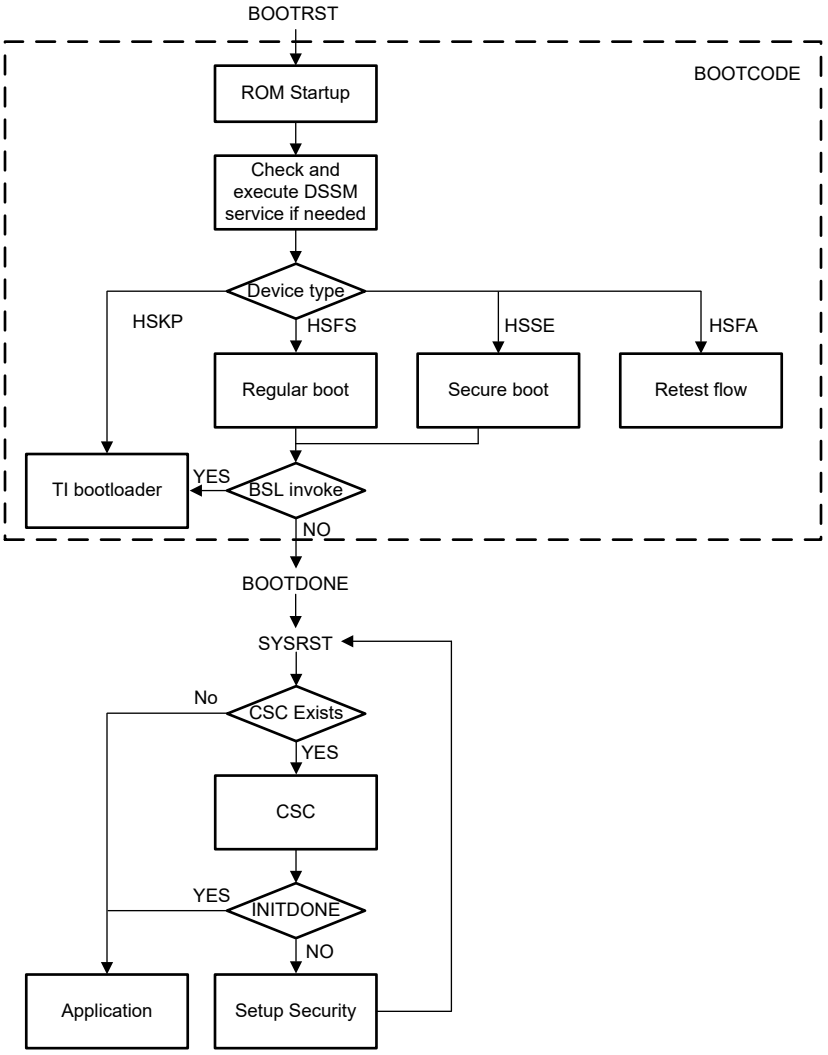


Figure 4-2. Secure Boot and Start Sequence

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 is executed. 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 CSC execution. At the second SYSRST, since INITDONE is YES, the main application is launched.

If CSC is not enable, then only one SYSRST will be issued, and the application will be launched then.

4.6.1 Secure Boot

MSPM33 C-series devices supports a immutable secure boot in ROM leveraging the device's lifecycle management. The immutable secure boot establishes a chain of trust that allows the execution of authenticated

and authorized firmware on the device, utilizing the device's lifecycle management transistions - from a provisioning state (HS-FS) to a permanaently secured operational state (HS-SE).

Secure boot provides mechanisms to validate image integrity through SHA-256 hashing, verify public key authenticity via key hashing, and perform ECDSA signature verification.

ROM secure boot supports the following features:

- Authentication of the firmware using ECDSA(Elliptic Curve Digital Signature Algorithm) P-256 curve
- Integrity of the firmware through SHA-256 hash verification
- Anti-rollback to prevent downgrade to older firmware that can be a potentail security vulberibility
- Key revocation to decomission compromised keys - support for upto 6 keys
- Industry standard MCUBoot trailer processing for security metadata
- Fast secure boot mode for optimized boot sequences

The Root of Trust for the authentication of the firmware image is based on the public keys stored in the immutable OTP memory. Secure boot uses these keys to verify the signature of the firmware that is signed by the corresponding private key. MSPM33 C-series devices supports upto 6 public keys. Secure boot uses the first public key for verification. The rest of the public keys are utilized if the first key is revoked

Secure boot is performed only when the device is in HS-SE state. In HS-SE state, secure boot is performed during one of the followign conditions:

- BOOTRST
- Power-on reset
- Wake up from Shutdown

4.6.2 Customer Secure Code (CSC)

If the user wants ot add additonaly features or functionality to the immutable secure boot on MSPM33 C-series devices then a Customer Secure Code (CSC) (also called secondary bootloader) can be implemented in the main Flash of the device. The ROM provides the followign features to support a CSC.

- Bank swap decision
- Initialization of the secure key store

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 previously (this is determined by reading the SYSCTL.SECCFG.SECSTATUS.INITDONE bit) and it directly calls the main application.

The customer owned secure code can leverage GSC to implement additional security capabilities such as:

- 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.

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.

Chapter 5

NVM (Flash)

The nonvolatile memory (NVM) system provides nonvolatile flash memory for storing executable code and data.

5.1 NVM Overview 297
5.2 Flash Memory Bank Organization 298
5.3 Flash Controller 299
5.4 Flash Programming Interface 307
5.5 Flash Read Interface 314
5.6 FLASHCTL Registers 316
5.7 FRI Registers 321

5.1 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.

5.1.1 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
- 128-bit flash word size (128+8+8 = 144 with ECC)
- Flash Controller Protection through Global Security Controller (GSC)
- Dynamic write/erase protection (configurable at runtime)
- Sector (2KB) and bank erase
- Two Code banks with total 1024 KB Main memory
- Separate Data Flash present (32 KB Main)
- Automatic hardware preverification to extend flash bank longevity
- Automatic hardware post-verification of program/erase
- ECC protection (SECDED)

5.1.2 System Components

The nonvolatile memory system consists of four components (listed below):

- The Flash Controller (for executing read/program/erase operations on the flash banks)
- The Flash Read Interface (FRI) (for restricting reads of the flash banks to the CPU and peripheral bus)
- The Flash Programming Interface (FPI) (for restricting program/erase of the flash banks)
- The Global Security Controller (GSC) (for providing lifecycle state information to the FRI and FPI)

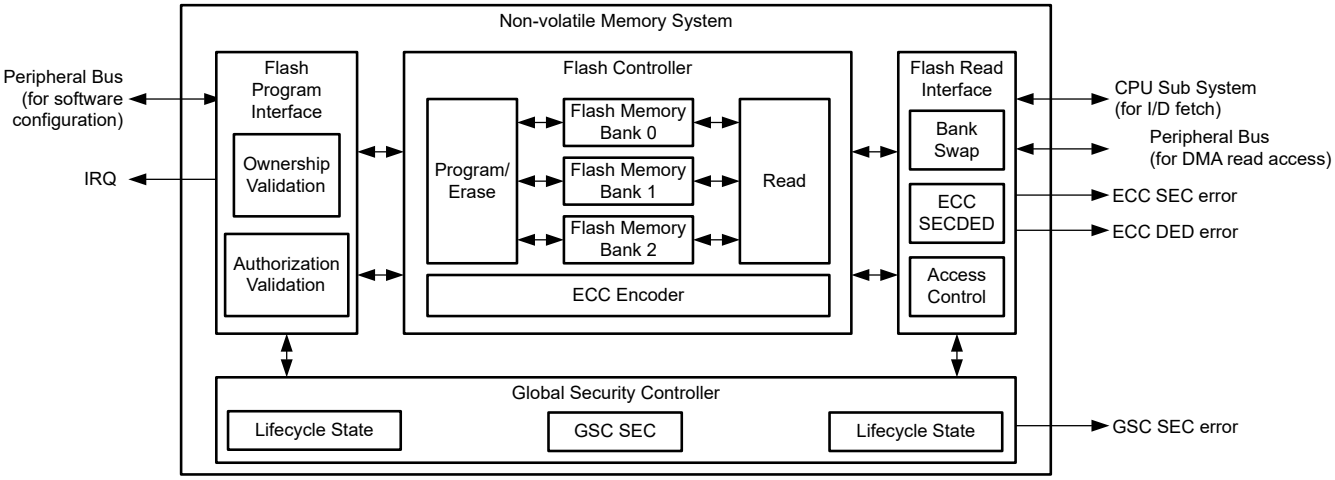


Figure 5-1. Non-volatile Memory System Components

5.1.3 Terminology

Key flash bank terms are defined in this section to be used as a reference for the rest of this chapter.

Table 5-1. 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)	128 data bits (144 bits with ECC)
Word line	Group of flash words within a sector, with maximum program operation limit before sector erase	16 flash words (256 data bytes, 32 ECC bytes)
Sector	Group of word lines that are erased together (minimum erase resolution of the flash memory)	8 word lines (2048 data bytes, 256 ECC bytes)

Table 5-1. 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

5.2 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.

5.2.1 Banks

The nonvolatile memory system provides support for up to 3 flash memory banks (enumerated as BANK0 through BANK2). 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 three flash banks (BANK0 through BANK2), with BANK0 and BANK1 serving as the main code banks, and BANK2 being used as the data flash.

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)

5.2.2 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: MAIN (Flash Memory), NONMAIN (Configuration NVM), and FACTORY.

Table 5-2. Code Bank Memory Regions

Flash Memory Region	Region Contents	Executable	Used by	Programmed by
MAIN (Flash Memory)	Application code and data	Yes	Application	User
NONMAIN (Configuration NVM)	Device boot configuration (BCR and BSL)	No	Boot ROM	TI, User
FACTORY	Analog trim, boot related, and other security information	No	Application	TI only (not modifiable)

Devices with multiple banks implement MAIN, NONMAIN, and FACTORY regions on BANK0 and BANK1, but BANK2 can only implement MAIN.

5.2.3 Addressing

The flash memory regions are assigned to address space in the system memory map.

The NONMAIN and FACTORY regions are assigned to the peripheral address space (0x8010.1800) as they do not contain any executable code. The CPU should not fetch executable instructions from this region.

The MAIN region is assigned to the code address space (0x0000.0000).

Note

The Data Flash MAIN region is assigned to the peripheral address space (0x8000.0000), and can only be used in special configurations.

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.

5.2.3.1 Flash Memory Map

The following table lists the system address space assignments. These assignments are consistent for all devices.

Table 5-3. Flash Region Memory Map

Bank	Region	Read Type	ECC Behavior	Base Address
0	MAIN	Instruction fetch or data read	Corrected	0x0000.0000
		Data read	ECC code	0x8020.0000
	NONMAIN	Data read	Corrected	0x8010.1800
			ECC code	0x8026.0300
	FACTORY	Data read	Corrected	0x8011.0000
			ECC code	0x8027.0000
1	MAIN	Instruction fetch or data read	Corrected	0x0008.0000
		Data read	ECC code	0x8021.0000
	NONMAIN	Data read	Corrected	0x8011.3800
			ECC code	0x8026.0700
	FACTORY	Data read	Corrected	0x8011.1000
			ECC code	0x8027.0200
Data Flash	MAIN	Instruction fetch or data read	Corrected	0x8000.0000
		Data read	ECC code	0x8024.0000

5.3 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 using low level register accesses to the flash controller, review the remainder of this section in detail.

5.3.1 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 Flash Controller Commands.

Table 5-4. 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.

5.3.2 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.

5.3.3 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 128 data bits (plus 2 ECC bytes) at a time, with control to limit the scope of a program operation to specific bytes within a 128-bit flash word.

Some devices additionally have support for a multi-word programming mode where 2 or 4 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.

5.3.3.1 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 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.

5.3.3.2 Programming Less Than One Flash Word

In general, the simplest way to program the flash memory is one flash word at a time (128 bits plus 2 ECC bytes). 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 128 bits of data at a time ensures that the 2 ECC bytes which correspond to the 128-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 128 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 128-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 128-bit flash word and the 2 ECC bytes 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.

Executing 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. CMDDATA0 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. CMDDATA0 is written with DATA2, CMDDATA1 is not used.
   b. CMDBYTEEN = 0x00C.
3. Program 16 bit at address 0x1004 (ECC is not programmed at this step):
   a. CMDDATA1 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. CMDDATA0 is written with DATA1 and DATA2.
b. CMDDATA1 is written with DATA3 and DATA4.
c. CMDBYTEEN = 0x1FF.

5.3.3.3 Target Data Alignment (Devices with Single Flash Word Programming Only)

For devices which only support single word programming, the CMDDATA0, CMDDATA1, CMDDATA2, CMDDATA3, 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, CMDDATA1 is always loaded with BIT63-BIT32, CMDDATA2 is always loaded with BIT95-BIT64, and CMDDATA3 is always loaded with BIT127-BIT96 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 128 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 (128-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).

5.3.3.4 Target Data Alignment (Devices With Multiword Programming)

For devices that support 2- or 4-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 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).

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 target data word can be written to the same 128-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 or 4 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, or 4 word programming operations.

Table 5-5. 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
CMDDATA1:0 / CMDECC0	CMDINDEX = 0	Target data word 0 if the target address ends in 0b0000	Target data word 0	Not supported
CMDDATA3:2 / CMDECC1	CMDINDEX = 1	Target data word 0 if the target address ends in 0b1000	Target data word 1	

Table 5-6. 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
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
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

5.3.3.5 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 (128-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 128 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 bytes. 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 (128 or 144 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.
   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.

5.3.4 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 (2KB) 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.

5.3.4.1 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.

5.3.4.2 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:
   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.

5.3.5 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.

5.3.5.1 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:
   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.

5.3.6 Command Diagnostics

The flash controller updates several software-readable registers to communicate information about an initiated operation.

5.3.6.1 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.

5.3.6.2 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.

5.3.6.3 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.

5.3.7 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).

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 5.3.4.2, 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

5.4 Flash Programming Interface

The Flash Programming Interface (FPI) provides two write protection mechanisms (flash resource ownership check, authorization check) 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 an error signal reported on the FPI-SEC error interface.

5.4.1 Flash Resource Ownership Check

This initial level security check simply checks if the code thread trying to perform program/erase operation has the current ownership (secure/non-secure & privilege/non-privilege) of the flash resource as defined in FPC_FLSEMSTAT.SEC and FPC_FLSEMSTAT.PRIV register fields. This check happens on the MMR interface going to the Flash wrapper. Once the ownership check passes, the program/erase write access is forwarded to the flash MMRs.

Note

This FPI resource ownership check doesn't consider debug information and this check is still done even if debugger makes an access. However, no error is sent to SEC in such cases. The reason is debugger security is already handled inside the M33 core and if the access is reaching FPI, it means, debugger can indeed grab the FLSEM and program flash.

Note

DMA cannot access flash. (Design note: Flash NW's VBUSP Peripheral bridge wouldn't have DMA access port)

5.4.2 Authorization Check

The GSC provides information on the security level, privilege level, and the write/erase protection level to the FPI. The FPI then uses these inputs to perform the autorization check and generate the error signal (*fxb_sys_addr_err*). The error signal is used to block commands not allowed by safety or security or if the system address is out of range. The values of the error signal are described in Table 5-7.

Table 5-7. Authorization Error Signal

#	Priority	Brief Description	Condition	fxb_sys_addr_err	Fault/Error
1	1 (highest)	Default when not executing command	Flash Controller is not currently executing a command (*sys_flash_off_done_mscbit* = 1) *sys_flash_off_done_mscbit* is the BUSY_N signal in the above timing waveform	1 (error, not authorized)†	N/A
2	2	IDLE/NOOP (0x0) and CLEARSTATUS (0x5) commands are always allowed	The *fxb_cmd_type* equals IDLE/NOOP (0x0) or CLEARSTATUS (0x5)	0 (no error, authorized)	N/A
3	2	Not a valid flash address for this controller configuration	System address (*fxb_sys_addr*) is not an address contained in Flash Controller	1 (error, not authorized)	ILLADDR
4	2	Reserved command	The *fxb_cmd_type* is reserved (0x7)	1 (error, not authorized)	ILLCMD
5	2	Reserved sizes	The *fxb_cmd_size* is equal to either: EIGHTWORD (0x3), or A reserved value (0x6 or 0x7)	1 (error, not authorized)	ILLSIZE
6	2	Unaligned address/ size	The *fxb_sys_addr* is not aligned to the *fxb_cmd_size* (this forces commands to not cross a sector boundary). Alignment is: 1. ONEWORD (0x0): alignment must be per 16 bytes (*fxb_sys_addr[3:0]* must equal 4'b0000) 2. TWOWORD (0x1): alignment must be per 32 bytes (*fxb_sys_addr[4:0]* must equal 5'b00000) 3. FOURWORD (0x2): alignment must be per 64 bytes (*fxb_sys_addr[5:0]* must equal 6'b000000) 4. EIGHTWORD (0x3): not allowed; covered by 'Reserved sizes' rule 5. SECTOR (0x4): alignment must be per 2048 bytes (*fxb_sys_addr[10:0]* must equal 11'b00000000000) 6. BANK (0x5): alignment is ignored	1 (error, not authorized)	ILLSIZE
7	2	Illegal combination of programming commands and size	a) The *fxb_cmd_type* equals PROGRAM (0x1), and *fxb_cmd_size* is equal to either: SECTOR (0x4), or BANK (0x5)	1 (error, not authorized)	ILLSIZE
8	2	Illegal combination of erase commands and size	1. The *fxb_cmd_type* equals ERASE (0x2), and 2. The *fxb_cmd_size* is equal to either: a. ONEWORD (0x0), or b. TWOWORD (0x1), or c. FOURWORD (0x2), or d. EIGHTWORD (0x3) -> this one is not required as it is covered already	1 (error, not authorized)	ILLSIZE

Table 5-7. Authorization Error Signal (continued)

#	Priority	Brief Description	Condition	fxb_sys_addr_err	Fault/Error
9	3	Modification of MAIN sectors by secure thread	a) The *fxb_cmd_type* equals either PROGRAM (0x1), or ERASE (0x2)‡, and b) The *fxb_region_id* calculated from *fxb_sys_addr* matches MAIN (0x0) and c) Sector (or 8sectors) bit value that fxb_sys_addr decodes based on FPC_WEPROT_REG_x = 0 and d) FPC_FLSEMSTAT.ASSIGNED = 1 and e) FPC_FLSEMSTAT.SEC = 1 and sector (or 8sectors) bit value that fxb_sys_addr decodes based on FPC_SECATTRIB_REG_x is i. ‘1’ (secure) or ii. ‘0’ (non-secure) and FPC_ATTRIBVIOLS_CONFIG.SECVIOL = 0 and f) Either FPC_FLSEMSTAT.PRIV = 1 and sector (or 8sectors) bit value that fxb_sys_addr decodes based on FPC_PRIVATTRIB_REG_x is i. ‘1’ (privilege) or ii. ‘0’ (non-privilege) and FPC_ATTRIBVIOLP_CONFIG.PRIVVIOL = 0 g) Or FPC_FLSEMSTAT.PRIV = 0 and sector (or 8sectors) bit value that fxb_sys_addr decodes based on FPC_PRIVATTRIB_REG_x is ‘0’ (non-privilege) h) Program/Erase is allowed to MAIN based on FlashRegionProtections	0 (no error, authorized)	N/A
10	3	Modification of MAIN sectors by non-secure thread	a) The *fxb_cmd_type* equals either PROGRAM (0x1), or ERASE (0x2)‡, and b) The *fxb_region_id* calculated from *fxb_sys_addr* matches MAIN (0x0) and c) Sector (or 8sectors) bit value that fxb_sys_addr decodes based on FPC_WEPROT_REG_x = 0 and d) FPC_FLSEMSTAT.ASSIGNED = 1 and e) FPC_FLSEMSTAT.SEC = 0 and sector (or 8sectors) bit value that fxb_sys_addr decodes based on FPC_SECATTRIB_REG_x is ‘0’ (non-secure) and f) Either FPC_FLSEMSTAT.PRIV = 1 and sector (or 8sectors) bit value that fxb_sys_addr decodes based on FPC_PRIVATTRIB_REG_x is i. ‘1’ (privilege) or ii. ‘0’ (non-privilege) and FPC_ATTRIBVIOLP_CONFIG.PRIVVIOL = 0 g) Or FPC_FLSEMSTAT.PRIV = 0 and sector (or 8sectors) bit value that fxb_sys_addr decodes based on FPC_PRIVATTRIB_REG_x is ‘0’ (non-privilege) h) Program/Erase is allowed to MAIN$^{\alpha}$ based on FlashRegionProtections	0 (no error, authorized)	N/A

Table 5-7. Authorization Error Signal (continued)

#	Priority	Brief Description	Condition	fxb_sys_addr_err	Fault/Error
11	3	Bank Erase of MAIN sectors	a) The *fxb_cmd_type* equals ERASE (0x2), and b) The *fxb_region_id* calculated from *fxb_sys_addr* matches MAIN (0x0), and c) The *fxb_cmd_size* is equal to BANK (0x5) Below are the conditions to be satisfied for successful erase of sectors. Authorization error is NOT to be generated if one or more below conditions below fail. a) FPC_FLSEMSTAT.ASSIGNED = 1 b) Sectors that are write protected through FPC_WEPROT_REG_x are not erased c) Sectors marked as secure are erased when FPC_FLSEMSTAT.SEC = 1 d) Sectors marked as non-secure are erased when i. Either FPC_FLSEMSTAT.SEC = 0 ii. Or FPC_FLSEMSTAT.SEC = 1 and FPC_ATTRIBVIOLS_CONFIG.SECVIOL = 0 e) Sectors marked as privilege are erased when FPC_FLSEMSTAT.PRIV = 1 f) Sectors marked as non-privilege are erased when i. Either FPC_FLSEMSTAT.PRIV = 0 ii. Or FPC_FLSEMSTAT.PRIV = 1 and FPC_ATTRIBVIOLP_CONFIG.PRIVVIOL = 0 g) Program/Erase is allowed to corresponding sector of NONMAIN based on FlashRegionProtections	0 (no error, authorized)	N/A
12	3	Modification of NONMAIN sectors by secure thread	a) The *fxb_cmd_type* equals either PROGRAM (0x1), or ERASE (0x2)‡, and b) The *fxb_region_id* calculated from *fxb_sys_addr* matches NONMAIN (0x1) and c) Sub-sector† bit value that fxb_sys_addr decodes based on FPC_NONMAIN_WEPROT_REG_x = 0 and d) FPC_FLSEMSTAT.ASSIGNED = 1 and e) FPC_FLSEMSTAT.SEC = 1 and sub-sector bit value that fxb_sys_addr decodes based on FPC_NONMAIN_SECATTRIB_REG_x is i. ‘1’ (secure) or ii. ‘0’ (non-secure) and FPC_ATTRIBVIOLS_CONFIG.SECVIOL = 0 and f) Either FPC_FLSEMSTAT.PRIV = 1 and sub-sector bit value that fxb_sys_addr decodes based on FPC_NONMAIN_PRIVATTRIB_REG_x is i. ‘1’ (privilege) or ii. ‘0’ (non-privilege) and FPC_ATTRIBVIOLP_CONFIG.PRIVVIOL = 0 g) Or FPC_FLSEMSTAT.PRIV = 0 and sub-sector bit value that fxb_sys_addr decodes based on FPC_NONMAIN_PRIVATTRIB_REG_x is ‘0’ (non-privilege) h) Program/Erase is allowed to corresponding sector of NONMAIN based on FlashRegionProtections	0 (no error, authorized)	N/A

Table 5-7. Authorization Error Signal (continued)

#	Priority	Brief Description	Condition	fxb_sys_addr_err	Fault/Error
13	3	Modification of NONMAIN sectors by non-secure thread	a) The *fxb_cmd_type* equals either PROGRAM (0x1), or ERASE (0x2)‡, and b) The *fxb_region_id* calculated from *fxb_sys_addr* matches NONMAIN (0x1) and c) Sub-sector bit value that fxb_sys_addr decodes based on FPC_NONMAIN_WEPROT_REG_x = 0 and d) FPC_FLSEMSTAT.ASSIGNED = 1 and e) FPC_FLSEMSTAT.SEC = 0 and sub-sector bit value that fxb_sys_addr decodes based on FPC_NONMAIN_SECATTRIB_REG_x is '0' (non-secure) and f) Either FPC_FLSEMSTAT.PRIV = 1 and sub-sector bit value that fxb_sys_addr decodes based on FPC_NONMAIN_PRIVATTRIB_REG_x is i. '1' (privilege) or ii. '0' (non-privilege) and FPC_ATTRIBVIOLP_CONFIG.PRIVVIOL = 0 g) Or FPC_FLSEMSTAT.PRIV = 0 and sub-sector bit value that fxb_sys_addr decodes based on FPC_NONMAIN_PRIVATTRIB_REG_x is '0' (non-privilege) h) Program/Erase is allowed to corresponding sector of NONMAIN based on FlashRegionProtections	0 (no error, authorized)	N/A
14	3	Read verify check from secure context	Following conditions must satisfy: a) The fxb_cmd_type equals READVERIFY (0x3) and b) The fxb_region_id calculated from fxb_sys_addr matches MAIN (0x0), and c) FPC_FLSEMSTAT.ASSIGNED = 1 and d) FPC_FLSEMSTAT.SEC = 1 and sector (or 8sectors) bit value that fxb_sys_addr decodes based on FPC_SECATTRIB_REG_x is i. 1' (secure) or ii. '0' (non-secure) and FPC_ATTRIBVIOLS_CONFIG.SECVIOL = 0 and e) Either FPC_FLSEMSTAT.PRIV = 1 and sector (or 8sectors) bit value that fxb_sys_addr decodes based on FPC_PRIVATTRIB_REG_x is i. '1' (privilege) or ii. '0' (non-privilege) and FPC_ATTRIBVIOLP_CONFIG.PRIVVIOL = 0 f) Or FPC_FLSEMSTAT.PRIV = 0 and sector (or 8sectors) bit value that fxb_sys_addr decodes based on FPC_PRIVATTRIB_REG_x is '0' (non-privilege) g) Read-Verify command is allowed to MAIN based on FlashRegionProtections (Prog permissions used)	0 (no error, authorized)	N/A

Table 5-7. Authorization Error Signal (continued)

#	Priority	Brief Description	Condition	fxb_sys_addr_err	Fault/Error
15	3	Read verify check from non-secure context	Following conditions must satisfy: a) The fxb_cmd_type equals READVERIFY (0x3) and b) The fxb_region_id calculated from fxb_sys_addr matches MAIN (0x0), and c) FPC_FLSEMSTAT.ASSIGNED = 1 and d) FPC_FLSEMSTAT.SEC = 0 and sector (or 8sectors) bit value that fxb_sys_addr decodes based on FPC_SECATTRIB_REG_x is '0' (non-secure) e) Either FPC_FLSEMSTAT.PRIV = 1 and sector (or 8sectors) bit value that fxb_sys_addr decodes based on FPC_PRIVATTRIB_REG_x is i. '1' (privilege) or ii. '0' (non-privilege) and FPC_ATTRIBVIOLP_CONFIG.PRIVVIOL = 0 f) Or FPC_FLSEMSTAT.PRIV = 0 and sector (or 8sectors) bit value that fxb_sys_addr decodes based on FPC_PRIVATTRIB_REG_x is '0' (non-privilege) g) Read-Verify command is allowed to MAIN based on FlashRegionProtections (Prog permissions used)	0 (no error, authorized)	N/A
16	3	Mode change to read mode always allowed	a) The *fxb_cmd_type* equals MODECHANGE (0x4), and *fxb_cmd_mode* is READ (0x0) b) FPC_FLSEMSTAT.ASSIGNED = 1	0 (no error, authorized)	N/A
17	3	Mode change to read margin modes only allowed by Secure thread	a) Following conditions must satisfy: a) The *fxb_cmd_type* equals MODECHANGE (0x4), and b) The *fxb_cmd_mode* is either: a. RDMARG0 (0x2), or b. RDMARG1 (0x4), or c. RDMARG0B (0x6), or d. RDMARG1B (0x7), and c) FPC_FLSEMSTAT.ASSIGNED = 1 and FPC_FLSEMSTAT.PRIV = 1 and FPC_FLSEMSTAT.SEC = 1 and	0 (no error, authorized)	N/A
18	4 (lowest)	Else	Else (default) – this ensures that any condition not explicitly given permission above will not be allowed	1 (error, not authorized)	Depends on *fxb_cmd_type*(ILLPROG, ILLERASE, ILLRDVER, ILLMODECH)

† Sub-sector refers to region of NONMAIN sector

‡ Bank erase (fxb_cmd_size = 0x5) is secured by the write/erase protect input signals

α Code corresponding to programming a HDP region is not to be kept after the code corresponding to setting of FPC_HDPPROT_CONTROL.HDPPROT[0/1]ACCEN.

Note

This authorization checks do not consider debug ownership (FLSEMSTAT.DBGACC) anywhere and authorization error is generated even if debugger makes an access. However, no error is sent to SEC in such cases.

5.4.3 FPI SEC Error Handling

The Global Security Control (GSC) module provides access control information for the FPI. With this information, the FPI grants access to write or erase secure/non-secure and privileged/unpriveleged. Un-granted write requests to controlled memory will results GSC SEC errors and sent out on the error interface, documented in Table 5-8.

Table 5-8. FPI SEC Error Interface

Error Name	Dir	Default	Description
FPI_SEC_ERROR	O	0	FPI error to SEC of GSC
FPI_SEC_ERROR_TYPE[7:0]	O	0	0x0: Reserved 0x1: ILLADDR 0x2: ILLPROG 0x4: ILLERASE 0x8: ILLRDVER 0x10: ILLMODECH 0x20: ILLCMD 0x40: ILLSIZE 0x80: FLC_MMR_ACCESS_ERROR Note: In case of FLCMMR access error and authorization error, FLC access error is reported on this bus.
FPI_SEC_ERROR_ADDRESS[31:0]	O	0	For an FLC error, the following bus from the appropriate FLC's FXB port is passed on: • ILLADDR/ILLPROG/ILLERASE/ILLRDVER/ILLMODECH/ ILLBANKERASE → fxb_sys_addr • ILLCMD → fxb_cmd_type; upper bits are zeroed • ILLSIZE → fxb_cmd_size; upper bits are zeroed

5.4.4 Bank Erase Protection

For each flash bank, the Flash Controller has a set of write/erase protection input signals, collectively called WEPROT signals. When a sector's associated WEPROT signal is active, the sector cannot be programmed or erased. Each bank has a 32-bit bus (*weprot_main_a0_mscbus*) where each signal corresponds to one of the first 32 sectors in the bank. Additionally, as many signals as are needed are provided to cover additional sectors in the bank, 8 sectors at a time (e.g., *weprot_main_b0_mscbus[31:4]*, *weprot_main_c0_mscbus[31:0]* – note that the first 4 in the 'b0' bus are skipped because those 32 sectors are covered by the 'a0' bus).

The WEPROT signals will only be used to protect bank erase, therefore the GSC bits that correspond to the WEPROT signals are called BEPROT. No error is thrown by the Flash Controller due to the WEPROT signals when attempted to erase without corresponding permissions. This is because some sectors will be allowed to be erased and others will not, according to the user's policies. The disallowed sectors should silently fail as it is expected behavior and the command correctly erasing the expected sectors should be rewarded with a pass.

Table 5-9. Flash Controller WEPROT Signal Mapping for a 512KB bank

Example Address †	Bank # (each 2 KB)	Sector # (each 2 KB)	Signal
0x1000.0000 – 0x1000.07FF	0	0	*weprot_main_a0_mscbus[0]*
0x1000.0800 – 0x1000.0FFF	0	1	*weprot_main_a0_mscbus[1]*
0x1000.1000 – 0x1000.17FF	0	2	*weprot_main_a0_mscbus[2]*
0x1000.1800 – 0x1000.1FFF	0	3	*weprot_main_a0_mscbus[3]*

Table 5-9. Flash Controller WEPROT Signal Mapping for a 512KB bank (continued)

Example Address †	Bank # (each 2 KB)	Sector # (each 2 KB)	Signal
…	..	..	..
0x1000.F800 – 0x1000.FFFF	0	31	*weprot_main_a0_mscbus[31]*
0x1001.0000 – 0x1001.3FFF	0	32-39	*weprot_main_b0_mscbus[4]*
0x1001.4000 – 0x1001.7FFF	0	40-47	*weprot_main_b0_mscbus[5]*
…	..	..	..
0x1007.C000 – 0x1007.FFFF	0	248-255	*weprot_main_b0_mscbus[31]*

Note

The BEPROT bits have to be made ‘0’ in Test mode

Note

NONMAIN bank erase is only allowed in Test mode

5.5 Flash Read Interface

The Flash Read Interface (FRI) 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, detection and reporting of ECC SEC, ECC DED, and GSC SEC errors.

5.5.1 Bank Address Swapping

Swapping of the MAIN regions of the banks within the address space 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 5-10 gives an example of the mapping before and after a bank swap is requested for a device with a 1024KB main flash split across 2 banks (512KB each).

Table 5-10. Bank Address Swap Translation

Bank and Region	Address Space Before Swap	Address Space After Swap
BANK0 MAIN	0x0000.0000 – 0x0007.FFFF	0x0008.0000 – 0x000F.FFFF
BANK1 MAIN	0x0008.0000 – 0x000F.FFFF	0x0000.0000 – 0x0007.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.

5.5.2 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.

5.5.2.1 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.

5.5.2.2 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.

5.5.3 GSC SEC Error Handling

The Global Security Control (GSC) module provides access control information for the FRI. With this information, the FRI grants access to secure/non-secure, privileged/unpriveleged, and hide protected memory. Un-granted read requests to controlled memory will results GSC SEC errors and sent out on the error interface.

The error interface contains the following information:

- Error pulse
- Error address
- Error type - 3 bit information
  - Bit 0 – Secure vs non-secure violation
  - Bit 1 – Privilege violation
  - Bit 2 – Hide Protection violation

5.6 FLASHCTL Registers

Table 5-11 lists the memory-mapped registers for the FLASHCTL registers. All register offset addresses not listed in Table 5-11 should be considered as reserved locations and the register contents should not be modified.

Table 5-11. FLASHCTL Registers

Offset	Acronym	Register Name	Section
11D0h	CMDWEPROTA	Command Write Erase Protect A Register	Section 5.6.1
11D4h	CMDWEPROTB	Command Write Erase Protect B Register	Section 5.6.2
1210h	CMDWEPROTNM	Command Write Erase Protect Non-Main Register	Section 5.6.3
13D0h	STATCMD	Command Status Register	Section 5.6.4

Complex bit access types are encoded to fit into small table cells. Table 5-12 shows the codes that are used for access types in this section.

Table 5-12. FLASHCTL 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

5.6.1 CMDWEPROTA Register (Offset = 11D0h) [Reset = 00000000h]

CMDWEPROTA is shown in Table 5-13.

Return to the Summary Table.

Command WriteErase Protect A Register 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.

Table 5-13. 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. 0h = Minimum value of [VAL] FFFFFFFFh = Maximum value of [VAL]

5.6.2 CMDWEPROTB Register (Offset = 11D4h) [Reset = 00000000h]

CMDWEPROTB is shown in Table 5-14.

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 multiple cases for how these protect bits are applied: 1. Single-bank system, CMDWEPROTA register present: The first 32 sectors are protected via the CMDWEPROTA register. Thus, the protection given by the bits in CMDWEPROTB begin with sector 32. 2. Single-bank system, CMDWEPROTA register not present: The protection given by the bits in CMDWEPROTB begin with sector 0. 3. Multi-bank system, CMDWEPROTA register present - Bank 0: 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. 4. Multi-bank system, CMDWEPROTA register present, 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. 5. Multi-bank system, CMDWEPROTA register not present: The bits in CMDWEPROTB will protect any of the 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.

Table 5-14. 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. 0h = Minimum value of [VAL] FFFFFFFFh = Maximum value of [VAL]

5.6.3 CMDWEPROTNM Register (Offset = 1210h) [Reset = 00000000h]

CMDWEPROTNM is shown in Table 5-15.

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.

Table 5-15. 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. 0h = Minimum value of [VAL] FFFFFFFFh = Maximum value of [VAL]

5.6.4 STATCMD Register (Offset = 13D0h) [Reset = 00000000h]

STATCMD is shown in Table 5-16.

Return to the Summary Table.

Command Status Register This register contains status regarding completion and errors of command execution.

Table 5-16. 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. 0h = No Fail 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. 0h = No Fail 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. 0h = No Fail 1h = Fail
6	FAILILLADDR	R	0h	Command failed due to the use of an illegal address 0h = No Fail 1h = Fail
5	FAILVERIFY	R	0h	Command failed due to verify error 0h = No Fail 1h = Fail
4	FAILWEPROT	R	0h	Command failed due to Write/Erase Protect Sector Violation 0h = No Fail 1h = Fail
3	RESERVED	R	0h	Reserved
2	CMDINPROGRESS	R	0h	Command In Progress 0h = Complete 1h = In Progress
1	CMDPASS	R	0h	Command Pass - valid when CMD_DONE field is 1 0h = Fail 1h = Pass
0	CMDDONE	R	0h	Command Done 0h = Not Done 1h = Done

5.7 FRI Registers

Table 5-17 lists the memory-mapped registers for the FRI registers. All register offset addresses not listed in Table 5-17 should be considered as reserved locations and the register contents should not be modified.

Table 5-17. FRI Registers

Offset	Acronym	Register Name	Section
1000h	FRDCNTL	Flash Read Control Register	Section 5.7.1
100Ch	FRD_INTF_CTRL	Flash Read Interface Control Register	Section 5.7.2
1010h	DTB_MUXSEL	Flash Read Interface DTB Mux select	Section 5.7.3
1100h	ECC_ENABLE	ECC Enable	Section 5.7.4
1104h	FECC_CTRL	ECC Control	Section 5.7.5

Complex bit access types are encoded to fit into small table cells. Table 5-18 shows the codes that are used for access types in this section.

Table 5-18. FRI 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

5.7.1 FRDCNTL Register (Offset = 1000h) [Reset = 02000201h]

FRDCNTL is shown in Table 5-19.

Return to the Summary Table.

Flash Read Control Register

Table 5-19. FRDCNTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	Reserved
27-24	RESERVED	R/W	0h	
23-12	RESERVED	R	0h	Reserved
11-8	RWAIT	R/W	2h	Random read waitstate These bits indicate how many waitstates are added to a flash read/fetch access. The RWAIT value can be set anywhere from 0 to 0xF. For a flash access, data is returned in RWAIT+1 SYSCLK cycles. Note: The required wait states for each SYSCLK frequency can be found in the device data manual. Reset type: SYSRSn
7-1	RESERVED	R	0h	Reserved
0	WS0_MODE	R/W	1h	When set, waitstate of the flash will be forced to 0 Waitstate. When this bit is set, RWAIT and TRIMENGRRWAIT values will be ignored. Reset type: SYSRSn

5.7.2 FRD_INTF_CTRL Register (Offset = 100Ch) [Reset = 00000000h]

FRD_INTF_CTRL is shown in Table 5-20.

Return to the Summary Table.

Flash Read Interface Control Register

Table 5-20. FRD_INTF_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-3	RESERVED	R	0h	Reserved
2	CODE_CACHE_EN	R/W	0h	Code cache enable. 0 A value of 0 disables the Code cache. 1 A value of 1 enables the Code cache. Reset type: SYSRSn
1	DATA_CACHE_EN	R/W	0h	Data cache enable. 0 A value of 0 disables the data cache. 1 A value of 1 enables the data cache. Reset type: SYSRSn
0	RESERVED	R/W	0h	

5.7.3 DTB_MUXSEL Register (Offset = 1010h) [Reset = 00000000h]

DTB_MUXSEL is shown in Table 5-21.

Return to the Summary Table.

Flash Read Interface DTB Mux select

Table 5-21. DTB_MUXSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1-0	DTB_MUX_SEL_FRI	R/W	0h	DTB Mux Select for Flash read interface signals to be exported out on DTB bus. Based on the MUXSELECT value, corresponding banks' DTB signals will be exported out. 00 - Bank0 01 - Bank1 10 - Bank2 11 - Reserved Reset type: SYSRSn

5.7.4 ECC_ENABLE Register (Offset = 1100h) [Reset = 0000000Ah]

ECC_ENABLE is shown in Table 5-22.

Return to the Summary Table.

ECC Enable

Table 5-22. ECC_ENABLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-4	RESERVED	R	0h	Reserved
3-0	ENABLE	R/W	Ah	ECC enable. A value of 0xA would enable ECC. Any other value would disable ECC. Reset type: SYSRSn

5.7.5 FECC_CTRL Register (Offset = 1104h) [Reset = 00000000h]

FECC_CTRL is shown in Table 5-23.

Return to the Summary Table.

ECC Control

Table 5-23. FECC_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-2	RESERVED	R	0h	Reserved
1-0	ECC_TEST_EN	R/W	0h	ECC test mode enable. 00 ECC test mode disabled 01 ECC test mode enabled, one of the 64 data bits is flipped and fed to the redundant ECC logic (on both ECC logic low and ECC logc high blocks). 11 ECC test mode enabled, Two of the 64 data bits are flipped and fed to the redundant ECC logic (on both ECC logic low and ECC logc high blocks). 10 Reserved Reset type: SYSRSn

Chapter 6

EAM

The Error Aggregator Module (EAM) provides a log of errors related to memory access and Error Correction Code (ECC) failures in the device.

6.1 EAM Introduction 328
6.2 EAM Operation 328
6.3 EAM Registers 330

6.1 EAM Introduction

The EAM module's purpose is to aggregate safety and security errors from multiple sources. The EAM then stores these errors in the Memory Mapped Registers (MMR's). Each group has their own set of fixed priority which is defined in the Section 6.2.

6.2 EAM Operation

Each aggregator stores the errors from various sources with their own fixed priority. The fixed priority is determined by the fixed priority arbiter. The fixed priority table uses 1 as the highest priority and the largest number being the lowest priority. See the corresponding error aggregator on details on the corresponding fixed priority arbiter. Information on which address the error occurred, corresponding error type and error flag are stored in the respective register.

When an error occurs the following steps are taken:

1. Error source is selected based on the fixed priority arbiter
2. Error flag bit, address, and id are latched into respective registers corresponding to the selected error
3. Selected error will be forwarded to SYSCTL generating an NMI or interrupt
4. Wait for software to write to the clear register
5. After the clear register is written to the error type, flag, and address will also be cleared

During step 4 the corresponding clear register must be written to by software for the next error to properly propagate. To see the corresponding register to clear please see *EAM Registers*.

6.2.1 Security Error Aggregator

The security error aggregator is used to handle the security errors generated from SRAM (SYSMEM), Flash Read Interface (FRI), Flash Program Interface (FPI), CPU, DMA0, and the Global Security Controller (GSC). These modules all generate different types of security errors that are propagated to the security error aggregator. After receiving these errors, the security error aggregator generates an security error non-maskable interrupt (NMI). Detailed information about the NMI can be seen in the SYSCTL Events section.

When one of the security errors occur, the security fixed priority arbiter selects the error source based on the Table 6-1. After the security fixed priority arbiter selects an error, the corresponding error is stored in the SECURITY_ERR_FLAG register. Then the error is forwarded to SYSCTL and causes a security error NMI. This transaction can be seen in the Figure 6-1.

Table 6-1. Security Error Aggregator's Fixed Priority table

Priority	Source
1 (Highest Priority)	MEM0
2	MEM1
3	MEM2
4	MEM3
5	FRI
6	FPI
7	CPU
8	DMA0
10	GSC

Figure 6-1. Security Error Aggregator Block Diagram

To properly understand the error that has occurred, information about the error can be found in the SECURITY_ERR_MSTID and SECURITY_ERR_ADDR registers. To clear the error and propagate the next error, the SECURITY_ERR_CLR register must be written to. Please see the *EAM Registers* section for a more detailed description on these registers.

6.2.2 Safety Error Aggregator

The safety error aggregator is the other error aggregator in the EAM. This error aggregator handles multiple safety errors caused by ECC failures.When the safety error aggregator receives an ECC error the module causes a NMI or an interrupt depending on the error. Information on the different types of errors can be seen in the SYSCTL Events section. These ECC errors can occur from the Flash Read Interface (FRI) or the SRAM memory (SYSMEM). Information on the ECC error can be seen in the ECC Error Handling section of the flash chapter and information on the SRAM errors can be seen in the SRAM region section of the Architecture chapter. The key difference is that the safety error aggregator can generate NMIs as well as interrupts to handle the different types of ECC errors. See Table 6-2 for details on which interrupts are generated from the corresponding error.

Table 6-2. Safety Error Aggregator NMI and Interrupt table

Error	Resulting Interrupt
Flash Read ECC SEC	FLASHSEC Interrupt
Flash Read ECC DED	FLASHDED NMI
SRAM MEM0 - SRAM MEM3 Parity error	SRAMPARITY NMI

Due to the single bit errors (SEC) and dual bit errors (DED) generating different types of interrupts there are 4 sets of MMR's for this error aggregator: FRI_SEC, FRI_DED, . Each one of these groups of MMR's has an individual set of MSTID, ADDR, FLAG, and CLR registers. To see detailed descriptions on these registers please see *EAM Registers*.

For a visual representation of the safety error aggregator module, please see to find the configuration of the source selectors, fixed priority arbiters, and segregation module. The fixed priority arbiter is not needed for the flash read interface as there is only one type of flash error from the SEC and DED interfaces. To see the different priority levels please see table Table 6-3.

Table 6-3. Safety Error Aggregator's Fixed Priority table

Priority	Source
1 (Highest Priority)	MEM0
2	MEM1
3	MEM2
4	MEM3

6.3 EAM Registers

Table 6-4 lists the memory-mapped registers for the EAM registers. All register offset addresses not listed in Table 6-4 should be considered as reserved locations and the register contents should not be modified.

Table 6-4. EAM Registers

Offset	Acronym	Register Name	Section
1000h	REVISION	IP revision id register	Section 6.3.1
1010h	SECURITY_ERR_FLAG	Security error flag	Section 6.3.2
1014h	SECURITY_ERR_CLR	Security error clear	Section 6.3.3
1018h	SECURITY_ERR_MSTID	Security error master id	Section 6.3.4
101Ch	SECURITY_ERR_ADDR	Security error address	Section 6.3.5
1030h	FRI_SEC_FLAG	FRI SEC flag	Section 6.3.6
1034h	FRI_SEC_CLR	FRI SEC clear	Section 6.3.7
1038h	FRI_SEC_ADDR	FRI SEC address	Section 6.3.8
103Ch	FRI_SEC_MSTID	FRI SEC master id	Section 6.3.9
1050h	FRI_DED_FLAG	FRI DED flag	Section 6.3.10
1054h	FRI_DED_CLR	FRI DED clear	Section 6.3.11
1058h	FRI_DED_ADDR	FRI DED address	Section 6.3.12
105Ch	FRI_DED_MSTID	FRI DED master id	Section 6.3.13
1070h	SYSMEM_SEC_FLAG	SYSMEM SEC flag	Section 6.3.14
1074h	SYSMEM_SEC_CLR	SYSMEM SEC clear	Section 6.3.15
1078h	SYSMEM_SEC_ADDR	SYSMEM SEC address	Section 6.3.16
107Ch	SYSMEM_SEC_MSTID	SYSMEM SEC master id	Section 6.3.17
1090h	SYSMEM_DED_FLAG	SYSMEM DED flag	Section 6.3.18
1094h	SYSMEM_DED_CLR	SYSMEM DED clear	Section 6.3.19
1098h	SYSMEM_DED_ADDR	SYSMEM DED address	Section 6.3.20
109Ch	SYSMEM_DED_MSTID	SYSMEM DED master id	Section 6.3.21

Complex bit access types are encoded to fit into small table cells. Table 6-5 shows the codes that are used for access types in this section.

Table 6-5. EAM Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W1C	W 1C	Write 1 to clear
Reset or Default Value		
-n		Value after reset or the default value

6.3.1 REVISION Register (Offset = 1000h) [Reset = 00000000h]

REVISION is shown in Table 6-6.

Return to the Summary Table.

IP revision id register

Table 6-6. REVISION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R-0	0h	Reserved
0	REVISION	R	0h	Revision register Reset type: PORESETn

6.3.2 SECURITY_ERR_FLAG Register (Offset = 1010h) [Reset = 00000000h]

SECURITY_ERR_FLAG is shown in Table 6-7.

Return to the Summary Table.

Security error flag

Table 6-7. SECURITY_ERR_FLAG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R-0	0h	Reserved
11	FLSEM_ACCESS_ERROR	R	0h	FLSEM ACCESS ERROR flag Reset type: PORESETn
10	FLC_MMR_ACCESS_ERROR	R	0h	FLC MMR ACCESS ERROR flag Reset type: PORESETn
9	FPI_ILLSIZE	R	0h	FPI ILLSIZE flag Reset type: PORESETn
8	FPI_ILLCMD	R	0h	FPI ILLCMD flag Reset type: PORESETn
7	FPI_ILLMODECH	R	0h	FPI ILLMODECH flag Reset type: PORESETn
6	FPI_ILLRDVER	R	0h	FPI ILLRDVER flag Reset type: PORESETn
5	FPI_ILLERASE	R	0h	FPI ILLERASE flag Reset type: PORESETn
4	FPI_ILLPROG	R	0h	FPI ILLPROG flag Reset type: PORESETn
3	FPI_ILLADDR	R	0h	FPI ILLADDR flag Reset type: PORESETn
2	HDP	R	0h	Hide protection error flag Reset type: PORESETn
1	PRIV	R	0h	Privilege error flag Reset type: PORESETn
0	SECURE	R	0h	Secure error flag Reset type: PORESETn

6.3.3 SECURITY_ERR_CLR Register (Offset = 1014h) [Reset = 00000000h]

SECURITY_ERR_CLR is shown in Table 6-8.

Return to the Summary Table.

Security error clear

Table 6-8. SECURITY_ERR_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R-0	0h	Reserved
11	FLSEM_ACCESS_ERROR	R-0/W1C	0h	writing '1' will clear SECURITY_ERR_FLAG[FLSEM_ACCESS_ERROR] register. Reset type: PORESETn
10	FLC_MMR_ACCESS_ERROR	R-0/W1C	0h	writing '1' will clear SECURITY_ERR_FLAG[FLC_MMR_ACCESS_ERROR] register. Reset type: PORESETn
9	FPI_ILLSIZE	R-0/W1C	0h	writing '1' will clear SECURITY_ERR_FLAG[FPI_ILLSIZE] register. Reset type: PORESETn
8	FPI_ILLCMD	R-0/W1C	0h	writing '1' will clear SECURITY_ERR_FLAG[FPI_ILLCMD] register. Reset type: PORESETn
7	FPI_ILLMODECH	R-0/W1C	0h	writing '1' will clear SECURITY_ERR_FLAG[FPI_ILLMODECH] register. Reset type: PORESETn
6	FPI_ILLRDVER	R-0/W1C	0h	writing '1' will clear SECURITY_ERR_FLAG[FPI_ILLRDVER] register. Reset type: PORESETn
5	FPI_ILLERASE	R-0/W1C	0h	writing '1' will clear SECURITY_ERR_FLAG[FPI_ILLERASE] register. Reset type: PORESETn
4	FPI_ILLPROG	R-0/W1C	0h	writing '1' will clear SECURITY_ERR_FLAG[FPI_ILLPROG] register. Reset type: PORESETn
3	FPI_ILLADDR	R-0/W1C	0h	writing '1' will clear SECURITY_ERR_FLAG[FPI_ILLADDR] register. Reset type: PORESETn
2	HDP	R-0/W1C	0h	writing '1' will clear SECURITY_ERR_FLAG[HDP] register. Reset type: PORESETn
1	PRIV	R-0/W1C	0h	writing '1' will clear SECURITY_ERR_FLAG[PRIV] register. Reset type: PORESETn
0	SECURE	R-0/W1C	0h	writing '1' will clear SECURITY_ERR_FLAG[SEC] register. Reset type: PORESETn

6.3.4 SECURITY_ERR_MSTID Register (Offset = 1018h) [Reset = 00000000h]

SECURITY_ERR_MSTID is shown in Table 6-9.

Return to the Summary Table.

Security error master id

Table 6-9. SECURITY_ERR_MSTID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7-0	MSTID	R	0h	Security error master ID. This field will be cleared along with the corresponding flag clear. Reset type: PORESETn

6.3.5 SECURITY_ERR_ADDR Register (Offset = 101Ch) [Reset = 00000000h]

SECURITY_ERR_ADDR is shown in Table 6-10.

Return to the Summary Table.

Security error address

Table 6-10. SECURITY_ERR_ADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDRESS	R	0h	Security error address. This field will be cleared along with the corresponding flag clear. Reset type: PORESETn

6.3.6 FRI_SEC_FLAG Register (Offset = 1030h) [Reset = 00000000h]

FRI_SEC_FLAG is shown in Table 6-11.

Return to the Summary Table.

FRI single error correction flag

Table 6-11. FRI_SEC_FLAG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R-0	0h	Reserved
0	SEC	R	0h	Single error correction error Reset type: PORESETn

6.3.7 FRI_SEC_CLR Register (Offset = 1034h) [Reset = 00000000h]

FRI_SEC_CLR is shown in Table 6-12.

Return to the Summary Table.

FRI single error correction clear

Table 6-12. FRI_SEC_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R-0	0h	Reserved
0	SEC	R-0/W1C	0h	writing '1' will clear FRI_SEC_FLAG[SEC] register. Reset type: PORESETn

6.3.8 FRI_SEC_ADDR Register (Offset = 1038h) [Reset = 00000000h]

FRI_SEC_ADDR is shown in Table 6-13.

Return to the Summary Table.

FRI single error correction address

Table 6-13. FRI_SEC_ADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDRESS	R	0h	Single error correction error address. This field will be cleared along with the corresponding flag clear. Reset type: PORESETn

6.3.9 FRI_SEC_MSTID Register (Offset = 103Ch) [Reset = 00000000h]

FRI_SEC_MSTID is shown in Table 6-14.

Return to the Summary Table.

FRI single error correction master id

Table 6-14. FRI_SEC_MSTID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7-0	MSTID	R	0h	Single error correction error master ID. This field will be cleared along with the corresponding flag clear. Reset type: PORESETn

6.3.10 FRI_DED_FLAG Register (Offset = 1050h) [Reset = 00000000h]

FRI_DED_FLAG is shown in Table 6-15.

Return to the Summary Table.

FRI double error detection flag

Table 6-15. FRI_DED_FLAG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1	DIAG_DED	R	0h	diagnostic Double error detect error Reset type: PORESETn
0	DED	R	0h	Double error detect error Reset type: PORESETn

6.3.11 FRI_DED_CLR Register (Offset = 1054h) [Reset = 00000000h]

FRI_DED_CLR is shown in Table 6-16.

Return to the Summary Table.

FRI double error detection clear

Table 6-16. FRI_DED_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1	DIAG_DED	R-0/W1C	0h	writing '1' will clear FRI_DED_FLAG[DIAG_DED] register. Reset type: PORESETn
0	DED	R-0/W1C	0h	writing '1' will clear FRI_DED_FLAG[DED] register. Reset type: PORESETn

6.3.12 FRI_DED_ADDR Register (Offset = 1058h) [Reset = 00000000h]

FRI_DED_ADDR is shown in Table 6-17.

Return to the Summary Table.

FRI double error detection address

Table 6-17. FRI_DED_ADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDRESS	R	0h	Double error detection error address. This field will be cleared along with the corresponding flag clear. Reset type: PORESETn

6.3.13 FRI_DED_MSTID Register (Offset = 105Ch) [Reset = 00000000h]

FRI_DED_MSTID is shown in Table 6-18.

Return to the Summary Table.

FRI double error detection master id

Table 6-18. FRI_DED_MSTID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7-0	MSTID	R	0h	Double error detection error master ID. This field will be cleared along with the corresponding flag clear. Reset type: PORESETn

6.3.14 SYSMEM_SEC_FLAG Register (Offset = 1070h) [Reset = 00000000h]

SYSMEM_SEC_FLAG is shown in Table 6-19.

Return to the Summary Table.

SYSMEM single error correction flag

Table 6-19. SYSMEM_SEC_FLAG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1	WR_SEC	R	0h	diagnostic single error correction error during write Reset type: PORESETn
0	RD_SEC	R	0h	Single error correction error during read Reset type: PORESETn

6.3.15 SYSMEM_SEC_CLR Register (Offset = 1074h) [Reset = 00000000h]

SYSMEM_SEC_CLR is shown in Table 6-20.

Return to the Summary Table.

SYSMEM single error correction clear

Table 6-20. SYSMEM_SEC_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1	WR_SEC	R-0/W1C	0h	writing '1' will clear SYSMEM_SEC_FLAG[WR_SEC] register. Reset type: PORESETn
0	RD_SEC	R-0/W1C	0h	writing '1' will clear SYSMEM_SEC_FLAG[RD_SEC] register. Reset type: PORESETn

6.3.16 SYSMEM_SEC_ADDR Register (Offset = 1078h) [Reset = 00000000h]

SYSMEM_SEC_ADDR is shown in Table 6-21.

Return to the Summary Table.

SYSMEM single error correction address

Table 6-21. SYSMEM_SEC_ADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDRESS	R	0h	Single error correction error address. This field will be cleared along with the corresponding flag clear. Reset type: PORESETn

6.3.17 SYSMEM_SEC_MSTID Register (Offset = 107Ch) [Reset = 00000000h]

SYSMEM_SEC_MSTID is shown in Table 6-22.

Return to the Summary Table.

SYSMEM single error correction master id

Table 6-22. SYSMEM_SEC_MSTID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7-0	MSTID	R	0h	Single error correction error master ID. This field will be cleared along with the corresponding flag clear. Reset type: PORESETn

6.3.18 SYSMEM_DED_FLAG Register (Offset = 1090h) [Reset = 00000000h]

SYSMEM_DED_FLAG is shown in Table 6-23.

Return to the Summary Table.

SYSMEM double error detection flag

Table 6-23. SYSMEM_DED_FLAG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1	WR_DED	R	0h	diagnostic Double error detect error during write Reset type: PORESETn
0	RD_DED	R	0h	Double error detect error during read Reset type: PORESETn

6.3.19 SYSMEM_DED_CLR Register (Offset = 1094h) [Reset = 00000000h]

SYSMEM_DED_CLR is shown in Table 6-24.

Return to the Summary Table.

SYSMEM double error detection clear

Table 6-24. SYSMEM_DED_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1	WR_DED	R-0/W1C	0h	writing '1' will clear SYSMEM_DED_FLAG[WR_DED] register. Reset type: PORESETn
0	RD_DED	R-0/W1C	0h	writing '1' will clear SYSMEM_DED_FLAG[RD_DED] register. Reset type: PORESETn

6.3.20 SYSMEM_DED_ADDR Register (Offset = 1098h) [Reset = 00000000h]

SYSMEM_DED_ADDR is shown in Table 6-25.

Return to the Summary Table.

SYSMEM double error detection address

Table 6-25. SYSMEM_DED_ADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDRESS	R	0h	Double error detection error address. This field will be cleared along with the corresponding flag clear. Reset type: PORESETn

6.3.21 SYSMEM_DED_MSTID Register (Offset = 109Ch) [Reset = 00000000h]

SYSMEM_DED_MSTID is shown in Table 6-26.

Return to the Summary Table.

SYSMEM double error detection master id

Table 6-26. SYSMEM_DED_MSTID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7-0	MSTID	R	0h	Double error detection error master ID. This field will be cleared along with the corresponding flag clear. Reset type: PORESETn

Chapter 7

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.

7.1 DMA Overview 353
7.2 DMA Operation 354
7.3 DMA Registers 365

7.1 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 7-1.

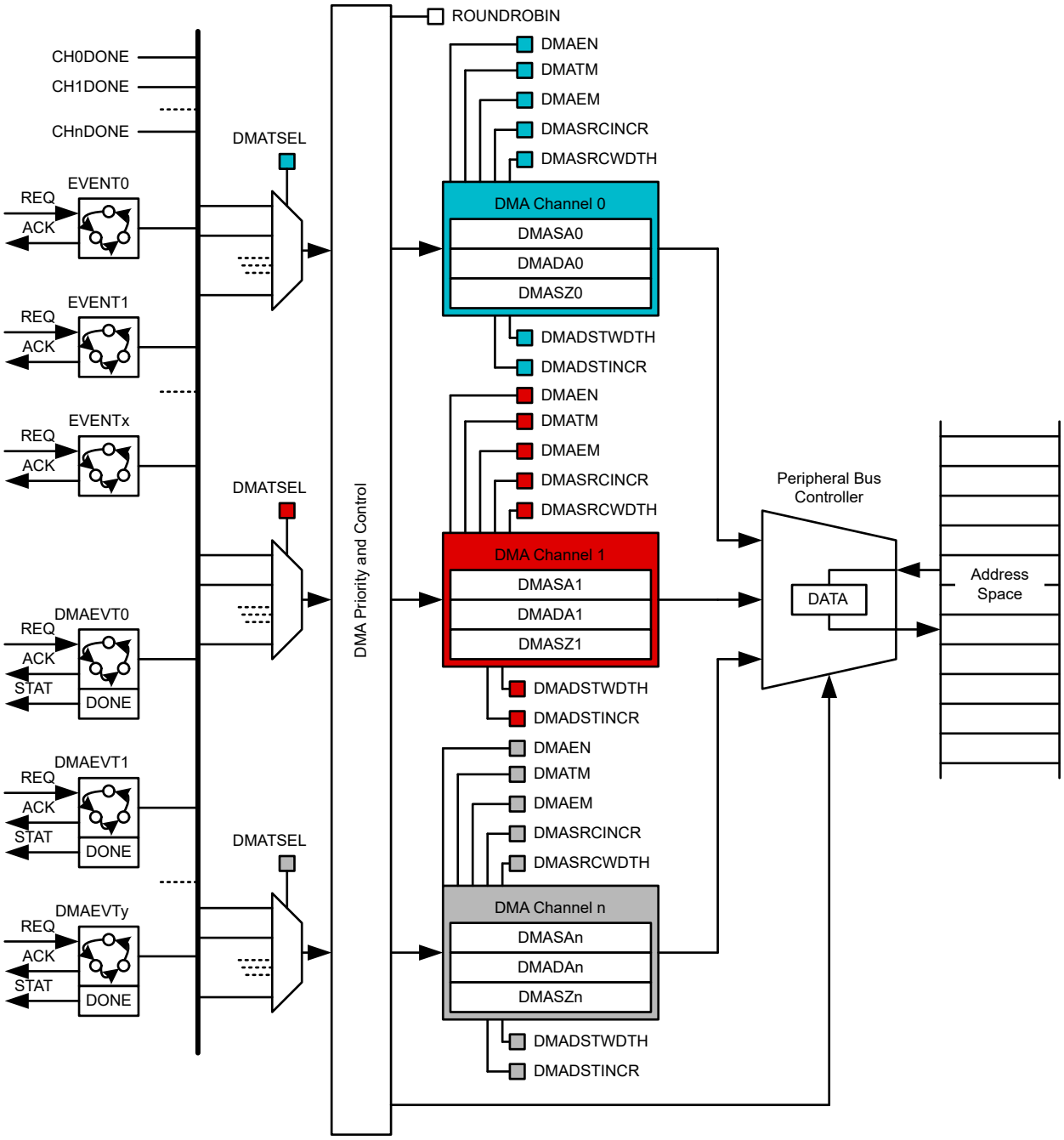


Figure 7-1. DMA Block Diagram

7.2 DMA Operation

The DMA controller is configured with user software. The setup and operation of the DMA is discussed in the following sections.

7.2.1 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:

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 7.2.4.1, Section 7.2.4.2 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.

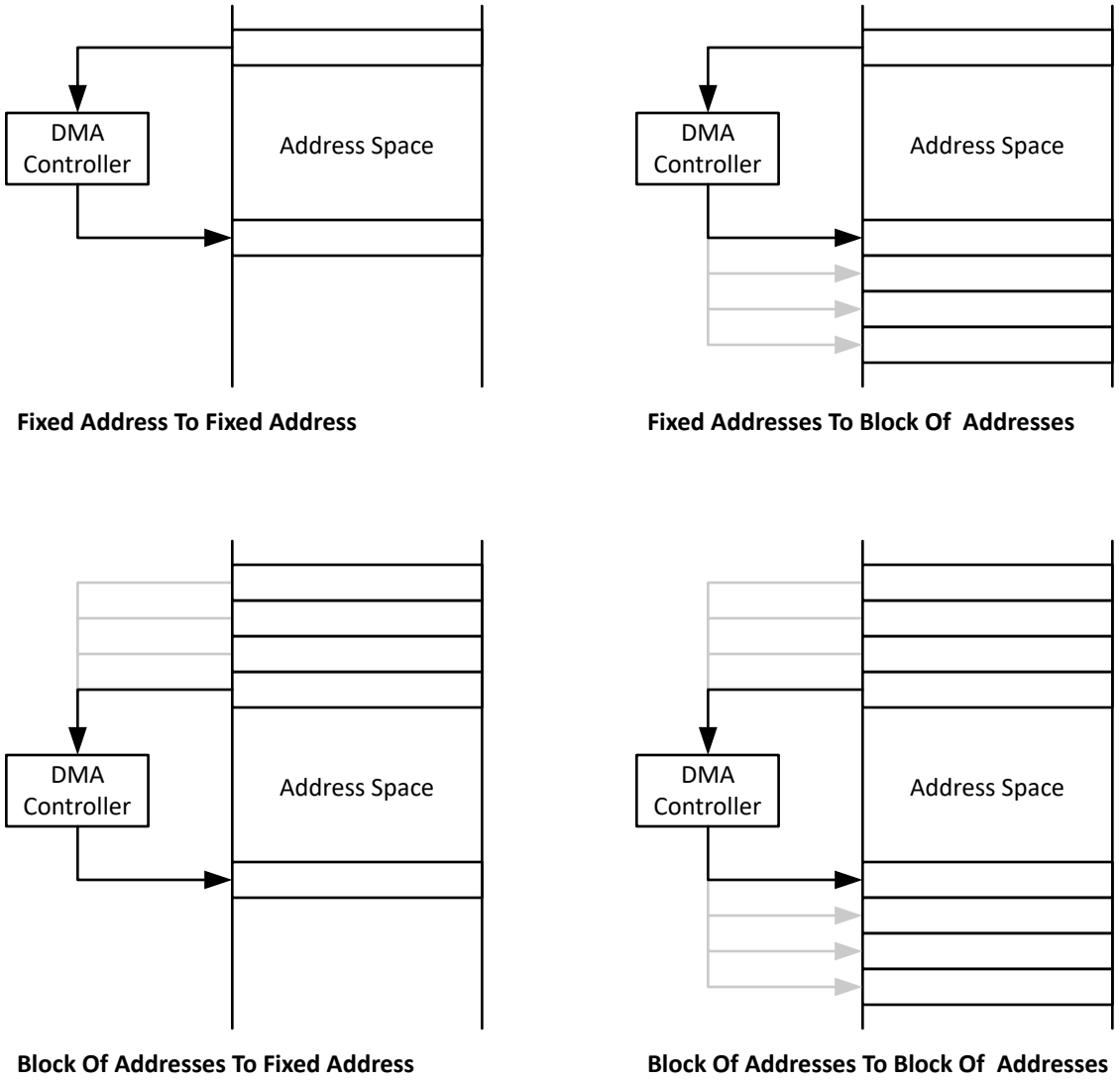


Figure 7-2. Basic DMA Addressing Modes

7.2.2 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.

Table 7-1. 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	–	–	✓	✓

7.2.3 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 7-2. 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

7.2.3.1 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.

The DMAEN bit is cleared automatically when DMASZ decrements to zero and must be set again for another transfer to occur.

7.2.3.2 Block Transfer

In block transfer mode (DMATM = 1), a transfer of a complete block of data occurs after one trigger. Block transfer mode is available in basic DMA channels only.

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.

7.2.3.3 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).

7.2.3.4 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.

7.2.3.5 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.

7.2.4 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 7-3. 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

Table 7-3. DMA Extended Modes (continued)

DMAEM	Extended Mode	Description
3	Table mode	Used to help configure a table of peripheral control registers

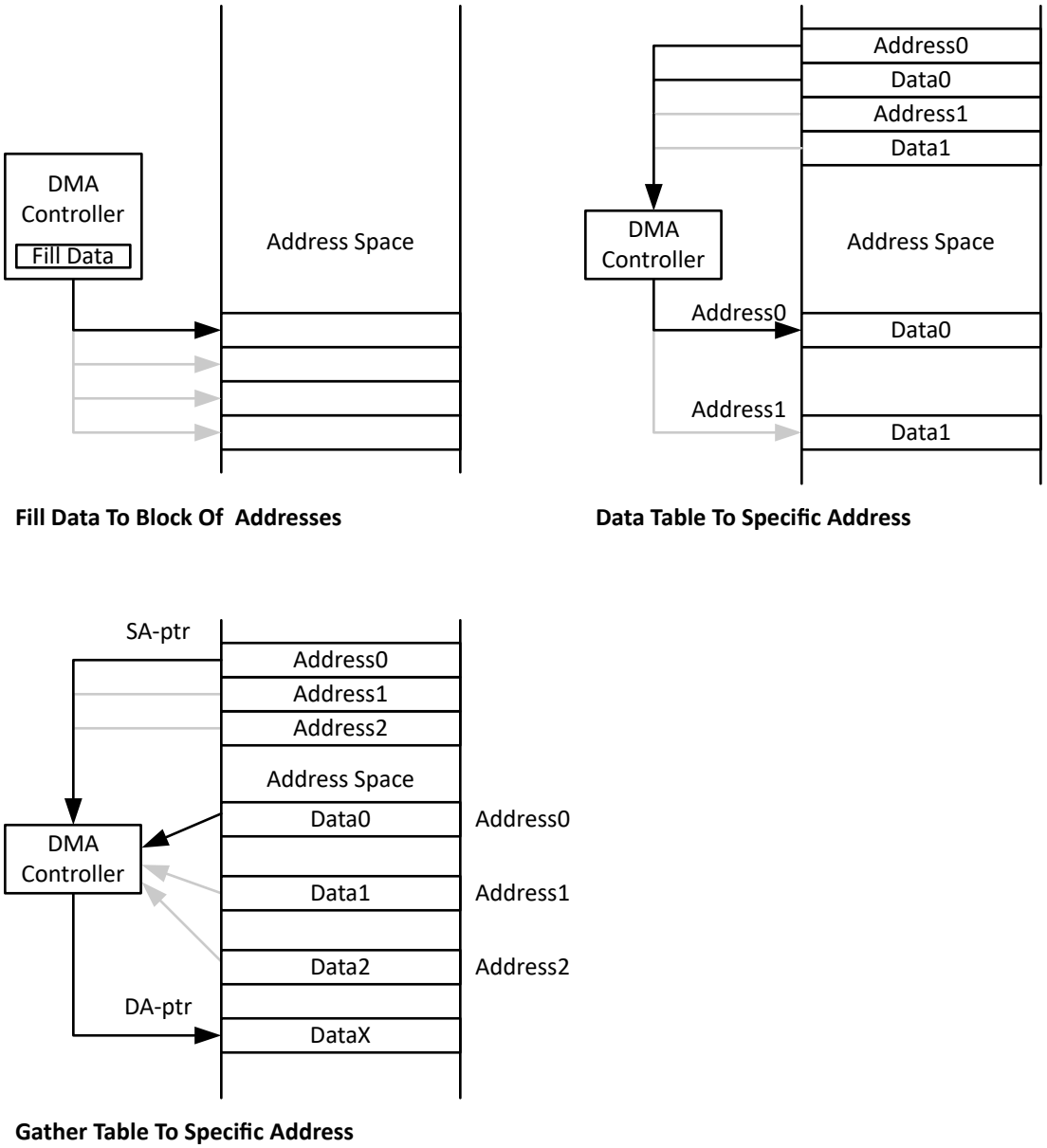


Figure 7-3. DMA Extended Modes Diagram

7.2.4.1 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 7-4 for how to use DMASRCWDTH in fill mode.

Table 7-4. 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.

7.2.4.2 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 7-5 is an example of a table in memory compatible with the DMA table mode.

Table 7-5. Example of an Incremental Table Compatible with DMA Table Mode

Table Address	Table Data
0x0000	Address 0
0x0004	Data 0
0x0008	Address 1
0x000C	Data 1

7.2.5 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.

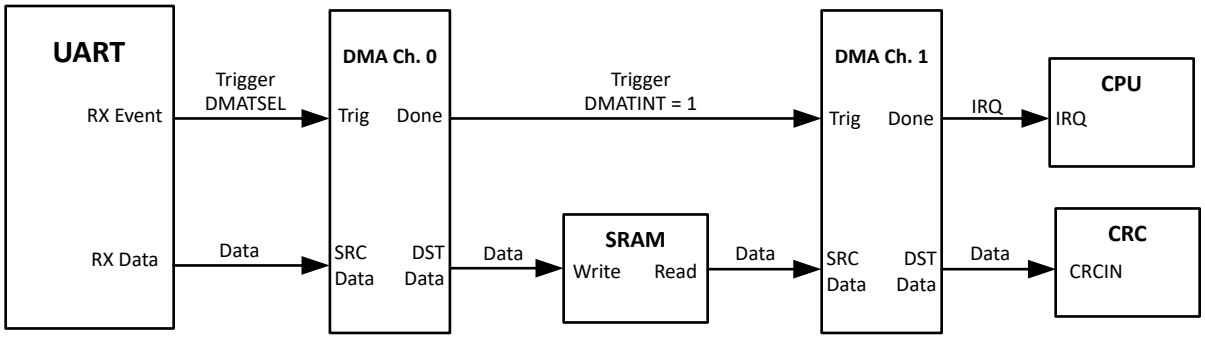


Figure 7-4. DMA Cascading Channels

7.2.6 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.

7.2.7 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 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 7-6. 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

7.2.8 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.

7.2.9 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.

7.2.10 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.

7.2.11 DMA Trigger Event Status

The DMA controller supports dedicated DMA events. See Section 8.2.2 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.

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).

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.

7.2.12 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.

7.2.12.1 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.

7.2.12.2 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.

7.2.13 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 nonexisting memory range. If an address error occurs, the interrupt index IIDX[j].STAT flags a DMA address error (11h). Address error interrupts can be masked, set, and cleared using the ADDERR bit.

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. If a protected address error or a data error occurs, the EAM registers will reflect an ECC error.

7.2.14 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 7-7.

Table 7-7. 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

7.3 DMA Registers

Table 7-8 lists the memory-mapped registers for the DMA registers. All register offset addresses not listed in Table 7-8 should be considered as reserved locations and the register contents should not be modified.

Table 7-8. 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 7-9 shows the codes that are used for access types in this section.

Table 7-9. 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		

Table 7-9. 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.

7.3.1 FSUB_0 (Offset = 400h) [Reset = 00000000h]

FSUB_0 is shown in Figure 7-5 and described in Table 7-10.

Return to the Summary Table.

Subscriber port

Figure 7-5. 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 7-10. FSUB_0 Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	CHANID	R/W	0h	0 = disconnected. 1-255 = connected to channelID = CHANID. 0h = A value of 0 specifies that the event is not connected FFh = Consult your device datasheet as the actual allowed maximum may be less than 255.

7.3.2 FSUB_1 (Offset = 404h) [Reset = 00000000h]

FSUB_1 is shown in Figure 7-6 and described in Table 7-11.

Return to the Summary Table.

Subscriber port

Figure 7-6. 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 7-11. FSUB_1 Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	CHANID	R/W	0h	0 = disconnected. 1-255 = connected to channelID = CHANID. 0h = A value of 0 specifies that the event is not connected FFh = Consult your device datasheet as the actual allowed maximum may be less than 255.

7.3.3 FPUB_1 (Offset = 444h) [Reset = 00000000h]

FPUB_1 is shown in Figure 7-7 and described in Table 7-12.

Return to the Summary Table.

Publisher port

Figure 7-7. 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 7-12. FPUB_1 Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	CHANID	R/W	0h	0 = disconnected. 1-255 = connected to channelID = CHANID. 0h = A value of 0 specifies that the event is not connected FFh = Consult your device datasheet as the actual allowed maximum may be less than 255.

7.3.4 PDBGCTL (Offset = 1018h) [Reset = 00000000h]

PDBGCTL is shown in Figure 7-8 and described in Table 7-13.

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 7-8. 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 7-13. 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' 0h = The peripheral will halt immediately, even if the resultant state will result in corruption if the system is restarted 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 0h = The peripheral freezes functionality while the Core Halted input is asserted and resumes when it is deasserted. 1h = The peripheral ignores the state of the Core Halted input

7.3.5 IIDX (Offset = 1020h) [Reset = 00000000h]

IIDX is shown in Figure 7-9 and described in Table 7-14.

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 7-9. 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 7-14. IIDX Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	STAT	R	0h	Interrupt index status 00h = No bit is set means there is no pending interrupt request 01h = DMA Channel 0 size counter reached zero (DMASZ=0). 02h = DMA Channel 1 size counter reached zero (DMASZ=0). 03h = DMA Channel 2 size counter reached zero (DMASZ=0). 04h = DMA Channel 3 size counter reached zero (DMASZ=0). 05h = DMA Channel 4 size counter reached zero (DMASZ=0). 06h = DMA Channel 5 size counter reached zero (DMASZ=0). 07h = DMA Channel 6 size counter reached zero (DMASZ=0). 08h = DMA Channel 7 size counter reached zero (DMASZ=0). 09h = DMA Channel 8 size counter reached zero (DMASZ=0). 0Ah = DMA Channel 9 size counter reached zero (DMASZ=0). 0Bh = DMA Channel 10 size counter reached zero (DMASZ=0). 0Ch = DMA Channel 11 size counter reached zero (DMASZ=0). 0Dh = DMA Channel 12 size counter reached zero (DMASZ=0). 0Eh = DMA Channel 13 size counter reached zero (DMASZ=0). 0Fh = DMA Channel 14 size counter reached zero (DMASZ=0). 10h = DMA Channel 15 size counter reached zero (DMASZ=0). 11h = PRE-IRQ event for DMA Channel 0. 12h = PRE-IRQ event for DMA Channel 1. 13h = PRE-IRQ event for DMA Channel 2. 14h = PRE-IRQ event for DMA Channel 3. 15h = PRE-IRQ event for DMA Channel 4. 16h = PRE-IRQ event for DMA Channel 5. 17h = PRE-IRQ event for DMA Channel 6. 18h = PRE-IRQ event for DMA Channel 7. 19h = DMA address error, SRC address not reachable. 1Ah = DMA data error, SRC data might be corrupted (PAR or ECC error).

7.3.6 IMASK (Offset = 1028h) [Reset = 00000000h]

IMASK is shown in Figure 7-10 and described in Table 7-15.

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 7-10. 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 7-15. 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). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
24	ADDRERR	R/W	0h	DMA address error, SRC address not reachable. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
23	PREIRQCH7	R/W	0h	Pre-IRQ for Channel 7. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
22	PREIRQCH6	R/W	0h	Pre-IRQ for Channel 6. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
21	PREIRQCH5	R/W	0h	Pre-IRQ for Channel 5. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
20	PREIRQCH4	R/W	0h	Pre-IRQ for Channel 4. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
19	PREIRQCH3	R/W	0h	Pre-IRQ for Channel 3. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
18	PREIRQCH2	R/W	0h	Pre-IRQ for Channel 2. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
17	PREIRQCH1	R/W	0h	Pre-IRQ for Channel 1. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit

Table 7-15. 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. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
15	DMACH15	R/W	0h	DMA Channel 15 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
14	DMACH14	R/W	0h	DMA Channel 14 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
13	DMACH13	R/W	0h	DMA Channel 13 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
12	DMACH12	R/W	0h	DMA Channel 12 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
11	DMACH11	R/W	0h	DMA Channel 11 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
10	DMACH10	R/W	0h	DMA Channel 10 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
9	DMACH9	R/W	0h	DMA Channel 9 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
8	DMACH8	R/W	0h	DMA Channel 8 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
7	DMACH7	R/W	0h	DMA Channel 7 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
6	DMACH6	R/W	0h	DMA Channel 6 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
5	DMACH5	R/W	0h	DMA Channel 5 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
4	DMACH4	R/W	0h	DMA Channel 4 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
3	DMACH3	R/W	0h	DMA Channel 3 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
2	DMACH2	R/W	0h	DMA Channel 2 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit

Table 7-15. 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). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
0	DMACH0	R/W	0h	DMA Channel 0 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit

7.3.7 RIS (Offset = 1030h) [Reset = 00000000h]

RIS is shown in Figure 7-11 and described in Table 7-16.

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 7-11. 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 7-16. 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). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
24	ADDRERR	R	0h	DMA address error, SRC address not reachable. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
23	PREIRQCH7	R	0h	Pre-IRQ for Channel 7. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
22	PREIRQCH6	R	0h	Pre-IRQ for Channel 6. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
21	PREIRQCH5	R	0h	Pre-IRQ for Channel 5. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
20	PREIRQCH4	R	0h	Pre-IRQ for Channel 4. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
19	PREIRQCH3	R	0h	Pre-IRQ for Channel 3. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
18	PREIRQCH2	R	0h	Pre-IRQ for Channel 2. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit

Table 7-16. RIS Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17	PREIRQCH1	R	0h	Pre-IRQ for Channel 1. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
16	PREIRQCH0	R	0h	Pre-IRQ for Channel 0. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
15	DMACH15	R	0h	DMA Channel 15 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
14	DMACH14	R	0h	DMA Channel 14 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
13	DMACH13	R	0h	DMA Channel 13 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
12	DMACH12	R	0h	DMA Channel 12 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
11	DMACH11	R	0h	DMA Channel 11 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
10	DMACH10	R	0h	DMA Channel 10 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
9	DMACH9	R	0h	DMA Channel 9 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
8	DMACH8	R	0h	DMA Channel 8 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
7	DMACH7	R	0h	DMA Channel 7 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
6	DMACH6	R	0h	DMA Channel 6 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
5	DMACH5	R	0h	DMA Channel 5 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
4	DMACH4	R	0h	DMA Channel 4 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
3	DMACH3	R	0h	DMA Channel 3 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred

Table 7-16. 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). 0h = Interrupt did not occur 1h = Interrupt occurred
1	DMACH1	R	0h	DMA Channel 1 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
0	DMACH0	R	0h	DMA Channel 0 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred

7.3.8 MIS (Offset = 1038h) [Reset = 00000000h]

MIS is shown in Figure 7-12 and described in Table 7-17.

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 7-12. 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 7-17. 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). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
24	ADDRERR	R	0h	DMA address error, SRC address not reachable. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
23	PREIRQCH7	R	0h	Pre-IRQ for Channel 7. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
22	PREIRQCH6	R	0h	Pre-IRQ for Channel 6. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
21	PREIRQCH5	R	0h	Pre-IRQ for Channel 5. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
20	PREIRQCH4	R	0h	Pre-IRQ for Channel 4. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
19	PREIRQCH3	R	0h	Pre-IRQ for Channel 3. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
18	PREIRQCH2	R	0h	Pre-IRQ for Channel 2. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
17	PREIRQCH1	R	0h	Pre-IRQ for Channel 1. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit

Table 7-17. MIS Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	PREIRQCH0	R	0h	Pre-IRQ for Channel 0. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
15	DMACH15	R	0h	DMA Channel 15 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
14	DMACH14	R	0h	DMA Channel 14 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
13	DMACH13	R	0h	DMA Channel 13 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
12	DMACH12	R	0h	DMA Channel 12 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
11	DMACH11	R	0h	DMA Channel 11 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
10	DMACH10	R	0h	DMA Channel 10 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
9	DMACH9	R	0h	DMA Channel 9 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
8	DMACH8	R	0h	DMA Channel 8 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
7	DMACH7	R	0h	DMA Channel 7 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
6	DMACH6	R	0h	DMA Channel 6 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
5	DMACH5	R	0h	DMA Channel 5 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
4	DMACH4	R	0h	DMA Channel 4 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
3	DMACH3	R	0h	DMA Channel 3 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
2	DMACH2	R	0h	DMA Channel 2 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred

Table 7-17. 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). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
0	DMACH0	R	0h	DMA Channel 0 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred

7.3.9 ISET (Offset = 1040h) [Reset = 00000000h]

ISET is shown in Figure 7-13 and described in Table 7-18.

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 7-13. 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 7-18. 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). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
24	ADDRERR	W	0h	DMA address error, SRC address not reachable. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
23	PREIRQCH7	W	0h	Pre-IRQ for Channel 7. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
22	PREIRQCH6	W	0h	Pre-IRQ for Channel 6. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
21	PREIRQCH5	W	0h	Pre-IRQ for Channel 5. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
20	PREIRQCH4	W	0h	Pre-IRQ for Channel 4. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
19	PREIRQCH3	W	0h	Pre-IRQ for Channel 3. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
18	PREIRQCH2	W	0h	Pre-IRQ for Channel 2. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit

Table 7-18. ISET Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17	PREIRQCH1	W	0h	Pre-IRQ for Channel 1. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
16	PREIRQCH0	W	0h	Pre-IRQ for Channel 0. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
15	DMACH15	W	0h	DMA Channel 0 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
14	DMACH14	W	0h	DMA Channel 14 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
13	DMACH13	W	0h	DMA Channel 13 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
12	DMACH12	W	0h	DMA Channel 12 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
11	DMACH11	W	0h	DMA Channel 11 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
10	DMACH10	W	0h	DMA Channel 0 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
9	DMACH9	W	0h	DMA Channel 9 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
8	DMACH8	W	0h	DMA Channel 8 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
7	DMACH7	W	0h	DMA Channel 7 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
6	DMACH6	W	0h	DMA Channel 6 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
5	DMACH5	W	0h	DMA Channel 5 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
4	DMACH4	W	0h	DMA Channel 4 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
3	DMACH3	W	0h	DMA Channel 3 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt

Table 7-18. 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). 0h = Writing 0 has no effect 1h = Set interrupt
1	DMACH1	W	0h	DMA Channel 1 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
0	DMACH0	W	0h	DMA Channel 0 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt

7.3.10 ICLR (Offset = 1048h) [Reset = 00000000h]

ICLR is shown in Figure 7-14 and described in Table 7-19.

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 7-14. 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 7-19. 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). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
24	ADDRERR	W	0h	DMA address error, SRC address not reachable. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
23	PREIRQCH7	W	0h	Pre-IRQ for Channel 7. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
22	PREIRQCH6	W	0h	Pre-IRQ for Channel 6. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
21	PREIRQCH5	W	0h	Pre-IRQ for Channel 5. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
20	PREIRQCH4	W	0h	Pre-IRQ for Channel 4. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
19	PREIRQCH3	W	0h	Pre-IRQ for Channel 3. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
18	PREIRQCH2	W	0h	Pre-IRQ for Channel 2. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
17	PREIRQCH1	W	0h	Pre-IRQ for Channel 1. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit

Table 7-19. ICLR Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	PREIRQCH0	W	0h	Pre-IRQ for Channel 0. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
15	DMACH15	W	0h	DMA Channel 15 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
14	DMACH14	W	0h	DMA Channel 14 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
13	DMACH13	W	0h	DMA Channel 13 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
12	DMACH12	W	0h	DMA Channel 12 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
11	DMACH11	W	0h	DMA Channel 11 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
10	DMACH10	W	0h	DMA Channel 10 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
9	DMACH9	W	0h	DMA Channel 9 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
8	DMACH8	W	0h	DMA Channel 8 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
7	DMACH7	W	0h	DMA Channel 7 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
6	DMACH6	W	0h	DMA Channel 6 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
5	DMACH5	W	0h	DMA Channel 5 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
4	DMACH4	W	0h	DMA Channel 4 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
3	DMACH3	W	0h	DMA Channel 3 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
2	DMACH2	W	0h	DMA Channel 2 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt

Table 7-19. 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). 0h = Writing 0 has no effect 1h = Clear interrupt
0	DMACH0	W	0h	DMA Channel 0 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt

7.3.11 IIDX (Offset = 1050h) [Reset = 00000000h]

IIDX is shown in Figure 7-15 and described in Table 7-20.

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 7-15. 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 7-20. IIDX Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	STAT	R	0h	Interrupt index status 00h = No bit is set means there is no pending interrupt request 01h = DMA Channel 0 size counter reached zero (DMASZ=0). 02h = DMA Channel 1 size counter reached zero (DMASZ=0). 03h = DMA Channel 2 size counter reached zero (DMASZ=0). 04h = DMA Channel 3 size counter reached zero (DMASZ=0). 05h = DMA Channel 4 size counter reached zero (DMASZ=0). 06h = DMA Channel 5 size counter reached zero (DMASZ=0). 07h = DMA Channel 6 size counter reached zero (DMASZ=0). 08h = DMA Channel 7 size counter reached zero (DMASZ=0). 09h = DMA Channel 8 size counter reached zero (DMASZ=0). 0Ah = DMA Channel 9 size counter reached zero (DMASZ=0). 0Bh = DMA Channel 10 size counter reached zero (DMASZ=0). 0Ch = DMA Channel 11 size counter reached zero (DMASZ=0). 0Dh = DMA Channel 12 size counter reached zero (DMASZ=0). 0Eh = DMA Channel 13 size counter reached zero (DMASZ=0). 0Fh = DMA Channel 14 size counter reached zero (DMASZ=0). 10h = DMA Channel 15 size counter reached zero (DMASZ=0). 11h = PRE-IRQ event for DMA Channel 0. 12h = PRE-IRQ event for DMA Channel 1. 13h = PRE-IRQ event for DMA Channel 2. 14h = PRE-IRQ event for DMA Channel 3. 15h = PRE-IRQ event for DMA Channel 4. 16h = PRE-IRQ event for DMA Channel 5. 17h = PRE-IRQ event for DMA Channel 6. 18h = PRE-IRQ event for DMA Channel 7. 19h = DMA address error, SRC address not reachable. 1Ah = DMA data error, SRC data might be corrupted (PAR or ECC error).

7.3.12 IMASK (Offset = 1058h) [Reset = 00000000h]

IMASK is shown in Figure 7-16 and described in Table 7-21.

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 7-16. 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 7-21. 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). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
24	ADDRERR	R/W	0h	DMA address error, SRC address not reachable. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
23	PREIRQCH7	R/W	0h	Pre-IRQ for Channel 7. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
22	PREIRQCH6	R/W	0h	Pre-IRQ for Channel 6. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
21	PREIRQCH5	R/W	0h	Pre-IRQ for Channel 5. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
20	PREIRQCH4	R/W	0h	Pre-IRQ for Channel 4. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
19	PREIRQCH3	R/W	0h	Pre-IRQ for Channel 3. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
18	PREIRQCH2	R/W	0h	Pre-IRQ for Channel 2. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
17	PREIRQCH1	R/W	0h	Pre-IRQ for Channel 1. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit

Table 7-21. 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. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
15	DMACH15	R/W	0h	DMA Channel 15 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
14	DMACH14	R/W	0h	DMA Channel 14 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
13	DMACH13	R/W	0h	DMA Channel 13 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
12	DMACH12	R/W	0h	DMA Channel 12 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
11	DMACH11	R/W	0h	DMA Channel 11 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
10	DMACH10	R/W	0h	DMA Channel 10 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
9	DMACH9	R/W	0h	DMA Channel 9 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
8	DMACH8	R/W	0h	DMA Channel 8 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
7	DMACH7	R/W	0h	DMA Channel 7 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
6	DMACH6	R/W	0h	DMA Channel 6 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
5	DMACH5	R/W	0h	DMA Channel 5 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
4	DMACH4	R/W	0h	DMA Channel 4 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
3	DMACH3	R/W	0h	DMA Channel 3 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
2	DMACH2	R/W	0h	DMA Channel 2 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit

Table 7-21. 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). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
0	DMACH0	R/W	0h	DMA Channel 0 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit

7.3.13 RIS (Offset = 1060h) [Reset = 00000000h]

RIS is shown in Figure 7-17 and described in Table 7-22.

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 7-17. 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 7-22. 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). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
24	ADDRERR	R	0h	DMA address error, SRC address not reachable. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
23	PREIRQCH7	R	0h	Pre-IRQ for Channel 7. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
22	PREIRQCH6	R	0h	Pre-IRQ for Channel 6. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
21	PREIRQCH5	R	0h	Pre-IRQ for Channel 5. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
20	PREIRQCH4	R	0h	Pre-IRQ for Channel 4. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
19	PREIRQCH3	R	0h	Pre-IRQ for Channel 3. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
18	PREIRQCH2	R	0h	Pre-IRQ for Channel 2. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit

Table 7-22. RIS Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17	PREIRQCH1	R	0h	Pre-IRQ for Channel 1. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
16	PREIRQCH0	R	0h	Pre-IRQ for Channel 0. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
15	DMACH15	R	0h	DMA Channel 15 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
14	DMACH14	R	0h	DMA Channel 14 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
13	DMACH13	R	0h	DMA Channel 13 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
12	DMACH12	R	0h	DMA Channel 12 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
11	DMACH11	R	0h	DMA Channel 11 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
10	DMACH10	R	0h	DMA Channel 10 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
9	DMACH9	R	0h	DMA Channel 9 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
8	DMACH8	R	0h	DMA Channel 8 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
7	DMACH7	R	0h	DMA Channel 7 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
6	DMACH6	R	0h	DMA Channel 6 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
5	DMACH5	R	0h	DMA Channel 5 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
4	DMACH4	R	0h	DMA Channel 4 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
3	DMACH3	R	0h	DMA Channel 3 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred

Table 7-22. 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). 0h = Interrupt did not occur 1h = Interrupt occurred
1	DMACH1	R	0h	DMA Channel 1 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
0	DMACH0	R	0h	DMA Channel 0 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred

7.3.14 MIS (Offset = 1068h) [Reset = 00000000h]

MIS is shown in Figure 7-18 and described in Table 7-23.

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 7-18. 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 7-23. 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). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
24	ADDRERR	R	0h	DMA address error, SRC address not reachable. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
23	PREIRQCH7	R	0h	Pre-IRQ for Channel 7. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
22	PREIRQCH6	R	0h	Pre-IRQ for Channel 6. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
21	PREIRQCH5	R	0h	Pre-IRQ for Channel 5. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
20	PREIRQCH4	R	0h	Pre-IRQ for Channel 4. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
19	PREIRQCH3	R	0h	Pre-IRQ for Channel 3. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
18	PREIRQCH2	R	0h	Pre-IRQ for Channel 2. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
17	PREIRQCH1	R	0h	Pre-IRQ for Channel 1. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit

Table 7-23. MIS Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	PREIRQCH0	R	0h	Pre-IRQ for Channel 0. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
15	DMACH15	R	0h	DMA Channel 15 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
14	DMACH14	R	0h	DMA Channel 14 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
13	DMACH13	R	0h	DMA Channel 13 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
12	DMACH12	R	0h	DMA Channel 12 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
11	DMACH11	R	0h	DMA Channel 11 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
10	DMACH10	R	0h	DMA Channel 10 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
9	DMACH9	R	0h	DMA Channel 9 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
8	DMACH8	R	0h	DMA Channel 8 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
7	DMACH7	R	0h	DMA Channel 7 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
6	DMACH6	R	0h	DMA Channel 6 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
5	DMACH5	R	0h	DMA Channel 5 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
4	DMACH4	R	0h	DMA Channel 4 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
3	DMACH3	R	0h	DMA Channel 3 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
2	DMACH2	R	0h	DMA Channel 2 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred

Table 7-23. 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). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
0	DMACH0	R	0h	DMA Channel 0 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred

7.3.15 ISET (Offset = 1070h) [Reset = 00000000h]

ISET is shown in Figure 7-19 and described in Table 7-24.

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 7-19. 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 7-24. 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). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
24	ADDRERR	W	0h	DMA address error, SRC address not reachable. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
23	PREIRQCH7	W	0h	Pre-IRQ for Channel 7. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
22	PREIRQCH6	W	0h	Pre-IRQ for Channel 6. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
21	PREIRQCH5	W	0h	Pre-IRQ for Channel 5. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
20	PREIRQCH4	W	0h	Pre-IRQ for Channel 4. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
19	PREIRQCH3	W	0h	Pre-IRQ for Channel 3. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
18	PREIRQCH2	W	0h	Pre-IRQ for Channel 2. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit

Table 7-24. ISET Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17	PREIRQCH1	W	0h	Pre-IRQ for Channel 1. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
16	PREIRQCH0	W	0h	Pre-IRQ for Channel 0. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
15	DMACH15	W	0h	DMA Channel 0 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
14	DMACH14	W	0h	DMA Channel 14 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
13	DMACH13	W	0h	DMA Channel 13 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
12	DMACH12	W	0h	DMA Channel 12 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
11	DMACH11	W	0h	DMA Channel 11 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
10	DMACH10	W	0h	DMA Channel 0 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
9	DMACH9	W	0h	DMA Channel 9 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
8	DMACH8	W	0h	DMA Channel 8 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
7	DMACH7	W	0h	DMA Channel 7 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
6	DMACH6	W	0h	DMA Channel 6 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
5	DMACH5	W	0h	DMA Channel 5 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
4	DMACH4	W	0h	DMA Channel 4 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
3	DMACH3	W	0h	DMA Channel 3 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt

Table 7-24. 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). 0h = Writing 0 has no effect 1h = Set interrupt
1	DMACH1	W	0h	DMA Channel 1 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
0	DMACH0	W	0h	DMA Channel 0 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt

7.3.16 ICLR (Offset = 1078h) [Reset = 00000000h]

ICLR is shown in Figure 7-20 and described in Table 7-25.

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 7-20. 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 7-25. 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). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
24	ADDRERR	W	0h	DMA address error, SRC address not reachable. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
23	PREIRQCH7	W	0h	Pre-IRQ for Channel 7. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
22	PREIRQCH6	W	0h	Pre-IRQ for Channel 6. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
21	PREIRQCH5	W	0h	Pre-IRQ for Channel 5. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
20	PREIRQCH4	W	0h	Pre-IRQ for Channel 4. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
19	PREIRQCH3	W	0h	Pre-IRQ for Channel 3. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
18	PREIRQCH2	W	0h	Pre-IRQ for Channel 2. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
17	PREIRQCH1	W	0h	Pre-IRQ for Channel 1. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit

Table 7-25. ICLR Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	PREIRQCH0	W	0h	Pre-IRQ for Channel 0. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
15	DMACH15	W	0h	DMA Channel 15 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
14	DMACH14	W	0h	DMA Channel 14 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
13	DMACH13	W	0h	DMA Channel 13 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
12	DMACH12	W	0h	DMA Channel 12 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
11	DMACH11	W	0h	DMA Channel 11 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
10	DMACH10	W	0h	DMA Channel 10 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
9	DMACH9	W	0h	DMA Channel 9 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
8	DMACH8	W	0h	DMA Channel 8 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
7	DMACH7	W	0h	DMA Channel 7 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
6	DMACH6	W	0h	DMA Channel 6 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
5	DMACH5	W	0h	DMA Channel 5 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
4	DMACH4	W	0h	DMA Channel 4 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
3	DMACH3	W	0h	DMA Channel 3 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
2	DMACH2	W	0h	DMA Channel 2 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt

Table 7-25. 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). 0h = Writing 0 has no effect 1h = Clear interrupt
0	DMACH0	W	0h	DMA Channel 0 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt

7.3.17 EVT_MODE (Offset = 10E0h) [Reset = 00000000h]

EVT_MODE is shown in Figure 7-21 and described in Table 7-26.

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 7-21. 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 7-26. 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 0h = The interrupt or event line is disabled. 1h = The interrupt or event line is in software mode. Software must clear the RIS. 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 0h = The interrupt or event line is disabled. 1h = The interrupt or event line is in software mode. Software must clear the RIS. 2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag.

7.3.18 DESC (Offset = 10FCh) [Reset = 00000000h]

DESC is shown in Figure 7-22 and described in Table 7-27.

Return to the Summary Table.

This register identifies the peripheral and its exact version.

Figure 7-22. 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 7-27. 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. 0h = Smallest value 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). 0h = Smallest value (1 channel) Fh = Highest value (16 channel)
11-8	RESERVED	R	0h	
7-4	MAJREV	R	0h	Major rev of the IP 0h = Smallest value Fh = Highest possible value
3-0	MINREV	R	0h	Minor rev of the IP 0h = Smallest value Fh = Highest possible value

7.3.19 DMAPRIO (Offset = 1100h) [Reset = 00000000h]

DMAPRIO is shown in Figure 7-23 and described in Table 7-28.

Return to the Summary Table.

DMA Channel Priority Control

Figure 7-23. 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 7-28. 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 0h = There is no burst size, the whole block transfer is completed on one transfer without interruption 1h = The burst size is 8, after 9 transfers the block transfer is interrupted and the priority is reevaluated 2h = The burst size is 16, after 17 transfers the block transfer is interrupted and the priority is reevaluated 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. 0h = Roundrobin priority disabled, DMA channel priority is fixed: DMA0-DMA1-DMA2-...-DMA16 1h = Roundrobin priority enabled, DMA channel priority changes with each transfer

7.3.20 DMATCTL[j] (Offset = 1110h + formula) [Reset = 00000000h]

DMATCTL[j] is shown in Figure 7-24 and described in Table 7-29.

Return to the Summary Table.

DMA Trigger Control

Offset = 1110h + (j * 4h); where j = 0h to Fh

Figure 7-24. 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 7-29. DMATCTL[j] Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7	DMATINT	R/W	0h	DMA Trigger by Internal Channel 0h = DMATSEL will define external trigger select as transfer trigger. 1h = DMATSEL will define internal channel as transfer trigger select. 0-> Channel0-done, 1-> Channel1-done, ...
6	RESERVED	R	0h	
5-0	DMATSEL	R/W	0h	DMA Trigger Select Note: Reference the datasheet of the device to see the specific trigger mapping. 00h = Software trigger request 3Fh = Highest possible value

7.3.21 DMACTL[j] (Offset = 1200h + formula) [Reset = 00000000h]

DMACTL[j] is shown in Figure 7-25 and described in Table 7-30.

Return to the Summary Table.

DMA Channel Control

Offset = 1200h + (j * 10h); where j = 0h to Fh

Figure 7-25. 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 7-30. DMACTL[j] Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R	0h	
29-28	DMATM	R/W	0h	DMA transfer mode register 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. 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. 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. 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. 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	

Table 7-30. DMACTL[j] Field Descriptions (continued)

Bit	Field	Type	Reset	Description
25-24	DMAEM	R/W	0h	DMA extended mode 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. 0h = Normal mode is related to transfers from SRC to DST 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. 2h = Fill mode will copy the SA register content as data to DA 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. 0h = Address is unchanged (+0) 2h = Decremented by 1 (-1 * DMADSTWDTH) 3h = Incremented by 1 (+1 * DMADSTWDTH) 8h = Stride size 2 (+2 * DMADSTWDTH) 9h = Stride size 3 (+3 * DMADSTWDTH) Ah = Stride size 4 (+4 * DMADSTWDTH) Bh = Stride size 5 (+5 * DMADSTWDTH) Ch = Stride size 6 (+6 * DMADSTWDTH) Dh = Stride size 7 (+7 * DMADSTWDTH) Eh = Stride size 8 (+8 * DMADSTWDTH) 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. 0h = Address is unchanged (+0) 2h = Decremented by 1 (-1 * DMASRCWDTH) 3h = Incremented by 1 (+1 * DMASRCWDTH) 8h = Stride size 2 (+2 * DMASRCWDTH) 9h = Stride size 3 (+3 * DMASRCWDTH) Ah = Stride size 4 (+4 * DMASRCWDTH) Bh = Stride size 5 (+5 * DMASRCWDTH) Ch = Stride size 6 (+6 * DMASRCWDTH) Dh = Stride size 7 (+7 * DMASRCWDTH) Eh = Stride size 8 (+8 * DMASRCWDTH) 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. 0h = Destination data width is BYTE (8-bit) 1h = Destination data width is HALF-WORD (16-bit) 2h = Destination data width is WORD (32-bit) 3h = Destination data width is LONG-WORD (64-bit) 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	

Table 7-30. 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. 0h = Source data width is BYTE (8-bit) 1h = Source data width is HALF-WORD (16-bit) 2h = Source data width is WORD (32-bit) 3h = Source data width is LONG-WORD (64-bit) 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. 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. 0h = Pre-IRQ event disabled. 1h = Issure Pre-IRQ event when DMASZ=1 2h = Issure Pre-IRQ event when DMASZ=2 3h = Issure Pre-IRQ event when DMASZ=4 4h = Issure Pre-IRQ event when DMASZ=8 5h = Issure Pre-IRQ event when DMASZ=32 6h = Issure Pre-IRQ event when DMASZ=64 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. If channel is configured as SW trigger (DMATCTL=0), the AUTOEN will set the DMAEN and DMAREQ. 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. 0h = No automatic DMA enable 1h = Automatic DMA enable on DMASA register write. 2h = Automatic DMA enable on DMADA register write. 3h = Automatic DMA enable on DMASZ register write.
1	DMAEN	R/W	0h	DMA enable 0h = DMA channel disabled 1h = DMA channel enabled
0	DMAREQ	R/W	0h	DMA request. Software-controlled DMA start. DMAREQ is reset automatically. 0h = Default read value 1h = DMA transfer request (start DMA)

7.3.22 DMASA[j] (Offset = 1204h + formula) [Reset = 00000000h]

DMASA[j] is shown in Figure 7-26 and described in Table 7-31.

Return to the Summary Table.

DMA Channel Source Address

Offset = 1204h + (j * 10h); where j = 0h to Fh

Figure 7-26. 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 7-31. DMASA[j] Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDR	R/W	0h	DMA Channel Source Address 0h = Smallest value FFFFFFFFh = Highest possible value

7.3.23 DMADA[j] (Offset = 1208h + formula) [Reset = 00000000h]

DMADA[j] is shown in Figure 7-27 and described in Table 7-32.

Return to the Summary Table.

DMA Channel Destination Address

Offset = 1208h + (j * 10h); where j = 0h to Fh

Figure 7-27. 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 7-32. DMADA[j] Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDR	R/W	0h	DMA Channel Destination Address 0h = Smallest value FFFFFFFFh = Highest possible value

7.3.24 DMASZ[j] (Offset = 120Ch + formula) [Reset = 00000000h]

DMASZ[j] is shown in Figure 7-28 and described in Table 7-33.

Return to the Summary Table.

DMA Channel Size

Offset = 120Ch + (j * 10h); where j = 0h to Fh

Figure 7-28. 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 7-33. 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 0h = Smallest value FFFFh = Highest possible value

Chapter 8
Events

The event manager provides the peripheral-to-peripheral, peripheral-to-DMA, and peripheral-to-CPU (IRQ) event connections.

8.1 Events Overview.....414
8.2 Events Operation.....418

8.1 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.

8.1.1 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.

8.1.2 Event Subscriber

Event subscribers are included within the processor, the DMA, and certain peripherals (see Section 8.1.4). 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.

8.1.3 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.

8.1.3.1 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.

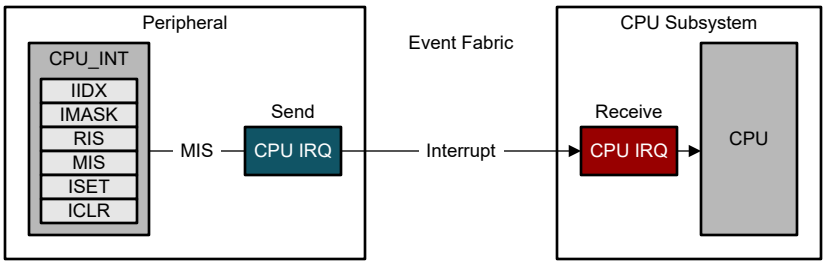


Figure 8-1. CPU Interrupt (Fixed Event Route)

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 8.2.5.3 for guidance on setting and clearing interrupt status with the event management registers.

8.1.3.2 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 8-2.

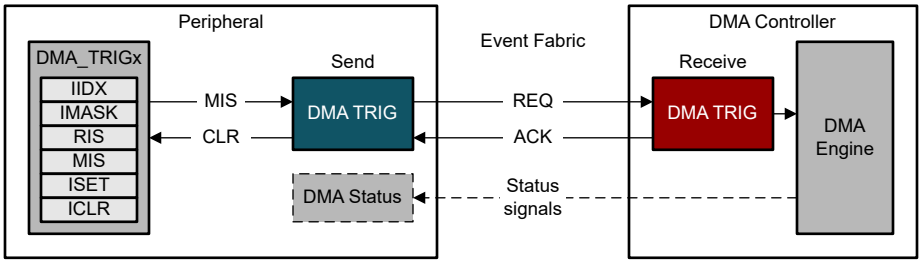


Figure 8-2. DMA Route

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 8-3 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.

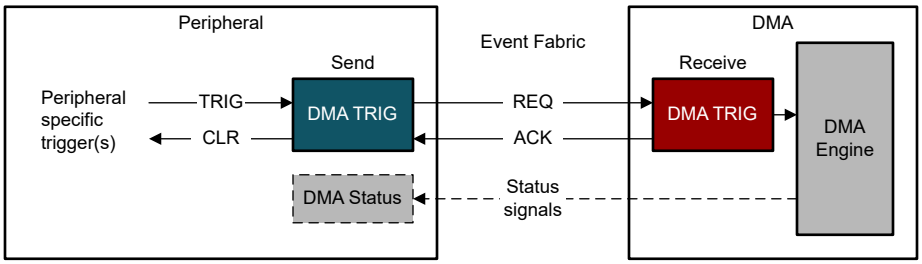


Figure 8-3. DMA Route without Event Management Registers

8.1.3.3 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 8-4.

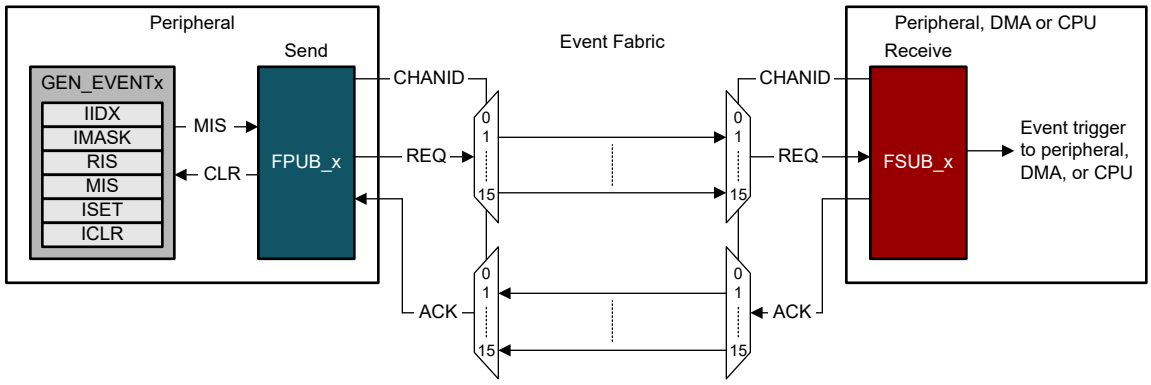


Figure 8-4. Generic Route

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.

8.1.4 Event Routing Map

The event capabilities of each peripheral type are shown in Figure 8-5. 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.

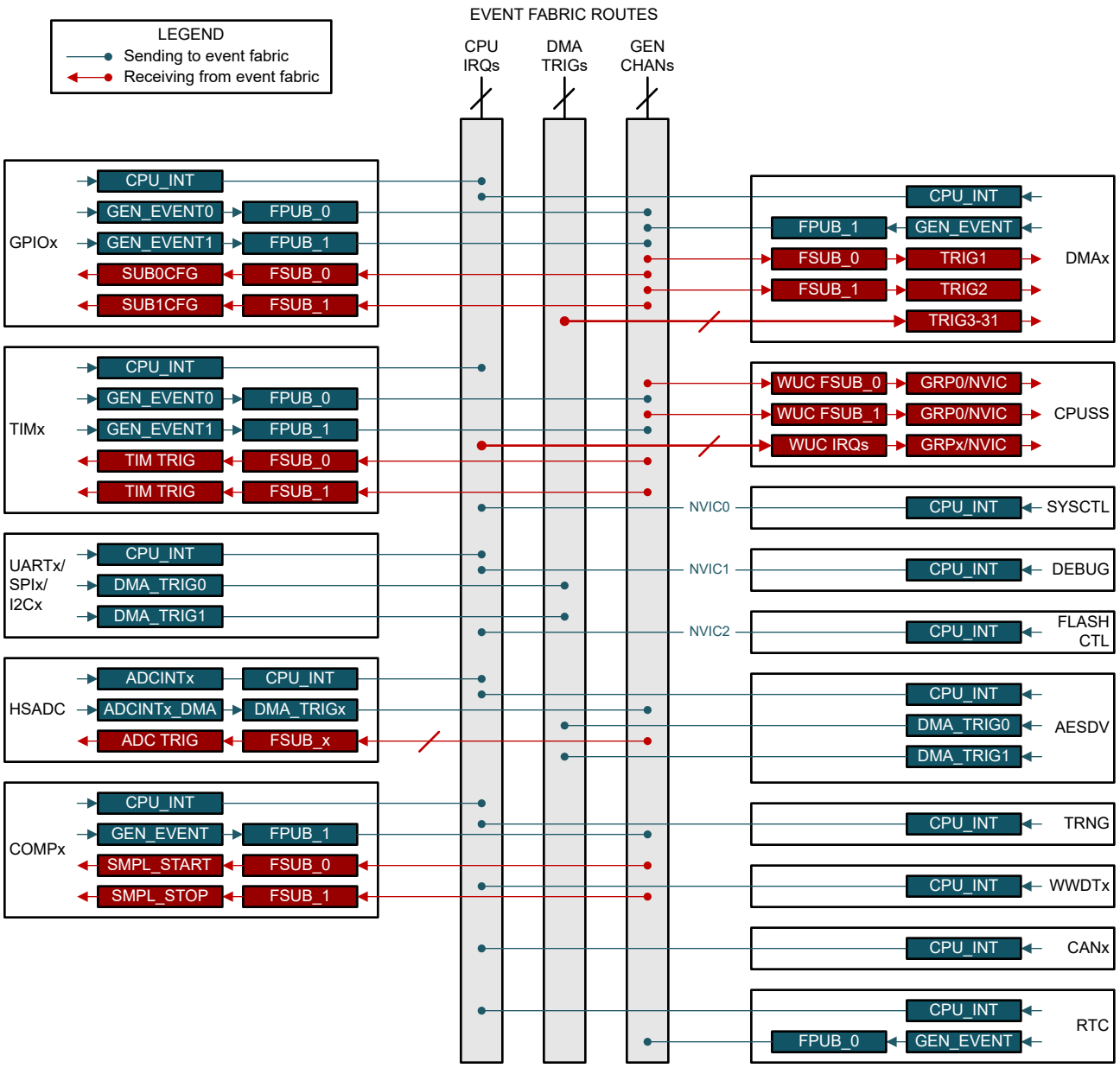


Figure 8-5. Event Map

8.1.5 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.

8.2 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.

8.2.1 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.

8.2.2 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-

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 8.1.4.

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.

8.2.3 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.

8.2.4 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.

8.2.5 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 8-1 for which event management groups are mapped to specific functions in the group name for a peripheral's "Registers" section.

Table 8-1. 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)

8.2.5.1 Event Registers

The event management register set contains 6 standard registers: RIS, IMASK, MIS, ISET, ICLR, and IIDX, given in Table 8-2. The event registers are interconnected as shown in Figure 8-6.

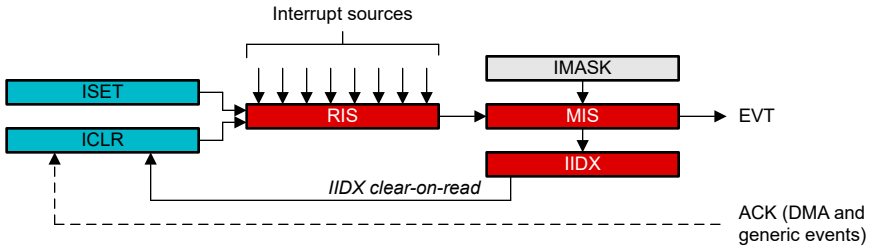


Figure 8-6. Event Management Register Relationship

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.

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 8-2. 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.

8.2.5.2 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.

8.2.5.3 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

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();
  }
}
```

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();
  }
}
```

8.2.5.4 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.

Chapter 9
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).

9.1 IOMUX Overview 424
9.2 IOMUX Operation 427
9.3 IOMUX Registers 431

9.1 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)

9.1.1 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 9-1 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 9-1. 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

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 9-1. 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.

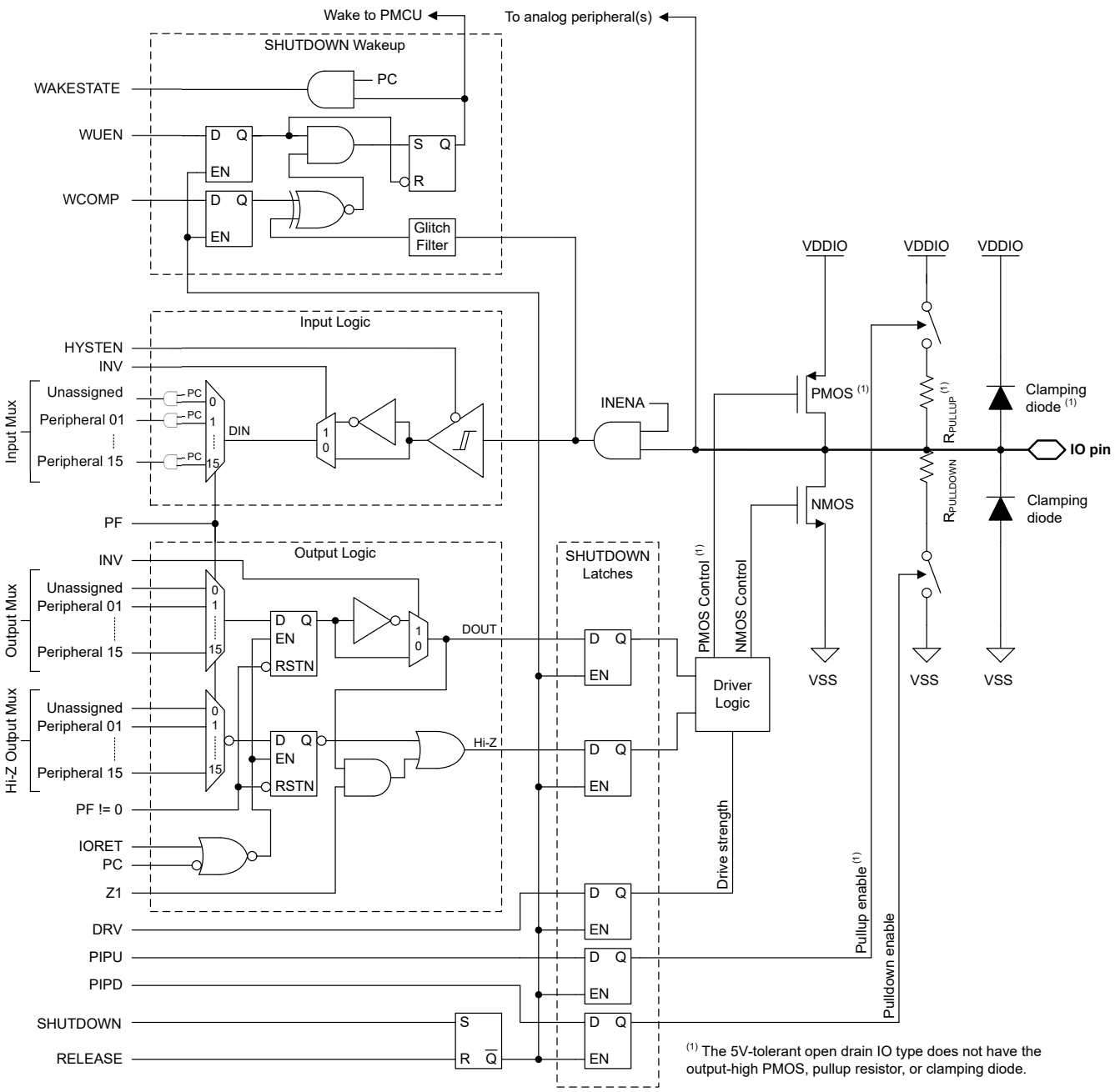


Figure 9-1. Superset IO Slice

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

- 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.

9.2 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.

9.2.1 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).

9.2.2 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 9-2.

Table 9-2. 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.

9.2.3 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.

9.2.4 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.

After SHUTDOWN is entered, a level match on any pin configured for wakeup 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).

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.

9.2.5 Pullup/Pulldown Resistors

Programmable 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.

9.2.6 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.

9.2.7 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 9-2.

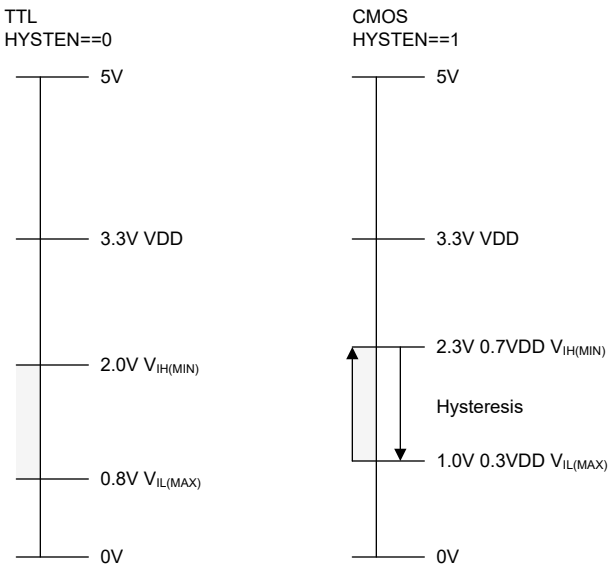


Figure 9-2. Input Logic Levels - 5V Tolerant Open Drain Digital IO

9.3 IOMUX Registers

Table 9-3 lists the memory-mapped registers for the IOMUX registers. All register offset addresses not listed in Table 9-3 should be considered as reserved locations and the register contents should not be modified.

Table 9-3. 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 9-4 shows the codes that are used for access types in this section.

Table 9-4. 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

9.3.1 PINCM (Offset = 4h) [Reset = X]

PINCM is shown in Figure 9-3 and described in Table 9-5.

Return to the Summary Table.

Pin Control Management Register

Figure 9-3. 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 9-5. PINCM Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit 0h = Wakeup on a match of 0 1h = Wakeup on a match of 1
27	WUEN	R/W	0h	Wakeup Enable bit 0h = wakeup is disabled. 1h = wakeup is enabled
26	INV	R/W	0h	Data inversion selection 0h = Data inversion is disabled. 1h = Data inversion is enabled
25	HIZ1	R/W	0h	High output value will tri-state the output when this bit is enabled 0h = open-drain is disabled. 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 0h = Drive setting of 0 selected 1h = Drive setting of 1 selected
19	HYSTEN	R/W	0h	Hysteresis Enable Control Selection 0h = hysteresis is disabled. 1h = hysteresis is enabled
18	INENA	R/W	0h	Input Enable Control Selection 0h = Input enable is disabled. 1h = Input enable is enabled.
17	PIPU	R/W	0h	Pull Up control selection 0h = Pull up is disabled. 1h = Pull up is enabled
16	PIPD	R/W	0h	Pull Down control selection 0h = Pull down is disabled. 1h = Pull down is enabled
15-14	RESERVED	R/W	0h	

Table 9-5. PINCM Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13	WAKESTAT	R	0h	This has the IOPAD WAKEUP signal as status bit. 0h = wakeup source is NOT from this IOCELL 1h = wakeup source is from this IOCELL
12-8	RESERVED	R/W	0h	
7	PC	R/W	0h	Peripheral is “Connected” 0h = The output of the peripheral (and its output enable) will not propagate to the IOCELL 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 0h = Reserved as unconnected 3Fh = An encoding per function that can be connected to this pin.

Chapter 10

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.

10.1 GPIO Overview 435
10.2 GPIO Operation 435
10.3 GPIO Registers 439

10.1 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
- "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.

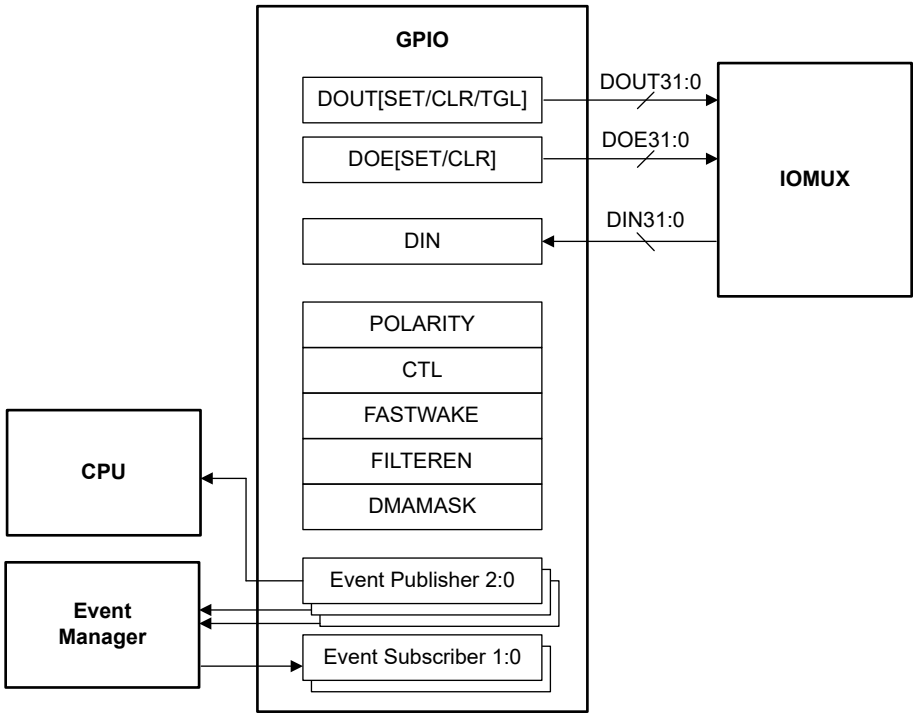


Figure 10-1. GPIO Block Diagram

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 9. 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.

10.2 GPIO Operation

The GPIO peripheral is configured with user software. The setup and operation of the GPIO is discussed in the following sections.

10.2.1 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 10-1. 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

10.2.2 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.

10.2.3 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.

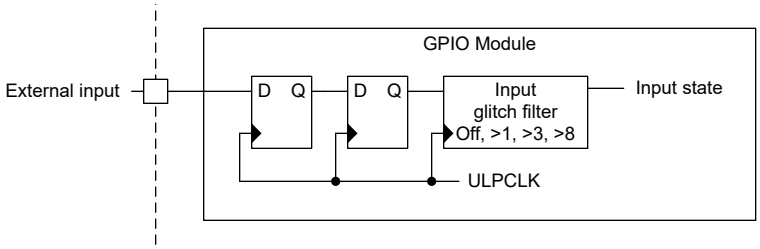


Figure 10-2. GPIO Input Synchronizer

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.

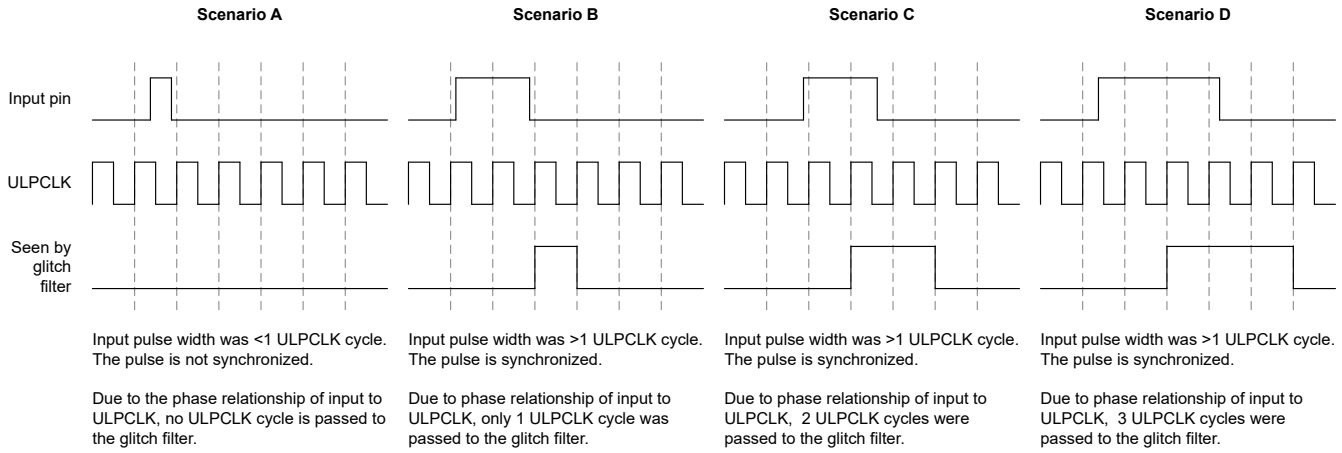


Figure 10-3. GPIO Input Synchronizer and Glitch Filtering Scenarios

- 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 32MHz, resulting in the input synchronization logic and glitch filter running at 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.

10.2.4 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.

10.2.5 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

10.3 GPIO Registers

Table 10-2 lists the memory-mapped registers for the GPIO registers. All register offset addresses not listed in Table 10-2 should be considered as reserved locations and the register contents should not be modified.

Table 10-2. 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

Table 10-2. 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 10-3 shows the codes that are used for access types in this section.

Table 10-3. 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 K	Write Write protected by a key
Reset or Default Value		
-n		Value after reset or the default value

10.3.1 FSUB_0 (Offset = 400h) [Reset = 00000000h]

FSUB_0 is shown in Figure 10-4 and described in Table 10-4.

Return to the Summary Table.

Subscriber port

Figure 10-4. 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 10-4. FSUB_0 Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3-0	CHANID	R/W	0h	0 = disconnected. 1-15 = connected to channelID = CHANID. 0h = A value of 0 specifies that the event is not connected Fh = Consult your device data sheet as the actual allowed maximum may be less than 15.

10.3.2 FSUB_1 (Offset = 404h) [Reset = 00000000h]

FSUB_1 is shown in Figure 10-5 and described in Table 10-5.

Return to the Summary Table.

Subscriber port

Figure 10-5. 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 10-5. FSUB_1 Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3-0	CHANID	R/W	0h	0 = disconnected. 1-15 = connected to channelID = CHANID. 0h = A value of 0 specifies that the event is not connected Fh = Consult your device data sheet as the actual allowed maximum may be less than 15.

10.3.3 FPUB_0 (Offset = 444h) [Reset = 00000000h]

FPUB_0 is shown in Figure 10-6 and described in Table 10-6.

Return to the Summary Table.

Publisher port

Figure 10-6. 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 10-6. FPUB_0 Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3-0	CHANID	R/W	0h	0 = disconnected. 1-15 = connected to channelID = CHANID. 0h = A value of 0 specifies that the event is not connected Fh = Consult your device data sheet as the actual allowed maximum may be less than 15.

10.3.4 FPUB_1 (Offset = 448h) [Reset = 00000000h]

FPUB_1 is shown in Figure 10-7 and described in Table 10-7.

Return to the Summary Table.

Publisher port

Figure 10-7. 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 10-7. FPUB_1 Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3-0	CHANID	R/W	0h	0 = disconnected. 1-15 = connected to channelID = CHANID. 0h = A value of 0 specifies that the event is not connected Fh = Consult your device data sheet as the actual allowed maximum may be less than 15.

10.3.5 PWREN (Offset = 800h) [Reset = 00000000h]

PWREN is shown in Figure 10-8 and described in Table 10-8.

Return to the Summary Table.

Register to control the power state

Figure 10-8. 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 10-8. PWREN Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to allow Power State Change 26h = KEY to allow write access to this register
23-1	RESERVED	R/W	0h	
0	ENABLE	K	0h	Enable the power KEY must be set to 26h to write to this bit. 0h = Disable Power 1h = Enable Power

10.3.6 RSTCTL (Offset = 804h) [Reset = 00000000h]

RSTCTL is shown in Figure 10-9 and described in Table 10-9.

Return to the Summary Table.

Register to control reset assertion and de-assertion

Figure 10-9. 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 10-9. RSTCTL Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	Unlock key 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 KEY must be set to B1h to write to this bit. 0h = Writing 0 has no effect 1h = Clear reset sticky bit
0	RESETASSERT	WK	0h	Assert reset to the peripheral KEY must be set to B1h to write to this bit. 0h = Writing 0 has no effect 1h = Assert reset

10.3.7 STAT (Offset = 814h) [Reset = 00000000h]

STAT is shown in Figure 10-10 and described in Table 10-10.

Return to the Summary Table.

peripheral enable and reset status

Figure 10-10. 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 10-10. 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 0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register 1h = The peripheral was reset since the last bit clear
15-0	RESERVED	R	0h	

10.3.8 CLKOVR (Offset = 1010h) [Reset = 00000000h]

CLKOVR is shown in Figure 10-11 and described in Table 10-11.

Return to the Summary Table.

This register overrides the functional clock request by this peripheral to the system

Figure 10-11. 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 10-11. 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 0h = Run/ungate functional clock 1h = Stop/gate functional clock
0	OVERRIDE	R/W	0h	Unlocks the functionality of RUN_STOP to override the automatic peripheral clock request 0h = Override disabled 1h = Override enabled

10.3.9 PDBGCTL (Offset = 1018h) [Reset = 00000001h]

PDBGCTL is shown in Figure 10-12 and described in Table 10-12.

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 10-12. 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 10-12. PDBGCTL Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	FREE	R/W	1h	Free run control 0h = The peripheral freezes functionality while the Core Halted input is asserted and resumes when it is deasserted. 1h = The peripheral ignores the state of the Core Halted input

10.3.10 IIDX (Offset = 1020h) [Reset = 00000000h]

IIDX is shown in Figure 10-13 and described in Table 10-13.

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 10-13. 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 10-13. IIDX Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	STAT	R	0h	Interrupt index status 0h = No bit is set means there is no pending interrupt request 1h = DIO0 interrupt 2h = DIO1 interrupt 3h = DIO2 interrupt 4h = DIO3 interrupt 5h = DIO4 interrupt 6h = DIO5 interrupt 7h = DIO6 interrupt 8h = DIO7 interrupt 9h = DIO8 interrupt Ah = DIO9 interrupt Bh = DIO10 interrupt Ch = DIO11 interrupt Dh = DIO12 interrupt Eh = DIO13 interrupt Fh = DIO14 interrupt 10h = DIO15 interrupt 11h = DIO16 interrupt 12h = DIO17 interrupt 13h = DIO18 interrupt 14h = DIO19 interrupt 15h = DIO20 interrupt 16h = DIO21 interrupt 17h = DIO22 interrupt 18h = DIO23 interrupt 19h = DIO24 interrupt 1Ah = DIO25 interrupt 1Bh = DIO26 interrupt 1Ch = DIO27 interrupt 1Dh = DIO28 interrupt 1Eh = DIO29 interrupt 1Fh = DIO30 interrupt 20h = DIO31 interrupt

10.3.11 IMASK (Offset = 1028h) [Reset = 00000000h]

IMASK is shown in Figure 10-14 and described in Table 10-14.

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 10-14. 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 10-14. IMASK Field Descriptions

Bit	Field	Type	Reset	Description
31	DIO31	R/W	0h	DIO31 event mask 0h = Event is masked 1h = Event is unmasked
30	DIO30	R/W	0h	DIO30 event mask 0h = Event is masked 1h = Event is unmasked
29	DIO29	R/W	0h	DIO29 event mask 0h = Event is masked 1h = Event is unmasked
28	DIO28	R/W	0h	DIO28 event mask 0h = Event is masked 1h = Event is unmasked
27	DIO27	R/W	0h	DIO27 event mask 0h = Event is masked 1h = Event is unmasked
26	DIO26	R/W	0h	DIO26 event mask 0h = Event is masked 1h = Event is unmasked
25	DIO25	R/W	0h	DIO25 event mask 0h = Event is masked 1h = Event is unmasked
24	DIO24	R/W	0h	DIO24 event mask 0h = Event is masked 1h = Event is unmasked
23	DIO23	R/W	0h	DIO23 event mask 0h = Event is masked 1h = Event is unmasked
22	DIO22	R/W	0h	DIO22 event mask 0h = Event is masked 1h = Event is unmasked

Table 10-14. IMASK Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	DIO21	R/W	0h	DIO21 event mask 0h = Event is masked 1h = Event is unmasked
20	DIO20	R/W	0h	DIO20 event mask 0h = Event is masked 1h = Event is unmasked
19	DIO19	R/W	0h	DIO19 event mask 0h = Event is masked 1h = Event is unmasked
18	DIO18	R/W	0h	DIO18 event mask 0h = Event is masked 1h = Event is unmasked
17	DIO17	R/W	0h	DIO17 event mask 0h = Event is masked 1h = Event is unmasked
16	DIO16	R/W	0h	DIO16 event mask 0h = Event is masked 1h = Event is unmasked
15	DIO15	R/W	0h	DIO15 event mask 0h = Event is masked 1h = Event is unmasked
14	DIO14	R/W	0h	DIO14 event mask 0h = Event is masked 1h = Event is unmasked
13	DIO13	R/W	0h	DIO13 event mask 0h = Event is masked 1h = Event is unmasked
12	DIO12	R/W	0h	DIO12 event mask 0h = Event is masked 1h = Event is unmasked
11	DIO11	R/W	0h	DIO11 event mask 0h = Event is masked 1h = Event is unmasked
10	DIO10	R/W	0h	DIO10 event mask 0h = Event is masked 1h = Event is unmasked
9	DIO9	R/W	0h	DIO9 event mask 0h = Event is masked 1h = Event is unmasked
8	DIO8	R/W	0h	DIO8 event mask 0h = Event is masked 1h = Event is unmasked
7	DIO7	R/W	0h	DIO7 event mask 0h = Event is masked 1h = Event is unmasked
6	DIO6	R/W	0h	DIO6 event mask 0h = Event is masked 1h = Event is unmasked
5	DIO5	R/W	0h	DIO5 event mask 0h = Event is masked 1h = Event is unmasked
4	DIO4	R/W	0h	DIO4 event mask 0h = Event is masked 1h = Event is unmasked
3	DIO3	R/W	0h	DIO3 event mask 0h = Event is masked 1h = Event is unmasked

Table 10-14. IMASK Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	DIO2	R/W	0h	DIO2 event mask 0h = Event is masked 1h = Event is unmasked
1	DIO1	R/W	0h	DIO1 event mask 0h = Event is masked 1h = Event is unmasked
0	DIO0	R/W	0h	DIO0 event mask 0h = Event is masked 1h = Event is unmasked

10.3.12 RIS (Offset = 1030h) [Reset = 00000000h]

RIS is shown in Figure 10-15 and described in Table 10-15.

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 10-15. 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 10-15. RIS Field Descriptions

Bit	Field	Type	Reset	Description
31	DIO31	R	0h	DIO31 event 0h = DIO31 event did not occur 1h = DIO31 event occurred
30	DIO30	R	0h	DIO30 event 0h = DIO30 event did not occur 1h = DIO30 event occurred
29	DIO29	R	0h	DIO29 event 0h = DIO29 event did not occur 1h = DIO29 event occurred
28	DIO28	R	0h	DIO28 event 0h = DIO28 event did not occur 1h = DIO28 event occurred
27	DIO27	R	0h	DIO27 event 0h = DIO27 event did not occur 1h = DIO27 event occurred
26	DIO26	R	0h	DIO26 event 0h = DIO26 event did not occur 1h = DIO26 event occurred
25	DIO25	R	0h	DIO25 event 0h = DIO25 event did not occur 1h = DIO25 event occurred
24	DIO24	R	0h	DIO24 event 0h = DIO24 event did not occur 1h = DIO24 event occurred
23	DIO23	R	0h	DIO23 event 0h = DIO23 event did not occur 1h = DIO23 event occurred
22	DIO22	R	0h	DIO22 event 0h = DIO22 event did not occur 1h = DIO22 event occurred

Table 10-15. RIS Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	DIO21	R	0h	DIO21 event 0h = DIO21 event did not occur 1h = DIO21 event occurred
20	DIO20	R	0h	DIO20 event 0h = DIO20 event did not occur 1h = DIO20 event occurred
19	DIO19	R	0h	DIO19 event 0h = DIO19 event did not occur 1h = DIO19 event occurred
18	DIO18	R	0h	DIO18 event 0h = DIO18 event did not occur 1h = DIO18 event occurred
17	DIO17	R	0h	DIO17 event 0h = DIO17 event did not occur 1h = DIO17 event occurred
16	DIO16	R	0h	DIO16 event 0h = DIO16 event did not occur 1h = DIO16 event occurred
15	DIO15	R	0h	DIO15 event 0h = DIO15 event did not occur 1h = DIO15 event occurred
14	DIO14	R	0h	DIO14 event 0h = DIO14 event did not occur 1h = DIO14 event occurred
13	DIO13	R	0h	DIO13 event 0h = DIO13 event did not occur 1h = DIO13 event occurred
12	DIO12	R	0h	DIO12 event 0h = DIO12 event did not occur 1h = DIO12 event occurred
11	DIO11	R	0h	DIO11 event 0h = DIO11 event did not occur 1h = DIO11 event occurred
10	DIO10	R	0h	DIO10 event 0h = DIO10 event did not occur 1h = DIO10 event occurred
9	DIO9	R	0h	DIO9 event 0h = DIO9 event did not occur 1h = DIO9 event occurred
8	DIO8	R	0h	DIO8 event 0h = DIO8 event did not occur 1h = DIO8 event occurred
7	DIO7	R	0h	DIO7 event 0h = DIO7 event did not occur 1h = DIO7 event occurred
6	DIO6	R	0h	DIO6 event 0h = DIO6 event did not occur 1h = DIO6 event occurred
5	DIO5	R	0h	DIO5 event 0h = DIO5 event did not occur 1h = DIO5 event occurred
4	DIO4	R	0h	DIO4 event 0h = DIO4 event did not occur 1h = DIO4 event occurred
3	DIO3	R	0h	DIO3 event 0h = DIO3 event did not occur 1h = DIO3 event occurred

Table 10-15. RIS Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	DIO2	R	0h	DIO2 event 0h = DIO2 event did not occur 1h = DIO2 event occurred
1	DIO1	R	0h	DIO1 event 0h = DIO1 event did not occur 1h = DIO1 event occurred
0	DIO0	R	0h	DIO0 event 0h = DIO0 event did not occur 1h = DIO0 event occurred

10.3.13 MIS (Offset = 1038h) [Reset = 00000000h]

MIS is shown in Figure 10-16 and described in Table 10-16.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Figure 10-16. 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 10-16. MIS Field Descriptions

Bit	Field	Type	Reset	Description
31	DIO31	R	0h	DIO31 event 0h = DIO31 event did not occur 1h = DIO31 event occurred
30	DIO30	R	0h	DIO30 event 0h = DIO30 event did not occur 1h = DIO30 event occurred
29	DIO29	R	0h	DIO29 event 0h = DIO29 event did not occur 1h = DIO29 event occurred
28	DIO28	R	0h	DIO28 event 0h = DIO28 event did not occur 1h = DIO28 event occurred
27	DIO27	R	0h	DIO27 event 0h = DIO27 event did not occur 1h = DIO27 event occurred
26	DIO26	R	0h	DIO26 event 0h = DIO26 event did not occur 1h = DIO26 event occurred
25	DIO25	R	0h	DIO25 event 0h = DIO25 event did not occur 1h = DIO25 event occurred
24	DIO24	R	0h	DIO24 event 0h = DIO24 event did not occur 1h = DIO24 event occurred
23	DIO23	R	0h	DIO23 event 0h = DIO23 event did not occur 1h = DIO23 event occurred
22	DIO22	R	0h	DIO22 event 0h = DIO22 event did not occur 1h = DIO22 event occurred

Table 10-16. MIS Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	DIO21	R	0h	DIO21 event 0h = DIO21 event did not occur 1h = DIO21 event occurred
20	DIO20	R	0h	DIO20 event 0h = DIO20 event did not occur 1h = DIO20 event occurred
19	DIO19	R	0h	DIO19 event 0h = DIO19 event did not occur 1h = DIO19 event occurred
18	DIO18	R	0h	DIO18 event 0h = DIO18 event did not occur 1h = DIO18 event occurred
17	DIO17	R	0h	DIO17 event 0h = DIO17 event did not occur 1h = DIO17 event occurred
16	DIO16	R	0h	DIO16 event 0h = DIO16 event did not occur 1h = DIO16 event occurred
15	DIO15	R	0h	DIO15 event 0h = DIO15 event did not occur 1h = DIO15 event occurred
14	DIO14	R	0h	DIO14 event 0h = DIO14 event did not occur 1h = DIO14 event occurred
13	DIO13	R	0h	DIO13 event 0h = DIO13 event did not occur 1h = DIO13 event occurred
12	DIO12	R	0h	DIO12 event 0h = DIO12 event did not occur 1h = DIO12 event occurred
11	DIO11	R	0h	DIO11 event 0h = DIO11 event did not occur 1h = DIO11 event occurred
10	DIO10	R	0h	DIO10 event 0h = DIO10 event did not occur 1h = DIO10 event occurred
9	DIO9	R	0h	DIO9 event 0h = DIO9 event did not occur 1h = DIO9 event occurred
8	DIO8	R	0h	DIO8 event 0h = DIO8 event did not occur 1h = DIO8 event occurred
7	DIO7	R	0h	DIO7 event 0h = DIO7 event did not occur 1h = DIO7 event occurred
6	DIO6	R	0h	DIO6 event 0h = DIO6 event did not occur 1h = DIO6 event occurred
5	DIO5	R	0h	DIO5 event 0h = DIO5 event did not occur 1h = DIO5 event occurred
4	DIO4	R	0h	DIO4 event 0h = DIO4 event did not occur 1h = DIO4 event occurred
3	DIO3	R	0h	DIO3 event 0h = DIO3 event did not occur 1h = DIO3 event occurred

Table 10-16. MIS Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	DIO2	R	0h	DIO2 event 0h = DIO2 event did not occur 1h = DIO2 event occurred
1	DIO1	R	0h	DIO1 event 0h = DIO1 event did not occur 1h = DIO1 event occurred
0	DIO0	R	0h	DIO0 event 0h = DIO0 event did not occur 1h = DIO0 event occurred

10.3.14 ISET (Offset = 1040h) [Reset = 00000000h]

ISET is shown in Figure 10-17 and described in Table 10-17.

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 10-17. 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 10-17. ISET Field Descriptions

Bit	Field	Type	Reset	Description
31	DIO31	W	0h	DIO31 event 0h = No effect 1h = Sets DIO31 in RIS register
30	DIO30	W	0h	DIO30 event 0h = No effect 1h = Sets DIO30 in RIS register
29	DIO29	W	0h	DIO29 event 0h = No effect 1h = Sets DIO29 in RIS register
28	DIO28	W	0h	DIO28 event 0h = No effect 1h = Sets DIO28 in RIS register
27	DIO27	W	0h	DIO27 event 0h = No effect 1h = Sets DIO27 in RIS register
26	DIO26	W	0h	DIO26 event 0h = No effect 1h = Sets DIO26 in RIS register
25	DIO25	W	0h	DIO25 event 0h = No effect 1h = Sets DIO25 in RIS register
24	DIO24	W	0h	DIO24 event 0h = No effect 1h = Sets DIO24 in RIS register
23	DIO23	W	0h	DIO23 event 0h = No effect 1h = Sets DIO23 in RIS register
22	DIO22	W	0h	DIO22 event 0h = No effect 1h = Sets DIO22 in RIS register

Table 10-17. ISET Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	DIO21	W	0h	DIO21 event 0h = No effect 1h = Sets DIO21 in RIS register
20	DIO20	W	0h	DIO20 event 0h = No effect 1h = Sets DIO20 in RIS register
19	DIO19	W	0h	DIO19 event 0h = No effect 1h = Sets DIO19 in RIS register
18	DIO18	W	0h	DIO18 event 0h = No effect 1h = Sets DIO18 in RIS register
17	DIO17	W	0h	DIO17 event 0h = No effect 1h = Sets DIO17 in RIS register
16	DIO16	W	0h	DIO16 event 0h = No effect 1h = Sets DIO16 in RIS register
15	DIO15	W	0h	DIO15 event 0h = No effect 1h = Sets DIO15 in RIS register
14	DIO14	W	0h	DIO14 event 0h = No effect 1h = Sets DIO14 in RIS register
13	DIO13	W	0h	DIO13 event 0h = No effect 1h = Sets DIO13 in RIS register
12	DIO12	W	0h	DIO12 event 0h = No effect 1h = Sets DIO12 in RIS register
11	DIO11	W	0h	DIO11 event 0h = No effect 1h = Sets DIO11 in RIS register
10	DIO10	W	0h	DIO10 event 0h = No effect 1h = Sets DIO10 in RIS register
9	DIO9	W	0h	DIO9 event 0h = No effect 1h = Sets DIO9 in RIS register
8	DIO8	W	0h	DIO8 event 0h = No effect 1h = Sets DIO8 in RIS register
7	DIO7	W	0h	DIO7 event 0h = No effect 1h = Sets DIO7 in RIS register
6	DIO6	W	0h	DIO6 event 0h = No effect 1h = Sets DIO6 in RIS register
5	DIO5	W	0h	DIO5 event 0h = No effect 1h = Sets DIO5 in RIS register
4	DIO4	W	0h	DIO4 event 0h = No effect 1h = Sets DIO4 in RIS register
3	DIO3	W	0h	DIO3 event 0h = No effect 1h = Sets DIO3 in RIS register

Table 10-17. ISET Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	DIO2	W	0h	DIO2 event 0h = No effect 1h = Sets DIO2 in RIS register
1	DIO1	W	0h	DIO1 event 0h = No effect 1h = Sets DIO1 in RIS register
0	DIO0	W	0h	DIO0 event 0h = No effect 1h = Sets DIO0 in RIS register

10.3.15 ICLR (Offset = 1048h) [Reset = 00000000h]

ICLR is shown in Figure 10-18 and described in Table 10-18.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

Figure 10-18. 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 10-18. ICLR Field Descriptions

Bit	Field	Type	Reset	Description
31	DIO31	W	0h	DIO31 event 0h = No effect 1h = Clears DIO31 in RIS register
30	DIO30	W	0h	DIO30 event 0h = No effect 1h = Clears DIO30 in RIS register
29	DIO29	W	0h	DIO29 event 0h = No effect 1h = Clears DIO29 in RIS register
28	DIO28	W	0h	DIO28 event 0h = No effect 1h = Clears DIO28 in RIS register
27	DIO27	W	0h	DIO27 event 0h = No effect 1h = Clears DIO27 in RIS register
26	DIO26	W	0h	DIO26 event 0h = No effect 1h = Clears DIO26 in RIS register
25	DIO25	W	0h	DIO25 event 0h = No effect 1h = Clears DIO25 in RIS register
24	DIO24	W	0h	DIO24 event 0h = No effect 1h = Clears DIO24 in RIS register
23	DIO23	W	0h	DIO23 event 0h = No effect 1h = Clears DIO23 in RIS register
22	DIO22	W	0h	DIO22 event 0h = No effect 1h = Clears DIO22 in RIS register

Table 10-18. ICLR Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	DIO21	W	0h	DIO21 event 0h = No effect 1h = Clears DIO21 in RIS register
20	DIO20	W	0h	DIO20 event 0h = No effect 1h = Clears DIO20 in RIS register
19	DIO19	W	0h	DIO19 event 0h = No effect 1h = Clears DIO19 in RIS register
18	DIO18	W	0h	DIO18 event 0h = No effect 1h = Clears DIO18 in RIS register
17	DIO17	W	0h	DIO17 event 0h = No effect 1h = Clears DIO17 in RIS register
16	DIO16	W	0h	DIO16 event 0h = No effect 1h = Clears DIO16 in RIS register
15	DIO15	W	0h	DIO15 event 0h = No effect 1h = Clears DIO15 in RIS register
14	DIO14	W	0h	DIO14 event 0h = No effect 1h = Clears DIO14 in RIS register
13	DIO13	W	0h	DIO13 event 0h = No effect 1h = Clears DIO13 in RIS register
12	DIO12	W	0h	DIO12 event 0h = No effect 1h = Clears DIO12 in RIS register
11	DIO11	W	0h	DIO11 event 0h = No effect 1h = Clears DIO11 in RIS register
10	DIO10	W	0h	DIO10 event 0h = No effect 1h = Clears DIO10 in RIS register
9	DIO9	W	0h	DIO9 event 0h = No effect 1h = Clears DIO9 in RIS register
8	DIO8	W	0h	DIO8 event 0h = No effect 1h = Clears DIO8 in RIS register
7	DIO7	W	0h	DIO7 event 0h = No effect 1h = Clears DIO7 in RIS register
6	DIO6	W	0h	DIO6 event 0h = No effect 1h = Clears DIO6 in RIS register
5	DIO5	W	0h	DIO5 event 0h = No effect 1h = Clears DIO5 in RIS register
4	DIO4	W	0h	DIO4 event 0h = No effect 1h = Clears DIO4 in RIS register
3	DIO3	W	0h	DIO3 event 0h = No effect 1h = Clears DIO3 in RIS register

Table 10-18. ICLR Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	DIO2	W	0h	DIO2 event 0h = No effect 1h = Clears DIO2 in RIS register
1	DIO1	W	0h	DIO1 event 0h = No effect 1h = Clears DIO1 in RIS register
0	DIO0	W	0h	DIO0 event 0h = No effect 1h = Clears DIO0 in RIS register

10.3.16 IIDX (Offset = 1050h) [Reset = 00000000h]

IIDX is shown in Figure 10-19 and described in Table 10-19.

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 10-19. 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 10-19. IIDX Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	STAT	R	0h	Interrupt index status 0h = No bit is set means there is no pending interrupt request 1h = DIO0 interrupt 2h = DIO1 interrupt 3h = DIO2 interrupt 4h = DIO3 interrupt 5h = DIO4 interrupt 6h = DIO5 interrupt 7h = DIO6 interrupt 8h = DIO7 interrupt 9h = DIO8 interrupt Ah = DIO9 interrupt Bh = DIO10 interrupt Ch = DIO11 interrupt Dh = DIO12 interrupt Eh = DIO13 interrupt Fh = DIO14 interrupt 10h = DIO15 interrupt

10.3.17 IMASK (Offset = 1058h) [Reset = 00000000h]

IMASK is shown in Figure 10-20 and described in Table 10-20.

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 10-20. 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 10-20. IMASK Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	
15	DIO15	R/W	0h	DIO15 event mask 0h = Event is masked 1h = Event is unmasked
14	DIO14	R/W	0h	DIO14 event mask 0h = Event is masked 1h = Event is unmasked
13	DIO13	R/W	0h	DIO13 event mask 0h = Event is masked 1h = Event is unmasked
12	DIO12	R/W	0h	DIO12 event mask 0h = Event is masked 1h = Event is unmasked
11	DIO11	R/W	0h	DIO11 event mask 0h = Event is masked 1h = Event is unmasked
10	DIO10	R/W	0h	DIO10 event mask 0h = Event is masked 1h = Event is unmasked
9	DIO9	R/W	0h	DIO9 event mask 0h = Event is masked 1h = Event is unmasked
8	DIO8	R/W	0h	DIO8 event mask 0h = Event is masked 1h = Event is unmasked
7	DIO7	R/W	0h	DIO7 event mask 0h = Event is masked 1h = Event is unmasked
6	DIO6	R/W	0h	DIO6 event mask 0h = Event is masked 1h = Event is unmasked

Table 10-20. IMASK Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	DIO5	R/W	0h	DIO5 event mask 0h = Event is masked 1h = Event is unmasked
4	DIO4	R/W	0h	DIO4 event mask 0h = Event is masked 1h = Event is unmasked
3	DIO3	R/W	0h	DIO3 event mask 0h = Event is masked 1h = Event is unmasked
2	DIO2	R/W	0h	DIO2 event mask 0h = Event is masked 1h = Event is unmasked
1	DIO1	R/W	0h	DIO1 event mask 0h = Event is masked 1h = Event is unmasked
0	DIO0	R/W	0h	DIO0 event mask 0h = Event is masked 1h = Event is unmasked

10.3.18 RIS (Offset = 1060h) [Reset = 00000000h]

RIS is shown in Figure 10-21 and described in Table 10-21.

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 10-21. 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 10-21. RIS Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15	DIO15	R	0h	DIO15 event 0h = DIO15 event did not occur 1h = DIO15 event occurred
14	DIO14	R	0h	DIO14 event 0h = DIO14 event did not occur 1h = DIO14 event occurred
13	DIO13	R	0h	DIO13 event 0h = DIO13 event did not occur 1h = DIO13 event occurred
12	DIO12	R	0h	DIO12 event 0h = DIO12 event did not occur 1h = DIO12 event occurred
11	DIO11	R	0h	DIO11 event 0h = DIO11 event did not occur 1h = DIO11 event occurred
10	DIO10	R	0h	DIO10 event 0h = DIO10 event did not occur 1h = DIO10 event occurred
9	DIO9	R	0h	DIO9 event 0h = DIO9 event did not occur 1h = DIO9 event occurred
8	DIO8	R	0h	DIO8 event 0h = DIO8 event did not occur 1h = DIO8 event occurred
7	DIO7	R	0h	DIO7 event 0h = DIO7 event did not occur 1h = DIO7 event occurred

Table 10-21. RIS Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	DIO6	R	0h	DIO6 event 0h = DIO6 event did not occur 1h = DIO6 event occurred
5	DIO5	R	0h	DIO5 event 0h = DIO5 event did not occur 1h = DIO5 event occurred
4	DIO4	R	0h	DIO4 event 0h = DIO4 event did not occur 1h = DIO4 event occurred
3	DIO3	R	0h	DIO3 event 0h = DIO3 event did not occur 1h = DIO3 event occurred
2	DIO2	R	0h	DIO2 event 0h = DIO2 event did not occur 1h = DIO2 event occurred
1	DIO1	R	0h	DIO1 event 0h = DIO1 event did not occur 1h = DIO1 event occurred
0	DIO0	R	0h	DIO0 event 0h = DIO0 event did not occur 1h = DIO0 event occurred

10.3.19 MIS (Offset = 1068h) [Reset = 00000000h]

MIS is shown in Figure 10-22 and described in Table 10-22.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Figure 10-22. 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 10-22. MIS Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15	DIO15	R	0h	DIO15 event 0h = DIO15 event did not occur 1h = DIO15 event occurred
14	DIO14	R	0h	DIO14 event 0h = DIO14 event did not occur 1h = DIO14 event occurred
13	DIO13	R	0h	DIO13 event 0h = DIO13 event did not occur 1h = DIO13 event occurred
12	DIO12	R	0h	DIO12 event 0h = DIO12 event did not occur 1h = DIO12 event occurred
11	DIO11	R	0h	DIO11 event 0h = DIO11 event did not occur 1h = DIO11 event occurred
10	DIO10	R	0h	DIO10 event 0h = DIO10 event did not occur 1h = DIO10 event occurred
9	DIO9	R	0h	DIO9 event 0h = DIO9 event did not occur 1h = DIO9 event occurred
8	DIO8	R	0h	DIO8 event 0h = DIO8 event did not occur 1h = DIO8 event occurred
7	DIO7	R	0h	DIO7 event 0h = DIO7 event did not occur 1h = DIO7 event occurred
6	DIO6	R	0h	DIO6 event 0h = DIO6 event did not occur 1h = DIO6 event occurred

Table 10-22. MIS Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	DIO5	R	0h	DIO5 event 0h = DIO5 event did not occur 1h = DIO5 event occurred
4	DIO4	R	0h	DIO4 event 0h = DIO4 event did not occur 1h = DIO4 event occurred
3	DIO3	R	0h	DIO3 event 0h = DIO3 event did not occur 1h = DIO3 event occurred
2	DIO2	R	0h	DIO2 event 0h = DIO2 event did not occur 1h = DIO2 event occurred
1	DIO1	R	0h	DIO1 event 0h = DIO1 event did not occur 1h = DIO1 event occurred
0	DIO0	R	0h	DIO0 event 0h = DIO0 event did not occur 1h = DIO0 event occurred

10.3.20 ISET (Offset = 1070h) [Reset = 00000000h]

ISET is shown in Figure 10-23 and described in Table 10-23.

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 10-23. 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 10-23. ISET Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	W	0h	
15	DIO15	W	0h	DIO15 event 0h = No effect 1h = Sets DIO15 in RIS register
14	DIO14	W	0h	DIO14 event 0h = No effect 1h = Sets DIO14 in RIS register
13	DIO13	W	0h	DIO13 event 0h = No effect 1h = Sets DIO13 in RIS register
12	DIO12	W	0h	DIO12 event 0h = No effect 1h = Sets DIO12 in RIS register
11	DIO11	W	0h	DIO11 event 0h = No effect 1h = Sets DIO11 in RIS register
10	DIO10	W	0h	DIO10 event 0h = No effect 1h = Sets DIO10 in RIS register
9	DIO9	W	0h	DIO9 event 0h = No effect 1h = Sets DIO9 in RIS register
8	DIO8	W	0h	DIO8 event 0h = No effect 1h = Sets DIO8 in RIS register
7	DIO7	W	0h	DIO7 event 0h = No effect 1h = Sets DIO7 in RIS register

Table 10-23. ISET Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	DIO6	W	0h	DIO6 event 0h = No effect 1h = Sets DIO6 in RIS register
5	DIO5	W	0h	DIO5 event 0h = No effect 1h = Sets DIO5 in RIS register
4	DIO4	W	0h	DIO4 event 0h = No effect 1h = Sets DIO4 in RIS register
3	DIO3	W	0h	DIO3 event 0h = No effect 1h = Sets DIO3 in RIS register
2	DIO2	W	0h	DIO2 event 0h = No effect 1h = Sets DIO2 in RIS register
1	DIO1	W	0h	DIO1 event 0h = No effect 1h = Sets DIO1 in RIS register
0	DIO0	W	0h	DIO0 event 0h = No effect 1h = Sets DIO0 in RIS register

10.3.21 ICLR (Offset = 1078h) [Reset = 00000000h]

ICLR is shown in Figure 10-24 and described in Table 10-24.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

Figure 10-24. 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 10-24. ICLR Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	W	0h	
15	DIO15	W	0h	DIO15 event 0h = No effect 1h = Clears DIO15 in RIS register
14	DIO14	W	0h	DIO14 event 0h = No effect 1h = Clears DIO14 in RIS register
13	DIO13	W	0h	DIO13 event 0h = No effect 1h = Clears DIO13 in RIS register
12	DIO12	W	0h	DIO12 event 0h = No effect 1h = Clears DIO12 in RIS register
11	DIO11	W	0h	DIO11 event 0h = No effect 1h = Clears DIO11 in RIS register
10	DIO10	W	0h	DIO10 event 0h = No effect 1h = Clears DIO10 in RIS register
9	DIO9	W	0h	DIO9 event 0h = No effect 1h = Clears DIO9 in RIS register
8	DIO8	W	0h	DIO8 event 0h = No effect 1h = Clears DIO8 in RIS register
7	DIO7	W	0h	DIO7 event 0h = No effect 1h = Clears DIO7 in RIS register
6	DIO6	W	0h	DIO6 event 0h = No effect 1h = Clears DIO6 in RIS register

Table 10-24. ICLR Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	DIO5	W	0h	DIO5 event 0h = No effect 1h = Clears DIO5 in RIS register
4	DIO4	W	0h	DIO4 event 0h = No effect 1h = Clears DIO4 in RIS register
3	DIO3	W	0h	DIO3 event 0h = No effect 1h = Clears DIO3 in RIS register
2	DIO2	W	0h	DIO2 event 0h = No effect 1h = Clears DIO2 in RIS register
1	DIO1	W	0h	DIO1 event 0h = No effect 1h = Clears DIO1 in RIS register
0	DIO0	W	0h	DIO0 event 0h = No effect 1h = Clears DIO0 in RIS register

10.3.22 IIDX (Offset = 1080h) [Reset = 00000000h]

IIDX is shown in Figure 10-25 and described in Table 10-25.

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 10-25. 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 10-25. IIDX Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	STAT	R	0h	Interrupt index status 0h = No bit is set means there is no pending interrupt request 1h = DIO0 interrupt 2h = DIO1 interrupt 3h = DIO2 interrupt 4h = DIO3 interrupt 5h = DIO4 interrupt 6h = DIO5 interrupt 7h = DIO6 interrupt 8h = DIO7 interrupt 9h = DIO8 interrupt Ah = DIO9 interrupt Bh = DIO10 interrupt Ch = DIO11 interrupt Dh = DIO12 interrupt Eh = DIO13 interrupt Fh = DIO14 interrupt 10h = DIO15 interrupt

10.3.23 IMASK (Offset = 1088h) [Reset = 00000000h]

IMASK is shown in Figure 10-26 and described in Table 10-26.

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 10-26. 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 10-26. IMASK Field Descriptions

Bit	Field	Type	Reset	Description
31	DIO31	R/W	0h	DIO31 event mask 0h = Event is masked 1h = Event is unmasked
30	DIO30	R/W	0h	DIO30 event mask 0h = Event is masked 1h = Event is unmasked
29	DIO29	R/W	0h	DIO29 event mask 0h = Event is masked 1h = Event is unmasked
28	DIO28	R/W	0h	DIO28 event mask 0h = Event is masked 1h = Event is unmasked
27	DIO27	R/W	0h	DIO27 event mask 0h = Event is masked 1h = Event is unmasked
26	DIO26	R/W	0h	DIO26 event mask 0h = Event is masked 1h = Event is unmasked
25	DIO25	R/W	0h	DIO25 event mask 0h = Event is masked 1h = Event is unmasked
24	DIO24	R/W	0h	DIO24 event mask 0h = Event is masked 1h = Event is unmasked
23	DIO23	R/W	0h	DIO23 event mask 0h = Event is masked 1h = Event is unmasked
22	DIO22	R/W	0h	DIO22 event mask 0h = Event is masked 1h = Event is unmasked

Table 10-26. IMASK Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	DIO21	R/W	0h	DIO21 event mask 0h = Event is masked 1h = Event is unmasked
20	DIO20	R/W	0h	DIO20 event mask 0h = Event is masked 1h = Event is unmasked
19	DIO19	R/W	0h	DIO19 event mask 0h = Event is masked 1h = Event is unmasked
18	DIO18	R/W	0h	DIO18 event mask 0h = Event is masked 1h = Event is unmasked
17	DIO17	R/W	0h	DIO17 event mask 0h = Event is masked 1h = Event is unmasked
16	DIO16	R/W	0h	DIO16 event mask 0h = Event is masked 1h = Event is unmasked
15-0	RESERVED	R/W	0h	

10.3.24 RIS (Offset = 1090h) [Reset = 00000000h]

RIS is shown in Figure 10-27 and described in Table 10-27.

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 10-27. 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 10-27. RIS Field Descriptions

Bit	Field	Type	Reset	Description
31	DIO31	R	0h	DIO31 event 0h = DIO31 event did not occur 1h = DIO31 event occurred
30	DIO30	R	0h	DIO30 event 0h = DIO30 event did not occur 1h = DIO30 event occurred
29	DIO29	R	0h	DIO29 event 0h = DIO29 event did not occur 1h = DIO29 event occurred
28	DIO28	R	0h	DIO28 event 0h = DIO28 event did not occur 1h = DIO28 event occurred
27	DIO27	R	0h	DIO27 event 0h = DIO27 event did not occur 1h = DIO27 event occurred
26	DIO26	R	0h	DIO26 event 0h = DIO26 event did not occur 1h = DIO26 event occurred
25	DIO25	R	0h	DIO25 event 0h = DIO25 event did not occur 1h = DIO25 event occurred
24	DIO24	R	0h	DIO24 event 0h = DIO24 event did not occur 1h = DIO24 event occurred
23	DIO23	R	0h	DIO23 event 0h = DIO23 event did not occur 1h = DIO23 event occurred
22	DIO22	R	0h	DIO22 event 0h = DIO22 event did not occur 1h = DIO22 event occurred

Table 10-27. RIS Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	DIO21	R	0h	DIO21 event 0h = DIO21 event did not occur 1h = DIO21 event occurred
20	DIO20	R	0h	DIO20 event 0h = DIO20 event did not occur 1h = DIO20 event occurred
19	DIO19	R	0h	DIO19 event 0h = DIO19 event did not occur 1h = DIO19 event occurred
18	DIO18	R	0h	DIO18 event 0h = DIO18 event did not occur 1h = DIO18 event occurred
17	DIO17	R	0h	DIO17 event 0h = DIO17 event did not occur 1h = DIO17 event occurred
16	DIO16	R	0h	DIO16 event 0h = DIO16 event did not occur 1h = DIO16 event occurred
15-0	RESERVED	R	0h	

10.3.25 MIS (Offset = 1098h) [Reset = 00000000h]

MIS is shown in Figure 10-28 and described in Table 10-28.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Figure 10-28. 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 10-28. MIS Field Descriptions

Bit	Field	Type	Reset	Description
31	DIO31	R	0h	DIO31 event 0h = DIO31 event did not occur 1h = DIO31 event occurred
30	DIO30	R	0h	DIO30 event 0h = DIO30 event did not occur 1h = DIO30 event occurred
29	DIO29	R	0h	DIO29 event 0h = DIO29 event did not occur 1h = DIO29 event occurred
28	DIO28	R	0h	DIO28 event 0h = DIO28 event did not occur 1h = DIO28 event occurred
27	DIO27	R	0h	DIO27 event 0h = DIO27 event did not occur 1h = DIO27 event occurred
26	DIO26	R	0h	DIO26 event 0h = DIO26 event did not occur 1h = DIO26 event occurred
25	DIO25	R	0h	DIO25 event 0h = DIO25 event did not occur 1h = DIO25 event occurred
24	DIO24	R	0h	DIO24 event 0h = DIO24 event did not occur 1h = DIO24 event occurred
23	DIO23	R	0h	DIO23 event 0h = DIO23 event did not occur 1h = DIO23 event occurred
22	DIO22	R	0h	DIO22 event 0h = DIO22 event did not occur 1h = DIO22 event occurred

Table 10-28. MIS Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	DIO21	R	0h	DIO21 event 0h = DIO21 event did not occur 1h = DIO21 event occurred
20	DIO20	R	0h	DIO20 event 0h = DIO20 event did not occur 1h = DIO20 event occurred
19	DIO19	R	0h	DIO19 event 0h = DIO19 event did not occur 1h = DIO19 event occurred
18	DIO18	R	0h	DIO18 event 0h = DIO18 event did not occur 1h = DIO18 event occurred
17	DIO17	R	0h	DIO17 event 0h = DIO17 event did not occur 1h = DIO17 event occurred
16	DIO16	R	0h	DIO16 event 0h = DIO16 event did not occur 1h = DIO16 event occurred
15-0	RESERVED	R	0h	

10.3.26 ISET (Offset = 10A0h) [Reset = 00000000h]

ISET is shown in Figure 10-29 and described in Table 10-29.

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 10-29. 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 10-29. ISET Field Descriptions

Bit	Field	Type	Reset	Description
31	DIO31	W	0h	DIO31 event 0h = No effect 1h = Sets DIO31 in RIS register
30	DIO30	W	0h	DIO30 event 0h = No effect 1h = Sets DIO30 in RIS register
29	DIO29	W	0h	DIO29 event 0h = No effect 1h = Sets DIO29 in RIS register
28	DIO28	W	0h	DIO28 event 0h = No effect 1h = Sets DIO28 in RIS register
27	DIO27	W	0h	DIO27 event 0h = No effect 1h = Sets DIO27 in RIS register
26	DIO26	W	0h	DIO26 event 0h = No effect 1h = Sets DIO26 in RIS register
25	DIO25	W	0h	DIO25 event 0h = No effect 1h = Sets DIO25 in RIS register
24	DIO24	W	0h	DIO24 event 0h = No effect 1h = Sets DIO24 in RIS register
23	DIO23	W	0h	DIO23 event 0h = No effect 1h = Sets DIO23 in RIS register
22	DIO22	W	0h	DIO22 event 0h = No effect 1h = Sets DIO22 in RIS register

Table 10-29. ISET Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	DIO21	W	0h	DIO21 event 0h = No effect 1h = Sets DIO21 in RIS register
20	DIO20	W	0h	DIO20 event 0h = No effect 1h = Sets DIO20 in RIS register
19	DIO19	W	0h	DIO19 event 0h = No effect 1h = Sets DIO19 in RIS register
18	DIO18	W	0h	DIO18 event 0h = No effect 1h = Sets DIO18 in RIS register
17	DIO17	W	0h	DIO17 event 0h = No effect 1h = Sets DIO17 in RIS register
16	DIO16	W	0h	DIO16 event 0h = No effect 1h = Sets DIO16 in RIS register
15-0	RESERVED	W	0h	

10.3.27 ICLR (Offset = 10A8h) [Reset = 00000000h]

ICLR is shown in Figure 10-30 and described in Table 10-30.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

Figure 10-30. 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 10-30. ICLR Field Descriptions

Bit	Field	Type	Reset	Description
31	DIO31	W	0h	DIO31 event 0h = No effect 1h = Clears DIO31 in RIS register
30	DIO30	W	0h	DIO30 event 0h = No effect 1h = Clears DIO30 in RIS register
29	DIO29	W	0h	DIO29 event 0h = No effect 1h = Clears DIO29 in RIS register
28	DIO28	W	0h	DIO28 event 0h = No effect 1h = Clears DIO28 in RIS register
27	DIO27	W	0h	DIO27 event 0h = No effect 1h = Clears DIO27 in RIS register
26	DIO26	W	0h	DIO26 event 0h = No effect 1h = Clears DIO26 in RIS register
25	DIO25	W	0h	DIO25 event 0h = No effect 1h = Clears DIO25 in RIS register
24	DIO24	W	0h	DIO24 event 0h = No effect 1h = Clears DIO24 in RIS register
23	DIO23	W	0h	DIO23 event 0h = No effect 1h = Clears DIO23 in RIS register
22	DIO22	W	0h	DIO22 event 0h = No effect 1h = Clears DIO22 in RIS register

Table 10-30. ICLR Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	DIO21	W	0h	DIO21 event 0h = No effect 1h = Clears DIO21 in RIS register
20	DIO20	W	0h	DIO20 event 0h = No effect 1h = Clears DIO20 in RIS register
19	DIO19	W	0h	DIO19 event 0h = No effect 1h = Clears DIO19 in RIS register
18	DIO18	W	0h	DIO18 event 0h = No effect 1h = Clears DIO18 in RIS register
17	DIO17	W	0h	DIO17 event 0h = No effect 1h = Clears DIO17 in RIS register
16	DIO16	W	0h	DIO16 event 0h = No effect 1h = Clears DIO16 in RIS register
15-0	RESERVED	W	0h	

10.3.28 EVT_MODE (Offset = 10E0h) [Reset = 00000029h]

EVT_MODE is shown in Figure 10-31 and described in Table 10-31.

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 10-31. 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 10-31. 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 0h = The interrupt or event line is disabled. 1h = The interrupt or event line is in software mode. Software must clear the RIS. 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 0h = The interrupt or event line is disabled. 1h = The interrupt or event line is in software mode. Software must clear the RIS. 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 0h = The interrupt or event line is disabled. 1h = The interrupt or event line is in software mode. Software must clear the RIS. 2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag.

10.3.29 DESC (Offset = 10FCh) [Reset = 16110000h]

DESC is shown in Figure 10-32 and described in Table 10-32.

Return to the Summary Table.

This register identifies the peripheral and its exact version.

Figure 10-32. 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 10-32. 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. 0h = Smallest value FFFFh = Highest possible value
15-12	FEATUREVER	R	0h	Feature Set for the module *instance* 0h = Smallest value Fh = Highest possible value
11-8	RESERVED	R	0h	
7-4	MAJREV	R	0h	Major rev of the IP 0h = Smallest value Fh = Highest possible value
3-0	MINREV	R	0h	Minor rev of the IP 0h = Smallest value Fh = Highest possible value

10.3.30 DOUT3_0 (Offset = 1200h) [Reset = 00000000h]

DOUT3_0 is shown in Figure 10-33 and described in Table 10-33.

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 10-33. 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 10-33. 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 1h = Output is set to 1

10.3.31 DOUT7_4 (Offset = 1204h) [Reset = 00000000h]

DOUT7_4 is shown in Figure 10-34 and described in Table 10-34.

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 10-34. 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 10-34. 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 1h = Output is set to 1

10.3.32 DOUT11_8 (Offset = 1208h) [Reset = 00000000h]

DOUT11_8 is shown in Figure 10-35 and described in Table 10-35.

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 10-35. 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 10-35. 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 1h = Output is set to 1

10.3.33 DOUT15_12 (Offset = 120Ch) [Reset = 00000000h]

DOUT15_12 is shown in Figure 10-36 and described in Table 10-36.

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 10-36. 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 10-36. 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 1h = Output is set to 1

10.3.34 DOUT19_16 (Offset = 1210h) [Reset = 00000000h]

DOUT19_16 is shown in Figure 10-37 and described in Table 10-37.

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 10-37. 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 10-37. 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 1h = Output is set to 1

10.3.35 DOUT23_20 (Offset = 1214h) [Reset = 00000000h]

DOUT23_20 is shown in Figure 10-38 and described in Table 10-38.

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 10-38. 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 10-38. 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 1h = Output is set to 1

10.3.36 DOUT27_24 (Offset = 1218h) [Reset = 00000000h]

DOUT27_24 is shown in Figure 10-39 and described in Table 10-39.

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 10-39. 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 10-39. 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 1h = Output is set to 1

10.3.37 DOUT31_28 (Offset = 121Ch) [Reset = 00000000h]

DOUT31_28 is shown in Figure 10-40 and described in Table 10-40.

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 10-40. 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 10-40. 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 1h = Output is set to 1

10.3.38 DOUT31_0 (Offset = 1280h) [Reset = 00000000h]

DOUT31_0 is shown in Figure 10-41 and described in Table 10-41.

Return to the Summary Table.

Data output for pins configured as DIO31 to DIO0.

Figure 10-41. 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 10-41. 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 1h = Output is set to 1

Table 10-41. 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 1h = Output is set to 1

Table 10-41. 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 1h = Output is set to 1

10.3.39 DOUTSET31_0 (Offset = 1290h) [Reset = 00000000h]

DOUTSET31_0 is shown in Figure 10-42 and described in Table 10-42.

Return to the Summary Table.

Writing 1 to a bit position in this register sets the corresponding bit in the DOUT31_0 register.

Figure 10-42. 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 10-42. 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. Writing 0 has no effect. 0h = No effect 1h = Sets DIO31 in DOUT31_0
30	DIO30	W	0h	Writing 1 to this bit sets the DIO30 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO30 in DOUT31_0
29	DIO29	W	0h	Writing 1 to this bit sets the DIO29 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO29 in DOUT31_0
28	DIO28	W	0h	Writing 1 to this bit sets the DIO28 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO28 in DOUT31_0
27	DIO27	W	0h	Writing 1 to this bit sets the DIO27 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO27 in DOUT31_0
26	DIO26	W	0h	Writing 1 to this bit sets the DIO26 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO26 in DOUT31_0
25	DIO25	W	0h	Writing 1 to this bit sets the DIO25 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO25 in DOUT31_0
24	DIO24	W	0h	Writing 1 to this bit sets the DIO24 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO24 in DOUT31_0

Table 10-42. 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. Writing 0 has no effect. 0h = No effect 1h = Sets DIO23 in DOUT31_0
22	DIO22	W	0h	Writing 1 to this bit sets the DIO22 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO22 in DOUT31_0
21	DIO21	W	0h	Writing 1 to this bit sets the DIO21 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO21 in DOUT31_0
20	DIO20	W	0h	Writing 1 to this bit sets the DIO20 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO20 in DOUT31_0
19	DIO19	W	0h	Writing 1 to this bit sets the DIO19 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO19 in DOUT31_0
18	DIO18	W	0h	Writing 1 to this bit sets the DIO18 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO18 in DOUT31_0
17	DIO17	W	0h	Writing 1 to this bit sets the DIO17 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO17 in DOUT31_0
16	DIO16	W	0h	Writing 1 to this bit sets the DIO16 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO16 in DOUT31_0
15	DIO15	W	0h	Writing 1 to this bit sets the DIO15 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO15 in DOUT31_0
14	DIO14	W	0h	Writing 1 to this bit sets the DIO14 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO14 in DOUT31_0
13	DIO13	W	0h	Writing 1 to this bit sets the DIO13 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO13 in DOUT31_0
12	DIO12	W	0h	Writing 1 to this bit sets the DIO12 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO12 in DOUT31_0
11	DIO11	W	0h	Writing 1 to this bit sets the DIO11 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO11 in DOUT31_0
10	DIO10	W	0h	Writing 1 to this bit sets the DIO10 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO10 in DOUT31_0

Table 10-42. 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. Writing 0 has no effect. 0h = No effect 1h = Sets DIO9 in DOUT31_0
8	DIO8	W	0h	Writing 1 to this bit sets the DIO8 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO8 in DOUT31_0
7	DIO7	W	0h	Writing 1 to this bit sets the DIO7 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO7 in DOUT31_0
6	DIO6	W	0h	Writing 1 to this bit sets the DIO6 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO6 in DOUT31_0
5	DIO5	W	0h	Writing 1 to this bit sets the DIO5 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO5 in DOUT31_0
4	DIO4	W	0h	Writing 1 to this bit sets the DIO4 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO4 in DOUT31_0
3	DIO3	W	0h	Writing 1 to this bit sets the DIO3 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO3 in DOUT31_0
2	DIO2	W	0h	Writing 1 to this bit sets the DIO2 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO2 in DOUT31_0
1	DIO1	W	0h	Writing 1 to this bit sets the DIO1 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO1 in DOUT31_0
0	DIO0	W	0h	Writing 1 to this bit sets the DIO0 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO0 in DOUT31_0

10.3.40 DOUTCLR31_0 (Offset = 12A0h) [Reset = 00000000h]

DOUTCLR31_0 is shown in Figure 10-43 and described in Table 10-43.

Return to the Summary Table.

Writing 1 to a bit position in this register clears the corresponding bit in the DOUT31_0 register.

Figure 10-43. 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 10-43. 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. Writing 0 has no effect. 0h = No effect 1h = Clears DIO31 in DOUT31_0
30	DIO30	W	0h	Writing 1 to this bit clears the DIO30 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO30 in DOUT31_0
29	DIO29	W	0h	Writing 1 to this bit clears the DIO29 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO29 in DOUT31_0
28	DIO28	W	0h	Writing 1 to this bit clears the DIO28 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO28 in DOUT31_0
27	DIO27	W	0h	Writing 1 to this bit clears the DIO27 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO27 in DOUT31_0
26	DIO26	W	0h	Writing 1 to this bit clears the DIO26 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO26 in DOUT31_0
25	DIO25	W	0h	Writing 1 to this bit clears the DIO25 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO25 in DOUT31_0
24	DIO24	W	0h	Writing 1 to this bit clears the DIO24 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO24 in DOUT31_0

Table 10-43. 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. Writing 0 has no effect. 0h = No effect 1h = Clears DIO23 in DOUT31_0
22	DIO22	W	0h	Writing 1 to this bit clears the DIO22 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO22 in DOUT31_0
21	DIO21	W	0h	Writing 1 to this bit clears the DIO21 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO21 in DOUT31_0
20	DIO20	W	0h	Writing 1 to this bit clears the DIO20 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO20 in DOUT31_0
19	DIO19	W	0h	Writing 1 to this bit clears the DIO19 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO19 in DOUT31_0
18	DIO18	W	0h	Writing 1 to this bit clears the DIO18 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO18 in DOUT31_0
17	DIO17	W	0h	Writing 1 to this bit clears the DIO17 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO17 in DOUT31_0
16	DIO16	W	0h	Writing 1 to this bit clears the DIO16 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO16 in DOUT31_0
15	DIO15	W	0h	Writing 1 to this bit clears the DIO15 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO15 in DOUT31_0
14	DIO14	W	0h	Writing 1 to this bit clears the DIO14 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO14 in DOUT31_0
13	DIO13	W	0h	Writing 1 to this bit clears the DIO13 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO13 in DOUT31_0
12	DIO12	W	0h	Writing 1 to this bit clears the DIO12 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO12 in DOUT31_0
11	DIO11	W	0h	Writing 1 to this bit clears the DIO11 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO11 in DOUT31_0
10	DIO10	W	0h	Writing 1 to this bit clears the DIO10 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO10 in DOUT31_0

Table 10-43. 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. Writing 0 has no effect. 0h = No effect 1h = Clears DIO9 in DOUT31_0
8	DIO8	W	0h	Writing 1 to this bit clears the DIO8 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO8 in DOUT31_0
7	DIO7	W	0h	Writing 1 to this bit clears the DIO7 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO7 in DOUT31_0
6	DIO6	W	0h	Writing 1 to this bit clears the DIO6 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO6 in DOUT31_0
5	DIO5	W	0h	Writing 1 to this bit clears the DIO5 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO5 in DOUT31_0
4	DIO4	W	0h	Writing 1 to this bit clears the DIO4 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO4 in DOUT31_0
3	DIO3	W	0h	Writing 1 to this bit clears the DIO3 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO3 in DOUT31_0
2	DIO2	W	0h	Writing 1 to this bit clears the DIO2 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO2 in DOUT31_0
1	DIO1	W	0h	Writing 1 to this bit clears the DIO1 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO1 in DOUT31_0
0	DIO0	W	0h	Writing 1 to this bit clears the DIO0 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO0 in DOUT31_0

10.3.41 DOUTTGL31_0 (Offset = 12B0h) [Reset = 00000000h]

DOUTTGL31_0 is shown in Figure 10-44 and described in Table 10-44.

Return to the Summary Table.

Writing 1 to a bit position in this register will invert the corresponding DIO output.

Figure 10-44. 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 10-44. DOUTTGL31_0 Field Descriptions

Bit	Field	Type	Reset	Description
31	DIO31	W	0h	This bit is used to toggle DIO31 output. 0h = No effect 1h = Toggle output
30	DIO30	W	0h	This bit is used to toggle DIO30 output. 0h = No effect 1h = Toggle output
29	DIO29	W	0h	This bit is used to toggle DIO29 output. 0h = No effect 1h = Toggle output
28	DIO28	W	0h	This bit is used to toggle DIO28 output. 0h = No effect 1h = Toggle output
27	DIO27	W	0h	This bit is used to toggle DIO27 output. 0h = No effect 1h = Toggle output
26	DIO26	W	0h	This bit is used to toggle DIO26 output. 0h = No effect 1h = Toggle output
25	DIO25	W	0h	This bit is used to toggle DIO25 output. 0h = No effect 1h = Toggle output
24	DIO24	W	0h	This bit is used to toggle DIO24 output. 0h = No effect 1h = Toggle output
23	DIO23	W	0h	This bit is used to toggle DIO23 output. 0h = No effect 1h = Toggle output
22	DIO22	W	0h	This bit is used to toggle DIO22 output. 0h = No effect 1h = Toggle output

Table 10-44. DOUTTGL31_0 Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	DIO21	W	0h	This bit is used to toggle DIO21 output. 0h = No effect 1h = Toggle output
20	DIO20	W	0h	This bit is used to toggle DIO20 output. 0h = No effect 1h = Toggle output
19	DIO19	W	0h	This bit is used to toggle DIO19 output. 0h = No effect 1h = Toggle output
18	DIO18	W	0h	This bit is used to toggle DIO18 output. 0h = No effect 1h = Toggle output
17	DIO17	W	0h	This bit is used to toggle DIO17 output. 0h = No effect 1h = Toggle output
16	DIO16	W	0h	This bit is used to toggle DIO16 output. 0h = No effect 1h = Toggle output
15	DIO15	W	0h	This bit is used to toggle DIO15 output. 0h = No effect 1h = Toggle output
14	DIO14	W	0h	This bit is used to toggle DIO14 output. 0h = No effect 1h = Toggle output
13	DIO13	W	0h	This bit is used to toggle DIO13 output. 0h = No effect 1h = Toggle output
12	DIO12	W	0h	This bit is used to toggle DIO12 output. 0h = No effect 1h = Toggle output
11	DIO11	W	0h	This bit is used to toggle DIO11 output. 0h = No effect 1h = Toggle output
10	DIO10	W	0h	This bit is used to toggle DIO10 output. 0h = No effect 1h = Toggle output
9	DIO9	W	0h	This bit is used to toggle DIO9 output. 0h = No effect 1h = Toggle output
8	DIO8	W	0h	This bit is used to toggle DIO8 output. 0h = No effect 1h = Toggle output
7	DIO7	W	0h	This bit is used to toggle DIO7 output. 0h = No effect 1h = Toggle output
6	DIO6	W	0h	This bit is used to toggle DIO6 output. 0h = No effect 1h = Toggle output
5	DIO5	W	0h	This bit is used to toggle DIO5 output. 0h = No effect 1h = Toggle output
4	DIO4	W	0h	This bit is used to toggle DIO4 output. 0h = No effect 1h = Toggle output
3	DIO3	W	0h	This bit is used to toggle DIO3 output. 0h = No effect 1h = Toggle output

Table 10-44. DOUTTGL31_0 Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	DIO2	W	0h	This bit is used to toggle DIO2 output. 0h = No effect 1h = Toggle output
1	DIO1	W	0h	This bit is used to toggle DIO1 output. 0h = No effect 1h = Toggle output
0	DIO0	W	0h	This bit is used to toggle DIO0 output. 0h = No effect 1h = Toggle output

10.3.42 DOE31_0 (Offset = 12C0h) [Reset = 00000000h]

DOE31_0 is shown in Figure 10-45 and described in Table 10-45.

Return to the Summary Table.

This register is used to enable the data outputs for DIO31 to DIO0.

Figure 10-45. 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 10-45. DOE31_0 Field Descriptions

Bit	Field	Type	Reset	Description
31	DIO31	R/W	0h	Enables data output for DIO31. 0h = Output disabled 1h = Output enabled
30	DIO30	R/W	0h	Enables data output for DIO30. 0h = Output disabled 1h = Output enabled
29	DIO29	R/W	0h	Enables data output for DIO29. 0h = Output disabled 1h = Output enabled
28	DIO28	R/W	0h	Enables data output for DIO28. 0h = Output disabled 1h = Output enabled
27	DIO27	R/W	0h	Enables data output for DIO27. 0h = Output disabled 1h = Output enabled
26	DIO26	R/W	0h	Enables data output for DIO26. 0h = Output disabled 1h = Output enabled
25	DIO25	R/W	0h	Enables data output for DIO25. 0h = Output disabled 1h = Output enabled
24	DIO24	R/W	0h	Enables data output for DIO24. 0h = Output disabled 1h = Output enabled
23	DIO23	R/W	0h	Enables data output for DIO23. 0h = Output disabled 1h = Output enabled
22	DIO22	R/W	0h	Enables data output for DIO22. 0h = Output disabled 1h = Output enabled

Table 10-45. DOE31_0 Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	DIO21	R/W	0h	Enables data output for DIO21. 0h = Output disabled 1h = Output enabled
20	DIO20	R/W	0h	Enables data output for DIO20. 0h = Output disabled 1h = Output enabled
19	DIO19	R/W	0h	Enables data output for DIO19. 0h = Output disabled 1h = Output enabled
18	DIO18	R/W	0h	Enables data output for DIO18. 0h = Output disabled 1h = Output enabled
17	DIO17	R/W	0h	Enables data output for DIO17. 0h = Output disabled 1h = Output enabled
16	DIO16	R/W	0h	Enables data output for DIO16. 0h = Output disabled 1h = Output enabled
15	DIO15	R/W	0h	Enables data output for DIO15. 0h = Output disabled 1h = Output enabled
14	DIO14	R/W	0h	Enables data output for DIO14. 0h = Output disabled 1h = Output enabled
13	DIO13	R/W	0h	Enables data output for DIO13. 0h = Output disabled 1h = Output enabled
12	DIO12	R/W	0h	Enables data output for DIO12. 0h = Output disabled 1h = Output enabled
11	DIO11	R/W	0h	Enables data output for DIO11. 0h = Output disabled 1h = Output enabled
10	DIO10	R/W	0h	Enables data output for DIO10. 0h = Output disabled 1h = Output enabled
9	DIO9	R/W	0h	Enables data output for DIO9. 0h = Output disabled 1h = Output enabled
8	DIO8	R/W	0h	Enables data output for DIO8. 0h = Output disabled 1h = Output enabled
7	DIO7	R/W	0h	Enables data output for DIO7. 0h = Output disabled 1h = Output enabled
6	DIO6	R/W	0h	Enables data output for DIO6. 0h = Output disabled 1h = Output enabled
5	DIO5	R/W	0h	Enables data output for DIO5. 0h = Output disabled 1h = Output enabled
4	DIO4	R/W	0h	Enables data output for DIO4. 0h = Output disabled 1h = Output enabled
3	DIO3	R/W	0h	Enables data output for DIO3. 0h = Output disabled 1h = Output enabled

Table 10-45. DOE31_0 Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	DIO2	R/W	0h	Enables data output for DIO2. 0h = Output disabled 1h = Output enabled
1	DIO1	R/W	0h	Enables data output for DIO1. 0h = Output disabled 1h = Output enabled
0	DIO0	R/W	0h	Enables data output for DIO0. 0h = Output disabled 1h = Output enabled

10.3.43 DOESET31_0 (Offset = 12D0h) [Reset = 00000000h]

DOESET31_0 is shown in Figure 10-46 and described in Table 10-46.

Return to the Summary Table.

Writing 1 to a bit position in this register sets the corresponding bit in the DOE31_0 register.

Figure 10-46. 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 10-46. 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. Writing 0 has no effect. 0h = No effect 1h = Sets DIO31 in DOE31_0
30	DIO30	W	0h	Writing 1 to this bit sets the DIO30 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO30 in DOE31_0
29	DIO29	W	0h	Writing 1 to this bit sets the DIO29 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO29 in DOE31_0
28	DIO28	W	0h	Writing 1 to this bit sets the DIO28 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO28 in DOE31_0
27	DIO27	W	0h	Writing 1 to this bit sets the DIO27 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO27 in DOE31_0
26	DIO26	W	0h	Writing 1 to this bit sets the DIO26 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO26 in DOE31_0
25	DIO25	W	0h	Writing 1 to this bit sets the DIO25 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO25 in DOE31_0
24	DIO24	W	0h	Writing 1 to this bit sets the DIO24 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO24 in DOE31_0

Table 10-46. 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. Writing 0 has no effect. 0h = No effect 1h = Sets DIO23 in DOE31_0
22	DIO22	W	0h	Writing 1 to this bit sets the DIO22 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO22 in DOE31_0
21	DIO21	W	0h	Writing 1 to this bit sets the DIO21 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO21 in DOE31_0
20	DIO20	W	0h	Writing 1 to this bit sets the DIO20 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO20 in DOE31_0
19	DIO19	W	0h	Writing 1 to this bit sets the DIO19 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO19 in DOE31_0
18	DIO18	W	0h	Writing 1 to this bit sets the DIO18 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO18 in DOE31_0
17	DIO17	W	0h	Writing 1 to this bit sets the DIO17 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO17 in DOE31_0
16	DIO16	W	0h	Writing 1 to this bit sets the DIO16 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO16 in DOE31_0
15	DIO15	W	0h	Writing 1 to this bit sets the DIO15 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO15 in DOE31_0
14	DIO14	W	0h	Writing 1 to this bit sets the DIO14 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO14 in DOE31_0
13	DIO13	W	0h	Writing 1 to this bit sets the DIO13 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO13 in DOE31_0
12	DIO12	W	0h	Writing 1 to this bit sets the DIO12 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO12 in DOE31_0
11	DIO11	W	0h	Writing 1 to this bit sets the DIO11 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO11 in DOE31_0
10	DIO10	W	0h	Writing 1 to this bit sets the DIO10 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO10 in DOE31_0

Table 10-46. 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. 0h = No effect 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. 0h = No effect 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. 0h = No effect 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. 0h = No effect 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. 0h = No effect 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. 0h = No effect 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. 0h = No effect 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. 0h = No effect 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. 0h = No effect 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. 0h = No effect 1h = Sets DIO0 in DOE31_0

10.3.44 DOECLR31_0 (Offset = 12E0h) [Reset = 00000000h]

DOECLR31_0 is shown in Figure 10-47 and described in Table 10-47.

Return to the Summary Table.

Writing 1 to a bit position in this register clears the corresponding bit in the DOE31_0 register.

Figure 10-47. 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 10-47. 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. Writing 0 has no effect. 0h = No effect 1h = Clears DIO31 in DOE31_0
30	DIO30	W	0h	Writing 1 to this bit clears the DIO30 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO30 in DOE31_0
29	DIO29	W	0h	Writing 1 to this bit clears the DIO29 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO29 in DOE31_0
28	DIO28	W	0h	Writing 1 to this bit clears the DIO28 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO28 in DOE31_0
27	DIO27	W	0h	Writing 1 to this bit clears the DIO27 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO27 in DOE31_0
26	DIO26	W	0h	Writing 1 to this bit clears the DIO26 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO26 in DOE31_0
25	DIO25	W	0h	Writing 1 to this bit clears the DIO25 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO25 in DOE31_0
24	DIO24	W	0h	Writing 1 to this bit clears the DIO24 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO24 in DOE31_0

Table 10-47. 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. Writing 0 has no effect. 0h = No effect 1h = Clears DIO23 in DOE31_0
22	DIO22	W	0h	Writing 1 to this bit clears the DIO22 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO22 in DOE31_0
21	DIO21	W	0h	Writing 1 to this bit clears the DIO21 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO21 in DOE31_0
20	DIO20	W	0h	Writing 1 to this bit clears the DIO20 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO20 in DOE31_0
19	DIO19	W	0h	Writing 1 to this bit clears the DIO19 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO19 in DOE31_0
18	DIO18	W	0h	Writing 1 to this bit clears the DIO18 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO18 in DOE31_0
17	DIO17	W	0h	Writing 1 to this bit clears the DIO17 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO17 in DOE31_0
16	DIO16	W	0h	Writing 1 to this bit clears the DIO16 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO16 in DOE31_0
15	DIO15	W	0h	Writing 1 to this bit clears the DIO15 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO15 in DOE31_0
14	DIO14	W	0h	Writing 1 to this bit clears the DIO14 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO14 in DOE31_0
13	DIO13	W	0h	Writing 1 to this bit clears the DIO13 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO13 in DOE31_0
12	DIO12	W	0h	Writing 1 to this bit clears the DIO12 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO12 in DOE31_0
11	DIO11	W	0h	Writing 1 to this bit clears the DIO11 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO11 in DOE31_0
10	DIO10	W	0h	Writing 1 to this bit clears the DIO10 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO10 in DOE31_0

Table 10-47. 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. Writing 0 has no effect. 0h = No effect 1h = Clears DIO9 in DOE31_0
8	DIO8	W	0h	Writing 1 to this bit clears the DIO8 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO8 in DOE31_0
7	DIO7	W	0h	Writing 1 to this bit clears the DIO7 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO7 in DOE31_0
6	DIO6	W	0h	Writing 1 to this bit clears the DIO6 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO6 in DOE31_0
5	DIO5	W	0h	Writing 1 to this bit clears the DIO5 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO5 in DOE31_0
4	DIO4	W	0h	Writing 1 to this bit clears the DIO4 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO4 in DOE31_0
3	DIO3	W	0h	Writing 1 to this bit clears the DIO3 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO3 in DOE31_0
2	DIO2	W	0h	Writing 1 to this bit clears the DIO2 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO2 in DOE31_0
1	DIO1	W	0h	Writing 1 to this bit clears the DIO1 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO1 in DOE31_0
0	DIO0	W	0h	Writing 1 to this bit clears the DIO0 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO0 in DOE31_0

10.3.45 DIN3_0 (Offset = 1300h) [Reset = 00000000h]

DIN3_0 is shown in Figure 10-48 and described in Table 10-48.

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 10-48. 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 10-48. 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. 0h = Input value is 0 1h = Input value is 1
23-17	RESERVED	R	0h	
16	DIO2	R	0h	This bit reads the data input value of DIO2. 0h = Input value is 0 1h = Input value is 1
15-9	RESERVED	R	0h	
8	DIO1	R	0h	This bit reads the data input value of DIO1. 0h = Input value is 0 1h = Input value is 1
7-1	RESERVED	R	0h	
0	DIO0	R	0h	This bit reads the data input value of DIO0. 0h = Input value is 0 1h = Input value is 1

10.3.46 DIN7_4 (Offset = 1304h) [Reset = 00000000h]

DIN7_4 is shown in Figure 10-49 and described in Table 10-49.

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 10-49. 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 10-49. 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. 0h = Input value is 0 1h = Input value is 1
23-17	RESERVED	R	0h	
16	DIO6	R	0h	This bit reads the data input value of DIO6. 0h = Input value is 0 1h = Input value is 1
15-9	RESERVED	R	0h	
8	DIO5	R	0h	This bit reads the data input value of DIO5. 0h = Input value is 0 1h = Input value is 1
7-1	RESERVED	R	0h	
0	DIO4	R	0h	This bit reads the data input value of DIO4. 0h = Input value is 0 1h = Input value is 1

10.3.47 DIN11_8 (Offset = 1308h) [Reset = 00000000h]

DIN11_8 is shown in Figure 10-50 and described in Table 10-50.

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 10-50. 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 10-50. 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. 0h = Input value is 0 1h = Input value is 1
23-17	RESERVED	R	0h	
16	DIO10	R	0h	This bit reads the data input value of DIO10. 0h = Input value is 0 1h = Input value is 1
15-9	RESERVED	R	0h	
8	DIO9	R	0h	This bit reads the data input value of DIO9. 0h = Input value is 0 1h = Input value is 1
7-1	RESERVED	R	0h	
0	DIO8	R	0h	This bit reads the data input value of DIO8. 0h = Input value is 0 1h = Input value is 1

10.3.48 DIN15_12 (Offset = 130Ch) [Reset = 00000000h]

DIN15_12 is shown in Figure 10-51 and described in Table 10-51.

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 10-51. 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 10-51. 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. 0h = Input value is 0 1h = Input value is 1
23-17	RESERVED	R	0h	
16	DIO14	R	0h	This bit reads the data input value of DIO14. 0h = Input value is 0 1h = Input value is 1
15-9	RESERVED	R	0h	
8	DIO13	R	0h	This bit reads the data input value of DIO13. 0h = Input value is 0 1h = Input value is 1
7-1	RESERVED	R	0h	
0	DIO12	R	0h	This bit reads the data input value of DIO12. 0h = Input value is 0 1h = Input value is 1

10.3.49 DIN19_16 (Offset = 1310h) [Reset = 00000000h]

DIN19_16 is shown in Figure 10-52 and described in Table 10-52.

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 10-52. 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 10-52. 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. 0h = Input value is 0 1h = Input value is 1
23-17	RESERVED	R	0h	
16	DIO18	R	0h	This bit reads the data input value of DIO18. 0h = Input value is 0 1h = Input value is 1
15-9	RESERVED	R	0h	
8	DIO17	R	0h	This bit reads the data input value of DIO17. 0h = Input value is 0 1h = Input value is 1
7-1	RESERVED	R	0h	
0	DIO16	R	0h	This bit reads the data input value of DIO16. 0h = Input value is 0 1h = Input value is 1

10.3.50 DIN23_20 (Offset = 1314h) [Reset = 00000000h]

DIN23_20 is shown in Figure 10-53 and described in Table 10-53.

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 10-53. 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 10-53. 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. 0h = Input value is 0 1h = Input value is 1
23-17	RESERVED	R	0h	
16	DIO22	R	0h	This bit reads the data input value of DIO22. 0h = Input value is 0 1h = Input value is 1
15-9	RESERVED	R	0h	
8	DIO21	R	0h	This bit reads the data input value of DIO21. 0h = Input value is 0 1h = Input value is 1
7-1	RESERVED	R	0h	
0	DIO20	R	0h	This bit reads the data input value of DIO20. 0h = Input value is 0 1h = Input value is 1

10.3.51 DIN27_24 (Offset = 1318h) [Reset = 00000000h]

DIN27_24 is shown in Figure 10-54 and described in Table 10-54.

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 10-54. 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 10-54. 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. 0h = Input value is 0 1h = Input value is 1
23-17	RESERVED	R	0h	
16	DIO26	R	0h	This bit reads the data input value of DIO26. 0h = Input value is 0 1h = Input value is 1
15-9	RESERVED	R	0h	
8	DIO25	R	0h	This bit reads the data input value of DIO25. 0h = Input value is 0 1h = Input value is 1
7-1	RESERVED	R	0h	
0	DIO24	R	0h	This bit reads the data input value of DIO24. 0h = Input value is 0 1h = Input value is 1

10.3.52 DIN31_28 (Offset = 131Ch) [Reset = 00000000h]

DIN31_28 is shown in Figure 10-55 and described in Table 10-55.

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 10-55. 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 10-55. 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. 0h = Input value is 0 1h = Input value is 1
23-17	RESERVED	R	0h	
16	DIO30	R	0h	This bit reads the data input value of DIO30. 0h = Input value is 0 1h = Input value is 1
15-9	RESERVED	R	0h	
8	DIO29	R	0h	This bit reads the data input value of DIO29. 0h = Input value is 0 1h = Input value is 1
7-1	RESERVED	R	0h	
0	DIO28	R	0h	This bit reads the data input value of DIO28. 0h = Input value is 0 1h = Input value is 1

10.3.53 DIN31_0 (Offset = 1380h) [Reset = 00000000h]

DIN31_0 is shown in Figure 10-56 and described in Table 10-56.

Return to the Summary Table.

Data input value for pins configured as DIO31 to DIO0.

Figure 10-56. 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 10-56. DIN31_0 Field Descriptions

Bit	Field	Type	Reset	Description
31	DIO31	R	0h	This bit reads the data input value of DIO31. 0h = Input value is 0 1h = Input value is 1
30	DIO30	R	0h	This bit reads the data input value of DIO30. 0h = Input value is 0 1h = Input value is 1
29	DIO29	R	0h	This bit reads the data input value of DIO29. 0h = Input value is 0 1h = Input value is 1
28	DIO28	R	0h	This bit reads the data input value of DIO28. 0h = Input value is 0 1h = Input value is 1
27	DIO27	R	0h	This bit reads the data input value of DIO27. 0h = Input value is 0 1h = Input value is 1
26	DIO26	R	0h	This bit reads the data input value of DIO26. 0h = Input value is 0 1h = Input value is 1
25	DIO25	R	0h	This bit reads the data input value of DIO25. 0h = Input value is 0 1h = Input value is 1
24	DIO24	R	0h	This bit reads the data input value of DIO24. 0h = Input value is 0 1h = Input value is 1
23	DIO23	R	0h	This bit reads the data input value of DIO23. 0h = Input value is 0 1h = Input value is 1
22	DIO22	R	0h	This bit reads the data input value of DIO22. 0h = Input value is 0 1h = Input value is 1

Table 10-56. DIN31_0 Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	DIO21	R	0h	This bit reads the data input value of DIO21. 0h = Input value is 0 1h = Input value is 1
20	DIO20	R	0h	This bit reads the data input value of DIO20. 0h = Input value is 0 1h = Input value is 1
19	DIO19	R	0h	This bit reads the data input value of DIO19. 0h = Input value is 0 1h = Input value is 1
18	DIO18	R	0h	This bit reads the data input value of DIO18. 0h = Input value is 0 1h = Input value is 1
17	DIO17	R	0h	This bit reads the data input value of DIO17. 0h = Input value is 0 1h = Input value is 1
16	DIO16	R	0h	This bit reads the data input value of DIO16. 0h = Input value is 0 1h = Input value is 1
15	DIO15	R	0h	This bit reads the data input value of DIO15. 0h = Input value is 0 1h = Input value is 1
14	DIO14	R	0h	This bit reads the data input value of DIO14. 0h = Input value is 0 1h = Input value is 1
13	DIO13	R	0h	This bit reads the data input value of DIO13. 0h = Input value is 0 1h = Input value is 1
12	DIO12	R	0h	This bit reads the data input value of DIO12. 0h = Input value is 0 1h = Input value is 1
11	DIO11	R	0h	This bit reads the data input value of DIO11. 0h = Input value is 0 1h = Input value is 1
10	DIO10	R	0h	This bit reads the data input value of DIO10. 0h = Input value is 0 1h = Input value is 1
9	DIO9	R	0h	This bit reads the data input value of DIO9. 0h = Input value is 0 1h = Input value is 1
8	DIO8	R	0h	This bit reads the data input value of DIO8. 0h = Input value is 0 1h = Input value is 1
7	DIO7	R	0h	This bit reads the data input value of DIO7. 0h = Input value is 0 1h = Input value is 1
6	DIO6	R	0h	This bit reads the data input value of DIO6. 0h = Input value is 0 1h = Input value is 1
5	DIO5	R	0h	This bit reads the data input value of DIO5. 0h = Input value is 0 1h = Input value is 1
4	DIO4	R	0h	This bit reads the data input value of DIO4. 0h = Input value is 0 1h = Input value is 1
3	DIO3	R	0h	This bit reads the data input value of DIO3. 0h = Input value is 0 1h = Input value is 1

Table 10-56. DIN31_0 Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	DIO2	R	0h	This bit reads the data input value of DIO2. 0h = Input value is 0 1h = Input value is 1
1	DIO1	R	0h	This bit reads the data input value of DIO1. 0h = Input value is 0 1h = Input value is 1
0	DIO0	R	0h	This bit reads the data input value of DIO0. 0h = Input value is 0 1h = Input value is 1

10.3.54 POLARITY15_0 (Offset = 1390h) [Reset = 00000000h]

POLARITY15_0 is shown in Figure 10-57 and described in Table 10-57.

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 10-57. 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 10-57. POLARITY15_0 Field Descriptions

Bit	Field	Type	Reset	Description
31-30	DIO15	R/W	0h	Enables and configures edge detection polarity for DIO15. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
29-28	DIO14	R/W	0h	Enables and configures edge detection polarity for DIO14. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
27-26	DIO13	R/W	0h	Enables and configures edge detection polarity for DIO13. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
25-24	DIO12	R/W	0h	Enables and configures edge detection polarity for DIO12. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
23-22	DIO11	R/W	0h	Enables and configures edge detection polarity for DIO11. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
21-20	DIO10	R/W	0h	Enables and configures edge detection polarity for DIO10. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
19-18	DIO9	R/W	0h	Enables and configures edge detection polarity for DIO9. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
17-16	DIO8	R/W	0h	Enables and configures edge detection polarity for DIO8. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event

Table 10-57. POLARITY15_0 Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-14	DIO7	R/W	0h	Enables and configures edge detection polarity for DIO7. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
13-12	DIO6	R/W	0h	Enables and configures edge detection polarity for DIO6. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
11-10	DIO5	R/W	0h	Enables and configures edge detection polarity for DIO5. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
9-8	DIO4	R/W	0h	Enables and configures edge detection polarity for DIO4. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
7-6	DIO3	R/W	0h	Enables and configures edge detection polarity for DIO3. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
5-4	DIO2	R/W	0h	Enables and configures edge detection polarity for DIO2. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
3-2	DIO1	R/W	0h	Enables and configures edge detection polarity for DIO1. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
1-0	DIO0	R/W	0h	Enables and configures edge detection polarity for DIO0. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event

10.3.55 POLARITY31_16 (Offset = 13A0h) [Reset = 00000000h]

POLARITY31_16 is shown in Figure 10-58 and described in Table 10-58.

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 10-58. 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 10-58. POLARITY31_16 Field Descriptions

Bit	Field	Type	Reset	Description
31-30	DIO31	R/W	0h	Enables and configures edge detection polarity for DIO31. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
29-28	DIO30	R/W	0h	Enables and configures edge detection polarity for DIO30. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
27-26	DIO29	R/W	0h	Enables and configures edge detection polarity for DIO29. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
25-24	DIO28	R/W	0h	Enables and configures edge detection polarity for DIO28. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
23-22	DIO27	R/W	0h	Enables and configures edge detection polarity for DIO27. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
21-20	DIO26	R/W	0h	Enables and configures edge detection polarity for DIO26. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
19-18	DIO25	R/W	0h	Enables and configures edge detection polarity for DIO25. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
17-16	DIO24	R/W	0h	Enables and configures edge detection polarity for DIO24. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event

Table 10-58. POLARITY31_16 Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-14	DIO23	R/W	0h	Enables and configures edge detection polarity for DIO23. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
13-12	DIO22	R/W	0h	Enables and configures edge detection polarity for DIO22. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
11-10	DIO21	R/W	0h	Enables and configures edge detection polarity for DIO21. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
9-8	DIO20	R/W	0h	Enables and configures edge detection polarity for DIO20. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
7-6	DIO19	R/W	0h	Enables and configures edge detection polarity for DIO19. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
5-4	DIO18	R/W	0h	Enables and configures edge detection polarity for DIO18. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
3-2	DIO17	R/W	0h	Enables and configures edge detection polarity for DIO17. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
1-0	DIO16	R/W	0h	Enables and configures edge detection polarity for DIO16. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event

10.3.56 CTL (Offset = 1400h) [Reset = 00000000h]

CTL is shown in Figure 10-59 and described in Table 10-59.

Return to the Summary Table.

GPIO Control Register

Figure 10-59. 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 10-59. 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 0h = The global control of fastwake is not enabled, per bit fast wake feature depends on FASTWAKE.DIN 1h = The global control of fastwake is enabled

10.3.57 FASTWAKE (Offset = 1404h) [Reset = 00000000h]

FASTWAKE is shown in Figure 10-60 and described in Table 10-60.

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 10-60. 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 10-60. FASTWAKE Field Descriptions

Bit	Field	Type	Reset	Description
31	DIN31	R/W	0h	Enable fastwake feature for DIN31 0h = fastwake feature is disabled 1h = fastwake feature is enabled
30	DIN30	R/W	0h	Enable fastwake feature for DIN30 0h = fastwake feature is disabled 1h = fastwake feature is enabled
29	DIN29	R/W	0h	Enable fastwake feature for DIN29 0h = fastwake feature is disabled 1h = fastwake feature is enabled
28	DIN28	R/W	0h	Enable fastwake feature for DIN29 0h = fastwake feature is disabled 1h = fastwake feature is enabled
27	DIN27	R/W	0h	Enable fastwake feature for DIN27 0h = fastwake feature is disabled 1h = fastwake feature is enabled
26	DIN26	R/W	0h	Enable fastwake feature for DIN26 0h = fastwake feature is disabled 1h = fastwake feature is enabled
25	DIN25	R/W	0h	Enable fastwake feature for DIN25 0h = fastwake feature is disabled 1h = fastwake feature is enabled
24	DIN24	R/W	0h	Enable fastwake feature for DIN24 0h = fastwake feature is disabled 1h = fastwake feature is enabled
23	DIN23	R/W	0h	Enable fastwake feature for DIN23 0h = fastwake feature is disabled 1h = fastwake feature is enabled
22	DIN22	R/W	0h	Enable fastwake feature for DIN22 0h = fastwake feature is disabled 1h = fastwake feature is enabled

Table 10-60. FASTWAKE Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	DIN21	R/W	0h	Enable fastwake feature for DIN21 0h = fastwake feature is disabled 1h = fastwake feature is enabled
20	DIN20	R/W	0h	Enable fastwake feature for DIN20 0h = fastwake feature is disabled 1h = fastwake feature is enabled
19	DIN19	R/W	0h	Enable fastwake feature for DIN19 0h = fastwake feature is disabled 1h = fastwake feature is enabled
18	DIN18	R/W	0h	Enable fastwake feature for DIN18 0h = fastwake feature is disabled 1h = fastwake feature is enabled
17	DIN17	R/W	0h	Enable fastwake feature for DIN17 0h = fastwake feature is disabled 1h = fastwake feature is enabled
16	DIN16	R/W	0h	Enable fastwake feature for DIN16 0h = fastwake feature is disabled 1h = fastwake feature is enabled
15	DIN15	R/W	0h	Enable fastwake feature for DIN15 0h = fastwake feature is disabled 1h = fastwake feature is enabled
14	DIN14	R/W	0h	Enable fastwake feature for DIN14 0h = fastwake feature is disabled 1h = fastwake feature is enabled
13	DIN13	R/W	0h	Enable fastwake feature for DIN13 0h = fastwake feature is disabled 1h = fastwake feature is enabled
12	DIN12	R/W	0h	Enable fastwake feature for DIN12 0h = fastwake feature is disabled 1h = fastwake feature is enabled
11	DIN11	R/W	0h	Enable fastwake feature for DIN11 0h = fastwake feature is disabled 1h = fastwake feature is enabled
10	DIN10	R/W	0h	Enable fastwake feature for DIN10 0h = fastwake feature is disabled 1h = fastwake feature is enabled
9	DIN9	R/W	0h	Enable fastwake feature for DIN9 0h = fastwake feature is disabled 1h = fastwake feature is enabled
8	DIN8	R/W	0h	Enable fastwake feature for DIN8 0h = fastwake feature is disabled 1h = fastwake feature is enabled
7	DIN7	R/W	0h	Enable fastwake feature for DIN7 0h = fastwake feature is disabled 1h = fastwake feature is enabled
6	DIN6	R/W	0h	Enable fastwake feature for DIN6 0h = fastwake feature is disabled 1h = fastwake feature is enabled
5	DIN5	R/W	0h	Enable fastwake feature for DIN5 0h = fastwake feature is disabled 1h = fastwake feature is enabled
4	DIN4	R/W	0h	Enable fastwake feature for DIN4 0h = fastwake feature is disabled 1h = fastwake feature is enabled
3	DIN3	R/W	0h	Enable fastwake feature for DIN3 0h = fastwake feature is disabled 1h = fastwake feature is enabled

Table 10-60. FASTWAKE Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	DIN2	R/W	0h	Enable fastwake feature for DIN2 0h = fastwake feature is disabled 1h = fastwake feature is enabled
1	DIN1	R/W	0h	Enable fastwake feature for DIN1 0h = fastwake feature is disabled 1h = fastwake feature is enabled
0	DIN0	R/W	0h	Enable fastwake feature for DIN0 0h = fastwake feature is disabled 1h = fastwake feature is enabled

10.3.58 SUB0CFG (Offset = 1500h) [Reset = 00000000h]

SUB0CFG is shown in Figure 10-61 and described in Table 10-61.

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 10-61. 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 10-61. 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 0h = specific bit targeted by the subscriber action is bit0 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. 0h = Selected DIO pins are set 1h = Selected DIO pins are cleared 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. 0h = Subscriber 0 event is disabled 1h = Subscriber 0 event is enabled

10.3.59 FILTEREN15_0 (Offset = 1508h) [Reset = 00000000h]

FILTEREN15_0 is shown in Figure 10-62 and described in Table 10-62.

Return to the Summary Table.

Programmable counter length of digital glitch filter for DIN0-DIN15

Figure 10-62. 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 10-62. FILTEREN15_0 Field Descriptions

Bit	Field	Type	Reset	Description
31-30	DIN15	R/W	0h	Programmable counter length of digital glitch filter for DIN15 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
29-28	DIN14	R/W	0h	Programmable counter length of digital glitch filter for DIN14 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
27-26	DIN13	R/W	0h	Programmable counter length of digital glitch filter for DIN13 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
25-24	DIN12	R/W	0h	Programmable counter length of digital glitch filter for DIN12 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
23-22	DIN11	R/W	0h	Programmable counter length of digital glitch filter for DIN11 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
21-20	DIN10	R/W	0h	Programmable counter length of digital glitch filter for DIN10 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
19-18	DIN9	R/W	0h	Programmable counter length of digital glitch filter for DIN9 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
17-16	DIN8	R/W	0h	Programmable counter length of digital glitch filter for DIN8 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample

Table 10-62. 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 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
13-12	DIN6	R/W	0h	Programmable counter length of digital glitch filter for DIN6 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
11-10	DIN5	R/W	0h	Programmable counter length of digital glitch filter for DIN5 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
9-8	DIN4	R/W	0h	Programmable counter length of digital glitch filter for DIN4 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
7-6	DIN3	R/W	0h	Programmable counter length of digital glitch filter for DIN3 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
5-4	DIN2	R/W	0h	Programmable counter length of digital glitch filter for DIN2 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
3-2	DIN1	R/W	0h	Programmable counter length of digital glitch filter for DIN1 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
1-0	DIN0	R/W	0h	Programmable counter length of digital glitch filter for DIN0 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample

10.3.60 FILTEREN31_16 (Offset = 150Ch) [Reset = 00000000h]

FILTEREN31_16 is shown in Figure 10-63 and described in Table 10-63.

Return to the Summary Table.

Programmable counter length of digital glitch filter for DIN16-DIN31

Figure 10-63. 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 10-63. FILTEREN31_16 Field Descriptions

Bit	Field	Type	Reset	Description
31-30	DIN31	R/W	0h	Programmable counter length of digital glitch filter for DIN31 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
29-28	DIN30	R/W	0h	Programmable counter length of digital glitch filter for DIN30 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
27-26	DIN29	R/W	0h	Programmable counter length of digital glitch filter for DIN29 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
25-24	DIN28	R/W	0h	Programmable counter length of digital glitch filter for DIN28 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
23-22	DIN27	R/W	0h	Programmable counter length of digital glitch filter for DIN27 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
21-20	DIN26	R/W	0h	Programmable counter length of digital glitch filter for DIN26 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
19-18	DIN25	R/W	0h	Programmable counter length of digital glitch filter for DIN25 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
17-16	DIN24	R/W	0h	Programmable counter length of digital glitch filter for DIN24 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample

Table 10-63. 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 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
13-12	DIN22	R/W	0h	Programmable counter length of digital glitch filter for DIN22 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
11-10	DIN21	R/W	0h	Programmable counter length of digital glitch filter for DIN21 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
9-8	DIN20	R/W	0h	Programmable counter length of digital glitch filter for DIN20 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
7-6	DIN19	R/W	0h	Programmable counter length of digital glitch filter for DIN19 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
5-4	DIN18	R/W	0h	Programmable counter length of digital glitch filter for DIN18 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
3-2	DIN17	R/W	0h	Programmable counter length of digital glitch filter for DIN17 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
1-0	DIN16	R/W	0h	Programmable counter length of digital glitch filter for DIN16 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample

10.3.61 DMAMASK (Offset = 1510h) [Reset = 00000000h]

DMAMASK is shown in Figure 10-64 and described in Table 10-64.

Return to the Summary Table.

DMA MASK which indicates which bit lanes the DMA is allowed to modify.

Figure 10-64. 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 10-64. DMAMASK Field Descriptions

Bit	Field	Type	Reset	Description
31	DOUT31	R/W	0h	DMA is allowed to modify DOUT31 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
30	DOUT30	R/W	0h	DMA is allowed to modify DOUT30 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
29	DOUT29	R/W	0h	DMA is allowed to modify DOUT29 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
28	DOUT28	R/W	0h	DMA is allowed to modify DOUT28 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
27	DOUT27	R/W	0h	DMA is allowed to modify DOUT27 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
26	DOUT26	R/W	0h	DMA is allowed to modify DOUT26 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
25	DOUT25	R/W	0h	DMA is allowed to modify DOUT25 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
24	DOUT24	R/W	0h	DMA is allowed to modify DOUT24 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
23	DOUT23	R/W	0h	DMA is allowed to modify DOUT23 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
22	DOUT22	R/W	0h	DMA is allowed to modify DOUT22 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane

Table 10-64. DMAMASK Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	DOUT21	R/W	0h	DMA is allowed to modify DOUT21 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
20	DOUT20	R/W	0h	DMA is allowed to modify DOUT20 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
19	DOUT19	R/W	0h	DMA is allowed to modify DOUT19 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
18	DOUT18	R/W	0h	DMA is allowed to modify DOUT18 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
17	DOUT17	R/W	0h	DMA is allowed to modify DOUT17 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
16	DOUT16	R/W	0h	DMA is allowed to modify DOUT16 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
15	DOUT15	R/W	0h	DMA is allowed to modify DOUT15 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
14	DOUT14	R/W	0h	DMA is allowed to modify DOUT14 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
13	DOUT13	R/W	0h	DMA is allowed to modify DOUT13 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
12	DOUT12	R/W	0h	DMA is allowed to modify DOUT12 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
11	DOUT11	R/W	0h	DMA is allowed to modify DOUT11 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
10	DOUT10	R/W	0h	DMA is allowed to modify DOUT10 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
9	DOUT9	R/W	0h	DMA is allowed to modify DOUT9 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
8	DOUT8	R/W	0h	DMA is allowed to modify DOUT8 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
7	DOUT7	R/W	0h	DMA is allowed to modify DOUT7 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
6	DOUT6	R/W	0h	DMA is allowed to modify DOUT6 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
5	DOUT5	R/W	0h	DMA is allowed to modify DOUT5 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
4	DOUT4	R/W	0h	DMA is allowed to modify DOUT4 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
3	DOUT3	R/W	0h	DMA is allowed to modify DOUT3 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane

Table 10-64. DMAMASK Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	DOUT2	R/W	0h	DMA is allowed to modify DOUT2 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
1	DOUT1	R/W	0h	DMA is allowed to modify DOUT1 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
0	DOUT0	R/W	0h	DMA is allowed to modify DOUT0 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane

10.3.62 SUB1CFG (Offset = 1520h) [Reset = 00000000h]

SUB1CFG is shown in Figure 10-65 and described in Table 10-65.

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 10-65. 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 10-65. 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 0h = specific bit targeted by the subscriber action is bit16 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. 0h = Selected DIO pins are set 1h = Selected DIO pins are cleared 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. 0h = Subscriber 1 event is disabled 1h = Subscriber 1 event is enabled

Chapter 11

Global Security Controller

The Global Security Controller (GSC) peripheral is a system module which is used to configure the security firewalls for secure and non-secure context of the application. After configuration, it detects out of context transaction to the flash, SRAM and peripherals, and generates NMI for the secure CPU context.

11.1 GSC Introduction 548
11.2 GSC Operation 548
11.3 GSC Registers 557

11.1 GSC Introduction

The Global Security Controller (GSC) configures the secure and privilege attribitue for resources on the device. The GSC consists of multiple blocks that ensure the device's access properties are seen same by all bus initiators. In essence, it provides firewall access based on secure and priviliege property of the application code and initiators.

11.1.1 GSC Features

- Configures secure and non-secure attribute for memory and peripherals
- Configures privilege and unprivileged attribute for memory and peripherals
- Configures secure and privilege attribute for non-secure and non-privilege aware initators
- Write and erase protection for flash between secure and non-secure context
- Direct monitoring of initiator access to memory and peripherals to generate NMI in case of access violation
- Logs violation information in the EAM for the initiator and the destination address
- Hide protection control for flash in 2 kB sector size granularity

11.2 GSC Operation

The GSC comprises of three important sub-blocks as shown in Figure 11-1. The Peripheral Protection Controller (PPC), the SRAM Protection Controller (SPC) and the Flash Protection Controller (FPC).

11.2.1 Functional Block Diagram

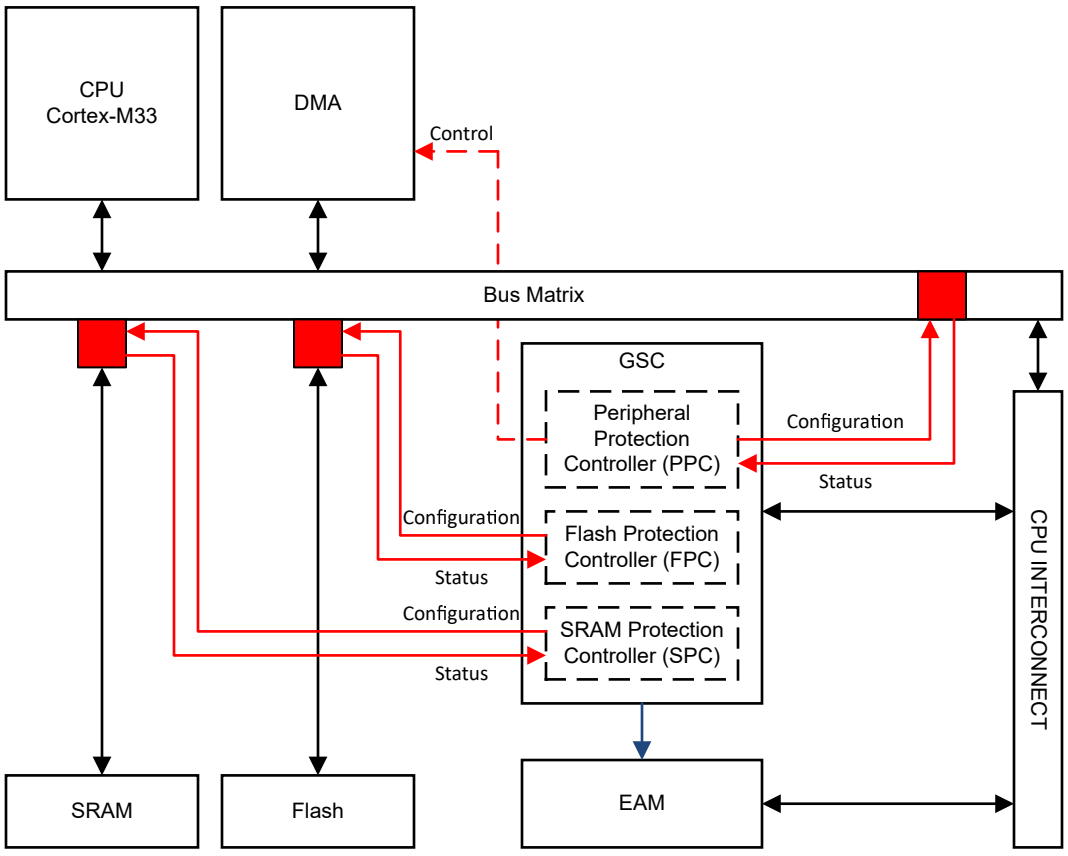


Figure 11-1. GSC Functional Block Diagram

11.2.2 Peripheral Protection Controller

The PPC setups the firewall for peripheral access. There are two sets of peripheral firewall register that exist for every peripheral instance. The register PPC_SECATTRIB for a peripheral configures the firewall for secure

or non-secure access to a peripheral instance. The register PPC_PRIVATTRIB for a peripheral configures the firewall for privilege or unprivliege access.

When a peripheral is configured as a secure peripheral then only an initiator which is operating in secure context can access the peripheral configuration, status or data registers. Any initiator access not in the secure context shall generate a non-maskable interrupt (NMI) which is logged into the EAM module.

When a peripheral is configured as a privilege peripheral then only an initiator which is operating in privilege context can access the peripheral configuration, status or data registers. Any initiator access not in the privilege context shall generate a non-maskable interrupt (NMI) which is logged into the EAM module.

In addition to generating an NMI, any write access shall be ignored and a read access shall return all zeroes to the corresponding initiator. This prevents a non-secure or unpriviliged code or initiator from accessing or modifying data in the peripheral space at run time.

Note

All peripheral instances have a PPC_SECATTRIB but not PPC_PRIVATTRIB register. Please refer to the GSC registers.

Some peripheral firewalls are configured with secure and privilege attribute enabled. This allows the application code from leveraging a default secure setting of the device to implement secure boot loaders or authentication code.

11.2.2.1 DMA controller security

The DMA is uniquely handled within the GSC framework as it is both an initiator and a target. When the DMA is configured for secure access then both the initiator and the target ports are set to secure state.

The power on reset value for DMA instance 0 is always secure and privilege. Only the CPU in secure mode can configure the DMA0 channel registers. Once configured, all write and read access generated by DMA0 initiator ports are also secure. This ensure that after the device is powered on and the CPU begins code execution in secure mode, the DMA0 is in a secure state for moving data between peripherals and memory and does not require any additional configuration.

The power on reset value for DMA instance 1 is always non-secure and unprivileged. This allows the a non-secure application to request data transfer service without any special handling.

Note

The GSC is also secure and privilege at power on reset. A non-secure or unprivileged application cannot directly modify the DMA property. This must be handled within a secure application.

As both DMA may be required for efficient handling of data transfer, the secure core may change both DMA instances to be available to the application. To ensure the seamless reassignment of the DMA instances, it is strongly recommended that the DMA be disabled and reset before changing the secure and privilege attribute.

11.2.3 SRAM Protection Controller

The SRAM Protection Controller (SPC) configures the firewall for the SRAM and monitors the transactions from the initiators. Any violation of the security property set for the SRAM results in a NMI being triggered and the violation logged in the EAM. Just like the PPC sub-block, the SPC provides two sets of SRAM firewall registers, SPC_SECATTRIB[x] and SPC_PRIVATTRIB[x].

To manage the SRAM firewall efficiently, the SRAM is partitioned into 16 kB pages. Each SPC_SECATTRIB[x] and SPC_PRIVATTRIB[x] register controls 4 such pages. Further granular control is provided within each page control field in sets of 8-bits as shown in Figure 11-2.

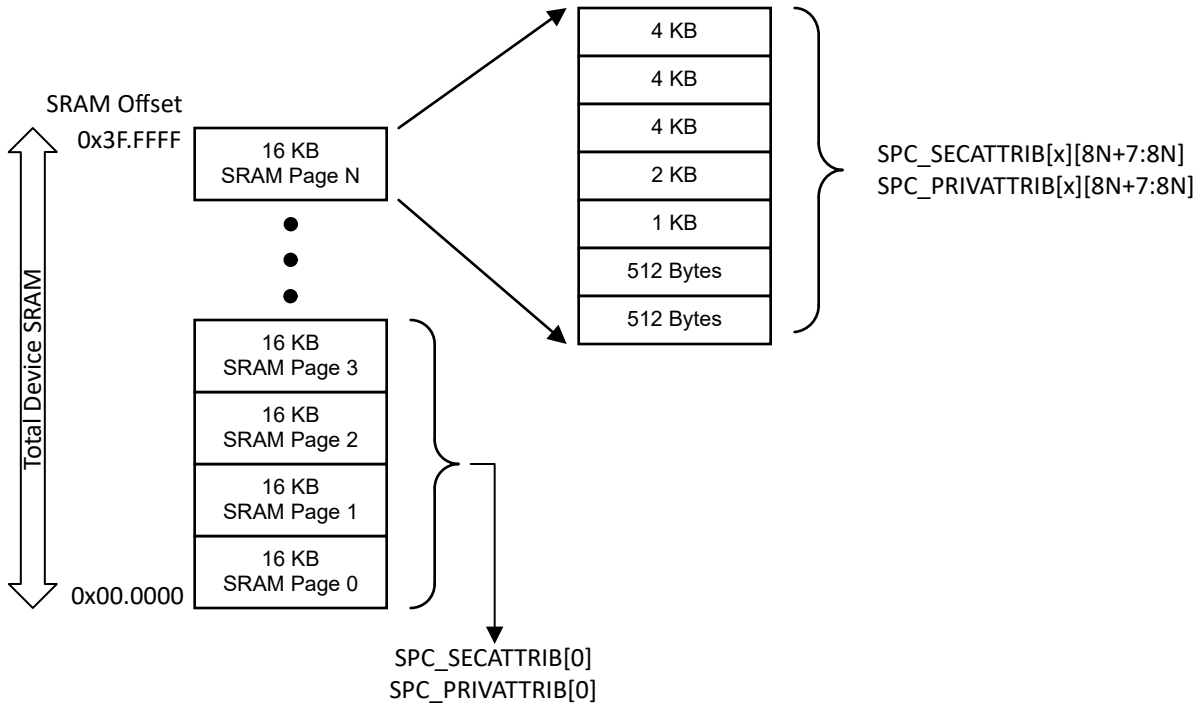


Figure 11-2. SRAM mapping to Page and Chunk

Each page is comprised of seven chunks of variable size. As a result, out of the 8-bits available for a page, only seven are used with the most significant bit being reserved. The chunks in a page are comprised of the following size SRAM

- 3 chunks of 4 kB
- 1 chunk of 2 kB and 1 kB each
- 2 chunks of 512 bytes

11.2.3.1 SRAM Page Use Model

Every 16 kB SRAM page chunks, can be configured independently for secure and privilege attributes to maxmize SRAM usage for different aspects of the application code.

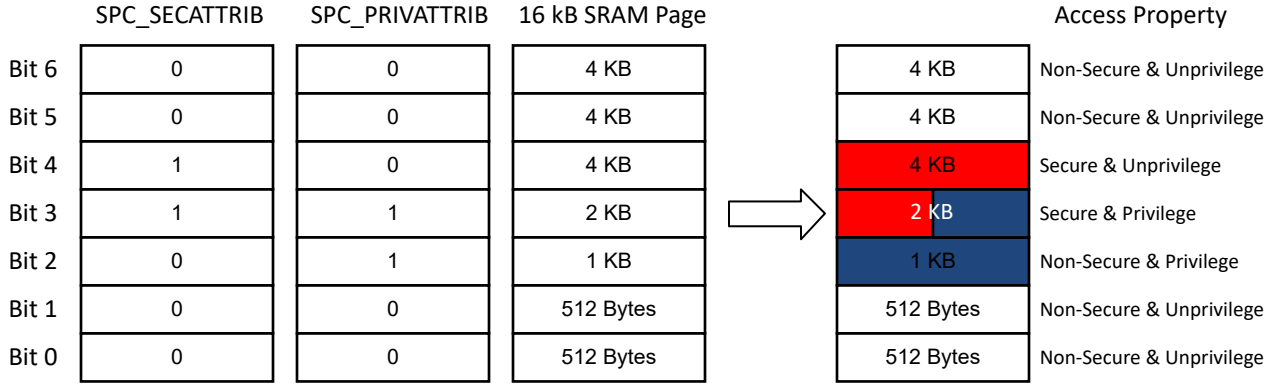


Figure 11-3. SRAM Page Use Model Illustration

As shown in the example in Figure 11-3, one of the 16 kB SRAM page is provisioned such that:

- 4 kB of SRAM can be accessed by a secure and unprivileged initiator
- 2 kB of SRAM can be accessed by a secure and privileged initiator
- 1 kB of SRAM can be accessed by a non-secure and privileged initiator

- 9 kB of SRAM can be accessed by a non-secure and unprivileged initiator

The SPC allows for chunks within a page and for pages within the entire SRAM with secure and privilege property, to be non-continguous. Additionally, by default the SPC is configurable by a secure and privileged initiator, which allows the secure context to dynamically allocate memory property. This is extremely useful, when SRAM has to be allocated for a critical application code from the common memory pool.

Note

Before changing the SRAM chunk memory property, it is the secure application's responsibility to ensure that all initiators are properly configured and any data in SRAM is zero initialized.

11.2.4 Flash Protection Controller

The Flash Protection Controller (FPC) configures the firewall for the program and data flash and monitors the transactions from the initiators. Any violation of the security property set for the flash results in an NMI being triggered and the violation logged in the EAM. Just like the PPC sub-block, the FPC provides two sets of flash firewall registers, FPC_SECATTRIBA[x] and FPC_PRIVATTRIBA[x] for bank 0 and FPC_SECATTRIBB[x] and FPC_PRIVATTRIBB[x] for bank 1. In addition, the FPC also provides flash write and erase protection registers, FPC_WEPROTA[x], FPC_WEPROTB[x] and FPC_WEPROTA_DFLASH for program and data flash respectively.

Note

For security configuration of Flash Programming Interface, refer to Section 5.4.

To manage the flash firewalls efficiently, the FPC uses the sector information for each flash bank. Each flash bank consists of multiple 2 kB flash sectors. An illustration of a 1MB flash is shown in Figure 11-4.

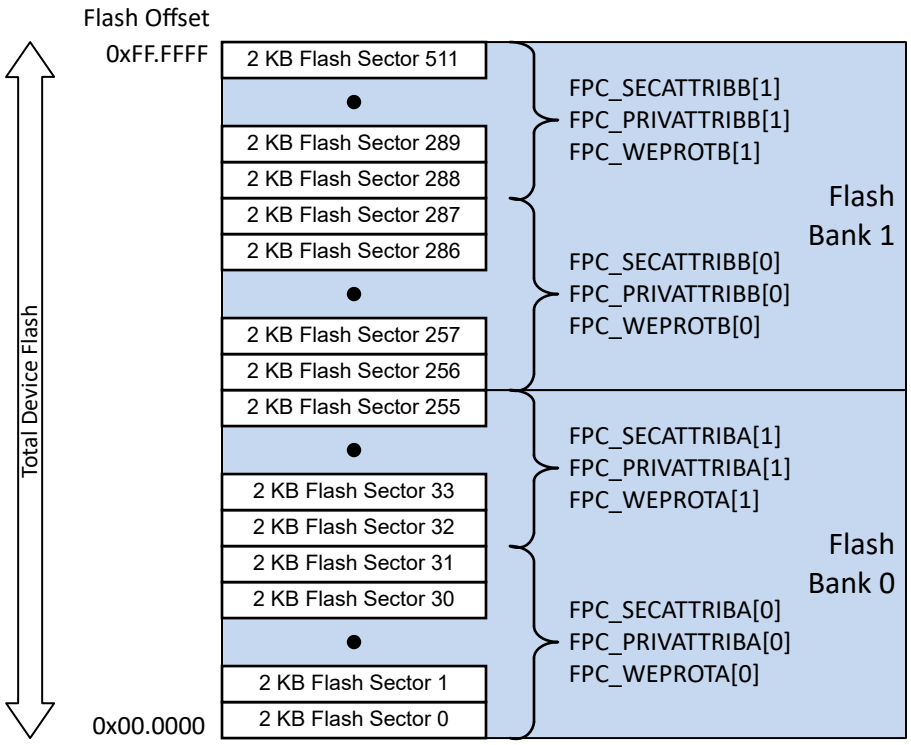


Figure 11-4. Flash mapping to Sector

In a dual-bank flash, security property registers with the suffix A controls the lower address space in Bank-0 and suffix B controls the upper address space in Bank-1.

11.2.4.1 Flash Bank Security Implementation

As mentioned earlier, the firewall control for the program flash bank is at a 2kB sector level. To ensure maximum flexibility and reasonable ease of use, each pair of security and privilege attribute register configures firewalls slightly differently.

- FPC_SECATTRIBAx, FPC_PRIVATTRIBAx, FPC_WEPROTAx: Each bit is mapped to individual 2kB sector starting at sector 0 upto sector 31. All 32-bits of the register are used for the mapping with bit-0 mapped to sector 0 and bit 31 mapped to sector 31.
- FPC_SECATTRIBBx, FPC_PRIVATTRIBBx, FPC_WEPROTBx: Each bit is mapped to a set of 8 2kB sectors starting at sector 32 upto sector 255. For a 1MB flash size, as there are 224 sectors left after removing the first 32 sectors, the registers have 28 bits used. Bit-0 is mapped to sectors 32 to 39 and bit-27 is mapped to sector 248 to sector 255.

The Table 11-1 shows some example of the register mapping for different flash sizes. The "x" in the register name should be replace by "0" for lower bank and "1" for upper bank, in a dual-bank configuration. In a single bank configuration it shall only have "0".

Table 11-1. FPC Firewall Register Decoding for 1 MB Flash

Flash Secure Access Register	Flash Privilege Access Register	Flash Write-Erase Protect Register	Flash Sector (1 MB)	Flash Sector (512 kB)
FPC_SECATTRIBAx[0]	FPC_PRIVATTRIBAx[0]	FPC_WEPROTAx[0]	0	0
FPC_SECATTRIBAx[1]	FPC_PRIVATTRIBAx[1]	FPC_WEPROTAx[1]	1	1
FPC_SECATTRIBAx[2]	FPC_PRIVATTRIBAx[2]	FPC_WEPROTAx[2]	2	2
FPC_SECATTRIBAx[3]	FPC_PRIVATTRIBAx[3]	FPC_WEPROTAx[3]	3	3
FPC_SECATTRIBAx[4]	FPC_PRIVATTRIBAx[4]	FPC_WEPROTAx[4]	4	4
FPC_SECATTRIBAx[5]	FPC_PRIVATTRIBAx[5]	FPC_WEPROTAx[5]	5	5
FPC_SECATTRIBAx[6]	FPC_PRIVATTRIBAx[6]	FPC_WEPROTAx[6]	6	6
FPC_SECATTRIBAx[7]	FPC_PRIVATTRIBAx[7]	FPC_WEPROTAx[7]	7	7
FPC_SECATTRIBAx[8]	FPC_PRIVATTRIBAx[8]	FPC_WEPROTAx[8]	8	8
FPC_SECATTRIBAx[9]	FPC_PRIVATTRIBAx[9]	FPC_WEPROTAx[9]	9	9
FPC_SECATTRIBAx[10]	FPC_PRIVATTRIBAx[10]	FPC_WEPROTAx[10]	10	10
FPC_SECATTRIBAx[11]	FPC_PRIVATTRIBAx[11]	FPC_WEPROTAx[11]	11	11
FPC_SECATTRIBAx[12]	FPC_PRIVATTRIBAx[12]	FPC_WEPROTAx[12]	12	12
FPC_SECATTRIBAx[13]	FPC_PRIVATTRIBAx[13]	FPC_WEPROTAx[13]	13	13
FPC_SECATTRIBAx[14]	FPC_PRIVATTRIBAx[14]	FPC_WEPROTAx[14]	14	14
FPC_SECATTRIBAx[15]	FPC_PRIVATTRIBAx[15]	FPC_WEPROTAx[15]	15	15
FPC_SECATTRIBAx[16]	FPC_PRIVATTRIBAx[16]	FPC_WEPROTAx[16]	16	16
FPC_SECATTRIBAx[17]	FPC_PRIVATTRIBAx[17]	FPC_WEPROTAx[17]	17	17
FPC_SECATTRIBAx[18]	FPC_PRIVATTRIBAx[18]	FPC_WEPROTAx[18]	18	18
FPC_SECATTRIBAx[19]	FPC_PRIVATTRIBAx[19]	FPC_WEPROTAx[19]	19	19
FPC_SECATTRIBAx[20]	FPC_PRIVATTRIBAx[20]	FPC_WEPROTAx[20]	20	20
FPC_SECATTRIBAx[21]	FPC_PRIVATTRIBAx[21]	FPC_WEPROTAx[21]	21	21
FPC_SECATTRIBAx[22]	FPC_PRIVATTRIBAx[22]	FPC_WEPROTAx[22]	22	22
FPC_SECATTRIBAx[23]	FPC_PRIVATTRIBAx[23]	FPC_WEPROTAx[23]	23	23
FPC_SECATTRIBAx[24]	FPC_PRIVATTRIBAx[24]	FPC_WEPROTAx[24]	24	24
FPC_SECATTRIBAx[25]	FPC_PRIVATTRIBAx[25]	FPC_WEPROTAx[25]	25	25

Table 11-1. FPC Firewall Register Decoding for 1 MB Flash (continued)

Flash Secure Access Register	Flash Privilege Access Register	Flash Write-Erase Protect Register	Flash Sector (1 MB)	Flash Sector (512 kB)
FPC_SECATTRIBAx[26]	FPC_PRIVATTRIBAx[26]	FPC_WEPROTAx[26]	26	26
FPC_SECATTRIBAx[27]	FPC_PRIVATTRIBAx[27]	FPC_WEPROTAx[27]	27	27
FPC_SECATTRIBAx[28]	FPC_PRIVATTRIBAx[28]	FPC_WEPROTAx[28]	28	28
FPC_SECATTRIBAx[29]	FPC_PRIVATTRIBAx[29]	FPC_WEPROTAx[29]	29	29
FPC_SECATTRIBAx[30]	FPC_PRIVATTRIBAx[30]	FPC_WEPROTAx[30]	30	30
FPC_SECATTRIBAx[31]	FPC_PRIVATTRIBAx[31]	FPC_WEPROTAx[31]	31	31
FPC_SECATTRIBBx[0]	FPC_PRIVATTRIBBx[0]	FPC_WEPROTBx[0]	32 - 39	32 - 39
FPC_SECATTRIBBx[1]	FPC_PRIVATTRIBBx[1]	FPC_WEPROTBx[1]	40 - 47	40 - 47
FPC_SECATTRIBBx[2]	FPC_PRIVATTRIBBx[2]	FPC_WEPROTBx[2]	48 - 55	48 - 55
FPC_SECATTRIBBx[3]	FPC_PRIVATTRIBBx[3]	FPC_WEPROTBx[3]	56 - 63	56 - 63
FPC_SECATTRIBBx[4]	FPC_PRIVATTRIBBx[4]	FPC_WEPROTBx[4]	64 - 71	64 - 71
FPC_SECATTRIBBx[5]	FPC_PRIVATTRIBBx[5]	FPC_WEPROTBx[5]	72 - 79	72 - 79
FPC_SECATTRIBBx[6]	FPC_PRIVATTRIBBx[6]	FPC_WEPROTBx[6]	80 - 87	80 - 87
FPC_SECATTRIBBx[7]	FPC_PRIVATTRIBBx[7]	FPC_WEPROTBx[7]	88 - 95	88 - 95
FPC_SECATTRIBBx[8]	FPC_PRIVATTRIBBx[8]	FPC_WEPROTBx[8]	96 - 103	96 - 103
FPC_SECATTRIBBx[9]	FPC_PRIVATTRIBBx[9]	FPC_WEPROTBx[9]	104 - 111	104 - 111
FPC_SECATTRIBBx[10]	FPC_PRIVATTRIBBx[10]	FPC_WEPROTBx[10]	112 - 119	112 - 119
FPC_SECATTRIBBx[11]	FPC_PRIVATTRIBBx[11]	FPC_WEPROTBx[11]	120 - 127	120 - 127
FPC_SECATTRIBBx[12]	FPC_PRIVATTRIBBx[12]	FPC_WEPROTBx[12]	128 - 135	DNU
FPC_SECATTRIBBx[13]	FPC_PRIVATTRIBBx[13]	FPC_WEPROTBx[13]	136 - 143	DNU
FPC_SECATTRIBBx[14]	FPC_PRIVATTRIBBx[14]	FPC_WEPROTBx[14]	144 - 151	DNU
FPC_SECATTRIBBx[15]	FPC_PRIVATTRIBBx[15]	FPC_WEPROTBx[15]	152 - 159	DNU
FPC_SECATTRIBBx[16]	FPC_PRIVATTRIBBx[16]	FPC_WEPROTBx[16]	160 - 167	DNU
FPC_SECATTRIBBx[17]	FPC_PRIVATTRIBBx[17]	FPC_WEPROTBx[17]	168 - 175	DNU
FPC_SECATTRIBBx[18]	FPC_PRIVATTRIBBx[18]	FPC_WEPROTBx[18]	176 - 183	DNU
FPC_SECATTRIBBx[19]	FPC_PRIVATTRIBBx[19]	FPC_WEPROTBx[19]	184 - 191	DNU
FPC_SECATTRIBBx[20]	FPC_PRIVATTRIBBx[20]	FPC_WEPROTx1[20]	192 - 199	DNU
FPC_SECATTRIBBx[21]	FPC_PRIVATTRIBBx[21]	FPC_WEPROTBx[21]	200 - 207	DNU
FPC_SECATTRIBBx[22]	FPC_PRIVATTRIBBx[22]	FPC_WEPROTBx[22]	208 - 215	DNU
FPC_SECATTRIBBx[23]	FPC_PRIVATTRIBBx[23]	FPC_WEPROTBx[23]	216 - 223	DNU
FPC_SECATTRIBBx[24]	FPC_PRIVATTRIBBx[24]	FPC_WEPROTBx[24]	224 - 231	DNU
FPC_SECATTRIBBx[25]	FPC_PRIVATTRIBBx[25]	FPC_WEPROTBx[25]	232 - 239	DNU
FPC_SECATTRIBBx[26]	FPC_PRIVATTRIBBx[26]	FPC_WEPROTBx[26]	240 - 247	DNU
FPC_SECATTRIBBx[27]	FPC_PRIVATTRIBBx[27]	FPC_WEPROTBx[27]	248 - 255	DNU

Note

Setting the bits marked as DNU (do not use) may have unpredictable results.

Similar to the program flash banks, some devices may implement an independent data flash bank. If the data bank is implemented, FPC_SECATTRIBAx_DFLASH, FPC_PRIVATTRIBAx_DFLASH and FPC_WEPROTAx_DFLASH are the firewall registers made available to the application. The data flash follows the same mapping for the register bit as the program flash. The first 32 sectors have one bit per sector and remaining data flash is 8 sectors per bit.

11.2.4.2 Flash Hide Protection

The FPC is also responsible for managing the Hide Protection (HDP) function. The HDP function allows an application to execute a section of code one time, and prevent subsequent read or execute access to the code. As an example, this is useful feature especially when running a second stage authentication code allowing main application to be validated before execution and prevent access to any stored symmetric or private keys.

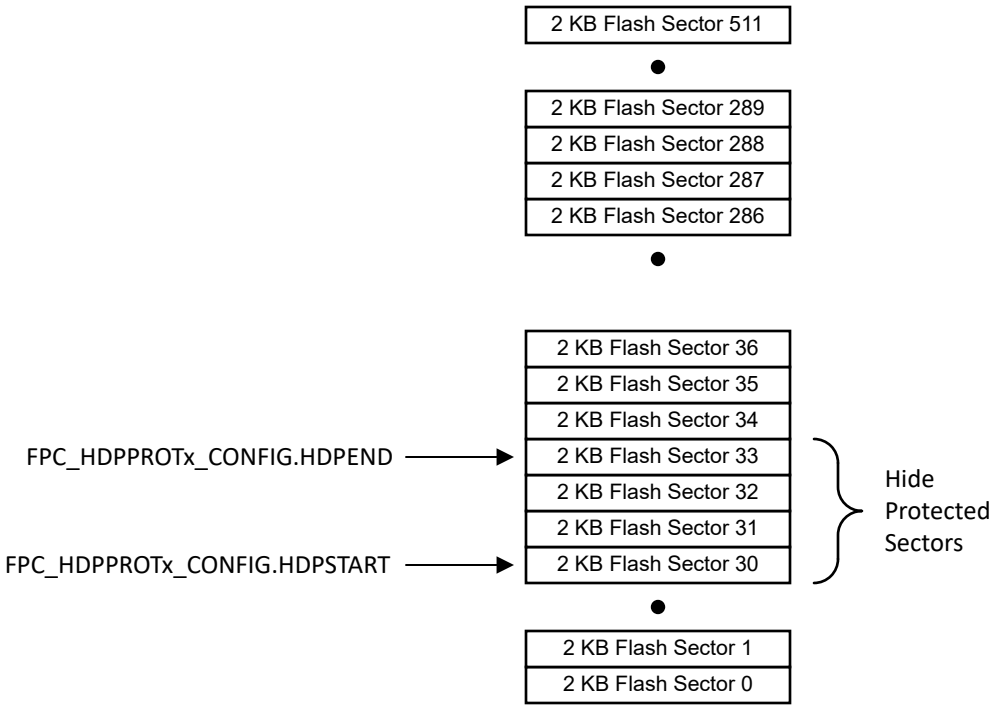


Figure 11-5. Flash Hide Protection Memory View

As shown in Figure 11-5, the HDPSTART and HDPEND fields in the FPC_HDPPROTx_CONFIG register define the start and end sector address which are to be hide protected. The Table 11-2 shows the valid configuration of values that define the sectors that can be hide protected.

Table 11-2. HDPSTART and HDPEND Rule

RULE	COMMENT
HDPSTART = HDPEND	Hide protection for exactly 1 sector identified by HDPSTART
HDPSTART < HDPEND	Hide protection for sectors from HDPSTART to HDPEND
HDPSTART > HDPEND	Invalid configuration. No hide protection

In addition to start and end sector address for HDP, there are two additional registers. The FPC_HDPPROT_CONTROL is used to validate the corresponding configuration register to be active. The FPC_HDPEN_CONTROL is to trigger the mechanism for hide protection. As shown in Figure 11-6, once the start sector, end sector and valid control bits are configured correctly, the HDP is ready. When CPU executes

the HDP region code, the HDP is armed. Any function call or jump outside of the HDP automatically enables the mechanism, preventing the application code from re-entering the HDP region. An access, execute or read, generates a non-maskable interrupt (NMI).

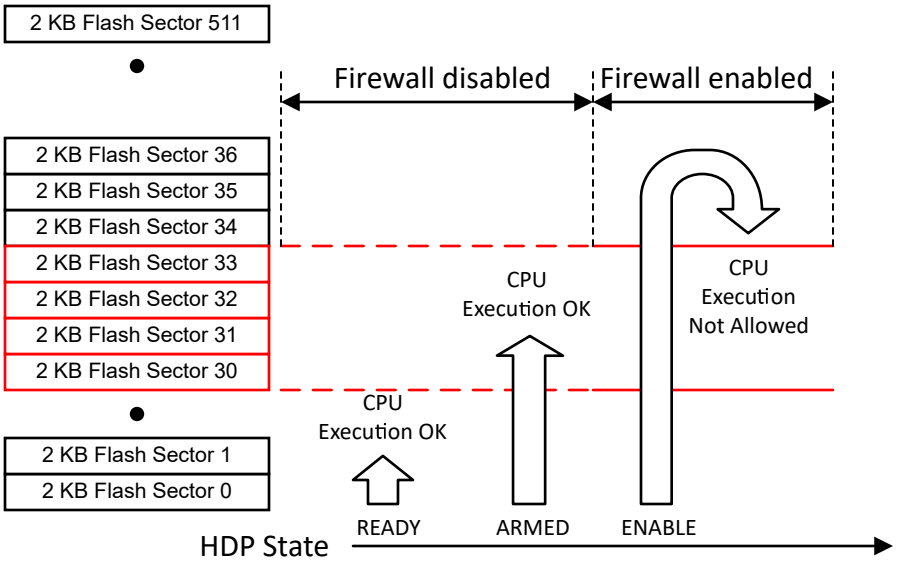


Figure 11-6. HDP execution mechanism

All the register are writeable once or can be set by the application code. System reset shall not allow the application to reconfigure and a BSL or higher reset is required.

Note

When executing code in HDP region followed by branch to SRAM, the SRAM code can access HDP region. Once all SRAM code execution is completed, it must branch back and exit HDP region with continued execution in flash to enable the HDP mechanism.

Note

If application does not want SRAM to access HDP region, then the code must implement a trampoline function to enable HDP before branching to SRAM code.

11.2.5 Strict Secure and Privilege Context Protection

The firewall configuration in GSC prevents non-secure and/or unprivileged initiator from accessing secure and/or privileged peripherals and memory. However secure and/or privileged initiator shall be able to acccess non-secure and/or unprivileged peripherals and memory. The attribute violation configuration register for PPC, SPC and FPC may however be configured to prevent the latter case.

This is typically the case where an application wants to avoid memory overflow or accidental peripheral access outside of the context. As shown in Table 11-3, on such an access, the GSC generates a NMI for the CPU. The corresponding write has no effect and read shall return all zero.

Table 11-3. Attribute Violation Rule Table

xPC_ATTRIBVIOL_CONFIG	Bus Transaction	Secure Target	Non-Secure Target
SECVIOL = 0	Secure Initiator	Allowed	Allowed
	Non-Secure Initiator	NMI	Allowed
SECVIOL = 1	Secure Initiator	Allowed	NMI
	Non-Secure Initiator	NMI	Allowed
PRIVVIOL = 0	Privileged Initiator	Allowed	Allowed
	Un-privileged Initator	NMI	Allowed

Table 11-3. Attribute Violation Rule Table (continued)

xPC_ATTRIBVIOL_CONFIG	Bus Transaction	Secure Target	Non-Secure Target
PRIVVIOL = 1	Privileged Initiator	Allowed	NMI
	Un-privileged Initator	NMI	Allowed

11.2.6 GSC Configuration Lock

The GSC also has the capability of locking and committing the security setting on a per region basis. There are total of 6 regions which can be individually locked or committed.

- Attribute Violation Configuration
- PPC Attribute
- SPC Attrribute
- FPC Attribute
- Vector Table Offset Register (VTOR)
- DMA trigger selection

When locked, the corresponding register in the region cannot be modified. To update a region register, the KEY of 0xA551 has to be written along with the modified register bit to unlock the register. When commit control is applied, then the lock register cannot be updated, even if the KEY is used.

The only mechanism to update when commit is applied, is a BOOTRST or higher for the Root of Trust to be executed, before the application can make changes.

11.3 GSC Registers

Table 11-4 lists the memory-mapped registers for the GSC registers. All register offset addresses not listed in Table 11-4 should be considered as reserved locations and the register contents should not be modified.

Table 11-4. GSC Registers

Offset	Acronym	Register Name	Section
1000h	SPC_ATTRIBVIOLS_CONFIG	SRAM Secure Access Attribute Violation Configuration Register	Section 11.3.1
1004h	SPC_ATTRIBVIOLP_CONFIG	SRAM Privilege Access Attribute Violation Configuration Register	Section 11.3.2
1010h	PPC_ATTRIBVIOLS_CONFIG	Peripheral Secure Access Attribute Violation Configuration Register	Section 11.3.3
1014h	PPC_ATTRIBVIOLP_CONFIG	Peripheral Privilege Access Attribute Violation Configuration Register	Section 11.3.4
1020h	FPC_ATTRIBVIOLS_CONFIG	Flash Secure Access Attribute Violation Configuration Register	Section 11.3.5
1024h	FPC_ATTRIBVIOLP_CONFIG	Flash Privilege Access Attribute Violation Configuration Register	Section 11.3.6
1040h	FPC_HDPPROT0_CONFIG	Flash Hide Protection for Bank0 Configuration Register	Section 11.3.7
1044h	FPC_HDPPROT1_CONFIG	Flash Hide Protection for Bank1 Configuration Register	Section 11.3.8
104Ch	FPC_HDPPROT_CONTROL	Flash Hide Protection Control Register	Section 11.3.9
1050h	FPC_HDPEN_CONTROL	Flash Hide Protection Enable Control Register	Section 11.3.10
1080h	FPC_HDPPROT_DBANK_CONFIG	Data Flash Hide Protection Configuration Register	Section 11.3.11
1084h	FPC_HDPPROT_DBANK_CONTROL	Data Flash Hide Protection Control Register	Section 11.3.12
1088h	FPC_HDPEN_DBANK_CONTROL	Data Flash Hide Protection Enable Control Register	Section 11.3.13
1100h	PPC_SECATTRIB_ADC	Peripheral Secure Access ADC	Section 11.3.14
1108h	PPC_SECATTRIB_TIMER	Peripheral Secure Access Timer	Section 11.3.15
1110h	PPC_SECATTRIB_UNICOMM	Peripheral Secure Access UniComm	Section 11.3.16
1118h	PPC_SECATTRIB_GPIO	Peripheral Secure Access GPIO	Section 11.3.17
1120h	PPC_SECATTRIB_MCAN	Peripheral Secure Access CAN	Section 11.3.18
1128h	PPC_SECATTRIB_I2S	Peripheral Secure Access I2S	Section 11.3.19
1130h	PPC_SECATTRIB_ACOMP	Peripheral Secure Access Analog Comp	Section 11.3.20
1138h	PPC_SECATTRIB_WDT	Peripheral Secure Access Watchdog	Section 11.3.21
1140h	PPC_SECATTRIB_LFSS	Peripheral Secure Access LFSS	Section 11.3.22
1148h	PPC_SECATTRIB_CRC	Peripheral Secure Access CRC	Section 11.3.23
1150h	PPC_SECATTRIB_AES	Peripheral Secure Access AES	Section 11.3.24
1158h	PPC_SECATTRIB_SHA	Peripheral Secure Access SHA	Section 11.3.25
1160h	PPC_SECATTRIB_PKA	Peripheral Secure Access PKA	Section 11.3.26
1168h	PPC_SECATTRIB_IOMUX	Peripheral Secure Access IOMUX	Section 11.3.27
1170h	PPC_SECATTRIB_QSPI	Peripheral Secure Access QSPI	Section 11.3.28
1178h	PPC_SECATTRIB_TRNG	Peripheral Secure Access TRNG	Section 11.3.29
1180h	PPC_SECATTRIB_DEBUGSS	Peripheral Secure Access DEBUGSS	Section 11.3.30
1188h	PPC_SECATTRIB_VREF	Peripheral Secure Access VREF	Section 11.3.31
1190h	PPC_SECATTRIB_KEYSTORECTL	Peripheral Secure Access KEYSTORECTL	Section 11.3.32
1198h	PPC_SECATTRIB_EAM	Peripheral Secure Access EAM	Section 11.3.33
11A0h	PPC_SECATTRIB_WUC	Peripheral Secure Access WUC	Section 11.3.34
11A8h	PPC_SECATTRIB_FRIREGS	Peripheral Secure Access FRIREGS	Section 11.3.35
11B0h	PPC_SECATTRIB_MEMCFG	Peripheral Secure Access MEMCFG	Section 11.3.36
1208h	PPC_SECATTRIB_SYSCTL	Peripheral Secure Access System Control	Section 11.3.37

Table 11-4. GSC Registers (continued)

Offset	Acronym	Register Name	Section
120Ch	PPC_PRIVATTRIB_SYSCTL	Peripheral Privilege Access System Control	Section 11.3.38
1210h	PPC_SECATTRIB_DMA	Peripheral Secure Access DMA	Section 11.3.39
1214h	PPC_PRIVATTRIB_DMA	Peripheral Privilege Access DMA	Section 11.3.40
1218h	PPC_SECATTRIB_GSC	Peripheral Secure Access GSC	Section 11.3.41
121Ch	PPC_PRIVATTRIB_GSC	Peripheral Privilege Access GSC	Section 11.3.42
1280h	SPC_SECATTRIB0	SRAM Secure Attribute-0 Register	Section 11.3.43
1284h	SPC_SECATTRIB1	SRAM Secure Attribute-1 Register	Section 11.3.44
1288h	SPC_SECATTRIB2	SRAM Secure Attribute-2 Register	Section 11.3.45
128Ch	SPC_SECATTRIB3	SRAM Secure Attribute-3 Register	Section 11.3.46
12C0h	SPC_PRIVATTRIB0	SRAM Privilege Attribute-0 Register	Section 11.3.47
12C4h	SPC_PRIVATTRIB1	SRAM Privilege Attribute-1 Register	Section 11.3.48
12C8h	SPC_PRIVATTRIB2	SRAM Privilege Attribute-2 Register	Section 11.3.49
12CCh	SPC_PRIVATTRIB3	SRAM Privilege Attribute-3 Register	Section 11.3.50
1300h	FPC_SECATTRIBA0	Flash Secure Attribute Register A0	Section 11.3.51
1304h	FPC_SECATTRIBA1	Flash Secure Attribute Register A1	Section 11.3.52
1308h	FPC_SECATTRIBB0	Flash Secure Attribute Register B0	Section 11.3.53
130Ch	FPC_SECATTRIBB1	Flash Secure Attribute Register B1	Section 11.3.54
1380h	FPC_SECATTRIBA_DFLASH	Data Flash Secure Attribute Register A	Section 11.3.55
1400h	FPC_PRIVATTRIBA0	Flash Privilege Attribute Register A0	Section 11.3.56
1404h	FPC_PRIVATTRIBA1	Flash Privilege Attribute Register A1	Section 11.3.57
1408h	FPC_PRIVATTRIBB0	Flash Privilege Attribute Register B0	Section 11.3.58
140Ch	FPC_PRIVATTRIBB1	Flash Privilege Attribute Register B1	Section 11.3.59
1480h	FPC_PRIVATTRIBA_DFLASH	Data Flash Privilege Attribute Register	Section 11.3.60
1600h	FPC_WEPROTA0	Flash Write Protect Attribute Register A0	Section 11.3.61
1604h	FPC_WEPROTA1	Flash Write Protect Attribute Register A1	Section 11.3.62
1608h	FPC_WEPROTB0	Flash Write Protect Attribute Register B0	Section 11.3.63
160Ch	FPC_WEPROTB1	Flash Write Protect Attribute Register B1	Section 11.3.64
1700h	FPC_WEPROTA_DFLASH	Data Flash Write Protect Attribute Register A	Section 11.3.65
1800h	FPC_FLSEMREQ	Flash semaphore request register	Section 11.3.66
1804h	FPC_FLSEMCLR	Flash semaphore release register	Section 11.3.67
1808h	FPC_FLSEMSTAT	Flash semaphore statys registers	Section 11.3.68
1A80h	DMA_TRIG_SEL	DMA Trigger Selection register	Section 11.3.69
1A84h	DMA_FIX_TRIG_SEL	DMA Fixed Trigger Selection register	Section 11.3.70
1B80h	VTOR_S	Secure Vector Table Offset Register	Section 11.3.71
1B84h	VTOR_NS	Non - Secure Vector Table Offset Register	Section 11.3.72
1D00h	DICE_CHECKSUM0	DICE checksum data register-0	Section 11.3.73
1D04h	DICE_CHECKSUM1	DICE checksum data register-1	Section 11.3.74
1D08h	DICE_CHECKSUM2	DICE checksum data register-2	Section 11.3.75
1D0Ch	DICE_CHECKSUM3	DICE checksum data register-3	Section 11.3.76
1D10h	DICE_CHECKSUM4	DICE checksum data register-4	Section 11.3.77
1D14h	DICE_CHECKSUM5	DICE checksum data register-5	Section 11.3.78
1D18h	DICE_CHECKSUM6	DICE checksum data register-6	Section 11.3.79
1D1Ch	DICE_CHECKSUM7	DICE checksum data register-7	Section 11.3.80
1D80h	GSC_LOCK	GSC Lock configuration register	Section 11.3.81
1D84h	GSC_COMMIT	GSC commit configuration register	Section 11.3.82

Table 11-4. GSC Registers (continued)

Offset	Acronym	Register Name	Section
1FFCh	GSC_REVISION	GSC Revision register	Section 11.3.83

Complex bit access types are encoded to fit into small table cells. Table 11-5 shows the codes that are used for access types in this section.

Table 11-5. GSC Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
WOnce	W Once	Write Write once
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.

11.3.1 SPC_ATTRIBVIOLS_CONFIG Register (Offset = 1000h) [Reset = 00000000h]

SPC_ATTRIBVIOLS_CONFIG is shown in Table 11-6.

Return to the Summary Table.

SRAM Secure Access Attribute Violation Configuration Register

Table 11-6. SPC_ATTRIBVIOLS_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	SECVIOL	R/W	0h	Controls the behavior of secure access to non-secure SRAM 0: Access is allowed 1: Access generates NMI Reset type: BOOTRST or Higher

11.3.2 SPC_ATTRIBVIOLP_CONFIG Register (Offset = 1004h) [Reset = 00000000h]

SPC_ATTRIBVIOLP_CONFIG is shown in Table 11-7.

Return to the Summary Table.

SRAM Privilege Access Attribute Violation Configuration Register

Table 11-7. SPC_ATTRIBVIOLP_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	PRIVVIOL	R/W	0h	Controls the behavior of privilege access to non-privilege SRAM 0: Access is allowed 1: Access generates NMI Reset type: BOOTRST or Higher

11.3.3 PPC_ATTRIBVIOLS_CONFIG Register (Offset = 1010h) [Reset = 00000000h]

PPC_ATTRIBVIOLS_CONFIG is shown in Table 11-8.

Return to the Summary Table.

Peripheral Secure Access Attribute Violation Configuration Register

Table 11-8. PPC_ATTRIBVIOLS_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	SECVIOL	R/W	0h	Controls the behavior of secure access to non-secure peripheral 0: Access is allowed 1: Access generates NMI Reset type: BOOTRST or Higher

11.3.4 PPC_ATTRIBVIOLP_CONFIG Register (Offset = 1014h) [Reset = 00000000h]

PPC_ATTRIBVIOLP_CONFIG is shown in Table 11-9.

Return to the Summary Table.

Peripheral Privilege Access Attribute Violation Configuration Register

Table 11-9. PPC_ATTRIBVIOLP_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	PRIVVIOL	R/W	0h	Controls the behavior of privilege access to non-privilege peripherals 0: Access is allowed 1: Access generates NMI Reset type: BOOTRST or Higher

11.3.5 FPC_ATTRIBVIOLS_CONFIG Register (Offset = 1020h) [Reset = 00000000h]

FPC_ATTRIBVIOLS_CONFIG is shown in Table 11-10.

Return to the Summary Table.

Flash Secure Access Attribute Violation Configuration Register

Table 11-10. FPC_ATTRIBVIOLS_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	SECVIOL	R/W	0h	Controls the behavior of secure access to non-secure Flash 0: Access is allowed 1: Access generates NMI Reset type: BOOTRST or Higher

11.3.6 FPC_ATTRIBVIOLP_CONFIG Register (Offset = 1024h) [Reset = 00000000h]

FPC_ATTRIBVIOLP_CONFIG is shown in Table 11-11.

Return to the Summary Table.

Flash Privilege Access Attribute Violation Configuration Register

Table 11-11. FPC_ATTRIBVIOLP_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	PRIVVIOL	R/W	0h	Controls the behavior of privilege access to non-privilege Flash 0: Access is allowed 1: Access generates NMI Reset type: BOOTRST or Higher

11.3.7 FPC_HDPPROT0_CONFIG Register (Offset = 1040h) [Reset = 00FF0000h]

FPC_HDPPROT0_CONFIG is shown in Table 11-12.

Return to the Summary Table.

Flash Hide Protection for Bank0 Configuration Register

Table 11-12. FPC_HDPPROT0_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-16	HDPSTART	WOnce	FFh	Sector number where hide protection starts Reset type: BSL Reset or Higher
15-8	RESERVED	R	0h	Reserved
7-0	HDPEND	WOnce	0h	Sector number where hide protection ends Reset type: BSL Reset or Higher

11.3.8 FPC_HDPPROT1_CONFIG Register (Offset = 1044h) [Reset = 00FF0000h]

FPC_HDPPROT1_CONFIG is shown in Table 11-13.

Return to the Summary Table.

Flash Hide Protection for Bank1 Configuration Register

Table 11-13. FPC_HDPPROT1_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-16	HDPSTART	WOnce	FFh	Sector number where hide protection starts Reset type: BSL Reset or Higher
15-8	RESERVED	R	0h	Reserved
7-0	HDPEND	WOnce	0h	Sector number where hide protection ends Reset type: BSL Reset or Higher

11.3.9 FPC_HDPPROT_CONTROL Register (Offset = 104Ch) [Reset = 00000000h]

FPC_HDPPROT_CONTROL is shown in Table 11-14.

Return to the Summary Table.

Flash Hide Protection Control Register

Table 11-14. FPC_HDPPROT_CONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1	HDPPROT1ACCDIS	R/W1S	0h	Enables the FPC_HDPPROT1_CONFIG to take effect. 0: HDP region access is enabled 1: HDP region access is disabled Reset type: BSL Reset or Higher
0	HDPPROT0ACCDIS	R/W1S	0h	Enables the FPC_HDPPROT0_CONFIG to take effect. 0: HDP region access is enabled 1: HDP region access is disabled Reset type: BSL Reset or Higher

11.3.10 FPC_HDPEN_CONTROL Register (Offset = 1050h) [Reset = 00000000h]

FPC_HDPEN_CONTROL is shown in Table 11-15.

Return to the Summary Table.

Flash Hide Protection Enable Control Register

Table 11-15. FPC_HDPEN_CONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	HDPEN	R/W1S	0h	HDP mode enable. This bit is set in the OTP during provisioning. 0: HDP mode is not used 1: HDP mode is enabled Reset type: BSL Reset or Higher

11.3.11 FPC_HDPPROT_DBANK_CONFIG Register (Offset = 1080h) [Reset = 000F0000h]

FPC_HDPPROT_DBANK_CONFIG is shown in Table 11-16.

Return to the Summary Table.

Data Flash Hide Protection Configuration Register

Table 11-16. FPC_HDPPROT_DBANK_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	Reserved
19-16	HDPSTART	WOnce	Fh	Sector number where hide protection starts Reset type: BSL Reset or Higher
15-4	RESERVED	R	0h	Reserved
3-0	HDPEND	WOnce	0h	Sector number where hide protection ends Reset type: BSL Reset or Higher

11.3.12 FPC_HDPPROT_DBANK_CONTROL Register (Offset = 1084h) [Reset = 00000000h]

FPC_HDPPROT_DBANK_CONTROL is shown in Table 11-17.

Return to the Summary Table.

Data Flash Hide Protection Control Register

Table 11-17. FPC_HDPPROT_DBANK_CONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	HDPPROTACCDIS	R/W1S	0h	Enables the HDP register to take effect for Databank 0: HDP region access is enabled 1: HDP region access is disabled Reset type: BSL Reset or Higher

11.3.13 FPC_HDPEN_DBANK_CONTROL Register (Offset = 1088h) [Reset = 00000000h]

FPC_HDPEN_DBANK_CONTROL is shown in Table 11-18.

Return to the Summary Table.

Data Flash Hide Protection Enable Control Register

Table 11-18. FPC_HDPEN_DBANK_CONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	HDPEN	R/W1S	0h	HDP mode enable. This bit is set in the OTP during provisioning and copied from the OTP memory during boot by BOOTROM. 0: HDP mode is not used 1: HDP mode is enabled Reset type: BSL Reset or Higher

11.3.14 PPC_SECATTRIB_ADC Register (Offset = 1100h) [Reset = 00000000h]

PPC_SECATTRIB_ADC is shown in Table 11-19.

Return to the Summary Table.

Peripheral Secure Access ADC

Table 11-19. PPC_SECATTRIB_ADC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1	SEC_ADC1	R/W	0h	Secure access enabled to ADC1 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
0	SEC_ADC0	R/W	0h	Secure access enabled to ADC0 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher

11.3.15 PPC_SECATTRIB_TIMER Register (Offset = 1108h) [Reset = 00000000h]

PPC_SECATTRIB_TIMER is shown in Table 11-20.

Return to the Summary Table.

Peripheral Secure Access Timer

Table 11-20. PPC_SECATTRIB_TIMER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25	SEC_TIMG8_1	R/W	0h	Secure access enabled to TIMG8_1 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
24	SEC_TIMG8_0	R/W	0h	Secure access enabled to TIMG8_0 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
23-17	RESERVED	R	0h	Reserved
16	SEC_TIMG12_0	R/W	0h	Secure access enabled to TIMG12_0 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
15-12	RESERVED	R	0h	Reserved
11	SEC_TIMG4_3	R/W	0h	Secure access enabled to TIMG4_3 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
10	SEC_TIMG4_2	R/W	0h	Secure access enabled to TIMG4_2 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
9	SEC_TIMG4_1	R/W	0h	Secure access enabled to TIMG4_1 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
8	SEC_TIMG4_0	R/W	0h	Secure access enabled to TIMG4_0 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
7-2	RESERVED	R	0h	Reserved
1	SEC_TIMA0_1	R/W	0h	Secure access enabled to TIMA0_1 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
0	SEC_TIMA0_0	R/W	0h	Secure access enabled to TIMA0_0 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher

11.3.16 PPC_SECATTRIB_UNICOMM Register (Offset = 1110h) [Reset = 00000000h]

PPC_SECATTRIB_UNICOMM is shown in Table 11-21.

Return to the Summary Table.

Peripheral Secure Access UniComm

Table 11-21. PPC_SECATTRIB_UNICOMM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R	0h	Reserved
10	SEC_S2U3	R/W	0h	Secure access enabled to UniComm S2U3 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
9	SEC_S2U2	R/W	0h	Secure access enabled to UniComm S2U2 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
8	SEC_S2U1	R/W	0h	Secure access enabled to UniComm S2U1 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
7	SEC_S2U0	R/W	0h	Secure access enabled to UniComm S2U0 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
6	SEC_S1U5	R/W	0h	Secure access enabled to UniComm S1U5 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
5	SEC_S1U4	R/W	0h	Secure access enabled to UniComm S1U4 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
4	SEC_S1U3	R/W	0h	Secure access enabled to UniComm S1U3 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
3	SEC_S1U2	R/W	0h	Secure access enabled to UniComm S1U2 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
2	SEC_S1U0	R/W	0h	Secure access enabled to UniComm S1U0 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
1	SEC_S0U2	R/W	0h	Secure access enabled to UniComm S0U2 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
0	SEC_S0U1	R/W	0h	Secure access enabled to UniComm S0U1 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher

11.3.17 PPC_SECATTRIB_GPIO Register (Offset = 1118h) [Reset = 00000000h]

PPC_SECATTRIB_GPIO is shown in Table 11-22.

Return to the Summary Table.

Peripheral Secure Access GPIO

Table 11-22. PPC_SECATTRIB_GPIO Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2	SEC_GPIOC	R/W	0h	Secure access enabled to GPIO C 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
1	SEC_GPIOB	R/W	0h	Secure access enabled to GPIO B 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
0	SEC_GPIOA	R/W	0h	Secure access enabled to GPIO A 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher

11.3.18 PPC_SECATTRIB_MCAN Register (Offset = 1120h) [Reset = 00000000h]

PPC_SECATTRIB_MCAN is shown in Table 11-23.

Return to the Summary Table.

Peripheral Secure Access CAN

Table 11-23. PPC_SECATTRIB_MCAN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1	SEC_MCAN1	R/W	0h	Secure access enabled to MCAN 1 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
0	SEC_MCAN0	R/W	0h	Secure access enabled to MCAN 0 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher

11.3.19 PPC_SECATTRIB_I2S Register (Offset = 1128h) [Reset = 00000000h]

PPC_SECATTRIB_I2S is shown in Table 11-24.

Return to the Summary Table.

Peripheral Secure Access I2S

Table 11-24. PPC_SECATTRIB_I2S Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1	SEC_I2S1	R/W	0h	Secure access enabled to I2S 1 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
0	SEC_I2S0	R/W	0h	Secure access enabled to I2S 0 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher

11.3.20 PPC_SECATTRIB_ACOMP Register (Offset = 1130h) [Reset = 00000000h]

PPC_SECATTRIB_ACOMP is shown in Table 11-25.

Return to the Summary Table.

Peripheral Secure Access Analog Comp

Table 11-25. PPC_SECATTRIB_ACOMP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1	SEC_ACOMP1	R/W	0h	Secure access enabled to Comparator 1 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
0	SEC_ACOMP0	R/W	0h	Secure access enabled to Comparator 0 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher

11.3.21 PPC_SECATTRIB_WDT Register (Offset = 1138h) [Reset = 00000000h]

PPC_SECATTRIB_WDT is shown in Table 11-26.

Return to the Summary Table.

Peripheral Secure Access Watchdog

Table 11-26. PPC_SECATTRIB_WDT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	SEC_WWDT	R/W	0h	Secure access enabled to Windowed Watchdog 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher

11.3.22 PPC_SECATTRIB_LFSS Register (Offset = 1140h) [Reset = 00000000h]

PPC_SECATTRIB_LFSS is shown in Table 11-27.

Return to the Summary Table.

Peripheral Secure Access LFSS

Table 11-27. PPC_SECATTRIB_LFSS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	SEC_LFSS	R/W	0h	Secure access enabled to LFSS (including peripherals inside LFSS e.g. IWDT) 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher

11.3.23 PPC_SECATTRIB_CRC Register (Offset = 1148h) [Reset = 00000001h]

PPC_SECATTRIB_CRC is shown in Table 11-28.

Return to the Summary Table.

Peripheral Secure Access CRC

Table 11-28. PPC_SECATTRIB_CRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	SEC_CRC	R/W	1h	Secure access enabled to CRC 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher

11.3.24 PPC_SECATTRIB_AES Register (Offset = 1150h) [Reset = 00000000h]

PPC_SECATTRIB_AES is shown in Table 11-29.

Return to the Summary Table.

Peripheral Secure Access AES

Table 11-29. PPC_SECATTRIB_AES Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	SEC_AES	R/W	0h	Secure access enabled to AES 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher

11.3.25 PPC_SECATTRIB_SHA Register (Offset = 1158h) [Reset = 00000000h]

PPC_SECATTRIB_SHA is shown in Table 11-30.

Return to the Summary Table.

Peripheral Secure Access SHA

Table 11-30. PPC_SECATTRIB_SHA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	SEC_SHA	R/W	0h	Secure access enabled to SHA 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher

11.3.26 PPC_SECATTRIB_PKA Register (Offset = 1160h) [Reset = 00000000h]

PPC_SECATTRIB_PKA is shown in Table 11-31.

Return to the Summary Table.

Peripheral Secure Access PKA

Table 11-31. PPC_SECATTRIB_PKA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	SEC_PKA	R/W	0h	Secure access enabled to PKA 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher

11.3.27 PPC_SECATTRIB_IOMUX Register (Offset = 1168h) [Reset = 00000000h]

PPC_SECATTRIB_IOMUX is shown in Table 11-32.

Return to the Summary Table.

Peripheral Secure Access IOMUX

Table 11-32. PPC_SECATTRIB_IOMUX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	SEC_IOMUX	R/W	0h	Secure access enabled to IOMUX 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher

11.3.28 PPC_SECATTRIB_QSPI Register (Offset = 1170h) [Reset = 00000000h]

PPC_SECATTRIB_QSPI is shown in Table 11-33.

Return to the Summary Table.

Peripheral Secure Access QSPI

Table 11-33. PPC_SECATTRIB_QSPI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	SEC_QSPI	R/W	0h	Secure access enabled to QSPI 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher

11.3.29 PPC_SECATTRIB_TRNG Register (Offset = 1178h) [Reset = 00000001h]

PPC_SECATTRIB_TRNG is shown in Table 11-34.

Return to the Summary Table.

Peripheral Secure Access TRNG

Table 11-34. PPC_SECATTRIB_TRNG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	SEC_TRNG	R/W	1h	Secure access enabled to TRNG 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher

11.3.30 PPC_SECATTRIB_DEBUGSS Register (Offset = 1180h) [Reset = 00000001h]

PPC_SECATTRIB_DEBUGSS is shown in Table 11-35.

Return to the Summary Table.

Peripheral Secure Access DEBUGSS

Table 11-35. PPC_SECATTRIB_DEBUGSS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	SEC_DEBUGSS	R/W	1h	Secure access enabled to DEBUGSS 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher

11.3.31 PPC_SECATTRIB_VREF Register (Offset = 1188h) [Reset = 00000000h]

PPC_SECATTRIB_VREF is shown in Table 11-36.

Return to the Summary Table.

Peripheral Secure Access VREF

Table 11-36. PPC_SECATTRIB_VREF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	SEC_VREF	R/W	0h	Secure access enabled to VREF 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher

11.3.32 PPC_SECATTRIB_KEYSTORECTL Register (Offset = 1190h) [Reset = 00000000h]

PPC_SECATTRIB_KEYSTORECTL is shown in Table 11-37.

Return to the Summary Table.

Peripheral Secure Access KEYSTORECTL

Table 11-37. PPC_SECATTRIB_KEYSTORECTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	SEC_KEYSTORECTL	R/W	0h	Secure access enabled to KEYSTORECTL 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher

11.3.33 PPC_SECATTRIB_EAM Register (Offset = 1198h) [Reset = 00000000h]

PPC_SECATTRIB_EAM is shown in Table 11-38.

Return to the Summary Table.

Peripheral Secure Access EAM

Table 11-38. PPC_SECATTRIB_EAM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	SEC_EAM	R/W	0h	Secure access enabled to EAM 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher

11.3.34 PPC_SECATTRIB_WUC Register (Offset = 11A0h) [Reset = 00000000h]

PPC_SECATTRIB_WUC is shown in Table 11-39.

Return to the Summary Table.

Peripheral Secure Access WUC

Table 11-39. PPC_SECATTRIB_WUC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	SEC_WUC	R/W	0h	Secure access enabled to WUC 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher

11.3.35 PPC_SECATTRIB_FRIREGS Register (Offset = 11A8h) [Reset = 00000000h]

PPC_SECATTRIB_FRIREGS is shown in Table 11-40.

Return to the Summary Table.

Peripheral Secure Access FRIREGS

Table 11-40. PPC_SECATTRIB_FRIREGS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	SEC_FRIREGS	R/W	0h	Secure access enabled to FRIREGS 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher

11.3.36 PPC_SECATTRIB_MEMCFG Register (Offset = 11B0h) [Reset = 00000000h]

PPC_SECATTRIB_MEMCFG is shown in Table 11-41.

Return to the Summary Table.

Peripheral Secure Access MEMCFG

Table 11-41. PPC_SECATTRIB_MEMCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	SEC_MEMCFG	R/W	0h	Secure access enabled to MEMCFG 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher

11.3.37 PPC_SECATTRIB_SYSCTL Register (Offset = 1208h) [Reset = 00000001h]

PPC_SECATTRIB_SYSCTL is shown in Table 11-42.

Return to the Summary Table.

Peripheral Secure Access System Control

Table 11-42. PPC_SECATTRIB_SYSCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	SEC_SYSCTL	R/W	1h	Secure access enabled to SYSCTL 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher

11.3.38 PPC_PRIVATTRIB_SYSCTL Register (Offset = 120Ch) [Reset = 00000001h]

PPC_PRIVATTRIB_SYSCTL is shown in Table 11-43.

Return to the Summary Table.

Peripheral Privilege Access System Control

Table 11-43. PPC_PRIVATTRIB_SYSCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	PRIV_SYSCTL	R/W	1h	Privilege access enabled to SYSCTL 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: BOOTRST or Higher

11.3.39 PPC_SECATTRIB_DMA Register (Offset = 1210h) [Reset = 00000001h]

PPC_SECATTRIB_DMA is shown in Table 11-44.

Return to the Summary Table.

Peripheral Secure Access DMA

Table 11-44. PPC_SECATTRIB_DMA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1	SEC_DMA1	R/W	0h	Secure access enabled to DMA1 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
0	SEC_DMA0	R/W	1h	Secure access enabled to DMA0 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher

11.3.40 PPC_PRIVATTRIB_DMA Register (Offset = 1214h) [Reset = 00000001h]

PPC_PRIVATTRIB_DMA is shown in Table 11-45.

Return to the Summary Table.

Peripheral Privilege Access DMA

Table 11-45. PPC_PRIVATTRIB_DMA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1	PRIV_DMA1	R/W	0h	Privilege access enabled to DMA1 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: BOOTRST or Higher
0	PRIV_DMA0	R/W	1h	Privilege access enabled to DMA0 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: BOOTRST or Higher

11.3.41 PPC_SECATTRIB_GSC Register (Offset = 1218h) [Reset = 00000001h]

PPC_SECATTRIB_GSC is shown in Table 11-46.

Return to the Summary Table.

Peripheral Secure Access GSC

Table 11-46. PPC_SECATTRIB_GSC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	SEC_GSC	R/W	1h	Secure access enabled to GSC 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher

11.3.42 PPC_PRIVATTRIB_GSC Register (Offset = 121Ch) [Reset = 00000001h]

PPC_PRIVATTRIB_GSC is shown in Table 11-47.

Return to the Summary Table.

Peripheral Privilege Access GSC

Table 11-47. PPC_PRIVATTRIB_GSC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	PRIV_GSC	R/W	1h	Privilege access enabled to GSC 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: BOOTRST or Higher

11.3.43 SPC_SECATTRIB0 Register (Offset = 1280h) [Reset = 7F7F7F7Fh]

SPC_SECATTRIB0 is shown in Table 11-48.

Return to the Summary Table.

SRAM Secure Attribute-0 Register

Table 11-48. SPC_SECATTRIB0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	Reserved
30	C3B2_4KB	R/W	1h	Secure access enabled to 3rd 4KB in chunk-3 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
29	C3B1_4KB	R/W	1h	Secure access enabled to 2nd 4KB in chunk-3 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
28	C3B0_4KB	R/W	1h	Secure access enabled to 1st 4KB in chunk-3 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
27	C3B0_2KB	R/W	1h	Secure access enabled to 2KB in chunk-3 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
26	C3B0_1KB	R/W	1h	Secure access enabled to 1KB in chunk-3 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
25	C3B1_512B	R/W	1h	Secure access enabled to 2nd 512 byte in chunk-3 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
24	C3B0_512B	R/W	1h	Secure access enabled to 1st 512 byte in chunk-3 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
23	RESERVED	R	0h	Reserved
22	C2B2_4KB	R/W	1h	Secure access enabled to 3rd 4KB in chunk-2 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
21	C2B1_4KB	R/W	1h	Secure access enabled to 2nd 4KB in chunk-2 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
20	C2B0_4KB	R/W	1h	Secure access enabled to 1st 4KB in chunk-2 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
19	C2B0_2KB	R/W	1h	Secure access enabled to 2KB in chunk-2 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher

Table 11-48. SPC_SECATTRIB0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	C2B0_1KB	R/W	1h	Secure access enabled to 1KB in chunk-2 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
17	C2B1_512B	R/W	1h	Secure access enabled to 2nd 512 byte in chunk-2 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
16	C2B0_512B	R/W	1h	Secure access enabled to 1st 512 byte in chunk-2 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
15	RESERVED	R	0h	Reserved
14	C1B2_4KB	R/W	1h	Secure access enabled to 3rd 4KB in chunk-1 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
13	C1B1_4KB	R/W	1h	Secure access enabled to 2nd 4KB in chunk-1 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
12	C1B0_4KB	R/W	1h	Secure access enabled to 1st 4KB in chunk-1 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
11	C1B0_2KB	R/W	1h	Secure access enabled to 2KB in chunk-1 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
10	C1B0_1KB	R/W	1h	Secure access enabled to 1KB in chunk-1 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
9	C1B1_512B	R/W	1h	Secure access enabled to 2nd 512 byte in chunk-1 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
8	C1B0_512B	R/W	1h	Secure access enabled to 1st 512 byte in chunk-1 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
7	RESERVED	R	0h	Reserved
6	C0B2_4KB	R/W	1h	Secure access enabled to 3rd 4KB in chunk-0 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
5	C0B1_4KB	R/W	1h	Secure access enabled to 2nd 4KB in chunk-0 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
4	C0B0_4KB	R/W	1h	Secure access enabled to 1st 4KB in chunk-0 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher

Table 11-48. SPC_SECATTRIB0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	C0B0_2KB	R/W	1h	Secure access enabled to 2KB in chunk-0 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
2	C0B0_1KB	R/W	1h	Secure access enabled to 1KB in chunk-0 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
1	C0B1_512B	R/W	1h	Secure access enabled to 2nd 512 byte in chunk-0 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
0	C0B0_512B	R/W	1h	Secure access enabled to 1st 512 byte in chunk-0 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher

11.3.44 SPC_SECATTRIB1 Register (Offset = 1284h) [Reset = 7F7F7F7Fh]

SPC_SECATTRIB1 is shown in Table 11-49.

Return to the Summary Table.

SRAM Secure Attribute-1 Register

Table 11-49. SPC_SECATTRIB1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	Reserved
30	C3B2_4KB	R/W	1h	Secure access enabled to 3rd 4KB in chunk-3 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
29	C3B1_4KB	R/W	1h	Secure access enabled to 2nd 4KB in chunk-3 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
28	C3B0_4KB	R/W	1h	Secure access enabled to 1st 4KB in chunk-3 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
27	C3B0_2KB	R/W	1h	Secure access enabled to 2KB in chunk-3 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
26	C3B0_1KB	R/W	1h	Secure access enabled to 1KB in chunk-3 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
25	C3B1_512B	R/W	1h	Secure access enabled to 2nd 512 byte in chunk-3 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
24	C3B0_512B	R/W	1h	Secure access enabled to 1st 512 byte in chunk-3 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
23	RESERVED	R	0h	Reserved
22	C2B2_4KB	R/W	1h	Secure access enabled to 3rd 4KB in chunk-2 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
21	C2B1_4KB	R/W	1h	Secure access enabled to 2nd 4KB in chunk-2 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
20	C2B0_4KB	R/W	1h	Secure access enabled to 1st 4KB in chunk-2 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
19	C2B0_2KB	R/W	1h	Secure access enabled to 2KB in chunk-2 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher

Table 11-49. SPC_SECATTRIB1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	C2B0_1KB	R/W	1h	Secure access enabled to 1KB in chunk-2 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
17	C2B1_512B	R/W	1h	Secure access enabled to 2nd 512 byte in chunk-2 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
16	C2B0_512B	R/W	1h	Secure access enabled to 1st 512 byte in chunk-2 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
15	RESERVED	R	0h	Reserved
14	C1B2_4KB	R/W	1h	Secure access enabled to 3rd 4KB in chunk-1 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
13	C1B1_4KB	R/W	1h	Secure access enabled to 2nd 4KB in chunk-1 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
12	C1B0_4KB	R/W	1h	Secure access enabled to 1st 4KB in chunk-1 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
11	C1B0_2KB	R/W	1h	Secure access enabled to 2KB in chunk-1 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
10	C1B0_1KB	R/W	1h	Secure access enabled to 1KB in chunk-1 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
9	C1B1_512B	R/W	1h	Secure access enabled to 2nd 512 byte in chunk-1 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
8	C1B0_512B	R/W	1h	Secure access enabled to 1st 512 byte in chunk-1 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
7	RESERVED	R	0h	Reserved
6	C0B2_4KB	R/W	1h	Secure access enabled to 3rd 4KB in chunk-0 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
5	C0B1_4KB	R/W	1h	Secure access enabled to 2nd 4KB in chunk-0 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
4	C0B0_4KB	R/W	1h	Secure access enabled to 1st 4KB in chunk-0 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher

Table 11-49. SPC_SECATTRIB1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	C0B0_2KB	R/W	1h	Secure access enabled to 2KB in chunk-0 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
2	C0B0_1KB	R/W	1h	Secure access enabled to 1KB in chunk-0 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
1	C0B1_512B	R/W	1h	Secure access enabled to 2nd 512 byte in chunk-0 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
0	C0B0_512B	R/W	1h	Secure access enabled to 1st 512 byte in chunk-0 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher

11.3.45 SPC_SECATTRIB2 Register (Offset = 1288h) [Reset = 7F7F7F7Fh]

SPC_SECATTRIB2 is shown in Table 11-50.

Return to the Summary Table.

SRAM Secure Attribute-2 Register

Table 11-50. SPC_SECATTRIB2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	Reserved
30	C3B2_4KB	R/W	1h	Secure access enabled to 3rd 4KB in chunk-3 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
29	C3B1_4KB	R/W	1h	Secure access enabled to 2nd 4KB in chunk-3 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
28	C3B0_4KB	R/W	1h	Secure access enabled to 1st 4KB in chunk-3 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
27	C3B0_2KB	R/W	1h	Secure access enabled to 2KB in chunk-3 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
26	C3B0_1KB	R/W	1h	Secure access enabled to 1KB in chunk-3 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
25	C3B1_512B	R/W	1h	Secure access enabled to 2nd 512 byte in chunk-3 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
24	C3B0_512B	R/W	1h	Secure access enabled to 1st 512 byte in chunk-3 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
23	RESERVED	R	0h	Reserved
22	C2B2_4KB	R/W	1h	Secure access enabled to 3rd 4KB in chunk-2 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
21	C2B1_4KB	R/W	1h	Secure access enabled to 2nd 4KB in chunk-2 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
20	C2B0_4KB	R/W	1h	Secure access enabled to 1st 4KB in chunk-2 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
19	C2B0_2KB	R/W	1h	Secure access enabled to 2KB in chunk-2 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher

Table 11-50. SPC_SECATTRIB2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	C2B0_1KB	R/W	1h	Secure access enabled to 1KB in chunk-2 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
17	C2B1_512B	R/W	1h	Secure access enabled to 2nd 512 byte in chunk-2 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
16	C2B0_512B	R/W	1h	Secure access enabled to 1st 512 byte in chunk-2 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
15	RESERVED	R	0h	Reserved
14	C1B2_4KB	R/W	1h	Secure access enabled to 3rd 4KB in chunk-1 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
13	C1B1_4KB	R/W	1h	Secure access enabled to 2nd 4KB in chunk-1 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
12	C1B0_4KB	R/W	1h	Secure access enabled to 1st 4KB in chunk-1 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
11	C1B0_2KB	R/W	1h	Secure access enabled to 2KB in chunk-1 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
10	C1B0_1KB	R/W	1h	Secure access enabled to 1KB in chunk-1 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
9	C1B1_512B	R/W	1h	Secure access enabled to 2nd 512 byte in chunk-1 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
8	C1B0_512B	R/W	1h	Secure access enabled to 1st 512 byte in chunk-1 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
7	RESERVED	R	0h	Reserved
6	C0B2_4KB	R/W	1h	Secure access enabled to 3rd 4KB in chunk-0 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
5	C0B1_4KB	R/W	1h	Secure access enabled to 2nd 4KB in chunk-0 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
4	C0B0_4KB	R/W	1h	Secure access enabled to 1st 4KB in chunk-0 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher

Table 11-50. SPC_SECATTRIB2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	C0B0_2KB	R/W	1h	Secure access enabled to 2KB in chunk-0 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
2	C0B0_1KB	R/W	1h	Secure access enabled to 1KB in chunk-0 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
1	C0B1_512B	R/W	1h	Secure access enabled to 2nd 512 byte in chunk-0 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
0	C0B0_512B	R/W	1h	Secure access enabled to 1st 512 byte in chunk-0 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher

11.3.46 SPC_SECATTRIB3 Register (Offset = 128Ch) [Reset = 7F7F7F7Fh]

SPC_SECATTRIB3 is shown in Table 11-51.

Return to the Summary Table.

SRAM Secure Attribute-3 Register

Table 11-51. SPC_SECATTRIB3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	Reserved
30	C3B2_4KB	R/W	1h	Secure access enabled to 3rd 4KB in chunk-3 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
29	C3B1_4KB	R/W	1h	Secure access enabled to 2nd 4KB in chunk-3 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
28	C3B0_4KB	R/W	1h	Secure access enabled to 1st 4KB in chunk-3 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
27	C3B0_2KB	R/W	1h	Secure access enabled to 2KB in chunk-3 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
26	C3B0_1KB	R/W	1h	Secure access enabled to 1KB in chunk-3 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
25	C3B1_512B	R/W	1h	Secure access enabled to 2nd 512 byte in chunk-3 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
24	C3B0_512B	R/W	1h	Secure access enabled to 1st 512 byte in chunk-3 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
23	RESERVED	R	0h	Reserved
22	C2B2_4KB	R/W	1h	Secure access enabled to 3rd 4KB in chunk-2 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
21	C2B1_4KB	R/W	1h	Secure access enabled to 2nd 4KB in chunk-2 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
20	C2B0_4KB	R/W	1h	Secure access enabled to 1st 4KB in chunk-2 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
19	C2B0_2KB	R/W	1h	Secure access enabled to 2KB in chunk-2 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher

Table 11-51. SPC_SECATTRIB3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	C2B0_1KB	R/W	1h	Secure access enabled to 1KB in chunk-2 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
17	C2B1_512B	R/W	1h	Secure access enabled to 2nd 512 byte in chunk-2 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
16	C2B0_512B	R/W	1h	Secure access enabled to 1st 512 byte in chunk-2 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
15	RESERVED	R	0h	Reserved
14	C1B2_4KB	R/W	1h	Secure access enabled to 3rd 4KB in chunk-1 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
13	C1B1_4KB	R/W	1h	Secure access enabled to 2nd 4KB in chunk-1 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
12	C1B0_4KB	R/W	1h	Secure access enabled to 1st 4KB in chunk-1 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
11	C1B0_2KB	R/W	1h	Secure access enabled to 2KB in chunk-1 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
10	C1B0_1KB	R/W	1h	Secure access enabled to 1KB in chunk-1 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
9	C1B1_512B	R/W	1h	Secure access enabled to 2nd 512 byte in chunk-1 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
8	C1B0_512B	R/W	1h	Secure access enabled to 1st 512 byte in chunk-1 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
7	RESERVED	R	0h	Reserved
6	C0B2_4KB	R/W	1h	Secure access enabled to 3rd 4KB in chunk-0 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
5	C0B1_4KB	R/W	1h	Secure access enabled to 2nd 4KB in chunk-0 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
4	C0B0_4KB	R/W	1h	Secure access enabled to 1st 4KB in chunk-0 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher

Table 11-51. SPC_SECATTRIB3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	C0B0_2KB	R/W	1h	Secure access enabled to 2KB in chunk-0 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
2	C0B0_1KB	R/W	1h	Secure access enabled to 1KB in chunk-0 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
1	C0B1_512B	R/W	1h	Secure access enabled to 2nd 512 byte in chunk-0 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
0	C0B0_512B	R/W	1h	Secure access enabled to 1st 512 byte in chunk-0 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher

11.3.47 SPC_PRIVATTRIB0 Register (Offset = 12C0h) [Reset = 7F7F7F7Fh]

SPC_PRIVATTRIB0 is shown in Table 11-52.

Return to the Summary Table.

SRAM Privilege Attribute-0 Register

Table 11-52. SPC_PRIVATTRIB0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	Reserved
30	C3B2_4KB	R/W	1h	privilege access enabled to 3rd 4KB in chunk-3 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
29	C3B1_4KB	R/W	1h	privilege access enabled to 2nd 4KB in chunk-3 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
28	C3B0_4KB	R/W	1h	privilege access enabled to 1st 4KB in chunk-3 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
27	C3B0_2KB	R/W	1h	privilege access enabled to 2KB in chunk-3 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
26	C3B0_1KB	R/W	1h	privilege access enabled to 1KB in chunk-3 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
25	C3B1_512B	R/W	1h	privilege access enabled to 2nd 512 byte in chunk-3 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
24	C3B0_512B	R/W	1h	privilege access enabled to 1st 512 byte in chunk-3 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
23	RESERVED	R	0h	Reserved
22	C2B2_4KB	R/W	1h	privilege access enabled to 3rd 4KB in chunk-2 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
21	C2B1_4KB	R/W	1h	privilege access enabled to 2nd 4KB in chunk-2 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
20	C2B0_4KB	R/W	1h	privilege access enabled to 1st 4KB in chunk-2 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
19	C2B0_2KB	R/W	1h	privilege access enabled to 2KB in chunk-2 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher

Table 11-52. SPC_PRIVATTRIB0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	C2B0_1KB	R/W	1h	privilege access enabled to 1KB in chunk-2 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
17	C2B1_512B	R/W	1h	privilege access enabled to 2nd 512 byte in chunk-2 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
16	C2B0_512B	R/W	1h	privilege access enabled to 1st 512 byte in chunk-2 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
15	RESERVED	R	0h	Reserved
14	C1B2_4KB	R/W	1h	privilege access enabled to 3rd 4KB in chunk-1 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
13	C1B1_4KB	R/W	1h	privilege access enabled to 2nd 4KB in chunk-1 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
12	C1B0_4KB	R/W	1h	privilege access enabled to 1st 4KB in chunk-1 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
11	C1B0_2KB	R/W	1h	privilege access enabled to 2KB in chunk-1 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
10	C1B0_1KB	R/W	1h	privilege access enabled to 1KB in chunk-1 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
9	C1B1_512B	R/W	1h	privilege access enabled to 2nd 512 byte in chunk-1 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
8	C1B0_512B	R/W	1h	privilege access enabled to 1st 512 byte in chunk-1 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
7	RESERVED	R	0h	Reserved
6	C0B2_4KB	R/W	1h	privilege access enabled to 3rd 4KB in chunk-0 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
5	C0B1_4KB	R/W	1h	privilege access enabled to 2nd 4KB in chunk-0 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
4	C0B0_4KB	R/W	1h	privilege access enabled to 1st 4KB in chunk-0 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher

Table 11-52. SPC_PRIVATTRIB0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	C0B0_2KB	R/W	1h	privilege access enabled to 2KB in chunk-0 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
2	C0B0_1KB	R/W	1h	privilege access enabled to 1KB in chunk-0 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
1	C0B1_512B	R/W	1h	privilege access enabled to 2nd 512 byte in chunk-0 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
0	C0B0_512B	R/W	1h	privilege access enabled to 1st 512 byte in chunk-0 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher

11.3.48 SPC_PRIVATTRIB1 Register (Offset = 12C4h) [Reset = 7F7F7F7Fh]

SPC_PRIVATTRIB1 is shown in Table 11-53.

Return to the Summary Table.

SRAM Privilege Attribute-1 Register

Table 11-53. SPC_PRIVATTRIB1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	Reserved
30	C3B2_4KB	R/W	1h	privilege access enabled to 3rd 4KB in chunk-3 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
29	C3B1_4KB	R/W	1h	privilege access enabled to 2nd 4KB in chunk-3 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
28	C3B0_4KB	R/W	1h	privilege access enabled to 1st 4KB in chunk-3 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
27	C3B0_2KB	R/W	1h	privilege access enabled to 2KB in chunk-3 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
26	C3B0_1KB	R/W	1h	privilege access enabled to 1KB in chunk-3 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
25	C3B1_512B	R/W	1h	privilege access enabled to 2nd 512 byte in chunk-3 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
24	C3B0_512B	R/W	1h	privilege access enabled to 1st 512 byte in chunk-3 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
23	RESERVED	R	0h	Reserved
22	C2B2_4KB	R/W	1h	privilege access enabled to 3rd 4KB in chunk-2 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
21	C2B1_4KB	R/W	1h	privilege access enabled to 2nd 4KB in chunk-2 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
20	C2B0_4KB	R/W	1h	privilege access enabled to 1st 4KB in chunk-2 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
19	C2B0_2KB	R/W	1h	privilege access enabled to 2KB in chunk-2 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher

Table 11-53. SPC_PRIVATTRIB1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	C2B0_1KB	R/W	1h	privilege access enabled to 1KB in chunk-2 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
17	C2B1_512B	R/W	1h	privilege access enabled to 2nd 512 byte in chunk-2 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
16	C2B0_512B	R/W	1h	privilege access enabled to 1st 512 byte in chunk-2 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
15	RESERVED	R	0h	Reserved
14	C1B2_4KB	R/W	1h	privilege access enabled to 3rd 4KB in chunk-1 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
13	C1B1_4KB	R/W	1h	privilege access enabled to 2nd 4KB in chunk-1 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
12	C1B0_4KB	R/W	1h	privilege access enabled to 1st 4KB in chunk-1 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
11	C1B0_2KB	R/W	1h	privilege access enabled to 2KB in chunk-1 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
10	C1B0_1KB	R/W	1h	privilege access enabled to 1KB in chunk-1 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
9	C1B1_512B	R/W	1h	privilege access enabled to 2nd 512 byte in chunk-1 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
8	C1B0_512B	R/W	1h	privilege access enabled to 1st 512 byte in chunk-1 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
7	RESERVED	R	0h	Reserved
6	C0B2_4KB	R/W	1h	privilege access enabled to 3rd 4KB in chunk-0 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
5	C0B1_4KB	R/W	1h	privilege access enabled to 2nd 4KB in chunk-0 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
4	C0B0_4KB	R/W	1h	privilege access enabled to 1st 4KB in chunk-0 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher

Table 11-53. SPC_PRIVATTRIB1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	C0B0_2KB	R/W	1h	privilege access enabled to 2KB in chunk-0 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
2	C0B0_1KB	R/W	1h	privilege access enabled to 1KB in chunk-0 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
1	C0B1_512B	R/W	1h	privilege access enabled to 2nd 512 byte in chunk-0 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
0	C0B0_512B	R/W	1h	privilege access enabled to 1st 512 byte in chunk-0 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher

11.3.49 SPC_PRIVATTRIB2 Register (Offset = 12C8h) [Reset = 7F7F7F7Fh]

SPC_PRIVATTRIB2 is shown in Table 11-54.

Return to the Summary Table.

SRAM Privilege Attribute-2 Register

Table 11-54. SPC_PRIVATTRIB2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	Reserved
30	C3B2_4KB	R/W	1h	privilege access enabled to 3rd 4KB in chunk-3 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
29	C3B1_4KB	R/W	1h	privilege access enabled to 2nd 4KB in chunk-3 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
28	C3B0_4KB	R/W	1h	privilege access enabled to 1st 4KB in chunk-3 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
27	C3B0_2KB	R/W	1h	privilege access enabled to 2KB in chunk-3 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
26	C3B0_1KB	R/W	1h	privilege access enabled to 1KB in chunk-3 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
25	C3B1_512B	R/W	1h	privilege access enabled to 2nd 512 byte in chunk-3 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
24	C3B0_512B	R/W	1h	privilege access enabled to 1st 512 byte in chunk-3 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
23	RESERVED	R	0h	Reserved
22	C2B2_4KB	R/W	1h	privilege access enabled to 3rd 4KB in chunk-2 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
21	C2B1_4KB	R/W	1h	privilege access enabled to 2nd 4KB in chunk-2 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
20	C2B0_4KB	R/W	1h	privilege access enabled to 1st 4KB in chunk-2 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
19	C2B0_2KB	R/W	1h	privilege access enabled to 2KB in chunk-2 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher

Table 11-54. SPC_PRIVATTRIB2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	C2B0_1KB	R/W	1h	privilege access enabled to 1KB in chunk-2 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
17	C2B1_512B	R/W	1h	privilege access enabled to 2nd 512 byte in chunk-2 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
16	C2B0_512B	R/W	1h	privilege access enabled to 1st 512 byte in chunk-2 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
15	RESERVED	R	0h	Reserved
14	C1B2_4KB	R/W	1h	privilege access enabled to 3rd 4KB in chunk-1 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
13	C1B1_4KB	R/W	1h	privilege access enabled to 2nd 4KB in chunk-1 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
12	C1B0_4KB	R/W	1h	privilege access enabled to 1st 4KB in chunk-1 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
11	C1B0_2KB	R/W	1h	privilege access enabled to 2KB in chunk-1 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
10	C1B0_1KB	R/W	1h	privilege access enabled to 1KB in chunk-1 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
9	C1B1_512B	R/W	1h	privilege access enabled to 2nd 512 byte in chunk-1 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
8	C1B0_512B	R/W	1h	privilege access enabled to 1st 512 byte in chunk-1 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
7	RESERVED	R	0h	Reserved
6	C0B2_4KB	R/W	1h	privilege access enabled to 3rd 4KB in chunk-0 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
5	C0B1_4KB	R/W	1h	privilege access enabled to 2nd 4KB in chunk-0 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
4	C0B0_4KB	R/W	1h	privilege access enabled to 1st 4KB in chunk-0 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher

Table 11-54. SPC_PRIVATTRIB2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	C0B0_2KB	R/W	1h	privilege access enabled to 2KB in chunk-0 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
2	C0B0_1KB	R/W	1h	privilege access enabled to 1KB in chunk-0 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
1	C0B1_512B	R/W	1h	privilege access enabled to 2nd 512 byte in chunk-0 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
0	C0B0_512B	R/W	1h	privilege access enabled to 1st 512 byte in chunk-0 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher

11.3.50 SPC_PRIVATTRIB3 Register (Offset = 12CCh) [Reset = 7F7F7F7Fh]

SPC_PRIVATTRIB3 is shown in Table 11-55.

Return to the Summary Table.

SRAM Privilege Attribute-3 Register

Table 11-55. SPC_PRIVATTRIB3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	Reserved
30	C3B2_4KB	R/W	1h	privilege access enabled to 3rd 4KB in chunk-3 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
29	C3B1_4KB	R/W	1h	privilege access enabled to 2nd 4KB in chunk-3 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
28	C3B0_4KB	R/W	1h	privilege access enabled to 1st 4KB in chunk-3 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
27	C3B0_2KB	R/W	1h	privilege access enabled to 2KB in chunk-3 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
26	C3B0_1KB	R/W	1h	privilege access enabled to 1KB in chunk-3 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
25	C3B1_512B	R/W	1h	privilege access enabled to 2nd 512 byte in chunk-3 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
24	C3B0_512B	R/W	1h	privilege access enabled to 1st 512 byte in chunk-3 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
23	RESERVED	R	0h	Reserved
22	C2B2_4KB	R/W	1h	privilege access enabled to 3rd 4KB in chunk-2 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
21	C2B1_4KB	R/W	1h	privilege access enabled to 2nd 4KB in chunk-2 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
20	C2B0_4KB	R/W	1h	privilege access enabled to 1st 4KB in chunk-2 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
19	C2B0_2KB	R/W	1h	privilege access enabled to 2KB in chunk-2 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher

Table 11-55. SPC_PRIVATTRIB3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	C2B0_1KB	R/W	1h	privilege access enabled to 1KB in chunk-2 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
17	C2B1_512B	R/W	1h	privilege access enabled to 2nd 512 byte in chunk-2 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
16	C2B0_512B	R/W	1h	privilege access enabled to 1st 512 byte in chunk-2 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
15	RESERVED	R	0h	Reserved
14	C1B2_4KB	R/W	1h	privilege access enabled to 3rd 4KB in chunk-1 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
13	C1B1_4KB	R/W	1h	privilege access enabled to 2nd 4KB in chunk-1 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
12	C1B0_4KB	R/W	1h	privilege access enabled to 1st 4KB in chunk-1 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
11	C1B0_2KB	R/W	1h	privilege access enabled to 2KB in chunk-1 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
10	C1B0_1KB	R/W	1h	privilege access enabled to 1KB in chunk-1 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
9	C1B1_512B	R/W	1h	privilege access enabled to 2nd 512 byte in chunk-1 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
8	C1B0_512B	R/W	1h	privilege access enabled to 1st 512 byte in chunk-1 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
7	RESERVED	R	0h	Reserved
6	C0B2_4KB	R/W	1h	privilege access enabled to 3rd 4KB in chunk-0 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
5	C0B1_4KB	R/W	1h	privilege access enabled to 2nd 4KB in chunk-0 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
4	C0B0_4KB	R/W	1h	privilege access enabled to 1st 4KB in chunk-0 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher

Table 11-55. SPC_PRIVATTRIB3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	C0B0_2KB	R/W	1h	privilege access enabled to 2KB in chunk-0 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
2	C0B0_1KB	R/W	1h	privilege access enabled to 1KB in chunk-0 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
1	C0B1_512B	R/W	1h	privilege access enabled to 2nd 512 byte in chunk-0 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
0	C0B0_512B	R/W	1h	privilege access enabled to 1st 512 byte in chunk-0 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher

11.3.51 FPC_SECATTRIBA0 Register (Offset = 1300h) [Reset = FFFFFFFFh]

FPC_SECATTRIBA0 is shown in Table 11-56.

Return to the Summary Table.

Flash Secure Attribute Register A0

Table 11-56. FPC_SECATTRIBA0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SEC_PAGE	R/W	FFFFFFFFh	Secure access enable for first 32 sectors of flash with each bit representing 1 sector of 2KB 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher

11.3.52 FPC_SECATTRIBA1 Register (Offset = 1304h) [Reset = FFFFFFFFh]

FPC_SECATTRIBA1 is shown in Table 11-57.

Return to the Summary Table.

Flash Secure Attribute Register A1

Table 11-57. FPC_SECATTRIBA1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SEC_PAGE	R/W	FFFFFFFFh	Secure access enable for first 32 sectors of flash with each bit representing 1 sector of 2KB 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher

11.3.53 FPC_SECATTRIBB0 Register (Offset = 1308h) [Reset = FFFFFFF0h]

FPC_SECATTRIBB0 is shown in Table 11-58.

Return to the Summary Table.

Flash Secure Attribute Register B0

Table 11-58. FPC_SECATTRIBB0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	SEC_PAGE	R/W	0FFFFFFFh	Secure access enable for 64KB-512KB flash with each bit representing 8 sectors of 2KB 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
3-0	RESERVED	R	0h	Reserved

11.3.54 FPC_SECATTRIBB1 Register (Offset = 130Ch) [Reset = FFFFFFF0h]

FPC_SECATTRIBB1 is shown in Table 11-59.

Return to the Summary Table.

Flash Secure Attribute Register B1

Table 11-59. FPC_SECATTRIBB1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	SEC_PAGE	R/W	0FFFFFFFh	Secure access enable for 64KB-512KB flash with each bit representing 8 sectors of 2KB 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher
3-0	RESERVED	R	0h	Reserved

11.3.55 FPC_SECATTRIBA_DFLASH Register (Offset = 1380h) [Reset = 0000FFFFh]

FPC_SECATTRIBA_DFLASH is shown in Table 11-60.

Return to the Summary Table.

Data Flash Secure Attribute Register A

Table 11-60. FPC_SECATTRIBA_DFLASH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-0	SEC_PAGE_DFLASH	R/W	FFFFh	Secure access enable for first 16 sectors data flash page with each bit representing 1 sector of 2KB 0: Non-secure access is enabled 1: Secure access is enabled Reset type: BOOTRST or Higher

11.3.56 FPC_PRIVATTRIBA0 Register (Offset = 1400h) [Reset = FFFFFFFFh]

FPC_PRIVATTRIBA0 is shown in Table 11-61.

Return to the Summary Table.

Flash Privilege Attribute Register A0

Table 11-61. FPC_PRIVATTRIBA0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PRIV_PAGE	R/W	FFFFFFFFh	Privilege access enable for first 32 sectors of flash with each bit representing 1 sector of 2KB 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: BOOTRST or Higher

11.3.57 FPC_PRIVATTRIBA1 Register (Offset = 1404h) [Reset = FFFFFFFFh]

FPC_PRIVATTRIBA1 is shown in Table 11-62.

Return to the Summary Table.

Flash Privilege Attribute Register A1

Table 11-62. FPC_PRIVATTRIBA1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PRIV_PAGE	R/W	FFFFFFFFh	Privilege access enable for first 32 sectors of flash with each bit representing 1 sector of 2KB 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: BOOTRST or Higher

11.3.58 FPC_PRIVATTRIBB0 Register (Offset = 1408h) [Reset = FFFFFFF0h]

FPC_PRIVATTRIBB0 is shown in Table 11-63.

Return to the Summary Table.

Flash Privilege Attribute Register B0

Table 11-63. FPC_PRIVATTRIBB0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	PRIV_PAGE	R/W	0FFFFFFFh	Privilege access enable for 64KB-512KB flash with each bit representing 8 sectors of 2KB 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
3-0	RESERVED	R	0h	Reserved

11.3.59 FPC_PRIVATTRIBB1 Register (Offset = 140Ch) [Reset = FFFFFFF0h]

FPC_PRIVATTRIBB1 is shown in Table 11-64.

Return to the Summary Table.

Flash Privilege Attribute Register B1

Table 11-64. FPC_PRIVATTRIBB1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	PRIV_PAGE	R/W	0FFFFFFFh	Privilege access enable for 64KB-512KB flash with each bit representing 8 sectors of 2KB 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: BOOTRST or Higher
3-0	RESERVED	R	0h	Reserved

11.3.60 FPC_PRIVATTRIBA_DFLASH Register (Offset = 1480h) [Reset = 0000FFFFh]

FPC_PRIVATTRIBA_DFLASH is shown in Table 11-65.

Return to the Summary Table.

Data Flash Privilege Attribute Register

Table 11-65. FPC_PRIVATTRIBA_DFLASH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-0	PRIV_PAGE_DFLASH	R/W	FFFFh	Privilege access enable for first 16 sectors data flash page with each bit representing 1 sector of 2KB 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: BOOTRST or Higher

11.3.61 FPC_WEPROTA0 Register (Offset = 1600h) [Reset = 00000000h]

FPC_WEPROTA0 is shown in Table 11-66.

Return to the Summary Table.

Flash Write Protect Attribute Register A0

Table 11-66. FPC_WEPROTA0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	WEPROT_PAGE	R/W1S	0h	Write protection enable for first 32 sectors of flash with each bit representing 1 sector of 2KB 0: Page is not erase/program protected 1: Page is erase/program protected Reset type: BOOTRST or Higher

11.3.62 FPC_WEPROTA1 Register (Offset = 1604h) [Reset = 00000000h]

FPC_WEPROTA1 is shown in Table 11-67.

Return to the Summary Table.

Flash Write Protect Attribute Register A1

Table 11-67. FPC_WEPROTA1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	WEPROT_PAGE	R/W1S	0h	Write protection enable for first 32 sectors of flash with each bit representing 1 sector of 2KB 0: Page is not erase/program protected 1: Page is erase/program protected Reset type: BOOTRST or Higher

11.3.63 FPC_WEPROTB0 Register (Offset = 1608h) [Reset = 00000000h]

FPC_WEPROTB0 is shown in Table 11-68.

Return to the Summary Table.

Flash Write Protect Attribute Register B0

Table 11-68. FPC_WEPROTB0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	WEPROT_PAGE	R/W1S	0h	Write Protection enable for 512KB-1024KB flash with each bit representing 8 sectors of 2KB 0: Page is not erase/program protected 1: Page is erase/program protected Reset type: BOOTRST or Higher

11.3.64 FPC_WEPROTB1 Register (Offset = 160Ch) [Reset = 00000000h]

FPC_WEPROTB1 is shown in Table 11-69.

Return to the Summary Table.

Flash Write Protect Attribute Register B1

Table 11-69. FPC_WEPROTB1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	WEPROT_PAGE	R/W1S	0h	Write Protection enable for 512KB-1024KB flash with each bit representing 8 sectors of 2KB 0: Page is not erase/program protected 1: Page is erase/program protected Reset type: BOOTRST or Higher

11.3.65 FPC_WEPROTA_DFLASH Register (Offset = 1700h) [Reset = 00000000h]

FPC_WEPROTA_DFLASH is shown in Table 11-70.

Return to the Summary Table.

Data Flash Write Protect Attribute Register A

Table 11-70. FPC_WEPROTA_DFLASH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	WEPROT_PAGE_DFLASH	R/W1S	0h	No functionality implemented for these bits Reset type: BOOTRST or Higher
15-0	WEPROT_PAGE_DFLASH	R/W1S	0h	Write protection enable for first 16 sectors data flash page with each bit representing 1 sector of 2KB 0: Page is not erase/program protected 1: Page is erase/program protected Reset type: BOOTRST or Higher

11.3.66 FPC_FLSEMREQ Register (Offset = 1800h) [Reset = 00000000h]

FPC_FLSEMREQ is shown in Table 11-71.

Return to the Summary Table.

Flash semaphore request register

Table 11-71. FPC_FLSEMREQ Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	REQ	R-0/W1S	0h	If the FLSEMSTAT.ASSIGNED bit is cleared, writing a '1' to this bit causes: 1) Load the SECURE attribute of the code performing the write into the FLSEMSTAT.SEC bit field, and 2) Load the Privilage attribute of the code performing the write into the FLSEMSTAT.PRIV bit field, and 3) Set the FLSEMSTAT.ASSIGNED bit. If the FLSEMSTAT.ASSIGNED is already set when a write to this bit occurs, the write will be ignored. If the above conditions are not met during the write, the write will be ignored with no error indicator. It is advised that the writing code perform a read of the FLSEMSTAT register to ensure it was set after writing to this bit. Reset type: BOOTRST or Higher

11.3.67 FPC_FLSEMCLR Register (Offset = 1804h) [Reset = 00000000h]

FPC_FLSEMCLR is shown in Table 11-72.

Return to the Summary Table.

Flash semaphore release register

Table 11-72. FPC_FLSEMCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	CLR	R-0/W1S	0h	If the following conditions are met, a write of '1' to this bit causes the FLSEMSTAT register to its reset state: 1) The FLSEMSTAT.SEC of the code performing the write is 0x1 and FLSEMSTAT.PRIV of the code performing the write is 0x1 (i.e. secure privilage code can force relinquishing of the semaphore) or 2) FLSEMSTAT.ASSIGNED bit is set, and a) FLSEMSTAT.SEC matches security attribute of the code performing the write, and b) FLSEMSTAT.PRIV matches privilege attribute code performing the write. If the above conditions are not met during the write, the write will be ignored with no error indicated. It is advised that the writing code perform a read of the FLSEMSTAT register to ensure it was cleared after writing to this bit. Reset type: BOOTRST or Higher

11.3.68 FPC_FLSEMSTAT Register (Offset = 1808h) [Reset = 00000000h]

FPC_FLSEMSTAT is shown in Table 11-73.

Return to the Summary Table.

Flash semaphore statys registers

Table 11-73. FPC_FLSEMSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31	ASSIGNED	R	0h	SEC and PRIV fields are valid when this bit is set. 0 : SEC and PRIV are unassigned and flash controller is not accessible 1 : SEC and PRIV values are valid and the flash controller is currently assigned to them Reset type: BOOTRST or Higher
30	MATCH	R	0h	On a read, this bit will reflect whether the reader's SEC and PRIV matches the ownership of the flash controller. 0 : Code performing read does not own the flash controller semaphore 1 : Code performing read owns the flash controller semphore. This avoids code from being required to know whether the current code segment owns the flash controller configuration. Reset type: BOOTRST or Higher
29-17	RESERVED	R	0h	Reserved
16	DBGACC	R	0h	Defines the flash controller semaphore owners debug access indication 0x0 : functional access 0x1 :Debug access Reset type: BOOTRST or Higher
15-9	RESERVED	R	0h	Reserved
8	PRIV	R	0h	Defines the flash controller semaphore owners privilage attribute 0x0 : non-privilage 0x1 :prvilage Reset type: BOOTRST or Higher
7-1	RESERVED	R	0h	Reserved
0	SEC	R	0h	Defines the flash controller semaphore owners security attribute 0x0 : non-secure 0x1 :secure Reset type: BOOTRST or Higher

11.3.69 DMA_TRIG_SEL Register (Offset = 1A80h) [Reset = 00000000h]

DMA_TRIG_SEL is shown in Table 11-74.

Return to the Summary Table.

DMA Trigger Selection register

Table 11-74. DMA_TRIG_SEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	Reserved
27-0	SEL_TRIG	R/W	0h	Select signal for DMA triggers to be routed to DMA0/DMA1 0: DMA0 will be selected 1: DMA1 will be selected Note: SEL_TRIG[9:6] is not used. If a specific DMA is selected, the triggers should only go to that DMA Reset type: BOOTRST or Higher

11.3.70 DMA_FIX_TRIG_SEL Register (Offset = 1A84h) [Reset = 00000000h]

DMA_FIX_TRIG_SEL is shown in Table 11-75.

Return to the Summary Table.

DMA Fixed Trigger Selection register

Table 11-75. DMA_FIX_TRIG_SEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R	0h	Reserved
10-0	SEL_TRIG	R/W	0h	Select signal for Fixed DMA triggers to be routed to DMA0/DMA1 0: DMA0 will be selected 1: DMA1 will be selected If a specific DMA is selected, the triggers should only go to that DMA Reset type: BOOTRST or Higher

11.3.71 VTOR_S Register (Offset = 1B80h) [Reset = 11000001h]

VTOR_S is shown in Table 11-76.

Return to the Summary Table.

Secure Vector Table Offset Register

Table 11-76. VTOR_S Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	VTOR_S	R/W	11000001h	value of INITSVTOR port at CPUSS boundary Reset type: BSL Reset or Higher

11.3.72 VTOR_NS Register (Offset = 1B84h) [Reset = 01000001h]

VTOR_NS is shown in Table 11-77.

Return to the Summary Table.

Non - Secure Vector Table Offset Register

Table 11-77. VTOR_NS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	VTOR_NS	R/W	01000001h	value of INITNSVTOR port at CPUSS boundary Reset type: BSL Reset or Higher

11.3.73 DICE_CHECKSUM0 Register (Offset = 1D00h) [Reset = 00000000h]

DICE_CHECKSUM0 is shown in Table 11-78.

Return to the Summary Table.

DICE checksum data register-0

Table 11-78. DICE_CHECKSUM0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CID	WOnce	0h	The register stores the 32-byte DICE Composite Device Identifier (CDI) value Reset type: BOOTRST or Higher

11.3.74 DICE_CHECKSUM1 Register (Offset = 1D04h) [Reset = 00000000h]

DICE_CHECKSUM1 is shown in Table 11-79.

Return to the Summary Table.

DICE checksum data register-1

Table 11-79. DICE_CHECKSUM1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CID	WOnce	0h	The register stores the 32-byte DICE Composite Device Identifier (CDI) value Reset type: BOOTRST or Higher

11.3.75 DICE_CHECKSUM2 Register (Offset = 1D08h) [Reset = 00000000h]

DICE_CHECKSUM2 is shown in Table 11-80.

Return to the Summary Table.

DICE checksum data register-2

Table 11-80. DICE_CHECKSUM2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CID	WOnce	0h	The register stores the 32-byte DICE Composite Device Identifier (CDI) value Reset type: BOOTRST or Higher

11.3.76 DICE_CHECKSUM3 Register (Offset = 1D0Ch) [Reset = 00000000h]

DICE_CHECKSUM3 is shown in Table 11-81.

Return to the Summary Table.

DICE checksum data register-3

Table 11-81. DICE_CHECKSUM3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CID	WOnce	0h	The register stores the 32-byte DICE Composite Device Identifier (CDI) value Reset type: BOOTRST or Higher

11.3.77 DICE_CHECKSUM4 Register (Offset = 1D10h) [Reset = 00000000h]

DICE_CHECKSUM4 is shown in Table 11-82.

Return to the Summary Table.

DICE checksum data register-4

Table 11-82. DICE_CHECKSUM4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CID	WOnce	0h	The register stores the 32-byte DICE Composite Device Identifier (CDI) value Reset type: BOOTRST or Higher

11.3.78 DICE_CHECKSUM5 Register (Offset = 1D14h) [Reset = 00000000h]

DICE_CHECKSUM5 is shown in Table 11-83.

Return to the Summary Table.

DICE checksum data register-5

Table 11-83. DICE_CHECKSUM5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CID	WOnce	0h	The register stores the 32-byte DICE Composite Device Identifier (CDI) value Reset type: BOOTRST or Higher

11.3.79 DICE_CHECKSUM6 Register (Offset = 1D18h) [Reset = 00000000h]

DICE_CHECKSUM6 is shown in Table 11-84.

Return to the Summary Table.

DICE checksum data register-6

Table 11-84. DICE_CHECKSUM6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CID	WOnce	0h	The register stores the 32-byte DICE Composite Device Identifier (CDI) value Reset type: BOOTRST or Higher

11.3.80 DICE_CHECKSUM7 Register (Offset = 1D1Ch) [Reset = 00000000h]

DICE_CHECKSUM7 is shown in Table 11-85.

Return to the Summary Table.

DICE checksum data register-7

Table 11-85. DICE_CHECKSUM7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CID	WOnce	0h	The register stores the 32-byte DICE Composite Device Identifier (CDI) value Reset type: BOOTRST or Higher

11.3.81 GSC_LOCK Register (Offset = 1D80h) [Reset = 00000000h]

GSC_LOCK is shown in Table 11-86.

Return to the Summary Table.

GSC Lock configuration register

Table 11-86. GSC_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	KEY	R-0/W	0h	Write 0xA551 to update the register Reset type: BOOTRST or Higher
15	RESERVED	R	0h	
14-7	RESERVED	R-0	0h	Reserved
6	DMA_TRIG_SEL	R/W	0h	Configuration impacts registers DMA_TRIG_SEL* 0: Register configuration is not locked. 1: Register configuration is locked. Once register is locked - reads give the MMR data and writes are ignored Reset type: BOOTRST or Higher
5	VTOR	R/W	0h	Configuration impacts registers VTOR_* 0: Register configuration is not locked. 1: Register configuration is locked. Once register is locked - reads give the MMR data and writes are ignored Reset type: SYSRST or Higher
4	RESERVED	R-0	0h	Reserved
3	FPC_AATRIB	R/W	0h	Configuration impacts registers FPC_SECATTRIB_* and FPC_PRIVATTRIB_* 0: Register configuration is not locked. 1: Register configuration is locked. Once register is locked - reads give the MMR data and writes are ignored Reset type: BOOTRST or Higher
2	SPC_AATRIB	R/W	0h	Configuration impacts registers SPC_SECATTRIB_* and SPC_PRIVATTRIB_* 0: Register configuration is not locked. 1: Register configuration is locked. Once register is locked - reads give the MMR data and writes are ignored Reset type: BOOTRST or Higher
1	PPC_AATRIB	R/W	0h	Configuration impacts registers PPC_SECATTRIB_* and PPC_PRIVATTRIB_* 0: Register configuration is not locked. 1: Register configuration is locked. Once register is locked - reads give the MMR data and writes are ignored Reset type: BOOTRST or Higher
0	ATTRIBVIOL_CONFIG	R/W	0h	Configuration impacts registers SPC_ATTRIBVIOLS_CONFIG, SPC_ATTRIBVIOLP_CONFIG, PPC_ATTRIBVIOLS_CONFIG, PPC_ATTRIBVIOLP_CONFIG, FPC_ATTRIBVIOLS_CONFIG, FPC_ATTRIBVIOLP_CONFIG 0: Register configuration is not locked. 1: Register configuration is locked. Once register is locked - reads give the MMR data and writes are ignored Reset type: BOOTRST or Higher

11.3.82 GSC_COMMIT Register (Offset = 1D84h) [Reset = 00000000h]

GSC_COMMIT is shown in Table 11-87.

Return to the Summary Table.

GSC commit configuration register

Table 11-87. GSC_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	KEY	R-0/W	0h	Write 0xA442 to update the register Reset type: BOOTRST or Higher
15	RESERVED	R	0h	
14-7	RESERVED	R-0	0h	Reserved
6	DMA_TRIG_SEL	R/W1S	0h	0: Register lock configuration is not committed. 1: Register configuration is committed. Commit configuration blocks the writes going to the lock register. Once configuration is committed, only reset can change the configuration. Reset type: BOOTRST or Higher
5	VTOR	R/W1S	0h	0: Register lock configuration is not committed. 1: Register configuration is committed. Commit configuration blocks the writes going to the lock register. Once configuration is committed, only reset can change the configuration. Reset type: SYSRST or Higher
4	RESERVED	R-0	0h	Reserved
3	FPC_AATRIB	R/W1S	0h	0: Register lock configuration is not committed. 1: Register configuration is committed. Commit configuration blocks the writes going to the lock register. Once configuration is committed, only reset can change the configuration. Reset type: BOOTRST or Higher
2	SPC_AATRIB	R/W1S	0h	0: Register lock configuration is not committed. 1: Register configuration is committed. Commit configuration blocks the writes going to the lock register. Once configuration is committed, only reset can change the configuration. Reset type: BOOTRST or Higher
1	PPC_AATRIB	R/W1S	0h	0: Register lock configuration is not committed. 1: Register configuration is committed. Commit configuration blocks the writes going to the lock register. Once configuration is committed, only reset can change the configuration. Reset type: BOOTRST or Higher
0	ATTRIBVIOL_CONFIG	R/W1S	0h	0: Register lock configuration is not committed. 1: Register configuration is committed. Commit configuration blocks the writes going to the lock register. Once configuration is committed, only reset can change the configuration. Reset type: BOOTRST or Higher

11.3.83 GSC_REVISION Register (Offset = 1FFCh) [Reset = 00000000h]

GSC_REVISION is shown in Table 11-88.

Return to the Summary Table.

GSC Revision register

Table 11-88. GSC_REVISION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-8	MAJREV	R	0h	This hardcoded field defines the major revision of the IP. Reset type: BOOTRST or Higher
7-0	MINREV	R	0h	This hardcoded field defines the minor revision of the IP. Reset type: BOOTRST or Higher

Chapter 12

PKA

The Public Key Accelerator (PKA) module supports mathematical operations needed for elliptic curves (ECC) and RSA asymetric cryptography.

12.1 PKA Introduction 660
12.2 PKA Operation 660
12.3 PKA Registers 666

12.1 PKA Introduction

The PKA is an integrated module for public key acceleration to offload the computation of intensive public cryptography operations.

12.1.1 PKA features

- The PKA engine provides the following basic operations
  - Large vector addition, subtraction and combined addition and subtraction
  - Large vector shift left or right
  - Large vector multiplication and division (with or without quotient)
  - Large vector compare and copy
- The PKA engine provides the following complex operations:
  - Large vector unsigned value modular exponentiation
  - Large vector unsigned value modular exponentiation using the CRT method with pre-calculated Q inverse vector
  - Modular inversion
  - ECC operations on two type of curve: Montgomery curves like Curve25519 and Curve448, and any curve of the form $y^2=x^3+ax+b$ (mod p).
  - ECC point addition/doubling on elliptic curve with affine or projective points as input/output
  - ECC point multiplication on elliptic curve
- Illegal state and timing attack detection

12.2 PKA Operation

12.2.1 Functional Block Diagram

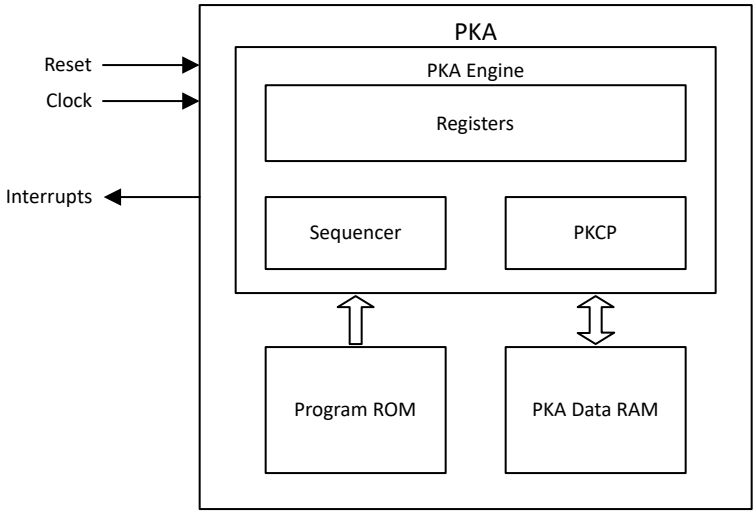


Figure 12-1. PKA Functional Block Diagram

12.2.2 Theory of Operations

The PKA engine consists of a Public Key Co-Processor (PKCP) that performs all the large number mathematical operations. The PKCP is controlled by the sequencer. The sequencer operates from a configuration ROM memory which provides the necessary set of operations for the PKCP.

12.2.2.1 PKCP

The PKCP engine is designed to perform basic mathematical operations for big number vectors stored in the PKA data RAM. The PKA data RAM uses the little endian format for all operations.

Table 12-1. PKCP Basic Commands

Command	Mathematical Operation	Vector A	Vector B	Vector C	Vector D
Multiply	A x B = C	Multiplicand	Multiplier	Product	-
Add	A + B = C	Addend	Addend	Sum	-
Subtract	A - B = C	Minuend	Subtrahend	Difference	-
AddSub	A + C - B = D	Addend	Subtrahend	Addend	Result
Right Shift	A >> Shift = C	Input	-	Result	-
Left Shift	A << Shift = C	Input	-	Result	-
Divide	A mod B = C A div B = D	Dividend	Divisor	Remainder	Quotient
Modulo	A mod B = C	Dividend	Divisor	Remainder	-
Compare	A = B A < B A > B	Input1	Input2	-	-
Copy	A = C	Input	-	Result	-

To obtain correct results, the input vectors must meet the requirements as shown in Table 12-2. Input restrictions are not checked by the PKCP, unless otherwise noted, and the host application must check the same.

Table 12-2. PKCP Operation Restrictions for Input Vectors

Command	Requirements
Multiply	0 < ALen, BLen ≤ Max_Len
Add	0 < ALen, BLen ≤ Max_Len
Subtract	0 < ALen, BLen ≤ Max_Len Result must be positive (A ≥ B)
AddSub	0 < ALen ≤ Max_Len (All operands have ALen as length) Result must be positive ((A + C) ≥ B)
Right Shift	0 < ALen ≤ Max_Len
Left Shift	0 < ALen ≤ Max_Len
Divide, Modulo	1 < BLen ≤ ALen ≤ Max_Len Most signifcant 32-bit word of B operand cannot be zero (this is checked)
Compare	0 < ALen ≤ Max_Len (All operands have ALen as length)
Copy	0 < ALen ≤ Max_Len

Note

Alen and Blen indicate the size of vectors A and B in 32-bit words. Max_Len equals 130 words.

The application is responsible for allocating a block of contiguous memory in the PKA Data RAM for the result vector as shown in Table 12-3.

Table 12-3. PKCP Result Vector Memory Allocation

Command	Result Vector	Result Vector Length (32-bit words)
Multiply	C	ALen + BLen + 3 words
Add	C	Max(ALen, BLen) + 1. The extra word contains the carry-out of the addition
Subtract	C	Max(ALen, BLen)

Table 12-3. PKCP Result Vector Memory Allocation (continued)

Command	Result Vector	Result Vector Length (32-bit words)
AddSub	D	ALen + 1
Right Shift	C	ALen
Left Shift	C	ALen + 1 (when shift value is non-zero) ALen (when shift value is zero)
Divide	C	Remainder = BLen + 1
	D	Quotient = ALen - BLen + 1
Modulo	C	Remainder = BLen + 1
Compare	None	Compare updates PKA_COMPARE register
Copy	C	ALen

Input vectors for an operation are always allowed to overlap in memory. gives restrictions for the overal of output and input vectors of the operation.

Table 12-4. PKCP Result / Input Vector Overlap Restrictions

Command	Result Vector	Restrictions
Multiply	C	No overlap with A or B vectors allowed
Add Subtract	C	May overlap with A and/or B vector, provided the start address of the C vector does not lie above the start address of the vector(s) with which it overlaps
AddSub	D	May overlap with A, B and/or C vector, provided the start address of the D vector does not lie above the start address of the vector(s) with which it overlaps
Right Shift Left Shift	C	May overlap with A vector, provided the start address of the C vector does not lie above the start address of the A vector
Divide	C	No overlap with A, B or D vectors allowed
	D	No overlap with A, B or D vectors allowed
Modulo	C	No overlap with A or B vectors allowed
Compare	None	Compare does not write a result vector
Copy	C	Same restrictions as for Right/Left Shift, copy of a vector to a lower address is always allowed even if source and destination overlap

12.2.2.2 Sequencer

Basic and complex commands are started by writing to the PKA function register, by setting the "RUN" bit in combination with one or more other bits. Table 12-5 and Table 12-6 show the supported basic and complex commands that are available with the PKA engine.

Table 12-5. Basic commands

Bits [11:4, 2:0]	Operation	Function
0000 0000 001	Multiply	Perform a multiply operation
0000 0000 010	AddSub	Perform a combined Add/Subtract operation
0000 0001 000	Add	Perform an add operation
0000 0010 000	Subtract	Perform a subtract operation
0000 0100 000	Rshift	Perform a right shift operation
0000 1000 000	Lshift	Perform a left shift operation
0001 0000 000	Divide	Perform a divide operation

Table 12-5. Basic commands (continued)

Bits [11:4, 2:0]	Operation	Function
0010 0000 000	Modulo	Perform a modulo operation
0100 0000 000	Compare	Perform a compare operation
1000 0000 000	Copy	Perform a copy operation
0000 0000 000	None	No operation. Use this value when selecting a complex command
All other values	Reserved	Do not use

Table 12-6. Complex Commands

Bits [18:16, 15:12]	Operation	Function
000 1000	None	No operation. Use this value when selecting a basic command
000 1001	MODEXP-CRT	Modular eponentiation using the CRT methid with selected number of pre-compued odd-powers
000 1010	ECmontMUL	Montgomery Curve25519 or Curve448 point multiply
000 1011	ECpADD	Point Add/Double on a prime field curve using affine input and output
000 1101	ECpMUL	Point Multiply on a prime fiel curve using affine input and output
000 1110	MODEXP	Modular exponentiation with selected number of pre-computed odd-powers
000 1111	MODINV	Modular inverse, using integer arithmetic
001 1000	ECpADDxyz	Point Add/Double on a Prime Field Curve using projective input and output
001 1001	ECpMULxyz	Point Multiply on a Prime Field Curve using affine input and projective output
001 1010	ECpSCALExyz	Replace the projective result (X, Y, Z) with an equivalent result having Z=1, i.e. (X, Y, Z) → X/Z, Y/Z, 1), using Prime Field arithmetic
All other values	Reserved	Do not use

12.2.3 Complex Commands

Complex commands are provided to the PKA easier to use and maximize the performance. Each complex command glues together a sequence of basic commands to create a higher-level operation that helps implement publick key algorithms such as RSA, ECDSA and ECDH.

Table 12-7. Complex Commands Summary

Command	Complex Operation
MODEXP	M^E (mod N) → R
MODEXP-CRT	CRTpq((M mod P)^{Dp} mode P, (M mod Q)^{Dq} mod Q, Qinv) → R, where CRTpq(a, b, Qinv) = ((a - b) x Qinv mode P) x Q + b
MODINV	1 / Z (mod N) → R
ECpADD	Affine point addition/doubling on prime curve $y^2 = x^3 + Ax + B$ (mod P)
ECpADDxyz	Projective point addition/doubling on prime curve $y^2 = x^3 + Ax + B$ (mod P)
ECpMUL	Affine point multiplication on prime curve $y^2 = x^3 + Ax + B$ (mod P)
ECpMULxyz	Affine to projective point multiplication on prime curve $y^2 = x^3 + Ax + B$ (mod P)
ECpSCALExyz	Projective to affine point scaling on prime curve $y^2 = x^3 + Ax + B$ (mod P)
ECmontMUL	Affine point multiplicsation on Montgomery curve $y^2 = x^3 + Ax^2 + B$ (mod P)

12.2.4 Command Execution and Status

From an application perspective, there is not much difference between running a basic command or running a complex command, except for the following:

- Complex commands return a status code.
- Complex command do not support double-buffering. When current command is still executing, the values for the next command cannot be written.
- Complex commands may be aborted.

The status code that appear in the sequencer register are shown in Table 12-8. When bit 0 has the value '1' it indicates that the PKA is ready for the next command. If the bit is '1', some special condition is indicated that does not have to be an error.

Table 12-8. Sequencer Status Code

Status Code	Description
0x01	Command completed successfully
0x03	Modulus value rejected because it is even
0x05	Exponent/Scalar equal to zero
0x07	Modulus too short
0x09	Exponent/Scalar equal to one
0x0B	Number of odd powers out of allowed range of 0 to 16
0x0D	EC point addition or multiplication resulted in point-at-infinity
0x0F	Operation aborted by Host
0x13	Unexpected at-infinity result for intermediate calculation
0x17	No modular inverse exists
0x1B	Point off curve condition for input or output point
0x1F	Bad length value
0x21	Requested command is invalid or not supported
0x23	Invalid argument
0x27	Result error
0x2B	HW signaled an error
0xAA	PKA is busy
0xFF	General Error

12.2.5 Initialization

PKA module operations require the application to perform basic setup.

1. Reset the peripheal using PKA.RSTCTL register
2. Enable the power to PKA module using the PKA.PWREN
3. Enable the zeroization of the PKA Data RAM by setting the zeroization bit in GSC.PPC_PKACTRL.
4. Wait for the PKA.PKA_SEQ_CTRL LOGIC_BUSY bit to become '0'
5. Clear the zeroization control bit in GSC.PPC_PKACTRL.

For more details on running cryptographic operations using the PKA module, refer to the MSPM33 SDK.

12.2.6 Interrupts support

The PKA module has an Advanced Interrupt Conroller (AIC) that can be configured to generate an interrupt when certain operations are complete or error conditions are encountered. The AIC is managed using the following register groups:

- Enable register
- Acknowledge register to clear edge detected interrupts
- Status registers for raw interrupt source and masked sources
- Configuration status register for module version and options

12.2.6.1 Interrupt Sources

The PKA modules provides 3 interrupt sources which can be configured to source a CPU interrupt event.

Table 12-9. PKA CPU Interrupt Event Conditions

BIT	Name	Description
1	PKA_INT	PKCP and sequencer ready interrupt
5	SL_ERROR_INT	Configuration port error interrupt
6	PROT_INT	Protection error interrupt

12.3 PKA Registers

Table 12-10 lists the memory-mapped registers for the PKA registers. All register offset addresses not listed in Table 12-10 should be considered as reserved locations and the register contents should not be modified.

Table 12-10. PKA Registers

Offset	Acronym	Register Name	Section
800h	PWREN	Power enable	Section 12.3.1
804h	RSTCTL	Reset Control	Section 12.3.2
814h	STAT	Status Register	Section 12.3.3
00014000h	PKA_APTR	A operand address offset	Section 12.3.4
00014004h	PKA_BPTR	B operand address offset	Section 12.3.5
00014008h	PKA_CPTR	C operand address offset	Section 12.3.6
0001400Ch	PKA_DPTR	D operand address offset	Section 12.3.7
00014010h	PKA_ALENGTH	Length of A operand	Section 12.3.8
00014014h	PKA_BLENGTH	Length of B operand	Section 12.3.9
00014018h	PKA_SHIFT	Bits to shift	Section 12.3.10
0001401Ch	PKA_FUNCTION	control bits to start basic PKCP as well as complex Sequencer operations	Section 12.3.11
00014020h	PKA_COMPARE	Result of compare	Section 12.3.12
00014024h	PKA_MSW	MS non-zero word address	Section 12.3.13
00014028h	PKA_DIVMSW	MS non-zero word address for remainder (MOD and DIV ops)	Section 12.3.14
00014030h	PKA_ISTA_CTRL	This register provides control and status bits related to the Illegal State and Timing Attack features of the PKCP	Section 12.3.15
000140C4h	PKCP_PRIO_CTRL	PKCP or Sequencer RAM Access Priority Control	Section 12.3.16
000140C8h	PKA_SEQ_CTRL	PKA Squencer control or status	Section 12.3.17
000140D0h	PKA_FW_OPTIONS	FW options register	Section 12.3.18
000140F4h	PKA_OPTIONS	HW configured options	Section 12.3.19
000140F8h	PKA_SW_REV	FW revision numbers, capabilities	Section 12.3.20
000140FCh	PKA_REVISION	PKA HW revision, EIP code	Section 12.3.21
00016000h + formula	PKA_RAM_DATA_y	PKA DATA RAM - 512 Locations - 2KB	Section 12.3.22
00018000h	AIC_POL_CTRL	Input Polarity Control Register	Section 12.3.23
00018004h	AIC_TYPE_CTRL	Input Type Control Register	Section 12.3.24
00018008h	AIC_ENABLE_CTRL	Interrupt Enable Control Register	Section 12.3.25
0001800Ch	AIC_RAW_STAT	Raw Interrupt Pending Status Register	Section 12.3.26
0001800Ch	AIC_ENABLE_SET	Interrupt Enable Register	Section 12.3.27
00018010h	AIC_ENABLED_STAT	Masked Interrupt Pending Status Register	Section 12.3.28
00018010h	AIC_ACK	Interrupt Acknowledge Register	Section 12.3.29
00018014h	AIC_ENABLE_CLR	Interrupt Disable Register	Section 12.3.30
00018018h	AIC_OPTIONS	AIC Configuration Register	Section 12.3.31
0001801Ch	AIC_VERSION	AIC HW revision, EIP code	Section 12.3.32
0001BFF8h	EIP150_OPTIONS	PKP Configuration Register	Section 12.3.33
0001BFFCh	EIP150_REVISION	PKP HW revision, EIP code	Section 12.3.34

Complex bit access types are encoded to fit into small table cells. Table 12-11 shows the codes that are used for access types in this section.

Table 12-11. PKA Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
WK	W K	Write 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.

12.3.1 PWREN Register (Offset = 800h) [Reset = 00000000h]

PWREN is shown in Table 12-12.

Return to the Summary Table.

Register to control the power state

Table 12-12. PWREN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to allow Power State Change 26h = KEY to allow write access to this register
23-1	RESERVED	R/W	0h	
0	ENABLE	R/WK	0h	Enable the power KEY must be set to 26h to write to this bit. 0h = Disable Power 1h = Enable Power

12.3.2 RSTCTL Register (Offset = 804h) [Reset = 00000000h]

RSTCTL is shown in Table 12-13.

Return to the Summary Table.

Register to control reset assertion and de-assertion

Table 12-13. RSTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	Unlock key 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 KEY must be set to B1h to write to this bit. 0h = Writing 0 has no effect 1h = Clear reset sticky bit
0	RESETASSERT	WK	0h	Assert reset to the peripheral KEY must be set to B1h to write to this bit. 0h = Writing 0 has no effect 1h = Assert reset

12.3.3 STAT Register (Offset = 814h) [Reset = 00000000h]

STAT is shown in Table 12-14.

Return to the Summary Table.

peripheral enable and reset status

Table 12-14. 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 0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register 1h = The peripheral was reset since the last bit clear
15-0	RESERVED	R	0h	

12.3.4 PKA_APTR Register (Offset = 00014000h) [Reset = 00000000h]

PKA_APTR is shown in Table 12-15.

Return to the Summary Table.

A operand address offset

Table 12-15. PKA_APTR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R	0h	Reserved
10-0	APTR	R/W	0h	This register specifies the location of Vector A within the PKA Data RAM. Vectors are identified through the location of their least-significant 32-bit word. Note that bit [0] must be zero to ensure that the vector starts at an 8-byte boundary

12.3.5 PKA_BPTR Register (Offset = 00014004h) [Reset = 00000000h]

PKA_BPTR is shown in Table 12-16.

Return to the Summary Table.

B operand address offset

Table 12-16. PKA_BPTR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R	0h	Reserved
10-0	BPTR	R/W	0h	This register specifies the location of Vector B within the PKA Data RAM. Vectors are identified through the location of their least-significant 32-bit word. Note that bit [0] must be zero to ensure that the vector starts at an 8-byte boundary

12.3.6 PKA_CPTR Register (Offset = 00014008h) [Reset = 00000000h]

PKA_CPTR is shown in Table 12-17.

Return to the Summary Table.

C operand address offset

Table 12-17. PKA_CPTR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R	0h	Reserved
10-0	CPTR	R/W	0h	This register specifies the location of Vector C within the PKA Data RAM. Vectors are identified through the location of their least-significant 32-bit word. Note that bit [0] must be zero to ensure that the vector starts at an 8-byte boundary

12.3.7 PKA_DPTR Register (Offset = 0001400Ch) [Reset = 00000000h]

PKA_DPTR is shown in Table 12-18.

Return to the Summary Table.

D operand address offset

Table 12-18. PKA_DPTR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R	0h	Reserved
10-0	DPTR	R/W	0h	This register specifies the location of Vector D within the PKA Data RAM. Vectors are identified through the location of their least-significant 32-bit word. Note that bit [0] must be zero to ensure that the vector starts at an 8-byte boundary

12.3.8 PKA_ALENGTH Register (Offset = 00014010h) [Reset = 00000000h]

PKA_ALENGTH is shown in Table 12-19.

Return to the Summary Table.

Length of A operand

Table 12-19. PKA_ALENGTH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	0h	Reserved
9-0	ALENGTH	R/W	0h	This register specifies the length (in 32-bit words) of Vector A

12.3.9 PKA_BLENGTH Register (Offset = 00014014h) [Reset = 00000000h]

PKA_BLENGTH is shown in Table 12-20.

Return to the Summary Table.

Length of B operand

Table 12-20. PKA_BLENGTH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	0h	Reserved
9-0	PKA_BLENGTH	R/W	0h	This register specifies the length (in 32-bit words) of Vector B

12.3.10 PKA_SHIFT Register (Offset = 00014018h) [Reset = 00000000h]

PKA_SHIFT is shown in Table 12-21.

Return to the Summary Table.

Bits to shift

Table 12-21. PKA_SHIFT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	Reserved
4-0	BITS_TO_SHIFT	R/W	0h	This register specifies the number of bits to shift the input vector in the range 0 to 31 during a Rshift or Lshift operation

12.3.11 PKA_FUNCTION Register (Offset = 0001401Ch) [Reset = 00000000h]

PKA_FUNCTION is shown in Table 12-22.

Return to the Summary Table.

control bits to start basic PKCP as well as complex Sequencer operations

Table 12-22. PKA_FUNCTION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	Reserved
24	STALLRESULT	R/W	0h	When written with a 1b, updating of the PKA_COMPARE, PKA_MSW and PKA_DIVMSW registers, as well as resetting the Run bit is stalled beyond the point that a running operation is finished
23-19	RESERVED	R	0h	Reserved
18-16	SEQOPEXT	R/W	0h	These bits select the complex Sequencer operation to perform. The encoding of these operations is determined by the Sequencer firmware
15	RUNFUNC	R/W	0h	The Host sets this bit to instruct the PKA module to begin processing the basic PKCP or complex Sequencer operation. This bit is reset low automatically when the operation is complete
14-12	SEQOP	R/W	0h	These bits select the complex Sequencer operation to perform. The encoding of these operations is determined by the Sequencer firmware
11	COPY	R/W	0h	Perform copy operation
10	COMPARE	R/W	0h	Perform compare operation
9	MODULO	R/W	0h	Perform modulo operation
8	DIVIDE	R/W	0h	Perform divide operation
7	LSHIFT	R/W	0h	Perform left shift operation
6	RSHIFT	R/W	0h	Perform right shift operation
5	SUBTRACT	R/W	0h	Perform subtract operation
4	ADD	R/W	0h	Perform add operation.
3	MSONEBIT	R/W	0h	Loads the location of the Most Significant one bit within the result word indicated in the PKA_MSW register into bits [4:0] of the PKA_DIVMSW register, can only be used with basic PKCP operations, except for Divide, Modulo and Compare
2	RESERVED	R	0h	Reserved
1	ADDSUB	R/W	0h	Perform combined add, subtract operation
0	MULTIPLY	R/W	0h	Perform multiply operation

12.3.12 PKA_COMPARE Register (Offset = 00014020h) [Reset = 00000001h]

PKA_COMPARE is shown in Table 12-23.

Return to the Summary Table.

This register provides the result of a basic PKCP Compare operation

Table 12-23. PKA_COMPARE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2	AGTB	R	0h	Vector A > Vector B
1	ALTB	R	0h	Vector A < Vector B
0	AEQB	R	1h	Vector A = Vector B

12.3.13 PKA_MSW Register (Offset = 00014024h) [Reset = 00008000h]

PKA_MSW is shown in Table 12-24.

Return to the Summary Table.

Address of the most significant non-zero 32-bit word of the result vector in PKA Data RAM.

Table 12-24. PKA_MSW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	ZERO_RESULT_VECTOR	R	1h	The result vector is all zeroes, ignore the address returned in bits 10 to 0
14-11	RESERVED	R	0h	Reserved
10-0	ADDRESS	R	0h	Address of the most significant non zero 32 bit word of the result vector in PKA Data RAM

12.3.14 PKA_DIVMSW Register (Offset = 00014028h) [Reset = 00008000h]

PKA_DIVMSW is shown in Table 12-25.

Return to the Summary Table.

Address of the most significant non-zero 32-bit word of the Remainder result vector in PKA Data RAM

Table 12-25. PKA_DIVMSW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	ZERO_RESULT_VECTOR	R	1h	The remainder result vector is all zeroes, ignore the address returned in bits 10 to 0
14-11	RESERVED	R	0h	Reserved
10-0	MSW_ADDRESS	R	0h	Address of the most significant nonzero 32 bit word of the Remainder result vector in PKA Data RAM

12.3.15 PKA_ISTA_CTRL Register (Offset = 00014030h) [Reset = 00000000h]

PKA_ISTA_CTRL is shown in Table 12-26.

Return to the Summary Table.

This register provides control and status bits related to the Illegal State and Timing Attack features of the PKCP

Table 12-26. PKA_ISTA_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	FSM_ERROR	R/W	0h	This bit is set to 1b when one of the PKCP control FSMs enters an illegal state
30	PARAM_ERROR	R/W	0h	This bit is set to 1b when an illegal operation code selecting more than 1 operation at a time is written to the PKA_FUNCTION register. It is also set to 1b when the highest 32-bit word of the B vector (divisor) input to a PKCP DIV or MOD operation equals zero MOD operation equals zero
29-2	RESERVED	R	0h	Reserved
1	FORCE_ERROR	R/W	0h	1b = force FSM errors, eg jump to an illegal state, when starting basic PKCP operations, these should activate the fsm_error bit in this register
0	DUMMY_CORR	R/W	0h	1b = enable dummy correction cycles in the basic PKCP DIV and MOD operations. These dummy cycles remove the variability of these operations at the cost of a running time that can be up to around 70 percent longer

12.3.16 PKCP_PRIO_CTRL Register (Offset = 000140C4h) [Reset = 00000000h]

PKCP_PRIO_CTRL is shown in Table 12-27.

Return to the Summary Table.

This register is only present in configurations with ISTA features enabled. Bit [0] of this register controls data RAM access arbitration between the PKCP and Sequencer. Although it is accessible from the Host, control of it should be left to the firmware

Table 12-27. PKCP_PRIO_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RAM_ECC_TEST	R/W	0h	Depending on ram_test_dual, internal registers ram_ecc_test_0 and ram_ecc_test_1 will be set, or only ram_ecc_test_0
15	RAM_TEST_ONCE	R/W	0h	Writing 1b inserts errors only once, on the next write to the PKA Data RAM, all the ram_ecc_test bits will be reset to zero to prevent further error insertions
14-12	RAM_ECC_B1	R/W	0h	Bit number for bit to flip in Bytes selected in field ram_ecc_test_1
11	RAM_TEST_DUAL	R/W	0h	Set this bit to 1b to produce dual errors, prior to setting ram_ecc_test. When this bit is enabled the next write to ram_ecc_test will set internal registers ram_ecc_test_0 and ram_ecc_test_1, When set to 0b, only ram_ecc_test_0 will be set
10-8	RAM_ECC_B0	R/W	0h	Bit number for bit to flip in Bytes selected in field ram_ecc_test_0
7-1	RESERVED	R	0h	Reserved
0	PKCP_RAM_PRIO	R/W	0h	0b = Sequencer has priority for accessing data RAM. 1b = PKCP has priority for accessing data RAM.

12.3.17 PKA_SEQ_CTRL Register (Offset = 000140C8h) [Reset = 00000000h]

PKA_SEQ_CTRL is shown in Table 12-28.

Return to the Summary Table.

PKA Sequencer Control or Status

Table 12-28. PKA_SEQ_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESET_FOR_SEQUENCER	R/W	0h	reset for sequencer
30-29	PROGRAM_PAGE	R/W	0h	This field provides two page select bits for accessing the Sequencer program RAM through the Host interface when that is larger than 2K words. Value 00b selects page 0, the lowest page, value 01b selects page 1, etcetera
28-24	RESERVED	R	0h	Reserved
23	RAM_ECC_CORR	R/W	0h	Overall correctable RAM SECDED EDC error detected status bit.
22	SEQUENCER_UNLOCKED	R/W	0h	This bit is set to 1b when the main Sequencer receives a SW reset during a complex operation, thereby unlocking the Sequencer. The bit is sticky: It can be cleared again by writing a 1b to this location. Writing a 0b has no effect
21	RESERVED	R	0h	Reserved
20	SEQ_ECC_ERROR	R/W	0h	Sequencer uncorrectable RAM SECDED EDC error. This bit reads 1b when an uncorrectable error occurred while the Sequencer reads the PKA Data RAM
19	RAM_ECC_ERROR	R/W	0h	Overall uncorrectable RAM SECDED EDC error status bit,This bit reads 1b when one or more uncorrectable error(s) occurred on the PKA Data RAM
18	ZERO_BUSY	R	0h	When 1b is returned this read only bit indicates that the zeroization logic is busy. During zeroization the PKA cannot be used
17	PARITY_ERROR	R/W	0h	A 1b indicates parity error was detected
16	HAMMING_ERROR	R/W	0h	A 1b indicates a mismatch was detected between bits 28:24 of a Sequencer instruction word and the Hamming protection code calculated from bits 23:0 of that same instruction word. This Hamming code detects at least 2 bit errors in bits 23:0. The state of this bit is OR-ed into the protection_int output of the engine
15-8	SEQUENCER_STATUS	R	0h	These read only bits are used by the Sequencer to communicate status to the outside world. Bit 8 is also used as Sequencer interrupt, with the complement of this bit OR-ed into the Run bit in PKA_FUNCTION
7-0	SW_TRIGGERS	R/W	0h	These bits can be used by software to trigger Sequencer operations

12.3.18 PKA_FW_OPTIONS Register (Offset = 000140D0h) [Reset = 00000000h]

PKA_FW_OPTIONS is shown in Table 12-29.

Return to the Summary Table.

The FW options register is used by the firmware for storing available capabilities and options

Table 12-29. PKA_FW_OPTIONS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	MISCELLANEOUS	R	0h	xxx1 : Reserved for Binary curve support xx1x : Reserved for SCAP functionality extension x1xx : Fixed-Point ECp multiplication is supported 1xxx : Shamir's trick is configured for ECDSApVERIFY and SM2DSApVERIFY
27-18	DIGITAL_SIGNATURE_SUPPORT	R	0h	xxxxxxxxx1 : DSASIGN is supported xxxxxxxx1x : DSAVERIFY is supported xxxxxxx1xx : ECDSApSIGN is supported xxxxxx1xxx : ECDSApVERIFY is supported xxxxx1xxxx : SM2DSApSIGN is supported xxxx1xxxxx : SM2DSApVERIFY is supported xxx1xxxxxx : Reserved for SCAP command xx1xxxxxxx : ditto x1xxxxxxxx : ditto 1xxxxxxxxx : ditto
17-14	MONT_CURVE_SUPPORT	R	0h	xxx1 : ECmontMUL is supported xx1x : Reserved for SCAP command x1xx : Not used, reserved for future use 1xxx : ditto
13-6	NIST_CURVE_SUPPORT	R	0h	xxxxxxx1 : ECpMUL is supported xxxxxx1x : ECpMULxyz is supported xxxxx1xx : Reserved for SCAP command xxxx1xxx : ECpADD is supported xxx1xxxx : ECpADDxyz is supported xx1xxxxx : Not used, reserved for future use x1xxxxxx : ditto 1xxxxxxx : ditto
5-0	RSA_SUPPORT	R	0h	xxxxx1 : ModExp is supported xxxx1x : ModExpCRT is supported xxx1xx : ModInv is supported xx1xxx : Reserved for SCAP command x1xxxx : ditto 1xxxxx : ditto

12.3.19 PKA_OPTIONS Register (Offset = 000140F4h) [Reset = 00002121h]

PKA_OPTIONS is shown in Table 12-30.

Return to the Summary Table.

HW configured options,This register provides the Host with a means to determine the hardware configuration implemented in this Public Key Accelerator – focused on options that have an effect on software interacting with the module.

Table 12-30. PKA_OPTIONS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-14	RESERVED	R	0h	Reserved
13	ZEROIZATION	R	1h	Value 1b indicates that hardware zeroization logic is present
12	MMM3A	R	0h	When an LNME is present, this bit is 1b to indicate the MMM3A operation is implemented, otherwise this bit equals 0b
11	INT_MASKING	R	0h	Value 0b (default) indicates that the main interrupt output (bit [1] of the interrupts output bus) is the direct complement of the Run bit in the PKA_CONTROL register, value 1b indicates that interrupt masking logic is present for this output.
10-8	PROTECTION_OPTION	R	1h	Value 0 indicates that no additional protection features are present. Value 1 indicates that Illegal State and Timing Attack (ISTA) protection features are present. Value 3 indicates that ISTA protection features are present together with the PROT feature (detection of invalid Sequencer opcodes). Other values are reserved
7	PROGRAM_RAM	R	0h	Value 1b indicates Sequencer program storage in RAM, value 0b in ROM
6-5	SEQUENCER_CONFIGURATION	R	1h	Value 1 indicates a standard Sequencer, other values reserved.
4-2	LNME_CONFIGURATION	R	0h	Value 0 indicates NO LNME, value 1 indicates a standard LNME (with a single PE chain), value 2 indicates a dual LNME (with two parallel PE chains), other values reserved
1-0	PKCP_CONFIGURATION	R	1h	Value 1 indicates a PKCP with a 16x16 multiplier, value 2 indicates a PKCP with a 32x32 multiplier, other values reserved

12.3.20 PKA_SW_REV Register (Offset = 000140F8h) [Reset = 00000000h]

PKA_SW_REV is shown in Table 12-31.

Return to the Summary Table.

PKA Firmware Revision and Capabilities Register. This register allows the Host access to the internal firmware revision number of the Public Key Accelerator for software driver matching and diagnostic purposes. This register also contains a field that globally encodes the capabilities of the embedded firmware

Table 12-31. PKA_SW_REV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	FIRMWARE_CAPABILITIES	R	0h	4 bits binary encoding for the functionality implemented in the firmware
27-24	MAJOR_FW_REVISION	R	0h	4 bits binary encoding of the major firmware revision number
23-20	MINOR_FW_REVISION	R	0h	4 bits binary encoding of the minor firmware revision number
19-16	FW_PATCH_LEVEL	R	0h	4 bits binary encoding of the firmware patch level, initial release will carry value zero. Patches are used to remove bugs without changing the functionality or interface of a module
15-8	RESERVED	R	0h	Write 1 to indicate data is available
7-0	RESERVED	R	0h	Input Data on which hash is computed

12.3.21 PKA_REVISION Register (Offset = 000140FCh) [Reset = 0333E31Ch]

PKA_REVISION is shown in Table 12-32.

Return to the Summary Table.

This register allows the Host access to the hardware revision number of the Public Key Accelerator for software driver matching and diagnostic purposes

Table 12-32. PKA_REVISION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	Bits should be ignored on a read
27-24	MAJOR_HW_REVISION	R	3h	4 bits binary encoding of the major hardware revision number
23-20	MINOR_HW_REVISION	R	3h	4 bits binary encoding of the minor hardware revision number
19-16	HW_PATCH_LEVEL	R	3h	4 bits binary encoding of the hardware patch level, initial release will carry value zero. Patches are used to remove bugs without changing the functionality or interface of a module
15-8	COMPLEMENT_OF_BASIC_EIP_NUMBER	R	E3h	Bit-by-bit logic complement of bits [7:0], EIP-28 gives 0xE3
7-0	BASIC_EIP_NUMBER	R	1Ch	8 bits binary encoding of the EIP number, EIP-28 gives 0x1C

12.3.22 PKA_RAM_DATA_y Register (Offset = 00016000h + formula) [Reset = 00000000h]

PKA_RAM_DATA_y is shown in Table 12-33.

Return to the Summary Table.

The 2KB PKA Data RAM is accessible in the top half of the Public Key Accelerator address space. If the Sequencer program is stored in RAM, this PKA Data RAM starts in the address space at the same location of the Sequencer program RAM.

Offset = 00016000h + (y * 4h); where y = 0h to 1FFh

Table 12-33. PKA_RAM_DATA_y Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R/W	0h	Vector Data. The last 36 to 92 bytes (9 to 23 words of 32 bits) of PKA Data RAM are used as Sequencer scratchpad during execution of complex operations. The PKA Data RAM is not accessible when the Sequencer program is stored in RAM and the Reset control bit in the PKA_SEQ_CTRL register is 1b

12.3.23 AIC_POL_CTRL Register (Offset = 00018000h) [Reset = 0000007Fh]

AIC_POL_CTRL is shown in Table 12-34.

Return to the Summary Table.

Individual polarity (high level/rising edge or low level/falling edge) control bits per interrupt input. The signal is as follows: 0=Low level/falling edge 1=High level/rising edge

Table 12-34. AIC_POL_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	Reserved
6	PROTECTION_INT	R/W	1h	1 = high level/rising edge
5	SL_ERR_INT	R/W	1h	1 = high level/rising edge
4	RESERVED	R	0h	Reserved
3	TRNG_IRQ	R/W	1h	1 = high level/rising edge
2	PKA_INT_2	R/W	1h	1 = high level/rising edge
1	PKA_INT_1	R/W	1h	1 = high level/rising edge
0	PKA_INT_0	R/W	1h	1 = high level/rising edge

12.3.24 AIC_TYPE_CTRL Register (Offset = 00018004h) [Reset = 0000007Fh]

AIC_TYPE_CTRL is shown in Table 12-35.

Return to the Summary Table.

Signal type (level or edge) control bits for each interrupt input. 0= level (the interrupt source level determines the raw status) 1= edge (the interrupt source is connected to edge detector and output of edge detector determines the raw status)

Table 12-35. AIC_TYPE_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	Reserved
6	PROTECTION_INT	R/W	1h	1 = Edge
5	SL_ERR_INT	R/W	1h	1 = Edge
4	RESERVED	R	0h	Reserved
3	TRNG_IRQ	R/W	1h	1 = Edge
2	PKA_INT_2	R/W	1h	1 = Edge
1	PKA_INT_1	R/W	1h	1 = Edge
0	PKA_INT_0	R/W	1h	1 = Edge

12.3.25 AIC_ENABLE_CTRL Register (Offset = 00018008h) [Reset = 00000000h]

AIC_ENABLE_CTRL is shown in Table 12-36.

Return to the Summary Table.

Individual enable control bits per interrupt input

Table 12-36. AIC_ENABLE_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	Reserved
6-5	ENABLECONTROLBIT56	R/W	0h	0 = Disabled 1 = Enabled
4	RESERVED	R	0h	Reserved
3-0	ENABLECONTROL	R/W	0h	0 = Disabled 1 = Enabled

12.3.26 AIC_RAW_STAT Register (Offset = 0001800Ch) [Reset = 00000000h]

AIC_RAW_STAT is shown in Table 12-37.

Return to the Summary Table.

These bits reflect the status of the interrupts after polarity control and optional edge detection (i.e. just before masking with the bits in the AIC_ENABLE_CTRL register)

Table 12-37. AIC_RAW_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	Reserved
6-5	RAWSTATUSBIT56	R	0h	0 = Inactive 1 = Pending
4	RESERVED	R	0h	Reserved
3-0	RAWSTATUS	R	0h	0 = Inactive 1 = Pending

12.3.27 AIC_ENABLE_SET Register (Offset = 0001800Ch) [Reset = 00000000h]

AIC_ENABLE_SET is shown in Table 12-38.

Return to the Summary Table.

These bits reflect the status of the interrupts gated with the enable bits of the AIC_ENABLE_CTRL Register

Table 12-38. AIC_ENABLE_SET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	Reserved
6-5	ENABLESETBIT56	W	0h	0 = Inactive 1 = Pending
4	RESERVED	R	0h	Reserved
3-0	ENABLESET	W	0h	0 = Inactive 1 = Pending

12.3.28 AIC_ENABLED_STAT Register (Offset = 00018010h) [Reset = 00000000h]

AIC_ENABLED_STAT is shown in Table 12-39.

Return to the Summary Table.

These bits reflect the status of the interrupts gated with the enable bits of the AIC_ENABLE_CTRL Register

Table 12-39. AIC_ENABLED_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	Reserved
6-5	ENABLESTATUSBIT56	R	0h	0 = Inactive 1 = Pending
4	RESERVED	R	0h	Reserved
3-0	ENABLESTATUS	R	0h	0 = Inactive 1 = Pending

12.3.29 AIC_ACK Register (Offset = 00018010h) [Reset = 00000000h]

AIC_ACK is shown in Table 12-40.

Return to the Summary Table.

Used to clear edge-detect interrupts. A ‘1’ written to any one of the bit locations acknowledges the respective interrupt and clears the status bit

Table 12-40. AIC_ACK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	Reserved
6-5	ACKNOWLEDGEBIT56	W	0h	0 = No effect. 1 = Acknowledge the interrupt signal, clears the status bit. The activated bit of Acknowledge will be cleared internally
4	RESERVED	R	0h	Reserved
3-0	ACKNOWLEDGE	W	0h	0 = No effect. 1 = Acknowledge the interrupt signal, clears the status bit. The activated bit of Acknowledge will be cleared internally

12.3.30 AIC_ENABLE_CLR Register (Offset = 00018014h) [Reset = 00000000h]

AIC_ENABLE_CLR is shown in Table 12-41.

Return to the Summary Table.

Interrupt Disable Register

Table 12-41. AIC_ENABLE_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	W	0h	Reserved
6-5	ENABLECLEARBIT56	W	0h	0 = No effect. 1 = Acknowledge the interrupt signal, clears the status bit. The activated bit of Acknowledge will be cleared internally
4	RESERVED	W	0h	Reserved
3-0	ENABLECLEAR	W	0h	Individual interrupt disable bits per interrupt input 0 = No effect. 1 = Resets the corresponding bit in the AIC_ENABLE_CTRL register to '0', i.e. disables that interrupt. The activated bit of EnableClear will be cleared internally

12.3.31 AIC_OPTIONS Register (Offset = 00018018h) [Reset = 00000007h]

AIC_OPTIONS is shown in Table 12-42.

Return to the Summary Table.

AIC Configuration Register

Table 12-42. AIC_OPTIONS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	Reserved
5-0	NUMBER_OF_INPUTS	R	7h	The EIP-201 supports up to 32 interrupt sources but the EIP-150 only has a maximum of 7 internal interrupt sources when all options are present: PKA (3), TRNG (1), reserved for backward compatibility (1), bus interface error (1) and PKA protection (1).

12.3.32 AIC_VERSION Register (Offset = 0001801Ch) [Reset = 014036C9h]

AIC_VERSION is shown in Table 12-43.

Return to the Summary Table.

AIC HW revision, EIP code

Table 12-43. AIC_VERSION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	Reserved
27-24	MAJOR_REVISION	R	1h	These bits encode the major version number for this module
23-20	MINOR_REVISION	R	4h	These bits encode the minor version number for this module
19-16	PATCH_LEVEL	R	0h	These bits encode the hardware patch level for this module – they start at value 0 on the first release
15-8	COMPLEMENT_OF_BASIC_EIP_NUMBER	R	36h	These bits simply contain the complement of bits [7:0] (0x36), used by a driver to ascertain that the AIC_VERSION register is indeed read
7-0	BASIC_EIP_NUMBER	R	C9h	These bits encode the Rambus EIP number for the AIC. As the AIC is called the EIP-201 in Rambus's IP numbering so this field contains the value 201 (decimal) or 0xC9

12.3.33 EIP150_OPTIONS Register (Offset = 0001BFF8h) [Reset = 00000D01h]

EIP150_OPTIONS is shown in Table 12-44.

Return to the Summary Table.

PKP Configuration Register

Table 12-44. EIP150_OPTIONS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R	0h	Reserved
11	RAM_ZEROIZE	R	1h	When set to 1: indicates that PKA/LMNE RAM ZEROIZE logic is present
10	AIC_PRESENT	R	1h	When set to 1: indicates that an EIP-201 AIC is included in the EIP-150
9	EIP76_PRESENT	R	0h	When set to 1: indicates that an EIP-76 TRNG is included in the EIP-150
8	EIP28_PRESENT	R	1h	When set to 1: indicates that an EIP-28 PKA is included in the EIP-150
7-4	RESERVED	R	0h	Reserved
3-0	SLAVE_INTERFACE	R	1h	This field decodes the type slave interface that used for the EIP-150. 0b0001 : AHB slave interface. 0b0011 : AXI slave interface. 0b1001 : Asynchronous AHB slave interface. 0b1101 : Asynchronous APB slave interface. Other values are reserved for future use

12.3.34 EIP150_REVISION Register (Offset = 0001BFFCh) [Reset = 03456996h]

EIP150_REVISION is shown in Table 12-45.

Return to the Summary Table.

PKP HW revision, EIP code

Table 12-45. EIP150_REVISION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	Reserved
27-24	MAJOR_REVISION	R	3h	These bits encode the major version number for this module
23-20	MINOR_REVISION	R	4h	These bits encode the minor version number for this module
19-16	PATCH_LEVEL	R	5h	encoding for hardware patch level, they start at value 0 on the first release
15-8	BIT_BY_BIT_COMPLEMENT_OF_EIP_NUMBER	R	69h	These bits simply contain the complement of bits 7 down to 0 used by a driver to ascertain that the EIP150_REVISION register is indeed read
7-0	EIP_NUMBER	R	96h	These bits encode the Rambus EIP number for the EIP-150

Chapter 13

AES

The AES accelerator module accelerates encryption and decryption operations in hardware based on the FIPS PUB 197 AES.

13.1 AES Overview 703
13.2 AESADV Operation 704
13.3 AESADV Registers 724

13.1 AES Overview

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.

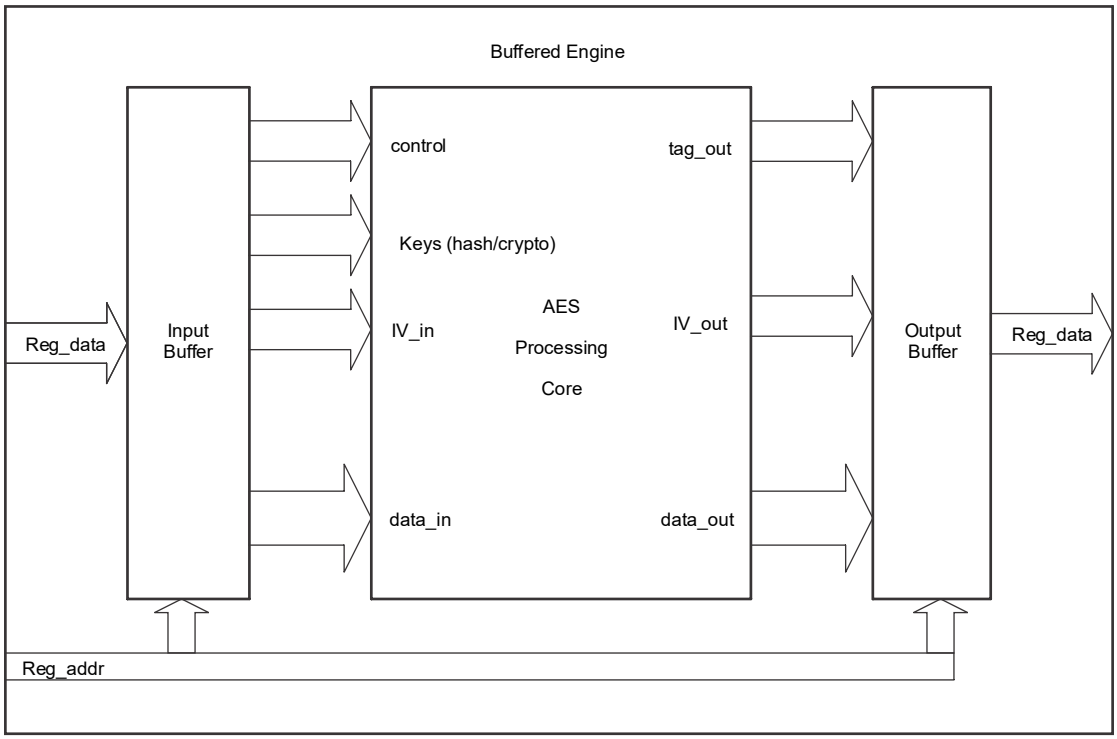


Figure 13-1. AES Block Diagram

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 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.

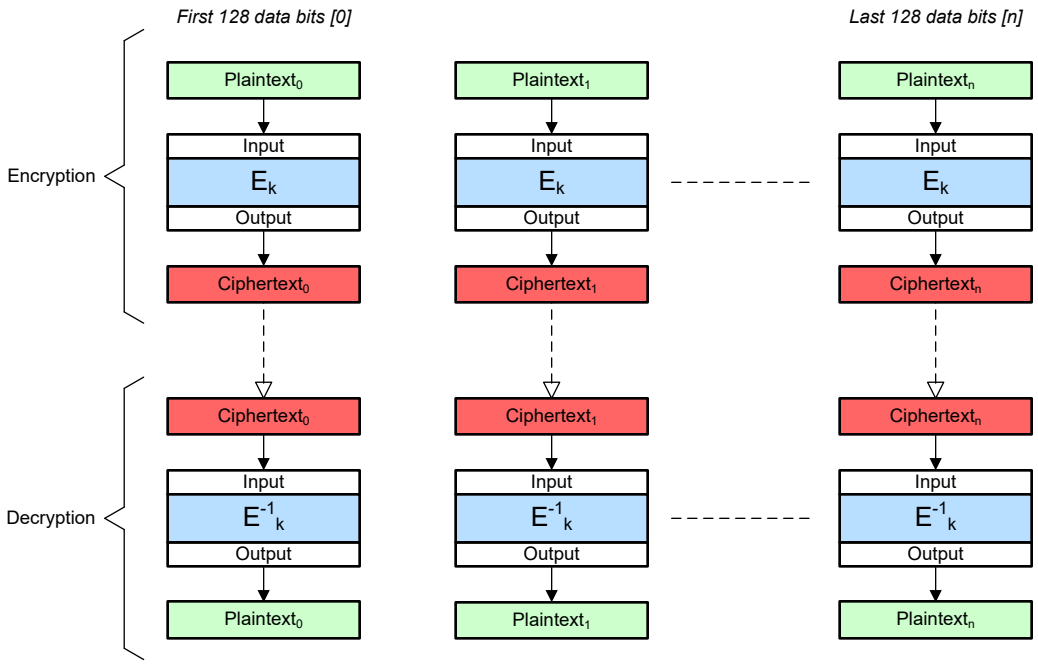


Figure 13-2. ECB Cipher

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

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.

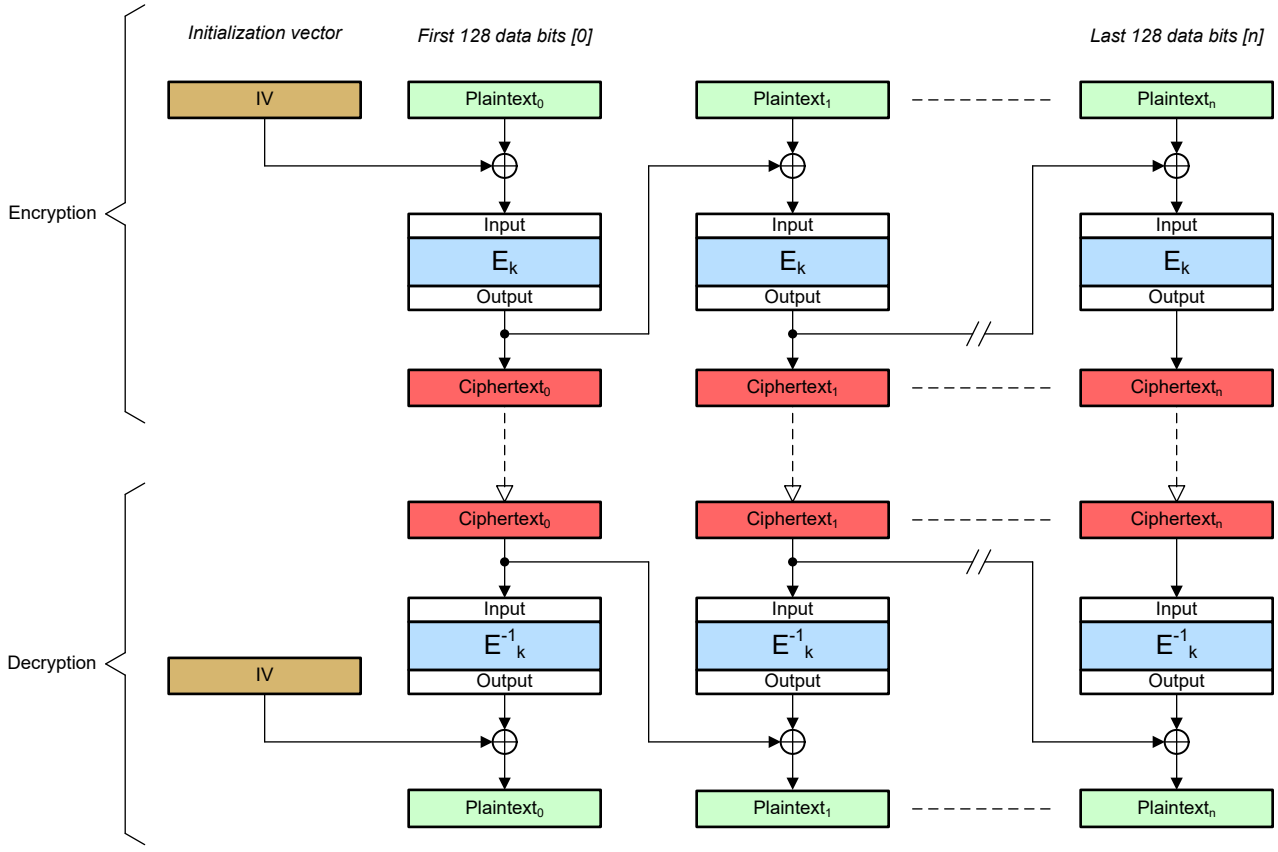


Figure 13-3. CBC Cipher

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

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.

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.

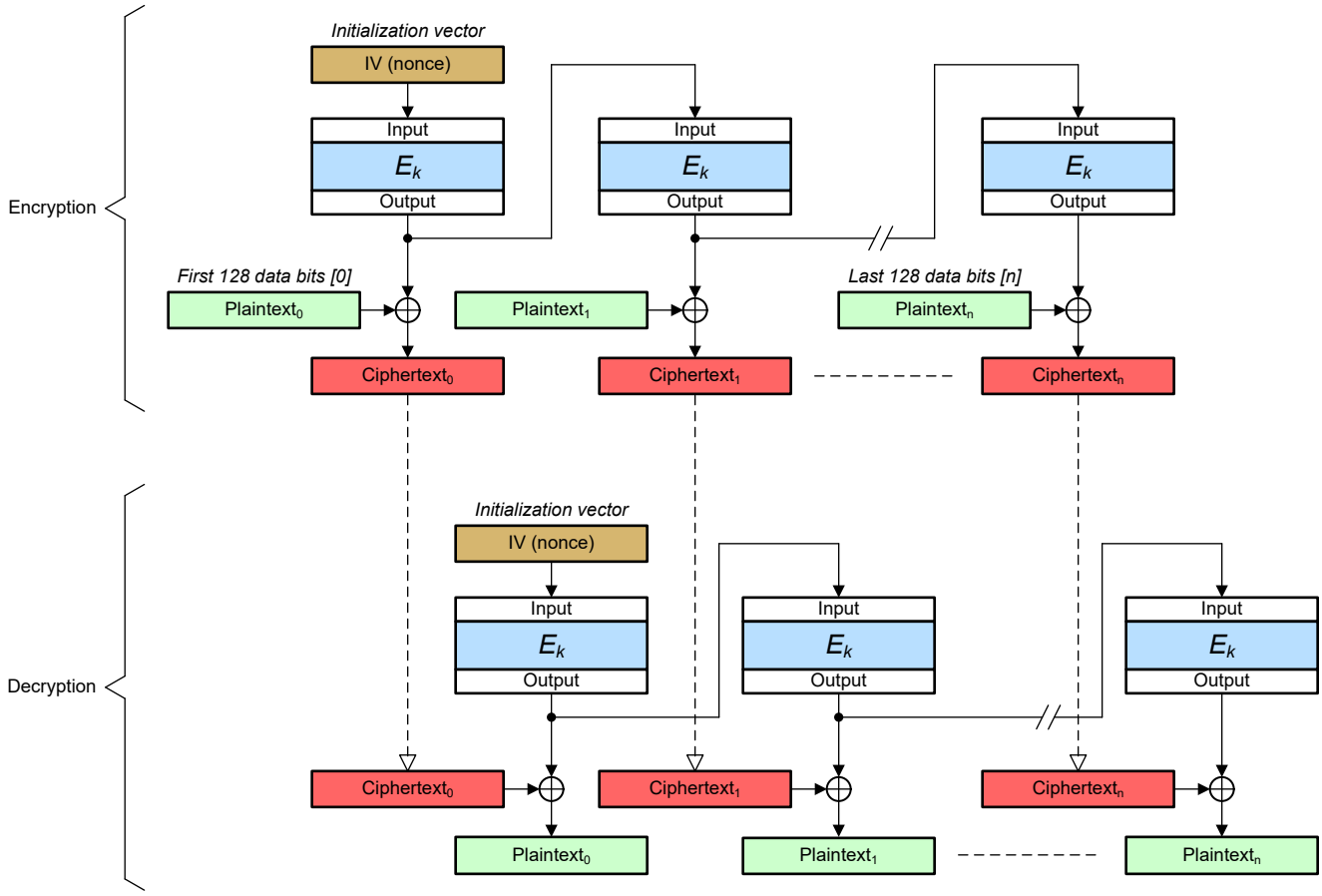


Figure 13-4. OFB Cipher

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

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.

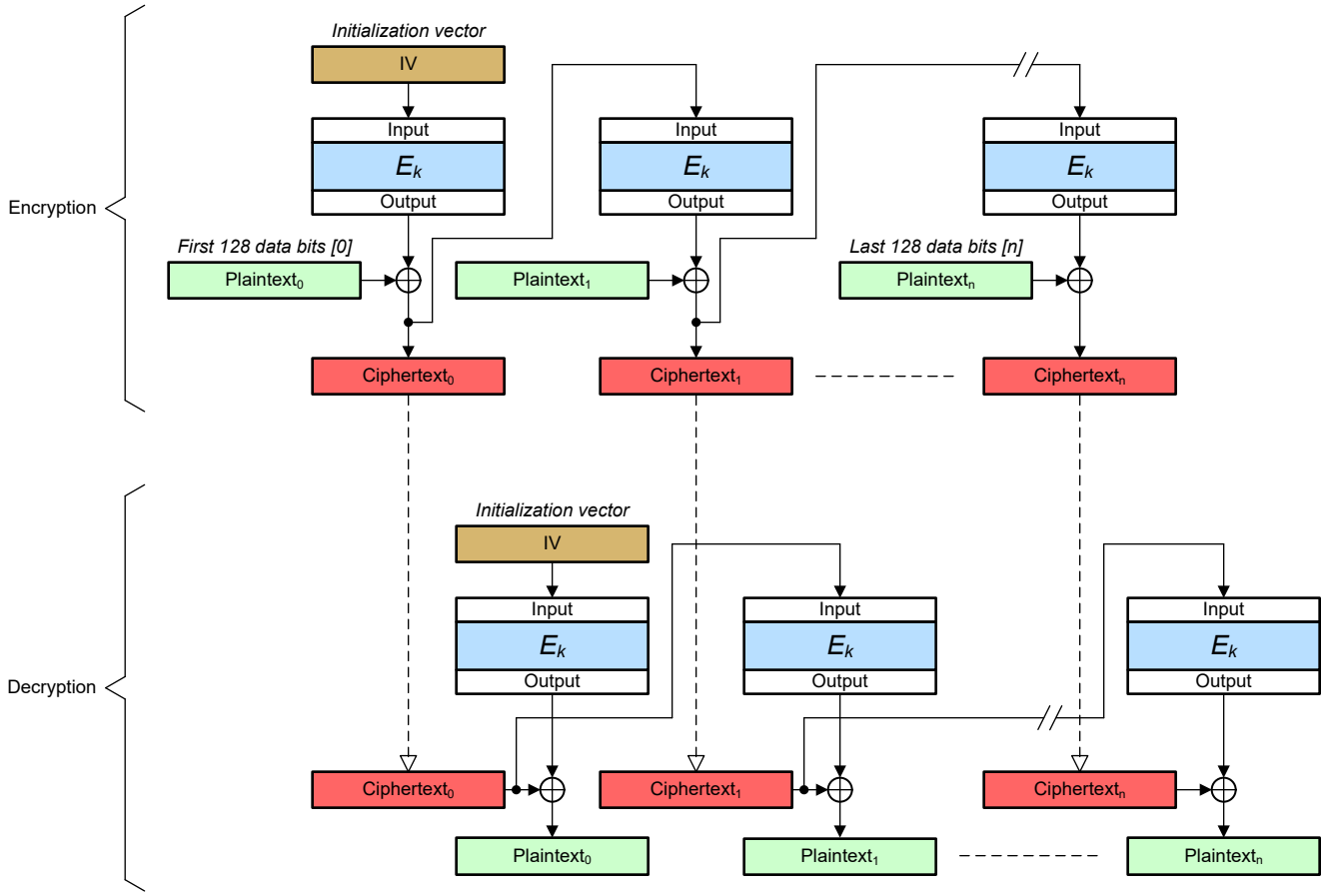


Figure 13-5. CFB Cipher

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

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.

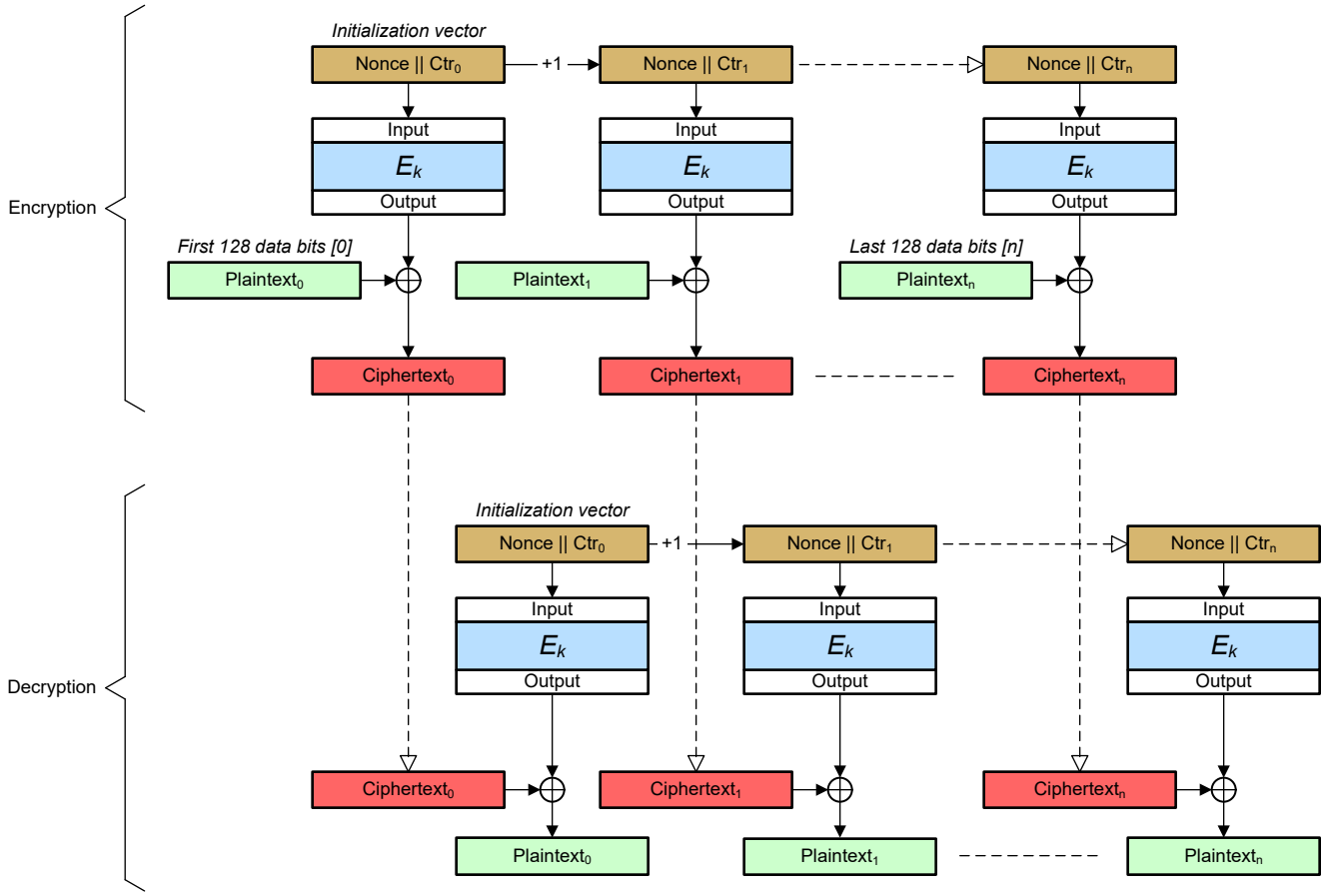


Figure 13-6. CTR Cipher

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

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.

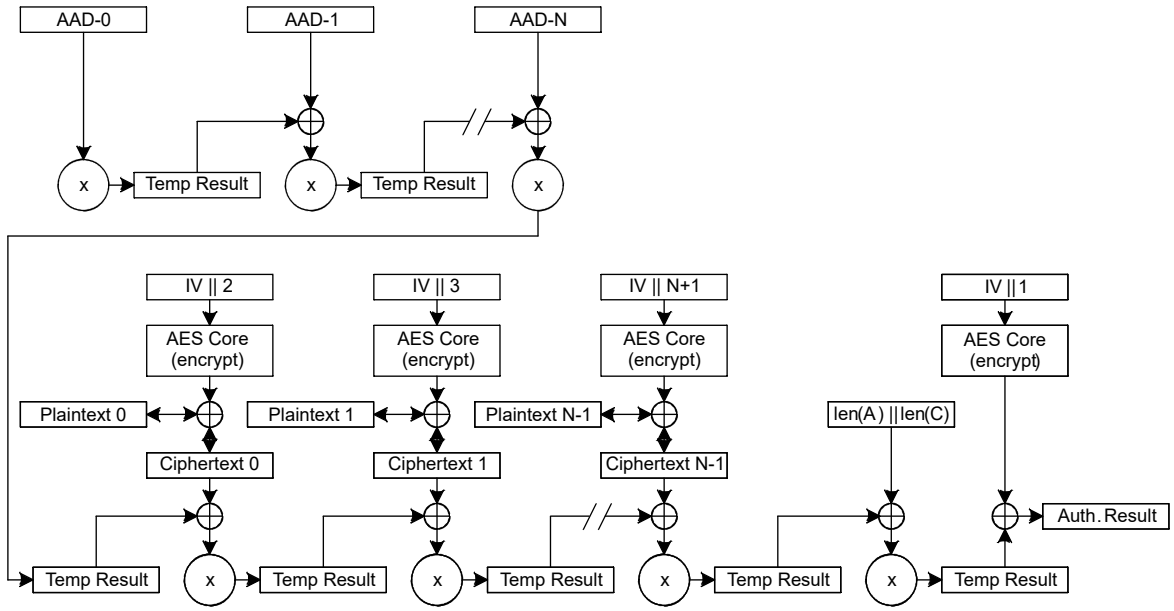


Figure 13-7. GCM Protocol Operation

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.

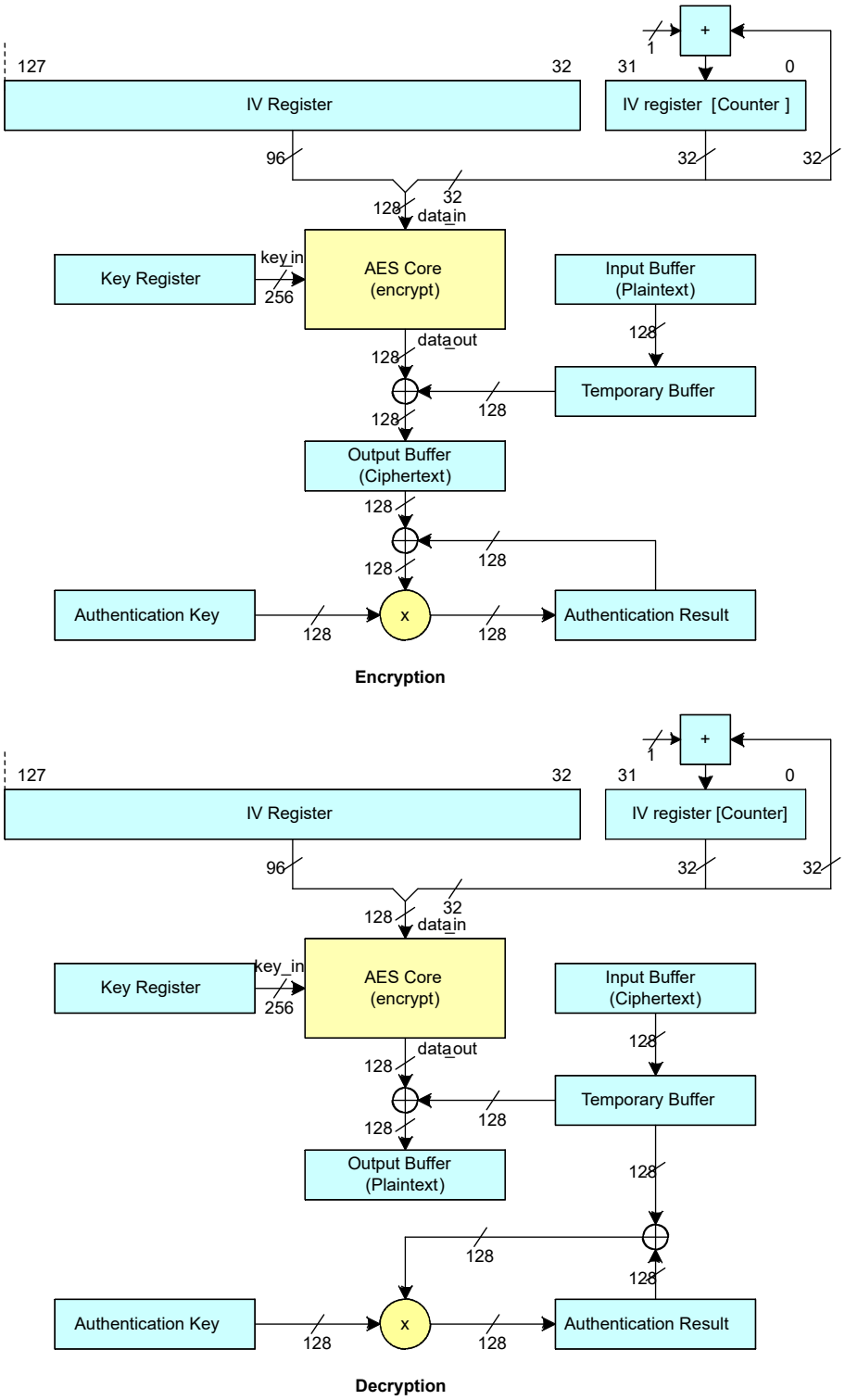


Figure 13-8. GCM Operation on a Block

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.

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.

- 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. Enabke 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

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
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.

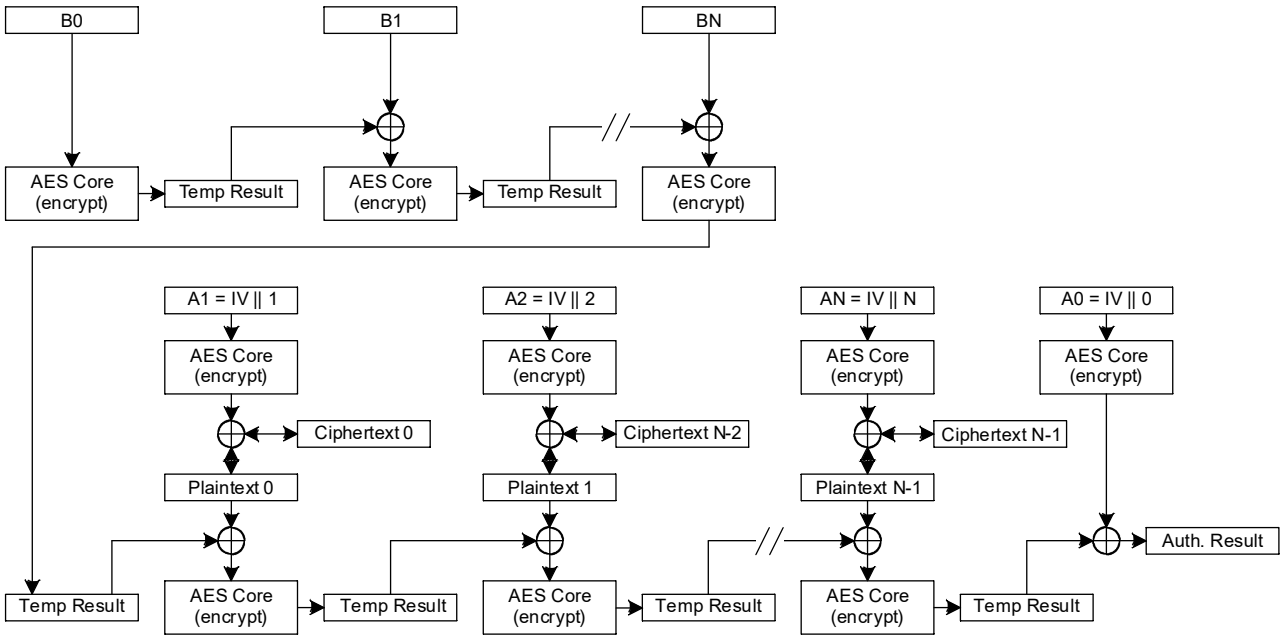


Figure 13-9. CCM Protocol Operation

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.

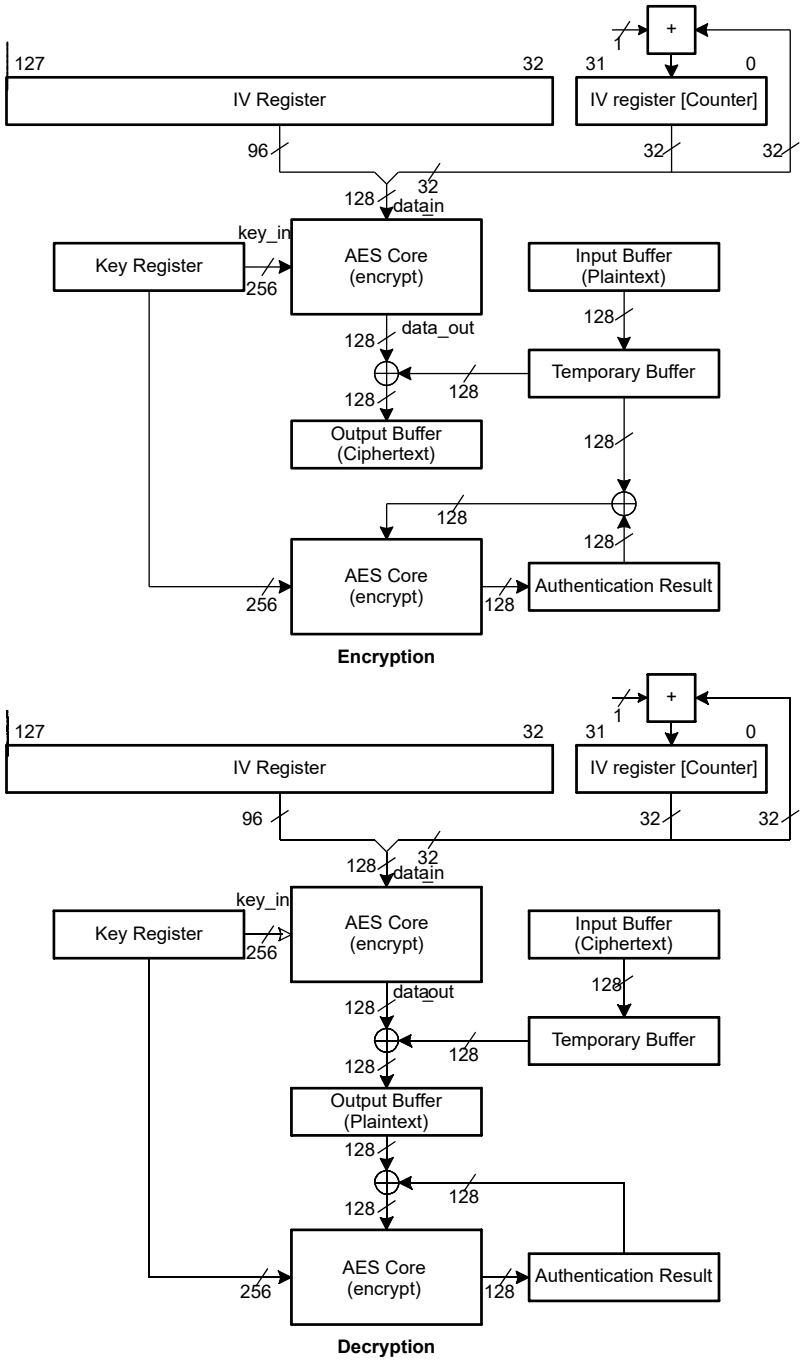


Figure 13-10. CCM Operation

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
- 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

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)

Table 13-3. AES CPU Interrupt Event Conditions (CPU_EVENT) (continued)

Index (IIDX)	Name	Description
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.
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 Section 8.2.5 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 Section 8.2.5 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 Section 8.2.5 for guidance on configuring the event registers for DMA triggers.

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

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

Table 13-7. AESADV Access Type Codes (continued)

Access Type	Code	Description
Write Type		
W	W	Write
WK	W K	Write Write protected by a key
Reset or Default Value		
-n		Value after reset or the default value

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. 0h = The CPU is not connected. 1h = The CPU is connected.
0	RESERVED	R	0h	

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 26h = KEY to allow write access to this register
23-1	RESERVED	R	0h	
0	ENABLE	R/WK	0h	Enable the power KEY must be set to 26h to write to this bit. 0h = Disable Power 1h = Enable Power

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 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 KEY must be set to B1h to write to this bit. 0h = Writing 0 has no effect 1h = Clear reset sticky bit
0	RESETASSERT	WK	0h	Assert reset to the peripheral KEY must be set to B1h to write to this bit. 0h = Writing 0 has no effect 1h = Assert reset

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 0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register 1h = The peripheral was reset since the last bit clear
15-0	RESERVED	R	0h	

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 1h = The peripheral ignores the state of the Core Halted input

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 00h = No interrupt pending 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) 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) 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. 4h = This bit indicates that the context data registers can be overwritten, and the CPU is permitted to write new context.

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. 0h = Clear Interrupt Mask 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. 0h = Clear Interrupt Mask 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) 0h = Clear Interrupt Mask 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) 0h = Clear Interrupt Mask 1h = Set Interrupt Mask

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. 0h = Clear Interrupt Mask 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. 0h = Clear Interrupt Mask 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) 0h = Clear Interrupt Mask 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) 0h = Interrupt did not occur 1h = Interrupt occurred

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. 0h = Clear Interrupt Mask 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. 0h = Clear Interrupt Mask 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) 0h = Clear Interrupt Mask 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) 0h = Interrupt did not occur 1h = Interrupt occurred

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. 0h = Clear Interrupt Mask 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. 0h = Clear Interrupt Mask 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) 0h = Clear Interrupt Mask 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) 0h = Writing 0 has no effect 1h = Set Interrupt

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. 0h = Clear Interrupt Mask 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. 0h = Clear Interrupt Mask 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) 0h = Clear Interrupt Mask 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) 0h = Writing 0 has no effect 1h = Clear Interrupt

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 00h = No interrupt pending 1h = AES trigger 0 DMA (Data Input trigger)

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. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask

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 0h = Interrupt did not occur 1h = Interrupt occurred

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 0h = Interrupt did not occur 1h = Interrupt occurred

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 0h = Writing 0 has no effect 1h = Set Interrupt

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 0h = Writing 0 has no effect 1h = Clear Interrupt

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 00h = No interrupt pending 1h = AES DMA Trigger 1 (Data Output trigger)

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. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask

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 0h = Interrupt did not occur 1h = Interrupt occurred

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 0h = Interrupt did not occur 1h = Interrupt occurred

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 0h = Writing 0 has no effect 1h = Set Interrupt

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 0h = Writing 0 has no effect 1h = Clear Interrupt

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 0h = The interrupt or event line is disabled. 1h = The interrupt or event line is in software mode. Software must clear the RIS. 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 0h = The interrupt or event line is disabled. 1h = The interrupt or event line is in software mode. Software must clear the RIS. 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 0h = The interrupt or event line is disabled. 1h = The interrupt or event line is in software mode. Software must clear the RIS. 2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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. 0h = Not ready 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. 0h = Not ready 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. 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. 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. 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. 0h = No effect 1h = Enable

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. 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. 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. 0h = No effect 1h = Enable
27	GET_DIGEST	R/W	0h	Interrupt processing and generate an intermediate digest during an AES-GCM or AES-CCM operation. 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. 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. 0h = No effect 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]. If CCM/GCM is not selected: If this bit is set to '1b', full block AES output feedback mode (OFB-128) is selected. If CCM/GCM is selected: Continue processing of an interrupted AES-GCM or AES-CCM operation in the AAD phase. 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. 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. 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. 0h = Length is 1 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. 0h = Length is 1 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. 0h = Disable CBC mode 1h = Select CBC mode

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. 1h = GHASH with H loaded and Y0-encrypted forced to 0. 2h = GHASH with H loaded and Y0-encrypted calculated internally 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. 0h = Disable CBC mode 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. 0h = Disable CBC mode 1h = Select CBC mode
9	ICM	R/W	0h	When the CFB bit is set, specifies the CFB mode feedback width: 0h = Disable CBC mode 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. When the CFB bit is set, specifies the CFB mode feedback width: 0h = CFB-128 mode 1h = CFB-1 mode 2h = CFB-8 mode 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. 0h = Disable CBC mode 1h = Select CBC mode
5	CBC	R/W	0h	If set to ‘1b’, cipher-block-chaining (CBC) mode is selected. 0h = Disable CBC mode 1h = Select CBC mode
4-3	KEYSIZE	R/W	0h	Specifies the encryption strength / key width 1h = 128-bit key 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. 0h = Decryption 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’. 0h = Not Ready 1h = Ready

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. 0h = Not Ready 1h = Ready

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. 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. A write to this register triggers the engine to start using this context. This is valid for all modes except GCM and CCM. 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. All modes must have a length > 0. For the combined modes, it is allowed to have one of the lengths equal to zero. 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. 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.

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. 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. A write to this register triggers the engine to start using this context. This is valid for all modes except GCM and CCM. 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. All modes must have a length > 0. For the combined modes, it is allowed to have one of the lengths equal to zero. 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. 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.

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) 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. A write to this register triggers the engine to start using this context for GCM and CCM.

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

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

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

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

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. 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.

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. 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.

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. 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.

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. 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.

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. In order to allow user write, perform a module reset. 0h = User write to KEY MMR is allowed 1h = User write to KEY MMR is disabled. Writing has no effect.

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

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

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

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.

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. 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. 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. Also refer to the get_digest and gcm_ccm_continue bits in AES_CTRL register.

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. 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. 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. Also refer to the get_digest and gcm_ccm_continue bits in AES_CTRL register.

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. When this bit is 1b, input and output data acknowledge is based on DMA handshake signals. 0h = Disable DMA based data handshake 1h = Enables DMA based handshake

Chapter 14

SHA2

The SHA2 (Secure Hash Algorithm-2) provides a set of cryptographic hash functions in accordance with NIST standards. This chapter describes the operations of the SHA module.

14.1 SHA Introduction 795
14.2 SHA Operation 795
14.3 SHA Auto-Feed Mode 798
14.4 SHA Registers 799

14.1 SHA Introduction

The SHA cryptographic peripheral supports a FIPS compliant secure hash algorithm (SHA-224, SHA-256) and the hash-based message authentication (HMAC) that can be used for message authentication applications.

14.1.1 SHA features

- Secure Hash Standard compliant implementation for FIPS PUB 180-2 and FIPS PUB 180-3
- HMAC support for all algorithms compliant with FIP PUB 198-1
- High performance hash with performance on one hash iteration (7.88 bits) per clock cycle for faster throughput
- Supports HMAC and basic hash operation for SHA-224 and SHA-256
- Hash and HMAC context switching
- Supports MAC key XOR and message padding
- Supports MAC key shorter, equal or larger than algorithm block size
- Supports automatic message data scheduling
- Supports message sizes up to 2^{64} bits in increments of 8-bits
- Autonomous support for DMA request generation based on message length to reduce CPU overhead
- Supports both constant and incremental address for input message parsing

14.2 SHA Operation

14.2.1 Functional Block Diagram

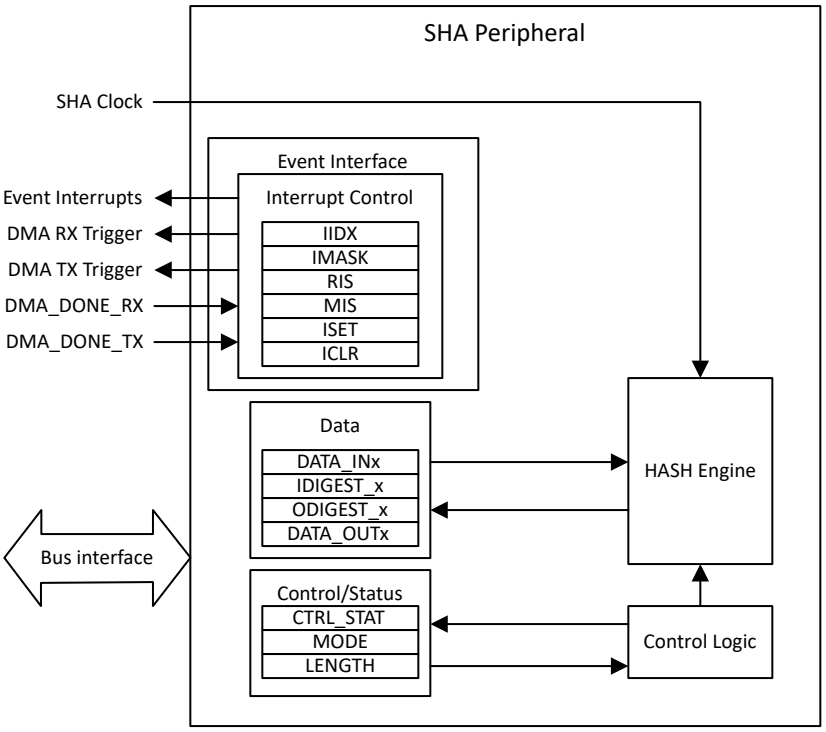


Figure 14-1. SHA Functional Block Diagram

14.2.2 HMAC Controller

The HMAC controoer generates request for data in and monitors the input data and request for data. It also controls the HASH/HMAC engine by initiating the start and providing the control information.

- New data is loaded into the engine when the engine is ready and host controller indicates one of the input available signals.

- The HASH/HMAC engine is started when the engine is ready and host asserts one of the input available signals.
- Output is made available when the enginer has finished processing a data block and the host controller has read out the previous data.

All of the above operations mentioned are performed over a wide internal bus to ensure that data load from the registers to the engine and result load from the engine to registers is performed in a single clock cycle.

14.2.3 HASH/HMAC Engine

The Hash/HMAC engine performs the SHA-2 hash computation with 512-bit data block. Additionally, an intermediate digest or initial digest value can also be loaded for hash computation.

When the hash core is idle or done, a new hash operation can be started. Any additional information needed by the hash core (mode, data to process, input digest if not starting from algorithm constants) must be provided by programming the SHA registers before the core starts a new operation. The input data must be submitted in multiples of the data block size. When the last block of data is provided, the message padding block in the core will automatically add the message padding according to the hash algorithm specification to finish the operation.

14.2.3.1 HMAC processing with MAC Key Input

The hash core expects two inputs; a MAC key and the message. Figure 14-2 shows the HMAC processing flow in time key size smaller than or equal to the block size. If the length of the MAC key is smaller than the block size, the application must pad the keys with extra zeroes at the end of the key. The total length of the key must equal the block size.

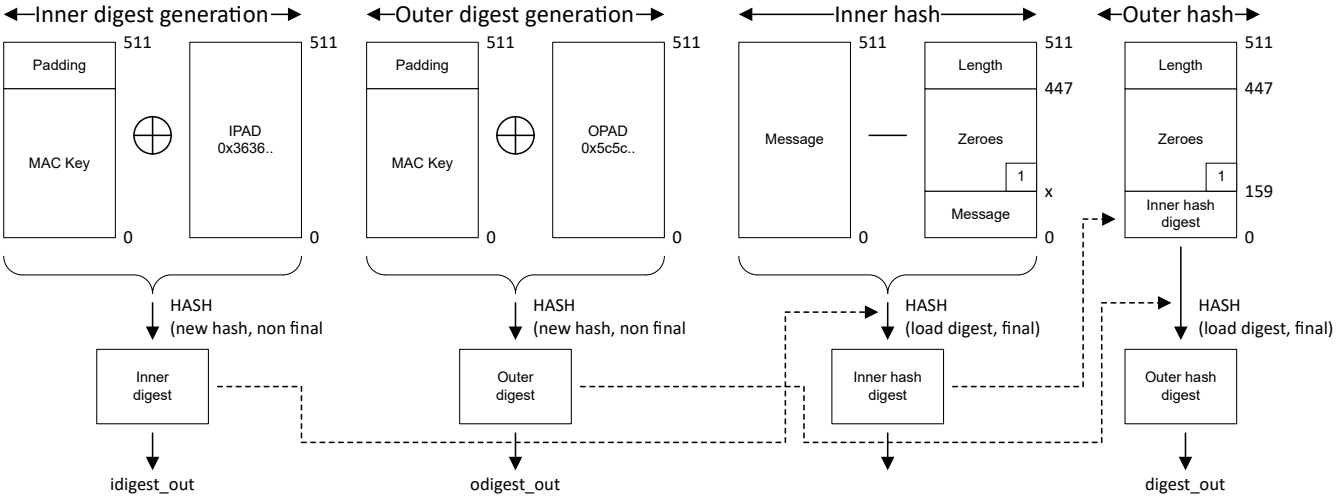


Figure 14-2. HMAC Processing with Short Key

As shown in Figure 14-3 when the MAC key is longer than the block size, input data buffer is first loaded with the key equal to block size with the MODE_IN.MAC_KEY register bit set. As a full hash must be calculated over the MACK Key, and this result be used as the actual key for HMAC, the length information and pad message must be programmed with the last block of MAC key information.

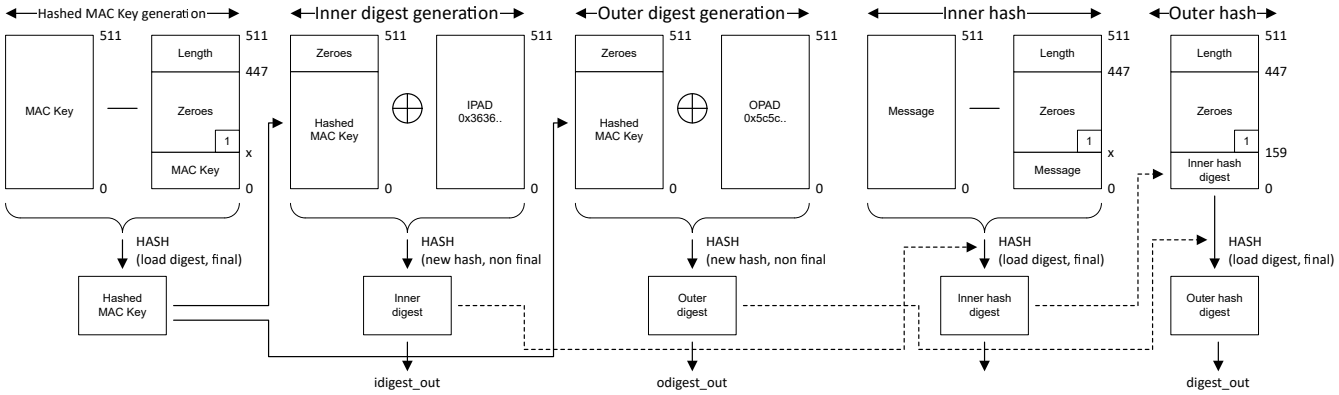


Figure 14-3. HMAC Processing with Long Key

14.2.3.2 HMAC processing with digest

Generation of the inner digest and outer digest takes two full hash operations. For multiple HMAC operations using the same MAC key, this adds processing time. To improve the performance, it is possible for the first HMAC operation to use the MAC key input and any subsequent HMAC operation to use the inner digest and outer digest from its internal registers as shown in Figure 14-4.

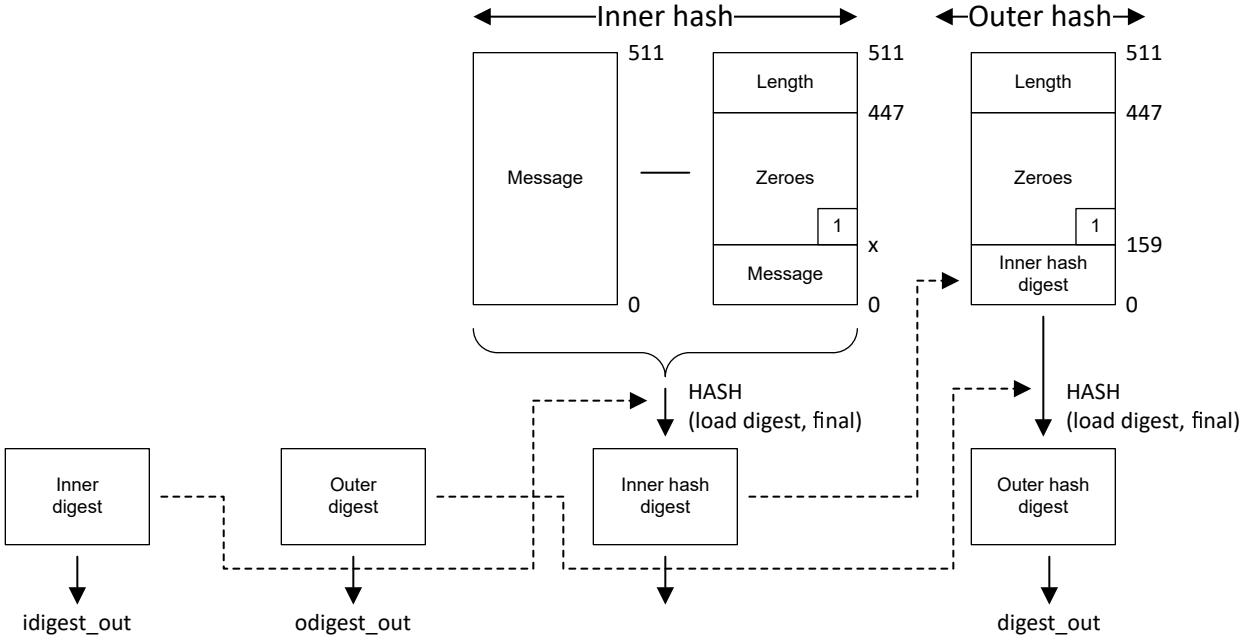


Figure 14-4. HMAC processing with reuse of digest

The length information in such cases must be applied with the message data length incremented with one block size to accommodate the digest length of the padded MAC key block.

14.2.3.3 HMAC processing with reload digest

When multiple HMAC operations need to be performed, using the same MACK keys, but the operation can be interrupted by other hash operations, the performance of the hash operation depends on how fast the MAC keys can be hashed. To increase overall performance the SHA peripheral provides the capability of reading out the inner and outer digest which can be re-loaded for the next operation as shown in Figure 14-5.

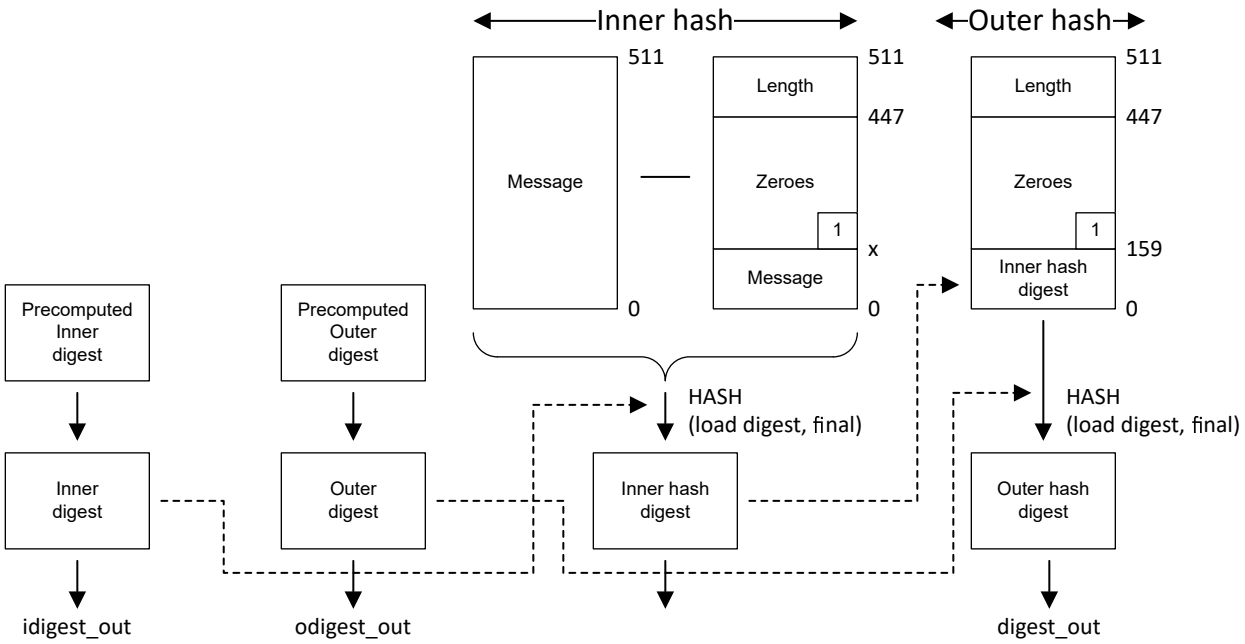


Figure 14-5. HMAC processing with reload digest

The length information in such cases must be applied with the message data length incremented with one block size to accommodate the digest length of the padded MAC key block.

14.3 SHA Auto-Feed Mode

SHA operates on a block size of data and to use the mode the peripheral has individually addressable registers to access the input or output data. Additionally the control register must be written every time to indicate a new block of data is available for processing. However when processing a very large data block the CPU has to be involved to set the appropriate control bits.

To reduce the complexity of implementation, the SHA peripheral features an auto-feed mode using a fixed data register SHAW_DATA_FIXED (offset 0x1100). The use of the mode is described below:

- The CPU sets all the control information which includes mode, length.
- The DMA channel is programmed with the source address as the location of data to be hashed and destination as the SHAW_DATA_FIXED.
- The DMA is started by priming the SHAW_DMA_START_TRIGGER register which generates a DMA request and the DMA transfer one block size of data from the source to the SHA peripheral.
- The auto-feed mechanism automatically generates the control signal for the HASH/HMAC engine.
- Once the data is digested, the next DMA trigger is generated until the length programmed is less than one block size
- On the last block of the data the auto-feed mechanism generates the last message block trigger internally and then interrupts the CPU for the hash data to be read.

Note

When using the auto-feed mode for the SHA peripheral, the operation must not be interrupted. If the operation is interrupted by a write or read, the SHA peripheral aborts the hash computation and no intermediate data provided.

14.4 SHA Registers

Table 14-1 lists the memory-mapped registers for the SHA registers. All register offset addresses not listed in Table 14-1 should be considered as reserved locations and the register contents should not be modified.

Table 14-1. SHA Registers

Offset	Acronym	Register Name	Section
800h	PWREN	Power enable	Section 14.4.1
804h	RSTCTL	Reset Control	Section 14.4.2
814h	STAT	Status Register	Section 14.4.3
1020h	SHAW_PUB0_IIDX	PUB0 IIDX register	Section 14.4.4
1028h	SHAW_PUB0_IMASK	PUB0 IMASK register	Section 14.4.5
1030h	SHAW_PUB0_RIS	PUB0 RIS register	Section 14.4.6
1038h	SHAW_PUB0_MIS	PUB0 MIS register	Section 14.4.7
1040h	SHAW_PUB0_ISET	PUB0 ISET register	Section 14.4.8
1048h	SHAW_PUB0_ICLR	PUB0 ICLR register	Section 14.4.9
1050h	SHAW_PUB1_IIDX	PUB1 IIDX register	Section 14.4.10
1058h	SHAW_PUB1_IMASK	PUB1 IIDX register	Section 14.4.11
1060h	SHAW_PUB1_RIS	PUB1 RIS register	Section 14.4.12
1068h	SHAW_PUB1_MIS	PUB1 MIS register	Section 14.4.13
1070h	SHAW_PUB1_ISET	PUB1 ISET register	Section 14.4.14
1078h	SHAW_PUB1_ICLR	PUB1 ICLR register	Section 14.4.15
1080h	SHAW_DMA_START_TRIGGER	DMA Start Trigger Register	Section 14.4.16
1100h	SHAW_DATA_FIXED	Fixed Address Wrie Register	Section 14.4.17
1200h	EIP59_DATA_IN0	SHA Data IN 0 register	Section 14.4.18
1204h	EIP59_DATA_IN1	SHA Data IN 1 register	Section 14.4.19
1208h	EIP59_DATA_IN2	SHA Data IN 2 register	Section 14.4.20
120Ch	EIP59_DATA_IN3	SHA Data IN 3 register	Section 14.4.21
1210h	EIP59_DATA_IN4	SHA Data IN 4 register	Section 14.4.22
1214h	EIP59_DATA_IN5	SHA Data IN 5 register	Section 14.4.23
1218h	EIP59_DATA_IN6	SHA Data IN 6 register	Section 14.4.24
121Ch	EIP59_DATA_IN7	SHA Data IN 7 register	Section 14.4.25
1220h	EIP59_DATA_IN8	SHA Data IN 8 register	Section 14.4.26
1224h	EIP59_DATA_IN9	SHA Data IN 9 register	Section 14.4.27
1228h	EIP59_DATA_IN10	SHA Data IN 10 register	Section 14.4.28
122Ch	EIP59_DATA_IN11	SHA Data IN 11 register	Section 14.4.29
1230h	EIP59_DATA_IN12	SHA Data IN 12 register	Section 14.4.30
1234h	EIP59_DATA_IN13	SHA Data IN 13 register	Section 14.4.31
1238h	EIP59_DATA_IN14	SHA Data IN 14 register	Section 14.4.32
123Ch	EIP59_DATA_IN15	SHA Data IN 15 register	Section 14.4.33
1240h	EIP59_IO_BUF_CTRL_STAT	SHA Status Register	Section 14.4.34
1244h	EIP59_MODE_IN	SHA Mode Register	Section 14.4.35
1248h	EIP59_LENGTH_IN_L	Data Length Register LSB	Section 14.4.36
124Ch	EIP59_LENGTH_IN_H	Data Length Register MSB	Section 14.4.37
1250h	EIP59_IDIGEST_A	SHA IDIGEST Register	Section 14.4.38
1254h	EIP59_IDIGEST_B	SHA IDIGEST Register	Section 14.4.39
1258h	EIP59_IDIGEST_C	SHA IDIGEST Register	Section 14.4.40
125Ch	EIP59_IDIGEST_D	SHA IDIGEST Register	Section 14.4.41

Table 14-1. SHA Registers (continued)

Offset	Acronym	Register Name	Section
1260h	EIP59_IDIGEST_E	SHA IDIGEST Register	Section 14.4.42
1264h	EIP59_IDIGEST_F	SHA IDIGEST Register	Section 14.4.43
1268h	EIP59_IDIGEST_G	SHA IDIGEST Register	Section 14.4.44
126Ch	EIP59_IDIGEST_H	SHA IDIGEST Register	Section 14.4.45
1270h	EIP59_ODIGEST_A	SHA ODIGEST Register	Section 14.4.46
1274h	EIP59_ODIGEST_B	SHA ODIGEST Register	Section 14.4.47
1278h	EIP59_ODIGEST_C	SHA ODIGEST Register	Section 14.4.48
127Ch	EIP59_ODIGEST_D	SHA ODIGEST Register	Section 14.4.49
1280h	EIP59_ODIGEST_E	SHA ODIGEST Register	Section 14.4.50
1284h	EIP59_ODIGEST_F	SHA ODIGEST Register	Section 14.4.51
1288h	EIP59_ODIGEST_G	SHA ODIGEST Register	Section 14.4.52
128Ch	EIP59_ODIGEST_H	SHA ODIGEST Register	Section 14.4.53
1290h	EIP59_DIGEST_A	SHA DIGEST OUT Result Register	Section 14.4.54
1294h	EIP59_DIGEST_B	SHA DIGEST OUT Result Register	Section 14.4.55
1298h	EIP59_DIGEST_C	SHA DIGEST OUT Result Register	Section 14.4.56
129Ch	EIP59_DIGEST_D	SHA DIGEST OUT Result Register	Section 14.4.57
12A0h	EIP59_DIGEST_E	SHA DIGEST OUT Result Register	Section 14.4.58
12A4h	EIP59_DIGEST_F	SHA DIGEST OUT Result Register	Section 14.4.59
12A8h	EIP59_DIGEST_G	SHA DIGEST OUT Result Register	Section 14.4.60
12ACh	EIP59_DIGEST_H	SHA DIGEST OUT Result Register	Section 14.4.61
12F8h	EIP59_CONFIG	SHA Configuration Register	Section 14.4.62
12FCh	EIP59_VERSION	SHA Version Register	Section 14.4.63

Complex bit access types are encoded to fit into small table cells. Table 14-2 shows the codes that are used for access types in this section.

Table 14-2. SHA Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set
WK	W K	Write Write protected by a key
Reset or Default Value		
-n		Value after reset or the default value

14.4.1 PWREN Register (Offset = 800h) [Reset = 00000000h]

PWREN is shown in Table 14-3.

Return to the Summary Table.

Register to control the power state

Table 14-3. PWREN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to allow Power State Change 26h = KEY to allow write access to this register
23-1	RESERVED	R/W	0h	
0	ENABLE	R/WK	0h	Enable the power KEY must be set to 26h to write to this bit. 0h = Disable Power 1h = Enable Power

14.4.2 RSTCTL Register (Offset = 804h) [Reset = 00000000h]

RSTCTL is shown in Table 14-4.

Return to the Summary Table.

Register to control reset assertion and de-assertion

Table 14-4. RSTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	Unlock key 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 KEY must be set to B1h to write to this bit. 0h = Writing 0 has no effect 1h = Clear reset sticky bit
0	RESETASSERT	WK	0h	Assert reset to the peripheral KEY must be set to B1h to write to this bit. 0h = Writing 0 has no effect 1h = Assert reset

14.4.3 STAT Register (Offset = 814h) [Reset = 00000000h]

STAT is shown in Table 14-5.

Return to the Summary Table.

peripheral enable and reset status

Table 14-5. 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 0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register 1h = The peripheral was reset since the last bit clear
15-0	RESERVED	R	0h	

14.4.4 SHAW_PUB0_IIDX Register (Offset = 1020h) [Reset = 00000000h]

SHAW_PUB0_IIDX is shown in Table 14-6.

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.

Table 14-6. SHAW_PUB0_IIDX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	STAT	R	0h	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. Reset type: SYSRSn

14.4.5 SHAW_PUB0_IMASK Register (Offset = 1028h) [Reset = 00000000h]

SHAW_PUB0_IMASK is shown in Table 14-7.

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 14-7. SHAW_PUB0_IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	OUTPUTRDY	R/W	0h	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. Reset type: SYSRSn

14.4.6 SHAW_PUB0_RIS Register (Offset = 1030h) [Reset = 00000000h]

SHAW_PUB0_RIS is shown in Table 14-8.

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.

Table 14-8. SHAW_PUB0_RIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	OUTPUTRDY	R	0h	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. Reset type: SYSRSn

14.4.7 SHAW_PUB0_MIS Register (Offset = 1038h) [Reset = 00000000h]

SHAW_PUB0_MIS is shown in Table 14-9.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Table 14-9. SHAW_PUB0_MIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	OUTPUTRDY	R	0h	Masked interrupt status. This is an AND of the IMASK and RIS registers. Reset type: SYSRSn

14.4.8 SHAW_PUB0_ISET Register (Offset = 1040h) [Reset = 00000000h]

SHAW_PUB0_ISET is shown in Table 14-10.

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.

Table 14-10. SHAW_PUB0_ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	OUTPUTRDY	R-0/W1S	0h	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. Reset type: SYSRSn

14.4.9 SHAW_PUB0_ICLR Register (Offset = 1048h) [Reset = 00000000h]

SHAW_PUB0_ICLR is shown in Table 14-11.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

Table 14-11. SHAW_PUB0_ICLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	OUTPUTRDY	R-0/W1C	0h	Interrupt clear. Write a 1 to clear corresponding Interrupt. Reset type: SYSRSn

14.4.10 SHAW_PUB1_IIDX Register (Offset = 1050h) [Reset = 00000000h]

SHAW_PUB1_IIDX is shown in Table 14-12.

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.

Table 14-12. SHAW_PUB1_IIDX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	STAT	R	0h	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. Reset type: SYSRSn

14.4.11 SHAW_PUB1_IMASK Register (Offset = 1058h) [Reset = 00000000h]

SHAW_PUB1_IMASK is shown in Table 14-13.

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 14-13. SHAW_PUB1_IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	DMA_START_TRIGGER	R/W	0h	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. Reset type: SYSRSn

14.4.12 SHAW_PUB1_RIS Register (Offset = 1060h) [Reset = 00000000h]

SHAW_PUB1_RIS is shown in Table 14-14.

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.

Table 14-14. SHAW_PUB1_RIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	DMA_START_TRIGGER	R	0h	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. Reset type: SYSRSn

14.4.13 SHAW_PUB1_MIS Register (Offset = 1068h) [Reset = 00000000h]

SHAW_PUB1_MIS is shown in Table 14-15.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Table 14-15. SHAW_PUB1_MIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	DMA_START_TRIGGER	R	0h	Masked interrupt status. This is an AND of the IMASK and RIS registers. Reset type: SYSRSn

14.4.14 SHAW_PUB1_ISET Register (Offset = 1070h) [Reset = 00000000h]

SHAW_PUB1_ISET is shown in Table 14-16.

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.

Table 14-16. SHAW_PUB1_ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	DMA_START_TRIGGER	R-0/W1S	0h	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. Reset type: SYSRSn

14.4.15 SHAW_PUB1_ICLR Register (Offset = 1078h) [Reset = 00000000h]

SHAW_PUB1_ICLR is shown in Table 14-17.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

Table 14-17. SHAW_PUB1_ICLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	DMA_START_TRIGGER	R-0/W1C	0h	Interrupt clear. Write a 1 to clear corresponding Interrupt. Reset type: SYSRSn

14.4.16 SHAW_DMA_START_TRIGGER Register (Offset = 1080h) [Reset = 00000000h]

SHAW_DMA_START_TRIGGER is shown in Table 14-18.

Return to the Summary Table.

Writing 1 to this register starts DMA transfer, DMA is configured before writing to this register

Table 14-18. SHAW_DMA_START_TRIGGER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	START_TRIGGER	R-0/W1C	0h	Trigger to start DMA Reset type: SYSRSn

14.4.17 SHAW_DATA_FIXED Register (Offset = 1100h) [Reset = 00000000h]

SHAW_DATA_FIXED is shown in Table 14-19.

Return to the Summary Table.

DMA to be configured to write to a fixed address (SHAW_DMA_DATA_IN) register, indirectly writes to registers in EIP59

Table 14-19. SHAW_DATA_FIXED Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA_IN	R-0/W	0h	Input Data on which hash is computed Reset type: SYSRSn

14.4.18 EIP59_DATA_IN0 Register (Offset = 1200h) [Reset = 00000000h]

EIP59_DATA_IN0 is shown in Table 14-20.

Return to the Summary Table.

SHA Data IN 0 register

Table 14-20. EIP59_DATA_IN0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA_IN	R-0/W	0h	Input Data on which hash is computed Reset type: SYSRSn

14.4.19 EIP59_DATA_IN1 Register (Offset = 1204h) [Reset = 00000000h]

EIP59_DATA_IN1 is shown in Table 14-21.

Return to the Summary Table.

SHA Data IN 1 register

Table 14-21. EIP59_DATA_IN1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA_IN	R-0/W	0h	Input Data on which hash is computed Reset type: SYSRSn

14.4.20 EIP59_DATA_IN2 Register (Offset = 1208h) [Reset = 00000000h]

EIP59_DATA_IN2 is shown in Table 14-22.

Return to the Summary Table.

SHA Data IN 2 register

Table 14-22. EIP59_DATA_IN2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA_IN	R-0/W	0h	Input Data on which hash is computed Reset type: SYSRSn

14.4.21 EIP59_DATA_IN3 Register (Offset = 120Ch) [Reset = 00000000h]

EIP59_DATA_IN3 is shown in Table 14-23.

Return to the Summary Table.

SHA Data IN 3 register

Table 14-23. EIP59_DATA_IN3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA_IN	R-0/W	0h	Input Data on which hash is computed Reset type: SYSRSn

14.4.22 EIP59_DATA_IN4 Register (Offset = 1210h) [Reset = 00000000h]

EIP59_DATA_IN4 is shown in Table 14-24.

Return to the Summary Table.

SHA Data IN 4 register

Table 14-24. EIP59_DATA_IN4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA_IN	R-0/W	0h	Input Data on which hash is computed Reset type: SYSRSn

14.4.23 EIP59_DATA_IN5 Register (Offset = 1214h) [Reset = 00000000h]

EIP59_DATA_IN5 is shown in Table 14-25.

Return to the Summary Table.

SHA Data IN 5 register

Table 14-25. EIP59_DATA_IN5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA_IN	R-0/W	0h	Input Data on which hash is computed Reset type: SYSRSn

14.4.24 EIP59_DATA_IN6 Register (Offset = 1218h) [Reset = 00000000h]

EIP59_DATA_IN6 is shown in Table 14-26.

Return to the Summary Table.

SHA Data IN 6 register

Table 14-26. EIP59_DATA_IN6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA_IN	R-0/W	0h	Input Data on which hash is computed Reset type: SYSRSn

14.4.25 EIP59_DATA_IN7 Register (Offset = 121Ch) [Reset = 00000000h]

EIP59_DATA_IN7 is shown in Table 14-27.

Return to the Summary Table.

SHA Data IN 7 register

Table 14-27. EIP59_DATA_IN7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA_IN	R-0/W	0h	Input Data on which hash is computed Reset type: SYSRSn

14.4.26 EIP59_DATA_IN8 Register (Offset = 1220h) [Reset = 00000000h]

EIP59_DATA_IN8 is shown in Table 14-28.

Return to the Summary Table.

SHA Data IN 8 register

Table 14-28. EIP59_DATA_IN8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA_IN	R-0/W	0h	Input Data on which hash is computed Reset type: SYSRSn

14.4.27 EIP59_DATA_IN9 Register (Offset = 1224h) [Reset = 00000000h]

EIP59_DATA_IN9 is shown in Table 14-29.

Return to the Summary Table.

SHA Data IN 9 register

Table 14-29. EIP59_DATA_IN9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA_IN	R-0/W	0h	Input Data on which hash is computed Reset type: SYSRSn

14.4.28 EIP59_DATA_IN10 Register (Offset = 1228h) [Reset = 00000000h]

EIP59_DATA_IN10 is shown in Table 14-30.

Return to the Summary Table.

SHA Data IN 10 register

Table 14-30. EIP59_DATA_IN10 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA_IN	R-0/W	0h	Input Data on which hash is computed Reset type: SYSRSn

14.4.29 EIP59_DATA_IN11 Register (Offset = 122Ch) [Reset = 00000000h]

EIP59_DATA_IN11 is shown in Table 14-31.

Return to the Summary Table.

SHA Data IN 11 register

Table 14-31. EIP59_DATA_IN11 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA_IN	R-0/W	0h	Input Data on which hash is computed Reset type: SYSRSn

14.4.30 EIP59_DATA_IN12 Register (Offset = 1230h) [Reset = 00000000h]

EIP59_DATA_IN12 is shown in Table 14-32.

Return to the Summary Table.

SHA Data IN 12 register

Table 14-32. EIP59_DATA_IN12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA_IN	R-0/W	0h	Input Data on which hash is computed Reset type: SYSRSn

14.4.31 EIP59_DATA_IN13 Register (Offset = 1234h) [Reset = 00000000h]

EIP59_DATA_IN13 is shown in Table 14-33.

Return to the Summary Table.

SHA Data IN 13 register

Table 14-33. EIP59_DATA_IN13 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA_IN	R-0/W	0h	Input Data on which hash is computed Reset type: SYSRSn

14.4.32 EIP59_DATA_IN14 Register (Offset = 1238h) [Reset = 00000000h]

EIP59_DATA_IN14 is shown in Table 14-34.

Return to the Summary Table.

SHA Data IN 14 register

Table 14-34. EIP59_DATA_IN14 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA_IN	R-0/W	0h	Input Data on which hash is computed Reset type: SYSRSn

14.4.33 EIP59_DATA_IN15 Register (Offset = 123Ch) [Reset = 00000000h]

EIP59_DATA_IN15 is shown in Table 14-35.

Return to the Summary Table.

SHA Data IN 15 register

Table 14-35. EIP59_DATA_IN15 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA_IN	R-0/W	0h	Input Data on which hash is computed Reset type: SYSRSn

14.4.34 EIP59_IO_BUF_CTRL_STAT Register (Offset = 1240h) [Reset = 00000000h]

EIP59_IO_BUF_CTRL_STAT is shown in Table 14-36.

Return to the Summary Table.

SHA Status Register

Table 14-36. EIP59_IO_BUF_CTRL_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	RESERVED	R	0h	Reserved
13	ODIGEST_OUT_AV	R	0h	Status to indicate odigest out is available Reset type: SYSRSn
12	IDIGEST_OUT_AV	R	0h	Status to indicate idigest out is available Reset type: SYSRSn
11-8	RESERVED	R	0h	Reserved
7	GET_DIGEST	R-0/W	0h	Write 1 to get the digest on the digest out registers Reset type: SYSRSn
6	PAD_MESSAGE	R-0/W	0h	Write 1 to pad the message Reset type: SYSRSn
5	ODIGEST_IN_AV	R-0/W	0h	Write 1 to indicate odigest is available Reset type: SYSRSn
4	IDIGEST_IN_AV	R-0/W	0h	Write 1 to indicate idigest is available Reset type: SYSRSn
3	MODE_IN_AV	R-0/W	0h	Write 1 to indicate mode is available Reset type: SYSRSn
2	LENGTH_IN_AV	R-0/W	0h	Write 1 to indicate length is available Reset type: SYSRSn
1	DATA_IN_AV	R/W	0h	Indicates that the EIP59_DATA_IN_x registers contain a new input data block for processing. The Host must write a 1b to this bit to start an operation on the data in EIP59_DATA_IN_x as soon as the EIP-59 engine is ready to process new input data. Alternatively, when DMA is selected, asserting input signal dma_in_clear to 1b must be used instead to start processing new input data. As an exception, the data_in_av bit must only be used to start processing a zero length message. Writing a '0' to the data_in_av bit has no effect. This bit is automatically cleared (i.e. reads 0b) when the Wide bus EIP-59 engine has copied the EIP59_DATA_IN_x contents. This bit reads as 1b between the time it was set by the Host and the Wide bus EIP-59 engine has started processing the new input block. Reset type: SYSRSn
0	OUTBUF_FULL	R-0/W1C	0h	Indicates output is available, needs to be written 1 to clear this bit before starting new hash Reset type: SYSRSn

14.4.35 EIP59_MODE_IN Register (Offset = 1244h) [Reset = 00000000h]

EIP59_MODE_IN is shown in Table 14-37.

Return to the Summary Table.

SHA Mode Register

Table 14-37. EIP59_MODE_IN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	Reserved
6	SHA_224	R/W	0h	Write 1 to compute hash in sha 224 mode, while reading back this information will be available in 4rd bit instead of 6th bit and 6th bit remains reserved Reset type: SYSRSn
5	SHA_256	R/W	0h	Write 1 to compute hash in sha 256 mode, while reading back this information will be available in 3rd bit instead of 5th bit and 5th bit remains reserved Reset type: SYSRSn
4-3	RESERVED	R	0h	Reserved
2	NEW_HASH	R/W	0h	Write 1 to indicate new hash computation Reset type: SYSRSn
1	MAC_KEY	R/W	0h	Write 1 to indicate mac key computation Reset type: SYSRSn
0	HASH_HMAC	R/W	0h	Write 1 to indicate if it is hmac operation Reset type: SYSRSn

14.4.36 EIP59_LENGTH_IN_L Register (Offset = 1248h) [Reset = 00000000h]

EIP59_LENGTH_IN_L is shown in Table 14-38.

Return to the Summary Table.

Data Length Register LSB

Table 14-38. EIP59_LENGTH_IN_L Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LENGTH_LSB	R-0/W	0h	Data Length LSB Reset type: SYSRSn

14.4.37 EIP59_LENGTH_IN_H Register (Offset = 124Ch) [Reset = 00000000h]

EIP59_LENGTH_IN_H is shown in Table 14-39.

Return to the Summary Table.

Data Length Register MSB

Table 14-39. EIP59_LENGTH_IN_H Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LENGTH_MSB	R-0/W	0h	Data Length MSB Reset type: SYSRSn

14.4.38 EIP59_IDIGEST_A Register (Offset = 1250h) [Reset = 00000000h]

EIP59_IDIGEST_A is shown in Table 14-40.

Return to the Summary Table.

SHA IDIGEST Register

Table 14-40. EIP59_IDIGEST_A Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	IDIGEST_A	R/W	0h	Inner Digest A Reset type: SYSRSn

14.4.39 EIP59_IDIGEST_B Register (Offset = 1254h) [Reset = 00000000h]

EIP59_IDIGEST_B is shown in Table 14-41.

Return to the Summary Table.

SHA IDIGEST Register

Table 14-41. EIP59_IDIGEST_B Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	IDIGEST_B	R/W	0h	Inner Digest B Reset type: SYSRSn

14.4.40 EIP59_IDIGEST_C Register (Offset = 1258h) [Reset = 00000000h]

EIP59_IDIGEST_C is shown in Table 14-42.

Return to the Summary Table.

SHA IDIGEST Register

Table 14-42. EIP59_IDIGEST_C Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	IDIGEST_C	R/W	0h	Inner Digest C Reset type: SYSRSn

14.4.41 EIP59_IDIGEST_D Register (Offset = 125Ch) [Reset = 00000000h]

EIP59_IDIGEST_D is shown in Table 14-43.

Return to the Summary Table.

SHA IDIGEST Register

Table 14-43. EIP59_IDIGEST_D Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	IDIGEST_D	R/W	0h	Inner Digest D Reset type: SYSRSn

14.4.42 EIP59_IDIGEST_E Register (Offset = 1260h) [Reset = 00000000h]

EIP59_IDIGEST_E is shown in Table 14-44.

Return to the Summary Table.

SHA IDIGEST Register

Table 14-44. EIP59_IDIGEST_E Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	IDIGEST_E	R/W	0h	Inner Digest E Reset type: SYSRSn

14.4.43 EIP59_IDIGEST_F Register (Offset = 1264h) [Reset = 00000000h]

EIP59_IDIGEST_F is shown in Table 14-45.

Return to the Summary Table.

SHA IDIGEST Register

Table 14-45. EIP59_IDIGEST_F Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	IDIGEST_F	R/W	0h	Inner Digest F Reset type: SYSRSn

14.4.44 EIP59_IDIGEST_G Register (Offset = 1268h) [Reset = 00000000h]

EIP59_IDIGEST_G is shown in Table 14-46.

Return to the Summary Table.

SHA IDIGEST Register

Table 14-46. EIP59_IDIGEST_G Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	IDIGEST_G	R/W	0h	Inner Digest G Reset type: SYSRSn

14.4.45 EIP59_IDIGEST_H Register (Offset = 126Ch) [Reset = 00000000h]

EIP59_IDIGEST_H is shown in Table 14-47.

Return to the Summary Table.

SHA IDIGEST Register

Table 14-47. EIP59_IDIGEST_H Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	IDIGEST_H	R/W	0h	Inner Digest H Reset type: SYSRSn

14.4.46 EIP59_ODIGEST_A Register (Offset = 1270h) [Reset = 00000000h]

EIP59_ODIGEST_A is shown in Table 14-48.

Return to the Summary Table.

SHA ODIGEST Register

Table 14-48. EIP59_ODIGEST_A Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ODIGEST_A	R/W	0h	Outer Digest A Reset type: SYSRSn

14.4.47 EIP59_ODIGEST_B Register (Offset = 1274h) [Reset = 00000000h]

EIP59_ODIGEST_B is shown in Table 14-49.

Return to the Summary Table.

SHA ODIGEST Register

Table 14-49. EIP59_ODIGEST_B Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ODIGEST_B	R/W	0h	Outer Digest B Reset type: SYSRSn

14.4.48 EIP59_ODIGEST_C Register (Offset = 1278h) [Reset = 00000000h]

EIP59_ODIGEST_C is shown in Table 14-50.

Return to the Summary Table.

SHA ODIGEST Register

Table 14-50. EIP59_ODIGEST_C Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ODIGEST_C	R/W	0h	Outer Digest C Reset type: SYSRSn

14.4.49 EIP59_ODIGEST_D Register (Offset = 127Ch) [Reset = 00000000h]

EIP59_ODIGEST_D is shown in Table 14-51.

Return to the Summary Table.

SHA ODIGEST Register

Table 14-51. EIP59_ODIGEST_D Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ODIGEST_D	R/W	0h	Outer Digest D Reset type: SYSRSn

14.4.50 EIP59_ODIGEST_E Register (Offset = 1280h) [Reset = 00000000h]

EIP59_ODIGEST_E is shown in Table 14-52.

Return to the Summary Table.

SHA ODIGEST Register

Table 14-52. EIP59_ODIGEST_E Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ODIGEST_E	R/W	0h	Outer Digest E Reset type: SYSRSn

14.4.51 EIP59_ODIGEST_F Register (Offset = 1284h) [Reset = 00000000h]

EIP59_ODIGEST_F is shown in Table 14-53.

Return to the Summary Table.

SHA ODIGEST Register

Table 14-53. EIP59_ODIGEST_F Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ODIGEST_F	R/W	0h	Outer Digest F Reset type: SYSRSn

14.4.52 EIP59_ODIGEST_G Register (Offset = 1288h) [Reset = 00000000h]

EIP59_ODIGEST_G is shown in Table 14-54.

Return to the Summary Table.

SHA ODIGEST Register

Table 14-54. EIP59_ODIGEST_G Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ODIGEST_G	R/W	0h	Outer Digest G Reset type: SYSRSn

14.4.53 EIP59_ODIGEST_H Register (Offset = 128Ch) [Reset = 00000000h]

EIP59_ODIGEST_H is shown in Table 14-55.

Return to the Summary Table.

SHA ODIGEST Register

Table 14-55. EIP59_ODIGEST_H Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ODIGEST_H	R/W	0h	Outer Digest H Reset type: SYSRSn

14.4.54 EIP59_DIGEST_A Register (Offset = 1290h) [Reset = 00000000h]

EIP59_DIGEST_A is shown in Table 14-56.

Return to the Summary Table.

SHA DIGEST OUT Result Register

Table 14-56. EIP59_DIGEST_A Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DIGEST_A	R	0h	Digest Output A Reset type: SYSRSn

14.4.55 EIP59_DIGEST_B Register (Offset = 1294h) [Reset = 00000000h]

EIP59_DIGEST_B is shown in Table 14-57.

Return to the Summary Table.

SHA DIGEST OUT Result Register

Table 14-57. EIP59_DIGEST_B Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DIGEST_B	R	0h	Digest Output B Reset type: SYSRSn

14.4.56 EIP59_DIGEST_C Register (Offset = 1298h) [Reset = 00000000h]

EIP59_DIGEST_C is shown in Table 14-58.

Return to the Summary Table.

SHA DIGEST OUT Result Register

Table 14-58. EIP59_DIGEST_C Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DIGEST_C	R	0h	Digest Output C Reset type: SYSRSn

14.4.57 EIP59_DIGEST_D Register (Offset = 129Ch) [Reset = 00000000h]

EIP59_DIGEST_D is shown in Table 14-59.

Return to the Summary Table.

SHA DIGEST OUT Result Register

Table 14-59. EIP59_DIGEST_D Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DIGEST_D	R	0h	Digest Output D Reset type: SYSRSn

14.4.58 EIP59_DIGEST_E Register (Offset = 12A0h) [Reset = 00000000h]

EIP59_DIGEST_E is shown in Table 14-60.

Return to the Summary Table.

SHA DIGEST OUT Result Register

Table 14-60. EIP59_DIGEST_E Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DIGEST_E	R	0h	Digest Output E Reset type: SYSRSn

14.4.59 EIP59_DIGEST_F Register (Offset = 12A4h) [Reset = 00000000h]

EIP59_DIGEST_F is shown in Table 14-61.

Return to the Summary Table.

SHA DIGEST OUT Result Register

Table 14-61. EIP59_DIGEST_F Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DIGEST_F	R	0h	Digest Output F Reset type: SYSRSn

14.4.60 EIP59_DIGEST_G Register (Offset = 12A8h) [Reset = 00000000h]

EIP59_DIGEST_G is shown in Table 14-62.

Return to the Summary Table.

SHA DIGEST OUT Result Register

Table 14-62. EIP59_DIGEST_G Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DIGEST_G	R	0h	Digest Output G Reset type: SYSRSn

14.4.61 EIP59_DIGEST_H Register (Offset = 12ACh) [Reset = 00000000h]

EIP59_DIGEST_H is shown in Table 14-63.

Return to the Summary Table.

SHA DIGEST OUT Result Register

Table 14-63. EIP59_DIGEST_H Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DIGEST_H	R	0h	Digest Output H Reset type: SYSRSn

14.4.62 EIP59_CONFIG Register (Offset = 12F8h) [Reset = 01400019h]

EIP59_CONFIG is shown in Table 14-64.

Return to the Summary Table.

SHA Configuration Register

Table 14-64. EIP59_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CONFIG	R	01400019h	Configuration Register = 0x01400019 Reset type: SYSRSn

14.4.63 EIP59_VERSION Register (Offset = 12FCh) [Reset = 2240C43Bh]

EIP59_VERSION is shown in Table 14-65.

Return to the Summary Table.

SHA Version Register

Table 14-65. EIP59_VERSION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	VERSION	R	2240C43Bh	Version Register = 0x2240C43B Reset type: SYSRSn

Chapter 15

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.

15.1 CRC Overview 865
15.2 CRC Operation 865
15.3 CRCP0 Registers 868

15.1 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 15-1. CRC and CRC-P Key Features

15.1.1 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.

$$f(x)=x^{16}+x^{12}+x^{5}+1 \quad (11)$$

The CRC16-CCITT digest size is 16 bits (half-word).

15.1.2 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.

$$f(x)=x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^{8}+x^{7}+x^{5}+x^{4}+x^{2}+x+1 \quad (12)$$

The digest size for CRC32-ISO3309 is 32 bits.

15.2 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.

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.

15.2.1 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.

15.2.2 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).

15.2.2.1 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.

15.2.2.2 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-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:

- **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).

15.2.2.3 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.

15.2.2.4 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 15-1. 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.

15.2.2.5 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).

15.3 CRCP0 Registers

Table 15-2 lists the memory-mapped registers for the CRCP0 registers. All register offset addresses not listed in Table 15-2 should be considered as reserved locations and the register contents should not be modified.

Table 15-2. 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 15-3 shows the codes that are used for access types in this section.

Table 15-3. 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 K	Write 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.

15.3.1 PWREN (Offset = 800h) [Reset = 00000000h]

PWREN is shown in Figure 15-2 and described in Table 15-4.

Return to the Summary Table.

Register to control the power state

Figure 15-2. 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 15-4. PWREN Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to allow Power State Change 26h = KEY to allow write access to this register
23-1	RESERVED	R	0h	
0	ENABLE	R/WK	0h	Enable the power KEY must be set to 26h to write to this bit. 0h = Disable Power 1h = Enable Power

15.3.2 RSTCTL (Offset = 804h) [Reset = 00000000h]

RSTCTL is shown in Figure 15-3 and described in Table 15-5.

Return to the Summary Table.

Register to control reset assertion and de-assertion

Figure 15-3. 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 15-5. RSTCTL Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	Unlock key 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 KEY must be set to B1h to write to this bit. 0h = Writing 0 has no effect 1h = Clear reset sticky bit
0	RESETASSERT	WK	0h	Assert reset to the peripheral KEY must be set to B1h to write to this bit. 0h = Writing 0 has no effect 1h = Assert reset

15.3.3 STAT (Offset = 814h) [Reset = 00000000h]

STAT is shown in Figure 15-4 and described in Table 15-6.

Return to the Summary Table.

peripheral enable and reset status

Figure 15-4. 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 15-6. 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 0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register 1h = The peripheral was reset since the last bit clear
15-0	RESERVED	R	0h	

15.3.4 CLKSEL (Offset = 1004h) [Reset = 00000001h]

CLKSEL is shown in Figure 15-5 and described in Table 15-7.

Return to the Summary Table.

Clock source selection

Figure 15-5. 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 15-7. CLKSEL Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	MCLK_SEL	R	1h	Selects main clock (MCLK) if enabled 0h = Does not select this clock as a source 1h = Select this clock as a source

15.3.5 DESC (Offset = 10FCh) [Reset = 00000000h]

DESC is shown in Figure 15-6 and described in Table 15-8.

Return to the Summary Table.

This register identifies the peripheral and its exact version.

Figure 15-6. 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 15-8. 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. 0h = Smallest value FFFFh = Highest possible value
15-12	FEATUREVER	R	8h	Feature Set for the module *instance* 0h = Smallest value 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 0h = Smallest value Fh = Highest possible value
7-4	MAJREV	R	1h	Major rev of the IP 0h = Smallest value Fh = Highest possible value
3-0	MINREV	R	0h	Minor rev of the IP 0h = Smallest value Fh = Highest possible value

15.3.6 CRCCTRL (Offset = 1100h) [Reset = 00000000h]

CRCCTRL is shown in Figure 15-7 and described in Table 15-9.

Return to the Summary Table.

CRC Control Register. Configuration control of the CRC.

Figure 15-7. 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 15-9. 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. 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. B1 is returned as B0 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 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 0h = Output byteswapping is disabled 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. 0h = LSB is lowest memory address and first to be processed. 1h = LSB is highest memory address and last to be processed.

Table 15-9. 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. 0h = Bit order is not reversed. 1h = Bit order is reversed.
0	POLYSIZE	R/W	0h	This bit indicates which CRC calculation is performed by the generator. 0h = CRC-32 ISO-3309 calulation is performed 1h = CRC-16 CCITT is performed

15.3.7 CRCSEED (Offset = 1104h) [Reset = 00000000h]

CRCSEED is shown in Figure 15-8 and described in Table 15-10.

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 15-8. 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 15-10. CRCSEED Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SEED	W	0h	Seed Data 00000000h = Mnimum value FFFFFFFFh = Maximum value

15.3.8 CRCIN (Offset = 1108h) [Reset = 00000000h]

CRCIN is shown in Figure 15-9 and described in Table 15-11.

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 15-9. 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 15-11. CRCIN Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	W	0h	Input Data 00000000h = Mnimum value FFFFFFFFh = Maximum value

15.3.9 CRCOUT (Offset = 110Ch) [Reset = 00000000h]

CRCOUT is shown in Figure 15-10 and described in Table 15-12.

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 15-10. 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 15-12. CRCOUT Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESULT	R	0h	Result 00000000h = Mnimum value FFFFFFFFh = Maximum value

15.3.10 CRCPOLY (Offset = 1110h) [Reset = 00000000h]

CRCPOLY is shown in Figure 15-11 and described in Table 15-13.

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 15-11. 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 15-13. CRCPOLY Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	04C11DB7h	Polynomial definition	

15.3.11 CRCIN_IDX[y] (Offset = 1800h + formula) [Reset = 00000000h]

CRCIN_IDX[y] is shown in Figure 15-12 and described in Table 15-14.

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 15-12. 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 15-14. CRCIN_IDX[y] Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	HW	0h	Input Data 00000000h = Minimum value FFFFFFFFh = Maximum value

Chapter 16

Keystore

16.1 Overview.....882
16.2 Detailed Description.....882
16.3 KEYSTORECTL Registers.....883

16.1 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.

16.2 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

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.

16.3 KEYSTORECTL Registers

Table 16-1 lists the memory-mapped registers for the KEYSTORECTL registers. All register offset addresses not listed in Table 16-1 should be considered as reserved locations and the register contents should not be modified.

Table 16-1. 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 16-2 shows the codes that are used for access types in this section.

Table 16-2. 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

16.3.1 CFG (Offset = 1100h) [Reset = 00000001h]

CFG is shown in Figure 16-1 and described in Table 16-3.

Return to the Summary Table.

Key-store configuration register.

Figure 16-1. 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 16-3. 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. Can not exceed the total number of slots present in the hardware / 2. For example, if SYS_N_SLOTS = 4, then at most 2 256-bit keys can be held in the key-store. Incorrect setting of this field will be reported via STATUS register 0h = No 256-bit keys 1h = One 256-bit key 2h = Two 256-bit keys 3h = Three 256-bit keys 4h = Four 256-bit keys

16.3.2 KEYWR (Offset = 1104h) [Reset = 00000000h]

KEYWR is shown in Figure 16-2 and described in Table 16-4.

Return to the Summary Table.

Register to configure a key write operation

Figure 16-2. 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 16-4. 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. NOTE: SW needs to ensure that it is writing to the correct slots. The slot numbering is from 0 through SYS_N_SLOTS-1. Each slot is a 128-bit slot. Therefore, when writing a 256-bit key, two slots will need to budgeted. The selected slot for the 256-bit key needs to be an even numbered slot (0 or 2). Also all the 256-bit keys need to be stored in lower numbered slots and 128-bit keys stored in higher numbered slots. 0h = Slot 0 1h = Slot 1 2h = Slot 2 3h = Slot 3 4h = Slot 4 5h = Slot 5 6h = Slot 6 7h = Slot 7
3	RESERVED	R	0h	
2-0	KEYSZSEL	R/W	0h	Key size selection. Selection of 128 or 256 bit keys 0h = 256 bit key 1h = 128 bit key

16.3.3 KEYRD (Offset = 1108h) [Reset = 00000000h]

KEYRD is shown in Figure 16-3 and described in Table 16-5.

Return to the Summary Table.

Register to configure a key transfer operation - transfer from keystore into a crypto engine

Figure 16-3. 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 16-5. 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. NOTE: SW needs to ensure that it is reading from the correct slots. The slot numbering is from 0 through SYS_N_SLOTS-1. Each slot is a 128-bit slot. Therefore, when reading a 256-bit key, two adjacent slots will be read. 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 0h = 256 bit key 1h = 128 bit key

16.3.4 STATUS (Offset = 110Ch) [Reset = 00020001h]

STATUS is shown in Figure 16-4 and described in Table 16-6.

Return to the Summary Table.

Status register

Figure 16-4. 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 16-6. 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 0h = Two slots 1h = Three slots 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 0h = Valid configuration 1h = Key-store has not been configured. NK256 has not been set. 2h = Invalid value for NK256 field in CFG. 3h = Busy receiving a key deposit 4h = Busy transmitting a key to a crypto engine 5h = Invalid key slot selection for writing 6h = Invalid key slot selection for reading

16.3.5 KEYIN (Offset = 1110h) [Reset = 00000000h]

KEYIN is shown in Figure 16-5 and described in Table 16-7.

Return to the Summary Table.

Write 32-bit key value by writing to this register. This is not readable.

Figure 16-5. 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 16-7. KEYIN Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	W	0h	32-bit key data

16.3.6 KEYLOCK (Offset = 1114h) [Reset = 00000000h]

KEYLOCK is shown in Figure 16-6 and described in Table 16-8.

Return to the Summary Table.

Writing 1 to this will lock the key access to the particular one hot encoded slot

Figure 16-6. 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 16-8. 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.

Chapter 17

TRNG

The true random number generator (TRNG) block is an entropy source which can be used to generate random bit sequences.

17.1 TRNG Overview 891
17.2 TRNG Operation 891
17.3 TRNG Registers 898

17.1 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

17.2 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 17-1 shows the TRNG blocks.

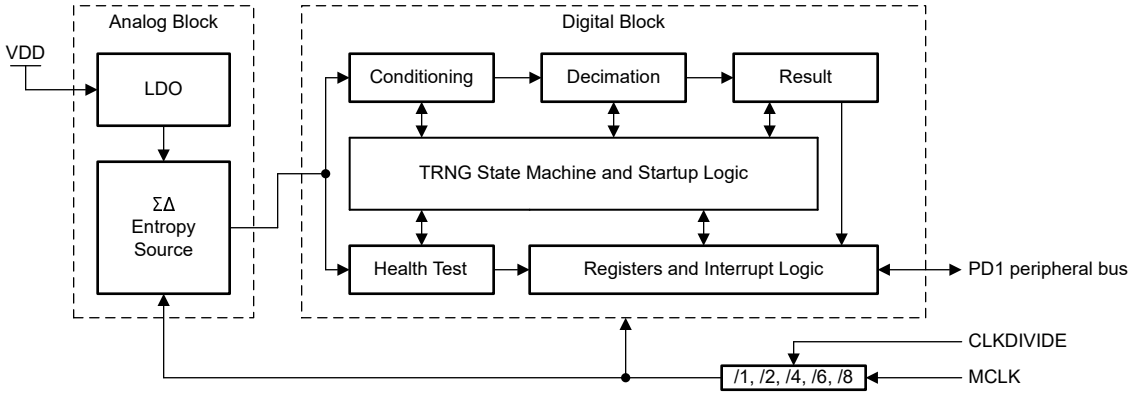


Figure 17-1. TRNG Block Diagram

17.2.1 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.

17.2.2 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 13 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.

$$t_{GENERATE}=(32*(DECIM\_RATE+1) / f_{TRNG} \quad (13)$$

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.

17.2.3 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.

17.2.4 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.

17.2.4.1 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

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.**

17.2.4.2 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

17.2.4.3 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.

17.2.4.3.1 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.

17.2.4.3.2 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

in Table 17-1 are violated. For example, the test will fail if more than 912 samples within a 1024 sample window are the same bit value.

Table 17-1. 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 17-1 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.

17.2.4.3.3 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.

17.2.5 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.

17.2.5.1 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 17-2.

Table 17-2. 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.

17.2.5.1.1 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 17-3.

Table 17-3. 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 17-2.

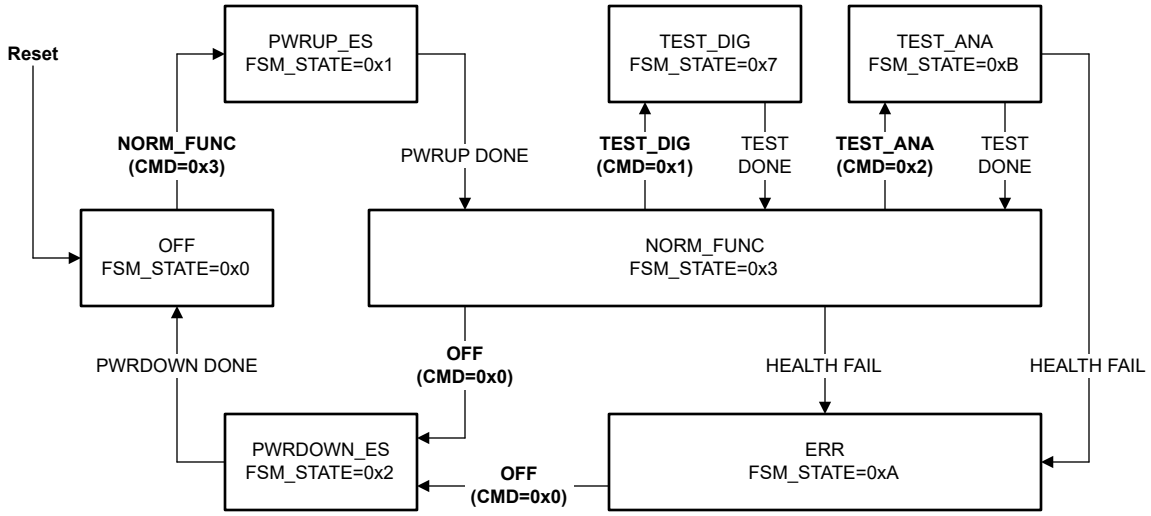


Figure 17-2. TRNG State Flow

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.

17.2.5.2 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:

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, and 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, random bits are 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.

17.2.5.3 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 17-4 lists the TRNG events.

Table 17-4. 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

17.2.5.3.1 CPU Interrupt Event Publisher (CPU_INT)

The TRNG module provides four interrupt sources which can be configured to source a CPU interrupt event. Table 17-5 lists the TRNG interrupt conditions.

Table 17-5. 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 8.2.5 for guidance on configuring these registers for CPU interrupts.

17.3 TRNG Registers

Table 17-6 lists the memory-mapped registers for the TRNG registers. All register offset addresses not listed in Table 17-6 should be considered as reserved locations and the register contents should not be modified.

Table 17-6. 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 17-7 shows the codes that are used for access types in this section.

Table 17-7. TRNG Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
WK	W K	Write Write protected by a key
Reset or Default Value		
-n		Value after reset or the default value

17.3.1 PWREN (Offset = 800h) [Reset = 00000000h]

PWREN is shown in Figure 17-3 and described in Table 17-8.

Return to the Summary Table.

Register to control the power state

Figure 17-3. 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 17-8. PWREN Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to allow Power State Change 26h = KEY to allow write access to this register
23-1	RESERVED	R/W	0h	
0	ENABLE	R/WK	0h	Enable the power KEY must be set to 26h to write to this bit. 0h = Disable Power 1h = Enable Power

17.3.2 RSTCTL (Offset = 804h) [Reset = 00000000h]

RSTCTL is shown in Figure 17-4 and described in Table 17-9.

Return to the Summary Table.

Register to control reset assertion and de-assertion

Figure 17-4. 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 17-9. RSTCTL Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	Unlock key 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 KEY must be set to B1h to write to this bit. 0h = Writing 0 has no effect 1h = Clear reset sticky bit
0	RESETASSERT	WK	0h	Assert reset to the peripheral KEY must be set to B1h to write to this bit. 0h = Writing 0 has no effect 1h = Assert reset

17.3.3 STAT (Offset = 814h) [Reset = 00000000h]

STAT is shown in Figure 17-5 and described in Table 17-10.

Return to the Summary Table.

peripheral enable and reset status

Figure 17-5. 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 17-10. 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 0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register 1h = The peripheral was reset since the last bit clear
15-0	RESERVED	R	0h	

17.3.4 IIDX (Offset = 1020h) [Reset = X]

IIDX is shown in Figure 17-6 and described in Table 17-11.

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 17-6. 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 17-11. IIDX Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	X	
7-0	STAT	R	0h	Interrupt index status 0h = No bit is set means there is no pending interrupt request 1h = Indicates that a health test has failed. The TRNG is in an error state until the interrupt is cleared. 2h = Indicates that the just issued command was rejected and is not being performed. 3h = Indicates that the current command/mode is done. This may have different meanings based on the mode: OFF --> Power has been turned off PWRUP_DIG --> Digital powerup tests are done PWRUP_ANA --> Analog powerup tests are done NORM_FUNC --> No IRQ, since mode runs indefinitely until a new command is issued 4h = Indicates that the captured word buffer is ready to be copied to memory

17.3.5 IMASK (Offset = 1028h) [Reset = X]

IMASK is shown in Figure 17-7 and described in Table 17-12.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt mask is set.

Figure 17-7. 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 17-12. 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. 0h = Interrupt is masked out 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 0h = Interrupt is masked out 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. 0h = Interrupt is masked out 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. 0h = Interrupt is masked out 1h = Interrupt will request an interrupt service routine and corresponding bit in MIS will be set

17.3.6 RIS (Offset = 1030h) [Reset = X]

RIS is shown in Figure 17-8 and described in Table 17-13.

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 17-8. 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 17-13. 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. 0h = IRQ_CAPTURED_READY did not occur 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. 0h = IRQ_CMD_DONE did not occur 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. 0h = IRQ_CMD_FAIL did not occur 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. 0h = IRQ_CAPTURED_READY did not occur 1h = IRQ_CAPTURED_READY occurred

17.3.7 MIS (Offset = 1038h) [Reset = X]

MIS is shown in Figure 17-9 and described in Table 17-14.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Figure 17-9. 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 17-14. 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. 0h = IRQ_CAPTURED_READY did not request an interrupt service routine 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. 0h = IRQ_CAPTURED_READY did not request an interrupt service routine 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. 0h = IRQ_CMD_FAIL did not request an interrupt service routine 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. 0h = IRQ_CAPTURED_READY did not request an interrupt service routine 1h = IRQ_CAPTURED_READY requests an interrupt service routine

17.3.8 ISET (Offset = 1040h) [Reset = 00000000h]

ISET is shown in Figure 17-10 and described in Table 17-15.

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 17-10. 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 17-15. 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. 0h = Writing a 0 has no effect 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. 0h = Writing a 0 has no effect. 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. 0h = Writing a 0 has no effect. 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. 0h = Writing a 0 has no effect 1h = RIS bit corresponding to HEALTH_FAIL is set

17.3.9 ICLR (Offset = 1048h) [Reset = 00000000h]

ICLR is shown in Figure 17-11 and described in Table 17-16.

Return to the Summary Table.

Write a 1 to clear corresponding Interrupt.

Figure 17-11. 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 17-16. 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. 0h = Writing a 0 has no effect 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. 0h = Writing a 0 has no effect. 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. 0h = Writing a 0 has no effect. 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. 0h = Writing a 0 has no effect 1h = RIS bit corresponding to CAPTURED_READY is cleared

17.3.10 DESC (Offset = 10FCh) [Reset = 05110000h]

DESC is shown in Figure 17-12 and described in Table 17-17.

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 17-12. 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 17-17. 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

17.3.11 CTL (Offset = 1100h) [Reset = X]

CTL is shown in Figure 17-13 and described in Table 17-18.

Return to the Summary Table.

Affects various parameters of the TRNG system

Figure 17-13. 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 17-18. 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 b00 = Disabled b01 = rise at 10us, fall at 50us b10 = rise at 10us, fall at 70us (default) b11 = rise at 10us, fall at 90us
18-17	PWRUP_PCHRG_CFG	R/W	2h	Configure PCHARGE sequence length b00 = Disabled b01 = 20 us PCHARGE b10 = 30 us PCHARGE (default) 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 0x0 = Decimation by 1 (no decimation) 0x1 = Decimation by 2 (Skip every other sample) … 0x7 = Decimation by 8 (Take every 8th sample)
7-2	RESERVED	R/W	X	

Table 17-18. 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. 00 --> OFF 01 --> PWRUP_DIG 10 --> PWRUP_ANA 11 --> NORM_FUNC 0h = Turns the power off of the analog source and clocks the digital interface 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 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 3h = Normal operating mode for TRNG. All components are turned on.

17.3.12 STAT (Offset = 1104h) [Reset = X]

STAT is shown in Figure 17-14 and described in Table 17-19.

Return to the Summary Table.

Status register that informs health test result, last issued command, and current FSM state

Figure 17-14. 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 17-19. 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 States: 0000: OFF 0001: PWRUP_ES 0011: NORM_FUNC 0111: TEST_DIG 1011: TEST_ANA 1010: ERROR 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. 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. 00 --> OFF 01 --> PWRUP_DIG 10 --> PWRUP_ANA 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.

17.3.13 DATA_CAPTURE (Offset = 1108h) [Reset = 00000000h]

DATA_CAPTURE is shown in Figure 17-15 and described in Table 17-20.

Return to the Summary Table.

Captured data from decimation block

Figure 17-15. 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 17-20. DATA_CAPTURE Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BUFFER	R	0h	Captured Data from the Decimation Block

17.3.14 TEST_RESULTS (Offset = 110Ch) [Reset = X]

TEST_RESULTS is shown in Figure 17-16 and described in Table 17-21.

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 17-16. 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 17-21. 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 Bit 0 - decim_test0 (decim = 0x0) Bit 1 - decim_test1 (decim = 0x1) ...

17.3.15 CLKDIVIDE (Offset = 1110h) [Reset = 00000000h]

CLKDIVIDE is shown in Figure 17-17 and described in Table 17-22.

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 17-17. 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 17-22. 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 0h = Do not divide clock source 1h = Divide clock source by 2 3h = Divide clock source by 4 5h = Divide clock source by 6 7h = Divide clock source by 8

Chapter 18

HSADC

The analog-to-digital converter (ADC) module described in this chapter is a high speed 12-bit SAR ADC.

18.1 Introduction....916
18.2 HSADC Operation....917
18.3 ADC_LITE_REGS Registers....935
18.4 ADC_LITE_RESULT_REGS Registers....1044

18.1 Introduction

The ADC module is a successive approximation (SAR) style ADC . The ADC is composed of a core and a wrapper. The core is composed of the analog circuits which include the channel select MUX, the sample-and-hold (S/H) circuit, the successive approximation circuits, voltage reference circuits, and other analog support circuits. The wrapper is composed of the digital circuits that configure and control the ADC. These circuits include the logic for programmable conversions, result registers, interfaces to analog circuits, interfaces to the peripheral buses, post-processing circuits, and interfaces to other on-chip modules.

Each ADC module consists of a single sample-and-hold (S/H) circuit. The ADC module is designed to be duplicated multiple times on the same chip, allowing simultaneous sampling or independent operation of multiple ADCs. The ADC wrapper is start-of-conversion (SOC) based (see Section 18.2.2).

18.1.1 Features

Below are the high-level features supported by HS-ADC. For detailed list of features supported, refer to device specific datasheet.

- Must support total 32 channels/instance
- 4 sequencer/instance
- Window comparator
- Support for internal reference and external reference
- MSP style of DMA/Interrupt triggers
- ADC Conversion sequencer

18.1.2 Block Diagram

Figure 18-1 shows the block diagram for the ADC core and ADC wrapper.

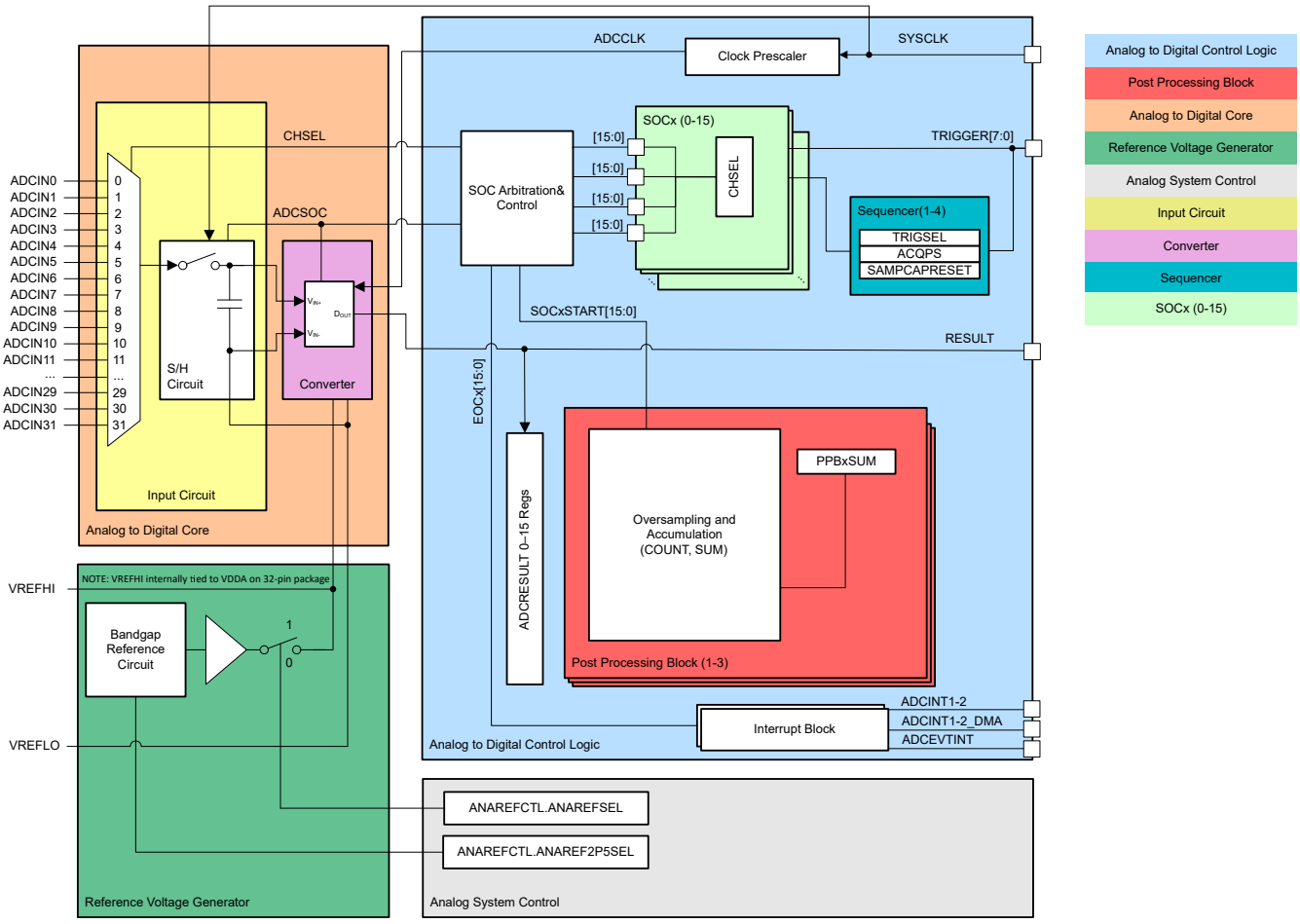


Figure 18-1. ADC Module Block Diagram

Note

- The ADC block diagram reflects the number of ADC channels internally configurable on the device. The actual number of available external ADC inputs varies depending on device part number and package type.

18.2 HSADC Operation

This section provides details on the different functionality of the HSADC block that are used for analog to digital conversion operations.

18.2.1 ADC Configurability

Some ADC configurations are individually controlled by the SOCs, while others are globally controlled per ADC module. Table 18-1 summarizes the basic ADC options and the level of configurability. The subsequent sections discuss these configurations.

Table 18-1. ADC Options and Configuration Levels

Options	Configurability
Clock	Per module(1)
Reference Voltage Source	Not configurable (external or internal reference only)
Trigger Source	Per Sequencer
Converted Channel	Per SOC
Acquisition Window Duration	Per Sequencer
EOC Location	Per module

(1) Writing these values differently to different ADC modules can cause the ADCs to operate asynchronously. See *Ensuring Synchronous Operation* for guidance on when the ADCs are operating synchronously or asynchronously.

18.2.1.1 ADC Clock Configuration

The base ADC clock is provided directly by the system clock. is used to generate the ADC acquisition window. The register ADCCTL2 has a PRESCALE field that determines the ADCCLK. ADCCLK is used to clock the converter, and is only active during the conversion phase. At all other times, including during the sample-and-hold window, the ADCCLK signal is gated off.

Choosing an Acquisition Window Duration.

Note

To determine an appropriate value for ADCCTL2.PRESCALE, see the device data sheet for the maximum allowable ADCCLK frequencies.

18.2.1.2 Voltage Reference

The M33 HSADC utilizes the VREF moudle to provide the VREFHI and VREFLO signals. The M33 VREF module can provide internal voltage references as well as external references. Refer to Chapter 19 for more details.

18.2.1.3 Signal Mode

In single-ended mode, the input voltage to the converter is sampled through a single pin (ADCINx), referenced to VREFLO.

18.2.1.3.1 Expected Conversion Results

Based on a given analog input voltage, the expected digital conversion is given in Table 18-2. Fractional values are truncated.

Table 18-2. Analog to 12-bit Digital Formulas

Analog Input	Digital Result
when ADCINy ≤ VREFLO	ADCRESULTx = 0
when VREFLO < ADCINy < VREFHI	$ADCRESULTx = 4096 \left(\frac{ADCINy - VREFLO}{VREFHI - VREFLO} \right)$
when ADCINy ≥ VREFHI	ADCRESULTx = 4095

18.2.1.3.2 Interpreting Conversion Results

Based on a given ADC conversion result, the corresponding analog input is given in Table 18-3. This corresponds to the center of the possible range of analog voltages that can produce this conversion result.

Table 18-3. 12-Bit Digital-to-Analog Formulas

Digital Value	Analog Equivalent
when ADCRESULTy = 0	$\text{ADCINx} \leq \text{VREFLO}$ (14)
when 0 < ADCRESULTy < 4095	$\text{ADCINx} = (\text{VREFHI} - \text{VREFLO})\left(\frac{\text{ADCRESULTy}}{4096}\right) + \text{VREFLO}$ (15)
when ADCRESULTy = 4095	$\text{ADCINx} \geq \text{VREFHI}$ (16)

18.2.2 SOC Principle of Operation

The ADC conversion sequencing is accomplished through configurable start-of-conversions (SOCs). Each M33 SOC is a configuration set defining the single conversion of a single channel. In that set, there are two configurations: the channel to convert and whether or not limit comparison is enabled. Upon receiving the start signal for an SOC, the wrapper makes sure that the specified channel is captured and the limit comparison is performed if enabled.

Multiple SOCs can be configured for the same channel as desired. Configuring multiple SOCs to use the same trigger allows the trigger to generate a sequence of conversions. Configuring multiple SOCs to use the same trigger and channel allows for oversampling.

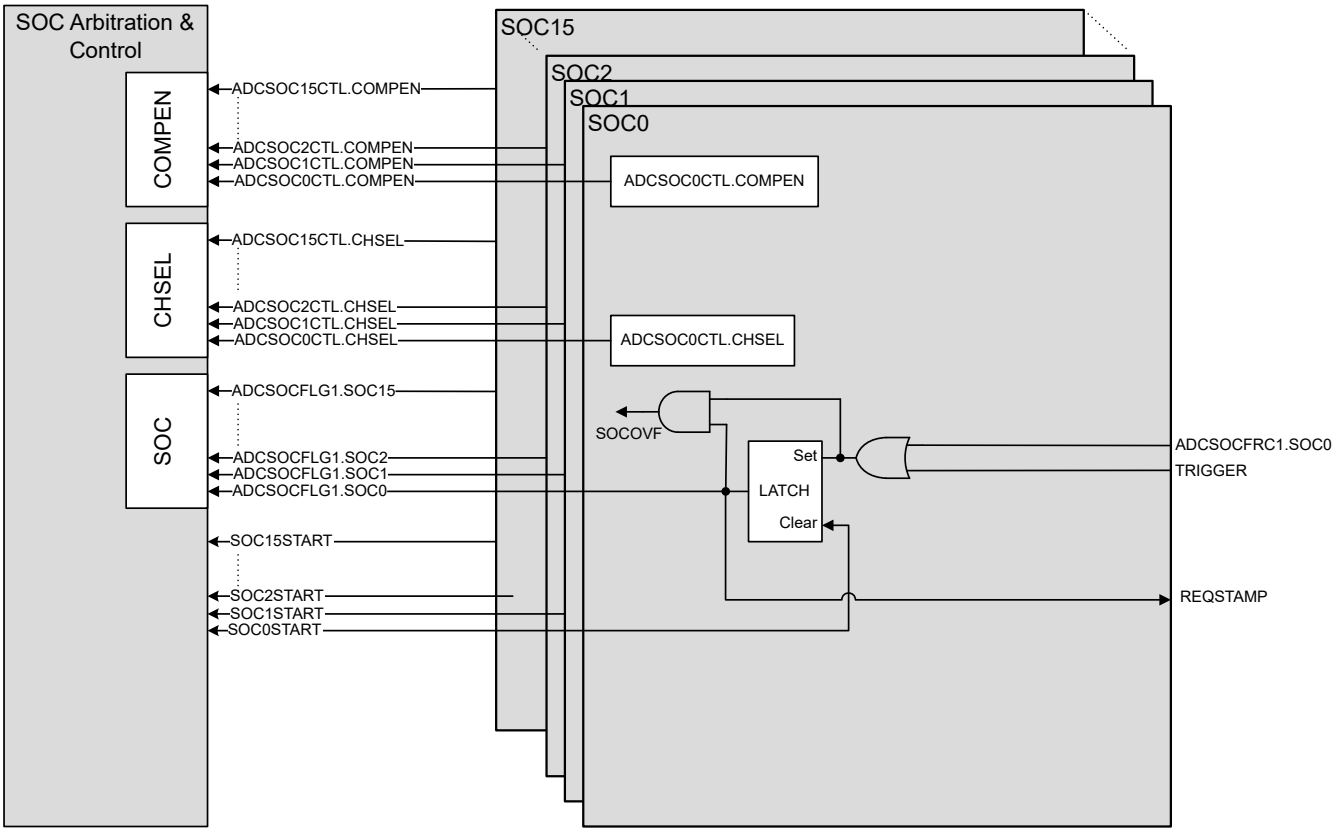


Figure 18-2. SOC Block Diagram

18.2.2.1 HSADC Sequencer Operation

The M33 ADC supports up to four sequencers, which can be used to sequence the order in which ADC channels are converted. Each sequencer is associated with multiple SOCs, and the number of SOCs within a sequencer is configured using the ADCSEQ{#}CONFIG.SEQSTART and ADCSEQ{#+1}CONFIG.SEQSTART-1 or ADCSEQCTL.SEQEND registers. Sequencers are enabled using the ADCSEQ{#}CONFIG.SEQENABLE register.

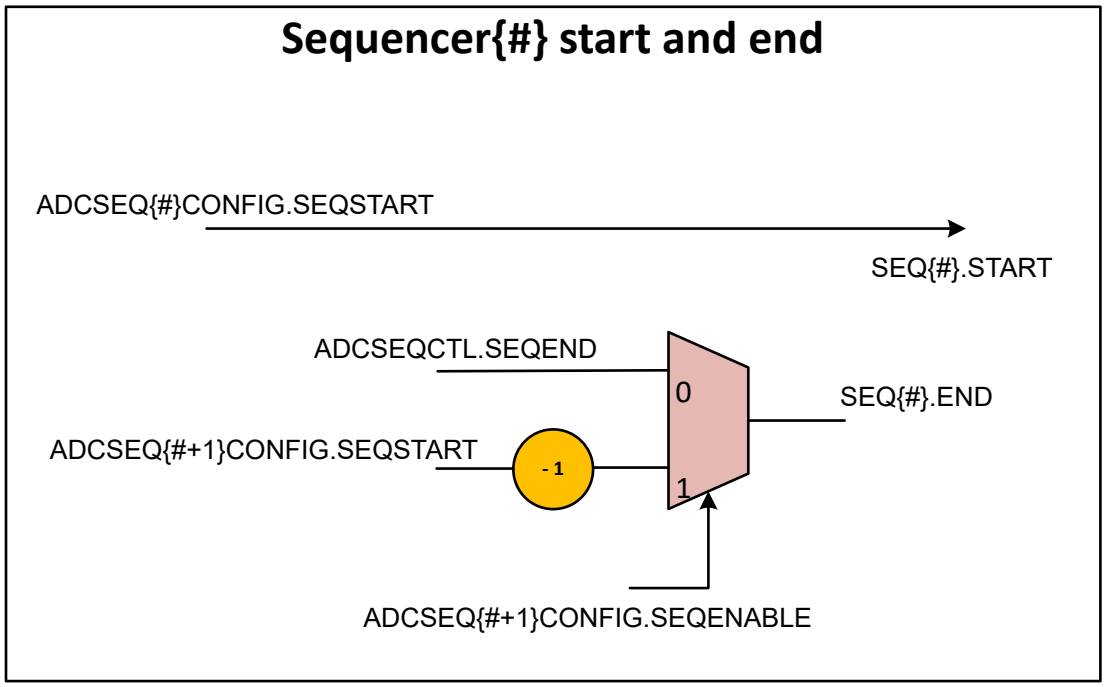


Figure 18-3. Sequencer start and end

Once a trigger is received for a sequencer and the ADC Wrapper is ready for arbitration, the sequencer will start converting in ascending order of SOCs. The first SOC in the sequence to convert will be determined by ADCSEQ{#}CONFIG.SEQSTART, followed by subsequent SOCs until the end of the sequencer is reached.

Multiple sequencers can be configured to use the same trigger if desired. If multiple sequencers are triggered at the same time, the conversions within the sequence happens in the order of priority: SEQ1 followed by SEQ2, SEQ3, and SEQ4.

If a trigger for higher priority sequence occurs while a lower priority sequence is ongoing, the higher priority sequence can either be serviced as soon as the ongoing SOC conversion is complete or serviced after the lower priority sequence is complete. This can be configured using ADCSEQCTL.SEQPREEMPT register.

The ADCSEQCTL.SEQPREEMPT register configuration is as follows

- Pre-empt disabled (0x): The ongoing sequence will fully complete before switching to higher priority sequence.
- Pre-empt enabled and will not restart aborted SEQ (10): The ongoing sequence will be discarded (after completing any ongoing conversion) and arbitration will switch to higher priority sequence. The discarded/aborted sequence will not be re-started after completing higher priority sequence.
- Pre-empt enabled and will restart aborted SEQ (11): The ongoing sequence will be discarded (after completing any ongoing conversion) and arbitration will switch to higher priority sequence. The discarded/aborted sequence will be re-started from the beginning of the sequence after completing all the higher priority sequence(s).

Sequencer{#} enable and overflow logic illustrates the logical representation of sequencer{#} enable and overflow logic, which generates the overflow condition when the new trigger is received while a previous trigger is ongoing or pending.

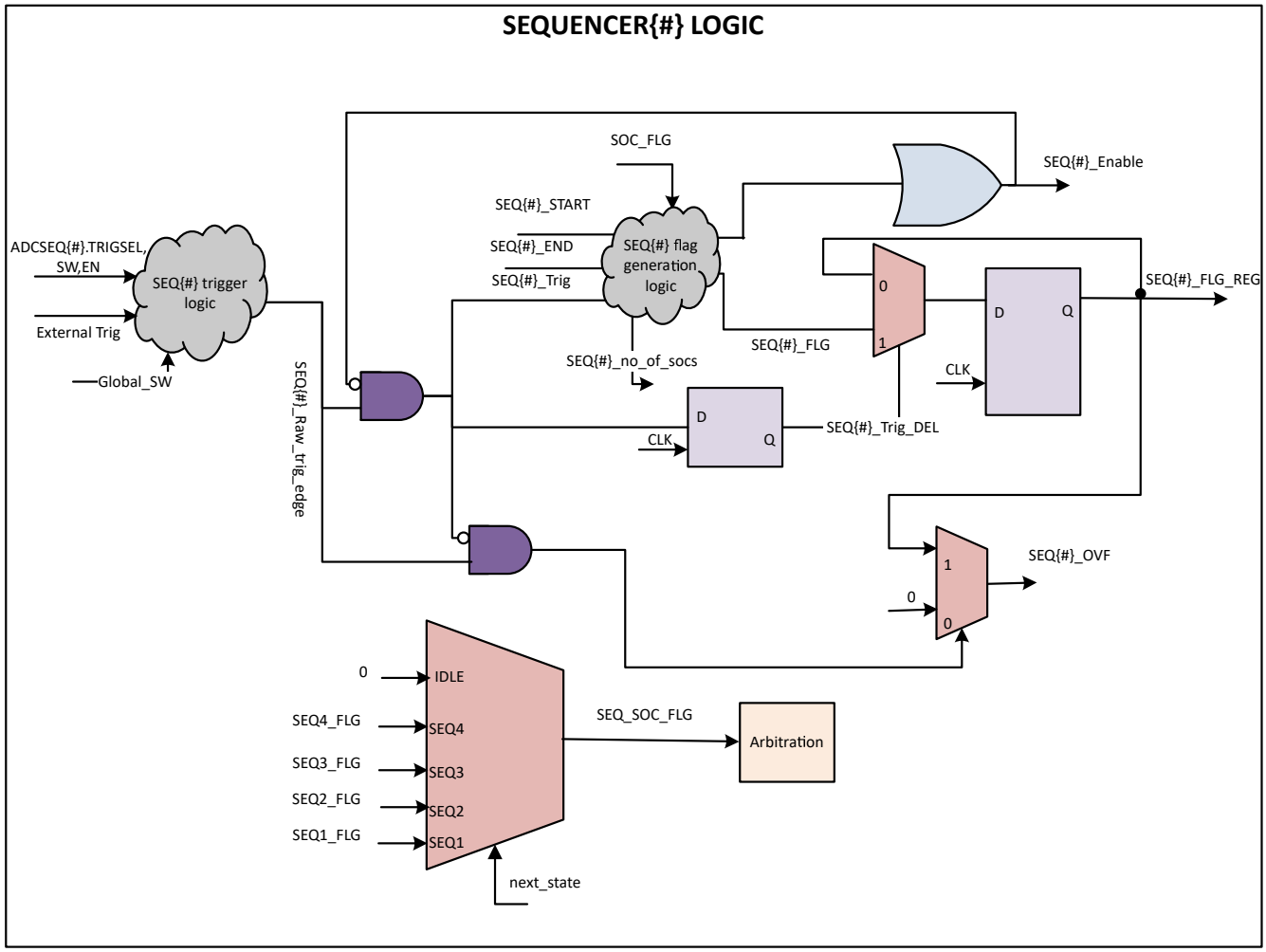


Figure 18-4. Sequencer{#} enable and overflow logic

Sequencer pre-empt restart & discard logic illustrates the logical representation of sequencer{#} restart & discard logic.

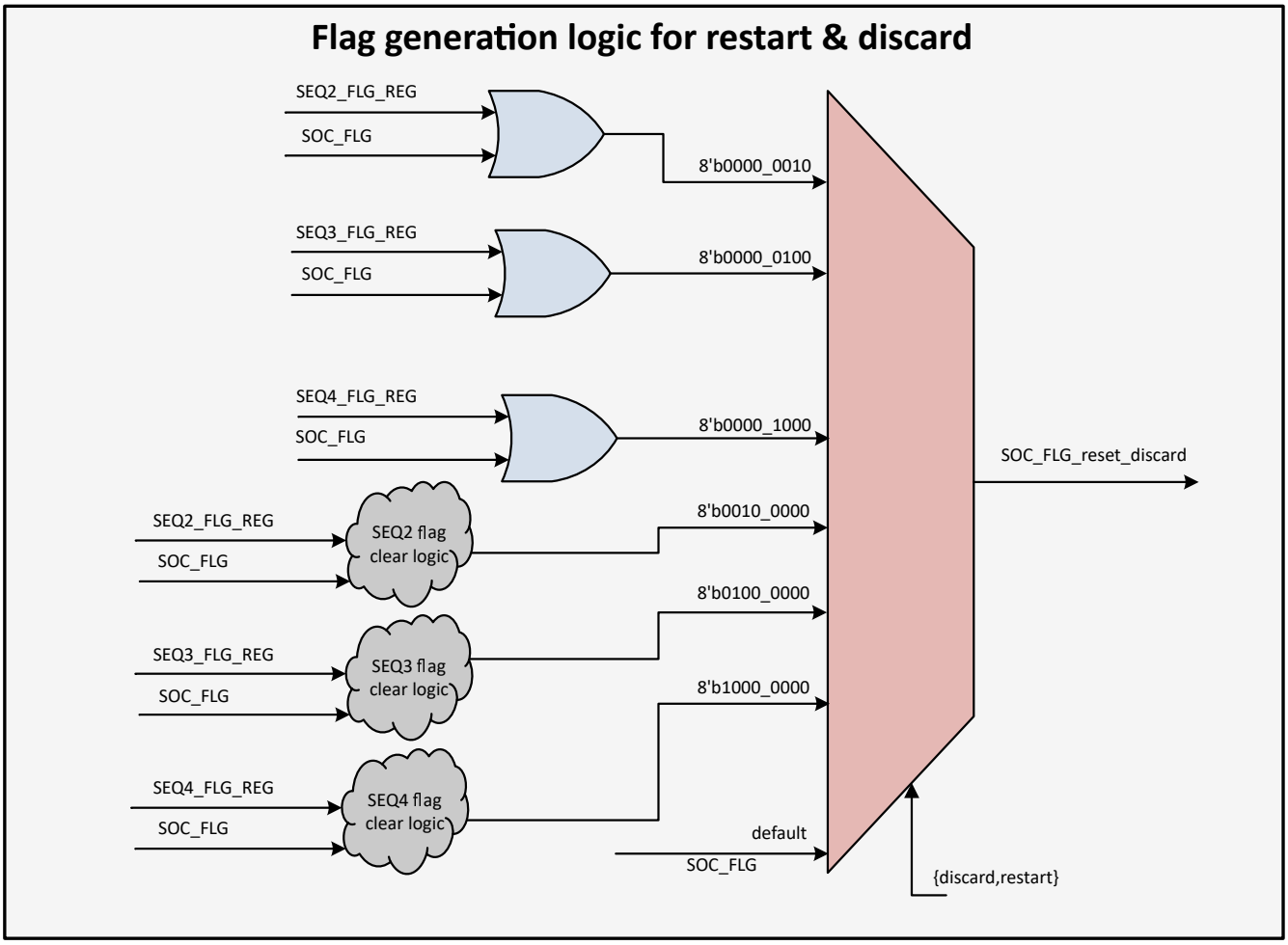


Figure 18-5. Sequencer pre-empt restart & discard logic

Additionally, oversampling and averaging can be enabled for the SOCs within the sequence using the Post Processing Blocking (PPB) averaging feature. Each sequencer is associated with the corresponding PPB block. PPB{#}LIMIT defines the number of samples per SOCs.

If oversampling is enabled, each SOC will be sampled and converted 'N' times (N – total number of samples configured) before moving to the next SOC.

Each sequencer is associated with the corresponding ADCPPB{#}TRIPHI and ADCPPB{#}TRIPLO registers. This can be used to compare whether ADCRESULT is within the configured TRIPHI and TRIPLO limits or exceeded either of the limits.

Since the limit comparator settings are common across all SOCs within a Sequencer, limit comparison for a given SOC can be enabled or disabled using the ADCSOC{#}CTL.COMPEN field of the corresponding SOC.

18.2.2.2 SOC Configuration

Each SOC has a configuration register, ADCSOCxCTL. Within this register, SOCx can be configured to select a specific channel to convert and to enable or disable the threshold comparator

18.2.2.3 Trigger Operation

Each SOC can be configured to start on one of many input triggers. The primary trigger select for SEQx is in the ADCSEQxCONFIG.TRIGSEL register, which can select between:

- Disabled (Software only)
- Generic Subscribers (GEN_SUB) 0-3

18.2.2.4 ADC Acquisition (Sample and Hold) Window

External signal sources vary in the ability to drive an analog signal quickly and effectively. To achieve rated resolution, the signal source needs to charge the sampling capacitor in the ADC core to within 0.5 LSBs of the signal voltage. The acquisition window is the amount of time the sampling capacitor is allowed to charge and is configurable per SOCx by the register.

ACQPS is a 9-bit 8-bit register field that can be set to a value between 0 and 255. 511, resulting in an acquisition window duration of: The 2 upper bits (ADCSOCxCTL.ACQPS[7:6]) configure the base duration and cycle prescalers. The base duration can be set to values of 0, 64, 192, or 448 with corresponding cycle prescalers of 1, 2, 4, and 16 respectively. The lower 6 bits (ADCSOCxCTL.ACQPS[5:0]) in the register field configure the additional cycles to be multiplied by the prescaler and added to the base duration.

Acquisition Window = (ACQPS + 1) ˣ (System Clock () cycle time)

Acquisition Window [SYSCLK cycles] = (Base duration) + ((Additional cycles + 1)*(Prescaler))

- The acquisition window duration is based on the System Clock (SYSCLK), not the ADC clock (ADCCLK).
- The selected acquisition window duration must be at least as long as one ADCCLK cycle.
- The data sheet specifies a minimum acquisition window duration (in nanoseconds). The user is responsible for selecting an acquisition window duration that meets this requirement.

18.2.2.5 Sample Capacitor Reset

In certain systems where the ADC successively samples multiple signal sources, memory crosstalk can occur. Memory crosstalk is the tendency of the ADC conversion to be pulled towards the value of the previous conversion, due to inadequate acquisition/settling time. This happens because the ADC sample capacitor voltage starts near the previously converted voltage, then settles towards the newly applied voltage on the current channel. If the acquisition window is not long enough for the sample capacitor to settle, this can result in some sample error reflected in the ADC conversion.

The 12-bit ADC modules in this device include a sample capacitor reset feature to help mitigate memory crosstalk. When sample capacitor reset is enabled, after every conversion, the sampling capacitor voltage is reset to the VREFLO either the VREFLO or VREFHI/2 voltage, depending on the configuration of the ADCSOCxCTL.SAMPCAPRESETSEL bit. This reset takes an extra ADCCLK cycle to complete. The sample capacitor reset function is inactive by default for each SOC. If desired, the application can activate sample capacitor reset by writing 0 to the SAMPCAPRESETDISABLE bit in the ADCSOCxCTL register. When sample capacitor reset is active, overall ADC throughput is slightly decreased due to the extra ADCCLK cycle in the conversion period.

18.2.2.6 ADC Input Models

For single-ended operation, the ADC input characteristics for values in the single-ended input model (see Figure 18-6) can be found in the device data sheet.

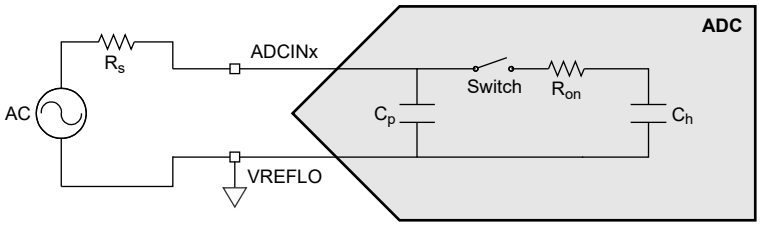


Figure 18-6. Single-Ended Input Model

These input models must be used along with actual signal source impedance to determine the acquisition window duration. See *Choosing an Acquisition Window Duration* for more information.

18.2.2.7 Channel Selection

Each SOC can be configured to convert any of the ADC channels. This behavior is selected for SOCx by the ADCSOCxCTL.CHSEL register. This is summarized in Table 18-4. For pin location of ADC inputs, refer to the device specific datasheet.

Table 18-4. Channel Selection of Input Pins

Input Mode	CHSEL	Input
Single-Ended	0	ADCIN0
	1	ADCIN1
	2	ADCIN2
	3	ADCIN3
	4	ADCIN4
	5	ADCIN5
	6	ADCIN6
	7	ADCIN7
	8	ADCIN8
	9	ADCIN9
	10	ADCIN10
	11	ADCIN11
	12	ADCIN12
	13	ADCIN13
	14	ADCIN14
	15	ADCIN15

18.2.3 EOC and Interrupt Operation

Each SOC has a corresponding end-of-conversion (EOC) signal. This EOC signal can be used to trigger an ADC interrupt. The ADC can be configured to generate the EOC pulse at either the end of the acquisition window or at the end of the voltage conversion. This is configured using the bit INTPULSEPOS in the ADCCTL1 register. See Section 18.2.6 for exact EOC pulse location.

Each ADC module has configurable ADC interrupts. These interrupts can be triggered by any of the EOC signals. The flag bit for each ADCINT can be read directly to determine if the associated SOC is complete or the interrupt can be passed on to the .

Note

The ADCCTL1.ADCBSY bit being clear does not indicate that all conversions in a set of SOCs have completed, only that the ADC is ready to process the next conversion. To determine if a sequence of SOCs is complete, link an ADCINT flag to the last SOC in the sequence and monitor that ADCINT flag.

Figure 18-7 shows a block diagram of the ADC interrupt structure.

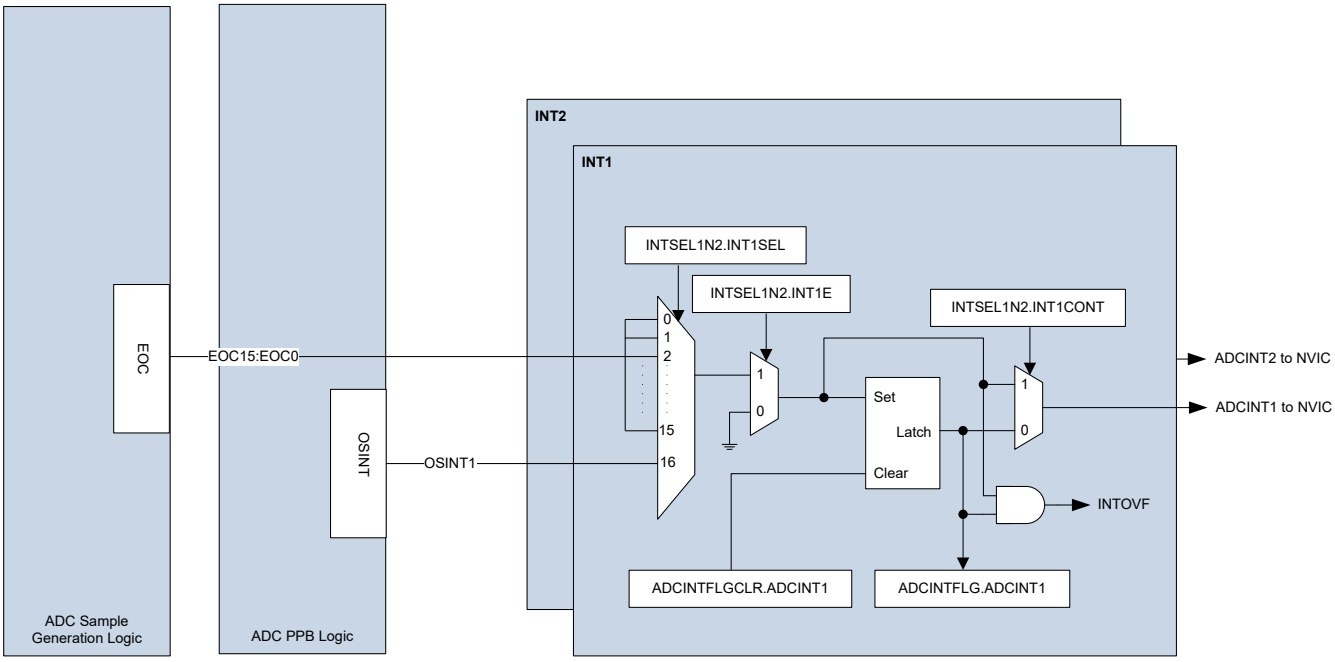


Figure 18-7. ADC End-of-Conversion (EOC) Signal Interrupts

18.2.3.1 Interrupt Overflow

If the EOC signal sets a flag in the ADCINTFLG register, but that flag is already set, an interrupt overflow occurs. By default, overflow interrupts are not passed on to the module. When an overflow occurs on a given flag in the ADCINTFLG register, the corresponding flag in the ADCINOVF register is set. This overflow flag is only used to detect that an overflow has occurred; the flag does not block further interrupts from propagating to the module.

When an ADC interrupt overflow occurs, the application must check the appropriate ADCINTOVF flag inside the ISR or in the background loop and take appropriate action when an overflow is detected.

18.2.3.2 Continue to Interrupt Mode

The INTxCONT bits in the ADCINTSEL1N2 and ADCINTSEL3N4 registers configure how interrupts are handled when an ADCINTFLG has not yet been cleared from a prior interrupt. This mode is disabled by default and additional overlapping interrupts are not issued to the . By activating this mode, ADC interrupts always reach the . If interrupts occur while ADCINTFLG is set, the ADCINTOVF register remains set regardless of the configuration of the INTxCONT bits.

18.2.3.3 Early Interrupt Configuration Mode

Enabling early interrupt mode can allow the application to enter the ADC interrupt service routine before the ADC results are ready. This allows the application to do any necessary pre-work so that the application can act on the ADC results immediately when the ADC results become available. If the timing of the early interrupt is too early, then the application needs to waste time until the updated ADC results become available. To prevent this situation, the time the ADC interrupt is entered in early interrupt mode is configurable by way of the DELAY field in the ADCINTCYCLE register.

- To use the configurable interrupt time, the ADC must be in early interrupt mode. To achieve this, clear the bit INTPULSEPOS to 0 in ADCCTL1.
- The DELAY value in the ADCINTCYCLE register sets the number of additional cycles after the falling edge of the SOC pulse before the ADCINT flag is set.
- If the value of DELAY goes beyond EOC, the ADC interrupt is generated along with EOC.
- Writing values to DELAY when INTPULSEPOS is set to 1 does not have any effect on the interrupt generation.

18.2.4 Post-Processing Blocks

Each ADC module contains post-processing blocks (PPB). These blocks can be associated with any of the RESULT registers using the ADCPPBxCONFIG.CONFIG bit field. The post-processing blocks have the ability to:

- Aggregate successive samples using sum , max, and min calculations
- Flag a zero-crossing point, with the option to trip a PWM and generate an interrupt
- Flag a high or low compare limit, with the option to trip a PWM and generate an interrupt

Figure 18-8 presents the structure of each PPB. Subsequent sections explain the use of each submodule.

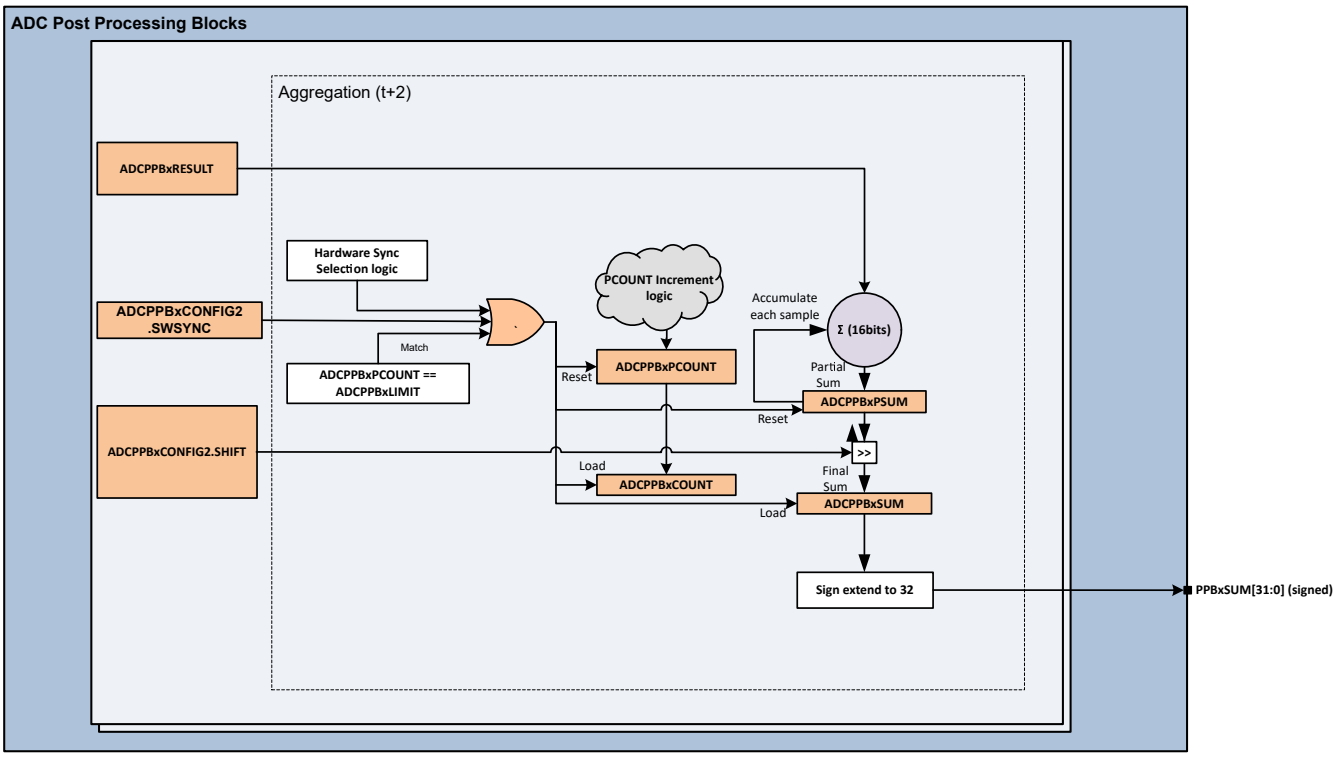


Figure 18-8. ADC Post-Processing Blocks (PPB) Block Diagram

18.2.4.1 PPB Limit Detection and Zero-Crossing Detection

Many applications perform a limit check against the ADC conversion results. The PPB can automatically perform a check against a high limit, a low limit, or whenever ADCPPBxRESULT changes sign. Based on these comparisons, the PPB can generate a trip to the PWM and an interrupt automatically, lowering the sample to ePWM latency and reducing software overhead. This functionality also enables safety-conscious applications to trip the ePWM based on an out-of-range ADC conversion without any CPU intervention.

To enable this functionality, first point the ADCPPBxCONFIG.CONFIG to the desired SOC, then write a value to one or both of the registers ADCPPBxTRIPHI.LIMITHI and ADCPPBxTRIPLO.LIMITLO (zero-crossing detection does not require further configuration). Whenever these limits are exceeded, the PPBxTRIPHI bit or PPBxTRIPLO bit is set in the ADCEVTSTAT register. Note that the PPBxZERO bit in the ADCEVTSTAT register is gated by end-of-conversion (EOC), not by the sign change in the ADCPPBxRESULT register. The ADCEVTCLR register has corresponding bits to clear these event flags. The ADCEVTSEL register has corresponding bits that allow the events to propagate through to the PWM. The ADCEVTINTSEL register has corresponding bits that allow the events to propagate through to the .

One interrupt is shared between all the PPBs for a given ADC module as shown in the following figure.

Note

- If different actions need to be taken for different PPB events from the same ADC module, then the ADCEVTINT ISR has to read the PPB event flags in the ADCEVTSTAT register to determine which event caused the interrupt.
- If different ePWM trips need to be generated separately for high compare, low compare, and zero-crossing, this can be achieved by pointing multiple PPBs to the same SOC.
- The zero-crossing detect circuit considers a result of zero to be positive.

18.2.4.1.1 PPB Digital Trip Filter

The ADC provides digital filters on the TRIPHI and TRIPLO signals to help prevent unwanted threshold trips. Each digital filter works on a window of FIFO samples (SAMPWIN) taken from the threshold comparator output. The filter output resolves to the majority value of the sample window, where majority is defined by the threshold (THRESH) value. If the majority threshold is not satisfied, the filter output remains unchanged. Figure 18-9 shows a block diagram of the PPB digital trip filter.

For proper operation, the value of THRESH must be greater than SAMPWIN / 2, and less than or equal to SAMPWIN.

A prescale function (CLKPRESCALE) determines the filter sampling rate. The filter FIFO captures one sample for every CLKPRESCALE cycles of SYSCLK. Old data from the FIFO is discarded.

The digital trip filters are disabled by default. To enable and configure the digital trip filters, write the desired values to the ADCPPBTRIPxFILCTL and ADCPPBTRIPxFILCLKCTL registers. Note that for SAMPWIN, THRESH and PRESCALE, the filter adds 1 to the value in the register field (for example, write 15 to SAMPWIN to enable a filter sample window of 16). To clear the FIFO and re-initialize the filter, write 1 to the FILINIT bit in the ADCPPBTRIPxFILCTL register.

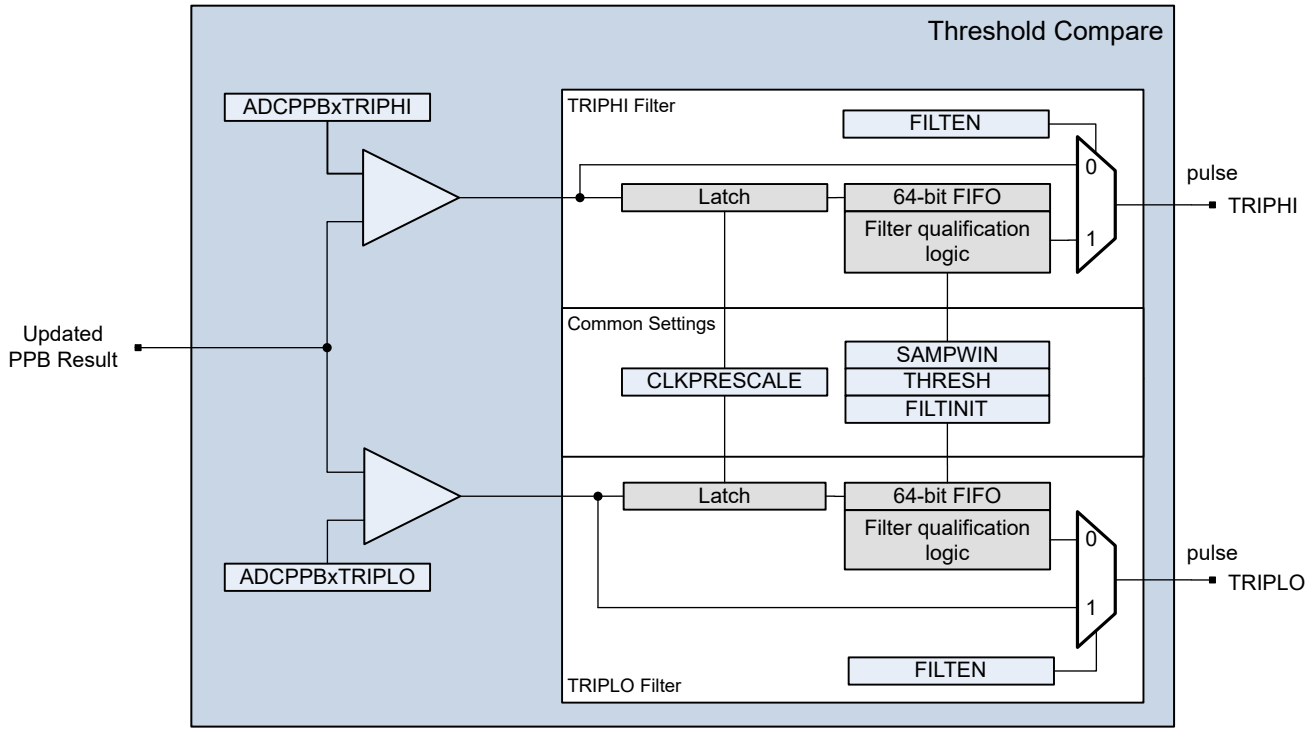


Figure 18-9. ADC PPB Limit Filter Logic

18.2.4.2 PPB Oversampling

This ADC has built-in support for oversampling in the post-processing block, including an accumulator, min/max for peak detection, and outlier removal. The oversampling support module exists at the output of the sample correction module, as shown in Figure 18-8. The oversampling module works by accumulating results in partial registers until either the sample count limit defined in the ADCPPBxLIMIT register is reached or the software forces a sync event by writing to the SWSYNC bit in the ADCPPBxCONFIG2 register.

18.2.4.2.1 Accumulation and Average Functions

At the end of each ADC sample conversion, the PPB updates the partial result registers ADCPPBxPSUM with the newly processed conversion result from the ADCPPBxRESULT register, and the partial conversion count register (ADCPPBxPCOUNT) increments by 1. When the partial conversion count equals the limit defined in ADCPPBxLIMIT, or the PPB receives a software sync signal, the PPB takes the following actions:

1. The PPB loads the values of the respective partial result registers into the final result registers ADCPPBxPSUM.
2. The PPB loads the partial count in ADCPPBxPCOUNT into the final conversion count register ADCPPBxCOUNT.
3. The partial count register and partial result registers reset to zero.
4. The ADC generates an oversampling interrupt (OSINTx) event pulse, which triggers a CPU interrupt if so configured in the ADCINTSEL1N2 or ADCINTSEL3N4 registers.

The PPB can also be configured to generate an oversampling interrupt when there is a hardware or software sync event. To trigger an OSINTx pulse when a sync event occurs, write 1 to the OSINTSEL bit in the ADCPPBxCONFIG2 register.

The PPB can automatically compute the average of the accumulated samples if ADCPPBxLIMIT is set to a power of 2 (up to a maximum of 1024 samples). To perform automatic averaging over 2^n samples, set the SHIFT field in the ADCPPBxCONFIG2 register to n. When this field is set, the PPB divides the value of ADCPPBxPSUM by 2^n before loading into ADCPPBxSUM.

To compute an average from the accumulated sum when the number of samples is not a power of 2, divide the value of ADCPPBxSUM by the value of ADCPPBxCOUNT using the CPU.

18.2.5 Power-Up Sequence

Upon device power-up or system level reset, the ADC is powered down and disabled. When powering up the ADC, the following sequence must be used:

1. Set the desired ADC clock divider in the PRESCALE field of ADCCTL2. Note that ADCCLK must be divided down to meet the maximum ADCCLK frequency provided in the device data sheet.
2. Power up the ADC by setting the ADCPWDNZ bit in ADCCTL1.
3. Allow a delay before sampling. See the data sheet for the necessary time.

If multiple ADCs are powered up simultaneously, steps 1 and step 3 can each be done for all ADCs in one write instruction. Also, only one delay is necessary as long as the delay occurs after all the ADCs have begun powering up.

18.2.6 ADC Timings

The process of converting an analog voltage to a digital value is broken down into an S+H phase and a conversion phase. The ADC sample and hold circuits (S+H) are clocked by while the ADC conversion process is clocked by ADCCLK. ADCCLK is generated by dividing down based on the PRESCALE field in the ADCCTL2 register.

The S+H duration is the value of the ACQPS field of the SOC being converted, plus one, times the period. The user must make sure that this duration exceeds both 1 ADCCLK period and the minimum S+H duration specified in the data sheet. The conversion time is approximately. See the timing diagrams and tables in Section 18.2.6.1 for exact timings.

18.2.6.1 ADC Timing Diagrams

The following diagrams show the ADC conversion timings for two SOCs given the following assumptions:

- SOC0 and SOC1 are configured to use the same trigger.
- No other SOCs are converting or pending when the trigger occurs.
- The round robin pointer is in a state that causes SOC0 to convert first.
- ADCINTSEL is configured to set an ADCINT flag upon end of conversion for SOC0 (whether this flag propagates through to the CPU to cause an interrupt is determined by the configurations in the module).

Table 18-5 describes the parameters in the following timing diagrams.

Table 18-6 and Table 18-7 lists the ADC timings.

Table 18-5. ADC Timing Parameter Descriptions

Parameter	Description
t_{SH}	The duration of the S+H window. At the end of this window, the value on the S+H capacitor becomes the voltage to be converted into a digital value. The duration is given by (ACQPS + 1) cycles. ACQPS can be configured individually for each SOC, so t_{SH} is not necessarily the same for different SOCs. **Note:** The value on the S+H capacitor is captured approximately 5ns before the end of the S+H window regardless of device clock settings.
t_{LAT}	The time from the end of the S+H window until the ADC results latch in the ADCRESULTx register. If the ADCRESULTx register is read before this time, the previous conversion results are returned.
t_{EOC}	The time from the end of the S+H window until the S+H window for the next ADC conversion can begin.
t_{INT}	The time from the end of the S+H window until an ADCINT flag is set (if configured). If the INTPULSEPOS bit in the ADCCTL1 register is set, t_{INT} coincides with the end of conversion (EOC) signal. If the INTPULSEPOS bit is 0, and the OFFSET field in the ADCINTCYCLE register is not 0, then there is a delay of OFFSET cycles before the ADCINT flag is set. This delay can be used to enter the ISR when the sample is ready. If the INTPULSEPOS bit is 0, t_{INT} coincides with the end of the S+H window. If t_{INT} triggers a read of the ADC result register (by triggering an ISR that reads the result), care must be taken to make sure the read occurs after the results latch (otherwise, the previous results are read).

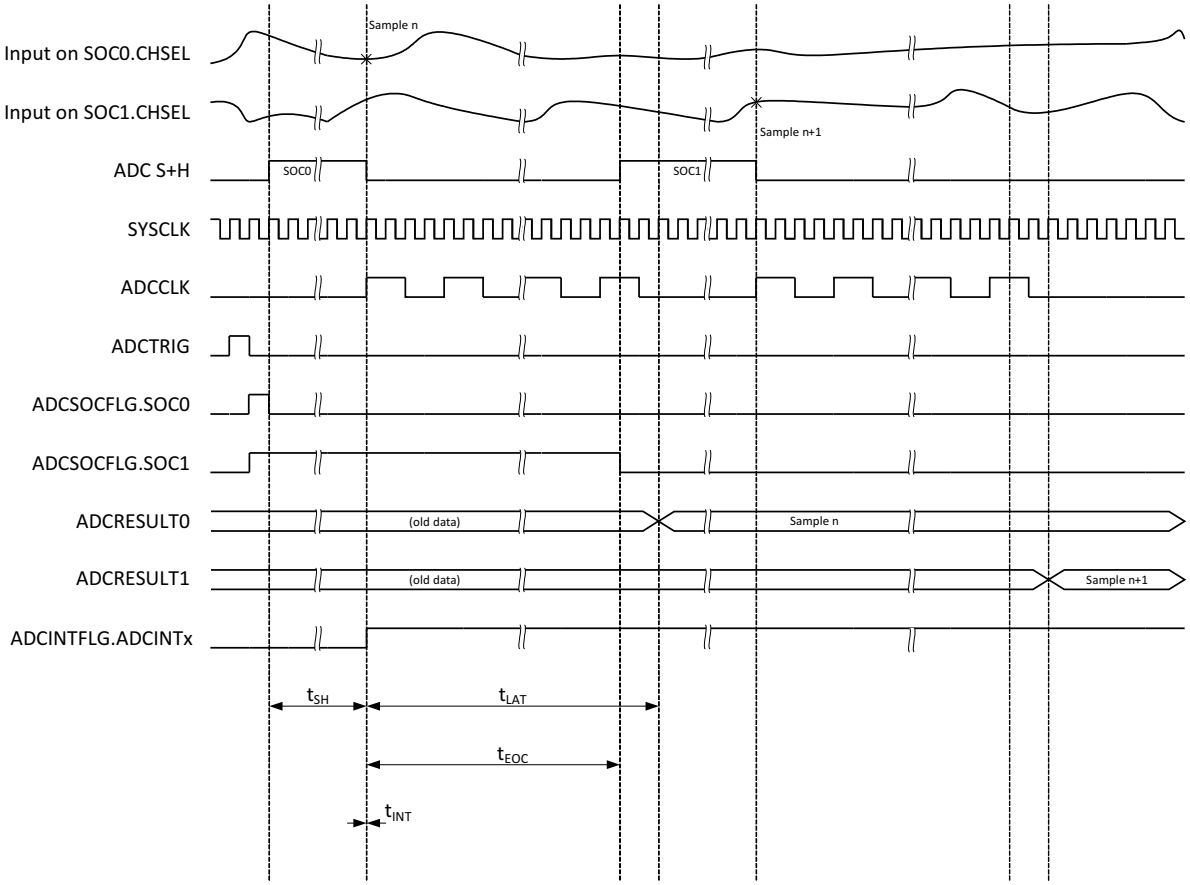


Figure 18-10. ADC Timings for 12-bit Mode in Early Interrupt Mode

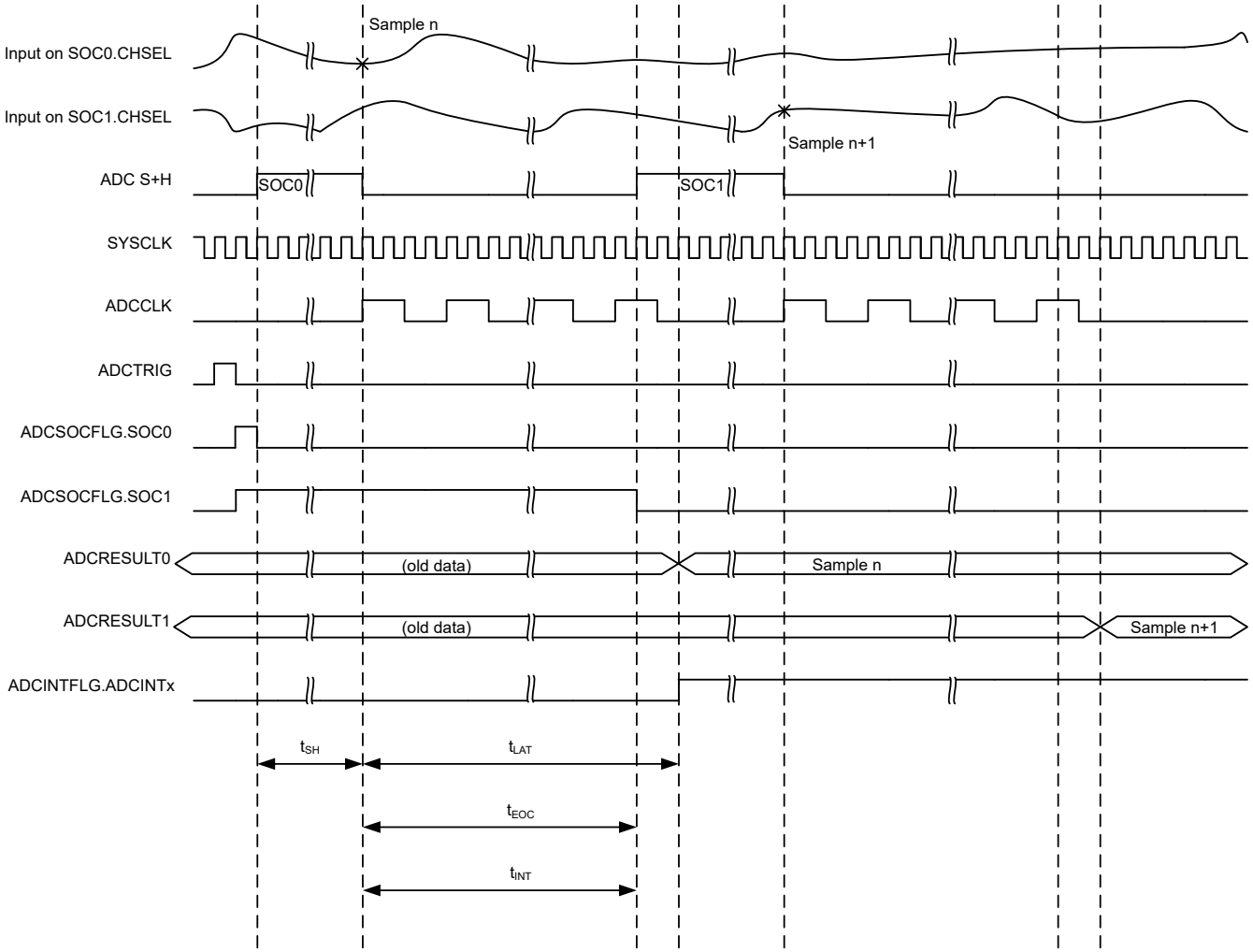


Figure 18-11. ADC Timings for 12-bit Mode in Late Interrupt Mode

Table 18-6. ADC Timings with SAMPCAPRESETSEL = 0

ADCCLK Prescale		SYSCLK Cycles				
ADCCTL2. PRESCALE	Prescale Ratio	t_{EOC}	t_{LAT}	t_{INT} (Early)	t_{INT} (Late)	t_{DMA}
0	1	12	16	1	12	16
2	2	24	28	1	24	28
4	3	36	40	1	36	40
6	4	48	52	1	48	52
8	5	60	64	1	60	64
10	6	72	76	1	72	76
12	7	84	88	1	84	88
14	8	96	100	1	96	100

Table 18-7. ADC Timings with SAMPCAPRESETSEL = 1

ADCCLK Prescale		SYSCLK Cycles				
ADCCTL2. PRESCALE	Prescale Ratio	t_{EOC}	t_{LAT}	t_{INT} (Early)	t_{INT} (Late)	t_{DMA}
0	1	11	15	1	11	15
2	2	22	26	1	22	26
4	3	33	37	1	22	37
6	4	44	48	1	44	48
8	5	55	59	1	55	59
10	6	66	70	1	66	70
12	7	77	81	1	77	81
14	8	88	92	1	88	92

Note

It is recommended only to configure whole numbered integer prescalers in ADCCTL2.PRESCALE.

18.3 ADC_LITE_REGS Registers

Table 18-8 lists the memory-mapped registers for the ADC_LITE_REGS registers. All register offset addresses not listed in Table 18-8 should be considered as reserved locations and the register contents should not be modified.

Table 18-8. ADC_LITE_REGS Registers

Offset	Acronym	Register Name	Section
400h	FSUB_0	Subscriber Port 0	Section 18.3.1
404h	FSUB_1	Subscriber Port 1	Section 18.3.2
408h	FSUB_2	Subscriber Port 2	Section 18.3.3
40Ch	FSUB_3	Subscriber Port 3	Section 18.3.4
410h	FSUB_4	Subscriber Port 4	Section 18.3.5
414h	FSUB_5	Subscriber Port 5	Section 18.3.6
444h	FPUB_0	Publisher port 0	Section 18.3.7
448h	FPUB_1	Publisher port 1	Section 18.3.8
44Ch	FPUB_2	Publisher port 2	Section 18.3.9
450h	FPUB_3	Publisher port 3	Section 18.3.10
800h	PWREN	Power enable	Section 18.3.11
804h	RSTCTL	Reset Control	Section 18.3.12
814h	STAT	Status Register	Section 18.3.13
1000h	ADCCTL1	ADC Control 1 Register	Section 18.3.14
1004h	ADCCTL2	ADC Control 2 Register	Section 18.3.15
1010h	ADCINTSEL	ADC Interrupt 1, 2, 3 and 4 Selection Register	Section 18.3.16
1014h	ADCDMAINTSEL	ADC DMA Interrupt 1, 2, 3 and 4 Selection Register	Section 18.3.17
1018h	ADCRAWINTFLG	ADC Raw Interrupt Flag Register	Section 18.3.18
101Ch	ADCINTFLG	ADC Interrupt Flag Register	Section 18.3.19
1020h	ADCINTFLGFRC	ADC Interrupt Flag Force Register	Section 18.3.20
1024h	ADCINTFLGCLR	ADC Interrupt Flag Clear Register	Section 18.3.21
1028h	ADCINTOVF	ADC Interrupt Overflow Register	Section 18.3.22
102Ch	ADCINTOVFCLR	ADC Interrupt Overflow Clear Register	Section 18.3.23
103Ch	ADCSOCFLG1	ADC SOC Flag 1 Register	Section 18.3.24
1044h	ADCSOCOVF1	ADC SOC Overflow 1 Register	Section 18.3.25
1048h	ADCSOCOVFCLR1	ADC SOC Overflow Clear 1 Register	Section 18.3.26
104Ch	ADCSOC0CTL	ADC SOC0 Control Register	Section 18.3.27
1050h	ADCSOC1CTL	ADC SOC1 Control Register	Section 18.3.28
1054h	ADCSOC2CTL	ADC SOC2 Control Register	Section 18.3.29
1058h	ADCSOC3CTL	ADC SOC3 Control Register	Section 18.3.30
105Ch	ADCSOC4CTL	ADC SOC4 Control Register	Section 18.3.31
1060h	ADCSOC5CTL	ADC SOC5 Control Register	Section 18.3.32
1064h	ADCSOC6CTL	ADC SOC6 Control Register	Section 18.3.33
1068h	ADCSOC7CTL	ADC SOC7 Control Register	Section 18.3.34
106Ch	ADCSOC8CTL	ADC SOC8 Control Register	Section 18.3.35
1070h	ADCSOC9CTL	ADC SOC9 Control Register	Section 18.3.36
1074h	ADCSOC10CTL	ADC SOC10 Control Register	Section 18.3.37
1078h	ADCSOC11CTL	ADC SOC11 Control Register	Section 18.3.38
107Ch	ADCSOC12CTL	ADC SOC12 Control Register	Section 18.3.39
1080h	ADCSOC13CTL	ADC SOC13 Control Register	Section 18.3.40

Table 18-8. ADC_LITE_REGS Registers (continued)

Offset	Acronym	Register Name	Section
1084h	ADCSOC14CTL	ADC SOC14 Control Register	Section 18.3.41
1088h	ADCSOC15CTL	ADC SOC15 Control Register	Section 18.3.42
10CCh	ADCEVTSTAT	ADC Event Status Register	Section 18.3.43
10D0h	ADCEVTCLR	ADC Event Clear Register	Section 18.3.44
10D4h	ADCEVTSEL	ADC Event Selection Register	Section 18.3.45
10D8h	ADCEVTINTSEL	ADC Event Interrupt Selection Register	Section 18.3.46
10E4h	ADCREV	ADC Revision Register	Section 18.3.47
10E8h	ADCOFFTRIM	ADC Offset Trim Register 1	Section 18.3.48
1100h	ADCPPB1CONFIG	ADC PPB1 Config Register	Section 18.3.49
1110h	ADCPPB1TRIPHI	ADC PPB1 Trip High Register	Section 18.3.50
1114h	ADCPPB1TRIPLO	ADC PPB1 Trip Low/Trigger Time Stamp Register	Section 18.3.51
1120h	ADCPPB2CONFIG	ADC PPB2 Config Register	Section 18.3.52
1130h	ADCPPB2TRIPHI	ADC PPB2 Trip High Register	Section 18.3.53
1134h	ADCPPB2TRIPLO	ADC PPB2 Trip Low/Trigger Time Stamp Register	Section 18.3.54
1140h	ADCPPB3CONFIG	ADC PPB3 Config Register	Section 18.3.55
1150h	ADCPPB3TRIPHI	ADC PPB3 Trip High Register	Section 18.3.56
1154h	ADCPPB3TRIPLO	ADC PPB3 Trip Low/Trigger Time Stamp Register	Section 18.3.57
1160h	ADCPPB4CONFIG	ADC PPB4 Config Register	Section 18.3.58
1170h	ADCPPB4TRIPHI	ADC PPB4 Trip High Register	Section 18.3.59
1174h	ADCPPB4TRIPLO	ADC PPB4 Trip Low/Trigger Time Stamp Register	Section 18.3.60
1180h	ADCINTCYCLE	ADC Early Interrupt Generation Cycle	Section 18.3.61
119Ch	ADCREV2	ADC Wrapper Revision Register	Section 18.3.62
1200h	ADCPPB1LIMIT	ADC PPB1Conversion Count Limit Register	Section 18.3.63
1204h	ADCPPB1PCOUNT	ADC PPB1 Partial Conversion Count Register	Section 18.3.64
1208h	ADCPPB1CONFIG2	ADC PPB1 Sum Shift Register	Section 18.3.65
120Ch	ADCPPB1PSUM	ADC PPB1 Partial Sum Register	Section 18.3.66
1240h	ADCPPB2LIMIT	ADC PPB2Conversion Count Limit Register	Section 18.3.67
1244h	ADCPPB2PCOUNT	ADC PPB2 Partial Conversion Count Register	Section 18.3.68
1248h	ADCPPB2CONFIG2	ADC PPB2 Sum Shift Register	Section 18.3.69
124Ch	ADCPPB2PSUM	ADC PPB2 Partial Sum Register	Section 18.3.70
1280h	ADCPPB3LIMIT	ADC PPB3Conversion Count Limit Register	Section 18.3.71
1284h	ADCPPB3PCOUNT	ADC PPB3 Partial Conversion Count Register	Section 18.3.72
1288h	ADCPPB3CONFIG2	ADC PPB3 Sum Shift Register	Section 18.3.73
128Ch	ADCPPB3PSUM	ADC PPB3 Partial Sum Register	Section 18.3.74
12C0h	ADCPPB4LIMIT	ADC PPB4Conversion Count Limit Register	Section 18.3.75
12C4h	ADCPPB4PCOUNT	ADC PPB4 Partial Conversion Count Register	Section 18.3.76
12C8h	ADCPPB4CONFIG2	ADC PPB4 Sum Shift Register	Section 18.3.77
12CCh	ADCPPB4PSUM	ADC PPB4 Partial Sum Register	Section 18.3.78
1320h	ADCSEQCTL	ADC Sequencer common control Register	Section 18.3.79
1324h	ADCSEQ1CONFIG	ADC Sequencer 1 Config register	Section 18.3.80
1328h	ADCSEQ2CONFIG	ADC Sequencer 2 Config register	Section 18.3.81
132Ch	ADCSEQ3CONFIG	ADC Sequencer 3 Config register	Section 18.3.82
1330h	ADCSEQ4CONFIG	ADC Sequencer 4 Config register	Section 18.3.83

Complex bit access types are encoded to fit into small table cells. Table 18-9 shows the codes that are used for access types in this section.

Table 18-9. ADC_LITE_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set
WK	W K	Write Write protected by a key
Reset or Default Value		
-n		Value after reset or the default value

18.3.1 FSUB_0 Register (Offset = 400h) [Reset = 00000000h]

FSUB_0 is shown in Table 18-10.

Return to the Summary Table.

Subscriber port

Table 18-10. FSUB_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3-0	CHANID	R/W	0h	0 = disconnected. 1-15 = connected to channelID = CHANID. 0h = A value of 0 specifies that the event is not connected Fh = Consult your device datasheet as the actual allowed maximum may be less than 15.

18.3.2 FSUB_1 Register (Offset = 404h) [Reset = 00000000h]

FSUB_1 is shown in Table 18-11.

Return to the Summary Table.

Subscriber port

Table 18-11. FSUB_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3-0	CHANID	R/W	0h	0 = disconnected. 1-15 = connected to channelID = CHANID. 0h = A value of 0 specifies that the event is not connected Fh = Consult your device datasheet as the actual allowed maximum may be less than 15.

18.3.3 FSUB_2 Register (Offset = 408h) [Reset = 00000000h]

FSUB_2 is shown in Table 18-12.

Return to the Summary Table.

Subscriber port

Table 18-12. FSUB_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3-0	CHANID	R/W	0h	0 = disconnected. 1-15 = connected to channelID = CHANID. 0h = A value of 0 specifies that the event is not connected Fh = Consult your device datasheet as the actual allowed maximum may be less than 15.

18.3.4 FSUB_3 Register (Offset = 40Ch) [Reset = 00000000h]

FSUB_3 is shown in Table 18-13.

Return to the Summary Table.

Subscriber port

Table 18-13. FSUB_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3-0	CHANID	R/W	0h	0 = disconnected. 1-15 = connected to channelID = CHANID. 0h = A value of 0 specifies that the event is not connected Fh = Consult your device datasheet as the actual allowed maximum may be less than 15.

18.3.5 FSUB_4 Register (Offset = 410h) [Reset = 00000000h]

FSUB_4 is shown in Table 18-14.

Return to the Summary Table.

Subscriber port

Table 18-14. FSUB_4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3-0	CHANID	R/W	0h	0 = disconnected. 1-15 = connected to channelID = CHANID. 0h = A value of 0 specifies that the event is not connected Fh = Consult your device datasheet as the actual allowed maximum may be less than 15.

18.3.6 FSUB_5 Register (Offset = 414h) [Reset = 00000000h]

FSUB_5 is shown in Table 18-15.

Return to the Summary Table.

Subscriber port

Table 18-15. FSUB_5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3-0	CHANID	R/W	0h	0 = disconnected. 1-15 = connected to channelID = CHANID. 0h = A value of 0 specifies that the event is not connected Fh = Consult your device datasheet as the actual allowed maximum may be less than 15.

18.3.7 FPUB_0 Register (Offset = 444h) [Reset = 00000000h]

FPUB_0 is shown in Table 18-16.

Return to the Summary Table.

Publisher port

Table 18-16. FPUB_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3-0	CHANID	R/W	0h	0 = disconnected. 1-15 = connected to channelID = CHANID. 0h = A value of 0 specifies that the event is not connected Fh = Consult your device datasheet as the actual allowed maximum may be less than 15.

18.3.8 FPUB_1 Register (Offset = 448h) [Reset = 00000000h]

FPUB_1 is shown in Table 18-17.

Return to the Summary Table.

Publisher port

Table 18-17. FPUB_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3-0	CHANID	R/W	0h	0 = disconnected. 1-15 = connected to channelID = CHANID. 0h = A value of 0 specifies that the event is not connected Fh = Consult your device datasheet as the actual allowed maximum may be less than 15.

18.3.9 FPUB_2 Register (Offset = 44Ch) [Reset = 00000000h]

FPUB_2 is shown in Table 18-18.

Return to the Summary Table.

Publisher port

Table 18-18. FPUB_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3-0	CHANID	R/W	0h	0 = disconnected. 1-15 = connected to channelID = CHANID. 0h = A value of 0 specifies that the event is not connected Fh = Consult your device datasheet as the actual allowed maximum may be less than 15.

18.3.10 FPUB_3 Register (Offset = 450h) [Reset = 00000000h]

FPUB_3 is shown in Table 18-19.

Return to the Summary Table.

Publisher port

Table 18-19. FPUB_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3-0	CHANID	R/W	0h	0 = disconnected. 1-15 = connected to channelID = CHANID. 0h = A value of 0 specifies that the event is not connected Fh = Consult your device datasheet as the actual allowed maximum may be less than 15.

18.3.11 PWREN Register (Offset = 800h) [Reset = 00000000h]

PWREN is shown in Table 18-20.

Return to the Summary Table.

Register to control the power state

Table 18-20. PWREN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to allow Power State Change 26h = KEY to allow write access to this register
23-1	RESERVED	R	0h	
0	ENABLE	R/WK	0h	Enable the power KEY must be set to 26h to write to this bit. 0h = Disable Power 1h = Enable Power

18.3.12 RSTCTL Register (Offset = 804h) [Reset = 00000000h]

RSTCTL is shown in Table 18-21.

Return to the Summary Table.

Register to control reset assertion and de-assertion

Table 18-21. RSTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	Unlock key 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 KEY must be set to B1h to write to this bit. 0h = Writing 0 has no effect 1h = Clear reset sticky bit
0	RESETASSERT	WK	0h	Assert reset to the peripheral KEY must be set to B1h to write to this bit. 0h = Writing 0 has no effect 1h = Assert reset

18.3.13 STAT Register (Offset = 814h) [Reset = 00000000h]

STAT is shown in Table 18-22.

Return to the Summary Table.

peripheral enable and reset status

Table 18-22. STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	
16	RESETSTKYCLR	R	0h	This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register 0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register 1h = The peripheral was reset since the last bit clear
15-0	RESERVED	R	0h	

18.3.14 ADCCTL1 Register (Offset = 1000h) [Reset = 00000000h]

ADCCTL1 is shown in Table 18-23.

Return to the Summary Table.

ADC Control 1 Register

Table 18-23. ADCCTL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	RESERVED	R	0h	Reserved
13	ADCBSY	R	0h	ADC Busy. Set when ADC SOC is generated, cleared by hardware four ADC clocks after negative edge of S/H pulse. Used by the ADC state machine to determine if ADC is available to sample. 0 ADC is available to sample next channel 1 ADC is busy and cannot sample another channel Reset type: SYSRSn
12	RESERVED	R	0h	Reserved
11-8	ADCBSYCHN	R	0h	ADC Busy Channel. Set when an ADC Start of Conversion (SOC) is generated. When ADCBSY=0: holds the value of the last converted SOC When ADCBSY=1: reflects the SOC currently being processed 0h SOC0 is currently processing or was last SOC converted 1h SOC1 is currently processing or was last SOC converted 2h SOC2 is currently processing or was last SOC converted 3h SOC3 is currently processing or was last SOC converted 4h SOC4 is currently processing or was last SOC converted 5h SOC5 is currently processing or was last SOC converted 6h SOC6 is currently processing or was last SOC converted 7h SOC7 is currently processing or was last SOC converted 8h SOC8 is currently processing or was last SOC converted 9h SOC9 is currently processing or was last SOC converted Ah SOC10 is currently processing or was last SOC converted Bh SOC11 is currently processing or was last SOC converted Ch SOC12 is currently processing or was last SOC converted Dh SOC13 is currently processing or was last SOC converted Eh SOC14 is currently processing or was last SOC converted Fh SOC15 is currently processing or was last SOC converted Reset type: SYSRSn
7	ADCPWDNZ	R/W	0h	ADC Power Down (active low). This bit controls the power up and power down of all the analog circuitry inside the analog core. 0 All analog circuitry inside the core is powered down 1 All analog circuitry inside the core is powered up Reset type: SYSRSn
6-3	RESERVED	R	0h	Reserved
2	INTPULSEPOS	R/W	0h	ADC Interrupt Pulse Position. 0 Interrupt pulse generation occurs when ADC begins conversion (at the end of the acquisition window) plus a number of SYSCLK cycles as specified in the ADCINTCYCLE.OFFSET register. 1 Interrupt pulse generation occurs at the end of the conversion, 1 cycle prior to the ADC result latching into its result register Reset type: SYSRSn
1-0	RESERVED	R	0h	Reserved

18.3.15 ADCCTL2 Register (Offset = 1004h) [Reset = 00000000h]

ADCCTL2 is shown in Table 18-24.

Return to the Summary Table.

ADC Control 2 Register

Table 18-24. ADCCTL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R	0h	Reserved
12-9	RESERVED	R	0h	Reserved
8-6	RESERVED	R	0h	
5-4	RESERVED	R	0h	Reserved
3-0	PRESCALE	R/W	0h	ADC Clock Prescaler. 0000 ADCCLK = Input Clock / 1.0 0001 ADCCLK = Input Clock / 1.5 0010 ADCCLK = Input Clock / 2.0 0011 ADCCLK = Input Clock / 2.5 0100 ADCCLK = Input Clock / 3.0 0101 ADCCLK = Input Clock / 3.5 0110 ADCCLK = Input Clock / 4.0 0111 ADCCLK = Input Clock / 4.5 1000 ADCCLK = Input Clock / 5.0 1001 ADCCLK = Input Clock / 5.5 1010 ADCCLK = Input Clock / 6.0 1011 ADCCLK = Input Clock / 6.5 1100 ADCCLK = Input Clock / 7.0 1101 ADCCLK = Input Clock / 7.5 1110 ADCCLK = Input Clock / 8.0 1111 ADCCLK = Input Clock / 8.5 Note: Non-integer ADC clock dividers are not recommended. Reset type: SYSRSn

18.3.16 ADCINTSEL Register (Offset = 1010h) [Reset = 00000000h]

ADCINTSEL is shown in Table 18-25.

Return to the Summary Table.

ADC Interrupt 1, 2, 3 and 4 Selection Register

Table 18-25. ADCINTSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INT4E	R/W	0h	ADCINT4 Interrupt Enable 0 ADCINT4 is disabled 1 ADCINT4 is enabled Reset type: SYSRSn
30	INT4CONT	R/W	0h	ADCINT4 Continue to Interrupt Mode 0 No further ADCINT4 pulses are generated until ADCINT4 flag (in ADCINTFLG register) is cleared by user. 1 ADCINT4 pulses are generated whenever an EOC pulse is generated irrespective of whether the flag bit is cleared or not. Reset type: SYSRSn
29-28	RESERVED	R	0h	Reserved
27-24	INT4SEL	R/W	0h	ADCINT4 EOC Source Select 00h EOC0 is trigger for ADCINT4 01h EOC1 is trigger for ADCINT4 02h EOC2 is trigger for ADCINT4 03h EOC3 is trigger for ADCINT4 04h EOC4 is trigger for ADCINT4 05h EOC5 is trigger for ADCINT4 06h EOC6 is trigger for ADCINT4 07h EOC7 is trigger for ADCINT4 08h EOC8 is trigger for ADCINT4 09h EOC9 is trigger for ADCINT4 0Ah EOC10 is trigger for ADCINT4 0Bh EOC11 is trigger for ADCINT4 0Ch EOC12 is trigger for ADCINT4 0Dh EOC13 is trigger for ADCINT4 0Eh EOC14 is trigger for ADCINT4 0Fh EOC15 is trigger for ADCINT4 Note : When oversampling is enabled, the end of the last oversampled coversion will trigger the interrupts Reset type: SYSRSn
23	INT3E	R/W	0h	ADCINT3 Interrupt Enable 0 ADCINT3 is disabled 1 ADCINT3 is enabled Reset type: SYSRSn
22	INT3CONT	R/W	0h	ADCINT3 Continue to Interrupt Mode 0 No further ADCINT3 pulses are generated until ADCINT3 flag (in ADCINTFLG register) is cleared by user. 1 ADCINT3 pulses are generated whenever an EOC pulse is generated irrespective of whether the flag bit is cleared or not. Reset type: SYSRSn
21-20	RESERVED	R	0h	Reserved

Table 18-25. ADCINTSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-16	INT3SEL	R/W	0h	ADCINT3 EOC Source Select 00h EOC0 is trigger for ADCINT3 01h EOC1 is trigger for ADCINT3 02h EOC2 is trigger for ADCINT3 03h EOC3 is trigger for ADCINT3 04h EOC4 is trigger for ADCINT3 05h EOC5 is trigger for ADCINT3 06h EOC6 is trigger for ADCINT3 07h EOC7 is trigger for ADCINT3 08h EOC8 is trigger for ADCINT3 09h EOC9 is trigger for ADCINT3 0Ah EOC10 is trigger for ADCINT3 0Bh EOC11 is trigger for ADCINT3 0Ch EOC12 is trigger for ADCINT3 0Dh EOC13 is trigger for ADCINT3 0Eh EOC14 is trigger for ADCINT3 0Fh EOC15 is trigger for ADCINT3 Note : When oversampling is enabled, the end of the last oversampled coversion will trigger the interrupts Reset type: SYSRSn
15	INT2E	R/W	0h	ADCINT2 Interrupt Enable 0 ADCINT2 is disabled 1 ADCINT2 is enabled Reset type: SYSRSn
14	INT2CONT	R/W	0h	ADCINT2 Continue to Interrupt Mode 0 No further ADCINT2 pulses are generated until ADCINT2 flag (in ADCINTFLG register) is cleared by user. 1 ADCINT2 pulses are generated whenever an EOC pulse is generated irrespective of whether the flag bit is cleared or not. Reset type: SYSRSn
13-12	RESERVED	R	0h	Reserved
11-8	INT2SEL	R/W	0h	ADCINT2 EOC Source Select 00h EOC0 is trigger for ADCINT2 01h EOC1 is trigger for ADCINT2 02h EOC2 is trigger for ADCINT2 03h EOC3 is trigger for ADCINT2 04h EOC4 is trigger for ADCINT2 05h EOC5 is trigger for ADCINT2 06h EOC6 is trigger for ADCINT2 07h EOC7 is trigger for ADCINT2 08h EOC8 is trigger for ADCINT2 09h EOC9 is trigger for ADCINT2 0Ah EOC10 is trigger for ADCINT2 0Bh EOC11 is trigger for ADCINT2 0Ch EOC12 is trigger for ADCINT2 0Dh EOC13 is trigger for ADCINT2 0Eh EOC14 is trigger for ADCINT2 0Fh EOC15 is trigger for ADCINT2 Note : When oversampling is enabled, the end of the last oversampled coversion will trigger the interrupts Reset type: SYSRSn
7	INT1E	R/W	0h	ADCINT1 Interrupt Enable 0 ADCINT1 is disabled 1 ADCINT1 is enabled Reset type: SYSRSn
6	INT1CONT	R/W	0h	ADCINT1 Continue to Interrupt Mode 0 No further ADCINT1 pulses are generated until ADCINT1 flag (in ADCINTFLG register) is cleared by user. 1 ADCINT1 pulses are generated whenever an EOC pulse is generated irrespective of whether the flag bit is cleared or not. Reset type: SYSRSn

Table 18-25. ADCINTSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5-4	RESERVED	R	0h	Reserved
3-0	INT1SEL	R/W	0h	ADCINT1 EOC Source Select 00h EOC0 is trigger for ADCINT1 01h EOC1 is trigger for ADCINT1 02h EOC2 is trigger for ADCINT1 03h EOC3 is trigger for ADCINT1 04h EOC4 is trigger for ADCINT1 05h EOC5 is trigger for ADCINT1 06h EOC6 is trigger for ADCINT1 07h EOC7 is trigger for ADCINT1 08h EOC8 is trigger for ADCINT1 09h EOC9 is trigger for ADCINT1 0Ah EOC10 is trigger for ADCINT1 0Bh EOC11 is trigger for ADCINT1 0Ch EOC12 is trigger for ADCINT1 0Dh EOC13 is trigger for ADCINT1 0Eh EOC14 is trigger for ADCINT1 0Fh EOC15 is trigger for ADCINT1 Note : When oversampling is enabled, the end of the last oversampled coversion will trigger the interrupts Reset type: SYSRSn

18.3.17 ADCDMAINTSEL Register (Offset = 1014h) [Reset = 00000000h]

ADCDMAINTSEL is shown in Table 18-26.

Return to the Summary Table.

ADC DMA Interrupt 1, 2, 3 and 4 Selection Register

Table 18-26. ADCDMAINTSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	DMAINT4E	R/W	0h	ADCDMAINT4 Interrupt Enable 0 ADCDMAINT4 is disabled 1 ADCDMAINT4 is enabled Reset type: SYSRSn
30	DMAINT4CONT	R/W	0h	ADCDMAINT4 Continue to Interrupt Mode 0 No further ADCDMAINT4 pulses are generated until ADCDMAINT4 flag (in ADCDMAINTFLG register) is cleared by user. 1 ADCDMAINT4 pulses are generated whenever an EOC pulse is generated irrespective of whether the flag bit is cleared or not. Reset type: SYSRSn
29-28	RESERVED	R	0h	Reserved
27-24	DMAINT4SEL	R/W	0h	ADCDMAINT4 EOC Source Select 00h EOC0 is trigger for ADCDMAINT4 01h EOC1 is trigger for ADCDMAINT4 02h EOC2 is trigger for ADCDMAINT4 03h EOC3 is trigger for ADCDMAINT4 04h EOC4 is trigger for ADCDMAINT4 05h EOC5 is trigger for ADCDMAINT4 06h EOC6 is trigger for ADCDMAINT4 07h EOC7 is trigger for ADCDMAINT4 08h EOC8 is trigger for ADCDMAINT4 09h EOC9 is trigger for ADCDMAINT4 0Ah EOC10 is trigger for ADCDMAINT4 0Bh EOC11 is trigger for ADCDMAINT4 0Ch EOC12 is trigger for ADCDMAINT4 0Dh EOC13 is trigger for ADCDMAINT4 0Eh EOC14 is trigger for ADCDMAINT4 0Fh EOC15 is trigger for ADCDMAINT4 Note : When oversampling is enabled, the end of the last oversampled coversion will trigger the DMAINTerrupts Reset type: SYSRSn
23	DMAINT3E	R/W	0h	ADCDMAINT3 Interrupt Enable 0 ADCDMAINT3 is disabled 1 ADCDMAINT3 is enabled Reset type: SYSRSn
22	DMAINT3CONT	R/W	0h	ADCDMAINT3 Continue to Interrupt Mode 0 No further ADCDMAINT3 pulses are generated until ADCDMAINT3 flag (in ADCDMAINTFLG register) is cleared by user. 1 ADCDMAINT3 pulses are generated whenever an EOC pulse is generated irrespective of whether the flag bit is cleared or not. Reset type: SYSRSn
21-20	RESERVED	R	0h	Reserved

Table 18-26. ADCDMAINTSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-16	DMAINT3SEL	R/W	0h	ADCDMAINT3 EOC Source Select 00h EOC0 is trigger for ADCDMAINT3 01h EOC1 is trigger for ADCDMAINT3 02h EOC2 is trigger for ADCDMAINT3 03h EOC3 is trigger for ADCDMAINT3 04h EOC4 is trigger for ADCDMAINT3 05h EOC5 is trigger for ADCDMAINT3 06h EOC6 is trigger for ADCDMAINT3 07h EOC7 is trigger for ADCDMAINT3 08h EOC8 is trigger for ADCDMAINT3 09h EOC9 is trigger for ADCDMAINT3 0Ah EOC10 is trigger for ADCDMAINT3 0Bh EOC11 is trigger for ADCDMAINT3 0Ch EOC12 is trigger for ADCDMAINT3 0Dh EOC13 is trigger for ADCDMAINT3 0Eh EOC14 is trigger for ADCDMAINT3 0Fh EOC15 is trigger for ADCDMAINT3 Note : When oversampling is enabled, the end of the last oversampled coversion will trigger the DMAINTerrupts Reset type: SYSRSn
15	DMAINT2E	R/W	0h	ADCDMAINT2 Interrupt Enable 0 ADCDMAINT2 is disabled 1 ADCDMAINT2 is enabled Reset type: SYSRSn
14	DMAINT2CONT	R/W	0h	ADCDMAINT2 Continue to Interrupt Mode 0 No further ADCDMAINT2 pulses are generated until ADCDMAINT2 flag (in ADCDMAINTFLG register) is cleared by user. 1 ADCDMAINT2 pulses are generated whenever an EOC pulse is generated irrespective of whether the flag bit is cleared or not. Reset type: SYSRSn
13-12	RESERVED	R	0h	Reserved
11-8	DMAINT2SEL	R/W	0h	ADCDMAINT2 EOC Source Select 00h EOC0 is trigger for ADCDMAINT2 01h EOC1 is trigger for ADCDMAINT2 02h EOC2 is trigger for ADCDMAINT2 03h EOC3 is trigger for ADCDMAINT2 04h EOC4 is trigger for ADCDMAINT2 05h EOC5 is trigger for ADCDMAINT2 06h EOC6 is trigger for ADCDMAINT2 07h EOC7 is trigger for ADCDMAINT2 08h EOC8 is trigger for ADCDMAINT2 09h EOC9 is trigger for ADCDMAINT2 0Ah EOC10 is trigger for ADCDMAINT2 0Bh EOC11 is trigger for ADCDMAINT2 0Ch EOC12 is trigger for ADCDMAINT2 0Dh EOC13 is trigger for ADCDMAINT2 0Eh EOC14 is trigger for ADCDMAINT2 0Fh EOC15 is trigger for ADCDMAINT2 Note : When oversampling is enabled, the end of the last oversampled coversion will trigger the DMAINTerrupts Reset type: SYSRSn
7	DMAINT1E	R/W	0h	ADCDMAINT1 Interrupt Enable 0 ADCDMAINT1 is disabled 1 ADCDMAINT1 is enabled Reset type: SYSRSn
6	DMAINT1CONT	R/W	0h	ADCDMAINT1 Continue to Interrupt Mode 0 No further ADCDMAINT1 pulses are generated until ADCDMAINT1 flag (in ADCDMAINTFLG register) is cleared by user. 1 ADCDMAINT1 pulses are generated whenever an EOC pulse is generated irrespective of whether the flag bit is cleared or not. Reset type: SYSRSn

Table 18-26. ADCDMAINTSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5-4	RESERVED	R	0h	Reserved
3-0	DMAINT1SEL	R/W	0h	ADCDMAINT1 EOC Source Select 00h EOC0 is trigger for ADCDMAINT1 01h EOC1 is trigger for ADCDMAINT1 02h EOC2 is trigger for ADCDMAINT1 03h EOC3 is trigger for ADCDMAINT1 04h EOC4 is trigger for ADCDMAINT1 05h EOC5 is trigger for ADCDMAINT1 06h EOC6 is trigger for ADCDMAINT1 07h EOC7 is trigger for ADCDMAINT1 08h EOC8 is trigger for ADCDMAINT1 09h EOC9 is trigger for ADCDMAINT1 0Ah EOC10 is trigger for ADCDMAINT1 0Bh EOC11 is trigger for ADCDMAINT1 0Ch EOC12 is trigger for ADCDMAINT1 0Dh EOC13 is trigger for ADCDMAINT1 0Eh EOC14 is trigger for ADCDMAINT1 0Fh EOC15 is trigger for ADCDMAINT1 Note : When oversampling is enabled, the end of the last oversampled conversion will trigger the DMAINTerrupts Reset type: SYSRSn

18.3.18 ADCRAWINTFLG Register (Offset = 1018h) [Reset = 00000000h]

ADCRAWINTFLG is shown in Table 18-27.

Return to the Summary Table.

ADC Raw Interrupt Flag Register

Table 18-27. ADCRAWINTFLG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	Reserved
19	ADCDMARAWINT4	R	0h	ADC DMA Raw Interrupt 4 Flag. Reading these flags indicates if the associated ADCDMAINT condition occurred. This flag will be set irrespective of corresponding DMAINTE setting 0 Selected EOC/OSINT event did not occur 1 Selected EOC/OSINT event occurred Writing corresponding INTCLR bit in ADCINTFLGCLR register will clear this bit. Reset type: SYSRSn
18	ADCDMARAWINT3	R	0h	ADC DMA Raw Interrupt 3 Flag. Reading these flags indicates if the associated ADCDMAINT condition occurred. This flag will be set irrespective of corresponding DMAINTE setting 0 Selected EOC/OSINT event did not occur 1 Selected EOC/OSINT event occurred Writing corresponding INTCLR bit in ADCINTFLGCLR register will clear this bit. Reset type: SYSRSn
17	ADCDMARAWINT2	R	0h	ADC DMA Raw Interrupt 2 Flag. Reading these flags indicates if the associated ADCDMAINT condition occurred. This flag will be set irrespective of corresponding DMAINTE setting 0 Selected EOC/OSINT event did not occur 1 Selected EOC/OSINT event occurred Writing corresponding INTCLR bit in ADCINTFLGCLR register will clear this bit. Reset type: SYSRSn
16	ADCDMARAWINT1	R	0h	ADC DMA Raw Interrupt 1 Flag. Reading these flags indicates if the associated ADCDMAINT condition occurred. This flag will be set irrespective of corresponding DMAINTE setting 0 Selected EOC/OSINT event did not occur 1 Selected EOC/OSINT event occurred Writing corresponding INTCLR bit in ADCINTFLGCLR register will clear this bit. Reset type: SYSRSn
15-4	RESERVED	R	0h	Reserved
3	ADCRAWINT4	R	0h	ADC RAW Interrupt 4 Flag. Reading these flags indicates if the associated INT condition occurred. This flag will be set irrespective of corresponding INTE setting 0 Selected EOC/OSINT event did not occur 1 Selected EOC/OSINT event occured Writing corresponding INTCLR bit in ADCINTFLGCLR register will clear this bit. Reset type: SYSRSn
2	ADCRAWINT3	R	0h	ADC RAW Interrupt 3 Flag. Reading these flags indicates if the associated INT condition occurred. This flag will be set irrespective of corresponding INTE setting 0 Selected EOC/OSINT event did not occur 1 Selected EOC/OSINT event occured Writing corresponding INTCLR bit in ADCINTFLGCLR register will clear this bit. Reset type: SYSRSn

Table 18-27. ADCRAWINTFLG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	ADCRAWINT2	R	0h	ADC RAW Interrupt 2 Flag. Reading these flags indicates if the associated INT condition occurred. This flag will be set irrespective of corresponding INTE setting 0 Selected EOC/OSINT event did not occur 1 Selected EOC/OSINT event occured Writing corresponding INTCLR bit in ADCINTFLGCLR register will clear this bit. Reset type: SYSRSn
0	ADCRAWINT1	R	0h	ADC RAW Interrupt 1 Flag. Reading these flags indicates if the associated INT condition occurred. This flag will be set irrespective of corresponding INTE setting 0 Selected EOC/OSINT event did not occur 1 Selected EOC/OSINT event occurred Writing corresponding INTCLR bit in ADCINTFLGCLR register will clear this bit. Reset type: SYSRSn

18.3.19 ADCINTFLG Register (Offset = 101Ch) [Reset = 00000000h]

ADCINTFLG is shown in Table 18-28.

Return to the Summary Table.

ADC Interrupt Flag Register

Table 18-28. ADCINTFLG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	Reserved
19	ADCDMAINT4	R	0h	ADC DMA Interrupt 4 Flag. Reading these flags indicates if the associated ADCDMAINT pulse was generated since the last clear. 0 No ADC DMA interru4pt pulse generated 1 ADC DMA interrupt pulse generated If the ADC interrupt is placed in continue to interrupt mode (DMAINTSEL register) then further interrupt pulses are generated whenever a selected EOC event occurs even if the flag bit is set. If the continuous mode is not enabled, then no further interrupt pulses are generated until the user clears this flag bit using the ADCINTFLGCLR register. Rather, an ADC interrupt overflow event occurs in the ADCINTOVF register. Reset type: SYSRSn
18	ADCDMAINT3	R	0h	ADC DMA Interrupt 3 Flag. Reading these flags indicates if the associated ADCDMAINT pulse was generated since the last clear. 0 No ADC DMA interrupt pulse generated 1 ADC DMA interrupt pulse generated If the ADC interrupt is placed in continue to interrupt mode (DMAINTSEL register) then further interrupt pulses are generated whenever a selected EOC event occurs even if the flag bit is set. If the continuous mode is not enabled, then no further interrupt pulses are generated until the user clears this flag bit using the ADCINTFLGCLR register. Rather, an ADC interrupt overflow event occurs in the ADCINTOVF register. Reset type: SYSRSn
17	ADCDMAINT2	R	0h	ADC DMA Interrupt 2 Flag. Reading these flags indicates if the associated ADCDMAINT pulse was generated since the last clear. 0 No ADC DMA interrupt pulse generated 1 ADC DMA interrupt pulse generated If the ADC DMA interrupt is placed in continue to interrupt mode (DMAINTSEL register) then further interrupt pulses are generated whenever a selected EOC event occurs even if the flag bit is set. If the continuous mode is not enabled, then no further interrupt pulses are generated until the user clears this flag bit using the ADCINTFLGCLR register. Rather, an ADC interrupt overflow event occurs in the ADCINTOVF register. Reset type: SYSRSn
16	ADCDMAINT1	R	0h	ADC DMA Interrupt 1 Flag. Reading these flags indicates if the associated ADCDMAINT pulse was generated since the last clear. 0 No ADC DMA interrupt pulse generated 1 ADC DMA interrupt pulse generated If the ADC interrupt is placed in continue to interrupt mode (DMAINTSEL register) then further interrupt pulses are generated whenever a selected EOC event occurs even if the flag bit is set. If the continuous mode is not enabled, then no further interrupt pulses are generated until the user clears this flag bit using the ADCINTFLGCLR register. Rather, an ADC interrupt overflow event occurs in the ADCINTOVF register. Reset type: SYSRSn
15-12	RESERVED	R	0h	Reserved

Table 18-28. ADCINTFLG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	ADCINT4RESULT	R	0h	ADC Interrupt 4 Results Ready Flag. This flag is set when the conversions results associated with ADCINT4 latch into the corresponding results register. 0 Conversion results have not latched 1 Conversion results have latched This flag can be used in an ISR that is entered in early interrupt mode to ensure that the corresponding results are ready before proceeding to read the result register. This flag can be cleared via the ACK bit in the ADCINTFLGCLR that also clears the ADCINT4 flag. In case results latch and this flag is already set, the corresponding flag in ADCINTOVF is NOT set. This flag does NOT have to be cleared in order for ADCINT ISRs to propagate to the PIE. In case the associated SOC is associated with a PPB or PPBs, the flag will not be set until all associated PPB results latch. Reset type: SYSRSn
10	ADCINT3RESULT	R	0h	ADC Interrupt 3 Results Ready Flag. This flag is set when the conversions results associated with ADCINT3 latch into the corresponding results register. 0 Conversion results have not latched 1 Conversion results have latched This flag can be used in an ISR that is entered in early interrupt mode to ensure that the corresponding results are ready before proceeding to read the result register. This flag can be cleared via the ACK bit in the ADCINTFLGCLR that also clears the ADCINT3 flag. In case results latch and this flag is already set, the corresponding flag in ADCINTOVF is NOT set. This flag does NOT have to be cleared in order for ADCINT ISRs to propagate to the PIE. In case the associated SOC is associated with a PPB or PPBs, the flag will not be set until all associated PPB results latch. Reset type: SYSRSn
9	ADCINT2RESULT	R	0h	ADC Interrupt 2 Results Ready Flag. This flag is set when the conversions results associated with ADCINT2 latch into the corresponding results register. 0 Conversion results have not latched 1 Conversion results have latched This flag can be used in an ISR that is entered in early interrupt mode to ensure that the corresponding results are ready before proceeding to read the result register. This flag can be cleared via the ACK bit in the ADCINTFLGCLR that also clears the ADCINT2 flag. In case results latch and this flag is already set, the corresponding flag in ADCINTOVF is NOT set. This flag does NOT have to be cleared in order for ADCINT ISRs to propagate to the PIE. In case the associated SOC is associated with a PPB or PPBs, the flag will not be set until all associated PPB results latch. Reset type: SYSRSn

Table 18-28. ADCINTFLG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	ADCINT1RESULT	R	0h	ADC Interrupt 1 Results Ready Flag. This flag is set when the conversions results associated with ADCINT1 latch into the corresponding results register. 0 Conversion results have not latched 1 Conversion results have latched This flag can be used in an ISR that is entered in early interrupt mode to ensure that the corresponding results are ready before proceeding to read the result register. This flag can be cleared via the ACK bit in the ADCINTFLGCLR that also clears the ADCINT1 flag. In case results latch and this flag is already set, the corresponding flag in ADCINTOVF is NOT set. This flag does NOT have to be cleared in order for ADCINT ISRs to propagate to the PIE. In case the associated SOC is associated with a PPB or PPBs, the flag will not be set until all associated PPB results latch. Reset type: SYSRSn
7-4	RESERVED	R	0h	Reserved
3	ADCINT4	R	0h	ADC Interrupt 4 Flag. Reading these flags indicates if the associated ADCINT pulse was generated since the last clear. 0 No ADC interrupt pulse generated 1 ADC interrupt pulse generated If the ADC interrupt is placed in continue to interrupt mode (INTSEL register) then further interrupt pulses are generated whenever a selected EOC event occurs even if the flag bit is set. If the continuous mode is not enabled, then no further interrupt pulses are generated until the user clears this flag bit using the ADCINFLGCLR register. Rather, an ADC interrupt overflow event occurs in the ADCINTOVF register. Reset type: SYSRSn
2	ADCINT3	R	0h	ADC Interrupt 3 Flag. Reading these flags indicates if the associated ADCINT pulse was generated since the last clear. 0 No ADC interrupt pulse generated 1 ADC interrupt pulse generated If the ADC interrupt is placed in continue to interrupt mode (INTSEL register) then further interrupt pulses are generated whenever a selected EOC event occurs even if the flag bit is set. If the continuous mode is not enabled, then no further interrupt pulses are generated until the user clears this flag bit using the ADCINTFLGCLR register. Rather, an ADC interrupt overflow event occurs in the ADCINTOVF register. Reset type: SYSRSn
1	ADCINT2	R	0h	ADC Interrupt 2 Flag. Reading these flags indicates if the associated ADCINT pulse was generated since the last clear. 0 No ADC interrupt pulse generated 1 ADC interrupt pulse generated If the ADC interrupt is placed in continue to interrupt mode (INTSEL register) then further interrupt pulses are generated whenever a selected EOC event occurs even if the flag bit is set. If the continuous mode is not enabled, then no further interrupt pulses are generated until the user clears this flag bit using the ADCINTFLGCLR register. Rather, an ADC interrupt overflow event occurs in the ADCINTOVF register. Reset type: SYSRSn

Table 18-28. ADCINTFLG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	ADCINT1	R	0h	ADC Interrupt 1 Flag. Reading these flags indicates if the associated ADCINT pulse was generated since the last clear. 0 No ADC interrupt pulse generated 1 ADC interrupt pulse generated If the ADC interrupt is placed in continue to interrupt mode (INTSEL register) then further interrupt pulses are generated whenever a selected EOC event occurs even if the flag bit is set. If the continuous mode is not enabled, then no further interrupt pulses are generated until the user clears this flag bit using the ADCINTFLGCLR register. Rather, an ADC interrupt overflow event occurs in the ADCINTOVF register. Reset type: SYSRSn

18.3.20 ADCINTFLGFRC Register (Offset = 1020h) [Reset = 00000000h]

ADCINTFLGFRC is shown in Table 18-29.

Return to the Summary Table.

ADC Interrupt Flag Force Register

Table 18-29. ADCINTFLGFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	Reserved
19	ADCDMAINT4	R-0/W1S	0h	ADC DMA interrupt 4 Flag Force. Reads return 0. 0 No action 1 Forces ADCDMAINT4 flags in the ADCINTFLG and ADCRAWINTFLG registers. Reset type: SYSRSn
18	ADCDMAINT3	R-0/W1S	0h	ADC DMA interrupt 3 Flag Force. Reads return 0. 0 No action 1 Forces ADCDMAINT3 flags in the ADCINTFLG and ADCRAWINTFLG registers. Reset type: SYSRSn
17	ADCDMAINT2	R-0/W1S	0h	ADC DMA interrupt 2 Flag Force. Reads return 0. 0 No action 1 Forces ADCDMAINT2 flags in the ADCINTFLG and ADCRAWINTFLG registers. Reset type: SYSRSn
16	ADCDMAINT1	R-0/W1S	0h	ADC DMA interrupt 1 Flag Force. Reads return 0. 0 No action 1 Forces ADCDMAINT1 flags in the ADCINTFLG and ADCRAWINTFLG registers. Reset type: SYSRSn
15-4	RESERVED	R	0h	Reserved
3	ADCINT4	R-0/W1S	0h	ADC Interrupt 4 Flag Force. Reads return 0. 0 No action 1 Forces ADCINT4 flags in the ADCINTFLG and ADCRAWINTFLG registers. Reset type: SYSRSn
2	ADCINT3	R-0/W1S	0h	ADC Interrupt 3 Flag Force. Reads return 0. 0 No action 1 Forces ADCINT3 flags in the ADCINTFLG and ADCRAWINTFLG registers. Reset type: SYSRSn
1	ADCINT2	R-0/W1S	0h	ADC Interrupt 2 Flag Force. Reads return 0. 0 No action 1 Forces ADCINT2 flags in the ADCINTFLG and ADCRAWINTFLG registers. Reset type: SYSRSn
0	ADCINT1	R-0/W1S	0h	ADC Interrupt 1 Flag Force. Reads return 0. 0 No action 1 Forces ADCINT1 flags in the ADCINTFLG and ADCRAWINTFLG registers. Reset type: SYSRSn

18.3.21 ADCINTFLGCLR Register (Offset = 1024h) [Reset = 00000000h]

ADCINTFLGCLR is shown in Table 18-30.

Return to the Summary Table.

ADC Interrupt Flag Clear Register

Table 18-30. ADCINTFLGCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	Reserved
19	ADCDMAINT4	R-0/W1C	0h	ADC DMA Interrupt 4 Flag Clear. Reads return 0. 0 No action 1 Clears ADDMACINT4 flags in the ADCINTFLG ,ADCRAWINTFLG registers. If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority and the overflow bit will not be set Reset type: SYSRSn
18	ADCDMAINT3	R-0/W1C	0h	ADC DMA Interrupt 3 Flag Clear. Reads return 0. 0 No action 1 Clears ADDMACINT3 flags in the ADCINTFLG ,ADCRAWINTFLG registers. If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority and the overflow bit will not be set Reset type: SYSRSn
17	ADCDMAINT2	R-0/W1C	0h	ADC DMA Interrupt 2 Flag Clear. Reads return 0. 0 No action 1 Clears ADDMACINT2 flags in the ADCINTFLG ,ADCRAWINTFLG registers. If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority and the overflow bit will not be set Reset type: SYSRSn
16	ADCDMAINT1	R-0/W1C	0h	ADC DMA Interrupt 1 Flag Clear. Reads return 0. 0 No action 1 Clears ADDMACINT1 flags in the ADCINTFLG ,ADCRAWINTFLG registers. If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority and the overflow bit will not be set Reset type: SYSRSn
15-4	RESERVED	R	0h	Reserved
3	ADCINT4	R-0/W1C	0h	ADC Interrupt 4 Flag Clear. Reads return 0. 0 No action 1 Clears ADCINT4 and ADCINT4RESULT flags in the ADCINTFLG,, ADCRAWINTFLG registers. If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority and the overflow bit will not be set Reset type: SYSRSn
2	ADCINT3	R-0/W1C	0h	ADC Interrupt 3 Flag Clear. Reads return 0. 0 No action 1 Clears ADCINT3 and ADCINT3RESULT flags in the ADCINTFLG, ADCRAWINTFLG registers. If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority and the overflow bit will not be set Reset type: SYSRSn

Table 18-30. ADCINTFLGCLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	ADCINT2	R-0/W1C	0h	ADC Interrupt 2 Flag Clear. Reads return 0. 0 No action 1 Clears ADCINT2 and ADCINT2RESULT flags in the ADCINTFLG, ADCRAWINTFLG registers. If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority and the overflow bit will not be set Reset type: SYSRSn
0	ADCINT1	R-0/W1C	0h	ADC Interrupt 1 Flag Clear. Reads return 0. 0 No action 1 Clears ADCINT1 and ADCINT1RESULT flags in the ADCINTFLG, ADCRAWINTFLG registers. If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority and the overflow bit will not be set Reset type: SYSRSn

18.3.22 ADCINTOVF Register (Offset = 1028h) [Reset = 00000000h]

ADCINTOVF is shown in Table 18-31.

Return to the Summary Table.

ADC Interrupt Overflow Register

Table 18-31. ADCINTOVF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	Reserved
19	ADCDMAINT4OVF	R	0h	ADC DMA Interrupt 4 Overflow Flags Indicates if an overflow occurred when generating ADCINT pulses. If the respective ADCINTFLG bit is set and a selected additional EOC trigger is generated, then an overflow condition occurs. 0 No ADC DMA Interrupt overflow event detected. 1 ADC DMA Interrupt overflow event detected. The overflow bit does not care about the continuous mode bit state. An overflow condition is generated irrespective of this mode selection. Reset type: SYSRSn
18	ADCDMAINT3OVF	R	0h	ADC DMA Interrupt 3 Overflow Flags Indicates if an overflow occurred when generating ADCINT pulses. If the respective ADCINTFLG bit is set and a selected additional EOC trigger is generated, then an overflow condition occurs. 0 No ADC DMA Interrupt overflow event detected. 1 ADC DMA Interrupt overflow event detected. The overflow bit does not care about the continuous mode bit state. An overflow condition is generated irrespective of this mode selection. Reset type: SYSRSn
17	ADCDMAINT2OVF	R	0h	ADC DMA Interrupt 2 Overflow Flags Indicates if an overflow occurred when generating ADCINT pulses. If the respective ADCINTFLG bit is set and a selected additional EOC trigger is generated, then an overflow condition occurs. 0 No ADC DMA Interrupt overflow event detected. 1 ADC DMA Interrupt overflow event detected. The overflow bit does not care about the continuous mode bit state. An overflow condition is generated irrespective of this mode selection. Reset type: SYSRSn
16	ADCDMAINT1OVF	R	0h	ADC DMA Interrupt 1 Overflow Flags Indicates if an overflow occurred when generating ADCINT pulses. If the respective ADCINTFLG bit is set and a selected additional EOC trigger is generated, then an overflow condition occurs. 0 No ADC DMA Interrupt overflow event detected. 1 ADC DMA Interrupt overflow event detected. The overflow bit does not care about the continuous mode bit state. An overflow condition is generated irrespective of this mode selection. Reset type: SYSRSn
15-4	RESERVED	R	0h	Reserved
3	ADCINT4OVF	R	0h	ADC Interrupt 4 Overflow Flags Indicates if an overflow occurred when generating ADCINT pulses. If the respective ADCINTFLG bit is set and a selected additional EOC trigger is generated, then an overflow condition occurs. 0 No ADC interrupt overflow event detected. 1 ADC Interrupt overflow event detected. The overflow bit does not care about the continuous mode bit state. An overflow condition is generated irrespective of this mode selection. Reset type: SYSRSn

Table 18-31. ADCINTOVF Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	ADCINT3OVF	R	0h	ADC Interrupt 3 Overflow Flags Indicates if an overflow occurred when generating ADCINT pulses. If the respective ADCINTFLG bit is set and a selected additional EOC trigger is generated, then an overflow condition occurs. 0 No ADC interrupt overflow event detected. 1 ADC Interrupt overflow event detected. The overflow bit does not care about the continuous mode bit state. An overflow condition is generated irrespective of this mode selection. Reset type: SYSRSn
1	ADCINT2OVF	R	0h	ADC Interrupt 2 Overflow Flags Indicates if an overflow occurred when generating ADCINT pulses. If the respective ADCINTFLG bit is set and a selected additional EOC trigger is generated, then an overflow condition occurs. 0 No ADC interrupt overflow event detected. 1 ADC Interrupt overflow event detected. The overflow bit does not care about the continuous mode bit state. An overflow condition is generated irrespective of this mode selection. Reset type: SYSRSn
0	ADCINT1OVF	R	0h	ADC Interrupt 1 Overflow Flags Indicates if an overflow occurred when generating ADCINT pulses. If the respective ADCINTFLG bit is set and a selected additional EOC trigger is generated, then an overflow condition occurs. 0 No ADC interrupt overflow event detected. 1 ADC Interrupt overflow event detected. The overflow bit does not care about the continuous mode bit state. An overflow condition is generated irrespective of this mode selection. Reset type: SYSRSn

18.3.23 ADCINTOVFCLR Register (Offset = 102Ch) [Reset = 00000000h]

ADCINTOVFCLR is shown in Table 18-32.

Return to the Summary Table.

ADC Interrupt Overflow Clear Register

Table 18-32. ADCINTOVFCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	Reserved
19	ADCDMAINT4OVF	R-0/W1C	0h	ADC DMA Interrupt 4 Overflow Clear Bits 0 No action. 1 Clears the respective overflow bit in the ADCINTOVF register. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCINTOVF register, then hardware has priority and the ADCINTOVF bit will be set. Reset type: SYSRSn
18	ADCDMAINT3OVF	R-0/W1C	0h	ADC DMA Interrupt 3 Overflow Clear Bits 0 No action. 1 Clears the respective overflow bit in the ADCINTOVF register. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCINTOVF register, then hardware has priority and the ADCINTOVF bit will be set. Reset type: SYSRSn
17	ADCDMAINT2OVF	R-0/W1C	0h	ADC DMA Interrupt 2 Overflow Clear Bits 0 No action. 1 Clears the respective overflow bit in the ADCINTOVF register. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCINTOVF register, then hardware has priority and the ADCINTOVF bit will be set. Reset type: SYSRSn
16	ADCDMAINT1OVF	R-0/W1C	0h	ADC DMA Interrupt 1 Overflow Clear Bits 0 No action. 1 Clears the respective overflow bit in the ADCINTOVF register. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCINTOVF register, then hardware has priority and the ADCINTOVF bit will be set. Reset type: SYSRSn
15-4	RESERVED	R	0h	Reserved
3	ADCINT4OVF	R-0/W1C	0h	ADC Interrupt 4 Overflow Clear Bits 0 No action. 1 Clears the respective overflow bit in the ADCINTOVF register. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCINTOVF register, then hardware has priority and the ADCINTOVF bit will be set. Reset type: SYSRSn
2	ADCINT3OVF	R-0/W1C	0h	ADC Interrupt 3 Overflow Clear Bits 0 No action. 1 Clears the respective overflow bit in the ADCINTOVF register. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCINTOVF register, then hardware has priority and the ADCINTOVF bit will be set. Reset type: SYSRSn
1	ADCINT2OVF	R-0/W1C	0h	ADC Interrupt 2 Overflow Clear Bits 0 No action. 1 Clears the respective overflow bit in the ADCINTOVF register. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCINTOVF register, then hardware has priority and the ADCINTOVF bit will be set. Reset type: SYSRSn

Table 18-32. ADCINTOVFCLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	ADCINT1OVF	R-0/W1C	0h	ADC Interrupt 1 Overflow Clear Bits 0 No action. 1 Clears the respective overflow bit in the ADCINTOVF register. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCINTOVF register, then hardware has priority and the ADCINTOVF bit will be set. Reset type: SYSRSn

18.3.24 ADCSOCFLG1 Register (Offset = 103Ch) [Reset = 00000000h]

ADCSOCFLG1 is shown in Table 18-33.

Return to the Summary Table.

ADC SOC Flag 1 Register

Table 18-33. ADCSOCFLG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	SOC15	R	0h	SOC15 Start of Conversion Flag. Indicates the state of SOC15 conversions. 0 No sample pending for SOC15. 1 Trigger has been received and sample is pending for SOC15. This bit will be automatically cleared when the SOC15 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
14	SOC14	R	0h	SOC14 Start of Conversion Flag. Indicates the state of SOC14 conversions. 0 No sample pending for SOC14. 1 Trigger has been received and sample is pending for SOC14. This bit will be automatically cleared when the SOC14 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
13	SOC13	R	0h	SOC13 Start of Conversion Flag. Indicates the state of SOC13 conversions. 0 No sample pending for SOC13. 1 Trigger has been received and sample is pending for SOC13. This bit will be automatically cleared when the SOC13 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
12	SOC12	R	0h	SOC12 Start of Conversion Flag. Indicates the state of SOC12 conversions. 0 No sample pending for SOC12. 1 Trigger has been received and sample is pending for SOC12. This bit will be automatically cleared when the SOC12 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn

Table 18-33. ADCSOCFLG1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	SOC11	R	0h	SOC11 Start of Conversion Flag. Indicates the state of SOC11 conversions. 0 No sample pending for SOC11. 1 Trigger has been received and sample is pending for SOC11. This bit will be automatically cleared when the SOC11 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
10	SOC10	R	0h	SOC10 Start of Conversion Flag. Indicates the state of SOC10 conversions. 0 No sample pending for SOC10. 1 Trigger has been received and sample is pending for SOC10. This bit will be automatically cleared when the SOC10 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
9	SOC9	R	0h	SOC9 Start of Conversion Flag. Indicates the state of SOC9 conversions. 0 No sample pending for SOC9. 1 Trigger has been received and sample is pending for SOC9. This bit will be automatically cleared when the SOC9 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
8	SOC8	R	0h	SOC8 Start of Conversion Flag. Indicates the state of SOC8 conversions. 0 No sample pending for SOC8. 1 Trigger has been received and sample is pending for SOC8. This bit will be automatically cleared when the SOC8 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
7	SOC7	R	0h	SOC7 Start of Conversion Flag. Indicates the state of SOC7 conversions. 0 No sample pending for SOC7. 1 Trigger has been received and sample is pending for SOC7. This bit will be automatically cleared when the SOC7 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn

Table 18-33. ADCSOCFLG1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	SOC6	R	0h	SOC6 Start of Conversion Flag. Indicates the state of SOC6 conversions. 0 No sample pending for SOC6. 1 Trigger has been received and sample is pending for SOC6. This bit will be automatically cleared when the SOC6 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
5	SOC5	R	0h	SOC5 Start of Conversion Flag. Indicates the state of SOC5 conversions. 0 No sample pending for SOC5. 1 Trigger has been received and sample is pending for SOC5. This bit will be automatically cleared when the SOC5 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
4	SOC4	R	0h	SOC4 Start of Conversion Flag. Indicates the state of SOC4 conversions. 0 No sample pending for SOC4. 1 Trigger has been received and sample is pending for SOC4. This bit will be automatically cleared when the SOC4 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
3	SOC3	R	0h	SOC3 Start of Conversion Flag. Indicates the state of SOC3 conversions. 0 No sample pending for SOC3. 1 Trigger has been received and sample is pending for SOC3. This bit will be automatically cleared when the SOC3 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
2	SOC2	R	0h	SOC2 Start of Conversion Flag. Indicates the state of SOC2 conversions. 0 No sample pending for SOC2. 1 Trigger has been received and sample is pending for SOC2. This bit will be automatically cleared when the SOC2 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn

Table 18-33. ADCSOCFLG1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	SOC1	R	0h	SOC1 Start of Conversion Flag. Indicates the state of SOC1 conversions. 0 No sample pending for SOC1. 1 Trigger has been received and sample is pending for SOC1. This bit will be automatically cleared when the SOC1 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
0	SOC0	R	0h	SOC0 Start of Conversion Flag. Indicates the state of SOC0 conversions. 0 No sample pending for SOC0. 1 Trigger has been received and sample is pending for SOC0. This bit will be automatically cleared when the SOC0 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn

18.3.25 ADCSOCOVF1 Register (Offset = 1044h) [Reset = 00000000h]

ADCSOCOVF1 is shown in Table 18-34.

Return to the Summary Table.

ADC SOC Overflow 1 Register

Table 18-34. ADCSOCOVF1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	SOC15OVF	R	0h	SOC15 Start of Conversion Overflow Flag. Indicates an SOC15 event was generated in hardware while an existing SOC15 event was already pending. 0 No SOC15 event overflow. 1 SOC15 event overflow. An overflow condition does not stop SOC15 events from being processed. It simply is an indication that a hardware trigger was missed. Reset type: SYSRSn
14	SOC14OVF	R	0h	SOC14 Start of Conversion Overflow Flag. Indicates an SOC14 event was generated in hardware while an existing SOC14 event was already pending. 0 No SOC14 event overflow. 1 SOC14 event overflow. An overflow condition does not stop SOC14 events from being processed. It simply is an indication that a hardware trigger was missed. Reset type: SYSRSn
13	SOC13OVF	R	0h	SOC13 Start of Conversion Overflow Flag. Indicates an SOC13 event was generated in hardware while an existing SOC13 event was already pending. 0 No SOC13 event overflow. 1 SOC13 event overflow. An overflow condition does not stop SOC13 events from being processed. It simply is an indication that a hardware trigger was missed. Reset type: SYSRSn
12	SOC12OVF	R	0h	SOC12 Start of Conversion Overflow Flag. Indicates an SOC12 event was generated in hardware while an existing SOC12 event was already pending. 0 No SOC12 event overflow. 1 SOC12 event overflow. An overflow condition does not stop SOC12 events from being processed. It simply is an indication that a hardware trigger was missed. Reset type: SYSRSn
11	SOC11OVF	R	0h	SOC11 Start of Conversion Overflow Flag. Indicates an SOC11 event was generated in hardware while an existing SOC11 event was already pending. 0 No SOC11 event overflow. 1 SOC11 event overflow. An overflow condition does not stop SOC11 events from being processed. It simply is an indication that a hardware trigger was missed. Reset type: SYSRSn

Table 18-34. ADCSOCOVF1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	SOC10OVF	R	0h	SOC10 Start of Conversion Overflow Flag. Indicates an SOC10 event was generated in hardware while an existing SOC10 event was already pending. 0 No SOC10 event overflow. 1 SOC10 event overflow. An overflow condition does not stop SOC10 events from being processed. It simply is an indication that a hardware trigger was missed. Reset type: SYSRSn
9	SOC9OVF	R	0h	SOC9 Start of Conversion Overflow Flag. Indicates an SOC9 event was generated in hardware while an existing SOC9 event was already pending. 0 No SOC9 event overflow. 1 SOC9 event overflow. An overflow condition does not stop SOC9 events from being processed. It simply is an indication that a hardware trigger was missed. Reset type: SYSRSn
8	SOC8OVF	R	0h	SOC8 Start of Conversion Overflow Flag. Indicates an SOC8 event was generated in hardware while an existing SOC8 event was already pending. 0 No SOC8 event overflow. 1 SOC8 event overflow. An overflow condition does not stop SOC8 events from being processed. It simply is an indication that a hardware trigger was missed. Reset type: SYSRSn
7	SOC7OVF	R	0h	SOC7 Start of Conversion Overflow Flag. Indicates an SOC7 event was generated in hardware while an existing SOC7 event was already pending. 0 No SOC7 event overflow. 1 SOC7 event overflow. An overflow condition does not stop SOC7 events from being processed. It simply is an indication that a hardware trigger was missed. Reset type: SYSRSn
6	SOC6OVF	R	0h	SOC6 Start of Conversion Overflow Flag. Indicates an SOC6 event was generated in hardware while an existing SOC6 event was already pending. 0 No SOC6 event overflow. 1 SOC6 event overflow. An overflow condition does not stop SOC6 events from being processed. It simply is an indication that a hardware trigger was missed. Reset type: SYSRSn
5	SOC5OVF	R	0h	SOC5 Start of Conversion Overflow Flag. Indicates an SOC5 event was generated in hardware while an existing SOC5 event was already pending. 0 No SOC5 event overflow. 1 SOC5 event overflow. An overflow condition does not stop SOC5 events from being processed. It simply is an indication that a hardware trigger was missed. Reset type: SYSRSn

Table 18-34. ADCSOCOVF1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	SOC4OVF	R	0h	SOC4 Start of Conversion Overflow Flag. Indicates an SOC4 event was generated in hardware while an existing SOC4 event was already pending. 0 No SOC4 event overflow. 1 SOC4 event overflow. An overflow condition does not stop SOC4 events from being processed. It simply is an indication that a hardware trigger was missed. Reset type: SYSRSn
3	SOC3OVF	R	0h	SOC3 Start of Conversion Overflow Flag. Indicates an SOC3 event was generated in hardware while an existing SOC3 event was already pending. 0 No SOC3 event overflow. 1 SOC3 event overflow. An overflow condition does not stop SOC3 events from being processed. It simply is an indication that a hardware trigger was missed. Reset type: SYSRSn
2	SOC2OVF	R	0h	SOC2 Start of Conversion Overflow Flag. Indicates an SOC2 event was generated in hardware while an existing SOC2 event was already pending. 0 No SOC2 event overflow. 1 SOC2 event overflow. An overflow condition does not stop SOC2 events from being processed. It simply is an indication that a hardware trigger was missed. Reset type: SYSRSn
1	SOC1OVF	R	0h	SOC1 Start of Conversion Overflow Flag. Indicates an SOC1 event was generated in hardware while an existing SOC1 event was already pending. 0 No SOC1 event overflow. 1 SOC1 event overflow. An overflow condition does not stop SOC1 events from being processed. It simply is an indication that a hardware trigger was missed. Reset type: SYSRSn
0	SOC0OVF	R	0h	SOC0 Start of Conversion Overflow Flag. Indicates an SOC0 event was generated in hardware while an existing SOC0 event was already pending. 0 No SOC0 event overflow. 1 SOC0 event overflow. An overflow condition does not stop SOC0 events from being processed. It simply is an indication that a hardware trigger was missed. Reset type: SYSRSn

18.3.26 ADCSOCOVFCLR1 Register (Offset = 1048h) [Reset = 00000000h]

ADCSOCOVFCLR1 is shown in Table 18-35.

Return to the Summary Table.

ADC SOC Overflow Clear 1 Register

Table 18-35. ADCSOCOVFCLR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	SOC15OVF	R-0/W1C	0h	SOC15 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC15 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC15 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
14	SOC14OVF	R-0/W1C	0h	SOC14 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC14 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC14 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
13	SOC13OVF	R-0/W1C	0h	SOC13 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC13 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC13 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
12	SOC12OVF	R-0/W1C	0h	SOC12 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC12 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC12 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
11	SOC11OVF	R-0/W1C	0h	SOC11 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC11 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC11 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn

Table 18-35. ADCSOCOVFCLR1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	SOC10OVF	R-0/W1C	0h	SOC10 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC10 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC10 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
9	SOC9OVF	R-0/W1C	0h	SOC9 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC9 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC9 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
8	SOC8OVF	R-0/W1C	0h	SOC8 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC8 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC8 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
7	SOC7OVF	R-0/W1C	0h	SOC7 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC7 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC7 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
6	SOC6OVF	R-0/W1C	0h	SOC6 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC6 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC6 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
5	SOC5OVF	R-0/W1C	0h	SOC5 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC5 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC5 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn

Table 18-35. ADCSOCOVFCLR1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	SOC4OVF	R-0/W1C	0h	SOC4 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC4 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC4 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
3	SOC3OVF	R-0/W1C	0h	SOC3 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC3 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC3 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
2	SOC2OVF	R-0/W1C	0h	SOC2 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC2 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC2 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
1	SOC1OVF	R-0/W1C	0h	SOC1 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC1 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC1 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
0	SOC0OVF	R-0/W1C	0h	SOC0 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC0 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC0 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn

18.3.27 ADCSOC0CTL Register (Offset = 104Ch) [Reset = 00000000h]

ADCSOC0CTL is shown in Table 18-36.

Return to the Summary Table.

ADC SOC0 Control Register

Table 18-36. ADCSOC0CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25	COMPEN	R/W	0h	SOC0 Threshold comparator enable. Reset type: SYSRSn
24-20	RESERVED	R	0h	Reserved
19-15	CHSEL	R/W	0h	SOC0 Channel Select. Selects the channel to be converted when SOC0 is received by the ADC. 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Reset type: SYSRSn
14-0	RESERVED	R	0h	Reserved

18.3.28 ADCSOC1CTL Register (Offset = 1050h) [Reset = 00000000h]

ADCSOC1CTL is shown in Table 18-37.

Return to the Summary Table.

ADC SOC1 Control Register

Table 18-37. ADCSOC1CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25	COMPEN	R/W	0h	SOC1 Threshold comparator enable. Reset type: SYSRSn
24-20	RESERVED	R	0h	Reserved
19-15	CHSEL	R/W	0h	SOC1 Channel Select. Selects the channel to be converted when SOC1 is received by the ADC. 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Reset type: SYSRSn
14-0	RESERVED	R	0h	Reserved

18.3.29 ADCSOC2CTL Register (Offset = 1054h) [Reset = 00000000h]

ADCSOC2CTL is shown in Table 18-38.

Return to the Summary Table.

ADC SOC2 Control Register

Table 18-38. ADCSOC2CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25	COMPEN	R/W	0h	SOC2 Threshold comparator enable. Reset type: SYSRSn
24-20	RESERVED	R	0h	Reserved
19-15	CHSEL	R/W	0h	SOC2 Channel Select. Selects the channel to be converted when SOC2 is received by the ADC. 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Reset type: SYSRSn
14-0	RESERVED	R	0h	Reserved

18.3.30 ADCSOC3CTL Register (Offset = 1058h) [Reset = 00000000h]

ADCSOC3CTL is shown in Table 18-39.

Return to the Summary Table.

ADC SOC3 Control Register

Table 18-39. ADCSOC3CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25	COMPEN	R/W	0h	SOC3 Threshold comparator enable. Reset type: SYSRSn
24-20	RESERVED	R	0h	Reserved
19-15	CHSEL	R/W	0h	SOC3 Channel Select. Selects the channel to be converted when SOC3 is received by the ADC. 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Reset type: SYSRSn
14-0	RESERVED	R	0h	Reserved

18.3.31 ADCSOC4CTL Register (Offset = 105Ch) [Reset = 00000000h]

ADCSOC4CTL is shown in Table 18-40.

Return to the Summary Table.

ADC SOC4 Control Register

Table 18-40. ADCSOC4CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25	COMPEN	R/W	0h	SOC4 Threshold comparator enable. Reset type: SYSRSn
24-20	RESERVED	R	0h	Reserved
19-15	CHSEL	R/W	0h	SOC4 Channel Select. Selects the channel to be converted when SOC4 is received by the ADC. 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Reset type: SYSRSn
14-0	RESERVED	R	0h	Reserved

18.3.32 ADCSOC5CTL Register (Offset = 1060h) [Reset = 00000000h]

ADCSOC5CTL is shown in Table 18-41.

Return to the Summary Table.

ADC SOC5 Control Register

Table 18-41. ADCSOC5CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25	COMPEN	R/W	0h	SOC5 Threshold comparator enable. Reset type: SYSRSn
24-20	RESERVED	R	0h	Reserved
19-15	CHSEL	R/W	0h	SOC5 Channel Select. Selects the channel to be converted when SOC5 is received by the ADC. 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Reset type: SYSRSn
14-0	RESERVED	R	0h	Reserved

18.3.33 ADCSOC6CTL Register (Offset = 1064h) [Reset = 00000000h]

ADCSOC6CTL is shown in Table 18-42.

Return to the Summary Table.

ADC SOC6 Control Register

Table 18-42. ADCSOC6CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25	COMPEN	R/W	0h	SOC6 Threshold comparator enable. Reset type: SYSRSn
24-20	RESERVED	R	0h	Reserved
19-15	CHSEL	R/W	0h	SOC6 Channel Select. Selects the channel to be converted when SOC6 is received by the ADC. 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Reset type: SYSRSn
14-0	RESERVED	R	0h	Reserved

18.3.34 ADCSOC7CTL Register (Offset = 1068h) [Reset = 00000000h]

ADCSOC7CTL is shown in Table 18-43.

Return to the Summary Table.

ADC SOC7 Control Register

Table 18-43. ADCSOC7CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25	COMPEN	R/W	0h	SOC7 Threshold comparator enable. Reset type: SYSRSn
24-20	RESERVED	R	0h	Reserved
19-15	CHSEL	R/W	0h	SOC7 Channel Select. Selects the channel to be converted when SOC7 is received by the ADC. 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Reset type: SYSRSn
14-0	RESERVED	R	0h	Reserved

18.3.35 ADCSOC8CTL Register (Offset = 106Ch) [Reset = 00000000h]

ADCSOC8CTL is shown in Table 18-44.

Return to the Summary Table.

ADC SOC8 Control Register

Table 18-44. ADCSOC8CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25	COMPEN	R/W	0h	SOC8 Threshold comparator enable. Reset type: SYSRSn
24-20	RESERVED	R	0h	Reserved
19-15	CHSEL	R/W	0h	SOC8 Channel Select. Selects the channel to be converted when SOC8 is received by the ADC. 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Reset type: SYSRSn
14-0	RESERVED	R	0h	Reserved

18.3.36 ADCSOC9CTL Register (Offset = 1070h) [Reset = 00000000h]

ADCSOC9CTL is shown in Table 18-45.

Return to the Summary Table.

ADC SOC9 Control Register

Table 18-45. ADCSOC9CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25	COMPEN	R/W	0h	SOC9 Threshold comparator enable. Reset type: SYSRSn
24-20	RESERVED	R	0h	Reserved
19-15	CHSEL	R/W	0h	SOC9 Channel Select. Selects the channel to be converted when SOC9 is received by the ADC. 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Reset type: SYSRSn
14-0	RESERVED	R	0h	Reserved

18.3.37 ADCSOC10CTL Register (Offset = 1074h) [Reset = 00000000h]

ADCSOC10CTL is shown in Table 18-46.

Return to the Summary Table.

ADC SOC10 Control Register

Table 18-46. ADCSOC10CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25	COMPEN	R/W	0h	SOC10 Threshold comparator enable. Reset type: SYSRSn
24-20	RESERVED	R	0h	Reserved
19-15	CHSEL	R/W	0h	SOC10 Channel Select. Selects the channel to be converted when SOC10 is received by the ADC. 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Reset type: SYSRSn
14-0	RESERVED	R	0h	Reserved

18.3.38 ADCSOC11CTL Register (Offset = 1078h) [Reset = 00000000h]

ADCSOC11CTL is shown in Table 18-47.

Return to the Summary Table.

ADC SOC11 Control Register

Table 18-47. ADCSOC11CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25	COMPEN	R/W	0h	SOC11 Threshold comparator enable. Reset type: SYSRSn
24-20	RESERVED	R	0h	Reserved
19-15	CHSEL	R/W	0h	SOC11 Channel Select. Selects the channel to be converted when SOC11 is received by the ADC. 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Reset type: SYSRSn
14-0	RESERVED	R	0h	Reserved

18.3.39 ADCSOC12CTL Register (Offset = 107Ch) [Reset = 00000000h]

ADCSOC12CTL is shown in Table 18-48.

Return to the Summary Table.

ADC SOC12 Control Register

Table 18-48. ADCSOC12CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25	COMPEN	R/W	0h	SOC12 Threshold comparator enable. Reset type: SYSRSn
24-20	RESERVED	R	0h	Reserved
19-15	CHSEL	R/W	0h	SOC12 Channel Select. Selects the channel to be converted when SOC12 is received by the ADC. 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Reset type: SYSRSn
14-0	RESERVED	R	0h	Reserved

18.3.40 ADCSOC13CTL Register (Offset = 1080h) [Reset = 00000000h]

ADCSOC13CTL is shown in Table 18-49.

Return to the Summary Table.

ADC SOC13 Control Register

Table 18-49. ADCSOC13CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25	COMPEN	R/W	0h	SOC13 Threshold comparator enable. Reset type: SYSRSn
24-20	RESERVED	R	0h	Reserved
19-15	CHSEL	R/W	0h	SOC13 Channel Select. Selects the channel to be converted when SOC13 is received by the ADC. 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Reset type: SYSRSn
14-0	RESERVED	R	0h	Reserved

18.3.41 ADCSOC14CTL Register (Offset = 1084h) [Reset = 00000000h]

ADCSOC14CTL is shown in Table 18-50.

Return to the Summary Table.

ADC SOC14 Control Register

Table 18-50. ADCSOC14CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25	COMPEN	R/W	0h	SOC14 Threshold comparator enable. Reset type: SYSRSn
24-20	RESERVED	R	0h	Reserved
19-15	CHSEL	R/W	0h	SOC14 Channel Select. Selects the channel to be converted when SOC14 is received by the ADC. 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Reset type: SYSRSn
14-0	RESERVED	R	0h	Reserved

18.3.42 ADCSOC15CTL Register (Offset = 1088h) [Reset = 00000000h]

ADCSOC15CTL is shown in Table 18-51.

Return to the Summary Table.

ADC SOC15 Control Register

Table 18-51. ADCSOC15CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25	COMPEN	R/W	0h	SOC15 Threshold comparator enable. Reset type: SYSRSn
24-20	RESERVED	R	0h	Reserved
19-15	CHSEL	R/W	0h	SOC15 Channel Select. Selects the channel to be converted when SOC15 is received by the ADC. 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Reset type: SYSRSn
14-0	RESERVED	R	0h	Reserved

18.3.43 ADCEVTSTAT Register (Offset = 10CCh) [Reset = 00000000h]

ADCEVTSTAT is shown in Table 18-52.

Return to the Summary Table.

ADC Event Status Register

Table 18-52. ADCEVTSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	PPB4INLIMIT	R	0h	Post Processing Block 4 Within trip limit Flag. When set indicates a digital compare within limit event has occurred. This will be set when the PPB result is either in between or equal to the TRIPHI and TRIPLO thresholds. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
14	PPB4ZERO	R	0h	Post Processing Block 4 Zero Crossing Flag. When set indicates the ADCPPB4RESULT register has changed sign. This bit is gated by EOC signal. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
13	PPB4TRIPLO	R	0h	Post Processing Block 4 Trip Low Flag. When set indicates a digital compare trip low event has occurred. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
12	PPB4TRIPHI	R	0h	Post Processing Block 4 Trip High Flag. When set indicates a digital compare trip high event has occurred. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
11	PPB3INLIMIT	R	0h	Post Processing Block 3 Within trip limit Flag. When set indicates a digital compare within limit event has occurred. This will be set when the PPB result is either in between or equal to the TRIPHI and TRIPLO thresholds. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn

Table 18-52. ADCEVTSTAT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	PPB3ZERO	R	0h	Post Processing Block 3 Zero Crossing Flag. When set indicates the ADCPPB3RESULT register has changed sign. This bit is gated by EOC signal. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
9	PPB3TRIPLO	R	0h	Post Processing Block 3 Trip Low Flag. When set indicates a digital compare trip low event has occurred. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
8	PPB3TRIPHI	R	0h	Post Processing Block 3 Trip High Flag. When set indicates a digital compare trip high event has occurred. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
7	PPB2INLIMIT	R	0h	Post Processing Block 2 Within trip limit Flag. When set indicates a digital compare within limit event has occurred. This will be set when the PPB result is either in between or equal to the TRIPHI and TRIPLO thresholds. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
6	PPB2ZERO	R	0h	Post Processing Block 2 Zero Crossing Flag. When set indicates the ADCPPB2RESULT register has changed sign. This bit is gated by EOC signal. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
5	PPB2TRIPLO	R	0h	Post Processing Block 2 Trip Low Flag. When set indicates a digital compare trip low event has occurred. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn

Table 18-52. ADCEVTSTAT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	PPB2TRIPHI	R	0h	Post Processing Block 2 Trip High Flag. When set indicates a digital compare trip high event has occurred. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
3	PPB1INLIMIT	R	0h	Post Processing Block 1 Within trip limit Flag. When set indicates a digital compare within limit event has occurred. This will be set when the PPB result is either in between or equal to the TRIPHI and TRIPLO thresholds. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
2	PPB1ZERO	R	0h	Post Processing Block 1 Zero Crossing Flag. When set indicates the ADCPPB1RESULT register has changed sign. This bit is gated by EOC signal. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
1	PPB1TRIPLO	R	0h	Post Processing Block 1 Trip Low Flag. When set indicates a digital compare trip low event has occurred. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
0	PPB1TRIPHI	R	0h	Post Processing Block 1 Trip High Flag. When set indicates a digital compare trip high event has occurred. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn

18.3.44 ADCEVTCLR Register (Offset = 10D0h) [Reset = 00000000h]

ADCEVTCLR is shown in Table 18-53.

Return to the Summary Table.

ADC Event Clear Register

Table 18-53. ADCEVTCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	PPB4INLIMIT	R-0/W1C	0h	Post Processing Block 4 Within trip limit flag Clear, Clears the corresponding within trip limit flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
14	PPB4ZERO	R-0/W1C	0h	Post Processing Block 4 Zero Crossing Clear. Clears the corresponding zero crossing flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
13	PPB4TRIPLO	R-0/W1C	0h	Post Processing Block 4 Trip Low Clear. Clears the corresponding trip low flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
12	PPB4TRIPHI	R-0/W1C	0h	Post Processing Block 4 Trip High Clear. Clears the corresponding trip high flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
11	PPB3INLIMIT	R-0/W1C	0h	Post Processing Block 3 Within trip limit flag Clear, Clears the corresponding within trip limit flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
10	PPB3ZERO	R-0/W1C	0h	Post Processing Block 3 Zero Crossing Clear. Clears the corresponding zero crossing flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
9	PPB3TRIPLO	R-0/W1C	0h	Post Processing Block 3 Trip Low Clear. Clears the corresponding trip low flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
8	PPB3TRIPHI	R-0/W1C	0h	Post Processing Block 3 Trip High Clear. Clears the corresponding trip high flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
7	PPB2INLIMIT	R-0/W1C	0h	Post Processing Block 2 Within trip limit flag Clear, Clears the corresponding within trip limit flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
6	PPB2ZERO	R-0/W1C	0h	Post Processing Block 2 Zero Crossing Clear. Clears the corresponding zero crossing flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn

Table 18-53. ADCEVTCLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	PPB2TRIPLO	R-0/W1C	0h	Post Processing Block 2 Trip Low Clear. Clears the corresponding trip low flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
4	PPB2TRIPHI	R-0/W1C	0h	Post Processing Block 2 Trip High Clear. Clears the corresponding trip high flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
3	PPB1INLIMIT	R-0/W1C	0h	Post Processing Block 1 Within trip limit flag Clear, Clears the corresponding within trip limit flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
2	PPB1ZERO	R-0/W1C	0h	Post Processing Block 1 Zero Crossing Clear. Clears the corresponding zero crossing flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
1	PPB1TRIPLO	R-0/W1C	0h	Post Processing Block 1 Trip Low Clear. Clears the corresponding trip low flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
0	PPB1TRIPHI	R-0/W1C	0h	Post Processing Block 1 Trip High Clear. Clears the corresponding trip high flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn

18.3.45 ADCEVTSEL Register (Offset = 10D4h) [Reset = 00000000h]

ADCEVTSEL is shown in Table 18-54.

Return to the Summary Table.

ADC Event Selection Register

Table 18-54. ADCEVTSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	PPB4INLIMIT	R/W	0h	Post Processing Block 4 Within trip limit event enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn
14	PPB4ZERO	R/W	0h	Post Processing Block 4 Zero Crossing Event Enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn
13	PPB4TRIPLO	R/W	0h	Post Processing Block 4 Trip Low Event Enable. Setting this bit allows the corresponding rising trip low flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn
12	PPB4TRIPHI	R/W	0h	Post Processing Block 4 Trip High Event Enable. Setting this bit allows the corresponding rising trip high flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn
11	PPB3INLIMIT	R/W	0h	Post Processing Block 3 Within trip limit event enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn
10	PPB3ZERO	R/W	0h	Post Processing Block 3 Zero Crossing Event Enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn
9	PPB3TRIPLO	R/W	0h	Post Processing Block 3 Trip Low Event Enable. Setting this bit allows the corresponding rising trip low flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn
8	PPB3TRIPHI	R/W	0h	Post Processing Block 3 Trip High Event Enable. Setting this bit allows the corresponding rising trip high flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn
7	PPB2INLIMIT	R/W	0h	Post Processing Block 2 Within trip limit event enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn
6	PPB2ZERO	R/W	0h	Post Processing Block 2 Zero Crossing Event Enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn

Table 18-54. ADCEVTSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	PPB2TRIPLO	R/W	0h	Post Processing Block 2 Trip Low Event Enable. Setting this bit allows the corresponding rising trip low flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn
4	PPB2TRIPHI	R/W	0h	Post Processing Block 2 Trip High Event Enable. Setting this bit allows the corresponding rising trip high flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn
3	PPB1INLIMIT	R/W	0h	Post Processing Block 1 Within trip limit event enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn
2	PPB1ZERO	R/W	0h	Post Processing Block 1 Zero Crossing Event Enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn
1	PPB1TRIPLO	R/W	0h	Post Processing Block 1 Trip Low Event Enable. Setting this bit allows the corresponding rising trip low flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn
0	PPB1TRIPHI	R/W	0h	Post Processing Block 1 Trip High Event Enable. Setting this bit allows the corresponding rising trip high flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn

18.3.46 ADCEVTINTSEL Register (Offset = 10D8h) [Reset = 00000000h]

ADCEVTINTSEL is shown in Table 18-55.

Return to the Summary Table.

ADC Event Interrupt Selection Register

Table 18-55. ADCEVTINTSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	PPB4INLIMIT	R/W	0h	Post Processing Block 4 Within trip limit Interrupt Enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event interrupt signal to the PIE. The flag must be cleared before it can produce additional interrupts to the PIE. Reset type: SYSRSn
14	PPB4ZERO	R/W	0h	Post Processing Block 4 Zero Crossing Interrupt Enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event interrupt signal to the PIE. The flag must be cleared before it can produce additional interrupts to the PIE. Reset type: SYSRSn
13	PPB4TRIPLO	R/W	0h	Post Processing Block 4 Trip Low Interrupt Enable. Setting this bit allows the corresponding rising trip low flag to activate the event interrupt signal to the PIE. The flag must be cleared before it can produce additional interrupts to the PIE. Reset type: SYSRSn
12	PPB4TRIPHI	R/W	0h	Post Processing Block 4 Trip High Interrupt Enable. Setting this bit allows the corresponding rising trip high flag to activate the event interrupt signal to the PIE. The flag must be cleared before it can produce additional interrupts to the PIE. Reset type: SYSRSn
11	PPB3INLIMIT	R/W	0h	Post Processing Block 3 Within trip limit Interrupt Enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event interrupt signal to the PIE. The flag must be cleared before it can produce additional interrupts to the PIE. Reset type: SYSRSn
10	PPB3ZERO	R/W	0h	Post Processing Block 3 Zero Crossing Interrupt Enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event interrupt signal to the PIE. The flag must be cleared before it can produce additional interrupts to the PIE. Reset type: SYSRSn
9	PPB3TRIPLO	R/W	0h	Post Processing Block 3 Trip Low Interrupt Enable. Setting this bit allows the corresponding rising trip low flag to activate the event interrupt signal to the PIE. The flag must be cleared before it can produce additional interrupts to the PIE. Reset type: SYSRSn
8	PPB3TRIPHI	R/W	0h	Post Processing Block 3 Trip High Interrupt Enable. Setting this bit allows the corresponding rising trip high flag to activate the event interrupt signal to the PIE. The flag must be cleared before it can produce additional interrupts to the PIE. Reset type: SYSRSn
7	PPB2INLIMIT	R/W	0h	Post Processing Block 2 Within trip limit Interrupt Enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event interrupt signal to the PIE. The flag must be cleared before it can produce additional interrupts to the PIE. Reset type: SYSRSn
6	PPB2ZERO	R/W	0h	Post Processing Block 2 Zero Crossing Interrupt Enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event interrupt signal to the PIE. The flag must be cleared before it can produce additional interrupts to the PIE. Reset type: SYSRSn

Table 18-55. ADCEVTINTSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	PPB2TRIPLO	R/W	0h	Post Processing Block 2 Trip Low Interrupt Enable. Setting this bit allows the corresponding rising trip low flag to activate the event interrupt signal to the PIE. The flag must be cleared before it can produce additional interrupts to the PIE. Reset type: SYSRSn
4	PPB2TRIPHI	R/W	0h	Post Processing Block 2 Trip High Interrupt Enable. Setting this bit allows the corresponding rising trip high flag to activate the event interrupt signal to the PIE. The flag must be cleared before it can produce additional interrupts to the PIE. Reset type: SYSRSn
3	PPB1INLIMIT	R/W	0h	Post Processing Block 1 Within trip limit Interrupt Enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event interrupt signal to the PIE. The flag must be cleared before it can produce additional interrupts to the PIE. Reset type: SYSRSn
2	PPB1ZERO	R/W	0h	Post Processing Block 1 Zero Crossing Interrupt Enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event interrupt signal to the PIE. The flag must be cleared before it can produce additional interrupts to the PIE. Reset type: SYSRSn
1	PPB1TRIPLO	R/W	0h	Post Processing Block 1 Trip Low Interrupt Enable. Setting this bit allows the corresponding rising trip low flag to activate the event interrupt signal to the PIE. The flag must be cleared before it can produce additional interrupts to the PIE. Reset type: SYSRSn
0	PPB1TRIPHI	R/W	0h	Post Processing Block 1 Trip High Interrupt Enable. Setting this bit allows the corresponding rising trip high flag to activate the event interrupt signal to the PIE. The flag must be cleared before it can produce additional interrupts to the PIE. Reset type: SYSRSn

18.3.47 ADCREV Register (Offset = 10E4h) [Reset = 00000006h]

ADCREV is shown in Table 18-56.

Return to the Summary Table.

ADC Revision Register

Table 18-56. ADCREV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	REV	R	0h	ADC Revision. To allow documentation of differences between revisions. First version is labeled as 00h. Reset type: SYSRSn
7-0	TYPE	R	6h	ADC Type. Always set to 6 for this HSADC-12b. Reset type: SYSRSn

18.3.48 ADCOFFTRIM Register (Offset = 10E8h) [Reset = 00000000h]

ADCOFFTRIM is shown in Table 18-57.

Return to the Summary Table.

ADC Offset Trim Register 1

Table 18-57. ADCOFFTRIM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	OFFTRIM	R/W	0h	ADC Offset Trim Adjusts the conversion results of the converter up or down to account for offset error in the ADC. A factory trim setting will be loaded during device boot. Offset can be corrected in the range of +7 to -8 LSBs. Value is 16*Offset in 8-bit 2's complement: 7 LSB (16*7) = 112 6 LSB (16*6) = 96 5 LSB (16*5) = 80 4 LSB (16*4) = 64 3 LSB (16*3) = 48 2 LSB (16*2) = 32 1 LSB (16*1) = 16 0 LSB (16*0) = 0 -1 LSB (16*(-1)) = 240 : : -7LSB(16*(-7)) = 144 Reset type: XRSn

18.3.49 ADCPPB1CONFIG Register (Offset = 1100h) [Reset = 00000000h]

ADCPPB1CONFIG is shown in Table 18-58.

Return to the Summary Table.

ADC PPB1 Config Register

Table 18-58. ADCPPB1CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7	RESERVED	R	0h	
6	RESERVED	R	0h	Reserved
5	CBCEN	R/W	0h	ADC Post Processing Block Cycle By Cycle Enable. When set, this bit enables the post conversion hardware processing circuit to automatically clear the ADCEVTSTAT on a conversion if the event condition is no longer present. Reset type: SYSRSn
4-0	RESERVED	R	0h	Reserved

18.3.50 ADCPPB1TRIPHI Register (Offset = 1110h) [Reset = 00000000h]

ADCPPB1TRIPHI is shown in Table 18-59.

Return to the Summary Table.

ADC PPB1 Trip High Register

Table 18-59. ADCPPB1TRIPHI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R	0h	Reserved
18-0	LIMITHI	R/W	0h	ADC Post Processing Block 1 Trip High Limit. This value sets the digital comparator trip high limit. When comparing to an ADCPPBxRESULT register, the upper bits will be ignored: - TRIPHI[18:13] will be ignored Reset type: SYSRSn

18.3.51 ADCPPB1TRIPLO Register (Offset = 1114h) [Reset = 00000000h]

ADCPPB1TRIPLO is shown in Table 18-60.

Return to the Summary Table.

ADC PPB1 Trip Low/Trigger Time Stamp Register

Table 18-60. ADCPPB1TRIPLO Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R	0h	Reserved
18-0	LIMITLO	R/W	0h	ADC Post Processing Block 1 Trip High Low Limit. This value sets the digital comparator trip low limit if ADCPPB1TRIPLO.LIMITLO2EN = 1. When comparing to an ADCPPBxRESULT register, the upper bits will be ignored: - TRIPLO[18:13] will be ignored in 12 bit mode Reset type: SYSRSn

18.3.52 ADCPPB2CONFIG Register (Offset = 1120h) [Reset = 00000000h]

ADCPPB2CONFIG is shown in Table 18-61.

Return to the Summary Table.

ADC PPB2 Config Register

Table 18-61. ADCPPB2CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7	RESERVED	R	0h	
6	RESERVED	R	0h	Reserved
5	CBCEN	R/W	0h	ADC Post Processing Block Cycle By Cycle Enable. When set, this bit enables the post conversion hardware processing circuit to automatically clear the ADCEVTSTAT on a conversion if the event condition is no longer present. Reset type: SYSRSn
4-0	RESERVED	R	0h	Reserved

18.3.53 ADCPPB2TRIPHI Register (Offset = 1130h) [Reset = 00000000h]

ADCPPB2TRIPHI is shown in Table 18-62.

Return to the Summary Table.

ADC PPB2 Trip High Register

Table 18-62. ADCPPB2TRIPHI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R	0h	Reserved
18-0	LIMITHI	R/W	0h	ADC Post Processing Block 2 Trip High Limit. This value sets the digital comparator trip high limit. When comparing to an ADCPPBxRESULT register, the upper bits will be ignored: - TRIPHI[18:13] will be ignored Reset type: SYSRSn

18.3.54 ADCPPB2TRIPLO Register (Offset = 1134h) [Reset = 00000000h]

ADCPPB2TRIPLO is shown in Table 18-63.

Return to the Summary Table.

ADC PPB2 Trip Low/Trigger Time Stamp Register

Table 18-63. ADCPPB2TRIPLO Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R	0h	Reserved
18-0	LIMITLO	R/W	0h	ADC Post Processing Block 2 Trip High Low Limit. This value sets the digital comparator trip low limit if ADCPPB2TRIPLO.LIMITLO2EN = 1. When comparing to an ADCPPBxRESULT register, the upper bits will be ignored: - TRIPLO[18:13] will be ignored in 12 bit mode Reset type: SYSRSn

18.3.55 ADCPPB3CONFIG Register (Offset = 1140h) [Reset = 00000000h]

ADCPPB3CONFIG is shown in Table 18-64.

Return to the Summary Table.

ADC PPB3 Config Register

Table 18-64. ADCPPB3CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7	RESERVED	R	0h	
6	RESERVED	R	0h	Reserved
5	CBCEN	R/W	0h	ADC Post Processing Block Cycle By Cycle Enable. When set, this bit enables the post conversion hardware processing circuit to automatically clear the ADCEVTSTAT on a conversion if the event condition is no longer present. Reset type: SYSRSn
4-0	RESERVED	R	0h	Reserved

18.3.56 ADCPPB3TRIPHI Register (Offset = 1150h) [Reset = 00000000h]

ADCPPB3TRIPHI is shown in Table 18-65.

Return to the Summary Table.

ADC PPB3 Trip High Register

Table 18-65. ADCPPB3TRIPHI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R	0h	Reserved
18-0	LIMITHI	R/W	0h	ADC Post Processing Block 3 Trip High Limit. This value sets the digital comparator trip high limit. When comparing to an ADCPPBxRESULT register, the upper bits will be ignored: - TRIPHI[18:13] will be ignored Reset type: SYSRSn

18.3.57 ADCPPB3TRIPLO Register (Offset = 1154h) [Reset = 00000000h]

ADCPPB3TRIPLO is shown in Table 18-66.

Return to the Summary Table.

ADC PPB3 Trip Low/Trigger Time Stamp Register

Table 18-66. ADCPPB3TRIPLO Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R	0h	Reserved
18-0	LIMITLO	R/W	0h	ADC Post Processing Block 3 Trip High Low Limit. This value sets the digital comparator trip low limit if ADCPPB3TRIPLO.LIMITLO2EN = 1. When comparing to an ADCPPBxRESULT register, the upper bits will be ignored: - TRIPLO[18:13] will be ignored in 12 bit mode Reset type: SYSRSn

18.3.58 ADCPPB4CONFIG Register (Offset = 1160h) [Reset = 00000000h]

ADCPPB4CONFIG is shown in Table 18-67.

Return to the Summary Table.

ADC PPB4 Config Register

Table 18-67. ADCPPB4CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7	RESERVED	R	0h	
6	RESERVED	R	0h	Reserved
5	CBCEN	R/W	0h	ADC Post Processing Block Cycle By Cycle Enable. When set, this bit enables the post conversion hardware processing circuit to automatically clear the ADCEVTSTAT on a conversion if the event condition is no longer present. Reset type: SYSRSn
4-0	RESERVED	R	0h	Reserved

18.3.59 ADCPPB4TRIPHI Register (Offset = 1170h) [Reset = 00000000h]

ADCPPB4TRIPHI is shown in Table 18-68.

Return to the Summary Table.

ADC PPB4 Trip High Register

Table 18-68. ADCPPB4TRIPHI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R	0h	Reserved
18-0	LIMITHI	R/W	0h	ADC Post Processing Block 4 Trip High Limit. This value sets the digital comparator trip high limit. When comparing to an ADCPPBxRESULT register, the upper bits will be ignored: - TRIPHI[18:13] will be ignored Reset type: SYSRSn

18.3.60 ADCPPB4TRIPLO Register (Offset = 1174h) [Reset = 00000000h]

ADCPPB4TRIPLO is shown in Table 18-69.

Return to the Summary Table.

ADC PPB4 Trip Low/Trigger Time Stamp Register

Table 18-69. ADCPPB4TRIPLO Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R	0h	Reserved
18-0	LIMITLO	R/W	0h	ADC Post Processing Block 4 Trip High Low Limit. This value sets the digital comparator trip low limit if ADCPPB4TRIPLO.LIMITLO2EN = 1. When comparing to an ADCPPBxRESULT register, the upper bits will be ignored: - TRIPLO[18:13] will be ignored in 12 bit mode Reset type: SYSRSn

18.3.61 ADCINTCYCLE Register (Offset = 1180h) [Reset = 00000000h]

ADCINTCYCLE is shown in Table 18-70.

Return to the Summary Table.

ADC Early Interrupt Generation Cycle

Table 18-70. ADCINTCYCLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	Reserved
5-0	DELAY	R/W	0h	ADC Early Interrupt Generation Cycle Delay: Defines the delay from the fall edge of ADCSOC in terms of system clock cycles, for the interrupt to be generated. Reset type: SYSRSn

18.3.62 ADCREV2 Register (Offset = 119Ch) [Reset = 00000006h]

ADCREV2 is shown in Table 18-71.

Return to the Summary Table.

ADC Wrapper Revision Register

Table 18-71. ADCREV2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	WRAPPERREV	R	0h	ADC Revision. To allow documentation of differences between revisions. First version is labeled as 00h. Reset type: SYSRSn
7-0	WRAPPERTYPE	R	6h	ADC Wrapper Type. Always set to 6 for this ADC. Reset type: SYSRSn

18.3.63 ADCPPB1LIMIT Register (Offset = 1200h) [Reset = 00000000h]

ADCPPB1LIMIT is shown in Table 18-72.

Return to the Summary Table.

ADC PPB1Conversion Count Limit Register

Table 18-72. ADCPPB1LIMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	LIMIT	R/W	0h	Post Processing Block 1 Oversampling Limit. Defines the number of conversions to accumulate before PSUM is automatically loaded into SUM. 0 = No - accumulation 1 = 2 conversions are accumulated 2 = 4 conversions are accumulated 3 = 8 conversions are accumulated 4 = 16 conversions are accumulated 5 = 32 conversions are accumulated 6 = 64 conversions are accumulated Reset type: SYSRSn

18.3.64 ADCPPB1PCOUNT Register (Offset = 1204h) [Reset = 00000000h]

ADCPPB1PCOUNT is shown in Table 18-73.

Return to the Summary Table.

ADC PPB1 Partial Conversion Count Register

Table 18-73. ADCPPB1PCOUNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	Reserved
5-0	PCOUNT	R	0h	Post Processing Block 1 Oversampling Partial Count. Each time a new result propagates through the PPB signal chain and accumulates into ADCPPB1PSUM this register is incremented by 1. This register is reset when either a count-match event occurs (PCOUNT = LIMIT) or PPB1 receives a sync. event. This result is available 1 SYSCLK cycle after the associated ADCPPB1RESULT is available. This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB1RESULT timing information). Reset type: SYSRSn

18.3.65 ADCPPB1CONFIG2 Register (Offset = 1208h) [Reset = 00000000h]

ADCPPB1CONFIG2 is shown in Table 18-74.

Return to the Summary Table.

ADC PPB1 Sum Shift Register

Table 18-74. ADCPPB1CONFIG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	COMPSEL	R/W	0h	Post Processing Block 1 Compare Source Select. This field determines whether ADCPPB1RESULT or ADCPPB1SUM is used for the threshold compare. 0 = ADCPPB1RESULT is used for compare logic 1 = ADCPPB1SUM is used for compare logic Reset type: SYSRSn
14-13	RESERVED	R	0h	Reserved
12	RESERVED	R	0h	
11	SWSYNC	R-0/W1S	0h	PPB 1 software force sync. On a sync. event, all partial registers transfer to the final registers and are then reset. Note: In the case where the software force occurs at the same time that a new sample is added to the PSUM and the PSUM is being used for a high or low limit compare, then the comparison will not occur. Reset type: SYSRSn
10-7	RESERVED	R	0h	Reserved
6-4	SYNCINSEL	R/W	0h	PPB 1 sync. input select. On a sync. event, all partial registers transfer to the final registers and are then reset. Refer to SOC spec for details Reset type: SYSRSn
3	RESERVED	R	0h	Reserved
2-0	SHIFT	R/W	0h	Post Processing Block 1 right shift. Defines the number of bits to right shift PSUM before loading into the final SUM. 0 : no right shift 1 : SUM = PSUM >> 1 2 : SUM = PSUM >> 2 ... 7 : SUM = PSUM >> 7 Reset type: SYSRSn

18.3.66 ADCPPB1PSUM Register (Offset = 120Ch) [Reset = 00000000h]

ADCPPB1PSUM is shown in Table 18-75.

Return to the Summary Table.

ADC PPB1 Partial Sum Register

Table 18-75. ADCPPB1PSUM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 15. Reset type: SYSRSn
15-0	PSUM	R	0h	Post Processing Block 1 Oversampling Partial Sum. Each time a new result propagates through the PPB signal chain and latches into ADCPPB1RESULT the result is subsequently accumulated into this register. This register is reset when either a count-match event occurs (PCOUNT = LIMIT) or PPB1 receives a sync. event. This result is available 1 SYSCLK cycle after the associated ADCPPB1RESULT is available. This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC. In the case of multiple PPBs associated with the same SOC, the lowest numbered PPB's result will be available 2 SYSCLK cycles after the associated ADCRESULT. Subsequent results (in order from lowest numbered PPB to highest) will each become available every 2-3 SYSCLK cycles. Reset type: SYSRSn

18.3.67 ADCPPB2LIMIT Register (Offset = 1240h) [Reset = 00000000h]

ADCPPB2LIMIT is shown in Table 18-76.

Return to the Summary Table.

ADC PPB2Conversion Count Limit Register

Table 18-76. ADCPPB2LIMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	LIMIT	R/W	0h	Post Processing Block 2 Oversampling Limit. Defines the number of conversions to accumulate before PSUM is automatically loaded into SUM. 0 = No - accumulation 1 = 2 conversions are accumulated 2 = 4 conversions are accumulated 3 = 8 conversions are accumulated 4 = 16 conversions are accumulated 5 = 32 conversions are accumulated 6 = 64 conversions are accumulated Reset type: SYSRSn

18.3.68 ADCPPB2PCOUNT Register (Offset = 1244h) [Reset = 00000000h]

ADCPPB2PCOUNT is shown in Table 18-77.

Return to the Summary Table.

ADC PPB2 Partial Conversion Count Register

Table 18-77. ADCPPB2PCOUNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	Reserved
5-0	PCOUNT	R	0h	Post Processing Block 2 Oversampling Partial Count. Each time a new result propagates through the PPB signal chain and accumulates into ADCPPB2PSUM this register is incremented by 1. This register is reset when either a count-match event occurs (PCOUNT = LIMIT) or PPB2 receives a sync. event. This result is available 1 SYSCLK cycle after the associated ADCPPB2RESULT is available. This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB2RESULT timing information). Reset type: SYSRSn

18.3.69 ADCPPB2CONFIG2 Register (Offset = 1248h) [Reset = 00000000h]

ADCPPB2CONFIG2 is shown in Table 18-78.

Return to the Summary Table.

ADC PPB2 Sum Shift Register

Table 18-78. ADCPPB2CONFIG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	COMPSEL	R/W	0h	Post Processing Block 2 Compare Source Select. This field determines whether ADCPPB2RESULT or ADCPPB2SUM is used for the threshold compare. 0 = ADCPPB2RESULT is used for compare logic 1 = ADCPPB2SUM is used for compare logic Reset type: SYSRSn
14-13	RESERVED	R	0h	Reserved
12	RESERVED	R	0h	
11	SWSYNC	R-0/W1S	0h	PPB 2 software force sync. On a sync. event, all partial registers transfer to the final registers and are then reset. Note: In the case where the software force occurs at the same time that a new sample is added to the PSUM and the PSUM is being used for a high or low limit compare, then the comparison will not occur. Reset type: SYSRSn
10-7	RESERVED	R	0h	Reserved
6-4	SYNCINSEL	R/W	0h	PPB 2 sync. input select. On a sync. event, all partial registers transfer to the final registers and are then reset. Refer to SOC spec for details Reset type: SYSRSn
3	RESERVED	R	0h	Reserved
2-0	SHIFT	R/W	0h	Post Processing Block 2 right shift. Defines the number of bits to right shift PSUM before loading into the final SUM. 0 : no right shift 1 : SUM = PSUM >> 1 2 : SUM = PSUM >> 2 ... 7 : SUM = PSUM >> 7 Reset type: SYSRSn

18.3.70 ADCPPB2PSUM Register (Offset = 124Ch) [Reset = 00000000h]

ADCPPB2PSUM is shown in Table 18-79.

Return to the Summary Table.

ADC PPB2 Partial Sum Register

Table 18-79. ADCPPB2PSUM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 15. Reset type: SYSRSn
15-0	PSUM	R	0h	Post Processing Block 2 Oversampling Partial Sum. Each time a new result propagates through the PPB signal chain and latches into ADCPPB2RESULT the result is subsequently accumulated into this register. This register is reset when either a count-match event occurs (PCOUNT = LIMIT) or PPB2 receives a sync. event. This result is available 1 SYSCLK cycle after the associated ADCPPB2RESULT is available. This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC. In the case of multiple PPBs associated with the same SOC, the lowest numbered PPB's result will be available 2 SYSCLK cycles after the associated ADCRESULT. Subsequent results (in order from lowest numbered PPB to highest) will each become available every 2-3 SYSCLK cycles. Reset type: SYSRSn

18.3.71 ADCPPB3LIMIT Register (Offset = 1280h) [Reset = 00000000h]

ADCPPB3LIMIT is shown in Table 18-80.

Return to the Summary Table.

ADC PPB3Conversion Count Limit Register

Table 18-80. ADCPPB3LIMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	LIMIT	R/W	0h	Post Processing Block 3 Oversampling Limit. Defines the number of conversions to accumulate before PSUM is automatically loaded into SUM. 0 = No - accumulation 1 = 2 conversions are accumulated 2 = 4 conversions are accumulated 3 = 8 conversions are accumulated 4 = 16 conversions are accumulated 5 = 32 conversions are accumulated 6 = 64 conversions are accumulated Reset type: SYSRSn

18.3.72 ADCPPB3PCOUNT Register (Offset = 1284h) [Reset = 00000000h]

ADCPPB3PCOUNT is shown in Table 18-81.

Return to the Summary Table.

ADC PPB3 Partial Conversion Count Register

Table 18-81. ADCPPB3PCOUNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	Reserved
5-0	PCOUNT	R	0h	Post Processing Block 3 Oversampling Partial Count. Each time a new result propagates through the PPB signal chain and accumulates into ADCPPB3PSUM this register is incremented by 1. This register is reset when either a count-match event occurs (PCOUNT = LIMIT) or PPB3 receives a sync. event. This result is available 1 SYSCLK cycle after the associated ADCPPB3RESULT is available. This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB3RESULT timing information). Reset type: SYSRSn

18.3.73 ADCPPB3CONFIG2 Register (Offset = 1288h) [Reset = 00000000h]

ADCPPB3CONFIG2 is shown in Table 18-82.

Return to the Summary Table.

ADC PPB3 Sum Shift Register

Table 18-82. ADCPPB3CONFIG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	COMPSEL	R/W	0h	Post Processing Block 3 Compare Source Select. This field determines whether ADCPPB3RESULT or ADCPPB3SUM is used for the threshold compare. 0 = ADCPPB3RESULT is used for compare logic 1 = ADCPPB3SUM is used for compare logic Reset type: SYSRSn
14-13	RESERVED	R	0h	Reserved
12	RESERVED	R	0h	
11	SWSYNC	R-0/W1S	0h	PPB 3 software force sync. On a sync. event, all partial registers transfer to the final registers and are then reset. Note: In the case where the software force occurs at the same time that a new sample is added to the PSUM and the PSUM is being used for a high or low limit compare, then the comparison will not occur. Reset type: SYSRSn
10-7	RESERVED	R	0h	Reserved
6-4	SYNCINSEL	R/W	0h	PPB 3 sync. input select. On a sync. event, all partial registers transfer to the final registers and are then reset. Refer to SOC spec for details Reset type: SYSRSn
3	RESERVED	R	0h	Reserved
2-0	SHIFT	R/W	0h	Post Processing Block 3 right shift. Defines the number of bits to right shift PSUM before loading into the final SUM. 0 : no right shift 1 : SUM = PSUM >> 1 2 : SUM = PSUM >> 2 ... 7 : SUM = PSUM >> 7 Reset type: SYSRSn

18.3.74 ADCPPB3PSUM Register (Offset = 128Ch) [Reset = 00000000h]

ADCPPB3PSUM is shown in Table 18-83.

Return to the Summary Table.

ADC PPB3 Partial Sum Register

Table 18-83. ADCPPB3PSUM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 15. Reset type: SYSRSn
15-0	PSUM	R	0h	Post Processing Block 3 Oversampling Partial Sum. Each time a new result propagates through the PPB signal chain and latches into ADCPPB3RESULT the result is subsequently accumulated into this register. This register is reset when either a count-match event occurs (PCOUNT = LIMIT) or PPB3 receives a sync. event. This result is available 1 SYSCLK cycle after the associated ADCPPB3RESULT is available. This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC. In the case of multiple PPBs associated with the same SOC, the lowest numbered PPB's result will be available 2 SYSCLK cycles after the associated ADCRESULT. Subsequent results (in order from lowest numbered PPB to highest) will each become available every 2-3 SYSCLK cycles. Reset type: SYSRSn

18.3.75 ADCPPB4LIMIT Register (Offset = 12C0h) [Reset = 00000000h]

ADCPPB4LIMIT is shown in Table 18-84.

Return to the Summary Table.

ADC PPB4Conversion Count Limit Register

Table 18-84. ADCPPB4LIMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	LIMIT	R/W	0h	Post Processing Block 4 Oversampling Limit. Defines the number of conversions to accumulate before PSUM is automatically loaded into SUM. 0 = No - accumulation 1 = 2 conversions are accumulated 2 = 4 conversions are accumulated 3 = 8 conversions are accumulated 4 = 16 conversions are accumulated 5 = 32 conversions are accumulated 6 = 64 conversions are accumulated Reset type: SYSRSn

18.3.76 ADCPPB4PCOUNT Register (Offset = 12C4h) [Reset = 00000000h]

ADCPPB4PCOUNT is shown in Table 18-85.

Return to the Summary Table.

ADC PPB4 Partial Conversion Count Register

Table 18-85. ADCPPB4PCOUNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	Reserved
5-0	PCOUNT	R	0h	Post Processing Block 4 Oversampling Partial Count. Each time a new result propagates through the PPB signal chain and accumulates into ADCPPB4PSUM this register is incremented by 1. This register is reset when either a count-match event occurs (PCOUNT = LIMIT) or PPB4 receives a sync. event. This result is available 1 SYSCLK cycle after the associated ADCPPB4RESULT is available. This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB4RESULT timing information). Reset type: SYSRSn

18.3.77 ADCPPB4CONFIG2 Register (Offset = 12C8h) [Reset = 00000000h]

ADCPPB4CONFIG2 is shown in Table 18-86.

Return to the Summary Table.

ADC PPB4 Sum Shift Register

Table 18-86. ADCPPB4CONFIG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	COMPSEL	R/W	0h	Post Processing Block 4 Compare Source Select. This field determines whether ADCPPB4RESULT or ADCPPB4SUM is used for the threshold compare. 0 = ADCPPB4RESULT is used for compare logic 1 = ADCPPB4SUM is used for compare logic Reset type: SYSRSn
14-13	RESERVED	R	0h	Reserved
12	RESERVED	R	0h	
11	SWSYNC	R-0/W1S	0h	PPB 4 software force sync. On a sync. event, all partial registers transfer to the final registers and are then reset. Note: In the case where the software force occurs at the same time that a new sample is added to the PSUM and the PSUM is being used for a high or low limit compare, then the comparison will not occur. Reset type: SYSRSn
10-7	RESERVED	R	0h	Reserved
6-4	SYNCINSEL	R/W	0h	PPB 4 sync. input select. On a sync. event, all partial registers transfer to the final registers and are then reset. Refer to SOC spec for details Reset type: SYSRSn
3	RESERVED	R	0h	Reserved
2-0	SHIFT	R/W	0h	Post Processing Block 4 right shift. Defines the number of bits to right shift PSUM before loading into the final SUM. 0 : no right shift 1 : SUM = PSUM >> 1 2 : SUM = PSUM >> 2 ... 7 : SUM = PSUM >> 7 Reset type: SYSRSn

18.3.78 ADCPPB4PSUM Register (Offset = 12CCh) [Reset = 00000000h]

ADCPPB4PSUM is shown in Table 18-87.

Return to the Summary Table.

ADC PPB4 Partial Sum Register

Table 18-87. ADCPPB4PSUM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 15. Reset type: SYSRSn
15-0	PSUM	R	0h	Post Processing Block 4 Oversampling Partial Sum. Each time a new result propagates through the PPB signal chain and latches into ADCPPB4RESULT the result is subsequently accumulated into this register. This register is reset when either a count-match event occurs (PCOUNT = LIMIT) or PPB4 receives a sync. event. This result is available 1 SYSCLK cycle after the associated ADCPPB4RESULT is available. This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC. In the case of multiple PPBs associated with the same SOC, the lowest numbered PPB's result will be available 2 SYSCLK cycles after the associated ADCRESULT. Subsequent results (in order from lowest numbered PPB to highest) will each become available every 2-3 SYSCLK cycles. Reset type: SYSRSn

18.3.79 ADCSEQCTL Register (Offset = 1320h) [Reset = 00000000h]

ADCSEQCTL is shown in Table 18-88.

Return to the Summary Table.

ADC Sequencer common control Register

Table 18-88. ADCSEQCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R	0h	Reserved
17-16	SEQPREEMPT	R/W	0h	SoC Sequence PREEMPT 0X: Pre-empt disabled 10: Pre-empt enabled and will not restart aborted Sequence 11: Pre-empt enabled and will restart aborted Sequence Reset type: SYSRSn
15-4	RESERVED	R	0h	Reserved
3-0	SEQEND	R/W	0h	END SOC of last enabled sequence Reset type: SYSRSn

18.3.80 ADCSEQ1CONFIG Register (Offset = 1324h) [Reset = 00000200h]

ADCSEQ1CONFIG is shown in Table 18-89.

Return to the Summary Table.

ADC Sequencer 1 Config register

Table 18-89. ADCSEQ1CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31	SEQENABLE	R/W	0h	SEQ1Enable Indicates whether the Sequence1 is enabled or not Reset type: SYSRSn
30	SEQSWFRC	R-0/W1S	0h	Write 1 to force a trigger to Sequencer 1 regardless of the value of Hardware TRIGGER. Always reads 0. Reset type: SYSRSn
29-23	RESERVED	R	0h	Reserved
22-20	TRIGSEL	R/W	0h	SOC1 Trigger Source Select. Along with the SOC1 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC1 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 0h ADCTRIG0 - Software only 1h - 7h Hardware triggers Reset type: SYSRSn
19	RESERVED	R	0h	Reserved
18-15	SEQSTART	R/W	0h	Beginning SOC of SEQ1 Reset type: SYSRSn
14-11	RESERVED	R	0h	Reserved
10	SAMPCAPRESETSEL	R/W	0h	SOC1 Sample Cap Reset Select : Resets sample cap after conversion to either VREFHI/2 or VREFLO 0 - The sample cap is reset to VREFLO after each conversion 1 - The sample cap is reset to VREFHI/2 after each conversion Reset type: SYSRSn
9	SAMPCAPRESETDISABLE	R/W	1h	SOC1 Sample Cap Reset enable : Resets sample cap after conversion. 0 - The sample cap is reset after each conversion 1 - The sample cap is not reset after each conversion Reset type: SYSRSn
8	RESERVED	R	0h	Reserved
7-0	ACQPS	R/W	0h	SOC1 Acquisition Prescale. Controls the sample and hold window for each SOC in this Seqeunce. The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The device datasheet will also specify a minimum sample and hold window duration. S+H window values is defined based on a combonation of ACQPS[7:6] and ACQPS[5:0] values If ACQPS[7:6] value is "00" S+H window = ACQPS[5:0] + 1 sysclk cycles "01" S+H window = 64 + ((ACQPS[5:0] +1) * 2) sysclk cycles "10" S+H window = 192 + ((ACQPS[5:0] +1) * 4) sysclk cycles "11" S+H window = 448 + ((ACQPS[5:0] +1) * 16) sysclk cycles Reset type: SYSRSn

18.3.81 ADCSEQ2CONFIG Register (Offset = 1328h) [Reset = 00000200h]

ADCSEQ2CONFIG is shown in Table 18-90.

Return to the Summary Table.

ADC Sequencer 2 Config register

Table 18-90. ADCSEQ2CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31	SEQENABLE	R/W	0h	SEQ2Enable Indicates whether the Sequence2 is enabled or not Reset type: SYSRSn
30	SEQSWFRC	R-0/W1S	0h	Write 1 to force a trigger to Sequencer 2 regardless of the value of Hardware TRIGGER. Always reads 0. Reset type: SYSRSn
29-23	RESERVED	R	0h	Reserved
22-20	TRIGSEL	R/W	0h	SOC2 Trigger Source Select. Along with the SOC2 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC2 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 0h ADCTRIG0 - Software only 1h - 7h Hardware triggers Reset type: SYSRSn
19	RESERVED	R	0h	Reserved
18-15	SEQSTART	R/W	0h	Beginning SOC of SEQ2 Reset type: SYSRSn
14-11	RESERVED	R	0h	Reserved
10	SAMPCAPRESETSEL	R/W	0h	SOC2 Sample Cap Reset Select : Resets sample cap after conversion to either VREFHI/2 or VREFLO 0 - The sample cap is reset to VREFLO after each conversion 1 - The sample cap is reset to VREFHI/2 after each conversion Reset type: SYSRSn
9	SAMPCAPRESETDISABLE	R/W	1h	SOC2 Sample Cap Reset enable : Resets sample cap after conversion. 0 - The sample cap is reset after each conversion 1 - The sample cap is not reset after each conversion Reset type: SYSRSn
8	RESERVED	R	0h	Reserved
7-0	ACQPS	R/W	0h	SOC2 Acquisition Prescale. Controls the sample and hold window for each SOC in this Seqeunce. The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The device datasheet will also specify a minimum sample and hold window duration. S+H window values is defined based on a combonation of ACQPS[7:6] and ACQPS[5:0] values If ACQPS[7:6] value is "00" S+H window = ACQPS[5:0] + 1 sysclk cycles "01" S+H window = 64 + ((ACQPS[5:0] +1) * 2) sysclk cycles "10" S+H window = 192 + ((ACQPS[5:0] +1) * 4) sysclk cycles "11" S+H window = 448 + ((ACQPS[5:0] +1) * 16) sysclk cycles Reset type: SYSRSn

18.3.82 ADCSEQ3CONFIG Register (Offset = 132Ch) [Reset = 00000200h]

ADCSEQ3CONFIG is shown in Table 18-91.

Return to the Summary Table.

ADC Sequencer 3 Config register

Table 18-91. ADCSEQ3CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31	SEQENABLE	R/W	0h	SEQ3Enable Indicates whether the Sequence3 is enabled or not Reset type: SYSRSn
30	SEQSWFRC	R-0/W1S	0h	Write 1 to force a trigger to Sequencer 3 regardless of the value of Hardware TRIGGER. Always reads 0. Reset type: SYSRSn
29-23	RESERVED	R	0h	Reserved
22-20	TRIGSEL	R/W	0h	SOC3 Trigger Source Select. Along with the SOC3 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC3 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 0h ADCTRIG0 - Software only 1h - 7h Hardware triggers Reset type: SYSRSn
19	RESERVED	R	0h	Reserved
18-15	SEQSTART	R/W	0h	Beginning SOC of SEQ3 Reset type: SYSRSn
14-11	RESERVED	R	0h	Reserved
10	SAMPCAPRESETSEL	R/W	0h	SOC3 Sample Cap Reset Select : Resets sample cap after conversion to either VREFHI/2 or VREFLO 0 - The sample cap is reset to VREFLO after each conversion 1 - The sample cap is reset to VREFHI/2 after each conversion Reset type: SYSRSn
9	SAMPCAPRESETDISABLE	R/W	1h	SOC3 Sample Cap Reset enable : Resets sample cap after conversion. 0 - The sample cap is reset after each conversion 1 - The sample cap is not reset after each conversion Reset type: SYSRSn
8	RESERVED	R	0h	Reserved
7-0	ACQPS	R/W	0h	SOC3 Acquisition Prescale. Controls the sample and hold window for each SOC in this Seqeunce. The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The device datasheet will also specify a minimum sample and hold window duration. S+H window values is defined based on a combonation of ACQPS[7:6] and ACQPS[5:0] values If ACQPS[7:6] value is "00" S+H window = ACQPS[5:0] + 1 sysclk cycles "01" S+H window = 64 + ((ACQPS[5:0] +1) * 2) sysclk cycles "10" S+H window = 192 + ((ACQPS[5:0] +1) * 4) sysclk cycles "11" S+H window = 448 + ((ACQPS[5:0] +1) * 16) sysclk cycles Reset type: SYSRSn

18.3.83 ADCSEQ4CONFIG Register (Offset = 1330h) [Reset = 00000200h]

ADCSEQ4CONFIG is shown in Table 18-92.

Return to the Summary Table.

ADC Sequencer 4 Config register

Table 18-92. ADCSEQ4CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31	SEQENABLE	R/W	0h	SEQ4Enable Indicates whether the Sequence4 is enabled or not Reset type: SYSRSn
30	SEQSWFRC	R-0/W1S	0h	Write 1 to force a trigger to Sequencer 4 regardless of the value of Hardware TRIGGER. Always reads 0. Reset type: SYSRSn
29-23	RESERVED	R	0h	Reserved
22-20	TRIGSEL	R/W	0h	SOC4 Trigger Source Select. Along with the SOC4 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC4 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 0h ADCTRIG0 - Software only 1h - 7h Hardware triggers Reset type: SYSRSn
19	RESERVED	R	0h	Reserved
18-15	SEQSTART	R/W	0h	Beginning SOC of SEQ4 Reset type: SYSRSn
14-11	RESERVED	R	0h	Reserved
10	SAMPCAPRESETSEL	R/W	0h	SOC4 Sample Cap Reset Select : Resets sample cap after conversion to either VREFHI/2 or VREFLO 0 - The sample cap is reset to VREFLO after each conversion 1 - The sample cap is reset to VREFHI/2 after each conversion Reset type: SYSRSn
9	SAMPCAPRESETDISABLE	R/W	1h	SOC4 Sample Cap Reset enable : Resets sample cap after conversion. 0 - The sample cap is reset after each conversion 1 - The sample cap is not reset after each conversion Reset type: SYSRSn
8	RESERVED	R	0h	Reserved
7-0	ACQPS	R/W	0h	SOC4 Acquisition Prescale. Controls the sample and hold window for each SOC in this Seqeunce. The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The device datasheet will also specify a minimum sample and hold window duration. S+H window values is defined based on a combonation of ACQPS[7:6] and ACQPS[5:0] values If ACQPS[7:6] value is "00" S+H window = ACQPS[5:0] + 1 sysclk cycles "01" S+H window = 64 + ((ACQPS[5:0] +1) * 2) sysclk cycles "10" S+H window = 192 + ((ACQPS[5:0] +1) * 4) sysclk cycles "11" S+H window = 448 + ((ACQPS[5:0] +1) * 16) sysclk cycles Reset type: SYSRSn

18.4 ADC_LITE_RESULT_REGS Registers

Table 18-93 lists the memory-mapped registers for the ADC_LITE_RESULT_REGS registers. All register offset addresses not listed in Table 18-93 should be considered as reserved locations and the register contents should not be modified.

Table 18-93. ADC_LITE_RESULT_REGS Registers

Offset	Acronym	Register Name	Section
0h	ADCRESULT0	ADC Result 0 Register	Section 18.4.1
2h	ADCRESULT1	ADC Result 1 Register	Section 18.4.2
4h	ADCRESULT2	ADC Result 2 Register	Section 18.4.3
6h	ADCRESULT3	ADC Result 3 Register	Section 18.4.4
8h	ADCRESULT4	ADC Result 4 Register	Section 18.4.5
Ah	ADCRESULT5	ADC Result 5 Register	Section 18.4.6
Ch	ADCRESULT6	ADC Result 6 Register	Section 18.4.7
Eh	ADCRESULT7	ADC Result 7 Register	Section 18.4.8
10h	ADCRESULT8	ADC Result 8 Register	Section 18.4.9
12h	ADCRESULT9	ADC Result 9 Register	Section 18.4.10
14h	ADCRESULT10	ADC Result 10 Register	Section 18.4.11
16h	ADCRESULT11	ADC Result 11 Register	Section 18.4.12
18h	ADCRESULT12	ADC Result 12 Register	Section 18.4.13
1Ah	ADCRESULT13	ADC Result 13 Register	Section 18.4.14
1Ch	ADCRESULT14	ADC Result 14 Register	Section 18.4.15
1Eh	ADCRESULT15	ADC Result 15 Register	Section 18.4.16
40h	ADCPPB1RESULT	ADC Post Processing Block 1 Result Register	Section 18.4.17
44h	ADCPPB2RESULT	ADC Post Processing Block 2 Result Register	Section 18.4.18
48h	ADCPPB3RESULT	ADC Post Processing Block 3 Result Register	Section 18.4.19
4Ch	ADCPPB4RESULT	ADC Post Processing Block 4 Result Register	Section 18.4.20
50h	ADCPPB1SUM	ADC PPB 1 Final Sum Result Register	Section 18.4.21
54h	ADCPPB1COUNT	ADC PPB1 Final Conversion Count Register	Section 18.4.22
58h	ADCPPB2SUM	ADC PPB 2 Final Sum Result Register	Section 18.4.23
5Ch	ADCPPB2COUNT	ADC PPB2 Final Conversion Count Register	Section 18.4.24
60h	ADCPPB3SUM	ADC PPB 3 Final Sum Result Register	Section 18.4.25
64h	ADCPPB3COUNT	ADC PPB3 Final Conversion Count Register	Section 18.4.26
68h	ADCPPB4SUM	ADC PPB 4 Final Sum Result Register	Section 18.4.27
6Ch	ADCPPB4COUNT	ADC PPB4 Final Conversion Count Register	Section 18.4.28
C0h	ADCSEQ1FIFORESULT	ADC Sequence 1 FIFO Result Register	Section 18.4.29
C4h	ADCSEQ2FIFORESULT	ADC Sequence 2 FIFO Result Register	Section 18.4.30
C8h	ADCSEQ3FIFORESULT	ADC Sequence 3 FIFO Result Register	Section 18.4.31
CCh	ADCSEQ4FIFORESULT	ADC Sequence 4 FIFO Result Register	Section 18.4.32
D0h	ADCSEQ1FIFOSTATUS	ADC Sequence 1 FIFO Status	Section 18.4.33
D4h	ADCSEQ2FIFOSTATUS	ADC Sequence 2 FIFO Status	Section 18.4.34
D8h	ADCSEQ3FIFOSTATUS	ADC Sequence 3 FIFO Status	Section 18.4.35
DCh	ADCSEQ4FIFOSTATUS	ADC Sequence 4 FIFO Status	Section 18.4.36

Complex bit access types are encoded to fit into small table cells. Table 18-94 shows the codes that are used for access types in this section.

Table 18-94. ADC_LITE_RESULT_REGS Access Type Codes

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

18.4.1 ADCRESULT0 Register (Offset = 0h) [Reset = 0000h]

ADCRESULT0 is shown in Table 18-95.

Return to the Summary Table.

ADC Result 0 Register

Table 18-95. ADCRESULT0 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 0 12-bit ADC result. After the ADC completes a conversion of SOC0, the digital result is placed in this bit field. If oversampling is enabled, Final SUM (After any Shift if enabled) will be stored in this register after completing the specified number of samples. Reset type: SYSRSn

18.4.2 ADCRESULT1 Register (Offset = 2h) [Reset = 0000h]

ADCRESULT1 is shown in Table 18-96.

Return to the Summary Table.

ADC Result 1 Register

Table 18-96. ADCRESULT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 1 12-bit ADC result. After the ADC completes a conversion of SOC1, the digital result is placed in this bit field. If oversampling is enabled, Final SUM (After any Shift if enabled) will be stored in this register after completing the specified number of samples. Reset type: SYSRSn

18.4.3 ADCRESULT2 Register (Offset = 4h) [Reset = 0000h]

ADCRESULT2 is shown in Table 18-97.

Return to the Summary Table.

ADC Result 2 Register

Table 18-97. ADCRESULT2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 2 12-bit ADC result. After the ADC completes a conversion of SOC2, the digital result is placed in this bit field. If oversampling is enabled, Final SUM (After any Shift if enabled) will be stored in this register after completing the specified number of samples. Reset type: SYSRSn

18.4.4 ADCRESULT3 Register (Offset = 6h) [Reset = 0000h]

ADCRESULT3 is shown in Table 18-98.

Return to the Summary Table.

ADC Result 3 Register

Table 18-98. ADCRESULT3 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 3 12-bit ADC result. After the ADC completes a conversion of SOC3, the digital result is placed in this bit field. If oversampling is enabled, Final SUM (After any Shift if enabled) will be stored in this register after completing the specified number of samples. Reset type: SYSRSn

18.4.5 ADCRESULT4 Register (Offset = 8h) [Reset = 0000h]

ADCRESULT4 is shown in Table 18-99.

Return to the Summary Table.

ADC Result 4 Register

Table 18-99. ADCRESULT4 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 4 12-bit ADC result. After the ADC completes a conversion of SOC4, the digital result is placed in this bit field. If oversampling is enabled, Final SUM (After any Shift if enabled) will be stored in this register after completing the specified number of samples. Reset type: SYSRSn

18.4.6 ADCRESULT5 Register (Offset = Ah) [Reset = 0000h]

ADCRESULT5 is shown in Table 18-100.

Return to the Summary Table.

ADC Result 5 Register

Table 18-100. ADCRESULT5 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 5 12-bit ADC result. After the ADC completes a conversion of SOC5, the digital result is placed in this bit field. If oversampling is enabled, Final SUM (After any Shift if enabled) will be stored in this register after completing the specified number of samples. Reset type: SYSRSn

18.4.7 ADCRESULT6 Register (Offset = Ch) [Reset = 0000h]

ADCRESULT6 is shown in Table 18-101.

Return to the Summary Table.

ADC Result 6 Register

Table 18-101. ADCRESULT6 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 6 12-bit ADC result. After the ADC completes a conversion of SOC6, the digital result is placed in this bit field. If oversampling is enabled, Final SUM (After any Shift if enabled) will be stored in this register after completing the specified number of samples. Reset type: SYSRSn

18.4.8 ADCRESULT7 Register (Offset = Eh) [Reset = 0000h]

ADCRESULT7 is shown in Table 18-102.

Return to the Summary Table.

ADC Result 7 Register

Table 18-102. ADCRESULT7 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 7 12-bit ADC result. After the ADC completes a conversion of SOC7, the digital result is placed in this bit field. If oversampling is enabled, Final SUM (After any Shift if enabled) will be stored in this register after completing the specified number of samples. Reset type: SYSRSn

18.4.9 ADCRESULT8 Register (Offset = 10h) [Reset = 0000h]

ADCRESULT8 is shown in Table 18-103.

Return to the Summary Table.

ADC Result 8 Register

Table 18-103. ADCRESULT8 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 8 12-bit ADC result. After the ADC completes a conversion of SOC8, the digital result is placed in this bit field. If oversampling is enabled, Final SUM (After any Shift if enabled) will be stored in this register after completing the specified number of samples. Reset type: SYSRSn

18.4.10 ADCRESULT9 Register (Offset = 12h) [Reset = 0000h]

ADCRESULT9 is shown in Table 18-104.

Return to the Summary Table.

ADC Result 9 Register

Table 18-104. ADCRESULT9 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 9 12-bit ADC result. After the ADC completes a conversion of SOC9, the digital result is placed in this bit field. If oversampling is enabled, Final SUM (After any Shift if enabled) will be stored in this register after completing the specified number of samples. Reset type: SYSRSn

18.4.11 ADCRESULT10 Register (Offset = 14h) [Reset = 0000h]

ADCRESULT10 is shown in Table 18-105.

Return to the Summary Table.

ADC Result 10 Register

Table 18-105. ADCRESULT10 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 10 12-bit ADC result. After the ADC completes a conversion of SOC10, the digital result is placed in this bit field. If oversampling is enabled, Final SUM (After any Shift if enabled) will be stored in this register after completing the specified number of samples. Reset type: SYSRSn

18.4.12 ADCRESULT11 Register (Offset = 16h) [Reset = 0000h]

ADCRESULT11 is shown in Table 18-106.

Return to the Summary Table.

ADC Result 11 Register

Table 18-106. ADCRESULT11 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 11 12-bit ADC result. After the ADC completes a conversion of SOC11, the digital result is placed in this bit field. If oversampling is enabled, Final SUM (After any Shift if enabled) will be stored in this register after completing the specified number of samples. Reset type: SYSRSn

18.4.13 ADCRESULT12 Register (Offset = 18h) [Reset = 0000h]

ADCRESULT12 is shown in Table 18-107.

Return to the Summary Table.

ADC Result 12 Register

Table 18-107. ADCRESULT12 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 12 12-bit ADC result. After the ADC completes a conversion of SOC12, the digital result is placed in this bit field. If oversampling is enabled, Final SUM (After any Shift if enabled) will be stored in this register after completing the specified number of samples. Reset type: SYSRSn

18.4.14 ADCRESULT13 Register (Offset = 1Ah) [Reset = 0000h]

ADCRESULT13 is shown in Table 18-108.

Return to the Summary Table.

ADC Result 13 Register

Table 18-108. ADCRESULT13 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 13 12-bit ADC result. After the ADC completes a conversion of SOC13, the digital result is placed in this bit field. If oversampling is enabled, Final SUM (After any Shift if enabled) will be stored in this register after completing the specified number of samples. Reset type: SYSRSn

18.4.15 ADCRESULT14 Register (Offset = 1Ch) [Reset = 0000h]

ADCRESULT14 is shown in Table 18-109.

Return to the Summary Table.

ADC Result 14 Register

Table 18-109. ADCRESULT14 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 14 12-bit ADC result. After the ADC completes a conversion of SOC14, the digital result is placed in this bit field. If oversampling is enabled, Final SUM (After any Shift if enabled) will be stored in this register after completing the specified number of samples. Reset type: SYSRSn

18.4.16 ADCRESULT15 Register (Offset = 1Eh) [Reset = 0000h]

ADCRESULT15 is shown in Table 18-110.

Return to the Summary Table.

ADC Result 15 Register

Table 18-110. ADCRESULT15 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 15 12-bit ADC result. After the ADC completes a conversion of SOC15, the digital result is placed in this bit field. If oversampling is enabled, Final SUM (After any Shift if enabled) will be stored in this register after completing the specified number of samples. Reset type: SYSRSn

18.4.17 ADCPPB1RESULT Register (Offset = 40h) [Reset = 00000000h]

ADCPPB1RESULT is shown in Table 18-111.

Return to the Summary Table.

ADC Post Processing Block 1 Result Register

Table 18-111. ADCPPB1RESULT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 12. NOTE: If the conversion associated with this Post Processing Block is a 12-bit conversion, the SIGN bits extend down to bit 12, and all reflect the same value as bit 12. Reset type: SYSRSn
12-0	PPBRESULT	R	0h	ADC Post Processing Block Result 1 The result of the offset/reference subtraction post conversion processing is stored in this register. This result is available 1 SYSCLK cycle after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC. In the case of multiple PPBs associated with the same SOC, the lowest numbered PPB's result will be availble 1 SYSCLK cycle after the associated ADCRESULT and subsequent results (in order from lowest numbered PPB to highest) will each become available every 2-3 SYSCLK cycles (refer to the TRM for more detailed timing information). If ADCINTFLG is polled to determine when to read the PPBRESULT, it may be necessary to add one or more NOP instructions to ensure that the updated post conversion processing result has posted to the register. NOTE: If the conversion associated with this Post Processing Block is a 12-bit conversion, the PPBRESULT bits are limited to bits 12:0. Reset type: SYSRSn

18.4.18 ADCPPB2RESULT Register (Offset = 44h) [Reset = 00000000h]

ADCPPB2RESULT is shown in Table 18-112.

Return to the Summary Table.

ADC Post Processing Block 2 Result Register

Table 18-112. ADCPPB2RESULT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 12. NOTE: If the conversion associated with this Post Processing Block is a 12-bit conversion, the SIGN bits extend down to bit 12, and all reflect the same value as bit 12. Reset type: SYSRSn
12-0	PPBRESULT	R	0h	ADC Post Processing Block Result 2 The result of the offset/reference subtraction post conversion processing is stored in this register. This result is available 1 SYSCLK cycle after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC. In the case of multiple PPBs associated with the same SOC, the lowest numbered PPB's result will be availble 1 SYSCLK cycle after the associated ADCRESULT and subsequent results (in order from lowest numbered PPB to highest) will each become available every 2-3 SYSCLK cycles (refer to the TRM for more detailed timing information). If ADCINTFLG is polled to determine when to read the PPBRESULT, it may be necessary to add one or more NOP instructions to ensure that the updated post conversion processing result has posted to the register. NOTE: If the conversion associated with this Post Processing Block is a 12-bit conversion, the PPBRESULT bits are limited to bits 12:0. Reset type: SYSRSn

18.4.19 ADCPPB3RESULT Register (Offset = 48h) [Reset = 00000000h]

ADCPPB3RESULT is shown in Table 18-113.

Return to the Summary Table.

ADC Post Processing Block 3 Result Register

Table 18-113. ADCPPB3RESULT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 12. NOTE: If the conversion associated with this Post Processing Block is a 12-bit conversion, the SIGN bits extend down to bit 12, and all reflect the same value as bit 12. Reset type: SYSRSn
12-0	PPBRESULT	R	0h	ADC Post Processing Block Result 3 The result of the offset/reference subtraction post conversion processing is stored in this register. This result is available 1 SYSCLK cycle after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC. In the case of multiple PPBs associated with the same SOC, the lowest numbered PPB's result will be availble 1 SYSCLK cycle after the associated ADCRESULT and subsequent results (in order from lowest numbered PPB to highest) will each become available every 2-3 SYSCLK cycles (refer to the TRM for more detailed timing information). If ADCINTFLG is polled to determine when to read the PPBRESULT, it may be necessary to add one or more NOP instructions to ensure that the updated post conversion processing result has posted to the register. NOTE: If the conversion associated with this Post Processing Block is a 12-bit conversion, the PPBRESULT bits are limited to bits 12:0. Reset type: SYSRSn

18.4.20 ADCPPB4RESULT Register (Offset = 4Ch) [Reset = 00000000h]

ADCPPB4RESULT is shown in Table 18-114.

Return to the Summary Table.

ADC Post Processing Block 4 Result Register

Table 18-114. ADCPPB4RESULT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 12. NOTE: If the conversion associated with this Post Processing Block is a 12-bit conversion, the SIGN bits extend down to bit 12, and all reflect the same value as bit 12. Reset type: SYSRSn
12-0	PPBRESULT	R	0h	ADC Post Processing Block Result 4 The result of the offset/reference subtraction post conversion processing is stored in this register. This result is available 1 SYSCLK cycle after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC. In the case of multiple PPBs associated with the same SOC, the lowest numbered PPB's result will be availble 1 SYSCLK cycle after the associated ADCRESULT and subsequent results (in order from lowest numbered PPB to highest) will each become available every 2-3 SYSCLK cycles (refer to the TRM for more detailed timing information). If ADCINTFLG is polled to determine when to read the PPBRESULT, it may be necessary to add one or more NOP instructions to ensure that the updated post conversion processing result has posted to the register. NOTE: If the conversion associated with this Post Processing Block is a 12-bit conversion, the PPBRESULT bits are limited to bits 12:0. Reset type: SYSRSn

18.4.21 ADCPPB1SUM Register (Offset = 50h) [Reset = 00000000h]

ADCPPB1SUM is shown in Table 18-115.

Return to the Summary Table.

ADC PPB 1 Final Sum Result Register

Table 18-115. ADCPPB1SUM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 15. Reset type: SYSRSn
15-0	SUM	R	0h	Post Processing Block 1 Oversampling Final Sum. When either a count-match event occurs (PCOUNT = LIMIT) or PPB1 receives a sync. event, the value of PSUM is loaded into this register. In the case of a count-match event, the sum loaded into this register includes the value from the most recent conversion. The value from PSUM will be right shifted by the amount specified in the SHIFT register before being loaded into the final SUM result register. This result is available 1 SYSCLK cycle after the associated ADCPPB1RESULT is available (only in case of a count-match event). This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB1RESULT timing information). Reset type: SYSRSn

18.4.22 ADCPPB1COUNT Register (Offset = 54h) [Reset = 0000h]

ADCPPB1COUNT is shown in Table 18-116.

Return to the Summary Table.

ADC PPB1 Final Conversion Count Register

Table 18-116. ADCPPB1COUNT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-7	RESERVED	R	0h	Reserved
6-0	COUNT	R	0h	Post Processing Block 1 Oversampling Final Count. When either a count-match event occurs (PCOUNT = LIMIT) or PPB1 receives a sync. event, the value of PCOUNT is loaded into this register. This result is available 1 SYSCLK cycle after the associated ADCPPB1RESULT is available (only in case of a count-match event). This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB1RESULT timing information). Reset type: SYSRSn

18.4.23 ADCPPB2SUM Register (Offset = 58h) [Reset = 00000000h]

ADCPPB2SUM is shown in Table 18-117.

Return to the Summary Table.

ADC PPB 2 Final Sum Result Register

Table 18-117. ADCPPB2SUM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 15. Reset type: SYSRSn
15-0	SUM	R	0h	Post Processing Block 2 Oversampling Final Sum. When either a count-match event occurs (PCOUNT = LIMIT) or PPB2 receives a sync. event, the value of PSUM is loaded into this register. In the case of a count-match event, the sum loaded into this register includes the value from the most recent conversion. The value from PSUM will be right shifted by the amount specified in the SHIFT register before being loaded into the final SUM result register. This result is available 1 SYSCLK cycle after the associated ADCPPB2RESULT is available (only in case of a count-match event). This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB2RESULT timing information). Reset type: SYSRSn

18.4.24 ADCPPB2COUNT Register (Offset = 5Ch) [Reset = 0000h]

ADCPPB2COUNT is shown in Table 18-118.

Return to the Summary Table.

ADC PPB2 Final Conversion Count Register

Table 18-118. ADCPPB2COUNT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-7	RESERVED	R	0h	Reserved
6-0	COUNT	R	0h	Post Processing Block 2 Oversampling Final Count. When either a count-match event occurs (PCOUNT = LIMIT) or PPB2 receives a sync. event, the value of PCOUNT is loaded into this register. This result is available 1 SYSCLK cycle after the associated ADCPPB2RESULT is available (only in case of a count-match event). This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB2RESULT timing information). Reset type: SYSRSn

18.4.25 ADCPPB3SUM Register (Offset = 60h) [Reset = 00000000h]

ADCPPB3SUM is shown in Table 18-119.

Return to the Summary Table.

ADC PPB 3 Final Sum Result Register

Table 18-119. ADCPPB3SUM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 15. Reset type: SYSRSn
15-0	SUM	R	0h	Post Processing Block 3 Oversampling Final Sum. When either a count-match event occurs (PCOUNT = LIMIT) or PPB3 receives a sync. event, the value of PSUM is loaded into this register. In the case of a count-match event, the sum loaded into this register includes the value from the most recent conversion. The value from PSUM will be right shifted by the amount specified in the SHIFT register before being loaded into the final SUM result register. This result is available 1 SYSCLK cycle after the associated ADCPPB3RESULT is available (only in case of a count-match event). This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB3RESULT timing information). Reset type: SYSRSn

18.4.26 ADCPPB3COUNT Register (Offset = 64h) [Reset = 0000h]

ADCPPB3COUNT is shown in Table 18-120.

Return to the Summary Table.

ADC PPB3 Final Conversion Count Register

Table 18-120. ADCPPB3COUNT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-7	RESERVED	R	0h	Reserved
6-0	COUNT	R	0h	Post Processing Block 3 Oversampling Final Count. When either a count-match event occurs (PCOUNT = LIMIT) or PPB3 receives a sync. event, the value of PCOUNT is loaded into this register. This result is available 1 SYSCLK cycle after the associated ADCPPB3RESULT is available (only in case of a count-match event). This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB3RESULT timing information). Reset type: SYSRSn

18.4.27 ADCPPB4SUM Register (Offset = 68h) [Reset = 00000000h]

ADCPPB4SUM is shown in Table 18-121.

Return to the Summary Table.

ADC PPB 4 Final Sum Result Register

Table 18-121. ADCPPB4SUM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 15. Reset type: SYSRSn
15-0	SUM	R	0h	Post Processing Block 4 Oversampling Final Sum. When either a count-match event occurs (PCOUNT = LIMIT) or PPB4 receives a sync. event, the value of PSUM is loaded into this register. In the case of a count-match event, the sum loaded into this register includes the value from the most recent conversion. The value from PSUM will be right shifted by the amount specified in the SHIFT register before being loaded into the final SUM result register. This result is available 1 SYSCLK cycle after the associated ADCPPB4RESULT is available (only in case of a count-match event). This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB4RESULT timing information). Reset type: SYSRSn

18.4.28 ADCPPB4COUNT Register (Offset = 6Ch) [Reset = 0000h]

ADCPPB4COUNT is shown in Table 18-122.

Return to the Summary Table.

ADC PPB4 Final Conversion Count Register

Table 18-122. ADCPPB4COUNT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-7	RESERVED	R	0h	Reserved
6-0	COUNT	R	0h	Post Processing Block 4 Oversampling Final Count. When either a count-match event occurs (PCOUNT = LIMIT) or PPB4 receives a sync. event, the value of PCOUNT is loaded into this register. This result is available 1 SYSCLK cycle after the associated ADCPPB4RESULT is available (only in case of a count-match event). This will be 2 SYSCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB4RESULT timing information). Reset type: SYSRSn

18.4.29 ADCSEQ1FIFORESULT Register (Offset = C0h) [Reset = 00000000h]

ADCSEQ1FIFORESULT is shown in Table 18-123.

Return to the Summary Table.

ADC Sequence 1 FIFO Result Register

Table 18-123. ADCSEQ1FIFORESULT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SEQFIFORESULT	R	0h	Sequencer 1 Result in FIFO mode Reset type: SYSRSn

18.4.30 ADCSEQ2FIFORESULT Register (Offset = C4h) [Reset = 00000000h]

ADCSEQ2FIFORESULT is shown in Table 18-124.

Return to the Summary Table.

ADC Sequence 2 FIFO Result Register

Table 18-124. ADCSEQ2FIFORESULT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SEQFIFORESULT	R	0h	Sequencer 2 Result in FIFO mode Reset type: SYSRSn

18.4.31 ADCSEQ3FIFORESULT Register (Offset = C8h) [Reset = 00000000h]

ADCSEQ3FIFORESULT is shown in Table 18-125.

Return to the Summary Table.

ADC Sequence 3 FIFO Result Register

Table 18-125. ADCSEQ3FIFORESULT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SEQFIFORESULT	R	0h	Sequencer 3 Result in FIFO mode Reset type: SYSRSn

18.4.32 ADCSEQ4FIFORESULT Register (Offset = CCh) [Reset = 00000000h]

ADCSEQ4FIFORESULT is shown in Table 18-126.

Return to the Summary Table.

ADC Sequence 4 FIFO Result Register

Table 18-126. ADCSEQ4FIFORESULT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SEQFIFORESULT	R	0h	Sequencer 4 Result in FIFO mode Reset type: SYSRSn

18.4.33 ADCSEQ1FIFOSTATUS Register (Offset = D0h) [Reset = 00000000h]

ADCSEQ1FIFOSTATUS is shown in Table 18-127.

Return to the Summary Table.

ADC Sequence 1 FIFO Status

Table 18-127. ADCSEQ1FIFOSTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	Reserved
4-1	SEQFIFOPENDREAD	R	0h	Sequencer 1 FIFO pending reads Note: 1. This field is only updated when after the last byte of ADCSEQ1FIFORESULT is read. 2. The Pending read information will not be updated for debug reads. 3. Once all the data in the FIFO has been read, FIFO will wrap back. Subsequent reads will read from the beginning of the FIFO. Reset type: SYSRSn
0	SEQFIFOVALID	R	0h	Sequencer 1 FIFO Valid 0 - All Conversions associated with the current Sequencer are NOT complete. 1 - Conversion from all the SOCs associated with the Sequencer are complete and result is available to be read. Reset type: SYSRSn

18.4.34 ADCSEQ2FIFOSTATUS Register (Offset = D4h) [Reset = 00000000h]

ADCSEQ2FIFOSTATUS is shown in Table 18-128.

Return to the Summary Table.

ADC Sequence 2 FIFO Status

Table 18-128. ADCSEQ2FIFOSTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	Reserved
4-1	SEQFIFOPENDREAD	R	0h	Sequencer 2 FIFO pending reads Note: 1. This field is only updated when after the last byte of ADCSEQ2FIFORESULT is read. 2. The Pending read information will not be updated for debug reads. 3. Once all the data in the FIFO has been read, FIFO will wrap back. Subsequent reads will read from the beginning of the FIFO. Reset type: SYSRSn
0	SEQFIFOVALID	R	0h	Sequencer 2 FIFO Valid 0 - All Conversions associated with the current Sequencer are NOT complete. 1 - Conversion from all the SOCs associated with the Sequencer are complete and result is available to be read. Reset type: SYSRSn

18.4.35 ADCSEQ3FIFOSTATUS Register (Offset = D8h) [Reset = 00000000h]

ADCSEQ3FIFOSTATUS is shown in Table 18-129.

Return to the Summary Table.

ADC Sequence 3 FIFO Status

Table 18-129. ADCSEQ3FIFOSTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	Reserved
4-1	SEQFIFOPENDREAD	R	0h	Sequencer 3 FIFO pending reads Note: 1. This field is only updated when after the last byte of ADCSEQ3FIFORESULT is read. 2. The Pending read information will not be updated for debug reads. 3. Once all the data in the FIFO has been read, FIFO will wrap back. Subsequent reads will read from the beginning of the FIFO. Reset type: SYSRSn
0	SEQFIFOVALID	R	0h	Sequencer 3 FIFO Valid 0 - All Conversions associated with the current Sequencer are NOT complete. 1 - Conversion from all the SOCs associated with the Sequencer are complete and result is available to be read. Reset type: SYSRSn

18.4.36 ADCSEQ4FIFOSTATUS Register (Offset = DCh) [Reset = 00000000h]

ADCSEQ4FIFOSTATUS is shown in Table 18-130.

Return to the Summary Table.

ADC Sequence 4 FIFO Status

Table 18-130. ADCSEQ4FIFOSTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	Reserved
4-1	SEQFIFOPENDREAD	R	0h	Sequencer 4 FIFO pending reads Note: 1. This field is only updated when after the last byte of ADCSEQ4FIFORESULT is read. 2. The Pending read information will not be updated for debug reads. 3. Once all the data in the FIFO has been read, FIFO will wrap back. Subsequent reads will read from the beginning of the FIFO. Reset type: SYSRSn
0	SEQFIFOVALID	R	0h	Sequencer 4 FIFO Valid 0 - All Conversions associated with the current Sequencer are NOT complete. 1 - Conversion from all the SOCs associated with the Sequencer are complete and result is available to be read. Reset type: SYSRSn

Chapter 19

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.

19.1 VREF Overview 1083
19.2 VREF Operation 1084
19.3 VREF Registers 1087

19.1 VREF Overview

The VREF module for the MSPM33C3 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

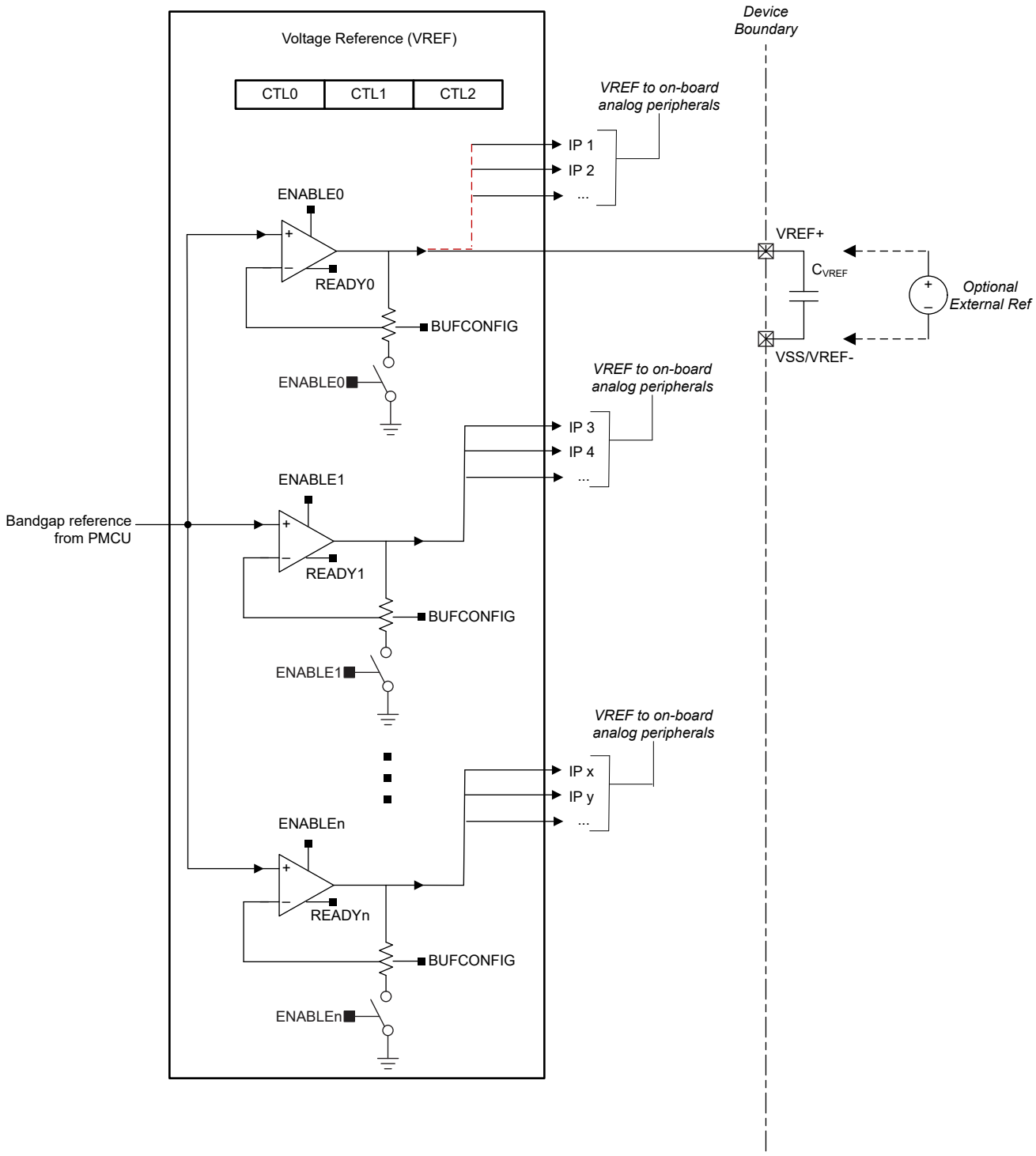


Figure 19-1. MSPM33 VREF Superset Block Diagram

19.2 VREF Operation

The VREF module is configured with user software. The setup and operation of VREF is discussed in the following sections.

19.2.1 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. The first time the VREF is enabled, the READY bit will remain cleared until the VREF is started and settled, after which the READY bit will be set by hardware. If the VREF is disabled, the READY bit will be cleared back to zero by hardware. If the VREF is re-enabled later, the READY bit will not be set, and application software must manage the VREF startup time.

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

19.2.2 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.

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.

19.2.3 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 HSADC

The HSADC peripheral leverages the internal configurable reference or an external reference automastically.

The user must make sure that the VREF reference buffer has settled before triggering an ADC conversion when using an internal reference as the HSADC voltage reference (VR+). Refer to the device-specific data sheet for VREF settling times and refer to the HSADC chapter for more details on all of the reference options available for the ADC peripheral.

VREF to COMP

The comparator peripheral can leverage the 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 OPA

The OPA peripheral can leverage the external reference for setting a voltage bias on the noninverting input terminal of the amplifier using the PSEL control bits in the CFG register.

Refer to the OPA chapter for more details on all of the channels available for the noninverting amplifier input terminal of the amplifier.

19.3 VREF Registers

Table 19-1 lists the memory-mapped registers for the VREF registers. All register offset addresses not listed in Table 19-1 should be considered as reserved locations and the register contents should not be modified.

Table 19-1. VREF Registers

Offset	Acronym	Register Name	Section
800h	PWREN	Power enable	Section 19.3.1
804h	RSTCTL	Reset Control	Section 19.3.2
814h	STAT	Status Register	Section 19.3.3
1000h	CLKDIV	Clock Divider	Section 19.3.4
1008h	CLKSEL	Clock Selection	Section 19.3.5
1100h	CTL0	Control 0	Section 19.3.6
1104h	CTL1	Control 1	Section 19.3.7
1108h	CTL2	Control 2	Section 19.3.8
110Ch	V2IEN	V2I ENABLE REGISTER	Section 19.3.9

Complex bit access types are encoded to fit into small table cells. Table 19-2 shows the codes that are used for access types in this section.

Table 19-2. VREF Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
WK	W K	Write Write protected by a key
Reset or Default Value		
-n		Value after reset or the default value

19.3.1 PWREN Register (Offset = 800h) [Reset = 00000000h]

PWREN is shown in Table 19-3.

Return to the Summary Table.

Register to control the power state

Table 19-3. PWREN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to allow Power State Change 26h = KEY to allow write access to this register
23-1	RESERVED	R/W	0h	
0	ENABLE	R/WK	0h	Enable the power KEY must be set to 26h to write to this bit. 0h = Disable Power 1h = Enable Power

19.3.2 RSTCTL Register (Offset = 804h) [Reset = 00000000h]

RSTCTL is shown in Table 19-4.

Return to the Summary Table.

Register to control reset assertion and de-assertion

Table 19-4. RSTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	Unlock key 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 KEY must be set to B1h to write to this bit. 0h = Writing 0 has no effect 1h = Clear reset sticky bit
0	RESETASSERT	WK	0h	Assert reset to the peripheral KEY must be set to B1h to write to this bit. 0h = Writing 0 has no effect 1h = Assert reset

19.3.3 STAT Register (Offset = 814h) [Reset = 00000000h]

STAT is shown in Table 19-5.

Return to the Summary Table.

peripheral enable and reset status

Table 19-5. 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 0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register 1h = The peripheral was reset since the last bit clear
15-0	RESERVED	R	0h	

19.3.4 CLKDIV Register (Offset = 1000h) [Reset = 00000000h]

CLKDIV is shown in Table 19-6.

Return to the Summary Table.

Clock Divider

Table 19-6. CLKDIV Register 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 to be used in sample and hold logic

19.3.5 CLKSEL Register (Offset = 1008h) [Reset = 00000000h]

CLKSEL is shown in Table 19-7.

Return to the Summary Table.

Clock Selection

Table 19-7. CLKSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	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/W	0h	

19.3.6 CTL0 Register (Offset = 1100h) [Reset = 00000000h]

CTL0 is shown in Table 19-8.

Return to the Summary Table.

Control 0 register.

Table 19-8. CTL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R/W	0h	
8	SHMODE	R/W	0h	This bit enable sample and hold mode 0h = Sample and hold mode is disable 1h = Sample and hold mode is enable
7	BUFCONFIG	R/W	0h	These bits configure output buffer. 0h = output 2p5v : Configure Output Buffer to 2.5v 1h = output 1p4v : Configure Output Buffer to 1.4v
6-2	RESERVED	R/W	0h	
1	COMP_VREF_ENABLE	R/W	0h	Comparator Vref Enable 0h = COMP VREF is disabled 1h = COMP VREF is enabled
0	ENABLE	R/W	0h	This bit enables the VREF module. 0h = VREF is disabled 1h = VREF is enabled

19.3.7 CTL1 Register (Offset = 1104h) [Reset = 00000000h]

CTL1 is shown in Table 19-9.

Return to the Summary Table.

Control 1 register.

Table 19-9. CTL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	0h	
1	VREFLOSEL	R/W	0h	This bit select VREFLO pin
0	READY	R	0h	These bits defines status of VREF 0h = VREF output is not ready 1h = VREF output is ready

19.3.8 CTL2 Register (Offset = 1108h) [Reset = 00000000h]

CTL2 is shown in Table 19-10.

Return to the Summary Table.

Control 2 register.

Table 19-10. CTL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	HCYCLE	R/W	0h	Hold cycle count Total cycles of module clock for hold phase when VREF is working in sample and hold mode in STANDBY to save power. Please refer VREF section of datasheet for recommended values of sample and hold times. 0h = smallest hold cycle FFFFh = largest hold cycle
15-0	SHCYCLE	R/W	0h	Sample and Hold cycle count Total cycles of module clock for sample and hold phase when VREF is working in sample and hold mode in STANDBY to save power. This field should be greater than HCYCLE field. The difference between this field and HCYCLE gives the number of cycles of sample phase. Please refer VREF section of datasheet for recommended values of sample and hold times. 0h = smallest sample and hold cycle count FFFFh = largest sample and hold cycle

19.3.9 V2IEN Register (Offset = 110Ch) [Reset = 00000000h]

V2IEN is shown in Table 19-11.

Return to the Summary Table.

V2I ENABLE register.

Table 19-11. V2IEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	V2I_EN	R/W	0h	V2I enable

Chapter 20

COMP

The comparator module (COMP) is an analog voltage comparator. This chapter describes the operation of the comparator module.

20.1 Comparator Overview 1098
20.2 Comparator Operation 1100
20.3 COMP Registers 1108

20.1 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
- 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
- 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
- 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
- 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

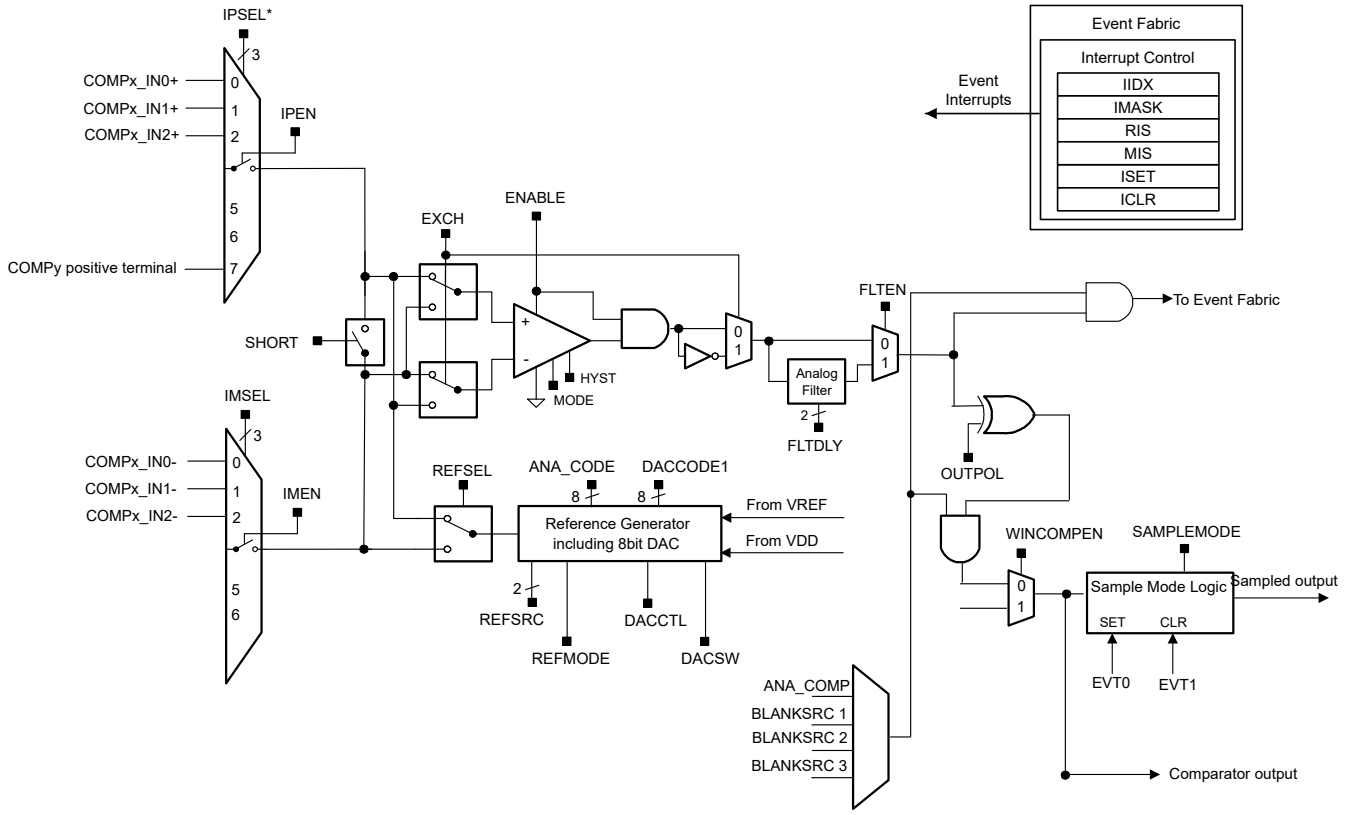


Figure 20-1. Comparator Block Diagram

20.2 Comparator Operation

The comparator module is configured by user software. The setup and operation of comparator is discussed in the following sections.

20.2.1 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. When the comparator is off, OUT is low when OUTPOL bit is set to 0, and OUT is high when OUTPOL bit is set to 1.

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.

20.2.2 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:

- Connection of an external signal 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, IMSEL and EXCH settings in the register COMPx.CTL0. To change the configuration of the comparator it must be first disabled, reconfigured as needed and then re-enabled.

20.2.3 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.

20.2.4 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 20-1).

Table 20-1. Typical Output Filter Delay Settings

FLTDLY bit in COMPx.CTL1	Typical filter delay	Current	Cap
0	60ns	1.4uA	350fF
1	440ns	400nA	350fF
2	1.01us	200nA	350fF
3	2.2us	100nA	350fF

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 20-2). 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.

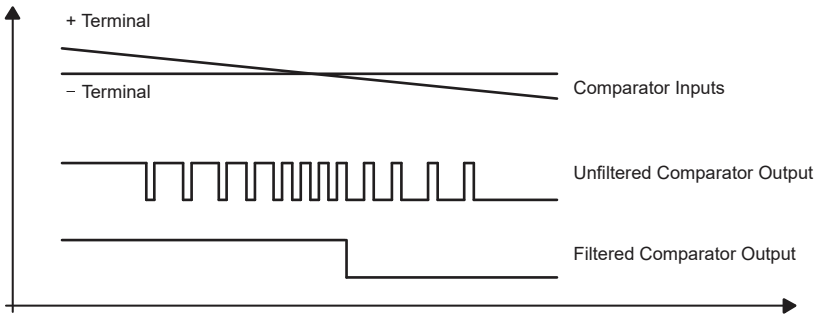


Figure 20-2. Analog Filter Response at the Output of the Comparator

Note

The output filter only works for the comparator when it operates in fast mode.

20.2.5 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, the output of the comparator is captured only 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.

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.

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)

20.2.6 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.

During this delay period, the comparator ignores any further changes in the input signal and maintains its current output state. 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.

20.2.7 Reference Voltage Generator

Figure 20-3 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.

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 also selected on the same terminal using IPSEL/IPEN or IMSEL/IMEN bits in COMPx.CTL0 register then the comparator channel selection takes precedence.

DAC8 output is also connected to the internal analog OPA module through OPAx (see *OPA Block Diagram*).

Figure 20-3. Reference Voltage Generator Block Diagram

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.
- 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.

20.2.8 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 1 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.

20.2.9 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 20-4)

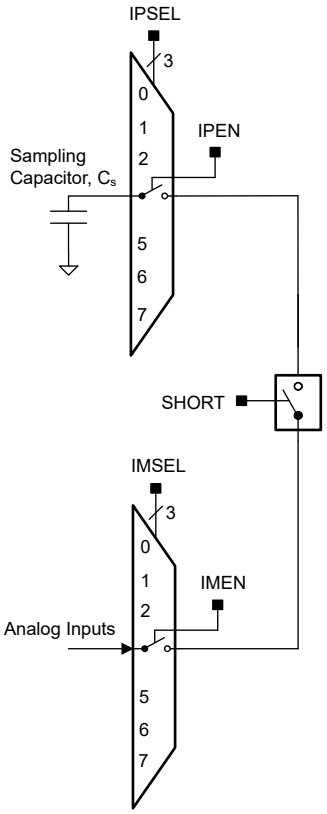


Figure 20-4. Comparator Short Switch

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 17 .

$$T_{au} = (R_I + R_S)\text{x}C_S \quad (17)$$

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.

20.2.10 Analog Comparison Feature

The MSPM33C-series of devices includes an external sequencer that allows the programmer to make multiple comparisons to preset voltages while in stop mode. This sequencer can also be used to generate comparator events that wake up the device if any of these inputs cross the reference voltage. To accomplish this, a set of memory mapped registers (MMR) are used to control the DACCODE0 and VMON input. To see how these signals are routed see Figure 20-5. Please see your device's data sheet for information on which mux input the VMON pin is.

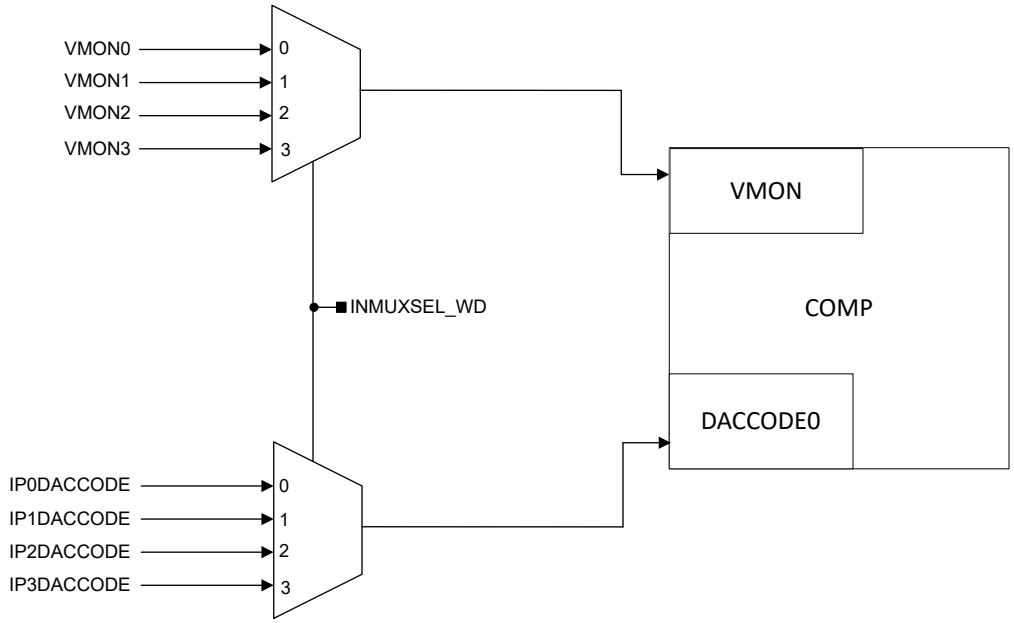


Figure 20-5. Analog signal comparison diagram

The input into the VMON pin of the comparator is selected by the ANACMPCTRCTL MMR. The corresponding IP0DACCODE - IP3DACCODE that the VMON signals is compared to is stored n the ANACMPDACCODE MMR. These both are controlled by the ANACMPWD MMR. There are multiple ways to sequence through the DACCODES and VMON signals. In software the user can manually write to the register and sequence through the VMON values and measure the outputs. The other option is to use the ANACMPCTRCTL register to set a comparison period and a blanking period. The comparison period determines how long the voltage is held and compared against the corresponding DACCODE. The blanking period determines how much time is taken between incrementing the ANACMPCTRCTL register and starting the comparison period. To start the sequencing of the ANACMPWD MMR the ANACMPWKUPEN bit must be set in the ANACMPWKUPCTL register. The sequencing increments the INMUXSEL_WD bits in the ANACMPWD register.

When an COMP event occurs with the ANACMPWKUPEN enabled, a bit in the ANACMPCHLSTS register is set corresponding to which VMON voltage crossed the threshold. This gives the programmer an understanding of which VMON value exceeded the corresponding DACCODE when the interrupt occurs. To clear the corresponding bit the programmer must write to the corresponding bit in the ANACMPCHLSTSCLR register.

When setting the comparison period and blanking period, the CMPPRD and BLKCMPPRD bits in the ANACMPCTRCTL register are used. The CMPPRD bits configure the comparison period. The BLKCMPPRD bits are used for selecting the blank period but is calculated by adding blanking and comparison period. To calculate the set blank period subtract the CMPPRD from the BLKCMPPRD register. The programmer needs sure the BLKCMPPRD is always greater than the CMPPRD register to guarentee there is blanking time on the device.

Using the Analog Comparison Feature While the device is in sleep mode

The typical use case for this feature is to wake up the device from sleep while comparing to multiple reference voltages. Below are the typical steps used to configure the registers for the analog comparison wake up feature.

1. Configure the ANACMPDACCODE voltages with the corresponding reference voltage values
2. Set the corresponding comparison window and blanking period in the ANACMPCTRCTL register.
3. Set the ANACMPWKUPEN pin
4. Put the device into stop mode

After the device is put in sleep mode the INMUXSEL_WD multiplex's through the DACCODE and VMON values. Once the comparator outputs a high voltage COMP interrupt is generated.

20.2.11 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 20-2.

Table 20-2. 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
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

20.2.11.1 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 20-3. 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 8.2.5 for guidance on configuring the Event registers for CPU interrupts.

20.2.11.2 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 20-4) 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 8.1.3.3 and Section 8.2.3 for how generic event route works.

Table 20-4. 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 8.2.5 for guidance on configuring the Event registers.

20.2.11.3 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.

20.3 COMP Registers

Table 20-5 lists the memory-mapped registers for the COMP registers. All register offset addresses not listed in Table 20-5 should be considered as reserved locations and the register contents should not be modified.

Table 20-5. COMP Registers

Offset	Acronym	Register Name	Section
400h	FSUB_0	Subscriber Port 0	Section 20.3.1
404h	FSUB_1	Subscriber Port 1	Section 20.3.2
444h	FPUB_1	Publisher port 1	Section 20.3.3
800h	PWREN	Power enable	Section 20.3.4
804h	RSTCTL	Reset Control	Section 20.3.5
808h	CLKCFG	Peripheral Clock Configuration Register	Section 20.3.6
814h	STAT	Status Register	Section 20.3.7
1020h + formula	IIDX_j	Interrupt index	Section 20.3.8
1028h + formula	IMASK_j	Interrupt mask	Section 20.3.9
1030h + formula	RIS_j	Raw interrupt status	Section 20.3.10
1038h + formula	MIS_j	Masked interrupt status	Section 20.3.11
1040h + formula	ISET_j	Interrupt set	Section 20.3.12
1048h + formula	ICLR_j	Interrupt clear	Section 20.3.13
10E0h	EVT_MODE	Event Mode	Section 20.3.14
10FCh	DESC	Module Description	Section 20.3.15
1100h	CTL0	Control 0	Section 20.3.16
1104h	CTL1	Control 1	Section 20.3.17
1108h	CTL2	Control 2	Section 20.3.18
110Ch	CTL3	Control 3	Section 20.3.19
1120h	STAT	Status	Section 20.3.20
1140h	ANACMPWKUPCTL	Analog comparison wakeup control register	Section 20.3.21
1144h	ANACMPCTRCTL	Analog comparison counter control register	Section 20.3.22
1148h	ANACMPDACCODE	Analog comparison DAC codes	Section 20.3.23
114Ch	ANACMPWD	Window operation control for analog comparison	Section 20.3.24
1150h	ANACMPCHSTS	Status flag to indicate input channel event genration	Section 20.3.25
1154h	ANACMPCHSTSCLR	Clear register for ANACMPCHSTS status register	Section 20.3.26

Complex bit access types are encoded to fit into small table cells. Table 20-6 shows the codes that are used for access types in this section.

Table 20-6. COMP Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
RH	R H	Read Set or cleared by hardware
Write Type		
W	W	Write

Table 20-6. COMP Access Type Codes (continued)

Access Type	Code	Description
WK	W K	Write 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.

20.3.1 FSUB_0 Register (Offset = 400h) [Reset = 00000000h]

FSUB_0 is shown in Table 20-7.

Return to the Summary Table.

Subscriber port

Table 20-7. FSUB_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3-0	CHANID	R/W	0h	0 = disconnected. 1-15 = connected to channelID = CHANID. 0h = A value of 0 specifies that the event is not connected Fh = Consult your device datasheet as the actual allowed maximum may be less than 15.

20.3.2 FSUB_1 Register (Offset = 404h) [Reset = 00000000h]

FSUB_1 is shown in Table 20-8.

Return to the Summary Table.

Subscriber port

Table 20-8. FSUB_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3-0	CHANID	R/W	0h	0 = disconnected. 1-15 = connected to channelID = CHANID. 0h = A value of 0 specifies that the event is not connected Fh = Consult your device datasheet as the actual allowed maximum may be less than 15.

20.3.3 FPUB_1 Register (Offset = 444h) [Reset = 00000000h]

FPUB_1 is shown in Table 20-9.

Return to the Summary Table.

Publisher port

Table 20-9. FPUB_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3-0	CHANID	R/W	0h	0 = disconnected. 1-15 = connected to channelID = CHANID. 0h = A value of 0 specifies that the event is not connected Fh = Consult your device datasheet as the actual allowed maximum may be less than 15.

20.3.4 PWREN Register (Offset = 800h) [Reset = 00000000h]

PWREN is shown in Table 20-10.

Return to the Summary Table.

Register to control the power state

Table 20-10. PWREN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to allow Power State Change 26h = KEY to allow write access to this register
23-1	RESERVED	R/W	0h	
0	ENABLE	R/WK	0h	Enable the power KEY must be set to 26h to write to this bit. 0h = Disable Power 1h = Enable Power

20.3.5 RSTCTL Register (Offset = 804h) [Reset = 00000000h]

RSTCTL is shown in Table 20-11.

Return to the Summary Table.

Register to control reset assertion and de-assertion

Table 20-11. RSTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	Unlock key 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 KEY must be set to B1h to write to this bit. 0h = Writing 0 has no effect 1h = Clear reset sticky bit
0	RESETASSERT	WK	0h	Assert reset to the peripheral KEY must be set to B1h to write to this bit. 0h = Writing 0 has no effect 1h = Assert reset

20.3.6 CLKCFG Register (Offset = 808h) [Reset = 00000000h]

CLKCFG is shown in Table 20-12.

Return to the Summary Table.

Peripheral Clock Configuration Register

Table 20-12. CLKCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow State Change -- 0xA9 A9h = Key value to be used in writing to this register for the write to take effect.
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 0h = disable COMP to request SYSOSC 1h = enable COMP to request SYSOSC
7-0	RESERVED	R/W	0h	

20.3.7 STAT Register (Offset = 814h) [Reset = 00000000h]

STAT is shown in Table 20-13.

Return to the Summary Table.

peripheral enable and reset status

Table 20-13. 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 0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register 1h = The peripheral was reset since the last bit clear
15-0	RESERVED	R	0h	

20.3.8 IIDX_j Register (Offset = 1020h + formula) [Reset = 00000000h]

IIDX_j is shown 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 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.

Offset = 1020h + (j * 30h); where j = 0h to 1h

Table 20-14. IIDX_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1-0	STAT	R	0h	Interrupt index status 0h = No pending interrupt 2h = Comparator output interrupt 3h = Comparator output inverted interrupt 4h = Comparator output ready interrupt

20.3.9 IMASK_j Register (Offset = 1028h + formula) [Reset = 00000000h]

IMASK_j is shown in Table 20-15.

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."

Offset = 1028h + (j * 30h); where j = 0h to 1h

Table 20-15. IMASK_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	OUTRDYIFG	R/W	0h	Masks OUTRDYIFG 0h = Interrupt is masked out 1h = Interrupt will request an interrupt service routine and corresponding bit in MIS will be set
2	COMPINVIFG	R/W	0h	Masks COMPINVIFG 0h = Interrupt is masked out 1h = Interrupt will request an interrupt service routine and corresponding bit in MIS will be set
1	COMPIFG	R/W	0h	Masks COMPIFG 0h = Interrupt is masked out 1h = Interrupt will request an interrupt service routine and corresponding bit in MIS will be set
0	RESERVED	R/W	0h	

20.3.10 RIS_j Register (Offset = 1030h + formula) [Reset = 00000000h]

RIS_j is shown 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.

Offset = 1030h + (j * 30h); where j = 0h to 1h

Table 20-16. RIS_j 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. 0h = No interrupt pending 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. 0h = No interrupt pending 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. 0h = No interrupt pending 1h = Interrupt pending
0	RESERVED	R	0h	

20.3.11 MIS_j Register (Offset = 1038h + formula) [Reset = 00000000h]

MIS_j is shown in Table 20-17.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Offset = 1038h + (j * 30h); where j = 0h to 1h

Table 20-17. MIS_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3	OUTRDYIFG	R	0h	Masked interrupt status for OUTRDYIFG 0h = OUTRDYIFG does not request an interrupt service routine 1h = OUTRDYIFG requests an interrupt service routine
2	COMPINVIFG	R	0h	Masked interrupt status for COMPINVIFG 0h = COMPINVIFG does not request an interrupt service routine 1h = COMPINVIFG requests an interrupt service routine
1	COMPIFG	R	0h	Masked interrupt status for COMPIFG 0h = COMPIFG does not request an interrupt service routine 1h = COMPIFG requests an interrupt service routine
0	RESERVED	R	0h	

20.3.12 ISET_j Register (Offset = 1040h + formula) [Reset = 00000000h]

ISET_j is shown 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.

Offset = 1040h + (j * 30h); where j = 0h to 1h

Table 20-18. ISET_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	W	0h	
3	OUTRDYIFG	W	0h	Sets OUTRDYIFG in RIS register 0h = Writing a 0 has no effect 1h = RIS bit corresponding to OUTRDYIFG is set
2	COMPINVIFG	W	0h	Sets COMPINVIFG in RIS register 0h = Writing a 0 has no effect 1h = RIS bit corresponding to COMPINVIFG is set
1	COMPIFG	W	0h	Sets COMPIFG in RIS register 0h = Writing a 0 has no effect 1h = RIS bit corresponding to COMPIFG is set
0	RESERVED	W	0h	

20.3.13 ICLR_j Register (Offset = 1048h + formula) [Reset = 00000000h]

ICLR_j is shown in Table 20-19.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

Offset = 1048h + (j * 30h); where j = 0h to 1h

Table 20-19. ICLR_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	W	0h	
3	OUTRDYIFG	W	0h	Clears OUTRDYIFG in RIS register 0h = Writing a 0 has no effect 1h = RIS bit corresponding to OUTRDYIFG is cleared
2	COMPINVIFG	W	0h	Clears COMPINVIFG in RIS register 0h = Writing a 0 has no effect 1h = RIS bit corresponding to COMPINVIFG is cleared
1	COMPIFG	W	0h	Clears COMPIFG in RIS register 0h = Writing a 0 has no effect 1h = RIS bit corresponding to COMPIFG is cleared
0	RESERVED	W	0h	

20.3.14 EVT_MODE Register (Offset = 10E0h) [Reset = 00000009h]

EVT_MODE is shown in Table 20-20.

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)

Table 20-20. EVT_MODE Register 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.INT_EVENT[0] 0h = The interrupt or event line is disabled. 1h = Event handled by software. Software must clear the associated RIS flag. 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.INT_EVENT[0] 0h = The interrupt or event line is disabled. 1h = Event handled by software. Software must clear the associated RIS flag. 2h = Event handled by hardware. The hardware (another module) clears automatically the associated RIS flag.

20.3.15 DESC Register (Offset = 10FCh) [Reset = 00000000h]

DESC is shown in Table 20-21.

Return to the Summary Table.

This register identifies the peripheral and its exact version.

Table 20-21. 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

20.3.16 CTL0 Register (Offset = 1100h) [Reset = 00000000h]

CTL0 is shown in Table 20-22.

Return to the Summary Table.

Control 0 register.

Table 20-22. CTL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	IMEN	R/W	0h	Channel input enable for the negative terminal of the comparator. 0h = Selected analog input channel for negative terminal is disabled 1h = Selected analog input channel for negative terminal is enabled
30-19	RESERVED	R/W	0h	
18-16	IMSEL	R/W	0h	Channel input selected for the negative terminal of the comparator if IMEN is set to 1. 0h = Channel 0 selected 1h = Channel 1 selected 2h = Channel 2 selected 3h = Channel 3 selected 4h = Channel 4 selected 5h = Channel 5 selected 6h = Channel 6 selected 7h = Channel 7 selected
15	IPEN	R/W	0h	Channel input enable for the positive terminal of the comparator. 0h = Selected analog input channel for positive terminal is disabled 1h = Selected analog input channel for positive terminal is enabled
14-3	RESERVED	R/W	0h	
2-0	IPSEL	R/W	0h	Channel input selected for the positive terminal of the comparator if IPEN is set to 1. 0h = Channel 0 selected 1h = Channel 1 selected 2h = Channel 2 selected 3h = Channel 3 selected 4h = Channel 4 selected 5h = Channel 5 selected 6h = Channel 6 selected 7h = Channel 7 selected

20.3.17 CTL1 Register (Offset = 1104h) [Reset = 00000000h]

CTL1 is shown in Table 20-23.

Return to the Summary Table.

Control 1 register.

Table 20-23. CTL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R/W	0h	
12	WINCOMPEN	R/W	0h	This bit enables window comparator operation of comparator. 0h = window comparator is disable 1h = window comparator is enable
11	DACOUTEN	R/W	0h	This bit enables DAC8 output to be connected to the pin. 0h = Disable the DAC8 output connectivity to pin. 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. 0h = Typical filter delay of 70 ns 1h = Typical filter delay of 500 ns 2h = Typical filter delay of 1200 ns 3h = Typical filter delay of 2700 ns
8	FLTEN	R/W	0h	This bit enables the analog filter at comparator output. 0h = Comparator output filter is disabled 1h = Comparator output filter is enabled
7	OUTPOL	R/W	0h	This bit selects the comparator output polarity. 0h = Comparator output is non-inverted 1h = Comparator output is inverted
6-5	HYST	R/W	0h	These bits select the hysteresis setting of the comparator. 0h = No hysteresis 1h = Low hysteresis, typical 10mV 2h = Medium hysteresis, typical 20mV 3h = High hysteresis, typical 30mV
4	IES	R/W	0h	This bit selected the interrupt edge for COMPIFG and COMPINVIFG. 0h = Rising edge sets COMPIFG and falling edge sets COMPINVIFG 1h = Falling edge sets COMPIFG and rising edge sets COMPINVIFG
3	SHORT	R/W	0h	This bit shorts the positive and negative input terminals of the comparator. 0h = Comparator positive and negative input terminals are not shorted 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. 0h = Comparator inputs not exchanged and output not inverted 1h = Comparator inputs exchanged and output inverted
1	MODE	R/W	0h	This bit selects the comparator operating mode. 0h = Comparator is in fast mode 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. 0h = Comparator is off 1h = Comparator is on

20.3.18 CTL2 Register (Offset = 1108h) [Reset = 00000000h]

CTL2 is shown in Table 20-24.

Return to the Summary Table.

Control 2 register.

Table 20-24. CTL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R/W	0h	
24	SAMPMODE	R/W	0h	Enable sampled mode of comparator. 0h = Sampled mode disabled 1h = Sampled mode enabled
23-18	RESERVED	R/W	0h	
17	DACSW	R/W	0h	This bit selects between DACCODE0 and DACCODE1 to 8-bit DAC when DACCTL bit is 1. 0h = DACCODE0 selected for 8-bit DAC 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. 0h = Comparator output controls selection between DACCODE0 and DACCODE1 1h = DACSW bit controls selection between DACCODE0 and DACCODE1
15-11	RESERVED	R/W	0h	
10-8	BLANKSRC	R/W	0h	These bits select the blanking source for the comparator. 0h = Blanking source disabled 1h = Select Blanking Source 1 2h = Select Blanking Source 2 3h = Select Blanking Source 3 4h = Select Blanking Source 4 5h = Select Blanking Source 5 6h = Select Blanking Source 6
7	REFSEL	R/W	0h	This bit selects if the selected reference voltage is applied to positive or negative terminal of the comparator. 0h = If EXCH bit is 0, the selected reference is applied to positive terminal. If EXCH bit is 1, the selected reference is applied to negative terminal. 1h = If EXCH bit is 0, the selected reference is applied to negative terminal. If EXCH bit is 1, the selected reference is applied to positive terminal.
6	RESERVED	R/W	0h	
5-3	REFSRC	R/W	0h	These bits select the reference source for the comparator. 0h = Reference voltage generator is disabled (local reference buffer as well as DAC). 1h = VDDA selected as the reference source to DAC and DAC output applied as reference to comparator. 2h = VREF selected as reference to DAC and DAC output applied as reference to comparator. 3h = In devices where internal VREF is buffered and hookedup to extrernal VREF pin, VREF applied as reference to comparator. DAC is switched off. 5h = VDDA is used as comparator reference. 6h = Internal reference selected as the reference source to DAC and DAC output applied as reference to comparator. 7h = Internal VREF is used as the source of comparator. Not all devices will have this option.
2-1	RESERVED	R/W	0h	

Table 20-24. CTL2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
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. 0h = ULP_REF bandgap, local reference buffer and 8-bit DAC inside comparator operate in static mode. 1h = ULP_REF bandgap, local reference buffer and 8-bit DAC inside comparator operate in sampled mode.

20.3.19 CTL3 Register (Offset = 110Ch) [Reset = 00000000h]

CTL3 is shown in Table 20-25.

Return to the Summary Table.

Control 3 register.

Table 20-25. CTL3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	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. 0h = Minimum DAC code value FFh = Minimum DAC code value
15-8	RESERVED	R/W	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. 0h = Minimum DAC code value FFh = Minimum DAC code value

20.3.20 STAT Register (Offset = 1120h) [Reset = 00000000h]

STAT is shown in Table 20-26.

Return to the Summary Table.

Status register.

Table 20-26. 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. 0h = Comparator output is low 1h = Comparator output is high

20.3.21 ANACMPWKUPCTL Register (Offset = 1140h) [Reset = 00000000h]

ANACMPWKUPCTL is shown in Table 20-27.

Return to the Summary Table.

Analog comparison wakeup control register

Table 20-27. ANACMPWKUPCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	Analog comparison wakeup enable bit	R/W	0h	0: Analog comparison wakeup is disabled (back ward compatible with existing IP) 1 : Analog comparison feature is enabled

20.3.22 ANACMPCTRCTL Register (Offset = 1144h) [Reset = 00000000h]

ANACMPCTRCTL is shown in Table 20-28.

Return to the Summary Table.

Analog comparison counter control register

Table 20-28. ANACMPCTRCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-8	CMPPRD	R/W	0h	Comparision period
7-0	BLKCMPPRD	R/W	0h	Blanking and comparision period

20.3.23 ANACMPDACCODE Register (Offset = 1148h) [Reset = 00000000h]

ANACMPDACCODE is shown in Table 20-29.

Return to the Summary Table.

Analog comparison DAC codes

Table 20-29. ANACMPDACCODE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	IP3DACCODE	R/W	0h	Input3 daccode value
23-16	IP2DACCODE	R/W	0h	Input2 daccode value
15-8	IP1DACCODE	R/W	0h	Input1 daccode value
7-0	IP0DACCODE	R/W	0h	Input0 daccode value

20.3.24 ANACMPWD Register (Offset = 114Ch) [Reset = 00000000h]

ANACMPWD is shown in Table 20-30.

Return to the Summary Table.

Window operation control for analog comparison

Table 20-30. ANACMPWD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-1	INMUXSEL_WD	R/W	0h	Input mux (Static) selection for window mode
0	WDEN	R/W	0h	0 : Window mode disable 1 : Window mode enable

20.3.25 ANACMPCHSTS Register (Offset = 1150h) [Reset = 00000000h]

ANACMPCHSTS is shown in Table 20-31.

Return to the Summary Table.

Status flag to indicate input channel event genration

Table 20-31. ANACMPCHSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3	IN3EVT	R	0h	Input3 caused event
2	IN2EVT	R	0h	Input2 caused event
1	IN1EVT	R	0h	Input1 caused event
0	IN0EVT	RH	0h	Input0 caused event

20.3.26 ANACMPCHSTSCLR Register (Offset = 1154h) [Reset = 00000000h]

ANACMPCHSTSCLR is shown in Table 20-32.

Return to the Summary Table.

Clear register for ANACMPCHSTS status register

Table 20-32. ANACMPCHSTSCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	W	0h	
3	IN3EVTCLR	W	0h	Write a 1 to clear corresponding Interrupt.
2	IN2EVTCLR	W	0h	Write a 1 to clear corresponding Interrupt.
1	IN1EVTCLR	W	0h	Write a 1 to clear corresponding Interrupt.
0	IN0EVTCLR	W	0h	Write a 1 to clear corresponding Interrupt.

Chapter 21
UNICOMM

The UNICOMM module is a run time configurable peripheral capable of supporting UART, I^2C and SPI.

21.1 Overview 1138
21.2 Unicomm Architecture 1140
21.3 High-Level Initialization 1145
21.4 UNICOMM/SPGSS Registers 1146

21.1 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. The Table 21-1 table below 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 21-1. 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

Table 21-1. 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 - Chapter 22
- SPI - Chapter 24
- I2CC / I2CT - Chapter 23

21.1.1 Block Diagram

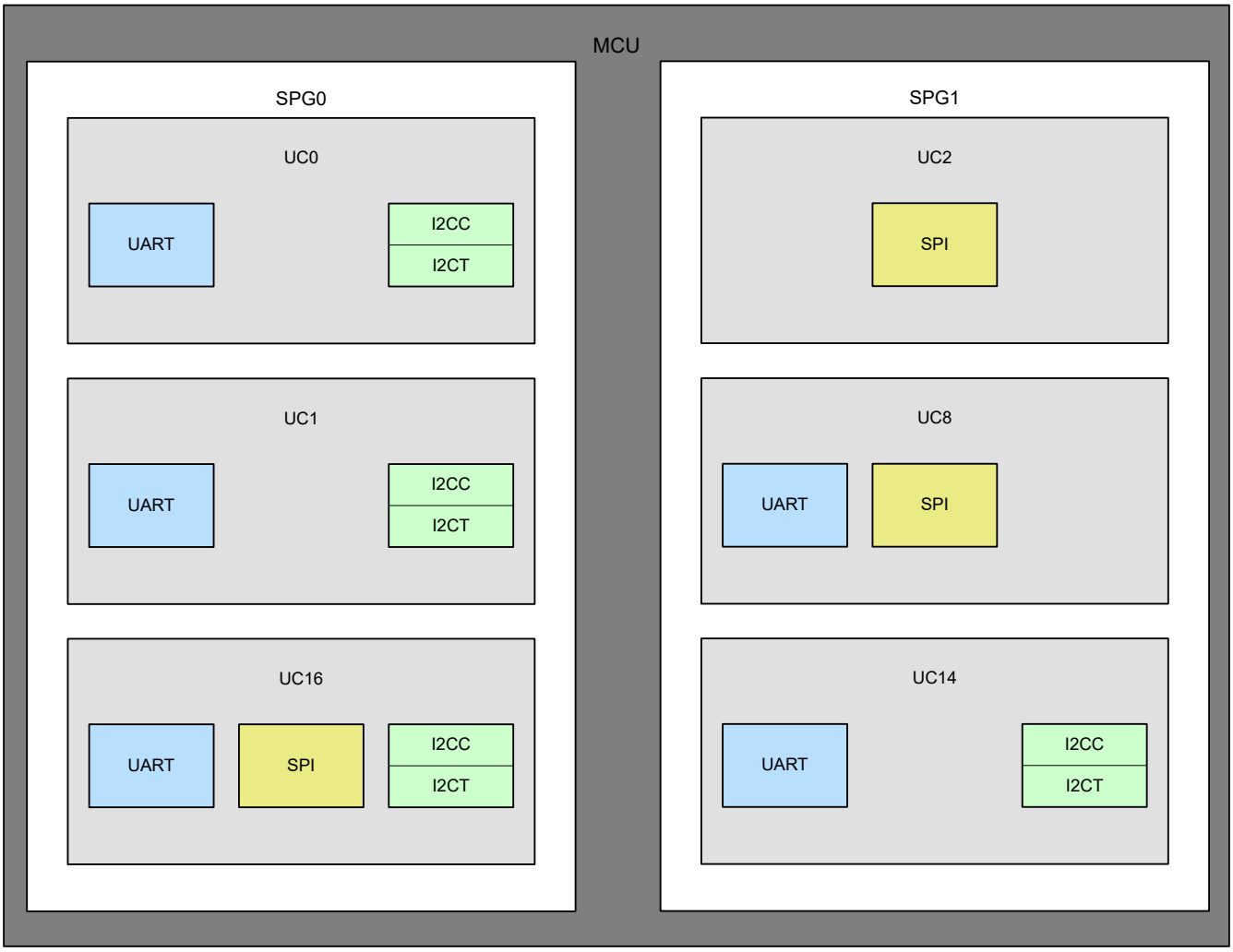


Figure 21-1. Unicomm Block Diagram

21.2 Unicomm Architecture

21.2.1 Serial Peripheral Group (SPG) Configurations

The Scalable Peripheral Group (SPG) level configurations consist of I2C pairing operation in compliance with the SMBUS protocol.

Note

Users not employing either of the loopback or I2C pairing test features can skip to Section 21.2.2.

21.2.1.1 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.

I2C pairing connects the SDA line to the SDA line and the SCL line to the SCL line between two different UNICOMM instances inside the same scalable peripheral group (SPG). 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.

21.2.2 FIFO Operation

Each UNICOMM peripheral has two dedicated FIFOs, one for receive operations and one for transmit operations. The FIFO level varies by device, but is indicated in the device-specific data sheet. The FIFOs reduce service overhead by efficiently storing data to be sent while the transmitter is busy or storing received data to be processed while the CPU is busy. The TXDATA and RXDATA registers are used to interface with the transmit and receive FIFOs respectively.

The FIFO interrupt threshold for each UNICOMM instance is configured in the IFLS register within the peripheral-specific interface registers, even though the configuration options are the same across all IP modes.

21.2.2.1 Receive FIFO Interrupt Thresholds

The configurable FIFO condition options for the Receive (RX) FIFO are:

- FIFO is full
- FIFO is at least 1/16 full - "not empty"
- FIFO is ¼ full
- FIFO is ½ full
- FIFO is ¾ full
- FIFO is 15/16 full - "almost full"
- FIFO is 1/16 full - "almost empty"

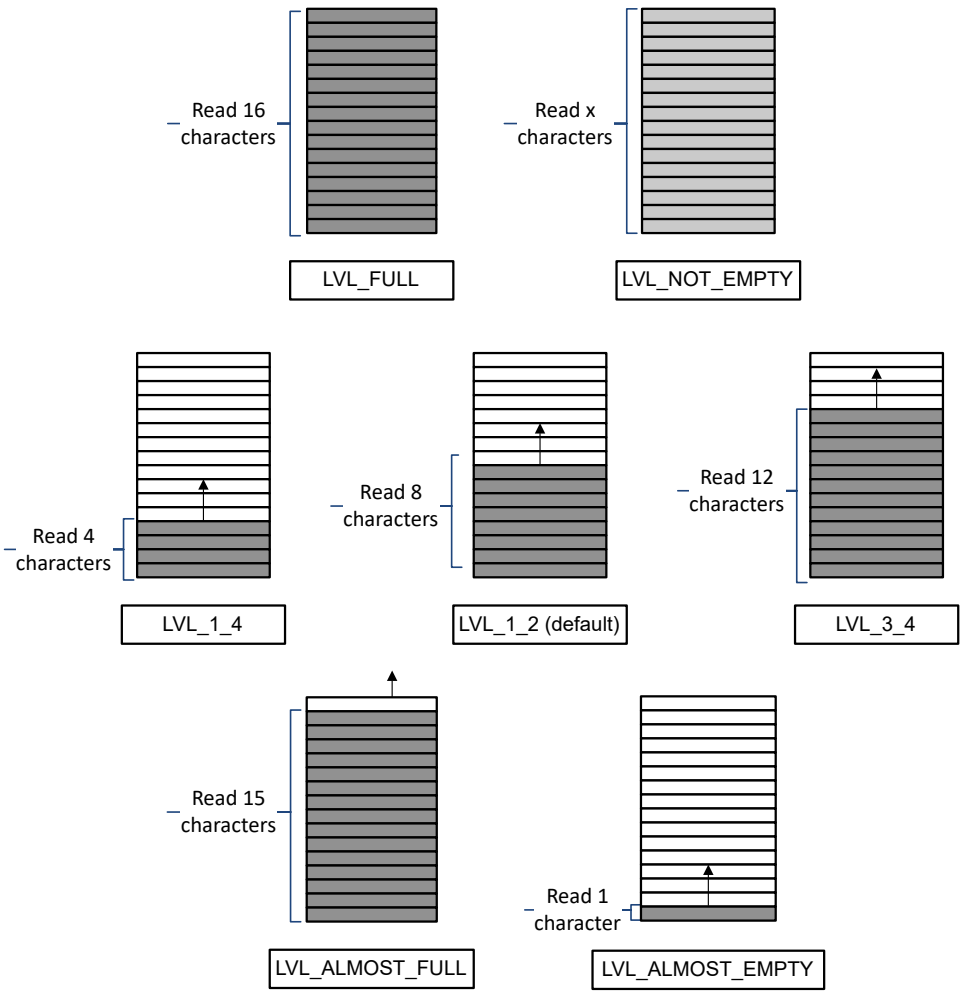


Figure 21-2. UNICOMM Receive FIFO Interrupt Thresholds

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) equal how many are present in the FIFO. For the "not empty" condition, the application can write one element at a time while the condition is satisfied.

Note

Debugger reads of RXDATA have no effect on the contents in the RX FIFO.

21.2.2.2 Transmitter FIFO Interrupt Thresholds

The configurable FIFO condition options for the Transmit (TX) FIFO are:

- FIFO is empty
- FIFO is at least 1/16 empty - "not full"
- FIFO is ¼ full
- FIFO is ½ full
- FIFO is ¾ full
- FIFO is 15/16 full - "almost full"
- FIFO is 1/16 full - "almost empty"

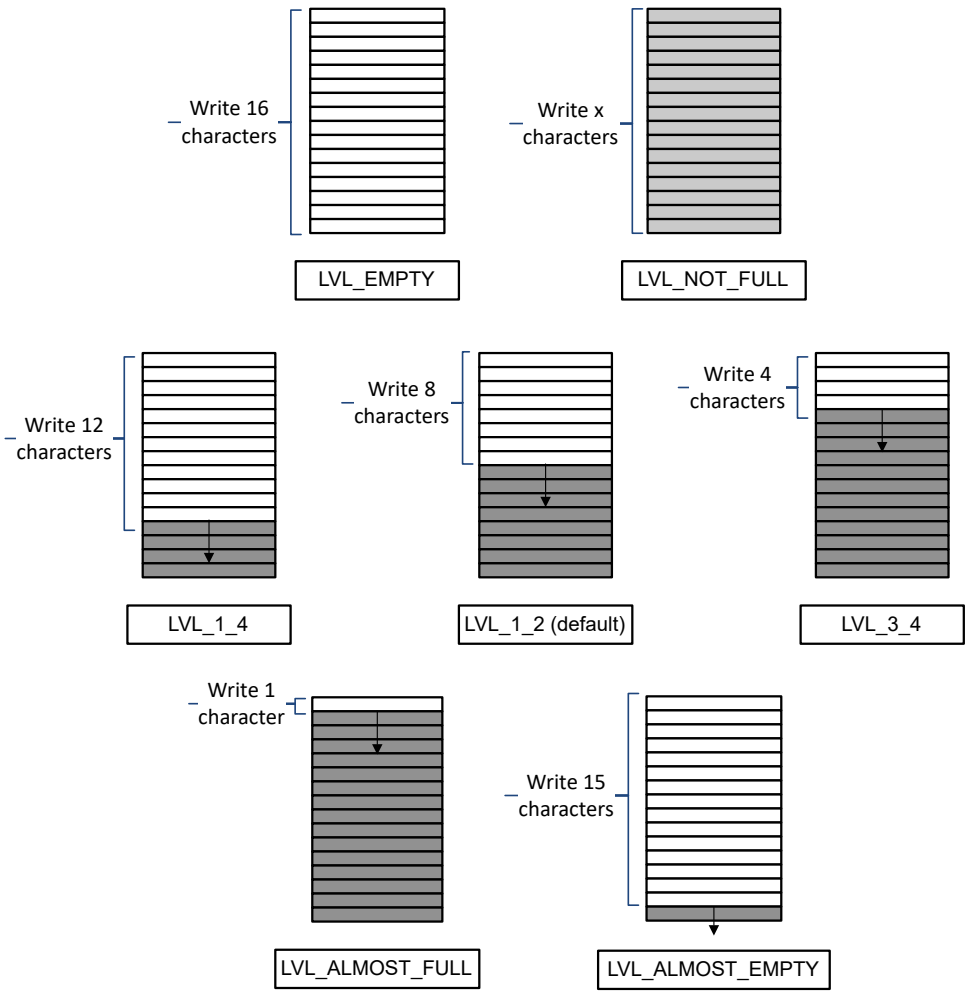


Figure 21-3. UNICOMM Transmit FIFO Interrupt Thresholds

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 empty in the FIFO. Often this value is the highest possible even factor of the number of characters that are sent in each frame. For the "not full" condition, the application can write one element at a time while the condition is satisfied.

Note

Debugger writes of TXDATA will add elements to the TX FIFO the same as a CPU write.

21.2.2.3 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 interrupt threshold configuration in the RXIFSEL or TXIFSEL fields is recommended.

Note

Avoid performing step 1 repeatedly before full sequence is complete.

21.2.3 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.

21.2.4 Interrupts

Each UNICOMM peripheral contains the six standard interrupt Section 8.2.5.1: IIDX, IMASK, RIS, MIS, ISET and ICLR in the CPU_INT register group. These configuration, control, and status registers dictate how the UNICOMM module interfaces with the device interrupt controller module. Unique conditional flags are available in these registers to generate CPU interrupts for each specific peripheral mode (UART, SPI, I2CC, or I2CT). Each UNICOMM instance has one interrupt line connected to the interrupt controller.

21.2.4.1 Receive Interrupt Sequence

The receive interrupt flags for each peripheral specific UNICOMM instance are described below. These flags are raised when there is data present in the RX FIFO that is ready to be read by the CPU or DMA.

- UART peripheral mode - RXINT flag
- SPI peripheral mode - RX flag
- I2CC peripheral mode - RXTRG flag
- I2CT peripheral mode - RXTRG flag

Below is the high-level sequence for receiving a fixed amount of data with interrupts.

1. The RX FIFO contains greater than or equal to the number of bytes/words designated by the programmed trigger interrupt threshold in IFLS.RXIFSEL.
2. The interrupt flag for the configured peripheral mode is set and an interrupt is triggered on the CPU.
3. In the Interrupt Service Routine (ISR), the CPU reads the fixed amount of data from the RXDATA register (repeated reads). This data is then read from the RX FIFO automatically by hardware.

Depending on the peripheral mode configured, additional steps before and/or after this sequence may be required to receive the data using the corresponding protocol. See the peripheral-specific UNICOMM chapters for more details about the software steps needed for each IP.

21.2.4.2 Transmit Interrupt Sequence

The transmit interrupt flags for each peripheral specific UNICOMM instance are described below. These flags are raised when there is space present in the TX FIFO for the CPU or DMA to write data.

- UART peripheral mode - TXINT flag

- SPI peripheral mode - TX flag
- I2CC peripheral mode - TXTRG flag
- I2CT peripheral mode - TXTRG flag

Below is the high-level sequence for transmitting a fixed amount of data with interrupts.

1. The TX FIFO contains less than or equal to the number of bytes/words designated by the programmed trigger interrupt threshold in IFLS.TXIFSEL.
2. The interrupt flag for the configured peripheral mode is set and an interrupt is triggered on the CPU.
3. In the Interrupt Service Routine (ISR), the CPU writes the fixed amount of data to the TXDATA register. This data is then written to the TX FIFO automatically by hardware.

Depending on the peripheral mode configured, additional steps before and/or after this sequence may be required to transmit the data using the corresponding protocol. See the peripheral-specific UNICOMM chapters for more details about the software steps needed for each IP.

21.2.5 DMA Operation

Each UNICOMM peripheral has access to the DMA module on the device and can be configured to trigger DMA channels. From the UNICOMM point-of-view, DMA triggers are setup through the registers in the DMA_TRIG_RX and DMA_TRIG_TX register groups. Different conditionals are available to trigger the DMA_TRIG_RX line and the DMA_TRIG_TX line depending on the IPMODE configured. See the below sections from the peripheral-specific chapters for the options available.

- UART mode DMA conditionals: UART DMA Event Support
- SPI mode DMA conditionals: SPI DMA Event Support
- I2CC/I2CT mode DMA conditionals: I2C DMA Event Support

Note

Some modules only have one available conditional per DMA line.

See *DMA Trigger* for how to configure one of the UNICOMM signals as the trigger source for a DMA channel in the DMA registers.

21.3 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, configure the peripheral mode for the UCx instance in the IPMODE.SELECT field from the Section 21.4.1 to operate as a UART, SPI, I2C Controller, or I2C Target.
2. (Optionally) Configure the inter-module internal loopback/pairing settings in Section 21.4.2 between two UCx modules in the same SPG. See the UNICOMM Section 21.1.1 to determine which UCx pairings are available on this device.
   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.
3. Enable the pin functionality for each GPIO being used in the peripheral configuration in the Chapter 9 registers.
4. Put the UNICOMM instance into reset by setting the RSCTL.RESETASSERT register bit from Section 21.4.1.
5. Enable power to the UNICOMM instance by writing "0hB1" to the KEY field and setting the EN bit for the PWREN register in Section 21.4.1.
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 be MCLKDIV2, 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.

7. Pull the UNICOMM instance out of reset by clearing the RSCTL.RESETASSERT register bit in Section 21.4.1.

21.4 UNICOMM/SPGSS Registers

21.4.1 UNICOMM Registers

This Section describes the UNICOMM Registers.

21.4.1.1 UNICOMM Registers

Table 21-2 lists the memory-mapped registers for the UNICOMM registers. All register offset addresses not listed in Table 21-2 should be considered as reserved locations and the register contents should not be modified.

Table 21-2. 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 21-3 shows the codes that are used for access types in this section.

Table 21-3. UNICOMM Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
WK	W K	Write Write protected by a key
Reset or Default Value		
-n		Value after reset or the default value

21.4.1.1.1 PWREN (Offset = 800h) [Reset = 00000000h]

PWREN is shown in Figure 21-4 and described in Table 21-4.

Return to the Summary Table.

Register to control the power state

Figure 21-4. 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 21-4. PWREN Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to allow Power State Change 26h = KEY to allow write access to this register
23-1	RESERVED	R	0h	
0	ENABLE	R/WK	0h	Enable the power KEY must be set to 26h to write to this bit. 0h = Disable Power 1h = Enable Power

21.4.1.1.2 RSTCTL (Offset = 804h) [Reset = 00000000h]

RSTCTL is shown in Figure 21-5 and described in Table 21-5.

Return to the Summary Table.

Register to control reset assertion and de-assertion

Figure 21-5. 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 21-5. RSTCTL Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	Unlock key 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 KEY must be set to B1h to write to this bit. 0h = Writing 0 has no effect 1h = Clear reset sticky bit
0	RESETASSERT	WK	0h	Assert reset to the peripheral KEY must be set to B1h to write to this bit. 0h = Writing 0 has no effect 1h = Assert reset

21.4.1.1.3 CLKCFG (Offset = 808h) [Reset = 00000000h]

CLKCFG is shown in Figure 21-6 and described in Table 21-6.

Return to the Summary Table.

Peripheral Clock Configuration Register

Figure 21-6. 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 21-6. CLKCFG Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow State Change -- 0xA9 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 0h = Not block async clock request 1h = Block async clock request
7-0	RESERVED	R	0h	

21.4.1.1.4 STAT (Offset = 814h) [Reset = 00000000h]

STAT is shown in Figure 21-7 and described in Table 21-7.

Return to the Summary Table.

peripheral enable and reset status

Figure 21-7. 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 21-7. 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 0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register 1h = The peripheral was reset since the last bit clear
15-0	RESERVED	R	0h	

21.4.1.1.5 IPMODE (Offset = 1100h) [Reset = 00000000h]

IPMODE is shown in Figure 21-8 and described in Table 21-8.

Return to the Summary Table.

Used for selecting mode - UART / I2C Controller / I2C Peripheral / SPI

Figure 21-8. 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 21-8. IPMODE Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1-0	SELECT	R/W	0h	Free run control 0h = Default functionality; UNICOMM in UART mode 1h = UNICOMM in SPI Mode 2h = UNICOMM in I2C Controller Mode 3h = UNICOMM in I2C Peripheral/Target Mode

21.4.2 SPG Registers

This Section describes the SPG Registers.

21.4.2.1 SPGSS Registers

Table 21-9 lists the memory-mapped registers for the SPGSS registers. All register offset addresses not listed in Table 21-9 should be considered as reserved locations and the register contents should not be modified.

Table 21-9. 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 21-10 shows the codes that are used for access types in this section.

Table 21-10. 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

21.4.2.1.1 PAIR0 (Offset = 1C0h) [Reset = 00000000h]

PAIR0 is shown in Figure 21-9 and described in Table 21-11.

Return to the Summary Table.

Pairing of I2C for Multi-Controller/SMBUS applications

Figure 21-9. 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 21-11. 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 0h = UNICOMM peripheral/target for loopback. 1h = UNICOMM peripheral/target for loopback. 2h = UNICOMM peripheral/target for loopback. 3h = UNICOMM peripheral/target for loopback. 4h = UNICOMM peripheral/target for loopback. 5h = UNICOMM peripheral/target for loopback. 6h = UNICOMM peripheral/target for loopback. 7h = UNICOMM peripheral/target for loopback. 8h = UNICOMM peripheral/target for loopback. 9h = UNICOMM peripheral/target for loopback. Ah = UNICOMM peripheral/target for loopback. Bh = UNICOMM peripheral/target for loopback. Ch = UNICOMM peripheral/target for loopback. Dh = UNICOMM peripheral/target for loopback. Eh = UNICOMM peripheral/target for loopback. Fh = UNICOMM peripheral/target for loopback.
11-8	CTL	R/W	0h	UNICOMM instance for loopback. This instance will be treated as the controller. 0h = UNICOMM controller for loopback. 1h = UNICOMM controller for loopback. 2h = UNICOMM controller for loopback. 3h = UNICOMM controller for loopback. 4h = UNICOMM controller for loopback. 5h = UNICOMM controller for loopback. 6h = UNICOMM controller for loopback. 7h = UNICOMM controller for loopback. 8h = UNICOMM controller for loopback. 9h = UNICOMM controller for loopback. Ah = UNICOMM controller for loopback. Bh = UNICOMM controller for loopback. Ch = UNICOMM controller for loopback. Dh = UNICOMM controller for loopback. Eh = UNICOMM controller for loopback. Fh = UNICOMM controller for loopback.
7-1	RESERVED	R	0h	
0	EN	R/W	0h	Enable 0h = Disable 1h = Enable

21.4.2.1.2 PAIR1 (Offset = 1C4h) [Reset = 00000000h]

PAIR1 is shown in Figure 21-10 and described in Table 21-12.

Return to the Summary Table.

Pairing of I2C for Multi-Controller/SMBUS applications

Figure 21-10. 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 21-12. 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 0h = UNICOMM peripheral/target for loopback. 1h = UNICOMM peripheral/target for loopback. 2h = UNICOMM peripheral/target for loopback. 3h = UNICOMM peripheral/target for loopback. 4h = UNICOMM peripheral/target for loopback. 5h = UNICOMM peripheral/target for loopback. 6h = UNICOMM peripheral/target for loopback. 7h = UNICOMM peripheral/target for loopback. 8h = UNICOMM peripheral/target for loopback. 9h = UNICOMM peripheral/target for loopback. Ah = UNICOMM peripheral/target for loopback. Bh = UNICOMM peripheral/target for loopback. Ch = UNICOMM peripheral/target for loopback. Dh = UNICOMM peripheral/target for loopback. Eh = UNICOMM peripheral/target for loopback. Fh = UNICOMM peripheral/target for loopback.
11-8	CTL	R/W	0h	UNICOMM instance for loopback. This instance will be treated as the controller. 0h = UNICOMM controller for loopback. 1h = UNICOMM controller for loopback. 2h = UNICOMM controller for loopback. 3h = UNICOMM controller for loopback. 4h = UNICOMM controller for loopback. 5h = UNICOMM controller for loopback. 6h = UNICOMM controller for loopback. 7h = UNICOMM controller for loopback. 8h = UNICOMM controller for loopback. 9h = UNICOMM controller for loopback. Ah = UNICOMM controller for loopback. Bh = UNICOMM controller for loopback. Ch = UNICOMM controller for loopback. Dh = UNICOMM controller for loopback. Eh = UNICOMM controller for loopback. Fh = UNICOMM controller for loopback.
7-1	RESERVED	R	0h	
0	EN	R/W	0h	Enable 0h = Disable 1h = Enable

21.4.2.1.3 PAIR2 (Offset = 1C8h) [Reset = 00000000h]

PAIR2 is shown in Figure 21-11 and described in Table 21-13.

Return to the Summary Table.

Pairing of I2C for Multi-Controller/SMBUS applications

Figure 21-11. 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 21-13. 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 0h = UNICOMM peripheral/target for loopback. 1h = UNICOMM peripheral/target for loopback. 2h = UNICOMM peripheral/target for loopback. 3h = UNICOMM peripheral/target for loopback. 4h = UNICOMM peripheral/target for loopback. 5h = UNICOMM peripheral/target for loopback. 6h = UNICOMM peripheral/target for loopback. 7h = UNICOMM peripheral/target for loopback. 8h = UNICOMM peripheral/target for loopback. 9h = UNICOMM peripheral/target for loopback. Ah = UNICOMM peripheral/target for loopback. Bh = UNICOMM peripheral/target for loopback. Ch = UNICOMM peripheral/target for loopback. Dh = UNICOMM peripheral/target for loopback. Eh = UNICOMM peripheral/target for loopback. Fh = UNICOMM peripheral/target for loopback.
11-8	CTL	R/W	0h	UNICOMM instance for loopback. This instance will be treated as the controller. 0h = UNICOMM controller for loopback. 1h = UNICOMM controller for loopback. 2h = UNICOMM controller for loopback. 3h = UNICOMM controller for loopback. 4h = UNICOMM controller for loopback. 5h = UNICOMM controller for loopback. 6h = UNICOMM controller for loopback. 7h = UNICOMM controller for loopback. 8h = UNICOMM controller for loopback. 9h = UNICOMM controller for loopback. Ah = UNICOMM controller for loopback. Bh = UNICOMM controller for loopback. Ch = UNICOMM controller for loopback. Dh = UNICOMM controller for loopback. Eh = UNICOMM controller for loopback. Fh = UNICOMM controller for loopback.
7-1	RESERVED	R	0h	
0	EN	R/W	0h	Enable 0h = Disable 1h = Enable

21.4.2.1.4 PAIR3 (Offset = 1CCh) [Reset = 00000000h]

PAIR3 is shown in Figure 21-12 and described in Table 21-14.

Return to the Summary Table.

Pairing of I2C for Multi-Controller/SMBUS applications

Figure 21-12. 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 21-14. 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 0h = UNICOMM peripheral/target for loopback. 1h = UNICOMM peripheral/target for loopback. 2h = UNICOMM peripheral/target for loopback. 3h = UNICOMM peripheral/target for loopback. 4h = UNICOMM peripheral/target for loopback. 5h = UNICOMM peripheral/target for loopback. 6h = UNICOMM peripheral/target for loopback. 7h = UNICOMM peripheral/target for loopback. 8h = UNICOMM peripheral/target for loopback. 9h = UNICOMM peripheral/target for loopback. Ah = UNICOMM peripheral/target for loopback. Bh = UNICOMM peripheral/target for loopback. Ch = UNICOMM peripheral/target for loopback. Dh = UNICOMM peripheral/target for loopback. Eh = UNICOMM peripheral/target for loopback. Fh = UNICOMM peripheral/target for loopback.
11-8	CTL	R/W	0h	UNICOMM instance for loopback. This instance will be treated as the controller. 0h = UNICOMM controller for loopback. 1h = UNICOMM controller for loopback. 2h = UNICOMM controller for loopback. 3h = UNICOMM controller for loopback. 4h = UNICOMM controller for loopback. 5h = UNICOMM controller for loopback. 6h = UNICOMM controller for loopback. 7h = UNICOMM controller for loopback. 8h = UNICOMM controller for loopback. 9h = UNICOMM controller for loopback. Ah = UNICOMM controller for loopback. Bh = UNICOMM controller for loopback. Ch = UNICOMM controller for loopback. Dh = UNICOMM controller for loopback. Eh = UNICOMM controller for loopback. Fh = UNICOMM controller for loopback.
7-1	RESERVED	R	0h	
0	EN	R/W	0h	Enable 0h = Disable 1h = Enable

21.4.2.1.5 PAIR4 (Offset = 1D0h) [Reset = 00000000h]

PAIR4 is shown in Figure 21-13 and described in Table 21-15.

Return to the Summary Table.

Pairing of I2C for Multi-Controller/SMBUS applications

Figure 21-13. 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 21-15. 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 0h = UNICOMM peripheral/target for loopback. 1h = UNICOMM peripheral/target for loopback. 2h = UNICOMM peripheral/target for loopback. 3h = UNICOMM peripheral/target for loopback. 4h = UNICOMM peripheral/target for loopback. 5h = UNICOMM peripheral/target for loopback. 6h = UNICOMM peripheral/target for loopback. 7h = UNICOMM peripheral/target for loopback. 8h = UNICOMM peripheral/target for loopback. 9h = UNICOMM peripheral/target for loopback. Ah = UNICOMM peripheral/target for loopback. Bh = UNICOMM peripheral/target for loopback. Ch = UNICOMM peripheral/target for loopback. Dh = UNICOMM peripheral/target for loopback. Eh = UNICOMM peripheral/target for loopback. Fh = UNICOMM peripheral/target for loopback.
11-8	CTL	R/W	0h	UNICOMM instance for loopback. This instance will be treated as the controller. 0h = UNICOMM controller for loopback. 1h = UNICOMM controller for loopback. 2h = UNICOMM controller for loopback. 3h = UNICOMM controller for loopback. 4h = UNICOMM controller for loopback. 5h = UNICOMM controller for loopback. 6h = UNICOMM controller for loopback. 7h = UNICOMM controller for loopback. 8h = UNICOMM controller for loopback. 9h = UNICOMM controller for loopback. Ah = UNICOMM controller for loopback. Bh = UNICOMM controller for loopback. Ch = UNICOMM controller for loopback. Dh = UNICOMM controller for loopback. Eh = UNICOMM controller for loopback. Fh = UNICOMM controller for loopback.
7-1	RESERVED	R	0h	
0	EN	R/W	0h	Enable 0h = Disable 1h = Enable

21.4.2.1.6 PAIR5 (Offset = 1D4h) [Reset = 00000000h]

PAIR5 is shown in Figure 21-14 and described in Table 21-16.

Return to the Summary Table.

Pairing of I2C for Multi-Controller/SMBUS applications

Figure 21-14. 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 21-16. 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 0h = UNICOMM peripheral/target for loopback. 1h = UNICOMM peripheral/target for loopback. 2h = UNICOMM peripheral/target for loopback. 3h = UNICOMM peripheral/target for loopback. 4h = UNICOMM peripheral/target for loopback. 5h = UNICOMM peripheral/target for loopback. 6h = UNICOMM peripheral/target for loopback. 7h = UNICOMM peripheral/target for loopback. 8h = UNICOMM peripheral/target for loopback. 9h = UNICOMM peripheral/target for loopback. Ah = UNICOMM peripheral/target for loopback. Bh = UNICOMM peripheral/target for loopback. Ch = UNICOMM peripheral/target for loopback. Dh = UNICOMM peripheral/target for loopback. Eh = UNICOMM peripheral/target for loopback. Fh = UNICOMM peripheral/target for loopback.
11-8	CTL	R/W	0h	UNICOMM instance for loopback. This instance will be treated as the controller. 0h = UNICOMM controller for loopback. 1h = UNICOMM controller for loopback. 2h = UNICOMM controller for loopback. 3h = UNICOMM controller for loopback. 4h = UNICOMM controller for loopback. 5h = UNICOMM controller for loopback. 6h = UNICOMM controller for loopback. 7h = UNICOMM controller for loopback. 8h = UNICOMM controller for loopback. 9h = UNICOMM controller for loopback. Ah = UNICOMM controller for loopback. Bh = UNICOMM controller for loopback. Ch = UNICOMM controller for loopback. Dh = UNICOMM controller for loopback. Eh = UNICOMM controller for loopback. Fh = UNICOMM controller for loopback.
7-1	RESERVED	R	0h	
0	EN	R/W	0h	Enable 0h = Disable 1h = Enable

21.4.2.1.7 PAIR6 (Offset = 1D8h) [Reset = 00000000h]

PAIR6 is shown in Figure 21-15 and described in Table 21-17.

Return to the Summary Table.

Pairing of I2C for Multi-Controller/SMBUS applications

Figure 21-15. 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 21-17. 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 0h = UNICOMM peripheral/target for loopback. 1h = UNICOMM peripheral/target for loopback. 2h = UNICOMM peripheral/target for loopback. 3h = UNICOMM peripheral/target for loopback. 4h = UNICOMM peripheral/target for loopback. 5h = UNICOMM peripheral/target for loopback. 6h = UNICOMM peripheral/target for loopback. 7h = UNICOMM peripheral/target for loopback. 8h = UNICOMM peripheral/target for loopback. 9h = UNICOMM peripheral/target for loopback. Ah = UNICOMM peripheral/target for loopback. Bh = UNICOMM peripheral/target for loopback. Ch = UNICOMM peripheral/target for loopback. Dh = UNICOMM peripheral/target for loopback. Eh = UNICOMM peripheral/target for loopback. Fh = UNICOMM peripheral/target for loopback.
11-8	CTL	R/W	0h	UNICOMM instance for loopback. This instance will be treated as the controller. 0h = UNICOMM controller for loopback. 1h = UNICOMM controller for loopback. 2h = UNICOMM controller for loopback. 3h = UNICOMM controller for loopback. 4h = UNICOMM controller for loopback. 5h = UNICOMM controller for loopback. 6h = UNICOMM controller for loopback. 7h = UNICOMM controller for loopback. 8h = UNICOMM controller for loopback. 9h = UNICOMM controller for loopback. Ah = UNICOMM controller for loopback. Bh = UNICOMM controller for loopback. Ch = UNICOMM controller for loopback. Dh = UNICOMM controller for loopback. Eh = UNICOMM controller for loopback. Fh = UNICOMM controller for loopback.
7-1	RESERVED	R	0h	
0	EN	R/W	0h	Enable 0h = Disable 1h = Enable

21.4.2.1.8 PAIR7 (Offset = 1DCh) [Reset = 00000000h]

PAIR7 is shown in Figure 21-16 and described in Table 21-18.

Return to the Summary Table.

Pairing of I2C for Multi-Controller/SMBUS applications

Figure 21-16. 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 21-18. 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 0h = UNICOMM peripheral/target for loopback. 1h = UNICOMM peripheral/target for loopback. 2h = UNICOMM peripheral/target for loopback. 3h = UNICOMM peripheral/target for loopback. 4h = UNICOMM peripheral/target for loopback. 5h = UNICOMM peripheral/target for loopback. 6h = UNICOMM peripheral/target for loopback. 7h = UNICOMM peripheral/target for loopback. 8h = UNICOMM peripheral/target for loopback. 9h = UNICOMM peripheral/target for loopback. Ah = UNICOMM peripheral/target for loopback. Bh = UNICOMM peripheral/target for loopback. Ch = UNICOMM peripheral/target for loopback. Dh = UNICOMM peripheral/target for loopback. Eh = UNICOMM peripheral/target for loopback. Fh = UNICOMM peripheral/target for loopback.
11-8	CTL	R/W	0h	UNICOMM instance for loopback. This instance will be treated as the controller. 0h = UNICOMM controller for loopback. 1h = UNICOMM controller for loopback. 2h = UNICOMM controller for loopback. 3h = UNICOMM controller for loopback. 4h = UNICOMM controller for loopback. 5h = UNICOMM controller for loopback. 6h = UNICOMM controller for loopback. 7h = UNICOMM controller for loopback. 8h = UNICOMM controller for loopback. 9h = UNICOMM controller for loopback. Ah = UNICOMM controller for loopback. Bh = UNICOMM controller for loopback. Ch = UNICOMM controller for loopback. Dh = UNICOMM controller for loopback. Eh = UNICOMM controller for loopback. Fh = UNICOMM controller for loopback.
7-1	RESERVED	R	0h	
0	EN	R/W	0h	Enable 0h = Disable 1h = Enable

Chapter 22

UNICOMM UART

This section describes the functionality of a UNICOMM module when configured to operate as an UART. 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 UART functionality on that UNICOMM instance is disabled and all register reads return a value of zero.

22.1 UART Overview 1163
22.2 UART Operation 1164
22.3 UNICOMMUART Registers 1186

22.1 UART Overview

22.1.1 Purpose of the Peripheral

The UART interface is used to communicate between a MSPM33 device and another device supporting the standard UART protocol. Additionally, select UNICOMM instances include features that support communication with the standard LIN (Local Interconnect Network), ISO7816 (Smart Card), and IrDA (Infrared Data Association) protocols, as well as hardware flow control (CTS/RTS) and multi-processor communication mode. This document outlines the functionality of each mode. The device data sheet describes which modes are available in a given UNICOMM-UART configuration.

22.1.2 Features

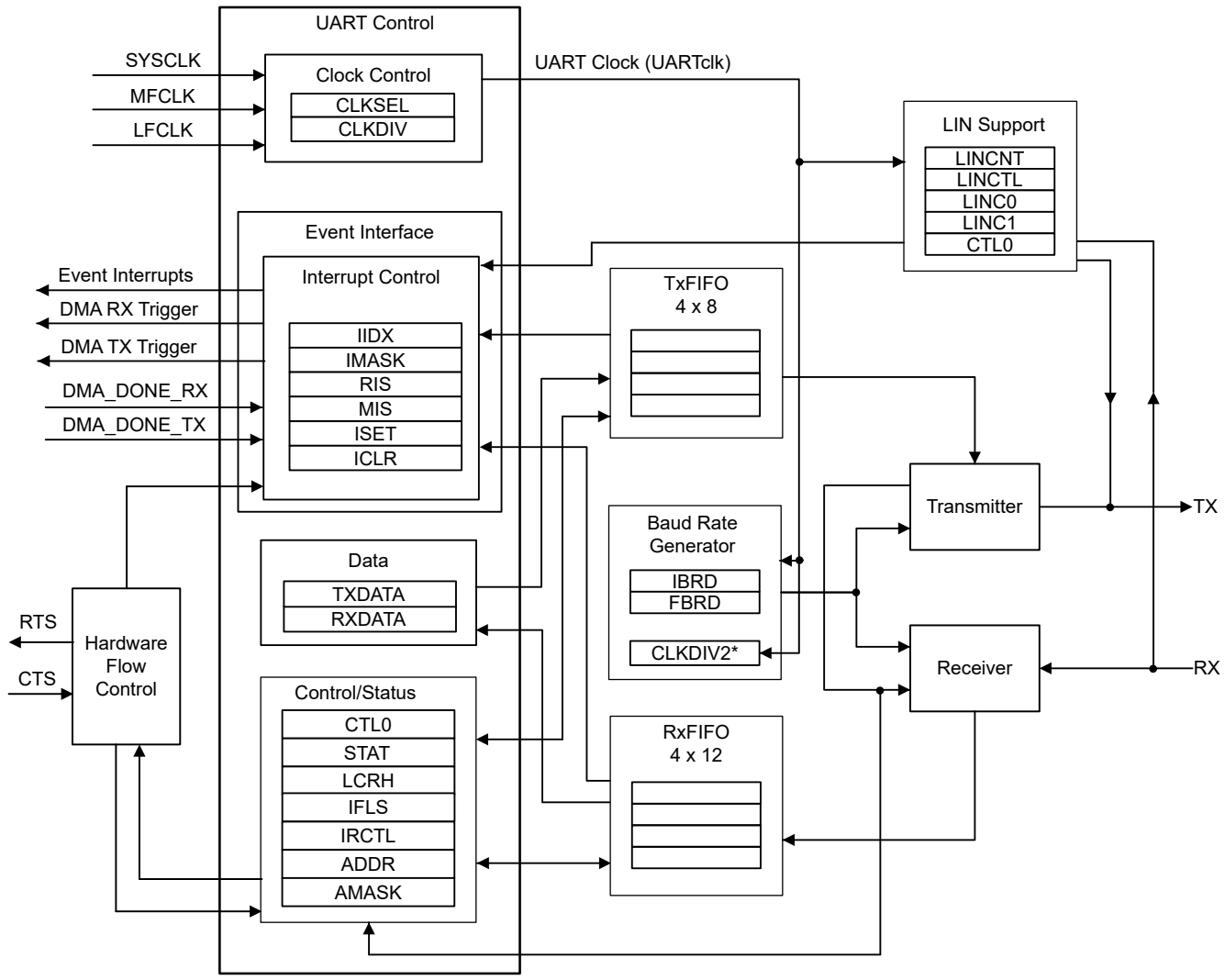
The UNICOMM-UART peripheral mode of the UNICOMM module includes the following features:

- 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 signals
  - Programmable baud-rate generation with oversampling by 16, 8 or 3
- Active in all low-power mode including stop and standby mode for instances available in power domain 0 (PD0)
- Support for waking up SYSOSC via an asynchronous fast clock request upon start bit detection when operating in low power modes
- Support for transmit to receive loopback mode operation
- Extensive protocol supported. See device data sheet for list of support protocols and features

Table 22-1. Optional UNICOMM-UART feature support

UNICOMM-UART Feature Tag	Feature description
UART-RX-TIMEOUT	Receive timeout and Line timeout
UART-IDLELINE-MULTIPROC	Idle-Line Multiprocessor
UART-FLOW-CONTROL	Flow control (CTS/RTS) with support for RS-485
UART-MULTIDROP-9-BIT	9-bit UART mode for multidrop systems with addressable peripherals
UART-EXT-DRIVER	External driver output enable
UART-SMARTCARD	ISO7816 smart card mode
UART-LIN	Local Interconnect Network (LIN)
UART-DALI-MANCHESTER	IEC62386 Digital Addressable Lighting Interface (DALI)
UART-IRDA	IrDA encoding and decoding
UART-FIFO	RX and TX FIFO
UART-DMA	Direct memory access (DMA)

22.1.3 Functional Block Diagram



*CLKDIV2 is for IrDA mode only

Figure 22-1. UART Functional Block Diagram

22.2 UART Operation

This section describes the operation of the UART peripheral.

22.2.1 Clock Control

The UART internal functional clock is selected and divided from the functional clock of the IP.

- Use 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 the CLKDIV register to select the divide ratio of the UART functional clock. The options available are divide by 1 through divide by 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 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 will be ready to start receiving and transmitting data.

22.2.2 General Architecture and Protocol

The UART protocol transmits and receives characters at a programmable bit rate, called a baud rate, shared between the two communicating devices. This is done asychronously, meaning there is no connected clock pin for this protocol. The configured baud rate is generated by each device's functional clock and is used by both the UNICOMM UART transmitter and receiver submodules.

In general, configuration registers can only be programmed when the UNICOMM UART module is disabled (the ENABLE bit in the CTL0 register is cleared). One exception is the baud-rate divisor registers (IBRD and FBRD), which can be modified without disabling the UART. If the UNICOMM UART is disabled during an ongoing transmit or receive operation, the current transaction completes before the UART halts operation.

22.2.2.1 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. Note that this functionatlity is only available in UNICOMM-UART instances that implement the UART-FLOW-CONTROL feature.

- 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.

22.2.2.2 Transmit and Receive Logic

The transmit logic performs parallel-to-serial conversion on the data written to the TXDATA register. In UNICOMM-UART instances that support the UART-FIFO feature, this data can be buffered in a 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 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.

In the UNICOMMUART Registers, the CTL0.ENABLE bit is used to enable and disable the UNICOMM-UART module, the CTL0.TXE and CTL0.RXE bits are used to enable the UART transmitter and UART receiver, the LCRH.WLEN bit is used to configure the number of data bits transmitted or received in each frame, the LCRH.PEN is used to enable parity mode, and the LCRH.STP2 is used to send two stop bits (rather than the default one bit).

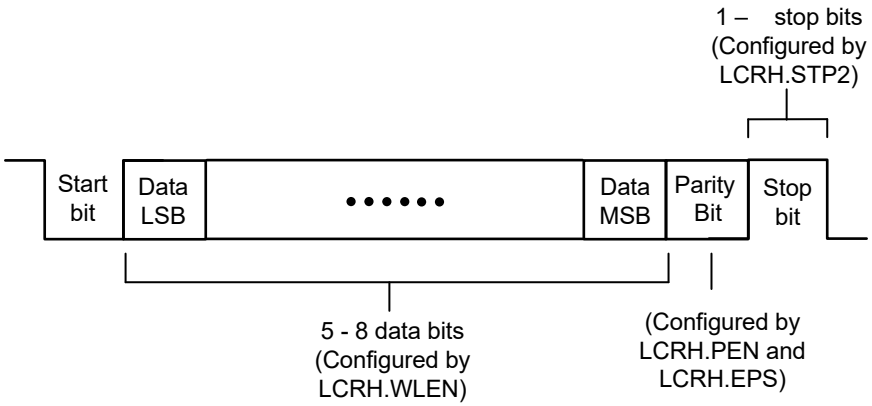


Figure 22-2. UART Character Frame

22.2.2.3 Bit Sampling

UNICOMM UART supports oversampling at 3x, 8x, and 16x the UART bus baud-rate.

- By default, CTL0.HSE is set to 0, i.e.16X oversampling is selected. 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 aforementioned scenarios assume an 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.

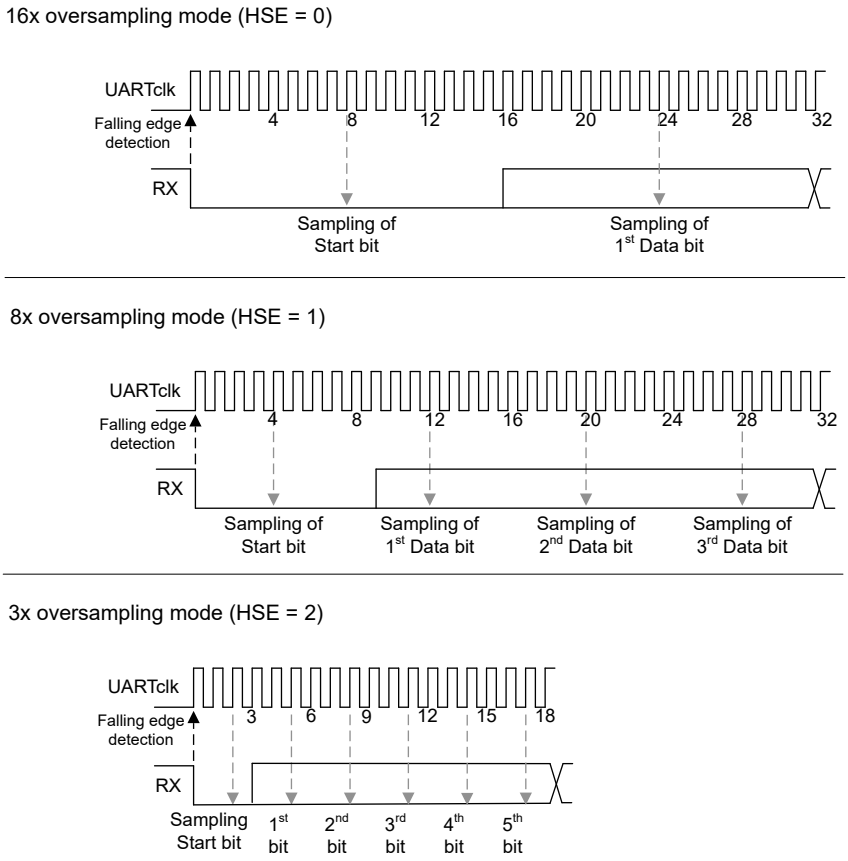


Figure 22-3. UART Oversampling mode

22.2.2.4 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 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:

$$BRD = UART\ Clock / (Oversampling \times Baud\ rate) \quad (18)$$

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 UARTx.IBRD register. The 6-bit fractional number must be loaded into the UARTx.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.

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 22-4. Baud Rate Configuration

When updating the baud-rate divisor (IBRD or IFRD), the LCRH register must also be written, so any 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 IBRD and FBRD registers are not updated until transmission or reception of the current character is complete.

22.2.2.5 Data Transmission

Data received or transmitted is stored in two FIFOs, where the receive FIFO has an extra four bits per character for status information.

Transmit data:

For transmission, data is written into the TXDATA register. If the UART is enabled, a write to TXDATA causes a data frame to start 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 as soon as data is written to the transmit FIFO (that is, if the FIFO is non-empty) and remains asserted while data is being transmitted. The BUSY bit goes low only when the transmit FIFO is empty, and the last character has been transmitted to the shift register, including the stop bits. The UART can indicate that it is busy even though the UART can no longer be enabled. STAT.BUSY also is set during the generation of a BREAK signal.

Receive data:

When the UART receiver is in an idle state and data input transitions to a low level (i.e. START condition detect), the receive counter starts running and data is sampled as per the oversampling settings in CTL0.HSE, except for the START/STOP bits. During the START bit, the receiver checks that every sampled value reads '0', and if not, the receiver state machine transitions back to the IDLE state. During the STOP bit/s, the receiver checks that every sampled value reads '1'. If not, the framing error flag (RXDATA.FRMERR) is set. When a full word is received, data is stored in the receive buffer/FIFO along with any associated error bits. The STAT.BUSY flag is set to high when a START condition is detected and set to low after a STOP bit has been received (i.e. after the entire duration of STOP bit).

22.2.2.6 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 22-2. 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 22-2. 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 a 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. Address bits are included in the parity calculation if included in the character. 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 the receive input is held low for longer than a full character transmission time, meaning the start bit, all received data, parity bit, and stop bit/bits are 0. When a break condition is detected, the BRKERR bit is set.

(1) Framing error and break condition 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.

The UART module flag status can also be checked by reading the STAT register, as shown in Table 22-3.

Table 22-3. 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. 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 the 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	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.

22.2.2.7 DMA Operation

The section applies to UNICOMM-UART configurations which support the UART-DMA feature.

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.

- 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.

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.

22.2.2.8 Internal 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

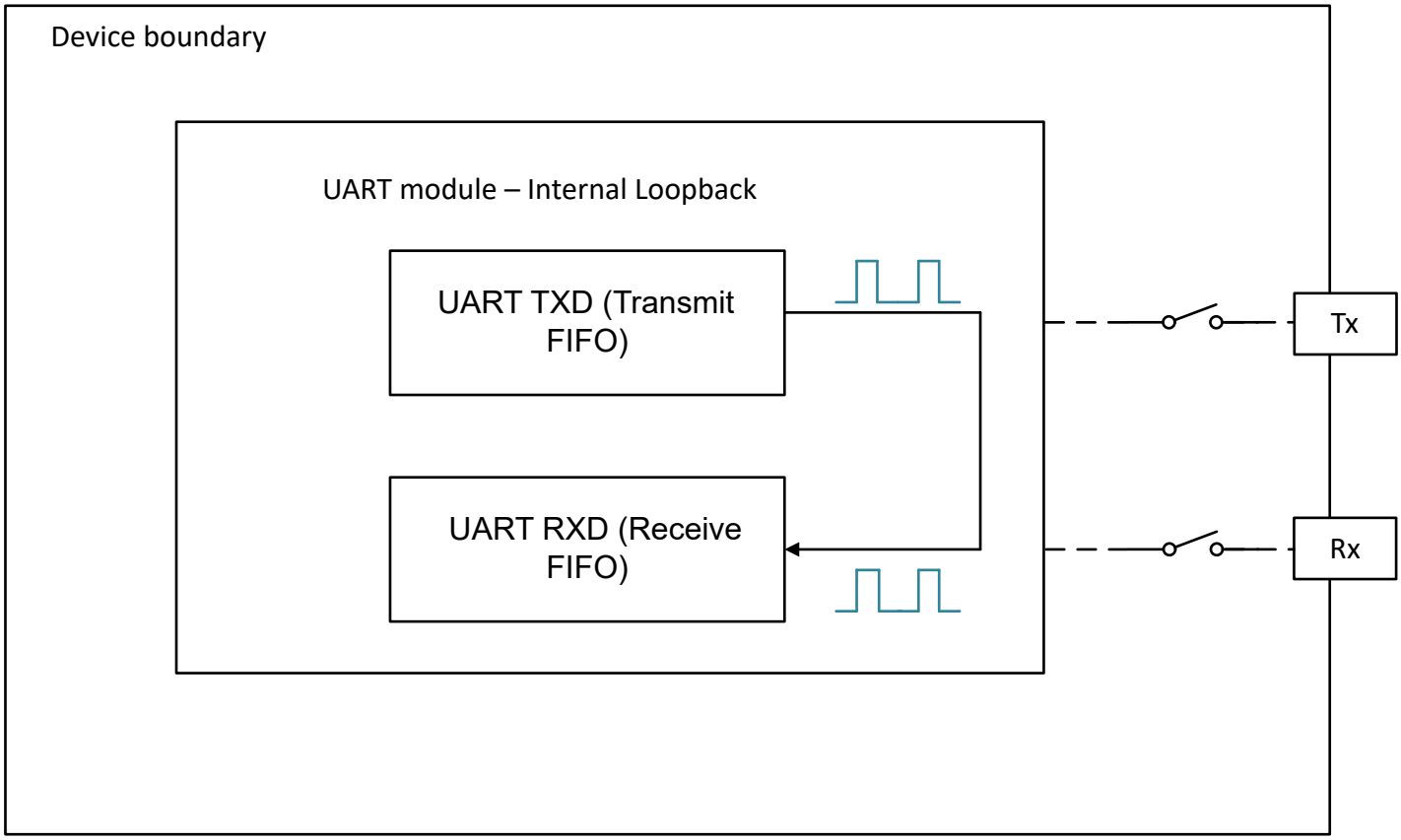


Figure 22-5. UART Loopback mode

22.2.3 Additional Protocol and Feature Support

22.2.3.1 Local Interconnect Network (LIN) Support

This section applies to UNICOMM-UART configurations which support the UART-LIN feature.

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.

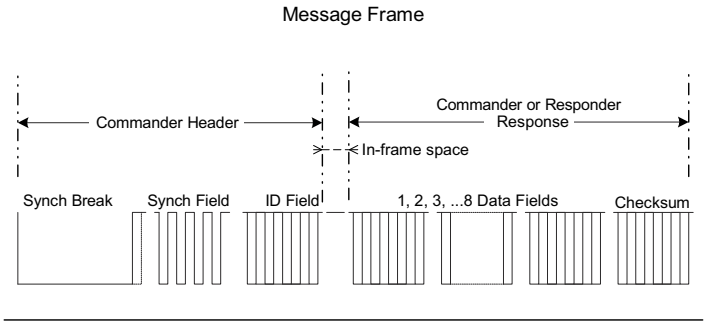


Figure 22-6. LIN Protocol Message Structure

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). This feature is also needed to achieve a LIN protocol on UNICOMM-UART.

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.

22.2.3.1.1 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.

22.2.3.1.2 LIN Responder Receive

Receiving the Break Field

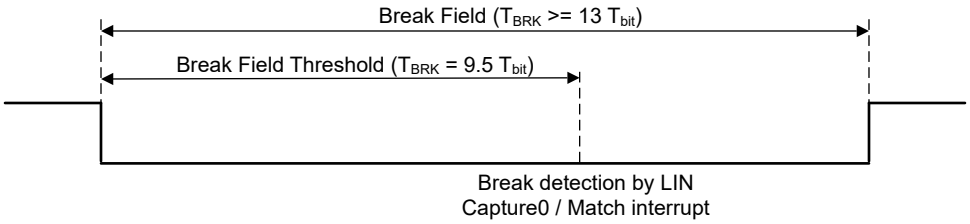


Figure 22-7. LIN Break Field Detection

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 (UARTx.LINCNT = 0)
2. Enable counter compare match mode (UARTx.LINCTL.LINC0_MATCH = 1)
3. Load UARTx.LINC0 (Counter Capture 0 register) with the counter value corresponding to 9.5 x T_{bit} (refer to the *LIN Standards and Specifications* for details on this timing).
4. Enable the LINC0 match interrupt (CPU_INT.IMASK.LINC0 = 1)
5. Setup LIN count control (configure the UARTx.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.

Receiving the Sync Field

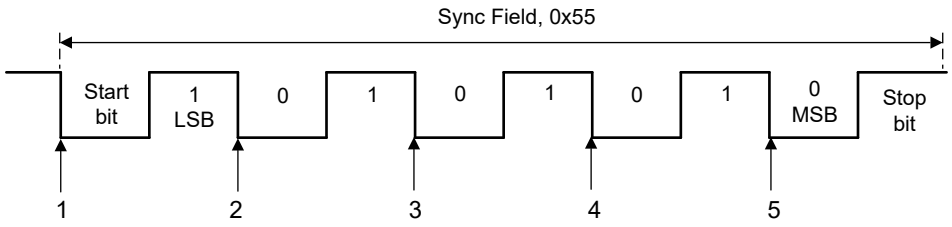


Figure 22-8. LIN Sync Field Detection

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 (see Figure 22-8). After software detects a valid BREAK field, the counter can be set to measure the SYNC field. Figure 22-8 shows the SYNC byte format. 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 (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)

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.

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.

22.2.3.1.3 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 tranmission may 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 transmittion.

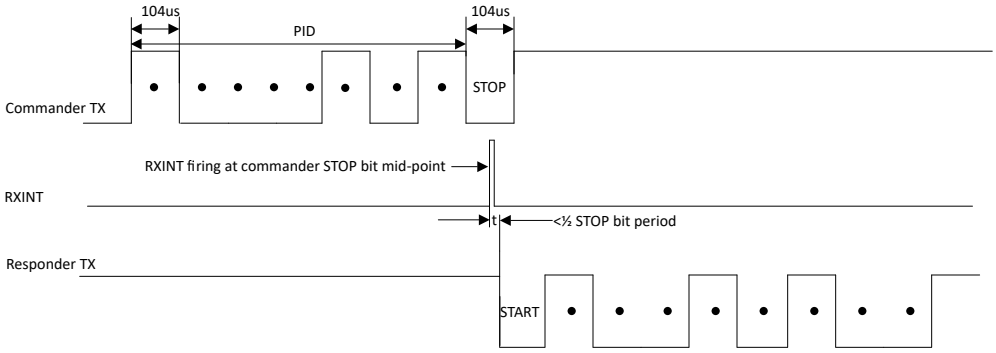


Figure 22-9. LIN Responder Transmission Delay

22.2.3.2 Flow Control

This section applies to UNICOMM-UART configurations which support the UART-FLOW-CONTROL feature.

In UART mode (CTL0.MODE set to 0), hardware flow control between two devices is accomplished 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, the CTS input expects a low signal on a send request as shown in Figure 22-10.

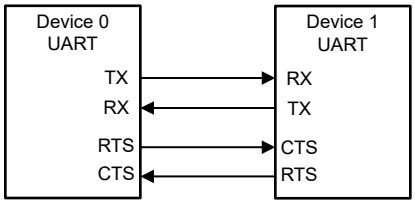


Figure 22-10. Flow Control

The CTS input controls the transmitter. The transmitter may only transmit data when the CTS input is asserted low. When RTS flow control is enabled, the 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 as shown in Table 22-4.

Table 22-4. 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

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 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.

22.2.3.3 RS485 Support

This section applies to UNICOMM-UART configurations which support the UART-EXT-DRIVER feature.

RS485 is a standard used in serial communications systems. This standard can be used effectively over long distances and in electrically noisy environments. Multiple receivers can be connected to such a network. These characteristics make RS-485 useful in industrial control systems and similar applications.

With the RS485 direction signal an external RS485 PHY can be controlled. The RTS I/O is used in this mode for the direction signal. The signal is set automatically to high once a data transmit is started. It will stay set between bytes if they are sent back to back. If a data receive is ongoing a new transmit should be delayed till this data has been received and the direction signal has been set to transmit.

Data exchange sequence as show in Figure 22-11:

- 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

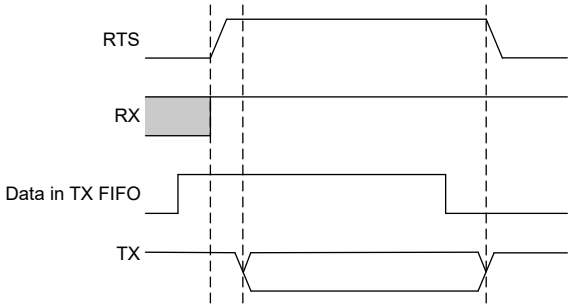


Figure 22-11. RS485 Data Exchange

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 functional clocks between RTS assertion and beginning of START bit
- EXTDIR_HOLD bits define number of functional clocks between STOP (entire bit duration) and RTS de-assertion

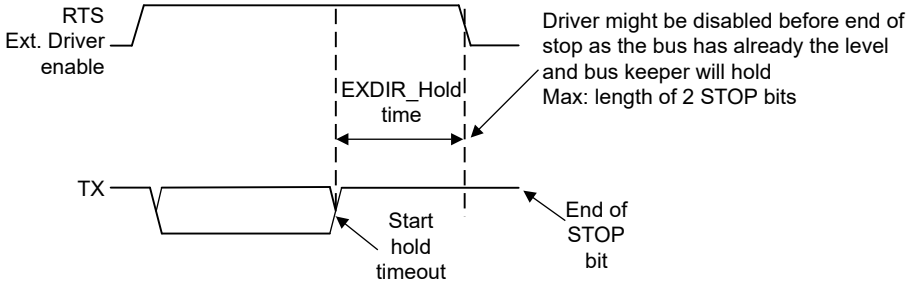


Figure 22-12. RS485 external driver control

22.2.3.4 FIFO Operation

This section applies to UNICOMM-UART configurations which support the UART-FIFO feature. Configurations which do not support this feature still interface with the TXDATA and RXDATA registers, but do not contain any FIFO features.

The UART has two FIFOs with a depth of 4 entries, one for transmit and one for receive. The FIFOs are accessed through the UART Data (TXDATA/RXDATA) registers. Read operations of the RXDATA register return a 12-bit value consisting of 8 data bits and 4 error flags. Write operations to TXDATA place 8-bit data in the transmit FIFO.

Out of reset, both FIFOs are disabled and act as 1-byte-deep holding registers. The FIFOs are enabled by setting the FEN bit in UARTx.CTL0. FIFO status can be monitored through the UARTx.STAT register and the interrupt events.

Hardware monitors empty, full and overrun conditions

The UARTx.STAT register contains empty and full flags (TXFE, TXFF, RXFE, and RXFF bits), and 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. A write is just lost, in case it overruns the transmit FIFO. If the FIFOs are disabled, the empty and full flags are set according to the status of the 1-byte-deep holding registers. When receiving more data than the FIFO can capture the oldest data will be overwritten with the received data.

The trigger point at which the FIFOs generate interrupts is controlled through the UARTx.IFLS register. Both FIFOs can be individually configured to trigger interrupts at different levels. Available configurations for transmit FIFO include 3/4, 1/2, 1/4 and empty, for receive FIFO 1/4, 1/2, 3/4 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. Out of reset, both FIFOs are configured to trigger an interrupt at the 1/2 mark. The FIFO integrity is not guaranteed under the following conditions:

- After a UART Send Break has been initiated (LCRH.BRK = 1)
- If the software disables the UART in the middle of a transmission with data in the FIFO, and then re-enables it.

22.2.3.5 Idle-Line Multiprocessor

This section applies to UNICOMM-UART configurations which support the UART-IDLELINE-MULTIPROC feature.

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 22-13). 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.

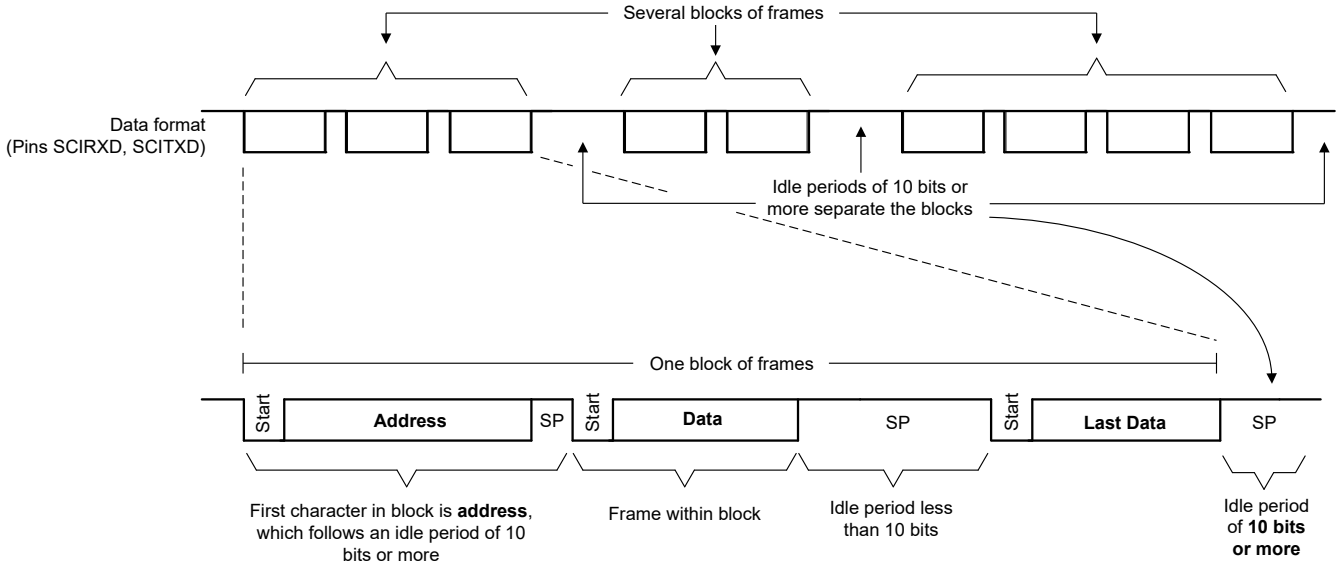


Figure 22-13. Idle-Line Multiprocessor

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.

22.2.3.6 9-Bit UART Mode

This sections applies to UNICOMM-UART configurations that support the UART-MULTIDROP-9-BIT feature.

9-bit mode is enabled by setting MODE bit to ADDR9BIT in the CTL0 register. This feature is useful in a multi-drop configuration of the UART where a single controller connected to multiple peripherals can communicate with a particular peripheral through its address or 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 bit.

Receive Transaction:

In 9-bit mode, a 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 UARTx.ADDR register:

- If the address matches, a 9-bit mode address match interrupt (ADDR_MATCH) is generated, if enabled and further data get received.
- If the address does not match, the address byte and the subsequent data bytes are not recieved.

The address can be predefined 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. By default, the UART.AMASK is 0xFF, meaning that only the specified address must match

Transmit Transaction:

All the send transactions 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 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. For data byte transmissions, 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 with EPS bit cleared.

9^{th} bit handling:

Table 22-5. 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)

22.2.3.7 DALI Protocol

This section applies to UNICOMM-UART configurations which support the UART-DALI-MANCHESTER feature.

DALI stands for Digital Addressable Lighting Interface. It is an International Standard (IEC 62386) lighting control system, providing a single interface for all electronic control gear (light sources) and electronic control devices (lighting controllers).

The UART module supports the low level DALI Protocol sending and receiving the bit streams for 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 user needs to ensure to write second byte to buffer before first byte has been shifted out. The hardware will then send the two bytes without inserting the stop bits in-between. Otherwise the stop bits will be sent and the data will be handled like a backward frame.

Receiving data:

The UART module in DALI mode will check the 9th bit after the start bit to detect a Forward frame or backward frame. If this bit does have a change of the phase (= no stop bit) a forward frame is detected and:

- Address compare is done in software or hardware depending on MASK
- Address and data are always transferred into the RX Buffer

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.

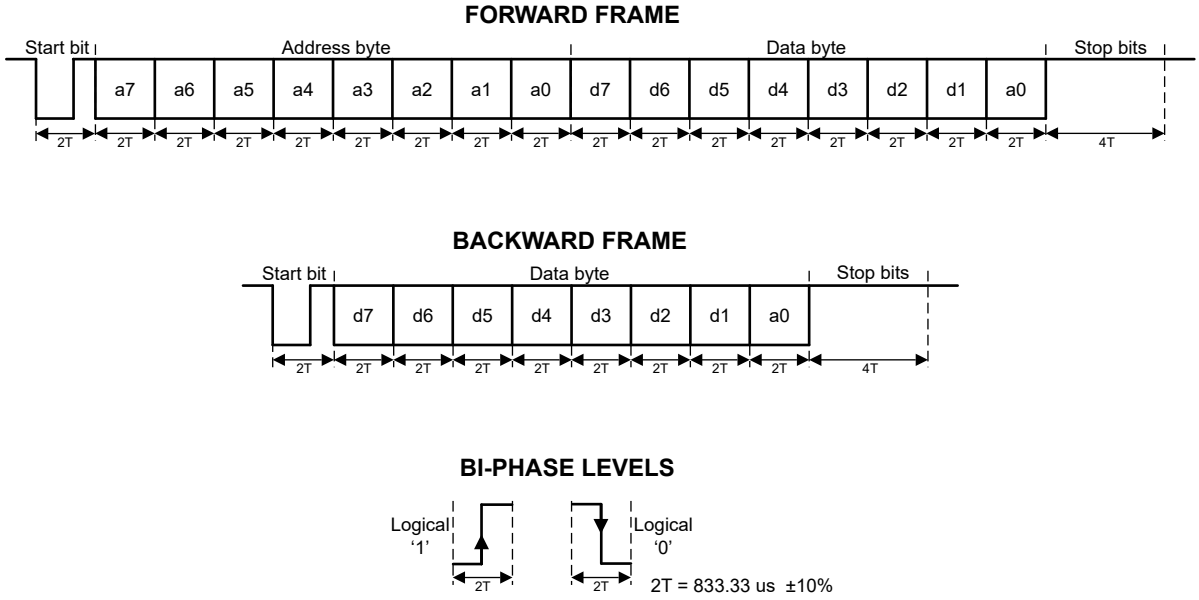


Figure 22-14. DALI Protocol

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

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.

22.2.3.8 Manchester Encoding and Decoding

This section applies to UNICOMM-UART configurations which support the UART-DALI-MANCHESTER feature.

UART provides option to receive and transmit Manchester encoded data. The function is enabled by the MENC bit in CTL0 register to generate the IEEE 802.3 compliant waveform. 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.

22.2.3.9 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 22-15). The pulse duration is defined by IRTXPL bits specifying the number of one-half clock periods of the clock selected by IRTXCLK bit.

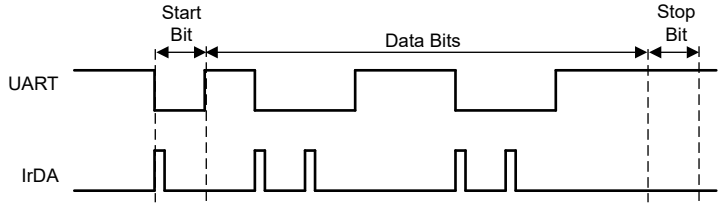


Figure 22-15. IrDA Protocol

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.)

22.2.3.10 ISO7816 Smart Card Support

The UART offers basic support to allow communication with an ISO7816 smart card. When configuring the MODE bits to smart card (0x4) of the CTL0 register, 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 oversampling of 16 is used; LCRH.STP2 & CTL0.HSE bits are ignored.

If a parity error is detected during a transmission, RXD is pulled low during the second stop bit. In this case, the UART aborts the transmission, it flushes the transmit FIFO and discards any data it contains. Additionally it 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 will drive the line low and a parity interrupt flag is asserted. The transmitter responds based on the value of this bit.

22.2.3.11 Address Detection

This section applies to UNICOMM-UART configurations which support any of the following features: UART-DALI, UART-MULTIDROP-9-BIT, UART-IDLELINE-MULTIPROC.

The ADDR register is used to set the specific address that should be matched with receiving 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 22-6. 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
Address mismatch	Address and Data get dropped	Address and Data get dropped	Address and Data get dropped

22.2.3.12 Glitch Suppression

Digital filter

Digital filter is based on the UART functional clock. The DGFSEL bits in the GFCTL register can be programmed to provide glitch suppression on the RX line and assure proper signal values. The glitch suppression value is in terms of functional clocks. All signals are delayed internally when glitch suppression is nonzero. For example, if DGFSEL is set to 0x5, 5 clocks should be added onto the calculation for the expected transaction time. The DGFSEL need to be configured for the glitch suppression pulse width to be shorter than 1/3 of a normal data pulse, to avoid a normal pulse is filtered unexpectedly.

Analog filter

The analog glitch suppression on the RX line is based on the analog glitch filter and it can be selected with the AGFSEL bits in the UARTx.GFCTL register. See data sheet for the select-able glitch filter values. The analog glitch filter is enabled with the AGFEN, if not set the input signals will be passed through to the UART module without filtering.

22.2.4 Low Power Operation

(More information will be added in next revision)

22.2.5 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 UART interface external pull up or down resistors might be required.

22.2.6 UART Initialization

To enable and initialize the UART, perform the following initialization steps:

1. Configure the module to UNICOMM-UART mode in the IPMODE.SELECT bits at the UNICOMM top level
2. Configure RX and TX pin functions by using the IOMUX registers.
3. Reset the peripheral using RSTCTL register at the UNICOMM top level
4. Enable the power to UART peripheral using the PWREN register top level
5. Select the UART function clock source and divide options using CLKSEL and CLKDIV registers.
6. Disable the UART by clearing the CTL0.ENABLE bit.
7. Use the baud-rate equation in Section 22.2.2.4 to calculate the IBRD and FBRD registers.
8. Write the integer portion of the BRD to the IBRD register.
9. Write the fractional portion of the BRD to the FBRD register.
10. Write the desired communication mode settings to the CTL0 register (all CTL0 fields other than ENABLE).
11. Write the desired serial parameter configurations to the LCRH register.
12. Configure the desired FIFO trigger levels in the IFLS register.
13. Enable desired interrupts and/or DMA event by using CPU_INT, DMA_TRIG_RX, DMA_TRIG_TX group IMASK registers.
14. 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.

22.2.7 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 22-7.

Table 22-7. 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
DMA trigger	Publisher	UART	DMA	DMA event route	DMA_TRIG_TX registers	Fixed interrupt route from UART to DMA

22.2.7.1 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 22-8. 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 field. More information provided below.
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. More information provided below.
0x0C	TXINT	UART transmit interrupt. More information provided below.
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.
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 UARTx.IFLS.RXTOSEL bits.

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.

The receive timeout interrupt is asserted when the receive FIFO is not empty, and no further data is received specified time in the UARTx.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:

- 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.

22.2.7.2 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 configured in a flexible way to trigger the DMA for receive or transmit events with the trigger conditions in Table 22-9and Table 22-10.

DMA_TRIG_RX line is used for triggering the DMA to do a receive data transfer and DMA_TRIG_TX line is used for triggering the DMA to do a transmit data transfer.

Table 22-9. 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.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, 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 22-10. 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 8.2.5 for guidance on configuring the Event registers and Section 8.1.3.2 for on how DMA trigger event works.

22.2.8 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 22-11. Debug Mode Peripheral Behavior

PDBGCTL.FREE	PDBGCTL.SOFT	Function
1	**x**	Modules continues operation

Table 22-11. Debug Mode Peripheral Behavior (continued)

PDBGCTL.FREE	PDBGCTL.SOFT	Function
0	**0**	Module stops immediately
0	**1**	Module stops after the next transfer has been finished

22.3 UNICOMMUART Registers

Table 22-12 lists the memory-mapped registers for the UNICOMMUART registers. All register offset addresses not listed in Table 22-12 should be considered as reserved locations and the register contents should not be modified.

Table 22-12. UNICOMMUART Registers

Offset	Acronym	Register Name	Section
1000h	CLKDIV	Clock Divider	Section 22.3.1
1008h	CLKSEL	Clock Select for Ultra Low Power peripherals	Section 22.3.2
1020h	IIDX	Interrupt index	Section 22.3.3
1028h	IMASK	Interrupt mask	Section 22.3.4
1030h	RIS	Raw interrupt status	Section 22.3.5
1038h	MIS	Masked interrupt status	Section 22.3.6
1040h	ISET	Interrupt set	Section 22.3.7
1048h	ICLR	Interrupt clear	Section 22.3.8
1058h	IMASK	Interrupt mask	Section 22.3.9
1060h	RIS	Raw interrupt status	Section 22.3.10
1068h	MIS	Masked interrupt status	Section 22.3.11
1070h	ISET	Interrupt set	Section 22.3.12
1088h	IMASK	Interrupt mask	Section 22.3.13
1090h	RIS	Raw interrupt status	Section 22.3.14
1098h	MIS	Masked interrupt status	Section 22.3.15
10A0h	ISET	Interrupt set	Section 22.3.16
10E4h	INTCTL	Interrupt control register	Section 22.3.17
1100h	CTL0	UART Control Register 0	Section 22.3.18
1104h	LCRH	UART Line Control Register	Section 22.3.19
1108h	STAT	UART Status Register	Section 22.3.20
110Ch	IFLS	UART Interrupt FIFO Level Select Register	Section 22.3.21
1110h	IBRD	UART Integer Baud-Rate Divisor Register	Section 22.3.22
1114h	FBRD	UART Fractional Baud-Rate Divisor Register	Section 22.3.23
1118h	GFCTL	Glitch Filter Control	Section 22.3.24
1120h	TXDATA	UART Transmit Data Register	Section 22.3.25
1124h	RXDATA	UART Receive Data Register	Section 22.3.26
1130h	LINCNT	UART LIN Mode Counter Register	Section 22.3.27
1134h	LINCTL	UART LIN Mode Control Register	Section 22.3.28
1138h	LINC0	UART LIN Mode Capture 0 Register	Section 22.3.29
113Ch	LINC1	UART LIN Mode Capture 1 Register	Section 22.3.30
1140h	IRCTL	eUSCI_Ax IrDA Control Word Register	Section 22.3.31
1148h	AMASK	Self Address Mask Register	Section 22.3.32
114Ch	ADDR	Self Address Register	Section 22.3.33

Complex bit access types are encoded to fit into small table cells. Table 22-13 shows the codes that are used for access types in this section.

Table 22-13. UNICOMMUART Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read

Table 22-13. UNICOMMUART Access Type Codes (continued)

Access Type	Code	Description
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value

22.3.1 CLKDIV Register (Offset = 1000h) [Reset = 00000000h]

CLKDIV is shown in Table 22-14.

Return to the Summary Table.

This register is used to specify module-specific divide ratio of the functional clock

Table 22-14. CLKDIV Register 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 Division factor 0 : DIV_BY_1 1 : DIV_BY_2 .... 63: DIV_BY_64 0h = Do not divide clock source 1h = Divide clock source by 2 2h = Divide clock source by 3 3h = Divide clock source by 4 4h = Divide clock source by 5 5h = Divide clock source by 6 6h = Divide clock source by 7 7h = Divide clock source by 8 3Ch = Divide clock source by 61 3Dh = Divide clock source by 62 3Eh = Divide clock source by 63 3Fh = Divide clock source by 64

22.3.2 CLKSEL Register (Offset = 1008h) [Reset = 00000000h]

CLKSEL is shown in Table 22-15.

Return to the Summary Table.

Clock source selection for peripherals

Table 22-15. CLKSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11	ASYNC_PLL_SEL	R/W	0h	Asynchronous PLL selected as source 0h = Does not select this clock as a source 1h = Select this clock as a source
10	ASYNC_HFCLK_SEL	R/W	0h	Asynchronous HFCLK selected as source 0h = Does not select this clock as a source 1h = Select this clock as a source
9	ASYNC_SYSCLK_SEL	R/W	0h	Asynchronous sysclk selected as source 0h = Does not select this clock as a source 1h = Select this clock as a source
8	ASYNC_LFCLK_SEL	R/W	0h	Asynchronous lfclk selected as source 0h = Does not select this clock as a source 1h = Select this clock as a source
7-4	RESERVED	R/W	0h	
3	BUSCLK_SEL	R/W	0h	Selects BUS CLK as clock source if enabled 0h = Does not select this clock as a source 1h = Select this clock as a source
2	MFCLK_SEL	R/W	0h	Selects MFCLK as clock source if enabled 0h = Does not select this clock as a source 1h = Select this clock as a source
1	LFCLK_SEL	R/W	0h	Selects LFCLK as clock source if enabled 0h = Does not select this clock as a source 1h = Select this clock as a source
0	RESERVED	R/W	0h	

22.3.3 IIDX Register (Offset = 1020h) [Reset = 00000000h]

IIDX is shown in Table 22-16.

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.

Table 22-16. IIDX Register 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 highes priority interrupt index. A read clears the corresponding interrupt flag in RIS and MIS registers. 15h-1Fh = Reserved 00h = No interrupt pending 01h = UART receive time-out interrupt; Interrupt Flag: RT; Interrupt Priority: Highest 02h = UART framing error interrupt; Interrupt Flag: FE 03h = UART parity error interrupt; Interrupt Flag: PE 04h = UART break error interrupt; Interrupt Flag: BE 05h = UART receive overrun error interrupt; Interrupt Flag: OE 06h = Negative edge on UARTxRXD interrupt; Interrupt Flag: RXNE 07h = Positive edge on UARTxRXD interrupt; Interrupt Flag: RXPE 08h = LIN capture 0 / match interrupt; Interrupt Flag: LINC0 09h = LIN capture 1 interrupt; Interrupt Flag: LINC1 0Ah = LIN hardware counter overflow interrupt; Interrupt Flag: LINOVF 0Bh = UART receive interrupt; Interrupt Flag: RX 0Ch = UART transmit interrupt; Interrupt Flag: TX 0Dh = UART end of transmission interrupt (transmit serializer empty); Interrupt Flag: EOT 0Eh = 9-bit mode address match interrupt; Interrupt Flag: ADDR_MATCH Fh = UART Clear to Send Modem interrupt; Interrupt Flag: CTS 10h = DMA DONE on RX 11h = DMA DONE on TX 12h = Noise Error Event 13h = DMA PRE IRQ INTERRUPT 14h = DMA PRE IRQ INTERRUPT 15h = UART line time-out interrupt; Interrupt Flag

22.3.4 IMASK Register (Offset = 1028h) [Reset = 00000000h]

IMASK is shown in Table 22-17.

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.

Table 22-17. IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	RESERVED	R/W	0h	
20	LTOUT	R/W	0h	Enable UARTOUT Line Time-Out Interrupt. 0h = Clear Interrupt Mask 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 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
16	DMA_DONE_TX	R/W	0h	Enable DMA Done on TX Event Channel Interrupt 0h = Interrupt disabled 1h = Set Interrupt Mask
15	DMA_DONE_RX	R/W	0h	Enable DMA Done on RX Event Channel Interrupt 0h = Interrupt disabled 1h = Set Interrupt Mask
14	CTS	R/W	0h	Enable UART Clear to Send Modem Interrupt. 0h = Interrupt disabled 1h = Set Interrupt Mask
13	ADDR_MATCH	R/W	0h	Enable Address Match Interrupt. 0h = Clear Interrupt Mask 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. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
11	TXINT	R/W	0h	Enable UART Transmit Interrupt. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
10	RXINT	R/W	0h	Enable UART Receive Interrupt. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
9	LINOVF	R/W	0h	Enable LIN Hardware Counter Overflow Interrupt. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
8	LINC1	R/W	0h	Enable LIN Capture 1 Interrupt. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
7	LINC0	R/W	0h	Enable LIN Capture 0 / Match Interrupt . 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
6	RXPE	R/W	0h	Enable Positive Edge on UARTxRXD Interrupt. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
5	RXNE	R/W	0h	Enable Negative Edge on UARTxRXD Interrupt. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask

Table 22-17. IMASK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	OVRERR	R/W	0h	Enable UART Receive Overrun Error Interrupt. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
3	BRKERR	R/W	0h	Enable UART Break Error Interrupt. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
2	PARERR	R/W	0h	Enable UART Parity Error Interrupt. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
1	FRMERR	R/W	0h	Enable UART Framing Error Interrupt. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
0	RTOUT	R/W	0h	Enable UARTOUT Receive Time-Out Interrupt. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask

22.3.5 RIS Register (Offset = 1030h) [Reset = 00000000h]

RIS is shown in Table 22-18.

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.

Table 22-18. RIS Register 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 occured 0h = Interrupt did not occur 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 0h = Interrupt did not occur 1h = Interrupt occurred
16	DMA_DONE_TX	R	0h	DMA Done on TX Event Channel Interrupt 0h = Interrupt disabled 1h = Interrupt occurred
15	DMA_DONE_RX	R	0h	DMA Done on RX Event Channel Interrupt 0h = Interrupt disabled 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'. 0h = Interrupt disabled 1h = Interrupt occurred
13	ADDR_MATCH	R	0h	Address Match Interrupt. 0h = Interrupt did not occur 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. 0h = Interrupt did not occur 1h = Interrupt occurred
11	TXINT	R	0h	UART Transmit Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
10	RXINT	R	0h	UART Receive Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
9	LINOVF	R	0h	LIN Hardware Counter Overflow Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
8	LINC1	R	0h	LIN Capture 1 Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
7	LINC0	R	0h	LIN Capture 0 / Match Interrupt . 0h = Interrupt did not occur 1h = Interrupt occurred

Table 22-18. RIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	RXPE	R	0h	Positive Edge on UARTxRXD Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
5	RXNE	R	0h	Negative Edge on UARTxRXD Interrupt. 0h = Interrupt did not occur 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 0h = Interrupt did not occur 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' 0h = Interrupt did not occur 1h = Interrupt occurred
2	PARERR	R	0h	UART Parity Error Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
1	FRMERR	R	0h	UART Framing Error Interrupt. Framing error interrupt is raised when a "low" stop bit is detected. 0h = Interrupt did not occur 1h = Interrupt occurred
0	RTOUT	R	0h	UARTOUT Receive Time-Out Interrupt. Interupt is raised when a time-out has lapsed and there are still items in the FIFO 0h = Interrupt did not occur 1h = Interrupt occurred

22.3.6 MIS Register (Offset = 1038h) [Reset = 00000000h]

MIS is shown in Table 22-19.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Table 22-19. MIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	RESERVED	R	0h	
20	LTOUT	R	0h	Masked UARTOUT Line Time-Out Interrupt. 0h = Interrupt did not occur 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 0h = Interrupt did not occur 1h = Interrupt occurred
16	DMA_DONE_TX	R	0h	Masked DMA Done on TX Event Channel Interrupt 0h = Interrupt did not occur 1h = Interrupt occurred
15	DMA_DONE_RX	R	0h	Masked DMA Done on RX Event Channel Interrupt 0h = Interrupt did not occur 1h = Interrupt occurred
14	CTS	R	0h	Masked UART Clear to Send Modem Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
13	ADDR_MATCH	R	0h	Masked Address Match Interrupt. 0h = Interrupt did not occur 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. 0h = Interrupt did not occur 1h = Interrupt occurred
11	TXINT	R	0h	Masked UART Transmit Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
10	RXINT	R	0h	Masked UART Receive Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
9	LINOVF	R	0h	Masked LIN Hardware Counter Overflow Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
8	LINC1	R	0h	Masked LIN Capture 1 Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
7	LINC0	R	0h	Masked LIN Capture 0 / Match Interrupt . 0h = Interrupt did not occur 1h = Interrupt occurred
6	RXPE	R	0h	Masked Positive Edge on UARTxRXD Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
5	RXNE	R	0h	Masked Negative Edge on UARTxRXD Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred

Table 22-19. MIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	OVRERR	R	0h	Masked UART Receive Overrun Error Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
3	BRKERR	R	0h	Masked UART Break Error Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
2	PARERR	R	0h	Masked UART Parity Error Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
1	FRMERR	R	0h	Masked UART Framing Error Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
0	RTOUT	R	0h	Masked UARTOUT Receive Time-Out Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred

22.3.7 ISET Register (Offset = 1040h) [Reset = 00000000h]

ISET is shown in Table 22-20.

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.

Table 22-20. ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	RESERVED	W	0h	
20	LTOUT	W	0h	Set UARTOUT Line Time-Out Interrupt. 0h = Writing 0 has no effect 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 0h = Writing this has no effect 1h = Set the interrupt
16	DMA_DONE_TX	W	0h	Set DMA Done on TX Event Channel Interrupt 0h = Interrupt disabled 1h = Set Interrupt
15	DMA_DONE_RX	W	0h	Set DMA Done on RX Event Channel Interrupt 0h = Interrupt disabled 1h = Set Interrupt
14	CTS	W	0h	Set UART Clear to Send Modem Interrupt. 0h = Writing 0 has no effect 1h = Set Interrupt
13	ADDR_MATCH	W	0h	Set Address Match Interrupt. 0h = Writing 0 has no effect 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. 0h = Writing 0 has no effect 1h = Set Interrupt
11	TXINT	W	0h	Set UART Transmit Interrupt. 0h = Writing 0 has no effect 1h = Set Interrupt
10	RXINT	W	0h	Set UART Receive Interrupt. 0h = Writing 0 has no effect 1h = Set Interrupt
9	LINOVF	W	0h	Set LIN Hardware Counter Overflow Interrupt. 0h = Writing 0 has no effect 1h = Set Interrupt
8	LINC1	W	0h	Set LIN Capture 1 Interrupt. 0h = Writing 0 has no effect 1h = Set Interrupt
7	LINC0	W	0h	Set LIN Capture 0 / Match Interrupt . 0h = Writing 0 has no effect 1h = Set Interrupt
6	RXPE	W	0h	Set Positive Edge on UARTxRXD Interrupt. 0h = Writing 0 has no effect 1h = Set Interrupt
5	RXNE	W	0h	Set Negative Edge on UARTxRXD Interrupt. 0h = Writing 0 has no effect 1h = Set Interrupt

Table 22-20. ISET Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	OVRERR	W	0h	Set UART Receive Overrun Error Interrupt. 0h = Writing 0 has no effect 1h = Set Interrupt
3	BRKERR	W	0h	Set UART Break Error Interrupt. 0h = Writing 0 has no effect 1h = Set Interrupt
2	PARERR	W	0h	Set UART Parity Error Interrupt. 0h = Writing 0 has no effect 1h = Set Interrupt
1	FRMERR	W	0h	Set UART Framing Error Interrupt. 0h = Writing 0 has no effect 1h = Set Interrupt
0	RTOUT	W	0h	Set UARTOUT Receive Time-Out Interrupt. 0h = Writing 0 has no effect 1h = Set Interrupt

22.3.8 ICLR Register (Offset = 1048h) [Reset = 00000000h]

ICLR is shown in Table 22-21.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

Table 22-21. ICLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	RESERVED	W	0h	
20	LTOUT	W	0h	Clear UARTOUT Line Time-Out Interrupt. 0h = Writing 0 has no effect 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 0h = Writing 0 has no effect 1h = Clear Interrupt
16	DMA_DONE_TX	W	0h	Clear DMA Done on TX Event Channel Interrupt 0h = Interrupt disabled 1h = Clear Interrupt
15	DMA_DONE_RX	W	0h	Clear DMA Done on RX Event Channel Interrupt 0h = Interrupt disabled 1h = Clear Interrupt
14	CTS	W	0h	Clear UART Clear to Send Modem Interrupt. 0h = Writing 0 has no effect 1h = Clear Interrupt
13	ADDR_MATCH	W	0h	Clear Address Match Interrupt. 0h = Writing 0 has no effect 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. 0h = Writing 0 has no effect 1h = Clear Interrupt
11	TXINT	W	0h	Clear UART Transmit Interrupt. 0h = Writing 0 has no effect 1h = Clear Interrupt
10	RXINT	W	0h	Clear UART Receive Interrupt. 0h = Writing 0 has no effect 1h = Clear Interrupt
9	LINOVF	W	0h	Clear LIN Hardware Counter Overflow Interrupt. 0h = Writing 0 has no effect 1h = Clear Interrupt
8	LINC1	W	0h	Clear LIN Capture 1 Interrupt. 0h = Writing 0 has no effect 1h = Clear Interrupt
7	LINC0	W	0h	Clear LIN Capture 0 / Match Interrupt . 0h = Writing 0 has no effect 1h = Clear Interrupt
6	RXPE	W	0h	Clear Positive Edge on UARTxRXD Interrupt. 0h = Writing 0 has no effect 1h = Clear Interrupt
5	RXNE	W	0h	Clear Negative Edge on UARTxRXD Interrupt. 0h = Writing 0 has no effect 1h = Clear Interrupt

Table 22-21. ICLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	OVRERR	W	0h	Clear UART Receive Overrun Error Interrupt. 0h = Writing 0 has no effect 1h = Clear Interrupt
3	BRKERR	W	0h	Clear UART Break Error Interrupt. 0h = Writing 0 has no effect 1h = Clear Interrupt
2	PARERR	W	0h	Clear UART Parity Error Interrupt. 0h = Writing 0 has no effect 1h = Clear Interrupt
1	FRMERR	W	0h	Clear UART Framing Error Interrupt. 0h = Writing 0 has no effect 1h = Clear Interrupt
0	RTOUT	W	0h	Clear UARTOUT Receive Time-Out Interrupt. 0h = Writing 0 has no effect 1h = Clear Interrupt

22.3.9 IMASK Register (Offset = 1058h) [Reset = 00000000h]

IMASK is shown in Table 22-22.

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 22-22. IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R/W	0h	
10	RXINT	R/W	0h	Enable UART Receive Interrupt. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
9-1	RESERVED	R/W	0h	
0	RTOUT	R/W	0h	Enable UARTOUT Receive Time-Out Interrupt. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask

22.3.10 RIS Register (Offset = 1060h) [Reset = 00000000h]

RIS is shown in Table 22-23.

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.

Table 22-23. RIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R	0h	
10	RXINT	R	0h	UART Receive Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
9-1	RESERVED	R	0h	
0	RTOUT	R	0h	UARTOUT Receive Time-Out Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred

22.3.11 MIS Register (Offset = 1068h) [Reset = 00000000h]

MIS is shown in Table 22-24.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Table 22-24. MIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R	0h	
10	RXINT	R	0h	Masked UART Receive Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
9-1	RESERVED	R	0h	
0	RTOUT	R	0h	Masked UARTOUT Receive Time-Out Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred

22.3.12 ISET Register (Offset = 1070h) [Reset = 00000000h]

ISET is shown in Table 22-25.

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.

Table 22-25. ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	W	0h	
10	RXINT	W	0h	Set UART Receive Interrupt. 0h = Writing 0 has no effect 1h = Set Interrupt
9-1	RESERVED	W	0h	
0	RTOUT	W	0h	Set UARTOUT Receive Time-Out Interrupt. 0h = Writing 0 has no effect 1h = Set Interrupt

22.3.13 IMASK Register (Offset = 1088h) [Reset = 00000000h]

IMASK is shown in Table 22-26.

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 22-26. IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11	TXINT	R/W	0h	Enable UART Transmit Interrupt. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
10-0	RESERVED	R/W	0h	

22.3.14 RIS Register (Offset = 1090h) [Reset = 00000000h]

RIS is shown in Table 22-27.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking.

Table 22-27. RIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R	0h	
11	TXINT	R	0h	UART Transmit Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
10-0	RESERVED	R	0h	

22.3.15 MIS Register (Offset = 1098h) [Reset = 00000000h]

MIS is shown in Table 22-28.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Table 22-28. MIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R	0h	
11	TXINT	R	0h	Masked UART Transmit Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
10-0	RESERVED	R	0h	

22.3.16 ISET Register (Offset = 10A0h) [Reset = 00000000h]

ISET is shown in Table 22-29.

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.

Table 22-29. ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	W	0h	
11	TXINT	W	0h	Set UART Transmit Interrupt. 0h = Writing 0 has no effect 1h = Set Interrupt
10-0	RESERVED	W	0h	

22.3.17 INTCTL Register (Offset = 10E4h) [Reset = 00000000h]

INTCTL is shown in Table 22-30.

Return to the Summary Table.

Interrupt control register

Table 22-30. INTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	W	0h	
0	INTEVAL	W	0h	Writing a 1 to this field re-evaluates the interrupt sources. 0h = The interrupt or event line is disabled. 1h = The interrupt or event line is in software mode. Software must clear the RIS.

22.3.18 CTL0 Register (Offset = 1100h) [Reset = 00000038h]

CTL0 is shown in Table 22-31.

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.

Table 22-31. CTL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R/W	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. 0h = Least significant bit is sent first in the protocol packet 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 0h = Majority voting is disabled 1h = Majority voting is enabled
17	RESERVED	R/W	0h	
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). 0h = 16x oversampling. 1h = 8x oversampling. 2h = 3x oversampling. IrDA, Manchester and DALI not supported when 3x oversampling is enabled.
14	CTSEN	R/W	0h	Enable Clear To Send 0h = CTS hardware flow control is disabled. 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 0h = RTS hardware flow control is disabled. 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. 0h = Signal not RTS 1h = Signal RTS
11	RESERVED	R/W	0h	

Table 22-31. CTL0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10-8	MODE	R/W	0h	Set the communication mode and protocol used. (Not defined settings uses the default setting: 0) 0h = Normal operation 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. 2h = The UART operates in IDLE Line Mode 3h = The UART operates in 9 Bit Address mode 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. 5h = DALI Mode:
7	MENC	R/W	0h	Manchester Encode enable 0h = Disable Manchester Encoding 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. 0h = TXD pin is low 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. 0h = TXD pin can not be controlled by TXD_OUT 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. 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. 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. 0h = The receive section of the UART is disabled. 1h = The receive section of the UART is enabled.
2	LBE	R/W	0h	UART Loop Back Enable 0h = Normal operation. 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. 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. 0h = Disable Module 1h = Enable module

22.3.19 LCRH Register (Offset = 1104h) [Reset = 00000000h]

LCRH is shown in Table 22-32.

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.

Table 22-32. LCRH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R/W	0h	
26	SUSPEND	R/W	0h	Suspend external communication 0h = Functional mode resumed 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.) 0h = Smallest value 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 0h = Smallest value 1Fh = Highest possible value
15-8	RESERVED	R/W	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 0h = Normal operation 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. 0h = Disable Stick Parity 1h = Enable Stick Parity
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: 0h = 5 bits (default) 1h = 6 bits 2h = 7 bits 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. 0h = One stop bit is transmitted at the end of a frame. 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.

Table 22-32. LCRH Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
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 0h = Odd parity is performed, which checks for an odd number of 1s. 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 0h = Parity is disabled and no parity bit is added to the data frame. 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 0h = Normal operation 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).

22.3.20 STAT Register (Offset = 1108h) [Reset = 00000000h]

STAT is shown in Table 22-33.

Return to the Summary Table.

UART Status Register

Table 22-33. STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R	0h	
10	SENDIDLE	R	0h	TX FIFO Clear Status 0h = IDLE condition has not yet been sent on the line 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. 0h = IDLE has not been detected before last received character. (In idle-line multiprocessor mode). 1h = IDLE has been detected before last received character. (In idle-line multiprocessor mode).
8	CTS	R	1h	Clear To Send 0h = The CTS signal is not asserted (high). 1h = The CTS signal is asserted (low).
7	TXCLR	R	0h	TX FIFO Clear Status 0h = FIFO is not cleared 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. 0h = The transmitter is not full. 1h = transmit FIFO is full.
5	TXFE	R	1h	UART Transmit FIFO Empty 0h = The transmitter has data to transmit. 1h = transmit FIFO is empty.
4	RXCLR	R	0h	RX FIFO Clear Status 0h = FIFO is not cleared 1h = FIFO clear is complete
3	RXFF	R	0h	UART Receive FIFO Full 0h = The receiver can receive data. 1h = receive FIFO is full.
2	RXFE	R	1h	UART Receive FIFO Empty 0h = The receiver is not empty. 1h = eceive 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 (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). In IDLE_Line mode the Busy signal also stays set during the idle time generation. 0h = The UART is not busy. 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.

22.3.21 IFLS Register (Offset = 110Ch) [Reset = 00000022h]

IFLS is shown in Table 22-34.

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.

Table 22-34. IFLS Register 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 bittimes a RX interrupt is set even if the FIFO level is not reached. A value of 0 disables this function. 0h = Smallest value Fh = Highest possible value
7	RXCLR	R/W	0h	RX FIFO CLEAR. Setting this bit will clear the RX FIFO contents. 0h = Disable FIFO clear 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. 1h = RX FIFO >= 1/4 full 2h = RX FIFO >= 1/2 full (default) 3h = RX FIFO >= 3/4 full 4h = Opposite of empty 5h = RX FIFO is full 6h = RX_FIFO >= (MAX_FIFO_LEN -1) 7h = RX_FIFO <= 1
3	TXCLR	R/W	0h	TX FIFO CLEAR. Setting this bit will clear the TX FIFO contents. 0h = Disable FIFO clear 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. 1h = TX FIFO <= 3/4 empty 2h = TX FIFO <= 1/2 empty (default) 3h = TX FIFO <= 1/4 empty 4h = Opposite of full 5h = TX FIFO is empty 6h = TX FIFO <= 1 7h = TX_FIFO >= (MAX_FIFO_LEN -1)

22.3.22 IBRD Register (Offset = 1110h) [Reset = 00000000h]

IBRD is shown in Table 22-35.

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.

Table 22-35. IBRD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	
15-0	DIVINT	R/W	0h	Integer Baud-Rate Divisor 0h = Smallest value FFFFh = Highest possible value

22.3.23 FBRD Register (Offset = 1114h) [Reset = 00000000h]

FBRD is shown in Table 22-36.

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.

Table 22-36. FBRD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R/W	0h	
5-0	DIVFRAC	R/W	0h	Fractional Baud-Rate Divisor 0h = Smallest value 3Fh = Highest possible value

22.3.24 GFCTL Register (Offset = 1118h) [Reset = 0000000Xh]

GFCTL is shown in Table 22-37.

Return to the Summary Table.

This register control the glitch filter on the RX input.

Table 22-37. GFCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R/W	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. In IRDA mode: The minimum pulse length for receive is given by: t(MIN) = (DGFSEL) / f(IRTXCLK) 0h = Bypass GF 3Fh = Highest Possible Value

22.3.25 TXDATA Register (Offset = 1120h) [Reset = 00000000h]

TXDATA is shown in Table 22-38.

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.

Table 22-38. TXDATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	W	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. 0h = Smallest value FFh = Highest possible value

22.3.26 RXDATA Register (Offset = 1124h) [Reset = 00000000h]

RXDATA is shown in Table 22-39.

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.

Table 22-39. RXDATA Register 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. 0h = No noise error occurred 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. 0h = No break condition has occurred 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. 0h = No parity error has occurred 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. 0h = No framing error has occurred 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. 0h = Smallest value FFh = Highest possible value

22.3.27 LINCNT Register (Offset = 1130h) [Reset = 00000000h]

LINCNT is shown in Table 22-40.

Return to the Summary Table.

UART LIN Mode Counter Register

Table 22-40. LINCNT Register 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. 0h = Smallest value FFFFh = Highest possible value

22.3.28 LINCTL Register (Offset = 1134h) [Reset = 00000000h]

LINCTL is shown in Table 22-41.

Return to the Summary Table.

UART LIN Mode Control Register

Table 22-41. LINCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	
7	EN_FRM_ERR	R/W	0h	Enable FRAME ERROR 0h = Disable Frame Error Generation If STOP bit of '0' is received, then, FRAME ERROR is not generated 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. 0h = Counter compare match mode disabled (capture mode enabled) 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. 0h = Capture counter on positive UARTxRXD edge disabled 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. 0h = Capture counter on negative UARTxRXD edge disabled 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. 0h = Count while low Signal on UARTxRXD disabled 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 0h = Zero on negative edge disabled 1h = Zero on negative edge enabled
0	CTRENA	R/W	0h	LIN Counter Enable. LIN counter will only count when enabled. 0h = Counter disabled 1h = Counter enabled

22.3.29 LINC0 Register (Offset = 1138h) [Reset = 00000000h]

LINC0 is shown in Table 22-42.

Return to the Summary Table.

UART LIN Mode Capture 0 Register

Table 22-42. LINC0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	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. 0h = Smallest value FFFFh = Highest possible value

22.3.30 LINC1 Register (Offset = 113Ch) [Reset = 00000000h]

LINC1 is shown in Table 22-43.

Return to the Summary Table.

UART LIN Mode Capture 1 Register

Table 22-43. LINC1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	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) 0h = Smallest value FFFFh = Highest possible value

22.3.31 IRCTL Register (Offset = 1140h) [Reset = 00000000h]

IRCTL is shown in Table 22-44.

Return to the Summary Table.

IrDA Control Register

Table 22-44. IRCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R/W	0h	
9	IRRXPL	R/W	0h	IrDA receive input UCAxRXD polarity 0h = HIGH : IrDA transceiver delivers a high pulse when a light pulse is seen 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. Pulse length t(PULSE) = (IRTXPLx + 1) / [2 * f(IRTXCLK)] (IRTXCLK = functional clock of the UART) 0h = Smallest value 3Fh = Highest possible value
1	IRTXCLK	R/W	0h	IrDA transmit pulse clock select 0h (R/W) = IrDA encode data is based on the functional clock. 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 0h (R/W) = IrDA encoder/decoder disabled 1h (R/W) = IrDA encoder/decoder enabled

22.3.32 AMASK Register (Offset = 1148h) [Reset = 00000000h]

AMASK is shown in Table 22-45.

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.

Table 22-45. AMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	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. 0h = Smallest value FFh = Highest possible value

22.3.33 ADDR Register (Offset = 114Ch) [Reset = 00000000h]

ADDR is shown in Table 22-46.

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.

Table 22-46. ADDR Register 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. 0h = Smallest value FFh = Highest possible value

Chapter 23

UNICOMM I2C

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.

23.1 Overview 1229
23.2 Peripheral Functional Description 1230
23.3 UNICOMMI2CC Registers 1253
23.4 UNICOMMI2CT Registers 1293

23.1 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).

23.1.1 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.

23.1.2 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

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 23-1.

UNICOMM-I2CC (Controller) Feature Tag	Feature description
I2CC-ANALOG-FILTER	Analog glitch filter for input signals
I2CC-DIGITAL-FILTER	Digital glitch filter for input signals
I2CC-BURST	Hardware managed burst control send multiple bytes
I2CC-SMBUS	Hardware support for SMBus
I2CC-DMA	Support for DMA triggers
I2CC-EXTEND-CLKDIV	Clock divider control

Table 23-2.

UNICOMM-I2CT (Target) Feature Tag	Feature description
I2CT-ANALOG-FILTER	Analog glitch filter for input signals
I2CT-DIGITAL-FILTER	Digital glitch filter for input signals
I2CT-SECOND-TARGET-ADDR	Support for second target address
I2CT-SMBUS	Hardware support for SMBus including Quick Command and clock-low timeout detection
I2CT-DMA	Support for DMA triggers
I2CT-CLKDIV	Clock divider control

23.1.3 Functional Block Diagram

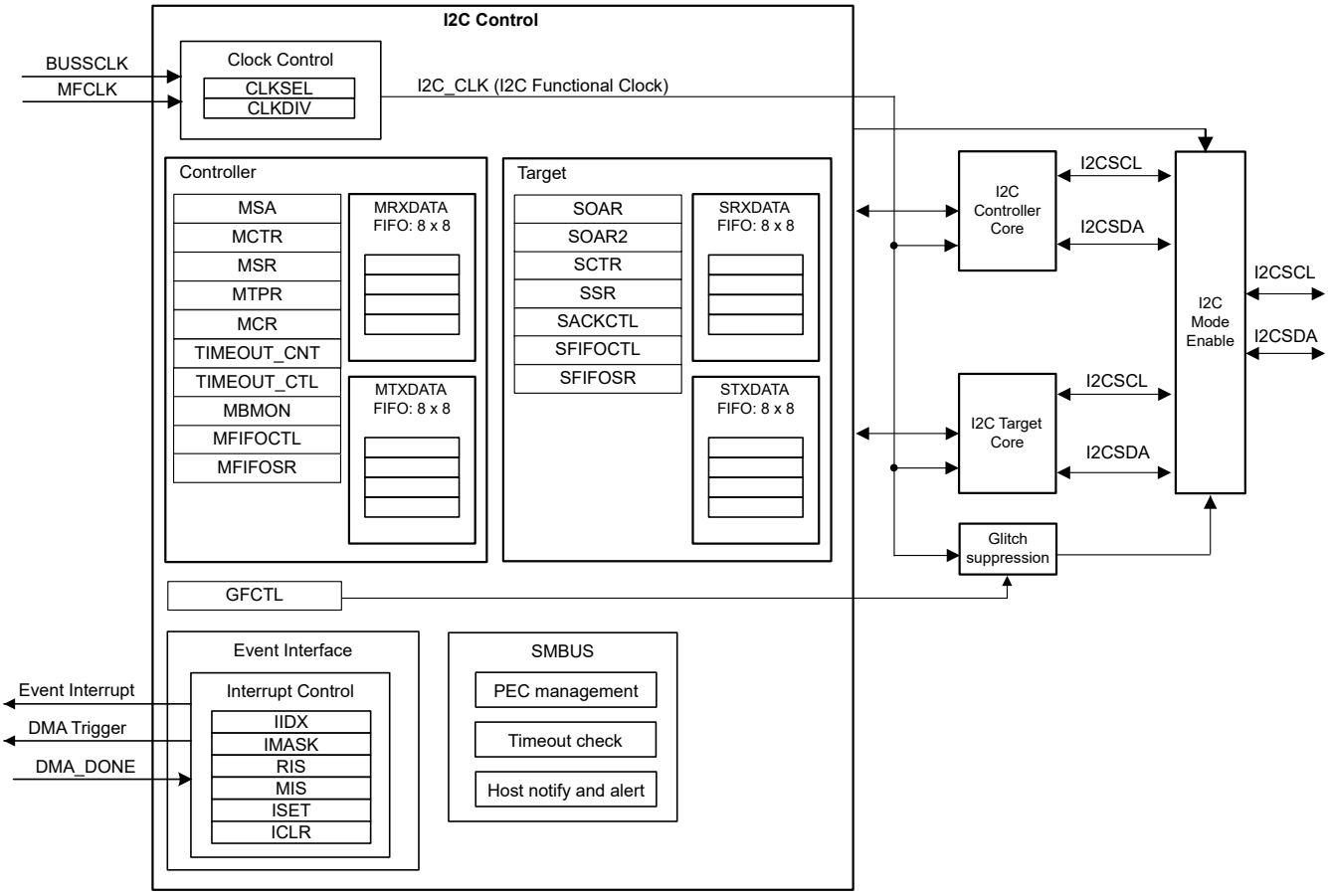


Figure 23-1. I2C Functional Block Diagram

23.2 Peripheral Functional Description

23.2.1 Clock Control

23.2.1.1 Clock Select and I²C Speed

Standard, Fast, and Fast Plus modes are selected using a value in the I²C Controller Timer Period (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 I2C_CLK functional clock frequency and the TPR field, as well as the fixed SCL_LP , and SCL_HP values 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. the CLKDIV.RATIO register field to select divide ratio of I²C functional clock, options are from divide by 1 to 8.
- 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
- TPR is the programmed value of the TPR field of the TPR register. This value is determined by replacing the known variables in the equation below and solving for TPR.

The I²C freqency is calculated as follows:

$$I2C\_FREQ = I2C\_CLK / ((1+TPR) * (SCL\_LP + SCL\_HP)) \quad (19)$$

For example, if the I²C functional clock frequency is 32 MHz and target SCL frequency is 400 kHz:

I2C_CLK = 32 MHz

I2C_FREQ = 400 kHz

SCL_LP = 6 , SCL_HP = 4

TPR = (I2C_CLK / (I2C_FREQ * (4 + 6))) - 1

TPR = 7 (0x07)

Table 23-3. 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 Plus 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 I²C functional clock must be greater than or equal to 20 times the SCL frequency, I2C_CLK ≥ 20 × I2C_FREQ.

The following minimum functional clock frequencies are required when running certain I2C clock speeds:

- I2C_CLK ≥ 2 MHz when working with I²C speed 0 to 100 kHz
- I2C_CLK ≥ 8 MHz when working with I²C speed 100 to 400 kHz
- I2C_CLK ≥ 20 MHz when working with I²C speed 400 kHz to 1 MHz

23.2.1.2 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 I²C module by setting the PWREN.ENABLE bit from the UNICOMM top level, the module is ready to start receiving and transmitting data.

23.2.2 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 23-4. I²C signal descriptions

Device Pin	Function
I2Cx_SCL	I²C clock signal
I2Cx_SDA	I²C data signal

23.2.3 General Architecture

23.2.3.1 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 transaction is defined as the time between a valid START and STOP condition—as described in Figure 23-2. 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.

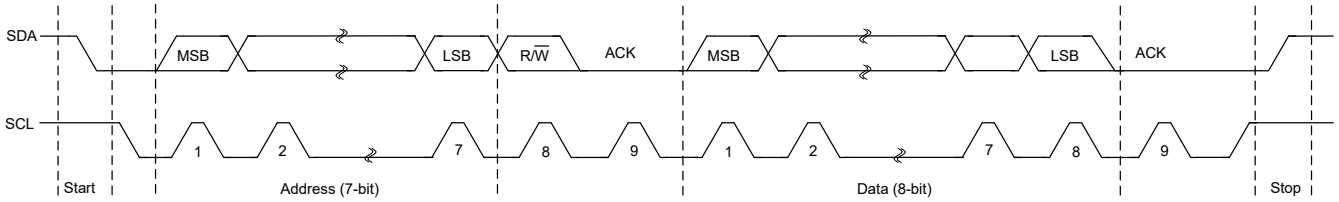


Figure 23-2. Module Data Transfer

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 23-3), otherwise START or STOP conditions are generated.

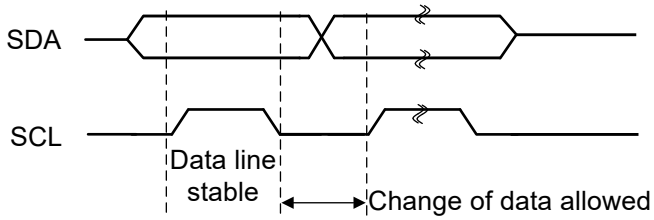


Figure 23-3. Data Change Period

23.2.3.2 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.

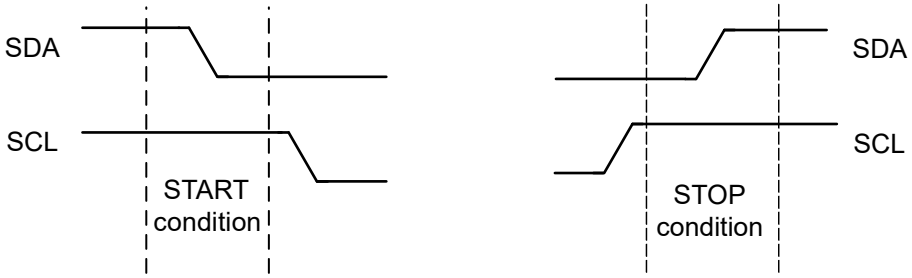


Figure 23-4. START and STOP Conditions

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 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 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 23.2.4.1

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 23.2.4.2

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 I2Cx.MSR.CLKTO is set once both SCL and SDA are pulled to high.
- BUSY is also set after a START/Repeated START, 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 there is no ongoing transfer.

23.2.3.3 Data Format with 7-Bit Address

Data transfers follow the format shown in Figure 23-5. 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 23.2.3.4 .

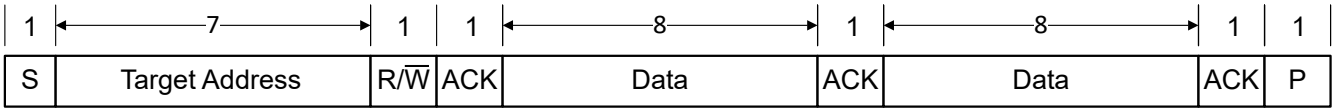


Figure 23-5. Data Format with 7-Bit Address

With the CTR.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.

23.2.3.4 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.

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.

If the I^2C 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.

23.2.3.5 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 23-6. After a RESTART is issued, the target address is again sent out with the new data direction specified by the R/W bit.

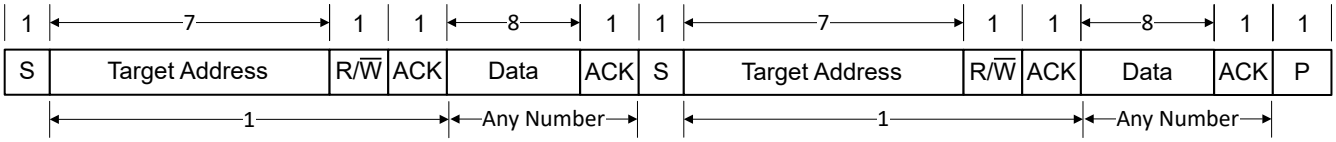


Figure 23-6. Data Format with Repeated Start

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.

23.2.3.6 Clock Low Timeout (CLKTO)

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 TIMEOUT_CTL.TCNTLA field. The value programmed in the TIMEOUT_CTL.TCNTLA register field has to be greater than 0x01 to enable the CLKTO 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 functional clocks FCLK ticks I2Cclk ticks where the TPR is the programmable timer period. The Controller Clock Low Timeout counter counts for the entire time SCL is held low continuously. When SCL is high, the TIMEOUT_CNT.TCNTA is reloaded with the value in the TIMEOUT_CTL.TCNTLA register, 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 = I2C_CLK period * (1 + TPR) * 12
- Timeout counter = TIMEOUT_CTL.TCNTA 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 23.2.1.1 on how TPR is calculated. One timeout period is equal to (1 / 20 MHz) × (1 + 19) × 12 or 12 µs. Programming the TIMEOUT_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 CLKTO bit in the I^2C Controller Status CONTROLLER.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 MBMON register to help determine the state of the remote target.

In the event of a CLKTO condition, application software must choose how it intends to attempt bus recovery. If a CLKTO 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 low or high timeout configuration must be set only during initialization and not when the I2C module is in an active state.

23.2.3.7 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 I^2C 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 CTR.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 I^2C 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 I^2C 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.

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 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.

23.2.3.8 Dual Address

The I^{2}C target interface supports dual address capability for the target. An additional programmable I^{2}C Target Own Address Register I2Cx.SOAR2 is provided and can be matched if enabled. When dual address is disabled (I2Cx.SOAR2.OAR2EN=0), the I^{2}C target provides an ACK on the bus if the address matches the OAR field in the I2Cx.SOAR register. When dual address mode is enabled (I2Cx.SOAR2.OAR2EN=1), the I^{2}C 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, there is either a primary address match or no OAR2 address match.

23.2.3.9 Arbitration

A controller can start a transfer only if the bus is in the IDLE state. If two or more controllers generate a START condition within the minimum hold time of the START condition, an arbitration scheme takes place on the SDA line while SCL is high (see Figure 23-7) . 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.

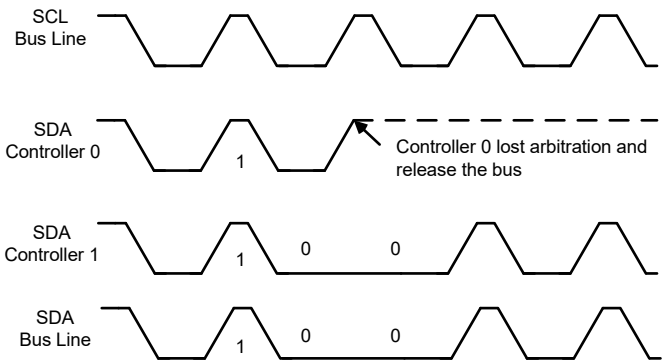


Figure 23-7. Arbitration Procedure Between Two Controller Transmitters

When an arbitration loss is detected, the I2Cx.MSR.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^{2}C controller has initiated a transfer, the application software needs to execute the following steps to correctly handle the arbitration loss:

- Flush the TX FIFO
- 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.

23.2.3.10 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 held low by the device with the longest low period. The other devices must wait until the SCL is released before starting high periods.

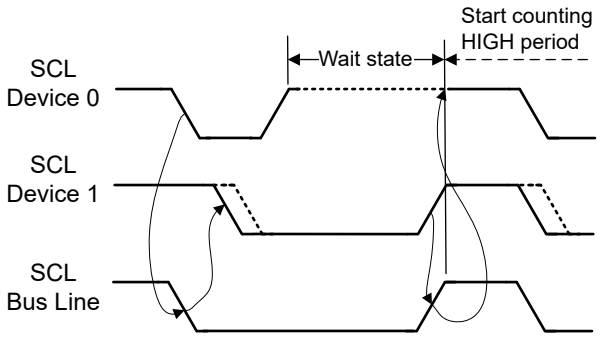


Figure 23-8. Clock Synchronization

23.2.3.11 Glitch Suppression

The I^{2}C module supports glitch suppression on the SCL and SDA lines to meet the 50ns glitch suppression as specified in the I^{2}C specification.

Analog glitch Filter

In UNICOMM-I2CC configurations that support I2CC-ANALOG-FILTER, by default, an analog glitch filter is enabled and configured to suppress spikes with a pulse width up to 50ns. I^{2}C spec advises to suppress noise spikes of less than 50ns. The user can disable this filter by clearing the GFCTL.AGFEN bit and the suppression pulse width can also be configured by using GFCTL.AGFEN. The analog glitch filter can only be used to wake up the I2C in low-power mode.

Digital Glitch Filter

In UNICOMM-I2CC configurations that support I2CC-DIGITAL-FILTER, 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^{2}C functional clocks. 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. For more information please see the GFCTL register description. The digital glitch filter can't be used to wake up the I2C from low-power mode.

Table 23-5. Glitch Suppression Filter

	Analog Glitch Filter	Digital Glitch Filter
Default	Default enabled with 50ns	Default bypassed
Pulse width of suppressed spikes	Configurable 5ns, 10ns, 25ns, 50ns	Programmable I^{2}C clock cycle 1, 2, 3, 4, 8, 16, 31
Benefits	Available without needing clock	• Programmable length to provide extra filtering . • Stable filtering length.
Limitation	Variation with temperature, voltage, process	• Does not work in low-power mode wakeup when there is no sufficient clock . • Only enabled 3 clock cycles after start of I^{2}C packet.

Note

The glitch filter configuration within the register GFCTL must only be modified while the I^{2}C module for controller and target is not enabled.

23.2.3.12 Burst Mode

I2C Controller burst mode allows a sequence of data transfers using DMA or software to handle the data in the buffer/FIFO. Burst mode is enabled by setting the CTR.MBLEN 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 and send out a STOP/RE-START condition by writing to the START/STOP fields.

23.2.3.13 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 transfer 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.

23.2.3.14 Low Power Operation

I^2C 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 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 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

(More information will be added in next revision)

23.2.4 Protocol Descriptions & Initialization

23.2.4.1 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.

23.2.4.1.1 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 (Available in UNICOMM-I2CC modules that support the I2CC-BURST feature)
- 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 23-6. 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 23-7. 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 23-8. 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 23-9. 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.

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 23-10. 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 23-11. 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 23-12. 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 23-13. 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

23.2.4.1.2 Controller Mode Operation

I²C Controller Initialization

1. Configure the module to UNICOMM-I2CC mode in the IPMODE.SELECT bits at the UNICOMM top level
2. Configure SDA and SCL pin functions and select as input by using the IOMUX registers.
3. Reset the peripheral using RSTCTL register at the UNICOMM level.
4. Enable the power to peripheral using PWREN register at the UNICOMM level.
5. Select and configure the I²C clock using the CLKCTL and CLKDIV registers.
6. Set the desired SCL clock speed of by writing the TPR bit in TPR 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 TPR register with the value of 0x13.
7. Specify the target address and mode (transmit or receive) for the next operation by writing the TA register.
8. If controller is transmitting data, user can place data (byte) to be transmitted in the data register by writing the TXDATA register with the desired data.
9. Configure the controller transmit or receive mode by writing the CTR register. For more information on how to configure CTR register for different mode please see Controller Configuration section.
10. Enable desired interrupts and/or DMA event by using CPU_INT.IMASK register.

I²C Controller Status

The application can read the SR register to check the current state of the I²C controller.

I²C Controller Receiver Mode

For controller to start receive data out of the idle mode, 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 it detects that the bus is free. All the process below should be followed.

TA.DIR is set to 1 to enable receive mode, CTR.START is set to generate start condition, CTR.BLEN can be programmed to indicate the number of bytes (n) for the receive operation. CTR.ACK and CTR.STOP bit can be set or clear based on user configuration. CTR.FRM_START 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 RXDONE (0x01) interrupt in CPU_INT.IIDX register is set to indicate that controller receive transaction is completed. User can use the RXFIFOTRG (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 the RXTRIG bit in I2Cx.MFIFOCTL register. The flow chart of controller receiver mode is shown in Figure 23-9.

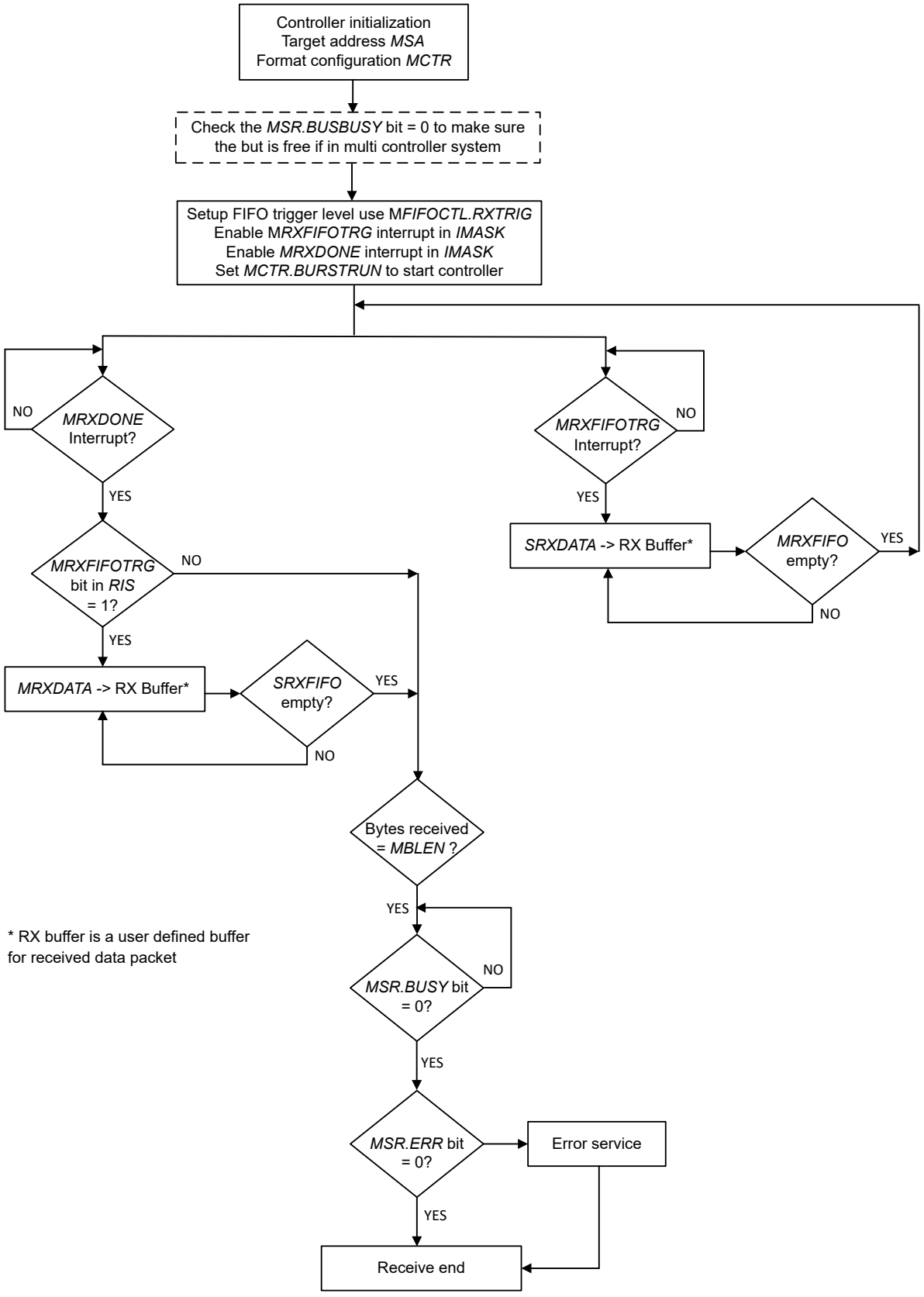


Figure 23-9. Controller Receiver Mode

I²C Controller transmitter Mode

For 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 it detects that the bus is free. The data written into TXDATA is transmitted if arbitration is not lost during transmission of the target address. The process below should be followed.

TA.DIR is cleared to enable transmit mode, CTR.START is set to generate start condition, CTR.BLEN can be programmed to indicate the number of bytes (n) for the transmit operation. CTR.STOP bit can be set or clear based on user configuration. CTR.FRM_START 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 TXDONE (0x02) interrupt in CPU_INT.IIDX register is set to indicate that controller transmit transaction is completed. User can also set use the TXEMPTY (0x06) interrupt in CPU_INT.IIDX register to see if the TXFIFO 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 receiver mode is shown in Figure 23-10.

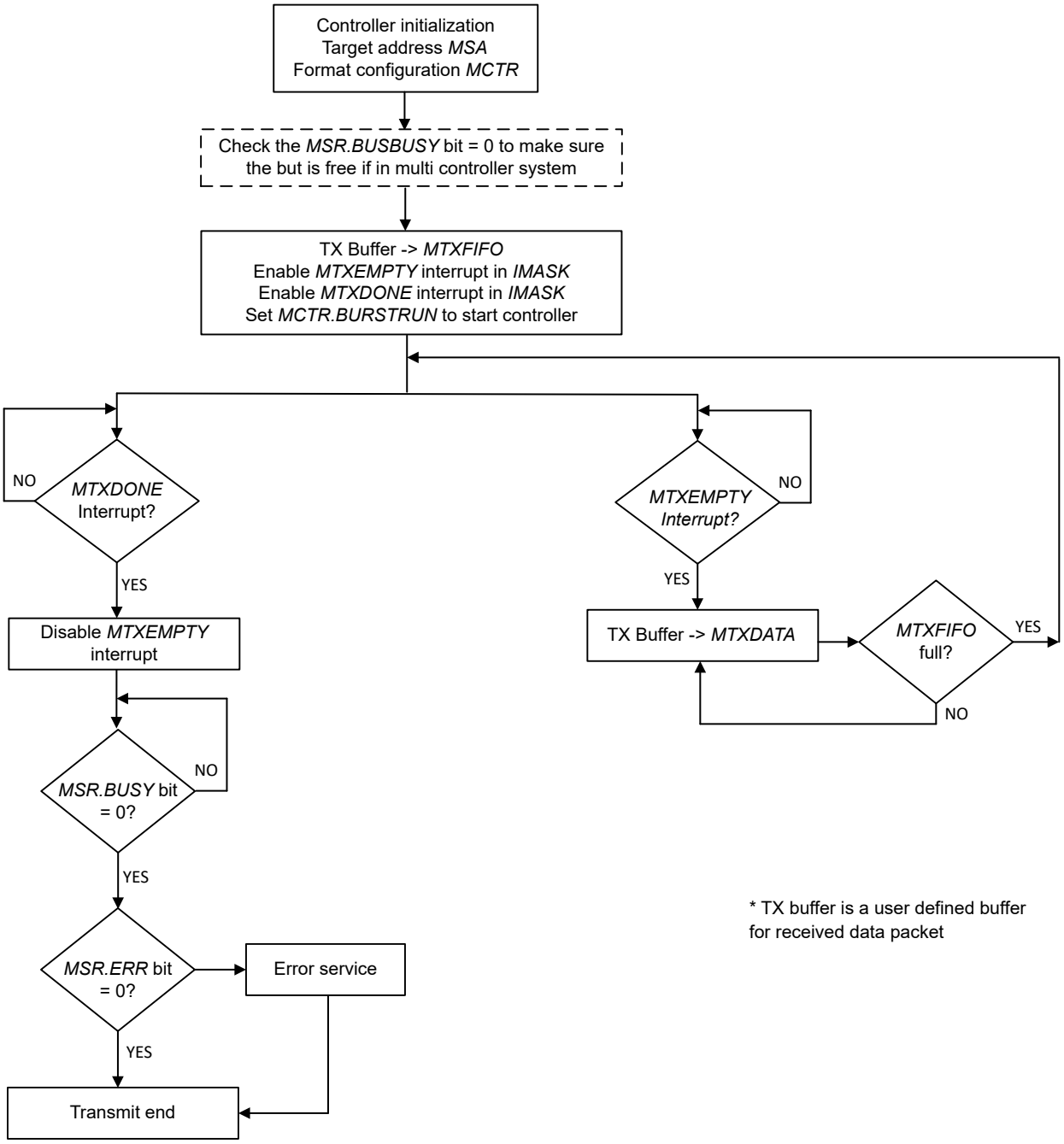


Figure 23-10. Controller Transmitter Mode

23.2.4.2 I²C Target Mode

23.2.4.2.1 Target Mode Operation

I²C Target Initialization

1. Configure the module to UNICOMM-I2CT mode in the IPMODE.SELECT bits at the UNICOMM top level
2. Configure SDA and SCL pin functions and select as input by using the IOMUX registers.
3. Reset the peripheral using RSTCTL register at the UNICOMM level
4. Enable the power to peripheral using PWREN register at the UNICOMM level
5. Select and configure the I²C clock using the CLKCTL and CLKDIV registers.

6. Configure at least one target address by writing the 7-bit address to OAR register. The additional target address can be enabled and configured by using OAR2 register in UNICOMM-I2CT configuraitons that support the I2CT-SECOND-TARGET-ADDR feature.
7. Enable desired interrupts and/or DMA event by using CPU_INT.IMASK register.
8. The general call response can be enabled by setting the GENCALL bit in CTR register.
9. Enable the I^2C target mode by setting the ENABLE bit in CTR register.

I^2C Target Status

User can read the SR register to check the current state of the I^2C target.

I^2C 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.

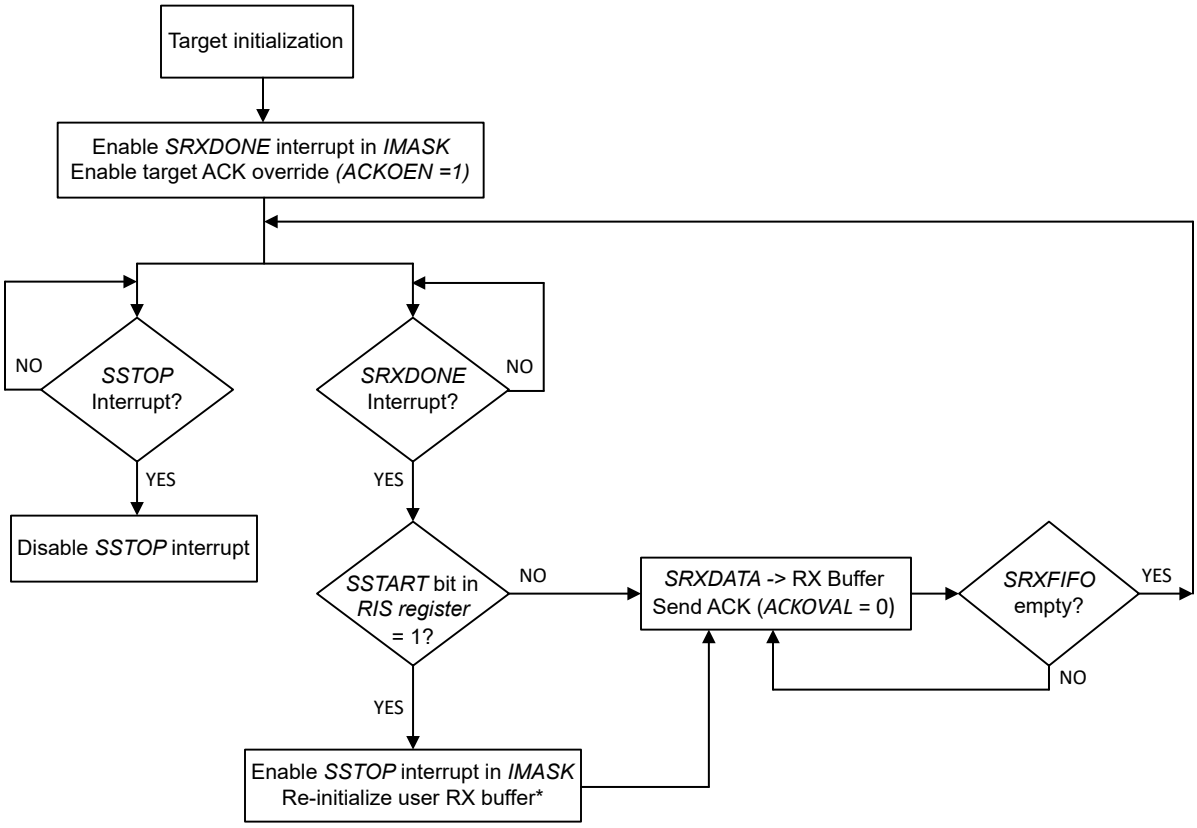
After the a data byte is received, the RXDONE interrupt in CPU_INT.IIDX register is set to indicate that a byte has been received. The I^2C module automatically acknowledges the received data or user can choice to manually send acknowledge after each byte received by configuring the ACKCTL register.

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 set use the RXFIFOTRG 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 the FIFOCTL register.

The RXDONE approach could be used if target wants to slow down communication to evaluate reception of every byte, while the RXFIFOTRG approach is used to maximize throughput and avoid clock stretching.

The flow chart of using RXDONE and RXFIFOTRG interrupt to read the receive data are shown in Figure 23-11 and Figure 23-12 respectively.



* RX buffer is a user defined buffer for received data packet

Figure 23-11. Target Receiver Mode using SRXDONE and ACK Override

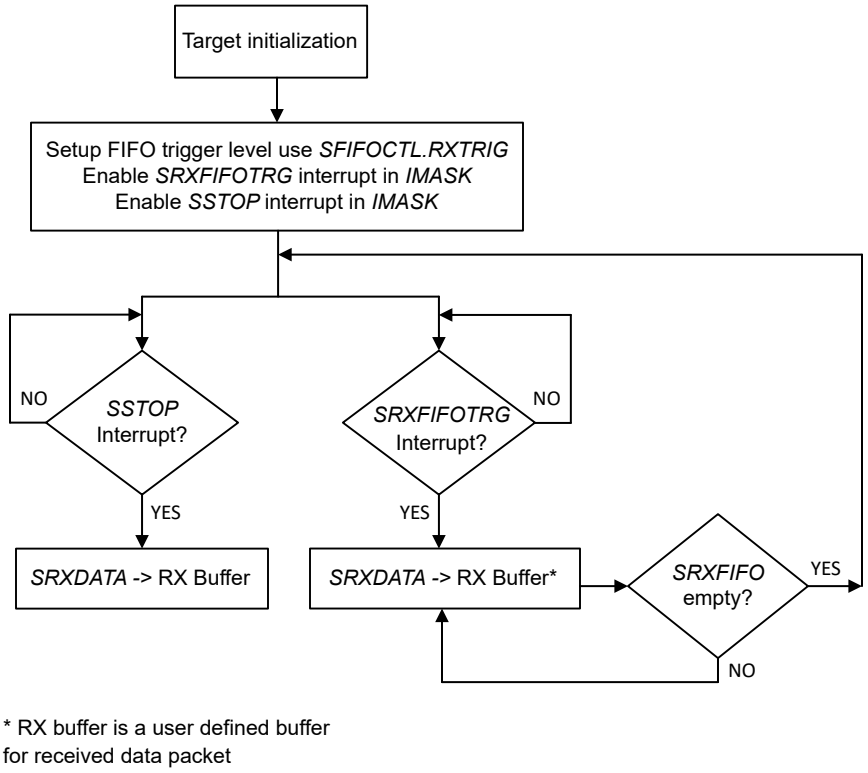


Figure 23-12. Target Receiver Mode using SRXFIFOTRG and Automatic ACK

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.

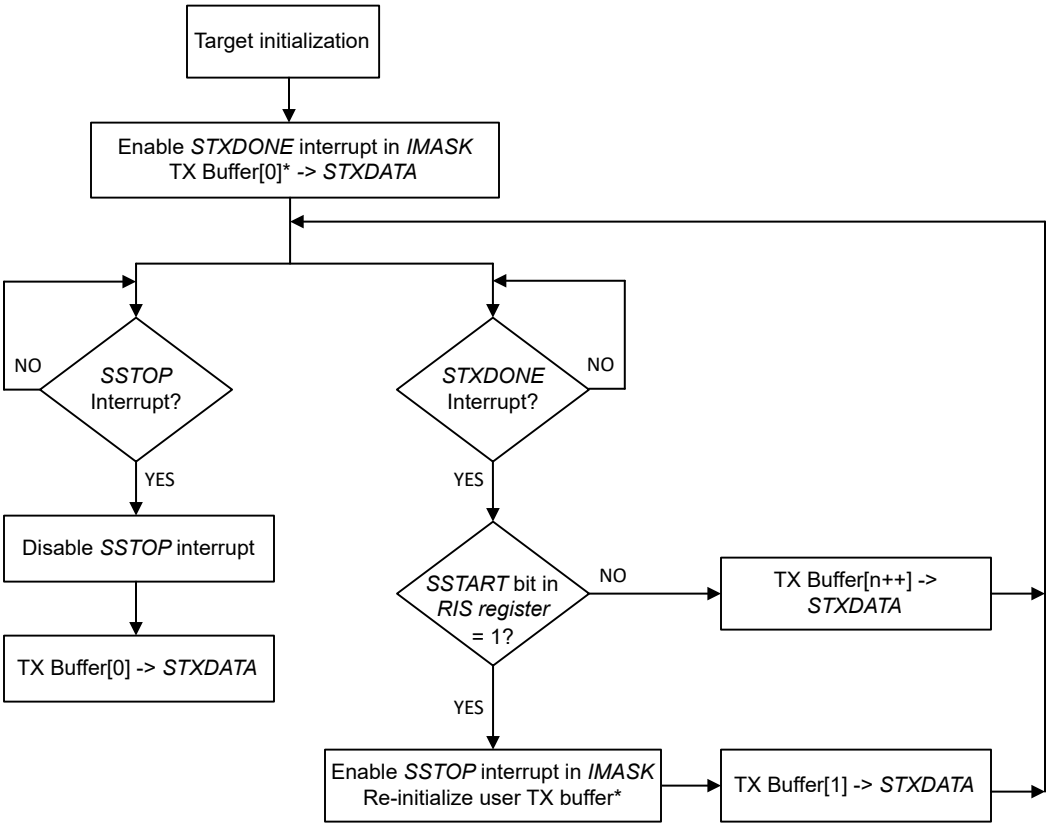
After a data byte is transmitted, the TXDONE interrupt in CPU_INT.IIDX register is set to indicate that a byte has been transmitted.

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.

User can also set use the TXFIFOTRG interrupt in CPU_INT.IIDX register to load the data to the transmit FIFO. This interrupt will trigger when the transmit FIFO contains <= defined bytes. The trigger level can be defined by using TXFIFOTRG bit in the FIFOCTL register.

The TXDONE approach could be used if target wants to slow down communication to evaluate transmit of every byte, while the TXFIFOTRG 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 23-13 and Figure 23-14.



* TX buffer is a user defined buffer for transmit data packet

Figure 23-13. Target Transmitter Mode using STXDONE

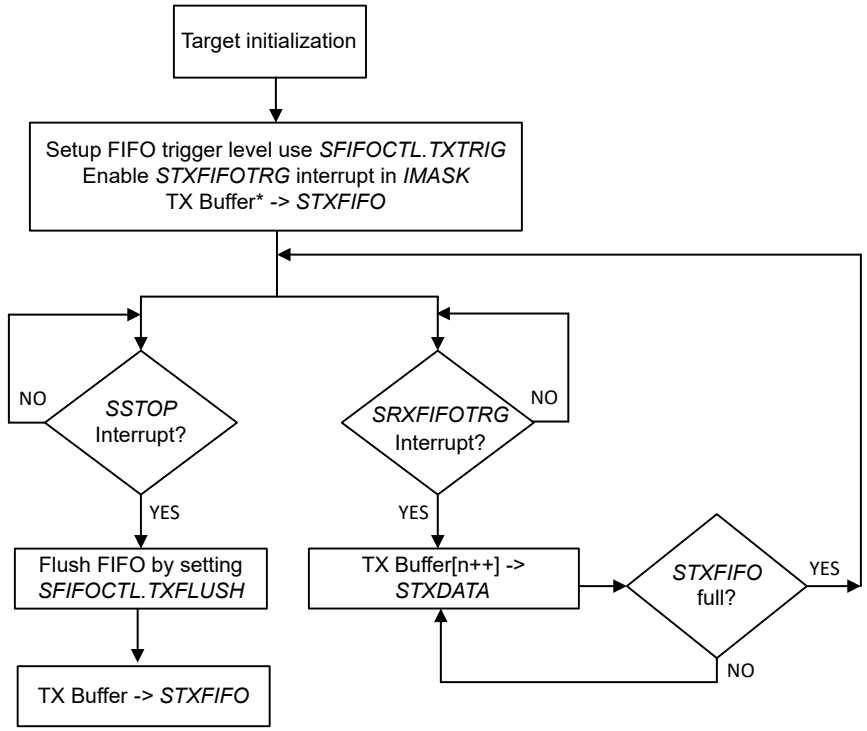


Figure 23-14. Target Transmitter Mode using STXFIFOTRG

23.2.5 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. Resets must only be issued after terminating a transaction.

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 23-14 shows the behavior of status bits when Controller or Target gets disabled.

Table 23-14. Status Bits when Controller is Disabled

Register	Bit	Behavior when Controller is Disabled	Behavior when Target Disabled	Behavior after Controller/Target is 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
	BCNT	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

Table 23-14. Status Bits when Controller is Disabled (continued)

Register	Bit	Behavior when Controller is Disabled	Behavior when Target Disabled	Behavior after Controller/Target is Enabled
I2Cx.MBMON	SCL	Reset State	Don't care	Updates with Controller enable
	SDA	Reset State	Don't care	Updates with Controller enable

Table 23-15. 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

23.2.6 Initialization

Please see Section 23.2.4.1.2 for controller initialization and Section 23.2.4.2.1 for target initialization.

23.2.7 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_TRIG1, DMA_TRIG0) are used to setup the trigger signaling for the DMA through DMA event route.

The I^2C events are summarized in Table 23-16.

Table 23-16. 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_TRIG1 registers	Fixed interrupt route from I^2C to DMA
DMA trigger	Publisher	I^2C	DMA	DMA event route	DMA_TRIG0 registers	Fixed interrupt route from I^2C to DMA

23.2.7.1 CPU Interrupt Event Publisher (CPU_INT)

The I^2C 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 I^2C are:

Table 23-17. I^2C 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

Table 23-17. I²C CPU Interrupt Event Conditions for Controller (CPU_INT) (continued)

IIDX STAT	Name	Description
0x05	RXFULL	Controller RXFIFO full event. This interrupt is set if an RX FIFO is full.
0x07	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)
0x010	DMA_DONE_RX	Controller DMA TX done
0x011	DMA_DONE_TX	Controller DMA RX done

Table 23-18. 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.

23.2.7.2 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 23-19. 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

Table 23-19. I²C DMA Trigger Condition (DMA_TRIG1 and DMA_TRIG0) (continued)

IIDX STAT	Name	Description
0x04	STXFIFOTRG	Target transmit FIFO trigger. Trigger when Transmit FIFO contains <= defined bytes

The DMA trigger event configuration is managed with the DMA_TRIG1 and DMA_TRIG0 event management registers. See Section 8.2.5 for guidance on configuring the Event registers and Section 8.1.3.2 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 23-15, each DMA channel can be triggered by any of the conditions listed in Table 23-19 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.

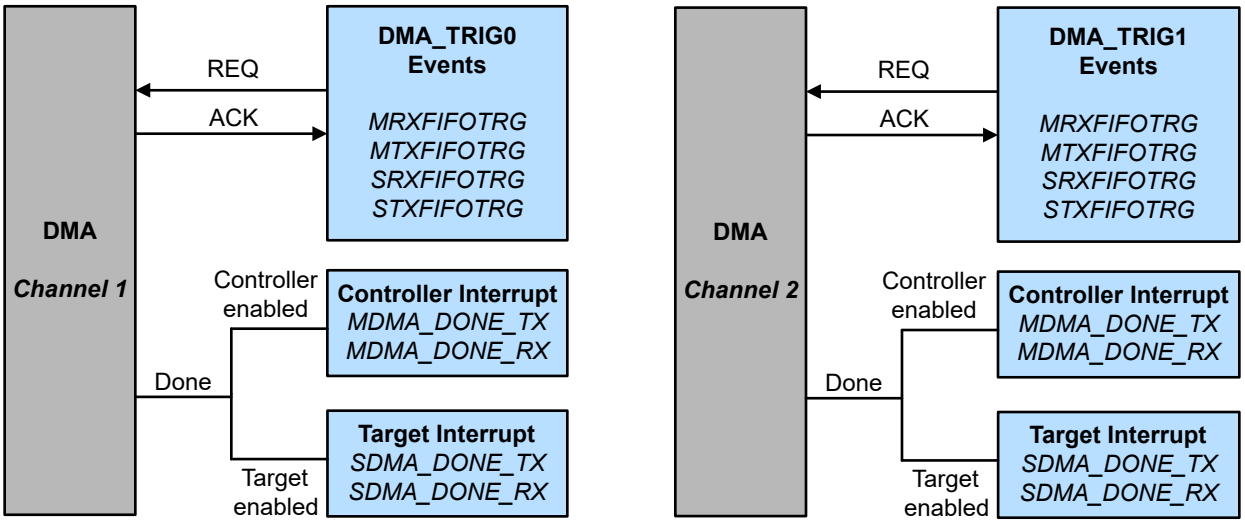


Figure 23-15. I²C DMA Trigger and Status

23.2.8 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 23-20. 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

23.3 UNICOMMI2CC Registers

Table 23-21 lists the memory-mapped registers for the UNICOMMI2CC registers. All register offset addresses not listed in Table 23-21 should be considered as reserved locations and the register contents should not be modified.

Table 23-21. UNICOMMI2CC Registers

Offset	Acronym	Register Name	Section
1000h	CLKDIV	Clock Divider	Section 23.3.1
1008h	CLKSEL	Clock Select for Ultra Low Power peripherals	Section 23.3.2
1020h	IIDX	Interrupt index	Section 23.3.3
1028h	IMASK	Interrupt mask	Section 23.3.4
1030h	RIS	Raw interrupt status	Section 23.3.5
1038h	MIS	Masked interrupt status	Section 23.3.6
1040h	ISET	Interrupt set	Section 23.3.7
1048h	ICLR	Interrupt clear	Section 23.3.8
1058h	IMASK	Interrupt mask	Section 23.3.9
1060h	RIS	Raw interrupt status	Section 23.3.10
1068h	MIS	Masked interrupt status	Section 23.3.11
1070h	ISET	Interrupt set	Section 23.3.12
1088h	IMASK	Interrupt mask	Section 23.3.13
1090h	RIS	Raw interrupt status	Section 23.3.14
1098h	MIS	Masked interrupt status	Section 23.3.15
10A0h	ISET	Interrupt set	Section 23.3.16
10E4h	INTCTL	Interrupt control register	Section 23.3.17
1100h	CTR	Control Register	Section 23.3.18
1104h	CR	Configuration	Section 23.3.19
1108h	SR	Status Register	Section 23.3.20
110Ch	IFLS	Interrupt FIFO Level Select Register	Section 23.3.21
1110h	TPR	Timer Period	Section 23.3.22
1118h	GFCTL	I2C Glitch Filter Control	Section 23.3.23
111Ch	BMON	Bus Monitor	Section 23.3.24
1120h	TXDATA	TXData	Section 23.3.25
1124h	RXDATA	RXData	Section 23.3.26
1128h	PECSR	PEC status register	Section 23.3.27
114Ch	TA	Target Address Register	Section 23.3.28
1150h	TIMEOUT_CNT	I2C Timeout Count Register	Section 23.3.29
1154h	TIMEOUT_CTL	I2C Timeout Count Control Register	Section 23.3.30
1158h	PECCTL	I2C PEC control register	Section 23.3.31

Complex bit access types are encoded to fit into small table cells. Table 23-22 shows the codes that are used for access types in this section.

Table 23-22. UNICOMMI2CC Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write

Table 23-22. UNICOMMI2CC Access Type Codes (continued)

Access Type	Code	Description
Reset or Default Value		
-n		Value after reset or the default value

23.3.1 CLKDIV Register (Offset = 1000h) [Reset = 00000000h]

CLKDIV is shown in Table 23-23.

Return to the Summary Table.

This register is used to specify module-specific divide ratio of the functional clock

Table 23-23. CLKDIV Register 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 Division factor 0 : DIV_BY_1 1 : DIV_BY_2 .... 63: DIV_BY_64 0h = Do not divide clock source 1h = Divide clock source by 2 2h = Divide clock source by 3 3h = Divide clock source by 4 4h = Divide clock source by 5 5h = Divide clock source by 6 6h = Divide clock source by 7 7h = Divide clock source by 8 3Ch = Divide clock source by 8 3Dh = Divide clock source by 8 3Eh = Divide clock source by 8 3Fh = Divide clock source by 8

23.3.2 CLKSEL Register (Offset = 1008h) [Reset = 00000000h]

CLKSEL is shown in Table 23-24.

Return to the Summary Table.

Clock source selection.

Table 23-24. CLKSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11	ASYNC_PLL_SEL	R/W	0h	Asynchronous PLL selected as source 0h = Does not select this clock as a source 1h = Select this clock as a source
10	ASYNC_HFCLK_SEL	R/W	0h	Asynchronous HFCLK selected as source 0h = Does not select this clock as a source 1h = Select this clock as a source
9	ASYNC_SYSCLK_SEL	R/W	0h	Asynchronous sysclk selected as source 0h = Does not select this clock as a source 1h = Select this clock as a source
8-4	RESERVED	R/W	0h	
3	BUSCLK_SEL	R/W	0h	Selects BUS CLK as clock source if enabled 0h = Does not select this clock as a source 1h = Select this clock as a source
2	MFCLK_SEL	R/W	0h	Selects MFCLK as clock source if enabled 0h = Does not select this clock as a source 1h = Select this clock as a source
1-0	RESERVED	R/W	0h	

23.3.3 IIDX Register (Offset = 1020h) [Reset = 00000000h]

IIDX is shown in Table 23-25.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index.

Table 23-25. 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 00h = No interrupt pending 01h = Data received 02h = data transmitted 03h = Receive Trigger 04h = Transmit Trigger 5h = RX FIFO FULL Event/interrupt pending 6h = Transmit FIFO/Buffer Empty Event/interrupt pending 08h = Address/Data NACK 09h = Start Event 0Ah = Stop Event 0Bh = Arbitration Lost Ch = PEC Receive Error Event Dh = Timeout A Event Eh = Timeout B Event 10h = DMA DONE on Channel RX 11h = DMA DONE on Channel TX 13h = DMA PRE IRQ INTERRUPT 14h = DMA PRE IRQ INTERRUPT

23.3.4 IMASK Register (Offset = 1028h) [Reset = 00000000h]

IMASK is shown in Table 23-26.

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 23-26. IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R/W	0h	
19	RESERVED	R	0h	Reserved
18	RESERVED	R	0h	Reserved
17	RESERVED	R/W	0h	
16	DMA_DONE_TX	R/W	0h	DMA Done on Event Channel TX 0h = Interrupt disabled 1h = Set Interrupt Mask
15	DMA_DONE_RX	R/W	0h	DMA Done on Event Channel RX 0h = Interrupt disabled 1h = Set Interrupt Mask
14	RESERVED	R/W	0h	
13	TIMEOUTB	R/W	0h	Timeout B Interrupt 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
12	TIMEOUTA	R/W	0h	TIMEOUTA interrupt 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
11	PEC_RX_ERR	R/W	0h	RX Pec Error Interrupt 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
10	ARBLOST	R/W	0h	Arbitration Lost interrupt 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
9	STOP	R/W	0h	STOP interrupt 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
8	START	R/W	0h	START interrupt 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
7	NACK	R/W	0h	NACK interrupt 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
6	RESERVED	R/W	0h	
5	TXEMPTY	R/W	0h	Transmit FIFO/Buffer Empty interrupt. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
4	RXFULL	R/W	0h	RXFIFO full event 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
3	TXTRG	R/W	0h	Transmit Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
2	RXTRG	R/W	0h	Receive Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
1	TXDONE	R/W	0h	Transmit Transaction completed Interrupt 0h = Clear Interrupt Mask 1h = Set Interrupt Mask

Table 23-26. IMASK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	RXDONE	R/W	0h	RXDONE interrupt 0h = Clear Interrupt Mask 1h = Set Interrupt Mask

23.3.5 RIS Register (Offset = 1030h) [Reset = 00000000h]

RIS is shown in Table 23-27.

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.

Table 23-27. RIS Register 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 Event Channel TX 0h = Interrupt disabled 1h = Set Interrupt Mask
15	DMA_DONE_RX	R	0h	DMA Done on Event Channel RX 0h = Interrupt disabled 1h = Set Interrupt Mask
14	RESERVED	R	0h	
13	TIMEOUTB	R	0h	Timeout B Interrupt 0h = Interrupt did not occur 1h = Interrupt occured
12	TIMEOUTA	R	0h	Timeout A Interrupt 0h = Interrupt did not occur 1h = Set Interrupt Mask
11	PEC_RX_ERR	R	0h	Receive PEC Error Interrupt: set when the calculated PEC does not match received PEC 0h = Interrupt did not occur 1h = Interrupt Occured
10	ARBLOST	R	0h	Arbitration Lost Interrupt: in multi-controller systems, when this controller loses out during arbitration 0h = Interrupt did not occur 1h = Set Interrupt Mask
9	STOP	R	0h	STOP Detection Interrupt 0h = Interrupt did not occur 1h = Set Interrupt Mask
8	START	R	0h	START Detection Interrupt 0h = Interrupt did not occur 1h = Set Interrupt Mask
7	NACK	R	0h	Address/Data NACK Interrupt: set when address or data nack is received 0h = Interrupt did not occur 1h = Set Interrupt Mask
6	RESERVED	R	0h	
5	TXEMPTY	R	0h	Transmit FIFO Empty: set when Transmit FIFO is empty 0h = Interrupt did not occur 1h = Interrupt occured
4	RXFULL	R	0h	RXFIFO full event. This interrupt is set when receive FIFO is full. 0h = Interrupt did not occur 1h = Interrupt occured
3	TXTRG	R	0h	Transmit Trigger When FIFO is present, as per IFLS settings When FIFO is not present, trigger when transmit buffer is empty 0h = Clear Interrupt Mask 1h = Set Interrupt Mask

Table 23-27. RIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	RXTRG	R	0h	Receive Trigger When FIFO is present, as per IFLS settings when FIFO is not present, this indicates when buffer is full 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
1	TXDONE	R	0h	Transmit Done Interrupt: TXDONE interrupt is raised when a burst length completes Or, in case of quick command, when a quick command with R/Wn bit set to '0' 0h = Interrupt did not occur 1h = Set Interrupt Mask
0	RXDONE	R	0h	Receive Done Interrupt: RXDONE interrupt is raised when a burst length completes Or, in case of quick command, when a quick command with R/Wn bit set to '1' 0h = Interrupt did not occur 1h = Set Interrupt Mask

23.3.6 MIS Register (Offset = 1038h) [Reset = 00000000h]

MIS is shown in Table 23-28.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Table 23-28. MIS Register 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 Event Channel TX 0h = Interrupt disabled 1h = Set Interrupt Mask
15	DMA_DONE_RX	R	0h	DMA Done on Event Channel RX 0h = Interrupt disabled 1h = Set Interrupt Mask
14	RESERVED	R	0h	
13	TIMEOUTB	R	0h	Timeout B Interrupt 0h = Clear interrupt mask 1h = Set interrupt mask
12	TIMEOUTA	R	0h	Timeout A Interrupt 0h = Interrupt did not occur 1h = Set Interrupt Mask
11	PEC_RX_ERR	R	0h	RX Pec Error Interrupt 0h = Clear interrupt mask 1h = Set interrupt mask
10	ARBLOST	R	0h	Arbitration Lost Interrupt 0h = Interrupt did not occur 1h = Set Interrupt Mask
9	STOP	R	0h	STOP Detection Interrupt 0h = Interrupt did not occur 1h = Set Interrupt Mask
8	START	R	0h	START Detection Interrupt 0h = Interrupt did not occur 1h = Set Interrupt Mask
7	NACK	R	0h	Address/Data NACK Interrupt 0h = Interrupt did not occur 1h = Set Interrupt Mask
6	RESERVED	R	0h	
5	TXEMPTY	R	0h	Transmit FIFO Empty masked interrupt. This interrupt is set if all data in the Transmit FIFO have been shifted out and the transmit goes into idle mode. 0h = Interrupt did not occur 1h = Interrupt occured
4	RXFULL	R	0h	Masked RXFIFO full event 0h = Interrupt did not occur 1h = Interrupt occured
3	TXTRG	R	0h	Transmit Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
2	RXTRG	R	0h	Receive Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask

Table 23-28. MIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	TXDONE	R	0h	Transmit Transaction completed Interrupt 0h = Interrupt did not occur 1h = Set Interrupt Mask
0	RXDONE	R	0h	Receive Data Interrupt 0h = Interrupt did not occur 1h = Set Interrupt Mask

23.3.7 ISET Register (Offset = 1040h) [Reset = 00000000h]

ISET is shown in Table 23-29.

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.

Table 23-29. ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	W	0h	
19	RESERVED	R	0h	Reserved
18	RESERVED	R	0h	Reserved
17	RESERVED	W	0h	
16	DMA_DONE_TX	W	0h	DMA Done on Event Channel TX 0h = Interrupt disabled 1h = Set Interrupt Mask
15	DMA_DONE_RX	W	0h	DMA Done on Event Channel RX 0h = Interrupt disabled 1h = Set Interrupt Mask
14	RESERVED	W	0h	
13	TIMEOUTB	W	0h	Timeout B Interrupt 0h = Writing 0 has no effect 1h = Set interrupt
12	TIMEOUTA	W	0h	Timeout A interrupt 0h = Writing 0 has no effect 1h = Set Interrupt Mask
11	PEC_RX_ERR	W	0h	RX Pec Error Interrupt 0h = Writing 0 has no effect 1h = Set interrupt
10	ARBLOST	W	0h	Arbitration Lost Interrupt 0h = Writing 0 has no effect 1h = Set Interrupt Mask
9	STOP	W	0h	STOP Detection Interrupt 0h = Writing 0 has no effect 1h = Set Interrupt Mask
8	START	W	0h	START Detection Interrupt 0h = Writing 0 has no effect 1h = Set Interrupt Mask
7	NACK	W	0h	Address/Data NACK Interrupt 0h = Writing 0 has no effect 1h = Set Interrupt Mask
6	RESERVED	W	0h	
5	TXEMPTY	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. 0h = Writing 0 has no effect 1h = Set Interrupt
4	RXFULL	W	0h	RXFIFO full event. 0h = Writing 0 has no effect 1h = Set Interrupt
3	TXTRG	W	0h	Transmit Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
2	RXTRG	W	0h	Receive Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask

Table 23-29. ISET Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	TXDONE	W	0h	Transmit Transaction completed Interrupt 0h = Writing 0 has no effect 1h = Set Interrupt Mask
0	RXDONE	W	0h	Receive Data Interrupt Signals that a byte has been received 0h = Writing 0 has no effect 1h = Set Interrupt Mask

23.3.8 ICLR Register (Offset = 1048h) [Reset = 00000000h]

ICLR is shown in Table 23-30.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

Table 23-30. ICLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	W	0h	
19	RESERVED	R	0h	Reserved
18	RESERVED	R	0h	Reserved
17	RESERVED	W	0h	
16	DMA_DONE_TX	W	0h	DMA Done on Event Channel TX 0h = Interrupt disabled 1h = Set Interrupt Mask
15	DMA_DONE_RX	W	0h	DMA Done on Event Channel RX 0h = Interrupt disabled 1h = Set Interrupt Mask
14	RESERVED	W	0h	
13	TIMEOUTB	W	0h	Timeout B Interrupt 0h = Writing 0 has no effect 1h = Clear Interrupt
12	TIMEOUTA	W	0h	Timeout A interrupt 0h = Writing 0 has no effect 1h = Set Interrupt Mask
11	PEC_RX_ERR	W	0h	RX Pec Error Interrupt 0h = Writing 0 has no effect 1h = Clear Interrupt
10	ARBLOST	W	0h	Arbitration Lost Interrupt 0h = Writing 0 has no effect 1h = Set Interrupt Mask
9	STOP	W	0h	STOP Detection Interrupt 0h = Writing 0 has no effect 1h = Set Interrupt Mask
8	START	W	0h	START Detection Interrupt 0h = Writing 0 has no effect 1h = Set Interrupt Mask
7	NACK	W	0h	Address/Data NACK Interrupt 0h = Writing 0 has no effect 1h = Set Interrupt Mask
6	RESERVED	W	0h	
5	TXEMPTY	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. 0h = Writing 0 has no effect 1h = Clear Interrupt
4	RXFULL	W	0h	RXFIFO full event. 0h = Writing 0 has no effect 1h = Clear Interrupt
3	TXTRG	W	0h	Transmit FIFO Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
2	RXTRG	W	0h	Receive Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask

Table 23-30. ICLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	TXDONE	W	0h	Transmit Transaction completed Interrupt 0h = Writing 0 has no effect 1h = Set Interrupt Mask
0	RXDONE	W	0h	Receive Data Interrupt Signals that a byte has been received 0h = Writing 0 has no effect 1h = Set Interrupt Mask

23.3.9 IMASK Register (Offset = 1058h) [Reset = 00000000h]

IMASK is shown in Table 23-31.

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 23-31. IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2	RXTRG	R/W	0h	Receive Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
1-0	RESERVED	R/W	0h	

23.3.10 RIS Register (Offset = 1060h) [Reset = 00000000h]

RIS is shown in Table 23-32.

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 23-32. RIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	
2	RXTRG	R	0h	Receive Trigger When FIFO is present, trigger when RX FIFO contains >= defined bytes when FIFO is not present, this indicates when buffer is full 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
1-0	RESERVED	R	0h	

23.3.11 MIS Register (Offset = 1068h) [Reset = 00000000h]

MIS is shown in Table 23-33.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Table 23-33. MIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	
2	RXTRG	R	0h	Receive Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
1-0	RESERVED	R	0h	

23.3.12 ISET Register (Offset = 1070h) [Reset = 00000000h]

ISET is shown in Table 23-34.

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 23-34. ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	W	0h	
2	RXTRG	W	0h	Receive Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
1-0	RESERVED	W	0h	

23.3.13 IMASK Register (Offset = 1088h) [Reset = 00000000h]

IMASK is shown in Table 23-35.

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 23-35. IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	TXTRG	R/W	0h	Transmit Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
2-0	RESERVED	R/W	0h	

23.3.14 RIS Register (Offset = 1090h) [Reset = 00000000h]

RIS is shown in Table 23-36.

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 23-36. RIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3	TXTRG	R	0h	Transmit Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
2-0	RESERVED	R	0h	

23.3.15 MIS Register (Offset = 1098h) [Reset = 00000000h]

MIS is shown in Table 23-37.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Table 23-37. MIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3	TXTRG	R	0h	Transmit Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
2-0	RESERVED	R	0h	

23.3.16 ISET Register (Offset = 10A0h) [Reset = 00000000h]

ISET is shown in Table 23-38.

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 23-38. ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	W	0h	
3	TXTRG	W	0h	Transmit Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
2-0	RESERVED	W	0h	

23.3.17 INTCTL Register (Offset = 10E4h) [Reset = 00000000h]

INTCTL is shown in Table 23-39.

Return to the Summary Table.

Interrupt control register

Table 23-39. INTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	W	0h	
0	INTEVAL	W	0h	Writing a 1 to this field re-evaluates the interrupt sources. 0h = The interrupt or event line is disabled. 1h = The interrupt or event line is in software mode. Software must clear the RIS.

23.3.18 CTR Register (Offset = 1100h) [Reset = 00000000h]

CTR is shown in Table 23-40.

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 Target Address (TA) 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 RXDATA 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 23-40. CTR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27-16	BLEN	R/W	0h	I2C transaction length This field contains the programmed length of bytes of the Transaction. 0h = Smallest value FFFh = Highest possible value
15-7	RESERVED	R/W	0h	
6	SUSPEND	R/W	0h	Suspend external communication 0h = Functional mode resumed 1h = External communication suspended
5	RD_ON_TXEMPTY	R/W	0h	Read on TX Empty 0h = No special behavior 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. 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	ACKOEN	R/W	0h	ACK overrride Enable 0h = No special behavior 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. Software needs to configure this bit to send the ACK or NACK. See field decoding in Table: MCTR Field decoding. 0h = The last received data byte of a transaction is not acknowledged automatically . 1h = The last received data byte of a transaction is acknowledged automatically .
2	STOP	R/W	0h	Generate STOP 0h = The controller does not generate the STOP condition. 1h = The controller generates the STOP condition. See field decoding in Table: MCTR Field decoding.

Table 23-40. CTR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	START	R/W	0h	Generate START 0h = The controller does not generate the START condition. 1h = The controller generates the START or repeated START condition. See field decoding in Table: MCTR Field decoding.
0	FRM_START	R/W	0h	Start Transfer 0h = In standard mode, this encoding means the Controller is unable to transmit or receive data. 1h = The Controller is able to transmit or receive data. See field decoding in Table: MCTR Field decoding.

23.3.19 CR Register (Offset = 1104h) [Reset = 00000000h]

CR is shown in Table 23-41.

Return to the Summary Table.

Configuration register

Table 23-41. CR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2	CLKSTRETCH	R/W	0h	Clock Stretching. This bit controls the support for clock stretching of the I2C bus. 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. 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	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. 0h = Disable MultiController mode. 1h = Enable MultiController mode.
0	ENABLE	R/W	0h	Enable 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. 0h = Disables operation. 1h = Enables operation.

23.3.20 SR Register (Offset = 1108h) [Reset = 00000000h]

SR is shown in Table 23-42.

Return to the Summary Table.

The status register indicates the state of the I2C bus controller.

Table 23-42. SR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	
27-16	BCNT	R	0h	Transaction Count This field contains the current count-down value of the transaction. 0h = Smallest value FFFh = Highest possible value
15	RESERVED	R	0h	
14	TXFF	R	0h	Transmit FIFO Full The meaning of this bit depends on the state of the FEN bit in the CTL0 register. 0h = The transmitter is not full. 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.
13	TXFE	R	1h	Transmit FIFO Empty The meaning of this bit depends on the state of the FEN bit in the CTL0 register. 0h = The transmitter has data to transmit. 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.
12	RXFF	R	0h	Receive FIFO Full The meaning of this bit depends on the state of the FEN bit in the CTL0 register. 0h = The receiver can receive data. 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.
11	RXFE	R	1h	Receive FIFO Empty The meaning of this bit depends on the state of the FEN bit in the CTL0 register. 0h = The receiver is not empty. 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.
10	TXCLR	R	0h	TX FIFO Clear Status 0h = FIFO is not cleared 1h = FIFO clear is complete
9	RXCLR	R	0h	RX FIFO Clear Status 0h = FIFO is not cleared 1h = FIFO clear is complete
8-7	RESERVED	R	0h	
6	BUSBSY	R	0h	I2C Bus is Busy Controller State Machine will wait until this bit to be 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 CTR register to start the transaction so that if SCL goes low it will trigger the BUSBSY. 0h = The I2C bus is idle. 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. 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 0h = The I2C controller is not idle. 1h = The I2C controller is idle.

Table 23-42. SR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	ARBLST	R	0h	Arbitration Lost 0h = The I2C controller won arbitration. 1h = The I2C controller lost arbitration.
3	DATACK	R	0h	Acknowledge Data 0h = The transmitted data was acknowledged 1h = The transmitted data was not acknowledged.
2	ADRACK	R	0h	Acknowledge Address 0h = The transmitted address was acknowledged 1h = The transmitted address was not acknowledged.
1	ERR	R	0h	Error The error can be from the Target address not being acknowledged or the transmit data not being acknowledged. 0h = No error was detected on the last operation. 1h = An error occurred on the last operation.
0	BUSY	R	0h	FSM Busy 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. 0h = The controller is idle. 1h = The controller is busy.

23.3.21 IFLS Register (Offset = 110Ch) [Reset = 00000022h]

IFLS is shown in Table 23-43.

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.

Table 23-43. IFLS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	
7	RXCLR	R/W	0h	RX FIFO CLEAR. Setting this bit will clear the RX FIFO contents. 0h = Disable FIFO clear 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. 1h = RX FIFO >= 1/4 full 2h = RX FIFO >= 1/2 full (default) 3h = RX FIFO >= 3/4 full 4h = Opposite of empty 5h = RX FIFO is full 6h = RX_FIFO >= (MAX_FIFO_LEN -1) 7h = RX_FIFO <= 1
3	TXCLR	R/W	0h	TX FIFO CLEAR. Setting this bit will clear the TX FIFO contents. 0h = Disable FIFO clear 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. 1h = TX FIFO <= 3/4 empty 2h = TX FIFO <= 1/2 empty (default) 3h = TX FIFO <= 1/4 empty 4h = Opposite of full 5h = TX FIFO is empty 6h = TX FIFO <= 1 7h = TX_FIFO >= (MAX_FIFO_LEN -1)

23.3.22 TPR Register (Offset = 1110h) [Reset = 00000001h]

TPR is shown in Table 23-44.

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 23-44. TPR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R/W	0h	
6-0	TPR	R/W	1h	Timer Period This field is used in the equation to configure SCL_PERIOD : SCL_PERIOD = (1 + TPR) × (SCL_LP + SCL_HP) × 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. 0h = Smallest value 7Fh = Highest possible value

23.3.23 GFCTL Register (Offset = 1118h) [Reset = 00000000h]

GFCTL is shown in Table 23-45.

Return to the Summary Table.

This register controls the glitch filter on the SCL and SDA lines

Table 23-45. GFCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R/W	0h	
8	AGFEN	R/W	0h	Analog Glitch Suppression Enable 0h = Analog Glitch Filter disable 1h = Analog Glitch Filter enable
7-3	RESERVED	R/W	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) 0h = Bypass 1h = 1 clock 2h = 2 clocks 3h = 3 clocks 4h = 4 clocks 5h = 8 clocks 6h = 16 clocks 7h = 31 clocks

23.3.24 BMON Register (Offset = 111Ch) [Reset = 00000003h]

BMON is shown in Table 23-46.

Return to the Summary Table.

This register is used to determine the SCL and SDA signal status.

Table 23-46. BMON Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1	SDA	R	1h	I2C SDA Status 0h = The I2CSDA signal is low. 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 0h = The I2CSCL signal is low. 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.

23.3.25 TXDATA Register (Offset = 1120h) [Reset = 00000000h]

TXDATA is shown in Table 23-47.

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).

Table 23-47. TXDATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	W	0h	
7-0	DATA	W	0h	Transmit Data This byte contains the data to be transferred during the next transaction. 0h = Smallest value FFh = Highest possible value

23.3.26 RXDATA Register (Offset = 1124h) [Reset = 00000000h]

RXDATA is shown in Table 23-48.

Return to the Summary Table.

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 23-48. RXDATA Register 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. 0h = Smallest value FFh = Highest possible value

23.3.27 PECSR Register (Offset = 1128h) [Reset = 00000000h]

PECSR is shown in Table 23-49.

Return to the Summary Table.

PEC Status Register

Table 23-49. 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. 0h = Indicates PEC check error did not occurr in the transaction that occurred before the last Stop 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. 0h = Indicates PEC was not checked in the transaction that occurred before the last Stop 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. 0h = Minimum Value 1FFh = Maximum Value

23.3.28 TA Register (Offset = 114Ch) [Reset = 00000000h]

TA is shown in Table 23-50.

Return to the Summary Table.

Target Address Register

Table 23-50. TA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	
15	MODE	R/W	0h	This bit selects the adressing mode to be used When 0, 7-bit addressing is used. When 1, 10-bit addressing is used. 0h = 7-bit addressing mode 1h = 10-bit addressing mode
14-11	RESERVED	R/W	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 MSA.MODE bit, the top 3 bits are don't care 0h = Smallest value 3FFh = Highest possible value
0	DIR	R/W	0h	Receive/Send The DIR bit specifies if the next operation is a Receive (High) or Transmit (Low). 0h = Transmit 1h = Receive 0h = in transmit mode. 1h = is in receive mode.

23.3.29 TIMEOUT_CNT Register (Offset = 1150h) [Reset = 00020002h]

TIMEOUT_CNT is shown in Table 23-51.

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 23-51. 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 0h = Smallest Value 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 0h = Smallest Value FFh = Highest possible value

23.3.30 TIMEOUT_CTL Register (Offset = 1154h) [Reset = 00020002h]

TIMEOUT_CTL is shown in Table 23-52.

Return to the Summary Table.

This register contains controls for Timeout Counters A and B

Table 23-52. TIMEOUT_CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	TCNTBEN	R/W	0h	Timeout Counter B Enable 0h = Disable Timeout Counter B 1h = Enable Timeout Counter B
30-24	RESERVED	R/W	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. Each count is equal to 1* clock period. For example, with 10MHz functional clock one timeout period will be equal to1*100ns. 0h = Smallest possible value FFh = Highest possible value
15	TCNTAEN	R/W	0h	Timeout Counter A Enable 0h = Disable Timeout Counter B 1h = Enable Timeout Counter B
14-8	RESERVED	R/W	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. 0h = Smallest Value FFh = Highest possible value

23.3.31 PECCTL Register (Offset = 1158h) [Reset = 00000000h]

PECCTL is shown in Table 23-53.

Return to the Summary Table.

PEC Control Register

Table 23-53. PECCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R/W	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 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. 0h = PEC is disabled 1h = PEC is enabled
11-9	RESERVED	R/W	0h	
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 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. 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. 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. Note that any write to the I2PECCTL Register will clear the current PEC Byte Count in the State Machine. 0h = Minimum Value 1FFh = Maximum Value

23.4 UNICOMMI2CT Registers

Table 23-54 lists the memory-mapped registers for the UNICOMMI2CT registers. All register offset addresses not listed in Table 23-54 should be considered as reserved locations and the register contents should not be modified.

Table 23-54. UNICOMMI2CT Registers

Offset	Acronym	Register Name	Section
1000h	CLKDIV	Clock Divider	Section 23.4.1
1008h	CLKSEL	Clock Select for Ultra Low Power peripherals	Section 23.4.2
1020h	IIDX	Interrupt index	Section 23.4.3
1028h	IMASK	Interrupt mask	Section 23.4.4
1030h	RIS	Raw interrupt status	Section 23.4.5
1038h	MIS	Masked interrupt status	Section 23.4.6
1040h	ISET	Interrupt set	Section 23.4.7
1048h	ICLR	Interrupt clear	Section 23.4.8
1058h	IMASK	Interrupt mask	Section 23.4.9
1060h	RIS	Raw interrupt status	Section 23.4.10
1068h	MIS	Masked interrupt status	Section 23.4.11
1070h	ISET	Interrupt set	Section 23.4.12
1088h	IMASK	Interrupt mask	Section 23.4.13
1090h	RIS	Raw interrupt status	Section 23.4.14
1098h	MIS	Masked interrupt status	Section 23.4.15
10A0h	ISET	Interrupt set	Section 23.4.16
10E4h	INTCTL	Interrupt control register	Section 23.4.17
1100h	CTR	I2C Target Control Register	Section 23.4.18
1104h	ACKCTL	I2C Target ACK Control	Section 23.4.19
1108h	SR	Status Register	Section 23.4.20
110Ch	IFLS	Interrupt FIFO Level Select Register	Section 23.4.21
1118h	GFCTL	I2C Glitch Filter Control	Section 23.4.22
1120h	TXDATA	I2C TXData	Section 23.4.23
1124h	RXDATA	I2C RXData	Section 23.4.24
1128h	PECSR	PEC status register	Section 23.4.25
1148h	OAR2	Own Address 2	Section 23.4.26
114Ch	OAR	I2C Own Address	Section 23.4.27
1150h	TIMEOUT_CNT	I2C Timeout Count Register	Section 23.4.28
1154h	TIMEOUT_CTL	I2C Timeout Count Control Register	Section 23.4.29
1158h	PECCTL	I2C PEC control register	Section 23.4.30

Complex bit access types are encoded to fit into small table cells. Table 23-55 shows the codes that are used for access types in this section.

Table 23-55. UNICOMMI2CT Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		

Table 23-55. UNICOMMI2CT Access Type Codes (continued)

Access Type	Code	Description
-n		Value after reset or the default value

23.4.1 CLKDIV Register (Offset = 1000h) [Reset = 00000000h]

CLKDIV is shown in Table 23-56.

Return to the Summary Table.

This register is used to specify module-specific divide ratio of the functional clock

Table 23-56. CLKDIV Register 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 Division factor 0 : DIV_BY_1 1 : DIV_BY_2 .... 63: DIV_BY_64 0h = Do not divide clock source 1h = Divide clock source by 2 2h = Divide clock source by 3 3h = Divide clock source by 4 4h = Divide clock source by 5 5h = Divide clock source by 6 6h = Divide clock source by 7 7h = Divide clock source by 8 3Ch = Divide clock source by 8 3Dh = Divide clock source by 8 3Eh = Divide clock source by 8 3Fh = Divide clock source by 8

23.4.2 CLKSEL Register (Offset = 1008h) [Reset = 00000000h]

CLKSEL is shown in Table 23-57.

Return to the Summary Table.

Clock source selection for peripherals

Table 23-57. CLKSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11	ASYNC_PLL_SEL	R/W	0h	Asynchronous PLL selected as source 0h = Does not select this clock as a source 1h = Select this clock as a source
10	ASYNC_HFCLK_SEL	R/W	0h	Asynchronous HFCLK selected as source 0h = Does not select this clock as a source 1h = Select this clock as a source
9	ASYNC_SYSCLK_SEL	R/W	0h	Asynchronous sysclk selected as source 0h = Does not select this clock as a source 1h = Select this clock as a source
8-4	RESERVED	R/W	0h	
3	BUSCLK_SEL	R/W	0h	Selects BUS CLK as clock source if enabled 0h = Does not select this clock as a source 1h = Select this clock as a source
2	MFCLK_SEL	R/W	0h	Selects MFCLK as clock source if enabled 0h = Does not select this clock as a source 1h = Select this clock as a source
1-0	RESERVED	R/W	0h	

23.4.3 IIDX Register (Offset = 1020h) [Reset = 00000000h]

IIDX is shown in Table 23-58.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index.

Table 23-58. 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 00h = No interrupt pending 1h = Receive Done Flag 2h = Transmit Done Flag 3h = receive FIFO Trigger Level 4h = transmit FIFO Trigger level 5h = RX FIFO FULL Event/interrupt pending 6h = Transmit FIFO/Buffer Empty Event/interrupt pending 7h = Target TX FIFO underflow 8h = Target RX FIFO overflow event 9h = General Call Event Ah = Start Event Bh = Stop Event Ch = PEC receive error event Dh = Timeout A Event Eh = Timeout B Event 10h = DMA DONE on Channel RX 11h = DMA DONE on Channel TX 12h = Arbitration Lost 13h = DMA PRE IRQ INTERRUPT 14h = DMA PRE IRQ INTERRUPT

23.4.4 IMASK Register (Offset = 1028h) [Reset = 00000000h]

IMASK is shown in Table 23-59.

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 23-59. IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R/W	0h	
19	RESERVED	R	0h	Reserved
18	RESERVED	R	0h	Reserved
17	ARBLOST	R/W	0h	Arbitration Lost Interrupt 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
16	DMA_DONE_TX	R/W	0h	DMA Done on Event Channel TX 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
15	DMA_DONE_RX	R/W	0h	DMA Done on Event Channel RX 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
14	RESERVED	R/W	0h	
13	TIMEOUTB	R/W	0h	Timeout B Interrupt 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
12	TIMEOUTA	R/W	0h	Timeout A Interrupt 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
11	PEC_RX_ERR	R/W	0h	RX Pec Error Interrupt 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
10	STOP	R/W	0h	Stop Condition Interrupt 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
9	START	R/W	0h	Start Condition Interrupt 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
8	GENCALL	R/W	0h	General Call Interrupt 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
7	RX_OVFL	R/W	0h	Target RX FIFO overflow 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
6	TX_UNFL	R/W	0h	Target TX FIFO underflow 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
5	TXEMPTY	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. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
4	RXFULL	R/W	0h	RXFIFO full event. This interrupt is set if an Target RX FIFO is full. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
3	TXTRG	R/W	0h	Transmit Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask

Table 23-59. IMASK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	RXTRG	R/W	0h	Receive Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
1	TXDONE	R/W	0h	Transmit Transaction completed Interrupt 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
0	RXDONE	R/W	0h	Receive Data Interrupt Signals that a byte has been received 0h = Clear Interrupt Mask 1h = Set Interrupt Mask

23.4.5 RIS Register (Offset = 1030h) [Reset = 00000000h]

RIS is shown in Table 23-60.

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 23-60. RIS Register 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	Arbitration Lost Interrupt 0h = Interrupt did not occur 1h = Set Interrupt Mask
16	DMA_DONE_TX	R	0h	DMA Done on Event Channel TX 0h = Clear interrupt 1h = Set interrupt
15	DMA_DONE_RX	R	0h	DMA Done on Event Channel RX 0h = Clear interrupt 1h = Set interrupt
14	RESERVED	R	0h	
13	TIMEOUTB	R	0h	Timeout B Interrupt 0h = Interrupt did not occur 1h = Interrupt occured
12	TIMEOUTA	R	0h	Timeout A Interrupt 0h = Interrupt did not occur 1h = Set Interrupt Mask
11	PEC_RX_ERR	R	0h	RX Pec Error Interrupt 0h = Interrupt did not occur 1h = Interrupt ocuured
10	STOP	R	0h	Stop Condition Interrupt: set on a STOP condition if this target was being addressed 0h = Clear Interrupt 1h = Interrupt is set when target is addressed and STOP condition is received
9	START	R	0h	Start Condition Interrupt: is set after a START condition is received and this target is addressed 0h = Clear interrupt 1h = Set when a START condition is receivied and address matches target's address
8	GENCALL	R	0h	General Call Interrupt: set when a general call is received 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
7	RX_OVFL	R	0h	Receive FIFO overflow 0h = Interrupt did not occur 1h = Interrupt Occured
6	TX_UNFL	R	0h	Transmit FIFO underflow 0h = Interrupt did not occur 1h = Interrupt occured
5	TXEMPTY	R	0h	Transmit FIFO Empty Interrupt. This interrupt is set if all data in the Transmit FIFO have been shifted out and FSM goes into idle mode. 0h = Interrupt did not occur 1h = Interrupt occured
4	RXFULL	R	0h	RXFIFO full event. This interrupt is set if an RX FIFO is full. 0h = Clear Interrupt Mask 1h = Interrupt occured

Table 23-60. RIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	TXTRG	R	0h	Transmit Trigger When FIFO is present, triggered as per IFLS settings When FIFO is not present, trigger when transmit buffer is empty 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
2	RXTRG	R	0h	Receive Trigger When FIFO is present, triggered as per IFLS settings when FIFO is not present, this indicates when buffer is full 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
1	TXDONE	R	0h	Transmission Done Interrupt: Set after a byte is transmitted 0h = Interrupt did not occur 1h = Set Interrupt Mask
0	RXDONE	R	0h	Receive Done Interrupt: Set after a byte is received 0h = Interrupt did not occur 1h = Set Interrupt Mask

23.4.6 MIS Register (Offset = 1038h) [Reset = 00000000h]

MIS is shown in Table 23-61.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Table 23-61. MIS Register 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	Arbitration Lost Interrupt 0h = Interrupt did not occur 1h = Set Interrupt Mask
16	DMA_DONE_TX	R	0h	DMA Done on Event Channel TX 0h = Clear MIS 1h = Set MIS
15	DMA_DONE_RX	R	0h	DMA Done on Event Channel RX 0h = Clear MIS 1h = Set MIS
14	RESERVED	R	0h	
13	TIMEOUTB	R	0h	Timeout B Interrupt 0h = Clear interrupt mask 1h = Set interrupt mask
12	TIMEOUTA	R	0h	Timeout A Interrupt 0h = Interrupt did not occur 1h = Set Interrupt Mask
11	PEC_RX_ERR	R	0h	Target RX Pec Error Interrupt 0h = Clear interrupt mask 1h = Set interrupt mask
10	STOP	R	0h	STOP Detection Interrupt 0h = Clear MIS 1h = Set MIS
9	START	R	0h	START Detection Interrupt 0h = Clear MIS 1h = Set MIS
8	GENCALL	R	0h	General Call Interrupt 0h = Interrupt did not occur 1h = Set Interrupt Mask
7	RX_OVFL	R	0h	Target RX FIFO overflow 0h = Clear interrupt mask 1h = Set interrupt mask
6	TX_UNFL	R	0h	Target TX FIFO underflow 0h = Clear interrupt mask 1h = Set interrupt mask
5	TXEMPTY	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. 0h = Interrupt did not occur 1h = Interrupt occured
4	RXFULL	R	0h	RXFIFO full event. This interrupt is set if an RX FIFO is full. 0h = Clear Interrupt Mask 1h = Interrupt occured
3	TXTRG	R	0h	Transmit Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask

Table 23-61. MIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	RXTRG	R	0h	Receive Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
1	TXDONE	R	0h	Transmit Transaction completed Interrupt 0h = Interrupt did not occur 1h = Set Interrupt Mask
0	RXDONE	R	0h	Receive Data Interrupt Signals that a byte has been received 0h = Interrupt did not occur 1h = Set Interrupt Mask

23.4.7 ISET Register (Offset = 1040h) [Reset = 00000000h]

ISET is shown in Table 23-62.

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 23-62. ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	W	0h	
19	RESERVED	R	0h	Reserved
18	RESERVED	R	0h	Reserved
17	ARBLOST	W	0h	Arbitration Lost Interrupt 0h = Writing 0 has no effect 1h = Set Interrupt Mask
16	DMA_DONE_TX	W	0h	DMA Done on Event Channel TX 0h = Writing 0 has no effect 1h = Set interrupt
15	DMA_DONE_RX	W	0h	DMA Done on Event Channel RX 0h = Writing 0 has no effect 1h = Set interrupt
14	RESERVED	W	0h	
13	TIMEOUTB	W	0h	Timeout B Interrupt 0h = Writing 0 has no effect 1h = Set interrupt
12	TIMEOUTA	W	0h	Timeout A interrupt 0h = Writing 0 has no effect 1h = Set Interrupt Mask
11	PEC_RX_ERR	W	0h	Target RX Pec Error Interrupt 0h = Writing 0 has no effect 1h = Set interrupt
10	STOP	W	0h	Stop Condition Interrupt 0h = Writing 0 has no effect 1h = Set interrupt
9	START	W	0h	Start Condition Interrupt 0h = Writing 0 has no effect 1h = Set interrupt
8	GENCALL	W	0h	General Call Interrupt 0h = Writing 0 has no effect 1h = Set Interrupt Mask
7	RX_OVFL	W	0h	Target RX FIFO overflow 0h = Writing 0 has no effect 1h = Set interrupt
6	TX_UNFL	W	0h	Target TX FIFO underflow 0h = Writing 0 has no effect 1h = Set interrupt
5	TXEMPTY	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. 0h = Writing 0 has no effect 1h = Set Interrupt
4	RXFULL	W	0h	RXFIFO full event. This interrupt is set if an RX FIFO is full. 0h = Clear Interrupt Mask 1h = Set Interrupt

Table 23-62. ISET Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	TXTRG	W	0h	Transmit Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
2	RXTRG	W	0h	Receive Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
1	TXDONE	W	0h	Target Transmit Transaction completed Interrupt 0h = Writing 0 has no effect 1h = Set Interrupt Mask
0	RXDONE	W	0h	Target Receive Data Interrupt Signals that a byte has been received 0h = Writing 0 has no effect 1h = Set Interrupt Mask

23.4.8 ICLR Register (Offset = 1048h) [Reset = 00000000h]

ICLR is shown in Table 23-63.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

Table 23-63. ICLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	W	0h	
19	RESERVED	R	0h	Reserved
18	RESERVED	R	0h	Reserved
17	ARBLOST	W	0h	Arbitration Lost Interrupt 0h = Writing 0 has no effect 1h = Set Interrupt Mask
16	DMA_DONE_TX	W	0h	DMA Done on Event Channel TX 0h = Writing 0 has no effect 1h = Clear interrupt
15	DMA_DONE_RX	W	0h	DMA Done on Event Channel RX 0h = Writing 0 has no effect 1h = Clear interrupt
14	RESERVED	W	0h	
13	TIMEOUTB	W	0h	Timeout B Interrupt 0h = Writing 0 has no effect 1h = Clear Interrupt
12	TIMEOUTA	W	0h	Timeout A interrupt 0h = Writing 0 has no effect 1h = Set Interrupt Mask
11	PEC_RX_ERR	W	0h	Target RX Pec Error Interrupt 0h = Writing 0 has no effect 1h = Clear Interrupt
10	STOP	W	0h	Target STOP Detection Interrupt 0h = Writing 0 has no effect 1h = Clear interrupt
9	START	W	0h	Target START Detection Interrupt 0h = Writing 0 has no effect 1h = Clear interrupt
8	GENCALL	W	0h	General Call Interrupt 0h = Writing 0 has no effect 1h = Set Interrupt Mask
7	RX_OVFL	W	0h	Target RX FIFO overflow 0h = Writing 0 has no effect 1h = Clear Interrupt
6	TX_UNFL	W	0h	Target TX FIFO underflow 0h = Writing 0 has no effect 1h = Clear Interrupt
5	TXEMPTY	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. 0h = Writing 0 has no effect 1h = Clear Interrupt
4	RXFULL	W	0h	RXFIFO full event. This interrupt is set if an RX FIFO is full. 0h = Clear Interrupt Mask 1h = Clear Interrupt
3	TXTRG	W	0h	Transmit Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask

Table 23-63. ICLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	RXTRG	W	0h	Receive Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
1	TXDONE	W	0h	Target Transmit Transaction completed Interrupt 0h = Writing 0 has no effect 1h = Set Interrupt Mask
0	RXDONE	W	0h	Target Receive Data Interrupt Signals that a byte has been received 0h = Writing 0 has no effect 1h = Set Interrupt Mask

23.4.9 IMASK Register (Offset = 1058h) [Reset = 00000000h]

IMASK is shown in Table 23-64.

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 23-64. IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2	RXTRG	R/W	0h	Receive Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
1-0	RESERVED	R/W	0h	

23.4.10 RIS Register (Offset = 1060h) [Reset = 00000000h]

RIS is shown in Table 23-65.

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 23-65. RIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	
2	RXTRG	R	0h	Receive Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
1-0	RESERVED	R	0h	

23.4.11 MIS Register (Offset = 1068h) [Reset = 00000000h]

MIS is shown in Table 23-66.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Table 23-66. MIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	
2	RXTRG	R	0h	Receive Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
1-0	RESERVED	R	0h	

23.4.12 ISET Register (Offset = 1070h) [Reset = 00000000h]

ISET is shown in Table 23-67.

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 23-67. ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	W	0h	
2	RXTRG	W	0h	Receive Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
1-0	RESERVED	W	0h	

23.4.13 IMASK Register (Offset = 1088h) [Reset = 00000000h]

IMASK is shown in Table 23-68.

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 23-68. IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	TXTRG	R/W	0h	Transmit Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
2-0	RESERVED	R/W	0h	

23.4.14 RIS Register (Offset = 1090h) [Reset = 00000000h]

RIS is shown in Table 23-69.

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 23-69. RIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3	TXTRG	R	0h	Transmit Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
2-0	RESERVED	R	0h	

23.4.15 MIS Register (Offset = 1098h) [Reset = 00000000h]

MIS is shown in Table 23-70.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Table 23-70. MIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3	TXTRG	R	0h	Transmit Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
2-0	RESERVED	R	0h	

23.4.16 ISET Register (Offset = 10A0h) [Reset = 00000000h]

ISET is shown in Table 23-71.

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 23-71. ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	W	0h	
3	TXTRG	W	0h	Transmit Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
2-0	RESERVED	W	0h	

23.4.17 INTCTL Register (Offset = 10E4h) [Reset = 00000000h]

INTCTL is shown in Table 23-72.

Return to the Summary Table.

Interrupt control register

Table 23-72. INTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	W	0h	
0	INTEVAL	W	0h	Writing a 1 to this field re-evaluates the interrupt sources. 0h = The interrupt or event line is disabled. 1h = The interrupt or event line is in software mode. Software must clear the RIS.

23.4.18 CTR Register (Offset = 1100h) [Reset = 00300404h]

CTR is shown in Table 23-73.

Return to the Summary Table.

Control Register

Table 23-73. CTR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	RESERVED	R/W	0h	
21	WUEN	R/W	1h	Target Wakeup Enable 0h = When 0, the Target is not allowed to clock stretch on START detection 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
20	CLKSTRETCH	R/W	1h	Clock Stretch Enable 0h = clock stretching is disabled 1h = clock stretching is enabled
19-12	RESERVED	R/W	0h	
11	SUSPEND	R/W	0h	Suspend external communication 0h = Functional mode resumed 1h = External communication suspended
10	RESERVED	R/W	0h	
9	EN_DEFDEVADR	R/W	0h	Enable Default device address 0h = When this bit is 0, the default device address is not matched. NOTE: it may still be matched if programmed inside SOAR/SOAR2. 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 0h = When this bit is 0, the alert response address is not matched. NOTE: it may still be matched if programmed inside SOAR/SOAR2 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 0h = When this bit is 0, the default host address is not matched NOTE: it may still be matched if programmed inside SOAR/SOAR2 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 0h = When 0, RIS:SRXFULL will be set when only the Target RX FIFO is full. This allows the SRXFULL 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. 1h = When 1, RIS:SRXFULL 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. 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. 0h = When 0, the TX FIFO empty signal to the Target State Machine indicates that the TX FIFO is empty. 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.

Table 23-73. CTR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	TXTRIG_TXMODE	R/W	0h	Tx Trigger when Target FSM is in Tx Mode 0h = No special behavior 1h = When 1, RIS:TXTRG 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. When cleared RIS:TXTRG will be set when the Target 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 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 0h = When 0, RIS:TXEMPTY will be set when only the Target TX FIFO is empty. This allows the TXEMPTY interrupt to be used to indicate that the I2C bus is being clock stretched and that Target TX data is required. 1h = When 1, RIS:TXEMPTY 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	RESERVED	R/W	0h	
1	GENCALL	R/W	0h	General call response enable Modify only when UCSWRST = 1. 0b = Do not respond to a general call 1b = Respond to a general call 0h = Do not respond to a general call 1h = Respond to a general call
0	ENABLE	R/W	0h	Setting this bit enables the module. 0h = Disables module operation. 1h = Enables module operation.

23.4.19 ACKCTL Register (Offset = 1104h) [Reset = 00000000h]

ACKCTL is shown in Table 23-74.

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 23-74. ACKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R/W	0h	
4	ACKOEN_ON_PECDONE	R/W	0h	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. 0h = No special behavior 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 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 State Machine 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. 0h = No special behavior 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 Target_SACKCTL.
2	ACKOEN_ON_START	R/W	0h	When set this bit will automatically enable target override acknowlede following a Start Condition. Status bit ACKOEN will reflect a value of '1' when automatic override enable is used. 0h = No special behavior 1h = When set this bit will automatically turn on the Target ACKOEN field following a Start Condition.
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. 0h = An ACK is sent indicating valid data or command. 1h = A NACK is sent indicating invalid data or command.
0	ACKOEN	R/W	0h	I2C Target ACK Override Enable Read SR.ACKOEN to check current status of this bit 0h = A response in not provided. 1h = An ACK or NACK is sent according to the value written to the ACKOVAL bit.

23.4.20 SR Register (Offset = 1108h) [Reset = 00000000h]

SR is shown in Table 23-75.

Return to the Summary Table.

Status register

Table 23-75. SR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-16	ADDRMATCH	R	0h	Indicates the address for which Target address match happened 0h = Minimum Value 3FFh = Maximum Value
15	ACKOEN	R	0h	Status of ACK Override Enable 0h = ACK override is disabled 1h = ACK override was enabled in design
14	TXFF	R	0h	Transmit FIFO Full The meaning of this bit depends on the state of the FEN bit in the CTL0 register. 0h = The transmitter is not full. 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.
13	TXFE	R	1h	Transmit FIFO Empty The meaning of this bit depends on the state of the FEN bit in the CTL0 register. 0h = The transmitter has data to transmit. 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.
12	RXFF	R	0h	Receive FIFO Full The meaning of this bit depends on the state of the FEN bit in the CTL0 register. 0h = The receiver can receive data. 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.
11	RXFE	R	1h	Receive FIFO Empty The meaning of this bit depends on the state of the FEN bit in the CTL0 register. 0h = The receiver is not empty. 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.
10	TXCLR	R	0h	TX FIFO Clear Status 0h = FIFO is not cleared 1h = FIFO clear is complete
9	RXCLR	R	0h	RX FIFO Clear Status 0h = FIFO is not cleared 1h = FIFO clear is complete
8	STALE_TXFIFO	R	0h	Stale Tx FIFO 0h = Tx FIFO is not stale 1h = The TX FIFO is stale. This occurs when the TX FIFO was not emptied during the previous I2C transaction.
7	TXMODE	R	0h	FSM is in TX MODE 0h = State Machine is not in TX_DATA, TX_WAIT, TX_ACK or ADDR_ACK state with the bus direction set to read. 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 is busy 0h = The I2C Bus is not busy 1h = The I2C Bus is busy. This is cleared on a timeout.
5	QCMDRW	R	0h	Quick Command Read / Write 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 0h = Quick command was a write 1h = Quick command was a read

Table 23-75. SR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
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 0h = The last transaction was a normal transaction or a transaction has not occurred. 1h = The last transaction was a Quick Command transaction.
3	OAR2SEL	R	0h	OAR2 Address Matched This bit gets reevaluated after every address comparison. 0h = Either the OAR2 address is not matched or the match is in legacy mode. 1h = OAR2 address matched and ACKed by the Target.
2	RXMODE	R	0h	Target FSM is in Rx MODE 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. 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 0h = No outstanding transmit request. 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 0h = No outstanding receive data. 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).

23.4.21 IFLS Register (Offset = 110Ch) [Reset = 00000022h]

IFLS is shown in Table 23-76.

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.

Table 23-76. IFLS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	
7	RXCLR	R/W	0h	RX FIFO CLEAR. Setting this bit will clear the RX FIFO contents. 0h = Disable FIFO clear 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. 1h = RX FIFO >= 1/4 full 2h = RX FIFO >= 1/2 full (default) 3h = RX FIFO >= 3/4 full 4h = Opposite of empty 5h = RX FIFO is full 6h = RX_FIFO >= (MAX_FIFO_LEN -1) 7h = RX_FIFO <= 1
3	TXCLR	R/W	0h	TX FIFO CLEAR. Setting this bit will clear the TX FIFO contents. 0h = Disable FIFO clear 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. 1h = TX FIFO <= 3/4 empty 2h = TX FIFO <= 1/2 empty (default) 3h = TX FIFO <= 1/4 empty 4h = Opposite of full 5h = TX FIFO is empty 6h = TX FIFO <= 1 7h = TX_FIFO >= (MAX_FIFO_LEN -1)

23.4.22 GFCTL Register (Offset = 1118h) [Reset = 00000000h]

GFCTL is shown in Table 23-77.

Return to the Summary Table.

This register controls the glitch filter on the SCL and SDA lines

Table 23-77. GFCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R/W	0h	
8	AGFEN	R/W	0h	Analog Glitch Suppression Enable 0h = Analog Glitch Filter disable 1h = Analog Glitch Filter enable
7-3	RESERVED	R/W	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) 0h = Bypass 1h = 1 clock 2h = 2 clocks 3h = 3 clocks 4h = 4 clocks 5h = 8 clocks 6h = 16 clocks 7h = 31 clocks

23.4.23 TXDATA Register (Offset = 1120h) [Reset = 00000000h]

TXDATA is shown in Table 23-78.

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).

Table 23-78. TXDATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	W	0h	
7-0	DATA	W	0h	Transmit Data This byte contains the data to be transferred during the next transaction. 0h = Smallest value FFh = Highest possible value

23.4.24 RXDATA Register (Offset = 1124h) [Reset = 00000000h]

RXDATA is shown in Table 23-79.

Return to the Summary Table.

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 23-79. RXDATA Register 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. 0h = Smallest value FFh = Highest possible value

23.4.25 PECSR Register (Offset = 1128h) [Reset = 00000000h]

PECSR is shown in Table 23-80.

Return to the Summary Table.

PEC Status Register

Table 23-80. 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. 0h = Indicates PEC check error did not occurr in the transaction that occurred before the last Stop 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. 0h = Indicates PEC was not checked in the transaction that occurred before the last Stop 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 State Machine. 0h = Minimum Value 1FFh = Maximum Value

23.4.26 OAR2 Register (Offset = 1148h) [Reset = 00000000h]

OAR2 is shown in Table 23-81.

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 23-81. OAR2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	RESERVED	R/W	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 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. 0h = Minimum Value 7Fh = Maximum Value
15-8	RESERVED	R/W	0h	
7	OAR2EN	R/W	0h	Own Address 2 Enable 0h = The alternate address is disabled. 1h = Enables the use of the alternate address in the OAR2 field.
6-0	OAR2	R/W	0h	Own Address 2This field specifies the alternate OAR2 address. 0h = Smallest value 7Fh = Highest possible value

23.4.27 OAR Register (Offset = 114Ch) [Reset = 00004000h]

OAR is shown in Table 23-82.

Return to the Summary Table.

This register consists of seven address bits that identify the I2C device on the I2C bus.

Table 23-82. OAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	
15	MODE	R/W	0h	This bit selects the adressing mode to be used. When 0, 7-bit addressing is used. When 1, 10-bit addressing is used. 0h = Enable 7-bit addressing 1h = Enable 10-bit addressing
14	OAREN	R/W	1h	Own Address Enable 0h = Disable OAR address 1h = Enable OAR address
13-10	RESERVED	R/W	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 I2CSOAR.MODE bit, the top 3 bits are don't care 0h = Smallest value 3FFh = Highest possible value

23.4.28 TIMEOUT_CNT Register (Offset = 1150h) [Reset = 00020002h]

TIMEOUT_CNT is shown in Table 23-83.

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 23-83. 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 0h = Smallest Value 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 0h = Smallest Value FFh = Highest possible value

23.4.29 TIMEOUT_CTL Register (Offset = 1154h) [Reset = 00020002h]

TIMEOUT_CTL is shown in Table 23-84.

Return to the Summary Table.

This register contains controls for Timeout Counters A and B

Table 23-84. TIMEOUT_CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	TCNTBEN	R/W	0h	Timeout Counter B Enable 0h = Disable Timeout Counter B 1h = Enable Timeout Counter B
30-24	RESERVED	R/W	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. Each count is equal to 1* clock period. For example, with 10MHz functional clock one timeout period will be equal to1*100ns. 0h = Smallest possible value FFh = Highest possible value
15	TCNTAEN	R/W	0h	Timeout Counter A Enable 0h = Disable Timeout Counter B 1h = Enable Timeout Counter B
14-8	RESERVED	R/W	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. 0h = Smallest Value FFh = Highest possible value

23.4.30 PECCTL Register (Offset = 1158h) [Reset = 00000000h]

PECCTL is shown in Table 23-85.

Return to the Summary Table.

PEC Control Register

Table 23-85. PECCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R/W	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 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. 0h = PEC transmission and check is disabled 1h = PEC transmission and check is enabled
11-9	RESERVED	R/W	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. In the normal 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. 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 0h = Minimum Value 1FFh = Maximum Value

Chapter 24

UNICOMM SPI

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.

24.1 Overview 1333
24.2 UNICOMM common infrastructure 1334
24.3 External Connections and Signal Descriptions 1334
24.4 SPI Operation 1336
24.5 UNICOMMSPI Registers 1348

24.1 Overview

The UNICOMM-SPI module provides a standardized serial interface for transferring data between MSPM33 devices and external devices (such as Sensors, Memory, ADCs, or DACs) using SPI protocols.

24.1.1 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.

24.1.2 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 FIFOs, overrun and timeout interrupts, and DMA done
- Programmable 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 24-1. Optional UNICOMM-SPI feature support

UNICOMM-SPI Feature Tag	Feature description
SPI-REPEAT	Repeat transfer mode operation to send last byte of information N number of times. Useful in readback scenarios
SPI-RECEIVE-TIMEOUT	Receive timeout flag is raised if there are contents in the FIFO and receive timeout period has elapsed. (supported for peripheral mode only)
SPI-COMMAND-DATA-CONTROL	Command data signal (shared with CS3) supports Motorola frame format
SPI-MULTIPLE-CS	Support for multiple chip select
SPI-DMA	Direct memory access (DMA) with separate channels for transmit and receive and support for transfer complete interrupt

24.1.3 Features supported in each variant

SoC specific features are decided by each SoC. Values presented here are typical values used by M0+ devices

Table 24-2. SPI variants and features

Feature	SoC specific	Basic	Advanved
Controller / Peripheral mode	No	Yes	Yes
Controller: 4-16 bits Target: 7-16 bits	No	Yes	Yes
Parity	No	Yes	Yes
Repeat Mode	No	No	Yes
Receive Timeout	No	No	Yes

Table 24-2. SPI variants and features (continued)

Feature	SoC specific	Basic	Advanved
Command / Data Control	No	No	Yes
Support for four chip selects	No	No	Yes
Packing	No	No	No
FIFO	Yes	Yes - 4 deep	Yes - 4 deep
DMA	Yes	No	Yes
LP-support	Yes	No	No

24.1.4 Functional Block Diagram

24.2 UNICOMM common infrastructure

For details on common infrastructure elements refer to SPGSS document (section UNICOMM common elements). Sections under common infrastructure are – resets, clocks, mode selection, pin overlay, FIFO depth, FIFO operation, Suspend communication, Interrupts and Events and Emulation Mode.

24.3 External Connections and Signal Descriptions

Figure 24-1 and Table 24-3 show an overview of the pin functions for different operation modes of the SPI module.

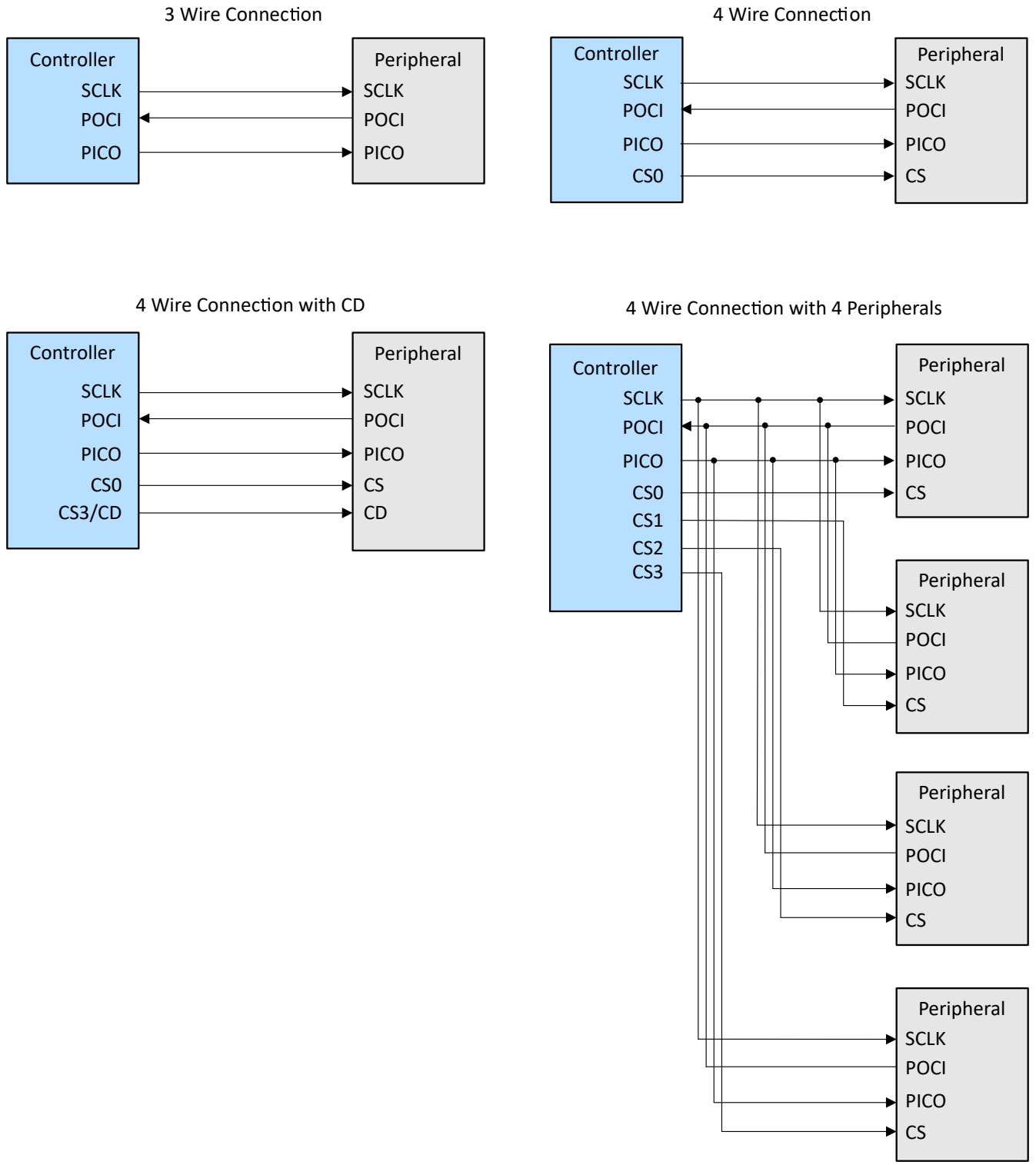


Figure 24-1. External Connections for Different SPI Configurations

Table 24-3. Pin Function Overview

Standard SPI	Feature
SCLK	**SPI clock** Controller mode: SCLK is an output Peripheral mode: SCLK is an input
PICO	**Controller out, peripheral in** Controller mode: PICO is the data output line Peripheral mode: PICO is the data input line
POCI	**Controller in, peripheral out** Controller mode: POCI is the data input line 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

24.4 SPI Operation

24.4.1 Clock Frequency Support

The maximum 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.

24.4.2 General Architecture

24.4.2.1 Chip Select and Command Handling

24.4.2.1.1 Chip Select Control

SPI can be configured to be 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 controlledby the 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 peripheralhas 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 theCS goes to the inactive state. When using the Motorola 4-wire or National Microwire mode, follow these constraints:

- The CS disable period must be longer than 2 functional clock cycles before the CS pin is re-asserted
- CTL0.CSCLR = 1: The transmit/receive bit counter is cleared when the CS signal disables the peripheral.

Following these constraints helps the peripheral to synchronize again on 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. SPI does not support asynchronous clock request generation.

24.4.2.1.2 Programmable CS-to-SCLK Setup Time

The CS_CLK_DLY field controls the delay between chip select assertion and the first SCLK edge. This delay can be programmed from 1 to 7 functional clock cycles. When not configured, no additional delay is inserted between CS and SCLK activation.

24.4.2.1.3 Automatic Chip Select Rotation

This feature operates in controller mode when CS_MASK_ROTATE_EN is enabled. The system automatically cycles through chip select lines in ascending order (from lowest to highest, e.g., 0→1→2→3) with each frame transmission.

The CSx_ROTATE_MASK bits determine which chip selects participate in the rotation sequence - only chip selects with their corresponding mask bit set will be included in the cycle.

Ex: If CS1_ROTATE_MASK and CS3_ROTATE_MASK are both set (with CS_MASK_ROTATE_EN enabled), the rotation sequence becomes:

- Frame 1: CS1 active
- Frame 2: CS3 active
- Frame 3: Returns to CS1 active
- Repeat the pattern

24.4.2.2 Command Data Control

When using the Motorola frame format, the CDMODE bit can be set to use the CS3/CD line as signal to distinguishbetween 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 lowfor the given numbers of bytes which are sent by the SPI, starting with the next value to be transmitted. After the number of bytes are transmitted the CD will go high automatically. If a value of 0xF is set the C/D stays low permanently, a value of 0 set 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 status of the FIFO should be checked to be empty and the SPI should be in Idle mode.

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 did count down to 0 and a new command package needs to be sent, CDMODE needs to be set first again, otherwise the next data is send as data with the CD pin signaling data mode.

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.

24.4.2.3 Data Format

The control bit CTL1.CP B defines the direction of the data input and output with most-significant-bit (MSB) or least-significant-bit (LSB) first. If the parity is enabled the parity bits is always received as last bit.

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 with 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 RX FIFO or RX buffer after the number of bits defined in CTL0.DSS register have 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 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 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 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 available for the peripheral on the CLK pins which needs to be configured for input. The Clock Select and divider control bits are not used. The CS input signal is used to select/enable the data receive path of the peripheral in 4 wire mode.

The SPI can be configured to work as Controller with CTL1.CP bit = 1. In Controller mode the clock needs to be generated by selecting the available clock sources with 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 to improve the robustness of the communication.

24.4.2.4 Delayed data sampling

In circumstances when the input data arrives at the POCI pin with some delay due to runtime conditions and the following input data sampling stage, the previous data would be sampled at the sampling clock edge. To compensate for such condition, a delayed sampling can be set with 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 with settingthe control register bits CLKCTL.DSAMPLE. The maximum allowed delay should not exceed the length of one data frame.

When CLKCTL.DSAMPLE is set to “zero”, delayed sampling will not be used.

24.4.2.5 SPI Clock Generation

The SPI includes a programmable bit rate clock divider and pre-scaler 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 UNICOMM Common elements 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)

SPISampling 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.

24.4.2.6 DMA Operation

The section applies to UNICOMM-SPI configurations which support the SPI-DMA feature.

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 24.4.13.2.

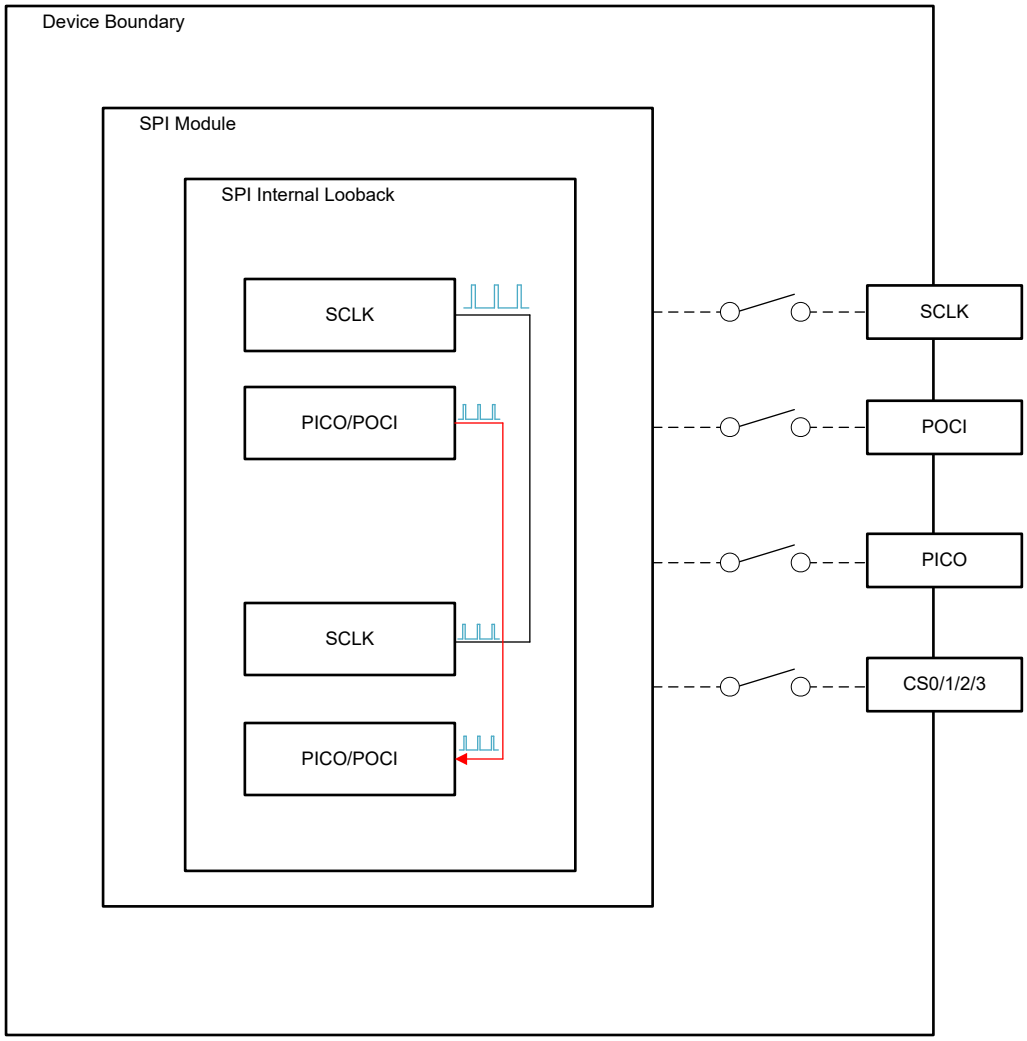
24.4.3 FIFO Size

FIFO width will be 16 bits without packing and 32 bits with packing. FIFO depth is as per variants list.

24.4.4 Internal Loopback Operation

Internal 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 serially transmitted into the RX FIFO. Thedata from the RX FIFO can be read to check whether correct transmission has occurred or not. The external toggling of the IOs has no effect when the module is set in the internal loopback mode.



24.4.5 Repeat Transfer mode

This function is available only in Controller mode. With the CTL1.REPEATTX bits, the last character transmit will be repeated as defined by the register bits. A valueof 0 in CTL1.REPEATTX bits will disable this mode.

The transfer will be started by writing a data into the TX Buffer. Then the data will be repeatedly sent with the same 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 buffer/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

24.4.6 Receive Timeout

In peripheral mode, application can detect receive timeout by programming CTL1.RXTIMEOUT and CLKCTL.SCR. Receive timeout is calculated as the time period from the last active clock edge to the programmed timeout period. Receive timeout flag is raised if there are contents in the FIFO and 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

24.4.7 Line Timeout

An additional bit, LTOUT (LINE timeout) is provided in UNICOMM. In peripheral mode, if no new transaction has been detected for receive timeout period after the last transaction, a line timeout interrupt is set (LTOUT) irrespective of FIFO/buffer contents.

24.4.8 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.

24.4.8.1 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 24-2 shows signal sequences for Motorola SPI format with SPO = 0 and SPH = 0.

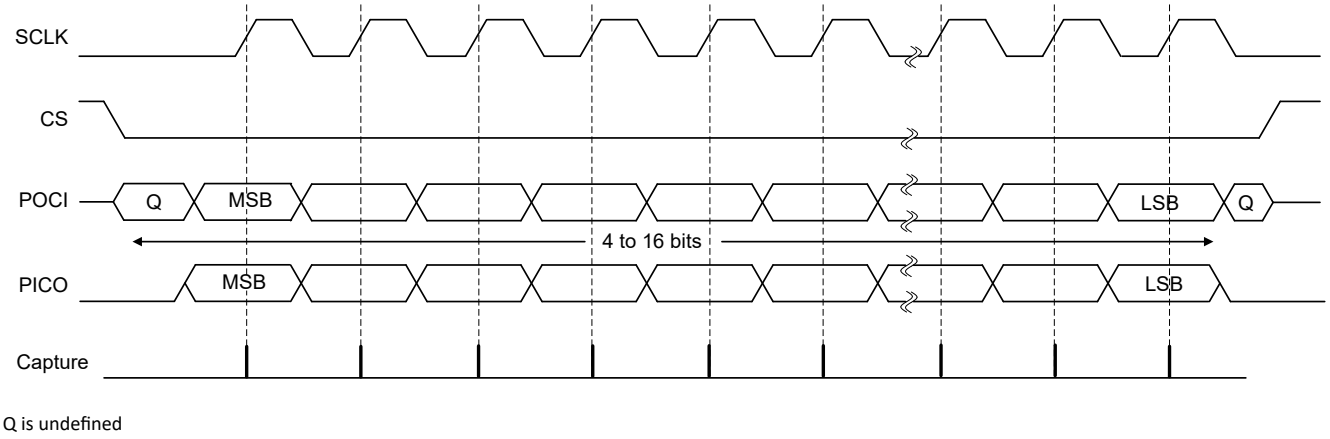


Figure 24-2. Motorola SPI Format With SPO = 0 and SPH = 0

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 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 24-3 shows the signal sequence for Motorola SPI format with SPO = 0 and SPH = 1.

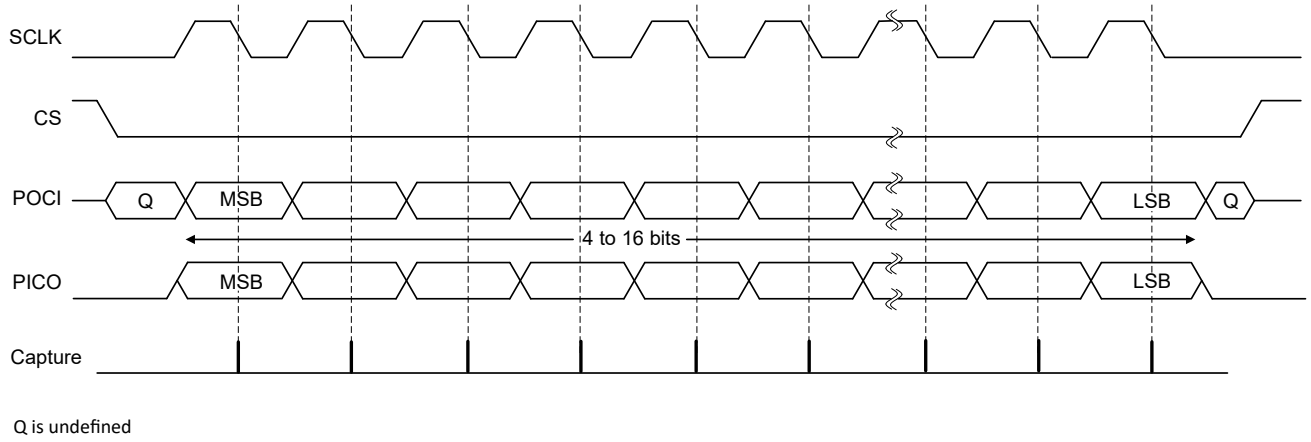


Figure 24-3. Motorola SPI Frame Format With SPO = 0 and SPH = 1

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

Figure 24-4 shows signal sequences for Motorola SPI format with SPO = 1 and SPH = 0.

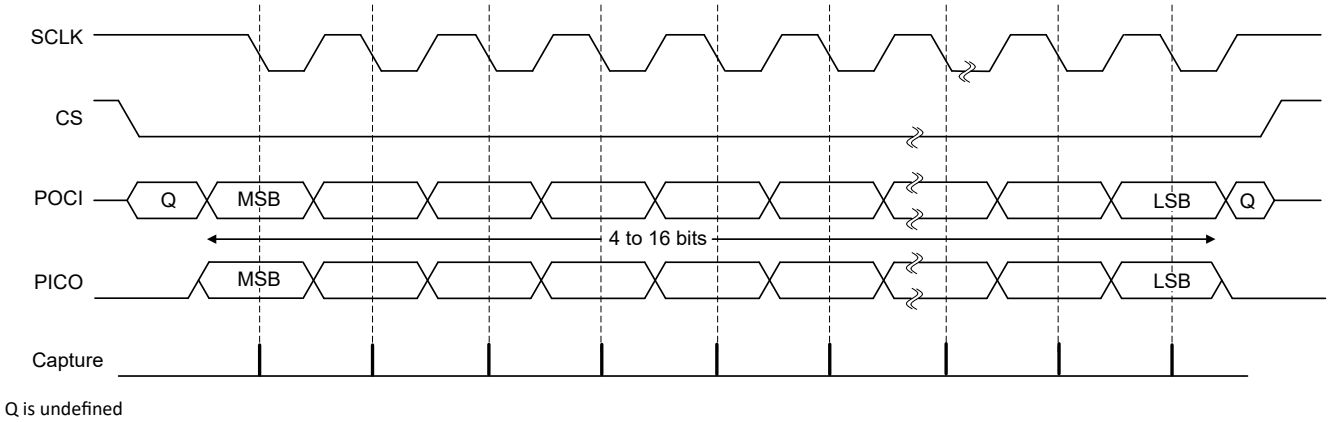


Figure 24-4. Motorola SPI Frame Format With SPO = 1 and SPH = 0

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 24-5shows the signal sequence for Motorola SPI format with SPO = 1 and SPH = 1.

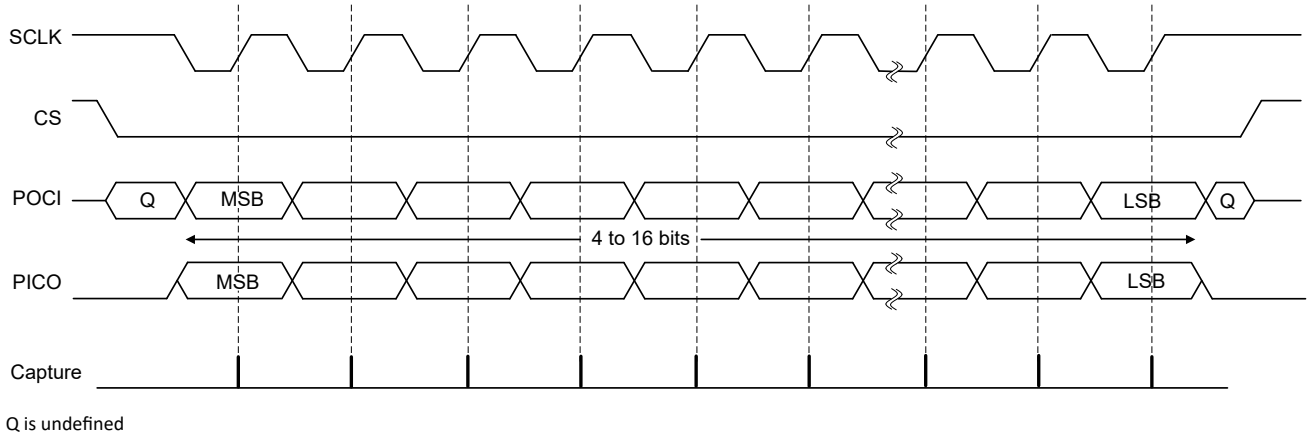


Figure 24-5. Motorola SPI Frame Format With SPO = 1 and SPH = 1

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.

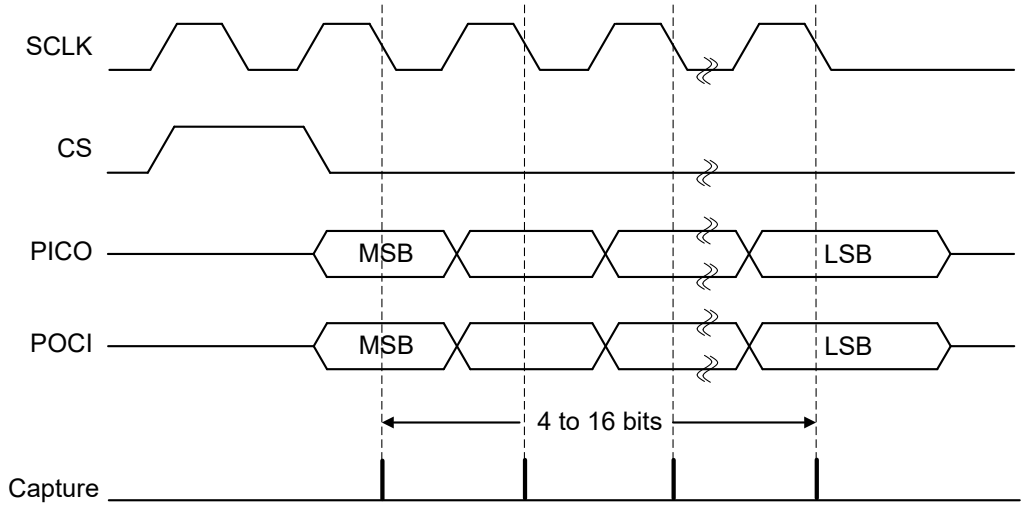
Frame Separation Control

In SPH=1 mode, the default behavior streams frames continuously without CS toggling. The DIS_CONT_FRAMES bit overrides this - when set, CS will de-assert and re-assert between each frame even when SPH=1.

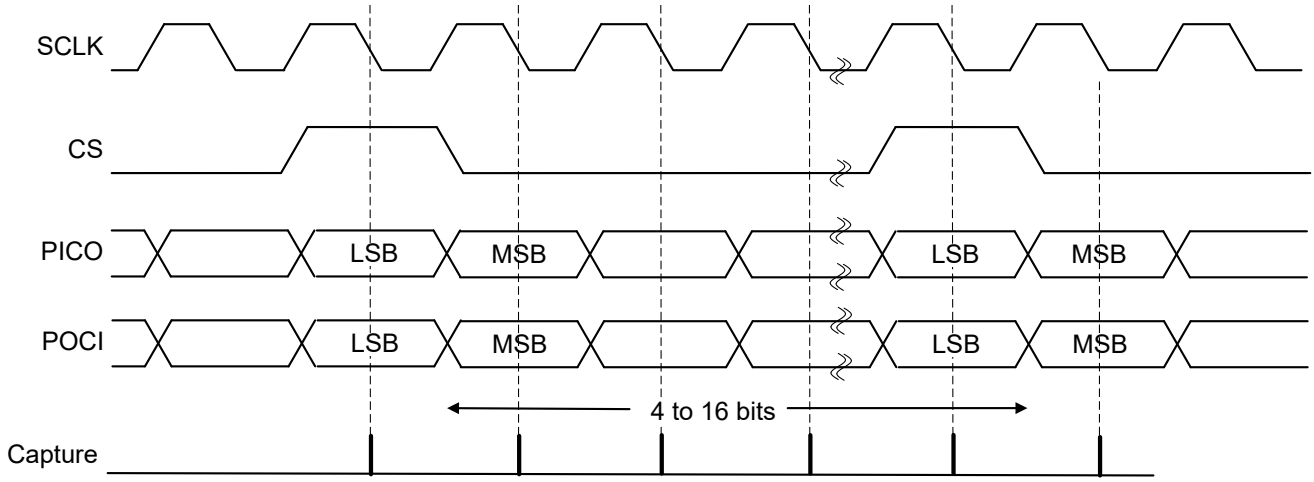
24.4.8.2 Texas Instruments Synchronous Serial Frame Format

The SPI peripheral is compatible with Texas Instruments Synchronous Serial frame format.

Figure 24-6 shows the TI synchronous serial frame format for a single and continuous transmitted frame.



Single Transmission Signal Sequences



Continuous Transmission Signal Sequences

Figure 24-6. 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.

24.4.9 Status Flags

The table below lists the status and their behavior. FIFO status bits are explained in the common SPGSS chapter.

Table 24-4. SPI variants and features

Status Bit	Description
BUSY	Controler Mode: 1. Busy bit is set (“ACTIVE”) when there is an ongoing transmission or reception 2. Busy bit is cleared (“IDLE”) where there is no ongoing transaction (receive or transmit) Peripheral Mode: 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 2. Busy bit is cleared (“IDLE”) when there is no ongoing transaction (receive or transmit)
CDMODE	Reflects current counter value and manual status 0x0: manual mode – C/D is low (data is being sent) 0xf: manual mode – C/D is high (command is being sent) 0x0 to 0xe: present byte count

24.4.10 Module pseudo static bit 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.

24.4.11 Reset Considerations

24.4.12 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.

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

24.4.13 Interrupt and Events Support

24.4.13.1 CPU Interrupt Event Publisher (CPU_INT)

24.4.13.2 DMA Trigger Publisher (DMA_TRIG_RX, DMA_TRIG_TX)

24.4.14 Emulation Modes

24.5 UNICOMMSPI Registers

Table 24-5 lists the memory-mapped registers for the UNICOMMSPI registers. All register offset addresses not listed in Table 24-5 should be considered as reserved locations and the register contents should not be modified.

Table 24-5. UNICOMMSPI Registers

Offset	Acronym	Register Name	Section
1000h	CLKDIV	Clock Divider	Section 24.5.1
1008h	CLKSEL	Clock Select for Ultra Low Power peripherals	Section 24.5.2
1020h	IIDX	Interrupt Index Register	Section 24.5.3
1028h	IMASK	Interrupt mask	Section 24.5.4
1030h	RIS	Raw interrupt status	Section 24.5.5
1038h	MIS	Masked interrupt status	Section 24.5.6
1040h	ISET	Interrupt set	Section 24.5.7
1048h	ICLR	Interrupt clear	Section 24.5.8
1058h	IMASK	Interrupt mask	Section 24.5.9
1060h	RIS	Raw interrupt status	Section 24.5.10
1068h	MIS	Masked interrupt status	Section 24.5.11
1070h	ISET	Interrupt set	Section 24.5.12
1088h	IMASK	Interrupt mask	Section 24.5.13
1090h	RIS	Raw interrupt status	Section 24.5.14
1098h	MIS	Masked interrupt status	Section 24.5.15
10A0h	ISET	Interrupt set	Section 24.5.16
10E4h	INTCTL	Interrupt control register	Section 24.5.17
1100h	CTL0	SPI control register 0	Section 24.5.18
1108h	STAT	Status Register	Section 24.5.19
110Ch	IFLS	Interrupt FIFO Level Select Register	Section 24.5.20
1110h	CLKCTL	Clock prescaler and divider register.	Section 24.5.21
1120h	TXDATA	TXDATA Register	Section 24.5.22
1124h	RXDATA	RXDATA Register	Section 24.5.23
114Ch	CTL1	SPI control register 1	Section 24.5.24

Complex bit access types are encoded to fit into small table cells. Table 24-6 shows the codes that are used for access types in this section.

Table 24-6. 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

24.5.1 CLKDIV Register (Offset = 1000h) [Reset = 00000000h]

CLKDIV is shown in Table 24-7.

Return to the Summary Table.

This register is used to specify module-specific divide ratio of the functional clock

Table 24-7. 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 0h = Do not divide clock source 1h = Divide clock source by 2 2h = Divide clock source by 3 3h = Divide clock source by 4 4h = Divide clock source by 5 5h = Divide clock source by 6 6h = Divide clock source by 7 7h = Divide clock source by 8

24.5.2 CLKSEL Register (Offset = 1008h) [Reset = 00000000h]

CLKSEL is shown in Table 24-8.

Return to the Summary Table.

Clock source selection for peripherals

Table 24-8. 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 0h = Does not select this clock as a source 1h = Select this clock as a source
10	ASYNC_HFCLK_SEL	R/W	0h	Asynchronous HFCLK selected as source 0h = Does not select this clock as a source 1h = Select this clock as a source
9	ASYNC_SYSCLK_SEL	R/W	0h	Asynchronous sys clock selected as source 0h = Does not select this clock as a source 1h = Select this clock as a source
8	ASYNC_LFCLK_SEL	R/W	0h	Asynchronous lfclk selected as source 0h = Does not select this clock as a source 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 0h = Does not select this clock as a source 1h = Select this clock as a source
2	MFCLK_SEL	R/W	0h	Selects MFCLK as clock source if enabled 0h = Does not select this clock as a source 1h = Select this clock as a source
1	LFCLK_SEL	R/W	0h	Selects LFCLK as clock source if enabled 0h = Does not select this clock as a source 1h = Select this clock as a source
0	RESERVED	R	0h	

24.5.3 IIDX Register (Offset = 1020h) [Reset = 00000000h] (CPU_INT)

IIDX is shown in Table 24-9.

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.

Table 24-9. IIDX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	STAT	R	0h	Interrupt index status 00h = No interrupt pending 1h = Transmit Parity Event/interrupt pending 2h = RX FIFO Overflow Event/interrupt pending 3h = Receive interrupt 4h = RX FIFO Full Interrupt 5h = TX FIFO underflow interrupt 6h = Transmit Event 7h = Transmit Buffer Empty Event/interrupt pending 9h = End of Transmit Event/interrupt pending Ah = SPI receive time-out interrupt 10h = DMA DONE on RX 11h = DMA DONE on TX 13h = DMA PRE IRQ RX INTERRUPT 14h = DMA PRE IRQ TX INTERRUPT 15h = SPI line time-out interrupt

24.5.4 IMASK Register (Offset = 1028h) [Reset = 00000000h] (CPU_INT)

IMASK is shown in Table 24-10.

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 24-10. IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	RESERVED	R	0h	
20	LTOUT	R/W	0h	Enable SPI Line Time-Out event mask. 0h = Clear Interrupt Mask 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	Enable DMA Done on TX Event Channel Interrupt 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
15	DMA_DONE_RX	R/W	0h	Enable DMA Done on RX Event Channel Interrupt 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
14-10	RESERVED	R	0h	
9	RTOUT	R/W	0h	Enable SPI Receive Time-Out event mask. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
8	IDLE	R/W	0h	SPI Idle event mask. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
7	RESERVED	R	0h	
6	TXEMPTY	R/W	0h	Transmit FIFO Empty event mask. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
5	TX	R/W	0h	Mask Transmit Event 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
4	TXFIFO_UNF	R/W	0h	TX FIFO underflow interrupt mask 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
3	RXFULL	R/W	0h	RX FIFO Full Interrupt Mask 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
2	RX	R/W	0h	Mask Receive Event 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
1	RXFIFO_OVF	R/W	0h	RXFIFO overflow event mask. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
0	PER	R/W	0h	Parity error event mask. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask

24.5.5 RIS Register (Offset = 1030h) [Reset = 00000000h] (CPU_INT)

RIS is shown in Table 24-11.

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 24-11. RIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	RESERVED	R	0h	
20	LTOUT	R	0h	SPI Line Time-Out event. This bit is set after timeout period has lapsed 0h = Interrupt did not occur 1h = Interrupt occurred
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 0h = Interrupt did not occur 1h = Interrupt occurred
15	DMA_DONE_RX	R	0h	DMA Done on RX Event Channel Interrupt 0h = Interrupt did not occur 1h = Interrupt occurred
14-10	RESERVED	R	0h	
9	RTOUT	R	0h	SPI Receive Time-Out event. This bit is set when there is still contents in the FIFO after the timeout period 0h = Interrupt did not occur 1h = Interrupt occurred
8	IDLE	R	0h	SPI has done finished transfers and changed into IDLE mode. This bit is set when BUSY goes low. 0h = Interrupt did not occur 1h = Interrupt occurred
7	RESERVED	R	0h	
6	TXEMPTY	R	0h	Transmit FIFO Empty interrupt mask. This interrupt is set if all data in the Transmit FIFO has moved to the shift register. 0h = Interrupt did not occur 1h = Interrupt occurred
5	TX	R	0h	Transmit event When FIFO is present, this bit is set when Tx FIFO threshold is reached When FIFO is not present, this bit is set when TX buffer is empty 0h = Interrupt did not occur 1h = Interrupt occurred
4	TXFIFO_UNF	R	0h	TX FIFO Underflow Interrupt 0h = Interrupt did not occur 1h = Interrupt occurred
3	RXFULL	R	0h	RX FIFO Full Interrupt 0h = Interrupt did not occur 1h = Interrupt occurred
2	RX	R	0h	Receive event When FIFO is present, this bit is set when FIFO threshold is reached When FIFO is not present, this bit is set when a frame is received 0h = Interrupt did not occur 1h = Interrupt occurred
1	RXFIFO_OVF	R	0h	RXFIFO overflow event. This interrupt is set if an RX FIFO overflow has been detected. 0h = Interrupt did not occur 1h = Interrupt occurred

Table 24-11. RIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	PER	R	0h	Parity error event: this bit is set if a Parity error has been detected 0h = Interrupt did not occur 1h = Interrupt occurred

24.5.6 MIS Register (Offset = 1038h) [Reset = 00000000h] (CPU_INT)

MIS is shown in Table 24-12.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Table 24-12. MIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	RESERVED	R	0h	
20	LTOUT	R	0h	Masked SPI Line Time-Out Interrupt. 0h = Interrupt did not occur or mask was not enabled 1h = Interrupt occured
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 0h = Interrupt did not occur or mask was not enabled 1h = Interrupt occurred
15	DMA_DONE_RX	R	0h	Masked DMA Done on RX Event Channel Interrupt 0h = Interrupt did not occur or mask was not enabled 1h = Interrupt occurred
14-10	RESERVED	R	0h	
9	RTOUT	R	0h	Masked SPI Receive Time-Out Interrupt. 0h = Interrupt did not occur or mask was not enabled 1h = Interrupt occured
8	IDLE	R	0h	Masked SPI IDLE mode event. 0h = Interrupt did not occur or mask was not enabled 1h = Interrupt occurred
7	RESERVED	R	0h	
6	TXEMPTY	R	0h	Masked Transmit FIFO Empty event. 0h = Interrupt did not occur or mask was not enabled 1h = Interrupt occurred
5	TX	R	0h	Masked Transmit FIFO event. This interrupt is set if the selected Transmit FIFO level has been reached. 0h = Interrupt did not occur or mask was not enabled 1h = Interrupt occurred
4	TXFIFO_UNF	R	0h	TX FIFO underflow interrupt 0h = Interrupt did not occur or mask was not enabled 1h = Interrupt occurred
3	RXFULL	R	0h	RX FIFO Full Interrupt 0h = Interrupt did not occur or mask was not enabled 1h = Interrupt occurred
2	RX	R	0h	Masked Receive Event 0h = Interrupt did not occur or mask was not enabled 1h = Interrupt occurred
1	RXFIFO_OVF	R	0h	Masked RXFIFO overflow event. This interrupt is set if an RX FIFO overflow has been detected. 0h = Interrupt did not occur or mask was not enabled 1h = Interrupt occurred
0	PER	R	0h	Masked Parity error event: this bit if a Parity error has been detected 0h = Interrupt did not occur or mask was not enabled 1h = Interrupt occurred

24.5.7 ISET Register (Offset = 1040h) [Reset = 00000000h] (CPU_INT)

ISET is shown in Table 24-13.

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 24-13. ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	RESERVED	R	0h	
20	LTOUT	W	0h	Set SPI Line Time-Out Event. 0h = Writing 0 has no effect 1h = Set Interrupt
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 0h = Writing 0 has no effect 1h = Set Interrupt
15	DMA_DONE_RX	W	0h	Set DMA Done on RX Event Channel Interrupt 0h = Writing 0 has no effect 1h = Set Interrupt
14-10	RESERVED	R	0h	
9	RTOUT	W	0h	Set SPI Receive Time-Out Event. 0h = Writing 0 has no effect 1h = Set Interrupt
8	IDLE	W	0h	Set SPI IDLE mode event. 0h = Writing 0 has no effect 1h = Set Interrupt
7	RESERVED	R	0h	
6	TXEMPTY	W	0h	Set Transmit FIFO Empty event. 0h = Writing 0 has no effect 1h = Set Interrupt
5	TX	W	0h	Set Transmit event. 0h = Writing 0 has no effect 1h = Set Interrupt
4	TXFIFO_UNF	W	0h	Set TX FIFO Underflow Event 0h = Writing has no effect 1h = Set interrupt
3	RXFULL	W	0h	Set RX FIFO Full Event 0h = Writing has no effect 1h = Set Interrupt
2	RX	W	0h	Set Receive event. 0h = Writing 0 has no effect 1h = Set Interrupt
1	RXFIFO_OVF	W	0h	Set RXFIFO overflow event. 0h = Writing 0 has no effect 1h = Set Interrupt
0	PER	W	0h	Set Parity error event. 0h = Writing 0 has no effect 1h = Set Interrupt

24.5.8 ICLR Register (Offset = 1048h) [Reset = 00000000h] (CPU_INT)

ICLR is shown in Table 24-14.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

Table 24-14. ICLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	RESERVED	R	0h	
20	LTOUT	W	0h	Clear SPI Line Time-Out Event. 0h = Writing 0 has no effect 1h = Clear Interrupt
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 0h = Writing 0 has no effect 1h = Clear Interrupt
15	DMA_DONE_RX	W	0h	Clear DMA Done on RX Event Channel Interrupt 0h = Writing 0 has no effect 1h = Clear Interrupt
14-10	RESERVED	R	0h	
9	RTOUT	W	0h	Clear SPI Receive Time-Out Event. 0h = Writing 0 has no effect 1h = Clear Interrupt
8	IDLE	W	0h	Clear SPI IDLE mode event. 0h = Writing 0 has no effect 1h = Clear Interrupt
7	RESERVED	R	0h	
6	TXEMPTY	W	0h	Clear Transmit FIFO Empty event. 0h = Writing 0 has no effect 1h = Clear Interrupt
5	TX	W	0h	Clear Transmit event. 0h = Writing 0 has no effect 1h = Clear Interrupt
4	TXFIFO_UNF	W	0h	Clear TXFIFO underflow event 0h = Writing has no effect 1h = Clear interrupt
3	RXFULL	W	0h	Clear RX FIFO underflow event 0h = Writing has no effect 1h = Clear interrupt
2	RX	W	0h	Clear Receive event. 0h = Writing 0 has no effect 1h = Clear Interrupt
1	RXFIFO_OVF	W	0h	Clear RXFIFO overflow event. 0h = Writing 0 has no effect 1h = Clear Interrupt
0	PER	W	0h	Clear Parity error event. 0h = Writing 0 has no effect 1h = Clear Interrupt

24.5.9 IMASK Register (Offset = 1058h) [Reset = 00000000h] (DMA_TRIG_RX)

IMASK is shown in Table 24-15.

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.

Table 24-15. IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	0h	
9	RTOUT	R/W	0h	SPI Receive Time-Out event mask. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
8-3	RESERVED	R	0h	
2	RX	R/W	0h	Receive FIFO event mask. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
1-0	RESERVED	R	0h	

24.5.10 RIS Register (Offset = 1060h) [Reset = 00000000h] (DMA_TRIG_RX)

RIS is shown in Table 24-16.

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.

Table 24-16. RIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	0h	
9	RTOUT	R	0h	SPI Receive Time-Out Event. This bit is set when there is still contents in the FIFO after the timeout period. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
8-3	RESERVED	R	0h	
2	RX	R	0h	Receive FIFO/BUFFER event. This interrupt is set if the selected Receive FIFO level has reached. In case there is no FIFO, this interrupt is set when BUFFER is FULL 0h = Interrupt did not occur 1h = Interrupt occurred
1-0	RESERVED	R	0h	

24.5.11 MIS Register (Offset = 1068h) [Reset = 00000000h] (DMA_TRIG_RX)

MIS is shown in Table 24-17.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Table 24-17. MIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	0h	
9	RTOUT	R	0h	SPI Receive Time-Out event . 0h = Interrupt did not occur or mask was not enabled. 1h = Interrupt Occurred
8-3	RESERVED	R	0h	
2	RX	R	0h	Receive FIFO/BUFFER event. 0h = Interrupt did not occur or mask was not enabled 1h = Interrupt occurred
1-0	RESERVED	R	0h	

24.5.12 ISET Register (Offset = 1070h) [Reset = 00000000h] (DMA_TRIG_RX)

ISET is shown in Table 24-18.

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.

Table 24-18. ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	0h	
9	RTOUT	W	0h	Set SPI Receive Time-Out event. 0h = Writing 0 has no effect 1h = Set Interrupt Mask
8-3	RESERVED	R	0h	
2	RX	W	0h	Set Receive FIFO event. 0h = Writing 0 has no effect 1h = Set Interrupt
1-0	RESERVED	R	0h	

24.5.13 IMASK Register (Offset = 1088h) [Reset = 00000000h] (DMA_TRIG_TX)

IMASK is shown in Table 24-19.

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 24-19. IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	
5	TX	R/W	0h	Transmit FIFO event mask. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
4-0	RESERVED	R	0h	

24.5.14 RIS Register (Offset = 1090h) [Reset = 00000000h] (DMA_TRIG_TX)

RIS is shown in Table 24-20.

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 24-20. RIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	
5	TX	R	0h	Transmit FIFO event: When TX FIFO is present, set as per FIFO thresholds. When FIFO is not present, set when TX BUFFER is empty. 0h = Interrupt did not occur 1h = Interrupt occurred
4-0	RESERVED	R	0h	

24.5.15 MIS Register (Offset = 1098h) [Reset = 00000000h] (DMA_TRIG_TX)

MIS is shown in Table 24-21.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Table 24-21. MIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	
5	TX	R	0h	Masked Transmit FIFO event 0h = Interrupt did not occur or mask was not enabled. 1h = Interrupt occurred
4-0	RESERVED	R	0h	

24.5.16 ISET Register (Offset = 10A0h) [Reset = 00000000h] (DMA_TRIG_TX)

ISET is shown in Table 24-22.

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 24-22. ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	
5	TX	W	0h	Set Transmit FIFO event. 0h = Writing 0 has no effect 1h = Set Interrupt
4-0	RESERVED	R	0h	

24.5.17 INTCTL Register (Offset = 10E4h) [Reset = 00000000h]

INTCTL is shown in Table 24-23.

Return to the Summary Table.

Interrupt control register

Table 24-23. 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. 0h = Writing 0 has no effect 1h = Interrupt Eval

24.5.18 CTL0 Register (Offset = 1100h) [Reset = 00000000h]

CTL0 is shown in Table 24-24.

Return to the Summary Table.

SPI control register 0

Table 24-24. CTL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28	CS_MASK_ROTATE_EN	R/W	0h	Chip select rotate mask enable 0h = No additional delay 1h = Enable
27	CS3_ROTATE_MASK	R/W	0h	Chip select rotate mask 0h = Disable 1h = Enable
26	CS2_ROTATE_MASK	R/W	0h	Chip select rotate mask 0h = Disable 1h = Enable
25	CS1_ROTATE_MASK	R/W	0h	Chip select rotate mask 0h = Disable 1h = Enable
24	CS0_ROTATE_MASK	R/W	0h	Chip select rotate mask 0h = Disable 1h = Enable
23-15	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. 0h = Disable automatic clear of shift register when CS goes to disable. 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 0h (R/W) = CS line select: 0 1h (R/W) = CS line select: 1 2h (R/W) = CS line select: 2 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. 0h = Data is captured on the first clock edge transition. 1h = Data is captured on the second clock edge transition.
8	SPO	R/W	0h	CLKOUT polarity (Motorola SPI frame format only) 0h = SPI produces a steady state LOW value on the CLKOUT 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 0h = Packing feature disabled 1h = Packing feature enabled
6-5	FRF	R/W	0h	Frame format Select 0h = Motorola SPI frame format (3 wire mode) 1h = Motorola SPI frame format (4 wire mode) 2h = TI synchronous serial frame format 3h = Reserved - not to be selected

Table 24-24. 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 3h (R/W) = Data Size Select bits: 4 4h (R/W) = Data Size Select bits: 5 5h (R/W) = Data Size Select bits: 6 6h (R/W) = Data Size Select bits: 7 7h (R/W) = Data Size Select bits: 8 8h (R/W) = Data Size Select bits: 9 9h (R/W) = Data Size Select bits: 10 Ah (R/W) = Data Size Select bits: 11 Bh (R/W) = Data Size Select bits: 12 Ch (R/W) = Data Size Select bits: 13 Dh (R/W) = Data Size Select bits: 14 Eh (R/W) = Data Size Select bits: 15 Fh (R/W) = Data Size Select bits: 16

24.5.19 STAT Register (Offset = 1108h) [Reset = 00000000h]

STAT is shown in Table 24-25.

Return to the Summary Table.

Status Register

Table 24-25. 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. 0h = Smallest value 0h = Manual mode: Data Fh = Manual mode: Command
11-9	RESERVED	R	0h	
8	BUSY	R	0h	Busy 0h = SPI is in idle mode. 1h = SPI is currently transmitting and/or receiving data
7	TXCLR	R	0h	TX FIFO Clear Status 0h = FIFO is not cleared 1h = FIFO clear is complete
6	TXFF	R	0h	Transmit FIFO FULL 0h = Transmit FIFO is not FULL 1h = Transmit FIFO is FULL
5	TXFE	R	1h	Transmit FIFO empty. 0h = Transmit FIFO is not empty. 1h = Transmit FIFO is empty.
4	RXCLR	R	0h	RX FIFO Clear Status 0h = FIFO is not cleared 1h = FIFO clear is complete
3	RXFF	R	0h	Receive FIFO FULL 0h = Receive FIFO is not FULL 1h = Receive FIFO is FULL
2	RXFE	R	1h	Receive FIFO empty. 0h = Receive FIFO is not empty. 1h = Receive FIFO is empty.
1-0	RESERVED	R	0h	

24.5.20 IFLS Register (Offset = 110Ch) [Reset = 00000022h]

IFLS is shown in Table 24-26.

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.

Table 24-26. 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. 0h = Disable FIFO clear 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. 1h = RX FIFO >= 1/4 full 2h = RX FIFO >= 1/2 full (default) 3h = RX FIFO >= 3/4 full 4h = Opposite of empty 5h = RX FIFO is full 6h = RX_FIFO >= (MAX_FIFO_LEN -1) 7h = RX_FIFO <= 1
3	TXCLR	R/W	0h	TX FIFO CLEAR. Setting this bit will clear the TX FIFO contents. 0h = Disable FIFO clear 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. 1h = TX FIFO <= 3/4 empty 2h = TX FIFO <= 1/2 empty (default) 3h = TX FIFO <= 1/4 empty 4h = Opposite of full 5h = TX FIFO is empty 6h = TX FIFO <= 1 7h = TX_FIFO >= (MAX_FIFO_LEN -1)

24.5.21 CLKCTL Register (Offset = 1110h) [Reset = 00000000h]

CLKCTL is shown in Table 24-27.

Return to the Summary Table.

Clock prescaler and divider register. This register contains the settings for the Clock prescaler and divider settings.

Table 24-27. 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 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. 0h = Delayed sampling is not used 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. 0h = Smallest value 3FFh = Highest possible value

24.5.22 TXDATA Register (Offset = 1120h) [Reset = 00000000h]

TXDATA is shown in Table 24-28.

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

Table 24-28. 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 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. 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. 0h = Smallest value FFFFh = Highest possible value

24.5.23 RXDATA Register (Offset = 1124h) [Reset = 00000000h]

RXDATA is shown in Table 24-29.

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.

Table 24-29. 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 receive FIFO, pointed to by the current FIFO write pointer. Received data less than 16 bits is automatically right justified in the receive buffer. 0h = Smallest value FFFFh = Highest possible value

24.5.24 CTL1 Register (Offset = 114Ch) [Reset = 00000004h]

CTL1 is shown in Table 24-30.

Return to the Summary Table.

SPI control register 1

Table 24-30. CTL1 Register 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. 0h = Smallest value 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. 0h = Smallest value 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. 0h = Manual mode: Data 0h = Smallest value Fh = Manual mode: Command
11	CDENABLE	R/W	0h	Command/Data Mode enable 0h = CS3 is used for Chip Select 1h = CS3 is used as CD signal
10	RESERVED	R	0h	
9	SUSPEND	R/W	0h	Suspend external communication 0h = Functional mode resumed 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. 0h = Parity transmission is disabled 1h = Parity transmission is enabled
7	RESERVED	R	0h	
6	PES	R/W	0h	Even Parity Select 0h = Odd Parity mode 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. 0h = Disable Parity receive function 1h = Enable Parity receive function

Table 24-30. CTL1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	MSB	R/W	0h	MSB first select. Controls the direction of the receive and transmit shift register. 0h = LSB first 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. 0h = SPI can drive the POCI output in peripheral mode. 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. 0h = Select Peripheral mode 1h = Select Controller Mode
1	LBM	R/W	0h	Loop back mode 0h = Disable loopback mode 1h = Enable loopback mode
0	ENABLE	R/W	0h	SPI enable 0h = Disable module function 1h = Enable module function

Chapter 25

QSPI

The quad serial peripheral interface (QSPI) module provides a standardized serial interface to transfer data between MSP devices and other external devices with QSPI interfaces (such as flash memory or high-refresh displays).

25.1 QSPI Overview 1377
25.2 QSPI Operation 1380
25.3 QSPI Registers 1391

25.1 QSPI Overview

The QSPI module provides a standardized serial interface to transfer data between MSP devices and other external devices with QSPI interfaces (such as flash memory or high-refresh displays).

25.1.1 Purpose of the Peripheral

QSPI is a peripheral used to communicate with a serial NOR flash over an augmented SPI bus where two additional data lines are provided along with COPI and CIPO signals. 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.

25.1.2 Features

The QSPI module has the following features:

- Serial clock for up to 40MHz to external NOR flash.
- Single, Bi (Dual) and Quad data read transfers per clock edge.
- 3- and 4-byte address for accessing the NOR flash.
- Interrupt to indicate FIFO status (full, not-full, empty).
- 8- or 16-bit length FIFO packing or unpacking between device and peripheral
- Programmable dummy clock insertion to match requirements from external NOR flash.
- Direct memory access controller interface (DMA):
  - Separate channels for transmit and receive.
  - Transfer complete interrupt.

25.1.3 Functional Block Diagram

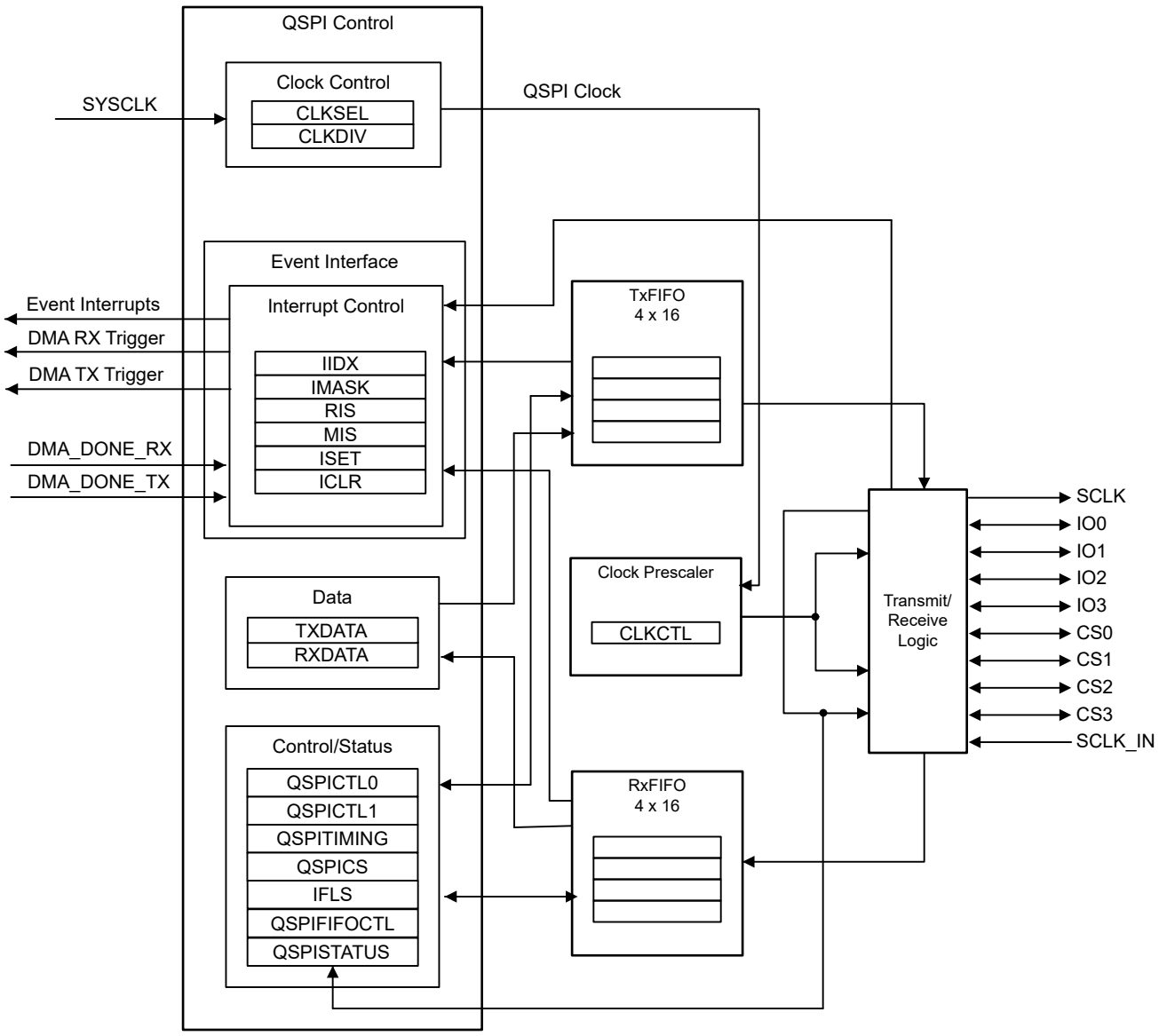


Figure 25-1. QSPI Functional Block Diagram

25.1.4 External Connections and Signal Descriptions

Figure 25-2 and Table 25-1 show an overview of the pin functions for different operation modes of the SPI module.

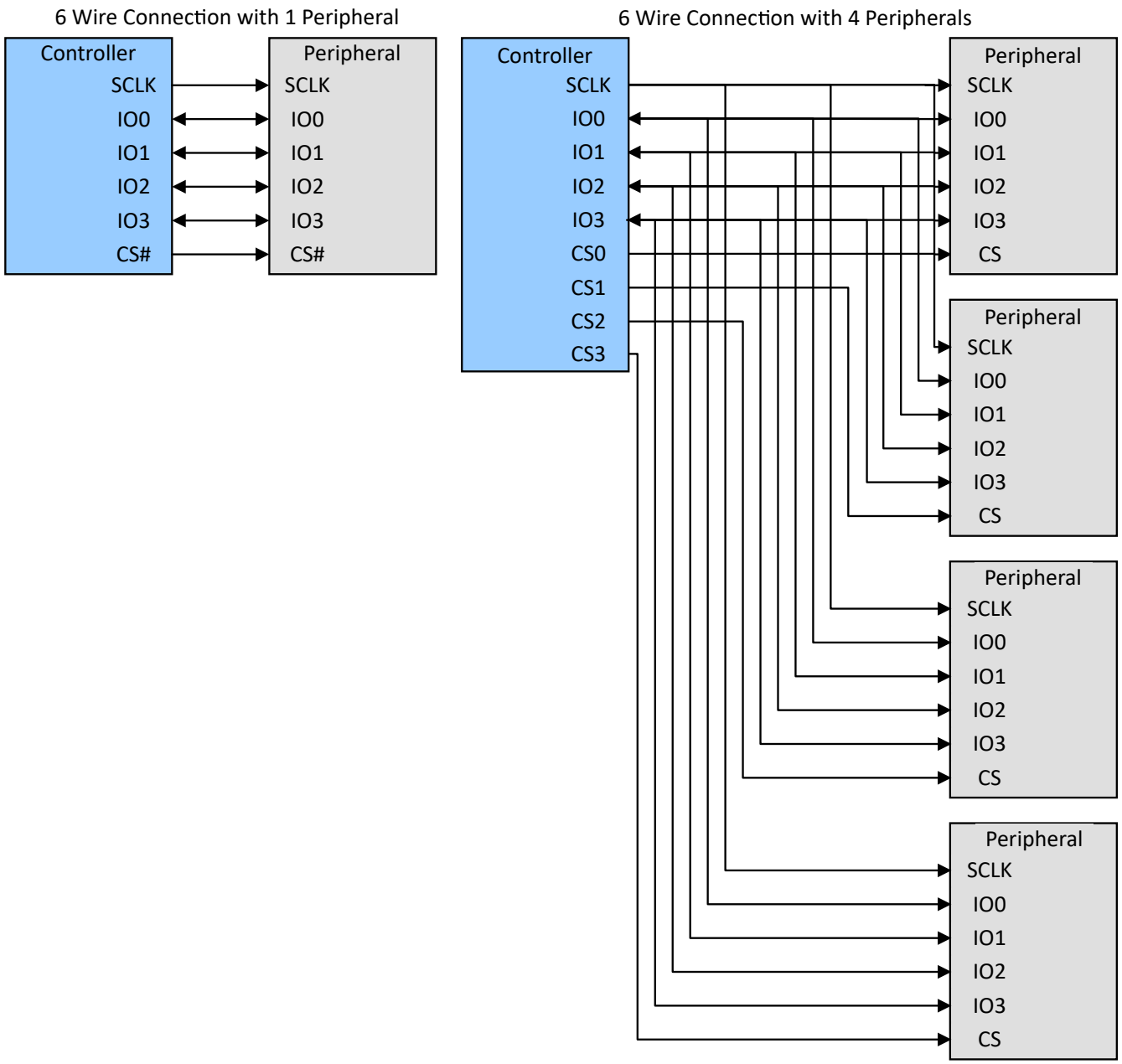


Figure 25-2. External Connections for Different QSPI Configurations

Table 25-1. Pin Function Overview

QSPI	Feature
SCLK	SPI clock
IO0	Input/Output Signal 0
IO1	Input/Output Signal 1
IO2	Input/Output Signal 2
IO3	Input/Output Signal 3
CS0	Chip select signal 0
CS1	Chip select signal 1, used with multiple peripherals
CS2	Chip select signal 2, used with multiple peripherals

Table 25-1. Pin Function Overview (continued)

QSPI	Feature
CS3	Chip select signal 3, used with multiple peripherals

25.2 QSPI Operation

25.2.1 Clock Control

The QSPI internal functional clock is selected and divided from the MCLK of BCLK clock domain in power domain 1 (PD1). Use QSPI.CLKDIV register to select the divide ratio of the QSPI function clock. Options are from divide by 1 to 8.The QSPI module must be enabled before being configured for use by using the ENABLE bit in QSPI.PWREN register. When the QSPI will be setup or the configuration should be changed the QSPI.CTL1->ENABLE bit should be cleared to avoid unpredictable behavior during the updates or for the first data receive or transmitted afterward.

The QSPI includes a programmable bit rate clock divider and prescaler to generate the serial output clock (SCLK). Bit rate supported is up to maximum 40MHz SCLK. "QSPI Clock" is the output after clock division performed according to ratio selected by the CLKDIV register.

$$\text{QSPI Clock} = \frac{Input\ Clock}{(1 + CLKDIV)} \tag{20}$$

QSPI Sampling Clock (SCLK) is the output after dividing the QSPI Clock by the Prescalar value.

$$\text{SCLK} = \frac{QSPI\ Clock}{((1 + SCR) \times 2)} \tag{21}$$

If the factor of two (*2) is set by CLKDIV the input clock must be at least 2 times faster than QSPI clock.

25.2.2 General Architecture

25.2.2.1 Chip Select Control

The CTL0.CSSEL bit selects which connected peripheral is addressed by the up to 4 CS signals. The selected signal is controlled during the transfers. The chip select signal needs to be provided by the controller and the chip select polarity can be inverted by configuring the PINCM.CSx.INV register.

The chip select can be toggled back to inactive 1 once the controller has completed the transfer using the QSPICS register. Additionally, the CS assertion and de-assertion delays with respect to external SCLK can be configured using the QSPITIMING register.

25.2.2.2 Data Format

With the control register bits QSPICTL0.QSPIDSIZE the bit length per transfer will be defined either 8 or 16 bits. A transfer will be triggered with writing to the TX buffer register. The data write needs to have the number of bits of the transfer. For example, if two bytesare written to the TX buffer but the length of the transfer is 8, the remaining 8 bits will be ignored. On the receive path the received data will be moved to the RXFIFO or RX buffer after the number of bits defined in QSPICTL0.DSIZE register have been received.

Clock polarity (CTL0.SPO) is used to control the clock polarity when data is not being transferred.

- 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.

Clock phase (CTL0.SPH) 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.

- 0h = Data is captured on the first clock edge transition.
- 1h = Data is captured on the second clock edge transition.

Refer to Motorola SPI frame for more details on SPO and SPH.

Note

Only the combinations SPO=0, SPH=0, and SPO=1, SPH=1 are possible in QSPI. Otherwise combinations are not supported, and mapped to SPO=0, SPH=0, even if configured by the user.

25.2.2.3 Delayed data sampling

In circumstances when the input data arrives at the IO pins with some delay due to runtime conditions and the following input data sampling stage, the previous data would be sampled at the sampling clock edge. To compensate for such condition, a delayed sampling can be set with the CLKCTL.DSAMPLE bits. The delay can be adjusted in steps of QSPI input clock steps with setting the control register bits CLKCTL.DSAMPLE. The maximum allowed delay should not exceed the length of one data frame.

25.2.2.4 Loopback mode

Similar to delayed data sampling, QSPI has an option to sample the data on a delayed clock. In this case, the SCLK is looped-back internally through an IO cell of the SOC and fed back to the QSPI. The delay in this case is fixed. The clock loopback can be enabled by setting the QSPITIMING.DATSMPLDY bit.

Note

The user needs to enable the INENA in the IOMUX->PINCM register for the pin corresponding to SCLK of QSPI to enable the loopback clock.

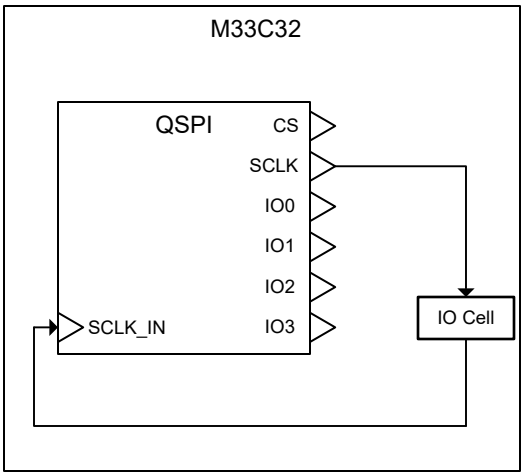


Figure 25-3. QSPI Clock Loopback

25.2.2.5 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 QSPI Data Register TXDATA.DATA, and data is stored in the FIFO until it is read out by the transmission logic. Parallel data is written into the TX FIFO before serial conversion and transmission to the attached peripheral, respectively, through the IO pins is done. User or software is responsible to make valid data available to the FIFO as needed. The SPI can be configured to generate an interrupt or a DMA request when the FIFO is empty. 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. Received data from the serial interface is stored in the buffer until read out by the CPU or DMA, which accesses the read FIFO by reading the RXDATA register. Serial data is received through the IO pins. As the access pointer for the FIFO will be

updated with each access, the data needs to be accessed by single transfers. With the FIFO fill level trigger signals located in the QSPISTAT register (TXFIFONF, TXFIFOE, RXFIFONF, RXFIFOE) the FIFO buffer allows an application to continuously stream serial data in one buffer while the application moves or process the data from the other buffer. If the FIFO is full and new data is written into the FIFO without reading data the RXFIFO overflow event is set.The receive FIFO has a RXFULL interrupt to indicate a FIFO full condition.

25.2.2.6 DMA Operation

The QSPI provides an interface to the DMA controller with separate channels for transmit and receive. The DMA operation of the QSPI is enabled through the QSPI Event and DMA register. When DMA operation is enabled, the QSPI 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. For the transmit channel a transfer request is asserted whenever the transmit FIFO contains lesser data 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. Once DMA transfer is completed, the CPU intervention shall be required for FIFO flush control and to de-assert the chip select.

25.2.2.7 Lower Power Mode

The QSPI module is located in power domain 1 (PD1) and as such is only active in RUN and SLEEP modes. If the QSPI module is enabled by application software, an entry into STOP or STANDBY low-power mode forces the QSPI module to be temporarily disabled while the device is in STOP or STANDBY mode.

25.2.3 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 QSPI interface external pull up or

25.2.4 Initialization

Note

It is not recommended to use QSPICTL0->QSPIFORMAT = 0x0, which configures QSPI to behave as legacy SPI module. If legacy SPI functionality is needed, the user should use the SPGSS.SPI modules in M33C32. The user should not configure and use any Legacy SPI *** registers mentioned in Section 2.11, except for CTL0->CSSEL, SPO,SPH and CTL1->ENABLE as needed. CTL1->CP should always be 1 as QSPI only works as a controller.

To enable and initialize the QSPI, the following steps are necessary:

1. Configure the IOMUX with the appropriate GPIO pins for which the QSPI 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 QSPI is configured using the following steps:

1. Ensure that the ENABLE bit in the CTL1 register is clear before making any configuration changes.

2. Configure the clock prescale divider by writing the CLKDIV register.
3. Ensure that CLKSEL always selects SYSCLK as the source clock as QSPI does not support MFCLK and LFCLK.
4. Ensure that MS bit in the CTL1 register is set as QSPI always works in controller mode.
5. Ensure that MSB bit in the CTL1 register is set to always send MSB first when using QSPI.
6. Ensure that PACKEN bit in the CTL0 register is always disabled as QSPI does not support 32-bit data packing.
7. Configure the clock divider by writing the CLKCTL register.
8. Configure the QSPI registers with based on the desired frame, data width and other special configurations.
9. Optionally configure DMA.
10. Enable the QSPI by setting the ENABLE bit in the CTL1 register.

25.2.5 QSPI Controller Description

25.2.5.1 Configuration Frame Access

When sending a configuration frame, the QSPI controller shall only use QSPIFORMAT of “0001” or “0010” for either SPI or QPI half-duplex mode.

Note

SPI Mode refers to the QSPI module emulating a SPI module, sending the command, instruction, and data on a single communication line.

Note

QPI Mode refers to the QSPI module sending the command, instruction, and data on all four comunication lines.

- The host must check QSPISTATUS.QSPIIDLE = ‘1’
- The host must program QSPICTL1 register
  - The number of bytes to send (instruction and payload) to QSPICTL1.TXCOUNT
  - QSPICTL1.RXCOUNT = 0 and QSPICTL1.QSPIPREFETCH = 0
- Write the instruction byte to the TXFIFO and monitor QSPISTATUS.TXFIFONF before writing additional payload byte(s).
- Check QSPISTATUS.QSPISTALL to be ‘1’
- Write QSPICS.CSHOLDCTL as ‘1’ and check QSPISTATUS.QSPIIDLE = ‘1'
- Write QSPICS.CSHOLDCTL = '0' to release the chip select for the subsequent transactions

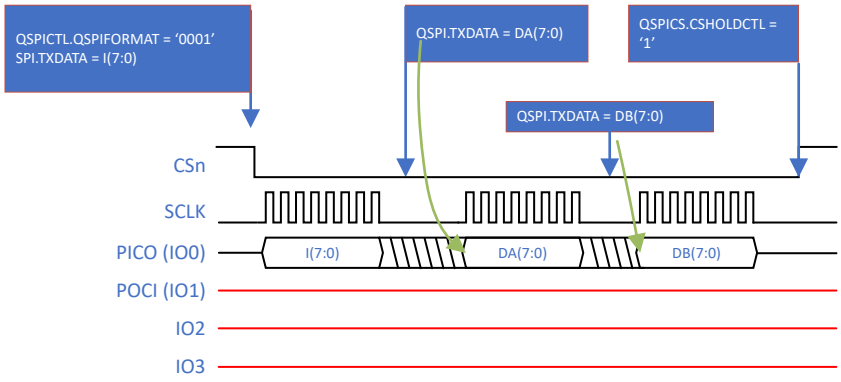


Figure 25-4. Configuration Frame Access-SPI Mode

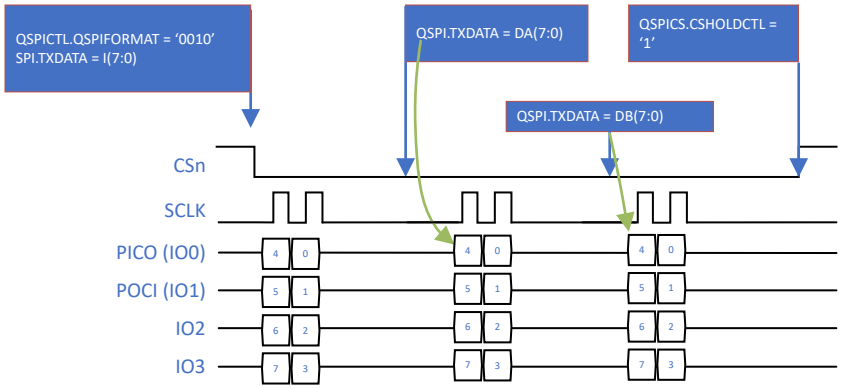


Figure 25-5. Configuration Frame Access-QPI Mode

25.2.5.2 Status Frame Access

When sending a status frame, the QSPI controller shall only use QSPIFORMAT of “0001” or “0010” for either SPI or QPI half-duplex mode.

- The host must check QSPISTATUS.QSPIIDLE = ‘1'
- The host must program QSPICTL1 register
  - The number of bytes to send (instruction and payload) to QSPICTL1.TXCOUNT
  - The number of bytes to receive to QSPICTL1.RXCOUNT and QSPICTL1.QSPIPREFETCH = 1
- Write the instruction byte to the TXFIFO and monitor QSPISTATUS.TXFIFONF before writing additional payload byte(s)
- Check QSPISTATUS.RXFIFOE to be ‘0’ and read the number of RXCOUNT status bytes
- Once RXCOUNT data bytes are received, write QSPICS.CSHOLDCTL as ‘1’ and check QSPISTATUS.QSPIIDLE = ‘1’
- Write QSPICS.CSHOLDCTL = '0' to release the chip select for the subsequent transactions

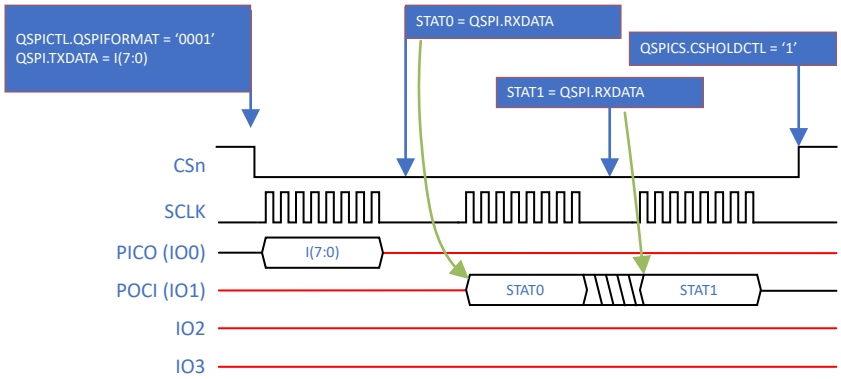


Figure 25-6. Status Frame Access-SPI Mode

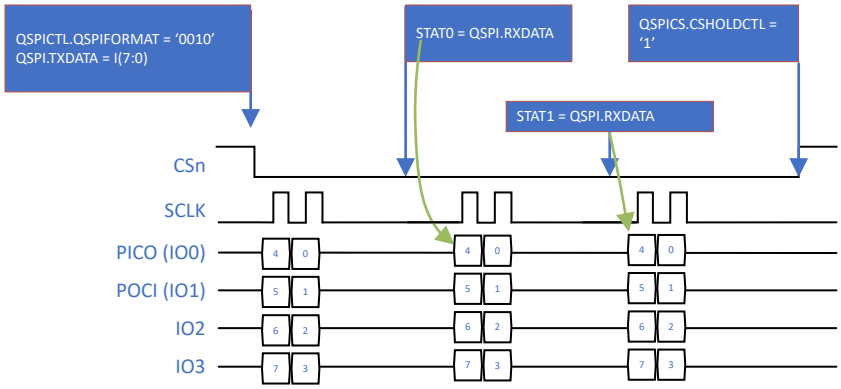


Figure 25-7. Status Frame Access-QPI Mode

25.2.5.3 Data packing and unpacking

The QSPIDSIZE bit is used for data packing and unpacking. When the field is programmed for byte (8-bit) access, all writes and read to/from the FIFO are with the lowest 8-bit containing valid data. When the field is programmed for half-word (16-bit) access, then all writes and read to/from the FIFO are with the lowest 16-bit containing valid data. FIFO is always accessed in little endian format to be compliant to the ARM architecture.

When data path is configured for 16-bit word mode, the CPU shall write the bytes starting LSB first.

- If the order of data to be sent on the bus is 0xAA, 0xBB, 0xCC then CPU shall write to the TXFIFO 0xBBAA and 0xXXCC
- If the order of data to be sent on the bus is 0x11, 0x22, 0x33, 0x44 then CPU shall write to the TXFIFO 0x2211 and 0x4433

When the CPU shall read the bytes from the RXFIFO, the LSB shall be the first byte received.

- If the order of data from external memory is 0x12, 0x34, 0x56 then CPU shall read 0x3412 and 0x0056 from the RXFIFO
- If the order of data from external memory is 0xAB, 0xCD, 0xEF, 0x78 then CPU shall read 0xCDAB and 0x78EF from the RXFIFO

Note

The transfer count shall always be number of bytes and not be dependent on the FIFO width being 8- or 16-bit. For a 16-bit FIFO, the transfer count shall be decremented by 2 and on the last entry only LSB sent.

25.2.5.4 Data Frame Access

When sending a data read frame in indirect mode, the QSPI controller shall only use the format with QSPIFORMAT[3] as '1' with other bits as per the required frame format. It must also program the QSPIPREFETCH, QSPIADDR, QSPIPERFBYTE, QSPICMDBYTE and QSPIDUMMYCLK for proper operation.

Once the programming is complete, the host shall write the address to the TXFIFO to prime the transaction. After the first byte is written to the TXFIFO, the QSPI controller shall start the bus transaction by sending the QSPICMDBYTE, followed by the address byte, QSPIPERFBYTE and dummy clocks. During the instruction and address phase (including the performance byte when enabled) the QSPI data lines will be driven by the host controller depending on the frame format. During the dummy clock phase and read data, the QSPI lines will only be input mode until the chip select is de-asserted. The bus shall not be sampled during dummy clock phase.

Note

Dummy clocks are required for interfacing with external NOR flash, and are needed to allow the NOR flash time to retrieve the requested data.

- The host must check QSPISTATUS.QSPIIDLE = '1'

- The host must program QSPICTL0 register only one time if the data frame format is not being changed
  - QSPIFORMAT for type of transfer
  - QSPICMDBYTE for the instruction opcode as per the flash datasheet
  - QSPIADDR for using 3- or 4-byte address field as per the flash datasheet
  - QSPIPERFBYTE and QSPIPERFMODE for the performance byte as per the flash datasheet
  - QSPIDUMMYCLK for number of dummy clocks to be sent as per the flash datasheet
- Case-1: When total number of bytes is less than 256
  - The host must program QSPICTL1 register
    - The number of bytes to receive to QSPICTL1.RXCOUNT and QSPICTL1.QSPIPREFETCH = 1
  - Write the address byte to the TXFIFO and monitor QSPISTATUS.TXFIFONF before writing additional address byte(s) to prime the transfer
  - Check QSPISTATUS.RXFIFOE to be ‘0’ and read the number of RXCOUNT data bytes
  - Once RXCOUNT data bytes are received, write QSPICS.CSHOLDCTL as ‘1’ and check QSPISTATUS.QSPIIDLE = ‘1’
  - Write QSPICS.CSHOLDCTL = '0' to release the chip select for the subsequent transactions
- Case-2: When total number of bytes is equal to or more than 256
  - The host must program QSPICTL1 register
    - For indefinite read set QSPICTL1.RXCOUNT = 0 and QSPICTL1.QSPIPREFETCH = 1
  - Write the address byte to the TXFIFO and monitor QSPISTATUS.TXFIFONF before writing additional address byte(s) to prime the transfer
  - Check QSPISTATUS.RXFIFOE to be ‘0’ and read the data bytes
  - Once required number of data bytes are received, wait for QSPISTATUS.QSPISTALL to be 1.
  - Write QSPICS.CSHOLDCTL as ‘1’ and check QSPISTATUS.QSPIIDLE = ‘1’
  - Write QSPICS.RXFIFOFLUSH as ‘1’, check QSPISTATUS.RXFIFOE = ‘1’ and QSPISTATUS.QSPISTALL = ‘0’
  - Write QSPICS.CSHOLDCTL = '0' to release the chip select for the subsequent transactions

25.2.5.4.1 SSS mode (QSPIFORMAT = 1000)

In SSS mode, the instruction, address and data phase on the bus is done in single lane mode. The host shall always send the instruction and address on IO0. The memory shall send the data on IO1.

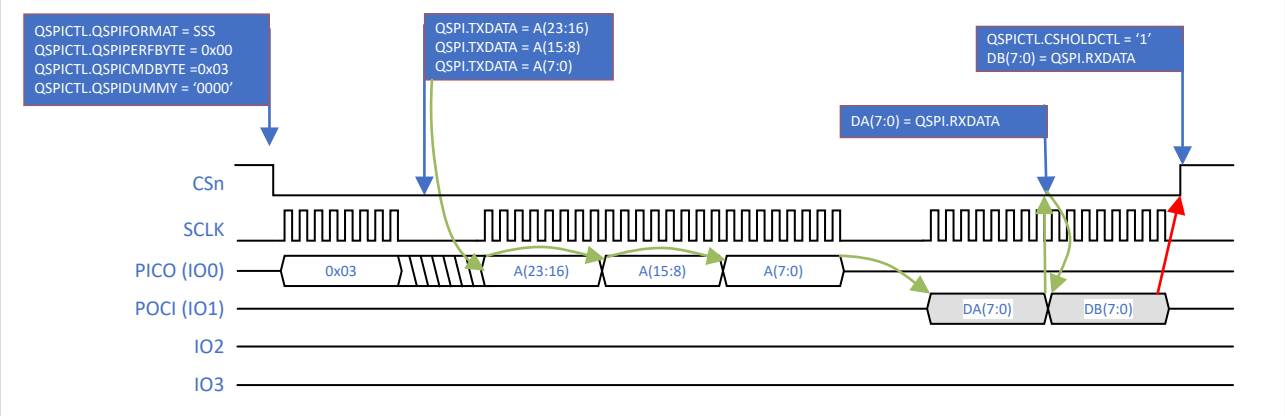


Figure 25-8. Data Read-SSS Format with 0 dummy clocks (Normal Read)

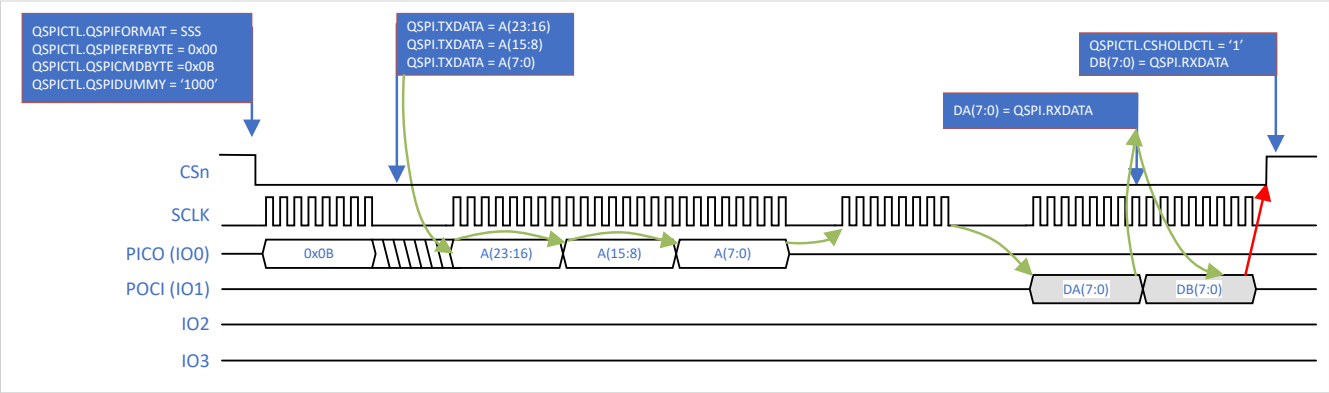


Figure 25-9. Data Read-SSS Format with 8 dummy clocks (Fast Read)

25.2.5.4.2 SSD mode (QSPIFORMAT = 1001)

In SSD mode, the host shall always send the instruction and address on IO0. The memory shall send the data on IO1 and IO0 with the most significant bit (MSb) first on IO1 and MSb-1 on IO0.

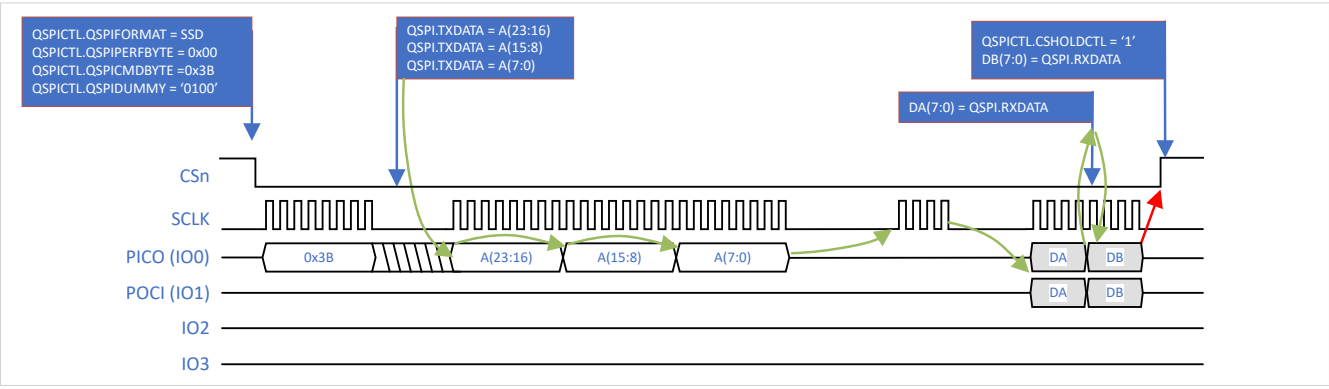


Figure 25-10. Data Read-SSD Format with 4 dummy clocks (Fast Read Dual Output)

25.2.5.4.3 SDD mode (QSPIFORMAT = 1010)

In SDD mode, the host shall always send the instruction on IO0 and address on IO1 & IO0 with the most significant bit (MSb) first on IO1 and MSb-1 on IO0. The memory shall send the data on IO1 and IO0 with the most significant bit (MSb) first on IO1 and MSb-1 on IO0.

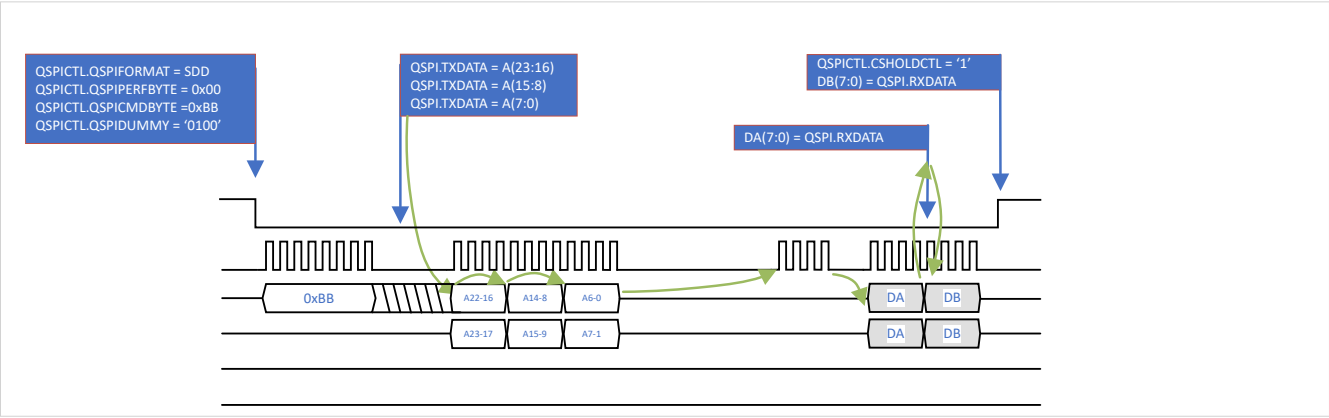


Figure 25-11. Data Read-SDD Format with 4 dummy clocks (Fast read Dual IO)

25.2.5.4.4 SSQ mode (QSPIFORMAT = 1011)

In SSQ mode, the host shall always send the instruction and address on IO0. The memory shall send the data on IO3, IO2, IO1 and IO0 with the most significant bit (MSb) first on IO3, MSb-1 on IO2, MSb-2 on IO1 and MSb-3 on IO0.

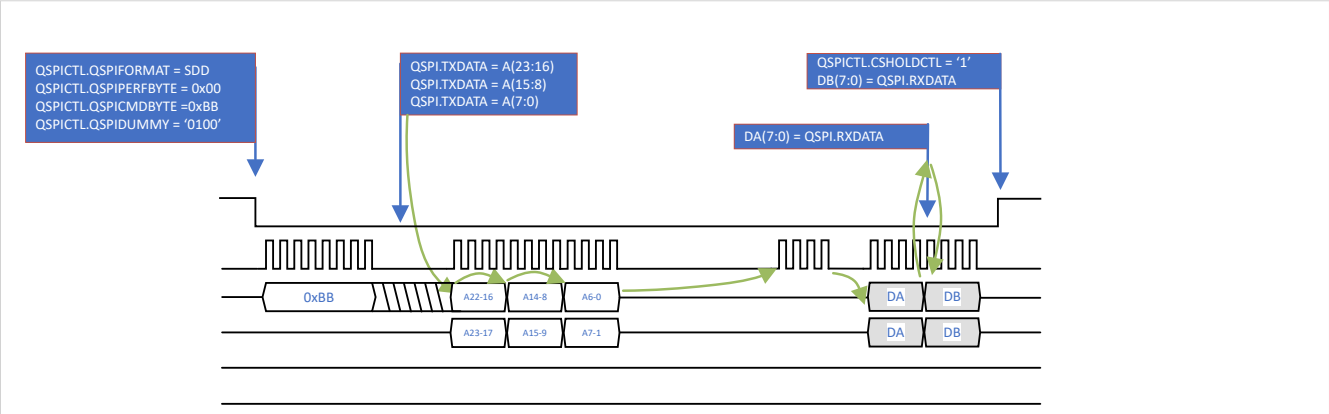


Figure 25-12. Data Read-SSQ Format with 2 dummy clocks (Fast Read Quad Output)

25.2.5.4.5 SQQ mode (QSPIFORMAT = 1100)

In SQQ mode, the host shall always send the instruction on IO0 and address on IO3, IO2, IO1 & IO0 with the most significant bit (MSb) first on IO3, MSb-1 on IO2, MSb-2 on IO1 and MSb-3 on IO0. The memory shall send the data on IO3, IO2, IO1 and IO0 with the most significant bit (MSb) first on IO3, MSb-1 on IO2, MSb-2 on IO1 and MSb-3 on IO0.

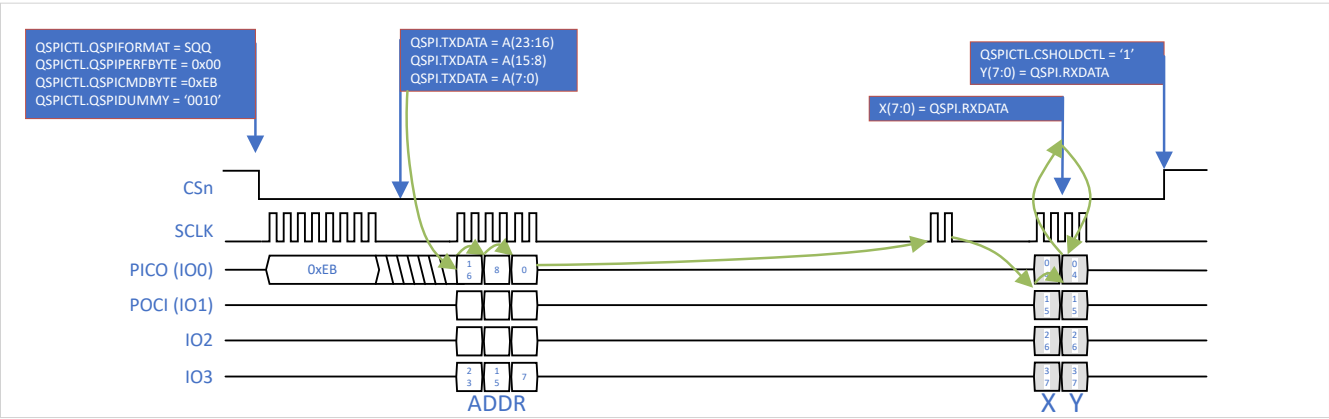


Figure 25-13. Data Read-SQQ Format with 2 dummy clocks (Fast read Quad IO)

25.2.5.4.6 QQQ mode (QSPIFORMAT = 1101)

In QQQ mode, the host shall always send the instruction and address on IO3, IO2, IO1 & IO0 with the most significant bit (MSb) first on IO3, MSb-1 on IO2, MSb-2 on IO1 and MSb-3 on IO0. The memory shall send the data on IO3, IO2, IO1 and IO0 with the most significant bit (MSb) first on IO3, MSb-1 on IO2, MSb-2 on IO1 and MSb-3 on IO0.

Note

QQQ mode is same as QPI mode

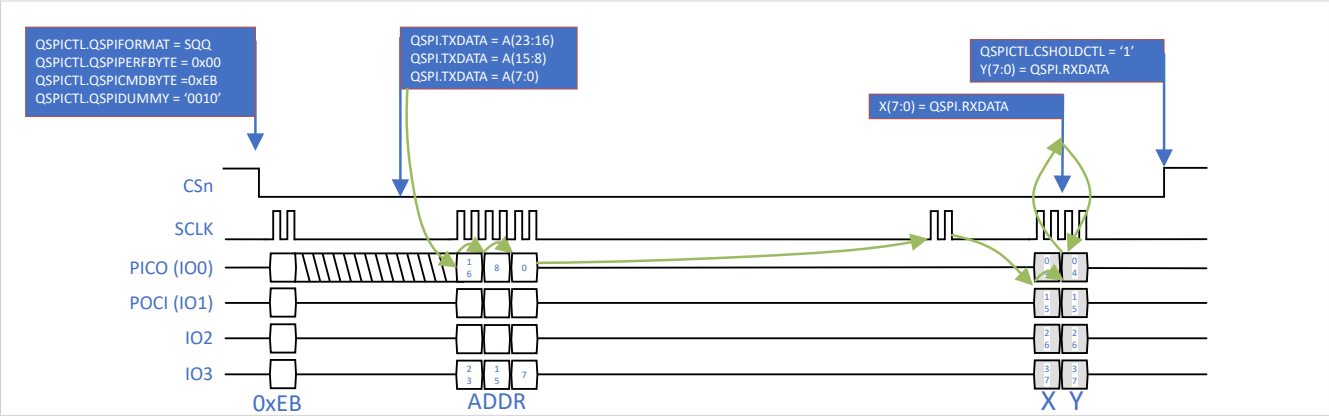


Figure 25-14. Data Read-QQQ Format with 2 dummy clocks

Note

To avoid bus stalls, it is recommended to read the status register to see if there is data in the RXDATA buffer before reading it. Since, this mode requires the CPU to read the QSPISTATUS register to check the FIFO status, the delay in issuing the status read, processing the result and then performing a FIFO read shall ensure that the FIFO shall have one byte of data in the fastest serial clock frequency.

25.2.6 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.

Table 25-2. QSPI Events

Event	Type	Source	Destination	Route	Configuration	Functionality
CPU interrupt	Publisher	QSPI	CPU Subsystem	Static route	CPU_INT registers	Fixed interrupt route from SPI to CPU
DMA trigger	Publisher	QSPI	DMA	DMA event route	DMA_TRIG_RX registers	Fixed interrupt route from QSPI RX to DMA
DMA trigger	Publisher	QSPI	DMA	DMA event route	DMA_TRIG_TX registers	Fixed interrupt route from QSPI TX to DMA

25.2.6.1 CPU Interrupt Event Publisher (CPU_INT)

The QSPI module provides 9 interrupt sources that can source a CPU interrupt event. SPI CPU_INT Trigger Condition lists the CPU interrupt events from the QSPI in order of decreasing priority.

Table 25-3. SPI CPU_INT Trigger Condition

IIDX STAT	Name	Description
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.
0x08	DMA_DONE1_RX	This interrupt is set if the RX DMA channel sends the DONE signal.

Table 25-3. SPI CPU_INT Trigger Condition (continued)

IIDX STAT	Name	Description
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 8.2.5 for guidance on configuring the Event registers.

25.2.6.2 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 SPI DMA_TRIG_RX DMA Trigger Condition and SPI DMA_TRIG_TX DMA Trigger Condition.

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 25-4. SPI DMA_TRIG_RX DMA Trigger Condition

IIDX STAT	Name	Description
0x04	RX	Receive FIFO event. This interrupt is set if the selected receive FIFO level has been reached.

Table 25-5. 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 8.2.5 for guidance on configuring the Event registers and Section 8.1.3.2 for on how DMA trigger event works.

25.2.7 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 25-6. 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

25.3 QSPI Registers

Table 25-7 lists the memory-mapped registers for the QSPI registers. All register offset addresses not listed in Table 25-7 should be considered as reserved locations and the register contents should not be modified.

Table 25-7. QSPI 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 Register	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 Register	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 Register	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
1100h	CTL0	SPI control register 0	Section 25.3.28
1104h	CTL1	SPI control register 1	Section 25.3.29
1108h	CLKCTL	Clock prescaler and divider register.	Section 25.3.30
110Ch	IFLS	Interrupt FIFO Level Select Register	Section 25.3.31
1110h	STAT	Status Register	Section 25.3.32
1130h	RXDATA	RXDATA Register	Section 25.3.33
1140h	TXDATA	TXDATA Register	Section 25.3.34
1200h	QSPICTL0	QSPI Control-0 register	Section 25.3.35
1204h	QSPICTL1	QSPI Control-1 Register	Section 25.3.36
1208h	QSPITIMING	The register is controls the timing setting for chip select assertion and de-assertion with respect to the serial clock, sampling delay for the data and chip-select behavior in performance mode	Section 25.3.37
120Ch	QSPIFIFOCTL	The register controls the flush operation in the peripheral FIFO	Section 25.3.38

Table 25-7. QSPI Registers (continued)

Offset	Acronym	Register Name	Section
1210h	QSPICSCTL	The register controls the toggling of chip select	Section 25.3.39
1220h	QSPIDMACTL	The register is used to generate a soft trigger for the DMA	Section 25.3.40
1234h	QSPISTATUS	The register contains the status bits for the QSPI controller and the FIFO	Section 25.3.41

Complex bit access types are encoded to fit into small table cells. Table 25-8 shows the codes that are used for access types in this section.

Table 25-8. QSPI Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
WK	W K	Write Write protected by a key
Reset or Default Value		
-n		Value after reset or the default value

25.3.1 PWREN Register (Offset = 800h) [Reset = 00000000h]

PWREN is shown in Table 25-9.

Return to the Summary Table.

Register to control the power state

Table 25-9. PWREN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to allow Power State Change 26h = KEY to allow write access to this register
23-1	RESERVED	R/W	0h	
0	ENABLE	R/WK	0h	Enable the power KEY must be set to 26h to write to this bit. 0h = Disable Power 1h = Enable Power

25.3.2 RSTCTL Register (Offset = 804h) [Reset = 00000000h]

RSTCTL is shown in Table 25-10.

Return to the Summary Table.

Register to control reset assertion and de-assertion

Table 25-10. RSTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	Unlock key 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 KEY must be set to B1h to write to this bit. 0h = Writing 0 has no effect 1h = Clear reset sticky bit
0	RESETASSERT	WK	0h	Assert reset to the peripheral KEY must be set to B1h to write to this bit. 0h = Writing 0 has no effect 1h = Assert reset

25.3.3 CLKCFG Register (Offset = 808h) [Reset = 00000000h]

CLKCFG is shown in Table 25-11.

Return to the Summary Table.

Peripheral Clock Configuration Register

Table 25-11. CLKCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow State Change -- 0xA9 A9h = key value to allow change field of GPRCM
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 0h = Not block async clock request 1h = Block async clock request
7-0	RESERVED	R/W	0h	

25.3.4 STAT Register (Offset = 814h) [Reset = 00000000h]

STAT is shown in Table 25-12.

Return to the Summary Table.

peripheral enable and reset status

Table 25-12. 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 0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register 1h = The peripheral was reset since the last bit clear
15-0	RESERVED	R	0h	

25.3.5 CLKDIV Register (Offset = 1000h) [Reset = 00000000h]

CLKDIV is shown in Table 25-13.

Return to the Summary Table.

This register is used to specify module-specific divide ratio of the functional clock

Table 25-13. CLKDIV Register 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 0h = Do not divide clock source 1h = Divide clock source by 2 2h = Divide clock source by 3 3h = Divide clock source by 4 4h = Divide clock source by 5 5h = Divide clock source by 6 6h = Divide clock source by 7 7h = Divide clock source by 8

25.3.6 CLKSEL Register (Offset = 1004h) [Reset = 00000000h]

CLKSEL is shown in Table 25-14.

Return to the Summary Table.

Clock source selection for peripherals

Table 25-14. CLKSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	SYSCLK_SEL	R/W	0h	Selects SYSCLK as clock source if enabled 0h = Does not select this clock as a source 1h = Select this clock as a source
2	MFCLK_SEL	R/W	0h	Selects MFCLK as clock source if enabled 0h = Does not select this clock as a source 1h = Select this clock as a source
1	LFCLK_SEL	R/W	0h	Selects LFCLK as clock source if enabled 0h = Does not select this clock as a source 1h = Select this clock as a source
0	RESERVED	R/W	0h	

25.3.7 PDBGCTL Register (Offset = 1018h) [Reset = 00000000h]

PDBGCTL is shown in Table 25-15.

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

Table 25-15. PDBGCTL Register 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' 0h = The peripheral will halt immediately, even if the resultant state will result in corruption if the system is restarted 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 0h = The peripheral freezes functionality while the Core Halted input is asserted and resumes when it is deasserted. 1h = The peripheral ignores the state of the Core Halted input

25.3.8 IIDX Register (Offset = 1020h) [Reset = 00000000h]

IIDX is shown in Table 25-16.

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.

Table 25-16. IIDX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	STAT	R	0h	Interrupt index status 00h = No interrupt pending 1h = RX FIFO Overflow Event/interrupt pending 2h = Transmit Parity Event/interrupt pending 3h = SPI receive time-out interrupt 4h = Receive Event/interrupt pending 5h = Transmit Event/interrupt pending 6h = Transmit Buffer Empty Event/interrupt pending 7h = End of Transmit Event/interrupt pending 8h = DMA Done for Receive Event/interrupt pending 9h = DMA Done for Transmit Event/interrupt pending Ah = TX FIFO underflow interrupt Bh = RX FIFO Full Interrupt

25.3.9 IMASK Register (Offset = 1028h) [Reset = 00000000h]

IMASK is shown in Table 25-17.

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.

Table 25-17. IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R/W	0h	
10	RXFULL	R/W	0h	RX FIFO Full Interrupt Mask 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
9	TXFIFO_UNF	R/W	0h	TX FIFO underflow interrupt mask 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
8	DMA_DONE_TX	R/W	0h	DMA Done 1 event for TX event mask. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
7	DMA_DONE_RX	R/W	0h	DMA Done 1 event for RX event mask. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
6	IDLE	R/W	0h	SPI Idle event mask. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
5	TXEMPTY	R/W	0h	Transmit FIFO Empty event mask. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
4	TX	R/W	0h	Transmit FIFO event mask. 0h = Clear Interrupt Mask 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 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
2	RTOUT	R/W	0h	Enable SPI Receive Time-Out event mask. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
1	PER	R/W	0h	Parity error event mask. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
0	RXFIFO_OVF	R/W	0h	RXFIFO overflow event mask. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask

25.3.10 RIS Register (Offset = 1030h) [Reset = 00000000h]

RIS is shown in Table 25-18.

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.

Table 25-18. RIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R	0h	
10	RXFULL	R	0h	RX FIFO Full Interrupt 0h = Interrupt did not occur 1h = Interrupt occurred
9	TXFIFO_UNF	R	0h	TX FIFO Underflow Interrupt 0h = Interrupt did not occur 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. 0h = Interrupt did not occur 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. 0h = Interrupt did not occur 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. 0h = Interrupt did not occur 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. 0h = Interrupt did not occur 1h = Interrupt occurred
4	TX	R	0h	Transmit FIFO event..This interrupt is set if the selected Transmit FIFO level has been reached. 0h = Interrupt did not occur 1h = Interrupt occurred
3	RX	R	0h	Receive FIFO event.This interrupt is set if the selected Receive FIFO level has been reached 0h = Interrupt did not occur 1h = Interrupt occurred
2	RTOUT	R	0h	SPI Receive Time-Out event. 0h = Interrupt did not occur 1h = Interrupt occurred
1	PER	R	0h	Parity error event: this bit is set if a Parity error has been detected 0h = Interrupt did not occur 1h = Interrupt occurred
0	RXFIFO_OVF	R	0h	RXFIFO overflow event. This interrupt is set if an RX FIFO overflow has been detected. 0h = Interrupt did not occur 1h = Interrupt occurred

25.3.11 MIS Register (Offset = 1038h) [Reset = 00000000h]

MIS is shown in Table 25-19.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Table 25-19. MIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R	0h	
10	RXFULL	R	0h	RX FIFO Full Interrupt 0h = Interrupt did not occur 1h = Interrupt occurred
9	TXFIFO_UNF	R	0h	TX FIFO underflow interrupt 0h = Interrupt did not occur 1h = Interrupt occurred
8	DMA_DONE_TX	R	0h	Masked DMA Done 1 event for TX. 0h = Interrupt did not occur 1h = Interrupt occurred
7	DMA_DONE_RX	R	0h	Masked DMA Done 1 event for RX. 0h = Interrupt did not occur 1h = Interrupt occurred
6	IDLE	R	0h	Masked SPI IDLE mode event. 0h = Interrupt did not occur 1h = Interrupt occurred
5	TXEMPTY	R	0h	Masked Transmit FIFO Empty event. 0h = Interrupt did not occur 1h = Interrupt occurred
4	TX	R	0h	Masked Transmit FIFO event. This interrupt is set if the selected Transmit FIFO level has been reached. 0h = Interrupt did not occur 1h = Interrupt occurred
3	RX	R	0h	Masked receive FIFO event.This interrupt is set if the selected Receive FIFO level has been reached 0h = Interrupt did not occur 1h = Interrupt occurred
2	RTOUT	R	0h	Masked SPI Receive Time-Out Interrupt. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
1	PER	R	0h	Masked Parity error event: this bit if a Parity error has been detected 0h = Interrupt did not occur 1h = Interrupt occurred
0	RXFIFO_OVF	R	0h	Masked RXFIFO overflow event. This interrupt is set if an RX FIFO overflow has been detected. 0h = Interrupt did not occur 1h = Interrupt occurred

25.3.12 ISET Register (Offset = 1040h) [Reset = 00000000h]

ISET is shown in Table 25-20.

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.

Table 25-20. ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	W	0h	
10	RXFULL	W	0h	Set RX FIFO Full Event 0h = Writing has no effect 1h = Set Interrupt
9	TXFIFO_UNF	W	0h	Set TX FIFO Underflow Event 0h = Writing has no effect 1h = Set interrupt
8	DMA_DONE_TX	W	0h	Set DMA Done 1 event for TX. 0h = Writing 0 has no effect 1h = Set Interrupt
7	DMA_DONE_RX	W	0h	Set DMA Done 1 event for RX. 0h = Writing 0 has no effect 1h = Set Interrupt
6	IDLE	W	0h	Set SPI IDLE mode event. 0h = Writing 0 has no effect 1h = Set Interrupt
5	TXEMPTY	W	0h	Set Transmit FIFO Empty event. 0h = Writing 0 has no effect 1h = Set Interrupt
4	TX	W	0h	Set Transmit FIFO event. 0h = Writing 0 has no effect 1h = Set Interrupt
3	RX	W	0h	Set Receive FIFO event. 0h = Writing 0 has no effect 1h = Set Interrupt
2	RTOUT	W	0h	Set SPI Receive Time-Out Event. 0h = Writing 0 has no effect 1h = Set Interrupt Mask
1	PER	W	0h	Set Parity error event. 0h = Writing 0 has no effect 1h = Set Interrupt
0	RXFIFO_OVF	W	0h	Set RXFIFO overflow event. 0h = Writing 0 has no effect 1h = Set Interrupt

25.3.13 ICLR Register (Offset = 1048h) [Reset = 00000000h]

ICLR is shown in Table 25-21.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

Table 25-21. ICLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	W	0h	
10	RXFULL	W	0h	Clear RX FIFO underflow event 0h = Writing has no effect 1h = Clear interrupt
9	TXFIFO_UNF	W	0h	Clear TXFIFO underflow event 0h = Writing has no effect 1h = Clear interrupt
8	DMA_DONE_TX	W	0h	Clear DMA Done 1 event for TX. 0h = Writing 0 has no effect 1h = Clear Interrupt
7	DMA_DONE_RX	W	0h	Clear DMA Done 1 event for RX. 0h = Writing 0 has no effect 1h = Clear Interrupt
6	IDLE	W	0h	Clear SPI IDLE mode event. 0h = Writing 0 has no effect 1h = Clear Interrupt
5	TXEMPTY	W	0h	Clear Transmit FIFO Empty event. 0h = Writing 0 has no effect 1h = Clear Interrupt
4	TX	W	0h	Clear Transmit FIFO event. 0h = Writing 0 has no effect 1h = Clear Interrupt
3	RX	W	0h	Clear Receive FIFO event. 0h = Writing 0 has no effect 1h = Clear Interrupt
2	RTOUT	W	0h	Clear SPI Receive Time-Out Event. 0h = Writing 0 has no effect 1h = Set Interrupt Mask
1	PER	W	0h	Clear Parity error event. 0h = Writing 0 has no effect 1h = Clear Interrupt
0	RXFIFO_OVF	W	0h	Clear RXFIFO overflow event. 0h = Writing 0 has no effect 1h = Clear Interrupt

25.3.14 IIDX Register (Offset = 1050h) [Reset = 00000000h]

IIDX is shown in Table 25-22.

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.

Table 25-22. IIDX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	STAT	R	0h	Interrupt index status 00h = No interrupt pending 3h = SPI receive time-out interrupt 4h = Receive Event/interrupt pending

25.3.15 IMASK Register (Offset = 1058h) [Reset = 00000000h]

IMASK is shown in Table 25-23.

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.

Table 25-23. IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	RX	R/W	0h	Receive FIFO event mask. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
2	RTOUT	R/W	0h	SPI Receive Time-Out event mask. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
1-0	RESERVED	R/W	0h	

25.3.16 RIS Register (Offset = 1060h) [Reset = 00000000h]

RIS is shown in Table 25-24.

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.

Table 25-24. RIS Register 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 0h = Interrupt did not occur 1h = Interrupt occurred
2	RTOUT	R	0h	SPI Receive Time-Out Event. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
1-0	RESERVED	R	0h	

25.3.17 MIS Register (Offset = 1068h) [Reset = 00000000h]

MIS is shown in Table 25-25.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Table 25-25. MIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3	RX	R	0h	Receive FIFO event mask. 0h = Interrupt did not occur 1h = Interrupt occurred
2	RTOUT	R	0h	SPI Receive Time-Out event mask. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
1-0	RESERVED	R	0h	

25.3.18 ISET Register (Offset = 1070h) [Reset = 00000000h]

ISET is shown in Table 25-26.

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.

Table 25-26. ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	W	0h	
3	RX	W	0h	Set Receive FIFO event. 0h = Writing 0 has no effect 1h = Set Interrupt
2	RTOUT	W	0h	Set SPI Receive Time-Out event. 0h = Writing 0 has no effect 1h = Set Interrupt Mask
1-0	RESERVED	W	0h	

25.3.19 ICLR Register (Offset = 1078h) [Reset = 00000000h]

ICLR is shown in Table 25-27.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

Table 25-27. ICLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	W	0h	
3	RX	W	0h	Clear Receive FIFO event. 0h = Writing 0 has no effect 1h = Clear Interrupt
2	RTOUT	W	0h	Clear SPI Receive Time-Out event. 0h = Writing 0 has no effect 1h = Set Interrupt Mask
1-0	RESERVED	W	0h	

25.3.20 IIDX Register (Offset = 1080h) [Reset = 00000000h]

IIDX is shown in Table 25-28.

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.

Table 25-28. IIDX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	STAT	R	0h	Interrupt index status 00h = No interrupt pending 5h = Transmit Event/interrupt pending

25.3.21 IMASK Register (Offset = 1088h) [Reset = 00000000h]

IMASK is shown in Table 25-29.

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.

Table 25-29. IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R/W	0h	
4	TX	R/W	0h	Transmit FIFO event mask. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
3-0	RESERVED	R/W	0h	

25.3.22 RIS Register (Offset = 1090h) [Reset = 00000000h]

RIS is shown in Table 25-30.

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.

Table 25-30. RIS Register 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. 0h = Interrupt did not occur 1h = Interrupt occurred
3-0	RESERVED	R	0h	

25.3.23 MIS Register (Offset = 1098h) [Reset = 00000000h]

MIS is shown in Table 25-31.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Table 25-31. MIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	
4	TX	R	0h	Masked Transmit FIFO event 0h = Interrupt did not occur 1h = Interrupt occurred
3-0	RESERVED	R	0h	

25.3.24 ISET Register (Offset = 10A0h) [Reset = 00000000h]

ISET is shown in Table 25-32.

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.

Table 25-32. ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	W	0h	
4	TX	W	0h	Set Transmit FIFO event. 0h = Writing 0 has no effect 1h = Set Interrupt
3-0	RESERVED	W	0h	

25.3.25 ICLR Register (Offset = 10A8h) [Reset = 00000000h]

ICLR is shown in Table 25-33.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

Table 25-33. ICLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	W	0h	
4	TX	W	0h	Clear Transmit FIFO event. 0h = Writing 0 has no effect 1h = Clear Interrupt
3-0	RESERVED	W	0h	

25.3.26 EVT_MODE Register (Offset = 10E0h) [Reset = 00000000h]

EVT_MODE is shown in Table 25-34.

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)

Table 25-34. EVT_MODE Register 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 none.INT_EVENT2 0h = The interrupt or event line is disabled. 1h = The interrupt or event line is in software mode. Software must clear the RIS. 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.INT_EVENT1 0h = The interrupt or event line is disabled. 1h = The interrupt or event line is in software mode. Software must clear the RIS. 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.INT_EVENT0 0h = The interrupt or event line is disabled. 1h = The interrupt or event line is in software mode. Software must clear the RIS. 2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag.

25.3.27 INTCTL Register (Offset = 10E4h) [Reset = 00000000h]

INTCTL is shown in Table 25-35.

Return to the Summary Table.

Interrupt control register

Table 25-35. INTCTL Register 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. 0h = The interrupt or event line is disabled. 1h = The interrupt or event line is in software mode. Software must clear the RIS.

25.3.28 CTL0 Register (Offset = 1100h) [Reset = 00000000h]

CTL0 is shown in Table 25-36.

Return to the Summary Table.

SPI control register 0

Table 25-36. CTL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R/W	0h	
14	CSCLR	R/W	0h	Clear shift register counter on CS inactive This bit is relevant only in the peripheral, CTL1.CP=0. 0h = Disable automatic clear of shift register when CS goes to disable. 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 0h (R/W) = CS line select: 0 1h (R/W) = CS line select: 1 2h (R/W) = CS line select: 2 3h (R/W) = CS line select: 3
11-10	RESERVED	R/W	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. 0h = Data is captured on the first clock edge transition. 1h = Data is captured on the second clock edge transition.
8	SPO	R/W	0h	CLKOUT polarity (Motorola SPI frame format only) 0h = SPI produces a steady state LOW value on the CLKOUT 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 0h = Packing feature disabled 1h = Packing feature enabled
6-5	FRF	R/W	0h	Frame format Select 0h = Motorola SPI frame format (3 wire mode) 1h = Motorola SPI frame format (4 wire mode) 2h = TI synchronous serial frame format 3h = National Microwire frame format
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 3h (R/W) = Data Size Select bits: 4 4h (R/W) = Data Size Select bits: 5 5h (R/W) = Data Size Select bits: 6 6h (R/W) = Data Size Select bits: 7 7h (R/W) = Data Size Select bits: 8 8h (R/W) = Data Size Select bits: 9 9h (R/W) = Data Size Select bits: 10 Ah (R/W) = Data Size Select bits: 11 Bh (R/W) = Data Size Select bits: 12 Ch (R/W) = Data Size Select bits: 13 Dh (R/W) = Data Size Select bits: 14 Eh (R/W) = Data Size Select bits: 15 Fh (R/W) = Data Size Select bits: 16

25.3.29 CTL1 Register (Offset = 1104h) [Reset = 00000004h]

CTL1 is shown in Table 25-37.

Return to the Summary Table.

SPI control register 1

Table 25-37. CTL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	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. 0h = Smallest value 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. 0h = Smallest value 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. 0h = Smallest value 0h = Manual mode: Data Fh = Manual mode: Command
11	CDENABLE	R/W	0h	Command/Data Mode enable 0h = CS3 is used for Chip Select 1h = CS3 is used as CD signal
10-9	RESERVED	R/W	0h	
8	PTEN	R/W	0h	Parity transmit enable If enabled, parity transmission will be done for both controller and peripheral modes. 0h = Parity transmission is disabled 1h = Parity transmission is enabled
7	RESERVED	R/W	0h	
6	PES	R/W	0h	Even Parity Select 0h = Odd Parity mode 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. 0h = Disable Parity receive function 1h = Enable Parity receive function

Table 25-37. CTL1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	MSB	R/W	0h	MSB first select. Controls the direction of the receive and transmit shift register. 0h = LSB first 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. 0h = SPI can drive the MISO output in peripheral mode. 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. 0h = Select Peripheral mode 1h = Select Controller Mode
1	LBM	R/W	0h	Loop back mode 0h = Disable loopback mode 1h = Enable loopback mode
0	ENABLE	R/W	0h	SPI enable 0h = Disable module function 1h = Enable module function

25.3.30 CLKCTL Register (Offset = 1108h) [Reset = 00000000h]

CLKCTL is shown in Table 25-38.

Return to the Summary Table.

Clock prescaler and divider register. This register contains the settings for the Clock prescaler and divider settings.

Table 25-38. CLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	DSAMPLE	R/W	0h	Delayed sampling value. 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. Note: High values of DSAMPLE can cause HOLD time violations and must be factored in the calculations. 0h = Smallest value Fh = Highest possible value
27-10	RESERVED	R/W	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. 0h = Smallest value 3FFh = Highest possible value

25.3.31 IFLS Register (Offset = 110Ch) [Reset = 00000012h]

IFLS is shown in Table 25-39.

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.

Table 25-39. IFLS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R/W	0h	
5-3	RXIFLSEL	R/W	2h	SPI Receive Interrupt FIFO Level Select The trigger points for the receive interrupt are as follows: 0h = Reserved 1h = RX FIFO >= 1/4 full 2h = RX FIFO >= 1/2 full (default) 3h = RX FIFO >= 3/4 full 4h = Reserved 5h = RX FIFO is full 6h = Reserved 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: 0h = Reserved 1h = TX FIFO <= 3/4 empty 2h = TX FIFO <= 1/2 empty (default) 3h = TX FIFO <= 1/4 empty 4h = Reserved 5h = TX FIFO is empty 6h = Reserved 7h = Trigger when TX FIFO has >= 1 frame free Should be used with DMA

25.3.32 STAT Register (Offset = 1110h) [Reset = 00000000h]

STAT is shown in Table 25-40.

Return to the Summary Table.

Status Register

Table 25-40. STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	
4	BUSY	R	0h	Busy 0h = SPI is in idle mode. 1h = SPI is currently transmitting and/or receiving data, or transmit FIFO is not empty.
3	RNF	R	1h	Receive FIFO not full 0h = Receive FIFO is full. 1h = Receive FIFO is not full.
2	RFE	R	1h	Receive FIFO empty. 0h = Receive FIFO is not empty. 1h = Receive FIFO is empty.
1	TNF	R	1h	Transmit FIFO not full 0h = Transmit FIFO is full. 1h = Transmit FIFO is not full.
0	TFE	R	1h	Transmit FIFO empty. 0h = Transmit FIFO is not empty. 1h = Transmit FIFO is empty.

25.3.33 RXDATA Register (Offset = 1130h) [Reset = 00000000h]

RXDATA is shown in Table 25-41.

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.

Table 25-41. 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 receive FIFO, pointed to by the current FIFO write pointer. Received data less than 16 bits is automatically right justified in the receive buffer. 0h = Smallest value FFFFFFFFh = Highest possible value

25.3.34 TXDATA Register (Offset = 1140h) [Reset = 00000000h]

TXDATA is shown in Table 25-42.

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

Table 25-42. TXDATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R/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 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. 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. 0h = Smallest value FFFFFFFFh = Highest possible value

25.3.35 QSPICTL0 Register (Offset = 1200h) [Reset = X0000000h]

QSPICTL0 is shown in Table 25-43.

Return to the Summary Table.

QSPI Control-0 register
This register must be accessed using a minimum write width of 32.

Table 25-43. QSPICTL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	QSPIMODE	R	0h	This bits selects which port is used for QSPI memory access 0h = Selects peripheral port for external flash access 1h = Selects direct port for external flash access
30	RESERVED	R/W	0h	
29	QSPIPERFMODE	R/W	0h	This bit enables or disables the performance mode byte insertion 0h = Performance mode is disabled 1h = Performance mode is enabled
28	QSPIADDRMODE	R/W	0h	The bit field sets the number of address byte to be used for communication with QSPI flash during a data read 0h = Use 3-byte address 1h = Use 4-byte address
27-24	QSPIFORMAT	R/W	0h	The bits provides the bus format to be used 0h = Legacy SPI format full-duplex 1h = SPI format half-duplex when not sending a data read frame 2h = QPI format half-duplex when not sending a data read frame 8h = SSS format when sending a data read frame 9h = SSD format when sending a data read frame Ah = SDD format when sending a data read frame Bh = SSQ format when sending a data read frame Ch = SQQ format when sending a data read frame Dh = QQQ format when sending a data read frame
23-16	QSPIPERFBYTE	R/W	0 h	The register bits provide the performance byte to be used during a flash read 0h = Samllest value FFh = Largest value
15-8	QSPICMDBYTE	R/W	0h	The register bits provide the instruction opcode to be used during a flash read 0h = Smallest value FFh = Largest value
7-6	QSPIDSIZE	R/W	0h	The register bits provide the packing and unpacking option from the FIFO to the flash interface 0h = Sets 8-bit data write and read from FIFO 2h = Sets 16-bit data write and read from FIFO 3h = Sets 32-bit data write and read from FIFO
5-4	RESERVED	R/W	0h	
3-0	QSPIDUMMYCLK	R/W	0h	The bits provides the number of dummy clocks to insert during a flash read operation 0h = Smallest value Fh = Largest possible value

25.3.36 QSPICTL1 Register (Offset = 1204h) [Reset = 00000000h]

QSPICTL1 is shown in Table 25-44.

Return to the Summary Table.

QSPI Control-1 Register
This register must be accessed using a minimum write width of 32.

Table 25-44. QSPICTL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	QSPIPREFETCH	R/W	0h	Controls prefetch of data when the read transaction is started 0h = Prefetch is disabled 1h = Prefetch is enabled
30-24	RESERVED	R/W	0h	
23-16	RXCOUNT	R/W	0h	
15-9	RESERVED	R/W	0h	
8-0	TXCOUNT	R/W	0h	

25.3.37 QSPITIMING Register (Offset = 1208h) [Reset = 00000000h]

QSPITIMING is shown in Table 25-45.

Return to the Summary Table.

The register is controls the timing setting for chip select assertion and de-assertion with respect to the serial clock, sampling delay for the data and chip-select behavior in performance mode
This register must be accessed using a minimum write width of 32.

Table 25-45. QSPITIMING Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	0h	
23-16	CSHIGHTIME	R	0h	When in performance mode, the parameter provides the number of bus clocks for which the CS shgall be high when an address change is detected 0h = Smallest value FFh = Largest value
15-9	RESERVED	R/W	0h	
8	DATASAMPLDLY	R/W	0h	When in QSPI mode, controls the sampling of the received data 0h = Received data is sampled on the rising edge of SCLK 1h = Received data is delayed by 1 SCLK before being sampled
7-5	RESERVED	R/W	0h	
4	CSASSERTDLY	R/W	0h	When in QSPI mode the bit controls the number of serial clock edge after chip select is asserted to start the SCLK 0h = SCLK is started 0.5 internal SCLK after CS is asserted 1h = SCLK is started 1.5 internal SCLK after CS is asserted
3-1	RESERVED	R/W	0h	
0	CSDEASSERTDLY	R/W	0h	When in QSPI mode the bit controls the number of serial clock edge after last SCLK to deassert the chip select 0h = Chip select is deasserted 0.5 internal SCLK after last edge of the external SCLK 1h = Chip select is deasserted 1.5 internal SCLK after last edge of the external SCLK

25.3.38 QSPIFIFOCTL Register (Offset = 120Ch) [Reset = 00000000h]

QSPIFIFOCTL is shown in Table 25-46.

Return to the Summary Table.

The register controls the flush operation in the peripheral FIFO
This register must be accessed using a minimum write width of 32.

Table 25-46. QSPIFIFOCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	W	0h	
1	TXFIFOFLUSH	R-0/W1S	0h	The bit is used to reset the TXFIFO pointers to flush any data from the preceding transaction. 0h = No effect 1h = Resets the TXFIFO and any corresponding status bit. The field will be auto-cleared after TXFIFO is reset
0	RXFIFOFLUSH	R-0/W1S	0h	The bit is used to reset the RXFIFO pointers to flush any data from the preceding transaction. 0h = No effect 1h = Resets the RXFIFO and any corresponding status bit. The field will be auto-cleared after RXFIFO is reset

25.3.39 QSPICSCTL Register (Offset = 1210h) [Reset = 00000000h]

QSPICSCTL is shown in Table 25-47.

Return to the Summary Table.

The register controls the toggling of chip select
This register must be accessed using a minimum write width of 32.

Table 25-47. QSPICSCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	CSHOLDCTL	R/W	0h	Controls the chip select hold feature during an active transaction when QSPIFORMAT is not 0x0 0h = No effect 1h = When written as 1, causes the chip select to be deasserted after the current bus transaction is completed
30-0	RESERVED	R/W	0h	

25.3.40 QSPIDMACTL Register (Offset = 1220h) [Reset = 00000000h]

QSPIDMACTL is shown in Table 25-48.

Return to the Summary Table.

The register is used to generate a soft trigger for the DMA
This register must be accessed using a minimum write width of 32.

Table 25-48. QSPIDMACTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	W	0h	
0	DMATRIG	R-0/W1S	0h	

25.3.41 QSPISTATUS Register (Offset = 1234h) [Reset = 00000000h]

QSPISTATUS is shown in Table 25-49.

Return to the Summary Table.

The register contains the status bits for the QSPI controller and the FIFO
This register must be accessed using a minimum write width of 32.

Table 25-49. QSPISTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7	QSPIIDLE	R	1h	Provides the state of the QSPI controller 0h = The QSPI has an active transaction ongoing or stalled and the chip select is low 1h = The QSPI does not have an active transaction and the chip select is high
6	QSPISTALL	R	0h	The bit indicates the state of the QSPI controller during an active transaction 0h = The QSPI is sending or receiving data bytes from the memory 1h = The QSPI is currently stalled on the TXFIFO empty to have data or RXFIFO empty to have space to receive data
5-4	RESERVED	R	0h	
3	TXFIFONF	R	0h	Transmit FIFO not full status 0h = Transmit FIFO is full 1h = Transmit FIFO is not full
2	TXFIFOE	R	1h	Transmit FIFO empty status 0h = Transmit FIFO is not empty 1h = Transmit FIFO is empty
1	RXFIFONF	R	1h	Receive FIFO not full status 0h = Receive FIFO is full 1h = Receive FIFO is not full
0	RXFIFOE	R	1h	Receive FIFO status empty status 0h = Receive FIFO is not empty 1h = Receive FIFO is empty

Chapter 26

CAN-FD

The Modular Controller Area Network (MCAN) peripheral supports both communication through classic CAN and CAN-FD protocols.

26.1 MCAN Overview 1436
26.2 MCAN Environment 1437
26.3 CAN Network Basics 1438
26.4 MCAN Functional Description 1440
26.5 MCAN Integration 1489
26.6 Interrupt and Event Support 1490
26.7 MCAN Registers 1492

26.1 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.

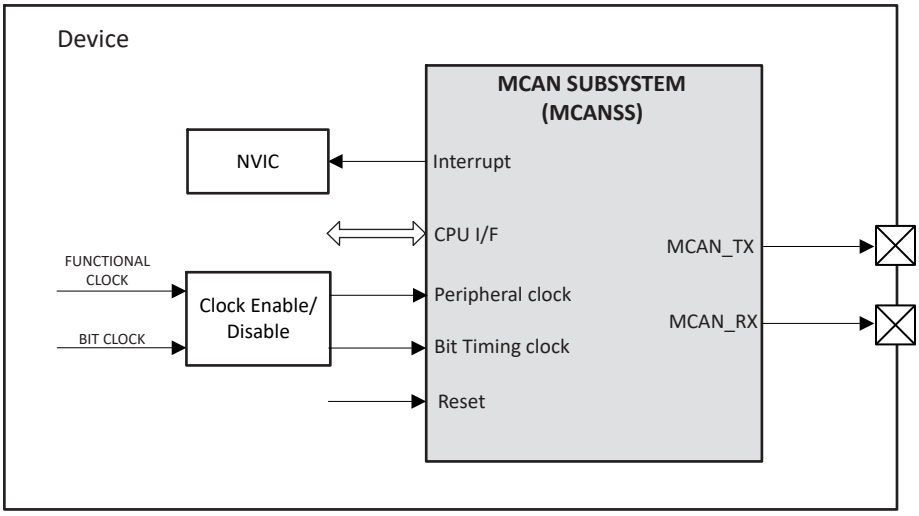


Figure 26-1. MCAN Block Diagram

26.1.1 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 14 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

26.2 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 26-2 shows typical MCAN wiring. Table 26-1 describes the external signals of the MCAN module.

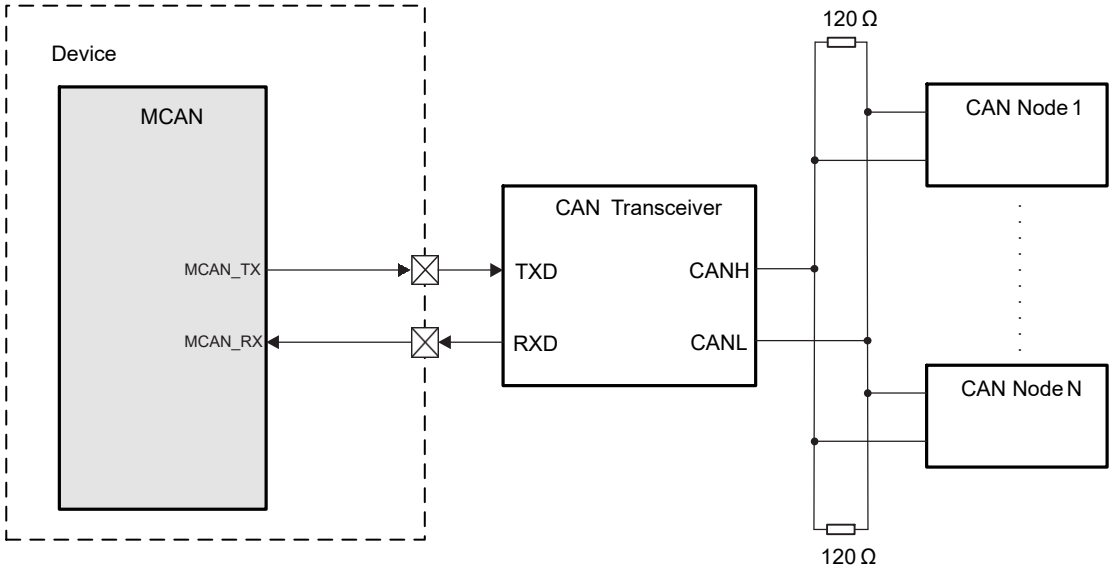


Figure 26-2. MCAN Typical Bus Wiring

Table 26-1. 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.

26.3 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)

- 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 (XTAL or HFCLK_IN) or the SYSPLL (SYSPLLCLK0 or SYSPLLCLK2X).
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).

26.4 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 26-3 shows the MCAN module block diagram, followed by the description of the MCAN module blocks.

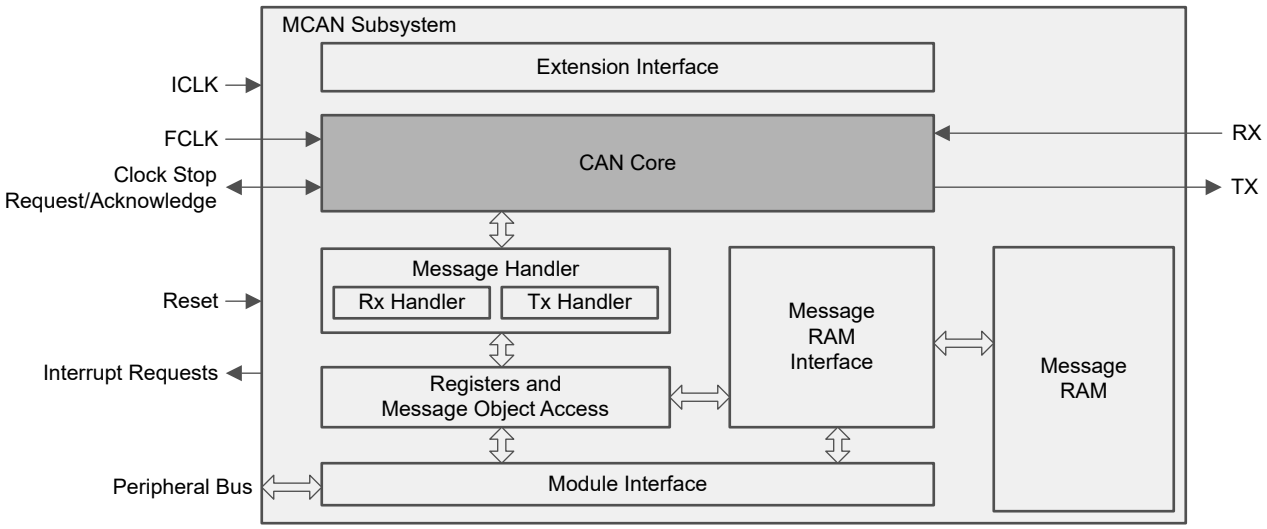


Figure 26-3. MCAN Block Diagram

- **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 26.4.19).
- **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).

26.4.1 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.*

26.4.2 Module Clocking Requirements

Two clocks are provided to the MCAN module:

- Host Clock : peripheral synchronous clock (MCAN_ICLK) as the general module clock source, and
- 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

26.4.3 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 26.4.12.1).

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.

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 26.4.14.2

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.

26.4.4 Operating Modes

The operating modes are discussed in the following sections.

26.4.4.1 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.

26.4.4.2 CAN Classic

CAN supports transmission of data up to 8 bytes with standard (11 bit) or (29 bit) identifier.

26.4.4.3 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 26-4 and described in Table 26-2, 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.

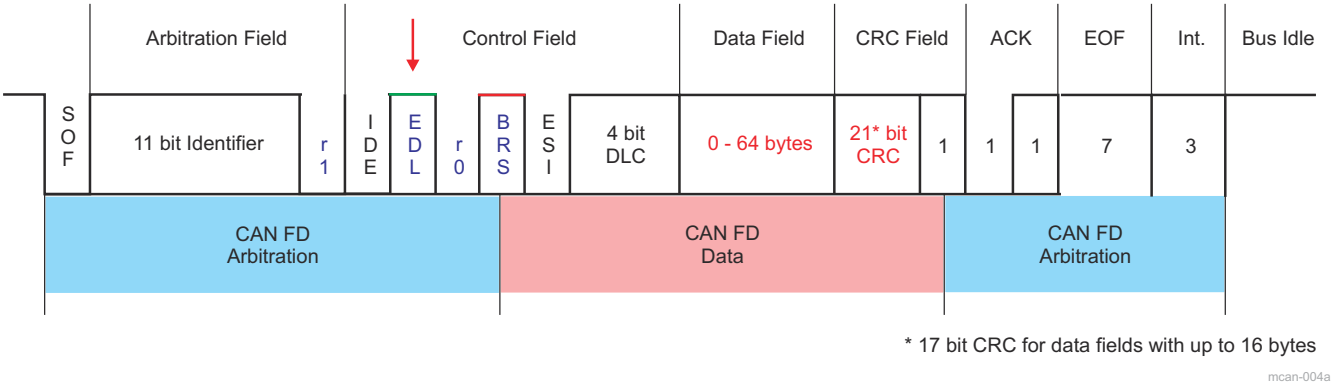


Figure 26-4. CAN FD Frame

Table 26-2. 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:

- 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 26-3.

Table 26-3. 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 26-5) 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.

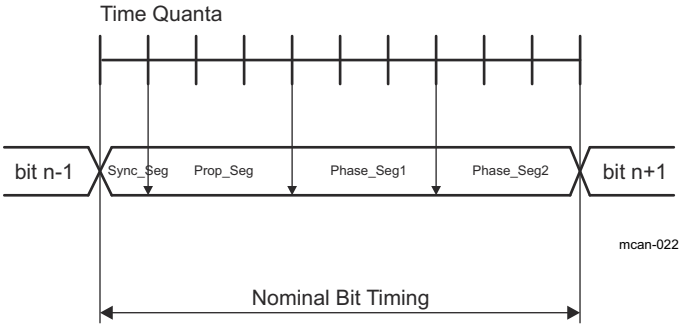


Figure 26-5. CAN Bit Timing

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.

26.4.5 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

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 26-4 shows the steps to configure the MCAN module.

Table 26-4. Steps to Configure MCAN Module

Step	Operation	Description	Pseudo Code
1	Initialize MCAN_CCCR	Set MCAN_CCCR.INIT bit and check that the bit has been set	INIT = 1; 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 FDOE = 0 for Classic CAN
4	Configure Bit Rate Switching	Set MCAN_CCCR.BRSE bit	BRSE = 1 for bit rate switching 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 and check that the bit has been cleared	INIT = 0; 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.

26.4.6 Transmitter Delay Compensation

26.4.6.1 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 26-6). 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.

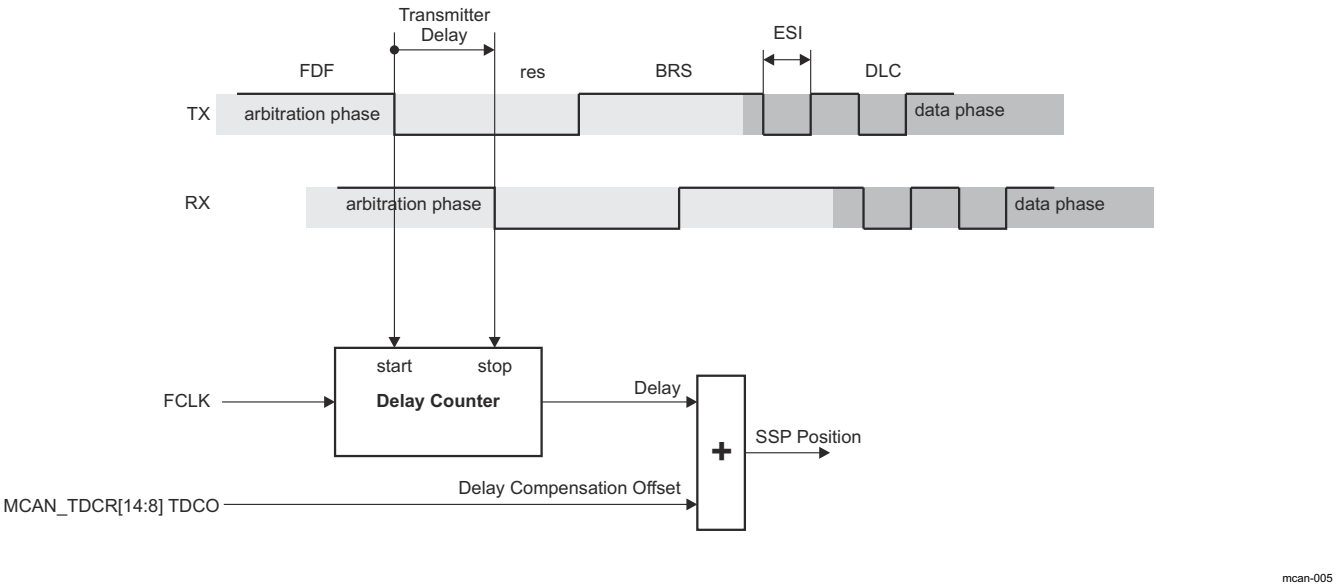


Figure 26-6. Transmitter Delay Measurement

26.4.6.2 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 26-6). 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.

26.4.7 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.

26.4.8 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 26-7 shows the connection of the TX and RX signals to the MCAN module in bus monitoring mode.

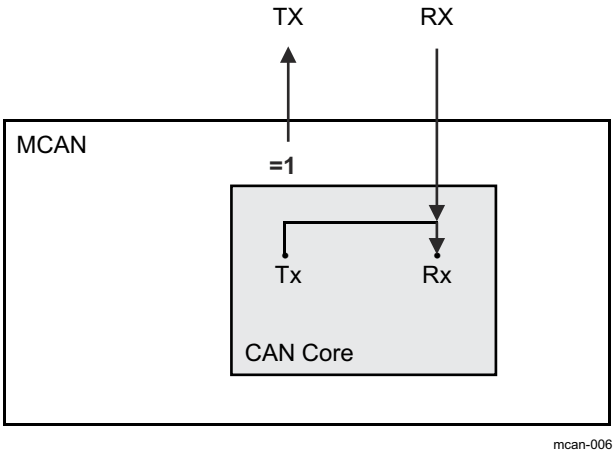


Figure 26-7. Connection of Signals in Bus Monitoring Mode

26.4.9 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).

26.4.9.1 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).

26.4.10 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.

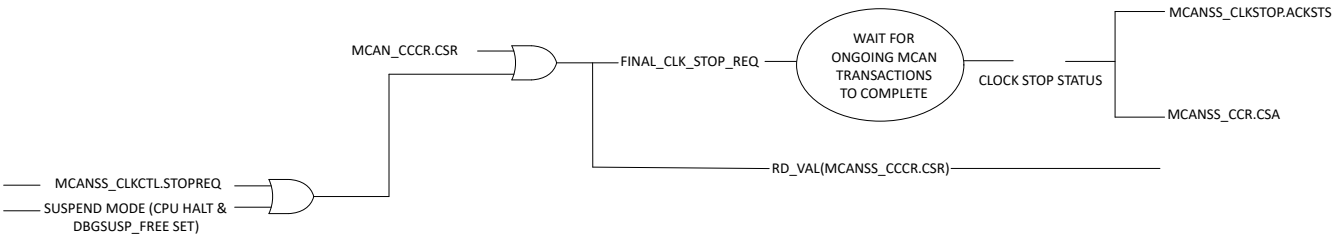


Figure 26-8. Clock Stop Request

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

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 26.4.10.2).

Clock stop and automatic wakeup is illustrated in Figure 26-9.

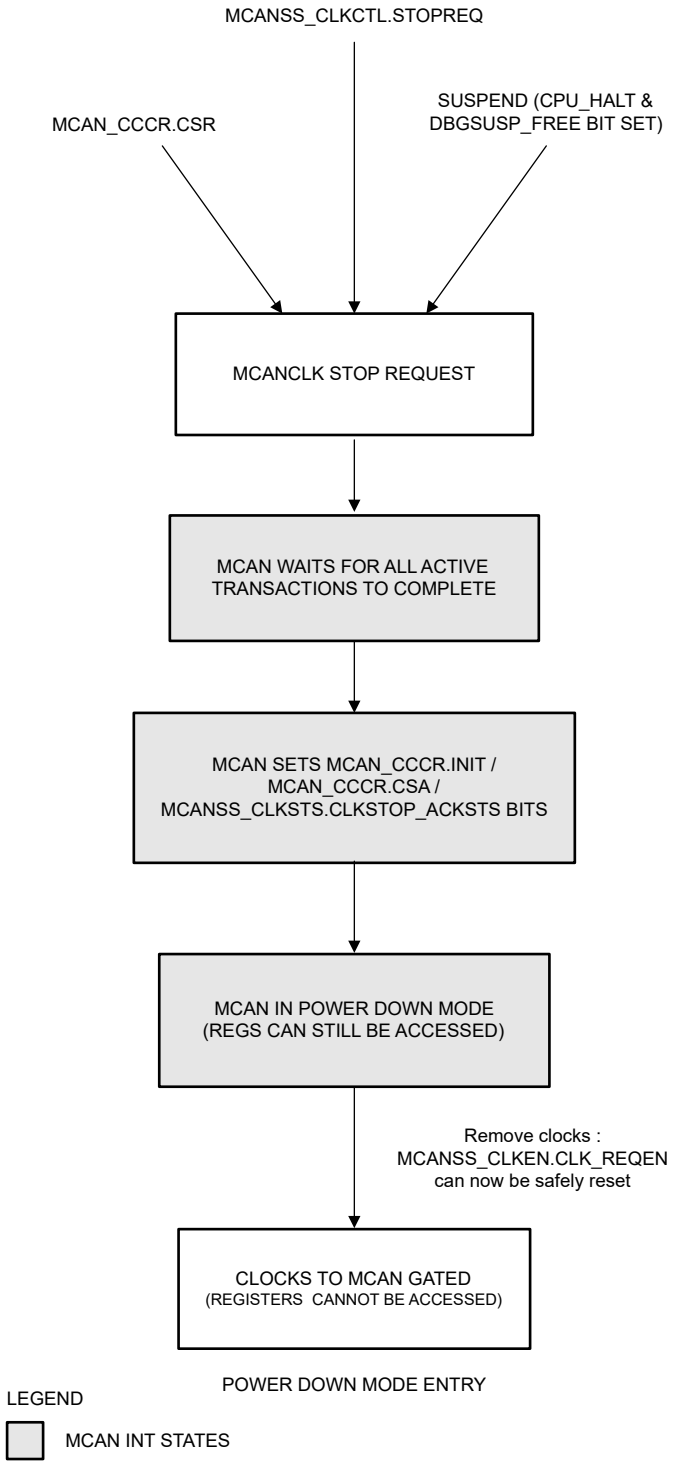


Figure 26-9. Power Down Entry

26.4.10.1 Suspend Mode

The MCAN module supports two suspend modes:

- Immediate
- Graceful

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 26.4.10.2). 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

26.4.10.2 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

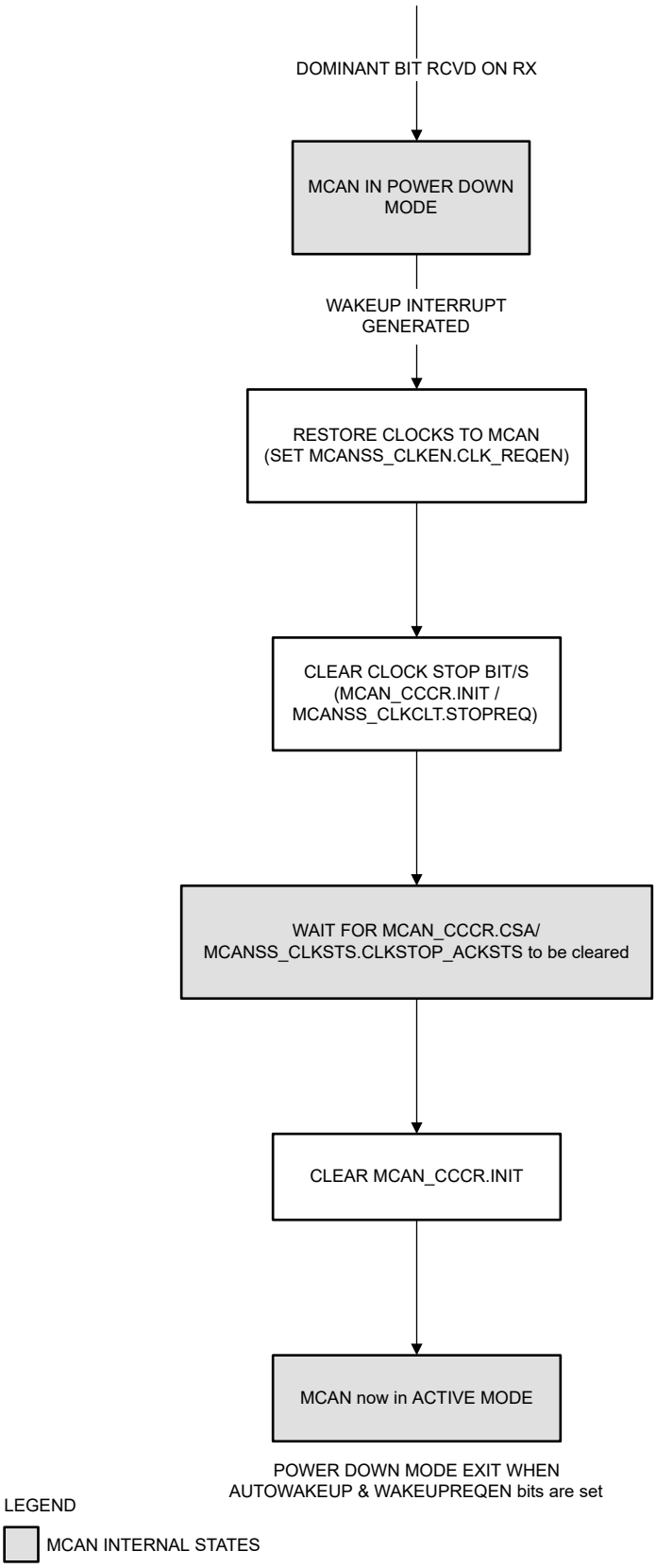


Figure 26-10. Auto Wakeup Enabled Exit from Power Down

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.

26.4.11 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.

26.4.11.1 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.

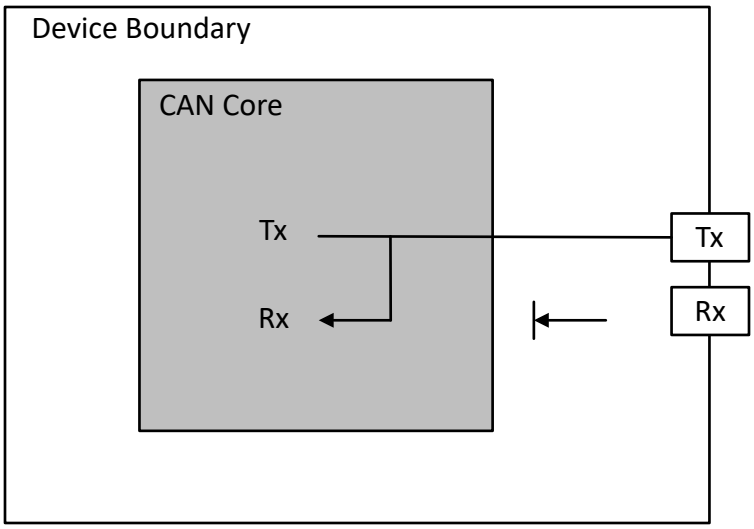


Figure 26-11. External Loop Back Mode

26.4.11.2 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

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 26-12 shows the connection of the TX and RX pins to the MCAN module in internal loop back mode.

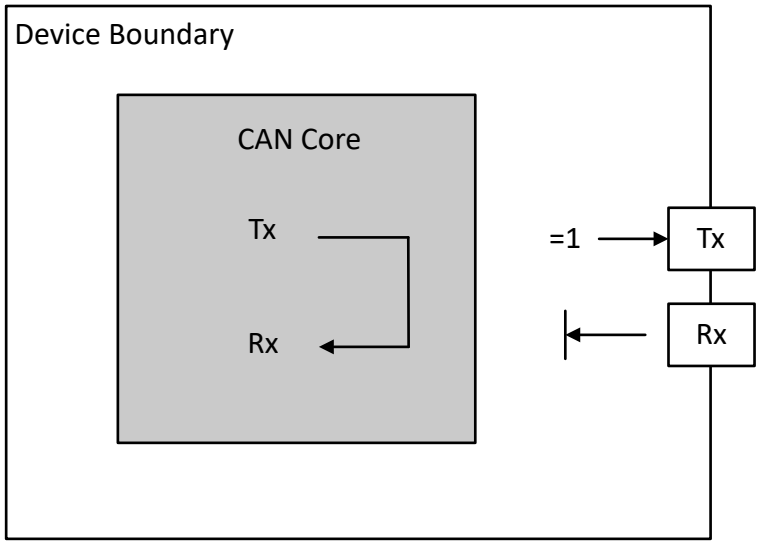


Figure 26-12. Internal Loop Back Mode

26.4.12 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 26.4.19.

26.4.12.1 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.

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.

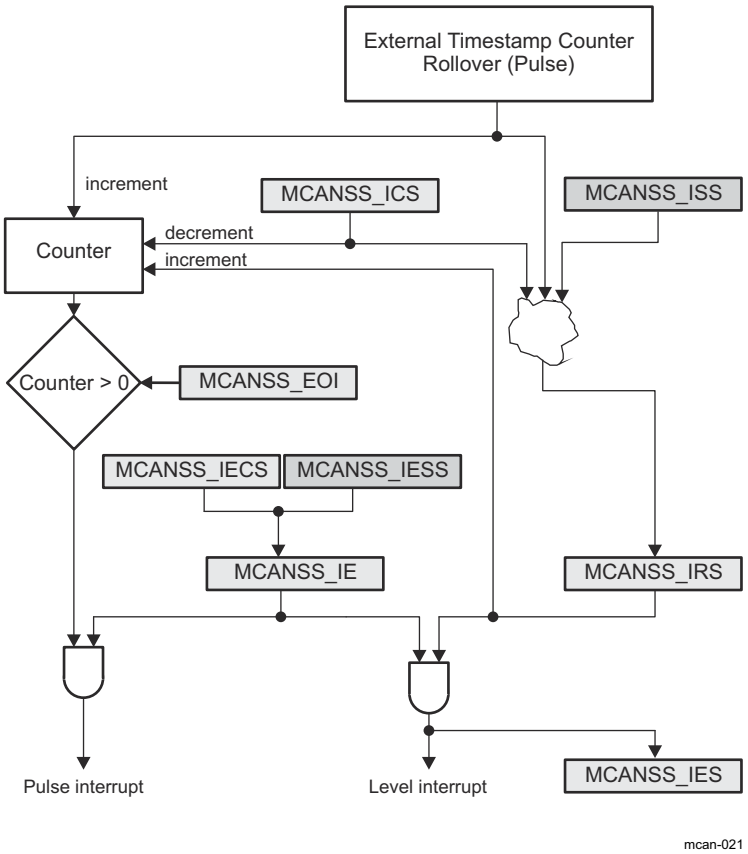


Figure 26-13. External Timestamp Counter Interrupt

26.4.13 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.

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.

26.4.14 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.

26.4.14.1 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.

26.4.14.2 MCAN ECC Aggregator

This section describes the functional details of the MCAN ECC aggregator module.

26.4.14.2.1 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.

26.4.14.2.2 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

26.4.14.3 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.

26.4.14.4 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.

26.4.15 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 26-5. 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

26.4.15.1 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.

26.4.15.2 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 26-6. 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

26.4.15.3 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.

26.4.15.4 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.

26.4.15.5 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.

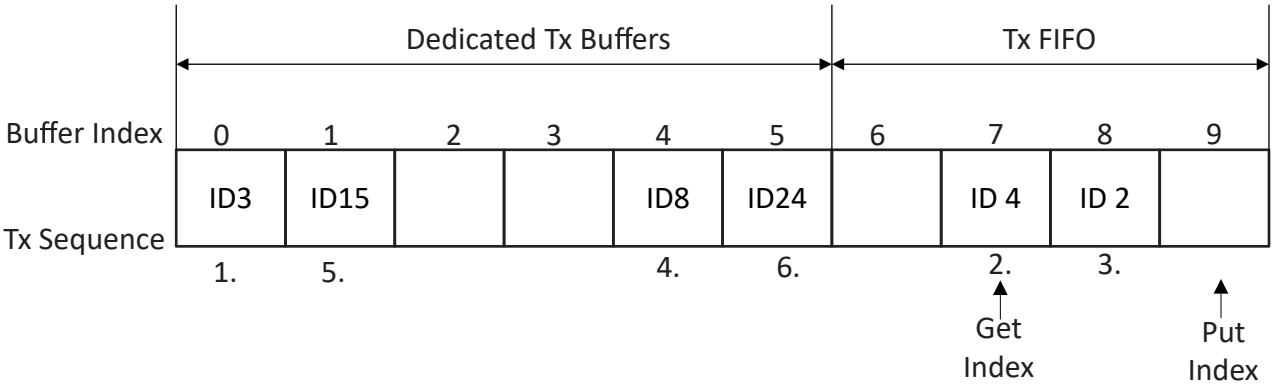


Figure 26-14. Mixed Dedicated Tx Buffers/ Tx FIFO (example)

26.4.15.6 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.

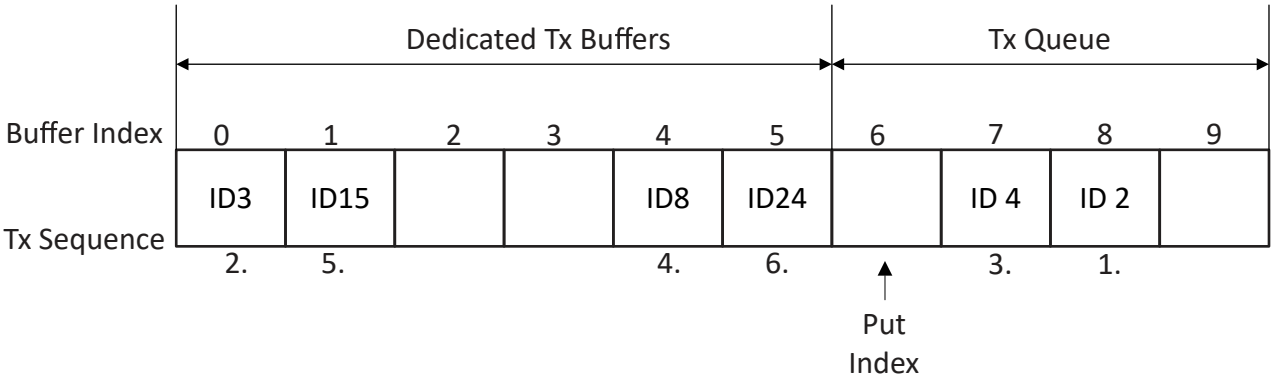


Figure 26-15. Mixed Dedicated Tx Buffers/ Tx FIFO (Example)

26.4.15.7 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.

26.4.15.8 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.

26.4.15.9 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.

26.4.16 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

26.4.16.1 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 26.4.19) 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 26.4.17.2 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.

26.4.16.1.1 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 26.4.19.5 and Section 26.4.19.6.

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.

26.4.16.1.2 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 26.4.19.5 and Section 26.4.19.6.

26.4.16.1.3 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.

26.4.16.1.4 Standard Message ID Filtering

Figure 26-16 shows the standard Message ID (11-bit ID) filtering flow. Section 26.4.19.5 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

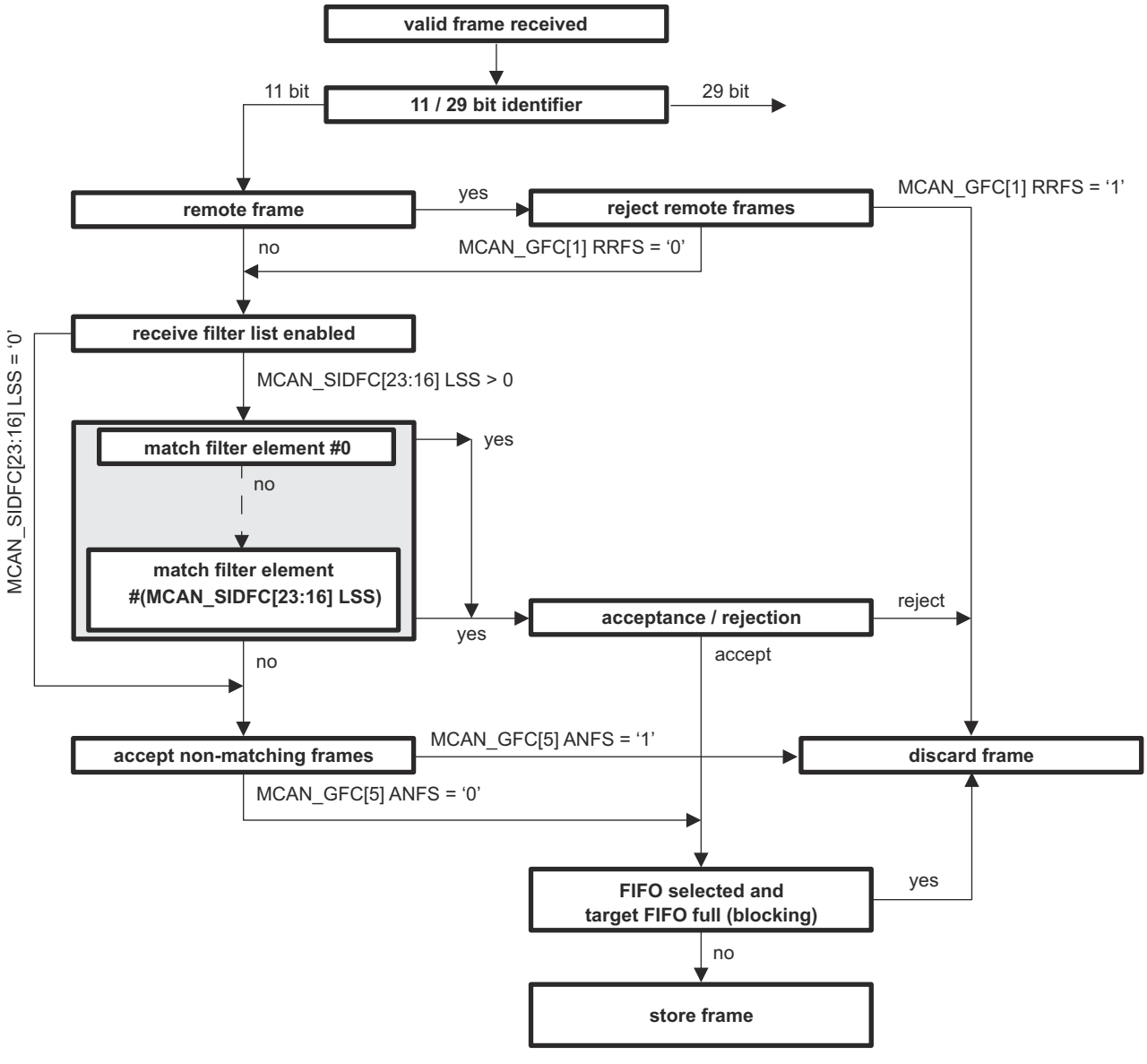


Figure 26-16. Standard Message ID Filter Path

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

26.4.16.1.5 Extended Message ID Filtering

Figure 26-17 shows the extended Message ID (29-bit ID) filtering flow. Section 26.4.19.6 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).

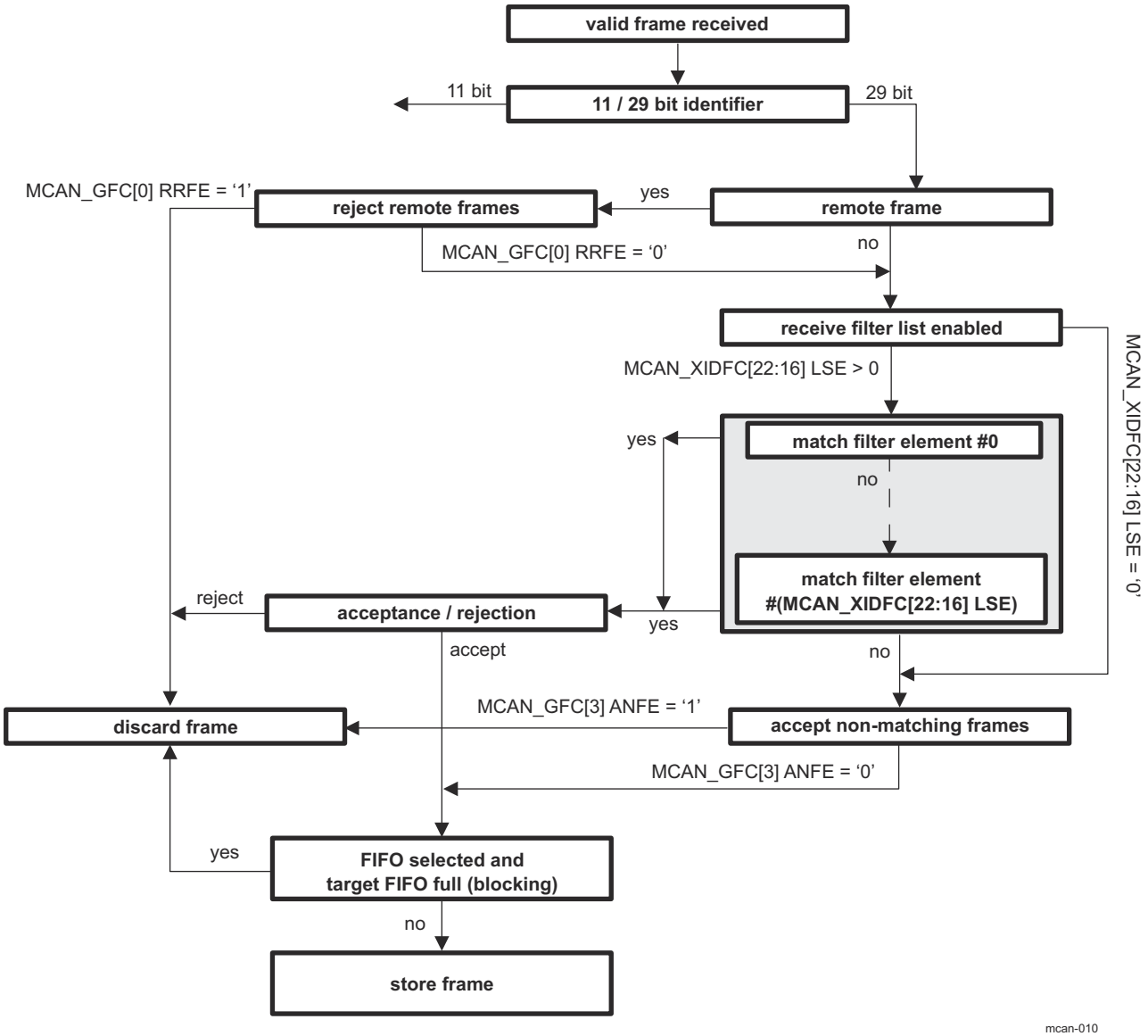


Figure 26-17. Extended Message ID Filter Path

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

26.4.17 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 26.4.16.1. The Rx FIFO element is described in Section 26.4.19.2.

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 26-18 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).

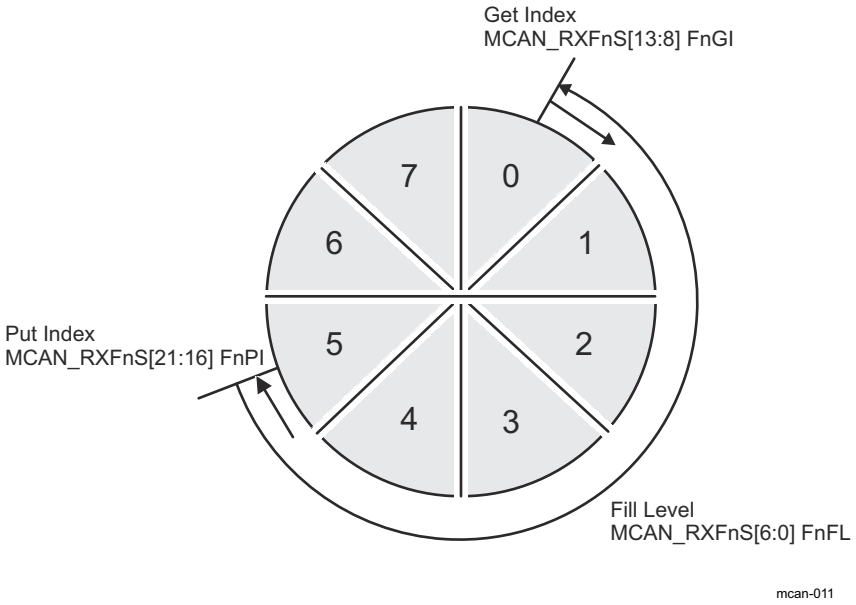


Figure 26-18. Rx FIFO Status

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 26-7 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 26-7. 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

26.4.17.1 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.

26.4.17.2 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 26-19 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).

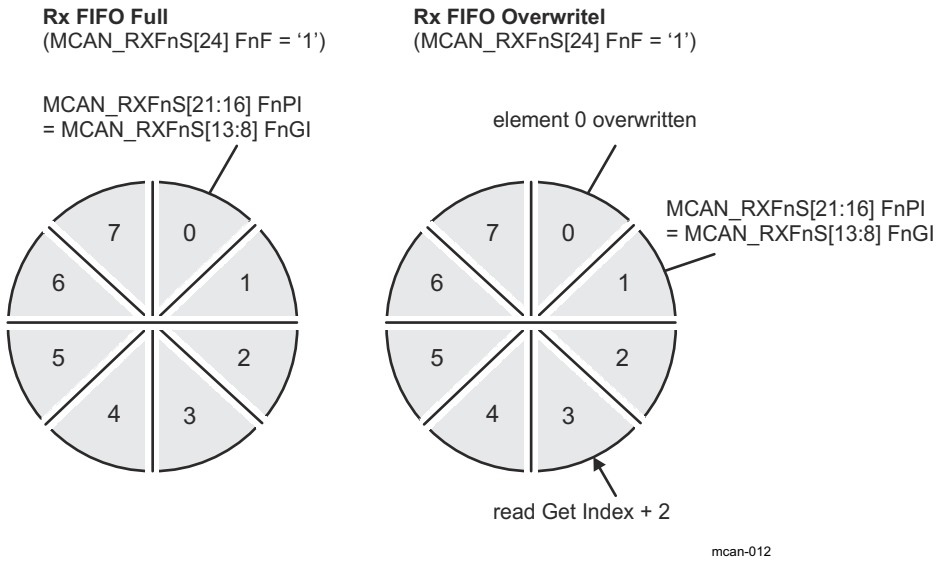


Figure 26-19. Rx FIFO Overflow Handling

26.4.18 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 26.4.19.5 and Section 26.4.19.6).

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 26-8 shows Example Filter Configuration for Rx buffers.

Table 26-8. Example Filter Configuration for Rx Buffers

Filter Element	SFID1[10:0] EFID1[28:0]	SFID2[10:9] EFID2[10:9]	SFID2[5:0] 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.

26.4.18.1 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

26.4.19 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

26.4.19.1 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

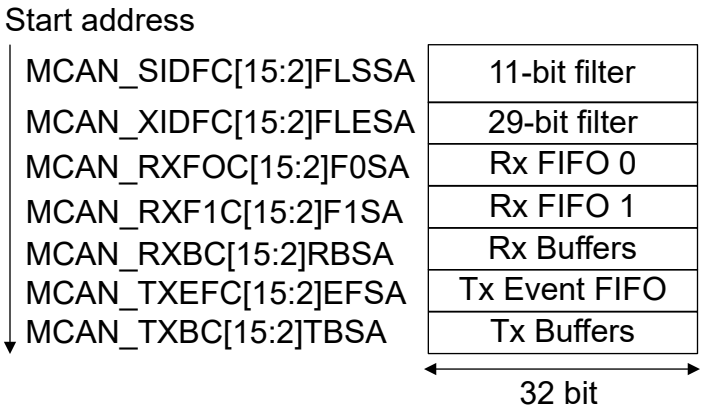


Figure 26-20. Message RAM Configuration

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 26-9. 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

Table 26-9. Message Transmission configurations (continued)

Frame Size, DLC Code (Bytes)	Element Size (Words)	Maximum Elements
64	18	14

26.4.19.2 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 26-21 shows the Rx Buffer/Rx FIFO element structure. Table 26-10 shows the Rx Buffer/Rx FIFO element field descriptions.

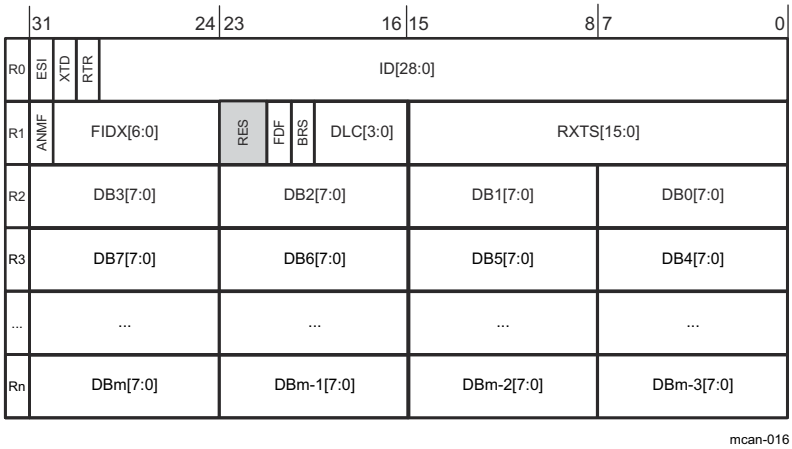


Figure 26-21. Rx Buffer/Rx FIFO Element Structure

Table 26-10. Rx Buffer/Rx FIFO Element Field Descriptions

Word	Bits	Field Name	Description
R0	31	ESI	Error State Indicator • 0x0: Transmitting node is error active • 0x1: Transmitting node is error passive
	30	XTD	Extended Identifier Signals to the Host CPU whether the received frame has a standard or extended identifier. • 0x0: 11-bit standard identifier • 0x1: 29-bit extended identifier
	29	RTR	Remote Transmission Request Signals to the Host CPU whether the received frame is a data frame or a remote frame. • 0x0: Received frame is a data frame • 0x1: Received frame is a remote frame **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 Standard or extended identifier depending on XTD bit. A standard identifier is stored into ID[28:18].

Table 26-10. Rx Buffer/Rx FIFO Element Field Descriptions (continued)

Word	Bits	Field Name	Description
R1	31	ANMF	Accepted Non-matching Frame Acceptance of non-matching frames can be enabled using the MCAN_GFC.ANFS and MCAN_GFC.ANFE fields. • 0x0: Received frame matching filter index FIDX field • 0x1: Received frame did not match any Rx filter element
	30:24	FIDX[6:0]	Filter Index 0x0-0x7F (0-127): Index of matching Rx acceptance filter element (invalid if ANMF = 1). Range is 0 to MCAN_SIDFC.LSS - 1 respectively MCAN_XIDFC.LSE - 1.
	23:22	RES	Reserved
	21	FDF	FD Format • 0x0: Standard frame format • 0x1: CAN FD frame format (new DLC-coding and CRC)
	20	BRS	Bit Rate Switch • 0x0: Frame received without bit rate switching • 0x1: Frame received with bit rate switching
	19:16	DLC[3:0]	Data Length Code • 0x0-0x8 (0-8): CAN + CAN FD: received frame has 0-8 data bytes • 0x9-0xF (9-15): CAN: received frame has 8 data bytes • 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 Timestamp Counter value captured on start of frame reception. 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.

26.4.19.3 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 26-22 shows the Tx Buffer element structure. Table 26-11 shows the Tx Buffer element field descriptions.

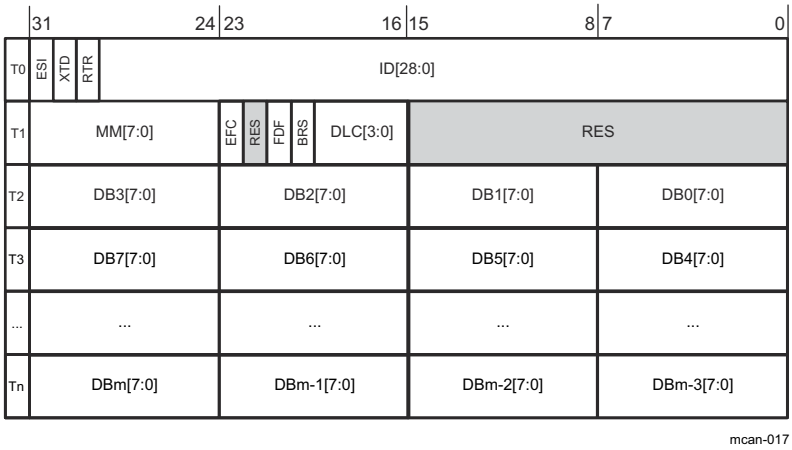


Figure 26-22. Tx Buffer Element Structure

Table 26-11. Tx Buffer Element Field Descriptions

Word	Bits	Field Name	Description
T0	31	ESI	Error State Indicator • 0x0: ESI bit in CAN FD format depends only on error passive flag • 0x1: ESI bit in CAN FD format transmitted recessive **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 • 0x0: 11-bit standard identifier • 0x1: 29-bit extended identifier
	29	RTR	Remote Transmission Request • 0x0: Transmit data frame • 0x1: Transmit remote frame **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 Standard or extended identifier depending on XTD bit. A standard identifier has to be written to ID[28:18].

Table 26-11. Tx Buffer Element Field Descriptions (continued)

Word	Bits	Field Name	Description
T1	31:24	MM[7:0]	Message Marker 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 26-12).
	23	EFC	Event FIFO Control • 0x0: Don't store Tx events • 0x1: Store Tx events
	22	RES	Reserved
	21	FDF	FD Format • 0x0: Frame transmitted in Classic CAN format • 0x1: Frame transmitted in CAN FD format
	20	BRS	Bit Rate Switch • 0x0: CAN FD frames transmitted without bit rate switching • 0x1: CAN FD frames transmitted with bit rate switching **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 • 0x0-0x8 (0-8): CAN + CAN FD: transmit frame has 0-8 data bytes • 0x9-0xF (9-15): CAN: transmit frame has 8 data bytes • 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.

26.4.19.4 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 26-23 shows the Tx Event FIFO element structure. Table 26-12 shows the Tx Event FIFO element field descriptions.

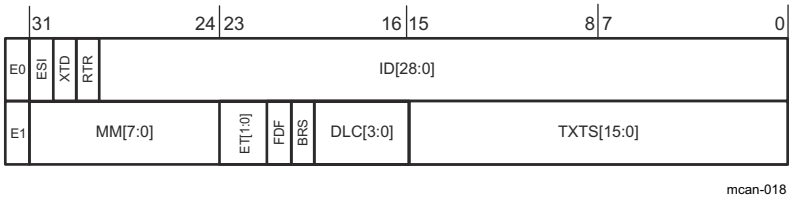


Figure 26-23. Tx Event FIFO Element Structure

Table 26-12. Tx Event FIFO Element Field Descriptions

Word	Bits	Field Name	Description
E0	31	ESI	Error State Indicator • 0x0: Transmitting node is error active • 0x1: Transmitting node is error passive
	30	XTD	Extended Identifier • 0x0: 11-bit standard identifier • 0x1: 29-bit extended identifier
	29	RTR	Remote Transmission Request • 0x0: Data frame transmitted • 0x1: Remote frame transmitted
	28:0	ID[28:0]	Identifier Standard or extended identifier depending on XTD bit. A standard identifier has to be written to ID[28:18].

Table 26-12. Tx Event FIFO Element Field Descriptions (continued)

Word	Bits	Field Name	Description
E1	31:24	MM[7:0]	Message Marker Copied from Tx Buffer into Tx Event FIFO element for identification of Tx message status (see also MM[7:0] field in Table 26-11).
	23:22	ET[1:0]	Event Type • 0x0: Reserved • 0x1: Tx event • 0x2: Transmission in spite of cancellation (always set for transmissions in DAR mode) • 0x3: Reserved
	21	FDF	FD Format • 0x0: Standard frame format • 0x1: CAN FD frame format (new DLC-coding and CRC)
	20	BRS	Bit Rate Switch • 0x0: Frame transmitted without bit rate switching • 0x1: Frame transmitted with bit rate switching
	19:16	DLC[3:0]	Data Length Code • 0x0-0x8 (0-8): CAN + CAN FD: frame with 0-8 data bytes transmitted • 0x9-0xF (9-15): CAN: frame with 8 data bytes transmitted • 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 Timestamp Counter value captured on start of frame transmission. Resolution depending on configuration of the Timestamp Counter Prescaler MCAN_TSCC.TCP filed.

26.4.19.5 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 26-24 shows the Standard Message ID Filter element structure. Table 26-13 shows the Standard Message ID Filter element field descriptions.

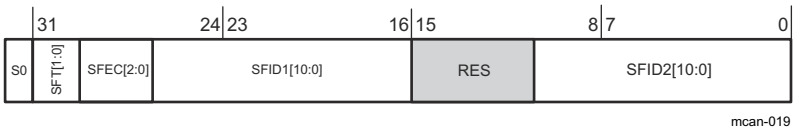


Figure 26-24. Standard Message ID Filter Element Structure

Table 26-13. Standard Message ID Filter Element Field Descriptions

Word	Bits	Field Name	Description
S0	31:30	SFT[1:0]	Standard Filter Type • 0x0: Range filter from SFID1 to SFID2 (SFID2 ≥ SFID1) • 0x1: Dual ID filter for SFID1 or SFID2 • 0x2: Classic filter: SFID1 = filter; SFID2 = mask • 0x3: Filter element disabled **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 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. • 0x0: Disable filter element • 0x1: Store in Rx FIFO 0 if filter matches • 0x2: Store in Rx FIFO 1 if filter matches • 0x3: Reject ID if filter matches • 0x4: Set priority if filter matches • 0x5: Set priority and store in FIFO 0 if filter matches • 0x6: Set priority and store in FIFO 1 if filter matches • 0x7: Store into Rx Buffer , configuration of SFT[1:0] ignored
	26:16	SFID1[10:0]	Standard Filter ID 1 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 This bit field has a different meaning depending on the configuration of SFEC: • SFEC = 001 - 110: Second ID of standard ID filter element • 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. • 0x0: Store message into an Rx Buffer • 0x1: Debug Message A • 0x2: Debug Message B • 0x3: Debug Message C **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. **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.

26.4.19.6 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 26-25 shows the Extended Message ID Filter element structure. Table 26-14 shows the Extended Message ID Filter element field descriptions.

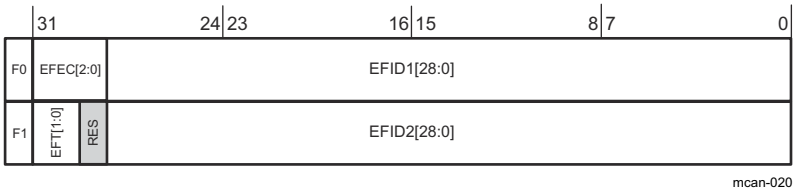


Figure 26-25. Extended Message ID Filter Element Structure

Table 26-14. Extended Message ID Filter Element Field Descriptions

Word	Bits	Field Name	Description
F0	31:29	EFEC[2:0]	Extended Filter Element Configuration 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. • 0x0: Disable filter element • 0x1: Store in Rx FIFO 0 if filter matches • 0x2: Store in Rx FIFO 1 if filter matches • 0x3: Reject ID if filter matches • 0x4: Set priority if filter matches • 0x5: Set priority and store in FIFO 0 if filter matches • 0x6: Set priority and store in FIFO 1 if filter matches • 0x7: Store into Rx Buffer or as debug message, configuration of EFT[1:0] ignored
	28:0	EFID1[28:0]	Extended Filter ID 1 First ID of extended ID filter element. 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 26.4.16.1.5) is used.

Table 26-14. Extended Message ID Filter Element Field Descriptions (continued)

Word	Bits	Field Name	Description
F1	31:30	EFT[1:0]	Extended Filter Type • 0x0: Range filter from EFID1 to EFID2 (EFID2 ≥ EFID1) • 0x1: Dual ID filter for EFID1 or EFID2 • 0x2: Classic filter: EFID1 = filter, EFID2 = mask • 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 This bit field has a different meaning depending on the configuration of EFEC: • EFEC = 001 - 110: Second ID of extended ID filter element • 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. • 0x0: Store message into an Rx Buffer • 0x1: Debug Message A • 0x2: Debug Message B • 0x3: Debug Message C **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. **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.

26.5 MCAN Integration

Figure 26-26 shows the integration of the MCAN module in the device.

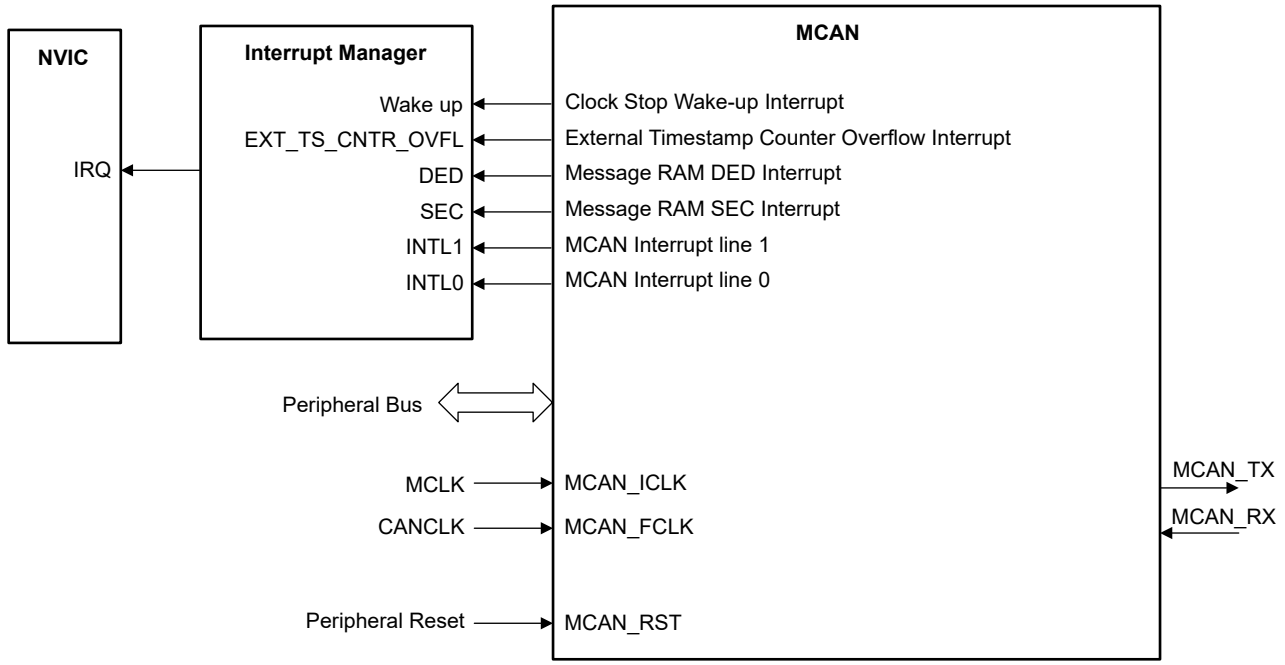


Figure 26-26. MCAN Integration

Table 26-15 summarizes the integration of the MCAN module in the device.

Table 26-15. 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

26.6 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 26-16.

Table 26-16. 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

26.6.1 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 26-17.

Table 26-17. 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

Table 26-17. 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 *Section 8.2.5* for guidance on configuring the event registers for CPU interrupts.

26.7 MCAN Registers

Table 26-18 lists the memory-mapped registers for the MCAN registers. All register offset addresses not listed in Table 26-18 should be considered as reserved locations and the register contents should not be modified.

Table 26-18. 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

Table 26-18. 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

Table 26-18. 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 26-19 shows the codes that are used for access types in this section.

Table 26-19. MCAN Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
RC	R C	Read to Clear
RS	R S	Read to Set
Write Type		

Table 26-19. MCAN Access Type Codes (continued)

Access Type	Code	Description
W	W	Write
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set
W1SQ	W 1S Q	Write 1 to set Qualified. A condition must be met for this operation to occur.
WD	W D	Write Decrement. Decrements the specified bit field by the amount written.
WI	W I	Write Increment. Increments the specified bit field by the amount written.
WK	W K	Write Write protected by a key
WQ	W Q	Write Qualified. A condition must be met for this operation to occur.
Reset or Default Value		
-n		Value after reset or the default value

26.7.1 CANRX (Offset = 6004h) [Reset = 00000000h]

CANRX is shown in Figure 26-27 and described in Table 26-20.

Return to the Summary Table.

CAN RX IO

Figure 26-27. 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 26-20. CANRX Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	
30	GFLT	R/W	0h	Glitch Filter Enable 0h = No internal glitch filter 1h = Use internal glitch filter
29	SLEW	R/W	0h	Reserved Slew Rate Control 0h = No Slew Rate Control 1h = Use Slew Rate Control
28	WCOMP	R/W	0h	Wake up compare value 0h = Match 0 will wake 1h = Match 1 will wake
27	WUEN	R/W	0h	Wake up enable 0h = Wake up not enabled 1h = Wake up enabled
26	INV	R/W	0h	Invert digital input/output relative to peripheral/GPIO 0h = Input and output are non-inverted 1h = Input and output are inverted
25	HIGHZ1	R/W	0h	High-Z instead of high output 0h = Pin can be driven high 1h = Pin is tri-stated instead of driven high
24	HIGHZ0	R/W	0h	High-Z instead of low output 0h = Pin can be driven low 1h = Pin is tri-stated instead of driven low
23	RESERVED	R	0h	
22-20	DRV	R/W	0h	Drive strength options 0h = Lowest drive strength 1h = Drive strength 2/8 2h = Drive strength 3/8 3h = Drive strength 4/8 4h = Drive strength 5/8 5h = Drive strength 6/8 6h = Drive strength 7/8 7h = Highest drive strength

Table 26-20. CANRX Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	HYSTEN	R/W	0h	Hysteresis enable 0h = No hysteresis 1h = Hysteresis on
18	INENA	R/W	0h	Input enable 0h = Inputs 0 to connected core 1h = Inputs IO pad value to connected core
17	PIPU	R/W	0h	Pull up enable 0h = No pull up 1h = Pull up
16	PIPD	R/W	0h	Pull down enable 0h = No pull down 1h = Pull down
15-14	GSTATE	R/W	0h	GPIO Channel State 0h = G-Channel is in Unassigned State 1h = G-Channel is in Handover State 2h = G-Channel is in Connected State and not Locked (That is F field is allowed to change without going back through Unassigned state) 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 0h = P-Channel is in Unassigned State 1h = P-Channel is in Handover State 2h = P-Channel is in Connected State and not Locked (That is F field is allowed to change without going back through Unassigned state) 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	

26.7.2 CANTX (Offset = 6008h) [Reset = 00000000h]

CANTX is shown in Figure 26-28 and described in Table 26-21.

Return to the Summary Table.

CAN TX IO

Figure 26-28. 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 26-21. CANTX Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	
30	GFLT	R/W	0h	Glitch Filter Enable 0h = No internal glitch filter 1h = Use internal glitch filter
29	SLEW	R/W	0h	Reserved Slew Rate Control 0h = No Slew Rate Control 1h = Use Slew Rate Control
28	WCOMP	R/W	0h	Wake up compare value 0h = Match 0 will wake 1h = Match 1 will wake
27	WUEN	R/W	0h	Wake up enable 0h = Wake up not enabled 1h = Wake up enabled
26	INV	R/W	0h	Invert digital input/output relative to peripheral/GPIO 0h = Input and output are non-inverted 1h = Input and output are inverted
25	HIGHZ1	R/W	0h	High-Z instead of high output 0h = Pin can be driven high 1h = Pin is tri-stated instead of driven high
24	HIGHZ0	R/W	0h	High-Z instead of low output 0h = Pin can be driven low 1h = Pin is tri-stated instead of driven low
23	RESERVED	R	0h	
22-20	DRV	R/W	0h	Drive strength options 0h = Lowest drive strength 1h = Drive strength 2/8 2h = Drive strength 3/8 3h = Drive strength 4/8 4h = Drive strength 5/8 5h = Drive strength 6/8 6h = Drive strength 7/8 7h = Highest drive strength

Table 26-21. CANTX Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	HYSTEN	R/W	0h	Hysteresis enable 0h = No hysteresis 1h = Hysteresis on
18	INENA	R/W	0h	Input enable 0h = Inputs 0 to connected core 1h = Inputs IO pad value to connected core
17	PIPU	R/W	0h	Pull up enable 0h = No pull up 1h = Pull up
16	PIPD	R/W	0h	Pull down enable 0h = No pull down 1h = Pull down
15-14	GSTATE	R/W	0h	GPIO Channel State 0h = G-Channel is in Unassigned State 1h = G-Channel is in Handover State 2h = G-Channel is in Connected State and not Locked (That is F field is allowed to change without going back through Unassigned state) 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 0h = P-Channel is in Unassigned State 1h = P-Channel is in Handover State 2h = P-Channel is in Connected State and not Locked (That is F field is allowed to change without going back through Unassigned state) 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	

26.7.3 CANRX (Offset = 6204h) [Reset = 00000000h]

CANRX is shown in Figure 26-29 and described in Table 26-22.

Return to the Summary Table.

FUPDATE version of CANRX

Figure 26-29. 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 26-22. 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 0h = Writing this value has no effect 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 0h = Select the P-Channel for the F update 1h = Select the G-Channel for the F update

26.7.4 CANTX (Offset = 6208h) [Reset = 00000000h]

CANTX is shown in Figure 26-30 and described in Table 26-23.

Return to the Summary Table.

FUPDATE version of CANTX

Figure 26-30. 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 26-23. 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 0h = Writing this value has no effect 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 0h = Select the P-Channel for the F update 1h = Select the G-Channel for the F update

26.7.5 CPU_CONNECT_0 (Offset = 6480h) [Reset = 00000000h]

CPU_CONNECT_0 is shown in Figure 26-31 and described in Table 26-24.

Return to the Summary Table.

Directly connect peripheral publisher port to application processor

Figure 26-31. 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 26-24. CPU_CONNECT_0 Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1	CPUSS0_CONN	R/W	0h	CPUSS0 connect bit. 0h = The CPU is not connected. 1h = The CPU is connected.
0	RESERVED	R	0h	

26.7.6 PWREN (Offset = 6800h) [Reset = 00000000h]

PWREN is shown in Figure 26-32 and described in Table 26-25.

Return to the Summary Table.

Register to control the power state

Figure 26-32. 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 26-25. PWREN Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to allow Power State Change 26h = KEY to allow write access to this register
23-1	RESERVED	R	0h	
0	ENABLE	R/WK	0h	Enable the power KEY must be set to 26h to write to this bit. 0h = Disable Power 1h = Enable Power

26.7.7 RSTCTL (Offset = 6804h) [Reset = 00000000h]

RSTCTL is shown in Figure 26-33 and described in Table 26-26.

Return to the Summary Table.

Register to control reset assertion and de-assertion

Figure 26-33. 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 26-26. RSTCTL Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	Unlock key 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 KEY must be set to B1h to write to this bit. 0h = Writing 0 has no effect 1h = Clear reset sticky bit
0	RESETASSERT	WK	0h	Assert reset to the peripheral KEY must be set to B1h to write to this bit. 0h = Writing 0 has no effect 1h = Assert reset

26.7.8 STAT (Offset = 6814h) [Reset = 00000000h]

STAT is shown in Figure 26-34 and described in Table 26-27.

Return to the Summary Table.

peripheral enable and reset status

Figure 26-34. 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 26-27. 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 0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register 1h = The peripheral was reset since the last bit clear
15-0	RESERVED	R	0h	

26.7.9 MCAN_CREL (Offset = 7000h) [Reset = 32380608h]

MCAN_CREL is shown in Figure 26-35 and described in Table 26-28.

Return to the Summary Table.

MCAN Core Release Register

Figure 26-35. 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 26-28. 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.

26.7.10 MCAN_ENDN (Offset = 7004h) [Reset = 87654321h]

MCAN_ENDN is shown in Figure 26-36 and described in Table 26-29.

Return to the Summary Table.

MCAN Endian Register

Figure 26-36. 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 26-29. 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.

26.7.11 MCAN_DBTP (Offset = 700Ch) [Reset = 00000A33h]

MCAN_DBTP is shown in Figure 26-37 and described in Table 26-30.

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 26-37. 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 26-30. MCAN_DBTP Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23	TDC	R/WQ	0h	Transmitter Delay Compensation 0 Transmitter Delay Compensation disabled 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. 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. 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. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.

Table 26-30. 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. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.

26.7.12 MCAN_TEST (Offset = 7010h) [Reset = 000000X0h]

MCAN_TEST is shown in Figure 26-38 and described in Table 26-31.

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 26-38. 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 26-31. 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. 0h = DOMINANT : The CAN bus is dominant (CAN RX pin = '0') 1h = RECESSIVE : The CAN bus is recessive (CAN RX pin = '1')
6-5	TX	R/WQ	0h	Control of Transmit Pin 00 CAN TX pin controlled by the CAN Core, updated at the end of the CAN bit time 01 Sample Point can be monitored at CAN TX pin 10 Dominant ('0') level at CAN TX pin 11 Recessive ('1') at CAN TX pin 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'. 0h = DISABLE : Reset value, Loop Back Mode is disabled 1h = ENABLE : Loop Back Mode is enabled
3-0	RESERVED	R	0h	

26.7.13 MCAN_RWD (Offset = 7014h) [Reset = 00000000h]

MCAN_RWD is shown in Figure 26-39 and described in Table 26-32.

Return to the Summary Table.

MCAN RAM Watchdog

Figure 26-39. 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 26-32. 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'.

26.7.14 MCAN_CCCR (Offset = 7018h) [Reset = 00000001h]

MCAN_CCCR is shown in Figure 26-40 and described in Table 26-33.

Return to the Summary Table.

MCAN CC Control Register

Figure 26-40. 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 26-33. 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. 0 CAN FD frame format according to ISO 11898-1:2015 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. 0 Transmit pause disabled 1 Transmit pause enabled Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
13	EFBI	R/WQ	0h	Edge Filtering during Bus Integration 0 Edge filtering disabled 1 Two consecutive dominant tq required to detect an edge for hard synchronization Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
12	PXHD	R/WQ	0h	Protocol Exception Handling Disable 0 Protocol exception handling enabled 1 Protocol exception handling disabled Note: When protocol exception handling is disabled, the MCAN will transmit an error frame when it detects a protocol exception condition. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
11-10	RESERVED	R	0h	

Table 26-33. MCAN_CCCR Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	BRSE	R/WQ	0h	Bit Rate Switch Enable 0 Bit rate switching for transmissions disabled 1 Bit rate switching for transmissions enabled Note: When CAN FD operation is disabled FDOE = '0', BRSE is not evaluated. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
8	FDOE	R/WQ	0h	Flexible Datarate Operation Enable 0 FD operation disabled 1 FD operation enabled Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
7	TEST	R/W1SQ	0h	Test Mode Enable 0 Normal operation, register TEST holds reset values 1 Test Mode, write access to register TEST enabled Qualified Write 1 to Set is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
6	DAR	R/WQ	0h	Disable Automatic Retransmission 0 Automatic retransmission of messages not transmitted successfully enabled 1 Automatic retransmission disabled 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. 0 Bus Monitoring Mode is disabled 1 Bus Monitoring Mode is enabled Qualified Write 1 to Set is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
4	CSR	R/W	0h	Clock Stop Request 0 No clock stop is requested 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 0 No clock stop acknowledged 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. 0 Normal CAN operation 1 Restricted Operation Mode active Qualified Write 1 to Set is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
1	CCE	R/WQ	0h	Configuration Change Enable 0 The CPU has no write access to the protected configuration registers 1 The CPU has write access to the protected configuration registers (while CCCR.INIT = '1') Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
0	INIT	R/W	1h	Initialization 0 Normal Operation 1 Initialization is started 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.

26.7.15 MCAN_NBTP (Offset = 701Ch) [Reset = 06000A03h]

MCAN_NBTP is shown in Figure 26-41 and described in Table 26-34.

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 26-41. 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 26-34. 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. 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. 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. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
7	RESERVED	R	0h	

Table 26-34. 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. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.

26.7.16 MCAN_TSCC (Offset = 7020h) [Reset = 00000000h]

MCAN_TSCC is shown in Figure 26-42 and described in Table 26-35.

Return to the Summary Table.

MCAN Timestamp Counter Configuration

Figure 26-42. 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 26-35. 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. Note: With CAN FD an external counter is required for timestamp generation (TSS = '10'). 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 00 Timestamp counter value always 0x0000 01 Timestamp counter value incremented according to TCP 10 External timestamp counter value used 11 Same as '00' Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.

26.7.17 MCAN_TSCV (Offset = 7024h) [Reset = 00000000h]

MCAN_TSCV is shown in Figure 26-43 and described in Table 26-36.

Return to the Summary Table.

MCAN Timestamp Counter Value

Figure 26-43. 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 26-36. 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. Note: A 'wrap around' is a change of the Timestamp Counter value from non-zero to zero not caused by write access to MCAN_TSCV.

26.7.18 MCAN_TOCC (Offset = 7028h) [Reset = FFFF0000h]

MCAN_TOCC is shown in Figure 26-44 and described in Table 26-37.

Return to the Summary Table.

MCAN Timeout Counter Configuration

Figure 26-44. 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 26-37. 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. 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. 00 Continuous operation 01 Timeout controlled by Tx Event FIFO 10 Timeout controlled by Rx FIFO 0 11 Timeout controlled by Rx FIFO 1 Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
0	ETOC	R/WQ	0h	Enable Timeout Counter 0 Timeout Counter disabled 1 Timeout Counter enabled Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.

26.7.19 MCAN_TOCV (Offset = 702Ch) [Reset = 0000FFFFh]

MCAN_TOCV is shown in Figure 26-45 and described in Table 26-38.

Return to the Summary Table.

MCAN Timeout Counter Value

Figure 26-45. 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 26-38. 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.

26.7.20 MCAN_ECR (Offset = 7040h) [Reset = 00000000h]

MCAN_ECR is shown in Figure 26-46 and described in Table 26-39.

Return to the Summary Table.

MCAN Error Counter Register

Figure 26-46. 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 26-39. 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. 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 0 The Receive Error Counter is below the error passive level of 128 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. 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. 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.

26.7.21 MCAN_PSR (Offset = 7044h) [Reset = 00000707h]

MCAN_PSR is shown in Figure 26-47 and described in Table 26-40.

Return to the Summary Table.

MCAN Protocol Status Register

Figure 26-47. 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 26-40. 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 0 No protocol exception event occurred since last read access 1 Protocol exception event occurred
13	RFDF	RC	0h	Received a CAN FD Message. This bit is set independent of acceptance filtering. 0 Since this bit was reset by the CPU, no CAN FD message has been received 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. 0 Last received CAN FD message did not have its BRS flag set 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. 0 Last received CAN FD message did not have its ESI flag set 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 0 The M_CAN is not Bus_Off 1 The M_CAN is in Bus_Off state

Table 26-40. MCAN_PSR Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	EW	R	0h	Warning Status 0 Both error counters are below the Error_Warning limit of 96 1 At least one of error counter has reached the Error_Warning limit of 96
5	EP	R	0h	Error Passive 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 1 The M_CAN is in the Error_Passive state
4-3	ACT	R	0h	Node Activity. Monitors the module's CAN communication state. 00 Synchronizing - node is synchronizing on CAN communication 01 Idle - node is neither receiver nor transmitter 10 Receiver - node is operating as receiver 11 Transmitter - node is operating as transmitter 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. 0 No Error: No error occurred since LEC has been reset by successful reception or transmission. 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. 2 Form Error: A fixed format part of a received frame has the wrong format. 3 AckError: The message transmitted by the MCAN was not acknowledged by another node. 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. 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). 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. 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. 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. 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.

26.7.22 MCAN_TDCR (Offset = 7048h) [Reset = 00000000h]

MCAN_TDCR is shown in Figure 26-48 and described in Table 26-41.

Return to the Summary Table.

MCAN Transmitter Delay Compensation Register

Figure 26-48. 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 26-41. 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. 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. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.

26.7.23 MCAN_IR (Offset = 7050h) [Reset = 80000000h]

MCAN_IR is shown in Figure 26-49 and described in Table 26-42.

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 26-49. 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 26-42. MCAN_IR Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R	0h	
29	ARA	R/W1C	0h	Access to Reserved Address 0 No access to reserved address occurred 1 Access to reserved address occurred
28	PED	R/W1C	0h	Protocol Error in Data Phase (Data Bit Time is used) 0 No protocol error in data phase 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) 0 No protocol error in arbitration phase 1 Protocol error in arbitration phase detected (PSR.LEC ? 0,7)
26	WDI	R/W1C	0h	Watchdog Interrupt 0 No Message RAM Watchdog event occurred 1 Message RAM Watchdog event due to missing READY
25	BO	R/W1C	0h	Bus_Off Status 0 Bus_Off status unchanged 1 Bus_Off status changed
24	EW	R/W1C	0h	Warning Status 0 Error_Warning status unchanged 1 Error_Warning status changed
23	EP	R/W1C	0h	Error Passive 0 Error_Passive status unchanged 1 Error_Passive status changed
22	ELO	R/W1C	0h	Error Logging Overflow 0 CAN Error Logging Counter did not overflow 1 Overflow of CAN Error Logging Counter occurred

Table 26-42. 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. 0 No bit error detected when reading from Message RAM 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. 0 No Rx Buffer updated 1 At least one received message stored into an Rx Buffer
18	TOO	R/W1C	0h	Timeout Occurred 0 No timeout 1 Timeout reached
17	MRAF	R/W1C	0h	Message RAM Access Failure. The flag is set, when the Rx Handler: - 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. - was not able to write a message to the Message RAM. In this case message storage is aborted. 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. 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. 0 No Message RAM access failure occurred 1 Message RAM access failure occurred
16	TSW	R/W1C	0h	Timestamp Wraparound 0 No timestamp counter wrap-around 1 Timestamp counter wrapped around
15	TEFL	R/W1C	0h	Tx Event FIFO Element Lost 0 No Tx Event FIFO element lost 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 0 Tx Event FIFO not full 1 Tx Event FIFO full
13	TEFW	R/W1C	0h	Tx Event FIFO Watermark Reached 0 Tx Event FIFO fill level below watermark 1 Tx Event FIFO fill level reached watermark
12	TEFN	R/W1C	0h	Tx Event FIFO New Entry 0 Tx Event FIFO unchanged 1 Tx Handler wrote Tx Event FIFO element
11	TFE	R/W1C	0h	Tx FIFO Empty 0 Tx FIFO non-empty 1 Tx FIFO empty
10	TCF	R/W1C	0h	Transmission Cancellation Finished 0 No transmission cancellation finished 1 Transmission cancellation finished
9	TC	R/W1C	0h	Transmission Completed 0 No transmission completed 1 Transmission completed

Table 26-42. MCAN_IR Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	HPM	R/W1C	0h	High Priority Message 0 No high priority message received 1 High priority message received
7	RF1L	R/W1C	0h	Rx FIFO 1 Message Lost 0 No Rx FIFO 1 message lost 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 0 Rx FIFO 1 not full 1 Rx FIFO 1 full
5	RF1W	R/W1C	0h	Rx FIFO 1 Watermark Reached 0 Rx FIFO 1 fill level below watermark 1 Rx FIFO 1 fill level reached watermark
4	RF1N	R/W1C	0h	Rx FIFO 1 New Message 0 No new message written to Rx FIFO 1 1 New message written to Rx FIFO 1
3	RF0L	R/W1C	0h	Rx FIFO 0 Message Lost 0 No Rx FIFO 0 message lost 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 0 Rx FIFO 0 not full 1 Rx FIFO 0 full
1	RF0W	R/W1C	0h	Rx FIFO 0 Watermark Reached 0 Rx FIFO 0 fill level below watermark 1 Rx FIFO 0 fill level reached watermark
0	RF0N	R/W1C	0h	Rx FIFO 0 New Message 0 No new message written to Rx FIFO 0 1 New message written to Rx FIFO 0

26.7.24 MCAN_IE (Offset = 7054h) [Reset = 00000000h]

MCAN_IE is shown in Figure 26-50 and described in Table 26-43.

Return to the Summary Table.

MCAN Interrupt Enable

Figure 26-50. 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 26-43. 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 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

Table 26-43. 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

26.7.25 MCAN_ILS (Offset = 7058h) [Reset = 00000000h]

MCAN_ILS is shown in Figure 26-51 and described in Table 26-44.

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 26-51. 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 26-44. MCAN_ILS Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R	0h	
29	ARAL	R/W	0h	Access to Reserved Address Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
28	PEDL	R/W	0h	Protocol Error in Data Phase Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
27	PEAL	R/W	0h	Protocol Error in Arbitration Phase Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
26	WDIL	R/W	0h	Watchdog Interrupt Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
25	BOL	R/W	0h	Bus_Off Status Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
24	EWL	R/W	0h	Warning Status Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
23	EPL	R/W	0h	Error Passive Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
22	ELOL	R/W	0h	Error Logging Overflow Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
21	BEUL	R/W	0h	Bit Error Uncorrected Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1

Table 26-44. MCAN_ILS Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20	BECL	R/W	0h	Bit Error Corrected Line 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 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
18	TOOL	R/W	0h	Timeout Occurred Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
17	MRAFL	R/W	0h	Message RAM Access Failure Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
16	TSWL	R/W	0h	Timestamp Wraparound Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
15	TEFLL	R/W	0h	Tx Event FIFO Element Lost Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
14	TEFFL	R/W	0h	Tx Event FIFO Full Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
13	TEFWL	R/W	0h	Tx Event FIFO Watermark Reached Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
12	TEFNL	R/W	0h	Tx Event FIFO New Entry Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
11	TFEL	R/W	0h	Tx FIFO Empty Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
10	TCFL	R/W	0h	Transmission Cancellation Finished Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
9	TCL	R/W	0h	Transmission Completed Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
8	HPML	R/W	0h	High Priority Message Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
7	RF1LL	R/W	0h	Rx FIFO 1 Message Lost Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
6	RF1FL	R/W	0h	Rx FIFO 1 Full Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
5	RF1WL	R/W	0h	Rx FIFO 1 Watermark Reached Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
4	RF1NL	R/W	0h	Rx FIFO 1 New Message Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
3	RF0LL	R/W	0h	Rx FIFO 0 Message Lost Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1

Table 26-44. MCAN_ILS Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	RF0FL	R/W	0h	Rx FIFO 0 Full Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
1	RF0WL	R/W	0h	Rx FIFO 0 Watermark Reached Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
0	RF0NL	R/W	0h	Rx FIFO 0 New Message Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1

26.7.26 MCAN_ILE (Offset = 705Ch) [Reset = 00000000h]

MCAN_ILE is shown in Figure 26-52 and described in Table 26-45.

Return to the Summary Table.

MCAN Interrupt Line Enable

Figure 26-52. 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 26-45. MCAN_ILE Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1	EINT1	R/W	0h	Enable Interrupt Line 1 0 Interrupt Line 1 is disabled 1 Interrupt Line 1 is enabled
0	EINT0	R/W	0h	Enable Interrupt Line 0 0 Interrupt Line 0 is disabled 1 Interrupt Line 0 is enabled

26.7.27 MCAN_GFC (Offset = 7080h) [Reset = 00000000h]

MCAN_GFC is shown in Figure 26-53 and described in Table 26-46.

Return to the Summary Table.

MCAN Global Filter Configuration

Figure 26-53. 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 26-46. 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. 00 Accept in Rx FIFO 0 01 Accept in Rx FIFO 1 10 Reject 11 Reject 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. 00 Accept in Rx FIFO 0 01 Accept in Rx FIFO 1 10 Reject 11 Reject Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
1	RRFS	R/WQ	0h	Reject Remote Frames Standard 0 Filter remote frames with 11-bit standard IDs 1 Reject all remote frames with 11-bit standard IDs Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
0	RRFE	R/WQ	0h	Reject Remote Frames Extended 0 Filter remote frames with 29-bit extended IDs 1 Reject all remote frames with 29-bit extended IDs Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.

26.7.28 MCAN_SIDFC (Offset = 7084h) [Reset = 00000000h]

MCAN_SIDFC is shown in Figure 26-54 and described in Table 26-47.

Return to the Summary Table.

MCAN Standard ID Filter Configuration

Figure 26-54. 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 26-47. MCAN_SIDFC Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23-16	LSS	R/WQ	0h	List Size Standard 0 No standard Message ID filter 1-128 Number of standard Message ID filter elements >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	

26.7.29 MCAN_XIDFC (Offset = 7088h) [Reset = 00000000h]

MCAN_XIDFC is shown in Figure 26-55 and described in Table 26-48.

Return to the Summary Table.

MCAN Extended ID Filter Configuration

Figure 26-55. 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 26-48. 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). Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
15-2	FLESA	R/WQ	0h	List Size Extended 0 No extended Message ID filter 1-64 Number of extended Message ID filter elements >64 Values greater than 64 are interpreted as 64 Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
1-0	RESERVED	R	0h	

26.7.30 MCAN_XIDAM (Offset = 7090h) [Reset = 1FFFFFFFh]

MCAN_XIDAM is shown in Figure 26-56 and described in Table 26-49.

Return to the Summary Table.

MCAN Extended ID and Mask

Figure 26-56. 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 26-49. 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. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.

26.7.31 MCAN_HPMS (Offset = 7094h) [Reset = 00000000h]

MCAN_HPMS is shown in Figure 26-57 and described in Table 26-50.

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 26-57. 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 26-50. 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. 0 Standard Filter List 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 00 No FIFO selected 01 FIFO message lost 10 Message stored in FIFO 0 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'.

26.7.32 MCAN_NDAT1 (Offset = 7098h) [Reset = 00000000h]

MCAN_NDAT1 is shown in Figure 26-58 and described in Table 26-51.

Return to the Summary Table.

MCAN New Data 1

Figure 26-58. 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 26-51. MCAN_NDAT1 Field Descriptions

Bit	Field	Type	Reset	Description
31	ND31	R/W1C	0h	New Data RX Buffer 31 0 Rx Buffer not updated 1 Rx Buffer updated from new message
30	ND30	R/W1C	0h	New Data RX Buffer 30 0 Rx Buffer not updated 1 Rx Buffer updated from new message
29	ND29	R/W1C	0h	New Data RX Buffer 29 0 Rx Buffer not updated 1 Rx Buffer updated from new message
28	ND28	R/W1C	0h	New Data RX Buffer 28 0 Rx Buffer not updated 1 Rx Buffer updated from new message
27	ND27	R/W1C	0h	New Data RX Buffer 27 0 Rx Buffer not updated 1 Rx Buffer updated from new message
26	ND26	R/W1C	0h	New Data RX Buffer 26 0 Rx Buffer not updated 1 Rx Buffer updated from new message
25	ND25	R/W1C	0h	New Data RX Buffer 25 0 Rx Buffer not updated 1 Rx Buffer updated from new message
24	ND24	R/W1C	0h	New Data RX Buffer 24 0 Rx Buffer not updated 1 Rx Buffer updated from new message
23	ND23	R/W1C	0h	New Data RX Buffer 23 0 Rx Buffer not updated 1 Rx Buffer updated from new message
22	ND22	R/W1C	0h	New Data RX Buffer 22 0 Rx Buffer not updated 1 Rx Buffer updated from new message

Table 26-51. MCAN_NDAT1 Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	ND21	R/W1C	0h	New Data RX Buffer 21 0 Rx Buffer not updated 1 Rx Buffer updated from new message
20	ND20	R/W1C	0h	New Data RX Buffer 20 0 Rx Buffer not updated 1 Rx Buffer updated from new message
19	ND19	R/W1C	0h	New Data RX Buffer 19 0 Rx Buffer not updated 1 Rx Buffer updated from new message
18	ND18	R/W1C	0h	New Data RX Buffer 18 0 Rx Buffer not updated 1 Rx Buffer updated from new message
17	ND17	R/W1C	0h	New Data RX Buffer 17 0 Rx Buffer not updated 1 Rx Buffer updated from new message
16	ND16	R/W1C	0h	New Data RX Buffer 16 0 Rx Buffer not updated 1 Rx Buffer updated from new message
15	ND15	R/W1C	0h	New Data RX Buffer 15 0 Rx Buffer not updated 1 Rx Buffer updated from new message
14	ND14	R/W1C	0h	New Data RX Buffer 14 0 Rx Buffer not updated 1 Rx Buffer updated from new message
13	ND13	R/W1C	0h	New Data RX Buffer 13 0 Rx Buffer not updated 1 Rx Buffer updated from new message
12	ND12	R/W1C	0h	New Data RX Buffer 12 0 Rx Buffer not updated 1 Rx Buffer updated from new message
11	ND11	R/W1C	0h	New Data RX Buffer 11 0 Rx Buffer not updated 1 Rx Buffer updated from new message
10	ND10	R/W1C	0h	New Data RX Buffer 10 0 Rx Buffer not updated 1 Rx Buffer updated from new message
9	ND9	R/W1C	0h	New Data RX Buffer 9 0 Rx Buffer not updated 1 Rx Buffer updated from new message
8	ND8	R/W1C	0h	New Data RX Buffer 8 0 Rx Buffer not updated 1 Rx Buffer updated from new message
7	ND7	R/W1C	0h	New Data RX Buffer 7 0 Rx Buffer not updated 1 Rx Buffer updated from new message
6	ND6	R/W1C	0h	New Data RX Buffer 6 0 Rx Buffer not updated 1 Rx Buffer updated from new message
5	ND5	R/W1C	0h	New Data RX Buffer 5 0 Rx Buffer not updated 1 Rx Buffer updated from new message
4	ND4	R/W1C	0h	New Data RX Buffer 4 0 Rx Buffer not updated 1 Rx Buffer updated from new message
3	ND3	R/W1C	0h	New Data RX Buffer 3 0 Rx Buffer not updated 1 Rx Buffer updated from new message

Table 26-51. MCAN_NDAT1 Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	ND2	R/W1C	0h	New Data RX Buffer 2 0 Rx Buffer not updated 1 Rx Buffer updated from new message
1	ND1	R/W1C	0h	New Data RX Buffer 1 0 Rx Buffer not updated 1 Rx Buffer updated from new message
0	ND0	R/W1C	0h	New Data RX Buffer 0 0 Rx Buffer not updated 1 Rx Buffer updated from new message

26.7.33 MCAN_NDAT2 (Offset = 709Ch) [Reset = 00000000h]

MCAN_NDAT2 is shown in Figure 26-59 and described in Table 26-52.

Return to the Summary Table.

MCAN New Data 2

Figure 26-59. 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 26-52. MCAN_NDAT2 Field Descriptions

Bit	Field	Type	Reset	Description
31	ND63	R/W1C	0h	New Data RX Buffer 63 0 Rx Buffer not updated 1 Rx Buffer updated from new message
30	ND62	R/W1C	0h	New Data RX Buffer 62 0 Rx Buffer not updated 1 Rx Buffer updated from new message
29	ND61	R/W1C	0h	New Data RX Buffer 61 0 Rx Buffer not updated 1 Rx Buffer updated from new message
28	ND60	R/W1C	0h	New Data RX Buffer 60 0 Rx Buffer not updated 1 Rx Buffer updated from new message
27	ND59	R/W1C	0h	New Data RX Buffer 59 0 Rx Buffer not updated 1 Rx Buffer updated from new message
26	ND58	R/W1C	0h	New Data RX Buffer 58 0 Rx Buffer not updated 1 Rx Buffer updated from new message
25	ND57	R/W1C	0h	New Data RX Buffer 57 0 Rx Buffer not updated 1 Rx Buffer updated from new message
24	ND56	R/W1C	0h	New Data RX Buffer 56 0 Rx Buffer not updated 1 Rx Buffer updated from new message
23	ND55	R/W1C	0h	New Data RX Buffer 55 0 Rx Buffer not updated 1 Rx Buffer updated from new message
22	ND54	R/W1C	0h	New Data RX Buffer 54 0 Rx Buffer not updated 1 Rx Buffer updated from new message

Table 26-52. MCAN_NDAT2 Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	ND53	R/W1C	0h	New Data RX Buffer 53 0 Rx Buffer not updated 1 Rx Buffer updated from new message
20	ND52	R/W1C	0h	New Data RX Buffer 52 0 Rx Buffer not updated 1 Rx Buffer updated from new message
19	ND51	R/W1C	0h	New Data RX Buffer 51 0 Rx Buffer not updated 1 Rx Buffer updated from new message
18	ND50	R/W1C	0h	New Data RX Buffer 50 0 Rx Buffer not updated 1 Rx Buffer updated from new message
17	ND49	R/W1C	0h	New Data RX Buffer 49 0 Rx Buffer not updated 1 Rx Buffer updated from new message
16	ND48	R/W1C	0h	New Data RX Buffer 48 0 Rx Buffer not updated 1 Rx Buffer updated from new message
15	ND47	R/W1C	0h	New Data RX Buffer 47 0 Rx Buffer not updated 1 Rx Buffer updated from new message
14	ND46	R/W1C	0h	New Data RX Buffer 46 0 Rx Buffer not updated 1 Rx Buffer updated from new message
13	ND45	R/W1C	0h	New Data RX Buffer 45 0 Rx Buffer not updated 1 Rx Buffer updated from new message
12	ND44	R/W1C	0h	New Data RX Buffer 44 0 Rx Buffer not updated 1 Rx Buffer updated from new message
11	ND43	R/W1C	0h	New Data RX Buffer 43 0 Rx Buffer not updated 1 Rx Buffer updated from new message
10	ND42	R/W1C	0h	New Data RX Buffer 42 0 Rx Buffer not updated 1 Rx Buffer updated from new message
9	ND41	R/W1C	0h	New Data RX Buffer 41 0 Rx Buffer not updated 1 Rx Buffer updated from new message
8	ND40	R/W1C	0h	New Data RX Buffer 40 0 Rx Buffer not updated 1 Rx Buffer updated from new message
7	ND39	R/W1C	0h	New Data RX Buffer 39 0 Rx Buffer not updated 1 Rx Buffer updated from new message
6	ND38	R/W1C	0h	New Data RX Buffer 38 0 Rx Buffer not updated 1 Rx Buffer updated from new message
5	ND37	R/W1C	0h	New Data RX Buffer 37 0 Rx Buffer not updated 1 Rx Buffer updated from new message
4	ND36	R/W1C	0h	New Data RX Buffer 36 0 Rx Buffer not updated 1 Rx Buffer updated from new message
3	ND35	R/W1C	0h	New Data RX Buffer 35 0 Rx Buffer not updated 1 Rx Buffer updated from new message

Table 26-52. MCAN_NDAT2 Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	ND34	R/W1C	0h	New Data RX Buffer 34 0 Rx Buffer not updated 1 Rx Buffer updated from new message
1	ND33	R/W1C	0h	New Data RX Buffer 33 0 Rx Buffer not updated 1 Rx Buffer updated from new message
0	ND32	R/W1C	0h	New Data RX Buffer 32 0 Rx Buffer not updated 1 Rx Buffer updated from new message

26.7.34 MCAN_RXF0C (Offset = 70A0h) [Reset = 00000000h]

MCAN_RXF0C is shown in Figure 26-60 and described in Table 26-53.

Return to the Summary Table.

MCAN Rx FIFO 0 Configuration

Figure 26-60. 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 26-53. 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. 0 FIFO 0 blocking mode 1 FIFO 0 overwrite mode Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
30-24	F0WM	R/WQ	0h	Rx FIFO 0 Watermark 0 Watermark interrupt disabled 1-64 Level for Rx FIFO 0 watermark interrupt (IR.RF0W) >64 Watermark interrupt disabled 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. 0 No Rx FIFO 0 1-64 Number of Rx FIFO 0 elements >64 Values greater than 64 are interpreted as 64 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). Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
1-0	RESERVED	R	0h	

26.7.35 MCAN_RXF0S (Offset = 70A4h) [Reset = 00000000h]

MCAN_RXF0S is shown in Figure 26-61 and described in Table 26-54.

Return to the Summary Table.

MCAN Rx FIFO 0 Status

Figure 26-61. 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 26-54. 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. 0 No Rx FIFO 0 message lost 1 Rx FIFO 0 message lost, also set after write attempt to Rx FIFO 0 of size zero Note: Overwriting the oldest message when RXF0C.F0OM = '1' will not set this flag.
24	F0F	R	0h	Rx FIFO 0 Full 0 Rx FIFO 0 not full 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.

26.7.36 MCAN_RXF0A (Offset = 70A8h) [Reset = 00000000h]

MCAN_RXF0A is shown in Figure 26-62 and described in Table 26-55.

Return to the Summary Table.

MCAN Rx FIFO 0 Acknowledge

Figure 26-62. 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 26-55. 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.

26.7.37 MCAN_RXBC (Offset = 70ACh) [Reset = 00000000h]

MCAN_RXBC is shown in Figure 26-63 and described in Table 26-56.

Return to the Summary Table.

MCAN Rx Buffer Configuration

Figure 26-63. 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 26-56. 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	

26.7.38 MCAN_RXF1C (Offset = 70B0h) [Reset = 00000000h]

MCAN_RXF1C is shown in Figure 26-64 and described in Table 26-57.

Return to the Summary Table.

MCAN Rx FIFO 1 Configuration

Figure 26-64. 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 26-57. 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. 0 FIFO 1 blocking mode 1 FIFO 1 overwrite mode Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
30-24	F1WM	R/WQ	0h	Rx FIFO 1 Watermark 0 Watermark interrupt disabled 1-64 Level for Rx FIFO 1 watermark interrupt (IR.RF1W) >64 Watermark interrupt disabled 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. 0 No Rx FIFO 1 1-64 Number of Rx FIFO 1 elements >64 Values greater than 64 are interpreted as 64 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	

26.7.39 MCAN_RXF1S (Offset = 70B4h) [Reset = 00000000h]

MCAN_RXF1S is shown in Figure 26-65 and described in Table 26-58.

Return to the Summary Table.

MCAN Rx FIFO 1 Status

Figure 26-65. 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 26-58. MCAN_RXF1S Field Descriptions

Bit	Field	Type	Reset	Description
31-30	DMS	R	0h	Debug Message Status 00 Idle state, wait for reception of debug messages 01 Debug message A received 10 Debug messages A, B received 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. 0 No Rx FIFO 1 message lost 1 Rx FIFO 1 message lost, also set after write attempt to Rx FIFO 1 of size zero Note: Overwriting the oldest message when RXF1C.F1OM = '1' will not set this flag.
24	F1F	R	0h	Rx FIFO 1 Full 0 Rx FIFO 1 not full 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.

26.7.40 MCAN_RXF1A (Offset = 70B8h) [Reset = 00000000h]

MCAN_RXF1A is shown in Figure 26-66 and described in Table 26-59.

Return to the Summary Table.

MCAN Rx FIFO 1 Acknowledge

Figure 26-66. 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 26-59. 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.

26.7.41 MCAN_RXESC (Offset = 70BCh) [Reset = 00000000h]

MCAN_RXESC is shown in Figure 26-67 and described in Table 26-60.

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 26-67. 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 26-60. 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 000 8 byte data field 001 12 byte data field 010 16 byte data field 011 20 byte data field 100 24 byte data field 101 32 byte data field 110 48 byte data field 111 64 byte data field 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. 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 000 8 byte data field 001 12 byte data field 010 16 byte data field 011 20 byte data field 100 24 byte data field 101 32 byte data field 110 48 byte data field 111 64 byte data field 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. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.

Table 26-60. 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 000 8 byte data field 001 12 byte data field 010 16 byte data field 011 20 byte data field 100 24 byte data field 101 32 byte data field 110 48 byte data field 111 64 byte data field 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. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.

26.7.42 MCAN_TXBC (Offset = 70C0h) [Reset = 00000000h]

MCAN_TXBC is shown in Figure 26-68 and described in Table 26-61.

Return to the Summary Table.

MCAN Tx Buffer Configuration

Figure 26-68. 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 26-61. MCAN_TXBC Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	
30	TFQM	R/WQ	0h	Tx FIFO/Queue Mode 0 Tx FIFO operation 1 Tx Queue operation Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
29-24	TFQS	R/WQ	0h	Transmit FIFO/Queue Size 0 No Tx FIFO/Queue 1-32 Number of Tx Buffers used for Tx FIFO/Queue >32 Values greater than 32 are interpreted as 32 Note: Be aware that the sum of TFQS and NDTB may be not greater than 32. There is no check for erroneous configurations. The Tx Buffers section in the Message RAM starts with the dedicated Tx Buffers. 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 0 No Dedicated Tx Buffers 1-32 Number of Dedicated Tx Buffers >32 Values greater than 32 are interpreted as 32 Note: Be aware that the sum of TFQS and NDTB may be not greater than 32. There is no check for erroneous configurations. The Tx Buffers section in the Message RAM starts with the dedicated Tx Buffers. 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). Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.

Table 26-61. MCAN_TXBC Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	RESERVED	R	0h	

26.7.43 MCAN_TXFQS (Offset = 70C4h) [Reset = 00000000h]

MCAN_TXFQS is shown in Figure 26-69 and described in Table 26-62.

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 26-69. 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 26-62. MCAN_TXFQS Field Descriptions

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21	TFQF	R	0h	Tx FIFO/Queue Full 0 Tx FIFO/Queue not full 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. 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'). 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').

26.7.44 MCAN_TXESC (Offset = 70C8h) [Reset = 00000000h]

MCAN_TXESC is shown in Figure 26-70 and described in Table 26-63.

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 26-70. 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 26-63. 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 000 8 byte data field 001 12 byte data field 010 16 byte data field 011 20 byte data field 100 24 byte data field 101 32 byte data field 110 48 byte data field 111 64 byte data field 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). Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.

26.7.45 MCAN_TXBRP (Offset = 70CCh) [Reset = 00000000h]

MCAN_TXBRP is shown in Figure 26-71 and described in Table 26-64.

Return to the Summary Table.

MCAN Tx Buffer Request Pending

Figure 26-71. 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 26-64. 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.

Table 26-64. 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. Each Tx Buffer has its own Transmission Request Pending bit. The bits are set via register TXBAR. The bits are reset after a requested transmission has completed or has been canceled via register TXBCR. TXBRP bits are set only for those Tx Buffers configured via TXBC. 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). 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. After a cancellation has been requested, a finished cancellation is signaled via TXBCF - after successful transmission together with the corresponding TXBTO bit - when the transmission has not yet been started at the point of cancellation - when the transmission has been aborted due to lost arbitration - when an error occurred during frame transmission In DAR mode all transmissions are automatically canceled if they are not successful. The corresponding TXBCF bit is set for all unsuccessful transmissions. 0 No transmission request pending 1 Transmission request pending 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.

26.7.46 MCAN_TXBAR (Offset = 70D0h) [Reset = 00000000h]

MCAN_TXBAR is shown in Figure 26-72 and described in Table 26-65.

Return to the Summary Table.

MCAN Tx Buffer Add Request

Figure 26-72. 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 26-65. 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.

Table 26-65. 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. 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. 0 No transmission request added 1 Transmission requested added 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. Qualified Write is possible only with CCCR.CCE='0'

26.7.47 MCAN_TXBCR (Offset = 70D4h) [Reset = 00000000h]

MCAN_TXBCR is shown in Figure 26-73 and described in Table 26-66.

Return to the Summary Table.

MCAN Tx Buffer Cancellation Request

Figure 26-73. 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 26-66. 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.

Table 26-66. 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. 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. 0 No cancellation pending 1 Cancellation pending Qualified Write is possible only with CCCR.CCE='0'

26.7.48 MCAN_TXBTO (Offset = 70D8h) [Reset = 00000000h]

MCAN_TXBTO is shown in Figure 26-74 and described in Table 26-67.

Return to the Summary Table.

MCAN Tx Buffer Transmission Occurred

Figure 26-74. 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 26-67. 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.

Table 26-67. 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. 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. 0 No transmission occurred 1 Transmission occurred

26.7.49 MCAN_TXBCF (Offset = 70DCh) [Reset = 00000000h]

MCAN_TXBCF is shown in Figure 26-75 and described in Table 26-68.

Return to the Summary Table.

MCAN Tx Buffer Cancellation Finished

Figure 26-75. 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 26-68. 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.

Table 26-68. 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. 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. 0 No transmit buffer cancellation 1 Transmit buffer cancellation finished

26.7.50 MCAN_TXBTIE (Offset = 70E0h) [Reset = 00000000h]

MCAN_TXBTIE is shown in Figure 26-76 and described in Table 26-69.

Return to the Summary Table.

MCAN Tx Buffer Transmission Interrupt Enable

Figure 26-76. 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 26-69. 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. 0 Transmission interrupt disabled 1 Transmission interrupt enable
30	TIE30	R/W	0h	Transmission Interrupt Enable 30. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
29	TIE29	R/W	0h	Transmission Interrupt Enable 29. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
28	TIE28	R/W	0h	Transmission Interrupt Enable 28. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
27	TIE27	R/W	0h	Transmission Interrupt Enable 27. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
26	TIE26	R/W	0h	Transmission Interrupt Enable 26. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
25	TIE25	R/W	0h	Transmission Interrupt Enable 25. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
24	TIE24	R/W	0h	Transmission Interrupt Enable 24. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable

Table 26-69. 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. 0 Transmission interrupt disabled 1 Transmission interrupt enable
22	TIE22	R/W	0h	Transmission Interrupt Enable 22. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
21	TIE21	R/W	0h	Transmission Interrupt Enable 21. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
20	TIE20	R/W	0h	Transmission Interrupt Enable 20. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
19	TIE19	R/W	0h	Transmission Interrupt Enable 19. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
18	TIE18	R/W	0h	Transmission Interrupt Enable 18. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
17	TIE17	R/W	0h	Transmission Interrupt Enable 17. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
16	TIE16	R/W	0h	Transmission Interrupt Enable 16. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
15	TIE15	R/W	0h	Transmission Interrupt Enable 15. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
14	TIE14	R/W	0h	Transmission Interrupt Enable 14. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
13	TIE13	R/W	0h	Transmission Interrupt Enable 13. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
12	TIE12	R/W	0h	Transmission Interrupt Enable 12. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
11	TIE11	R/W	0h	Transmission Interrupt Enable 11. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
10	TIE10	R/W	0h	Transmission Interrupt Enable 10. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable

Table 26-69. 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. 0 Transmission interrupt disabled 1 Transmission interrupt enable
8	TIE8	R/W	0h	Transmission Interrupt Enable 8. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
7	TIE7	R/W	0h	Transmission Interrupt Enable 7. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
6	TIE6	R/W	0h	Transmission Interrupt Enable 6. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
5	TIE5	R/W	0h	Transmission Interrupt Enable 5. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
4	TIE4	R/W	0h	Transmission Interrupt Enable 4. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
3	TIE3	R/W	0h	Transmission Interrupt Enable 3. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
2	TIE2	R/W	0h	Transmission Interrupt Enable 2. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
1	TIE1	R/W	0h	Transmission Interrupt Enable 1. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
0	TIE0	R/W	0h	Transmission Interrupt Enable 0. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable

26.7.51 MCAN_TXBCIE (Offset = 70E4h) [Reset = 00000000h]

MCAN_TXBCIE is shown in Figure 26-77 and described in Table 26-70.

Return to the Summary Table.

MCAN Tx Buffer Cancellation Finished Interrupt Enable

Figure 26-77. 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 26-70. 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. 0 Cancellation finished interrupt disabled 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. 0 Cancellation finished interrupt disabled 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. 0 Cancellation finished interrupt disabled 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. 0 Cancellation finished interrupt disabled 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. 0 Cancellation finished interrupt disabled 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. 0 Cancellation finished interrupt disabled 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. 0 Cancellation finished interrupt disabled 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. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled

Table 26-70. 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. 0 Cancellation finished interrupt disabled 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. 0 Cancellation finished interrupt disabled 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. 0 Cancellation finished interrupt disabled 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. 0 Cancellation finished interrupt disabled 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. 0 Cancellation finished interrupt disabled 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. 0 Cancellation finished interrupt disabled 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. 0 Cancellation finished interrupt disabled 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. 0 Cancellation finished interrupt disabled 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. 0 Cancellation finished interrupt disabled 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. 0 Cancellation finished interrupt disabled 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. 0 Cancellation finished interrupt disabled 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. 0 Cancellation finished interrupt disabled 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. 0 Cancellation finished interrupt disabled 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. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled

Table 26-70. 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. 0 Cancellation finished interrupt disabled 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. 0 Cancellation finished interrupt disabled 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. 0 Cancellation finished interrupt disabled 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. 0 Cancellation finished interrupt disabled 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. 0 Cancellation finished interrupt disabled 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. 0 Cancellation finished interrupt disabled 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. 0 Cancellation finished interrupt disabled 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. 0 Cancellation finished interrupt disabled 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. 0 Cancellation finished interrupt disabled 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. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled

26.7.52 MCAN_TXEFC (Offset = 70F0h) [Reset = 00000000h]

MCAN_TXEFC is shown in Figure 26-78 and described in Table 26-71.

Return to the Summary Table.

MCAN Tx Event FIFO Configuration

Figure 26-78. 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 26-71. MCAN_TXEFC Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R	0h	
29-24	EFWM	R/WQ	0h	Event FIFO Watermark 0 Watermark interrupt disabled 1-32 Level for Tx Event FIFO watermark interrupt (IR.TEFW) >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. 0 Tx Event FIFO disabled 1-32 Number of Tx Event FIFO elements >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	

26.7.53 MCAN_TXEFS (Offset = 70F4h) [Reset = 00000000h]

MCAN_TXEFS is shown in Figure 26-79 and described in Table 26-72.

Return to the Summary Table.

MCAN Tx Event FIFO Status

Figure 26-79. 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 26-72. 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. 0 No Tx Event FIFO element lost 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 0 Tx Event FIFO not full 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.

26.7.54 MCAN_TXEFA (Offset = 70F8h) [Reset = 00000000h]

MCAN_TXEFA is shown in Figure 26-80 and described in Table 26-73.

Return to the Summary Table.

MCAN Tx Event FIFO Acknowledge

Figure 26-80. 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 26-73. 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.

26.7.55 MCANSS_PID (Offset = 7200h) [Reset = 68E04901h]

MCANSS_PID is shown in Figure 26-81 and described in Table 26-74.

Return to the Summary Table.

MCAN Subsystem Revision Register

Figure 26-81. 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 26-74. 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

26.7.56 MCANSS_CTRL (Offset = 7204h) [Reset = 00000008h]

MCANSS_CTRL is shown in Figure 26-82 and described in Table 26-75.

Return to the Summary Table.

MCAN Subsystem Control Register

Figure 26-82. 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 26-75. 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. 0 External timestamp counter disabled 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. 0 Disable the automatic write to CCCR.INIT 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. 0 Disable wakeup request 1 Enables wakeup request
3	DBGSUSP_FREE	R/W	1h	Debug Suspend Free Bit. Enables debug suspend. 0 Honor debug suspend 1 Disregard debug suspend
2-0	RESERVED	R	0h	

26.7.57 MCANSS_STAT (Offset = 7208h) [Reset = 0000000Xh]

MCANSS_STAT is shown in Figure 26-83 and described in Table 26-76.

Return to the Summary Table.

MCAN Subsystem Status Register

Figure 26-83. 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 26-76. 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 Reflects the value of mcanss_enable_fdoe configuration port, -h = mcanss_enable_fdoe.
1	MEM_INIT_DONE	R	0h	Memory Initialization Done. 0 Message RAM initialization is in progress 1 Message RAM is initialized for use
0	RESET	R	0h	Soft Reset Status. 0 Not in reset 1 Reset is in progress

26.7.58 MCANSS_ICS (Offset = 720Ch) [Reset = 00000000h]

MCANSS_ICS is shown in Figure 26-84 and described in Table 26-77.

Return to the Summary Table.

MCAN Subsystem Interrupt Clear Shadow Register

Figure 26-84. 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 26-77. 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. 0 Write of '0' has no effect 1 Write of '1' clears the MCANSS_IRS.EXT_TS_CNTR_OVFL bit

26.7.59 MCANSS_IRS (Offset = 7210h) [Reset = 00000000h]

MCANSS_IRS is shown in Figure 26-85 and described in Table 26-78.

Return to the Summary Table.

MCAN Subsystem Interrupt Raw Status Register

Figure 26-85. 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 26-78. 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. 0 External timestamp counter has not overflowed 1 External timestamp counter has overflowed 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.

26.7.60 MCANSS_IECS (Offset = 7214h) [Reset = 00000000h]

MCANSS_IECS is shown in Figure 26-86 and described in Table 26-79.

Return to the Summary Table.

MCAN Subsystem Interrupt Enable Clear Shadow Register

Figure 26-86. 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 26-79. 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. 0 Write of '0' has no effect 1 Write of '1' clears the MCANSS_IES.EXT_TS_CNTR_OVFL bit

26.7.61 MCANSS_IE (Offset = 7218h) [Reset = 00000000h]

MCANSS_IE is shown in Figure 26-87 and described in Table 26-80.

Return to the Summary Table.

MCAN Subsystem Interrupt Enable Register

Figure 26-87. 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 26-80. 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.

26.7.62 MCANSS_IES (Offset = 721Ch) [Reset = 00000000h]

MCANSS_IES is shown in Figure 26-88 and described in Table 26-81.

Return to the Summary Table.

MCAN Subsystem Interrupt Enable Status

Figure 26-88. 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 26-81. 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. 0 External timestamp counter overflow interrupt is not enabled 1 External timestamp counter overflow interrupt is enabled

26.7.63 MCANSS_EOI (Offset = 7220h) [Reset = 00000000h]

MCANSS_EOI is shown in Figure 26-89 and described in Table 26-82.

Return to the Summary Table.

MCAN Subsystem End of Interrupt

Figure 26-89. 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 26-82. 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. 0x00 External TS Interrupt is cleared 0x01 MCAN(0) interrupt is cleared 0x02 MCAN(1) interrupt is cleared Other writes are ignored.

26.7.64 MCANSS_EXT_TS_PRESCALER (Offset = 7224h) [Reset = 00000000h]

MCANSS_EXT_TS_PRESCALER is shown in Figure 26-90 and described in Table 26-83.

Return to the Summary Table.

MCAN Subsystem External Timestamp Prescaler 0

Figure 26-90. 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 26-83. 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.

26.7.65 MCANSS_EXT_TS_UNSERVICED_INTR_CNTR (Offset = 7228h) [Reset = 00000000h]

MCANSS_EXT_TS_UNSERVICED_INTR_CNTR is shown in Figure 26-91 and described in Table 26-84.

Return to the Summary Table.

MCAN Subsystem External Timestamp Unserviced Interrupts Counter

Figure 26-91. 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 26-84. 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. The status of this bit field is affected by the MCANSS_IRS.EXT_TS_CNTR_OVFL bit field.

26.7.66 MCANERR_REV (Offset = 7400h) [Reset = 66A0EA00h]

MCANERR_REV is shown in Figure 26-92 and described in Table 26-85.

Return to the Summary Table.

MCAN Error Aggregator Revision Register

Figure 26-92. 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 26-85. 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

26.7.67 MCANERR_VECTOR (Offset = 7408h) [Reset = 00000000h]

MCANERR_VECTOR is shown in Figure 26-93 and described in Table 26-86.

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 26-93. 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 26-86. 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. 0x000 Message RAM ECC controller is selected Others Reserved (do not use) 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.

26.7.68 MCANERR_STAT (Offset = 740Ch) [Reset = 00000002h]

MCANERR_STAT is shown in Figure 26-94 and described in Table 26-87.

Return to the Summary Table.

MCAN Error Misc Status

Figure 26-94. 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 26-87. 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.

26.7.69 MCANERR_WRAP_REV (Offset = 7410h) [Reset = 66A46A02h]

MCANERR_WRAP_REV is shown in Figure 26-95 and described in Table 26-88.

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 26-95. 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 26-88. 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

26.7.70 MCANERR_CTRL (Offset = 7414h) [Reset = 00000187h]

MCANERR_CTRL is shown in Figure 26-96 and described in Table 26-89.

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 26-96. 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 26-89. 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

26.7.71 MCANERR_ERR_CTRL1 (Offset = 7418h) [Reset = 00000000h]

MCANERR_ERR_CTRL1 is shown in Figure 26-97 and described in Table 26-90.

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 26-97. 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 26-90. 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.

26.7.72 MCANERR_ERR_CTRL2 (Offset = 741Ch) [Reset = 00000000h]

MCANERR_ERR_CTRL2 is shown in Figure 26-98 and described in Table 26-91.

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 26-98. 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 26-91. 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

26.7.73 MCANERR_ERR_STAT1 (Offset = 7420h) [Reset = 00000000h]

MCANERR_ERR_STAT1 is shown in Figure 26-99 and described in Table 26-92.

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 26-99. 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 26-92. 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. 0 Bit 0 is in error 1 Bit 1 is in error 2 Bit 2 is in error 3 Bit 3 is in error ... 31 Bit 31 is in error >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 0 No SEC error while writeback pending 1 Indicates that successive single-bit errors have occurred while a writeback is still pending

Table 26-92. 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. 0 No double-bit error detected 1 One double-bit error was detected 2 Two double-bit errors were detected 3 Three double-bit errors were detected 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. 0 No single-bit error detected 1 One single-bit error was detected and corrected 2 Two single-bit errors were detected and corrected 3 Three single-bit errors were detected and corrected A write of a non-zero value to this bit field increments it by the value provided.

26.7.74 MCANERR_ERR_STAT2 (Offset = 7424h) [Reset = 00000000h]

MCANERR_ERR_STAT2 is shown in Figure 26-100 and described in Table 26-93.

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 26-100. 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 26-93. 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.

26.7.75 MCANERR_ERR_STAT3 (Offset = 7428h) [Reset = 00000000h]

MCANERR_ERR_STAT3 is shown in Figure 26-101 and described in Table 26-94.

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 26-101. 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 26-94. 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 0 No write back pending 1 An ECC data correction write back is pending

26.7.76 MCANERR_SEC_EOI (Offset = 743Ch) [Reset = 00000000h]

MCANERR_SEC_EOI is shown in Figure 26-102 and described in Table 26-95.

Return to the Summary Table.

MCAN Single Error Corrected End of Interrupt Register

Figure 26-102. 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 26-95. 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.

26.7.77 MCANERR_SEC_STATUS (Offset = 7440h) [Reset = 00000000h]

MCANERR_SEC_STATUS is shown in Figure 26-103 and described in Table 26-96.

Return to the Summary Table.

MCAN Single Error Corrected Interrupt Status Register

Figure 26-103. 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 26-96. 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 0 No SEC interrupt is pending 1 SEC interrupt is pending

26.7.78 MCANERR_SEC_ENABLE_SET (Offset = 7480h) [Reset = 00000000h]

MCANERR_SEC_ENABLE_SET is shown in Figure 26-104 and described in Table 26-97.

Return to the Summary Table.

MCAN Single Error Corrected Interrupt Enable Set Register

Figure 26-104. 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 26-97. 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.

26.7.79 MCANERR_SEC_ENABLE_CLR (Offset = 74C0h) [Reset = 00000000h]

MCANERR_SEC_ENABLE_CLR is shown in Figure 26-105 and described in Table 26-98.

Return to the Summary Table.

MCAN Single Error Corrected Interrupt Enable Clear Register

Figure 26-105. 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 26-98. 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.

26.7.80 MCANERR_DED_EOI (Offset = 753Ch) [Reset = 00000000h]

MCANERR_DED_EOI is shown in Figure 26-106 and described in Table 26-99.

Return to the Summary Table.

MCAN Double Error Detected End of Interrupt Register

Figure 26-106. 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 26-99. 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.

26.7.81 MCANERR_DED_STATUS (Offset = 7540h) [Reset = 00000000h]

MCANERR_DED_STATUS is shown in Figure 26-107 and described in Table 26-100.

Return to the Summary Table.

MCAN Double Error Detected Interrupt Status Register

Figure 26-107. 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 26-100. 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 0 No DED interrupt is pending 1 DED interrupt is pending

26.7.82 MCANERR_DED_ENABLE_SET (Offset = 7580h) [Reset = 00000000h]

MCANERR_DED_ENABLE_SET is shown in Figure 26-108 and described in Table 26-101.

Return to the Summary Table.

MCAN Double Error Detected Interrupt Enable Set Register

Figure 26-108. 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 26-101. 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.

26.7.83 MCANERR_DED_ENABLE_CLR (Offset = 75C0h) [Reset = 00000000h]

MCANERR_DED_ENABLE_CLR is shown in Figure 26-109 and described in Table 26-102.

Return to the Summary Table.

MCAN Double Error Detected Interrupt Enable Clear Register

Figure 26-109. 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 26-102. 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.

26.7.84 MCANERR_AGGR_ENABLE_SET (Offset = 7600h) [Reset = 00000000h]

MCANERR_AGGR_ENABLE_SET is shown in Figure 26-110 and described in Table 26-103.

Return to the Summary Table.

MCAN Error Aggregator Enable Set Register

Figure 26-110. 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 26-103. 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.

26.7.85 MCANERR_AGGR_ENABLE_CLR (Offset = 7604h) [Reset = 00000000h]

MCANERR_AGGR_ENABLE_CLR is shown in Figure 26-111 and described in Table 26-104.

Return to the Summary Table.

MCAN Error Aggregator Enable Clear Register

Figure 26-111. 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 26-104. 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.

26.7.86 MCANERR_AGGR_STATUS_SET (Offset = 7608h) [Reset = 00000000h]

MCANERR_AGGR_STATUS_SET is shown in Figure 26-112 and described in Table 26-105.

Return to the Summary Table.

MCAN Error Aggregator Status Set Register

Figure 26-112. 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 26-105. 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 2-bit saturating counter of the number of SVBUS timeout errors that have occurred since last cleared. 0 No timeout errors have occurred 1 One timeout error has occurred 2 Two timeout errors have occurred 3 Three timeout errors have occurred 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 2-bit saturating counter of the number of parity errors that have occurred since last cleared. 0 No parity errors have occurred 1 One parity error has occurred 2 Two parity errors have occurred 3 Three parity errors have occurred A write of a non-zero value to this bit field increments it by the value provided.

26.7.87 MCANERR_AGGR_STATUS_CLR (Offset = 760Ch) [Reset = 00000000h]

MCANERR_AGGR_STATUS_CLR is shown in Figure 26-113 and described in Table 26-106.

Return to the Summary Table.

MCAN Error Aggregator Status Clear Register

Figure 26-113. 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 26-106. 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 2-bit saturating counter of the number of SVBUS timeout errors that have occurred since last cleared. 0 No timeout errors have occurred 1 One timeout error has occurred 2 Two timeout errors have occurred 3 Three timeout errors have occurred 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 2-bit saturating counter of the number of parity errors that have occurred since last cleared. 0 No parity errors have occurred 1 One parity error has occurred 2 Two parity errors have occurred 3 Three parity errors have occurred A write of a non-zero value to this bit field decrements it by the value provided.

26.7.88 IIDX (Offset = 7820h) [Reset = 00000000h]

IIDX is shown in Figure 26-114 and described in Table 26-107.

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 26-114. 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 26-107. IIDX Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	STAT	R	0h	Interrupt index status 00h = No interrupt pending. 1h = MCAN Interrupt Line 0 interrupt pending. 2h = MCAN Interrupt Line 1 interrupt pending. 3h = Message RAM SEC (Single Error Correction) interrupt pending. 4h = Message RAM DED (Double Error Detection) interrupt pending. 5h = External Timestamp Counter Overflow interrupt pending. 6h = Clock Stop Wake Up interrupt pending.

26.7.89 IMASK (Offset = 7828h) [Reset = 00000000h]

IMASK is shown in Figure 26-115 and described in Table 26-108.

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 26-115. 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 26-108. IMASK Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	
5	WAKEUP	R/W	0h	Clock Stop Wake Up interrupt mask. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
4	EXT_TS_CNTR_OVFL	R/W	0h	External Timestamp Counter Overflow interrupt mask. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
3	DED	R/W	0h	Massage RAM DED interrupt mask. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
2	SEC	R/W	0h	Message RAM SEC interrupt mask. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
1	INTL1	R/W	0h	MCAN Interrupt Line 1 mask. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
0	INTL0	R/W	0h	MCAN Interrupt Line 0 mask. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask

26.7.90 RIS (Offset = 7830h) [Reset = 00000000h]

RIS is shown in Figure 26-116 and described in Table 26-109.

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 26-116. 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 26-109. RIS Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	
5	WAKEUP	R	0h	Clock Stop Wake Up interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
4	EXT_TS_CNTR_OVFL	R	0h	External Timestamp Counter Overflow interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
3	DED	R	0h	Message RAM DED interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
2	SEC	R	0h	Message RAM SEC interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
1	INTL1	R	0h	MCAN Interrupt Line 1. 0h = Interrupt did not occur 1h = Interrupt occurred
0	INTL0	R	0h	MCAN Interrupt Line 0. 0h = Interrupt did not occur 1h = Interrupt occurred

26.7.91 MIS (Offset = 7838h) [Reset = 00000000h]

MIS is shown in Figure 26-117 and described in Table 26-110.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Figure 26-117. 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 26-110. MIS Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	
5	WAKEUP	R	0h	Masked Clock Stop Wake Up interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
4	EXT_TS_CNTR_OVFL	R	0h	Masked External Timestamp Counter Overflow interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
3	DED	R	0h	Masked Message RAM DED interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
2	SEC	R	0h	Masked Message RAM SEC interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
1	INTL1	R	0h	Masked MCAN Interrupt Line 1. 0h = Interrupt did not occur 1h = Interrupt occurred
0	INTL0	R	0h	Masked MCAN Interrupt Line 0. 0h = Interrupt did not occur 1h = Interrupt occurred

26.7.92 ISET (Offset = 7840h) [Reset = 00000000h]

ISET is shown in Figure 26-118 and described in Table 26-111.

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 26-118. 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 26-111. ISET Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	
5	WAKEUP	W	0h	Set Clock Stop Wake Up interrupt. 0h = Writing 0 has no effect 1h = Set Interrupt
4	EXT_TS_CNTR_OVFL	W	0h	Set External Timestamp Counter Overflow interrupt. 0h = Writing 0 has no effect 1h = Set Interrupt
3	DED	W	0h	Set Message RAM DED interrupt. 0h = Writing 0 has no effect 1h = Set Interrupt
2	SEC	W	0h	Set Message RAM SEC interrupt. 0h = Writing 0 has no effect 1h = Set Interrupt
1	INTL1	W	0h	Set MCAN Interrupt Line 1. 0h = Writing 0 has no effect 1h = Set Interrupt
0	INTL0	W	0h	Set MCAN Interrupt Line 0. 0h = Writing 0 has no effect 1h = Set Interrupt

26.7.93 ICLR (Offset = 7848h) [Reset = 00000000h]

ICLR is shown in Figure 26-119 and described in Table 26-112.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

Figure 26-119. 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 26-112. ICLR Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	
5	WAKEUP	W	0h	Clear Clock Stop Wake Up interrupt. 0h = Writing 0 has no effect 1h = Clear Interrupt
4	EXT_TS_CNTR_OVFL	W	0h	Clear External Timestamp Counter Overflow interrupt. 0h = Writing 0 has no effect 1h = Clear Interrupt
3	DED	W	0h	Clear Message RAM DED interrupt. 0h = Writing 0 has no effect 1h = Clear Interrupt
2	SEC	W	0h	Clear Message RAM SEC interrupt. 0h = Writing 0 has no effect 1h = Clear Interrupt
1	INTL1	W	0h	Clear MCAN Interrupt Line 1. 0h = Writing 0 has no effect 1h = Clear Interrupt
0	INTL0	W	0h	Clear MCAN Interrupt Line 0. 0h = Writing 0 has no effect 1h = Clear Interrupt

26.7.94 EVT_MODE (Offset = 78E0h) [Reset = 00000000h]

EVT_MODE is shown in Figure 26-120 and described in Table 26-113.

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 26-120. 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 26-113. 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] 0h = The interrupt or event line is disabled. 1h = The interrupt or event line is in software mode. Software must clear the RIS. 2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag.

26.7.95 DESC (Offset = 78FCh) [Reset = 00000000h]

DESC is shown in Figure 26-121 and described in Table 26-114.

Return to the Summary Table.

This register identifies the peripheral and its exact version.

Figure 26-121. 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 26-114. 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. 0h = Smallest value FFFFh = Highest possible value
15-12	FEATUREVER	R	0x0	Feature Set for the module *instance* 0h = MCAN module with CAN-FD mode enabled 1h = MCAN module with CAN-FD mode disabled
11-8	RESERVED	R	0h	
7-4	MAJREV	R	0x0	Major rev of the IP 0h = Smallest value Fh = Highest possible value
3-0	MINREV	R	0x0	Minor rev of the IP 0h = Smallest value Fh = Highest possible value

26.7.96 MCANSS_CLKEN (Offset = 7900h) [Reset = 00000000h]

MCANSS_CLKEN is shown in Figure 26-122 and described in Table 26-115.

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 26-122. 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 26-115. 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 0h = MCAN module functional clock and Vbusp is not requested. These clocks are gated to the MCAN module. 1h = Setting this bit requests MCAN module functional clock and Vbusp. These clocks are not gated to MCAN module.

26.7.97 MCANSS_CLKDIV (Offset = 7904h) [Reset = 00000000h]

MCANSS_CLKDIV is shown in Figure 26-123 and described in Table 26-116.

Return to the Summary Table.

Needs to go to the Management aperture once available

Figure 26-123. 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 26-116. 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. 0h (R/W) = Divides input clock by 1 1h (R/W) = Divides input clock by 2 2h (R/W) = Divides input clock by 4 3h (R/W) = Divides input clock by 1

26.7.98 MCANSS_CLKCTL (Offset = 7908h) [Reset = 00000000h]

MCANSS_CLKCTL is shown in Figure 26-124 and described in Table 26-117.

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 26-124. 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 26-117. 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. 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). 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) 0h = Disable MCAN IP clock stop wakeup interrupt 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. Note: This bit can be reset by HW by Clock-Stop Wake-up via CAN RX Activity. See spec for more details. 0h = Disable MCAN-SS clock stop request 1h = Enable MCAN-SS clock stop request

26.7.99 MCANSS_CLKSTS (Offset = 790Ch) [Reset = 00000000h]

MCANSS_CLKSTS is shown in Figure 26-125 and described in Table 26-118.

Return to the Summary Table.

MCANSS clock stop status register to indicate status of clock stop mechanism

Figure 26-125. 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 26-118. 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. 0h = MCAN controller clock is not available to the MCAN IP. 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. 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. 0h = MCANSS_CLKCTL.STOPREQ bit has not been cleared by HW. 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 0h = No clock stop acknowledged. 1h = Clock stop has been acknowledged by MCAN IP; MCAN-SS may be clock gated by stopping both the CAN host and functional clocks.

Chapter 27

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.

27.1 I2S/TDM Introduction 1623
27.2 I2S/TDM Operation 1623
27.3 I2S Registers 1637

27.1 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.

27.1.1 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

27.2 I2S/TDM Operation

27.2.1 Functional Block Diagram

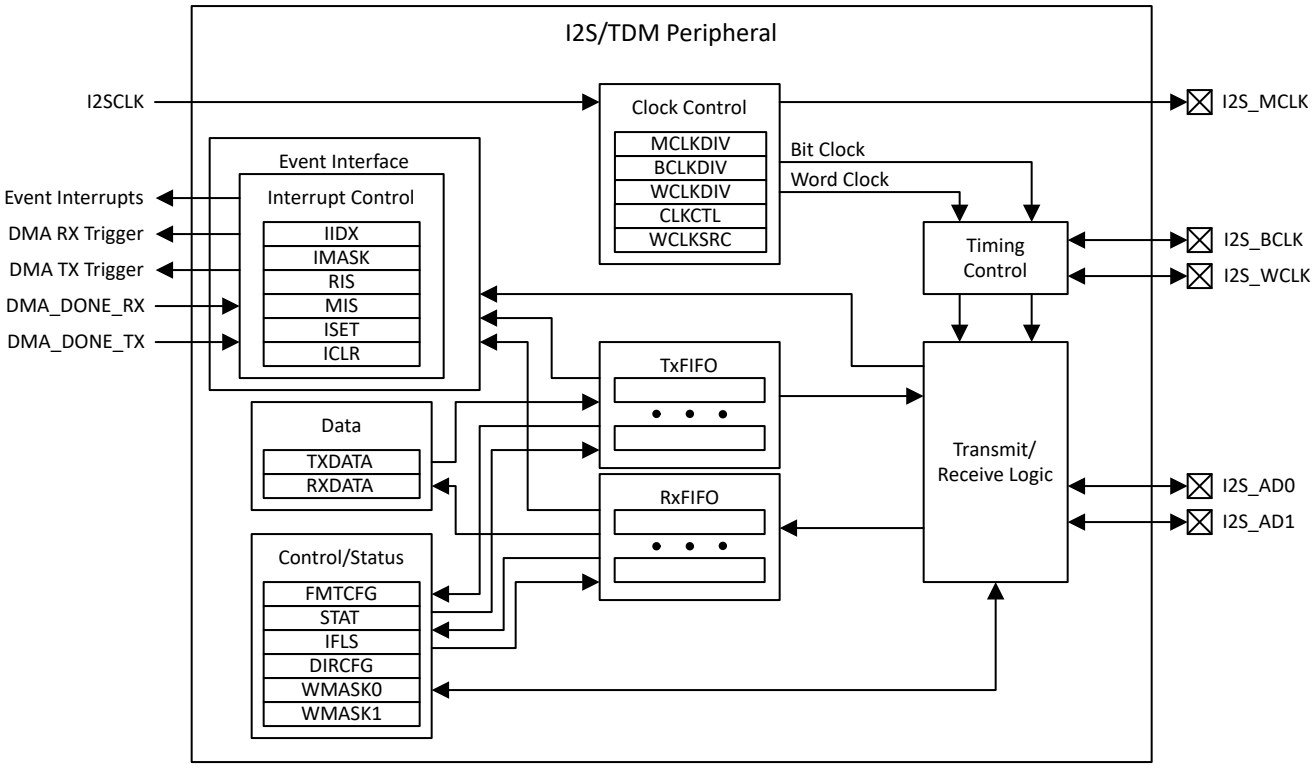


Figure 27-1. I2S/TDM Functional Block Diagram

Note

Refer to TxFIFO and RxFIFO depth in device specific data sheet.

27.2.2 Modes of Operation

The I2S module can be configured as a controller or target function.

27.2.2.1 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.

27.2.2.2 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.

27.2.3 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 22 and Equation 23 can be used to derive the clock frequency respectively.

$$BCLK\ Frequency = \frac{Module\ Clock\ Frequency}{BCLKDIV} \quad (22)$$

$$MCLK\ Frequency = \frac{Module\ Clock\ Frequency}{MCLKDIV} \quad (23)$$

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 24 defines the equation for calculating the WCLK frequency or frame rate.

$$WCLK\ Frequency\left(Frame\ Rate\right) = \frac{Module\ Clock\ Frequency}{(BCLKDIV \times (WCLKDIV[9:0] + 1))} \quad (24)$$

For frame format where the WCLK is high with a 50% duty cycle (dual-phase) the CLKCTL.WCLKPHASE is configured as 1, Equation 25 defines the equation for calculating the WCLK frequency or frame rate.

$$WCLK\ Frequency\left(Frame\ Rate\right) = \frac{Module\ Clock\ Frequency}{(BCLKDIV \times (2 \times WCLKDIV[9:0]))} \quad (25)$$

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 26 defines the equation for calculating the WCLK frequency or frame rate.

$$WCLK\ Frequency\left(Frame\ Rate\right) = \frac{Module\ Clock\ Frequency}{(BCLKDIV \times (WCLKDIV[7:0] + WCLKDIV[15:8]))} \quad (26)$$

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).

27.2.4 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 PCM, DSP and TDM, 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 PCM long frame, where I2S_WCLK is high for WCLKDIV.WDIV[7:0] and low for WCLKDIV.WDIV[15:8] I2S_BCLK

27.2.4.1 Frame and Word Length

The frame length is calculated as the number I2S_BCLK 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 27-2.

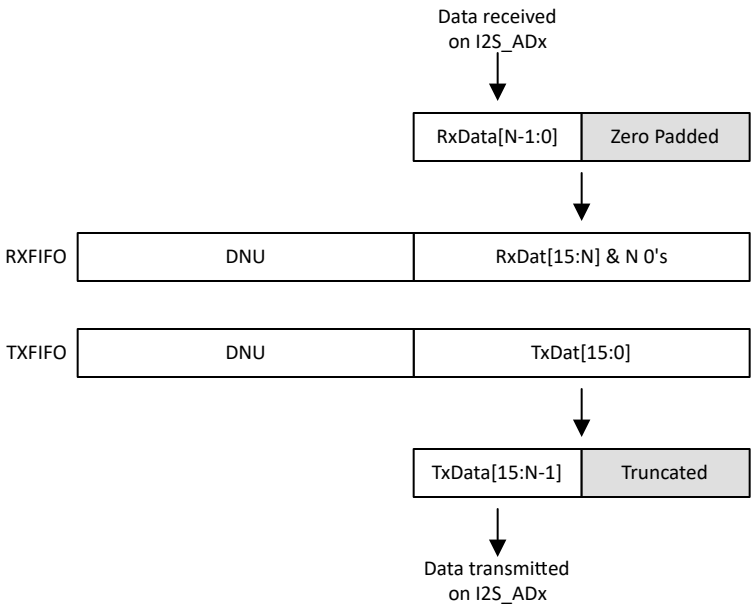


Figure 27-2. Data Mapping for MEMLEN32=0

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 27-3.

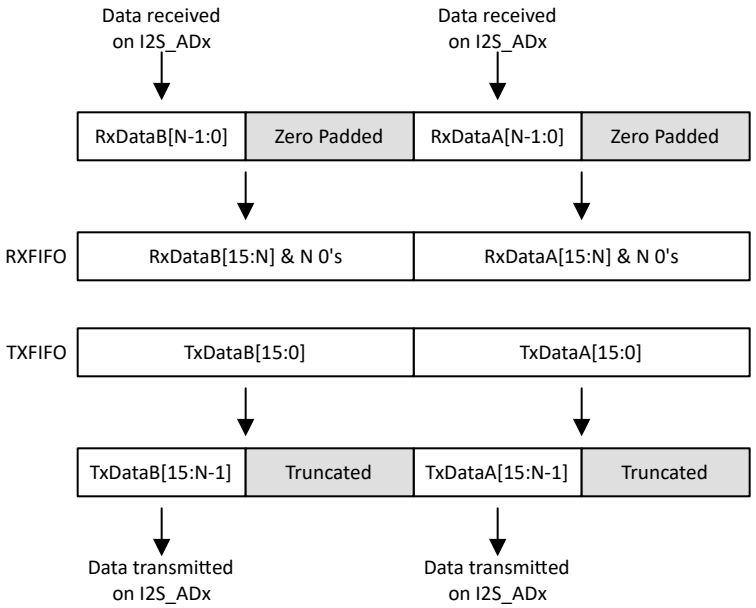


Figure 27-3. Data Mapping for MEMLEN32=1

27.2.4.2 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.

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.

27.2.4.3 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.

27.2.5 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.

27.2.5.1 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 27-4 to Figure 27-9 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.

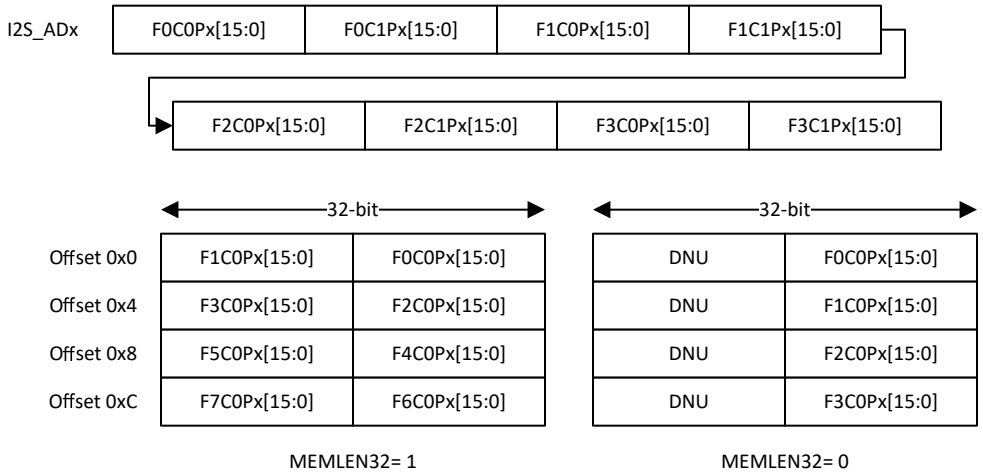


Figure 27-4. 16-bit Mono Single I2S_ADx to Memory Mapping

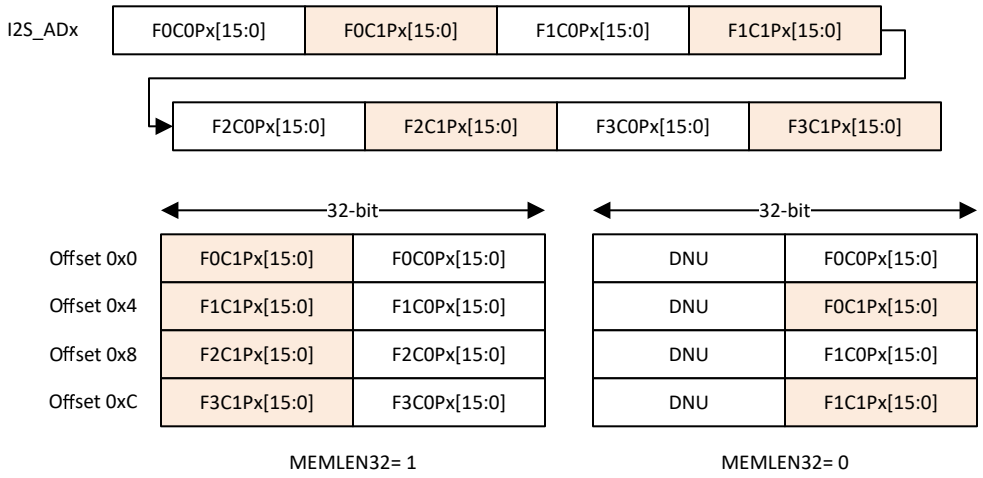
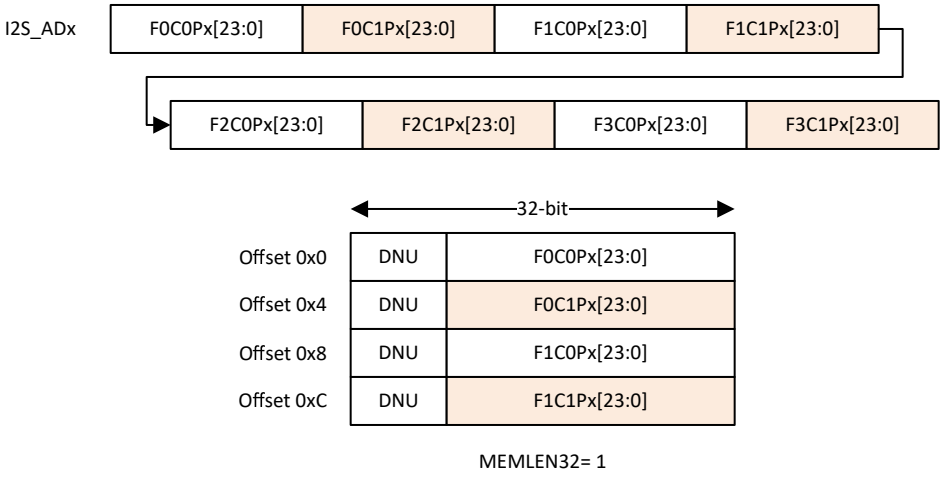


Figure 27-5. 16-bit Stereo Single I2S_ADx to Memory Mapping



Note

MEMLEN32=0 is not supported

Figure 27-6. 24-bit Stereo Single I2S_ADx to Memory Mapping

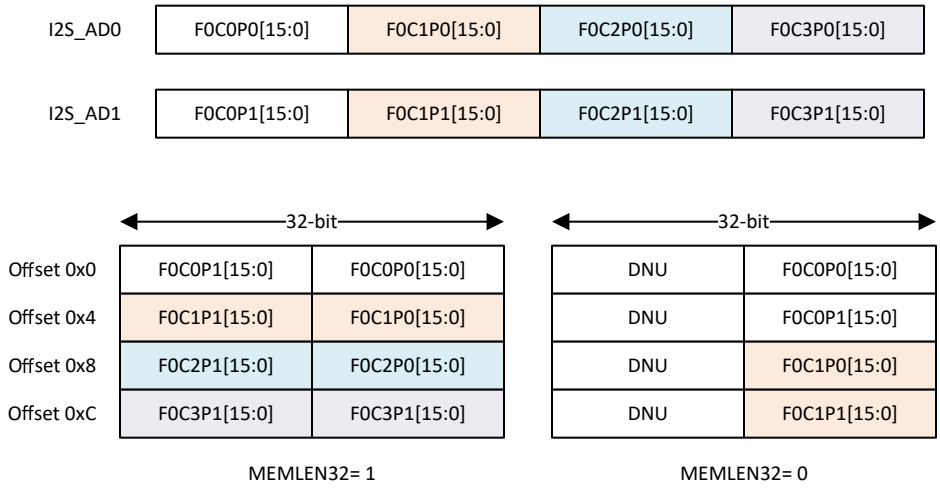


Figure 27-7. 16-bit TDM Dual I2S_ADx to Memory Mapping

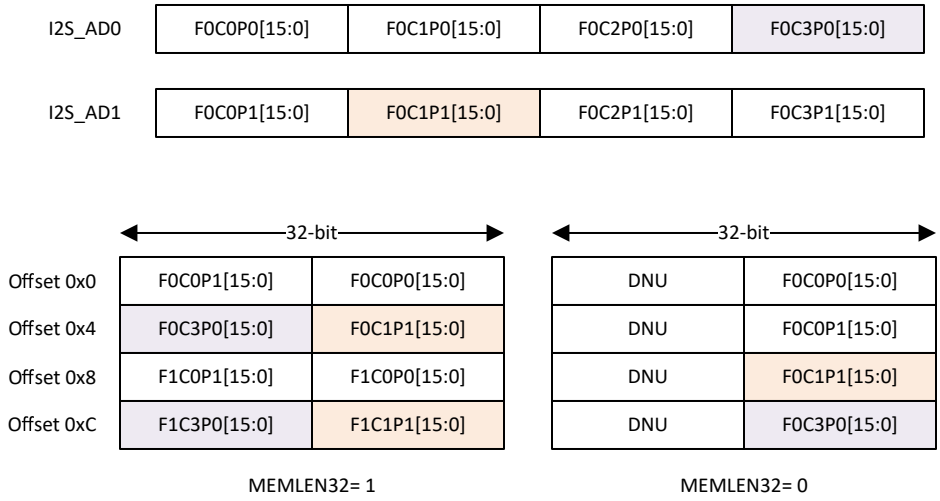


Figure 27-8. 16-bit TDM Dual I2S_ADx Slots Disabled to Memory Mapping

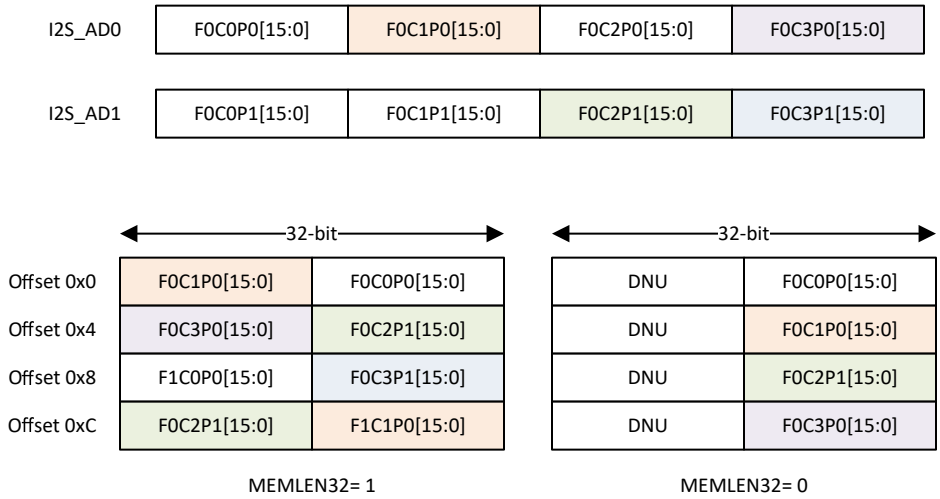


Figure 27-9. 16-bit TDM Dual I2S_ADx Slots Disabled to Memory Mapping

27.2.6 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 27-1. Subsequent sub-sections illustrate how the application can generate some of the most common audio formats. This is intended to be informational-only.

Table 27-1. 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	

27.2.6.1 I2S Format

I2S is a dual-phase format with a 50% I2S_WCLK duty cycle with two channels as shown in Figure 27-10 with the configuration example shown in Table 27-2. 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.

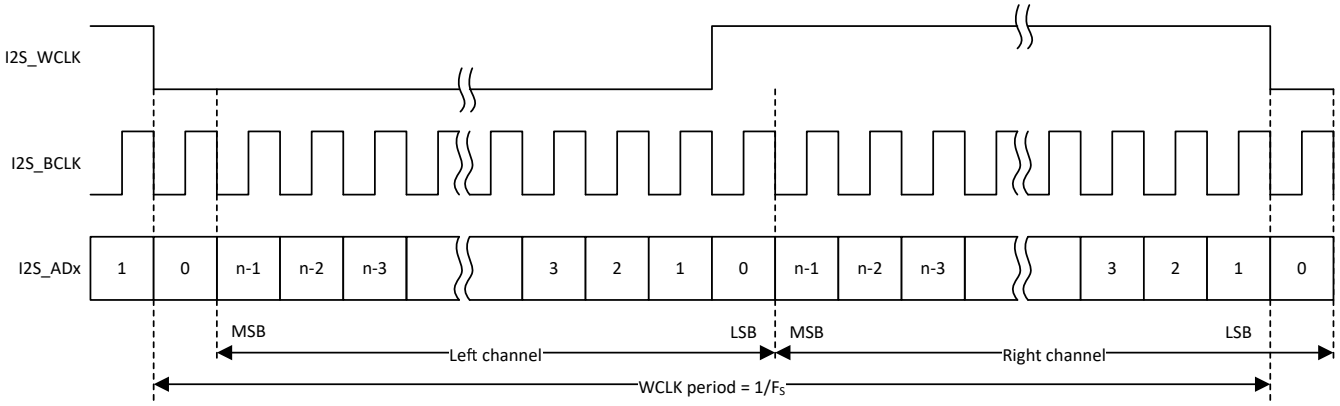


Figure 27-10. I2S Frame Format

Table 27-2. I2S Format Configuration

PARAMETER FIELD	VALUE
FMTCFG.DATADLY	1
FMTCFG.WORDLEN	Number of bits per sample in the left or right channel

Table 27-2. 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

27.2.6.2 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 27-11 with the configuration example shown in Table 27-3. 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.

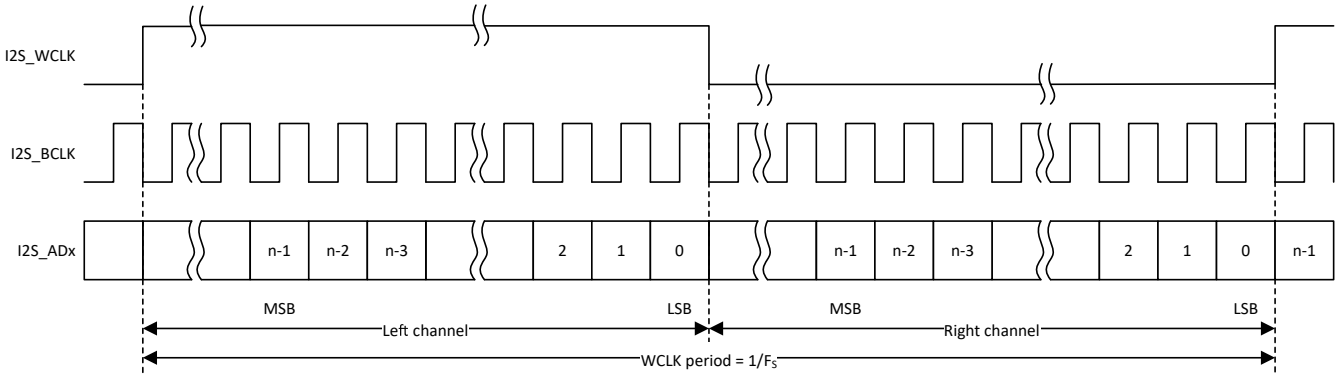


Figure 27-11. Right Justified Format

Table 27-3. 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

27.2.6.3 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 27-12 with the configuration examle shown in Table 27-4. The left channel is transferred when I2S_WCLK phase is high followed by the right channel when I2S_WCLK phase is low.

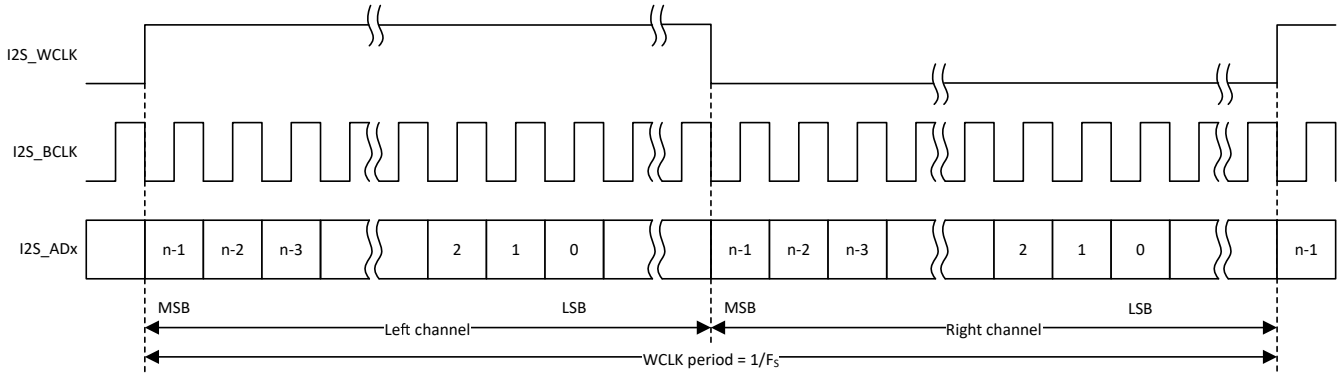


Figure 27-12. Left Justified Format

Table 27-4. 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

27.2.6.4 DSP Format

The DSP format is a single-phase format with I2S_WCLK high for one I2S_BCLK period as shown in Figure 27-13. 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 27-5.

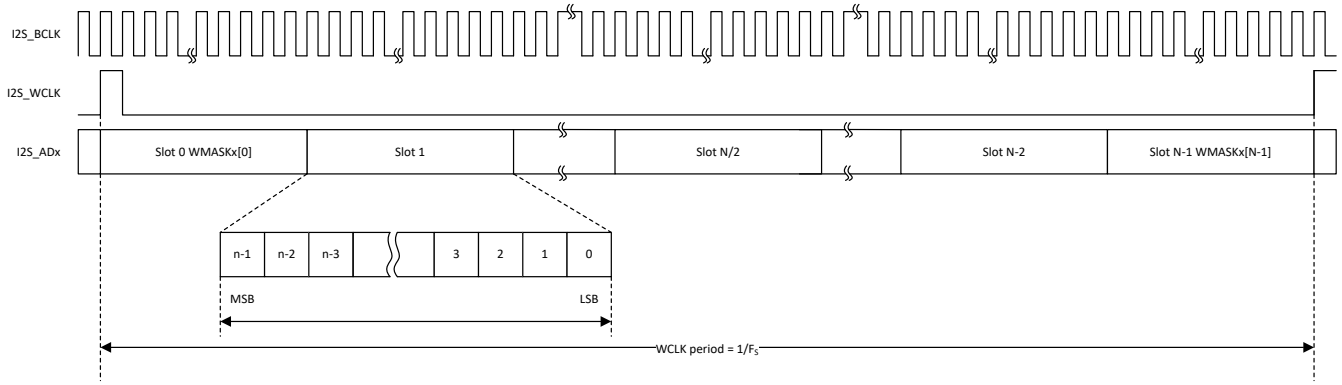


Figure 27-13. DSP Format

Table 27-5. 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

Table 27-5. 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

27.2.6.5 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 27-14. 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 27-6.

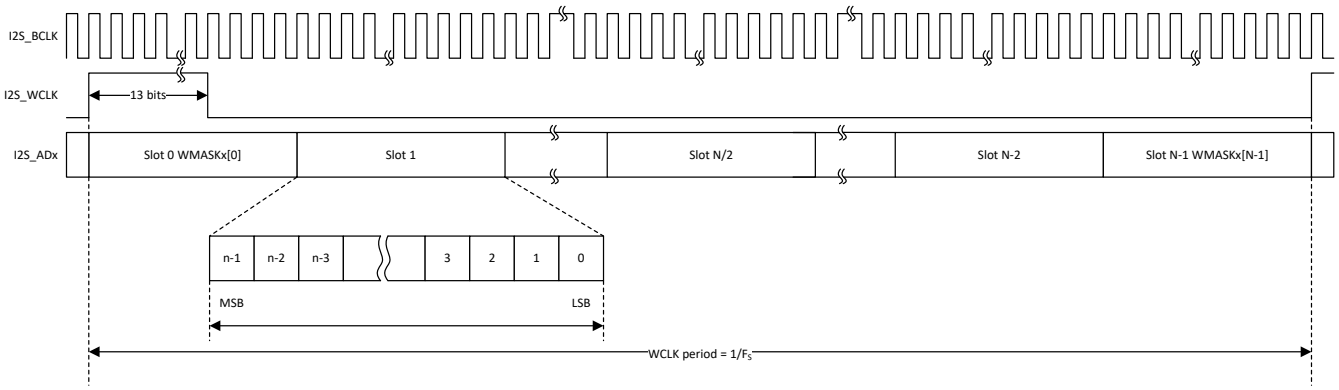


Figure 27-14. PCM Long Frame Format

Table 27-6. 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
WCLKDIV.WDIV[7:0]	13 - (Number of bits per samples x Number of slots)

27.2.6.6 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 27-15, 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 27-7.

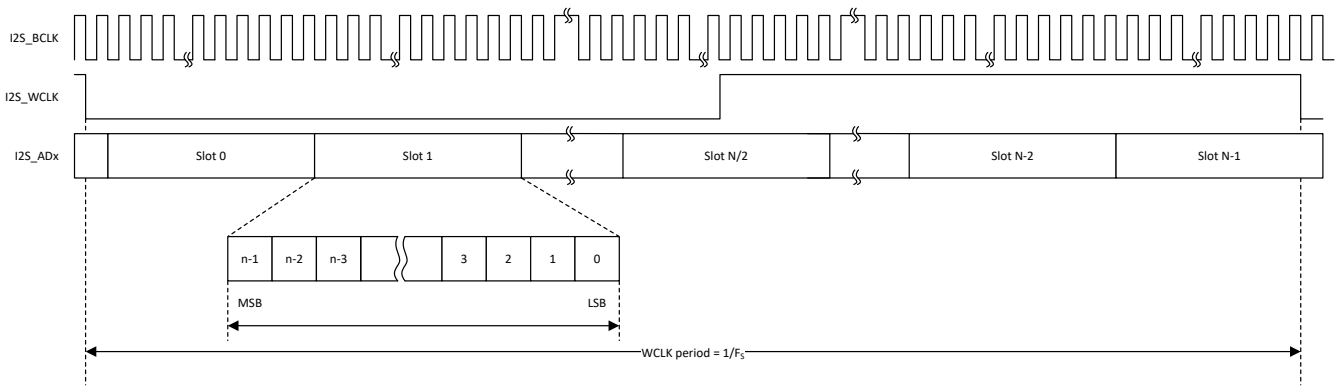


Figure 27-15. TDM Classic Format

Table 27-7. 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

27.2.7 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 27.2.6.
4. Setup the configuration for the channel configuration. Example channel configurations can be found in Section 27.2.5.
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 27.2.6.

3. Setup the configuration for the channel configuration. Example channel configurations can be found in Section 27.2.5.
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'.

27.2.8 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.

27.2.9 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 27-8.

Table 27-8. 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

27.2.9.1 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 27-9. 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.

Table 27-9. 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 8.2.5 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.

27.2.9.2 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 8.2.5 for guidance on configuring the Event registers and Section 8.1.3.2 for on how the DMA trigger event works.

27.2.10 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 27-10. 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

27.3 I2S Registers

Table 27-11 lists the memory-mapped registers for the I2S registers. All register offset addresses not listed in Table 27-11 should be considered as reserved locations and the register contents should not be modified.

Table 27-11. 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. This register can be used by software for diagnostics and safety checking purposes.		Go
1048h	ICLR	Interrupt clear register. 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

Table 27-11. 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 27-12 shows the codes that are used for access types in this section.

Table 27-12. I2S Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
WK	W K	Write Write protected by a key
Reset or Default Value		
-n		Value after reset or the default value

27.3.1 PWREN (Offset = 800h) [Reset = 00000000h]

PWREN is shown in Figure 27-16 and described in Table 27-13.

Return to the Summary Table.

Register to control the power state

Figure 27-16. 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 27-13. PWREN Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to allow Power State Change 26h = KEY to allow write access to this register
23-1	RESERVED	R	0h	
0	ENABLE	R/WK	0h	Enable the power KEY must be set to 26h to write to this bit. 0h = Disable Power 1h = Enable Power

27.3.2 RSTCTL (Offset = 804h) [Reset = 00000000h]

RSTCTL is shown in Figure 27-17 and described in Table 27-14.

Return to the Summary Table.

Register to control reset assertion and de-assertion

Figure 27-17. 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 27-14. RSTCTL Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	Unlock key 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 KEY must be set to B1h to write to this bit. 0h = Writing 0 has no effect 1h = Clear reset sticky bit
0	RESETASSERT	WK	0h	Assert reset to the peripheral KEY must be set to B1h to write to this bit. 0h = Writing 0 has no effect 1h = Assert reset

27.3.3 CLKCFG (Offset = 808h) [Reset = 00000000h]

CLKCFG is shown in Figure 27-18 and described in Table 27-15.

Return to the Summary Table.

Peripheral Clock Configuration Register

Figure 27-18. 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 27-15. CLKCFG Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow State Change -- 0xA9 A9h = key value to allow change field of GPRCM
23-2	RESERVED	R	0h	
1-0	DAICLK	R/W	0h	Audio Clock 0h = SYSOSC 1h = HF crystal 2h = PLL 3h = Reserved

27.3.4 STAT (Offset = 814h) [Reset = 00000000h]

STAT is shown in Figure 27-19 and described in Table 27-16.

Return to the Summary Table.

peripheral enable and reset status

Figure 27-19. 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 27-16. 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 0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register 1h = The peripheral was reset since the last bit clear
15-0	RESERVED	R	0h	

27.3.5 PDBGCTL (Offset = 1008h) [Reset = 00000003h]

PDBGCTL is shown in Figure 27-20 and described in Table 27-17.

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 27-20. 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 27-17. 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' 0h = The peripheral will halt immediately, even if the resultant state will result in corruption if the system is restarted 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 0h = The peripheral freezes functionality while the Core Halted input is asserted and resumes when it is de-asserted. 1h = The peripheral ignores the state of the Core Halted input

27.3.6 IIDX (Offset = 1020h) [Reset = 00000000h]

IIDX is shown in Figure 27-21 and described in Table 27-18.

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 27-21. 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 27-18. 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 00h = No interrupt pending 01h = WCLK Error 2h = Receive interrupt; Interrupt Flag: RX 3h = Transmit interrupt; Interrupt Flag: TX 4h = RX FIFO Overflow Event/interrupt pending 5h = TX FIFO underflow interrupt 10h = DMA DONE on RX 11h = DMA DONE on TX 13h = DMA PRE IRQ INTERRUPT 14h = DMA PRE IRQ INTERRUPT

27.3.7 IMASK (Offset = 1028h) [Reset = 00000000h]

IMASK is shown in Figure 27-22 and described in Table 27-19.

Return to the Summary Table.

Interrupt Mask Register Selects mask states of the flags in [IRQFLAGS.*] that contribute to the I2S_IRQ event.

Figure 27-22. 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 27-19. 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 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
15	DMA_DONE_RX	R/W	0h	Enable DMA Done on RX Event Channel Interrupt 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
14-5	RESERVED	R	0h	
4	TXFIFO_UNF	R/W	0h	TX FIFO underflow interrupt mask 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
3	RXFIFO_OVF	R/W	0h	RXFIFO overflow event mask. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
2	TXINT	R/W	0h	Enable Transmit Interrupt. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
1	RXINT	R/W	0h	Enable Receive Interrupt. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
0	WCLKERR	R/W	0h	WCLKERR interrupt mask 0h = Disable the interrupt mask 1h = Enable the interrupt mask

27.3.8 RIS (Offset = 1030h) [Reset = 00000000h]

RIS is shown in Figure 27-23 and described in Table 27-20.

Return to the Summary Table.

This registers gives the raw interrupt status

Figure 27-23. 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 27-20. 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 0h = Interrupt did not occur 1h = Interrupt occurred
15	DMA_DONE_RX	R	0h	DMA Done on RX Event Channel Interrupt 0h = Interrupt did not occur 1h = Interrupt occurred
14-5	RESERVED	R	0h	
4	TXFIFO_UNF	R	0h	TX FIFO Underflow Interrupt 0h = Interrupt did not occur 1h = Interrupt occurred
3	RXFIFO_OVF	R	0h	RXFIFO overflow event. This interrupt is set if an RX FIFO overflow has been detected. 0h = Interrupt did not occur 1h = Interrupt occurred
2	TXINT	R	0h	Transmit Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
1	RXINT	R	0h	Receive Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred

Table 27-20. 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. 0h = Interrupt did not occur 1h = Interrupt occurred

27.3.9 MIS (Offset = 1038h) [Reset = 00000000h]

MIS is shown in Figure 27-24 and described in Table 27-21.

Return to the Summary Table.

This registers gives the raw interrupt status

Figure 27-24. 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 27-21. 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 0h = Interrupt did not occur or mask was disabled. 1h = Interrupt occurred
15	DMA_DONE_RX	R	0h	Masked DMA Done on RX Event Channel Interrupt 0h = Interrupt did not occur or mask was disabled. 1h = Interrupt occurred
14-5	RESERVED	R	0h	
4	TXFIFO_UNF	R	0h	TX FIFO underflow interrupt 0h = Interrupt did not occur or mask was disabled. 1h = Interrupt occurred
3	RXFIFO_OVF	R	0h	Masked RXFIFO overflow event. This interrupt is set if an RX FIFO overflow has been detected. 0h = Interrupt did not occur or mask was disabled. 1h = Interrupt occurred
2	TXINT	R	0h	Masked Transmit Interrupt. 0h = Interrupt did not occur or mask was disabled. 1h = Interrupt occurred
1	RXINT	R	0h	Masked Receive Interrupt. 0h = Interrupt did not occur or mask was disabled. 1h = Interrupt occurred

Table 27-21. 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. 0h = Interrupt did not occur or mask was disabled. 1h = Interrupt occurred

27.3.10 ISET (Offset = 1040h) [Reset = 00000000h]

ISET is shown in Figure 27-25 and described in Table 27-22.

Return to the Summary Table.

Interrupt Set Register. This register can be used by software for diagnostics and safety checking purposes.

Figure 27-25. 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 27-22. 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 0h = Writing 0 has no effect 1h = Set Interrupt
15	DMA_DONE_RX	W	0h	Set DMA Done on RX Event Channel Interrupt 0h = Writing 0 has no effect 1h = Set Interrupt
14-5	RESERVED	R	0h	
4	TXFIFO_UNF	W	0h	Set TX FIFO Underflow Event 0h = Writing 0 has no effect 1h = Set interrupt
3	RXFIFO_OVF	W	0h	Set RXFIFO overflow event. 0h = Writing 0 has no effect 1h = Set Interrupt
2	TXINT	W	0h	Set Transmit Interrupt. 0h = Writing 0 has no effect 1h = Set Interrupt
1	RXINT	W	0h	Set Receive Interrupt. 0h = Writing 0 has no effect 1h = Set Interrupt
0	WCLKERR	W	0h	This field sets the interrupt WCLKERR 0h = Writing 0 has no effect 1h = Set interrupt

27.3.11 ICLR (Offset = 1048h) [Reset = 00000000h]

ICLR is shown in Figure 27-26 and described in Table 27-23.

Return to the Summary Table.

Interrupt clear register. This register allows software to clear interrupts.

Figure 27-26. 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 27-23. 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 0h = Writing 0 has no effect 1h = Clear Interrupt
15	DMA_DONE_RX	W	0h	Clear DMA Done on RX Event Channel Interrupt 0h = Writing 0 has no effect 1h = Clear Interrupt
14-5	RESERVED	R	0h	
4	TXFIFO_UNF	W	0h	Clear TXFIFO underflow event 0h = Writing 0 has no effect 1h = Clear interrupt
3	RXFIFO_OVF	W	0h	Clear RXFIFO overflow event. 0h = Writing 0 has no effect 1h = Clear Interrupt
2	TXINT	W	0h	Clear Transmit Interrupt. 0h = Writing 0 has no effect 1h = Clear Interrupt
1	RXINT	W	0h	Clear Receive Interrupt. 0h = Writing 0 has no effect 1h = Clear Interrupt
0	WCLKERR	W	0h	This field clears the interrupt WCLKERR 0h = Writing 0 has no effect 1h = Clear interrupt

27.3.12 IMASK (Offset = 1058h) [Reset = 00000000h]

IMASK is shown in Figure 27-27 and described in Table 27-24.

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 27-27. 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 27-24. IMASK Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1	RXINT	R/W	0h	Enable Receive Interrupt. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
0	RESERVED	R	0h	

27.3.13 RIS (Offset = 1060h) [Reset = 00000000h]

RIS is shown in Figure 27-28 and described in Table 27-25.

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 27-28. 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 27-25. RIS Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1	RXINT	R	0h	Receive Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
0	RESERVED	R	0h	

27.3.14 MIS (Offset = 1068h) [Reset = 00000000h]

MIS is shown in Figure 27-29 and described in Table 27-26.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Figure 27-29. 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 27-26. MIS Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1	RXINT	R	0h	Masked Receive Interrupt. 0h = Interrupt did not occur or mask was disabled. 1h = Interrupt occurred
0	RESERVED	R	0h	

27.3.15 ISET (Offset = 1070h) [Reset = 00000000h]

ISET is shown in Figure 27-30 and described in Table 27-27.

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 27-30. 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 27-27. ISET Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1	RXINT	W	0h	Set Receive Interrupt. 0h = Writing 0 has no effect 1h = Set Interrupt
0	RESERVED	R	0h	

27.3.16 IMASK (Offset = 1088h) [Reset = 00000000h]

IMASK is shown in Figure 27-31 and described in Table 27-28.

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 27-31. 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 27-28. IMASK Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	
2	TXINT	R/W	0h	Enable Transmit Interrupt. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
1-0	RESERVED	R	0h	

27.3.17 RIS (Offset = 1090h) [Reset = 00000000h]

RIS is shown in Figure 27-32 and described in Table 27-29.

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 27-32. 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 27-29. RIS Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	
2	TXINT	R	0h	Transmit Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
1-0	RESERVED	R	0h	

27.3.18 MIS (Offset = 1098h) [Reset = 00000000h]

MIS is shown in Figure 27-33 and described in Table 27-30.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Figure 27-33. 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 27-30. MIS Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	
2	TXINT	R	0h	Masked Transmit Interrupt. 0h = Interrupt did not occur or mask was disabled. 1h = Interrupt occurred
1-0	RESERVED	R	0h	

27.3.19 ISET (Offset = 10A0h) [Reset = 00000000h]

ISET is shown in Figure 27-34 and described in Table 27-31.

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 27-34. 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 27-31. ISET Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	
2	TXINT	W	0h	Set Transmit Interrupt. 0h = Writing 0 has no effect 1h = Set Interrupt
1-0	RESERVED	R	0h	

27.3.20 INTCTL (Offset = 10E4h) [Reset = 00000000h]

INTCTL is shown in Figure 27-35 and described in Table 27-32.

Return to the Summary Table.

Interrupt control register

Figure 27-35. 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 27-32. 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. 0h = The interrupt or event line is disabled. 1h = The interrupt or event line is in software mode. Software must clear the RIS.

27.3.21 FMTCFG (Offset = 1100h) [Reset = 00000170h]

FMTCFG is shown in Figure 27-36 and described in Table 27-33.

Return to the Summary Table.

This register configures the serial interface format

Figure 27-36. 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 27-33. 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. 0h = Disable Module 1h = Enable module
30-25	RESERVED	R	0h	
24	SUSPEND	R/W	0h	Suspend external communication 0h = Functional mode resumed 1h = External communication suspended
23-18	RESERVED	R	0h	
17-16	EMPTYSLOTOUTPUT	R/W	0h	The field configures the EMPTYSLOTOUTPUT data pin direction 0h = Send out zeroes in empty slots 1h = send out ones in empty slot 2h = Tristate line during empty slot 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. 0h = Zero BCLK periods - LJF and DSP formats 1h = One BCLK periods - I2S and DSP formats 2h = Two(Min) BCLK periods - RJF format 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 0h = 16BIT : 16-bit (one 16 bit access per sample) 1h = 32BIT : 32-bit(one 32-bit access per sample)

Table 27-33. 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 0h = Data is sampled on the negative edge and clocked out on the positive edge. 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 0h = Single-phase: DSP format 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.

27.3.22 CLKCTL (Offset = 1104h) [Reset = 00000000h]

CLKCTL is shown in Figure 27-37 and described in Table 27-34.

Return to the Summary Table.

This register controls internal audio clock

Figure 27-37. 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 27-34. CLKCTL Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3	MEN	R/W	0h	This field configures the MCLK generation 0h = Disable the generation 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 0h = Disables the generation 1h = Enable the generation

27.3.23 STAT (Offset = 1108h) [Reset = 00000000h]

STAT is shown in Figure 27-38 and described in Table 27-35.

Return to the Summary Table.

Status Register

Figure 27-38. 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 27-35. STAT Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7	TXCLR	R	0h	TX FIFO Clear Status 0h = FIFO is not cleared 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. 0h = The transmitter is not full. 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. 0h = The transmitter has data to transmit. 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 0h = FIFO is not cleared 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. 0h = The receiver can receive data. 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. 0h = The receiver is not empty. 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	

27.3.24 IFLS (Offset = 110Ch) [Reset = 00000022h]

IFLS is shown in Figure 27-39 and described in Table 27-36.

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 27-39. 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 27-36. 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. 0h = Disable FIFO clear 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. 1h = RX FIFO >= 1/4 full 2h = RX FIFO >= 1/2 full (default) 3h = RX FIFO >= 3/4 full 5h = RX FIFO is full 6h = RX_FIFO >= (MAX_FIFO_LEN -1) 7h = RX_FIFO <= 1
3	TXCLR	R/W	0h	TX FIFO CLEAR. Setting this bit will clear the TX FIFO contents. 0h = Disable FIFO clear 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. 1h = TX FIFO <= 3/4 empty 2h = TX FIFO <= 1/2 empty (default) 3h = TX FIFO <= 1/4 empty 5h = TX FIFO is empty 6h = TX FIFO <= 1 7h = TX_FIFO >= (MAX_FIFO_LEN -1)

27.3.25 WCLKSRC (Offset = 1110h) [Reset = X0000000h]

WCLKSRC is shown in Figure 27-40 and described in Table 27-37.

Return to the Summary Table.

This register configures the WCLK Source

Figure 27-40. 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 27-37. 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). 0h = Source is not inverted 1h = Source is inverted
1-0	WBCLKSRC	R/W	Xh	This field selects WCLK/BCLK source for I2S. 0h = None ('0') 1h = External WCLK generator, from pad 2h = Internal WCLK generator, from module PRCM 3h = Not supported. Will give same WCLK as 'NONE' ('00')

27.3.26 DIRCFG (Offset = 1118h) [Reset = 00000000h]

DIRCFG is shown in Figure 27-41 and described in Table 27-38.

Return to the Summary Table.

This register configures the direction of data pins(AD0/AD1)

Figure 27-41. 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 27-38. 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 0h = Not in use (disabled) 1h = Input 2h = Output 3h = Reserved
3-2	RESERVED	R	0h	
1-0	AD0	R/W	0h	The field configures the AD0 data pin direction 0h = Not in use (disabled) 1h = Input 2h = Output 3h = Reserved

27.3.27 TXDATA (Offset = 1120h) [Reset = 00000000h]

TXDATA is shown in Figure 27-42 and described in Table 27-39.

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 27-42. 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 27-39. 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 . 0h = Smallest value FFFFFFFFh = Highest possible value

27.3.28 RXDATA (Offset = 1124h) [Reset = 00000000h]

RXDATA is shown in Figure 27-43 and described in Table 27-40.

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 27-43. 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 27-40. 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 . 0h = Smallest value FFFFFFFFh = Highest possible value

27.3.29 WMASK0 (Offset = 1148h) [Reset = 00000000h]

WMASK0 is shown in Figure 27-44 and described in Table 27-41.

Return to the Summary Table.

This register configures the word selection mask for data pin 0(AD0)

Figure 27-44. 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 27-41. 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.

27.3.30 WMASK1 (Offset = 114Ch) [Reset = 00000000h]

WMASK1 is shown in Figure 27-45 and described in Table 27-42.

Return to the Summary Table.

This register configures the word selection mask for data pin 1(AD1)

Figure 27-45. 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 27-42. 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.

27.3.31 MCLKDIV (Offset = 1160h) [Reset = 00000000h]

MCLKDIV is shown in Figure 27-46 and described in Table 27-43.

Return to the Summary Table.

This field configures MCLK division ratio

Figure 27-46. 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 27-43. 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.

27.3.32 WCLKDIV (Offset = 1164h) [Reset = 00000000h]

WCLKDIV is shown in Figure 27-47 and described in Table 27-44.

Return to the Summary Table.

This field configures WCLK division ratio

Figure 27-47. 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 27-44. 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]

27.3.33 BCLKDIV (Offset = 1168h) [Reset = 00000000h]

BCLKDIV is shown in Figure 27-48 and described in Table 27-45.

Return to the Summary Table.

This field configures BCLK division ratio

Figure 27-48. 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 27-45. 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.

Chapter 28

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.

28.1 TIMx Overview 1676
28.2 TIMx Operation 1679
28.3 TIMx Registers 1727

28.1 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 28.1.3 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.

28.1.1 TIMG Overview

The TIMG module consists of 16-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.

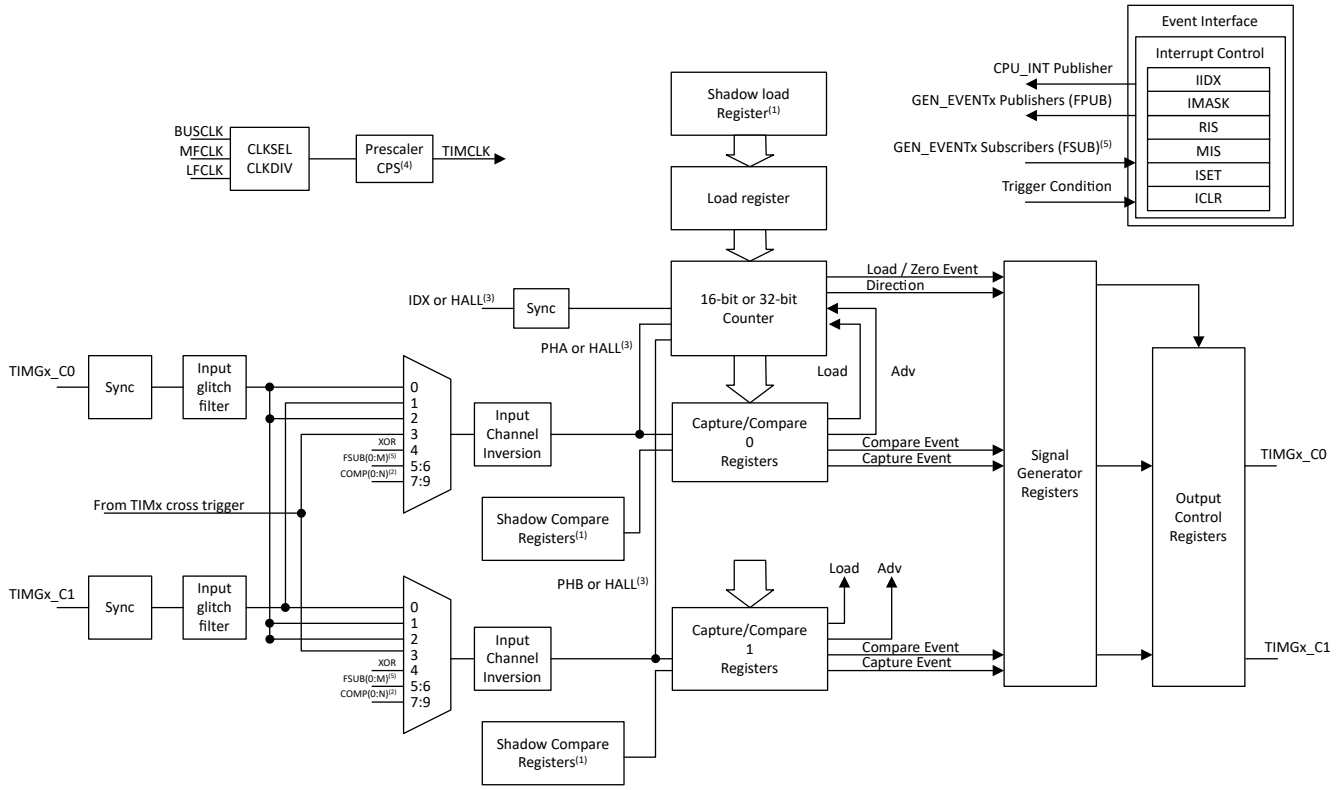
28.1.1.1 TIMG Features

Specific features for TIMG include:

- 16-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 28.2.4)
- Support for quadrature encoder interface (QEI) (see Section 28.2.3.1.3)
- 3-input Hall sensor mode for position sensing and speed computation (see Section 28.2.3.1.4)
- Support synchronization and cross trigger among different TIMx instances in the same power domain (see Section 28.2.7)
- Support CPU interrupt generation and cross peripherals (such as ADC, DAC, etc.) using the Event (see Section 28.2.9)

28.1.1.2 Functional Block Diagram

Figure 28-1 shows the TIMG block diagram.



(1) TIMG8_x and TIMG12_x do not have shadow load and compare registers
(2) Devices with comparator only
(3) TIMG8_x support QEI and Hall input mode
(4) Not supported on TIMG12_x
(5) Generic event (GEN_EVENTx) subscribers can be used to trigger any TIMx instance from any generic event publisher (GPIO, COMP, ADC, etc.)

Figure 28-1. TIMG Functional Block Diagram

28.1.2 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.

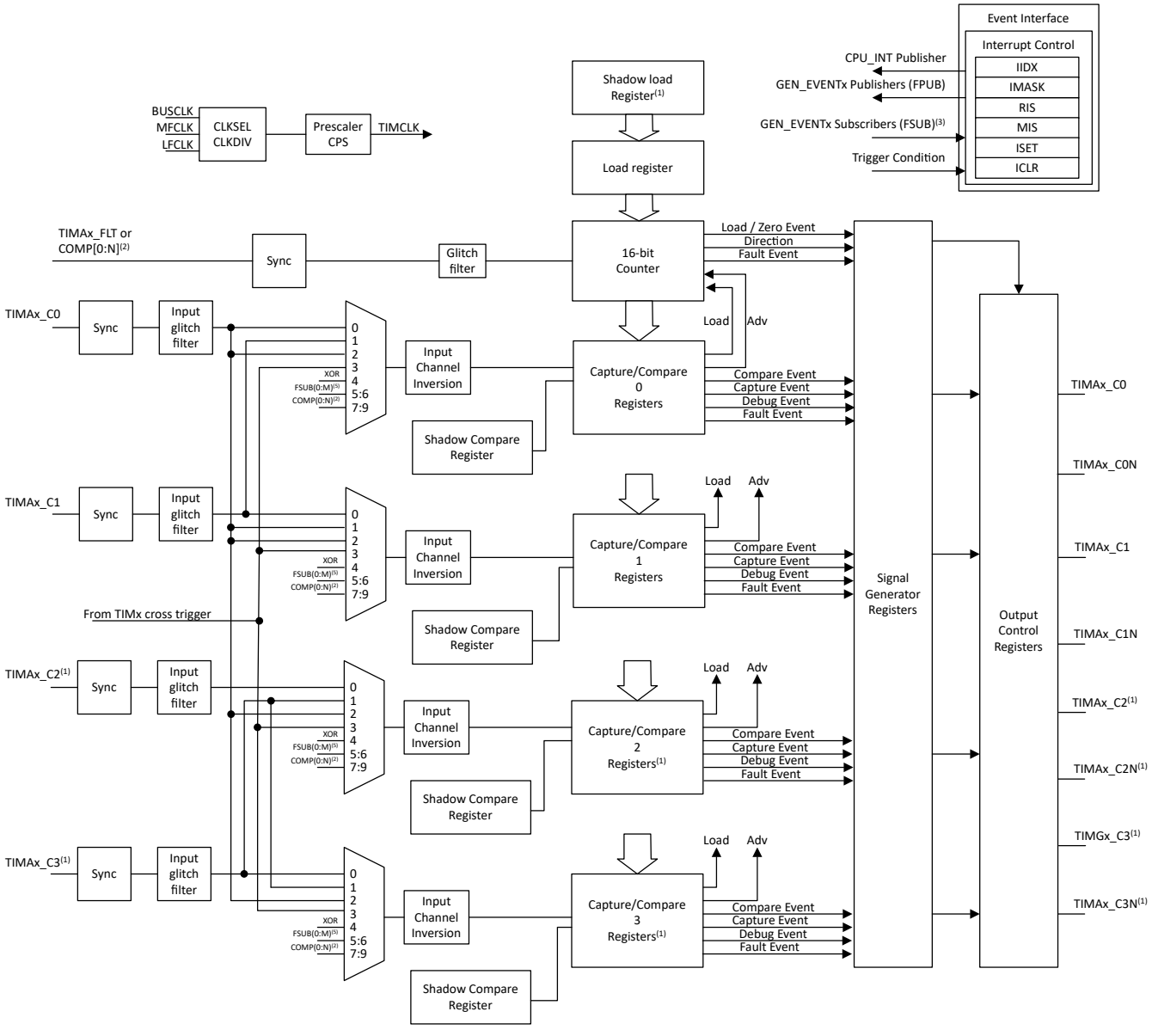
28.1.2.1 TIMA Features

- 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 28.2.1.2)
- 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 28.2.4)
- Complementary PWM output with programmable deadband insertion (see Section 28.2.5.2.4)
- Asymmetric PWM output (see Section 28.2.2.5)

- Fault handling mechanisms to ensure the output signals are in a safe user-defined state when a fault condition is encountered (see Section 28.2.6)
- Support synchronization and cross trigger among different TIMx instances in the same power domain (see Section 28.2.7)
- Support CPU interrupt generation and cross peripherals (such as ADC, DAC, etc.) using the Event (see Section 28.2.9)

28.1.2.2 Functional Block Diagram

Figure 28-2 shows the TIMA block diagram.



(1) TIMA0_x 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.)

Figure 28-2. TIMA Block Diagram

28.1.3 TIMx Instance Configuration

Table 28-1 shows the TIMx instance configuration for TIMA and TIMG instances.

Table 28-1. 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
TIMA0_x	16-bit	8-bit	Yes	4 + 2	8	Yes	Yes	Yes	Yes	Yes	-
TIMG4_x	16-bit	8-bit	-	2	2	-	Yes	Yes	-	-	-
TIMG8_x	16-bit	8-bit	-	2	2	-	-	-	-	-	Yes
TIMG12_x	32-bit	-	-	2	2	-	-	Yes	-	-	-

Note

On TIMA0_x 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

28.2 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.

Note

TIMx supports event subscribers and event publishers through event register arrays. For more information, please see the Events chapter.

28.2.1 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.

28.2.1.1 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.

28.2.1.1.1 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.
- 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.

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 27.

$$f_{TIMCLK} = \frac{f_{CLK\_SOURCE}}{((CLKDIV.RATIO + 1)*(CPS.PCNT + 1))} \quad (27)$$

28.2.1.1.2 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.

28.2.1.2 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 advancing, the repeat counter (TIMA.RC) advances when the counter reloads (TIMA.CTR = 0). Software can set the how many timer counter reloads occur until generating the interrupts and events by setting the TIMA.RCLD register. When TIMA.RC = TIMA.RCLD, the repeat counter is reset back to zero 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 28.2.6 and Section 28.2.10 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 28-15) are suppressed by setting the TIMA.CCCTL_xy[0/1].SCERCNEZ bit

Table 28-2 shows the repeat counter behavior with respect to the timer counter and repeat counter load value.

Table 28-2. Repeat Counter Behavior

TIMA.CTR is Advancing (+1)	Counter value	TIMA.RC = TIMA.RCLD	Repeat Counter Behavior	Suppress Load and Zero Events (SLZERCNEZ = 1)	Suppress Compare Events (SCERCNEZ = 1)
No	-	-	Does not advance	Yes	Yes
Yes	TIMA.CTR ≠ 0	-	Does not advance	Yes	Yes
Yes	TIMA.CTR = 0	No	Advance (+1)	Yes	Yes
Yes	TIMA.CTR = 0	Yes	TIMA.RC = 0	No	No

Repeat counter example

As shown in Figure 28-3, 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 (which occurs once TIMA.RC = TIMA.RCLD).

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.

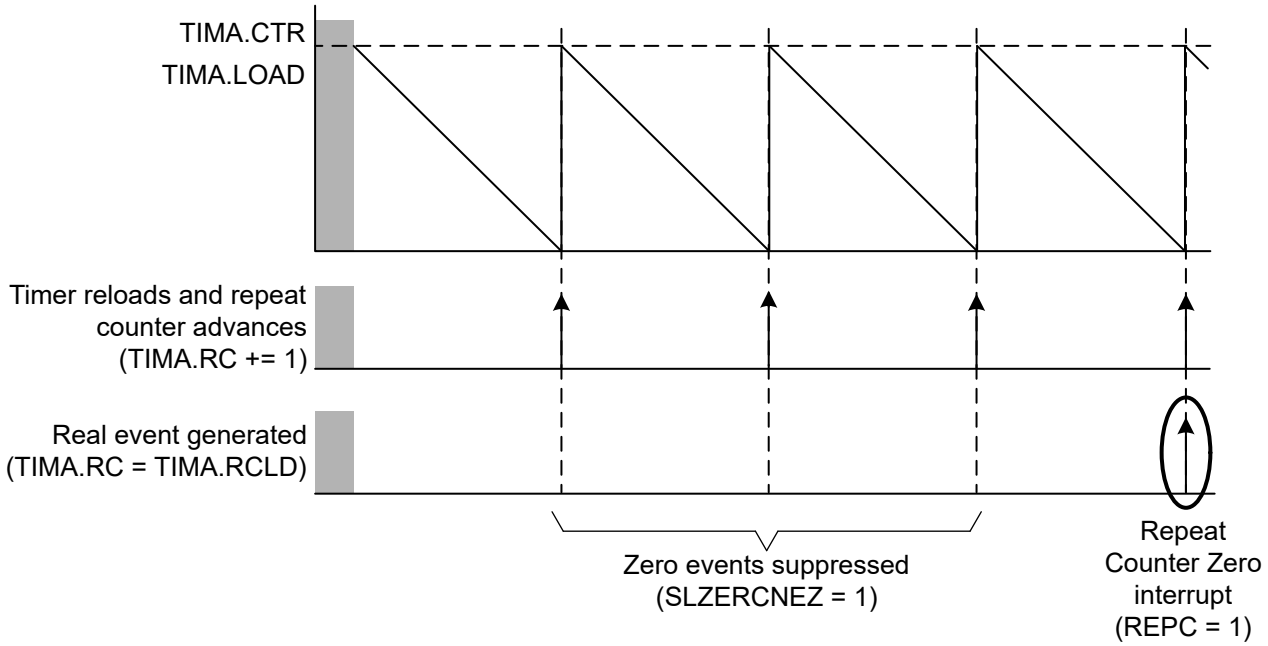


Figure 28-3. Event Suppressed by Repeat Counter with TIMx.RCLD = 4 in Down Counting Mode

28.2.2 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 28-3). After the counter is enabled, the timer will start counting from the TIMx.CTRCTL.CVAE setting.

Table 28-3. TIMx Counting Modes (CM)

TIMx.CTRCTL.CM	Counting Mode
0	Down
1	Up/Down
2	Up

Table 28-4. TIMx Counter Value after Enable (CVAE)

Count Value After Enable (CVAE)	Description	Supported Counting Modes
0	LOAD value	Up, up/down, down
1	Unchanged from current value	Up/down
2	Zero value	Up, up/down, down

28.2.2.1 One-shot and Periodic Modes

Figure 28-4 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.

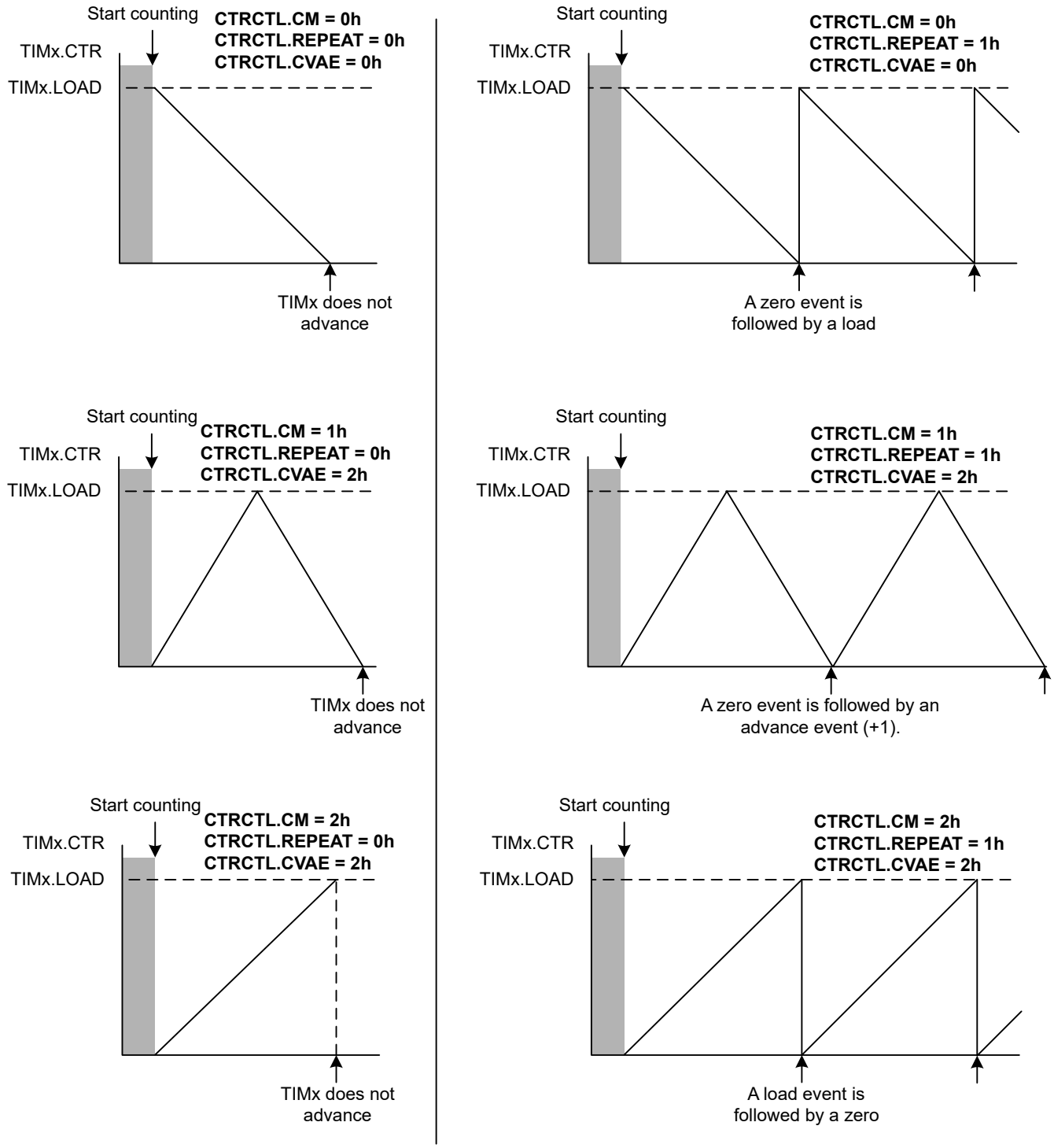


Figure 28-4. One-shot and Periodic Mode Behavior

28.2.2.2 Down Counting Mode

In down counting mode (CM = 0), TIMx counts from the value defined in TIMx.LOAD (CVAE = 0) 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 28-5.

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 28-6 shows the event generating cycle.

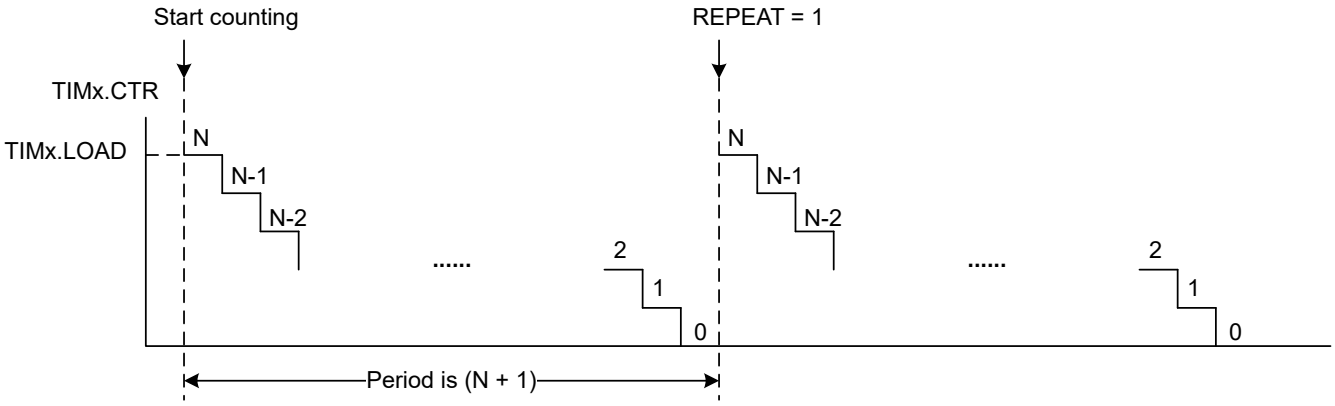


Figure 28-5. Down Counting Mode, CVAE = 0

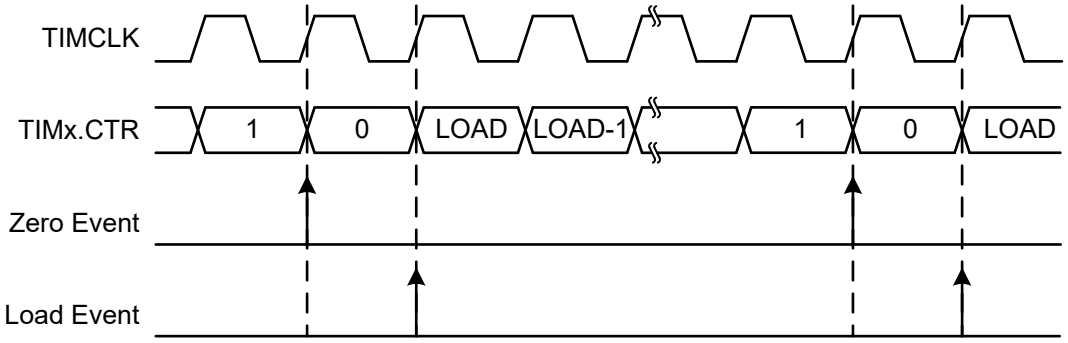


Figure 28-6. Down Counting Mode Event Generation

28.2.2.3 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 28-5. 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 TIMx counts back up to TIMx.LOAD value. A Load event is generated when it reaches TIMx.LOAD value.

Figure 28-7 shows TIMx counting in the down-up direction when TIMx.CTRCTL.CVAE = 0.

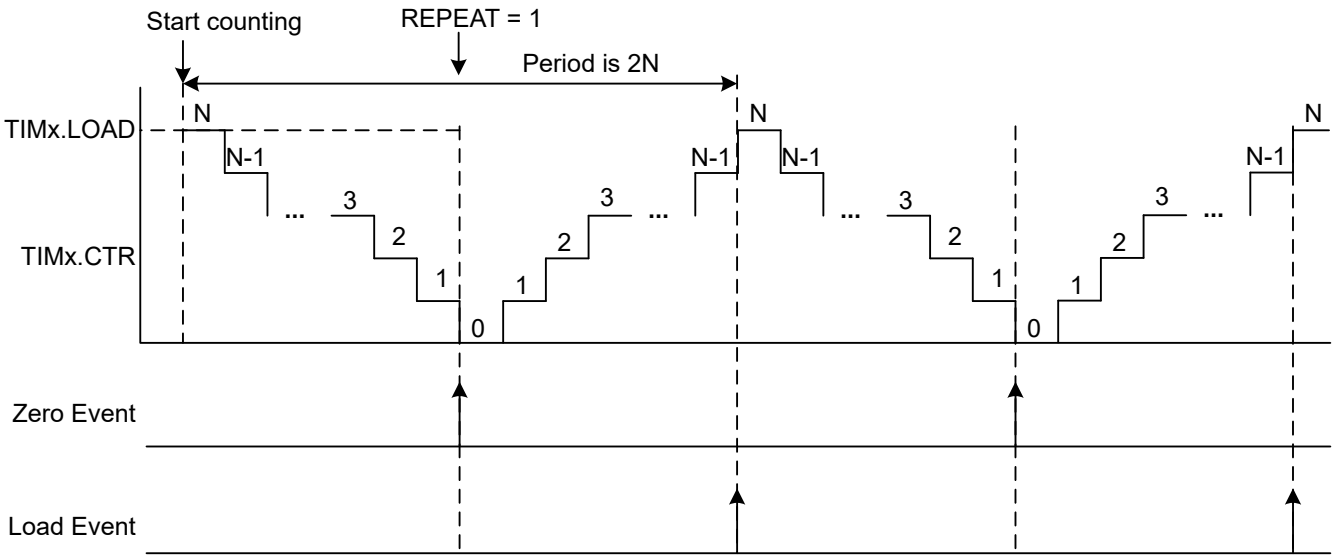


Figure 28-7. Down-up Counting Mode and Event Generation, CVAE = 0

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 28-8 shows TIMx counting in up-down direction when TIMxCTRCTL.CVAE = 2.

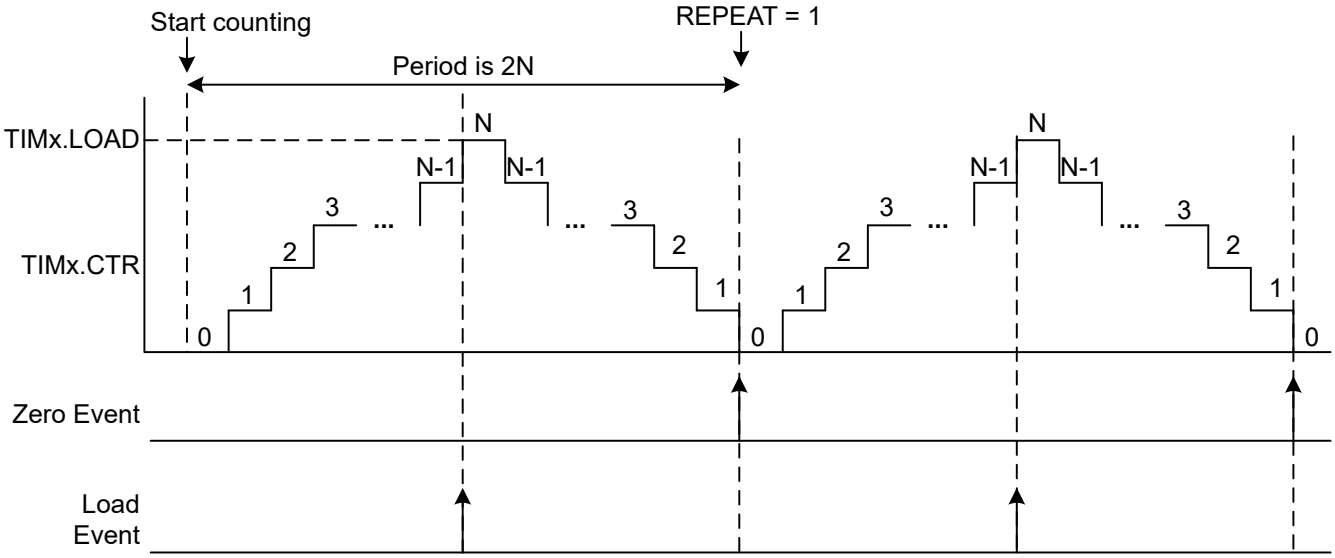


Figure 28-8. Up-down Counting Mode and Event Generation, CVAE = 2

28.2.2.4 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 28-5.

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 28-6 shows the event generating cycle.

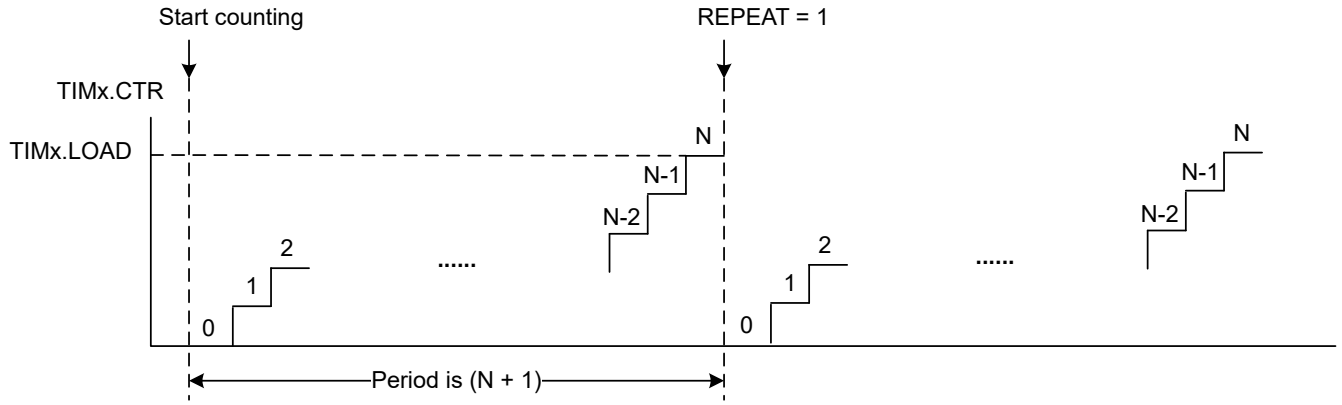


Figure 28-9. Up Counting Mode, CVAE = 2

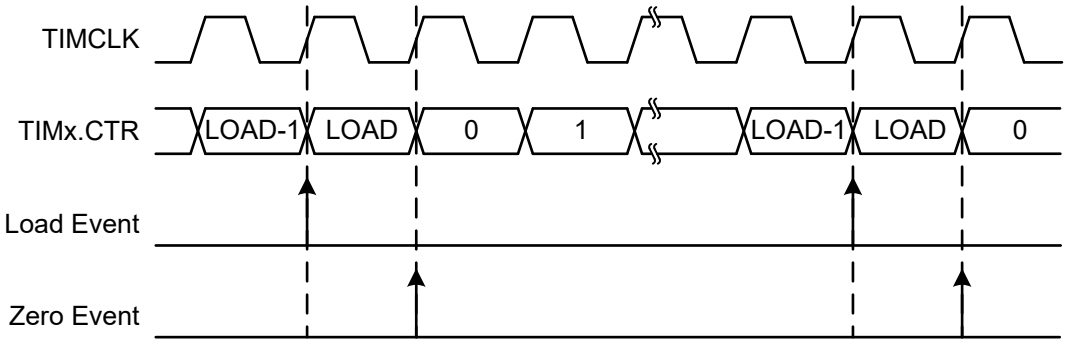


Figure 28-10. Up Counting Mode Event Generation

28.2.2.5 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 28-11 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.

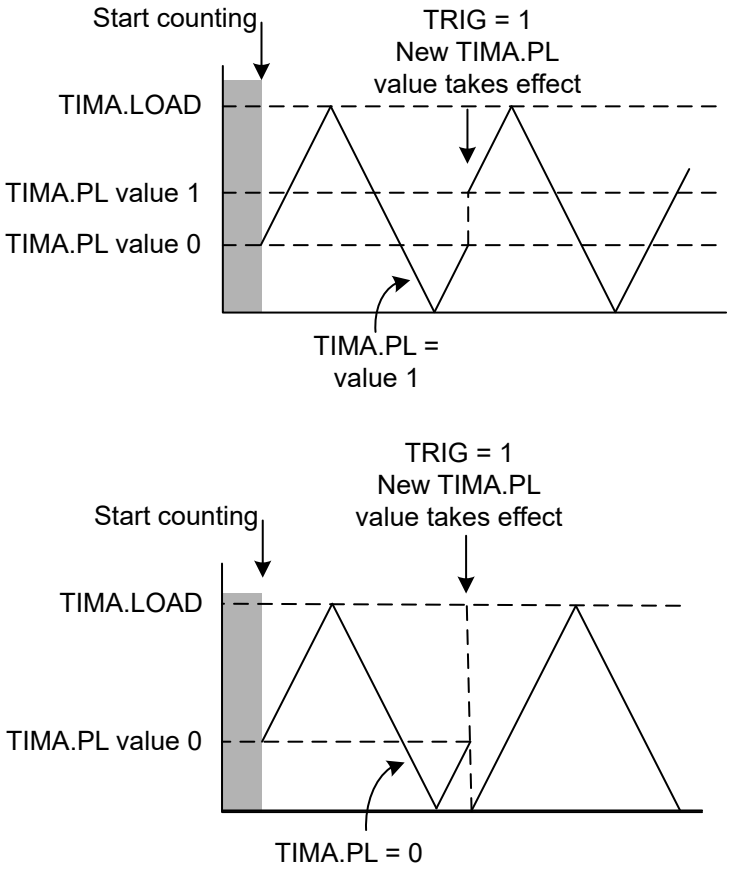


Figure 28-11. Phase Load Register Synchronization in Up-Down Mode

28.2.3 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.

28.2.3.1 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.

28.2.3.1.1 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 28-12 shows the block diagram for the TIMx capture block with two CC channels.

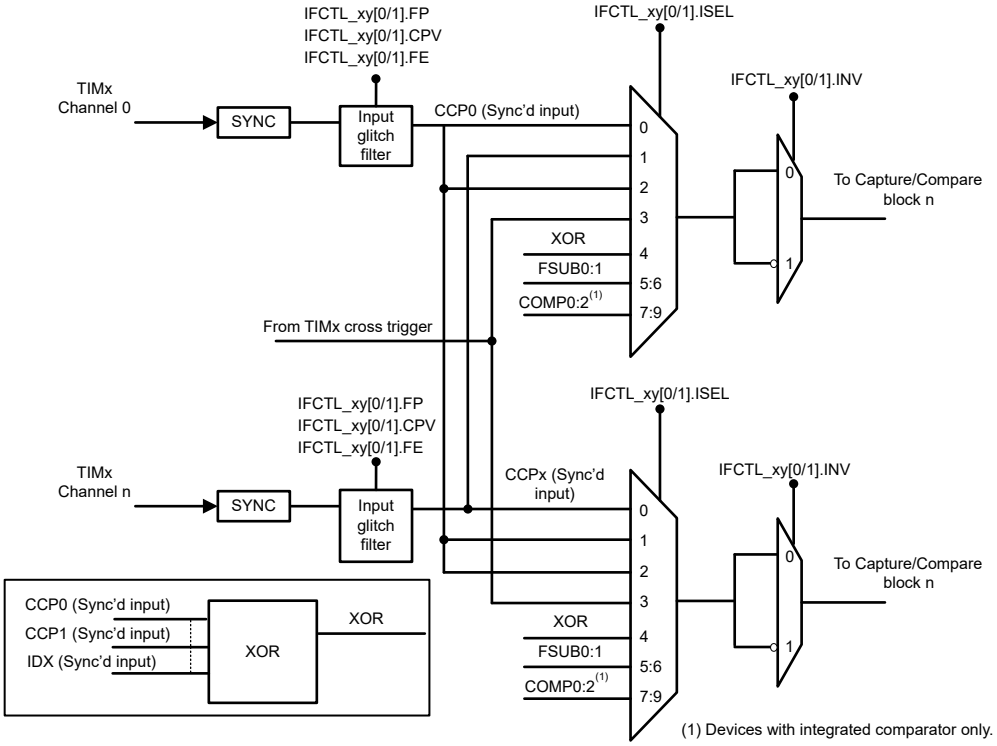


Figure 28-12. TIMx Capture Block Diagram

28.2.3.1.1.1 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.

28.2.3.1.1.2 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.

Table 28-6. 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 28-7. 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

28.2.3.1.1.3 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 28-8 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 28-8. 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 28.2.3.1.3
5h	3-input QEI mode. SeeSection 28.2.3.1.3

See Table 28-9 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 28-9. 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 28.2.3.1.3
5h	3-input QEI mode. SeeSection 28.2.3.1.3

See Table 28-10 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.

Table 28-10. 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 28.2.3.1.3
5h	3-input QEI mode. SeeSection 28.2.3.1.3

28.2.3.1.1.4 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 28-13 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.

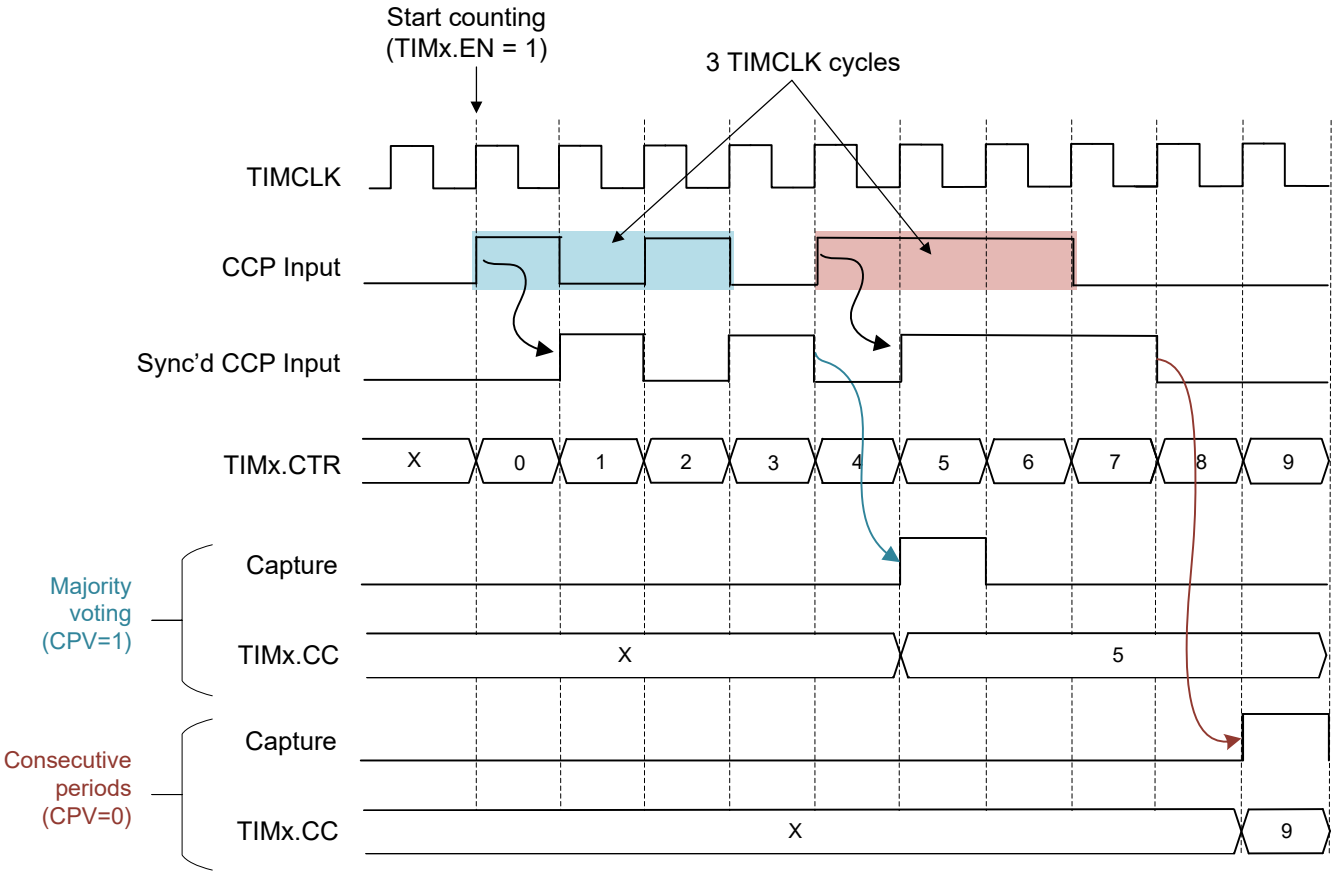


Figure 28-13. Consecutive Period and Majority Voting for Input Glitch Filtering using FP = 0 (3 TIMCLK cycles)

28.2.3.1.1.5 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 28-11. 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 28.2.3.1.4
5h	Subscriber event 0 (FSUB0). See Section 28.2.9.2
6h	Subscriber event 1 (FSUB1). See Section 28.2.9.2
7h	COMP0_OUT (devices with comparator only)
8h	COMP1_OUT (devices with comparator only)
9h	COMP2_OUT (devices with comparator only)

28.2.3.1.2 Use Cases

Several different use cases can be achieved in capture mode and are discussed in the following sections.

28.2.3.1.2.1 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 28.2.2)
   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 28.2.3.1.1.
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 28-14.

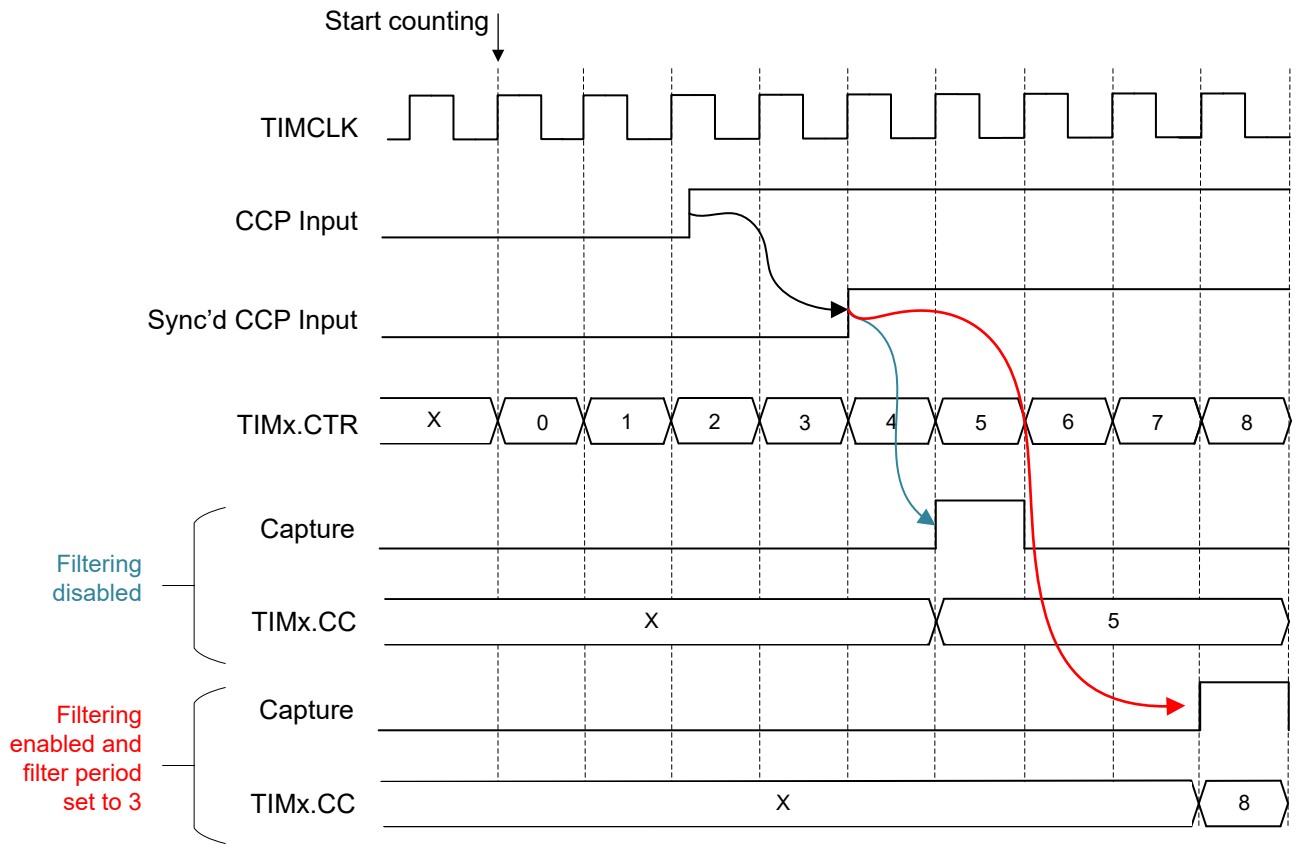


Figure 28-14. Edge Time Capture Mode in Up-Counting Mode, CVAE = 2

28.2.3.1.2.2 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 28.2.2)
   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 28.2.3.1.1.
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

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 28-14.

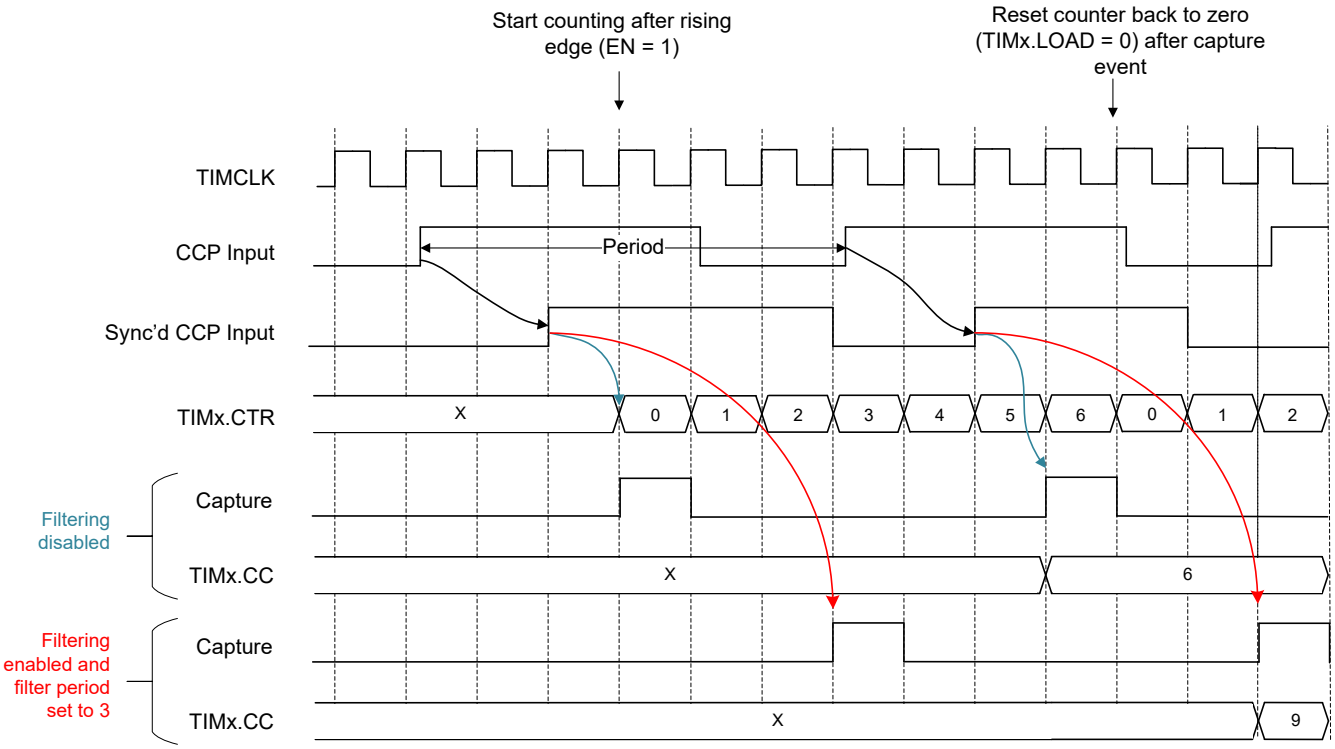


Figure 28-15. Period Capture Mode in Up-Counting Mode, CVAE = 2

28.2.3.1.2.3 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 28.2.2)
   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 28.2.3.1.1.

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 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 28-14.

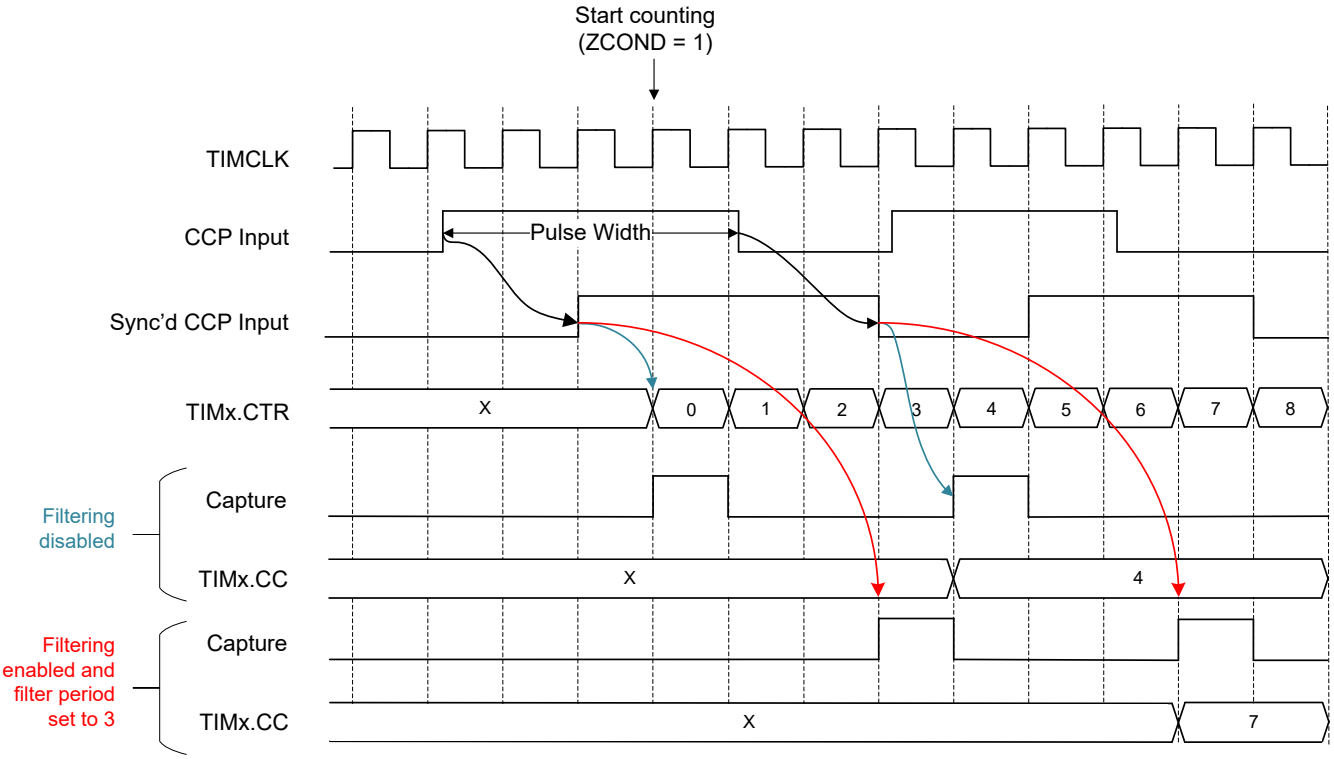


Figure 28-16. Pulse-Width Capture Mode

28.2.3.1.2.4 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 28-17.

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 28.2.2)
   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]),

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 28.2.3.1.1.
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 28-14.

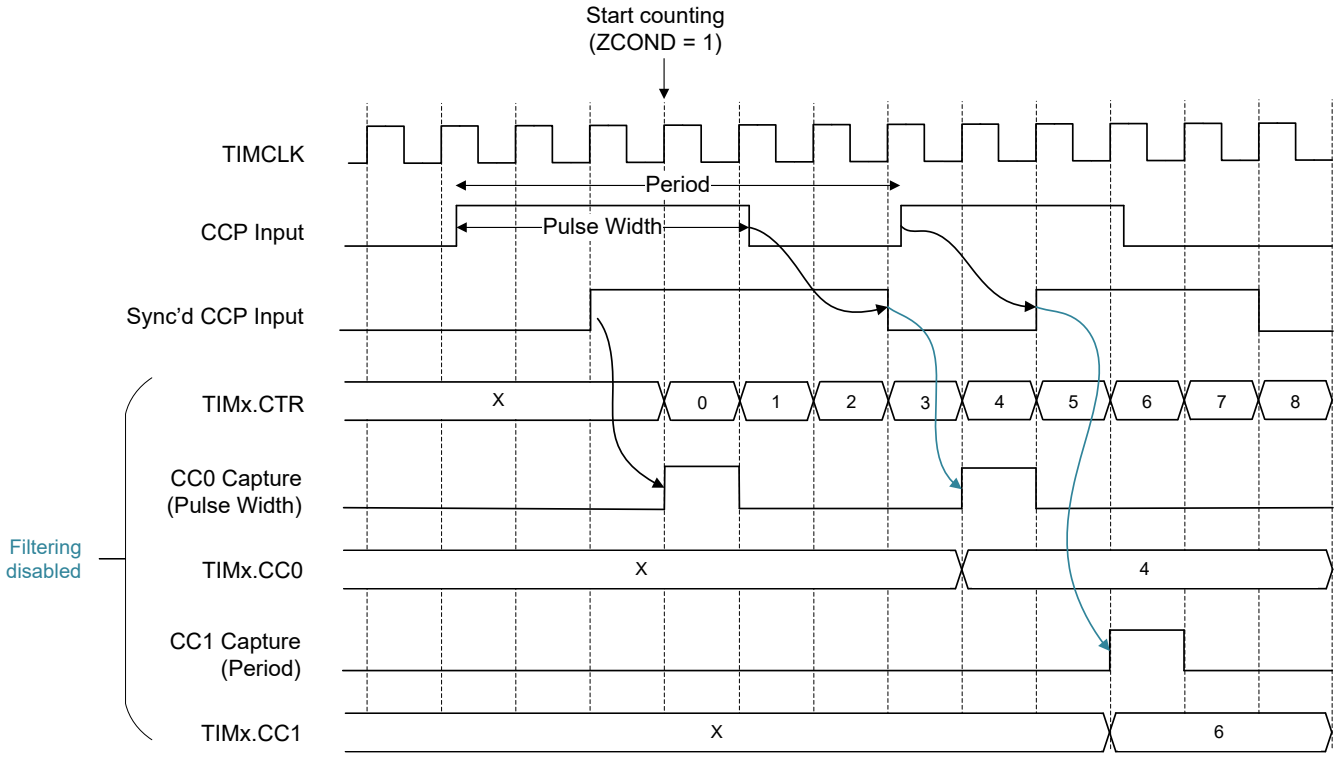


Figure 28-17. Combined Pulse-Width and Period Capture

28.2.3.1.3 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 28.1.3 and the device-specific data sheet for TIMG instances that support QEI / Hall Input Mode.

28.2.3.1.3.1 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 28-18 shows the state machine in the QEI interface to detect the directional rotation from the two CCP input signals, PHA and PHB.

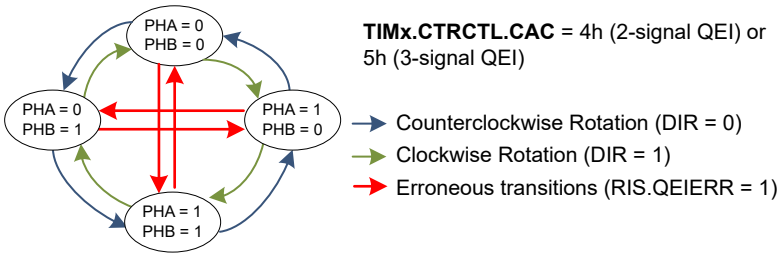


Figure 28-18. State Machine for 2-signal and 3-signal QEI mode

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 28.2.3.1.1, 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 28-12 and Figure 28-19.

Table 28-12. 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		

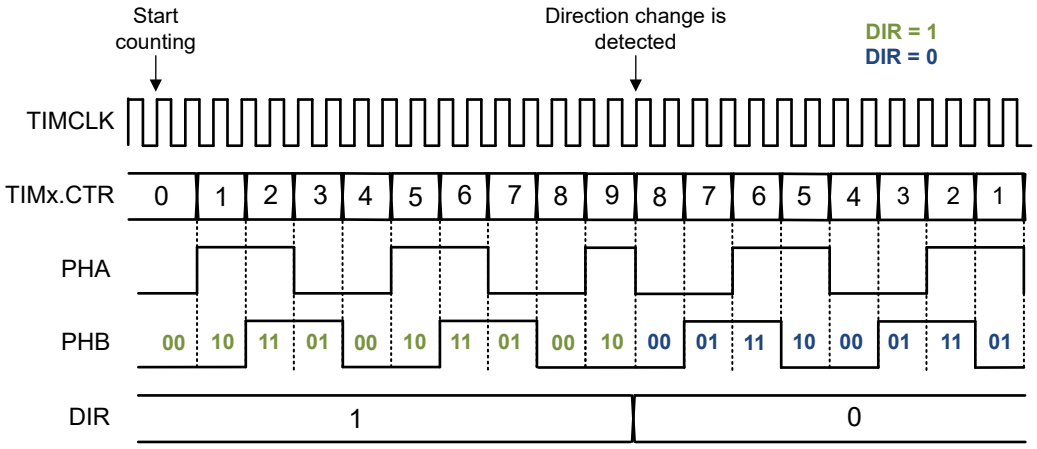


Figure 28-19. 2-Signal QEI Operation

28.2.3.1.3.2 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 28.2.3.1.1, 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 28-13 and

Table 28-13. 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)

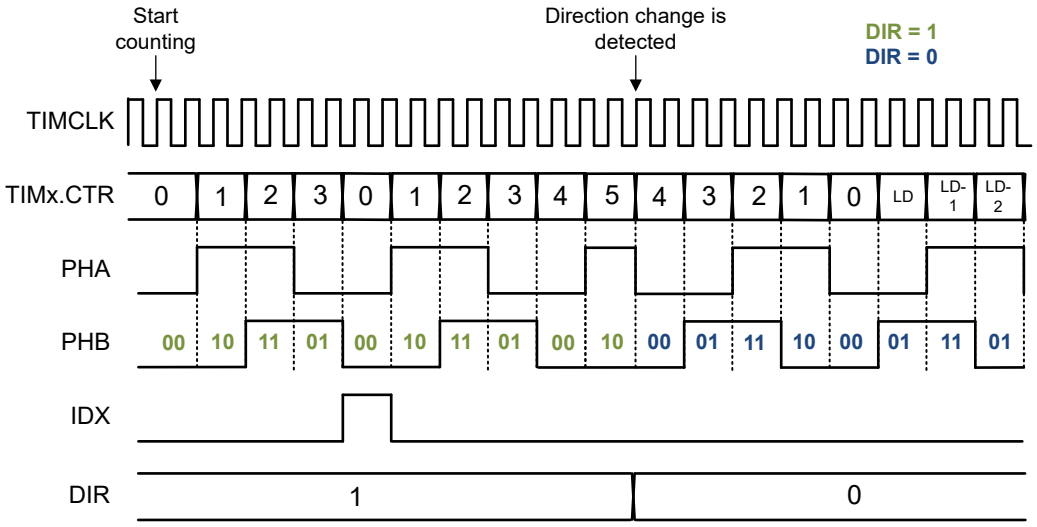


Figure 28-20. 2-Signal QEI with Index Input Operation

28.2.3.1.3.3 QEI Error Detection

The QEI module can detect erroneous transactions or state errors as shown in Figure 28-18. 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.

28.2.3.1.4 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 28.1.3 and the device-specific data sheet for TIMG instances that support QEI / Hall Input Mode.

Table 28-14 shows the signal mapping for Hall signals A (U), B (V), and C (W) to TIMG capture/compare input signals.

Table 28-14. 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.

As shown in Figure 28-21, 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 28-21 for XOR option in the input capture block diagram.

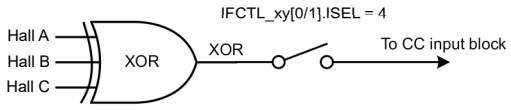


Figure 28-21. Hall 3-input XOR for Frequency Generator (FG) Signal to CC input block

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 28.2.3.1.2.2 and Section 28.2.3.1.2.3 on how to calculate period and pulse-width captures based on the XOR'ed input signal.

Figure 28-22 shows the input signal to the CC block which can be used for speed calculations.

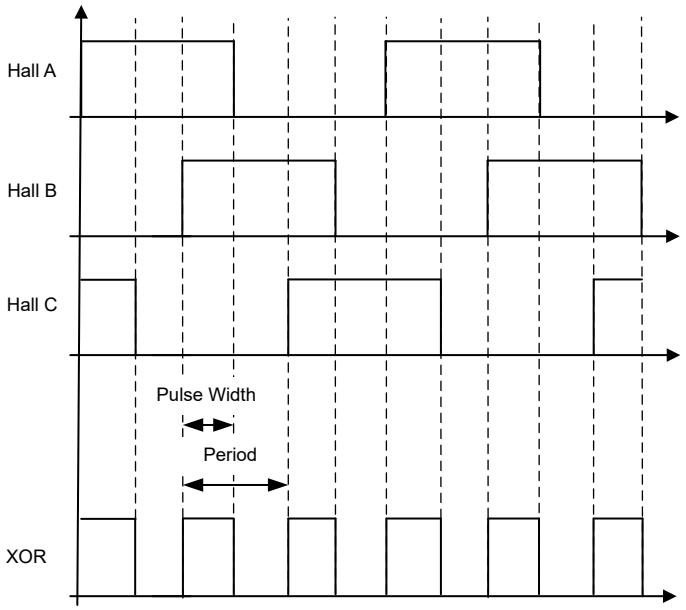


Figure 28-22. Hall 3-Input XOR Signal used with Pulse Width or Period Capture for Speed Computation

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.

28.2.3.2 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.

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 (CC2U) 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 28-15 shows the types of compare mode events that can be generated and conditions to generate the events.

Table 28-15. 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.

28.2.3.2.1 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 28.2.2)
   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 28.2.3.1.1, if desired.
6. Enable the counter by setting EN = 1.

Example using edge count operation using up-counting mode

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 28-23.

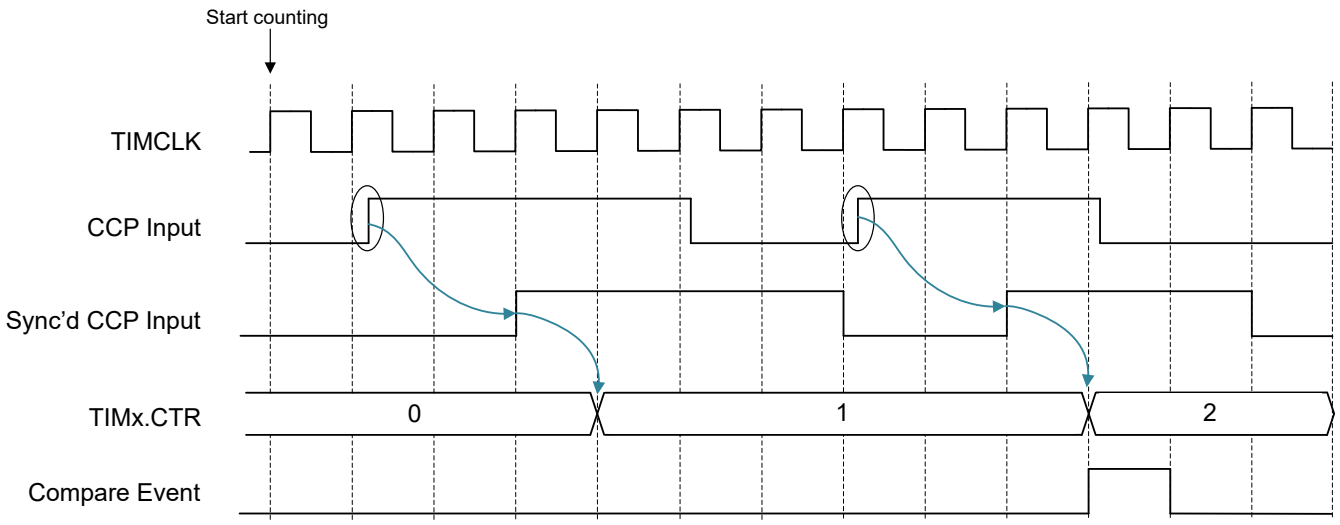


Figure 28-23. Edge Count Operation to Generate Compare Event (TIMx.CC = 2)

28.2.4 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 28.1.3 and the device-specific data sheet for TIMx instances that support Shadow Load and Shadow Compare.

28.2.4.1 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).

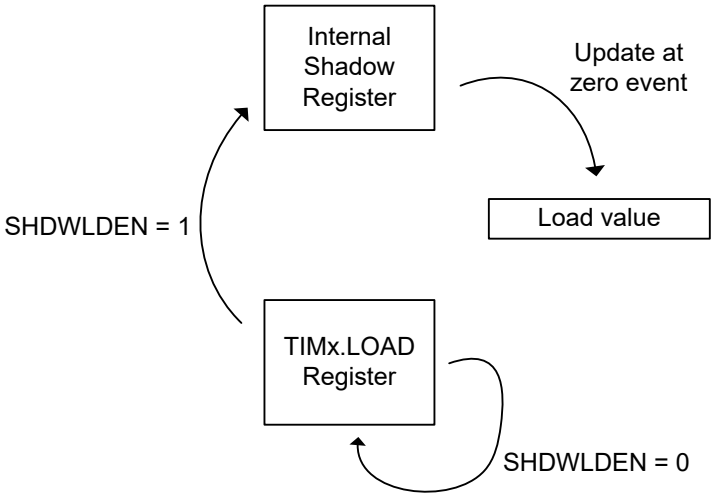
If the TIMx module has a shadow load feature, there is an internal shadow register for the load value (TIMx.LOAD). The shadow register will update the load value at a zero event as shown in Figure 28-24.

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.

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

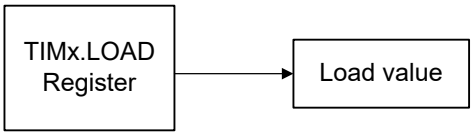


Figure 28-24. Load value updates for TIMx instances when EN=1

Note

To update the load value for shadow load capable timers, 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 28-25 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.

28.2.4.2 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 28-16 shows the settings for configuring when shadow compare and actions occur at different events.

Table 28-16. 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). **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. See Section 28.2.7.

Figure 28-25 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.

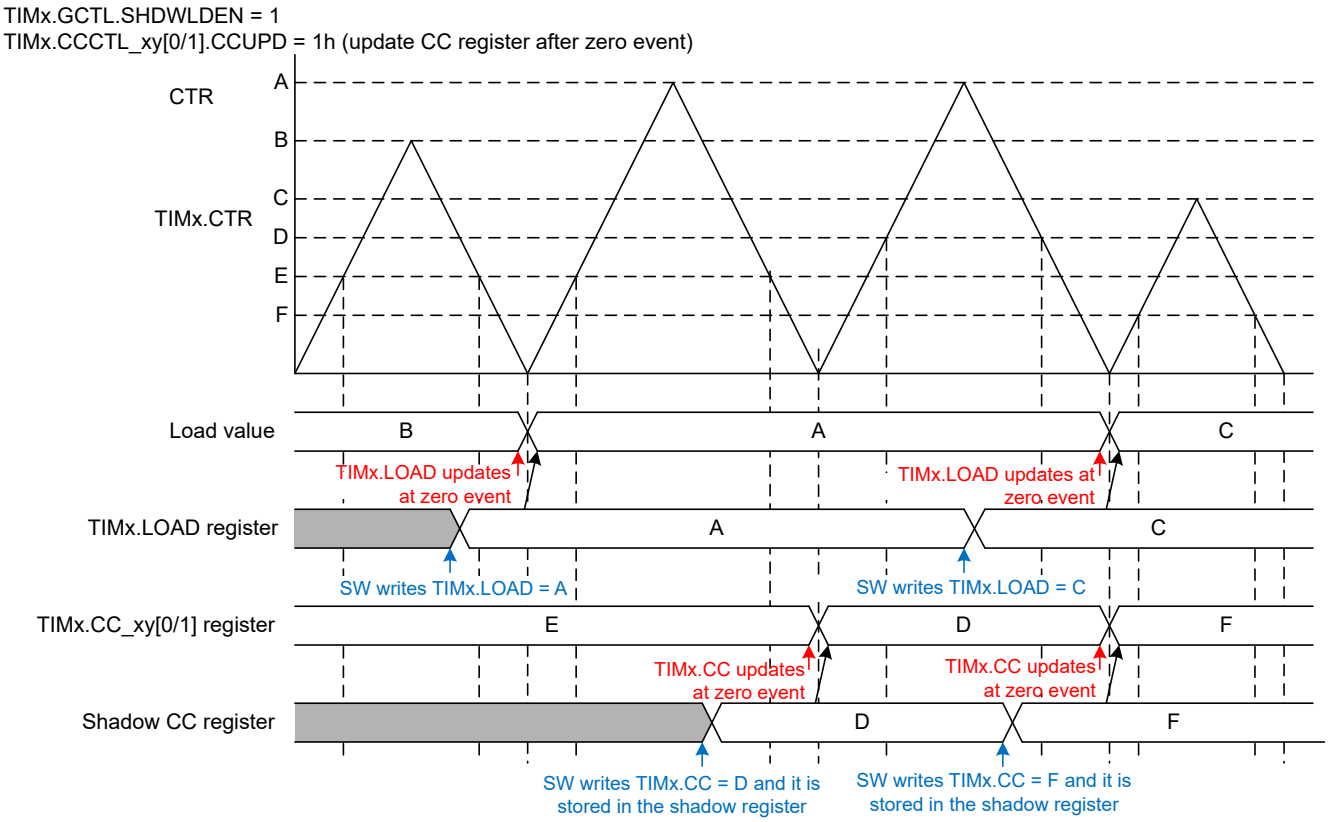


Figure 28-25. Shadow Load and Shadow Compare Taking Effect at Zero Event in Up/Down Mode

28.2.5 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 28-27 shows the TIMG output block diagram.

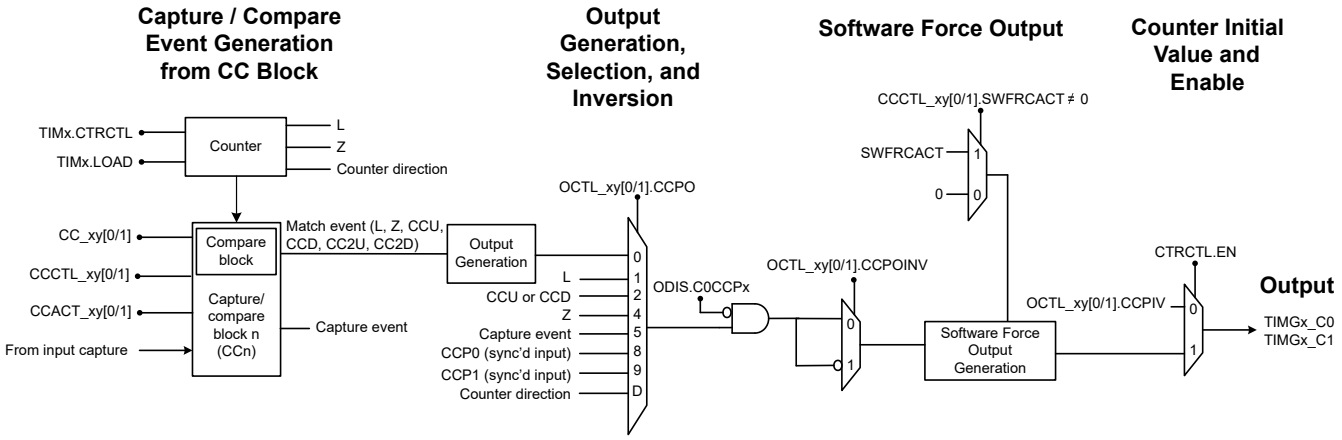


Figure 28-26. Output Connection for TIMG

Figure 28-27 shows the TIMA output block diagram.

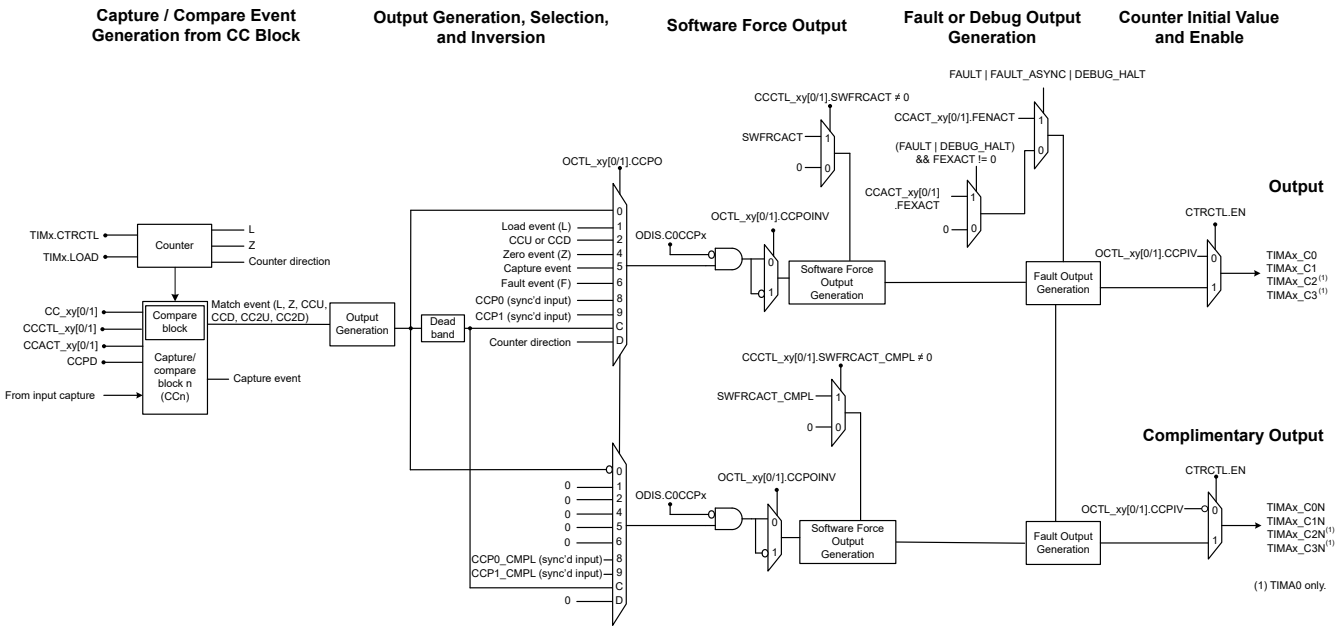


Figure 28-27. Output Connection for TIMA

Signal Generator Actions

Table 28-17 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 28-15.

Table 28-17. 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:

28.2.5.1 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 28-15 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 28.2.5.2.4.

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 28.2.5.3.

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 28.2.6 and Section 28.2.10.

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.

28.2.5.2 Use Cases

Several different use cases can be achieved with the output generator and are discussed in the following sections.

28.2.5.2.1 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 28.2.2)
   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 = 1 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. Enable the corresponding CCP output by setting ODIS.C0CCPn to 1 for the corresponding counter n.
9. Configure polarity of the signal using the CCPOINV bit, and configure CCPIV to specify the CCP output state while disabled.
10. 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 28-28 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)

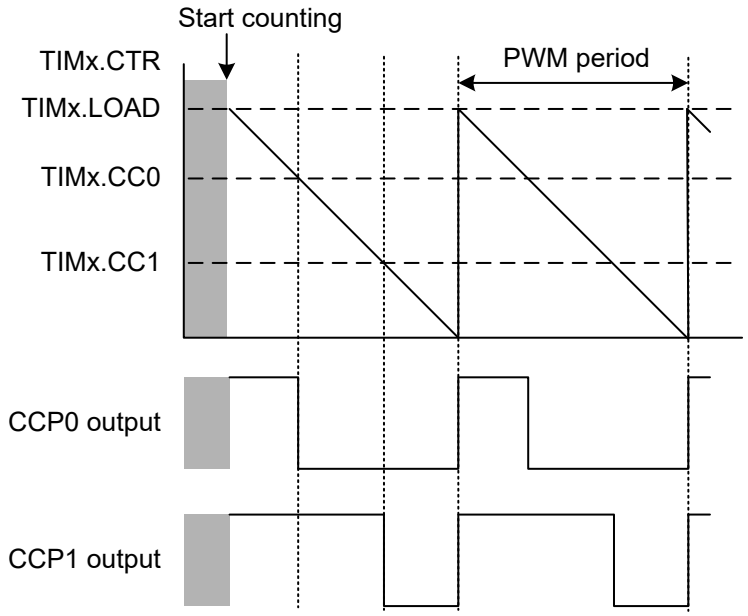


Figure 28-28. Edge-Aligned PWM Signals in Down-Counting Mode

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 28-28 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)

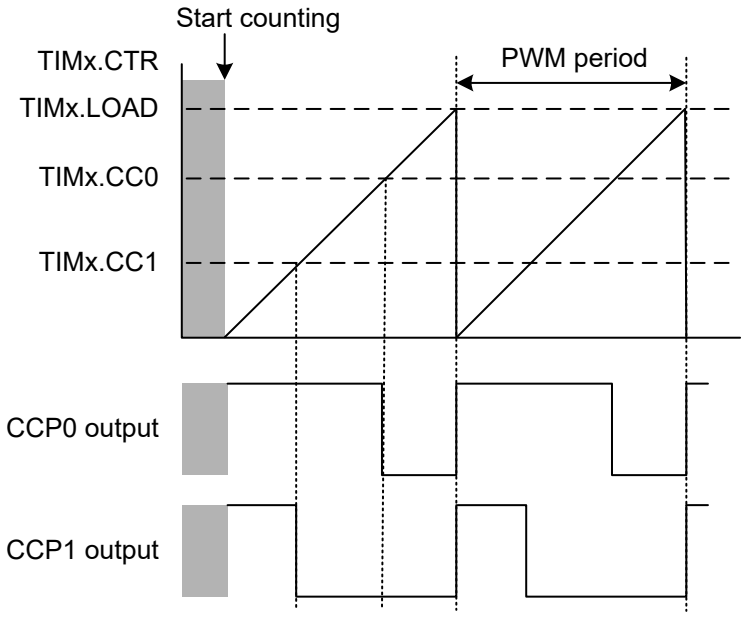


Figure 28-29. Edge-Aligned PWM Signals in Up-Counting Mode

28.2.5.2.2 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 28.2.2)
   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 = 1 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. Enable the corresponding CCP output by setting ODIS.C0CCPn to 1 for the corresponding counter n.
9. Configure polarity of the signal using the CCPOINV bit, and configure CCPIV to specify the CCP output state while disabled.
10. 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 28-30 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)

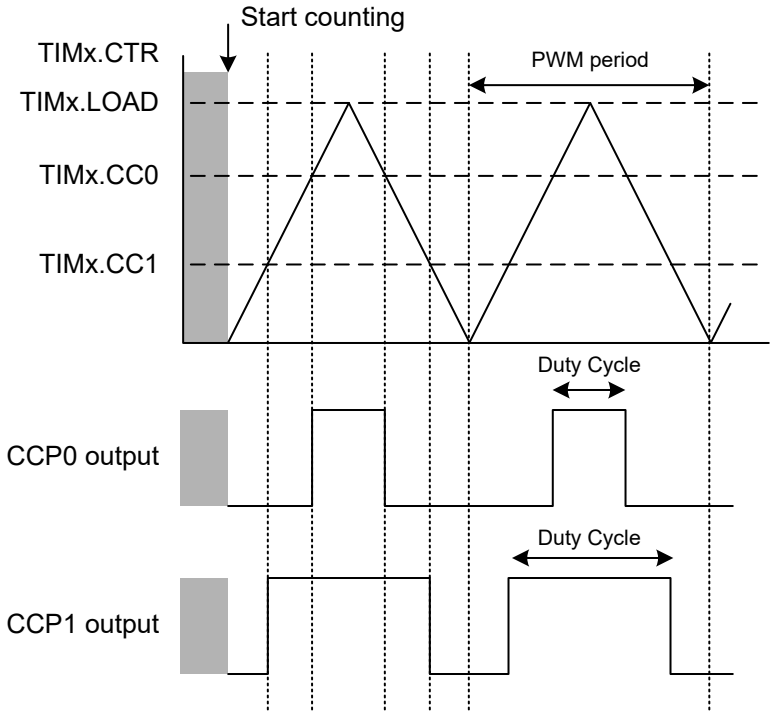


Figure 28-30. Center-Aligned PWM

28.2.5.2.3 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 28.2.2.5.

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 28.2.7.
2. Configure two center-aligned PWMs as described in Section 28.2.5.2.2 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 28.2.2.5.
4. Enable the counter by setting TIMA.CTRCTL.EN = 1.

Figure 28-31 shows an example of asymmetric PWM configuration using CCP channel 0 of TIMA0 and TIMA1.

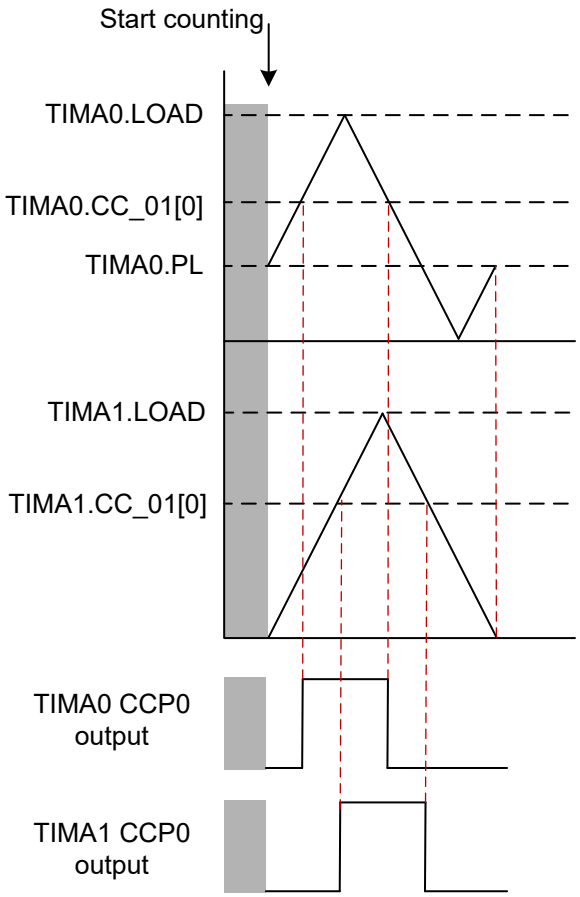


Figure 28-31. Asymmetric PWM Configuration with Phase Load for CCP channel 0 of TIMA0 and TIMA1

28.2.5.2.4 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 28-18 for the configuration and relationship between deadband mode and deadband width settings.

Table 28-18. 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. FALLDELAY is applied from **falling edge of reference PWM** to rising edge of CxN signal.	Any

Table 28-18. 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 28 and Equation 29.

$$RISEDELAY = f_{TIMCLK} \times t_{dead\_rise} \quad (28)$$

$$FALLDELAY = f_{TIMCLK} \times t_{dead\_fall} \quad (29)$$

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 28.2.5.2.1) or center-aligned PWM (Section 28.2.5.2.2) 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 28-32 for inserting configurable deadband using down counting mode, TIMA output channel 0, and edge-aligned PWM.

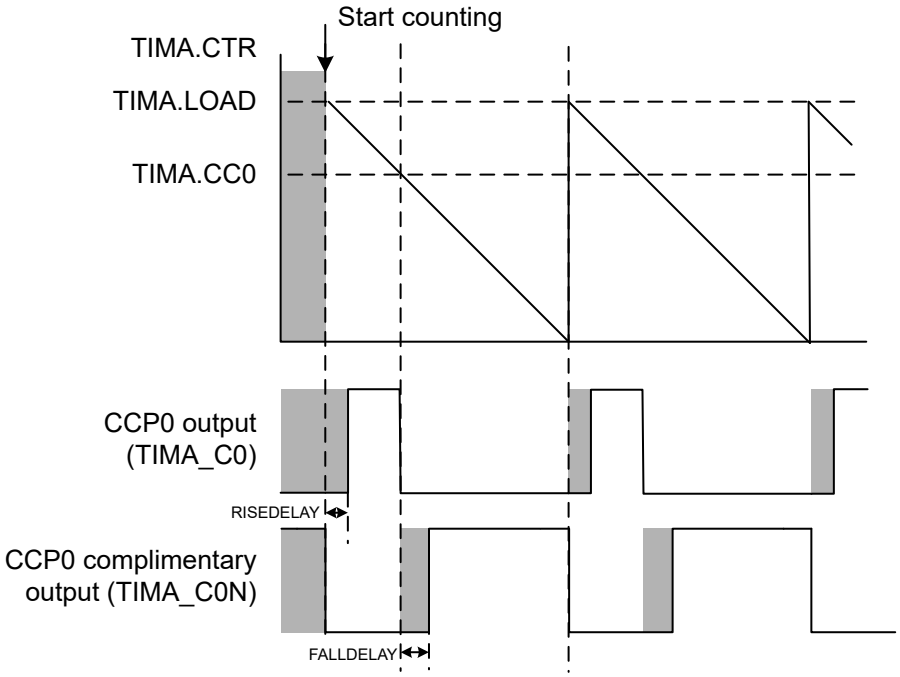


Figure 28-32. Deadband Insertion (Mode 0) for PWM in Down-Counting Mode

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 28-33 for inserting configurable deadband using up/down counting mode, TIMA output channel 0, and center-aligned PWMs with deadband.

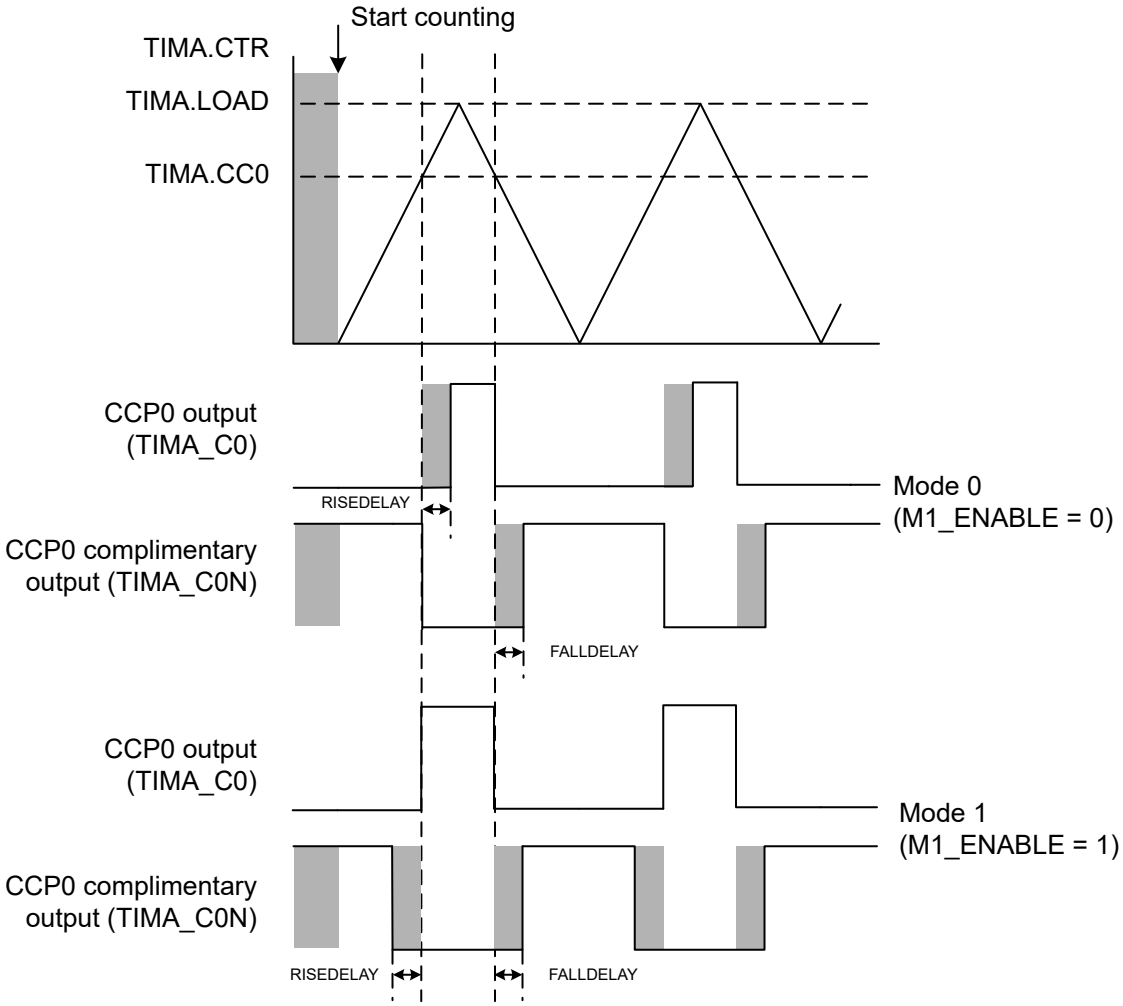


Figure 28-33. Dead Band Insertion for Center-Aligned PWM (Mode 0 and Mode 1)

28.2.5.3 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 28-19 shows the software force output action configuration options.

Table 28-19. 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

28.2.6 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 28-34 shows the asynchronous fault handler logic connections.

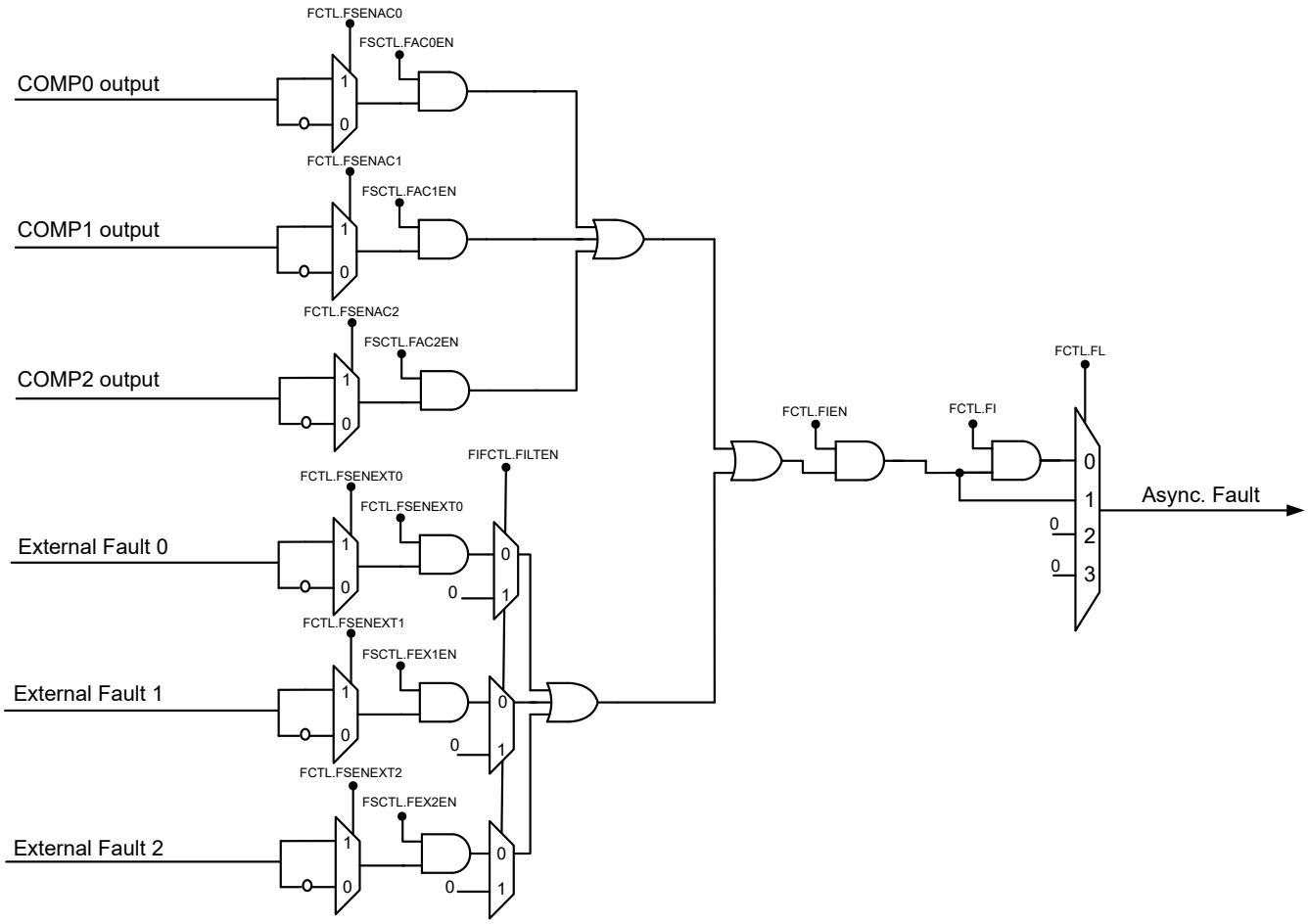


Figure 28-34. Asynchronous Fault Handler Connections

Figure 28-35 shows the synchronous fault handler logic connections.

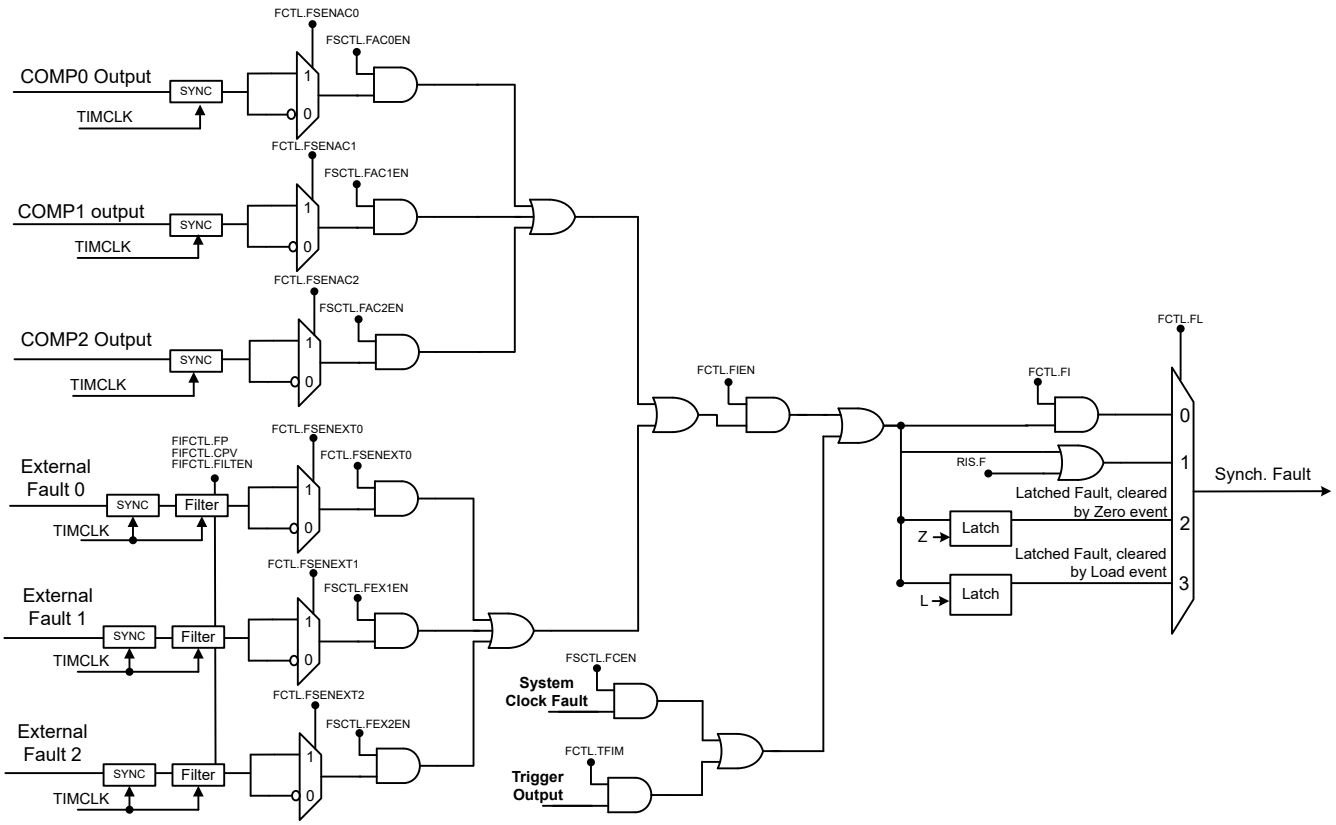


Figure 28-35. Synchronous Fault Handler Connections

Figure 28-36 shows the fault output generation logic connections.

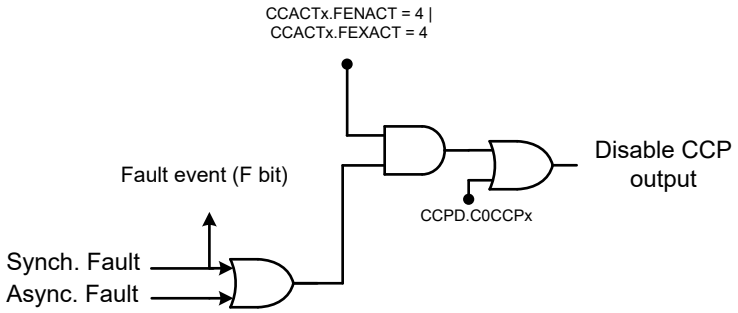


Figure 28-36. Fault Output Generation Connections

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.

28.2.6.1 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.

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 28-13 shows the difference between consecutive period and majority voting formats with a digital filter implemented to capture a fault input of 3 TIMCLK periods.

28.2.6.2 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 28-20.

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 28.2.7. The fault input mask can be enabled by setting the TIMA.FSCTL.TFIM bit.

Table 28-20. 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

28.2.6.3 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 28-21 and Figure 28-37.

Table 28-21. 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).

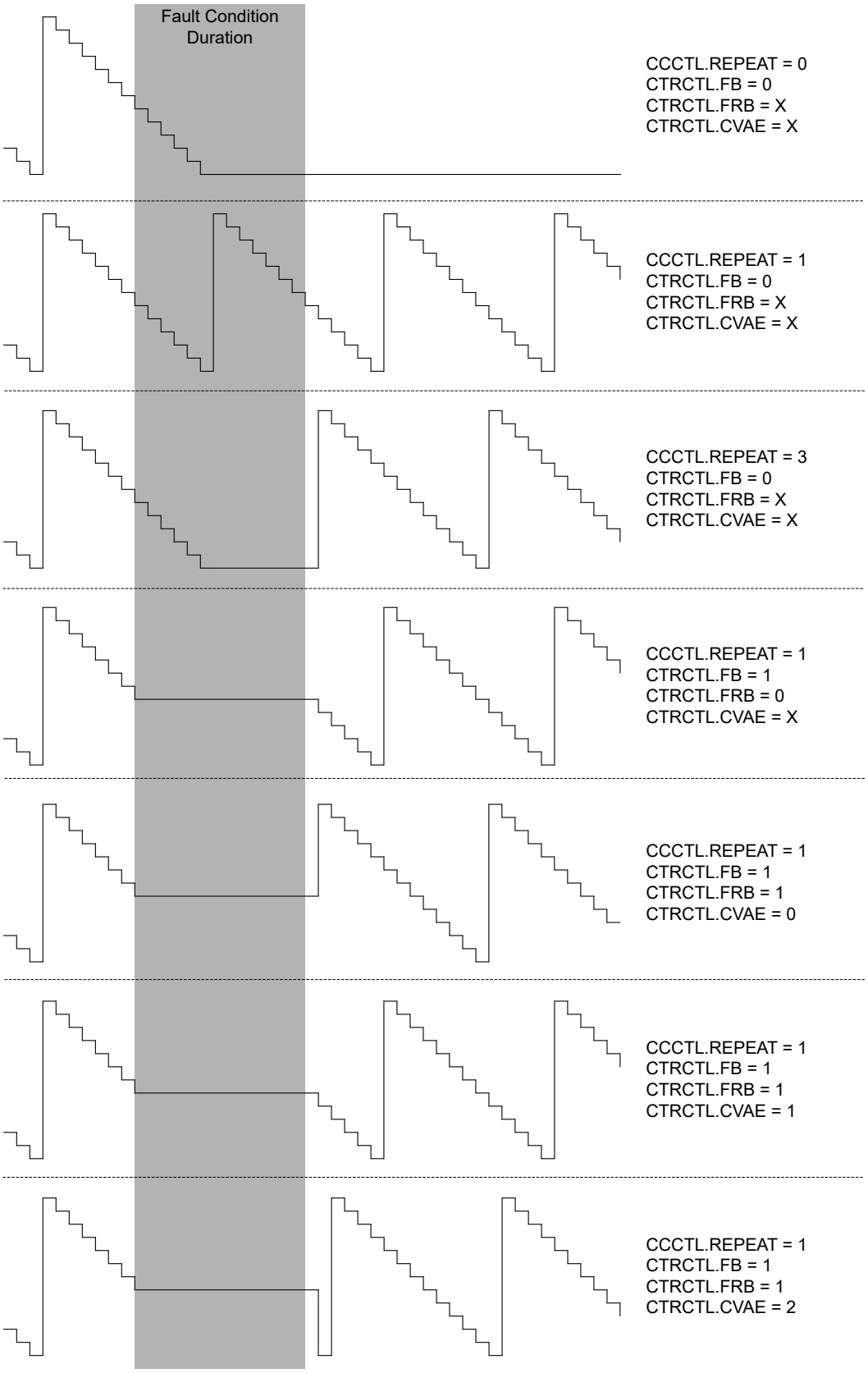


Figure 28-37. Counter Behavior in Fault Condition with TIMx.CTRCTL Register

28.2.6.4 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 28-22.

Table 28-22. 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).

28.2.7 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 28-38, 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.

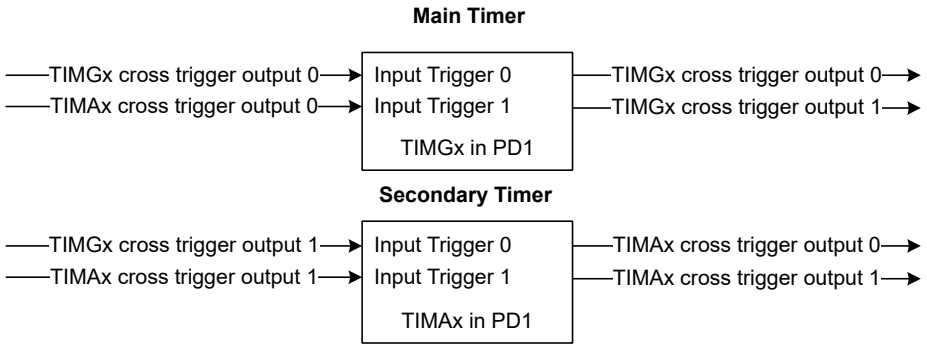


Figure 28-38. Cross Trigger Connections for Main Timer (TIMGx) and Secondary Timer (TIMAx) in Power Domain 1

28.2.7.1 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 28.2.5 for how to configure for PWM generation.
2. Select which cross trigger output needs to be generated. For example, in Figure 28-38, TIMGx cross trigger 1 can be used to trigger TIMAx and TIMGx cross trigger 0 can be used to trigger itself.
3. Enable the cross trigger output function by setting TIMx.CTTRIGCTL.CTEN bit to 1.
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 28-39 shows the connection logic and registers.

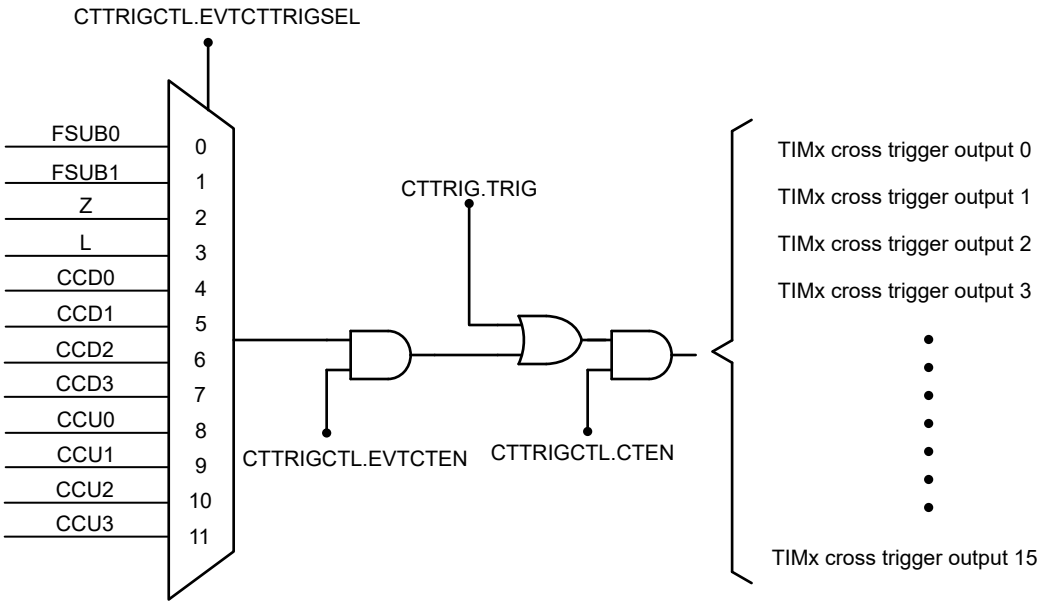


Figure 28-39. Main Timer Cross Trigger Output Configuration

28.2.7.2 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 28.2.5 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 28-38, 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 28-40 shows the logic connection.

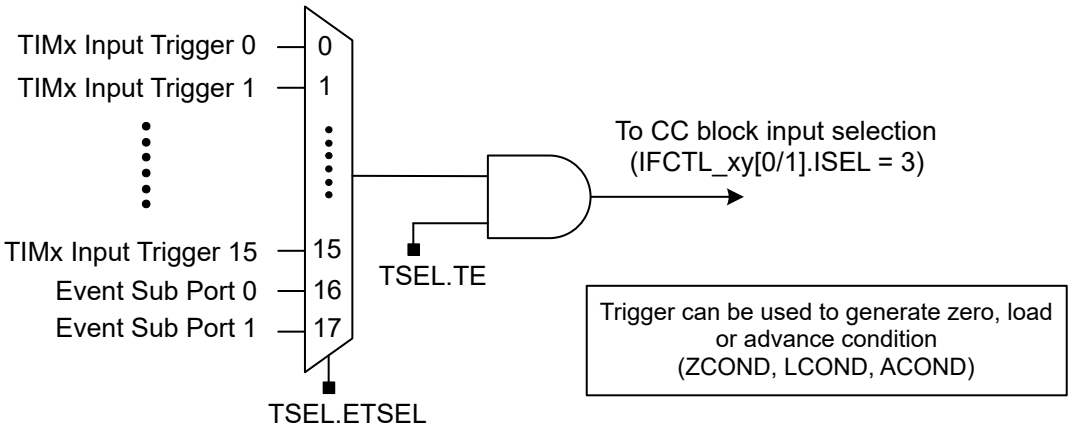


Figure 28-40. Secondary Timer Cross Trigger Input Configuration

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.

28.2.8 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.

28.2.9 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 8.2.5 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 28-23.

Table 28-23. 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
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

28.2.9.1 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 28-24 lists the CPU interrupt events from the TIMx in order of decreasing interrupt priority.

Table 28-24. 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

Table 28-24. TIMx CPU Interrupt Event Conditions (CPU_INT) (continued)

IIDX STAT	Name	Description	Timer Module
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 28.2.6. 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
0X1D	QEIERR	Direction change interrupt, used in QEI mode. This interrupt is only available for TIMG modules with QEI features.	TIMG

See Section 8.2.5 for guidance on configuring the event registers for CPU interrupts.

28.2.9.2 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 28-25) 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 8.1.3.3 and Section 8.2.3 for how generic event route works.

Table 28-25. 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

Table 28-25. TIMx Generic Event Conditions (GEN_EVENT0 and GEN_EVENT1) (continued)

IIDX STAT	Name	Description	Timer Module
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 28.2.6. 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
0x1D	QEIERR	Direction change interrupt, used in QEI mode. This interrupt is only available for TIMG modules with QEI features.	TIMG

See Section 8.2.5 for guidance on configuring the event registers.

28.2.9.3 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 28-12

   and Section 28.2.3.1.1.5.
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 28-12.
- 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.

28.2.10 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.

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 32.2.1.2 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.

28.3 TIMx Registers

Table 28-26 lists the memory-mapped registers for the TIMx registers. All register offset addresses not listed in Table 28-26 should be considered as reserved locations and the register contents should not be modified.

Table 28-26. 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_EVENT0	Go
1058h	IMASK	Interrupt mask	GEN_EVENT0	Go
1060h	RIS	Raw interrupt status	GEN_EVENT0	Go
1068h	MIS	Masked interrupt status	GEN_EVENT0	Go
1070h	ISET	Interrupt set	GEN_EVENT0	Go
1078h	ICLR	Interrupt clear	GEN_EVENT0	Go
1080h	IIDX	Interrupt index	GEN_EVENT1	Go
1088h	IMASK	Interrupt mask	GEN_EVENT1	Go
1090h	RIS	Raw interrupt status	GEN_EVENT1	Go
1098h	MIS	Masked interrupt status	GEN_EVENT1	Go
10A0h	ISET	Interrupt set	GEN_EVENT1	Go
10A8h	ICLR	Interrupt clear	GEN_EVENT1	Go
10E0h	EVT_MODE	Event Mode		Go
10FCh	DESC	Module Description		Go
1100h	CCPD	CCP Direction		Go
1104h	ODIS	Output Disable		Go
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

Table 28-26. TIMX Registers (continued)

Offset	Acronym	Register Name	Group	Section
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 28-27 shows the codes that are used for access types in this section.

Table 28-27. TIMx Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
WK	W K	Write 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.

Table 28-27. TIMx Access Type Codes (continued)

Access Type	Code	Description
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

28.3.1 FSUB_0 (Offset = 400h) [Reset = 00000000h]

FSUB_0 is shown in Figure 28-41 and described in Table 28-28.

Return to the Summary Table.

Subscriber port

Figure 28-41. 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 28-28. FSUB_0 Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3-0	CHANID	R/W	0h	0 = disconnected. 1-15 = connected to channelID = CHANID. 0h = A value of 0 specifies that the event is not connected Fh = Consult your device data sheet as the actual allowed maximum may be less than 15.

28.3.2 FSUB_1 (Offset = 404h) [Reset = 00000000h]

FSUB_1 is shown in Figure 28-42 and described in Table 28-29.

Return to the Summary Table.

Subscriber port

Figure 28-42. 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 28-29. FSUB_1 Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3-0	CHANID	R/W	0h	0 = disconnected. 1-15 = connected to channelID = CHANID. 0h = A value of 0 specifies that the event is not connected Fh = Consult your device data sheet as the actual allowed maximum may be less than 15.

28.3.3 FPUB_0 (Offset = 444h) [Reset = 00000000h]

FPUB_0 is shown in Figure 28-43 and described in Table 28-30.

Return to the Summary Table.

Publisher port

Figure 28-43. 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 28-30. FPUB_0 Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3-0	CHANID	R/W	0h	0 = disconnected. 1-15 = connected to channelID = CHANID. 0h = A value of 0 specifies that the event is not connected Fh = Consult your device data sheet as the actual allowed maximum may be less than 15.

28.3.4 FPUB_1 (Offset = 448h) [Reset = 00000000h]

FPUB_1 is shown in Figure 28-44 and described in Table 28-31.

Return to the Summary Table.

Publisher port

Figure 28-44. 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 28-31. FPUB_1 Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3-0	CHANID	R/W	0h	0 = disconnected. 1-15 = connected to channelID = CHANID. 0h = A value of 0 specifies that the event is not connected Fh = Consult your device data sheet as the actual allowed maximum may be less than 15.

28.3.5 PWREN (Offset = 800h) [Reset = 00000000h]

PWREN is shown in Figure 28-45 and described in Table 28-32.

Return to the Summary Table.

Register to control the power state

Figure 28-45. 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 28-32. PWREN Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to allow Power State Change 26h = KEY to allow write access to this register
23-1	RESERVED	R	0h	
0	ENABLE	R/WK	0h	Enable the power KEY must be set to 26h to write to this bit. 0h = Disable Power 1h = Enable Power

28.3.6 RSTCTL (Offset = 804h) [Reset = 00000000h]

RSTCTL is shown in Figure 28-46 and described in Table 28-33.

Return to the Summary Table.

Register to control reset assertion and de-assertion

Figure 28-46. 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 28-33. RSTCTL Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	Unlock key 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 KEY must be set to B1h to write to this bit. 0h = Writing 0 has no effect 1h = Clear reset sticky bit
0	RESETASSERT	WK	0h	Assert reset to the peripheral KEY must be set to B1h to write to this bit. 0h = Writing 0 has no effect 1h = Assert reset

28.3.7 STAT (Offset = 814h) [Reset = 00000000h]

STAT is shown in Figure 28-47 and described in Table 28-34.

Return to the Summary Table.

peripheral enable and reset status

Figure 28-47. 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 28-34. 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 0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register 1h = The peripheral was reset since the last bit clear
15-0	RESERVED	R	0h	

28.3.8 CLKDIV (Offset = 1000h) [Reset = 00000000h]

CLKDIV is shown in Figure 28-48 and described in Table 28-35.

Return to the Summary Table.

This register is used to specify module-specific divide ratio of the functional clock

Figure 28-48. 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 28-35. 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 0h = Do not divide clock source 1h = Divide clock source by 2 2h = Divide clock source by 3 3h = Divide clock source by 4 4h = Divide clock source by 5 5h = Divide clock source by 6 6h = Divide clock source by 7 7h = Divide clock source by 8

28.3.9 CLKSEL (Offset = 1008h) [Reset = 00000000h]

CLKSEL is shown in Figure 28-49 and described in Table 28-36.

Return to the Summary Table.

Clock Source Select Register

Figure 28-49. 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			RESERVED	BUSCLK_SEL	MFCLK_SEL	LFCLK_SEL	RESERVED
R-0h			R-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

Table 28-36. CLKSEL Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	
4	RESERVED	R	0h	Reserved
3	BUSCLK_SEL	R/W	0h	Selects BUSCLK as clock source if enabled 0h = Does not select this clock as a source 1h = Select this clock as a source
2	MFCLK_SEL	R/W	0h	Selects MFCLK as clock source if enabled 0h = Does not select this clock as a source 1h = Select this clock as a source
1	LFCLK_SEL	R/W	0h	Selects LFCLK as clock source if enabled 0h = Does not select this clock as a source 1h = Select this clock as a source
0	RESERVED	R	0h	

28.3.10 PDBGCTL (Offset = 1018h) [Reset = 00000000h]

PDBGCTL is shown in Figure 28-50 and described in Table 28-37.

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 28-50. 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 28-37. PDBGCTL Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1	SOFT	R/W	0h	Soft halt boundary control. This function is only available, if FREE is set to 'STOP' 0h = The peripheral will halt immediately, even if the resultant state will result in corruption if the system is restarted 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 0h = The peripheral freezes functionality while the Core Halted input is asserted and resumes when it is deasserted. 1h = The peripheral ignores the state of the Core Halted input

28.3.11 IIDX (Offset = 1020h) [Reset = 00000000h]

IIDX is shown in Figure 28-51 and described in Table 28-38.

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 28-51. 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 28-38. IIDX Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	STAT	R	0h	Interrupt index status 00h = No interrupt pending 01h = Interrupt Source: Zero event (Z) 02h = Interrupt Source: Load event (L) 05h = Interrupt Source: Capture or compare down event (CCD0) 06h = Interrupt Source: Capture or compare down event (CCD1) 07h = Interrupt Source: Capture or compare down event (CCD2) 08h = Interrupt Source: Capture or compare down event (CCD3) 09h = Interrupt Source: Capture or compare up event (CCU0) 0Ah = Interrupt Source: Capture or compare up event (CCU1) 0Bh = Interrupt Source: Capture or compare up event (CCU2) 0Ch = Interrupt Source: Capture or compare up event (CCU3) 0Dh = Interrupt Source: Compare down event (CCD4) 0Eh = Interrupt Source: Compare down event (CCD5) 0Fh = Interrupt Source: Compare down event (CCU4) 10h = Interrupt Source: Compare down event (CCU5) 19h = Interrupt Source: Fault Event generated an interrupt. (F) 1Ah = Interrupt Source: Trigger overflow (TOV) 1Bh = Interrupt Source: Repeat Counter Zero (REPC) 1Ch = Interrupt Source: Direction Change (DC) 1Dh = Interrupt Source:QEI Incorrect state transition error (QEIERR)

28.3.12 IMASK (Offset = 1028h) [Reset = 00000000h]

IMASK is shown in Figure 28-52 and described in Table 28-39.

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 28-52. 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 28-39. IMASK Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28	QEIERR	R/W	0h	QEIERR Event mask 0h = Disable Event 1h = Enable Event
27	DC	R/W	0h	Direction Change Event mask 0h = Disable Event 1h = Enable Event
26	REPC	R/W	0h	Repeat Counter Zero Event mask 0h = Disable Event 1h = Enable Event
25	TOV	R/W	0h	Trigger Overflow Event mask 0h = Disable Event 1h = Enable Event
24	F	R/W	0h	Fault Event mask 0h = Disable Event 1h = Enable Event
23-16	RESERVED	R	0h	
15	CCU5	R/W	0h	Compare UP event mask CCP5 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
14	CCU4	R/W	0h	Compare UP event mask CCP4 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
13	CCD5	R/W	0h	Compare DN event mask CCP5 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
12	CCD4	R/W	0h	Compare DN event mask CCP4 0h = Clear Interrupt Mask 1h = Set Interrupt Mask

Table 28-39. IMASK Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	CCU3	R/W	0h	Capture or Compare UP event mask CCP3 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
10	CCU2	R/W	0h	Capture or Compare UP event mask CCP2 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
9	CCU1	R/W	0h	Capture or Compare UP event mask CCP1 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
8	CCU0	R/W	0h	Capture or Compare UP event mask CCP0 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
7	CCD3	R/W	0h	Capture or Compare DN event mask CCP3 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
6	CCD2	R/W	0h	Capture or Compare DN event mask CCP2 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
5	CCD1	R/W	0h	Capture or Compare DN event mask CCP1 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
4	CCD0	R/W	0h	Capture or Compare DN event mask CCP0 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
3-2	RESERVED	R	0h	
1	L	R/W	0h	Load Event mask 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
0	Z	R/W	0h	Zero Event mask 0h = Disable Event 1h = Enable Event

28.3.13 RIS (Offset = 1030h) [Reset = 00000000h]

RIS is shown in Figure 28-53 and described in Table 28-40.

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 28-53. 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 28-40. 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. 0h = Event Cleared 1h = Event Set
27	DC	R	0h	Direction Change 0h = Event Cleared 1h = Event Set
26	REPC	R	0h	Repeat Counter Zero 0h = Event Cleared 1h = Event Set
25	TOV	R	0h	Trigger overflow 0h = Event Cleared 1h = Event Set
24	F	R	0h	Fault 0h = Event Cleared 1h = Event Set
23-16	RESERVED	R	0h	
15	CCU5	R	0h	Compare up event generated an interrupt CCP5 0h = Event Cleared 1h = Event Set
14	CCU4	R	0h	Compare up event generated an interrupt CCU4 0h = Event Cleared 1h = Event Set
13	CCD5	R	0h	Compare down event generated an interrupt CCD5 0h = Event Cleared 1h = Event Set

Table 28-40. RIS Field Descriptions (continued)

Bit	Field	Type	Reset	Description
12	CCD4	R	0h	Compare down event generated an interrupt CCD4 0h = Event Cleared 1h = Event Set
11	CCU3	R	0h	Capture or compare up event generated an interrupt CCP3 0h = Event Cleared 1h = Event Set
10	CCU2	R	0h	Capture or compare up event generated an interrupt CCP2 0h = Event Cleared 1h = Event Set
9	CCU1	R	0h	Capture or compare up event generated an interrupt CCP1 0h = Event Cleared 1h = Event Set
8	CCU0	R	0h	Capture or compare up event generated an interrupt CCP0 0h = Event Cleared 1h = Event Set
7	CCD3	R	0h	Capture or compare down event generated an interrupt CCP3 0h = Event Cleared 1h = Event Set
6	CCD2	R	0h	Capture or compare down event generated an interrupt CCP2 0h = Event Cleared 1h = Event Set
5	CCD1	R	0h	Capture or compare down event generated an interrupt CCP1 0h = Event Cleared 1h = Event Set
4	CCD0	R	0h	Capture or compare down event generated an interrupt CCP0 0h = Event Cleared 1h = Event Set
3-2	RESERVED	R	0h	
1	L	R	0h	Load event generated an interrupt. 0h = Event Cleared 1h = Event Set
0	Z	R	0h	Zero event generated an interrupt. 0h = Event Cleared 1h = Event Set

28.3.14 MIS (Offset = 1038h) [Reset = 00000000h]

MIS is shown in Figure 28-54 and described in Table 28-41.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Figure 28-54. 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 28-41. MIS Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28	QEIERR	R	0h	QEIERR 0h = Event Cleared 1h = Event Set
27	DC	R	0h	Direction Change 0h = Event Cleared 1h = Event Set
26	REPC	R	0h	Repeat Counter Zero 0h = Event Cleared 1h = Event Set
25	TOV	R	0h	Trigger overflow 0h = Event Cleared 1h = Event Set
24	F	R	0h	Fault 0h = Event Cleared 1h = Event Set
23-16	RESERVED	R	0h	
15	CCU5	R	0h	Compare up event generated an interrupt CCP5 0h = Event Cleared 1h = Event Set
14	CCU4	R	0h	Compare up event generated an interrupt CCP4 0h = Event Cleared 1h = Event Set
13	CCD5	R	0h	Compare down event generated an interrupt CCP5 0h = Event Cleared 1h = Event Set
12	CCD4	R	0h	Compare down event generated an interrupt CCP4 0h = Event Cleared 1h = Event Set

Table 28-41. MIS Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	CCU3	R	0h	Capture or compare up event generated an interrupt CCP3 0h = Event Cleared 1h = Event Set
10	CCU2	R	0h	Capture or compare up event generated an interrupt CCP2 0h = Event Cleared 1h = Event Set
9	CCU1	R	0h	Capture or compare up event generated an interrupt CCP1 0h = Event Cleared 1h = Event Set
8	CCU0	R	0h	Capture or compare up event generated an interrupt CCP0 0h = Event Cleared 1h = Event Set
7	CCD3	R	0h	Capture or compare down event generated an interrupt CCP3 0h = Event Cleared 1h = Event Set
6	CCD2	R	0h	Capture or compare down event generated an interrupt CCP2 0h = Event Cleared 1h = Event Set
5	CCD1	R	0h	Capture or compare down event generated an interrupt CCP1 0h = Event Cleared 1h = Event Set
4	CCD0	R	0h	Capture or compare down event generated an interrupt CCP0 0h = Event Cleared 1h = Event Set
3-2	RESERVED	R	0h	
1	L	R	0h	Load event generated an interrupt. 0h = Event Cleared 1h = Event Set
0	Z	R	0h	Zero event generated an interrupt. 0h = Event Cleared 1h = Event Set

28.3.15 ISET (Offset = 1040h) [Reset = 00000000h]

ISET is shown in Figure 28-55 and described in Table 28-42.

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 28-55. 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 28-42. ISET Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28	QEIERR	W	0h	QEIERR event SET 0h = Writing 0 has no effect. 1h = Event Set
27	DC	W	0h	Direction Change event SET 0h = Writing 0 has no effect. 1h = Event Set
26	REPC	W	0h	Repeat Counter Zero event SET 0h = Writing 0 has no effect. 1h = Event Set
25	TOV	W	0h	Trigger Overflow event SET 0h = Writing 0 has no effect. 1h = Event Set
24	F	W	0h	Fault event SET 0h = Writing 0 has no effect. 1h = Event Set
23-16	RESERVED	R	0h	
15	CCU5	W	0h	Compare up event 5 SET 0h = Writing 0 has no effect. 1h = Event Set
14	CCU4	W	0h	Compare up event 4 SET 0h = Writing 0 has no effect. 1h = Event Set
13	CCD5	W	0h	Compare down event 5 SET 0h = Writing 0 has no effect. 1h = Event Set
12	CCD4	W	0h	Compare down event 4 SET 0h = Writing 0 has no effect. 1h = Event Set

Table 28-42. ISET Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	CCU3	W	0h	Capture or compare up event SET 0h = Writing 0 has no effect. 1h = Event Set
10	CCU2	W	0h	Capture or compare up event SET 0h = Writing 0 has no effect. 1h = Event Set
9	CCU1	W	0h	Capture or compare up event SET 0h = Writing 0 has no effect. 1h = Event Set
8	CCU0	W	0h	Capture or compare up event SET 0h = Writing 0 has no effect. 1h = Event Set
7	CCD3	W	0h	Capture or compare down event SET 0h = Writing 0 has no effect. 1h = Event Set
6	CCD2	W	0h	Capture or compare down event SET 0h = Writing 0 has no effect. 1h = Event Set
5	CCD1	W	0h	Capture or compare down event SET 0h = Writing 0 has no effect. 1h = Event Set
4	CCD0	W	0h	Capture or compare down event SET 0h = Writing 0 has no effect. 1h = Event Set
3-2	RESERVED	R	0h	
1	L	W	0h	Load event SET 0h = Writing 0 has no effect. 1h = Event Set
0	Z	W	0h	Zero event SET 0h = Writing 0 has no effect. 1h = Event Set

28.3.16 ICLR (Offset = 1048h) [Reset = 00000000h]

ICLR is shown in Figure 28-56 and described in Table 28-43.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

Figure 28-56. 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 28-43. ICLR Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28	QEIERR	W	0h	QEIERR event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
27	DC	W	0h	Direction Change event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
26	REPC	W	0h	Repeat Counter Zero event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
25	TOV	W	0h	Trigger Overflow event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
24	F	W	0h	Fault event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
23-16	RESERVED	R	0h	
15	CCU5	W	0h	Compare up event 5 CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
14	CCU4	W	0h	Compare up event 4 CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
13	CCD5	W	0h	Compare down event 5 CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
12	CCD4	W	0h	Compare down event 4 CLEAR 0h = Writing 0 has no effect. 1h = Event Clear

Table 28-43. ICLR Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	CCU3	W	0h	Capture or compare up event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
10	CCU2	W	0h	Capture or compare up event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
9	CCU1	W	0h	Capture or compare up event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
8	CCU0	W	0h	Capture or compare up event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
7	CCD3	W	0h	Capture or compare down event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
6	CCD2	W	0h	Capture or compare down event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
5	CCD1	W	0h	Capture or compare down event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
4	CCD0	W	0h	Capture or compare down event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
3-2	RESERVED	R	0h	
1	L	W	0h	Load event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
0	Z	W	0h	Zero event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear

28.3.17 IIDX (Offset = 1050h) [Reset = 00000000h]

IIDX is shown in Figure 28-57 and described in Table 28-44.

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 28-57. 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 28-44. IIDX Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	STAT	R	0h	Interrupt index status 00h = No interrupt pending 01h = Interrupt Source: Zero event (Z) 02h = Interrupt Source: Load event (L) 05h = Interrupt Source: Capture or compare down event (CCD0) 06h = Interrupt Source: Capture or compare down event (CCD1) 07h = Interrupt Source: Capture or compare down event (CCD2) 08h = Interrupt Source: Capture or compare down event (CCD3) 09h = Interrupt Source: Capture or compare up event (CCU0) 0Ah = Interrupt Source: Capture or compare up event (CCU1) 0Bh = Interrupt Source: Capture or compare up event (CCU2) 0Ch = Interrupt Source: Capture or compare up event (CCU3) 0Dh = Interrupt Source: Compare down event (CCD4) 0Eh = Interrupt Source: Compare down event (CCD5) 0Fh = Interrupt Source: Compare down event (CCU4) 10h = Interrupt Source: Compare down event (CCU5) 19h = Interrupt Source: Fault Event generated an interrupt. (F) 1Ah = Interrupt Source: Trigger overflow (TOV) 1Bh = Interrupt Source: Repeat Counter Zero (REPC) 1Ch = Interrupt Source: Direction Change (DC) 1Dh = Interrupt Source:QEI Incorrect state transition error (QEIERR)

28.3.18 IMASK (Offset = 1058h) [Reset = 00000000h]

IMASK is shown in Figure 28-58 and described in Table 28-45.

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 28-58. 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 28-45. IMASK Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28	QEIERR	R/W	0h	QEIERR Event mask 0h = Disable Event 1h = Enable Event
27	DC	R/W	0h	Direction Change Event mask 0h = Disable Event 1h = Enable Event
26	REPC	R/W	0h	Repeat Counter Zero Event mask 0h = Disable Event 1h = Enable Event
25	TOV	R/W	0h	Trigger Overflow Event mask 0h = Disable Event 1h = Enable Event
24	F	R/W	0h	Fault Event mask 0h = Disable Event 1h = Enable Event
23-16	RESERVED	R	0h	
15	CCU5	R/W	0h	Compare UP event mask CCP5 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
14	CCU4	R/W	0h	Compare UP event mask CCP4 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
13	CCD5	R/W	0h	Compare DN event mask CCP5 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
12	CCD4	R/W	0h	Compare DN event mask CCP4 0h = Clear Interrupt Mask 1h = Set Interrupt Mask

Table 28-45. IMASK Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	CCU3	R/W	0h	Capture or Compare UP event mask CCP3 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
10	CCU2	R/W	0h	Capture or Compare UP event mask CCP2 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
9	CCU1	R/W	0h	Capture or Compare UP event mask CCP1 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
8	CCU0	R/W	0h	Capture or Compare UP event mask CCP0 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
7	CCD3	R/W	0h	Capture or Compare DN event mask CCP3 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
6	CCD2	R/W	0h	Capture or Compare DN event mask CCP2 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
5	CCD1	R/W	0h	Capture or Compare DN event mask CCP1 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
4	CCD0	R/W	0h	Capture or Compare DN event mask CCP0 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
3-2	RESERVED	R	0h	
1	L	R/W	0h	Load Event mask 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
0	Z	R/W	0h	Zero Event mask 0h = Disable Event 1h = Enable Event

28.3.19 RIS (Offset = 1060h) [Reset = 00000000h]

RIS is shown in Figure 28-59 and described in Table 28-46.

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 28-59. 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 28-46. 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. 0h = Event Cleared 1h = Event Set
27	DC	R	0h	Direction Change 0h = Event Cleared 1h = Event Set
26	REPC	R	0h	Repeat Counter Zero 0h = Event Cleared 1h = Event Set
25	TOV	R	0h	Trigger overflow 0h = Event Cleared 1h = Event Set
24	F	R	0h	Fault 0h = Event Cleared 1h = Event Set
23-16	RESERVED	R	0h	
15	CCU5	R	0h	Compare up event generated an interrupt CCP5 0h = Event Cleared 1h = Event Set
14	CCU4	R	0h	Compare up event generated an interrupt CCU4 0h = Event Cleared 1h = Event Set
13	CCD5	R	0h	Compare down event generated an interrupt CCD5 0h = Event Cleared 1h = Event Set

Table 28-46. RIS Field Descriptions (continued)

Bit	Field	Type	Reset	Description
12	CCD4	R	0h	Compare down event generated an interrupt CCD4 0h = Event Cleared 1h = Event Set
11	CCU3	R	0h	Capture or compare up event generated an interrupt CCP3 0h = Event Cleared 1h = Event Set
10	CCU2	R	0h	Capture or compare up event generated an interrupt CCP2 0h = Event Cleared 1h = Event Set
9	CCU1	R	0h	Capture or compare up event generated an interrupt CCP1 0h = Event Cleared 1h = Event Set
8	CCU0	R	0h	Capture or compare up event generated an interrupt CCP0 0h = Event Cleared 1h = Event Set
7	CCD3	R	0h	Capture or compare down event generated an interrupt CCP3 0h = Event Cleared 1h = Event Set
6	CCD2	R	0h	Capture or compare down event generated an interrupt CCP2 0h = Event Cleared 1h = Event Set
5	CCD1	R	0h	Capture or compare down event generated an interrupt CCP1 0h = Event Cleared 1h = Event Set
4	CCD0	R	0h	Capture or compare down event generated an interrupt CCP0 0h = Event Cleared 1h = Event Set
3-2	RESERVED	R	0h	
1	L	R	0h	Load event generated an interrupt. 0h = Event Cleared 1h = Event Set
0	Z	R	0h	Zero event generated an interrupt. 0h = Event Cleared 1h = Event Set

28.3.20 MIS (Offset = 1068h) [Reset = 00000000h]

MIS is shown in Figure 28-60 and described in Table 28-47.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Figure 28-60. 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 28-47. MIS Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28	QEIERR	R	0h	QEIERR 0h = Event Cleared 1h = Event Set
27	DC	R	0h	Direction Change 0h = Event Cleared 1h = Event Set
26	REPC	R	0h	Repeat Counter Zero 0h = Event Cleared 1h = Event Set
25	TOV	R	0h	Trigger overflow 0h = Event Cleared 1h = Event Set
24	F	R	0h	Fault 0h = Event Cleared 1h = Event Set
23-16	RESERVED	R	0h	
15	CCU5	R	0h	Compare up event generated an interrupt CCP5 0h = Event Cleared 1h = Event Set
14	CCU4	R	0h	Compare up event generated an interrupt CCP4 0h = Event Cleared 1h = Event Set
13	CCD5	R	0h	Compare down event generated an interrupt CCP5 0h = Event Cleared 1h = Event Set
12	CCD4	R	0h	Compare down event generated an interrupt CCP4 0h = Event Cleared 1h = Event Set

Table 28-47. MIS Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	CCU3	R	0h	Capture or compare up event generated an interrupt CCP3 0h = Event Cleared 1h = Event Set
10	CCU2	R	0h	Capture or compare up event generated an interrupt CCP2 0h = Event Cleared 1h = Event Set
9	CCU1	R	0h	Capture or compare up event generated an interrupt CCP1 0h = Event Cleared 1h = Event Set
8	CCU0	R	0h	Capture or compare up event generated an interrupt CCP0 0h = Event Cleared 1h = Event Set
7	CCD3	R	0h	Capture or compare down event generated an interrupt CCP3 0h = Event Cleared 1h = Event Set
6	CCD2	R	0h	Capture or compare down event generated an interrupt CCP2 0h = Event Cleared 1h = Event Set
5	CCD1	R	0h	Capture or compare down event generated an interrupt CCP1 0h = Event Cleared 1h = Event Set
4	CCD0	R	0h	Capture or compare down event generated an interrupt CCP0 0h = Event Cleared 1h = Event Set
3-2	RESERVED	R	0h	
1	L	R	0h	Load event generated an interrupt. 0h = Event Cleared 1h = Event Set
0	Z	R	0h	Zero event generated an interrupt. 0h = Event Cleared 1h = Event Set

28.3.21 ISET (Offset = 1070h) [Reset = 00000000h]

ISET is shown in Figure 28-61 and described in Table 28-48.

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 28-61. 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 28-48. ISET Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28	QEIERR	W	0h	QEIERR event SET 0h = Writing 0 has no effect. 1h = Event Set
27	DC	W	0h	Direction Change event SET 0h = Writing 0 has no effect. 1h = Event Set
26	REPC	W	0h	Repeat Counter Zero event SET 0h = Writing 0 has no effect. 1h = Event Set
25	TOV	W	0h	Trigger Overflow event SET 0h = Writing 0 has no effect. 1h = Event Set
24	F	W	0h	Fault event SET 0h = Writing 0 has no effect. 1h = Event Set
23-16	RESERVED	R	0h	
15	CCU5	W	0h	Compare up event 5 SET 0h = Writing 0 has no effect. 1h = Event Set
14	CCU4	W	0h	Compare up event 4 SET 0h = Writing 0 has no effect. 1h = Event Set
13	CCD5	W	0h	Compare down event 5 SET 0h = Writing 0 has no effect. 1h = Event Set
12	CCD4	W	0h	Compare down event 4 SET 0h = Writing 0 has no effect. 1h = Event Set

Table 28-48. ISET Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	CCU3	W	0h	Capture or compare up event SET 0h = Writing 0 has no effect. 1h = Event Set
10	CCU2	W	0h	Capture or compare up event SET 0h = Writing 0 has no effect. 1h = Event Set
9	CCU1	W	0h	Capture or compare up event SET 0h = Writing 0 has no effect. 1h = Event Set
8	CCU0	W	0h	Capture or compare up event SET 0h = Writing 0 has no effect. 1h = Event Set
7	CCD3	W	0h	Capture or compare down event SET 0h = Writing 0 has no effect. 1h = Event Set
6	CCD2	W	0h	Capture or compare down event SET 0h = Writing 0 has no effect. 1h = Event Set
5	CCD1	W	0h	Capture or compare down event SET 0h = Writing 0 has no effect. 1h = Event Set
4	CCD0	W	0h	Capture or compare down event SET 0h = Writing 0 has no effect. 1h = Event Set
3-2	RESERVED	R	0h	
1	L	W	0h	Load event SET 0h = Writing 0 has no effect. 1h = Event Set
0	Z	W	0h	Zero event SET 0h = Writing 0 has no effect. 1h = Event Set

28.3.22 ICLR (Offset = 1078h) [Reset = 00000000h]

ICLR is shown in Figure 28-62 and described in Table 28-49.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

Figure 28-62. 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 28-49. ICLR Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28	QEIERR	W	0h	QEIERR event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
27	DC	W	0h	Direction Change event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
26	REPC	W	0h	Repeat Counter Zero event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
25	TOV	W	0h	Trigger Overflow event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
24	F	W	0h	Fault event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
23-16	RESERVED	R	0h	
15	CCU5	W	0h	Compare up event 5 CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
14	CCU4	W	0h	Compare up event 4 CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
13	CCD5	W	0h	Compare down event 5 CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
12	CCD4	W	0h	Compare down event 4 CLEAR 0h = Writing 0 has no effect. 1h = Event Clear

Table 28-49. ICLR Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	CCU3	W	0h	Capture or compare up event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
10	CCU2	W	0h	Capture or compare up event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
9	CCU1	W	0h	Capture or compare up event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
8	CCU0	W	0h	Capture or compare up event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
7	CCD3	W	0h	Capture or compare down event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
6	CCD2	W	0h	Capture or compare down event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
5	CCD1	W	0h	Capture or compare down event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
4	CCD0	W	0h	Capture or compare down event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
3-2	RESERVED	R	0h	
1	L	W	0h	Load event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
0	Z	W	0h	Zero event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear

28.3.23 IIDX (Offset = 1080h) [Reset = 00000000h]

IIDX is shown in Figure 28-63 and described in Table 28-50.

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 28-63. 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 28-50. IIDX Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	STAT	R	0h	Interrupt index status 00h = No interrupt pending 01h = Interrupt Source: Zero event (Z) 02h = Interrupt Source: Load event (L) 05h = Interrupt Source: Capture or compare down event (CCD0) 06h = Interrupt Source: Capture or compare down event (CCD1) 07h = Interrupt Source: Capture or compare down event (CCD2) 08h = Interrupt Source: Capture or compare down event (CCD3) 09h = Interrupt Source: Capture or compare up event (CCU0) 0Ah = Interrupt Source: Capture or compare up event (CCU1) 0Bh = Interrupt Source: Capture or compare up event (CCU2) 0Ch = Interrupt Source: Capture or compare up event (CCU3) 0Dh = Interrupt Source: Compare down event (CCD4) 0Eh = Interrupt Source: Compare down event (CCD5) 0Fh = Interrupt Source: Compare down event (CCU4) 10h = Interrupt Source: Compare down event (CCU5) 19h = Interrupt Source: Fault Event generated an interrupt. (F) 1Ah = Interrupt Source: Trigger overflow (TOV) 1Bh = Interrupt Source: Repeat Counter Zero (REPC) 1Ch = Interrupt Source: Direction Change (DC) 1Dh = Interrupt Source:QEI Incorrect state transition error (QEIERR)

28.3.24 IMASK (Offset = 1088h) [Reset = 00000000h]

IMASK is shown in Figure 28-64 and described in Table 28-51.

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 28-64. 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 28-51. IMASK Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28	QEIERR	R/W	0h	QEIERR Event mask 0h = Disable Event 1h = Enable Event
27	DC	R/W	0h	Direction Change Event mask 0h = Disable Event 1h = Enable Event
26	REPC	R/W	0h	Repeat Counter Zero Event mask 0h = Disable Event 1h = Enable Event
25	TOV	R/W	0h	Trigger Overflow Event mask 0h = Disable Event 1h = Enable Event
24	F	R/W	0h	Fault Event mask 0h = Disable Event 1h = Enable Event
23-16	RESERVED	R	0h	
15	CCU5	R/W	0h	Compare UP event mask CCP5 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
14	CCU4	R/W	0h	Compare UP event mask CCP4 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
13	CCD5	R/W	0h	Compare DN event mask CCP5 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
12	CCD4	R/W	0h	Compare DN event mask CCP4 0h = Clear Interrupt Mask 1h = Set Interrupt Mask

Table 28-51. IMASK Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	CCU3	R/W	0h	Capture or Compare UP event mask CCP3 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
10	CCU2	R/W	0h	Capture or Compare UP event mask CCP2 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
9	CCU1	R/W	0h	Capture or Compare UP event mask CCP1 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
8	CCU0	R/W	0h	Capture or Compare UP event mask CCP0 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
7	CCD3	R/W	0h	Capture or Compare DN event mask CCP3 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
6	CCD2	R/W	0h	Capture or Compare DN event mask CCP2 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
5	CCD1	R/W	0h	Capture or Compare DN event mask CCP1 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
4	CCD0	R/W	0h	Capture or Compare DN event mask CCP0 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
3-2	RESERVED	R	0h	
1	L	R/W	0h	Load Event mask 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
0	Z	R/W	0h	Zero Event mask 0h = Disable Event 1h = Enable Event

28.3.25 RIS (Offset = 1090h) [Reset = 00000000h]

RIS is shown in Figure 28-65 and described in Table 28-52.

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 28-65. 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 28-52. 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. 0h = Event Cleared 1h = Event Set
27	DC	R	0h	Direction Change 0h = Event Cleared 1h = Event Set
26	REPC	R	0h	Repeat Counter Zero 0h = Event Cleared 1h = Event Set
25	TOV	R	0h	Trigger overflow 0h = Event Cleared 1h = Event Set
24	F	R	0h	Fault 0h = Event Cleared 1h = Event Set
23-16	RESERVED	R	0h	
15	CCU5	R	0h	Compare up event generated an interrupt CCP5 0h = Event Cleared 1h = Event Set
14	CCU4	R	0h	Compare up event generated an interrupt CCU4 0h = Event Cleared 1h = Event Set
13	CCD5	R	0h	Compare down event generated an interrupt CCD5 0h = Event Cleared 1h = Event Set

Table 28-52. RIS Field Descriptions (continued)

Bit	Field	Type	Reset	Description
12	CCD4	R	0h	Compare down event generated an interrupt CCD4 0h = Event Cleared 1h = Event Set
11	CCU3	R	0h	Capture or compare up event generated an interrupt CCP3 0h = Event Cleared 1h = Event Set
10	CCU2	R	0h	Capture or compare up event generated an interrupt CCP2 0h = Event Cleared 1h = Event Set
9	CCU1	R	0h	Capture or compare up event generated an interrupt CCP1 0h = Event Cleared 1h = Event Set
8	CCU0	R	0h	Capture or compare up event generated an interrupt CCP0 0h = Event Cleared 1h = Event Set
7	CCD3	R	0h	Capture or compare down event generated an interrupt CCP3 0h = Event Cleared 1h = Event Set
6	CCD2	R	0h	Capture or compare down event generated an interrupt CCP2 0h = Event Cleared 1h = Event Set
5	CCD1	R	0h	Capture or compare down event generated an interrupt CCP1 0h = Event Cleared 1h = Event Set
4	CCD0	R	0h	Capture or compare down event generated an interrupt CCP0 0h = Event Cleared 1h = Event Set
3-2	RESERVED	R	0h	
1	L	R	0h	Load event generated an interrupt. 0h = Event Cleared 1h = Event Set
0	Z	R	0h	Zero event generated an interrupt. 0h = Event Cleared 1h = Event Set

28.3.26 MIS (Offset = 1098h) [Reset = 00000000h]

MIS is shown in Figure 28-66 and described in Table 28-53.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Figure 28-66. 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 28-53. MIS Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28	QEIERR	R	0h	QEIERR 0h = Event Cleared 1h = Event Set
27	DC	R	0h	Direction Change 0h = Event Cleared 1h = Event Set
26	REPC	R	0h	Repeat Counter Zero 0h = Event Cleared 1h = Event Set
25	TOV	R	0h	Trigger overflow 0h = Event Cleared 1h = Event Set
24	F	R	0h	Fault 0h = Event Cleared 1h = Event Set
23-16	RESERVED	R	0h	
15	CCU5	R	0h	Compare up event generated an interrupt CCP5 0h = Event Cleared 1h = Event Set
14	CCU4	R	0h	Compare up event generated an interrupt CCP4 0h = Event Cleared 1h = Event Set
13	CCD5	R	0h	Compare down event generated an interrupt CCP5 0h = Event Cleared 1h = Event Set
12	CCD4	R	0h	Compare down event generated an interrupt CCP4 0h = Event Cleared 1h = Event Set

Table 28-53. MIS Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	CCU3	R	0h	Capture or compare up event generated an interrupt CCP3 0h = Event Cleared 1h = Event Set
10	CCU2	R	0h	Capture or compare up event generated an interrupt CCP2 0h = Event Cleared 1h = Event Set
9	CCU1	R	0h	Capture or compare up event generated an interrupt CCP1 0h = Event Cleared 1h = Event Set
8	CCU0	R	0h	Capture or compare up event generated an interrupt CCP0 0h = Event Cleared 1h = Event Set
7	CCD3	R	0h	Capture or compare down event generated an interrupt CCP3 0h = Event Cleared 1h = Event Set
6	CCD2	R	0h	Capture or compare down event generated an interrupt CCP2 0h = Event Cleared 1h = Event Set
5	CCD1	R	0h	Capture or compare down event generated an interrupt CCP1 0h = Event Cleared 1h = Event Set
4	CCD0	R	0h	Capture or compare down event generated an interrupt CCP0 0h = Event Cleared 1h = Event Set
3-2	RESERVED	R	0h	
1	L	R	0h	Load event generated an interrupt. 0h = Event Cleared 1h = Event Set
0	Z	R	0h	Zero event generated an interrupt. 0h = Event Cleared 1h = Event Set

28.3.27 ISET (Offset = 10A0h) [Reset = 00000000h]

ISET is shown in Figure 28-67 and described in Table 28-54.

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 28-67. 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 28-54. ISET Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28	QEIERR	W	0h	QEIERR event SET 0h = Writing 0 has no effect. 1h = Event Set
27	DC	W	0h	Direction Change event SET 0h = Writing 0 has no effect. 1h = Event Set
26	REPC	W	0h	Repeat Counter Zero event SET 0h = Writing 0 has no effect. 1h = Event Set
25	TOV	W	0h	Trigger Overflow event SET 0h = Writing 0 has no effect. 1h = Event Set
24	F	W	0h	Fault event SET 0h = Writing 0 has no effect. 1h = Event Set
23-16	RESERVED	R	0h	
15	CCU5	W	0h	Compare up event 5 SET 0h = Writing 0 has no effect. 1h = Event Set
14	CCU4	W	0h	Compare up event 4 SET 0h = Writing 0 has no effect. 1h = Event Set
13	CCD5	W	0h	Compare down event 5 SET 0h = Writing 0 has no effect. 1h = Event Set
12	CCD4	W	0h	Compare down event 4 SET 0h = Writing 0 has no effect. 1h = Event Set

Table 28-54. ISET Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	CCU3	W	0h	Capture or compare up event SET 0h = Writing 0 has no effect. 1h = Event Set
10	CCU2	W	0h	Capture or compare up event SET 0h = Writing 0 has no effect. 1h = Event Set
9	CCU1	W	0h	Capture or compare up event SET 0h = Writing 0 has no effect. 1h = Event Set
8	CCU0	W	0h	Capture or compare up event SET 0h = Writing 0 has no effect. 1h = Event Set
7	CCD3	W	0h	Capture or compare down event SET 0h = Writing 0 has no effect. 1h = Event Set
6	CCD2	W	0h	Capture or compare down event SET 0h = Writing 0 has no effect. 1h = Event Set
5	CCD1	W	0h	Capture or compare down event SET 0h = Writing 0 has no effect. 1h = Event Set
4	CCD0	W	0h	Capture or compare down event SET 0h = Writing 0 has no effect. 1h = Event Set
3-2	RESERVED	R	0h	
1	L	W	0h	Load event SET 0h = Writing 0 has no effect. 1h = Event Set
0	Z	W	0h	Zero event SET 0h = Writing 0 has no effect. 1h = Event Set

28.3.28 ICLR (Offset = 10A8h) [Reset = 00000000h]

ICLR is shown in Figure 28-68 and described in Table 28-55.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

Figure 28-68. 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 28-55. ICLR Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28	QEIERR	W	0h	QEIERR event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
27	DC	W	0h	Direction Change event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
26	REPC	W	0h	Repeat Counter Zero event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
25	TOV	W	0h	Trigger Overflow event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
24	F	W	0h	Fault event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
23-16	RESERVED	R	0h	
15	CCU5	W	0h	Compare up event 5 CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
14	CCU4	W	0h	Compare up event 4 CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
13	CCD5	W	0h	Compare down event 5 CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
12	CCD4	W	0h	Compare down event 4 CLEAR 0h = Writing 0 has no effect. 1h = Event Clear

Table 28-55. ICLR Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	CCU3	W	0h	Capture or compare up event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
10	CCU2	W	0h	Capture or compare up event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
9	CCU1	W	0h	Capture or compare up event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
8	CCU0	W	0h	Capture or compare up event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
7	CCD3	W	0h	Capture or compare down event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
6	CCD2	W	0h	Capture or compare down event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
5	CCD1	W	0h	Capture or compare down event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
4	CCD0	W	0h	Capture or compare down event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
3-2	RESERVED	R	0h	
1	L	W	0h	Load event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
0	Z	W	0h	Zero event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear

28.3.29 EVT_MODE (Offset = 10E0h) [Reset = 00000029h]

EVT_MODE is shown in Figure 28-69 and described in Table 28-56.

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 28-69. 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 28-56. 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 0h = The interrupt or event line is disabled. 1h = The interrupt or event line is in software mode. Software must clear the RIS. 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 0h = The interrupt or event line is disabled. 1h = The interrupt or event line is in software mode. Software must clear the RIS. 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 0h = The interrupt or event line is disabled. 1h = The interrupt or event line is in software mode. Software must clear the RIS. 2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag.

28.3.30 DESC (Offset = 10FCh) [Reset = 00000000h]

DESC is shown in Figure 28-70 and described in Table 28-57.

Return to the Summary Table.

This register identifies the peripheral and its exact version.

Figure 28-70. 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 28-57. DESC Field Descriptions

Bit	Field	Type	Reset	Description
31-16	MODULEID	R	0h	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. 0h = Smallest value FFFFh = Highest possible value
15-12	FEATUREVER	R	0h	Feature Set for the module *instance* 0h = Smallest value 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 0h = Smallest value Fh = Highest possible value
7-4	MAJREV	R	0h	Major rev of the IP 0h = Smallest value Fh = Highest possible value
3-0	MINREV	R	0h	Minor rev of the IP 0h = Smallest value Fh = Highest possible value

28.3.31 CCPD (Offset = 1100h) [Reset = 00000000h]

CCPD is shown in Figure 28-71 and described in Table 28-58.

Return to the Summary Table.

CCP Direction. Controls whether CCP is used as an input or an output.

Figure 28-71. 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				RESERVED	C0CCP2	C0CCP1	C0CCP0
R-0h				R-0h	R/W-0h	R/W-0h	R/W-0h

Table 28-58. CCPD Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3	RESERVED	R	0h	Reserved
2	C0CCP2	R/W	0h	CCP2 direction 0h = input 1h = Output
1	C0CCP1	R/W	0h	CCP1 direction 0h = Input 1h = Output
0	C0CCP0	R/W	0h	CCP0 direction 0h = Input 1h = Output

28.3.32 ODIS (Offset = 1104h) [Reset = 00000000h]

ODIS is shown in Figure 28-72 and described in Table 28-59.

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 28-72. 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 28-59. ODIS Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3	C0CCP3	R/W	0h	Counter CCP3 Disable Mask Defines whether CCP3 of Counter n is forced low or not 0h = Output function as selected by the OCTL register CCPO field are provided to occpout[2]. 1h = CCP output occpout[3] is forced low.
2	C0CCP2	R/W	0h	Counter CCP2 Disable Mask Defines whether CCP2 of Counter n is forced low or not 0h = Output function as selected by the OCTL register CCPO field are provided to output inversion block. 1h = CCP output occpout[2] is forced low.
1	C0CCP1	R/W	0h	Counter CCP1 Disable Mask Defines whether CCP0 of Counter n is forced low or not 0h = Output function as selected by the OCTL register CCPO field are provided to output inversion block. 1h = CCP output occpout[1] is forced low.
0	C0CCP0	R/W	0h	Counter CCP0 Disable Mask Defines whether CCP0 of Counter n is forced low or not 0h = Output function as selected by the OCTL register CCPO field are provided to output inversion block. 1h = CCP output occpout[0] is forced low.

28.3.33 CCLKCTL (Offset = 1108h) [Reset = 00000000h]

CCLKCTL is shown in Figure 28-73 and described in Table 28-60.

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 28-73. 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 28-60. CCLKCTL Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	CLKEN	R/W	0h	Clock Enable Disables the clock gating to the module. SW has to explicitly program the value to 0 to gate the clock. 0h = Clock is disabled. 1h = Clock is enabled

28.3.34 CPS (Offset = 110Ch) [Reset = 00000000h]

CPS is shown in Figure 28-74 and described in Table 28-61.

Return to the Summary Table.

The CPS register provides the value for the clock pre-scaler.

Figure 28-74. 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 28-61. CPS Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	PCNT	R/W	0h	Pre-Scale Count This field specifies the pre-scale count value. 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 generating a slower clock 0h = Minimum value FFh = Maximum Value

28.3.35 CPSV (Offset = 1110h) [Reset = 00000000h]

CPSV is shown in Figure 28-75 and described in Table 28-62.

Return to the Summary Table.

The CPSV register provides the ability to read the current clock prescale count value.

Figure 28-75. 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 28-62. CPSV Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	CPSVAL	R	0h	Current Prescale Count Value 0h = Minimum value FFh = Maximum Value

28.3.36 CTTRIGCTL (Offset = 1114h) [Reset = 00000000h]

CTTRIGCTL is shown in Figure 28-76 and described in Table 28-63.

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 28-76. 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 28-63. 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. 0h = Use FSUB0 as cross trigger source. 1h = Use FSUB1 as cross trigger source. 2h = Use Zero event as cross trigger source. 3h = Use Load event as cross trigger source. 4h = Use CCD0 event as cross trigger source. 5h = Use CCD1 event as cross trigger source. 6h = Use CCD2 event as cross trigger source. 7h = Use CCD3 event as cross trigger source. 8h = Use CCU0 event as cross trigger source. 9h = Use CCU1 event as cross trigger source. Ah = Use CCU2 event as cross trigger source. 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. 0h = Cross trigger generation disabled. 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. These cross triggers are connected to the respective timer trigger in of the other timer IPs in the SOC power domain. The timer cross trigger is essentially the combined logic of the HW and SW conditions controlling EN bit in the CTRCTL register. 0h = Cross trigger generation disabled. 1h = Cross trigger generation enabled

28.3.37 CTTRIG (Offset = 111Ch) [Reset = 00000000h]

CTTRIG is shown in Figure 28-77 and described in Table 28-64.

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 28-77. 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 28-64. CTTRIG Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	TRIG	W	0h	Generate Cross Trigger This bit when programmed will generate a synchronized trigger condition all the cross trigger enabled Timer instances including current timer instance. 0h = Cross trigger generation disabled 1h = Generate Cross trigger pulse

28.3.38 FSCTL (Offset = 1120h) [Reset = 00000000h]

FSCTL is shown in Figure 28-78 and described in Table 28-65.

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 28-78. 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 28-65. 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. 0h = Disable 1h = Enable
5	FEX1EN	R/W	0h	This field controls whether the fault is caused by external fault pin 1. 0h = Disable 1h = Enable
4	FEX0EN	R/W	0h	This field controls whether the fault is caused by external fault pin 0. 0h = Disable 1h = Enable
3	FAC2EN	R/W	0h	This field controls whether the fault is caused by COMP2 output. 0h = Disable 1h = Enable
2	FAC1EN	R/W	0h	This field controls whether the fault is caused by COMP1 output. 0h = Disable 1h = Enable
1	FAC0EN	R/W	0h	This field controls whether the fault signal is caused by COMP0 output. 0h = Disable 1h = Enable
0	FCEN	R/W	0h	This field controls whether the fault is caused by the system clock fault. 0h = Disable 1h = Enable

28.3.39 GCTL (Offset = 1124h) [Reset = 00000000h]

GCTL is shown in Figure 28-79 and described in Table 28-66.

Return to the Summary Table.

Global control register

Figure 28-79. 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 28-66. GCTL Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	SHDWLDEN	R/W	0h	Enables shadow to active load of buffered registers and register fields. 0h = Disable 1h = Enable

28.3.40 CTR (Offset = 1800h) [Reset = 00000000h]

CTR is shown in Figure 28-80 and described in Table 28-67.

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 28-80. 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 28-67. CTR Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	CCTR	R/W	0h	Current Counter value 0h = Minimum value 00FFFFFFh = Maximum Value

28.3.41 CTRCTL (Offset = 1804h) [Reset = 0000FF80h]

CTRCTL is shown in Figure 28-81 and described in Table 28-68.

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 28-81. 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 28-68. 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. 0h = The counter is set to the LOAD register value 1h = The counter value is unchanged from its current value which could have been initialized by software 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. 0h = Disabled 1h = Enabled
23	SLZERCNEZ	R/W	0h	Suppress Load and Zero Events if Repeat Counter is Not Equal to Zero. 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. 0h = Disabled. Z and L events are always generated from the counter when their conditions are generated. 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. 0h = Resume counting 1h = Perform the action as specified by the CVAE field.

Table 28-68. 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 0h = Continues counting 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. 0h = Resume counting 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. 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. 0h = CCCTL_0 ZCOND 1h = CCCTL_1 ZCOND 2h = CCCTL_2 ZCOND This value exists when there are 4 channels. 3h = CCCTL_3 ZCOND This value exists when there are 4 channels. 4h = Controlled by 2-input QEI mode This value exists when TIMER support QEI feature. 5h = Controlled by 3-input QEI mode 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. 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. 0h = CCCTL_0 ACOND 1h = CCCTL_1 ACOND 2h = CCCTL_2 ACOND This value exists when there are 4 channels. 3h = CCCTL_3 ACOND This value exists when there are 4 channels. 4h = Controlled by 2-input QEI mode This value exists when TIMER support QEI feature. 5h = Controlled by 3-input QEI mode 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. 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. 0h = CCCTL_0 LCOND 1h = CCCTL_1 LCOND 2h = CCCTL_2 LCOND This value exists when there are 4 channels. 3h = CCCTL_3 LCOND This value exists when there are 4 channels. 4h = Controlled by 2 input QEI mode. This value exists when TIMER support QEI feature. 5h = Controlled by 3 input QEI mode. This value exists when TIMER support QEI feature.
6	RESERVED	R	0h	

Table 28-68. CTRCTL Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5-4	CM	R/W	0h	Count Mode 0h = Down 1h = Up/Down 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. 0h = Does not automatically advance following a zero event. 1h = Continues to advance following a zero event. 2h = Reserved 3h = Continues to advance following a zero event if the debug mode is not in effect, or following the release of the debug mode. 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. 0h = Disabled 1h = Enabled

28.3.42 LOAD (Offset = 1808h) [Reset = 00000000h]

LOAD is shown in Figure 28-82 and described in Table 28-69.

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 28-82. 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 28-69. LOAD Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	LD	R/W	0h	Load Value 0h = Minimum value 00FFFFFFh = Maximum Value

28.3.43 CC_01[y] (Offset = 1810h + formula) [Reset = 00000000h]

CC_01[y] is shown in Figure 28-83 and described in Table 28-70.

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 28-83. 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 28-70. 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 0h = Minimum value FFFFh = Maximum Value

28.3.44 CC_23[y] (Offset = 1818h + formula) [Reset = 00000000h]

CC_23[y] is shown in Figure 28-84 and described in Table 28-71.

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 28-84. 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 28-71. 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 0h = Minimum value FFFFh = Maximum Value

28.3.45 CC_45[y] (Offset = 1820h + formula) [Reset = 00000000h]

CC_45[y] is shown in Figure 28-85 and described in Table 28-72.

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 28-85. 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 28-72. 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 0h = Minimum value FFFFh = Maximum Value

28.3.46 CCCTL_01[y] (Offset = 1830h + formula) [Reset = 00000000h]

CCCTL_01[y] is shown in Figure 28-86 and described in Table 28-73.

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 28-86. 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 28-73. CCCTL_01[y] Field Descriptions

Bit	Field	Type	Reset	Description
31-29	CC2SELD	R/W	0h	Selects the source second CCD event. 0h = Selects CCD from CC0. 1h = Selects CCD from CC1. 2h = Selects CCD from CC2. 3h = Selects CCD from CC3. 4h = Selects CCD from CC4. 5h = Selects CCD from CC5.

Table 28-73. CCCTL_01[y] Field Descriptions (continued)

Bit	Field	Type	Reset	Description
28-26	CCACTUPD	R/W	0h	CCACT shadow register Update Method This field controls how updates to the CCACT shadow register are performed 0h = Value written to the CCACT register has immediate effect. 1h = Following a zero event (CTR=0) Writes to the CCACTx_y register are stored in shadow register and transferred to CCACTx_y in the TIMCLK cycle following CTR equals 0. 2h = Following a CCD event (CTR=CC_xy) 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. 3h = Following a CCU event (CTR=CC_xy) 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. 4h = Following a zero event (CTR=0) or load event (CTR = LOAD) 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. 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. 5h = Following a zero event (CTR=0) with repeat count also zero (RC=0). 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. 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 This bit suppresses the generation of the compare (CCD, CCU and RC) events from the counter when the repeat counter (RC) value is not 0. 0h = CCD, CCU and RC events are always generated from the counter when their conditions are generated. 1h = CCD, CCU and RC events are generated from the counter when their conditions are generated and the RC register value is 0.
24-22	CC2SELU	R/W	0h	Selects the source second CCU event. 0h = Selects CCU from CC0. 1h = Selects CCU from CC1. 2h = Selects CCU from CC2. 3h = Selects CCU from CC3. 4h = Selects CCU from CC4. 5h = Selects CCU from CC5.

Table 28-73. CCCTL_01[y] Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	RESERVED	R	0h	
20-18	CCUPD	R/W	0h	Capture and Compare Update Method This field controls how updates to the shadow capture and compare register are performed (when operating in compare mode, COC=0). 0h = Writes to the CCx_y register is written to the register directly and has immediate effect. 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. 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. 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. 4h = Following a zero event(CTR=0) or load event (CTR=LOAD) Writes to the CCx_y register are stored in shadow register and transferred to ECCx_y in the TIMCLK cycle following CTR equals 0 or CTR. Equals LD. 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. 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. 6h = Following a TRIG pulse. Writes to the CCx_y register are stored in shadow register and transferred to CCx_y
17	COC	R/W	0h	Capture or Compare. Specifies whether the corresponding CC register is used as a capture register or a compare register (never both). 0h = Compare 1h = Capture
16-15	RESERVED	R	0h	

Table 28-73. CCCTL_01[y] Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14-12	ZCOND	R/W	0h	Zero Condition. This field specifies the condition that generates a zero pulse. 0h = CCP edges have no effect 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 change (assertion/de-assertion edge)
11	RESERVED	R	0h	
10-8	LCOND	R/W	0h	Load Condition. Specifies the condition that generates a load pulse. 0h = CCP edges have no effect 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 change (assertion/de-assertion edge)
7	RESERVED	R	0h	
6-4	ACOND	R/W	0h	Advance Condition. Specifies the condition that generates an advance pulse. 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 change (assertion/de-assertion edge) 5h = CCP High or Trigger assertion (level)
3	RESERVED	R	0h	
2-0	CCOND	R/W	0h	Capture Condition. Specifies the condition that generates a capture pulse. 0h = None (never captures) 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 change (assertion/de-assertion edge)

28.3.47 CCCTL_23[y] (Offset = 1838h + formula) [Reset = 00000000h]

CCCTL_23[y] is shown in Figure 28-87 and described in Table 28-74.

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 28-87. 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 28-74. CCCTL_23[y] Field Descriptions

Bit	Field	Type	Reset	Description
31-29	CC2SELD	R/W	0h	Selects the source second CCD event. 0h = Selects CCD from CC0. 1h = Selects CCD from CC1. 2h = Selects CCD from CC2. 3h = Selects CCD from CC3. 4h = Selects CCD from CC4. 5h = Selects CCD from CC5.

Table 28-74. CCCTL_23[y] Field Descriptions (continued)

Bit	Field	Type	Reset	Description
28-26	CCACTUPD	R/W	0h	CCACT shadow register Update Method This field controls how updates to the CCCACT shadow register are performed 0h = Value written to the CCACTx_y register has immediate effect. 1h = Following a zero event (CTR=0) Writes to the CCACTx_y register are stored in shadow register and transferred to CCACTx_y in the TIMCLK cycle following CTR equals 0. 2h = Following a CCD event (CTR=CC_xy) 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. 3h = Following a CCU event (CTR=cc_xy) 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. 4h = Following a zero event (CTR=0) or load event (CTR=LOAD) 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. 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. 5h = Following a zero event (CTR=0) with repeat count also zero (RC=0). 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. 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 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. 0h = CCD, CCU and RC events are always generated from the counter when their conditions are generated. 1h = CCD, CCU and RC events are generated from the counter when their conditions are generated and the RC register value is 0.

Table 28-74. CCCTL_23[y] Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24-22	CC2SELU	R/W	0h	Selects the source second CCU event. 0h = Selects CCU from CC0. 1h = Selects CCU from CC1. 2h = Selects CCU from CC2. 3h = Selects CCU from CC3. 4h = Selects CCU from CC4. 5h = Selects CCU from CC5.
21	RESERVED	R	0h	
20-18	CCUPD	R/W	0h	Capture and Compare Update Method This field controls how updates to the shadow capture and compare register are performed (when operating in compare mode, COC=0). 0h = Writes to the CCx_y register is written to the register directly and has immediate effect. 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. 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. 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. 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. 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. 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. 6h = Following a TRIG pulse. Writes to the CCx_y register are stored in shadow register and transferred to CCx_y #xD; 0.

Table 28-74. CCCTL_23[y] Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17	COC	R/W	0h	Capture or Compare. Specifies whether the corresponding CC register is used as a capture register or a compare register (never both). 0h = Compare 1h = Capture
16-15	RESERVED	R	0h	
14-12	ZCOND	R/W	0h	Zero Condition. This field specifies the condition that generates a zero pulse. 4h-Fh = Reserved 0h = CCP edges have no effect 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 change (assertion/de-assertion edge)
11	RESERVED	R	0h	
10-8	LCOND	R/W	0h	Load Condition. Specifies the condition that generates a load pulse. 4h-Fh = Reserved 0h = CCP edges have no effect 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 change (assertion/de-assertion edge)
7	RESERVED	R	0h	
6-4	ACOND	R/W	0h	Advance Condition. Specifies the condition that generates an advance pulse. 6h-Fh = Reserved 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 change (assertion/de-assertion edge) 5h = CCP High or Trigger assertion (level)
3	RESERVED	R	0h	
2-0	CCOND	R/W	0h	Capture Condition. Specifies the condition that generates a capture pulse. 4h-Fh = Reserved 0h = None (never captures) 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 change (assertion/de-assertion edge)

28.3.48 CCCTL_45[y] (Offset = 1840h + formula) [Reset = 00000000h]

CCCTL_45[y] is shown in Figure 28-88 and described in Table 28-75.

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 28-88. 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 28-75. 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 This bit suppresses the generation of the compare (CCD, CCU and RC) events from the counter when the repeat counter (RC) value is not 0. 0h = CCD, CCU and RC events are always generated from the counter when their conditions are generated. 1h = CCD, CCU and RC events are generated from the counter when their conditions are generated and the RC register value is 0.
24-21	RESERVED	R	0h	

Table 28-75. CCCTL_45[y] Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20-18	CCUPD	R/W	0h	Capture and Compare Update Method This field controls how updates to the shadow capture and compare register are performed (when operating in compare mode, COC=0). 0h = Writes to the CCx_y register is written to the register directly and has immediate effect. 1h = Following a zero event (CTR=0) Writes to the CCx_y register are stored in shadow register and transferred to ECCx_y in the TIMCLK cycle following CTR equals 0. 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. 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. 4h = Following a zero event (CTR=0) or load event (CTR=LOAD) 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. 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. 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. 6h = Following a TRIG pulse. Writes to the CCx_y register are stored in shadow register and transferred to CCx_y #xD; 0.
17-0	RESERVED	R	0h	

28.3.49 OCTL_01[y] (Offset = 1850h + formula) [Reset = 00000000h]

OCTL_01[y] is shown in Figure 28-89 and described in Table 28-76.

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 28-89. 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 28-76. 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). 0h = Low 1h = High
4	CCPOINV	R/W	0h	CCP Output Invert The output as selected by CCPO is conditionally inverted. 0h = No inversion 1h = Invert
3-0	CCPO	R/W	0h	CCP Output Source 0h = Signal generator value (for example, PWM, triggered PWM) 1h = Load event 2h = CCU event or CCD event 4h = Zero event 5h = Capture event 6h = Fault condition 8h = Mirror CCP of first capture and compare register to other capture compare blocks 9h = Mirror CCP of second capture and compare register in other capture compare blocks Ch = Signal generator output after deadband insertion Dh = Counter direction

28.3.50 OCTL_23[y] (Offset = 1858h + formula) [Reset = 00000000h]

OCTL_23[y] is shown in Figure 28-90 and described in Table 28-77.

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 28-90. 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 28-77. 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). 0h = Low 1h = High
4	CCPOINV	R/W	0h	CCP Output Invert The output as selected by CCPO is conditionally inverted. 0h = No inversion 1h = Invert
3-0	CCPO	R/W	0h	CCP Output Source 0h = Signal generator value (for example, PWM, triggered PWM) 1h = Load condition 2h = CCU event or CCD event 4h = Zero event 5h = Capture event 6h = Fault Condition 8h = Mirror CCP of first capture and compare register in other capture compare blocks 9h = Mirror CCP of second capture and compare register in other capture compare blocks. Ch = Deadband Inserted Output Dh = Counter direction

28.3.51 CCACT_01[y] (Offset = 1870h + formula) [Reset = 00000000h]

CCACT_01[y] is shown in Figure 28-91 and described in Table 28-78.

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 28-91. 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 28-78. 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 This field describes the resulting action of software force. This action has a shadow register, which will be updated under specific condition. So that this register cannot take into effect immediately. 0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event. 1h = CCP Complimentary output value is set high 2h = CCP Complimentary output value is set low
29-28	SWFRCACT	R/W	0h	CCP Output Action on Software Force Output This field describes the resulting action of software force. This action has a shadow register, which will be updated under specific condition. So that this register cannot take into effect immediately. 0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event. 1h = CCP output value is set high 2h = CCP output value is set low
27-25	FEXACT	R/W	0h	CCP Output Action on Fault Exit This field describes the resulting action of the signal generator upon exiting the fault condition. 0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event. 1h = CCP output value is set high 2h = CCP output value is set low 3h = CCP output value is toggled 4h = CCP output value is tristated

Table 28-78. CCACT_01[y] Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24-22	FENACT	R/W	0h	CCP Output Action on Fault Entry This field describes the resulting action of the signal generator upon detecting a fault. 0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event. 1h = CCP output value is set high 2h = CCP output value is set low 3h = CCP output value is toggled 4h = CCP output value is tristated
21-17	RESERVED	R	0h	
16-15	CC2UACT	R/W	0h	CCP Output Action on CC2U event. 0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event. 1h = CCP output value is set high 2h = CCP output value is set low 3h = CCP output value is toggled
14	RESERVED	R	0h	
13-12	CC2DACT	R/W	0h	CCP Output Action on CC2D event. 0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event. 1h = CCP output value is set high 2h = CCP output value is set low 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. 0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event. 1h = CCP output value is set high 2h = CCP output value is set low 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. 0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event. 1h = CCP output value is set high 2h = CCP output value is set low 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. 0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event. 1h = CCP output value is set high 2h = CCP output value is set low 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. 0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event. 1h = CCP output value is set high 2h = CCP output value is set low 3h = CCP output value is toggled

28.3.52 CCACT_23[y] (Offset = 1878h + formula) [Reset = 00000000h]

CCACT_23[y] is shown in Figure 28-92 and described in Table 28-79.

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 28-92. 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 28-79. 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 This field describes the resulting action of software force. This action has a shadow register, which will be updated under specific condition. So that this register cannot take into effect immediately. 0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event. 1h = CCP Complimentary output value is set high 2h = CCP Complimentary output value is set low
29-28	SWFRCACT	R/W	0h	CCP Output Action on Software Force Output This field describes the resulting action of software force. This action has a shadow register, which will be updated under specific condition. So that this register cannot take into effect immediately. 0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event. 1h = CCP output value is set high 2h = CCP output value is set low
27-25	FEXACT	R/W	0h	CCP Output Action on Fault Exit This field describes the resulting action of the signal generator upon exiting the fault condition. 0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event. 1h = CCP output value is set high 2h = CCP output value is set low 3h = CCP output value is toggled 4h = CCP output value is tristated

Table 28-79. CCACT_23[y] Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24-22	FENACT	R/W	0h	CCP Output Action on Fault Entry This field describes the resulting action of the signal generator upon detecting a fault. 0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event. 1h = CCP output value is set high 2h = CCP output value is set low 3h = CCP output value is toggled 4h = CCP output value is tristated
21-17	RESERVED	R	0h	
16-15	CC2UACT	R/W	0h	CCP Output Action on CC2U event. 0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event. 1h = CCP output value is set high 2h = CCP output value is set low 3h = CCP output value is toggled
14	RESERVED	R	0h	
13-12	CC2DACT	R/W	0h	CCP Output Action on CC2D event. 0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event. 1h = CCP output value is set high 2h = CCP output value is set low 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. 0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event. 1h = CCP output value is set high 2h = CCP output value is set low 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. 0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event. 1h = CCP output value is set high 2h = CCP output value is set low 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. 0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event. 1h = CCP output value is set high 2h = CCP output value is set low 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. 0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event. 1h = CCP output value is set high 2h = CCP output value is set low 3h = CCP output value is toggled

28.3.53 IFCTL_01[y] (Offset = 1880h + formula) [Reset = 00000000h]

IFCTL_01[y] is shown in Figure 28-93 and described in Table 28-80.

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 28-93. 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 28-80. IFCTL_01[y] Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R	0h	
12	FE	R/W	0h	Filter Enable This bit controls whether the input is filtered by the input filter or bypasses to the edge detect. 0h = Bypass. 1h = Filtered.
11	CPV	R/W	0h	Consecutive Period/Voting Select This bit controls whether the input filter uses a stricter consecutive period count or majority voting. 0h = Consecutive Periods The input must be at a specific logic level for the period defined by FP before it is passed to the filter output. 1h = Voting The filter ignores one clock of opposite logic over the filter period. That is, 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. That is, the input is sampled for FP timer clocks during filtering. 0h = The division factor is 3 1h = The division factor is 5 2h = The division factor is 8

Table 28-80. 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. 0h = Noninverted 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 0h = CCP of the corresponding capture compare unit 1h = Input pair CCPX of the capture compare unit. For CCP0 input pair is CCP1 and for CCP1 input pair is CCP0. 2h = CCP0 of the counter 3h = Trigger 4h = XOR of CCP inputs as input source (Used in Hall input mode). 5h = subscriber 0 event as input source. 6h = subscriber 1 event as input source. 7h = Comparator 0 output. 8h = Comparator 1 output. 9h = Comparator 2 output.

28.3.54 IFCTL_23[y] (Offset = 1888h + formula) [Reset = 00000000h]

IFCTL_23[y] is shown in Figure 28-94 and described in Table 28-81.

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 28-94. 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 28-81. IFCTL_23[y] Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R	0h	
12	FE	R/W	0h	Filter Enable This bit controls whether the input is filtered by the input filter or bypasses to the edge detect. 0h = Bypass. 1h = Filtered.
11	CPV	R/W	0h	Consecutive Period/Voting Select This bit controls whether the input filter uses a stricter consecutive period count or majority voting. 0h = Consecutive Periods The input must be at a specific logic level for the period defined by FP before it is passed to the filter output. 1h = Voting The filter ignores one clock of opposite logic over the filter period. That is, 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. That is, the input is sampled for FP timer clocks during filtering. 0h = The division factor is 3 1h = The division factor is 5 2h = The division factor is 8
7	INV	R/W	0h	Input Inversion This bit controls whether the selected input is inverted. 0h = Noninverted 1h = Inverted
6-4	RESERVED	R	0h	

Table 28-81. IFCTL_23[y] Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-0	ISEL	R/W	0h	Input Select (CCP0) This field selects the input source to the filter input. 4h-7h = Reserved 0h = CCP of the corresponding capture compare unit 1h = Input pair CCPX of the capture compare unit. For CCP0 input pair is CCP1 and for CCP1 input pair is CCP0. 2h = CCP0 of the counter 3h = Trigger 4h = XOR of CCP inputs as input source (Used in Hall input mode). 5h = subscriber 0 event as input source. 6h = subscriber 1 event as input source. 7h = Comparator 0 output. 8h = Comparator 1 output. 9h = Comparator 2 output.

28.3.55 PL (Offset = 18A0h) [Reset = 00000000h]

PL is shown in Figure 28-95 and described in Table 28-82.

Return to the Summary Table.

This is the phase load register.

Figure 28-95. 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 28-82. PL Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	PHASE	R/W	0h	Phase Load value 0h = Minimum value 00FFFFFFh = Maximum Value

28.3.56 DBCTL (Offset = 18A4h) [Reset = 00000000h]

DBCTL is shown in Figure 28-96 and described in Table 28-83.

Return to the Summary Table.

The DBCTL register controls the dead band insertion of the pulse width modulated output.

Figure 28-96. 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 28-83. DBCTL Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	
27-16	FALLDELAY	R/W	0h	Fall Delay The number of TIMCLK periods inserted between the fall edge of CCP signal and the rise edge of CCP complimentary signal. 0h = Minimum value FFFh = Maximum Value
15-13	RESERVED	R	0h	
12	M1_ENABLE	R/W	0h	Dead Band Mode 1 Enable. 0h = Disabled 1h = Enabled
11-0	RISEDELAY	R/W	0h	Rise Delay The number of TIMCLK periods inserted between the fall edge of CCP signal and the rise edge of CCP complimentary signal. 0h = Minimum value FFFh = Maximum Value

28.3.57 TSEL (Offset = 18B0h) [Reset = 00000000h]

TSEL is shown in Figure 28-97 and described in Table 28-84.

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 28-97. 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 28-84. TSEL Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	0h	
9	TE	R/W	0h	Trigger Enable. This selects whether a trigger is enabled or not for this counter 0x0 = Triggers are not used 0x1 = Triggers are used as selected by the ETSEL field 0h = Triggers are not used. 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. This selects which System Event is used if the input filter selects trigger. Triggers 0-15 are used to connect triggers generated by other timer modules. Refer to the SoC data sheet for details related to timer trigger sources. Triggers 16 and 17 are connected to event manager subscriber ports. Event lines 18-31 are reserved for future use. 0h = TRIGx = External trigger input from TIM x. 1h = TRIGx = External trigger input from TIM x. 2h = TRIGx = External trigger input from TIM x. 3h = TRIGx = External trigger input from TIM x. 4h = TRIGx = External trigger input from TIM x. 5h = TRIGx = External trigger input from TIM x. 6h = TRIGx = External trigger input from TIM x. 7h = TRIGx = External trigger input from TIM x. 8h = TRIGx = External trigger input from TIM x. 9h = TRIGx = External trigger input from TIM x. Ah = TRIGx = External trigger input from TIM x. Bh = TRIGx = External trigger input from TIM x. Ch = TRIGx = External trigger input from TIM x. Dh = TRIGx = External trigger input from TIM x. Eh = TRIGx = External trigger input from TIM x. Fh = TRIGx = External trigger input from TIM x. 10h = TRIG_SUBx = External trigger input from subscriber port x. 11h = TRIG_SUBx = External trigger input from subscriber port x.

28.3.58 RC (Offset = 18B4h) [Reset = 00000000h]

RC is shown in Figure 28-98 and described in Table 28-85.

Return to the Summary Table.

Repeat counter is to reduce interrupt overhead. The repeat counter provides the mechanism to suppress unnecessary 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 28-98. 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 28-85. RC Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	RC	R	0h	Repeat Counter Value 0h = Minimum value FFh = Maximum Value

28.3.59 RCLD (Offset = 18B8h) [Reset = 00000000h]

RCLD is shown in Figure 28-99 and described in Table 28-86.

Return to the Summary Table.

The load register value is transferred to the counter when the counter load input is asserted.

Figure 28-99. 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 28-86. RCLD Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	RCLD	R/W	0h	Repeat Counter Load Value This field provides the value loaded into the repeat counter at a load event following the repeat counter value equaling 0. 0h = Minimum value FFh = Maximum Value

28.3.60 QDIR (Offset = 18BCh) [Reset = 00000000h]

QDIR is shown in Figure 28-100 and described in Table 28-87.

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 28-100. 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 28-87. QDIR Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	DIR	R	0h	Direction of count 0h = Down (Phase B leads Phase A) 1h = Up (Phase A leads Phase B)

28.3.61 FCTL (Offset = 18D0h) [Reset = 00000000h]

FCTL is shown in Figure 28-101 and described in Table 28-88.

Return to the Summary Table.

The FCTL register controls the fault inputs, fault detection and error handling behavior.

Figure 28-101. 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 28-88. 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. 0h = Fault Input is active low. 1h = Fault Input is active high.
12	FSENEXT1	R/W	0h	Specifies whether the external fault pin1 high/low is treated as fault condition. 0h = Fault Input is active low. 1h = Fault Input is active high.
11	FSENEXT0	R/W	0h	Specifies whether the external fault pin0 high/low is treated as fault condition. 0h = Fault Input is active low. 1h = Fault Input is active high.
10	FSENAC2	R/W	0h	Specifies whether the COMP2 output high/low is treated as fault condition. 0h = Fault Input is active low. 1h = Fault Input is active high.
9	FSENAC1	R/W	0h	Specifies whether the COMP1 output high/low is treated as fault condition. 0h = Fault Input is active low. 1h = Fault Input is active high.
8	FSENAC0	R/W	0h	Specifies whether the COMP0 output high/low is treated as fault condition. 0h = Fault Input is active low. 1h = Fault Input is active high.
7	TFIM	R/W	0h	Trigger Fault Input Mask Specifies whether the selected trigger participates as a fault input. If enabled and the trigger asserts, the trigger is treated as a fault. 0h = Selected trigger does not participate in fault condition generation 1h = Selected trigger participates in fault condition generation
6-5	RESERVED	R	0h	

Table 28-88. FCTL Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4-3	FL	R/W	0h	Fault Latch mode Specifies whether the fault condition is latched and configures the latch clear conditions. 0h = Overall fault condition is not dependent on the F bit in RIS 1h = Overall fault condition is dependent on the F bit in RIS 2h = Fault condition is latched. Fault condition is cleared on a zero event if the fault input is 0. 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 Specifies whether the overall fault condition is dependent on the sensed fault pin. 0h = Overall Fault condition is not dependent on sensed input. 1h = Overall Fault condition is dependent on sensed input.
1	RESERVED	R	0h	
0	FIEN	R/W	0h	Fault Input Enable This bit enables the input for fault detection. 0h = Fault Input Disabled 1h = Fault Input Enabled

28.3.62 FIFCTL (Offset = 18D4h) [Reset = 00000000h]

FIFCTL is shown in Figure 28-102 and described in Table 28-89.

Return to the Summary Table.

The FIFCTL register controls the filtering for the fault input.

Figure 28-102. 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 28-89. FIFCTL Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	
4	FILTEN	R/W	0h	Filter Enable 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. 0h = Bypass 1h = Filtered.
3	CPV	R/W	0h	Consecutive Period/Voting Select This bit controls whether the input filter uses a stricter consecutive period count or majority voting. 0h = Consecutive Periods. The input must be at a specific logic level for the period defined by FP before it is passed to the filter output. 1h = Voting. The filter ignores one clock of opposite logic over the filter period, meaning that during 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 This field specifies the sample period for the input filter. The input is sampled for FP timer clocks during filtering. 0h = Filter Period 3 1h = Filter Period 5 2h = Filter Period 8

Chapter 29

Low Frequency Subsystem (LFSS)

29.1 Overview 1822
29.2 Clock System 1823
29.3 LFSS Reset Using VBAT 1824
29.4 Power Domains and Supply Detection 1824
29.5 Real Time Counter (RTC_x) 1827
29.6 Independent Watchdog Timer (IWDT) 1827
29.7 Tamper Input and Output 1828
29.8 Scratchpad Memory 1831
29.9 Lock Function of RTC, TIO, and IWDT 1831
29.10 LFSS Registers 1833

29.1 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. In some implementations, the LFSS can be powered by a separate battery backup domain called VBAT. The low-frequency clock has a typical frequency of 32kHz and is mainly intended for long-term timekeeping.

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 that 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)

In this combination, these modules stay active and functional through a power loss of the main supply (VDD). Figure 29-1 shows how the LFSS fits into the device in relationship to the main power management controller unit (PMCU) subsystems.

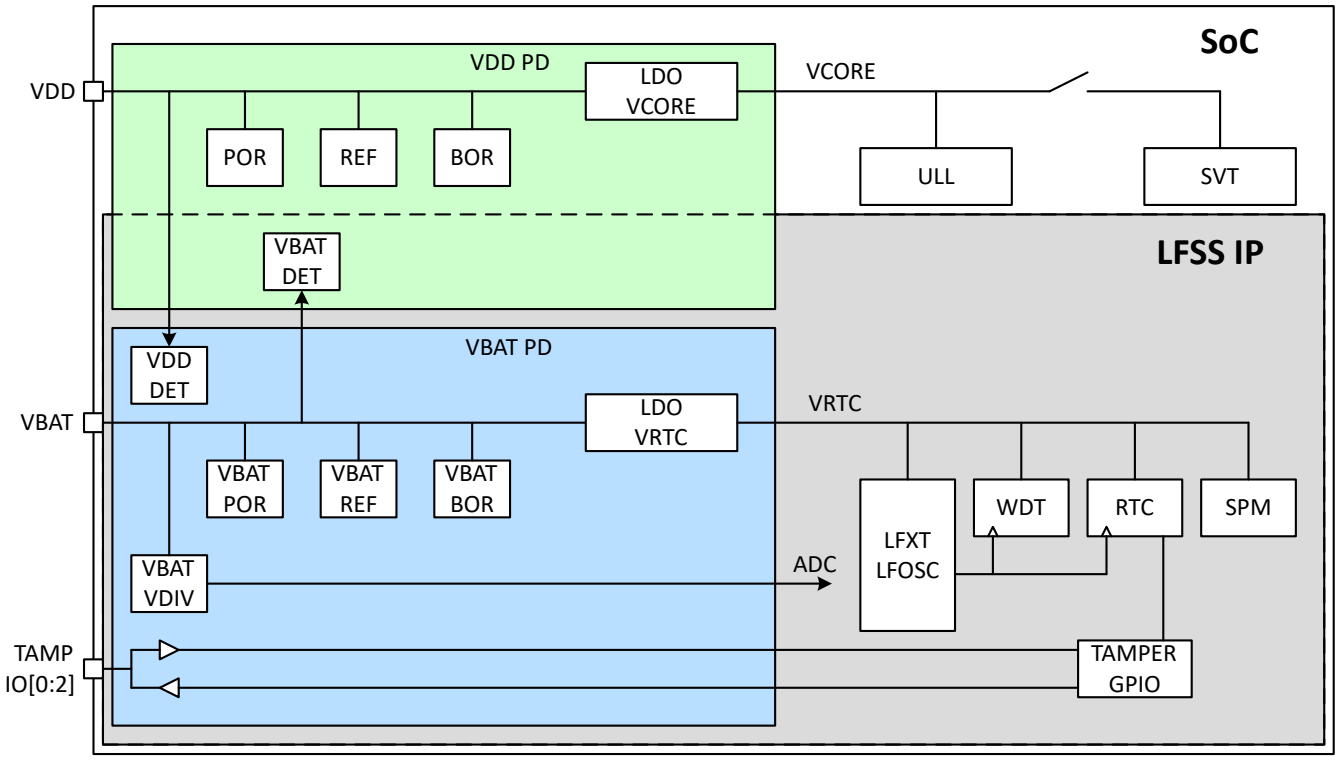


Figure 29-1. LFSS Overview

The components above include:

- **VDD / VDD PD** is the main supply power domain that powers the I/O and subregulated domains of the device.
- **VBAT / VBAT PD** is the battery backup supply power domain and powers a secondary power management unit (VBAT-PMU), which provides a regulated supply for the RTC and supporting circuits. This domain also powers the tamper I/O cells, WDT, and SPM block.
- **VCORE** is the regulated subdomain supply for most of the device digital logic .

- **VRTC** is the regulated subdomain supply powered by the VBAT supply. This supply powers the RTC, LFXT, LFOSC, SPM, WDT and tamper related circuits inside LFSS.
- **VBAT DET / VDD DET** is the detection circuit to monitor the presence of the VBAT / VDD supply accordingly.
- **VBAT VDIV** is a resistive voltage divider, which allows the measurement of the VBAT supply voltage by the internal ADC of the device.

29.2 Clock System

The low-frequency clock system is part of LFSS, but the control bits are located in the SYSCTL module. Figure 29-2 illustrates the components of the clock system.

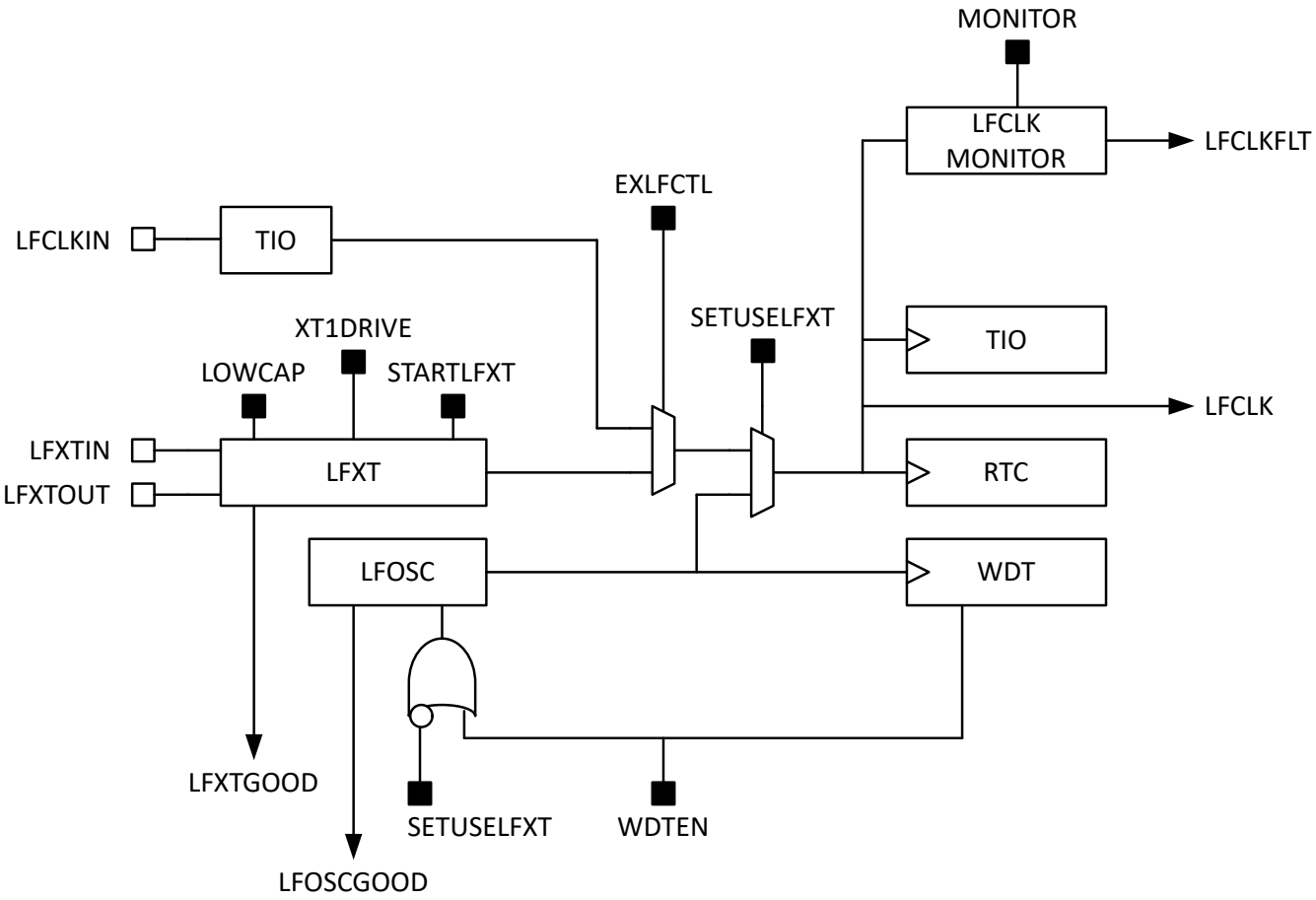


Figure 29-2. LFSS Low-Frequency Clock System

The control bits for LFCLK control is located inside the SYSCTL registers. The user control signals are latched in a shadow register inside LFSS-related logic. In case of a power loss of the VDD / VCORE domain, 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. This concept is similar to the GPIO sleep function in SHUTDOWN mode.

Note

The LFCLK is used in the device for other peripherals and is required for STANDBY mode by the device's power management unit. If the VBAT domain is not powered (for example, during VBAT battery replacement), the device does not have a valid LFCLK for these purposes. If the VBAT domain is not powered, only the SYSOSC derived clock is available, which cannot be used for STANDBY mode. There are three scenarios that need to be considered:

1. If the device is in STANDBY mode while the LFSS loses power, the VBAT power-down interrupt wakes the device from STANDBY mode.
2. If the device is in RUN mode and wants to transition into STANDBY mode while LFSS is not powered, the device transitions into STOP mode with a SYSOSC clock of 32MHz.
3. If the device is in RUN mode and the CPU clock is sourced by LFCLK, the loss of the VBAT domain switches the clock back to SYSOSC and issue an interrupt that VBAT is lost.

29.3 LFSS Reset Using VBAT

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.

29.4 Power Domains and Supply Detection

The LFSS needs to monitor the presence of the VBAT supply and the VDD supply independently. There is no dedicated always-on domain which can act as a trusted supervisor power domain to indicate which domain is powered and which one is not. The VBAT domain as well as the VDD domain can be powered or not powered independently as a legal operating scenario.

For example, when the main VDD supply fails, the VBAT domain will be powered by a battery to supply the LFXT and RTC. On the other hand, the battery on the VBAT domain can be replaced, which means a temporary power loss on the VBAT power domain, while the main VDD supply powers the device. The LFSS does not contain any cross-over switches that supplies the failing power domain from the powered domain.

The voltage levels on VDD and VBAT can also be very different and there is no requirement for one supply to be higher than the other one. For example, the device can be powered by a 1.8V LDO from the system LDO on the PCB while the VBAT supply is coming from a 3V coin cell battery. Alternatively, the device can be powered by a system LDO at 3.3V while the voltage of a nearly depleted battery drops down to 2.0V.

As shown in Figure 29-1, each domain has a Vth based power-on reset (POR) circuit for the initial initialization of its PMU circuitry. A bandgap-based reference system (REF) will provide an accurate voltage reference for the brown-out reset (BOR) circuit and the LDO.

The LFSS provides two additional voltage sense circuits to indicate whether the other 3V power domain is powered or not.

With this architecture, the LFSS supports four typical application use cases, which are shown in Figure 29-3.

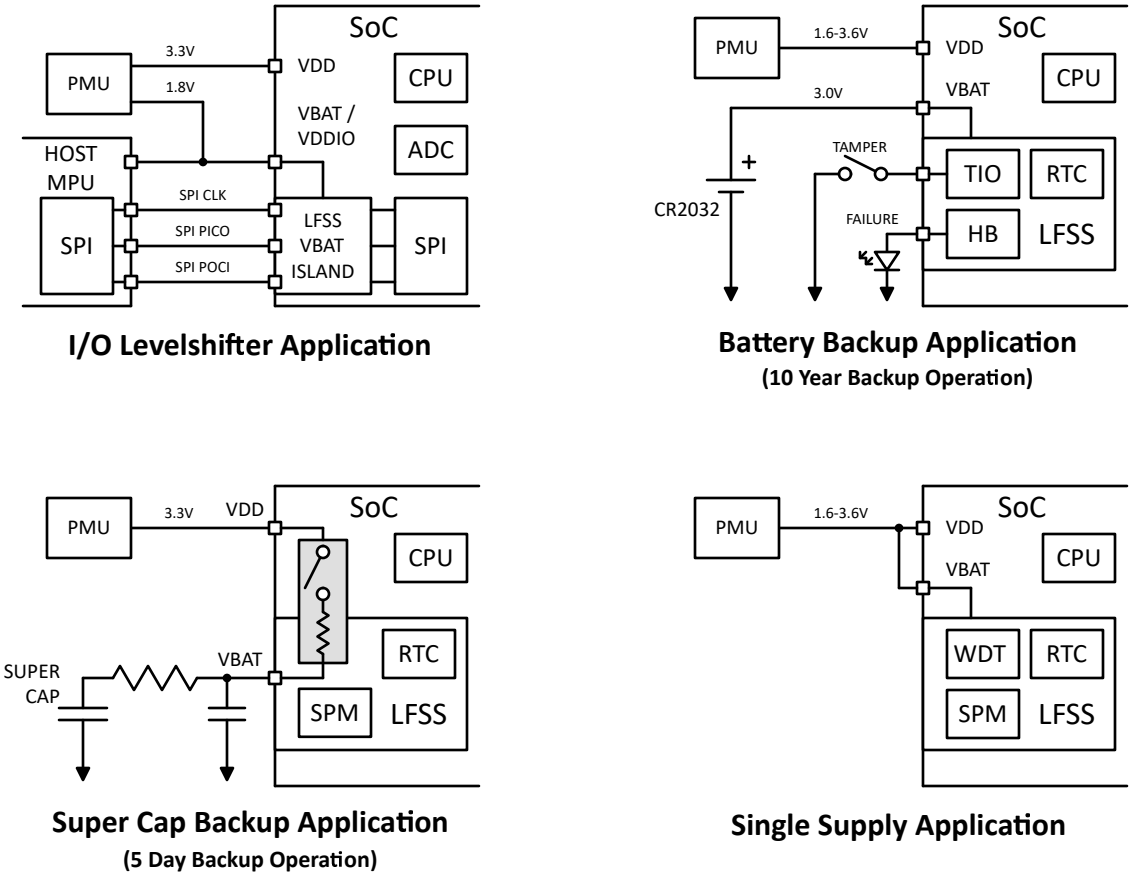


Figure 29-3. LFSS Typical Application Use Cases

29.4.1 Startup When VBAT Powers on First

When the VBAT domain is supplied first on a completely unpowered device, the LFSS will start with the asynchronous sequence of the VBAT PMU.

The release of the VBAT POR will start the VBAT REF, VBAT BOR and VBAT LDO.

When the reset is de-asserted, the LFOSC starts and provides a 32kHz clock that operates the VBAT REF and BOR circuit in sampled mode. If this state is reached, the VBAT waits for further configuration by software. In this

state, the overall power consumption of the LFSS is below the specified limit to enable 10-year lifetime from a coin cell battery.

29.4.2 Startup when VDD powers on first

When the VDD domain is supplied first on a completely unpowered device, the device will start with the asynchronous sequence of the device PMU. The device will start normally as if the VBAT domain is powered.

The difference is only related to the unavailability of the LFCLK for use in the device.

The VBAT-detector (which is powered by the VDD domain) will indicate the VBAT supply is not present yet and therefore ignore all digital signals from the VRTC domain. Since the VBAT domain contains the LFCLK circuit, there will be no LFCLK available in the device. This specifically means the device cannot go into STANBY mode, because the 32kHz is required for the STANDBY mode.

29.4.3 Behavior When VDD is Lost

Due to the required system-level decoupling capacitors on the VDD supply, the power loss is not instantaneous. When the VDD supply ramps down and reaches the BOR- level, the VCORE LDO is disabled. This signal propagates through the VBAT domain to the VRTC domain and logically isolates all signals coming from the VCORE domain.

Eventually, the VDD supply drops below the POR level and resets the whole VDD domain of the device. This level is also detected by the VDD detect circuit, which is powered by the VBAT domain. At this time, all signals from the VDD domain cannot be trusted and the VDD isolation signal isolates all signals coming from the VDD domain.

29.4.4 Behavior when VBAT is lost

A certain minimum decoupling capacitor will be required on the VBAT supply pin. This is mainly for noise immunity, but also to supply peak currents during switching activity on a high impedance power source. Specifically, the tamper I/O pins can create fairly high switching noise.

Due to the required decoupling cap, the loss of the VBAT supply will happen in a ramp, rather than a step function, even when the VBAT battery will suddenly be removed from the system. Due to the ramp, the VBAT-DET circuit will observe the loss of the VBAT domain and isolate all signals coming from the VRTC domain.

29.4.5 Behavior when the device goes into SHUTDOWN mode

The shutdown mode sequence is slightly different, since the VDD domain stays powered. The shutdown mode is initiated by software and will be stored in the shutdown mode register in the VDD domain. This mode register will disable the LDO. This signal will propagated through the VBAT domain to the VRTC domain and logically isolate all signals coming from the VCORE domain.

On a wakeup from shutdown mode, the device PMU will re-enable the REF system, BOR circuit and finally the LDO. Once the subregulated VCORE supply has reached the target level, the VRTC domain is again accessible by the VCOCRE domain.

29.4.6 Supercapacitor Charging Circuit

The VBAT PMU of the LFSS supports a charging circuit for an external supercapacitor connected to the VBAT supply pin. An overview of the charging circuit can be found in Figure 29-4.

The charging circuit contains an internal connect switch between the device VDD supply and the LFSS VBAT supply pin. The switch has an internal resistance and therefore limits the charge current. The supercapacitor needs to be connected with a series resistor to further limit the charge current. The external resistor needs to be in the same order as the internal switch resistor. This is important to start up the VBAT domain of the LFSS quickly.

The charging circuit is not active after power up of the device. The charging circuit needs to be enabled by SW.

For example, assume the charging circuit gets enabled with a completely depleted supercapacitor. In the initial state, the voltage over the capacitor is 0V, which means from modeling point of view this circuit node acts like

a virtual ground. In this case, the internal and external current limiting resistor builds a voltage divider. If both resistors are the same value, the resulting voltage on VBAT is 1/2 of VDD. This is typically enough to start the operation of the LFSS.

The charging circuit contains an internal comparator circuit to supervise the external VDD supply. Should the external VDD supply drop below the VBAT supply, the charger switch will get disabled. This will prevent the reverse current flow from the supercapacitor back into the main supply. When the external VDD supply rises above the VBAT supply, the charging switch will be closed again.

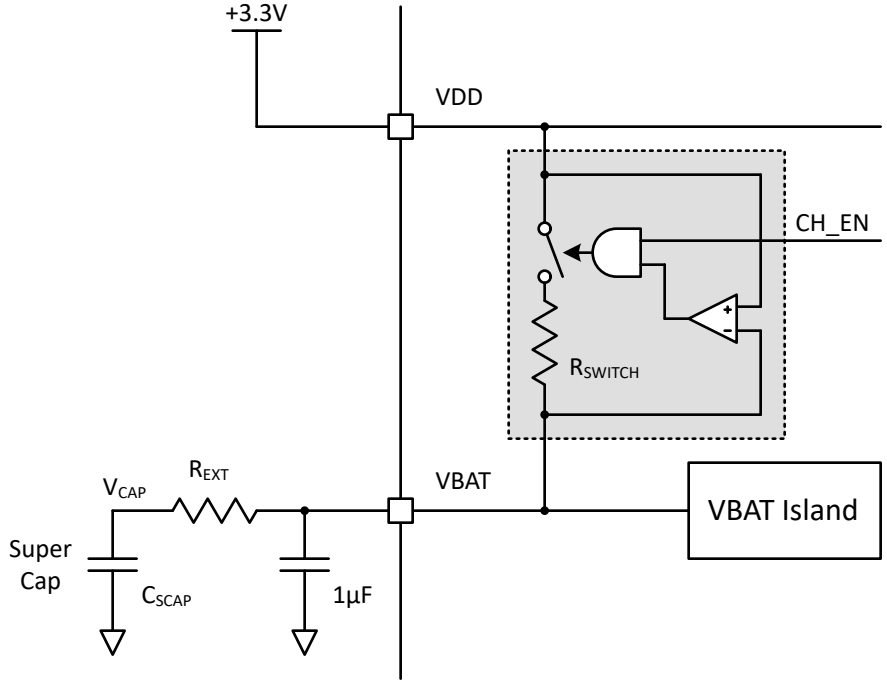


Figure 29-4. Supercapacitor Charging Circuit

29.5 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.

For more information on the LFSS real-time counter and various RTC instances, see the RTC chapter.

29.6 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.

29.7 Tamper Input and Output

The tamper I/O is a GPIO which is sourced by the VBAT power domain. The LFSS will contain up to 16 I/Os of this type. The I/Os have two modes of operation:

- In **IOMUX mode**, the tamper I/O input and output path are connected to the device IOMUX module and the I/Os can be used as a second function like SPI, UART, or timer PWM.
- In **Tamper mode**, the I/O is completely controlled by the LFSS and will remain functional during the power loss of the main supply or during SHUTDOWN mode.

The I/O pin output driver has enough drive capability to drive a low current LED from the VBAT domain.

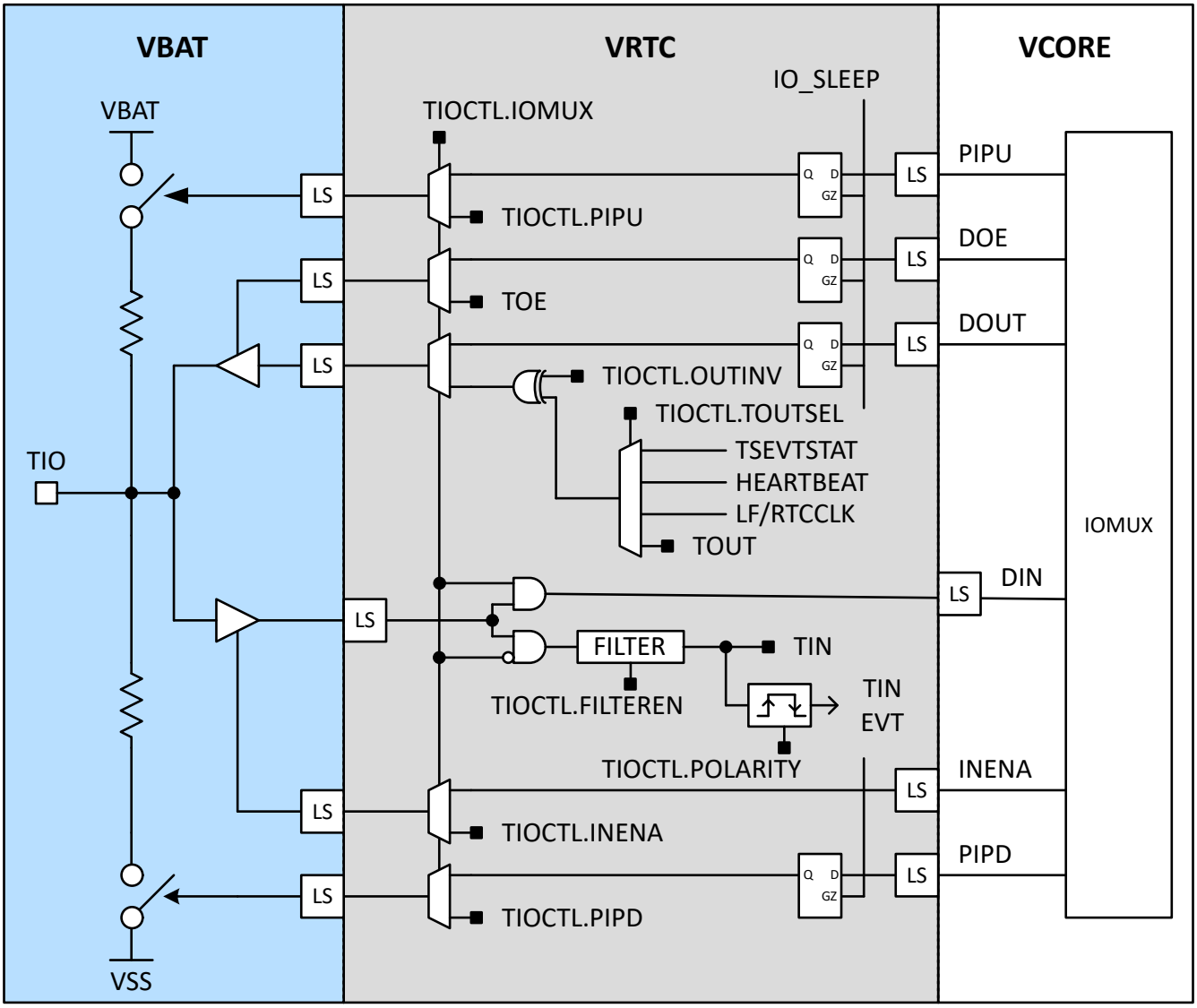


Figure 29-5. Tamper I/O Circuit Diagram

29.7.1 IOMUX Mode

The IOMUX mode is the default mode for LFSS. The IOMUX mode is mainly useful for applications where the VBAT and VDD are shorted, or when LFSS is used as a secondary supply to operate the second function at another voltage level as the main supply. This can be useful when the device is powered by a 3V domain to gain the full analog performance, but requires a SPI connection to a host CPU running at 1.8V. The LFSS will act as a device internal level shifter module for the SPI I/Os.

In general, the IOMUX mode gives the most software backward compatibility to other devices. By default, all existing software utilizing the GPIO or second function of the LFSS-controlled I/O will work as on other device that does not contain LFSS.

If the LFOSC is not used as a clock source for LFSS or the device, then LFSS supports the usage of the LFXIN/LFXOUT pins in IOMUX mode. This means the LFXIN and LFXOUT pins will be controlled by the IOMUX in the VCORE domain and second functions selected.

The limitation of the IOMUX mode is that the I/O is not functional when the main supply is lost, or when the device is in SHUTDOWN mode. In both cases, VCORE is not available and the control for the tamper I/O is not available.

In case of the SHUTDOWN mode, LFSS will latch the last status (I/O direction and pull up/down configuration) of the I/O similar to regular I/Os in SHUTDOWN mode. Similar to the standard device I/Os, the latch of the control signals for the LFSS controlled I/Os will be active until the SLEEP_IO is released by software.

This mode does not allow a wakeup of the device from SHUTDOWN mode. The tamper mode allows wakeup from SHUTDOWN mode (see Section 29.7.2).

29.7.2 Tamper Mode

In Tamper Mode, all controls and circuits are powered by the internal VRTC domain and will remain functional during loss of the VDD main supply or when the device is in SHUTDOWN mode.

Similar to the GPIO module, the tamper I/O can be configured in input or output mode. For the input mode operation, the I/O can be configured with an internal pull-up or pull-down resistor connected to the I/O pad.

The tamper mode also supports several LFSS-related internal secondary functions:

- Tamper event detection
- Timestamp event output
- Heartbeat generator
- LFCLK output

29.7.2.1 Tamper Event Detection

The tamper event detection allows to configure one or more tamper I/O's to trigger a timestamp event and to generate an interrupt to the CPU. The corresponding I/O (which acts as a tamper event trigger) needs to be configured as an input. To minimize false triggers, a digital filter circuit is inserted into the I/O path. Only pulses that are longer than the configured filter width will be registered as a tamper event. The tamper event trigger itself can be issued on a rising or falling edge of the filtered input signal.

The selectable input filter width is 65us, 100us, 155us and 250us.

29.7.2.2 Timestamp Event Output

The timestamp event can be triggered by an edge detection of any of the tamper I/O or by a power loss detection of the main supply (VDD). This event will capture the RTC state as a timestamp of the first or last occurrence of the timestamp event. The TSCTL register controls which source that triggers a capture event and whether the first of last event is captured. An illustration of the timestamp feature in context of the RTC can be seen in ?.

In the case that a capture event happens, the tamper I/O can be configured to display this status to the outside world via an I/O pin. The signal is sticky and will get set by the first timestamp event. It will stay active till the timestamp is cleared by SW (TSCTL.TSCLR).

29.7.2.3 Heartbeat Generator

The heartbeat generator allows the signaling of a certain operating state of LFSS to the outside world. The signal is a repetitive pulse with a configurable interval time and pulse width.

For example, it can be used as a trigger for an external watchdog timer to indicate the RTC is still working. Alternatively, it can be used to flash a LED in a failure scenario as a visual alarm indicator. The interval time and pulse width are selectable to tailor the heartbeat to the intended application. Figure 29-6 shows the circuit diagram of the heartbeat generator.

Selectable interval time: 125ms, 250ms, 500ms, 1s, 2s, 4s, 8s, 16s

Selectable pulse width: 1ms to 128ms in binary steps

Selectable heart beat mode: off, always-on, on with timestamp event, on with main supply fail

The selectable interval time and pulse width are dimensioned to allow the low power operation of an LED in blink/flash mode. For example, an average current load of a 2mA LED with a 1s period and 1ms flash configuration is 2µA.

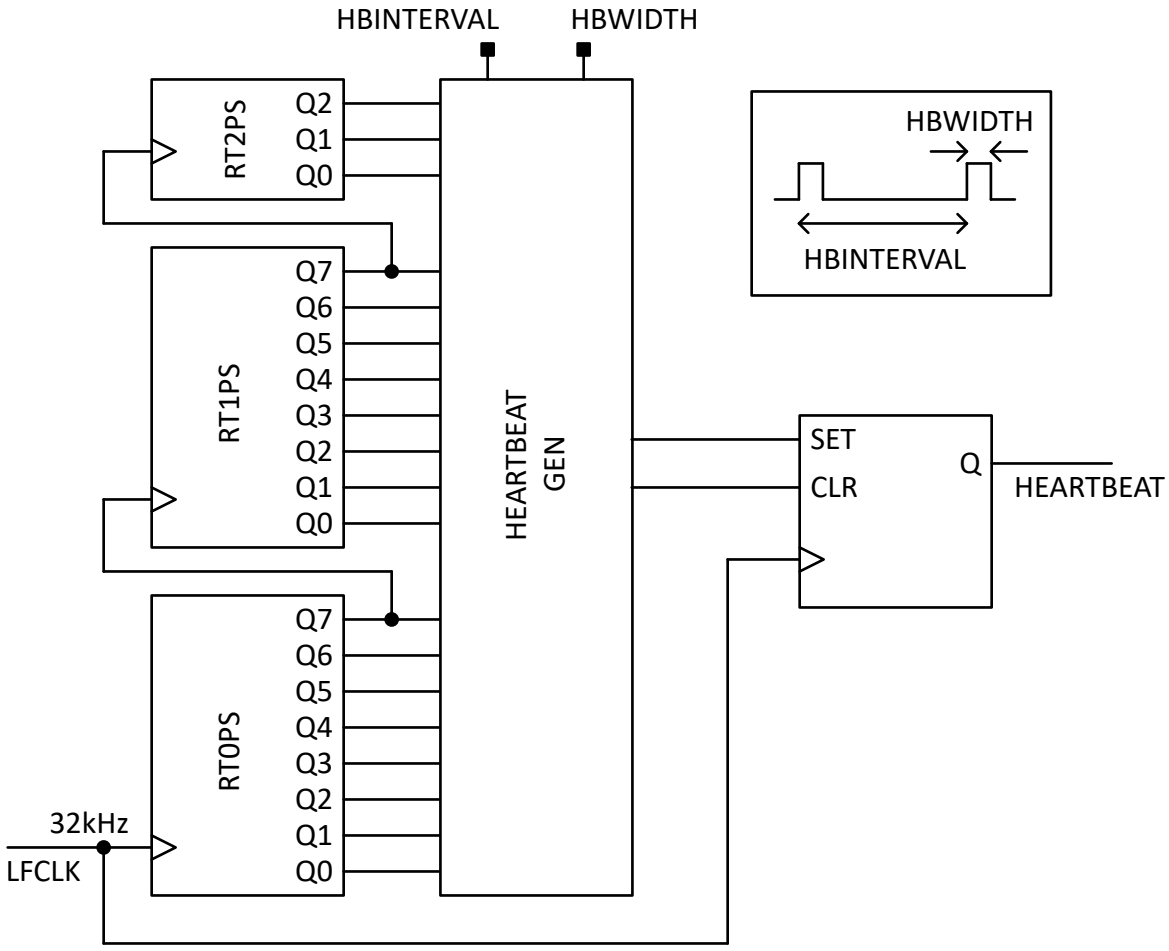


Figure 29-6. Heartbeat Generator Circuit Diagram

29.7.2.4 RTC Clock Output

For the calibration of the RTC, or to share the 32kHz signal with other devices in the system, the RTC can be routed to one of the tampers I/O's.

Selectable frequency: 32kHz, 512Hz, 256Hs, 1Hz

29.8 Scratchpad Memory

The LFSS scratchpad memory is a 16- to 256-byte register based memory that retains data similar to a nonvolatile memory as long as VBAT is supplied. This means the scratchpad memory retains data when the main VDD supply is lost or during the SHUTDOWN mode. However, the memory content is not retained when the VBAT domain is lost. The memory is organized in 4 to 64 words of 32-bit size. Each byte is single addressable for read and write.

To protect the memory against unwanted (accidental software initiated) writes, an LFSS memory write enable register is implemented. Each bit in the write enable register corresponds to the byte location in the scratchpad memory.

In the event of a tamper detect (time stamp event), the scratchpad memory erases memory locations that are enabled for erase. The indication is done by an LFSS memory tamper erase enable register. Each bit in the register corresponds to one-byte location in the scratchpad memory.

Figure 29-7 shows the memory.

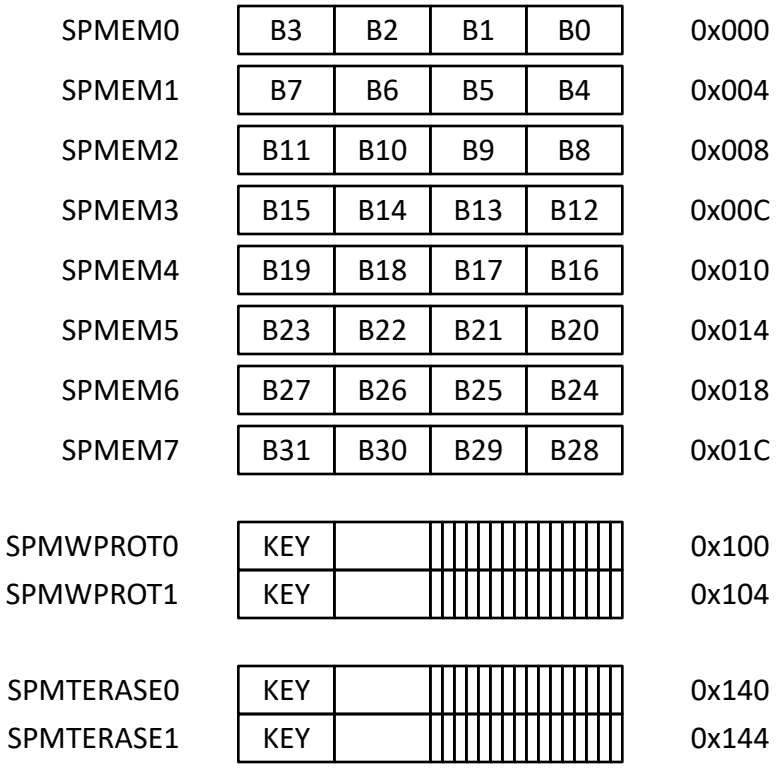


Figure 29-7. Scratchpad Memory

29.9 Lock Function of RTC, TIO, and IWDT

Safety or secure applications must make sure that the RTC value, IWDT, and TIO 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, IWDT configuration, and the TIO 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, IWDT restart function and the TIO output register are not affected by the lock bit.

29.10 LFSS Registers

Table 29-1 lists the memory-mapped registers for the LFSS 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. 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

Table 29-1. 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

Table 29-1. 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 29-2 shows the codes that are used for access types in this section.

Table 29-2. LFSS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Rmodify	R modify	Read
Write Type		
W	W	Write

Table 29-2. LFSS Access Type Codes (continued)

Access Type	Code	Description
WK	W K	Write 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.

29.10.1 FSUB_0 (Offset = 400h) [Reset = 00000000h]

FSUB_0 is shown in Figure 29-8 and described in Table 29-3.

Return to the Summary Table.

Subscriber port

Figure 29-8. 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 29-3. FSUB_0 Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3-0	CHANID	R/W	0h	0 = disconnected. 1-15 = connected to channelID = CHANID. 0h = A value of 0 specifies that the event is not connected Fh = Consult your device data sheet as the actual allowed maximum may be less than 15.

29.10.2 FPUB_0 (Offset = 444h) [Reset = 00000000h]

FPUB_0 is shown in Figure 29-9 and described in Table 29-4.

Return to the Summary Table.

Publisher port

Figure 29-9. 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 29-4. FPUB_0 Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3-0	CHANID	R/W	0h	0 = disconnected. 1-15 = connected to channelID = CHANID. 0h = A value of 0 specifies that the event is not connected Fh = Consult your device data sheet as the actual allowed maximum may be less than 15.

29.10.3 CLKSEL (Offset = 1004h) [Reset = 00000000h]

CLKSEL is shown in Figure 29-10 and described in Table 29-5.

Return to the Summary Table.

Clock source selection for ULP peripherals

Figure 29-10. 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 29-5. 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 0h = LFCLK is disabled 1h = LFCLK is enabled
0	RESERVED	R	0h	

29.10.4 IIDX (Offset = 1020h) [Reset = 00000000h]

IIDX is shown in Figure 29-11 and described in Table 29-6.

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 29-11. 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 29-6. IIDX Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	STAT	R	0h	Interrupt index status 00h = No interrupt pending 01h = RTC ready 02h = RTC time event 03h = RTC alarm 1 04h = RTC alarm 2 05h = RTC prescale timer 0 06h = RTC prescale timer 1 07h = RTC prescale timer 2 08h = Time stamp event 09h = Tamper I/O 0 event 0Ah = Tamper I/O 1 event 0Bh = Tamper I/O 2 event 0Ch = Tamper I/O 3 event 0Dh = Tamper I/O 4 event 0Eh = Tamper I/O 5 event 0Fh = Tamper I/O 6 event 10h = Tamper I/O 7 event 11h = Tamper I/O 8 event 12h = Tamper I/O 9 event 13h = Tamper I/O 10 event 14h = Tamper I/O 11 event 15h = Tamper I/O 12 event 16h = Tamper I/O 13 event 17h = Tamper I/O 14 event 18h = Tamper I/O 15 event

29.10.5 IMASK (Offset = 1028h) [Reset = 00000000h]

IMASK is shown in Figure 29-12 and described in Table 29-7.

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 29-12. 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 29-7. IMASK Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23	TIO15	R/W	0h	Tamper I/O 15 event 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
22	TIO14	R/W	0h	Tamper I/O 14 event 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
21	TIO13	R/W	0h	Tamper I/O 13 event 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
20	TIO12	R/W	0h	Tamper I/O 12 event 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
19	TIO11	R/W	0h	Tamper I/O 11 event 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
18	TIO10	R/W	0h	Tamper I/O 10 event 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
17	TIO9	R/W	0h	Tamper I/O 9 event 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
16	TIO8	R/W	0h	Tamper I/O 8 event 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
15	TIO7	R/W	0h	Tamper I/O 7 event 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
14	TIO6	R/W	0h	Tamper I/O 6 event 0h = Clear Interrupt Mask 1h = Set Interrupt Mask

Table 29-7. IMASK Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13	TIO5	R/W	0h	Tamper I/O 5 event 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
12	TIO4	R/W	0h	Tamper I/O 4 event 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
11	TIO3	R/W	0h	Tamper I/O 3 event 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
10	TIO2	R/W	0h	Tamper I/O 2 event 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
9	TIO1	R/W	0h	Tamper I/O 1 event 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
8	TIO0	R/W	0h	Tamper I/O 0 event 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
7	TSEVT	R/W	0h	Time stamp event 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
6	RT2PS	R/W	0h	RTC prescale timer 2 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
5	RT1PS	R/W	0h	RTC prescale timer 1 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
4	RT0PS	R/W	0h	RTC prescale timer 0 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
3	RTCA2	R/W	0h	RTC alarm 2 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
2	RTCA1	R/W	0h	RTC alarm 1 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
1	RTCTEV	R/W	0h	RTC time event 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
0	RTCRDY	R/W	0h	RTC ready 0h = Clear Interrupt Mask 1h = Set Interrupt Mask

29.10.6 RIS (Offset = 1030h) [Reset = 00000000h]

RIS is shown in Figure 29-13 and described in Table 29-8.

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 29-13. 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 29-8. RIS Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23	TIO15	R	0h	Tamper I/O 15 event 0h = Interrupt did not occur 1h = Interrupt occurred
22	TIO14	R	0h	Tamper I/O 14 event 0h = Interrupt did not occur 1h = Interrupt occurred
21	TIO13	R	0h	Tamper I/O 13 event 0h = Interrupt did not occur 1h = Interrupt occurred
20	TIO12	R	0h	Tamper I/O 12 event 0h = Interrupt did not occur 1h = Interrupt occurred
19	TIO11	R	0h	Tamper I/O 11 event 0h = Interrupt did not occur 1h = Interrupt occurred
18	TIO10	R	0h	Tamper I/O 10 event 0h = Interrupt did not occur 1h = Interrupt occurred
17	TIO9	R	0h	Tamper I/O 9 event 0h = Interrupt did not occur 1h = Interrupt occurred
16	TIO8	R	0h	Tamper I/O 8 event 0h = Interrupt did not occur 1h = Interrupt occurred
15	TIO7	R	0h	Tamper I/O 7 event 0h = Interrupt did not occur 1h = Interrupt occurred

Table 29-8. RIS Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14	TIO6	R	0h	Tamper I/O 6 event 0h = Interrupt did not occur 1h = Interrupt occurred
13	TIO5	R	0h	Tamper I/O 5 event 0h = Interrupt did not occur 1h = Interrupt occurred
12	TIO4	R	0h	Tamper I/O 4 event 0h = Interrupt did not occur 1h = Interrupt occurred
11	TIO3	R	0h	Tamper I/O 3 event 0h = Interrupt did not occur 1h = Interrupt occurred
10	TIO2	R	0h	Tamper I/O 2 event 0h = Interrupt did not occur 1h = Interrupt occurred
9	TIO1	R	0h	Tamper I/O 1 event 0h = Interrupt did not occur 1h = Interrupt occurred
8	TIO0	R	0h	Tamper I/O 0 event 0h = Interrupt did not occur 1h = Interrupt occurred
7	TSEVT	R	0h	Time stamp event 0h = Interrupt did not occur 1h = Interrupt occurred
6	RT2PS	R	0h	RTC prescale timer 2 0h = Interrupt did not occur 1h = Interrupt occurred
5	RT1PS	R	0h	RTC prescale timer 1 0h = Interrupt did not occur 1h = Interrupt occurred
4	RT0PS	R	0h	RTC prescale timer 0 0h = Interrupt did not occur 1h = Interrupt occurred
3	RTCA2	R	0h	RTC alarm 2 0h = Interrupt did not occur 1h = Interrupt occurred
2	RTCA1	R	0h	RTC alarm 1 0h = Interrupt did not occur 1h = Interrupt occurred
1	RTCTEV	R	0h	RTC time event 0h = Interrupt did not occur 1h = Interrupt occurred
0	RTCRDY	R	0h	RTC ready 0h = Interrupt did not occur 1h = Interrupt occurred

29.10.7 MIS (Offset = 1038h) [Reset = 00000000h]

MIS is shown in Figure 29-14 and described in Table 29-9.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Figure 29-14. 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 29-9. MIS Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23	TIO15	R	0h	Tamper I/O 15 event 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
22	TIO14	R	0h	Tamper I/O 14 event 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
21	TIO13	R	0h	Tamper I/O 13 event 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
20	TIO12	R	0h	Tamper I/O 12 event 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
19	TIO11	R	0h	Tamper I/O 11 event 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
18	TIO10	R	0h	Tamper I/O 10 event 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
17	TIO9	R	0h	Tamper I/O 9 event 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
16	TIO8	R	0h	Tamper I/O 8 event 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
15	TIO7	R	0h	Tamper I/O 7 event 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
14	TIO6	R	0h	Tamper I/O 6 event 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred

Table 29-9. MIS Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13	TIO5	R	0h	Tamper I/O 5 event 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
12	TIO4	R	0h	Tamper I/O 4 event 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
11	TIO3	R	0h	Tamper I/O 3 event 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
10	TIO2	R	0h	Tamper I/O 2 event 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
9	TIO1	R	0h	Tamper I/O 1 event 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
8	TIO0	R	0h	Tamper I/O 0 event 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
7	TSEVT	R	0h	Time stamp event 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
6	RT2PS	R	0h	RTC prescale timer 2 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
5	RT1PS	R	0h	RTC prescale timer 1 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
4	RT0PS	R	0h	RTC prescale timer 0 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
3	RTCA2	R	0h	RTC alarm 2 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
2	RTCA1	R	0h	RTC alarm 1 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
1	RTCTEV	R	0h	RTC time event 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
0	RTCRDY	R	0h	RTC ready 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred

29.10.8 ISET (Offset = 1040h) [Reset = 00000000h]

ISET is shown in Figure 29-15 and described in Table 29-10.

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 29-15. 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 29-10. ISET Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23	TIO15	W	0h	Tamper I/O 15 event 0h = Writing 0 has no effect 1h = Set interrupt
22	TIO14	W	0h	Tamper I/O 14 event 0h = Writing 0 has no effect 1h = Set interrupt
21	TIO13	W	0h	Tamper I/O 13 event 0h = Writing 0 has no effect 1h = Set interrupt
20	TIO12	W	0h	Tamper I/O 12 event 0h = Writing 0 has no effect 1h = Set interrupt
19	TIO11	W	0h	Tamper I/O 11 event 0h = Writing 0 has no effect 1h = Set interrupt
18	TIO10	W	0h	Tamper I/O 10 event 0h = Writing 0 has no effect 1h = Set interrupt
17	TIO9	W	0h	Tamper I/O 9 event 0h = Writing 0 has no effect 1h = Set interrupt
16	TIO8	W	0h	Tamper I/O 8 event 0h = Writing 0 has no effect 1h = Set interrupt
15	TIO7	W	0h	Tamper I/O 7 event 0h = Writing 0 has no effect 1h = Set interrupt

Table 29-10. ISET Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14	TIO6	W	0h	Tamper I/O 6 event 0h = Writing 0 has no effect 1h = Set interrupt
13	TIO5	W	0h	Tamper I/O 5 event 0h = Writing 0 has no effect 1h = Set interrupt
12	TIO4	W	0h	Tamper I/O 4 event 0h = Writing 0 has no effect 1h = Set interrupt
11	TIO3	W	0h	Tamper I/O 3 event 0h = Writing 0 has no effect 1h = Set interrupt
10	TIO2	W	0h	Tamper I/O 2 event 0h = Writing 0 has no effect 1h = Set interrupt
9	TIO1	W	0h	Tamper I/O 1 event 0h = Writing 0 has no effect 1h = Set interrupt
8	TIO0	W	0h	Tamper I/O 0 event 0h = Writing 0 has no effect 1h = Set interrupt
7	TSEVT	W	0h	Time stamp event 0h = Writing 0 has no effect 1h = Set interrupt
6	RT2PS	W	0h	RTC prescale timer 2 0h = Writing 0 has no effect 1h = Set interrupt
5	RT1PS	W	0h	RTC prescale timer 1 0h = Writing 0 has no effect 1h = Set interrupt
4	RT0PS	W	0h	RTC prescale timer 0 0h = Writing 0 has no effect 1h = Set interrupt
3	RTCA2	W	0h	RTC alarm 2 0h = Writing 0 has no effect 1h = Set interrupt
2	RTCA1	W	0h	RTC alarm 1 0h = Writing 0 has no effect 1h = Set interrupt
1	RTCTEV	W	0h	RTC time event 0h = Writing 0 has no effect 1h = Set interrupt
0	RTCRDY	W	0h	RTC ready 0h = Writing 0 has no effect 1h = Set interrupt

29.10.9 ICLR (Offset = 1048h) [Reset = 00000000h]

ICLR is shown in Figure 29-16 and described in Table 29-11.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

Figure 29-16. 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 29-11. ICLR Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23	TIO15	W	0h	Tamper I/O 15 event 0h = Writing 0 has no effect 1h = Clear interrupt
22	TIO14	W	0h	Tamper I/O 14 event 0h = Writing 0 has no effect 1h = Clear interrupt
21	TIO13	W	0h	Tamper I/O 13 event 0h = Writing 0 has no effect 1h = Clear interrupt
20	TIO12	W	0h	Tamper I/O 12 event 0h = Writing 0 has no effect 1h = Clear interrupt
19	TIO11	W	0h	Tamper I/O 11 event 0h = Writing 0 has no effect 1h = Clear interrupt
18	TIO10	W	0h	Tamper I/O 10 event 0h = Writing 0 has no effect 1h = Clear interrupt
17	TIO9	W	0h	Tamper I/O 9 event 0h = Writing 0 has no effect 1h = Clear interrupt
16	TIO8	W	0h	Tamper I/O 8 event 0h = Writing 0 has no effect 1h = Clear interrupt
15	TIO7	W	0h	Tamper I/O 7 event 0h = Writing 0 has no effect 1h = Clear interrupt
14	TIO6	W	0h	Tamper I/O 6 event 0h = Writing 0 has no effect 1h = Clear interrupt

Table 29-11. ICLR Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13	TIO5	W	0h	Tamper I/O 5 event 0h = Writing 0 has no effect 1h = Clear interrupt
12	TIO4	W	0h	Tamper I/O 4 event 0h = Writing 0 has no effect 1h = Clear interrupt
11	TIO3	W	0h	Tamper I/O 3 event 0h = Writing 0 has no effect 1h = Clear interrupt
10	TIO2	W	0h	Tamper I/O 2 event 0h = Writing 0 has no effect 1h = Clear interrupt
9	TIO1	W	0h	Tamper I/O 1 event 0h = Writing 0 has no effect 1h = Clear interrupt
8	TIO0	W	0h	Tamper I/O 0 event 0h = Writing 0 has no effect 1h = Clear interrupt
7	TSEVT	W	0h	Time stamp event 0h = Writing 0 has no effect 1h = Clear interrupt
6	RT2PS	W	0h	RTC prescale timer 2 0h = Writing 0 has no effect 1h = Clear interrupt
5	RT1PS	W	0h	RTC prescale timer 1 0h = Writing 0 has no effect 1h = Clear interrupt
4	RT0PS	W	0h	RTC prescale timer 0 0h = Writing 0 has no effect 1h = Clear interrupt
3	RTCA2	W	0h	RTC alarm 2 0h = Writing 0 has no effect 1h = Clear interrupt
2	RTCA1	W	0h	RTC alarm 1 0h = Writing 0 has no effect 1h = Clear interrupt
1	RTCTEV	W	0h	RTC time event 0h = Writing 0 has no effect 1h = Clear interrupt
0	RTCRDY	W	0h	RTC ready 0h = Writing 0 has no effect 1h = Clear interrupt

29.10.10 IIDX (Offset = 1050h) [Reset = 00000000h]

IIDX is shown in Figure 29-17 and described in Table 29-12.

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 29-17. 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 29-12. IIDX Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	STAT	R	0h	Interrupt index status 00h = No interrupt pending 01h = RTC ready 02h = RTC time event 03h = RTC alarm 1 04h = RTC alarm 2 05h = RTC prescale timer 0 06h = RTC prescale timer 1 07h = RTC prescale timer 2 08h = Time stamp event 09h = Tamper I/O 0 event 0Ah = Tamper I/O 1 event 0Bh = Tamper I/O 2 event 0Ch = Tamper I/O 3 event 0Dh = Tamper I/O 4 event 0Eh = Tamper I/O 5 event 0Fh = Tamper I/O 6 event 10h = Tamper I/O 7 event 11h = Tamper I/O 8 event 12h = Tamper I/O 9 event 13h = Tamper I/O 10 event 14h = Tamper I/O 11 event 15h = Tamper I/O 12 event 16h = Tamper I/O 13 event 17h = Tamper I/O 14 event 18h = Tamper I/O 15 event

29.10.11 IMASK (Offset = 1058h) [Reset = 00000000h]

IMASK is shown in Figure 29-18 and described in Table 29-13.

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 29-18. 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 29-13. IMASK Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23	TIO15	R/W	0h	Tamper I/O 15 event 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
22	TIO14	R/W	0h	Tamper I/O 14 event 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
21	TIO13	R/W	0h	Tamper I/O 13 event 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
20	TIO12	R/W	0h	Tamper I/O 12 event 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
19	TIO11	R/W	0h	Tamper I/O 11 event 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
18	TIO10	R/W	0h	Tamper I/O 10 event 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
17	TIO9	R/W	0h	Tamper I/O 9 event 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
16	TIO8	R/W	0h	Tamper I/O 8 event 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
15	TIO7	R/W	0h	Tamper I/O 7 event 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
14	TIO6	R/W	0h	Tamper I/O 6 event 0h = Clear Interrupt Mask 1h = Set Interrupt Mask

Table 29-13. IMASK Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13	TIO5	R/W	0h	Tamper I/O 5 event 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
12	TIO4	R/W	0h	Tamper I/O 4 event 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
11	TIO3	R/W	0h	Tamper I/O 3 event 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
10	TIO2	R/W	0h	Tamper I/O 2 event 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
9	TIO1	R/W	0h	Tamper I/O 1 event 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
8	TIO0	R/W	0h	Tamper I/O 0 event 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
7	TSEVT	R/W	0h	Time stamp event 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
6	RT2PS	R/W	0h	RTC prescale timer 2 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
5	RT1PS	R/W	0h	RTC prescale timer 1 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
4	RT0PS	R/W	0h	RTC prescale timer 0 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
3	RTCA2	R/W	0h	RTC alarm 2 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
2	RTCA1	R/W	0h	RTC alarm 1 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
1	RTCTEV	R/W	0h	RTC time event 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
0	RTCRDY	R/W	0h	RTC ready 0h = Clear Interrupt Mask 1h = Set Interrupt Mask

29.10.12 RIS (Offset = 1060h) [Reset = 00000000h]

RIS is shown in Figure 29-19 and described in Table 29-14.

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 29-19. 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 29-14. RIS Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23	TIO15	R	0h	Tamper I/O 15 event 0h = Interrupt did not occur 1h = Interrupt occurred
22	TIO14	R	0h	Tamper I/O 14 event 0h = Interrupt did not occur 1h = Interrupt occurred
21	TIO13	R	0h	Tamper I/O 13 event 0h = Interrupt did not occur 1h = Interrupt occurred
20	TIO12	R	0h	Tamper I/O 12 event 0h = Interrupt did not occur 1h = Interrupt occurred
19	TIO11	R	0h	Tamper I/O 11 event 0h = Interrupt did not occur 1h = Interrupt occurred
18	TIO10	R	0h	Tamper I/O 10 event 0h = Interrupt did not occur 1h = Interrupt occurred
17	TIO9	R	0h	Tamper I/O 9 event 0h = Interrupt did not occur 1h = Interrupt occurred
16	TIO8	R	0h	Tamper I/O 8 event 0h = Interrupt did not occur 1h = Interrupt occurred
15	TIO7	R	0h	Tamper I/O 7 event 0h = Interrupt did not occur 1h = Interrupt occurred

Table 29-14. RIS Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14	TIO6	R	0h	Tamper I/O 6 event 0h = Interrupt did not occur 1h = Interrupt occurred
13	TIO5	R	0h	Tamper I/O 5 event 0h = Interrupt did not occur 1h = Interrupt occurred
12	TIO4	R	0h	Tamper I/O 4 event 0h = Interrupt did not occur 1h = Interrupt occurred
11	TIO3	R	0h	Tamper I/O 3 event 0h = Interrupt did not occur 1h = Interrupt occurred
10	TIO2	R	0h	Tamper I/O 2 event 0h = Interrupt did not occur 1h = Interrupt occurred
9	TIO1	R	0h	Tamper I/O 1 event 0h = Interrupt did not occur 1h = Interrupt occurred
8	TIO0	R	0h	Tamper I/O 0 event 0h = Interrupt did not occur 1h = Interrupt occurred
7	TSEVT	R	0h	Time stamp event 0h = Interrupt did not occur 1h = Interrupt occurred
6	RT2PS	R	0h	RTC prescale timer 2 0h = Interrupt did not occur 1h = Interrupt occurred
5	RT1PS	R	0h	RTC prescale timer 1 0h = Interrupt did not occur 1h = Interrupt occurred
4	RT0PS	R	0h	RTC prescale timer 0 0h = Interrupt did not occur 1h = Interrupt occurred
3	RTCA2	R	0h	RTC alarm 2 0h = Interrupt did not occur 1h = Interrupt occurred
2	RTCA1	R	0h	RTC alarm 1 0h = Interrupt did not occur 1h = Interrupt occurred
1	RTCTEV	R	0h	RTC time event 0h = Interrupt did not occur 1h = Interrupt occurred
0	RTCRDY	R	0h	RTC ready 0h = Interrupt did not occur 1h = Interrupt occurred

29.10.13 MIS (Offset = 1068h) [Reset = 00000000h]

MIS is shown in Figure 29-20 and described in Table 29-15.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Figure 29-20. 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 29-15. MIS Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23	TIO15	R	0h	Tamper I/O 15 event 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
22	TIO14	R	0h	Tamper I/O 14 event 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
21	TIO13	R	0h	Tamper I/O 13 event 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
20	TIO12	R	0h	Tamper I/O 12 event 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
19	TIO11	R	0h	Tamper I/O 11 event 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
18	TIO10	R	0h	Tamper I/O 10 event 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
17	TIO9	R	0h	Tamper I/O 9 event 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
16	TIO8	R	0h	Tamper I/O 8 event 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
15	TIO7	R	0h	Tamper I/O 7 event 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
14	TIO6	R	0h	Tamper I/O 6 event 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred

Table 29-15. MIS Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13	TIO5	R	0h	Tamper I/O 5 event 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
12	TIO4	R	0h	Tamper I/O 4 event 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
11	TIO3	R	0h	Tamper I/O 3 event 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
10	TIO2	R	0h	Tamper I/O 2 event 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
9	TIO1	R	0h	Tamper I/O 1 event 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
8	TIO0	R	0h	Tamper I/O 0 event 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
7	TSEVT	R	0h	Time stamp event 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
6	RT2PS	R	0h	RTC prescale timer 2 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
5	RT1PS	R	0h	RTC prescale timer 1 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
4	RT0PS	R	0h	RTC prescale timer 0 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
3	RTCA2	R	0h	RTC alarm 2 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
2	RTCA1	R	0h	RTC alarm 1 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
1	RTCTEV	R	0h	RTC time event 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
0	RTCRDY	R	0h	RTC ready 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred

29.10.14 ISET (Offset = 1070h) [Reset = 00000000h]

ISET is shown in Figure 29-21 and described in Table 29-16.

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 29-21. 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 29-16. ISET Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23	TIO15	W	0h	Tamper I/O 15 event 0h = Writing 0 has no effect 1h = Set interrupt
22	TIO14	W	0h	Tamper I/O 14 event 0h = Writing 0 has no effect 1h = Set interrupt
21	TIO13	W	0h	Tamper I/O 13 event 0h = Writing 0 has no effect 1h = Set interrupt
20	TIO12	W	0h	Tamper I/O 12 event 0h = Writing 0 has no effect 1h = Set interrupt
19	TIO11	W	0h	Tamper I/O 11 event 0h = Writing 0 has no effect 1h = Set interrupt
18	TIO10	W	0h	Tamper I/O 10 event 0h = Writing 0 has no effect 1h = Set interrupt
17	TIO9	W	0h	Tamper I/O 9 event 0h = Writing 0 has no effect 1h = Set interrupt
16	TIO8	W	0h	Tamper I/O 8 event 0h = Writing 0 has no effect 1h = Set interrupt
15	TIO7	W	0h	Tamper I/O 7 event 0h = Writing 0 has no effect 1h = Set interrupt

Table 29-16. ISET Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14	TIO6	W	0h	Tamper I/O 6 event 0h = Writing 0 has no effect 1h = Set interrupt
13	TIO5	W	0h	Tamper I/O 5 event 0h = Writing 0 has no effect 1h = Set interrupt
12	TIO4	W	0h	Tamper I/O 4 event 0h = Writing 0 has no effect 1h = Set interrupt
11	TIO3	W	0h	Tamper I/O 3 event 0h = Writing 0 has no effect 1h = Set interrupt
10	TIO2	W	0h	Tamper I/O 2 event 0h = Writing 0 has no effect 1h = Set interrupt
9	TIO1	W	0h	Tamper I/O 1 event 0h = Writing 0 has no effect 1h = Set interrupt
8	TIO0	W	0h	Tamper I/O 0 event 0h = Writing 0 has no effect 1h = Set interrupt
7	TSEVT	W	0h	Time stamp event 0h = Writing 0 has no effect 1h = Set interrupt
6	RT2PS	W	0h	RTC prescale timer 2 0h = Writing 0 has no effect 1h = Set interrupt
5	RT1PS	W	0h	RTC prescale timer 1 0h = Writing 0 has no effect 1h = Set interrupt
4	RT0PS	W	0h	RTC prescale timer 0 0h = Writing 0 has no effect 1h = Set interrupt
3	RTCA2	W	0h	RTC alarm 2 0h = Writing 0 has no effect 1h = Set interrupt
2	RTCA1	W	0h	RTC alarm 1 0h = Writing 0 has no effect 1h = Set interrupt
1	RTCTEV	W	0h	RTC time event 0h = Writing 0 has no effect 1h = Set interrupt
0	RTCRDY	W	0h	RTC ready 0h = Writing 0 has no effect 1h = Set interrupt

29.10.15 ICLR (Offset = 1078h) [Reset = 00000000h]

ICLR is shown in Figure 29-22 and described in Table 29-17.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

Figure 29-22. 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 29-17. ICLR Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23	TIO15	W	0h	Tamper I/O 15 event 0h = Writing 0 has no effect 1h = Clear interrupt
22	TIO14	W	0h	Tamper I/O 14 event 0h = Writing 0 has no effect 1h = Clear interrupt
21	TIO13	W	0h	Tamper I/O 13 event 0h = Writing 0 has no effect 1h = Clear interrupt
20	TIO12	W	0h	Tamper I/O 12 event 0h = Writing 0 has no effect 1h = Clear interrupt
19	TIO11	W	0h	Tamper I/O 11 event 0h = Writing 0 has no effect 1h = Clear interrupt
18	TIO10	W	0h	Tamper I/O 10 event 0h = Writing 0 has no effect 1h = Clear interrupt
17	TIO9	W	0h	Tamper I/O 9 event 0h = Writing 0 has no effect 1h = Clear interrupt
16	TIO8	W	0h	Tamper I/O 8 event 0h = Writing 0 has no effect 1h = Clear interrupt
15	TIO7	W	0h	Tamper I/O 7 event 0h = Writing 0 has no effect 1h = Clear interrupt
14	TIO6	W	0h	Tamper I/O 6 event 0h = Writing 0 has no effect 1h = Clear interrupt

Table 29-17. ICLR Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13	TIO5	W	0h	Tamper I/O 5 event 0h = Writing 0 has no effect 1h = Clear interrupt
12	TIO4	W	0h	Tamper I/O 4 event 0h = Writing 0 has no effect 1h = Clear interrupt
11	TIO3	W	0h	Tamper I/O 3 event 0h = Writing 0 has no effect 1h = Clear interrupt
10	TIO2	W	0h	Tamper I/O 2 event 0h = Writing 0 has no effect 1h = Clear interrupt
9	TIO1	W	0h	Tamper I/O 1 event 0h = Writing 0 has no effect 1h = Clear interrupt
8	TIO0	W	0h	Tamper I/O 0 event 0h = Writing 0 has no effect 1h = Clear interrupt
7	TSEVT	W	0h	Time stamp event 0h = Writing 0 has no effect 1h = Clear interrupt
6	RT2PS	W	0h	RTC prescale timer 2 0h = Writing 0 has no effect 1h = Clear interrupt
5	RT1PS	W	0h	RTC prescale timer 1 0h = Writing 0 has no effect 1h = Clear interrupt
4	RT0PS	W	0h	RTC prescale timer 0 0h = Writing 0 has no effect 1h = Clear interrupt
3	RTCA2	W	0h	RTC alarm 2 0h = Writing 0 has no effect 1h = Clear interrupt
2	RTCA1	W	0h	RTC alarm 1 0h = Writing 0 has no effect 1h = Clear interrupt
1	RTCTEV	W	0h	RTC time event 0h = Writing 0 has no effect 1h = Clear interrupt
0	RTCRDY	W	0h	RTC ready 0h = Writing 0 has no effect 1h = Clear interrupt

29.10.16 EVT_MODE (Offset = 10E0h) [Reset = 00000000h]

EVT_MODE is shown in Figure 29-23 and described in Table 29-18.

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 29-23. 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 29-18. 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 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 1h = The interrupt or event line is in software mode. The software ISR clears the associated RIS flag.

29.10.17 DESC (Offset = 10FCh) [Reset = 00000000h]

DESC is shown in Figure 29-24 and described in Table 29-19.

Return to the Summary Table.

RTC Descriptor Register

Figure 29-24. 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 29-19. 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 0h = Smallest value 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. 0h = Smallest value Fh = Highest possible value
11-8	INSTNUM	R	0h	Instantiated version. Describes which instance of the module accessed. 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) 0h = Smallest value 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) 0h = Smallest value Fh = Highest possible value

29.10.18 CLKCTL (Offset = 1100h) [Reset = 00000000h]

CLKCTL is shown in Figure 29-25 and described in Table 29-20.

Return to the Summary Table.

RTC Clock Control Register.
This register can be made read only access by the RTCLOCK register.

Figure 29-25. 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 29-20. 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. 0h = 32kHz clock is not supplied to the RTC. 1h = 32kHz clock is supplied to the RTC.
30-0	RESERVED	R	0h	

29.10.19 DBGCTL (Offset = 1104h) [Reset = 00000000h]

DBGCTL is shown in Figure 29-26 and described in Table 29-21.

Return to the Summary Table.

RTC Module Debug Control Register

Figure 29-26. 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 29-21. DBGCTL Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1	DBGINT	R/W	0h	Debug Interrupt Enable. 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. 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. 0h = Counter is halted if CPU is in debug state. 1h = Continue to operate normally ignoring the debug state of the CPU.

29.10.20 CTL (Offset = 1108h) [Reset = 00000000h]

CTL is shown in Figure 29-27 and described in Table 29-22.

Return to the Summary Table.

RTC Control Register

Figure 29-27. 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 29-22. 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. 0h = Binary code selected 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) 0h = Generate RTC event every minute. 1h = Generate RTC event every hour. 2h = Generate RTC event at midnight. 3h = Generate RTC event at noon.

29.10.21 STA (Offset = 110Ch) [Reset = 00000000h]

STA is shown in Figure 29-28 and described in Table 29-23.

Return to the Summary Table.

RTC Status Register

Figure 29-28. 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 29-23. 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. 0h = Write to RTCTCMPx is unsuccessful 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. 0h = RTC temperature compensation in transition 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. 0h = RTC time values in transition 1h = RTC time values safe for reading.

29.10.22 CAL (Offset = 1110h) [Reset = 00000000h]

CAL is shown in Figure 29-29 and described in Table 29-24.

Return to the Summary Table.

RTC Clock Offset Calibration Register

Figure 29-29. 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 29-24. 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. 0h = 32kHz 1h = 512Hz 2h = 256Hz 3h = 1Hz
15	RTCOCALS	R/W	0h	Real-time clock offset error calibration sign. This bit decides the sign of offset error calibration. 0h = Down calibration. Frequency adjusted down. 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. 0h = Smallest value FFh = Highest possible value

29.10.23 TCMP (Offset = 1114h) [Reset = 00000000h]

TCMP is shown in Figure 29-30 and described in Table 29-25.

Return to the Summary Table.

RTC Temperature Compensation Register

Figure 29-30. 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 29-25. 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. 0h = Down calibration. Frequency adjusted down. 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. 00h = Smallest value FFh = Highest possible value

29.10.24 SEC (Offset = 1118h) [Reset = 0000XXXXh]

SEC is shown in Figure 29-31 and described in Table 29-26.

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 29-31. 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 29-26. 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. 0h = Smallest value 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. 0h = Smallest value 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. 00h = Smallest value 3Bh = Highest possible value

29.10.25 MIN (Offset = 111Ch) [Reset = 0000XXXXh]

MIN is shown in Figure 29-32 and described in Table 29-27.

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 29-32. 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 29-27. 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. 0h = Smallest value 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. 0h = Smallest value 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. 00h = Smallest value 3Bh = Highest possible value

29.10.26 HOUR (Offset = 1120h) [Reset = 0000XXXXh]

HOUR is shown in Figure 29-33 and described in Table 29-28.

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 29-33. 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 29-28. 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. 0h = Smallest value 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. 0h = Smallest value 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. 00h = Smallest value 17h = Highest possible value

29.10.27 DAY (Offset = 1124h) [Reset = 00XXXX0Xh]

DAY is shown in Figure 29-34 and described in Table 29-29.

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 29-34. 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 29-29. 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. 0h = Smallest value 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. 0h = Smallest value 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. 00h = Smallest value 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. 0h = Smallest value 6h = Highest possible value

29.10.28 MON (Offset = 1128h) [Reset = 0000XX0Xh]

MON is shown in Figure 29-35 and described in Table 29-30.

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 29-35. 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 29-30. 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. 0h = Smallest value 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. 0h = Smallest value 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. 0h = Smallest value Ch = Highest possible value

29.10.29 YEAR (Offset = 112Ch) [Reset = XXXX0XXXh]

YEAR is shown in Figure 29-36 and described in Table 29-31.

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 29-36. 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 29-31. 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. 0h = Smallest value 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. 0h = Smallest value 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. 0h = Smallest value 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. 0h = Smallest value 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. 0h = Smallest value 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. 00h = Smallest value FFh = Highest possible value

29.10.30 A1MIN (Offset = 1130h) [Reset = 0000XXXXh]

A1MIN is shown in Figure 29-37 and described in Table 29-32.

Return to the Summary Table.

RTC Minutes Alarm Register - Calendar Mode With Binary / BCD Format

Figure 29-37. 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-X			R/W-X			
7	6	5	4	3	2	1	0
AMINAEBIN	RESERVED	AMINBIN					
R/W-0h	R-0h	R/W-X					

Table 29-32. 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. 0h = No alarm 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. 0h = Smallest value 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. 0h = Smallest value 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. 0h = No alarm 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. 00h = Smallest value 3Bh = Highest possible value

29.10.31 A1HOUR (Offset = 1134h) [Reset = 0000XX0Xh]

A1HOUR is shown in Figure 29-38 and described in Table 29-33.

Return to the Summary Table.

RTC Hours Alarm Register - Calendar Mode With Binary / BCD Format

Figure 29-38. 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-X		R/W-X			
7	6	5	4	3	2	1	0
AHOURAEBIN	RESERVED		AHOURBIN				
R/W-0h	R-0h		R/W-X				

Table 29-33. 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. 0h = No alarm 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.. 0h = Smallest value 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. 0h = Smallest value 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. 0h = No alarm 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. 0h = Smallest value 17h = Highest possible value

29.10.32 A1DAY (Offset = 1138h) [Reset = 00XXXX0Xh]

A1DAY is shown in Figure 29-39 and described in Table 29-34.

Return to the Summary Table.

RTC Alarm Day Of Week / Month Register - Calendar Mode With Binary / BCD Format

Figure 29-39. 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-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 29-34. 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. 0h = No alarm 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. 0h = Smallest value 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. 0h = Smallest value 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. 0h = No alarm 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. 00h = Smallest value 1Fh = Highest possible value
7	ADOWAE	R/W	0h	Alarm Day of week enable. This bit are valid if RTCBCD=1 or RTCBCD=0. 0h = No alarm 1h = Alarm enabled
6-3	RESERVED	R	0h	

Table 29-34. 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. 0h = Smallest value 6h = Highest possible value

29.10.33 A2MIN (Offset = 113Ch) [Reset = 0000XXXXh]

A2MIN is shown in Figure 29-40 and described in Table 29-35.

Return to the Summary Table.

RTC Minutes Alarm Register - Calendar Mode With Binary / BCD Format

Figure 29-40. 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-X			R/W-X			
7	6	5	4	3	2	1	0
AMINAEBIN	RESERVED	AMINBIN					
R/W-0h	R-0h	R/W-X					

Table 29-35. 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. 0h = No alarm 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. 0h = Smallest value 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. 0h = Smallest value 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. 0h = No alarm 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. 00h = Smallest value 3Bh = Highest possible value

29.10.34 A2HOUR (Offset = 1140h) [Reset = 0000XX0Xh]

A2HOUR is shown in Figure 29-41 and described in Table 29-36.

Return to the Summary Table.

RTC Hours Alarm Register - Calendar Mode With Binary / BCD Format

Figure 29-41. 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-X		R/W-X			
7	6	5	4	3	2	1	0
AHOURAEBIN	RESERVED		AHOURBIN				
R/W-0h	R-0h		R/W-X				

Table 29-36. 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. 0h = No alarm 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.. 0h = Smallest value 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. 0h = Smallest value 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. 0h = No alarm 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. 0h = Smallest value 17h = Highest possible value

29.10.35 A2DAY (Offset = 1144h) [Reset = 00XXXX0Xh]

A2DAY is shown in Figure 29-42 and described in Table 29-37.

Return to the Summary Table.

RTC Alarm Day Of Week / Month Register - Calendar Mode With Binary / BCD Format

Figure 29-42. 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-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 29-37. 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. 0h = No alarm 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. 0h = Smallest value 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. 0h = Smallest value 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. 0h = No alarm 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. 00h = Smallest value 1Fh = Highest possible value
7	ADOWAE	R/W	0h	Alarm Day of week enable. This bit are valid if RTCBCD=1 or RTCBCD=0. 0h = No alarm 1h = Alarm enabled
6-3	RESERVED	R	0h	

Table 29-37. 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. 0h = Smallest value 6h = Highest possible value

29.10.36 PSCTL (Offset = 1148h) [Reset = 00000008h]

PSCTL is shown in Figure 29-43 and described in Table 29-38.

Return to the Summary Table.

RTC Prescale Timer Control Register

Figure 29-43. 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 29-38. PSCTL Field Descriptions

Bit	Field	Type	Reset	Description
31-21	RESERVED	R	0h	
20-18	RT1IP	R/W	0h	Prescale timer 1 interrupt interval 0h = Interval every 15.6 millisecond 1h = Interval every 31.3 millisecond 2h = Interval every 62.5 millisecond 3h = Interval every 125 millisecond 4h = Interval every 250 millisecond 5h = Interval every 500 millisecond 6h = Interval every 1 second 7h = Interval every 2 second
17-5	RESERVED	R	0h	
4-2	RT0IP	R/W	2h	Prescale timer 0 interrupt interval 2h = Interval every 244 microsecond 3h = Interval every 488 microsecond 4h = Interval every 0.98 millisecond 5h = Interval every 1.95 millisecond 6h = Interval every 3.91 millisecond 7h = Interval every 7.81 millisecond
1-0	RESERVED	R	0h	

29.10.37 EXTPSCTL (Offset = 114Ch) [Reset = 00000000h]

EXTPSCTL is shown in Figure 29-44 and described in Table 29-39.

Return to the Summary Table.

Extended Prescale Timer Control Register

Figure 29-44. 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 29-39. EXTPSCTL Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3-2	RT2PS	R/W	0h	Prescale timer 2 interrupt interval 0h = Interval every 4 second 1h = Interval every 8 second 2h = Interval every 16 second
1-0	RESERVED	R	0h	

29.10.38 TSSEC (Offset = 1150h) [Reset = 00000000h]

TSSEC is shown in Figure 29-45 and described in Table 29-40.

Return to the Summary Table.

RTC Second Time Stamp Capture - Calendar Mode With Binary / BCD Format

Figure 29-45. 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 29-40. 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. 0h = Smallest value 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. 0h = Smallest value 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. 00h = Smallest value 3Bh = Highest possible value

29.10.39 TSMIN (Offset = 1154h) [Reset = 00000000h]

TSMIN is shown in Figure 29-46 and described in Table 29-41.

Return to the Summary Table.

RTC Minutes Time Stamp Capture - Calendar Mode With Binary / BCD Format

Figure 29-46. 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 29-41. 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. 0h = Smallest value 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. 0h = Smallest value 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. 0h = Smallest value 3Bh = Highest possible value

29.10.40 TSHOUR (Offset = 1158h) [Reset = 00000000h]

TSHOUR is shown in Figure 29-47 and described in Table 29-42.

Return to the Summary Table.

RTC Hours Time Stamp Capture - Calendar Mode With Binary / BCD Format

Figure 29-47. 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 29-42. 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. 00h = Smallest value 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. 00h = Smallest value 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. 00h = Smallest value 17h = Highest possible value

29.10.41 TSDAY (Offset = 115Ch) [Reset = 00000000h]

TSDAY is shown in Figure 29-48 and described in Table 29-43.

Return to the Summary Table.

RTC Day Of Week / Month Time Stamp Capture - Calendar Mode With Binary / BCD Format

Figure 29-48. 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 29-43. 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. 0h = Smallest value 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. 0h = Smallest value 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. 00h = Smallest value 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. 0h = Smallest value 6h = Highest possible value

29.10.42 TSMON (Offset = 1160h) [Reset = 00000000h]

TSMON is shown in Figure 29-49 and described in Table 29-44.

Return to the Summary Table.

RTC Month Time Stamp Capture - Calendar Mode With Binary / BCD Format

Figure 29-49. 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 29-44. 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. 0h = Smallest value 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. 0h = Smallest value 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. 0h = Smallest value Ch = Highest possible value

29.10.43 TSYEAR (Offset = 1164h) [Reset = 00000000h]

TSYEAR is shown in Figure 29-50 and described in Table 29-45.

Return to the Summary Table.

RTC Years Time Stamp Capture - Calendar Mode With Binary / BCD Format

Figure 29-50. 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 29-45. 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. 0h = Smallest value 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. 0h = Smallest value 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. 0h = Smallest value 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. 0h = Smallest value 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. 0h = Smallest value 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. 00h = Smallest value FFh = Highest possible value

29.10.44 TSSTAT (Offset = 1168h) [Reset = 00000000h]

TSSTAT is shown in Figure 29-51 and described in Table 29-46.

Return to the Summary Table.

Time Stamp Status

Figure 29-51. 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 29-46. TSSTAT Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	
16	TSVDDEVT	R	0h	Loss of VDD caused time stamp event 0h = no event detected 1h = event detected
15	TSTIOEVT15	R	0h	Tamper I/O 15 caused time stamp event 0h = no event detected 1h = event detected
14	TSTIOEVT14	R	0h	Tamper I/O 14 caused time stamp event 0h = no event detected 1h = event detected
13	TSTIOEVT13	R	0h	Tamper I/O 13 caused time stamp event 0h = no event detected 1h = event detected
12	TSTIOEVT12	R	0h	Tamper I/O 12 caused time stamp event 0h = no event detected 1h = event detected
11	TSTIOEVT11	R	0h	Tamper I/O 11 caused time stamp event 0h = no event detected 1h = event detected
10	TSTIOEVT10	R	0h	Tamper I/O 10 caused time stamp event 0h = no event detected 1h = event detected
9	TSTIOEVT9	R	0h	Tamper I/O 9 caused time stamp event 0h = no event detected 1h = event detected
8	TSTIOEVT8	R	0h	Tamper I/O 8 caused time stamp event 0h = no event detected 1h = event detected
7	TSTIOEVT7	R	0h	Tamper I/O 7 caused time stamp event 0h = no event detected 1h = event detected

Table 29-46. TSSTAT Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	TSTIOEVT6	R	0h	Tamper I/O 6 caused time stamp event 0h = no event detected 1h = event detected
5	TSTIOEVT5	R	0h	Tamper I/O 5 caused time stamp event 0h = no event detected 1h = event detected
4	TSTIOEVT4	R	0h	Tamper I/O 4 caused time stamp event 0h = no event detected 1h = event detected
3	TSTIOEVT3	R	0h	Tamper I/O 3 caused time stamp event 0h = no event detected 1h = event detected
2	TSTIOEVT2	R	0h	Tamper I/O 2 caused time stamp event 0h = no event detected 1h = event detected
1	TSTIOEVT1	R	0h	Tamper I/O 1 caused time stamp event 0h = no event detected 1h = event detected
0	TSTIOEVT0	R	0h	Tamper I/O 0 caused time stamp event 0h = no event detected 1h = event detected

29.10.45 TSCTL (Offset = 116Ch) [Reset = 00000000h]

TSCTL is shown in Figure 29-52 and described in Table 29-47.

Return to the Summary Table.

time stamp control register

Figure 29-52. 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 29-47. TSCTL Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	R-0/W	0h	need to write (KEY=0xC5) to update this register 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. KEY must be set to C5h to write to this bit. 0h = Time stamp holds RTC capture at first occurrence of time stamp event. 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 KEY must be set to C5h to write to this bit. 0h = function disabled 1h = function enabled
15	TSTIOEN15	R/WK	0h	Time Stamp by Tamper I/O 15 enable KEY must be set to C5h to write to this bit. 0h = function disabled 1h = function enabled
14	TSTIOEN14	R/WK	0h	Time Stamp by Tamper I/O 14 enable KEY must be set to C5h to write to this bit. 0h = function disabled 1h = function enabled
13	TSTIOEN13	R/WK	0h	Time Stamp by Tamper I/O 13 enable KEY must be set to C5h to write to this bit. 0h = function disabled 1h = function enabled

Table 29-47. TSCTL Field Descriptions (continued)

Bit	Field	Type	Reset	Description
12	TSTIOEN12	R/WK	0h	Time Stamp by Tamper I/O 12 enable KEY must be set to C5h to write to this bit. 0h = function disabled 1h = function enabled
11	TSTIOEN11	R/WK	0h	Time Stamp by Tamper I/O 11 enable KEY must be set to C5h to write to this bit. 0h = function disabled 1h = function enabled
10	TSTIOEN10	R/WK	0h	Time Stamp by Tamper I/O 10 enable KEY must be set to C5h to write to this bit. 0h = function disabled 1h = function enabled
9	TSTIOEN9	R/WK	0h	Time Stamp by Tamper I/O 9 enable KEY must be set to C5h to write to this bit. 0h = function disabled 1h = function enabled
8	TSTIOEN8	R/WK	0h	Time Stamp by Tamper I/O 8 enable KEY must be set to C5h to write to this bit. 0h = function disabled 1h = function enabled
7	TSTIOEN7	R/WK	0h	Time Stamp by Tamper I/O 7 enable KEY must be set to C5h to write to this bit. 0h = function disabled 1h = function enabled
6	TSTIOEN6	R/WK	0h	Time Stamp by Tamper I/O 6 enable KEY must be set to C5h to write to this bit. 0h = function disabled 1h = function enabled
5	TSTIOEN5	R/WK	0h	Time Stamp by Tamper I/O 5 enable KEY must be set to C5h to write to this bit. 0h = function disabled 1h = function enabled
4	TSTIOEN4	R/WK	0h	Time Stamp by Tamper I/O 4 enable KEY must be set to C5h to write to this bit. 0h = function disabled 1h = function enabled
3	TSTIOEN3	R/WK	0h	Time Stamp by Tamper I/O 3 enable KEY must be set to C5h to write to this bit. 0h = function disabled 1h = function enabled
2	TSTIOEN2	R/WK	0h	Time Stamp by Tamper I/O 2 enable KEY must be set to C5h to write to this bit. 0h = function disabled 1h = function enabled
1	TSTIOEN1	R/WK	0h	Time Stamp by Tamper I/O 1 enable KEY must be set to C5h to write to this bit. 0h = function disabled 1h = function enabled
0	TSTIOEN0	R/WK	0h	Time Stamp by Tamper I/O 0 enable KEY must be set to C5h to write to this bit. 0h = function disabled 1h = function enabled

29.10.46 TSCLR (Offset = 1170h) [Reset = 00000000h]

TSCLR is shown in Figure 29-53 and described in Table 29-48.

Return to the Summary Table.

time stamp clear register

Figure 29-53. 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 29-48. TSCLR Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	R-0/W	0h	need to write (KEY=0xE2) to update this register 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. KEY must be set to E2h to write to this bit. 0h = Writing 0 has no effect 1h = clear time stamp event

29.10.47 LFSSRST (Offset = 11F0h) [Reset = 00000000h]

LFSSRST is shown in Figure 29-54 and described in Table 29-49.

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 29-54. 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 29-49. LFSSRST Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	R-0/W	0h	need to write (KEY=0x12) to update this register 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. KEY must be set to 12h to write to this bit. 0h = Writing this value has no effect. 1h = Request power on reset to the LFSS.

29.10.48 RTCLOCK (Offset = 11FCh) [Reset = 00000000h]

RTCLOCK is shown in Figure 29-55 and described in Table 29-50.

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 29-55. 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 29-50. RTCLOCK Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	R-0/W	0h	need to write (KEY=0x22) to update this register 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. KEY must be set to 22h to write to this bit. 0h = RTC counter is writable. 1h = RTC counter is read only access.

29.10.49 TIOCTL[y] (Offset = 1200h + formula) [Reset = 00000000h]

TIOCTL[y] is shown in Figure 29-56 and described in Table 29-51.

Return to the Summary Table.

tamper I/O control register

Offset = 1200h + (y * 4h); where y = 0h to Fh

Figure 29-56. 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 29-51. TIOCTL[y] Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	
19	OUTINV	R/W	0h	Output inversion enable 0h = The output inversion is disabled. 1h = The output inversion is enabled.
18	INENA	R/W	0h	input enable 0h = The input path is disabled. 1h = The input path is enabled.
17	PIPU	R/W	0h	pull up enable 0h = The pull-up function is disabled. 1h = The pull-up function is enabled.
16	PIPD	R/W	0h	pull down enable 0h = The pull-down function is disabled. 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 0h = no filter on the tamper I/O beyond CDC synchronization sample 1h = 1 FLCLK minimum sample (30us) 2h = 3 LFCLK minimum sample (100us) 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 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
7-6	RESERVED	R	0h	

Table 29-51. TIOCTL[y] Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5-4	TOUTSEL	Rmodify/W	0h	Selects the source for TOUT control 0h = The TOUT register is the source for TOUT 1h = The LFCLK is the source for TOUT 2h = The heart beat generator is the source for TOUT 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 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. 1h = The tamper I/O is controlled by the TIOCTL register and stays functional during loss of the main supply (VDD).

29.10.50 TOUT3_0 (Offset = 1280h) [Reset = 00000000h]

TOUT3_0 is shown in Figure 29-57 and described in Table 29-52.

Return to the Summary Table.

Tamper I/O output for pins configured as TIO3 to TIO0.

Figure 29-57. 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 29-52. 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 1h = Output is set to 1

29.10.51 TOUT7_4 (Offset = 1284h) [Reset = 00000000h]

TOUT7_4 is shown in Figure 29-58 and described in Table 29-53.

Return to the Summary Table.

Tamper I/O output for pins configured as TIO7 to TIO4.

Figure 29-58. 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 29-53. 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 1h = Output is set to 1

29.10.52 TOUT11_8 (Offset = 1288h) [Reset = 00000000h]

TOUT11_8 is shown in Figure 29-59 and described in Table 29-54.

Return to the Summary Table.

Tamper I/O output for pins configured as TIO11 to TIO8.

Figure 29-59. 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 29-54. 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 1h = Output is set to 1

29.10.53 TOUT15_12 (Offset = 128Ch) [Reset = 00000000h]

TOUT15_12 is shown in Figure 29-60 and described in Table 29-55.

Return to the Summary Table.

Tamper I/O output for pins configured as TIO15 to TIO12.

Figure 29-60. 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 29-55. 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 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. 0h = Output is set to 0 1h = Output is set to 1

29.10.54 TOE3_0 (Offset = 1290h) [Reset = 00000000h]

TOE3_0 is shown in Figure 29-61 and described in Table 29-56.

Return to the Summary Table.

Tamper I/O output enable for pins configured as TIO3 to TIO0.

Figure 29-61. 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 29-56. 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 0h = output disabled 1h = output enabled
23-17	RESERVED	R	0h	
16	TIO2	R/W	0h	Enables data output for tamper I/O 2 0h = output disabled 1h = output enabled
15-9	RESERVED	R	0h	
8	TIO1	R/W	0h	Enables data output for tamper I/O 1 0h = output disabled 1h = output enabled
7-1	RESERVED	R	0h	
0	TIO0	R/W	0h	Enables data output for tamper I/O 0 0h = output disabled 1h = output enabled

29.10.55 TOE7_4 (Offset = 1294h) [Reset = 00000000h]

TOE7_4 is shown in Figure 29-62 and described in Table 29-57.

Return to the Summary Table.

Tamper I/O output enable for pins configured as TIO7 to TIO4.

Figure 29-62. 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 29-57. 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 0h = output disabled 1h = output enabled
23-17	RESERVED	R	0h	
16	TIO6	R/W	0h	Enables data output for tamper I/O 6 0h = output disabled 1h = output enabled
15-9	RESERVED	R	0h	
8	TIO5	R/W	0h	Enables data output for tamper I/O 5 0h = output disabled 1h = output enabled
7-1	RESERVED	R	0h	
0	TIO4	R/W	0h	Enables data output for tamper I/O 4 0h = output disabled 1h = output enabled

29.10.56 TOE11_8 (Offset = 1298h) [Reset = 00000000h]

TOE11_8 is shown in Figure 29-63 and described in Table 29-58.

Return to the Summary Table.

Tamper I/O output enable for pins configured as TIO11 to TIO8.

Figure 29-63. 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 29-58. 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 0h = output disabled 1h = output enabled
23-17	RESERVED	R	0h	
16	TIO10	R/W	0h	Enables data output for tamper I/O 10 0h = output disabled 1h = output enabled
15-9	RESERVED	R	0h	
8	TIO9	R/W	0h	Enables data output for tamper I/O 9 0h = output disabled 1h = output enabled
7-1	RESERVED	R	0h	
0	TIO8	R/W	0h	Enables data output for tamper I/O 8 0h = output disabled 1h = output enabled

29.10.57 TOE15_12 (Offset = 129Ch) [Reset = 00000000h]

TOE15_12 is shown in Figure 29-64 and described in Table 29-59.

Return to the Summary Table.

Tamper I/O output enable for pins configured as TIO15 to TIO12.

Figure 29-64. 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 29-59. 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 0h = output disabled 1h = output enabled
23-17	RESERVED	R	0h	
16	TIO14	R/W	0h	Enables data output for tamper I/O 14 0h = output disabled 1h = output enabled
15-9	RESERVED	R	0h	
8	TIO13	R/W	0h	Enables data output for tamper I/O 13 0h = output disabled 1h = output enabled
7-1	RESERVED	R	0h	
0	TIO12	R/W	0h	Enables data output for tamper I/O 12 0h = output disabled 1h = output enabled

29.10.58 TIN3_0 (Offset = 12A0h) [Reset = 00000000h]

TIN3_0 is shown in Figure 29-65 and described in Table 29-60.

Return to the Summary Table.

Tamper I/O inputs for pins configured as TIO3 to TIO0.

Figure 29-65. 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 29-60. 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. 0h = Input value is 0 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. 0h = Input value is 0 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. 0h = Input value is 0 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. 0h = Input value is 0 1h = Input value is 1

29.10.59 TIN7_4 (Offset = 12A4h) [Reset = 00000000h]

TIN7_4 is shown in Figure 29-66 and described in Table 29-61.

Return to the Summary Table.

Tamper I/O inputs for pins configured as TIO7 to TIO4.

Figure 29-66. 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 29-61. 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. 0h = Input value is 0 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. 0h = Input value is 0 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. 0h = Input value is 0 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. 0h = Input value is 0 1h = Input value is 1

29.10.60 TIN11_8 (Offset = 12A8h) [Reset = 00000000h]

TIN11_8 is shown in Figure 29-67 and described in Table 29-62.

Return to the Summary Table.

Tamper I/O inputs for pins configured as TIO11 to TIO8.

Figure 29-67. 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 29-62. 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. 0h = Input value is 0 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. 0h = Input value is 0 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. 0h = Input value is 0 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. 0h = Input value is 0 1h = Input value is 1

29.10.61 TIN15_12 (Offset = 12ACh) [Reset = 00000000h]

TIN15_12 is shown in Figure 29-68 and described in Table 29-63.

Return to the Summary Table.

Tamper I/O inputs for pins configured as TIO15 to TIO12.

Figure 29-68. 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 29-63. 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. 0h = Input value is 0 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. 0h = Input value is 0 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. 0h = Input value is 0 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. 0h = Input value is 0 1h = Input value is 1

29.10.62 HEARTBEAT (Offset = 12C0h) [Reset = 00000000h]

HEARTBEAT is shown in Figure 29-69 and described in Table 29-64.

Return to the Summary Table.

The configuration register for the heart beat generator

Figure 29-69. 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 29-64. HEARTBEAT Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R	0h	
17-16	HBMODE	R/W	0h	Heart beat mode 0h = Heart beat disabled 1h = Heart beat always enabled 2h = Heart beat enabled when time stamp event detected 3h = Heart beat when VDD has fail condition
15-11	RESERVED	R	0h	
10-8	HBWIDTH	R/W	0h	Heart beat interval width 0h = Heart beat pulse width 1msec 1h = Heart beat pulse width 2msec 2h = Heart beat pulse width 4msec 3h = Heart beat pulse width 8msec 4h = Heart beat pulse width 16msec 5h = Heart beat pulse width 32msec 6h = Heart beat pulse width 64msec 7h = Heart beat pulse width 128msec
7-3	RESERVED	R	0h	
2-0	HBINTERVAL	R/W	0h	Heart beat interval 0h = Heart beat interval 0.125 sec 1h = Heart beat interval 0.25 sec 2h = Heart beat interval 0.5 sec 3h = Heart beat interval 1 sec 4h = Heart beat interval 2 sec 5h = Heart beat interval 4 sec 6h = Heart beat interval 8 sec 7h = Heart beat interval 16 sec

29.10.63 TIOLOCK (Offset = 12FCh) [Reset = 00000000h]

TIOLOCK is shown in Figure 29-70 and described in Table 29-65.

Return to the Summary Table.

The TIO lock bit protects the TIOCTL and HEARTBEAT registers from accidental updates.

Figure 29-70. 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 29-65. TIOLOCK Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	R-0/W	0h	need to write (KEY=0x18) to update this register 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. KEY must be set to 18h to write to this bit. 0h = Tamper I/O control is writable. 1h = Tamper I/O control is read only access.

29.10.64 WDTEN (Offset = 1300h) [Reset = 00000000h]

WDTEN is shown in Figure 29-71 and described in Table 29-66.

Return to the Summary Table.

Watchdog Timer Enable Register.
This register can be made read only access by the WDTLOCK register.

Figure 29-71. 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 29-66. 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. 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. KEY must be set to EEh to write to this bit. 0h = Disable WDT 1h = Enable WDT

29.10.65 WDTDBGCTL (Offset = 1304h) [Reset = 00000000h]

WDTDBGCTL is shown in Figure 29-72 and described in Table 29-67.

Return to the Summary Table.

Watchdog Timer Debug Control

Figure 29-72. 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 29-67. WDTDBGCTL Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	FREE	R/W	0h	Free run control 0h = The WDT freezes functionality while the CPU is Halted during debug and resumes when the CPU is active. 1h = The WDT ignores the state of the CPU Halted debug state.

29.10.66 WDTCTL (Offset = 1308h) [Reset = 00000043h]

WDTCTL is shown in Figure 29-73 and described in Table 29-68.

Return to the Summary Table.

Watchdog Timer Control Register.
This register can be made read only access by the WDTLOCK register.

Figure 29-73. 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 29-68. 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. 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. KEY must be set to C6h to write to this bit. 0h = Total timer count is 2^{25} 1h = Total timer count is 2^{21} 2h = Total timer count is 2^{18} 3h = Total timer count is 2^{15} 4h = Total timer count is 2^{12} (default) 5h = Total timer count is 2^{10} 6h = Total timer count is 2^{8} 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. KEY must be set to C6h to write to this bit. 0h = Minimum value 7h = Maximum value

29.10.67 WDTCNTRST (Offset = 130Ch) [Reset = 00000000h]

WDTCNTRST is shown in Figure 29-74 and described in Table 29-69.

Return to the Summary Table.

Watchdog Timer Counter Reset Register

Figure 29-74. 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 29-69. 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. 0h = Minimum value 000003A7h = VALUE to restart the WDT counter FFFFFFFFh = Maximum value

29.10.68 WDTSTAT (Offset = 1310h) [Reset = 00000000h]

WDTSTAT is shown in Figure 29-75 and described in Table 29-70.

Return to the Summary Table.

Watchdog Timer Status Register

Figure 29-75. 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 29-70. WDTSTAT Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	RUN	R	0h	Watchdog running status flag. 0h = Watchdog counter stopped. 1h = Watchdog running.

29.10.69 WDTLOCK (Offset = 13FCh) [Reset = 00000000h]

WDTLOCK is shown in Figure 29-76 and described in Table 29-71.

Return to the Summary Table.

The WDT lock bit protects the WDTEN and WDTCTL registers from accidental updates.

Figure 29-76. 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 29-71. WDTLOCK Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	R-0/W	0h	need to write (KEY=0xBD) to update this register 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. KEY must be set to BDh to write to this bit. 0h = Watchdog timer control is writable. 1h = Watchdog timer control is read only access.

29.10.70 SPMEM[y] (Offset = 1400h + formula) [Reset = 00000000h]

SPMEM[y] is shown in Figure 29-77 and described in Table 29-72.

Return to the Summary Table.

Scratch pad memory

Offset = 1400h + (y * 4h); where y = 0h to 1Fh

Figure 29-77. 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 29-72. SPMEM[y] Field Descriptions

Bit	Field	Type	Reset	Description
31-24	DATA3	R/W	0h	memory data byte 3 0h = Smallest value FFh = Highest possible value
23-16	DATA2	R/W	0h	memory data byte 2 0h = Smallest value FFh = Highest possible value
15-8	DATA1	R/W	0h	memory data byte 1 0h = Smallest value FFh = Highest possible value
7-0	DATA0	R/W	0h	memory data byte 0 0h = Smallest value FFh = Highest possible value

29.10.71 SPMWPROT0 (Offset = 1500h) [Reset = 00000000h]

SPMWPROT0 is shown in Figure 29-78 and described in Table 29-73.

Return to the Summary Table.

Scratch pad memory write protect 0

Figure 29-78. 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 29-73. SPMWPROT0 Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	R-0/W	0h	need to write (KEY=0xE8) to update this register 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 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
14	WP_3_2	R/WK	0h	write protect SPMEM3 - DATA2 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
13	WP_3_1	R/WK	0h	write protect SPMEM3 - DATA1 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
12	WP_3_0	R/WK	0h	write protect SPMEM3 - DATA0 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
11	WP_2_3	R/WK	0h	write protect SPMEM2 - DATA3 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
10	WP_2_2	R/WK	0h	write protect SPMEM2 - DATA2 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access

Table 29-73. SPMWPROT0 Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	WP_2_1	R/WK	0h	write protect SPMEM2 - DATA1 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
8	WP_2_0	R/WK	0h	write protect SPMEM2 - DATA0 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
7	WP_1_3	R/WK	0h	write protect SPMEM1 - DATA3 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
6	WP_1_2	R/WK	0h	write protect SPMEM1 - DATA2 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
5	WP_1_1	R/WK	0h	write protect SPMEM1 - DATA1 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
4	WP_1_0	R/WK	0h	write protect SPMEM1 - DATA0 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
3	WP_0_3	R/WK	0h	write protect SPMEM0 - DATA3 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
2	WP_0_2	R/WK	0h	write protect SPMEM0 - DATA2 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
1	WP_0_1	R/WK	0h	write protect SPMEM0 - DATA1 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
0	WP_0_0	R/WK	0h	write protect SPMEM0 - DATA0 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access

29.10.72 SPMWPROT1 (Offset = 1504h) [Reset = 00000000h]

SPMWPROT1 is shown in Figure 29-79 and described in Table 29-74.

Return to the Summary Table.

Scratch pad memory write protect 1

Figure 29-79. 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 29-74. SPMWPROT1 Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	R-0/W	0h	need to write (KEY=0xE8) to update this register 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 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
14	WP_7_2	R/WK	0h	write protect SPMEM7 - DATA2 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
13	WP_7_1	R/WK	0h	write protect SPMEM7 - DATA1 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
12	WP_7_0	R/WK	0h	write protect SPMEM7 - DATA0 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
11	WP_6_3	R/WK	0h	write protect SPMEM6 - DATA3 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
10	WP_6_2	R/WK	0h	write protect SPMEM6 - DATA2 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access

Table 29-74. SPMWPROT1 Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	WP_6_1	R/WK	0h	write protect SPMEM6 - DATA1 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
8	WP_6_0	R/WK	0h	write protect SPMEM6 - DATA0 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
7	WP_5_3	R/WK	0h	write protect SPMEM5 - DATA3 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
6	WP_5_2	R/WK	0h	write protect SPMEM5 - DATA2 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
5	WP_5_1	R/WK	0h	write protect SPMEM5 - DATA1 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
4	WP_5_0	R/WK	0h	write protect SPMEM5 - DATA0 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
3	WP_4_3	R/WK	0h	write protect SPMEM4 - DATA3 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
2	WP_4_2	R/WK	0h	write protect SPMEM4 - DATA2 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
1	WP_4_1	R/WK	0h	write protect SPMEM4 - DATA1 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
0	WP_4_0	R/WK	0h	write protect SPMEM4 - DATA0 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access

29.10.73 SPMWPROT2 (Offset = 1508h) [Reset = 00000000h]

SPMWPROT2 is shown in Figure 29-80 and described in Table 29-75.

Return to the Summary Table.

Scratch pad memory write protect 2

Figure 29-80. 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 29-75. SPMWPROT2 Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	R-0/W	0h	need to write (KEY=0xE8) to update this register 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 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
14	WP_11_2	R/WK	0h	write protect SPMEM11 - DATA2 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
13	WP_11_1	R/WK	0h	write protect SPMEM11 - DATA1 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
12	WP_11_0	R/WK	0h	write protect SPMEM11 - DATA0 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
11	WP_10_3	R/WK	0h	write protect SPMEM10 - DATA3 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
10	WP_10_2	R/WK	0h	write protect SPMEM610- DATA2 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access

Table 29-75. SPMWPROT2 Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	WP_10_1	R/WK	0h	write protect SPMEM10 - DATA1 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
8	WP_10_0	R/WK	0h	write protect SPMEM10 - DATA0 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
7	WP_9_3	R/WK	0h	write protect SPMEM9 - DATA3 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
6	WP_9_2	R/WK	0h	write protect SPMEM9 - DATA2 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
5	WP_9_1	R/WK	0h	write protect SPMEM9 - DATA1 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
4	WP_9_0	R/WK	0h	write protect SPMEM9 - DATA0 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
3	WP_8_3	R/WK	0h	write protect SPMEM8 - DATA3 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
2	WP_8_2	R/WK	0h	write protect SPMEM8 - DATA2 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
1	WP_8_1	R/WK	0h	write protect SPMEM8 - DATA1 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
0	WP_8_0	R/WK	0h	write protect SPMEM8 - DATA0 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access

29.10.74 SPMWPROT3 (Offset = 150Ch) [Reset = 00000000h]

SPMWPROT3 is shown in Figure 29-81 and described in Table 29-76.

Return to the Summary Table.

Scratch pad memory write protect 3

Figure 29-81. 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 29-76. SPMWPROT3 Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	R-0/W	0h	need to write (KEY=0xE8) to update this register 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 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
14	WP_15_2	R/WK	0h	write protect SPMEM15 - DATA2 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
13	WP_15_1	R/WK	0h	write protect SPMEM15 - DATA1 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
12	WP_15_0	R/WK	0h	write protect SPMEM15 - DATA0 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
11	WP_14_3	R/WK	0h	write protect SPMEM14 - DATA3 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
10	WP_14_2	R/WK	0h	write protect SPMEM14- DATA2 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access

Table 29-76. SPMWPROT3 Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	WP_14_1	R/WK	0h	write protect SPMEM14 - DATA1 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
8	WP_14_0	R/WK	0h	write protect SPMEM14 - DATA0 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
7	WP_13_3	R/WK	0h	write protect SPMEM13 - DATA3 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
6	WP_13_2	R/WK	0h	write protect SPMEM13 - DATA2 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
5	WP_13_1	R/WK	0h	write protect SPMEM13 - DATA1 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
4	WP_13_0	R/WK	0h	write protect SPMEM13 - DATA0 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
3	WP_12_3	R/WK	0h	write protect SPMEM12 - DATA3 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
2	WP_12_2	R/WK	0h	write protect SPMEM12 - DATA2 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
1	WP_12_1	R/WK	0h	write protect SPMEM12 - DATA1 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
0	WP_12_0	R/WK	0h	write protect SPMEM12 - DATA0 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access

29.10.75 SPMWPROT4 (Offset = 1510h) [Reset = 00000000h]

SPMWPROT4 is shown in Figure 29-82 and described in Table 29-77.

Return to the Summary Table.

Scratch pad memory write protect 4

Figure 29-82. 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 29-77. SPMWPROT4 Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	R-0/W	0h	need to write (KEY=0xE8) to update this register 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 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
14	WP_19_2	R/WK	0h	write protect SPMEM19 - DATA2 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
13	WP_19_1	R/WK	0h	write protect SPMEM19 - DATA1 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
12	WP_19_0	R/WK	0h	write protect SPMEM19 - DATA0 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
11	WP_18_3	R/WK	0h	write protect SPMEM18 - DATA3 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
10	WP_18_2	R/WK	0h	write protect SPMEM18- DATA2 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access

Table 29-77. SPMWPROT4 Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	WP_18_1	R/WK	0h	write protect SPMEM18 - DATA1 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
8	WP_18_0	R/WK	0h	write protect SPMEM18 - DATA0 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
7	WP_17_3	R/WK	0h	write protect SPMEM17 - DATA3 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
6	WP_17_2	R/WK	0h	write protect SPMEM17 - DATA2 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
5	WP_17_1	R/WK	0h	write protect SPMEM17 - DATA1 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
4	WP_17_0	R/WK	0h	write protect SPMEM17 - DATA0 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
3	WP_16_3	R/WK	0h	write protect SPMEM16 - DATA3 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
2	WP_16_2	R/WK	0h	write protect SPMEM16 - DATA2 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
1	WP_16_1	R/WK	0h	write protect SPMEM16 - DATA1 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
0	WP_16_0	R/WK	0h	write protect SPMEM16 - DATA0 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access

29.10.76 SPMWPROT5 (Offset = 1514h) [Reset = 00000000h]

SPMWPROT5 is shown in Figure 29-83 and described in Table 29-78.

Return to the Summary Table.

Scratch pad memory write protect 5

Figure 29-83. 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 29-78. SPMWPROT5 Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	R-0/W	0h	need to write (KEY=0xE8) to update this register 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 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
14	WP_23_2	R/WK	0h	write protect SPMEM23 - DATA2 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
13	WP_23_1	R/WK	0h	write protect SPMEM23 - DATA1 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
12	WP_23_0	R/WK	0h	write protect SPMEM23 - DATA0 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
11	WP_22_3	R/WK	0h	write protect SPMEM22 - DATA3 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
10	WP_22_2	R/WK	0h	write protect SPMEM22- DATA2 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access

Table 29-78. SPMWPROT5 Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	WP_22_1	R/WK	0h	write protect SPMEM22 - DATA1 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
8	WP_22_0	R/WK	0h	write protect SPMEM22 - DATA0 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
7	WP_21_3	R/WK	0h	write protect SPMEM21 - DATA3 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
6	WP_21_2	R/WK	0h	write protect SPMEM21 - DATA2 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
5	WP_21_1	R/WK	0h	write protect SPMEM21 - DATA1 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
4	WP_21_0	R/WK	0h	write protect SPMEM21 - DATA0 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
3	WP_20_3	R/WK	0h	write protect SPMEM20 - DATA3 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
2	WP_20_2	R/WK	0h	write protect SPMEM20 - DATA2 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
1	WP_20_1	R/WK	0h	write protect SPMEM20 - DATA1 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
0	WP_20_0	R/WK	0h	write protect SPMEM20 - DATA0 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access

29.10.77 SPMWPROT6 (Offset = 1518h) [Reset = 00000000h]

SPMWPROT6 is shown in Figure 29-84 and described in Table 29-79.

Return to the Summary Table.

Scratch pad memory write protect 6

Figure 29-84. 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 29-79. SPMWPROT6 Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	R-0/W	0h	need to write (KEY=0xE8) to update this register 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 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
14	WP_27_2	R/WK	0h	write protect SPMEM27 - DATA2 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
13	WP_27_1	R/WK	0h	write protect SPMEM27 - DATA1 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
12	WP_27_0	R/WK	0h	write protect SPMEM27 - DATA0 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
11	WP_26_3	R/WK	0h	write protect SPMEM26 - DATA3 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
10	WP_26_2	R/WK	0h	write protect SPMEM26- DATA2 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access

Table 29-79. SPMWPROT6 Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	WP_26_1	R/WK	0h	write protect SPMEM26 - DATA1 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
8	WP_26_0	R/WK	0h	write protect SPMEM26 - DATA0 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
7	WP_25_3	R/WK	0h	write protect SPMEM25 - DATA3 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
6	WP_25_2	R/WK	0h	write protect SPMEM25 - DATA2 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
5	WP_25_1	R/WK	0h	write protect SPMEM25 - DATA1 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
4	WP_25_0	R/WK	0h	write protect SPMEM25 - DATA0 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
3	WP_24_3	R/WK	0h	write protect SPMEM24 - DATA3 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
2	WP_24_2	R/WK	0h	write protect SPMEM24 - DATA2 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
1	WP_24_1	R/WK	0h	write protect SPMEM24 - DATA1 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
0	WP_24_0	R/WK	0h	write protect SPMEM24 - DATA0 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access

29.10.78 SPMWPROT7 (Offset = 151Ch) [Reset = 00000000h]

SPMWPROT7 is shown in Figure 29-85 and described in Table 29-80.

Return to the Summary Table.

Scratch pad memory write protect 7

Figure 29-85. 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 29-80. SPMWPROT7 Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	R-0/W	0h	need to write (KEY=0xE8) to update this register 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 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
14	WP_31_2	R/WK	0h	write protect SPMEM31 - DATA2 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
13	WP_31_1	R/WK	0h	write protect SPMEM31 - DATA1 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
12	WP_31_0	R/WK	0h	write protect SPMEM31 - DATA0 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
11	WP_30_3	R/WK	0h	write protect SPMEM30 - DATA3 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
10	WP_30_2	R/WK	0h	write protect SPMEM30- DATA2 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access

Table 29-80. SPMWPROT7 Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	WP_30_1	R/WK	0h	write protect SPMEM30 - DATA1 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
8	WP_30_0	R/WK	0h	write protect SPMEM30 - DATA0 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
7	WP_29_3	R/WK	0h	write protect SPMEM29 - DATA3 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
6	WP_29_2	R/WK	0h	write protect SPMEM29 - DATA2 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
5	WP_29_1	R/WK	0h	write protect SPMEM29 - DATA1 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
4	WP_29_0	R/WK	0h	write protect SPMEM29 - DATA0 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
3	WP_28_3	R/WK	0h	write protect SPMEM28 - DATA3 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
2	WP_28_2	R/WK	0h	write protect SPMEM28 - DATA2 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
1	WP_28_1	R/WK	0h	write protect SPMEM28 - DATA1 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access
0	WP_28_0	R/WK	0h	write protect SPMEM28 - DATA0 KEY must be set to E8h to write to this bit. 0h = SPMEM is read and write access 1h = SPMEM is read only access

29.10.79 SPMTERASE0 (Offset = 1540h) [Reset = 00000000h]

SPMTERASE0 is shown in Figure 29-86 and described in Table 29-81.

Return to the Summary Table.

Scratch pad memory tamper erase enable 0

Figure 29-86. 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 29-81. SPMTERASE0 Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	R-0/W	0h	need to write (KEY=0xA3) to update this register 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 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
14	TE_3_2	R/WK	0h	tamper erase enable SPMEM3 - DATA2 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
13	TE_3_1	R/WK	0h	tamper erase enable SPMEM3 - DATA1 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
12	TE_3_0	R/WK	0h	tamper erase enable SPMEM3 - DATA0 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
11	TE_2_3	R/WK	0h	tamper erase enable SPMEM2 - DATA3 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
10	TE_2_2	R/WK	0h	tamper erase enable SPMEM2 - DATA2 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event

Table 29-81. SPMTERASE0 Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	TE_2_1	R/WK	0h	tamper erase enable SPMEM2 - DATA1 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
8	TE_2_0	R/WK	0h	tamper erase enable SPMEM2 - DATA0 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
7	TE_1_3	R/WK	0h	tamper erase enable SPMEM1 - DATA3 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
6	TE_1_2	R/WK	0h	tamper erase enable SPMEM1 - DATA2 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
5	TE_1_1	R/WK	0h	tamper erase enable SPMEM1 - DATA1 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
4	TE_1_0	R/WK	0h	tamper erase enable SPMEM1 - DATA0 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
3	TE_0_3	R/WK	0h	tamper erase enable SPMEM0 - DATA3 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
2	TE_0_2	R/WK	0h	tamper erase enable SPMEM0 - DATA2 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
1	TE_0_1	R/WK	0h	tamper erase enable SPMEM0 - DATA1 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
0	TE_0_0	R/WK	0h	tamper erase enable SPMEM0 - DATA0 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event

29.10.80 SPMTERASE1 (Offset = 1544h) [Reset = 00000000h]

SPMTERASE1 is shown in Figure 29-87 and described in Table 29-82.

Return to the Summary Table.

Scratch pad memory tamper erase enable 1

Figure 29-87. 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 29-82. SPMTERASE1 Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	R-0/W	0h	need to write (KEY=0xA3) to update this register 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 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
14	TE_7_2	R/WK	0h	tamper erase enable SPMEM7 - DATA2 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
13	TE_7_1	R/WK	0h	tamper erase enable SPMEM7 - DATA1 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
12	TE_7_0	R/WK	0h	tamper erase enable SPMEM7 - DATA0 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
11	TE_6_3	R/WK	0h	tamper erase enable SPMEM6 - DATA3 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
10	TE_6_2	R/WK	0h	tamper erase enable SPMEM6 - DATA2 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event

Table 29-82. SPMTERASE1 Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	TE_6_1	R/WK	0h	tamper erase enable SPMEM6 - DATA1 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
8	TE_6_0	R/WK	0h	tamper erase enable SPMEM6 - DATA0 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
7	TE_5_3	R/WK	0h	tamper erase enable SPMEM5 - DATA3 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
6	TE_5_2	R/WK	0h	tamper erase enable SPMEM5 - DATA2 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
5	TE_5_1	R/WK	0h	tamper erase enable SPMEM5 - DATA1 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
4	TE_5_0	R/WK	0h	tamper erase enable SPMEM5 - DATA0 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
3	TE_4_3	R/WK	0h	tamper erase enable SPMEM4 - DATA3 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
2	TE_4_2	R/WK	0h	tamper erase enable SPMEM4 - DATA2 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
1	TE_4_1	R/WK	0h	tamper erase enable SPMEM4 - DATA1 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
0	TE_4_0	R/WK	0h	tamper erase enable SPMEM4 - DATA0 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event

29.10.81 SPMTERASE2 (Offset = 1548h) [Reset = 00000000h]

SPMTERASE2 is shown in Figure 29-88 and described in Table 29-83.

Return to the Summary Table.

Scratch pad memory tamper erase enable 2

Figure 29-88. 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 29-83. SPMTERASE2 Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	R-0/W	0h	need to write (KEY=0xA3) to update this register 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 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
14	TE_11_2	R/WK	0h	tamper erase enable SPMEM11 - DATA2 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
13	TE_11_1	R/WK	0h	tamper erase enable SPMEM11 - DATA1 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
12	TE_11_0	R/WK	0h	tamper erase enable SPMEM11 - DATA0 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
11	TE_10_3	R/WK	0h	tamper erase enable SPMEM10 - DATA3 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
10	TE_10_2	R/WK	0h	tamper erase enable SPMEM10 - DATA2 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event

Table 29-83. SPMTERASE2 Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	TE_10_1	R/WK	0h	tamper erase enable SPMEM10 - DATA1 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
8	TE_10_0	R/WK	0h	tamper erase enable SPMEM10 - DATA0 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
7	TE_9_3	R/WK	0h	tamper erase enable SPMEM9 - DATA3 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
6	TE_9_2	R/WK	0h	tamper erase enable SPMEM9 - DATA2 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
5	TE_9_1	R/WK	0h	tamper erase enable SPMEM9 - DATA1 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
4	TE_9_0	R/WK	0h	tamper erase enable SPMEM9 - DATA0 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
3	TE_8_3	R/WK	0h	tamper erase enable SPMEM8 - DATA3 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
2	TE_8_2	R/WK	0h	tamper erase enable SPMEM8 - DATA2 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
1	TE_8_1	R/WK	0h	tamper erase enable SPMEM8 - DATA1 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
0	TE_8_0	R/WK	0h	tamper erase enable SPMEM8 - DATA0 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event

29.10.82 SPMTERASE3 (Offset = 154Ch) [Reset = 00000000h]

SPMTERASE3 is shown in Figure 29-89 and described in Table 29-84.

Return to the Summary Table.

Scratch pad memory tamper erase enable 3

Figure 29-89. 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 29-84. SPMTERASE3 Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	R-0/W	0h	need to write (KEY=0xA3) to update this register 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 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
14	TE_15_2	R/WK	0h	tamper erase enable SPMEM15 - DATA2 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
13	TE_15_1	R/WK	0h	tamper erase enable SPMEM15 - DATA1 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
12	TE_15_0	R/WK	0h	tamper erase enable SPMEM15 - DATA0 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
11	TE_14_3	R/WK	0h	tamper erase enable SPMEM14 - DATA3 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
10	TE_14_2	R/WK	0h	tamper erase enable SPMEM14 - DATA2 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event

Table 29-84. SPMTERASE3 Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	TE_14_1	R/WK	0h	tamper erase enable SPMEM14 - DATA1 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
8	TE_14_0	R/WK	0h	tamper erase enable SPMEM14 - DATA0 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
7	TE_13_3	R/WK	0h	tamper erase enable SPMEM13 - DATA3 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
6	TE_13_2	R/WK	0h	tamper erase enable SPMEM13 - DATA2 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
5	TE_13_1	R/WK	0h	tamper erase enable SPMEM13 - DATA1 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
4	TE_13_0	R/WK	0h	tamper erase enable SPMEM13 - DATA0 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
3	TE_12_3	R/WK	0h	tamper erase enable SPMEM12 - DATA3 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
2	TE_12_2	R/WK	0h	tamper erase enable SPMEM12 - DATA2 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
1	TE_12_1	R/WK	0h	tamper erase enable SPMEM12 - DATA1 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
0	TE_12_0	R/WK	0h	tamper erase enable SPMEM12 - DATA0 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event

29.10.83 SPMTERASE4 (Offset = 1550h) [Reset = 00000000h]

SPMTERASE4 is shown in Figure 29-90 and described in Table 29-85.

Return to the Summary Table.

Scratch pad memory tamper erase enable 4

Figure 29-90. 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 29-85. SPMTERASE4 Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	R-0/W	0h	need to write (KEY=0xA3) to update this register 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 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
14	TE_19_2	R/WK	0h	tamper erase enable SPMEM19 - DATA2 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
13	TE_19_1	R/WK	0h	tamper erase enable SPMEM19 - DATA1 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
12	TE_19_0	R/WK	0h	tamper erase enable SPMEM19 - DATA0 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
11	TE_18_3	R/WK	0h	tamper erase enable SPMEM18 - DATA3 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
10	TE_18_2	R/WK	0h	tamper erase enable SPMEM18 - DATA2 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event

Table 29-85. SPMTERASE4 Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	TE_18_1	R/WK	0h	tamper erase enable SPMEM18 - DATA1 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
8	TE_18_0	R/WK	0h	tamper erase enable SPMEM18 - DATA0 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
7	TE_17_3	R/WK	0h	tamper erase enable SPMEM17 - DATA3 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
6	TE_17_2	R/WK	0h	tamper erase enable SPMEM17 - DATA2 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
5	TE_17_1	R/WK	0h	tamper erase enable SPMEM17 - DATA1 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
4	TE_17_0	R/WK	0h	tamper erase enable SPMEM17 - DATA0 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
3	TE_16_3	R/WK	0h	tamper erase enable SPMEM16 - DATA3 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
2	TE_16_2	R/WK	0h	tamper erase enable SPMEM16 - DATA2 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
1	TE_16_1	R/WK	0h	tamper erase enable SPMEM16 - DATA1 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
0	TE_16_0	R/WK	0h	tamper erase enable SPMEM16 - DATA0 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event

29.10.84 SPMTERASE5 (Offset = 1554h) [Reset = 00000000h]

SPMTERASE5 is shown in Figure 29-91 and described in Table 29-86.

Return to the Summary Table.

Scratch pad memory tamper erase enable 5

Figure 29-91. 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 29-86. SPMTERASE5 Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	R-0/W	0h	need to write (KEY=0xA3) to update this register 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 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
14	TE_23_2	R/WK	0h	tamper erase enable SPMEM23 - DATA2 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
13	TE_23_1	R/WK	0h	tamper erase enable SPMEM23 - DATA1 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
12	TE_23_0	R/WK	0h	tamper erase enable SPMEM23 - DATA0 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
11	TE_22_3	R/WK	0h	tamper erase enable SPMEM22 - DATA3 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
10	TE_22_2	R/WK	0h	tamper erase enable SPMEM22 - DATA2 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event

Table 29-86. SPMTERASE5 Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	TE_22_1	R/WK	0h	tamper erase enable SPMEM22 - DATA1 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
8	TE_22_0	R/WK	0h	tamper erase enable SPMEM22 - DATA0 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
7	TE_21_3	R/WK	0h	tamper erase enable SPMEM21 - DATA3 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
6	TE_21_2	R/WK	0h	tamper erase enable SPMEM21 - DATA2 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
5	TE_21_1	R/WK	0h	tamper erase enable SPMEM21 - DATA1 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
4	TE_21_0	R/WK	0h	tamper erase enable SPMEM21 - DATA0 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
3	TE_20_3	R/WK	0h	tamper erase enable SPMEM20 - DATA3 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
2	TE_20_2	R/WK	0h	tamper erase enable SPMEM20 - DATA2 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
1	TE_20_1	R/WK	0h	tamper erase enable SPMEM20 - DATA1 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
0	TE_20_0	R/WK	0h	tamper erase enable SPMEM20 - DATA0 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event

29.10.85 SPMTERASE6 (Offset = 1558h) [Reset = 00000000h]

SPMTERASE6 is shown in Figure 29-92 and described in Table 29-87.

Return to the Summary Table.

Scratch pad memory tamper erase enable 6

Figure 29-92. 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 29-87. SPMTERASE6 Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	R-0/W	0h	need to write (KEY=0xA3) to update this register 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 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
14	TE_27_2	R/WK	0h	tamper erase enable SPMEM27 - DATA2 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
13	TE_27_1	R/WK	0h	tamper erase enable SPMEM27 - DATA1 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
12	TE_27_0	R/WK	0h	tamper erase enable SPMEM27 - DATA0 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
11	TE_26_3	R/WK	0h	tamper erase enable SPMEM26 - DATA3 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
10	TE_26_2	R/WK	0h	tamper erase enable SPMEM26 - DATA2 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event

Table 29-87. SPMTERASE6 Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	TE_26_1	R/WK	0h	tamper erase enable SPMEM26 - DATA1 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
8	TE_26_0	R/WK	0h	tamper erase enable SPMEM26 - DATA0 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
7	TE_25_3	R/WK	0h	tamper erase enable SPMEM25 - DATA3 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
6	TE_25_2	R/WK	0h	tamper erase enable SPMEM25 - DATA2 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
5	TE_25_1	R/WK	0h	tamper erase enable SPMEM25 - DATA1 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
4	TE_25_0	R/WK	0h	tamper erase enable SPMEM25 - DATA0 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
3	TE_24_3	R/WK	0h	tamper erase enable SPMEM24 - DATA3 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
2	TE_24_2	R/WK	0h	tamper erase enable SPMEM24 - DATA2 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
1	TE_24_1	R/WK	0h	tamper erase enable SPMEM24 - DATA1 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
0	TE_24_0	R/WK	0h	tamper erase enable SPMEM24 - DATA0 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event

29.10.86 SPMTERASE7 (Offset = 155Ch) [Reset = 00000000h]

SPMTERASE7 is shown in Figure 29-93 and described in Table 29-88.

Return to the Summary Table.

Scratch pad memory tamper erase enable 7

Figure 29-93. 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 29-88. SPMTERASE7 Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	R-0/W	0h	need to write (KEY=0xA3) to update this register 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 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
14	TE_31_2	R/WK	0h	tamper erase enable SPMEM31 - DATA2 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
13	TE_31_1	R/WK	0h	tamper erase enable SPMEM31 - DATA1 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
12	TE_31_0	R/WK	0h	tamper erase enable SPMEM31 - DATA0 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
11	TE_30_3	R/WK	0h	tamper erase enable SPMEM30 - DATA3 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
10	TE_30_2	R/WK	0h	tamper erase enable SPMEM30 - DATA2 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event

Table 29-88. SPMTERASE7 Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	TE_30_1	R/WK	0h	tamper erase enable SPMEM30 - DATA1 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
8	TE_30_0	R/WK	0h	tamper erase enable SPMEM30 - DATA0 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
7	TE_29_3	R/WK	0h	tamper erase enable SPMEM29 - DATA3 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
6	TE_29_2	R/WK	0h	tamper erase enable SPMEM29 - DATA2 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
5	TE_29_1	R/WK	0h	tamper erase enable SPMEM29 - DATA1 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
4	TE_29_0	R/WK	0h	tamper erase enable SPMEM29 - DATA0 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
3	TE_28_3	R/WK	0h	tamper erase enable SPMEM28 - DATA3 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
2	TE_28_2	R/WK	0h	tamper erase enable SPMEM28 - DATA2 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
1	TE_28_1	R/WK	0h	tamper erase enable SPMEM28 - DATA1 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event
0	TE_28_0	R/WK	0h	tamper erase enable SPMEM28 - DATA0 KEY must be set to A3h to write to this bit. 0h = SPMEM is unmodified during tamper event 1h = SPMEM will be erased with tamper event

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 30-1 shows the block diagram of the IWDT.

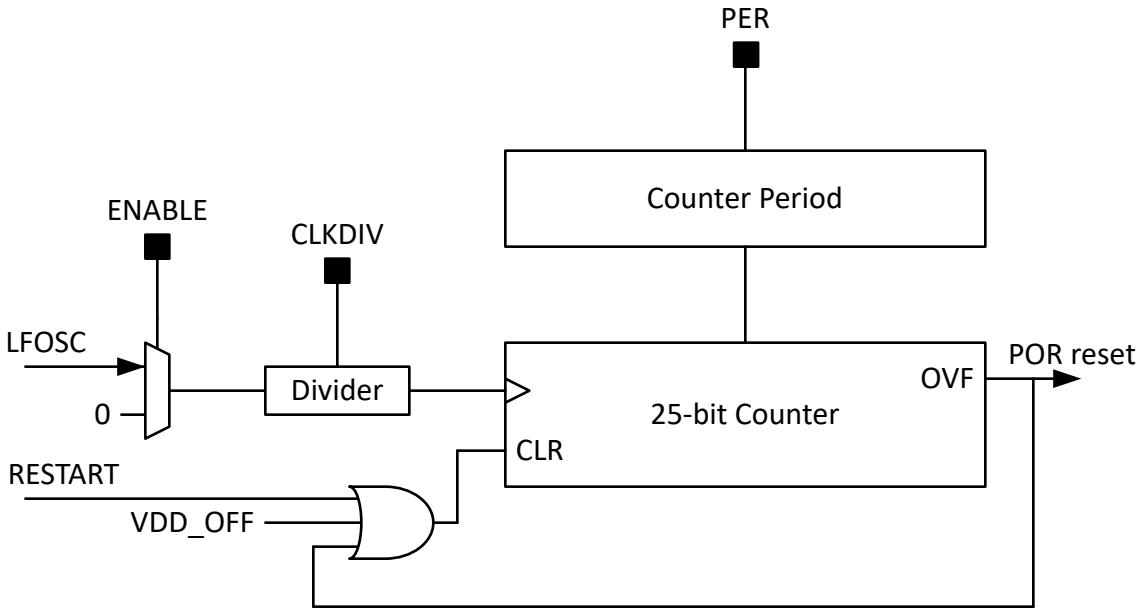


Figure 30-1. Independent Watchdog Timer (IWDT) Block Diagram

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.

30.1 1955
30.2 IWDT Clock Configuration 1955
30.3 IWDT Period Selection 1955
30.4 Debug Behavior of the IWDT 1955
30.5 IWDT Registers 1955

30.1

30.2 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).

30.3 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 30-1.

Table 30-1. 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 30-2 gives all possible IWDT periods for a given combination of PER and CLKDIV.

Table 30-2. 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

30.4 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.

30.5 IWDT Registers

For IWDT registers, refer to the Section 29.10 related to the watchdog timer.

Chapter 31

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.

31.1 WWDT Overview 1957
31.2 WWDT Operation 1958
31.3 WWDT Registers 1962

31.1 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 25-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

Devices may have 1 or 2 WWDT instances. A WWDT0 violation generates a BOOTRST, which resets the peripheral and CPU state and also causes the boot configuration routine (BCR) to run. A WWDT1 violation generates a SYSRST, which resets the peripheral and CPU state but does not trigger execution of the BCR. As such, WWDT1 is well suited for recovering from execution stalls that result from software execution, while WWDT0 is well suited for catching larger issues such as a corrupted trim value, at the expense of a longer reset time.

31.1.1 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 31-1. 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.

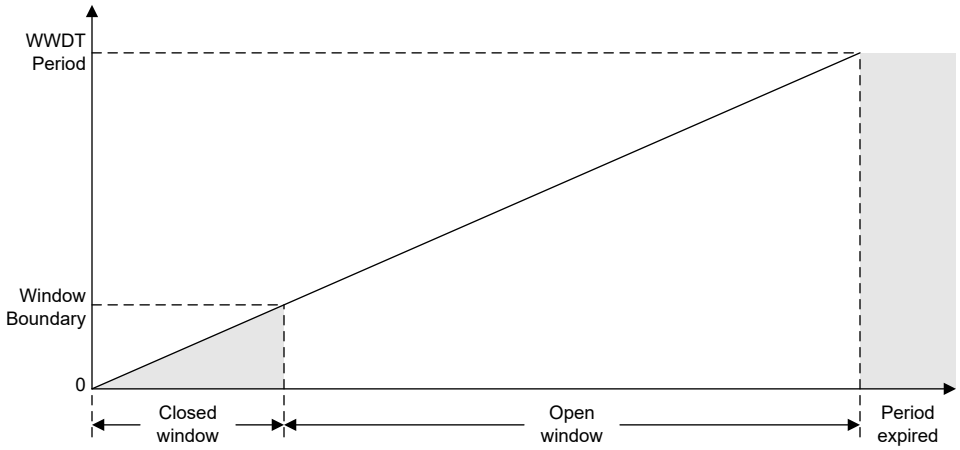


Figure 31-1. WWDT Functionality

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.

31.1.2 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.

31.2 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 31-2 shows the WWDT functional block diagram.

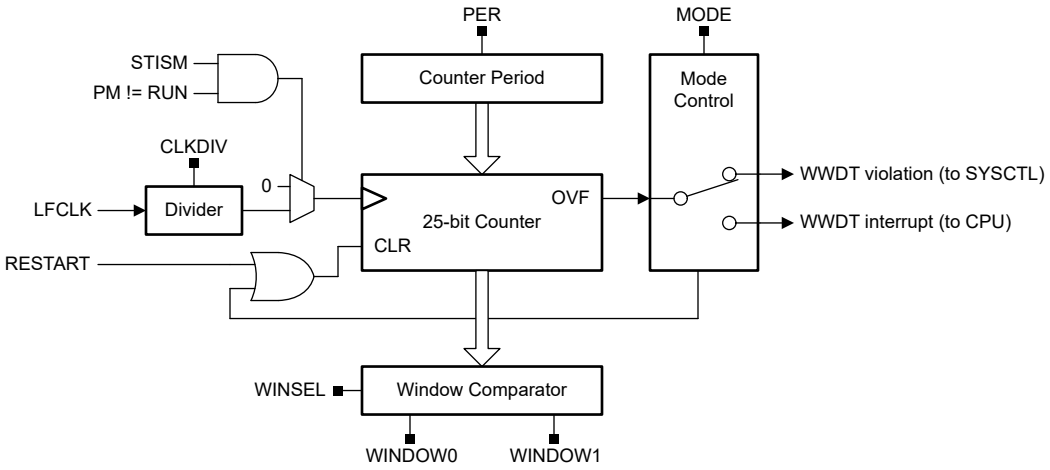


Figure 31-2. WWDT Diagram

31.2.1 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.

31.2.2 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).

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 31-3. In Figure 31-3, 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*.

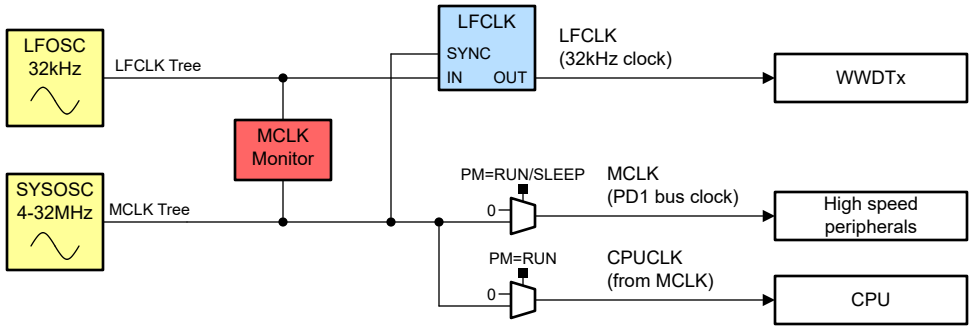


Figure 31-3. WWDT Simplified Clock Source Tree

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 25-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:

$$T_{WWDT} = (CLKDIV + 1) * PER_{COUNT} / 32768Hz \quad (30)$$

The total timer count PER_{COUNT} is selected to be one of 8 possible period count values, with the encoding given in Table 31-1.

Table 31-1. WWDT Period Total Timer Count

PER	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7
PER_{COUNT}	2^{25}	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 136.53 minutes. Table 31-2 gives all possible WWDT periods for a given combination of PER and CLKDIV.

Table 31-2. WWDT Period Timing Options

CLKDIV	PER							
	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7
	min	*min*	*s*	*s*	*s*	*ms*	*ms*	*ms*
0x0 (/1)	17.07	1.07	8.00	1.00	0.13	31.25	7.81	1.95

Table 31-2. WWDT Period Timing Options (continued)

CLKDIV	PER							
	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7
	min	*min*	*s*	*s*	*s*	*ms*	*ms*	*ms*
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	13.67
0x7 (/8)	136.53	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 31-2 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 31-3. 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).

31.2.3 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 sleep mode. 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.

31.2.4 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.

31.2.5 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 31-4 lists the WWDT events.

Table 31-4. 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

31.2.5.1 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 31-5.

Table 31-5. 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 8.2.5 for guidance on configuring the Event registers for CPU interrupts.

31.3 WWDT Registers

Table 31-6 lists the memory-mapped registers for the WWDT registers. All register offset addresses not listed in Table 31-6 should be considered as reserved locations and the register contents should not be modified.

Table 31-6. WWDT Registers

Offset	Acronym	Register Name	Section
800h	PWREN	Power enable	Section 31.3.1
804h	RSTCTL	Reset Control	Section 31.3.2
814h	STAT	Status Register	Section 31.3.3
1018h	PDBGCTL	Peripheral Debug Control	Section 31.3.4
1020h	IIDX	Interrupt index	Section 31.3.5
1028h	IMASK	Interrupt mask	Section 31.3.6
1030h	RIS	Raw interrupt status	Section 31.3.7
1038h	MIS	Masked interrupt status	Section 31.3.8
1040h	ISET	Interrupt set	Section 31.3.9
1048h	ICLR	Interrupt clear	Section 31.3.10
10E0h	EVT_MODE	Event Mode	Section 31.3.11
10FCh	DESC	Module Description	Section 31.3.12
1100h	WWDTCTL0	Window Watchdog Timer Control Register 0	Section 31.3.13
1104h	WWDTCTL1	Window Watchdog Timer Control Register 0	Section 31.3.14
1108h	WWDTCNTRST	Window Watchdog Timer Counter Reset Register	Section 31.3.15
110Ch	WWDTSTAT	Window Watchdog Timer Status Register	Section 31.3.16

Complex bit access types are encoded to fit into small table cells. Table 31-7 shows the codes that are used for access types in this section.

Table 31-7. 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 K	Write Write protected by a key
Reset or Default Value		
-n		Value after reset or the default value

31.3.1 PWREN Register (Offset = 800h) [Reset = 00000000h]

PWREN is shown in Figure 31-4 and described in Table 31-8.

Return to the Summary Table.

Register to control the power state

Figure 31-4. 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 31-8. PWREN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to allow Power State Change 26h = KEY to allow write access to this register
23-1	RESERVED	R/W	0h	
0	ENABLE	K	0h	Enable the power Note: For safety devices the power cannot be disabled once enabled. KEY must be set to 26h to write to this bit. 0h = Disable Power 1h = Enable Power

31.3.2 RSTCTL Register (Offset = 804h) [Reset = 00000000h]

RSTCTL is shown in Figure 31-5 and described in Table 31-9.

Return to the Summary Table.

Register to control reset assertion and de-assertion

Figure 31-5. 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 31-9. RSTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	Unlock key B1h = KEY to allow write access to this register
23-2	RESERVED	W	0h	
1	RESETSTKYCLR	WK	0h	Clear RESETSTKY KEY must be set to B1h to write to this bit. 0h = Writing 0 has no effect 1h = Clear reset sticky bit
0	RESETASSERT	WK	0h	Assert reset to the peripheral Note: For safety devices a watchdog reset by software is not possible. KEY must be set to B1h to write to this bit. 0h = Writing 0 has no effect 1h = Assert reset

31.3.3 STAT Register (Offset = 814h) [Reset = 00000000h]

STAT is shown in Figure 31-6 and described in Table 31-10.

Return to the Summary Table.

peripheral enable and reset status

Figure 31-6. 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 31-10. 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 0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register 1h = The peripheral was reset since the last bit clear
15-0	RESERVED	R	0h	

31.3.4 PDBGCTL Register (Offset = 1018h) [Reset = 00000000h]

PDBGCTL is shown in Figure 31-7 and described in Table 31-11.

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 31-7. 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 31-11. PDBGCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	FREE	R/W	0h	Free run control 0h = The peripheral freezes functionality while the Core Halted input is asserted and resumes when it is deasserted. 1h = The peripheral ignores the state of the Core Halted input

31.3.5 IIDX Register (Offset = 1020h) [Reset = 00000000h]

IIDX is shown in Figure 31-8 and described in Table 31-12.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index.

Figure 31-8. 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 31-12. 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. 0h = No interrupt pending 1h = Interval Timer Interrupt; Interrupt Flag: INTTIM; Interrupt Priority: Highest

31.3.6 IMASK Register (Offset = 1028h) [Reset = 00000000h]

IMASK is shown in Figure 31-9 and described in Table 31-13.

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 31-9. 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 31-13. IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	INTTIM	R/W	0h	Interval Timer Interrupt. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask

31.3.7 RIS Register (Offset = 1030h) [Reset = 00000000h]

RIS is shown in Figure 31-10 and described in Table 31-14.

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 31-10. 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 31-14. RIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	INTTIM	R	0h	Interval Timer Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred

31.3.8 MIS Register (Offset = 1038h) [Reset = 00000000h]

MIS is shown in Figure 31-11 and described in Table 31-15.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Figure 31-11. 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 31-15. MIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	INTTIM	R	0h	Interval Timer Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred

31.3.9 ISET Register (Offset = 1040h) [Reset = 00000000h]

ISET is shown in Figure 31-12 and described in Table 31-16.

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 31-12. 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 31-16. ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	W	0h	
0	INTTIM	W	0h	Interval Timer Interrupt. 0h = Writing 0 has no effect 1h = Set Interrupt

31.3.10 ICLR Register (Offset = 1048h) [Reset = 00000000h]

ICLR is shown in Figure 31-13 and described in Table 31-17.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

Figure 31-13. 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 31-17. ICLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	W	0h	
0	INTTIM	W	0h	Interval Timer Interrupt. 0h = Writing 0 has no effect 1h = Clear Interrupt

31.3.11 EVT_MODE Register (Offset = 10E0h) [Reset = 00000001h]

EVT_MODE is shown in Figure 31-14 and described in Table 31-18.

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 31-14. 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 31-18. 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] 0h = The interrupt or event line is disabled. 1h = The interrupt or event line is in software mode. Software must clear the RIS. 2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag.

31.3.12 DESC Register (Offset = 10FCh) [Reset = 1F117010h]

DESC is shown in Figure 31-15 and described in Table 31-19.

Return to the Summary Table.

This register identifies the peripheral and its exact version.

Figure 31-15. 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 31-19. 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. 0h = Smallest value FFFFh = Highest possible value
15-12	FEATUREVER	R	7h	Feature Set for the module *instance* 0h = Smallest value 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 0h = Smallest value Fh = Highest possible value
7-4	MAJREV	R	1h	Major rev of the IP 0h = Smallest value Fh = Highest possible value
3-0	MINREV	R	0h	Minor rev of the IP 0h = Smallest value Fh = Highest possible value

31.3.13 WWDTCTL0 Register (Offset = 1100h) [Reset = 00000043h]

WWDTCTL0 is shown in Figure 31-16 and described in Table 31-20.

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 31-16. 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 31-20. WWDTCTL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to allow write access to this register. Writing to this register with an incorrect key activates the WWDT error signal to the ESM. Read as 0. 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. The functionality of this bit requires that POLICY.HWCEN = 0. If POLICY.HWCEN = 1 the WWDT resets during sleep and needs re-configuration. Note: This bit has no effect for the global Window Watchdog as Sleep Mode is not supported. 0h = The WWDT continues to function in Sleep mode. 1h = The WWDT stops in Sleep mode and resumes where it was stopped after wakeup.
16	MODE	R/W	0h	Window Watchdog Timer Mode 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 1h = Interval Timer Mode. The WWDT acts as an interval timer. It generates an interrupt on timeout.
15	RESERVED	R/W	0h	

Table 31-20. 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). 0h = 0% (No closed Window) 1h = 12.50% of the total timer period is closed window 2h = 18.75% of the total timer period is closed window 3h = 25% of the total timer period is closed window 4h = 50% of the total timer period is closed window 5h = 75% of the total timer period is closed window 6h = 81.25% of the total timer period is closed window 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). 0h = 0% (No closed Window) 1h = 12.50% of the total timer period is closed window 2h = 18.75% of the total timer period is closed window 3h = 25% of the total timer period is closed window 4h = 50% of the total timer period is closed window 5h = 75% of the total timer period is closed window 6h = 81.25% of the total timer period is closed window 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. 0h = Total timer count is 2^{25} 1h = Total timer count is 2^{21} 2h = Total timer count is 2^{18} 3h = Total timer count is 2^{15} 4h = Total timer count is 2^{12} (default) 5h = Total timer count is 2^{10} 6h = Total timer count is 2^{8} 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. The clock divider is currently 4 bits. Bit 4 has no effect and should always be written with 0. 0h = Minimum value 7h = Maximum value

31.3.14 WWDTCTL1 Register (Offset = 1104h) [Reset = 00000000h]

WWDTCTL1 is shown in Figure 31-17 and described in Table 31-21.

Return to the Summary Table.

Window Watchdog Timer Control 1 Register

Figure 31-17. 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 31-21. WWDTCTL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to allow write access to this register. Writing to this register with an incorrect key activates the WWDT error signal to the ESM. Read as 0. BEh (W) = KEY to allow write access to this register
23-1	RESERVED	R/W	0h	
0	WINSEL	R/W	0h	Close Window Select 0h = In window mode field WINDOW0 of WDDTCTL0 defines the closed window size. 1h = In window mode field WINDOW1 of WDDTCTL0 defines the closed window size.

31.3.15 WWDTCNTRST Register (Offset = 1108h) [Reset = 00000000h]

WWDTCNTRST is shown in Figure 31-18 and described in Table 31-22.

Return to the Summary Table.

Window Watchdog Timer Counter Restart Register

Figure 31-18. 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 31-22. 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. Writing any other value causes an error generation to the ESM. Read as 0. 0h = Minimum value FFFFFFFFh = Maximum value

31.3.16 WWDTSTAT Register (Offset = 110Ch) [Reset = 00000000h]

WWDTSTAT is shown in Figure 31-19 and described in Table 31-23.

Return to the Summary Table.

Window Watchdog Timer Status Register
A write to this register has no effect.

Figure 31-19. 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 31-23. WWDTSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	RUN	R	0h	Watchdog running status flag. 0h = Watchdog counter stopped. 1h = Watchdog running.

Chapter 32

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.

32.1 DEBUGSS Overview 1981
32.2 DEBUGSS Operation 1984
32.3 DEBUGSS Registers 1991

32.1 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

32.1.1 Debug Interconnect

The DEBUGSS architecture is given in Figure 32-1.

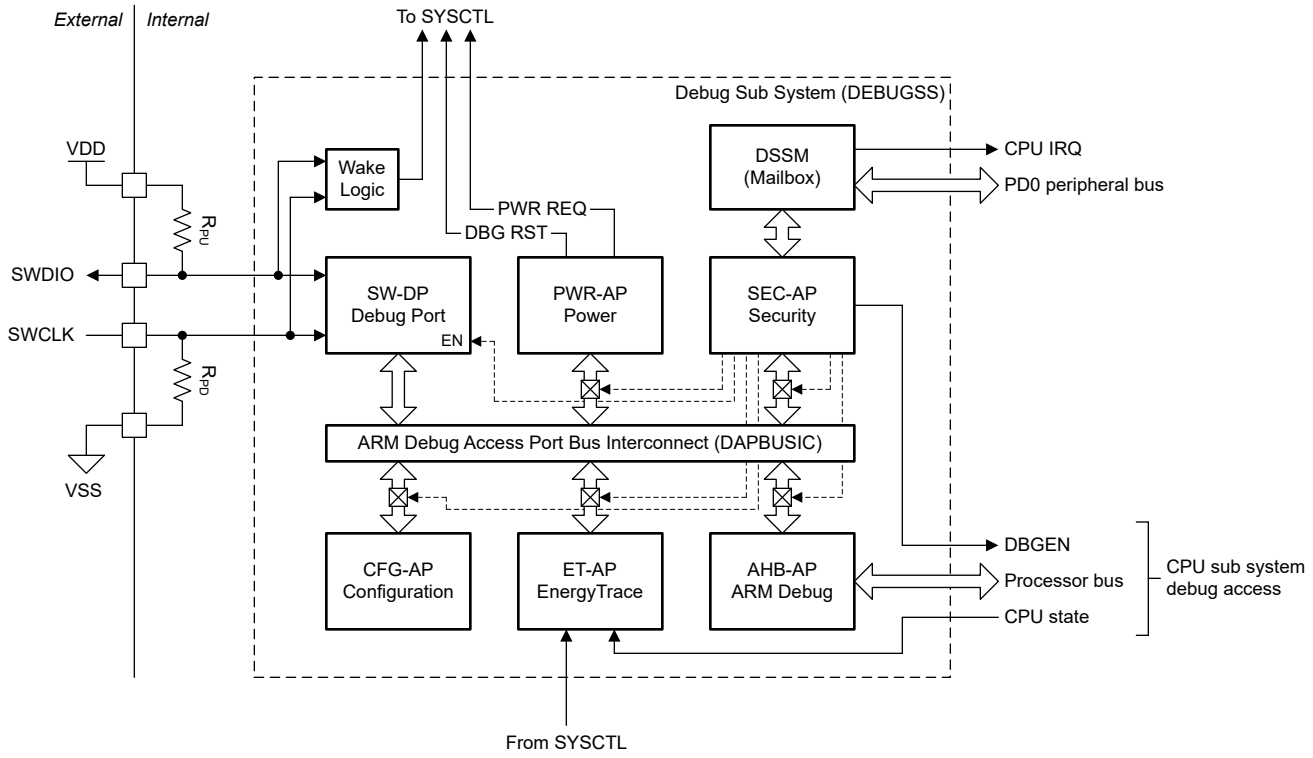


Figure 32-1. Debug Sub System Block Diagram

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 32.2.3).

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 32.1.3.

The SWD and SW-DP also contain signaling to the PMCU module to support debug-generated resets and operating mode changes (see Section 32.2.2).

32.1.2 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.

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.

32.1.3 Debug Access Ports

The debug access ports in the DEBUGSS are given in Table 32-1.

Table 32-1. 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)

Note

ET-AP is not available on MSPM33C devices.

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 32.2.1 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 32.2.4 for more information.

32.2 DEBUGSS Operation

The features and behaviors of the DEBUGSS module is discussed in the following sections.

32.2.1 Debug Features

The DEBUGSS supports processor debug, processor trace, peripheral debug, and energy state debug.

32.2.1.1 Processor Debug

The Arm Cortex-M33 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-M33 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

32.2.1.1.1 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);
```

32.2.1.1.2 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.

32.2.1.1.3 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-M33 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 32-2) which can be read out later by application software or the debug probe.

Table 32-2. MTB Buffer Memory

Start Address	End Address	Length
0x4040.3000	0x4040.3020	32B (4 trace packets)
0x5040.3000	0x5040.3020	32B (4 trace packets) when Secure Resource

32.2.1.2 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).

32.2.2 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.

In SHUTDOWN mode, any active debug connection is terminated as the debug logic is powered down with the device VCORE. While a debug connection to the DEBUGSS is not possible while the device is in SHUTDOWN 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 32-3.

Table 32-3. 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	-

32.2.3 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 32-4. 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 32-4. 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
AHB-AP (CPU Debug)	EN	EN w/ PW	DIS	DIS

The debug access control can be configured by writing predefined values into the NONMAIN BOOTCFG0.

Table 32-5. 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 data sheet for device-specific features.

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.

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 32-6. 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. 0x20E5F739C8D245C0E34A9C1301C12037DA29224D9B2C0738949C8ADC49AD34CA
   b. 0x2738a568
   c. 0x2ad52550
   d. 0x17722746
   e. 0x8dd2b55f
   f. 0x552c34bd
   g. 0x25560b90
   h. 0x67b0a3f3
   i. 0xd5c648f9
4. Reverse the output of the eight 32-bit words

   a. 0x20E5F739 → 0x39F7E520
   b. 0xC8D245C0 → 0xC045D2C8
   c. 0xE34A9C13 → 0x139C4AE3
   d. 0x01C12037 → 0x3720C101
   e. 0xDA29224D → 0x4D2229DA
   f. 0x9B2C0738 → 0x38072C9B
   g. 0x949C8ADC → 0xDC8A9C94
   h. 0x49AD34CA → 0xCA34AD49
5. Split the 128-bit value into four 32-bit words and write the value into the respective registers
   a. PWDFACTORYRESET[0]←0x39F7E520
   b. PWDFACTORYRESET[1]←0xC045D2C8
   c. PWDFACTORYRESET[2]←0x139C4AE3
   d. PWDFACTORYRESET[3]←0x3720C101
   e. PWDFACTORYRESET[4]←0x4D2229DA
   f. PWDFACTORYRESET[5]←0x38072C9B
   g. PWDFACTORYRESET[6]←0xDC8A9C94
   h. PWDFACTORYRESET[7]←0xCA34AD49
6. Repeat the same steps for any function that needs password protection

32.2.4 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 32-7. 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 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 32-8. 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		

32.2.4.1 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 32-9.

Table 32-9. 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

32.2.4.1.1 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 32-10.

Table 32-10. 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 8.2.5 for guidance on configuring the Event registers for CPU interrupts.

32.2.4.2 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 32-11 for possible DSSM commands and values.

Table 32-11. DSSM Commands

DSSM Command	DSSM Value
Factory Reset	0x020Ah
Mass Erase	0x020Ch
Password Authentication	0x030Eh
Data Exchange	0x00EEh

Table 32-11. DSSM Commands (continued)

DSSM Command	DSSM Value
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.

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.

32.3 DEBUGSS Registers

Table 32-12 lists the memory-mapped registers for the DEBUGSS registers. All register offset addresses not listed in Table 32-12 should be considered as reserved locations and the register contents should not be modified.

Table 32-12. 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 32-13 shows the codes that are used for access types in this section.

Table 32-13. DEBUGSS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
WK	W K	Write Write protected by a key
Reset or Default Value		
-n		Value after reset or the default value

32.3.1 IIDX (Offset = 1020h) [Reset = 00000000h]

IIDX is shown in Figure 32-2 and described in Table 32-14.

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 32-2. 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 32-14. IIDX Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	STAT	R	0h	Interrupt index status 0h = No pending interrupt request 1h = TX interrupt 2h = RX interrupt 3h = Power-up interrupt. A debug session has started. 4h = Power-up interrupt. A debug session has started.

32.3.2 IMASK (Offset = 1028h) [Reset = 00000000h]

IMASK is shown in Figure 32-3 and described in Table 32-15.

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 32-3. 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 32-15. IMASK Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	PWRDWNIFG	R/W	0h	Masks PWRDWNIFG in MIS register 0h = Interrupt is masked out 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 0h = Interrupt is masked out 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 0h = Interrupt is masked out 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 0h = Interrupt is masked out 1h = Interrupt will request an interrupt service routine and corresponding bit in MIS will be set

32.3.3 RIS (Offset = 1030h) [Reset = 00000000h]

RIS is shown in Figure 32-4 and described in Table 32-16.

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 32-4. 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 32-16. RIS Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3	PWRDWNIFG	R	0h	Raw interrupt status for PWRDWNIFG 0h = PWRUPIFG did not occur 1h = PWRUPIFG occurred
2	PWRUPIFG	R	0h	Raw interrupt status for PWRUPIFG 0h = PWRUPIFG did not occur 1h = PWRUPIFG occurred
1	RXIFG	R	0h	Raw interrupt status for RXIFG 0h = RXIFG did not occur 1h = RXIFG occurred
0	TXIFG	R	0h	Raw interrupt status for TXIFG 0h = TXIFG did not occur 1h = TXIFG occurred

32.3.4 MIS (Offset = 1038h) [Reset = 00000000h]

MIS is shown in Figure 32-5 and described in Table 32-17.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Figure 32-5. 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 32-17. MIS Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3	PWRDWNIFG	R	0h	Masked interrupt status for PWRDWNIFG 0h = PWRUPIFG did not request an interrupt service routine 1h = PWRUPIFG requests an interrupt service routine
2	PWRUPIFG	R	0h	Masked interrupt status for PWRUPIFG 0h = PWRUPIFG did not request an interrupt service routine 1h = PWRUPIFG requests an interrupt service routine
1	RXIFG	R	0h	Masked interrupt status for RXIFG 0h = RXIFG did not request an interrupt service routine 1h = RXIFG requests an interrupt service routine
0	TXIFG	R	0h	Masked interrupt status for TXIFG 0h = TXIFG did not request an interrupt service routine 1h = TXIFG requests an interrupt service routine

32.3.5 ISET (Offset = 1040h) [Reset = 00000000h]

ISET is shown in Figure 32-6 and described in Table 32-18.

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 32-6. 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 32-18. ISET Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	W	0h	
3	PWRDWNIFG	W	0h	Sets PWRDWNIFG in RIS register 0h = Writing a 0 has no effect 1h = RIS bit corresponding to PWRUPIFG is set
2	PWRUPIFG	W	0h	Sets PWRUPIFG in RIS register 0h = Writing a 0 has no effect 1h = RIS bit corresponding to PWRUPIFG is set
1	RXIFG	W	0h	Sets RXIFG in RIS register 0h = Writing a 0 has no effect 1h = RIS bit corresponding to RXIFG is set
0	TXIFG	W	0h	Sets TXIFG in RIS register 0h = Writing a 0 has no effect 1h = RIS bit corresponding to TXIFG is set

32.3.6 ICLR (Offset = 1048h) [Reset = 00000000h]

ICLR is shown in Figure 32-7 and described in Table 32-19.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

Figure 32-7. 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 32-19. ICLR Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	W	0h	
3	PWRDWNIFG	W	0h	Clears PWRDWNIFG in RIS register 0h = Writing a 0 has no effect 1h = RIS bit corresponding to PWRUPIFG is cleared
2	PWRUPIFG	W	0h	Clears PWRUPIFG in RIS register 0h = Writing a 0 has no effect 1h = RIS bit corresponding to PWRUPIFG is cleared
1	RXIFG	W	0h	Clears RXIFG in RIS register 0h = Writing a 0 has no effect 1h = RIS bit corresponding to RXIFG is cleared
0	TXIFG	W	0h	Clears TXIFG in RIS register 0h = Writing a 0 has no effect 1h = RIS bit corresponding to TXIFG is cleared

32.3.7 EVT_MODE (Offset = 10E0h) [Reset = 00000001h]

EVT_MODE is shown in Figure 32-8 and described in Table 32-20.

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 32-8. 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 32-20. 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 0h = The interrupt or event line is disabled. 1h = The interrupt or event line is in software mode. Software must clear the RIS. 2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag.

32.3.8 DESC (Offset = 10FCh) [Reset = 03400000h]

DESC is shown in Figure 32-9 and described in Table 32-21.

Return to the Summary Table.

This register identifies the peripheral and its exact version.

Figure 32-9. 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 32-21. 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

32.3.9 TXD (Offset = 1100h) [Reset = 00000000h]

TXD is shown in Figure 32-10 and described in Table 32-22.

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 32-10. 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 32-22. 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

32.3.10 TXCTL (Offset = 1104h) [Reset = 00000000h]

TXCTL is shown in Figure 32-11 and described in Table 32-23.

Return to the Summary Table.

Transmit control register

Figure 32-11. 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 32-23. 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. 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. 0h = TXD is empty 1h = TXD is full

32.3.11 RXD (Offset = 1108h) [Reset = 00000000h]

RXD is shown in Figure 32-12 and described in Table 32-24.

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 32-12. 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 32-24. RXD Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RX_DATA	R/W	0h	Contains data written by SM/OW.

32.3.12 RXCTL (Offset = 110Ch) [Reset = 00000000h]

RXCTL is shown in Figure 32-13 and described in Table 32-25.

Return to the Summary Table.

Receive control register

Figure 32-13. 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 32-25. 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. A read of the DSSM.RXD register by SWD Access Port will clear the RX field. 0h = RXD empty 1h = RXD full

32.3.13 SPECIAL_AUTH (Offset = 1200h) [Reset = 00000013h]

SPECIAL_AUTH is shown in Figure 32-14 and described in Table 32-26.

Return to the Summary Table.

This register is used to control ET-AP, DFT-TAP, SWD, CFG-AP and SEC-AP.

Figure 32-14. 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 32-26. 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. 0h = Disable PWR-AP 1h = Enable PWR-AP
5	AHBAPEN	R	0h	Disabling / enabling debug access to the M0+ Core via the AHB-AP DAP bus isolation. 0h = Disable AHB-AP 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. 0h = Disable CFG-AP 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. 0h = Disable ET+ -AP 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. 0h = Disable DFT-TAP 1h = Enable DFT-TAP
1	SWDPORTEN	R	1h	When asserted, the SW-DP functions normally. When deasserted, the SW-DP effectively disables all external debug access. 0h = Disable SWD port 1h = Enable SWD port

Table 32-26. 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. 0h = Disable SEC-AP 1h = Enable SEC-AP

32.3.14 APP_AUTH (Offset = 1210h) [Reset = 00000000h]

APP_AUTH is shown in Figure 32-15 and described in Table 32-27.

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 32-15. 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 32-27. APP_AUTH Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3	SPNIDEN	R	0h	Secure noninvasive debug enable. 0h = Invasive debug disabled 1h = Invasive debug enabled
2	SPIDEN	R	0h	Secure invasive debug enable. 0h = Invasive debug disabled 1h = Invasive debug enabled
1	NIDEN	R	0h	Controls noninvasive debug enable. 0h = Non-invasive debug disabled 1h = Non-invasive debug enabled
0	DBGEN	R	0h	Controls invasive debug enable. 0h = Invasive debug disabled 1h = Invasive debug enabled

Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

DATE	REVISION	NOTES
December 2025	*	Initial Release

This page intentionally left blank.

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.

Last updated 10/2025