CC13x2、CC26x2 SimpleLink™ 无线 MCU

Technical Reference Manual

Literature Number: ZHDU116G
JANUARY 2018 – REVISED JUNE 2024

目录

使用前必读.....65
关于本手册.....65
器件.....65
寄存器、字段和位调用.....65
相关文档.....66
1.1 商标.....67
1 结构概述.....68
1.1 目标应用.....69
1.2 概述.....69
1.3 功能概述.....72
1.3.1 Arm® Cortex®-M4F.....72
1.3.2 片上存储器.....73
1.3.3 无线电.....73
1.3.4 安全内核.....74
1.3.5 通用计时器.....75
1.3.6 直接存储器存取.....75
1.3.7 系统控制和时钟.....76
1.3.8 串行通信外设.....76
1.3.9 可编程 I/O.....79
1.3.10 传感器控制器.....79
1.3.11 随机数生成器.....80
1.3.12 cJTAG 及 JTAG.....80
1.3.13 电源系统.....81
2 Arm® Cortex®-M4F 处理器.....83
2.1 Arm® Cortex®-M4F 处理器简介.....84
2.2 方框图.....84
2.3 概述.....85
2.3.1 系统级接口.....85
2.3.2 集成可配置调试.....85
2.3.3 跟踪端口接口单元.....85
2.3.4 浮点单元 (FPU).....86
2.3.5 存储器保护单元 (MPU).....86
2.3.6 Arm® Cortex®-M4F 系统组件详情.....86
2.4 编程模型.....87
2.4.1 软件执行的处理器模式和特权等级.....87
2.4.2 栈.....87
2.4.3 异常和中断.....87
2.4.4 数据类型.....87
2.5 Arm® Cortex®-M4F 内核寄存器.....87
2.5.1 内核寄存器映射.....89
2.5.2 内核寄存器描述.....89
2.6 指令集概要.....102
2.6.1 Arm® Cortex®-M4F 指令.....102
2.6.2 加载和存储时序.....109
2.6.3 与其他 Cortex® 处理器实现二进制兼容.....110
2.7 浮点单元 (FPU).....111

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.7.1 关于 FPU....111
2.7.2 FPU 功能说明....111
2.7.3 FPU 编程器模型....116
2.8 存储器保护单元 (MPU)....117
2.8.1 关于 MPU....117
2.8.2 MPU 功能说明....117
2.8.3 MPU 编程器模型....117
2.9 Arm® Cortex®-M4F 处理器寄存器....118
2.9.1 CPU_DWT 寄存器....119
2.9.2 CPU_FPB 寄存器....146
2.9.3 CPU_ITM 寄存器....157
2.9.4 CPU_SCS 寄存器....198
2.9.5 CPU_TPIU 寄存器....297
3 存储器映射....310
3.1 存储器映射....310
4 Arm® Cortex®-M4F 外设....312
4.1 Arm® Cortex®-M4F 外设介绍....313
4.2 功能说明....313
4.2.1 SysTick....314
4.2.2 NVIC....314
4.2.3 SCB....315
4.2.4 ITM....316
4.2.5 FPB....316
4.2.6 TPIU....316
4.2.7 DWT....317
5 中断和事件....318
5.1 异常模型....319
5.1.1 异常状态....319
5.1.2 异常类型....320
5.1.3 异常处理程序....322
5.1.4 矢量表....322
5.1.5 异常优先级....323
5.1.6 中断优先级分组....323
5.1.7 异常进入和返回....324
5.2 故障处理....326
5.2.1 故障类型....326
5.2.2 故障升级和硬故障....327
5.2.3 故障状态寄存器和故障地址寄存器....327
5.2.4 锁定....327
5.3 事件结构....328
5.3.1 简介....328
5.3.2 事件结构概述....329
5.4 AON 事件结构....330
5.4.1 通用输入事件列表....330
5.4.2 事件订阅者....330
5.5 MCU 事件结构....331
5.5.1 通用输入事件列表....331
5.5.2 事件订阅者....335
5.6 AON 事件....336
5.7 中断和事件寄存器....337
5.7.1 AON_EVENT 寄存器....338
5.7.2 EVENT 寄存器....354
6 JTAG 接口....480
6.1 顶层调试系统....481
6.2 cJTAG....483
6.2.1 cJTAG 命令....485
6.2.2 编程序列....488

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

6.3 ICEPick 489
6.3.1 次级 TAP 490
6.3.2 ICEPick 寄存器 491
6.3.3 路由器扫描链 494
6.3.4 TAP 路由寄存器 495
6.4 ICEMelter 499
6.5 串行线查看器 (SWV) 500
6.6 引导中暂停 (HIB) 500
6.7 调试与关断 500
6.8 通过 WUC TAP 支持的调试功能 501
6.9 分析器寄存器 501
6.10 边界扫描 503
7 电源、复位和时钟管理 (PRCM) 506
7.1 简介 507
7.2 系统 CPU 模式 508
7.3 电源系统 508
7.3.1 内部 DC/DC 转换器和全局 LDO 508
7.4 数字电源分区 509
7.4.1 MCU_VD 511
7.4.2 AON_VD 511
7.5 时钟管理 511
7.5.1 系统时钟 511
7.5.2 MCU_VD 中的时钟 515
7.5.3 AON_VD 中的时钟 517
7.6 电源模式 517
7.6.1 启动状态 518
7.6.2 活动模式 518
7.6.3 空闲模式 518
7.6.4 待机模式 519
7.6.5 关断模式 520
7.7 复位 520
7.7.1 系统复位 520
7.7.2 MCU_VD 电源域及模块的复位 521
7.7.3 AON_VD 复位 521
7.8 PRCM 寄存器 521
7.8.1 DDI_0_OSC 寄存器 522
7.8.2 PRCM 寄存器 543
7.8.3 AON_PMCTL 寄存器 622
8 多功能指令存储器系统 (VIMS) 636
8.1 简介 637
8.2 VIMS 配置 638
8.2.1 VIMS 模式 638
8.2.2 VIMS 闪存线路缓冲器 640
8.2.3 VIMS 仲裁 640
8.2.4 VIMS 高速缓存 TAG 预取 640
8.3 VIMS 软件注意事项 640
8.3.1 闪存程序或者更新 641
8.3.2 VIMS 保持 641
8.4 ROM 642
8.5 闪存 642
8.5.1 闪存保护 642
8.5.2 存储器编程 642
8.5.3 闪存编程 643
8.5.4 电源管理要求 643
8.6 ROM 功能 645
8.7 VIMS 寄存器 645

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1 FLASH 寄存器 646
8.7.2 VIMS 寄存器 776
9 SRAM 779
9.1 简介 780
9.2 主要特性 780
9.3 数据保留 780
9.4 奇偶校验和 SRAM 错误支持 780
9.5 SRAM 自动初始化 780
9.6 奇偶校验调试行为 780
9.7 SRAM 寄存器 780
9.7.1 SRAM_MMR 寄存器 781
9.7.2 SRAM 寄存器 786
10 引导加载程序 792
10.1 引导加载程序功能 793
10.1.1 引导加载程序禁用 793
10.1.2 引导加载程序后门 793
10.2 引导加载程序接口 793
10.2.1 数据包处理 794
10.2.2 传输层 796
10.2.3 串行总线命令 797
11 器件配置 808
11.1 客户配置 (CCFG) 809
11.2 CCFG 寄存器 809
11.2.1 CCFG 寄存器 810
11.3 出厂配置 (FCFG) 840
11.4 FCFG 寄存器 840
11.4.1 FCFG1 寄存器 841
12 加密 928
12.1 AES 和哈希加密处理器简介 929
12.2 功能说明 929
12.2.1 调试功能 930
12.2.2 异常处理 930
12.3 电源管理和睡眠模式 930
12.4 硬件说明 931
12.4.1 AHB 从总线 931
12.4.2 AHB 主总线 931
12.4.3 中断 931
12.5 模块说明 932
12.5.1 简介 932
12.5.2 模块存储器映射 932
12.5.3 DMA 控制器 935
12.5.4 主控制和选择模块 937
12.5.5 AES 引擎 939
12.5.6 键区寄存器 943
12.6 AES 模块性能 943
12.6.1 简介 943
12.6.2 基于 DMA 的操作的性能 945
12.7 编程指南 945
12.7.1 复位后的一次性初始化 945
12.7.2 DMAC 及主控 946
12.7.3 哈希 947
12.7.4 加密和解密 954
12.7.5 异常处理 961
12.8 约定与合规性 982
12.8.1 本手册中使用的约定 982
12.8.2 合规性 983

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9 加密寄存器 983
12.9.1 CRYPTO 寄存器 984
13 I/O 控制器 (IOC) 1085
13.1 简介 1086
13.2 IOC 概述 1086
13.3 I/O 映射和配置 1087
13.3.1 基本 I/O 映射 1087
13.3.2 将 AUXIO 映射到 DIO 引脚 1087
13.3.3 通过 I/O 控制外部 LNA/PA（范围扩展器） 1087
13.3.4 将 32kHz 系统时钟（LF 时钟）映射至 DIO 1088
13.4 DIO 引脚上的边沿检测 1088
13.4.1 将 DIO 配置为 GPIO 输入，生成 EDGE DETECT 上的中断 1088
13.5 未使用的 I/O 引脚 1089
13.6 GPIO 1089
13.7 I/O 引脚功能 1090
13.8 外设 PORTID 1092
13.9 I/O 引脚 1093
13.9.1 输入/输出模式 1093
13.10 IOC 寄存器 1094
13.10.1 AON_IOC 寄存器 1095
13.10.2 GPIO 寄存器 1101
13.10.3 IOC 寄存器 1124
14 微型直接存储器存取 (µDMA) 1286
14.1 µDMA 简介 1287
14.2 方框图 1288
14.3 功能说明 1288
14.3.1 通道分配 1289
14.3.2 优先级 1290
14.3.3 仲裁大小 1290
14.3.4 请求类型 1290
14.3.5 通道配置 1291
14.3.6 传输模式 1293
14.3.7 传输大小和增量 1301
14.3.8 外设接口 1301
14.3.9 软件请求 1301
14.3.10 中断和错误 1301
14.4 初始化和配置 1303
14.4.1 模块初始化 1303
14.4.2 配置存储器到存储器传输 1303
14.5 µDMA 寄存器 1304
14.5.1 UDMA 寄存器 1305
15 计时器 1327
15.1 通用计时器 1328
15.2 方框图 1329
15.3 功能说明 1329
15.3.1 GPTM 复位条件 1330
15.3.2 计时器模式 1330
15.3.3 同步 GPT 模块 1338
15.3.4 访问级联的 16 位和 32 位 GPTM 寄存器值 1339
15.4 初始化和配置 1339
15.4.1 单次触发和周期计时器模式 1340
15.4.2 输入边沿计数模式 1340
15.4.3 输入边沿计时模式 1341
15.4.4 PWM 模式 1341
15.4.5 生成 DMA 触发事件 1342
15.5 GPTM 寄存器 1342

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

15.5.1 GPT 寄存器 1343
16 实时时钟 (RTC) 1378
16.1 简介 1379
16.2 功能规格 1379
16.2.1 功能概述 1379
16.2.2 自由运行计数器 1379
16.2.3 通道 1380
16.2.4 事件 1381
16.3 RTC 寄存器信息 1381
16.3.1 寄存器访问 1381
16.3.2 进入睡眠和从睡眠中唤醒 1381
16.3.3 AON_RTC:SYNC 寄存器 1382
16.4 RTC 寄存器 1382
16.4.1 AON_RTC 寄存器 1383
17 看门狗计时器 (WDT) 1399
17.1 简介 1400
17.2 功能说明 1400
17.3 初始化和配置 1401
17.4 WDT 寄存器 1401
17.4.1 WDT 寄存器 1402
18 真随机数发生器 (TRNG) 1412
18.1 简介 1413
18.2 方框图 1413
18.3 TRNG 软件复位 1414
18.4 中断请求 1414
18.5 TRNG 操作说明 1415
18.5.1 TRNG 关断 1415
18.5.2 TRNG 警报 1415
18.5.3 TRNG 熵 1416
18.6 TRNG 底层编程指南 1417
18.6.1 初始化 1417
18.7 TRNG 寄存器 1419
18.7.1 TRNG 寄存器 1420
19 AUX 域传感器控制器和外设 1444
19.1 简介 1445
19.1.1 AUX 方框图 1445
19.2 电源和时钟管理 1447
19.2.1 工作模式 1447
19.2.2 使用场景 1449
19.2.3 SCE 时钟仿真 1450
19.2.4 AUX RAM 保持 1450
19.3 传感器控制器 1450
19.3.1 Sensor Controller Studio 1450
19.3.2 传感器控制器引擎 (SCE) 1455
19.4 数字外设模块 1467
19.4.1 概述 1467
19.4.2 AIODIO 1467
19.4.3 SMPH 1470
19.4.4 SPIM 1470
19.4.5 时间数字转换器 (TDC) 1474
19.4.6 Timer01 1484
19.4.7 Timer2 1485
19.5 模拟外设模块 1493
19.5.1 概述 1493
19.5.2 模数转换器 (ADC) 1495
19.5.3 COMPA 1498
19.5.4 COMPB 1500

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.5.5 参考 DAC 1503
19.5.6 ISRC 1507
19.6 事件路由和使用 1509
19.6.1 AUX 事件总线 1509
19.6.2 在外部引脚上观察事件 1512
19.6.3 来自 MCU 域的事件 1512
19.6.4 发送到 MCU 域的事件 1512
19.6.5 来自 AON 域的事件 1513
19.6.6 发送到 AON 域的事件 1514
19.6.7 µDMA 接口 1514
19.7 传感器控制器别名寄存器空间 1515
19.8 AUX 域传感器控制器和外设寄存器 1520
19.8.1 ADI_4_AUX 寄存器 1521
19.8.2 AUX_AIODIO 寄存器 1534
19.8.3 AUX_EVCTL 寄存器 1558
19.8.4 AUX_SMPH 寄存器 1600
19.8.5 AUX_TDC 寄存器 1610
19.8.6 AUX_TIMER01 寄存器 1628
19.8.7 AUX_TIMER2 寄存器 1641
19.8.8 AUX_ANAIF 寄存器 1683
19.8.9 AUX_SYSIF 寄存器 1699
19.8.10 AUX_SPIM 寄存器 1751
19.8.11 AUX_MAC 寄存器 1761
19.8.12 AUX_SCE 寄存器 1803
20 电池监测器和温度传感器 (BATMON) 1814
20.1 简介 1815
20.2 功能说明 1815
20.3 BATMON 寄存器 1815
20.3.1 AON_BATMON 寄存器 1816
21 通用异步收发器 (UART) 1837
21.1 简介 1838
21.2 方框图 1839
21.3 信号描述 1839
21.4 功能说明 1839
21.4.1 发送和接收逻辑 1840
21.4.2 波特率生成 1840
21.4.3 数据传输 1840
21.4.4 调制解调器握手支持 1841
21.4.5 FIFO 操作 1842
21.4.6 中断 1842
21.4.7 环回操作 1843
21.5 用于连接 DMA 的接口 1844
21.6 初始化和配置 1845
21.7 UART 寄存器 1846
21.7.1 UART 寄存器 1847
22 同步串行接口 (SSI) 1868
22.1 简介 1869
22.2 方框图 1870
22.3 信号描述 1871
22.4 功能说明 1871
22.4.1 比特率生成 1871
22.4.2 FIFO 操作 1872
22.4.3 中断 1872
22.4.4 帧格式 1873
22.5 DMA 操作 1880
22.6 初始化和配置 1880

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

22.7 SSI 寄存器 1881
22.7.1 SSI 寄存器 1882
23 内部集成电路 (I2C) 1893
23.1 简介 1894
23.2 方框图 1894
23.3 功能说明 1895
23.3.1 I²C 总线功能概述 1895
23.3.2 可用速度模式 1897
23.3.3 中断 1898
23.3.4 环回操作 1898
23.3.5 命令序列流程图 1899
23.4 初始化和配置 1906
23.5 I²C 寄存器 1906
23.5.1 I2C 寄存器 1907
24 IC 间音频 (I²S) 1928
24.1 简介 1929
24.2 方框图 1930
24.3 信号描述 1931
24.4 功能说明 1931
24.4.1 依赖项 1931
24.4.2 引脚配置 1932
24.4.3 串行格式配置 1932
24.4.4 I²S 1932
24.4.5 左对齐 (LJF) 1933
24.4.6 右对齐 (RJF) 1934
24.4.7 DSP 1935
24.4.8 时钟配置 1936
24.5 存储器接口 1937
24.5.1 采样字长 1937
24.5.2 通道映射 1937
24.5.3 存储器中的采样存储 1938
24.5.4 DMA 运行 1939
24.6 采样时间戳发生器 1940
24.6.1 样片戳计数器 1941
24.6.2 启动触发 1941
24.6.3 采样时间戳捕获 1942
24.6.4 实现恒定音频延迟 1942
24.7 错误检测 1943
24.8 用途 1943
24.8.1 启动序列 1943
24.8.2 关断序列 1943
24.9 I²S 寄存器 1943
24.9.1 I2S 寄存器 1944
25 无线电 1975
25.1 RF 内核 1976
25.1.1 概要说明和概述 1976
25.2 无线电门铃 1978
25.2.1 特殊引导过程 1978
25.2.2 命令及状态寄存器和事件 1979
25.2.3 RF 内核中断 1979
25.2.4 无线电计时器 1980
25.3 RF 内核 HAL 1981
25.3.1 硬件支持 1981
25.3.2 固件支持 1981
25.3.3 命令定义 1994
25.3.4 数据队列操作的立即命令 2017

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.4 数据队列使用情况 2020
25.4.1 数据队列上的操作仅可用于内部无线电 CPU 操作 2020
25.4.2 无线电 CPU 使用模型 2023
25.5 IEEE 802.15.4 2024
25.5.1 IEEE 802.15.4 命令 2024
25.5.2 中断 2033
25.5.3 数据处理 2033
25.5.4 无线电操作命令 2034
25.5.5 即时命令 2047
25.6 低功耗 Bluetooth® 2048
25.6.1 低功耗 Bluetooth® 命令 2048
25.6.2 中断 2067
25.7 数据处理 2068
25.7.1 接收缓冲器 2068
25.7.2 发送缓冲器 2068
25.8 无线电操作命令说明 2069
25.8.1 Bluetooth®5 无线电设置命令 2069
25.8.2 用于低功耗 Bluetooth® 数据包传输的无线电操作命令 2069
25.8.3 编码 PHY 的编码选择 2071
25.8.4 参数覆盖 2072
25.8.5 链路层连接 2072
25.8.6 从器件命令 2077
25.8.7 主器件命令 2078
25.8.8 传统广播器 2079
25.8.9 Bluetooth® 5 广播器命令 2083
25.8.10 扫描器命令 2088
25.8.11 启动器命令 2095
25.8.12 通用接收器命令 2101
25.8.13 PHY 测试发送命令 2102
25.8.14 白名单处理 2104
25.8.15 退避过程 2104
25.8.16 AUX 指针处理 2105
25.8.17 器件地址的动态更改 2105
25.9 即时命令 2106
25.9.1 更新广播有效载荷命令 2106
25.10 专有无线电 2106
25.10.1 数据包格式 2106
25.10.2 命令 2106
25.10.3 中断 2114
25.10.4 数据处理 2114
25.10.5 无线电操作命令说明 2115
25.10.6 即时命令 2126
25.11 无线电寄存器 2126
25.11.1 RFC_RAT 寄存器 2127
25.11.2 RFC_DBELL 寄存器 2137
25.11.3 RFC_PWR 寄存器 2153
26 修订历史记录 2155

插图清单

图 1-1. CC13x2 和 CC26x2 方框图 70
图 1-2. CC13x2 和 CC26x2 电源系统 81
图 2-1. CPU 方框图 85
图 2-2. TPIU 方框图 86
图 2-3. Arm® Cortex®-M4F 寄存器组 88
图 2-4. FPU 寄存器组 111
图 2-5. CTRL 寄存器 121

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

图 2-6. CYCCNT 寄存器 123
图 2-7. CPICNT 寄存器 124
图 2-8. EXCCNT 寄存器 125
图 2-9. SLEEPCNT 寄存器 126
图 2-10. LSUCNT 寄存器 127
图 2-11. FOLDCNT 寄存器 128
图 2-12. PCSR 寄存器 129
图 2-13. COMP0 寄存器 130
图 2-14. MASK0 寄存器 131
图 2-15. FUNCTION0 寄存器 132
图 2-16. COMP1 寄存器 134
图 2-17. MASK1 寄存器 135
图 2-18. FUNCTION1 寄存器 136
图 2-19. COMP2 寄存器 138
图 2-20. MASK2 寄存器 139
图 2-21. FUNCTION2 寄存器 140
图 2-22. COMP3 寄存器 142
图 2-23. MASK3 寄存器 143
图 2-24. FUNCTION3 寄存器 144
图 2-25. CTRL 寄存器 147
图 2-26. REMAP 寄存器 148
图 2-27. COMP0 寄存器 149
图 2-28. COMP1 寄存器 150
图 2-29. COMP2 寄存器 151
图 2-30. COMP3 寄存器 152
图 2-31. COMP4 寄存器 153
图 2-32. COMP5 寄存器 154
图 2-33. COMP6 寄存器 155
图 2-34. COMP7 寄存器 156
图 2-35. STIM0 寄存器 159
图 2-36. STIM1 寄存器 160
图 2-37. STIM2 寄存器 161
图 2-38. STIM3 寄存器 162
图 2-39. STIM4 寄存器 163
图 2-40. STIM5 寄存器 164
图 2-41. STIM6 寄存器 165
图 2-42. STIM7 寄存器 166
图 2-43. STIM8 寄存器 167
图 2-44. STIM9 寄存器 168
图 2-45. STIM10 寄存器 169
图 2-46. STIM11 寄存器 170
图 2-47. STIM12 寄存器 171
图 2-48. STIM13 寄存器 172
图 2-49. STIM14 寄存器 173
图 2-50. STIM15 寄存器 174
图 2-51. STIM16 寄存器 175
图 2-52. STIM17 寄存器 176
图 2-53. STIM18 寄存器 177
图 2-54. STIM19 寄存器 178
图 2-55. STIM20 寄存器 179
图 2-56. STIM21 寄存器 180
图 2-57. STIM22 寄存器 181
图 2-58. STIM23 寄存器 182
图 2-59. STIM24 寄存器 183
图 2-60. STIM25 寄存器 184
图 2-61. STIM26 寄存器 185

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

图 2-62. STIM27 寄存器.....186
图 2-63. STIM28 寄存器.....187
图 2-64. STIM29 寄存器.....188
图 2-65. STIM30 寄存器.....189
图 2-66. STIM31 寄存器.....190
图 2-67. TER 寄存器.....191
图 2-68. TPR 寄存器.....193
图 2-69. TCR 寄存器.....194
图 2-70. LAR 寄存器.....196
图 2-71. LSR 寄存器.....197
图 2-72. ICTR 寄存器.....202
图 2-73. ACTLR 寄存器.....203
图 2-74. STCSR 寄存器.....204
图 2-75. STRVR 寄存器.....205
图 2-76. STCVR 寄存器.....206
图 2-77. STCR 寄存器.....207
图 2-78. NVIC_ISER0 寄存器.....208
图 2-79. NVIC_ISER1 寄存器.....211
图 2-80. NVIC_ICER0 寄存器.....212
图 2-81. NVIC_ICER1 寄存器.....215
图 2-82. NVIC_ISPR0 寄存器.....216
图 2-83. NVIC_ISPR1 寄存器.....219
图 2-84. NVIC_ICPR0 寄存器.....220
图 2-85. NVIC_ICPR1 寄存器.....223
图 2-86. NVIC_IABR0 寄存器.....224
图 2-87. NVIC_IABR1 寄存器.....227
图 2-88. NVIC_IPR0 寄存器.....228
图 2-89. NVIC_IPR1 寄存器.....229
图 2-90. NVIC_IPR2 寄存器.....230
图 2-91. NVIC_IPR3 寄存器.....231
图 2-92. NVIC_IPR4 寄存器.....232
图 2-93. NVIC_IPR5 寄存器.....233
图 2-94. NVIC_IPR6 寄存器.....234
图 2-95. NVIC_IPR7 寄存器.....235
图 2-96. NVIC_IPR8 寄存器.....236
图 2-97. NVIC_IPR9 寄存器.....237
图 2-98. CPUID 寄存器.....238
图 2-99. ICSR 寄存器.....239
图 2-100. VTOR 寄存器.....241
图 2-101. AIRCR 寄存器.....242
图 2-102. SCR 寄存器.....243
图 2-103. CCR 寄存器.....244
图 2-104. SHPR1 寄存器.....246
图 2-105. SHPR2 寄存器.....247
图 2-106. SHPR3 寄存器.....248
图 2-107. SHCSR 寄存器.....249
图 2-108. CFSR 寄存器.....251
图 2-109. HFSR 寄存器.....253
图 2-110. DFSR 寄存器.....254
图 2-111. MMFAR 寄存器.....255
图 2-112. BFAR 寄存器.....256
图 2-113. AFSR 寄存器.....257
图 2-114. ID_PFR0 寄存器.....258
图 2-115. ID_PFR1 寄存器.....259
图 2-116. ID_DFR0 寄存器.....260
图 2-117. ID_AFR0 寄存器.....261

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

图 2-118. ID_MMFR0 寄存器 262
图 2-119. ID_MMFR1 寄存器 263
图 2-120. ID_MMFR2 寄存器 264
图 2-121. ID_MMFR3 寄存器 265
图 2-122. ID_ISAR0 寄存器 266
图 2-123. ID_ISAR1 寄存器 267
图 2-124. ID_ISAR2 寄存器 268
图 2-125. ID_ISAR3 寄存器 269
图 2-126. ID_ISAR4 寄存器 270
图 2-127. CPACR 寄存器 271
图 2-128. MPU_TYPE 寄存器 272
图 2-129. MPU_CTRL 寄存器 273
图 2-130. MPU_RNR 寄存器 274
图 2-131. MPU_RBAR 寄存器 275
图 2-132. MPU_RASR 寄存器 276
图 2-133. MPU_RBAR_A1 寄存器 278
图 2-134. MPU_RASR_A1 寄存器 279
图 2-135. MPU_RBAR_A2 寄存器 280
图 2-136. MPU_RASR_A2 寄存器 281
图 2-137. MPU_RBAR_A3 寄存器 282
图 2-138. MPU_RASR_A3 寄存器 283
图 2-139. DHCSR 寄存器 284
图 2-140. DCRSR 寄存器 286
图 2-141. DCRDR 寄存器 287
图 2-142. DEMCR 寄存器 288
图 2-143. STIR 寄存器 290
图 2-144. FPCCR 寄存器 291
图 2-145. FPCAR 寄存器 293
图 2-146. FPDSCR 寄存器 294
图 2-147. MVFR0 寄存器 295
图 2-148. MVFR1 寄存器 296
图 2-149. SSPSR 寄存器 298
图 2-150. CSPSR 寄存器 299
图 2-151. ACPR 寄存器 300
图 2-152. SPPR 寄存器 301
图 2-153. FFSR 寄存器 302
图 2-154. FFCR 寄存器 303
图 2-155. FSCR 寄存器 304
图 2-156. CLAIMMASK 寄存器 305
图 2-157. CLAIMSET 寄存器 306
图 2-158. CLAIMTAG 寄存器 307
图 2-159. CLAIMCLR 寄存器 308
图 2-160. DEVID 寄存器 309
图 5-1. 矢量表 322
图 5-2. 异常栈帧 324
图 5-3. 事件结构概念 328
图 5-4. 事件结构概述（简化） 329
图 5-5. AON 事件结构中的 WUC 订阅者 330
图 5-6. MCUWUSEL 寄存器 339
图 5-7. MCUWUSEL1 寄存器 344
图 5-8. EVTOMCUSEL 寄存器 349
图 5-9. RTCSEL 寄存器 353
图 5-10. CPUIRQSEL0 寄存器 359
图 5-11. CPUIRQSEL1 寄存器 360
图 5-12. CPUIRQSEL2 寄存器 361
图 5-13. CPUIRQSEL3 寄存器 362

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

图 5-14. CPUIRQSEL4 寄存器 363
图 5-15. CPUIRQSEL5 寄存器 364
图 5-16. CPUIRQSEL6 寄存器 365
图 5-17. CPUIRQSEL7 寄存器 366
图 5-18. CPUIRQSEL8 寄存器 367
图 5-19. CPUIRQSEL9 寄存器 368
图 5-20. CPUIRQSEL10 寄存器 369
图 5-21. CPUIRQSEL11 寄存器 370
图 5-22. CPUIRQSEL12 寄存器 371
图 5-23. CPUIRQSEL13 寄存器 372
图 5-24. CPUIRQSEL14 寄存器 373
图 5-25. CPUIRQSEL15 寄存器 374
图 5-26. CPUIRQSEL16 寄存器 375
图 5-27. CPUIRQSEL17 寄存器 376
图 5-28. CPUIRQSEL18 寄存器 377
图 5-29. CPUIRQSEL19 寄存器 378
图 5-30. CPUIRQSEL20 寄存器 379
图 5-31. CPUIRQSEL21 寄存器 380
图 5-32. CPUIRQSEL22 寄存器 381
图 5-33. CPUIRQSEL23 寄存器 382
图 5-34. CPUIRQSEL24 寄存器 383
图 5-35. CPUIRQSEL25 寄存器 384
图 5-36. CPUIRQSEL26 寄存器 385
图 5-37. CPUIRQSEL27 寄存器 386
图 5-38. CPUIRQSEL28 寄存器 387
图 5-39. CPUIRQSEL29 寄存器 388
图 5-40. CPUIRQSEL30 寄存器 389
图 5-41. CPUIRQSEL31 寄存器 391
图 5-42. CPUIRQSEL32 寄存器 392
图 5-43. CPUIRQSEL33 寄存器 393
图 5-44. CPUIRQSEL34 寄存器 394
图 5-45. CPUIRQSEL35 寄存器 395
图 5-46. CPUIRQSEL36 寄存器 396
图 5-47. CPUIRQSEL37 寄存器 397
图 5-48. RFCSEL0 寄存器 398
图 5-49. RFCSEL1 寄存器 399
图 5-50. RFCSEL2 寄存器 400
图 5-51. RFCSEL3 寄存器 401
图 5-52. RFCSEL4 寄存器 402
图 5-53. RFCSEL5 寄存器 403
图 5-54. RFCSEL6 寄存器 404
图 5-55. RFCSEL7 寄存器 405
图 5-56. RFCSEL8 寄存器 406
图 5-57. RFCSEL9 寄存器 407
图 5-58. GPT0ACAPTSEL 寄存器 409
图 5-59. GPT0BCAPTSEL 寄存器 412
图 5-60. GPT1ACAPTSEL 寄存器 415
图 5-61. GPT1BCAPTSEL 寄存器 418
图 5-62. GPT2ACAPTSEL 寄存器 421
图 5-63. GPT2BCAPTSEL 寄存器 424
图 5-64. UDMACH1SSEL 寄存器 427
图 5-65. UDMACH1BSEL 寄存器 428
图 5-66. UDMACH2SSEL 寄存器 429
图 5-67. UDMACH2BSEL 寄存器 430
图 5-68. UDMACH3SSEL 寄存器 431
图 5-69. UDMACH3BSEL 寄存器 432

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

图 5-70. UDMACH4SSEL 寄存器 433
图 5-71. UDMACH4BSEL 寄存器 434
图 5-72. UDMACH5SSEL 寄存器 435
图 5-73. UDMACH5BSEL 寄存器 436
图 5-74. UDMACH6SSEL 寄存器 437
图 5-75. UDMACH6BSEL 寄存器 438
图 5-76. UDMACH7SSEL 寄存器 439
图 5-77. UDMACH7BSEL 寄存器 440
图 5-78. UDMACH8SSEL 寄存器 441
图 5-79. UDMACH8BSEL 寄存器 442
图 5-80. UDMACH9SSEL 寄存器 443
图 5-81. UDMACH9BSEL 寄存器 444
图 5-82. UDMACH10SSEL 寄存器 445
图 5-83. UDMACH10BSEL 寄存器 446
图 5-84. UDMACH11SSEL 寄存器 447
图 5-85. UDMACH11BSEL 寄存器 448
图 5-86. UDMACH12SSEL 寄存器 449
图 5-87. UDMACH12BSEL 寄存器 450
图 5-88. UDMACH13BSEL 寄存器 451
图 5-89. UDMACH14BSEL 寄存器 452
图 5-90. UDMACH15BSEL 寄存器 456
图 5-91. UDMACH16SSEL 寄存器 457
图 5-92. UDMACH16BSEL 寄存器 458
图 5-93. UDMACH17SSEL 寄存器 459
图 5-94. UDMACH17BSEL 寄存器 460
图 5-95. UDMACH21SSEL 寄存器 461
图 5-96. UDMACH21BSEL 寄存器 462
图 5-97. UDMACH22SSEL 寄存器 463
图 5-98. UDMACH22BSEL 寄存器 464
图 5-99. UDMACH23SSEL 寄存器 465
图 5-100. UDMACH23BSEL 寄存器 466
图 5-101. UDMACH24SSEL 寄存器 467
图 5-102. UDMACH24BSEL 寄存器 468
图 5-103. GPT3ACAPTSEL 寄存器 469
图 5-104. GPT3BCAPTSEL 寄存器 472
图 5-105. AUXSEL0 寄存器 475
图 5-106. CM3NMISEL0 寄存器 476
图 5-107. I2SSTMPSEL0 寄存器 477
图 5-108. FRZSEL0 寄存器 478
图 5-109. SWEV 寄存器 479
图 6-1. 顶层调试系统 481
图 6-2. JTAG 状态机 482
图 6-3. cJTAG 概念图 483
图 6-4. 数据移位寄存器 492
图 6-5. 指令寄存器 492
图 6-6. 旁路寄存器 492
图 6-7. 器件标识寄存器 492
图 6-8. 用户代码寄存器 493
图 6-9. ICEPick 标识寄存器 493
图 6-10. 连接寄存器 494
图 6-11. ROUTER DR 扫描链 494
图 6-12. 分析器寄存器 501
图 6-13. 边界扫描单元 503
图 7-1. 节能特性的层次结构 507
图 7-2. CC13x2 和 CC26x2 电源系统 509
图 7-3. CC13x2 和 CC26x2 中的数字电源分区 510

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

图 7-4. 时钟源 511
图 7-5. 系统时钟多路复用 514
图 7-6. MCU_VD 中的时钟 516
图 7-7. CTL0 寄存器 523
图 7-8. CTL1 寄存器 525
图 7-9. RADCEXTCFG 寄存器 526
图 7-10. AMPCOMPCTL 寄存器 527
图 7-11. AMPCOMPTH1 寄存器 528
图 7-12. AMPCOMPTH2 寄存器 529
图 7-13. ANABYPASSVAL1 寄存器 530
图 7-14. ANABYPASSVAL2 寄存器 531
图 7-15. ATESTCTL 寄存器 532
图 7-16. ADCDOUBLERNANOAMPCTL 寄存器 533
图 7-17. XOSCHFCTL 寄存器 534
图 7-18. LFOSCCTL 寄存器 535
图 7-19. RCOSCHFCTL 寄存器 536
图 7-20. RCOSCMFCTL 寄存器 537
图 7-21. STAT0 寄存器 538
图 7-22. STAT1 寄存器 540
图 7-23. STAT2 寄存器 542
图 7-24. INFRCLKDIVR 寄存器 547
图 7-25. INFRCLKDIVS 寄存器 548
图 7-26. INFRCLKDIVDS 寄存器 549
图 7-27. VDCTL 寄存器 550
图 7-28. CLKLOADCTL 寄存器 551
图 7-29. RFCCLKG 寄存器 553
图 7-30. VIMSCLKG 寄存器 554
图 7-31. SECDMACLKGR 寄存器 555
图 7-32. SECDMACLKGS 寄存器 557
图 7-33. SECDMACLKGDS 寄存器 558
图 7-34. GPIOCLKGR 寄存器 559
图 7-35. GPIOCLKGS 寄存器 560
图 7-36. GPIOCLKGDS 寄存器 561
图 7-37. GPTCLKGR 寄存器 562
图 7-38. GPTCLKGS 寄存器 563
图 7-39. GPTCLKGDS 寄存器 564
图 7-40. I2CCLKGR 寄存器 565
图 7-41. I2CCLKGS 寄存器 566
图 7-42. I2CCLKGDS 寄存器 567
图 7-43. UARTCLKGR 寄存器 568
图 7-44. UARTCLKGS 寄存器 569
图 7-45. UARTCLKGDS 寄存器 570
图 7-46. SSICLKGR 寄存器 571
图 7-47. SSICLKGS 寄存器 572
图 7-48. SSICLKGDS 寄存器 573
图 7-49. I2SCLKGR 寄存器 574
图 7-50. I2SCLKGS 寄存器 575
图 7-51. I2SCLKGDS 寄存器 576
图 7-52. SYSBUSCLKDIV 寄存器 577
图 7-53. CPUCLKDIV 寄存器 578
图 7-54. PERBUSCPUCLKDIV 寄存器 579
图 7-55. PERDMACLKDIV 寄存器 580
图 7-56. I2SBCLKSEL 寄存器 581
图 7-57. GPTCLKDIV 寄存器 582
图 7-58. I2SCLKCTL 寄存器 583
图 7-59. I2SMCLKDIV 寄存器 584

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

图 7-60. I2SBCLKDIV 寄存器 585
图 7-61. I2SWCLKDIV 寄存器 586
图 7-62. RESETSECDMA 寄存器 587
图 7-63. RESETGPIO 寄存器 588
图 7-64. RESETGPT 寄存器 589
图 7-65. RESETI2C 寄存器 590
图 7-66. RESETUART 寄存器 591
图 7-67. RESETSSI 寄存器 592
图 7-68. RESETI2S 寄存器 593
图 7-69. PDCTL0 寄存器 594
图 7-70. PDCTL0RFC 寄存器 595
图 7-71. PDCTL0SERIAL 寄存器 596
图 7-72. PDCTL0PERIPH 寄存器 597
图 7-73. PDSTAT0 寄存器 598
图 7-74. PDSTAT0RFC 寄存器 599
图 7-75. PDSTAT0SERIAL 寄存器 600
图 7-76. PDSTAT0PERIPH 寄存器 601
图 7-77. PDCTL1 寄存器 602
图 7-78. PDCTL1CPU 寄存器 603
图 7-79. PDCTL1RFC 寄存器 604
图 7-80. PDCTL1VIMS 寄存器 605
图 7-81. PDSTAT1 寄存器 606
图 7-82. PDSTAT1BUS 寄存器 607
图 7-83. PDSTAT1RFC 寄存器 608
图 7-84. PDSTAT1CPU 寄存器 609
图 7-85. PDSTAT1VIMS 寄存器 610
图 7-86. RFCBITS 寄存器 611
图 7-87. RFCMODESEL 寄存器 612
图 7-88. RFCMODEHWOPT 寄存器 613
图 7-89. PWRPROFSTAT 寄存器 614
图 7-90. MCUSRAMCFG 寄存器 615
图 7-91. RAMRETEN 寄存器 616
图 7-92. OSCIMSC 寄存器 617
图 7-93. OSCRIS 寄存器 618
图 7-94. OSCICR 寄存器 621
图 7-95. AUXSCECLK 寄存器 623
图 7-96. RAMCFG 寄存器 624
图 7-97. PWRCTL 寄存器 625
图 7-98. PWRSTAT 寄存器 626
图 7-99. SHUTDOWN 寄存器 627
图 7-100. RECHARGECFG 寄存器 628
图 7-101. RECHARGESTAT 寄存器 629
图 7-102. OSCCFG 寄存器 630
图 7-103. RESETCTL 寄存器 631
图 7-104. SLEEPCTL 寄存器 633
图 7-105. JTAGCFG 寄存器 634
图 7-106. JTAGUSERCODE 寄存器 635
图 8-1. VIMS 概述 637
图 8-2. VIMS 模式切换流程图 638
图 8-3. GPRAM 模式下的 VIMS 模块 638
图 8-4. VIMS 模块处于关闭模式 639
图 8-5. 缓存模式下的 VIMS 模块 639
图 8-6. 无 RAM 保留的软件预防措施 641
图 8-7. GPRAM 保持 642
图 8-8. 闪存电源状态 644
图 8-9. STAT 寄存器 652

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

图 8-10. CFG 寄存器.....653
图 8-11. SYSCODE_START 寄存器.....654
图 8-12. FLASH_SIZE 寄存器.....655
图 8-13. FWLOCK 寄存器.....656
图 8-14. FWFLAG 寄存器.....657
图 8-15. EFUSE 寄存器.....658
图 8-16. EFUSEADDR 寄存器.....659
图 8-17. DATAUPPER 寄存器.....660
图 8-18. DATALOWER 寄存器.....661
图 8-19. EFUSECFG 寄存器.....662
图 8-20. EFUSESTAT 寄存器.....663
图 8-21. ACC 寄存器.....664
图 8-22. BOUNDARY 寄存器.....665
图 8-23. EFUSEFLAG 寄存器.....666
图 8-24. EFUSEKEY 寄存器.....667
图 8-25. EFUSERELEASE 寄存器.....668
图 8-26. EFUSEPINS 寄存器.....669
图 8-27. EFUSECRA 寄存器.....670
图 8-28. EFUSEREAD 寄存器.....671
图 8-29. EFUSEPROGRAM 寄存器.....672
图 8-30. EFUSEERROR 寄存器.....673
图 8-31. SINGLEBIT 寄存器.....674
图 8-32. TWOBIT 寄存器.....675
图 8-33. SELFTESTCYC 寄存器.....676
图 8-34. SELFTESTSIGN 寄存器.....677
图 8-35. FRDCTL 寄存器.....678
图 8-36. FSPRD 寄存器.....679
图 8-37. FEDACCTL1 寄存器.....680
图 8-38. FEDACSTAT 寄存器.....681
图 8-39. FBPROT 寄存器.....682
图 8-40. FBSE 寄存器.....683
图 8-41. FBBUSY 寄存器.....684
图 8-42. FBAC 寄存器.....685
图 8-43. FBFALLBACK 寄存器.....686
图 8-44. FBPRDY 寄存器.....687
图 8-45. FPAC1 寄存器.....688
图 8-46. FPAC2 寄存器.....689
图 8-47. FMAC 寄存器.....690
图 8-48. FMSTAT 寄存器.....691
图 8-49. FLOCK 寄存器.....692
图 8-50. FVREADCT 寄存器.....693
图 8-51. FVHVCT1 寄存器.....694
图 8-52. FVHVCT2 寄存器.....695
图 8-53. FVHVCT3 寄存器.....696
图 8-54. FVNVCT 寄存器.....697
图 8-55. FVSLP 寄存器.....698
图 8-56. FVWLCT 寄存器.....699
图 8-57. FEFUSECTL 寄存器.....700
图 8-58. FEFUSESTAT 寄存器.....701
图 8-59. FEFUSEDATA 寄存器.....702
图 8-60. FSEQPMP 寄存器.....703
图 8-61. FBSTROBES 寄存器.....704
图 8-62. FPSTROBES 寄存器.....705
图 8-63. FBMODE 寄存器.....706
图 8-64. FTCR 寄存器.....707
图 8-65. FADDR 寄存器.....708

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

图 8-66. FTCTL 寄存器 709
图 8-67. FWPWRITE0 寄存器 710
图 8-68. FWPWRITE1 寄存器 711
图 8-69. FWPWRITE2 寄存器 712
图 8-70. FWPWRITE3 寄存器 713
图 8-71. FWPWRITE4 寄存器 714
图 8-72. FWPWRITE5 寄存器 715
图 8-73. FWPWRITE6 寄存器 716
图 8-74. FWPWRITE7 寄存器 717
图 8-75. FWPWRITE_ECC 寄存器 718
图 8-76. FSWSTAT 寄存器 719
图 8-77. FSM_GLBCTL 寄存器 720
图 8-78. FSM_STATE 寄存器 721
图 8-79. FSM_STAT 寄存器 722
图 8-80. FSM_CMD 寄存器 723
图 8-81. FSM_PE_OSU 寄存器 724
图 8-82. FSM_VSTAT 寄存器 725
图 8-83. FSM_PE_VSU 寄存器 726
图 8-84. FSM_CMP_VSU 寄存器 727
图 8-85. FSM_EX_VAL 寄存器 728
图 8-86. FSM_RD_H 寄存器 729
图 8-87. FSM_P_OH 寄存器 730
图 8-88. FSM_ERA_OH 寄存器 731
图 8-89. FSM_SAV_PPUL 寄存器 732
图 8-90. FSM_PE_VH 寄存器 733
图 8-91. FSM_PRG_PW 寄存器 734
图 8-92. FSM_ERA_PW 寄存器 735
图 8-93. FSM_SAV_ERA_PUL 寄存器 736
图 8-94. FSM_TIMER 寄存器 737
图 8-95. FSM_MODE 寄存器 738
图 8-96. FSM_PGM 寄存器 739
图 8-97. FSM_ERA 寄存器 740
图 8-98. FSM_PRG_PUL 寄存器 741
图 8-99. FSM_ERA_PUL 寄存器 742
图 8-100. FSM_STEP_SIZE 寄存器 743
图 8-101. FSM_PUL_CNTR 寄存器 744
图 8-102. FSM_EC_STEP_HEIGHT 寄存器 745
图 8-103. FSM_ST_MACHINE 寄存器 746
图 8-104. FSM_FLES 寄存器 747
图 8-105. FSM_WR_ENA 寄存器 748
图 8-106. FSM_ACC_PP 寄存器 749
图 8-107. FSM_ACC_EP 寄存器 750
图 8-108. FSM_ADDR 寄存器 751
图 8-109. FSM_SECTOR 寄存器 752
图 8-110. FMC_REV_ID 寄存器 753
图 8-111. FSM_ERR_ADDR 寄存器 754
图 8-112. FSM_PGM_MAXPUL 寄存器 755
图 8-113. FSM_EXECUTE 寄存器 756
图 8-114. FSM_SECTOR1 寄存器 757
图 8-115. FSM_SECTOR2 寄存器 758
图 8-116. FSM_BSLE0 寄存器 759
图 8-117. FSM_BSLE1 寄存器 760
图 8-118. FSM_BSLP0 寄存器 761
图 8-119. FSM_BSLP1 寄存器 762
图 8-120. FSM_PGM128 寄存器 763
图 8-121. FCFG_BANK 寄存器 764

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

图 8-122. FCFG_WRAPPER 寄存器 765
图 8-123. FCFG_BNK_TYPE 寄存器 766
图 8-124. FCFG_B0_START 寄存器 767
图 8-125. FCFG_B1_START 寄存器 768
图 8-126. FCFG_B2_START 寄存器 769
图 8-127. FCFG_B3_START 寄存器 770
图 8-128. FCFG_B4_START 寄存器 771
图 8-129. FCFG_B5_START 寄存器 772
图 8-130. FCFG_B6_START 寄存器 773
图 8-131. FCFG_B7_START 寄存器 774
图 8-132. FCFG_B0_SSIZE0 寄存器 775
图 8-133. STAT 寄存器 777
图 8-134. CTL 寄存器 778
图 9-1. PER_CTL 寄存器 782
图 9-2. PER_CHK 寄存器 783
图 9-3. PER_DBG 寄存器 784
图 9-4. MEM_CTL 寄存器 785
图 9-5. BANK0_y 寄存器 787
图 9-6. BANK1_y 寄存器 788
图 9-7. BANK2_y 寄存器 789
图 9-8. BANK3_y 寄存器 790
图 9-9. BANK4_y 寄存器 791
图 10-1. 发送和接收协议的序列图表 794
图 10-2. 串行总线数据包格式 795
图 11-1. EXT_LF_CLK 寄存器 812
图 11-2. MODE_CONF_1 寄存器 813
图 11-3. SIZE_AND_DIS_FLAGS 寄存器 815
图 11-4. MODE_CONF 寄存器 817
图 11-5. VOLT_LOAD_0 寄存器 819
图 11-6. VOLT_LOAD_1 寄存器 820
图 11-7. RTC_OFFSET 寄存器 821
图 11-8. FREQ_OFFSET 寄存器 822
图 11-9. IEEE_MAC_0 寄存器 823
图 11-10. IEEE_MAC_1 寄存器 824
图 11-11. IEEE_BLE_0 寄存器 825
图 11-12. IEEE_BLE_1 寄存器 826
图 11-13. BL_CONFIG 寄存器 827
图 11-14. ERASE_CONF 寄存器 828
图 11-15. CCFG_TI_OPTIONS 寄存器 829
图 11-16. CCFG_TAP_DAP_0 寄存器 830
图 11-17. CCFG_TAP_DAP_1 寄存器 831
图 11-18. IMAGE_VALID_CONF 寄存器 832
图 11-19. CCFG_PROT_31_0 寄存器 833
图 11-20. CCFG_PROT_63_32 寄存器 835
图 11-21. CCFG_PROT_95_64 寄存器 837
图 11-22. CCFG_PROT_127_96 寄存器 839
图 11-23. MISC_CONF_1 寄存器 845
图 11-24. MISC_CONF_2 寄存器 846
图 11-25. HPOSC_MEAS_5 寄存器 847
图 11-26. HPOSC_MEAS_4 寄存器 848
图 11-27. HPOSC_MEAS_3 寄存器 849
图 11-28. HPOSC_MEAS_2 寄存器 850
图 11-29. HPOSC_MEAS_1 寄存器 851
图 11-30. CONFIG_CC26_FE 寄存器 852
图 11-31. CONFIG_CC13_FE 寄存器 853
图 11-32. CONFIG_RF_COMMON 寄存器 854

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

图 11-33. CONFIG_SYNTH_DIV2_CC26_2G4 寄存器 855
图 11-34. CONFIG_SYNTH_DIV2_CC13_2G4 寄存器 856
图 11-35. CONFIG_SYNTH_DIV2_CC26_1G 寄存器 857
图 11-36. CONFIG_SYNTH_DIV2_CC13_1G 寄存器 858
图 11-37. CONFIG_SYNTH_DIV4_CC26 寄存器 859
图 11-38. CONFIG_SYNTH_DIV4_CC13 寄存器 860
图 11-39. CONFIG_SYNTH_DIV5 寄存器 861
图 11-40. CONFIG_SYNTH_DIV6_CC26 寄存器 862
图 11-41. CONFIG_SYNTH_DIV6_CC13 寄存器 863
图 11-42. CONFIG_SYNTH_DIV10 寄存器 864
图 11-43. CONFIG_SYNTH_DIV12_CC26 寄存器 865
图 11-44. CONFIG_SYNTH_DIV12_CC13 寄存器 866
图 11-45. CONFIG_SYNTH_DIV15 寄存器 867
图 11-46. CONFIG_SYNTH_DIV30 寄存器 868
图 11-47. FLASH_NUMBER 寄存器 869
图 11-48. FLASH_COORDINATE 寄存器 870
图 11-49. FLASH_E_P 寄存器 871
图 11-50. FLASH_C_E_P_R 寄存器 872
图 11-51. FLASH_P_R_PV 寄存器 873
图 11-52. FLASH_EH_SEQ 寄存器 874
图 11-53. FLASH_VHV_E 寄存器 875
图 11-54. FLASH_PP 寄存器 876
图 11-55. FLASH_PROG_EP 寄存器 877
图 11-56. FLASH_ERA_PW 寄存器 878
图 11-57. FLASH_VHV 寄存器 879
图 11-58. FLASH_VHV_PV 寄存器 880
图 11-59. FLASH_V 寄存器 881
图 11-60. USER_ID 寄存器 882
图 11-61. FLASH_OTP_DATA3 寄存器 884
图 11-62. ANA2_TRIM 寄存器 885
图 11-63. LDO_TRIM 寄存器 886
图 11-64. MAC_BLE_0 寄存器 887
图 11-65. MAC_BLE_1 寄存器 888
图 11-66. MAC_15_4_0 寄存器 889
图 11-67. MAC_15_4_1 寄存器 890
图 11-68. FLASH_OTP_DATA4 寄存器 891
图 11-69. MISC_TRIM 寄存器 893
图 11-70. RCOSC_HF_TEMPCOMP 寄存器 894
图 11-71. ICEPICK_DEVICE_ID 寄存器 895
图 11-72. FCFG1_REVISION 寄存器 896
图 11-73. MISC_OTP_DATA 寄存器 897
图 11-74. IOCONF 寄存器 898
图 11-75. CONFIG_IF_ADC 寄存器 899
图 11-76. CONFIG_OSC_TOP 寄存器 900
图 11-77. SOC_ADC_ABS_GAIN 寄存器 901
图 11-78. SOC_ADC_REL_GAIN 寄存器 902
图 11-79. SOC_ADC_OFFSET_INT 寄存器 903
图 11-80. SOC_ADC_REF_TRIM_AND_OFFSET_EXT 寄存器 904
图 11-81. AMPCOMP_TH1 寄存器 905
图 11-82. AMPCOMP_TH2 寄存器 906
图 11-83. AMPCOMP_CTRL1 寄存器 907
图 11-84. ANABYPASS_VALUE2 寄存器 908
图 11-85. VOLT_TRIM 寄存器 909
图 11-86. OSC_CONF 寄存器 910
图 11-87. FREQ_OFFSET 寄存器 912
图 11-88. MISC_OTP_DATA_1 寄存器 913

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

图 11-89. SHDW_DIE_ID_0 寄存器.....914
图 11-90. SHDW_DIE_ID_1 寄存器.....915
图 11-91. SHDW_DIE_ID_2 寄存器.....916
图 11-92. SHDW_DIE_ID_3 寄存器.....917
图 11-93. SHDW_OSC_BIAS_LDO_TRIM 寄存器.....918
图 11-94. SHDW_ANA_TRIM 寄存器.....919
图 11-95. OSC_CONF1 寄存器.....920
图 11-96. DAC_BIAS_CNF 寄存器.....921
图 11-97. TFW_PROBE 寄存器.....922
图 11-98. TFW_FT 寄存器.....923
图 11-99. DAC_CAL0 寄存器.....924
图 11-100. DAC_CAL1 寄存器.....925
图 11-101. DAC_CAL2 寄存器.....926
图 11-102. DAC_CAL3 寄存器.....927
图 12-1. DMA 控制器及集成.....935
图 12-2. 对称加密处理步骤.....944
图 12-3. HMAC 步骤.....951
图 12-4. PKA 引擎.....963
图 12-5. 运行时序.....980
图 12-6. 交错操作序列.....980
图 12-7. 基本 PKCP 操作序列.....981
图 12-8. DMACH0CTL 寄存器.....989
图 12-9. DMACH0EXTADDR 寄存器.....990
图 12-10. DMACH0LEN 寄存器.....991
图 12-11. DMASTAT 寄存器.....992
图 12-12. DMASWRESET 寄存器.....993
图 12-13. DMACH1CTL 寄存器.....994
图 12-14. DMACH1EXTADDR 寄存器.....995
图 12-15. DMACH1LEN 寄存器.....996
图 12-16. DMABUSCFG 寄存器.....997
图 12-17. DMAPORTERR 寄存器.....998
图 12-18. DMAHWVER 寄存器.....999
图 12-19. KEYWRITEAREA 寄存器.....1000
图 12-20. KEYWRITTENAREA 寄存器.....1003
图 12-21. KEYSIZE 寄存器.....1005
图 12-22. KEYREADAREA 寄存器.....1006
图 12-23. AESKEY2_y 寄存器.....1007
图 12-24. AESKEY3_y 寄存器.....1008
图 12-25. AESIV_y 寄存器.....1009
图 12-26. AESCTL 寄存器.....1010
图 12-27. AESDATALEN0 寄存器.....1012
图 12-28. AESDATALEN1 寄存器.....1013
图 12-29. AESAUTHLEN 寄存器.....1014
图 12-30. AESDATAOUT0 寄存器.....1015
图 12-31. AESDATAIN0 寄存器.....1016
图 12-32. AESDATAOUT1 寄存器.....1017
图 12-33. AESDATAIN1 寄存器.....1018
图 12-34. AESDATAOUT2 寄存器.....1019
图 12-35. AESDATAIN2 寄存器.....1020
图 12-36. AESDATAOUT3 寄存器.....1021
图 12-37. AESDATAIN3 寄存器.....1022
图 12-38. AESTAGOUT_y 寄存器.....1023
图 12-39. HASHDATAIN1 寄存器.....1024
图 12-40. HASHDATAIN2 寄存器.....1025
图 12-41. HASHDATAIN3 寄存器.....1026
图 12-42. HASHDATAIN4 寄存器.....1027

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

图 12-43. HASHDATAIN5 寄存器.....1028
图 12-44. HASHDATAIN6 寄存器.....1029
图 12-45. HASHDATAIN7 寄存器.....1030
图 12-46. HASHDATAIN8 寄存器.....1031
图 12-47. HASHDATAIN9 寄存器.....1032
图 12-48. HASHDATAIN10 寄存器.....1033
图 12-49. HASHDATAIN11 寄存器.....1034
图 12-50. HASHDATAIN12 寄存器.....1035
图 12-51. HASHDATAIN13 寄存器.....1036
图 12-52. HASHDATAIN14 寄存器.....1037
图 12-53. HASHDATAIN15 寄存器.....1038
图 12-54. HASHDATAIN16 寄存器.....1039
图 12-55. HASHDATAIN17 寄存器.....1040
图 12-56. HASHDATAIN18 寄存器.....1041
图 12-57. HASHDATAIN19 寄存器.....1042
图 12-58. HASHDATAIN20 寄存器.....1043
图 12-59. HASHDATAIN21 寄存器.....1044
图 12-60. HASHDATAIN22 寄存器.....1045
图 12-61. HASHDATAIN23 寄存器.....1046
图 12-62. HASHDATAIN24 寄存器.....1047
图 12-63. HASHDATAIN25 寄存器.....1048
图 12-64. HASHDATAIN26 寄存器.....1049
图 12-65. HASHDATAIN27 寄存器.....1050
图 12-66. HASHDATAIN28 寄存器.....1051
图 12-67. HASHDATAIN29 寄存器.....1052
图 12-68. HASHDATAIN30 寄存器.....1053
图 12-69. HASHDATAIN31 寄存器.....1054
图 12-70. HASHIOBUFCTRL 寄存器.....1055
图 12-71. HASHMODE 寄存器.....1057
图 12-72. HASHINLENL 寄存器.....1058
图 12-73. HASHINLENH 寄存器.....1059
图 12-74. HASHDIGESTA 寄存器.....1060
图 12-75. HASHDIGESTB 寄存器.....1061
图 12-76. HASHDIGESTC 寄存器.....1062
图 12-77. HASHDIGESTD 寄存器.....1063
图 12-78. HASHDIGESTE 寄存器.....1064
图 12-79. HASHDIGESTF 寄存器.....1065
图 12-80. HASHDIGESTG 寄存器.....1066
图 12-81. HASHDIGESTH 寄存器.....1067
图 12-82. HASHDIGESTI 寄存器.....1068
图 12-83. HASHDIGESTJ 寄存器.....1069
图 12-84. HASHDIGESTK 寄存器.....1070
图 12-85. HASHDIGESTL 寄存器.....1071
图 12-86. HASHDIGESTM 寄存器.....1072
图 12-87. HASHDIGESTN 寄存器.....1073
图 12-88. HASHDIGESTO 寄存器.....1074
图 12-89. HASHDIGESTP 寄存器.....1075
图 12-90. DMAPROTCTL 寄存器.....1077
图 12-91. SWRESET 寄存器.....1078
图 12-92. IRQTYPE 寄存器.....1079
图 12-93. IRQEN 寄存器.....1080
图 12-94. IRQCLR 寄存器.....1081
图 12-95. IRQSET 寄存器.....1082
图 12-96. IRQSTAT 寄存器.....1083
图 12-97. HWVER 寄存器.....1084
图 13-1. IOC 概述（简化版）.....1086

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

图 13-2. 通用 I/O 引脚 (简化版) 1093
图 13-3. IOSTRMIN 寄存器..... 1096
图 13-4. IOSTRMED 寄存器..... 1097
图 13-5. IOSTRMAX 寄存器..... 1098
图 13-6. CLK32KCTL 寄存器..... 1099
图 13-7. TCKCTL 寄存器..... 1100
图 13-8. DOUT3_0 寄存器..... 1102
图 13-9. DOUT7_4 寄存器..... 1103
图 13-10. DOUT11_8 寄存器..... 1104
图 13-11. DOUT15_12 寄存器..... 1105
图 13-12. DOUT19_16 寄存器..... 1106
图 13-13. DOUT23_20 寄存器..... 1107
图 13-14. DOUT27_24 寄存器..... 1108
图 13-15. DOUT31_28 寄存器..... 1109
图 13-16. DOUT31_0 寄存器..... 1110
图 13-17. DOUTSET31_0 寄存器..... 1112
图 13-18. DOUTCLR31_0 寄存器..... 1114
图 13-19. DOUTTGL31_0 寄存器..... 1116
图 13-20. DIN31_0 寄存器..... 1118
图 13-21. DOE31_0 寄存器..... 1120
图 13-22. EVFLAGS31_0 寄存器..... 1122
图 13-23. IOCFG0 寄存器..... 1126
图 13-24. IOCFG1 寄存器..... 1131
图 13-25. IOCFG2 寄存器..... 1136
图 13-26. IOCFG3 寄存器..... 1141
图 13-27. IOCFG4 寄存器..... 1146
图 13-28. IOCFG5 寄存器..... 1151
图 13-29. IOCFG6 寄存器..... 1156
图 13-30. IOCFG7 寄存器..... 1161
图 13-31. IOCFG8 寄存器..... 1166
图 13-32. IOCFG9 寄存器..... 1171
图 13-33. IOCFG10 寄存器..... 1176
图 13-34. IOCFG11 寄存器..... 1181
图 13-35. IOCFG12 寄存器..... 1186
图 13-36. IOCFG13 寄存器..... 1191
图 13-37. IOCFG14 寄存器..... 1196
图 13-38. IOCFG15 寄存器..... 1201
图 13-39. IOCFG16 寄存器..... 1206
图 13-40. IOCFG17 寄存器..... 1211
图 13-41. IOCFG18 寄存器..... 1216
图 13-42. IOCFG19 寄存器..... 1221
图 13-43. IOCFG20 寄存器..... 1226
图 13-44. IOCFG21 寄存器..... 1231
图 13-45. IOCFG22 寄存器..... 1236
图 13-46. IOCFG23 寄存器..... 1241
图 13-47. IOCFG24 寄存器..... 1246
图 13-48. IOCFG25 寄存器..... 1251
图 13-49. IOCFG26 寄存器..... 1256
图 13-50. IOCFG27 寄存器..... 1261
图 13-51. IOCFG28 寄存器..... 1266
图 13-52. IOCFG29 寄存器..... 1271
图 13-53. IOCFG30 寄存器..... 1276
图 13-54. IOCFG31 寄存器..... 1281
图 14-1. µDMA 方框图..... 1288
图 14-2. 乒乓 µDMA 事务示例..... 1294
图 14-3. 内存散聚、设置和配置..... 1296

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

图 14-4. 内存散聚、μDMA 复制顺序 1298
图 14-5. 外设散集、设置和配置 1299
图 14-6. 外设散聚、μDMA 复制顺序 1300
图 14-7. STATUS 寄存器 1307
图 14-8. CFG 寄存器 1309
图 14-9. CTRL 寄存器 1310
图 14-10. ALTCTRL 寄存器 1311
图 14-11. WAITONREQ 寄存器 1312
图 14-12. SOFTREQ 寄存器 1313
图 14-13. SETBURST 寄存器 1314
图 14-14. CLEARBURST 寄存器 1315
图 14-15. SETREQMASK 寄存器 1316
图 14-16. CLEARREQMASK 寄存器 1317
图 14-17. SETCHANNELEN 寄存器 1318
图 14-18. CLEARCHANNELEN 寄存器 1319
图 14-19. SETCHNLPRIALT 寄存器 1320
图 14-20. CLEARCHNLPRIALT 寄存器 1321
图 14-21. SETCHNLPRIORITY 寄存器 1322
图 14-22. CLEARCHNLPRIORITY 寄存器 1323
图 14-23. ERROR 寄存器 1324
图 14-24. REQDONE 寄存器 1325
图 14-25. DONEMASK 寄存器 1326
图 15-1. GPTM 模块方框图 1329
图 15-2. 输入边沿计数模式示例（向下计数） 1333
图 15-3. 输入边沿时间模式示例 1334
图 15-4. 16 位 PWM 模式示例 1336
图 15-5. CCP 输出，GPT:TnMATCHR > GPT:TnILR 1336
图 15-6. CCP 输出，GPT:TnMATCHR = GPT:TnILR 1337
图 15-7. CCP 输出，GPT:TnILR > GPT:TnMATCHR 1337
图 15-8. 计时器菊花链 1338
图 15-9. CFG 寄存器 1345
图 15-10. TAMR 寄存器 1346
图 15-11. TBMR 寄存器 1348
图 15-12. CTL 寄存器 1350
图 15-13. SYNC 寄存器 1352
图 15-14. IMR 寄存器 1353
图 15-15. RIS 寄存器 1355
图 15-16. MIS 寄存器 1357
图 15-17. ICLR 寄存器 1358
图 15-18. TAILR 寄存器 1359
图 15-19. TBILR 寄存器 1360
图 15-20. TAMATCHR 寄存器 1361
图 15-21. TBMATCHR 寄存器 1362
图 15-22. TAPR 寄存器 1363
图 15-23. TBPR 寄存器 1364
图 15-24. TAPMR 寄存器 1365
图 15-25. TBPMR 寄存器 1366
图 15-26. TAR 寄存器 1367
图 15-27. TBR 寄存器 1368
图 15-28. TAV 寄存器 1369
图 15-29. TBV 寄存器 1370
图 15-30. TAPS 寄存器 1371
图 15-31. TBPS 寄存器 1372
图 15-32. TAPV 寄存器 1373
图 15-33. TBPV 寄存器 1374
图 15-34. DMAEV 寄存器 1375

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

图 15-35. 版本寄存器 1376
图 15-36. ANDCCP 寄存器 1377
图 16-1. AON_RTC 通道 1380
图 16-2. CTL 寄存器 1384
图 16-3. EVFLAGS 寄存器 1386
图 16-4. SEC 寄存器 1387
图 16-5. SUBSEC 寄存器 1388
图 16-6. SUBSECINC 寄存器 1389
图 16-7. CHCTL 寄存器 1390
图 16-8. CH0CMP 寄存器 1391
图 16-9. CH1CMP 寄存器 1392
图 16-10. CH2CMP 寄存器 1393
图 16-11. CH2CMPINC 寄存器 1394
图 16-12. CH1CAPT 寄存器 1395
图 16-13. SYNC 寄存器 1396
图 16-14. TIME 寄存器 1397
图 16-15. SYNCLF 寄存器 1398
图 17-1. WDT 方框图 1401
图 17-2. LOAD 寄存器 1403
图 17-3. VALUE 寄存器 1404
图 17-4. CTL 寄存器 1405
图 17-5. ICR 寄存器 1406
图 17-6. RIS 寄存器 1407
图 17-7. MIS 寄存器 1408
图 17-8. TEST 寄存器 1409
图 17-9. INT_CAUS 寄存器 1410
图 17-10. LOCK 寄存器 1411
图 18-1. 随机数发生器方框图 1413
图 18-2. TRNG 轮询模式 1418
图 18-3. 中断服务例程 1419
图 18-4. OUT0 寄存器 1422
图 18-5. OUT1 寄存器 1423
图 18-6. IRQFLAGSTAT 寄存器 1424
图 18-7. IRQFLAGMASK 寄存器 1425
图 18-8. IRQFLAGCLR 寄存器 1426
图 18-9. CTL 寄存器 1427
图 18-10. CFG0 寄存器 1428
图 18-11. ALARMCNT 寄存器 1429
图 18-12. FROEN 寄存器 1430
图 18-13. FRODETUNE 寄存器 1431
图 18-14. ALARMMASK 寄存器 1432
图 18-15. ALARMSTOP 寄存器 1433
图 18-16. LFSR0 寄存器 1434
图 18-17. LFSR1 寄存器 1435
图 18-18. LFSR2 寄存器 1436
图 18-19. HWOPT 寄存器 1437
图 18-20. HWVER0 寄存器 1438
图 18-21. IRQSTATMASK 寄存器 1439
图 18-22. HWVER1 寄存器 1440
图 18-23. IRQSET 寄存器 1441
图 18-24. SWRESET 寄存器 1442
图 18-25. IRQSTAT 寄存器 1443
图 19-1. AUX 域方框图 1446
图 19-2. AUX 状态图 1447
图 19-3. ADC 窗口监控器 — 执行代码 1451
图 19-4. ADC 窗口监控器 1452

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

图 19-5. 任务测试 1453
图 19-6. 帮助查看器 1454
图 19-7. MAC 方框图 1461
图 19-8. MAC 时序图 1462
图 19-9. SCE 唤醒和事件接口 1463
图 19-10. 避免 VDDR 充电电源噪声 1466
图 19-11. AUX I/O 方框图 1469
图 19-12. AUX_SPIM 方框图 1471
图 19-13. SPI 计时示意图：PHA = 0 1473
图 19-14. SPI 计时示意图：PHA = 1 1473
图 19-15. AUX_TDC 方框图 1475
图 19-16. 相位宽度时序要求 1479
图 19-17. 频率测量波形 1479
图 19-18. 任意时间测量 1 1480
图 19-19. 任意时间测量 2 1481
图 19-20. 任意时间测量 3 1482
图 19-21. 任意时间测量 4 1483
图 19-22. 脉冲计数 1484
图 19-23. AUX_TIMER01 方框图 1485
图 19-24. AUX_TIMER2 方框图 1486
图 19-25. 周期脉冲宽度测量 1490
图 19-26. 中心对齐 PWM 1491
图 19-27. 边沿对齐的 PWM 1492
图 19-28. AUX 模拟方框图 1494
图 19-29. ADC 方框图 1495
图 19-30. COMPA 方框图 1499
图 19-31. COMPB 方框图 1501
图 19-32. 参考设计方框图 1503
图 19-33. DAC 采样时钟相位 1504
图 19-34. 采样时钟周期配置 1504
图 19-35. ISRC 方框图 1508
图 19-36. MUX0 寄存器 1522
图 19-37. MUX1 寄存器 1523
图 19-38. MUX2 寄存器 1524
图 19-39. MUX3 寄存器 1525
图 19-40. ISRC 寄存器 1526
图 19-41. COMP 寄存器 1527
图 19-42. MUX4 寄存器 1528
图 19-43. ADC0 寄存器 1529
图 19-44. ADC1 寄存器 1530
图 19-45. ADCREF0 寄存器 1531
图 19-46. ADCREF1 寄存器 1532
图 19-47. LPMBIAS 寄存器 1533
图 19-48. IOMODE 寄存器 1536
图 19-49. GPIODIE 寄存器 1541
图 19-50. IOPOE 寄存器 1542
图 19-51. GPIODOUT 寄存器 1543
图 19-52. GPIODIN 寄存器 1544
图 19-53. GPIODOUTSET 寄存器 1545
图 19-54. GPIODOUTCLR 寄存器 1546
图 19-55. GPIODOUTTGL 寄存器 1547
图 19-56. IO0PSEL 寄存器 1548
图 19-57. IO1PSEL 寄存器 1549
图 19-58. IO2PSEL 寄存器 1550
图 19-59. IO3PSEL 寄存器 1551
图 19-60. IO4PSEL 寄存器 1552

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

图 19-61. IO5PSEL 寄存器 1553
图 19-62. IO6PSEL 寄存器 1554
图 19-63. IO7PSEL 寄存器 1555
图 19-64. IOMODEL 寄存器 1556
图 19-65. IOMODEH 寄存器 1557
图 19-66. EVSTAT0 寄存器 1560
图 19-67. EVSTAT1 寄存器 1562
图 19-68. EVSTAT2 寄存器 1564
图 19-69. EVSTAT3 寄存器 1566
图 19-70. SCEWEVCFG0 寄存器 1568
图 19-71. SCEWEVCFG1 寄存器 1571
图 19-72. DMACTL 寄存器 1574
图 19-73. SWEVSET 寄存器 1575
图 19-74. EVTOAONFLAGS 寄存器 1576
图 19-75. EVTOAONPOL 寄存器 1577
图 19-76. EVTOAONFLAGSCLR 寄存器 1578
图 19-77. EVTOMCUFLAGS 寄存器 1579
图 19-78. EVTOMCUPOL 寄存器 1581
图 19-79. EVTOMCUFLAGSCLR 寄存器 1583
图 19-80. COMBEVTOMCUMASK 寄存器 1585
图 19-81. EVOBSCFG 寄存器 1587
图 19-82. PROGDLY 寄存器 1590
图 19-83. MANUAL 寄存器 1591
图 19-84. EVSTAT0L 寄存器 1592
图 19-85. EVSTAT0H 寄存器 1593
图 19-86. EVSTAT1L 寄存器 1594
图 19-87. EVSTAT1H 寄存器 1595
图 19-88. EVSTAT2L 寄存器 1596
图 19-89. EVSTAT2H 寄存器 1597
图 19-90. EVSTAT3L 寄存器 1598
图 19-91. EVSTAT3H 寄存器 1599
图 19-92. SMPH0 寄存器 1601
图 19-93. SMPH1 寄存器 1602
图 19-94. SMPH2 寄存器 1603
图 19-95. SMPH3 寄存器 1604
图 19-96. SMPH4 寄存器 1605
图 19-97. SMPH5 寄存器 1606
图 19-98. SMPH6 寄存器 1607
图 19-99. SMPH7 寄存器 1608
图 19-100. AUTOTAKE 寄存器 1609
图 19-101. CTL 寄存器 1611
图 19-102. STAT 寄存器 1612
图 19-103. RESULT 寄存器 1614
图 19-104. SATCFG 寄存器 1615
图 19-105. TRIGSRC 寄存器 1616
图 19-106. TRIGCNT 寄存器 1621
图 19-107. TRIGCNTLOAD 寄存器 1622
图 19-108. TRIGCNTCFG 寄存器 1623
图 19-109. PRECTL 寄存器 1624
图 19-110. PRECNTR 寄存器 1627
图 19-111. T0CFG 寄存器 1629
图 19-112. T0CTL 寄存器 1632
图 19-113. T0TARGET 寄存器 1633
图 19-114. T0CNTR 寄存器 1634
图 19-115. T1CFG 寄存器 1635
图 19-116. T1CTL 寄存器 1638

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

图 19-117. T1TARGET 寄存器 1639
图 19-118. T1CNTR 寄存器 1640
图 19-119. CTL 寄存器 1643
图 19-120. TARGET 寄存器 1645
图 19-121. SHDWTARGET 寄存器 1646
图 19-122. CNTR 寄存器 1647
图 19-123. PRECFG 寄存器 1648
图 19-124. EVCTL 寄存器 1649
图 19-125. PULSETRIG 寄存器 1650
图 19-126. CH0EVCFG 寄存器 1651
图 19-127. CH0CCFG 寄存器 1654
图 19-128. CH0PCC 寄存器 1657
图 19-129. CH0CC 寄存器 1658
图 19-130. CH1EVCFG 寄存器 1659
图 19-131. CH1CCFG 寄存器 1662
图 19-132. CH1PCC 寄存器 1665
图 19-133. CH1CC 寄存器 1666
图 19-134. CH2EVCFG 寄存器 1667
图 19-135. CH2CCFG 寄存器 1670
图 19-136. CH2PCC 寄存器 1673
图 19-137. CH2CC 寄存器 1674
图 19-138. CH3EVCFG 寄存器 1675
图 19-139. CH3CCFG 寄存器 1678
图 19-140. CH3PCC 寄存器 1681
图 19-141. CH3CC 寄存器 1682
图 19-142. ADCCTL 寄存器 1684
图 19-143. ADCFIFOSTAT 寄存器 1687
图 19-144. ADCFIFO 寄存器 1688
图 19-145. ADCTRIG 寄存器 1689
图 19-146. ISRCCTL 寄存器 1690
图 19-147. DACCTL 寄存器 1691
图 19-148. LPMBIASCTL 寄存器 1693
图 19-149. DACSMPLCTL 寄存器 1694
图 19-150. DACSMPLCFG0 寄存器 1695
图 19-151. DACSMPLCFG1 寄存器 1696
图 19-152. DACVALUE 寄存器 1697
图 19-153. DACSTAT 寄存器 1698
图 19-154. OPMODEREQ 寄存器 1701
图 19-155. OPMODEACK 寄存器 1702
图 19-156. PROGWU0CFG 寄存器 1703
图 19-157. PROGWU1CFG 寄存器 1706
图 19-158. PROGWU2CFG 寄存器 1709
图 19-159. PROGWU3CFG 寄存器 1712
图 19-160. SWWUTRIG 寄存器 1715
图 19-161. WUFLAGS 寄存器 1716
图 19-162. WUFLAGSCLR 寄存器 1717
图 19-163. WUGATE 寄存器 1719
图 19-164. VECCFG0 寄存器 1720
图 19-165. VECCFG1 寄存器 1721
图 19-166. VECCFG2 寄存器 1722
图 19-167. VECCFG3 寄存器 1723
图 19-168. VECCFG4 寄存器 1724
图 19-169. VECCFG5 寄存器 1725
图 19-170. VECCFG6 寄存器 1726
图 19-171. VECCFG7 寄存器 1727
图 19-172. EVSYNCRATE 寄存器 1728

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

图 19-173. PEROPRATE 寄存器 1729
图 19-174. ADCCLKCTL 寄存器 1730
图 19-175. TDCCLKCTL 寄存器 1731
图 19-176. TDCREFCLKCTL 寄存器 1732
图 19-177. TIMER2CLKCTL 寄存器 1733
图 19-178. TIMER2CLKSTAT 寄存器 1734
图 19-179. TIMER2CLKSWITCH 寄存器 1735
图 19-180. TIMER2DBGCTL 寄存器 1736
图 19-181. CLKSHIFTDET 寄存器 1737
图 19-182. RECHARGETRIG 寄存器 1738
图 19-183. RECHARGEDET 寄存器 1739
图 19-184. RTCSUBSECINC0 寄存器 1740
图 19-185. RTCSUBSECINC1 寄存器 1741
图 19-186. RTCSUBSECINCCTL 寄存器 1742
图 19-187. RTCSEC 寄存器 1743
图 19-188. RTCSUBSEC 寄存器 1744
图 19-189. RTCEVCLR 寄存器 1745
图 19-190. BATMONBAT 寄存器 1746
图 19-191. BATMONTEMP 寄存器 1747
图 19-192. TIMERHALT 寄存器 1748
图 19-193. TIMER2BRIDGE 寄存器 1749
图 19-194. SWPWRPROF 寄存器 1750
图 19-195. SPIMCFG 寄存器 1752
图 19-196. MISOCFG 寄存器 1753
图 19-197. MOSICTL 寄存器 1754
图 19-198. TX8 寄存器 1755
图 19-199. TX16 寄存器 1756
图 19-200. RX8 寄存器 1757
图 19-201. RX16 寄存器 1758
图 19-202. SCLKIDLE 寄存器 1759
图 19-203. DATAIDLE 寄存器 1760
图 19-204. OP0S 寄存器 1763
图 19-205. OP0U 寄存器 1764
图 19-206. OP1SMUL 寄存器 1765
图 19-207. OP1UMUL 寄存器 1766
图 19-208. OP1SMAC 寄存器 1767
图 19-209. OP1UMAC 寄存器 1768
图 19-210. OP1SADD16 寄存器 1769
图 19-211. OP1UADD16 寄存器 1770
图 19-212. OP1SADD32 寄存器 1771
图 19-213. OP1UADD32 寄存器 1772
图 19-214. CLZ 寄存器 1773
图 19-215. CLS 寄存器 1774
图 19-216. ACCSHIFT 寄存器 1775
图 19-217. ACCRESET 寄存器 1776
图 19-218. ACC15_0 寄存器 1777
图 19-219. ACC16_1 寄存器 1778
图 19-220. ACC17_2 寄存器 1779
图 19-221. ACC18_3 寄存器 1780
图 19-222. ACC19_4 寄存器 1781
图 19-223. ACC20_5 寄存器 1782
图 19-224. ACC21_6 寄存器 1783
图 19-225. ACC22_7 寄存器 1784
图 19-226. ACC23_8 寄存器 1785
图 19-227. ACC24_9 寄存器 1786
图 19-228. ACC25_10 寄存器 1787

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

图 19-229. ACC26_11 寄存器.....1788
图 19-230. ACC27_12 寄存器.....1789
图 19-231. ACC28_13 寄存器.....1790
图 19-232. ACC29_14 寄存器.....1791
图 19-233. ACC30_15 寄存器.....1792
图 19-234. ACC31_16 寄存器.....1793
图 19-235. ACC32_17 寄存器.....1794
图 19-236. ACC33_18 寄存器.....1795
图 19-237. ACC34_19 寄存器.....1796
图 19-238. ACC35_20 寄存器.....1797
图 19-239. ACC36_21 寄存器.....1798
图 19-240. ACC37_22 寄存器.....1799
图 19-241. ACC38_23 寄存器.....1800
图 19-242. ACC39_24 寄存器.....1801
图 19-243. ACC39_32 寄存器.....1802
图 19-244. CTL 寄存器.....1804
图 19-245. FETCHSTAT 寄存器.....1805
图 19-246. CPUSTAT 寄存器.....1806
图 19-247. WUSTAT 寄存器.....1807
图 19-248. REG1_0 寄存器.....1808
图 19-249. REG3_2 寄存器.....1809
图 19-250. REG5_4 寄存器.....1810
图 19-251. REG7_6 寄存器.....1811
图 19-252. LOOPADDR 寄存器.....1812
图 19-253. LOOPCNT 寄存器.....1813
图 20-1. CTL 寄存器.....1818
图 20-2. MEASCFG 寄存器.....1819
图 20-3. TEMPP0 寄存器.....1820
图 20-4. TEMPP1 寄存器.....1821
图 20-5. TEMPP2 寄存器.....1822
图 20-6. BATMONP0 寄存器.....1823
图 20-7. BATMONP1 寄存器.....1824
图 20-8. IOSTRP0 寄存器.....1825
图 20-9. FLASHPUMPP0 寄存器.....1826
图 20-10. BAT 寄存器.....1827
图 20-11. BATUPD 寄存器.....1828
图 20-12. TEMP 寄存器.....1829
图 20-13. TEMPUPD 寄存器.....1830
图 20-14. EVENTMASK 寄存器.....1831
图 20-15. EVENT 寄存器.....1832
图 20-16. BATTUL 寄存器.....1833
图 20-17. BATTLL 寄存器.....1834
图 20-18. TEMPUL 寄存器.....1835
图 20-19. TEMPLL 寄存器.....1836
图 21-1. UART 模块方框图.....1839
图 21-2. UART 字符帧.....1840
图 21-3. µDMA 示例.....1844
图 21-4. DR 寄存器.....1848
图 21-5. RSR 寄存器.....1850
图 21-6. ECR 寄存器.....1851
图 21-7. FR 寄存器.....1852
图 21-8. IBRD 寄存器.....1853
图 21-9. FBRD 寄存器.....1854
图 21-10. LCRH 寄存器.....1855
图 21-11. CTL 寄存器.....1857
图 21-12. IFLS 寄存器.....1859

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

图 21-13. IMSC 寄存器 1860
图 21-14. RIS 寄存器 1862
图 21-15. MIS 寄存器 1864
图 21-16. ICR 寄存器 1866
图 21-17. DMACTL 寄存器 1867
图 22-1. SSI 模块方框图 1870
图 22-2. TI 同步串行帧格式（单个传送） 1874
图 22-3. TI 同步串行帧格式（连续传送） 1874
图 22-4. SPO = 0 和 SPH = 0 时的 Motorola SPI 格式（单次发送） 1875
图 22-5. SPO = 0 和 SPH = 0 时的 Motorola SPI 格式（连续发送） 1875
图 22-6. SPO = 0 和 SPH = 1 时的 Motorola SPI 帧格式 1876
图 22-7. SPO = 1 和 SPH = 0 时的 Motorola SPI 帧格式（单次发送） 1877
图 22-8. SPO = 1 和 SPH = 0 时的 Motorola SPI 帧格式（连续发送） 1877
图 22-9. SPO = 1 和 SPH = 1 时的 Motorola SPI 帧格式 1878
图 22-10. MICROWIRE 帧格式（单帧） 1878
图 22-11. MICROWIRE 帧格式（连续传输） 1879
图 22-12. MICROWIRE 帧格式、SSIFss 输入设置及保持要求 1879
图 22-13. CR0 寄存器 1883
图 22-14. CR1 寄存器 1884
图 22-15. DR 寄存器 1885
图 22-16. SR 寄存器 1886
图 22-17. CPSR 寄存器 1887
图 22-18. IMSC 寄存器 1888
图 22-19. RIS 寄存器 1889
图 22-20. MIS 寄存器 1890
图 22-21. ICR 寄存器 1891
图 22-22. DMACR 寄存器 1892
图 23-1. I^2C 模块方框图 1894
图 23-2. I^2C 总线配置 1895
图 23-3. 启动和停止条件 1895
图 23-4. 使用 7 位地址完成数据传输 1896
图 23-5. 第一个字节中的 R/S 位 1896
图 23-6. 在 I^2C 总线上传输位期间的数据有效性 1896
图 23-7. 主器件单次传输 1899
图 23-8. 主器件单次接收 1900
图 23-9. 带重复启动条件的主器件传输 1901
图 23-10. 带重复启动条件的主器件接收 1902
图 23-11. 主器件在带重复启动条件的传输之后进行带重复启动条件的接收 1903
图 23-12. 主器件在带重复启动条件的接收之后进行带重复启动条件的传输 1904
图 23-13. 从器件命令序列 1905
图 23-14. SOAR 寄存器 1909
图 23-15. SSTAT 寄存器 1910
图 23-16. SCTL 寄存器 1911
图 23-17. SDR 寄存器 1912
图 23-18. SIMR 寄存器 1913
图 23-19. SRIS 寄存器 1914
图 23-20. SMIS 寄存器 1915
图 23-21. SICR 寄存器 1916
图 23-22. MSA 寄存器 1917
图 23-23. MSTAT 寄存器 1918
图 23-24. MCTRL 寄存器 1919
图 23-25. MDR 寄存器 1921
图 23-26. MTPR 寄存器 1922
图 23-27. MIMR 寄存器 1923
图 23-28. MRIS 寄存器 1924
图 23-29. MMIS 寄存器 1925

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

图 23-30. MICR 寄存器 1926
图 23-31. MCR 寄存器 1927
图 24-1. I²S 模块简化方框图 1930
图 24-2. I²S 串行格式 1932
图 24-3. LJF 串行格式 1933
图 24-4. RJF 串行格式 1934
图 24-5. DSP 串行格式（零数据延迟） 1935
图 24-6. 一个 ADx 引脚上的 16 位单声道 I²S、LJF 和 RJF 格式，存储器中显示六个帧 1938
图 24-7. 一个 ADx 引脚上的 16 位立体声 I²S、LJF 和 RJF 格式，存储器中显示三个帧 1938
图 24-8. 一个 ADx 引脚上的 24 位立体声 I²S、LJF 和 RJF 格式，存储器中显示两个帧 1938
图 24-9. AD0 和 AD1 引脚上的 16 位 I²S 格式，存储器中显示两个帧 1939
图 24-10. AD0 和 AD1 引脚上的 16 位 DSP 格式，存储器中显示两个帧 1939
图 24-11. 采样时间戳发生器 1941
图 24-12. AIFWCLKSRC 寄存器 1946
图 24-13. AIFDMACFG 寄存器 1947
图 24-14. AIFDIRCFG 寄存器 1948
图 24-15. AIFFMTCFG 寄存器 1949
图 24-16. AIFWMASK0 寄存器 1950
图 24-17. AIFWMASK1 寄存器 1951
图 24-18. AIFPWMVALUE 寄存器 1952
图 24-19. AIFINPTRNEXT 寄存器 1953
图 24-20. AIFINPTR 寄存器 1954
图 24-21. AIFOUTPTRNEXT 寄存器 1955
图 24-22. AIFOUTPTR 寄存器 1956
图 24-23. STMPCTL 寄存器 1957
图 24-24. STMPXCNTCAPT0 寄存器 1958
图 24-25. STMPXPER 寄存器 1959
图 24-26. STMPWCNTCAPT0 寄存器 1960
图 24-27. STMPWPER 寄存器 1961
图 24-28. STMPINTRIG 寄存器 1962
图 24-29. STMPOUTTRIG 寄存器 1963
图 24-30. STMPWSET 寄存器 1964
图 24-31. STMPWADD 寄存器 1965
图 24-32. STMPXPERMIN 寄存器 1966
图 24-33. STMPWCNT 寄存器 1967
图 24-34. STMPXCNT 寄存器 1968
图 24-35. STMPXCNTCAPT1 寄存器 1969
图 24-36. STMPWCNTCAPT1 寄存器 1970
图 24-37. IRQMASK 寄存器 1971
图 24-38. IRQFLAGS 寄存器 1972
图 24-39. IRQSET 寄存器 1973
图 24-40. IRQCLR 寄存器 1974
图 25-1. 具有外部依赖性的有限 RF 内核概述 1977
图 25-2. HAL 的硬件支持 1978
图 25-3. 用于无线操作命令和即时命令的 CMDR 寄存器 1982
图 25-4. 用于直接命令的 CMDR 寄存器 1982
图 25-5. CMDSTA 寄存器的格式 1983
图 25-6. RX 队列条目元素（装订字段可选） 2034
图 25-7. CSMA-CA 操作 2043
图 25-8. 接收缓冲器条目元素 2068
图 25-9. 标准数据包格式 2106
图 25-10. 高级数据包格式 2106
图 25-11. 接收缓冲区条目元素 2115
图 25-12. RATCNT 寄存器 2128
图 25-13. RATCH0VAL 寄存器 2129
图 25-14. RATCH1VAL 寄存器 2130

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

图 25-15. RATCH2VAL 寄存器.....2131
图 25-16. RATCH3VAL 寄存器.....2132
图 25-17. RATCH4VAL 寄存器.....2133
图 25-18. RATCH5VAL 寄存器.....2134
图 25-19. RATCH6VAL 寄存器.....2135
图 25-20. RATCH7VAL 寄存器.....2136
图 25-21. CMDR 寄存器.....2138
图 25-22. CMDSTA 寄存器.....2139
图 25-23. RFHWIFG 寄存器.....2140
图 25-24. RFHWIEN 寄存器.....2142
图 25-25. RFCPEIFG 寄存器.....2143
图 25-26. RFCPEIEN 寄存器.....2145
图 25-27. RFCPEISL 寄存器.....2147
图 25-28. RFACKIFG 寄存器.....2150
图 25-29. SYSGPOCTL 寄存器.....2151
图 25-30. PWMCLKEN 寄存器.....2154

表格清单

表 2-1. 处理器模式、特权级别和栈使用汇总.....87
表 2-2. 处理器寄存器映射.....89
表 2-3. Cortex® 通用寄存器 0 (R0).....89
表 2-4. Cortex® 通用寄存器 1 (R1).....90
表 2-5. Cortex® 通用寄存器 2 (R2).....90
表 2-6. Cortex® 通用寄存器 3 (R3).....90
表 2-7. Cortex® 通用寄存器 4 (R4).....91
表 2-8. Cortex® 通用寄存器 5 (R5).....91
表 2-9. Cortex® 通用寄存器 6 (R6).....91
表 2-10. Cortex® 通用寄存器 7 (R7).....92
表 2-11. Cortex® 通用寄存器 8 (R8).....92
表 2-12. Cortex® 通用寄存器 9 (R9).....92
表 2-13. Cortex® 通用寄存器 10 (R10).....93
表 2-14. Cortex® 通用寄存器 11 (R11).....93
表 2-15. Cortex® 通用寄存器 12 (R12).....93
表 2-16. 栈指针 (SP).....94
表 2-17. 链接寄存器 (LR).....94
表 2-18. 程序计数器 (PC).....95
表 2-19. PSR 组合.....95
表 2-20. 程序状态寄存器 (PSR) 或 (xPSR).....95
表 2-21. 优先级屏蔽寄存器 (PRIMASK).....98
表 2-22. 故障屏蔽寄存器 (FAULTMASK).....99
表 2-23. 基址优先级屏蔽寄存器 (BASEPRI).....100
表 2-24. 控制寄存器 (CONTROL).....101
表 2-25. Arm® Cortex®-M4F 指令集摘要.....102
表 2-26. Arm® Cortex®-M4F DSP 指令集摘要.....106
表 2-27. FPU 指令集.....112
表 2-28. 默认 NaN 值.....114
表 2-29. QNaN 和 SNaN 处理.....114
表 2-30. Arm® Cortex®-M4F 浮点系统寄存器.....116
表 2-31. MPU 寄存器.....117
表 2-32. Arm® Cortex®-M4F 处理器寄存器.....118
表 2-33. CPU_DWT 寄存器.....119
表 2-34. CPU_DWT 访问类型代码.....119
表 2-35. CTRL 寄存器字段说明.....121
表 2-36. CYCCNT 寄存器字段说明.....123
表 2-37. CPICNT 寄存器字段说明.....124
表 2-38. EXCCNT 寄存器字段说明.....125

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 2-39. SLEEPCNT 寄存器字段说明 126
表 2-40. LSUCNT 寄存器字段说明 127
表 2-41. FOLDCNT 寄存器字段说明 128
表 2-42. PCSR 寄存器字段说明 129
表 2-43. COMP0 寄存器字段说明 130
表 2-44. MASK0 寄存器字段说明 131
表 2-45. FUNCTION0 寄存器字段说明 132
表 2-46. COMP1 寄存器字段说明 134
表 2-47. MASK1 寄存器字段说明 135
表 2-48. FUNCTION1 寄存器字段说明 136
表 2-49. COMP2 寄存器字段说明 138
表 2-50. MASK2 寄存器字段说明 139
表 2-51. FUNCTION2 寄存器字段说明 140
表 2-52. COMP3 寄存器字段说明 142
表 2-53. MASK3 寄存器字段说明 143
表 2-54. FUNCTION3 寄存器字段说明 144
表 2-55. CPU_FPB 寄存器 146
表 2-56. CPU_FPB 访问类型代码 146
表 2-57. CTRL 寄存器字段说明 147
表 2-58. REMAP 寄存器字段说明 148
表 2-59. COMP0 寄存器字段说明 149
表 2-60. COMP1 寄存器字段说明 150
表 2-61. COMP2 寄存器字段说明 151
表 2-62. COMP3 寄存器字段说明 152
表 2-63. COMP4 寄存器字段说明 153
表 2-64. COMP5 寄存器字段说明 154
表 2-65. COMP6 寄存器字段说明 155
表 2-66. COMP7 寄存器字段说明 156
表 2-67. CPU_ITM 寄存器 157
表 2-68. CPU_ITM 访问类型代码 158
表 2-69. STIM0 寄存器字段说明 159
表 2-70. STIM1 寄存器字段说明 160
表 2-71. STIM2 寄存器字段说明 161
表 2-72. STIM3 寄存器字段说明 162
表 2-73. STIM4 寄存器字段说明 163
表 2-74. STIM5 寄存器字段说明 164
表 2-75. STIM6 寄存器字段说明 165
表 2-76. STIM7 寄存器字段说明 166
表 2-77. STIM8 寄存器字段说明 167
表 2-78. STIM9 寄存器字段说明 168
表 2-79. STIM10 寄存器字段说明 169
表 2-80. STIM11 寄存器字段说明 170
表 2-81. STIM12 寄存器字段说明 171
表 2-82. STIM13 寄存器字段说明 172
表 2-83. STIM14 寄存器字段说明 173
表 2-84. STIM15 寄存器字段说明 174
表 2-85. STIM16 寄存器字段说明 175
表 2-86. STIM17 寄存器字段说明 176
表 2-87. STIM18 寄存器字段说明 177
表 2-88. STIM19 寄存器字段说明 178
表 2-89. STIM20 寄存器字段说明 179
表 2-90. STIM21 寄存器字段说明 180
表 2-91. STIM22 寄存器字段说明 181
表 2-92. STIM23 寄存器字段说明 182
表 2-93. STIM24 寄存器字段说明 183
表 2-94. STIM25 寄存器字段说明 184

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 2-95. STIM26 寄存器字段说明 185
表 2-96. STIM27 寄存器字段说明 186
表 2-97. STIM28 寄存器字段说明 187
表 2-98. STIM29 寄存器字段说明 188
表 2-99. STIM30 寄存器字段说明 189
表 2-100. STIM31 寄存器字段说明 190
表 2-101. TER 寄存器字段说明 191
表 2-102. TPR 寄存器字段说明 193
表 2-103. TCR 寄存器字段说明 194
表 2-104. LAR 寄存器字段说明 196
表 2-105. LSR 寄存器字段说明 197
表 2-106. CPU_SCS 寄存器 198
表 2-107. CPU_SCS 访问类型代码 201
表 2-108. ICTR 寄存器字段说明 202
表 2-109. ACTLR 寄存器字段说明 203
表 2-110. STCSR 寄存器字段说明 204
表 2-111. STRVR 寄存器字段说明 205
表 2-112. STCVR 寄存器字段说明 206
表 2-113. STCR 寄存器字段说明 207
表 2-114. NVIC_ISER0 寄存器字段说明 208
表 2-115. NVIC_ISER1 寄存器字段说明 211
表 2-116. NVIC_ICER0 寄存器字段说明 212
表 2-117. NVIC_ICER1 寄存器字段说明 215
表 2-118. NVIC_ISPR0 寄存器字段说明 216
表 2-119. NVIC_ISPR1 寄存器字段说明 219
表 2-120. NVIC_ICPR0 寄存器字段说明 220
表 2-121. NVIC_ICPR1 寄存器字段说明 223
表 2-122. NVIC_IABR0 寄存器字段说明 224
表 2-123. NVIC_IABR1 寄存器字段说明 227
表 2-124. NVIC_IPR0 寄存器字段说明 228
表 2-125. NVIC_IPR1 寄存器字段说明 229
表 2-126. NVIC_IPR2 寄存器字段说明 230
表 2-127. NVIC_IPR3 寄存器字段说明 231
表 2-128. NVIC_IPR4 寄存器字段说明 232
表 2-129. NVIC_IPR5 寄存器字段说明 233
表 2-130. NVIC_IPR6 寄存器字段说明 234
表 2-131. NVIC_IPR7 寄存器字段说明 235
表 2-132. NVIC_IPR8 寄存器字段说明 236
表 2-133. NVIC_IPR9 寄存器字段说明 237
表 2-134. CPUID 寄存器字段说明 238
表 2-135. ICSR 寄存器字段说明 239
表 2-136. VTOR 寄存器字段说明 241
表 2-137. AIRCR 寄存器字段说明 242
表 2-138. SCR 寄存器字段说明 243
表 2-139. CCR 寄存器字段说明 244
表 2-140. SHPR1 寄存器字段说明 246
表 2-141. SHPR2 寄存器字段说明 247
表 2-142. SHPR3 寄存器字段说明 248
表 2-143. SHCSR 寄存器字段说明 249
表 2-144. CFSR 寄存器字段说明 251
表 2-145. HFSR 寄存器字段说明 253
表 2-146. DFSR 寄存器字段说明 254
表 2-147. MMFAR 寄存器字段说明 255
表 2-148. BFAR 寄存器字段说明 256
表 2-149. AFSR 寄存器字段说明 257
表 2-150. ID_PFR0 寄存器字段说明 258

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 2-151. ID_PFR1 寄存器字段说明 259
表 2-152. ID_DFR0 寄存器字段说明 260
表 2-153. ID_AFR0 寄存器字段说明 261
表 2-154. ID_MMFR0 寄存器字段说明 262
表 2-155. ID_MMFR1 寄存器字段说明 263
表 2-156. ID_MMFR2 寄存器字段说明 264
表 2-157. ID_MMFR3 寄存器字段说明 265
表 2-158. ID_ISAR0 寄存器字段说明 266
表 2-159. ID_ISAR1 寄存器字段说明 267
表 2-160. ID_ISAR2 寄存器字段说明 268
表 2-161. ID_ISAR3 寄存器字段说明 269
表 2-162. ID_ISAR4 寄存器字段说明 270
表 2-163. CPACR 寄存器字段说明 271
表 2-164. MPU_TYPE 寄存器字段说明 272
表 2-165. MPU_CTRL 寄存器字段说明 273
表 2-166. MPU_RNR 寄存器字段说明 274
表 2-167. MPU_RBAR 寄存器字段说明 275
表 2-168. MPU_RASR 寄存器字段说明 276
表 2-169. MPU_RBAR_A1 寄存器字段说明 278
表 2-170. MPU_RASR_A1 寄存器字段说明 279
表 2-171. MPU_RBAR_A2 寄存器字段说明 280
表 2-172. MPU_RASR_A2 寄存器字段说明 281
表 2-173. MPU_RBAR_A3 寄存器字段说明 282
表 2-174. MPU_RASR_A3 寄存器字段说明 283
表 2-175. DHCSR 寄存器字段说明 284
表 2-176. DCRSR 寄存器字段说明 286
表 2-177. DCRDR 寄存器字段说明 287
表 2-178. DEMCR 寄存器字段说明 288
表 2-179. STIR 寄存器字段说明 290
表 2-180. FPCCR 寄存器字段说明 291
表 2-181. FPCAR 寄存器字段说明 293
表 2-182. FPDSCR 寄存器字段说明 294
表 2-183. MVFR0 寄存器字段说明 295
表 2-184. MVFR1 寄存器字段说明 296
表 2-185. CPU_TPIU 寄存器 297
表 2-186. CPU_TPIU 访问类型代码 297
表 2-187. SSPSR 寄存器字段说明 298
表 2-188. CSPSR 寄存器字段说明 299
表 2-189. ACPR 寄存器字段说明 300
表 2-190. SPPR 寄存器字段说明 301
表 2-191. FFSR 寄存器字段说明 302
表 2-192. FFCR 寄存器字段说明 303
表 2-193. FSCR 寄存器字段说明 304
表 2-194. CLAIMMASK 寄存器字段说明 305
表 2-195. CLAIMSET 寄存器字段说明 306
表 2-196. CLAIMTAG 寄存器字段说明 307
表 2-197. CLAIMCLR 寄存器字段说明 308
表 2-198. DEVID 寄存器字段说明 309
表 3-1. 存储器映射 310
表 4-1. 内核外设寄存器区域 313
表 5-1. 异常类型 321
表 5-2. 异常返回行为 325
表 5-3. 故障 326
表 5-4. 故障状态和故障地址寄存器 327
表 5-5. MCU 事件结构输入事件 331
表 5-6. 冻结订阅者事件选择 335

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 5-7. AON 事件 336
表 5-8. AON_EVENT 寄存器 338
表 5-9. AON_EVENT 访问类型代码 338
表 5-10. MCUWUSEL 寄存器字段说明 339
表 5-11. MCUWUSEL1 寄存器字段说明 344
表 5-12. EVTOMCUSEL 寄存器字段说明 349
表 5-13. RTCSEL 寄存器字段说明 353
表 5-14. EVENT 寄存器 354
表 5-15. EVENT 访问类型代码 358
表 5-16. CPUIRQSEL0 寄存器字段说明 359
表 5-17. CPUIRQSEL1 寄存器字段说明 360
表 5-18. CPUIRQSEL2 寄存器字段说明 361
表 5-19. CPUIRQSEL3 寄存器字段说明 362
表 5-20. CPUIRQSEL4 寄存器字段说明 363
表 5-21. CPUIRQSEL5 寄存器字段说明 364
表 5-22. CPUIRQSEL6 寄存器字段说明 365
表 5-23. CPUIRQSEL7 寄存器字段说明 366
表 5-24. CPUIRQSEL8 寄存器字段说明 367
表 5-25. CPUIRQSEL9 寄存器字段说明 368
表 5-26. CPUIRQSEL10 寄存器字段说明 369
表 5-27. CPUIRQSEL11 寄存器字段说明 370
表 5-28. CPUIRQSEL12 寄存器字段说明 371
表 5-29. CPUIRQSEL13 寄存器字段说明 372
表 5-30. CPUIRQSEL14 寄存器字段说明 373
表 5-31. CPUIRQSEL15 寄存器字段说明 374
表 5-32. CPUIRQSEL16 寄存器字段说明 375
表 5-33. CPUIRQSEL17 寄存器字段说明 376
表 5-34. CPUIRQSEL18 寄存器字段说明 377
表 5-35. CPUIRQSEL19 寄存器字段说明 378
表 5-36. CPUIRQSEL20 寄存器字段说明 379
表 5-37. CPUIRQSEL21 寄存器字段说明 380
表 5-38. CPUIRQSEL22 寄存器字段说明 381
表 5-39. CPUIRQSEL23 寄存器字段说明 382
表 5-40. CPUIRQSEL24 寄存器字段说明 383
表 5-41. CPUIRQSEL25 寄存器字段说明 384
表 5-42. CPUIRQSEL26 寄存器字段说明 385
表 5-43. CPUIRQSEL27 寄存器字段说明 386
表 5-44. CPUIRQSEL28 寄存器字段说明 387
表 5-45. CPUIRQSEL29 寄存器字段说明 388
表 5-46. CPUIRQSEL30 寄存器字段说明 389
表 5-47. CPUIRQSEL31 寄存器字段说明 391
表 5-48. CPUIRQSEL32 寄存器字段说明 392
表 5-49. CPUIRQSEL33 寄存器字段说明 393
表 5-50. CPUIRQSEL34 寄存器字段说明 394
表 5-51. CPUIRQSEL35 寄存器字段说明 395
表 5-52. CPUIRQSEL36 寄存器字段说明 396
表 5-53. CPUIRQSEL37 寄存器字段说明 397
表 5-54. RFCSEL0 寄存器字段说明 398
表 5-55. RFCSEL1 寄存器字段说明 399
表 5-56. RFCSEL2 寄存器字段说明 400
表 5-57. RFCSEL3 寄存器字段说明 401
表 5-58. RFCSEL4 寄存器字段说明 402
表 5-59. RFCSEL5 寄存器字段说明 403
表 5-60. RFCSEL6 寄存器字段说明 404
表 5-61. RFCSEL7 寄存器字段说明 405
表 5-62. RFCSEL8 寄存器字段说明 406

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 5-63. RFCSEL9 寄存器字段说明.....407
表 5-64. GPT0ACAPTSEL 寄存器字段说明.....409
表 5-65. GPT0BCAPTSEL 寄存器字段说明.....412
表 5-66. GPT1ACAPTSEL 寄存器字段说明.....415
表 5-67. GPT1BCAPTSEL 寄存器字段说明.....418
表 5-68. GPT2ACAPTSEL 寄存器字段说明.....421
表 5-69. GPT2BCAPTSEL 寄存器字段说明.....424
表 5-70. UDMACH1SSEL 寄存器字段说明.....427
表 5-71. UDMACH1BSEL 寄存器字段说明.....428
表 5-72. UDMACH2SSEL 寄存器字段说明.....429
表 5-73. UDMACH2BSEL 寄存器字段说明.....430
表 5-74. UDMACH3SSEL 寄存器字段说明.....431
表 5-75. UDMACH3BSEL 寄存器字段说明.....432
表 5-76. UDMACH4SSEL 寄存器字段说明.....433
表 5-77. UDMACH4BSEL 寄存器字段说明.....434
表 5-78. UDMACH5SSEL 寄存器字段说明.....435
表 5-79. UDMACH5BSEL 寄存器字段说明.....436
表 5-80. UDMACH6SSEL 寄存器字段说明.....437
表 5-81. UDMACH6BSEL 寄存器字段说明.....438
表 5-82. UDMACH7SSEL 寄存器字段说明.....439
表 5-83. UDMACH7BSEL 寄存器字段说明.....440
表 5-84. UDMACH8SSEL 寄存器字段说明.....441
表 5-85. UDMACH8BSEL 寄存器字段说明.....442
表 5-86. UDMACH9SSEL 寄存器字段说明.....443
表 5-87. UDMACH9BSEL 寄存器字段说明.....444
表 5-88. UDMACH10SSEL 寄存器字段说明.....445
表 5-89. UDMACH10BSEL 寄存器字段说明.....446
表 5-90. UDMACH11SSEL 寄存器字段说明.....447
表 5-91. UDMACH11BSEL 寄存器字段说明.....448
表 5-92. UDMACH12SSEL 寄存器字段说明.....449
表 5-93. UDMACH12BSEL 寄存器字段说明.....450
表 5-94. UDMACH13BSEL 寄存器字段说明.....451
表 5-95. UDMACH14BSEL 寄存器字段说明.....452
表 5-96. UDMACH15BSEL 寄存器字段说明.....456
表 5-97. UDMACH16SSEL 寄存器字段说明.....457
表 5-98. UDMACH16BSEL 寄存器字段说明.....458
表 5-99. UDMACH17SSEL 寄存器字段说明.....459
表 5-100. UDMACH17BSEL 寄存器字段说明.....460
表 5-101. UDMACH21SSEL 寄存器字段说明.....461
表 5-102. UDMACH21BSEL 寄存器字段说明.....462
表 5-103. UDMACH22SSEL 寄存器字段说明.....463
表 5-104. UDMACH22BSEL 寄存器字段说明.....464
表 5-105. UDMACH23SSEL 寄存器字段说明.....465
表 5-106. UDMACH23BSEL 寄存器字段说明.....466
表 5-107. UDMACH24SSEL 寄存器字段说明.....467
表 5-108. UDMACH24BSEL 寄存器字段说明.....468
表 5-109. GPT3ACAPTSEL 寄存器字段说明.....469
表 5-110. GPT3BCAPTSEL 寄存器字段说明.....472
表 5-111. AUXSEL0 寄存器字段说明.....475
表 5-112. CM3NMISEL0 寄存器字段说明.....476
表 5-113. I2SSTMPSEL0 寄存器字段说明.....477
表 5-114. FRZSEL0 寄存器字段说明.....478
表 5-115. SWEV 寄存器字段说明.....479
表 6-1. 参考资料.....480
表 6-2. IEEE 1149.7 特性子集.....483
表 6-3. OScan 扫描数据包内容.....484

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 6-4. cJTAG 命令 486
表 6-5. 从 TAP 顺序 490
表 6-6. 寄存器汇总 491
表 6-7. 指令寄存器操作码 491
表 6-8. 器件标识寄存器说明 493
表 6-9. 用户代码寄存器说明 493
表 6-10. ICEPick 标识寄存器说明 493
表 6-11. 连接寄存器描述 494
表 6-12. ROUTER DR 扫描链说明 494
表 6-13. 控制块寄存器 495
表 6-14. All0s 寄存器 496
表 6-15. ICEPick 控制寄存器 496
表 6-16. ICEPick 链接模式寄存器 496
表 6-17. ICEPick TAP 链接模式 496
表 6-18. 测试 TAP 链接寄存器 497
表 6-19. 次级测试 TAP 寄存器 (STTR) 497
表 6-20. 调试 TAP 链接寄存器 497
表 6-21. 次级调试 TAP 寄存器 (SDTR) 498
表 6-22. 复位控制 499
表 6-23. 通过 WUC TAP 支持的调试功能 501
表 6-24. 分析器寄存器字段 501
表 6-25. 不同器件的边界扫描 I/O 504
表 7-1. 省电特性 507
表 7-2. TI-RTOS 中的电源模式 507
表 7-3. 系统 CPU 模式 508
表 7-4. 系统时钟 512
表 7-5. TI-RTOS 中定义的电源模式 517
表 7-6. 在待机模式下设置 CC13x2 及 CC26x2 的示例序列 519
表 7-7. 进入关断状态的序列示例 520
表 7-8. DDI_0_OSC 寄存器 522
表 7-9. DDI_0_OSC 访问类型代码 522
表 7-10. CTL0 寄存器字段说明 523
表 7-11. CTL1 寄存器字段说明 525
表 7-12. RADCEXTCFG 寄存器字段说明 526
表 7-13. AMPCOMPCTL 寄存器字段说明 527
表 7-14. AMPCOMPTH1 寄存器字段说明 528
表 7-15. AMPCOMPTH2 寄存器字段说明 529
表 7-16. ANABYPASSVAL1 寄存器字段说明 530
表 7-17. ANABYPASSVAL2 寄存器字段说明 531
表 7-18. ATESTCTL 寄存器字段说明 532
表 7-19. ADCDOUBLERNANOAMPCTL 寄存器字段说明 533
表 7-20. XOSCHFCTL 寄存器字段说明 534
表 7-21. LFOSCCTL 寄存器字段说明 535
表 7-22. RCOSCHFCTL 寄存器字段说明 536
表 7-23. RCOSCMFCTL 寄存器字段说明 537
表 7-24. STAT0 寄存器字段说明 538
表 7-25. STAT1 寄存器字段说明 540
表 7-26. STAT2 寄存器字段说明 542
表 7-27. PRCM 寄存器 543
表 7-28. PRCM 访问类型代码 546
表 7-29. INFRCLKDIVR 寄存器字段说明 547
表 7-30. INFRCLKDIVS 寄存器字段说明 548
表 7-31. INFRCLKDIVDS 寄存器字段说明 549
表 7-32. VDCTL 寄存器字段说明 550
表 7-33. CLKLOADCTL 寄存器字段说明 551
表 7-34. RFCCLKG 寄存器字段说明 553

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 7-35. VIMSCLKG 寄存器字段说明 554
表 7-36. SECDMACLKGR 寄存器字段说明 555
表 7-37. SECDMACLKGS 寄存器字段说明 557
表 7-38. SECDMACLKGDS 寄存器字段说明 558
表 7-39. GPIOCLKGR 寄存器字段说明 559
表 7-40. GPIOCLKGS 寄存器字段说明 560
表 7-41. GPIOCLKGDS 寄存器字段说明 561
表 7-42. GPTCLKGR 寄存器字段说明 562
表 7-43. GPTCLKGS 寄存器字段说明 563
表 7-44. GPTCLKGDS 寄存器字段说明 564
表 7-45. I2CCLKGR 寄存器字段说明 565
表 7-46. I2CCLKGS 寄存器字段说明 566
表 7-47. I2CCLKGDS 寄存器字段说明 567
表 7-48. UARTCLKGR 寄存器字段说明 568
表 7-49. UARTCLKGS 寄存器字段说明 569
表 7-50. UARTCLKGDS 寄存器字段说明 570
表 7-51. SSICLKGR 寄存器字段说明 571
表 7-52. SSICLKGS 寄存器字段说明 572
表 7-53. SSICLKGDS 寄存器字段说明 573
表 7-54. I2SCLKGR 寄存器字段说明 574
表 7-55. I2SCLKGS 寄存器字段说明 575
表 7-56. I2SCLKGDS 寄存器字段说明 576
表 7-57. SYSBUSCLKDIV 寄存器字段说明 577
表 7-58. CPUCLKDIV 寄存器字段说明 578
表 7-59. PERBUSCPUCLKDIV 寄存器字段说明 579
表 7-60. PERDMACLKDIV 寄存器字段说明 580
表 7-61. I2SBCLKSEL 寄存器字段说明 581
表 7-62. GPTCLKDIV 寄存器字段说明 582
表 7-63. I2SCLKCTL 寄存器字段说明 583
表 7-64. I2SMCLKDIV 寄存器字段说明 584
表 7-65. I2SBCLKDIV 寄存器字段说明 585
表 7-66. I2SWCLKDIV 寄存器字段说明 586
表 7-67. RESETSECDMA 寄存器字段说明 587
表 7-68. RESETGPIO 寄存器字段说明 588
表 7-69. RESETGPT 寄存器字段说明 589
表 7-70. RESETI2C 寄存器字段说明 590
表 7-71. RESETUART 寄存器字段说明 591
表 7-72. RESETSSI 寄存器字段说明 592
表 7-73. RESETI2S 寄存器字段说明 593
表 7-74. PDCTL0 寄存器字段说明 594
表 7-75. PDCTL0RFC 寄存器字段说明 595
表 7-76. PDCTL0SERIAL 寄存器字段说明 596
表 7-77. PDCTL0PERIPH 寄存器字段说明 597
表 7-78. PDSTAT0 寄存器字段说明 598
表 7-79. PDSTAT0RFC 寄存器字段说明 599
表 7-80. PDSTAT0SERIAL 寄存器字段说明 600
表 7-81. PDSTAT0PERIPH 寄存器字段说明 601
表 7-82. PDCTL1 寄存器字段说明 602
表 7-83. PDCTL1CPU 寄存器字段说明 603
表 7-84. PDCTL1RFC 寄存器字段说明 604
表 7-85. PDCTL1VIMS 寄存器字段说明 605
表 7-86. PDSTAT1 寄存器字段说明 606
表 7-87. PDSTAT1BUS 寄存器字段说明 607
表 7-88. PDSTAT1RFC 寄存器字段说明 608
表 7-89. PDSTAT1CPU 寄存器字段说明 609
表 7-90. PDSTAT1VIMS 寄存器字段说明 610

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 7-91. RFCBITS 寄存器字段说明.....611
表 7-92. RFCMODESEL 寄存器字段说明.....612
表 7-93. RFCMODEHWOPT 寄存器字段说明.....613
表 7-94. PWRPROFSTAT 寄存器字段说明.....614
表 7-95. MCUSRAMCFG 寄存器字段说明.....615
表 7-96. RAMRETEN 寄存器字段说明.....616
表 7-97. OSCIMSC 寄存器字段说明.....617
表 7-98. OSCRIS 寄存器字段说明.....618
表 7-99. OSCICR 寄存器字段说明.....621
表 7-100. AON_PMCTL 寄存器.....622
表 7-101. AON_PMCTL 访问类型代码.....622
表 7-102. AUXSCECLK 寄存器字段说明.....623
表 7-103. RAMCFG 寄存器字段说明.....624
表 7-104. PWRCTL 寄存器字段说明.....625
表 7-105. PWRSTAT 寄存器字段说明.....626
表 7-106. SHUTDOWN 寄存器字段说明.....627
表 7-107. RECHARGECFG 寄存器字段说明.....628
表 7-108. RECHARGESTAT 寄存器字段说明.....629
表 7-109. OSCCFG 寄存器字段说明.....630
表 7-110. RESETCTL 寄存器字段说明.....631
表 7-111. SLEEPCTL 寄存器字段说明.....633
表 7-112. JTAGCFG 寄存器字段说明.....634
表 7-113. JTAGUSERCODE 寄存器字段说明.....635
表 8-1. VIMS 存储器的有效保留组合.....641
表 8-2. CC13x2 和 CC26x2 存储器写入/擦除保护 (1) (2) (3).....643
表 8-3. FLASH 寄存器.....646
表 8-4. FLASH 访问类型代码.....651
表 8-5. STAT 寄存器字段说明.....652
表 8-6. CFG 寄存器字段说明.....653
表 8-7. SYSCODE_START 寄存器字段说明.....654
表 8-8. FLASH_SIZE 寄存器字段说明.....655
表 8-9. FWLOCK 寄存器字段说明.....656
表 8-10. FWFLAG 寄存器字段说明.....657
表 8-11. EFUSE 寄存器字段说明.....658
表 8-12. EFUSEADDR 寄存器字段说明.....659
表 8-13. DATAUPPER 寄存器字段说明.....660
表 8-14. DATALOWER 寄存器字段说明.....661
表 8-15. EFUSECFG 寄存器字段说明.....662
表 8-16. EFUSESTAT 寄存器字段说明.....663
表 8-17. ACC 寄存器字段说明.....664
表 8-18. BOUNDARY 寄存器字段说明.....665
表 8-19. EFUSEFLAG 寄存器字段说明.....666
表 8-20. EFUSEKEY 寄存器字段说明.....667
表 8-21. EFUSERELEASE 寄存器字段说明.....668
表 8-22. EFUSEPINS 寄存器字段说明.....669
表 8-23. EFUSECRA 寄存器字段说明.....670
表 8-24. EFUSEREAD 寄存器字段说明.....671
表 8-25. EFUSEPROGRAM 寄存器字段说明.....672
表 8-26. EFUSEERROR 寄存器字段说明.....673
表 8-27. SINGLEBIT 寄存器字段说明.....674
表 8-28. TWOBIT 寄存器字段说明.....675
表 8-29. SELFTESTCYC 寄存器字段说明.....676
表 8-30. SELFTESTSIGN 寄存器字段说明.....677
表 8-31. FRDCTL 寄存器字段说明.....678
表 8-32. FSPRD 寄存器字段说明.....679
表 8-33. FEDACCTL1 寄存器字段说明.....680

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 8-34. FEDACSTAT 寄存器字段说明.....681
表 8-35. FBPROT 寄存器字段说明.....682
表 8-36. FBSE 寄存器字段说明.....683
表 8-37. FBBUSY 寄存器字段说明.....684
表 8-38. FBAC 寄存器字段说明.....685
表 8-39. FBFALLBACK 寄存器字段说明.....686
表 8-40. FBPRDY 寄存器字段说明.....687
表 8-41. FPAC1 寄存器字段说明.....688
表 8-42. FPAC2 寄存器字段说明.....689
表 8-43. FMAC 寄存器字段说明.....690
表 8-44. FMSTAT 寄存器字段说明.....691
表 8-45. FLOCK 寄存器字段说明.....692
表 8-46. FVREADCT 寄存器字段说明.....693
表 8-47. FVHVCT1 寄存器字段说明.....694
表 8-48. FVHVCT2 寄存器字段说明.....695
表 8-49. FVHVCT3 寄存器字段说明.....696
表 8-50. FVNVCT 寄存器字段说明.....697
表 8-51. FVSLP 寄存器字段说明.....698
表 8-52. FVWLCT 寄存器字段说明.....699
表 8-53. FEFUSECTL 寄存器字段说明.....700
表 8-54. FEFUSESTAT 寄存器字段说明.....701
表 8-55. FEFUSEDATA 寄存器字段说明.....702
表 8-56. FSEQPMP 寄存器字段说明.....703
表 8-57. FBSTROBES 寄存器字段说明.....704
表 8-58. FPSTROBES 寄存器字段说明.....705
表 8-59. FBMODE 寄存器字段说明.....706
表 8-60. FTCR 寄存器字段说明.....707
表 8-61. FADDR 寄存器字段说明.....708
表 8-62. FTCTL 寄存器字段说明.....709
表 8-63. FWPWRITE0 寄存器字段说明.....710
表 8-64. FWPWRITE1 寄存器字段说明.....711
表 8-65. FWPWRITE2 寄存器字段说明.....712
表 8-66. FWPWRITE3 寄存器字段说明.....713
表 8-67. FWPWRITE4 寄存器字段说明.....714
表 8-68. FWPWRITE5 寄存器字段说明.....715
表 8-69. FWPWRITE6 寄存器字段说明.....716
表 8-70. FWPWRITE7 寄存器字段说明.....717
表 8-71. FWPWRITE_ECC 寄存器字段说明.....718
表 8-72. FSWSTAT 寄存器字段说明.....719
表 8-73. FSM_GLBCTL 寄存器字段说明.....720
表 8-74. FSM_STATE 寄存器字段说明.....721
表 8-75. FSM_STAT 寄存器字段说明.....722
表 8-76. FSM_CMD 寄存器字段说明.....723
表 8-77. FSM_PE_OSU 寄存器字段说明.....724
表 8-78. FSM_VSTAT 寄存器字段说明.....725
表 8-79. FSM_PE_VSU 寄存器字段说明.....726
表 8-80. FSM_CMP_VSU 寄存器字段说明.....727
表 8-81. FSM_EX_VAL 寄存器字段说明.....728
表 8-82. FSM_RD_H 寄存器字段说明.....729
表 8-83. FSM_P_OH 寄存器字段说明.....730
表 8-84. FSM_ERA_OH 寄存器字段说明.....731
表 8-85. FSM_SAV_PPUL 寄存器字段说明.....732
表 8-86. FSM_PE_VH 寄存器字段说明.....733
表 8-87. FSM_PRG_PW 寄存器字段说明.....734
表 8-88. FSM_ERA_PW 寄存器字段说明.....735
表 8-89. FSM_SAV_ERA_PUL 寄存器字段说明.....736

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 8-90. FSM_TIMER 寄存器字段说明 737
表 8-91. FSM_MODE 寄存器字段说明 738
表 8-92. FSM_PGM 寄存器字段说明 739
表 8-93. FSM_ERA 寄存器字段说明 740
表 8-94. FSM_PRG_PUL 寄存器字段说明 741
表 8-95. FSM_ERA_PUL 寄存器字段说明 742
表 8-96. FSM_STEP_SIZE 寄存器字段说明 743
表 8-97. FSM_PUL_CNTR 寄存器字段说明 744
表 8-98. FSM_EC_STEP_HEIGHT 寄存器字段说明 745
表 8-99. FSM_ST_MACHINE 寄存器字段说明 746
表 8-100. FSM_FLES 寄存器字段说明 747
表 8-101. FSM_WR_ENA 寄存器字段说明 748
表 8-102. FSM_ACC_PP 寄存器字段说明 749
表 8-103. FSM_ACC_EP 寄存器字段说明 750
表 8-104. FSM_ADDR 寄存器字段说明 751
表 8-105. FSM_SECTOR 寄存器字段说明 752
表 8-106. FMC_REV_ID 寄存器字段说明 753
表 8-107. FSM_ERR_ADDR 寄存器字段说明 754
表 8-108. FSM_PGM_MAXPUL 寄存器字段说明 755
表 8-109. FSM_EXECUTE 寄存器字段说明 756
表 8-110. FSM_SECTOR1 寄存器字段说明 757
表 8-111. FSM_SECTOR2 寄存器字段说明 758
表 8-112. FSM_BSLE0 寄存器字段说明 759
表 8-113. FSM_BSLE1 寄存器字段说明 760
表 8-114. FSM_BSLP0 寄存器字段说明 761
表 8-115. FSM_BSLP1 寄存器字段说明 762
表 8-116. FSM_PGM128 寄存器字段说明 763
表 8-117. FCFG_BANK 寄存器字段说明 764
表 8-118. FCFG_WRAPPER 寄存器字段说明 765
表 8-119. FCFG_BNK_TYPE 寄存器字段说明 766
表 8-120. FCFG_B0_START 寄存器字段说明 767
表 8-121. FCFG_B1_START 寄存器字段说明 768
表 8-122. FCFG_B2_START 寄存器字段说明 769
表 8-123. FCFG_B3_START 寄存器字段说明 770
表 8-124. FCFG_B4_START 寄存器字段说明 771
表 8-125. FCFG_B5_START 寄存器字段说明 772
表 8-126. FCFG_B6_START 寄存器字段说明 773
表 8-127. FCFG_B7_START 寄存器字段说明 774
表 8-128. FCFG_B0_SSIZE0 寄存器字段说明 775
表 8-129. VIMS 寄存器 776
表 8-130. VIMS 访问类型代码 776
表 8-131. STAT 寄存器字段说明 777
表 8-132. CTL 寄存器字段说明 778
表 9-1. SRAM_MMR 寄存器 781
表 9-2. SRAM_MMR 访问类型代码 781
表 9-3. PER_CTL 寄存器字段说明 782
表 9-4. PER_CHK 寄存器字段说明 783
表 9-5. PER_DBG 寄存器字段说明 784
表 9-6. MEM_CTL 寄存器字段说明 785
表 9-7. SRAM 寄存器 786
表 9-8. SRAM 访问类型代码 786
表 9-9. BANK0_y 寄存器字段说明 787
表 9-10. BANK1_y 寄存器字段说明 788
表 9-11. BANK2_y 寄存器字段说明 789
表 9-12. BANK3_y 寄存器字段说明 790
表 9-13. BANK4_y 寄存器字段说明 791

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 10-1. 协 议 确 认 和 非确认字节 795
表 10-2. 串行接口的配置 796
表 10-3. 支持的引导加载程序命令 797
表 10-4. 定义的状态值 801
表 10-5. 定义了 CCFG 字段 ID 和字段值 806
表 11-1. CCFG 寄存器 810
表 11-2. CCFG 访问类型代码 811
表 11-3. EXT_LF_CLK 寄存器字段说明 812
表 11-4. MODE_CONF_1 寄存器字段说明 813
表 11-5. SIZE_AND_DIS_FLAGS 寄存器字段说明 815
表 11-6. MODE_CONF 寄存器字段说明 817
表 11-7. VOLT_LOAD_0 寄存器字段说明 819
表 11-8. VOLT_LOAD_1 寄存器字段说明 820
表 11-9. RTC_OFFSET 寄存器字段说明 821
表 11-10. FREQ_OFFSET 寄存器字段说明 822
表 11-11. IEEE_MAC_0 寄存器字段说明 823
表 11-12. IEEE_MAC_1 寄存器字段说明 824
表 11-13. IEEE_BLE_0 寄存器字段说明 825
表 11-14. IEEE_BLE_1 寄存器字段说明 826
表 11-15. BL_CONFIG 寄存器字段说明 827
表 11-16. ERASE_CONF 寄存器字段说明 828
表 11-17. CCFG_TI_OPTIONS 寄存器字段说明 829
表 11-18. CCFG_TAP_DAP_0 寄存器字段说明 830
表 11-19. CCFG_TAP_DAP_1 寄存器字段说明 831
表 11-20. IMAGE_VALID_CONF 寄存器字段说明 832
表 11-21. CCFG_PROT_31_0 寄存器字段说明 833
表 11-22. CCFG_PROT_63_32 寄存器字段说明 835
表 11-23. CCFG_PROT_95_64 寄存器字段说明 837
表 11-24. CCFG_PROT_127_96 寄存器字段说明 839
表 11-25. FCFG1 寄存器 841
表 11-26. FCFG1 访问类型代码 844
表 11-27. MISC_CONF_1 寄存器字段说明 845
表 11-28. MISC_CONF_2 寄存器字段说明 846
表 11-29. HPOSC_MEAS_5 寄存器字段说明 847
表 11-30. HPOSC_MEAS_4 寄存器字段说明 848
表 11-31. HPOSC_MEAS_3 寄存器字段说明 849
表 11-32. HPOSC_MEAS_2 寄存器字段说明 850
表 11-33. HPOSC_MEAS_1 寄存器字段说明 851
表 11-34. CONFIG_CC26_FE 寄存器字段说明 852
表 11-35. CONFIG_CC13_FE 寄存器字段说明 853
表 11-36. CONFIG_RF_COMMON 寄存器字段说明 854
表 11-37. CONFIG_SYNTH_DIV2_CC26_2G4 寄存器字段说明 855
表 11-38. CONFIG_SYNTH_DIV2_CC13_2G4 寄存器字段说明 856
表 11-39. CONFIG_SYNTH_DIV2_CC26_1G 寄存器字段说明 857
表 11-40. CONFIG_SYNTH_DIV2_CC13_1G 寄存器字段说明 858
表 11-41. CONFIG_SYNTH_DIV4_CC26 寄存器字段说明 859
表 11-42. CONFIG_SYNTH_DIV4_CC13 寄存器字段说明 860
表 11-43. CONFIG_SYNTH_DIV5 寄存器字段说明 861
表 11-44. CONFIG_SYNTH_DIV6_CC26 寄存器字段说明 862
表 11-45. CONFIG_SYNTH_DIV6_CC13 寄存器字段说明 863
表 11-46. CONFIG_SYNTH_DIV10 寄存器字段说明 864
表 11-47. CONFIG_SYNTH_DIV12_CC26 寄存器字段说明 865
表 11-48. CONFIG_SYNTH_DIV12_CC13 寄存器字段说明 866
表 11-49. CONFIG_SYNTH_DIV15 寄存器字段说明 867
表 11-50. CONFIG_SYNTH_DIV30 寄存器字段说明 868

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 11-51. FLASH_NUMBER 寄存器字段说明 869
表 11-52. FLASH_COORDINATE 寄存器字段说明 870
表 11-53. FLASH_E_P 寄存器字段说明 871
表 11-54. FLASH_C_E_P_R 寄存器字段说明 872
表 11-55. FLASH_P_R_PV 寄存器字段说明 873
表 11-56. FLASH_EH_SEQ 寄存器字段说明 874
表 11-57. FLASH_VHV_E 寄存器字段说明 875
表 11-58. FLASH_PP 寄存器字段说明 876
表 11-59. FLASH_PROG_EP 寄存器字段说明 877
表 11-60. FLASH_ERA_PW 寄存器字段说明 878
表 11-61. FLASH_VHV 寄存器字段说明 879
表 11-62. FLASH_VHV_PV 寄存器字段说明 880
表 11-63. FLASH_V 寄存器字段说明 881
表 11-64. USER_ID 寄存器字段说明 882
表 11-65. FLASH_OTP_DATA3 寄存器字段说明 884
表 11-66. ANA2_TRIM 寄存器字段说明 885
表 11-67. LDO_TRIM 寄存器字段说明 886
表 11-68. MAC_BLE_0 寄存器字段说明 887
表 11-69. MAC_BLE_1 寄存器字段说明 888
表 11-70. MAC_15_4_0 寄存器字段说明 889
表 11-71. MAC_15_4_1 寄存器字段说明 890
表 11-72. FLASH_OTP_DATA4 寄存器字段说明 891
表 11-73. MISC_TRIM 寄存器字段说明 893
表 11-74. RCOSC_HF_TEMPCOMP 寄存器字段说明 894
表 11-75. ICEPICK_DEVICE_ID 寄存器字段说明 895
表 11-76. FCFG1_REVISION 寄存器字段说明 896
表 11-77. MISC_OTP_DATA 寄存器字段说明 897
表 11-78. IOCONF 寄存器字段说明 898
表 11-79. CONFIG_IF_ADC 寄存器字段说明 899
表 11-80. CONFIG_OSC_TOP 寄存器字段说明 900
表 11-81. SOC_ADC_ABS_GAIN 寄存器字段说明 901
表 11-82. SOC_ADC_REL_GAIN 寄存器字段说明 902
表 11-83. SOC_ADC_OFFSET_INT 寄存器字段说明 903
表 11-84. SOC_ADC_REF_TRIM_AND_OFFSET_EXT 寄存器字段说明 904
表 11-85. AMPCOMP_TH1 寄存器字段说明 905
表 11-86. AMPCOMP_TH2 寄存器字段说明 906
表 11-87. AMPCOMP_CTRL1 寄存器字段说明 907
表 11-88. ANABYPASS_VALUE2 寄存器字段说明 908
表 11-89. VOLT_TRIM 寄存器字段说明 909
表 11-90. OSC_CONF 寄存器字段说明 910
表 11-91. FREQ_OFFSET 寄存器字段说明 912
表 11-92. MISC_OTP_DATA_1 寄存器字段说明 913
表 11-93. SHDW_DIE_ID_0 寄存器字段说明 914
表 11-94. SHDW_DIE_ID_1 寄存器字段说明 915
表 11-95. SHDW_DIE_ID_2 寄存器字段说明 916
表 11-96. SHDW_DIE_ID_3 寄存器字段说明 917
表 11-97. SHDW_OSC_BIAS_LDO_TRIM 寄存器字段说明 918
表 11-98. SHDW_ANA_TRIM 寄存器字段说明 919
表 11-99. OSC_CONF1 寄存器字段说明 920
表 11-100. DAC_BIAS_CNF 寄存器字段说明 921
表 11-101. TFW_PROBE 寄存器字段说明 922
表 11-102. TFW_FT 寄存器字段说明 923
表 11-103. DAC_CAL0 寄存器字段说明 924
表 11-104. DAC_CAL1 寄存器字段说明 925
表 11-105. DAC_CAL2 寄存器字段说明 926
表 11-106. DAC_CAL3 寄存器字段说明 927

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 12-1. 详细的存储器映射 (1) 932
表 12-2. 支持的 DMAC 操作 937
表 12-3. ALGSEL 标志的有效组合 938
表 12-4. AES_KEY 939
表 12-5. AES_KEY 939
表 12-6. 对于 CCM 939
表 12-7. 对于 CBC-MAC 940
表 12-8. 对于 GCM 940
表 12-9. AES 初始化矢量寄存器 940
表 12-10. 初始化矢量，用于常规非 ECB 模式 (CBC/ECB CTR) 940
表 12-11. 对于 GCM 940
表 12-12. 初始化矢量，用于 CCM 941
表 12-13. 初始化矢量，用于 CBC-MAC 941
表 12-14. 每种工作模式的输入/输出块格式 942
表 12-15. AES 标签输出寄存器 942
表 12-16. 适用于 GCM、CCM 和 CBC-MAC 942
表 12-17. 基于 DMA 的操作的性能表 945
表 12-18. PKCP 操作 (1) 964
表 12-19. PKCP 操作的输入向量限制 964
表 12-20. 结果向量内存分配 965
表 12-21. PKCP 结果向量与向量重叠限制 965
表 12-22. ExpMod 操作 966
表 12-23. 操作限制 966
表 12-24. 结果向量和暂存区内存分配（起始于 PKA_DPTR） 967
表 12-25. 结果向量与输入向量的重叠限制 967
表 12-26. 所需 RAM 大小 968
表 12-27. 最大奇数幂数量 968
表 12-28. 示例 PKA RAM 向量分配 969
表 12-29. 基本 PKCP 功能的扩展 970
表 12-30. PKA_SHIFT 结果值 970
表 12-31. 操作限制 970
表 12-32. 结果向量与输入向量的重叠限制 970
表 12-33. 结果向量和暂存区内存分配 (1) 970
表 12-34. PKCP 操作所需时钟 972
表 12-35. ECC 操作 972
表 12-36. PKA_SHIFT 结果值 972
表 12-37. 操作限制 973
表 12-38. 输入和输出向量的重叠限制 973
表 12-39. 结果向量和暂存区的存储器分配 (1) (2) 973
表 12-40. PKA RAM 向量分配 974
表 12-41. PKCP 操作所需时钟 975
表 12-42. 五次模拟的性能值 976
表 12-43. ECC 操作性能值 976
表 12-44. ECC-MUL 操作性能值 977
表 12-45. 序列发生器 ROM 接口信息 978
表 12-46. 寄存器地址映射 978
表 12-47. PKCP 控制、序列发生器和状态、硬件和固件版本寄存器 979
表 12-48. 首字母缩写词 982
表 12-49. 公式和命名规则 983
表 12-50. CRYPTO 寄存器 984
表 12-51. CRYPTO 访问类型代码 987
表 12-52. DMACH0CTL 寄存器字段说明 989
表 12-53. DMACH0EXTADDR 寄存器字段说明 990
表 12-54. DMACH0LEN 寄存器字段说明 991
表 12-55. DMASTAT 寄存器字段说明 992
表 12-56. DMASWRESET 寄存器字段说明 993

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 12-57. DMACH1CTL 寄存器字段说明.....994
表 12-58. DMACH1EXTADDR 寄存器字段说明.....995
表 12-59. DMACH1LEN 寄存器字段说明.....996
表 12-60. DMABUSCFG 寄存器字段说明.....997
表 12-61. DMAPORTERR 寄存器字段说明.....998
表 12-62. DMAHWVER 寄存器字段说明.....999
表 12-63. KEYWRITEAREA 寄存器字段说明.....1000
表 12-64. KEYWRITTENAREA 寄存器字段说明.....1003
表 12-65. KEYSIZE 寄存器字段说明.....1005
表 12-66. KEYREADAREA 寄存器字段说明.....1006
表 12-67. AESKEY2_y 寄存器字段说明.....1007
表 12-68. AESKEY3_y 寄存器字段说明.....1008
表 12-69. AESIV_y 寄存器字段说明.....1009
表 12-70. AESCTL 寄存器字段说明.....1010
表 12-71. AESDATALEN0 寄存器字段说明.....1012
表 12-72. AESDATALEN1 寄存器字段说明.....1013
表 12-73. AESAUTHLEN 寄存器字段说明.....1014
表 12-74. AESDATAOUT0 寄存器字段说明.....1015
表 12-75. AESDATAIN0 寄存器字段说明.....1016
表 12-76. AESDATAOUT1 寄存器字段说明.....1017
表 12-77. AESDATAIN1 寄存器字段说明.....1018
表 12-78. AESDATAOUT2 寄存器字段说明.....1019
表 12-79. AESDATAIN2 寄存器字段说明.....1020
表 12-80. AESDATAOUT3 寄存器字段说明.....1021
表 12-81. AESDATAIN3 寄存器字段说明.....1022
表 12-82. AESTAGOUT_y 寄存器字段说明.....1023
表 12-83. HASHDATAIN1 寄存器字段说明.....1024
表 12-84. HASHDATAIN2 寄存器字段说明.....1025
表 12-85. HASHDATAIN3 寄存器字段说明.....1026
表 12-86. HASHDATAIN4 寄存器字段说明.....1027
表 12-87. HASHDATAIN5 寄存器字段说明.....1028
表 12-88. HASHDATAIN6 寄存器字段说明.....1029
表 12-89. HASHDATAIN7 寄存器字段说明.....1030
表 12-90. HASHDATAIN8 寄存器字段说明.....1031
表 12-91. HASHDATAIN9 寄存器字段说明.....1032
表 12-92. HASHDATAIN10 寄存器字段说明.....1033
表 12-93. HASHDATAIN11 寄存器字段说明.....1034
表 12-94. HASHDATAIN12 寄存器字段说明.....1035
表 12-95. HASHDATAIN13 寄存器字段说明.....1036
表 12-96. HASHDATAIN14 寄存器字段说明.....1037
表 12-97. HASHDATAIN15 寄存器字段说明.....1038
表 12-98. HASHDATAIN16 寄存器字段说明.....1039
表 12-99. HASHDATAIN17 寄存器字段说明.....1040
表 12-100. HASHDATAIN18 寄存器字段说明.....1041
表 12-101. HASHDATAIN19 寄存器字段说明.....1042
表 12-102. HASHDATAIN20 寄存器字段说明.....1043
表 12-103. HASHDATAIN21 寄存器字段说明.....1044
表 12-104. HASHDATAIN22 寄存器字段说明.....1045
表 12-105. HASHDATAIN23 寄存器字段说明.....1046
表 12-106. HASHDATAIN24 寄存器字段说明.....1047
表 12-107. HASHDATAIN25 寄存器字段说明.....1048
表 12-108. HASHDATAIN26 寄存器字段说明.....1049
表 12-109. HASHDATAIN27 寄存器字段说明.....1050
表 12-110. HASHDATAIN28 寄存器字段说明.....1051
表 12-111. HASHDATAIN29 寄存器字段说明.....1052
表 12-112. HASHDATAIN30 寄存器字段说明.....1053

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 12-113. HASHDATAIN31 寄存器字段说明 1054
表 12-114. HASHIOBUFCTRL 寄存器字段说明 1055
表 12-115. HASHMODE 寄存器字段说明 1057
表 12-116. HASHINLENL 寄存器字段说明 1058
表 12-117. HASHINLENH 寄存器字段说明 1059
表 12-118. HASHDIGESTA 寄存器字段说明 1060
表 12-119. HASHDIGESTB 寄存器字段说明 1061
表 12-120. HASHDIGESTC 寄存器字段说明 1062
表 12-121. HASHDIGESTD 寄存器字段说明 1063
表 12-122. HASHDIGESTE 寄存器字段说明 1064
表 12-123. HASHDIGESTF 寄存器字段说明 1065
表 12-124. HASHDIGESTG 寄存器字段说明 1066
表 12-125. HASHDIGESTH 寄存器字段说明 1067
表 12-126. HASHDIGESTI 寄存器字段说明 1068
表 12-127. HASHDIGESTJ 寄存器字段说明 1069
表 12-128. HASHDIGESTK 寄存器字段说明 1070
表 12-129. HASHDIGESTL 寄存器字段说明 1071
表 12-130. HASHDIGESTM 寄存器字段说明 1072
表 12-131. HASHDIGESTN 寄存器字段说明 1073
表 12-132. HASHDIGESTO 寄存器字段说明 1074
表 12-133. HASHDIGESTP 寄存器字段说明 1075
表 12-134. ALGSEL 寄存器字段说明 1076
表 12-135. DMAPROTCTL 寄存器字段说明 1077
表 12-136. SWRESET 寄存器字段说明 1078
表 12-137. IRQTYPE 寄存器字段说明 1079
表 12-138. IRQEN 寄存器字段说明 1080
表 12-139. IRQCLR 寄存器字段说明 1081
表 12-140. IRQSET 寄存器字段说明 1082
表 12-141. IRQSTAT 寄存器字段说明 1083
表 12-142. HWVER 寄存器字段说明 1084
表 13-1. PA 和 LNA 的 RF 内核数据信号 1087
表 13-2. CC13x2 和 CC26x2 引脚映射 1090
表 13-3. CC13x2 和 CC26x2 PORTID 1092
表 13-4. AON_IOC 寄存器 1095
表 13-5. AON_IOC 访问类型代码 1095
表 13-6. IOSTRMIN 寄存器字段说明 1096
表 13-7. IOSTRMED 寄存器字段说明 1097
表 13-8. IOSTRMAX 寄存器字段说明 1098
表 13-9. CLK32KCTL 寄存器字段说明 1099
表 13-10. TCKCTL 寄存器字段说明 1100
表 13-11. GPIO 寄存器 1101
表 13-12. GPIO 访问类型代码 1101
表 13-13. DOUT3_0 寄存器字段说明 1102
表 13-14. DOUT7_4 寄存器字段说明 1103
表 13-15. DOUT11_8 寄存器字段说明 1104
表 13-16. DOUT15_12 寄存器字段说明 1105
表 13-17. DOUT19_16 寄存器字段说明 1106
表 13-18. DOUT23_20 寄存器字段说明 1107
表 13-19. DOUT27_24 寄存器字段说明 1108
表 13-20. DOUT31_28 寄存器字段说明 1109
表 13-21. DOUT31_0 寄存器字段说明 1110
表 13-22. DOUTSET31_0 寄存器字段说明 1112
表 13-23. DOUTCLR31_0 寄存器字段说明 1114
表 13-24. DOUTTGL31_0 寄存器字段说明 1116
表 13-25. DIN31_0 寄存器字段说明 1118
表 13-26. DOE31_0 寄存器字段说明 1120

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-27. EVFLAGS31_0 寄存器字段说明.....1122
表 13-28. IOC 寄存器.....1124
表 13-29. IOC 访问类型代码.....1125
表 13-30. IOCFG0 寄存器字段说明.....1126
表 13-31. IOCFG1 寄存器字段说明.....1131
表 13-32. IOCFG2 寄存器字段说明.....1136
表 13-33. IOCFG3 寄存器字段说明.....1141
表 13-34. IOCFG4 寄存器字段说明.....1146
表 13-35. IOCFG5 寄存器字段说明.....1151
表 13-36. IOCFG6 寄存器字段说明.....1156
表 13-37. IOCFG7 寄存器字段说明.....1161
表 13-38. IOCFG8 寄存器字段说明.....1166
表 13-39. IOCFG9 寄存器字段说明.....1171
表 13-40. IOCFG10 寄存器字段说明.....1176
表 13-41. IOCFG11 寄存器字段说明.....1181
表 13-42. IOCFG12 寄存器字段说明.....1186
表 13-43. IOCFG13 寄存器字段说明.....1191
表 13-44. IOCFG14 寄存器字段说明.....1196
表 13-45. IOCFG15 寄存器字段说明.....1201
表 13-46. IOCFG16 寄存器字段说明.....1206
表 13-47. IOCFG17 寄存器字段说明.....1211
表 13-48. IOCFG18 寄存器字段说明.....1216
表 13-49. IOCFG19 寄存器字段说明.....1221
表 13-50. IOCFG20 寄存器字段说明.....1226
表 13-51. IOCFG21 寄存器字段说明.....1231
表 13-52. IOCFG22 寄存器字段说明.....1236
表 13-53. IOCFG23 寄存器字段说明.....1241
表 13-54. IOCFG24 寄存器字段说明.....1246
表 13-55. IOCFG25 寄存器字段说明.....1251
表 13-56. IOCFG26 寄存器字段说明.....1256
表 13-57. IOCFG27 寄存器字段说明.....1261
表 13-58. IOCFG28 寄存器字段说明.....1266
表 13-59. IOCFG29 寄存器字段说明.....1271
表 13-60. IOCFG30 寄存器字段说明.....1276
表 13-61. IOCFG31 寄存器字段说明.....1281
表 14-1. 通道分配.....1289
表 14-2. 请求类型支持.....1290
表 14-3. 控制结构内存映射.....1291
表 14-4. 通道控制结构.....1291
表 14-5. µDMA 读取示例：8 位外设.....1301
表 14-6. µDMA 中断分配.....1302
表 14-7. UDMA 寄存器.....1305
表 14-8. UDMA 访问类型代码.....1305
表 14-9. 状态寄存器字段说明.....1307
表 14-10. CFG 寄存器字段说明.....1309
表 14-11. CTRL 寄存器字段说明.....1310
表 14-12. ALTCTRL 寄存器字段说明.....1311
表 14-13. WAITONREQ 寄存器字段说明.....1312
表 14-14. SOFTREQ 寄存器字段说明.....1313
表 14-15. SETBURST 寄存器字段说明.....1314
表 14-16. CLEARBURST 寄存器字段说明.....1315
表 14-17. SETREQMASK 寄存器字段说明.....1316
表 14-18. CLEARREQMASK 寄存器字段说明.....1317
表 14-19. SETCHANNELEN 寄存器字段说明.....1318
表 14-20. CLEARCHANNELEN 寄存器字段说明.....1319
表 14-21. SETCHNLPRIALT 寄存器字段说明.....1320

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 14-22. CLEARCHNLPRIALT 寄存器字段说明.....1321
表 14-23. SETCHNLPRIORITY 寄存器字段说明.....1322
表 14-24. CLEARCHNLPRIORITY 寄存器字段说明.....1323
表 14-25. ERROR 寄存器字段说明.....1324
表 14-26. REQDONE 寄存器字段说明.....1325
表 14-27. DONEMASK 寄存器字段说明.....1326
表 15-1. 通用计时器功能.....1330
表 15-2. 带预分频器的 16 位计时器配置.....1332
表 15-3. 在输入边沿计数模式下计时器使能时的计数器值.....1332
表 15-4. 在输入事件计数模式下计时器使能时的计数器值.....1333
表 15-5. PWM 模式下定时器使能时的计数器值.....1335
表 15-6. GPTM 模式的超时操作.....1338
表 15-7. GPT 寄存器.....1343
表 15-8. GPT 访问类型代码.....1344
表 15-9. CFG 寄存器字段说明.....1345
表 15-10. TAMR 寄存器字段说明.....1346
表 15-11. TBMR 寄存器字段说明.....1348
表 15-12. CTL 寄存器字段说明.....1350
表 15-13. SYNC 寄存器字段说明.....1352
表 15-14. IMR 寄存器字段说明.....1353
表 15-15. RIS 寄存器字段说明.....1355
表 15-16. MIS 寄存器字段说明.....1357
表 15-17. ICLR 寄存器字段说明.....1358
表 15-18. TAILR 寄存器字段说明.....1359
表 15-19. TBILR 寄存器字段说明.....1360
表 15-20. TAMATCHR 寄存器字段说明.....1361
表 15-21. TBMATCHR 寄存器字段说明.....1362
表 15-22. TAPR 寄存器字段说明.....1363
表 15-23. TBPR 寄存器字段说明.....1364
表 15-24. TAPMR 寄存器字段说明.....1365
表 15-25. TBPMR 寄存器字段说明.....1366
表 15-26. TAR 寄存器字段说明.....1367
表 15-27. TBR 寄存器字段说明.....1368
表 15-28. TAV 寄存器字段说明.....1369
表 15-29. TBV 寄存器字段说明.....1370
表 15-30. TAPS 寄存器字段说明.....1371
表 15-31. TBPS 寄存器字段说明.....1372
表 15-32. TAPV 寄存器字段说明.....1373
表 15-33. TBPV 寄存器字段说明.....1374
表 15-34. DMAEV 寄存器字段说明.....1375
表 15-35. 版本寄存器字段说明.....1376
表 15-36. ANDCCP 寄存器字段说明.....1377
表 16-1. AON_RTC 寄存器.....1383
表 16-2. AON_RTC 访问类型代码.....1383
表 16-3. CTL 寄存器字段说明.....1384
表 16-4. EVFLAGS 寄存器字段说明.....1386
表 16-5. SEC 寄存器字段说明.....1387
表 16-6. SUBSEC 寄存器字段说明.....1388
表 16-7. SUBSECINC 寄存器字段说明.....1389
表 16-8. CHCTL 寄存器字段说明.....1390
表 16-9. CH0CMP 寄存器字段说明.....1391
表 16-10. CH1CMP 寄存器字段说明.....1392
表 16-11. CH2CMP 寄存器字段说明.....1393
表 16-12. CH2CMPINC 寄存器字段说明.....1394
表 16-13. CH1CAPT 寄存器字段说明.....1395
表 16-14. SYNC 寄存器字段说明.....1396

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 16-15. TIME 寄存器字段说明 1397
表 16-16. SYNCLF 寄存器字段说明 1398
表 17-1. WDT 寄存器 1402
表 17-2. WDT 访问类型代码 1402
表 17-3. LOAD 寄存器字段说明 1403
表 17-4. VALUE 寄存器字段说明 1404
表 17-5. CTL 寄存器字段说明 1405
表 17-6. ICR 寄存器字段说明 1406
表 17-7. RIS 寄存器字段说明 1407
表 17-8. MIS 寄存器字段说明 1408
表 17-9. TEST 寄存器字段说明 1409
表 17-10. INT_CAUS 寄存器字段说明 1410
表 17-11. LOCK 寄存器字段说明 1411
表 18-1. 事件 1414
表 18-2. 周边模块的初始化 1417
表 18-3. TRNG 初始化序列 1417
表 18-4. TRNG 中断模式 1419
表 18-5. TRNG 寄存器 1420
表 18-6. TRNG 访问类型代码 1421
表 18-7. OUT0 寄存器字段说明 1422
表 18-8. OUT1 寄存器字段说明 1423
表 18-9. IRQFLAGSTAT 寄存器字段说明 1424
表 18-10. IRQFLAGMASK 寄存器字段说明 1425
表 18-11. IRQFLAGCLR 寄存器字段说明 1426
表 18-12. CTL 寄存器字段说明 1427
表 18-13. CFG0 寄存器字段说明 1428
表 18-14. ALARMCNT 寄存器字段说明 1429
表 18-15. FROEN 寄存器字段说明 1430
表 18-16. FRODETUNE 寄存器字段说明 1431
表 18-17. ALARMMASK 寄存器字段说明 1432
表 18-18. ALARMSTOP 寄存器字段说明 1433
表 18-19. LFSR0 寄存器字段说明 1434
表 18-20. LFSR1 寄存器字段说明 1435
表 18-21. LFSR2 寄存器字段说明 1436
表 18-22. HWOPT 寄存器字段说明 1437
表 18-23. HWVER0 寄存器字段说明 1438
表 18-24. IRQSTATMASK 寄存器字段说明 1439
表 18-25. HWVER1 寄存器字段说明 1440
表 18-26. IRQSET 寄存器字段说明 1441
表 18-27. SWRESET 寄存器字段说明 1442
表 18-28. IRQSTAT 寄存器字段说明 1443
表 19-1. AUX 工作模式 1448
表 19-2. AUX 操作时钟速率 1449
表 19-3. 任务测试和任务调试 1453
表 19-4. 存储器字访问 1456
表 19-5. I/O 字访问 1456
表 19-6. I/O 位访问 1457
表 19-7. 寄存器访问 1457
表 19-8. 逻辑运算 1457
表 19-9. 算术运算 1458
表 19-10. 移位操作 1458
表 19-11. 程序流控制 1458
表 19-12. 程序流程条件 1459
表 19-13. 循环流控制 1459
表 19-14. 电源管理 1459
表 19-15. 其他 1460

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-16. SCE 事件接口 1460
表 19-17. 唤醒事件接口 1464
表 19-18. 唤醒矢量优先级 1464
表 19-19. 数字外设 1467
表 19-20. AUX I/O 控制 1468
表 19-21. 信标资源分配 1470
表 19-22. TDC 计数器时钟源 1477
表 19-23. TDC 基准时钟源 1477
表 19-24. 相位宽度时序要求 1479
表 19-25. 频率测量的时序要求 1479
表 19-26. 任意时间测量 1 1480
表 19-27. 任意时间测量 2 1481
表 19-28. 任意时间测量 3 1482
表 19-29. 任意时间测量 4 1483
表 19-30. 通道操作 1488
表 19-31. 模拟外设 1493
表 19-32. ADC 时钟源 1497
表 19-33. AUX 事件总线 1510
表 19-34. AUX 事件订阅者 1511
表 19-35. 异步事件检测 1511
表 19-36. 发送到 MCU 的事件 1513
表 19-37. 发送到 AON 的事件 1514
表 19-38. 别名寄存器映射 1515
表 19-39. AUX 域传感器控制器和外设寄存器 1520
表 19-40. ADI_4_AUX 寄存器 1521
表 19-41. ADI_4_AUX 访问类型代码 1521
表 19-42. MUX0 寄存器字段说明 1522
表 19-43. MUX1 寄存器字段说明 1523
表 19-44. MUX2 寄存器字段说明 1524
表 19-45. MUX3 寄存器字段说明 1525
表 19-46. ISRC 寄存器字段说明 1526
表 19-47. COMP 寄存器字段说明 1527
表 19-48. MUX4 寄存器字段说明 1528
表 19-49. ADC0 寄存器字段说明 1529
表 19-50. ADC1 寄存器字段说明 1530
表 19-51. ADCREF0 寄存器字段说明 1531
表 19-52. ADCREF1 寄存器字段说明 1532
表 19-53. LPMBIAS 寄存器字段说明 1533
表 19-54. AUX_AIODIO 寄存器 1534
表 19-55. AUX_AIODIO 访问类型代码 1534
表 19-56. IOMODE 寄存器字段说明 1536
表 19-57. GPIODIE 寄存器字段说明 1541
表 19-58. IOPOE 寄存器字段说明 1542
表 19-59. GPIODOUT 寄存器字段说明 1543
表 19-60. GPIODIN 寄存器字段说明 1544
表 19-61. GPIODOUTSET 寄存器字段说明 1545
表 19-62. GPIODOUTCLR 寄存器字段说明 1546
表 19-63. GPIODOUTTGL 寄存器字段说明 1547
表 19-64. IO0PSEL 寄存器字段说明 1548
表 19-65. IO1PSEL 寄存器字段说明 1549
表 19-66. IO2PSEL 寄存器字段说明 1550
表 19-67. IO3PSEL 寄存器字段说明 1551
表 19-68. IO4PSEL 寄存器字段说明 1552
表 19-69. IO5PSEL 寄存器字段说明 1553
表 19-70. IO6PSEL 寄存器字段说明 1554
表 19-71. IO7PSEL 寄存器字段说明 1555

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-72. IOMODEL 寄存器字段说明 1556
表 19-73. IOMODEH 寄存器字段说明 1557
表 19-74. AUX_EVCTL 寄存器 1558
表 19-75. AUX_EVCTL 访问类型代码 1559
表 19-76. EVSTAT0 寄存器字段说明 1560
表 19-77. EVSTAT1 寄存器字段说明 1562
表 19-78. EVSTAT2 寄存器字段说明 1564
表 19-79. EVSTAT3 寄存器字段说明 1566
表 19-80. SCEWEVCFG0 寄存器字段说明 1568
表 19-81. SCEWEVCFG1 寄存器字段说明 1571
表 19-82. DMACTL 寄存器字段说明 1574
表 19-83. SWEVSET 寄存器字段说明 1575
表 19-84. EVTOAONFLAGS 寄存器字段说明 1576
表 19-85. EVTOAONPOL 寄存器字段说明 1577
表 19-86. EVTOAONFLAGSCLR 寄存器字段说明 1578
表 19-87. EVTOMCUFLAGS 寄存器字段说明 1579
表 19-88. EVTOMCUPOL 寄存器字段说明 1581
表 19-89. EVTOMCUFLAGSCLR 寄存器字段说明 1583
表 19-90. COMBEVTOMCUMASK 寄存器字段说明 1585
表 19-91. EVOBSCFG 寄存器字段说明 1587
表 19-92. PROGDLY 寄存器字段说明 1590
表 19-93. MANUAL 寄存器字段说明 1591
表 19-94. EVSTAT0L 寄存器字段说明 1592
表 19-95. EVSTAT0H 寄存器字段说明 1593
表 19-96. EVSTAT1L 寄存器字段说明 1594
表 19-97. EVSTAT1H 寄存器字段说明 1595
表 19-98. EVSTAT2L 寄存器字段说明 1596
表 19-99. EVSTAT2H 寄存器字段说明 1597
表 19-100. EVSTAT3L 寄存器字段说明 1598
表 19-101. EVSTAT3H 寄存器字段说明 1599
表 19-102. AUX_SMPH 寄存器 1600
表 19-103. AUX_SMPH 访问类型代码 1600
表 19-104. SMPH0 寄存器字段说明 1601
表 19-105. SMPH1 寄存器字段说明 1602
表 19-106. SMPH2 寄存器字段说明 1603
表 19-107. SMPH3 寄存器字段说明 1604
表 19-108. SMPH4 寄存器字段说明 1605
表 19-109. SMPH5 寄存器字段说明 1606
表 19-110. SMPH6 寄存器字段说明 1607
表 19-111. SMPH7 寄存器字段说明 1608
表 19-112. AUTOTAKE 寄存器字段说明 1609
表 19-113. AUX_TDC 寄存器 1610
表 19-114. AUX_TDC 访问类型代码 1610
表 19-115. CTL 寄存器字段说明 1611
表 19-116. STAT 寄存器字段说明 1612
表 19-117. RESULT 寄存器字段说明 1614
表 19-118. SATCFG 寄存器字段说明 1615
表 19-119. TRIGSRC 寄存器字段说明 1616
表 19-120. TRIGCNT 寄存器字段说明 1621
表 19-121. TRIGCNTLOAD 寄存器字段说明 1622
表 19-122. TRIGCNTCFG 寄存器字段说明 1623
表 19-123. PRECTL 寄存器字段说明 1624
表 19-124. PRECNTR 寄存器字段说明 1627
表 19-125. AUX_TIMER01 寄存器 1628
表 19-126. AUX_TIMER01 访问类型代码 1628
表 19-127. T0CFG 寄存器字段说明 1629

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-128. T0CTL 寄存器字段说明.....1632
表 19-129. T0TARGET 寄存器字段说明.....1633
表 19-130. T0CNTR 寄存器字段说明.....1634
表 19-131. T1CFG 寄存器字段说明.....1635
表 19-132. T1CTL 寄存器字段说明.....1638
表 19-133. T1TARGET 寄存器字段说明.....1639
表 19-134. T1CNTR 寄存器字段说明.....1640
表 19-135. AUX_TIMER2 寄存器.....1641
表 19-136. AUX_TIMER2 访问类型代码.....1642
表 19-137. CTL 寄存器字段说明.....1643
表 19-138. TARGET 寄存器字段说明.....1645
表 19-139. SHDWTARGET 寄存器字段说明.....1646
表 19-140. CNTR 寄存器字段说明.....1647
表 19-141. PRECFG 寄存器字段说明.....1648
表 19-142. EVCTL 寄存器字段说明.....1649
表 19-143. PULSETRIG 寄存器字段说明.....1650
表 19-144. CH0EVCFG 寄存器字段说明.....1651
表 19-145. CH0CCFG 寄存器字段说明.....1654
表 19-146. CH0PCC 寄存器字段说明.....1657
表 19-147. CH0CC 寄存器字段说明.....1658
表 19-148. CH1EVCFG 寄存器字段说明.....1659
表 19-149. CH1CCFG 寄存器字段说明.....1662
表 19-150. CH1PCC 寄存器字段说明.....1665
表 19-151. CH1CC 寄存器字段说明.....1666
表 19-152. CH2EVCFG 寄存器字段说明.....1667
表 19-153. CH2CCFG 寄存器字段说明.....1670
表 19-154. CH2PCC 寄存器字段说明.....1673
表 19-155. CH2CC 寄存器字段说明.....1674
表 19-156. CH3EVCFG 寄存器字段说明.....1675
表 19-157. CH3CCFG 寄存器字段说明.....1678
表 19-158. CH3PCC 寄存器字段说明.....1681
表 19-159. CH3CC 寄存器字段说明.....1682
表 19-160. AUX_ANAIF 寄存器.....1683
表 19-161. AUX_ANAIF 访问类型代码.....1683
表 19-162. ADCCTL 寄存器字段说明.....1684
表 19-163. ADCFIFOSTAT 寄存器字段说明.....1687
表 19-164. ADCFIFO 寄存器字段说明.....1688
表 19-165. ADCTRIG 寄存器字段说明.....1689
表 19-166. ISRCCTL 寄存器字段说明.....1690
表 19-167. DACCTL 寄存器字段说明.....1691
表 19-168. LPMBIASCTL 寄存器字段说明.....1693
表 19-169. DACSMPLCTL 寄存器字段说明.....1694
表 19-170. DACSMPLCFG0 寄存器字段说明.....1695
表 19-171. DACSMPLCFG1 寄存器字段说明.....1696
表 19-172. DACVALUE 寄存器字段说明.....1697
表 19-173. DACSTAT 寄存器字段说明.....1698
表 19-174. AUX_SYSIF 寄存器.....1699
表 19-175. AUX_SYSIF 访问类型代码.....1700
表 19-176. OPMODEREQ 寄存器字段说明.....1701
表 19-177. OPMODEACK 寄存器字段说明.....1702
表 19-178. PROGWU0CFG 寄存器字段说明.....1703
表 19-179. PROGWU1CFG 寄存器字段说明.....1706
表 19-180. PROGWU2CFG 寄存器字段说明.....1709
表 19-181. PROGWU3CFG 寄存器字段说明.....1712
表 19-182. SWWUTRIG 寄存器字段说明.....1715
表 19-183. WUFLAGS 寄存器字段说明.....1716

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-184. WUFLAGSCLR 寄存器字段说明.....1717
表 19-185. WUGATE 寄存器字段说明.....1719
表 19-186. VECCFG0 寄存器字段说明.....1720
表 19-187. VECCFG1 寄存器字段说明.....1721
表 19-188. VECCFG2 寄存器字段说明.....1722
表 19-189. VECCFG3 寄存器字段说明.....1723
表 19-190. VECCFG4 寄存器字段说明.....1724
表 19-191. VECCFG5 寄存器字段说明.....1725
表 19-192. VECCFG6 寄存器字段说明.....1726
表 19-193. VECCFG7 寄存器字段说明.....1727
表 19-194. EVSYNCRATE 寄存器字段说明.....1728
表 19-195. PEROPRATE 寄存器字段说明.....1729
表 19-196. ADCCLKCTL 寄存器字段说明.....1730
表 19-197. TDCCLKCTL 寄存器字段说明.....1731
表 19-198. TDCREFCLKCTL 寄存器字段说明.....1732
表 19-199. TIMER2CLKCTL 寄存器字段说明.....1733
表 19-200. TIMER2CLKSTAT 寄存器字段说明.....1734
表 19-201. TIMER2CLKSWITCH 寄存器字段说明.....1735
表 19-202. TIMER2DBGCTL 寄存器字段说明.....1736
表 19-203. CLKSHIFTDET 寄存器字段说明.....1737
表 19-204. RECHARGETRIG 寄存器字段说明.....1738
表 19-205. RECHARGEDET 寄存器字段说明.....1739
表 19-206. RTCSUBSECINC0 寄存器字段说明.....1740
表 19-207. RTCSUBSECINC1 寄存器字段说明.....1741
表 19-208. RTCSUBSECINCCTL 寄存器字段说明.....1742
表 19-209. RTCSEC 寄存器字段说明.....1743
表 19-210. RTCSUBSEC 寄存器字段说明.....1744
表 19-211. RTCEVCLR 寄存器字段说明.....1745
表 19-212. BATMONBAT 寄存器字段说明.....1746
表 19-213. BATMONTEMP 寄存器字段说明.....1747
表 19-214. TIMERHALT 寄存器字段说明.....1748
表 19-215. TIMER2BRIDGE 寄存器字段说明.....1749
表 19-216. SWPWRPROF 寄存器字段说明.....1750
表 19-217. AUX_SPIM 寄存器.....1751
表 19-218. AUX_SPIM 访问类型代码.....1751
表 19-219. SPIMCFG 寄存器字段说明.....1752
表 19-220. MISOCFG 寄存器字段说明.....1753
表 19-221. MOSICTL 寄存器字段说明.....1754
表 19-222. TX8 寄存器字段说明.....1755
表 19-223. TX16 寄存器字段说明.....1756
表 19-224. RX8 寄存器字段说明.....1757
表 19-225. RX16 寄存器字段说明.....1758
表 19-226. SCLKIDLE 寄存器字段说明.....1759
表 19-227. DATAIDLE 寄存器字段说明.....1760
表 19-228. AUX_MAC 寄存器.....1761
表 19-229. AUX_MAC 访问类型代码.....1762
表 19-230. OP0S 寄存器字段说明.....1763
表 19-231. OP0U 寄存器字段说明.....1764
表 19-232. OP1SMUL 寄存器字段说明.....1765
表 19-233. OP1UMUL 寄存器字段说明.....1766
表 19-234. OP1SMAC 寄存器字段说明.....1767
表 19-235. OP1UMAC 寄存器字段说明.....1768
表 19-236. OP1SADD16 寄存器字段说明.....1769
表 19-237. OP1UADD16 寄存器字段说明.....1770
表 19-238. OP1SADD32 寄存器字段说明.....1771
表 19-239. OP1UADD32 寄存器字段说明.....1772

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-240. CLZ 寄存器字段说明 1773
表 19-241. CLS 寄存器字段说明 1774
表 19-242. ACCSHIFT 寄存器字段说明 1775
表 19-243. ACCRESET 寄存器字段说明 1776
表 19-244. ACC15_0 寄存器字段说明 1777
表 19-245. ACC16_1 寄存器字段说明 1778
表 19-246. ACC17_2 寄存器字段说明 1779
表 19-247. ACC18_3 寄存器字段说明 1780
表 19-248. ACC19_4 寄存器字段说明 1781
表 19-249. ACC20_5 寄存器字段说明 1782
表 19-250. ACC21_6 寄存器字段说明 1783
表 19-251. ACC22_7 寄存器字段说明 1784
表 19-252. ACC23_8 寄存器字段说明 1785
表 19-253. ACC24_9 寄存器字段说明 1786
表 19-254. ACC25_10 寄存器字段说明 1787
表 19-255. ACC26_11 寄存器字段说明 1788
表 19-256. ACC27_12 寄存器字段说明 1789
表 19-257. ACC28_13 寄存器字段说明 1790
表 19-258. ACC29_14 寄存器字段说明 1791
表 19-259. ACC30_15 寄存器字段说明 1792
表 19-260. ACC31_16 寄存器字段说明 1793
表 19-261. ACC32_17 寄存器字段说明 1794
表 19-262. ACC33_18 寄存器字段说明 1795
表 19-263. ACC34_19 寄存器字段说明 1796
表 19-264. ACC35_20 寄存器字段说明 1797
表 19-265. ACC36_21 寄存器字段说明 1798
表 19-266. ACC37_22 寄存器字段说明 1799
表 19-267. ACC38_23 寄存器字段说明 1800
表 19-268. ACC39_24 寄存器字段说明 1801
表 19-269. ACC39_32 寄存器字段说明 1802
表 19-270. AUX_SCE 寄存器 1803
表 19-271. AUX_SCE 访问类型代码 1803
表 19-272. CTL 寄存器字段说明 1804
表 19-273. FETCHSTAT 寄存器字段说明 1805
表 19-274. CPUSTAT 寄存器字段说明 1806
表 19-275. WUSTAT 寄存器字段说明 1807
表 19-276. REG1_0 寄存器字段说明 1808
表 19-277. REG3_2 寄存器字段说明 1809
表 19-278. REG5_4 寄存器字段说明 1810
表 19-279. REG7_6 寄存器字段说明 1811
表 19-280. LOOPADDR 寄存器字段说明 1812
表 19-281. LOOPCNT 寄存器字段说明 1813
表 20-1. AON_BATMON 寄存器 1816
表 20-2. AON_BATMON 访问类型代码 1816
表 20-3. CTL 寄存器字段说明 1818
表 20-4. MEASCFG 寄存器字段说明 1819
表 20-5. TEMPP0 寄存器字段说明 1820
表 20-6. TEMPP1 寄存器字段说明 1821
表 20-7. TEMPP2 寄存器字段说明 1822
表 20-8. BATMONP0 寄存器字段说明 1823
表 20-9. BATMONP1 寄存器字段说明 1824
表 20-10. IOSTRP0 寄存器字段说明 1825
表 20-11. FLASHPUMPP0 寄存器字段说明 1826
表 20-12. BAT 寄存器字段说明 1827
表 20-13. BATUPD 寄存器字段说明 1828
表 20-14. TEMP 寄存器字段说明 1829

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 20-15. TEMPUPD 寄存器字段说明 1830
表 20-16. EVENTMASK 寄存器字段说明 1831
表 20-17. EVENT 寄存器字段说明 1832
表 20-18. BATTUL 寄存器字段说明 1833
表 20-19. BATTLL 寄存器字段说明 1834
表 20-20. TEMPUL 寄存器字段说明 1835
表 20-21. TEMPLL 寄存器字段说明 1836
表 21-1. UART 信号 1839
表 21-2. 流控制模式 1841
表 21-3. UART 寄存器 1847
表 21-4. UART 访问类型代码 1847
表 21-5. DR 寄存器字段说明 1848
表 21-6. RSR 寄存器字段说明 1850
表 21-7. ECR 寄存器字段说明 1851
表 21-8. FR 寄存器字段说明 1852
表 21-9. IBRD 寄存器字段说明 1853
表 21-10. FBRD 寄存器字段说明 1854
表 21-11. LCRH 寄存器字段说明 1855
表 21-12. CTL 寄存器字段说明 1857
表 21-13. IFLS 寄存器字段说明 1859
表 21-14. IMSC 寄存器字段说明 1860
表 21-15. RIS 寄存器字段说明 1862
表 21-16. MIS 寄存器字段说明 1864
表 21-17. ICR 寄存器字段说明 1866
表 21-18. DMACTL 寄存器字段说明 1867
表 22-1. SSI 信号 1871
表 22-2. SSI 寄存器 1882
表 22-3. SSI 访问类型代码 1882
表 22-4. CR0 寄存器字段说明 1883
表 22-5. CR1 寄存器字段说明 1884
表 22-6. DR 寄存器字段说明 1885
表 22-7. SR 寄存器字段说明 1886
表 22-8. CPSR 寄存器字段说明 1887
表 22-9. IMSC 寄存器字段说明 1888
表 22-10. RIS 寄存器字段说明 1889
表 22-11. MIS 寄存器字段说明 1890
表 22-12. ICR 寄存器字段说明 1891
表 22-13. DMACR 寄存器字段说明 1892
表 23-1. I^2C 主计时器周期和速度模式的示例 1897
表 23-2. I2C 寄存器 1907
表 23-3. I2C 访问类型代码 1907
表 23-4. SOAR 寄存器字段说明 1909
表 23-5. SSTAT 寄存器字段说明 1910
表 23-6. SCTL 寄存器字段说明 1911
表 23-7. SDR 寄存器字段说明 1912
表 23-8. SIMR 寄存器字段说明 1913
表 23-9. SRIS 寄存器字段说明 1914
表 23-10. SMIS 寄存器字段说明 1915
表 23-11. SICR 寄存器字段说明 1916
表 23-12. MSA 寄存器字段说明 1917
表 23-13. MSTAT 寄存器字段说明 1918
表 23-14. MCTRL 寄存器字段说明 1919
表 23-15. MDR 寄存器字段说明 1921
表 23-16. MTPR 寄存器字段说明 1922
表 23-17. MIMR 寄存器字段说明 1923
表 23-18. MRIS 寄存器字段说明 1924

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 23-19. MMIS 寄存器字段说明.....1925
表 23-20. MICR 寄存器字段说明.....1926
表 23-21. MCR 寄存器字段说明.....1927
表 24-1. 串行音频引脚接口.....1931
表 24-2. I2S 寄存器.....1944
表 24-3. I2S 访问类型代码.....1945
表 24-4. AIFWCLKSRC 寄存器字段说明.....1946
表 24-5. AIFDMACFG 寄存器字段说明.....1947
表 24-6. AIFDIRCFG 寄存器字段说明.....1948
表 24-7. AIFFMTCFG 寄存器字段说明.....1949
表 24-8. AIFWMASK0 寄存器字段说明.....1950
表 24-9. AIFWMASK1 寄存器字段说明.....1951
表 24-10. AIFPWMVALUE 寄存器字段说明.....1952
表 24-11. AIFINPTRNEXT 寄存器字段说明.....1953
表 24-12. AIFINPTR 寄存器字段说明.....1954
表 24-13. AIFOUTPTRNEXT 寄存器字段说明.....1955
表 24-14. AIFOUTPTR 寄存器字段说明.....1956
表 24-15. STMPCTL 寄存器字段说明.....1957
表 24-16. STMPXCNTCAPT0 寄存器字段说明.....1958
表 24-17. STMPXPER 寄存器字段说明.....1959
表 24-18. STMPWCNTCAPT0 寄存器字段说明.....1960
表 24-19. STMPWPER 寄存器字段说明.....1961
表 24-20. STMPINTRIG 寄存器字段说明.....1962
表 24-21. STMPOUTTRIG 寄存器字段说明.....1963
表 24-22. STMPWSET 寄存器字段说明.....1964
表 24-23. STMPWADD 寄存器字段说明.....1965
表 24-24. STMPXPERMIN 寄存器字段说明.....1966
表 24-25. STMPWCNT 寄存器字段说明.....1967
表 24-26. STMPXCNT 寄存器字段说明.....1968
表 24-27. STMPXCNTCAPT1 寄存器字段说明.....1969
表 24-28. STMPWCNTCAPT1 寄存器字段说明.....1970
表 24-29. IRQMASK 寄存器字段说明.....1971
表 24-30. IRQFLAGS 寄存器字段说明.....1972
表 24-31. IRQSET 寄存器字段说明.....1973
表 24-32. IRQCLR 寄存器字段说明.....1974
表 25-1. 用于在引导过程中启用选定特性的 RFCBITS 格式.....1978
表 25-2. CMDSTA 寄存器中结果字节的值.....1983
表 25-3. 常见无线电操作状态代码.....1984
表 25-4. 触发条件定义字节的格式.....1987
表 25-5. 支持的触发条件类型.....1987
表 25-6. 外部事件触发条件的时间参数字段.....1987
表 25-7. 条件字节的格式.....1988
表 25-8. 条件规则.....1989
表 25-9. 无线电操作命令格式.....1990
表 25-10. 数据条目队列结构.....1990
表 25-11. 一般数据条目结构.....1991
表 25-12. 指针入口结构中的指针字段.....1992
表 25-13. 部分读取 RX 条目中的字段.....1992
表 25-14. 无线电操作结束命令.....1994
表 25-15. CMD_RADIO_SETUP 命令格式.....1995
表 25-16. 硬件寄存器覆盖条目的格式.....1996
表 25-17. 数组启动器的格式.....1996
表 25-18. ADI 寄存器覆盖条目的格式.....1997
表 25-19. 固件定义的参数覆盖条目的格式.....1997
表 25-20. MCE/RFE 覆盖模式条目的格式.....1997
表 25-21. 中心频率条目的格式.....1998

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-22. 20dBm PA TX 功率进入 1998
表 25-23. 列表结束条目的格式 1998
表 25-24. CMD_FS_POWERUP 命令格式 2000
表 25-25. CMD_FS 命令格式 2001
表 25-26. CMD_RX_TEST 命令格式 2002
表 25-27. CMD_TX_TEST 命令格式 2003
表 25-28. CMD_SYNC_STOP_RAT 命令格式 2004
表 25-29. CMD_SYNC_START_RAT 命令格式 2004
表 25-30. CMD_COUNT 命令格式 2005
表 25-31. CMD_COUNT 的其他结束原因 2005
表 25-32. CMD_SCH_IMM 命令格式 2005
表 25-33. CMD_SCH_IMM 的结束状态 2006
表 25-34. CMD_COUNT_BRANCH 命令格式 2006
表 25-35. CMD_COUNT_BRANCH 的其他结束原因 2006
表 25-36. CMD_PATTERN_CHECK 命令格式 2007
表 25-37. CMD_PATTERN_CHECK 的其他结束原因 2008
表 25-38. CMD_UPDATE_RADIO_SETUP 命令格式 2009
表 25-39. CMD_TRIGGER 命令格式 2009
表 25-40. CMD_GET_FW_INFO 命令格式 2010
表 25-41. CMD_READ_RFREG 命令格式 2011
表 25-42. CMD_SET_RAT_CMP 命令格式 2011
表 25-43. CMD_SET_RAT_CPT 命令格式 2012
表 25-44. CMD_DISABLE_RAT_CH 命令格式 2012
表 25-45. CMD_SET_RAT_OUTPUT 命令格式 2013
表 25-46. CMD_ARM_RAT_CH 命令格式 2013
表 25-47. CMD_DISARM_RAT_CH 命令格式 2014
表 25-48. CMD_SET_TX_POWER 命令格式 2014
表 25-49. CMD_SET_TX20_POWER 命令格式 2015
表 25-50. CMD_UPDATE_FS 命令格式 2015
表 25-51. CMD_MODIFY_FS 命令格式 2016
表 25-52. CMD_BUS_REQUEST 命令格式 2016
表 25-53. CMD_ADD_DATA_ENTRY 命令格式 2017
表 25-54. CMD_REMOVE_DATA_ENTRY 命令格式 2017
表 25-55. CMD_FLUSH_QUEUE 命令格式 2018
表 25-56. CMD_CLEAR_RX 命令格式 2018
表 25-57. CMD_REMOVE_PENDING_ENTRIES 命令格式 2019
表 25-58. 后台级 IEEE 802.15.4 无线电操作命令 2024
表 25-59. 前台级 IEEE 802.15.4 无线电操作命令 2024
表 25-60. 前台级别的常见无线电操作命令 2024
表 25-61. IEEE 802.15.4 立即命令 2024
表 25-62. IEEE 802.15.4 RX 命令结构 2025
表 25-63. IEEE 802.15.4 能量检测扫描命令结构 2025
表 25-64. IEEE 802.15.4 CSMA-CA 命令结构 2026
表 25-65. IEEE 802.15.4 TX 命令结构 2026
表 25-66. IEEE 802.15.4 接收 ACK 命令结构 2026
表 25-67. IEEE 802.15.4 修改 CCA 即时命令结构 2027
表 25-68. IEEE 802.15.4 修改帧过滤即时命令结构 2027
表 25-69. IEEE 802.15.4 启用或禁用源匹配条目即时命令结构 2027
表 25-70. IEEE 802.15.4 请求 CCA 状态即时命令结构 2028
表 25-71. RX 命令 2028
表 25-72. 接收队列条目配置位字段 2029
表 25-73. CCA 配置位字段 2029
表 25-74. 帧过滤配置位字段 2030
表 25-75. 帧类型过滤位字段 2030
表 25-76. 短地址条目结构 2032

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-77. 扩展地址列表结构 2032
表 25-78. 短地址列表结构 2032
表 25-79. 接收相关值/CRC 结果位字段 2032
表 25-80. 适用于 IEEE 802.15.4 的中断定义 2033
表 25-81. 前台和后台级别操作的允许组合 2034
表 25-82. IEEE 802.15.4 无线电操作状态代码 2035
表 25-83. RX 操作中递增计数器和引发中断的条件 2037
表 25-84. 接收操作结束 2039
表 25-85. CSMA-CA 操作结束 2044
表 25-86. 发送操作结束 2045
表 25-87. 接收 ACK 操作结束 2045
表 25-88. 后台级操作以 ABORT 结果结束 2046
表 25-89. 传统低功耗 Bluetooth® 无线电操作命令 2048
表 25-90. 低功耗 Bluetooth® 5 无线电操作命令 2049
表 25-91. 低功耗 Bluetooth® 即时命令 2049
表 25-92. 传统低功耗 Bluetooth® 无线电操作命令结构 (1) 2049
表 25-93. 低功耗 Bluetooth® 5 无线电操作命令结构 (1) 2050
表 25-94. 低功耗 Bluetooth® 5 无线电设置命令结构 (1) 2051
表 25-95. 更新广播有效载荷命令 (1) 2051
表 25-96. 传统从器件命令 (1) 2052
表 25-97. 旧主器件命令 (1) 2052
表 25-98. 旧广播器命令 (1) 2053
表 25-99. 传统扫描仪命令 (1) 2054
表 25-100. 传统启动器命令 (1) 2055
表 25-101. 通用 RX 命令 (1) 2055
表 25-102. TX 测试命令 (1) 2056
表 25-103. Bluetooth®5 从器件命令 (1) 2057
表 25-104. Bluetooth® 5 主命令 (1) 2057
表 25-105. 扩展的广播器命令 (1) 2058
表 25-106. 辅助通道广播器命令 (1) 2058
表 25-107. Bluetooth®5 扫描仪命令 (1) 2059
表 25-108. Bluetooth®5 启动器命令 (1) 2060
表 25-109. 主从命令 2062
表 25-110. 广播器命令 2062
表 25-111. 传统扫描器命令 2063
表 25-112. 传统发起器命令 2063
表 25-113. 低功耗 Bluetooth® 5 扫描器和发起器命令 2063
表 25-114. 通用 RX 命令 2064
表 25-115. 测试 TX 命令 2064
表 25-116. 接收队列条目配置位字段 (1) 2065
表 25-117. 序列号状态位字段 (1) 2065
表 25-118. 白名单结构 2065
表 25-119. 广播数据 ID 条目结构 2065
表 25-120. 旧命令的接收状态字节位字段 (1) 2066
表 25-121. 低功耗 Bluetooth®5 命令的接收状态字位字段 (1) 2066
表 25-122. 主器件和从器件数据包状态字节 2066
表 25-123. 通用扩展数据包条目格式 (1) 2067
表 25-124. 适用于低功耗 Bluetooth® 的中断定义 2067
表 25-125. 低功耗 Bluetooth® 无线电操作状态代码 2070
表 25-126. 主器件和从器件命令的编码选择 2071
表 25-127. 广播器、扫描仪和启动器命令的编码选择 2071
表 25-128. 对收到的数据包执行的操作 2072
表 25-129. 主器件和从器件命令中递增计数器和引发中断的条件 2074
表 25-130. 从器件运行结束 2077
表 25-131. 主器件操作结束 2078

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-132. 不同广播器命令的 PDU 类型 2079
表 25-133. 广播器根据收到的数据包采取的操作 2079
表 25-134. 要对收到的数据包执行的操作说明 2080
表 25-135. 可连接的非定向广播器操作结束 2081
表 25-136. 定向广播器操作结束 2082
表 25-137. 不可连接的广播器操作结束 2083
表 25-138. 可扫描非定向广播器操作结束 2083
表 25-139. 扩展广播器操作结束 2086
表 25-140. 可扫描广播器根据收到的数据包采取的操作 (extHdrInfo.advMode = 2) (1) 2087
表 25-141. 可连接广播器根据收到的数据包采取的操作 (extHdrInfo.advMode = 1) (1) 2087
表 25-142. 要对收到的数据包执行的操作说明 2087
表 25-143. 辅助通道广播器操作结束 2088
表 25-144. 过滤收到的广播器地址 2089
表 25-145. 过滤 ADV_DIRECT_IND 数据包中收到的发起器地址 2089
表 25-146. 扫描器对收到的数据包执行的操作 2089
表 25-147. 扫描器对收到的数据包执行的操作说明 2090
表 25-148. 扫描器在发送 SCAN_REQ 后对收到的数据包的操作 2090
表 25-149. 由扫描器对收到的扩展广播器数据包执行的操作 2091
表 25-150. 对扫描器接收到的扩展广播器数据包执行的操作说明 2091
表 25-151. 扫描器对收到的次级通道数据包执行的操作 2092
表 25-152. 对扫描器接收到的次级通道数据包执行的操作的说明 2093
表 25-153. 扫描器操作结束 2095
表 25-154. 筛选接收到的广播器地址 2096
表 25-155. 由启动器对收到的数据包执行的操作 2096
表 25-156. 对启动器接收到的数据包执行的操作说明 2096
表 25-157. 由启动器对收到的扩展广播器数据包执行的操作 2097
表 25-158. 对启动器接收到的扩展广播器数据包执行的操作说明 2097
表 25-159. 启动器对接收到的次级通道数据包执行的操作 2098
表 25-160. 对启动器接收到的次级通道数据包执行的操作说明 2098
表 25-161. 启动器对接收到的 AUX_CONNECT_RSP 数据包执行的操作 2099
表 25-162. 对启动器接收到的次级通道数据包执行的操作说明 2099
表 25-163. 启动器操作结束 2100
表 25-164. 通用 RX 操作结束 2102
表 25-165. 支持的 PHY 测试数据包类型 2103
表 25-166. PHY 测试 TX 操作结束 2103
表 25-167. 更新退避参数 2104
表 25-168. 专有无线电操作命令 2107
表 25-169. 专有立即数命令 2107
表 25-170. CMD_PROP_TX 命令结构 2107
表 25-171. CMD_PROP_TX_ADV 命令结构 2108
表 25-172. CMD_PROP_RX 和 CMD_PROP_RX_SNIFF 命令结构 2108
表 25-173. CMD_PROP_RX_ADV 和 CMD_PROP_RX_ADV_SNIFF 命令结构 2109
表 25-174. CMD_PROP_CS 命令结构 2110
表 25-175. CMD_PROP_RADIO_SETUP 和 CMD_PROP_RADIO_DIV_SETUP 命令结构 2110
表 25-176. CMD_PROP_SET_LEN 命令结构 2111
表 25-177. 接收命令 2111
表 25-178. CMD_PROP_RX_SNIFF 、CMD_PROP_RX_ADV_SNIFF 和 CMD_PROP_CS 的载波侦听字段 2113
表 25-179. 接收队列条目配置位字段 (1) 2113
表 25-180. 接收状态字节位字段 (1) 2113
表 25-181. 中断定义 2114
表 25-182. 专有无线电操作状态代码 2116
表 25-183. 接收器带宽设置 2117

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-184. 无线电 CMD_PROP_TX 和 CMD_PROP_TX_ADV 命令的结束状态 2119
表 25-185. 收到的数据包的中断、计数器和结果字段 (1) 2121
表 25-186. 无线电 CMD_PROP_RX 和 CMD_PROP_RX_ADV 命令结束 2122
表 25-187. 启用两个源时的通道状态 2124
表 25-188. CMD_PROP_CS 命令结束 2125
表 25-189. CMD_PROP_RX_SNIFF 和 CMD_PROP_RX_ADV_SNIFF 的其他结束状态 2126
表 25-190. RFC_RAT 寄存器 2127
表 25-191. RFC_RAT 访问类型代码 2127
表 25-192. RATCNT 寄存器字段说明 2128
表 25-193. RATCH0VAL 寄存器字段说明 2129
表 25-194. RATCH1VAL 寄存器字段说明 2130
表 25-195. RATCH2VAL 寄存器字段说明 2131
表 25-196. RATCH3VAL 寄存器字段说明 2132
表 25-197. RATCH4VAL 寄存器字段说明 2133
表 25-198. RATCH5VAL 寄存器字段说明 2134
表 25-199. RATCH6VAL 寄存器字段说明 2135
表 25-200. RATCH7VAL 寄存器字段说明 2136
表 25-201. RFC_DBELL 寄存器 2137
表 25-202. RFC_DBELL 访问类型代码 2137
表 25-203. CMDR 寄存器字段说明 2138
表 25-204. CMDSTA 寄存器字段说明 2139
表 25-205. RFHWIFG 寄存器字段说明 2140
表 25-206. RFHWIEN 寄存器字段说明 2142
表 25-207. RFCPEIFG 寄存器字段说明 2143
表 25-208. RFCPEIEN 寄存器字段说明 2145
表 25-209. RFCPEISL 寄存器字段说明 2147
表 25-210. RFACKIFG 寄存器字段说明 2150
表 25-211. SYSGPOCTL 寄存器字段说明 2151
表 25-212. RFC_PWR 寄存器 2153
表 25-213. RFC_PWR 访问类型代码 2153
表 25-214. PWMCLKEN 寄存器字段说明 2154

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

本技术参考手册介绍如何使用 CC13x2 和 CC26x2 SimpleLink™ 超低功耗无线微控制器 (MCU)。CC26x2 和 CC13x2 器件平台共享相同的 MCU 架构和大部分外设。CC26x2 器件中的无线电模块工作于 2.4GHz ISM 频段，而 CC13x2 器件中的无线电模块设计用于低于 1GHz 的频段。CC1352 器件是一款多频段无线 MCU，可同时在低于 1GHz 和 2.4GHz 频段工作。本文介绍的是整个器件平台，因此关于支持的模块和特性，请参阅各器件的单独数据表。

关于本手册

本文档按各个主要特性划分为相应的章节，每一章不仅说明每个模块的特性和功能，还阐述了用法。对于每个特性，都参考了相应操作系统驱动程序的说明文件。本文档不包含相应器件数据表中收集的器件或模块的性能特点。本手册适用于系统软件开发人员、硬件设计人员和应用程序开发人员。

器件

CC13x2 和 CC26x2 器件平台包含 2.4GHz (CC26x2 和 CC1352) 和
低于 1GHz (CC13x2) 射频，以及各种不同的存储器大小、外设和封装选项。所有器件均以处理应用层和协议栈的 Arm® Cortex®-M4F 系列处理器为中心，同时包含一个以 Arm® Cortex®-M0 处理器为中心的自主射频内核，负责所有底层射频控制和处理。提供网络处理器选项。

由于提供了各种不同的射频和 MCU 系统组合，使得这些器件系列非常适合几乎任何低功耗射频节点实现。

寄存器、字段和位调用

对调用应用的命名约定包括：

- 对于寄存器调用：<*模块名称*>.<*寄存器名称*>；例如： UART.UASR
- 对于位字段调用：
  - <*模块名称*>.<*寄存器名称*>*[End:Start]* <*字段名称*> *字段*；例如，UART.UASR[4:0] SPEED 位字段
  - <*字段名称*> *字段* <*模块名称*>.<*寄存器名称*>*[End:Start]*；例如，SPEED 位字段 UART.UASR[4:0]
- 对于位调用：
  - <*模块名称*>.<*寄存器名称*>*[pos]* <*位名称*> *位*；例如，UART.UASR[5] BIT_BY_CHAR 位
  - <*位名称*> *位* <*模块名称*>.<*寄存器名称*>*[pos]*；例如，BIT_BY_CHAR bit UART.UASR[5]

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

相关文档

以下相关文档可在 www.ti.com 上的 CC13x2 和 CC26x2 器件产品页面上找到：

1. CC1312：
   - CC1312R 数据表和勘误表（技术文档）
   - CC1312PSIP 数据表和勘误表（技术文档）
2. CC1352：
   - CC1352R 数据表和勘误表（技术文档）
   - CC1352P 数据表和勘误表（技术文档）
3. CC2642：
   - CC2642R 数据表和勘误表（技术文档）
4. CC2652：
   - CC2652R 数据表和勘误表（技术文档）
   - CC2652P 数据表和勘误表（技术文档）
   - CC2652RSIP 数据表和勘误表（技术文档）
   - CC2652PSIP 数据表和勘误表（技术文档）
5. *Cortex-M3/M4F 指令集技术用户手册*

备注

本列表所示文档截至出版日期为最新版本。请查看网站以获取其他文档、应用手册和白皮书。

其他相关文档如下：

6. The Institute of Electrical and Electronic Engineers, Inc.，*IEEE 标准测试接入端口和边界扫描架构*、*IEEE 标准 1149.1a 1993 和补充标准 1149.1b 1994*（请参见 IEEExplore.ieee.org）
7. The Institute of Electrical and Electronic Engineers, Inc.，*IEEE 1149.7 减少引脚和增强功能测试接入端口和边界扫描架构标准*（请参见 IEEExplore.ieee.org）
8. 美国国家标准与技术研究院，*NIST 特别刊物 800-38A，分组密码操作模式方法与技术建议*（请参见 NIST.gov）
9. 美国国家标准与技术研究院，*NIST 特别刊物 800-38D，分组密码操作模式建议：伽罗瓦/计数器模式 (GCM) 和 GMAC*（请参见 NIST.gov）
10. 美国国家标准与技术研究院，*FIPS 197，高级加密标准 (AES)*（请参见 NIST.gov）
11. Bluetooth SIG, Inc.，*蓝牙*规范 4.0、4.1、4.2 和 5.0 版（请参见 Bluetooth.com）
12. *Arm® v7-M 架构参考手册*（请参见 Arm.com）
13. *Arm® 调试接口 V5 架构规范*（请参见 Arm.com）

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

1 商标

SimpleLink™ and EnergyTrace™ are trademarks of Texas Instruments.
CoreSight™ is a trademark of Arm Limited (or its subsidiaries).
™ is a trademark of Motorola Trademark Holdings, LLC.
Arm®, Cortex®, Thumb®, Arm7®, AMBA®, and PrimeCell® are registered trademarks of Arm Limited (or its subsidiaries).
Zigbee® is a registered trademark of Zigbee Alliance.
所有商标均为其各自所有者的财产。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

章节 1

结构概述

SimpleLink™ 超低功耗无线 MCU 的 CC13x2 和 CC26x2 器件平台可以为各种应用提供解决方案。为帮助用户开发这些应用，本用户指南重点介绍了如何使用器件的不同构建块。有关详细的器件说明、完整的特性列表和性能数据，请参阅特定器件的数据表。以下各子部分提供了对相关信息的便捷访问，并指导读者阅读本文档中的不同章节。

器件平台片上系统 (SoC) 针对超低功耗进行了优化，同时提供快速且功能强大的 MCU 系统来实现较短的处理时间和高集成度。高达 48MHz 的 Arm® Cortex®-M4F 处理核心、闪存和多种外围设备的组合使 CC13x2 和 CC26x2 器件平台专为低功耗 RF 节点的单芯片实现或网络处理器实现而设计。

1.1 目标应用 69
1.2 概述 69
1.3 功能概述 72

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

1.1 目标应用

CC13x2 和 CC26x2 SimpleLink™ 超低功耗无线 MCU 平台专门用于低功耗无线应用，例如：

- 消费类电子产品
- 手机附件
- 运动和健身设备
- HID 应用
- 住宅和楼宇自动化
- 光控制
- 警报和安防
- 电子货架标签
- 接近标签
- 医疗
- 遥控
- 智能抄表
- 资产跟踪
- 无线传感器网络

1.2 概述

图 1-1 显示 CC13x2 和 CC26x2 器件平台的构建块。

CC13x2 和 CC26x2 器件平台具有以下特性：

- Arm® Cortex®-M4F 处理器内核
  - 48MHz RC 振荡器和 48MHz 晶体振荡器
  - 32kHz 晶体振荡器、32kHz RC 振荡器或低功耗 48MHz 晶体振荡器在低功耗模式下推算时钟以进行计时维护
  - Arm® Cortex® SysTick 计时器
  - 嵌套矢量中断控制器 (NVIC)
- 片上存储器
  - 352kB 闪存配 8kB 4 路组关联高速缓存 RAM，速度快、功耗低
  - 32kB 系统 RAM，可按 16kB 块配置其数据保持功能
- 电源管理
  - 宽电源电压范围
  - 用于降低功耗的高效片上 DC/DC 转换器
  - 器件零件的高粒度时钟门控和电源门控
  - 灵活的工作频率
    - 灵活的低功耗模式可在占空比应用中实现低能耗
- 传感器接口
  - 自主智能传感器接口，可以独立于系统 CPU 系统唤醒以执行传感器读取、收集数据并确定是否必须唤醒系统 CPU
  - 具有 8 个模拟输入通道的 12 位模数转换器 (ADC)
  - 低功耗模拟比较器
  - 串行通信接口

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

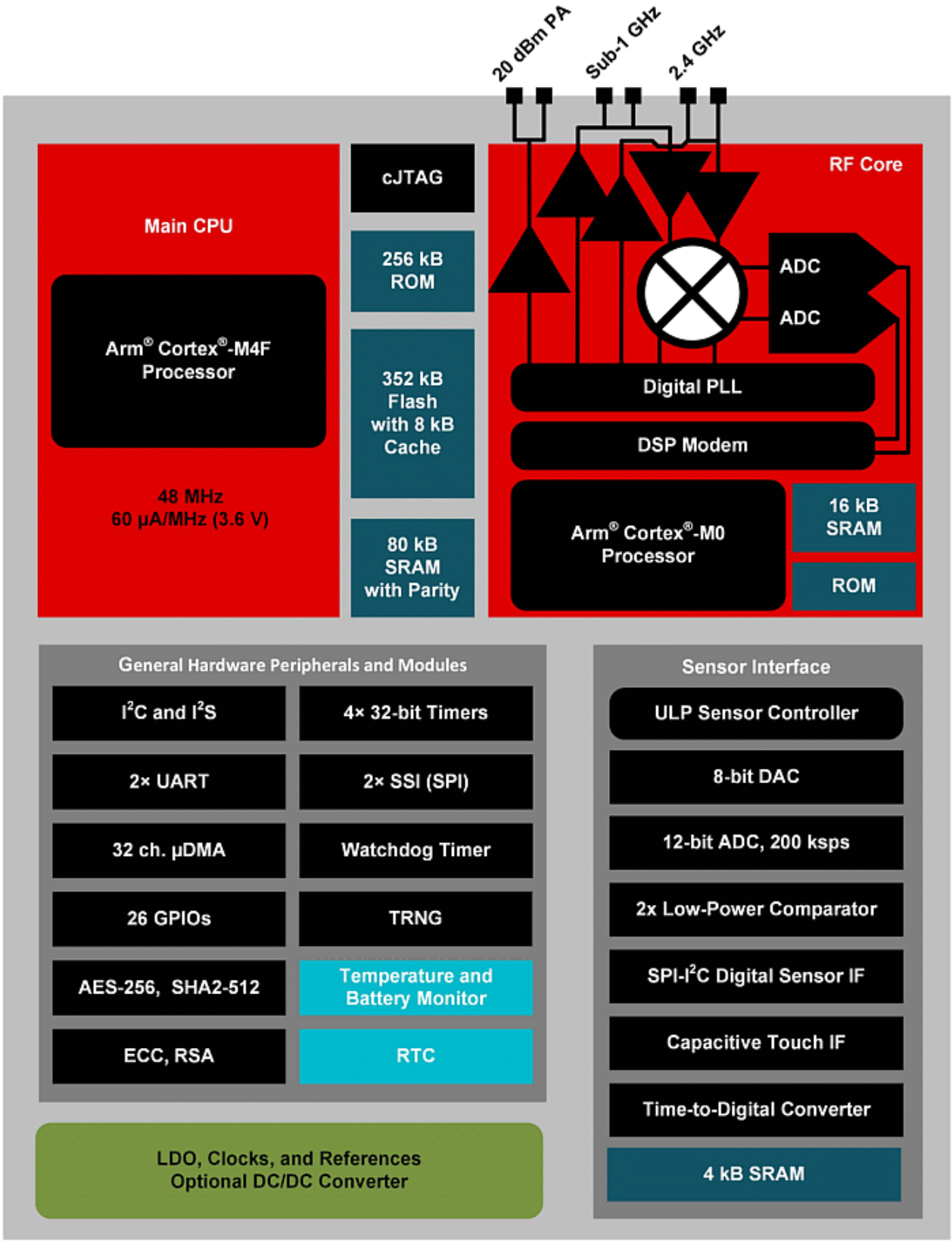


图 1-1. CC13x2 和 CC26x2 方框图

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

- 高级串行集成
  - 通用异步收发器 (UART)
  - 内部集成电路 (I^2C) 模块
  - 同步串行接口模块 (SSI)
  - 音频接口 I^2S 模块
- 系统集成
  - 直接存储器存取 (DMA) 控制器
  - 四个具有脉宽调制 (PWM) 功能和同步功能的 32 位计时器 (最多八个 16 位计时器)
  - 32kHz 实时时钟 (RTC)
  - 看门狗计时器
  - 片上温度和电源电压检测
  - 具备普通驱动或高驱动能力的 GPIO
  - 具备模拟功能的 GPIO，用于 ADC 和比较器
  - 完全灵活的数字引脚多路复用功能允许用作 GPIO 或任何外设功能
- 符合 IEEE 1149.7 标准的 2 引脚 cJTAG，支持传统 1149.1 JTAG
- 7mm × 7mm VQFN 和 7mm × 7mm MOT 封装

对于需要极端节能的应用，CC13x2 和 CC26x2 器件平台采用的电源管理系统可在长时间不活动期间高效地将器件调至低功耗状态。这款 MCU 配备加电和断电时序控制器、具有中断功能的 32 位睡眠计时器 (RTC) 和在所有电源模式下均具有数据保持功能的 80kB 超低泄漏 (ULL) RAM，非常适合电池供电的应用。

此外，CC13x2 和 CC26x2 器件平台充分利用了 Arm® 广泛可用的开发工具、SoC 基础设施 IP 应用和大型用户社区等优势。此外，这些微控制器采用兼容 Arm Thumb® 的 Thumb-2 指令集降低内存要求和成本。

TI 提供了一套用于快速将产品推向市场的完整解决方案，其中包括评估和开发板、白皮书和应用手册、易于使用的外设驱动程序库以及强大的支持、销售和分销商网络。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

1.3 功能概述

以下各分段概述了 CC13x2 及 CC26x2 器件平台的功能。

1.3.1 Arm® Cortex®-M4F

以下小节概述了 Arm® Cortex®-M4F 处理器内核和指令集、集成的系统计时器 (SysTick) 和 NVIC。

1.3.1.1 处理器内核

CC13x2 和 CC26x2 器件平台基于 Arm® Cortex®-M4F 处理器内核设计。Arm® Cortex®-M4F 处理器为高性能、低成本平台提供核心，可满足减少存储器和引脚数以及降低功耗的需求，同时提供出色的计算性能和系统中断响应能力。

处理器内核的特性如下：

- 针对小型嵌入式应用进行优化的 32 位 Arm® Cortex®-M4F 架构
- 出色处理性能与快速中断处理相结合
- Thumb-2 混合 16 位和 32 位指令集提供 32 位 Arm® 内核所能提供的高性能，但采用了通常在 8 位和 16 位器件中才会用到的更小存储器，特别是在微控制器级应用中采用的几千字节大小的存储器。
  - 单周期乘法指令和硬件除法
  - 原子位操作（位带操作），不仅最大限度利用了存储器空间，而且还简化了对外设的控制
  - 未对齐的数据访问，能够将数据高效打包到存储器中
- 快速代码执行可以降低处理器时钟速度或增加睡眠模式时间
- Harvard 架构的特点是采用独立指令和数据总线
- 高效的处理器内核、系统及存储器
- 硬件分频及快速乘法器
- 对时序敏感型应用的确定性、高性能中断处理
- 借助广泛的断点功能和通过电源模式进行调试来增强系统调试
- 紧凑型 JTAG 接口可以减少调试所需的引脚数
- 通过集成睡眠模式实现超低功耗
- 高达 48MHz 的运行频率

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

1.3.1.2 系统计时器 (SysTick)

Arm® Cortex®-M4F 处理器包含一个集成系统计时器 (SysTick)。SysTick 提供了一个简单的 24 位写时清除、递减、归零回绕计数器，并具有灵活的控制机制。该计数器有多种不同的使用方式；例如：

- 用作 RTOS 节拍计时器，它以可编程的速率（例如 100Hz）触发并调用 SysTick 例程
- 用作使用系统时钟 11 的高速警报计时器
- 用作可变速率警报或信号计时器 — 其持续时间范围取决于所使用的基准时钟和计数器的动态范围
- 作为简单计数器，用于测量完成时间以及使用的时间
- 基于超时或未超时情况的内部时钟源控制

1.3.1.3 嵌套向量中断控制器 (NVIC)

CC13x2 和 CC26x2 器件控制器包含 Arm® NVIC。NVIC 和 Arm® Cortex®-M4F 在处理器模式下对所有异常进行优先级排序和处理。发生异常时，处理器状态会自动保存到堆栈中，并在中断服务例程 (ISR) 结束时自动从堆栈恢复。中断向量的获取与状态保存并行进行，从而实现高效的中断进入。处理器支持尾链，即可以执行背靠背中断，没有状态保存和恢复开销。软件可对 7 个异常（系统处理程序）设置 8 个优先级，并可设置器件中断。

NVIC 的特性如下：

- 确定性、快速中断处理
  - 始终为 12 个周期，使用尾链时仅为 6 个周期
- 提供外部不可屏蔽中断 (NMI) 信号，用于立即执行 NMI 处理程序，适用于安全关键型应用
- 可动态重新调整中断优先级
- 通过所需寄存器操作的硬件实现方案来完成异常中断处理

1.3.1.4 系统控制块

系统控制块 (SCB) 提供系统实施信息和系统控制（系统异常的配置、控制和报告）。

1.3.2 片上存储器

以下各分段介绍了片上存储器模块。

1.3.2.1 SRAM

CC13x2 和 CC26x2 器件平台提供 80kB 低泄漏片上 SRAM，可在所有功耗模式下保留该 SRAM。可针对每个 16kB 块配置保留。此外，可以重新配置 8kB 闪存缓存 RAM 以作为正常系统 RAM 运行。由于读-修改-写 (RMW) 操作非常耗时，Arm® 在 Arm® Cortex®-M4F 处理器中引入了位结合技术。对于启用了位结合的处理器，存储器映射中的某些区域（SRAM 和外设空间）可以使用地址别名来访问单个原子操作中的单个位。

可以使用微 DMA (µDMA) 控制器将数据传输到 SRAM 或从 SRAM 传输数据。

1.3.2.2 闪存存储器

闪存块为器件提供了可在电路中编程的非易失性程序存储器。352kB 的闪存组织为一组可单独擦除的 8kB 页。擦除一个块会导致该块的全部内容重置为全 1。这些页可以单独保护。只读块不能擦除或编程，从而保护这些块的内容不被修改。除了保存程序代码和常量外，非易失性存储器还允许应用程序保存必须保留的数据，以便在器件重启后可用。使用此特性，用户能够利用已保存的特定网络数据，并且无需执行完整启动和“网络查找并加入”过程。

1.3.2.3 ROM

ROM 经过预编程，包含引导序列、器件驱动程序功能、低级协议栈组件和串行引导加载程序（SPI 或 UART）。

1.3.3 无线电

CC26x2 器件提供高度集成的低功耗 2.4GHz 无线电收发器，支持多种调制和数据包格式。CC1312 器件提供针对低于 1GHz 频段优化的类似功能。CC1352 器件是真正的双频带无线电，具有用于低于 1GHz 和 2.4GHz 运行的单独 RF 路径。无线电子系统在 MCU 和无线电之间提供了一个接口，使其可发出命令、读取状态，还能自动执行无线电事件并对其进行排序。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

1.3.4 安全内核

CC13x2 和 CC26x2 器件平台的安全内核配备了高级加密标准 (AES) 模块，支持 256 位密钥、本地密钥库和 DMA 功能。它还包含一个哈希引擎 (SHA-2) 和一个公钥加速引擎 (PKA)。

AES 引擎的特性如下：

- ECB、CBC、CBC-MAC、CTR、CCM 和 GCM 工作模式
- 118Mbps 吞吐量
- 安全密钥库存储器
- 低延迟
- SHA-224、SHA-256、SHA-384 和 SHA-512
- 用于椭圆曲线计算的硬件加速器

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

1.3.5 通用计时器

通用计时器可对驱动计时器输入引脚的外部事件进行计数或计时。每个 16 位或 32 位 GPTM 模块都提供两个 16 位的计时器/计数器，用户可以将它们配置成独立运行的计时器或事件计数器，或配置成一个 32 位计时器来运行。

通用定时器模块 (GPTM) 包含四个 16 位或 32 位 GPTM 模块，具有以下功能选项：

- 16 位或 32 位操作模式：
  - 16 位或 32 位可编程的单次定时器
  - 16 位或 32 位可编程的周期定时器
  - 具有 8 位预分频的 16 位通用定时器
  - 带有 8 位预分频器的 16 位输入沿计数或计时捕获模式
  - 带有 8 位预分频器的 16 位 PWM 模式以及软件编程实现的 PWM 信号反相输出
- 向上或向下计数
- 四个 32 位计数器或多达八个 16 位计数器
- 多达 8 个捕捉/比较引脚
- 多达四个 PWM 引脚（每个 32 位计时器有一个 PWM 引脚）
- 计时器模块的菊花链允许单个计时器启动多个时序事件
- 计时器同步允许所选计时器在同一时钟周期开始计数
- 调试期间，微控制器的 CPU 暂停时，用户可启用的停滞
- 能够确定从定时器中断断言到进入 ISR 之间经过的时间
- 使用 µDMA 控制器进行高效传输

1.3.5.1 看门狗计时器

看门狗计时器用于在系统由于软件错误而发生故障或者外部器件无法正确响应时重新获得控制权。当达到预定义的超时值时，看门狗计时器可以生成中断或复位。

1.3.5.2 常开域

AON 域包含始终启用的电路，但关断模式（数字电源关闭的情况）除外。此域包含以下内容：

- RTC 可用于将 CC13x2 和 CC26x2 器件平台从任何活动状态唤醒。RTC 包含三个匹配寄存器及一个比较寄存器。借助软件支持，RTC 可用于时钟和日历操作。RTC 由 32kHz RC 振荡器或 32kHz 晶体振荡器计时。
- 可通过软件访问电池监测器及温度传感器。电池监控器和温度传感器能够持续监控电池状态及粗略温度。

1.3.6 直接存储器存取

CC13x2 和 CC26x2 器件平台包含一个 DMA 控制器，称为 µDMA。µDMA 控制器以某种方式减少 Arm® Cortex®-M4F 处理器参与的数据传输任务，从而更加高效地使用处理器以及可用的总线带宽。µDMA 控制器能够执行存储器与外设之间的数据传输。µDMA 中的通道专用于每个受支持的片上模块，可以编程为自动执行外围设备和内存之间的传输，因为外围设备已准备好传输更多数据。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

1.3.7 系统控制和时钟

系统控制决定了 CC13x2 及 CC26x2 器件平台的整体运行方式。系统控制提供有关器件的信息，控制节能功能，控制器件和各个外设的时钟，并处理复位检测和报告。

- 功率控制：
  - 片上固定直流/直流转换器和低压降 (LDO) 稳压器
  - 处理内核数字逻辑和模拟电路的上电时序、断电时序及控制
  - 适用于微控制器的低功耗选项
  - 片上模块的低功耗选项：
    - 软件控制各个外设及存储器的关断
    - 在所有功耗模式下都会保留 RAM 及配置寄存器
  - 用于控制外部直流/直流稳压器的控制引脚选项
  - 可配置的从睡眠计时器唤醒或者任何 GPIO 中断
  - 电压监控电路
- 微控制器系统时钟的多个时钟源：
  - RC 振荡器 (HSRCOSC)：
    - 提供 48MHz 频率的片上资源
    - 48MHz 晶体振荡器 (HSXOSC) 是一个外部晶体中频率精确的时钟源，外部晶体通过 X48M_P 输入和 X48M_N 输出引脚连接。
    - 内部 32kHz RC 振荡器是一种提供 32kHz 频率的片上资源，在节能模式下使用并用于 RTC。
    - 32.768kHz 晶体振荡器是连接在 X32K_Q1 输入及 X32K_Q2 输出引脚上的外部晶体的频率精确时钟源
    - 非常适合精确的 RTC 操作或者同步网络计时
    - 可以使用其中一个数字输入/输出 (DIO) 引脚作为时钟输入，来提供外部 32.768kHz 时钟信号。
  - CPU 和外设时钟分频选项

1.3.8 串行通信外设

CC13x2 和 CC26x2 器件平台支持异步和同步串行通信，包括：

- 2 个 UART 模块
- I^2C 模块
- I^2S 模块
- 两个 SSI (SPI) 模块

以下小节更详细地介绍了每种通信功能。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

1.3.8.1 UART

UART 是一个用于 RS-232C 串行通信的集成电路。UART 包含一个传输器（并行转串行转换器）和一个接收器（串行转并行转换器）；两者单独提供时钟。

CC13x2 和 CC26x2 器件平台包含一个完全可编程的 UART。UART 可以通过接收 (RX)、传输 (TX)、调制解调器流控制和错误条件来生成单独屏蔽的中断。当任何中断被触发且未被屏蔽时，该模块会生成一个组合中断。

该 UART 具有以下特性：

- 可编程的波特率发生器支持高达 3Mbps 的速度
- 独立的 32 × 8 TX FIFO 和 32 × 16 RX FIFO 可减少 CPU 中断服务负载
- 可编程的 FIFO 长度，包括提供传统双缓冲接口的 1 字节深度操作
- FIFO 触发深度有如下级别可选：1/8、1/4、1/2、3/4 或 7/8
- 标准的异步通讯位：起始位、停止位、奇偶校验位
- 线中止的产生与检测；
- 完全可编程的串行接口特性：
  - 可包含 5、6、7 或 8 个数据位
  - 偶校验、奇校验、固定校验或无奇偶校验位生成与检测
  - 生成 1 或 2 个停止位
- 支持完整的调制解调器握手
- 可编程硬件流控制
- 标准 FIFO 级中断
- 使用 µDMA 控制器进行高效传输：
  - 独立的 TX 和 RX 通道
  - 当接收 FIFO 中有数据时产生单次请求；当接收 FIFO 到达预设的触发深度时产生猝发请求
  - 当发送 FIFO 中有空闲单元时产生单次请求；当发送 FIFO 到达预设的触发深度时产生猝发请求

1.3.8.2 I^2C

I^2C 总线通过两线设计（串行数据线路 SDA 和串行时钟线路 SCL）来提供双向数据传输。I^2C 总线可与诸如串行存储器（RAM 和 ROM）、网络设备、LCD、音频发生器等外部 I^2C 器件相连。在产品开发和制造过程中，I^2C 总线还可用于系统测试和诊断用途。

I^2C 总线上的每个器件可指定为主器件或从器件。每个 I^2C 模块都支持发送和接收数据（作为主器件或从器件），并且可以同时工作（作为主器件和从器件）。I^2C 主器件和从器件都可以生成中断。

CC13x2 和 CC26x2 器件平台包含一个 I^2C 模块，其功能如下：

- I^2C 总线上的器件可指定为主器件或从器件：
  - 支持作为主器件或从器件发送和接收数据
  - 同时支持主和从工作模式
- 四种 I^2C 模式：
  - 主器件发送
  - 主器件接收
  - 从器件发送
  - 从器件接收

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

- 两种传输速度：
  - 标准 (100kbps)
  - 快速 (400kbps)
- 时钟低电平超时中断
- 主器件和从器件中断生成：
  - 主器件会在 TX 或 RX 操作完成时产生中断（或者是因为错误而中止）
  - 当主器件传输或请求数据时，或者当检测到启动或停止条件时，从器件会生成中断
  - 主器件具有仲裁和时钟同步、多主器件支持以及 7 位寻址模式

1.3.8.3 I^2S

I^2S 模块使 CC13x2 和 CC26x2 设备平台能够与编解码器、DAC、ADC 或 DSP 等外部设备通信。这些器件仅支持 I^2S、RJF、LJF 及 DSP 等音频流格式；这些器件不支持外部器件配置。CC13x2 及 CC26x2 器件平台支持外部和内部生成的位时钟和字时钟（BCLK 和 WCLK）。

1.3.8.4 SSI

SSI 模块是一个 4 线双向通信接口，可在并行和串行之间转换数据。SSI 对从外设接收的数据执行串行到并行转换，并对发送到外设的数据执行并行到串行转换。SSI 可以配置为主器件或从器件。配置为从器件时，SSI 可配置为禁用其输出，从而允许将主器件与多个从器件耦合。TX 和 RX 路径通过独立的内部 FIFO 进行缓冲。

SSI 还包含一个可编程的比特率时钟分频器和预分频器，用于生成源自 SSI 输入时钟的输出串行时钟。比特率根据输入时钟生成，最大比特率由连接的外设决定。

CC13x2 和 CC26x2 器件平台包含两个 SSI 模块，具有以下特性：

- 可编程接口操作，支持 Freescale SPI、MICROWIRE 或 TI 同步串行接口
- 主/从运行
- 可编程的时钟位速率以及预分频器；
- 独立的 TX 和 RX FIFO，分别宽 16 位，深 8 个位置
- 可编程数据帧大小为 4 位到 16 位
- 内部环回测试模式，用于诊断和调试测试
- 基于 FIFO 的标准中断和 EoT 中断
- 使用 µDMA 控制器进行高效传输：
  - 独立的 TX 和 RX 通道
  - 当数据位于 FIFO 中时，发出单次接收请求；当 FIFO 包含四个条目时，发出突发请求
  - 当 FIFO 中有空间时，发出单次发送请求；当 FIFO 包含四个条目时，发出突发请求

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

1.3.9 可编程 I/O

I/O 引脚可以灵活用于各种连接。CC13x2 和 CC26x2 器件平台支持高度可配置的 I/O 引脚，这些引脚可通过 I/O 控制器多路复用为任何数字外设。

备注

模拟功能、传感器控制器连接和高驱动强度仅限于某些引脚。有关详细信息，请参阅 章节 13。

- 多达 31 个 GPIO，具体取决于配置
- 多达五个 8mA 驱动强度引脚
- 完全灵活的数字引脚复用功能，允许将每个引脚用作 GPIO 或任意多种外设功能之一
- GPIO 中断的可编程控制：
  - 针对每个引脚的中断生成屏蔽
  - 在上升或下降沿触发
- 通过地址线进行读取和写入操作中的位屏蔽
- 可以启动 µDMA 传输
- 可以在所有睡眠模式期间保留引脚状态
- 配置为数字输入的引脚是施密特触发
- DIO 配置的可编程控制：
  - 弱上拉或下拉电阻器
  - 数字输入使能

1.3.10 传感器控制器

传感器控制器包含可在省电模式下选择性启用的电路。该域中的外设可由传感器控制器（这是一个专有的功耗优化型 CPU，即传感器控制器引擎）控制，也可直接由系统 CPU 控制。传感器控制器引擎 CPU 可读取和监测传感器，或自主执行其他任务，从而降低功耗并减轻系统 CPU 的负担。

传感器控制器通过基于 PC 的配置工具进行设置，典型用例可能包括（但不限于）以下方面：

- 使用集成 ADC 的模拟传感器
- 使用 GPIO 以及位冲击 I^2C 和 SPI 的数字传感器
- 电容式感应
- 波形生成
- 键盘扫描
- 用于轮询旋转传感器的正交解码器
- 振荡器校准

传感器接口中的外设包括以下部分：

- 模拟比较器

  超低功耗模拟比较器可将 CC13x2 和 CC26x2 器件平台从任何活动状态唤醒。该比较器可使用可配置的内部基准。比较器的输出也可触发中断或触发 ADC。
- 电容式感应

  电容式感应并非 CC13x2 和 CC26x2 器件平台中的独立模块；相反，该功能是通过使用恒流源、时数转换器和比较器来实现的。该模块中的模拟比较器也可用作超低功耗比较器的高精度替代方案。传感器控制器负责基线跟踪、磁滞、滤波及其他相关功能。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

- ADC

  该 ADC 是一款 12 位、200ksps ADC，具有 8 个输入和一个内置电压基准。该 ADC 可由多种不同的源触发，包括计时器、I/O 引脚、软件、模拟比较器和 RTC。

  ADC 是一种将连续模拟电压转换为离散数字量的外设。ADC 模块具有 12 位转换分辨率，支持八个输入通道，以及电池电压内部分压和温度传感器。
- 低功耗 UART、SPI 和 I^2C 数字传感器接口

  模拟模块最多可连接八个不同的 I/O。

1.3.11 随机数生成器

随机数生成器生成真随机数，用于退避计算或安全密钥。

1.3.12 cJTAG 及 JTAG

联合测试行动组 (JTAG) 端口是一个 IEEE 标准，它定义了数字集成电路的测试接入端口 (TAP) 和边界扫描架构。JTAG 端口还提供一个标准化串行接口来控制相关测试逻辑。TAP、指令寄存器 (IR) 和数据寄存器 (DR) 可用于测试组装的印刷电路板 (PCB) 的互连，并获取有关元件的制造信息。JTAG 端口还提供了一种访问和控制为测试而设计的特性的方法，例如 I/O 引脚观察和控制、扫描测试和调试。紧凑型 JTAG (cJTAG) 接口具有以下特性：

- 符合 IEEE1149.1-1990 标准的 TAP 控制器
- IEEE 1149.7 cJTAG 接口
- ICEPick JTAG 路由器
- 用于存储 JTAG 指令的 4 位 IR 链
- IEEE 标准指令：BYPASS、IDCODE、SAMPLE 和 PRELOAD、EXTEST 和 INTEST
- Arm® 附加说明：APACC、DPACC 和 ABORT

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

1.3.13 电源系统

1.3.13.1 电源系统

CC13x2 和 CC26x2 器件平台使用多种电压电平。图 1-2 为电源系统的概况图。

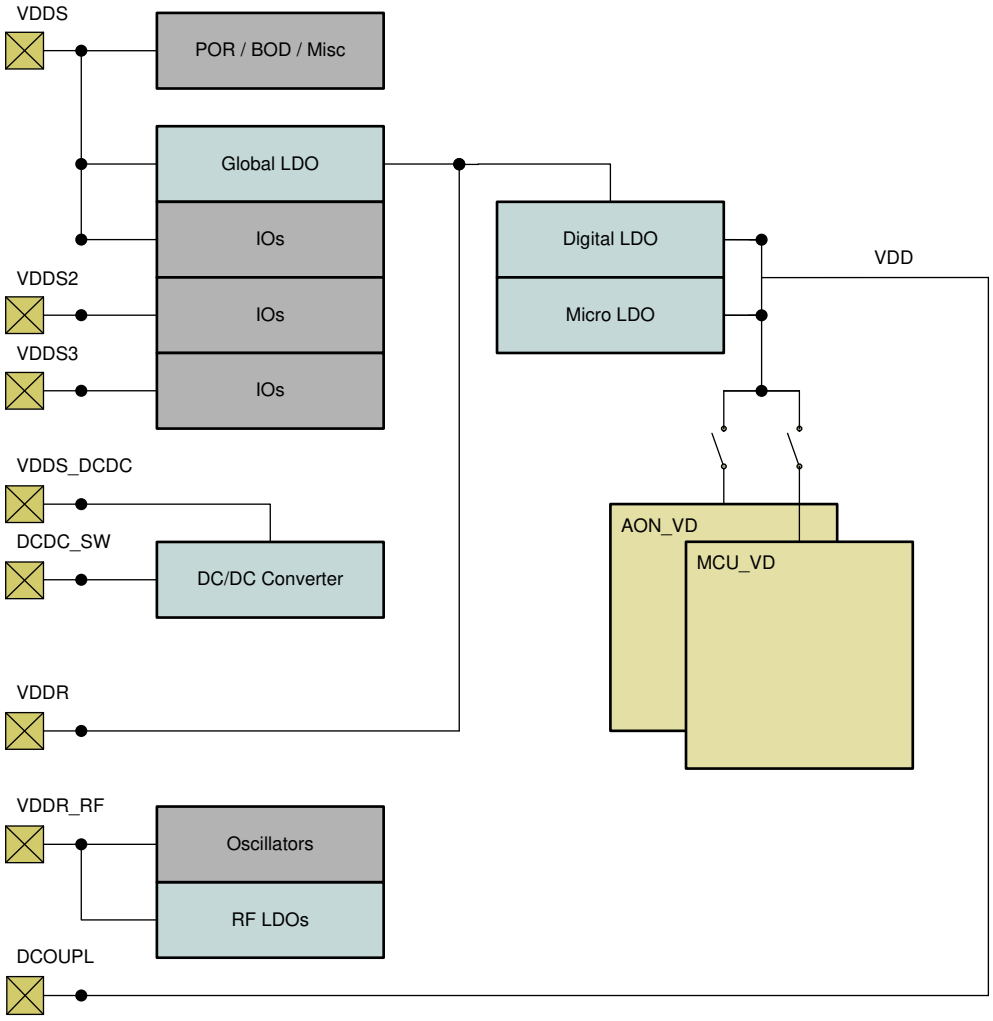


图 1-2. CC13x2 和 CC26x2 电源系统

1.3.13.1.1 VDDS

CC13x2 和 CC26x2 器件平台上的电池电压称作 VDDS（电源）。该电源在系统中具有最高电势，通常是用户提供的唯一电势。

备注

VDDS2 和 VDDS3 必须始终与 VDDS 处于同一电位。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

1.3.13.1.2 VDDR

两个 VDDR（稳压）引脚通常由其中一个内部稳压器供电。为了实现超低功耗，TI 建议使用内部直流/直流稳压器（有关该配置的更多详细信息，请参阅节 1.3.13.2）。

也可以选择使用全局 LDO。在这种情况下，两个 VDDR 引脚必须连接在一起。在这种情况下，VDDR 应具有 22µF 去耦电容器，而 VDDR_RF 应具有在各种参考设计中建议的去耦。在该设置中，VDDS_DCDC 应连接到 VDDS，并且 DCDC_SW 应保持悬空。

1.3.13.1.3 数字内核电源

CC13x2 和 CC26x2 器件平台的数字内核由一个连接到 VDDR 的 1.28V 稳压器供电。该稳压器的输出需要一个外部去耦电容器才能正常工作；此电容器必须连接到 DCOUPL 引脚。

备注

DCOUPL 引脚不能用于为外部电路供电。

当系统处于省电状态时，一个小型低功耗稳压器（微型 LDO）以有限的电流能力为数字域供电，以确保已启用的模块仍有电源。

1.3.13.1.4 其他内部电源

器件中的几个其他模块（例如频率合成器、RF 功率放大器等）具有独立的内部稳压器，其工作电压为 1.4V（模拟模块）或 1.28V（数字模块）。这些稳压器在需要时通过固件自动上电或断电。

1.3.13.2 直流/直流转换器

片上降压模式 DC/DC 转换器提供了一种降低器件功耗的简单方法。DC/DC 转换器集成在电源系统中，并通过系统控制器自动处理偏置和时钟。

DC/DC 转换器通过 AON_PMCTL:PWRCTL 寄存器进行控制。

要在系统处于运行状态时启用 DC/DC 转换器，必须设置 AON_PMCTL:PWRCTL.DCDC_ACTIVE 位。当器件处于待机模式时，也会定期使用 DC/DC 转换器，以维持 VDDR 域上的电压。

DC/DC 稳压器的输出电压通常被修整为 1.68V，但在某些应用场景中也会采用其他电压电平。电压电平由器件自动控制，用户无法更改。

备注

DC/DC 稳压器输出不能用于为外部电路供电。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

章节 2

Arm® Cortex®-M4F 处理器

CC13x2 和 CC26x2 器件平台基于 Arm® Cortex®-M4F 内核构建，为工厂自动化和控制、工业控制功率器件、楼宇和家居自动化以及步进电机控制等成本敏感型嵌入式微控制器应用提供高性能 32 位计算能力。

本章提供有关 Arm® Cortex®-M4F 处理器 CC13x2 和 CC26x2 器件平台实现的信息。

有关指令集的技术详情，请参阅 *Cortex-M3/M4F 指令集技术用户手册*。

2.1 Arm® Cortex®-M4F 处理器简介 84
2.2 方框图 84
2.3 概述 85
2.4 编程模型 87
2.5 Arm® Cortex®-M4F 内核寄存器 87
2.6 指令集概要 102
2.7 浮点单元 (FPU) 111
2.8 存储器保护单元 (MPU) 117
2.9 Arm® Cortex®-M4F 处理器寄存器 118

Copyright © 2026 Texas Instruments Incorporated

2.1 Arm® Cortex®-M4F 处理器简介

Arm® Cortex®-M4F 处理器作为一种高性能、低成本的平台，能够满足系统对于减小存储器、减少引脚数以及降低功耗的要求。包括以下特性：

- 针对小型嵌入式应用进行优化的 32 位 Arm® Cortex®-M4F 架构
- 出色的处理性能与快速的中断处理相结合
- Arm Thumb®-2 技术（采用混合 16 位和 32 位指令集）提供 32 位 Arm® 内核所能提供的高性能，但采用了通常在 8 位和 16 位器件中才会用到的更小存储器，特别是在微控制器级应用中采用的几千字节大小的存储器：
  - 单周期乘法指令和硬件除法
  - 原子位操作（位带操作），不仅最大限度利用了存储器空间，而且还简化了对外设的控制
  - 未对齐的数据访问，能够将数据高效打包到存储器中
- 快速代码执行可以降低处理器时钟速度或增加睡眠模式时间
- Harvard 架构的特点是采用独立指令和数据总线
- 高效的处理器内核、系统及存储器
- 面向硬件除法和快速数字信号处理的乘法累加运算
- 用于信号处理的饱和算术
- 对时序敏感型应用的确定性、高性能中断处理
- 用于单精度浮点算术功能的浮点单元 (FPU)
- 用于存储器管理和保护功能的存储器保护单元 (MPU)
- 通过广泛的断点和跟踪功能增强系统调试
- 通过数据匹配功能生成观察点以进行全面调试
  - DWT
  - JTAG 调试端口
  - FPB
- 从 Arm7® 处理器系列迁移到更高性能和电源效率
- 标准跟踪支持
  - ITM
  - 具备异步串行线输出 (SWO) 的 TPIU
- 针对单周期闪存存储器使用进行了优化
- 通过集成睡眠模式实现超低功耗
- 48MHz 工作频率

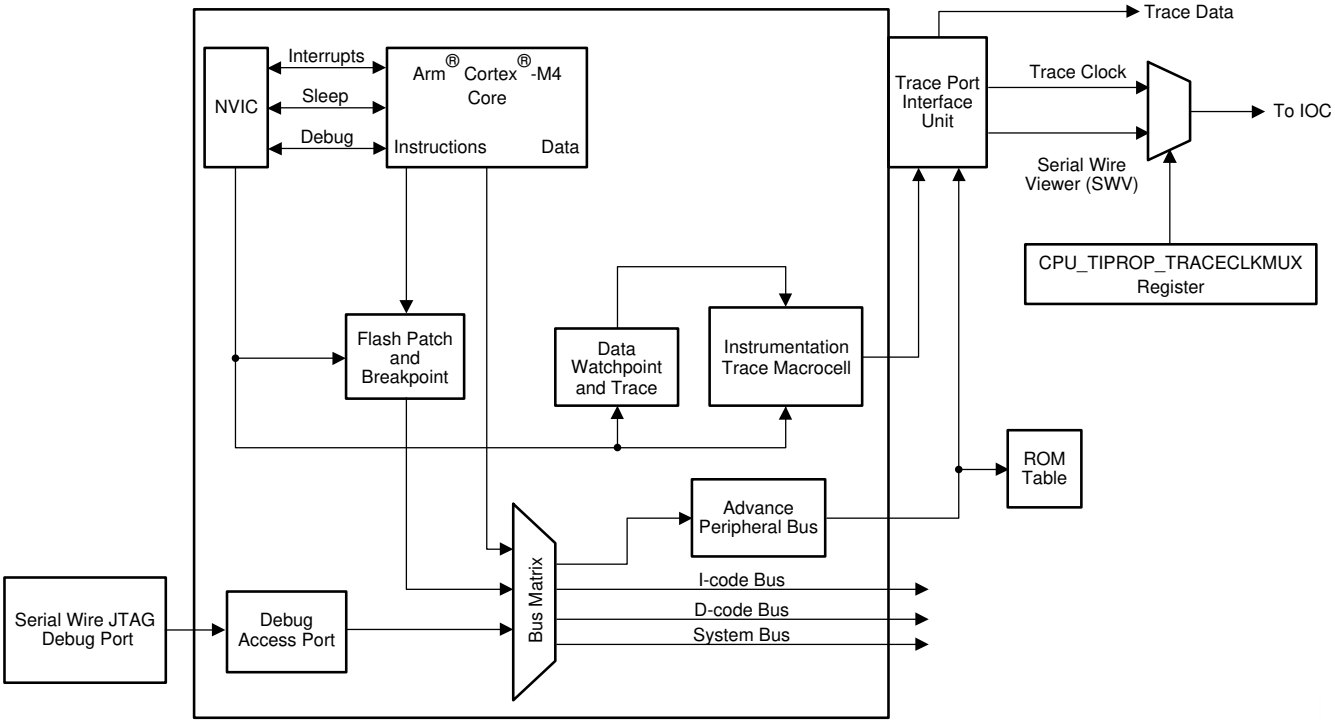
2.2 方框图

图 2-1 显示了核心处理器单元 (CPU) 方框图。Arm® Cortex®-M4F 处理器基于采用 3 级流水线 Harvard 架构的高性能处理器内核构建，因此它是要求苛刻的嵌入式应用的理想选择。该处理器采用高效的指令集和广泛优化的设计，提供高端的处理硬件、从而提供卓越的电源效率。该指令集包含一系列单周期和 SIMD 乘法以及带累加的乘法功能、饱和算术和专用硬件除法。

为了便于设计成本敏感型器件，Arm® Cortex®-M4F 处理器实现了紧密耦合的系统元件，可减少处理器面积，同时显著改善中断处理和系统调试功能。Arm® Cortex®-M4F 处理器采用基于 Thumb-2 技术的 Thumb 指令集版本，可确保高代码密度并减少程序存储器要求。Arm® Cortex®-M4F 指令集提供了现代 32 位架构所期望的卓越性能，具有 8 位和 16 位微控制器的高代码密度。

Arm® Cortex®-M4F 处理器与嵌套矢量中断控制器 (NVIC) 紧密集成，以提供中断服务例程 (ISR) 的快速执行，从而显著降低中断延迟。寄存器的硬件堆叠及暂停多次加载和存储多个操作的能力进一步降低了中断延迟。中断处理程序不需要任何汇编桩代码，从而消除了 ISR 中的代码开销。尾链优化还显著减少了从一个 ISR 切换至另一个 ISR 时的开销。为了优化低功耗设计，NVIC 集成了睡眠模式（包括深度睡眠模式），以便能够快速关断整个器件。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated



Copyright © 2017, Texas Instruments Incorporated

图 2-1. CPU 方框图

2.3 概述

2.3.1 系统级接口

Arm® Cortex®-M4F 处理器提供使用 AMBA® 技术的多个接口，以提供高速、低延迟存储器访问。该内核支持未对齐数据访问并实现原子位操作，从而实现更快的外设控制、系统自旋锁和线程安全布尔数据处理。

2.3.2 集成可配置调试

Arm® Cortex®-M4F 处理器通过串行线或 JTAG 调试端口 (SWJ-DP) 模块实现了完整的硬件调试解决方案。SWJ-DP 通过传统 JTAG 端口提供了对处理器和存储器的高系统可见性。有关 SWJ-DP 的详细信息，请参阅章节 6 和 *Arm 调试接口 V5 架构规范*。

对于系统跟踪，处理器集成了一个仪表跟踪宏单元 (ITM)，以及数据观察点和性能分析单元。为了实现对系统跟踪事件的简单且经济的性能分析，串行线查看器 (SWV) 可通过一个引脚来导出软件生成的消息、数据跟踪和性能分析信息流。

闪存补丁和断点单元 (FPB) 提供最多八个硬件断点比较器，供调试器使用。FPB 中的比较器还为 CODE 存储器区域内的程序代码提供多达八个字的重映射功能。重映射功能允许利用存储在闪存只读区域的应用程序来修补片上 SRAM 或闪存的其他区域。如果需要补丁，应用程序会编程 FPB 以重映射多个地址。访问这些地址时，访问将重定向到 FPB 配置中指定的重映射表。

有关 Arm® Cortex®-M4F 调试功能的更多信息，请参阅 *Arm 调试接口 V5 架构规范*。

2.3.3 跟踪端口接口单元

图 2-2 显示了跟踪端口接口单元 (TPIU) 方框图。TPIU 充当来自 ITM 的 Arm® Cortex®-M4F 跟踪数据和片外跟踪端口分析器之间的桥梁。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

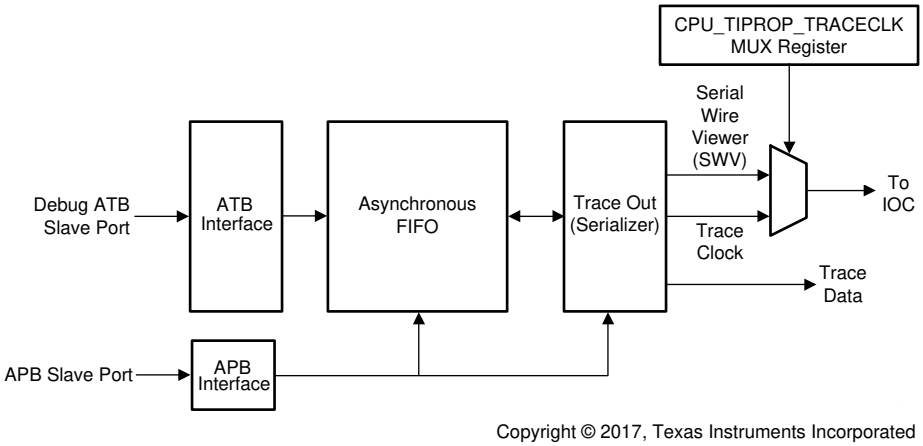


图 2-2. TPIU 方框图

2.3.4 浮点单元 (FPU)

Arm® Cortex®-M4F 处理器包括一个浮点单元 (FPU)，此类单元是 ARMv7-M 浮点扩展 (FPv4-SP) 单精度型号的实现。它提供浮点计算功能，符合 ANSI/IEEE Std 754-2008 IEEE 二进制浮点运算标准（称为 IEEE 754 标准）。FPU 支持 *ARMv7-M 架构参考手册* 中所述的所有单精度数据处理指令和数据类型（请参阅节 2.7）。

2.3.5 存储器保护单元 (MPU)

Arm® Cortex®-M4F 处理器包括一个支持标准 ARMv7 保护存储器系统架构模型的存储器保护单元 (MPU)（请参阅节 2.8）。

2.3.6 Arm® Cortex®-M4F 系统组件详情

Arm® Cortex®-M4F 包括以下系统组件：

- **SysTick：**24 位倒计时计时器，可用作实时操作系统 (RTOS) 节拍计时器或简单计数器（请参阅节 4.2.1）
- **嵌套向量中断控制器：**嵌入式中断控制器 (INTC)，支持低延迟中断处理（请参阅节 4.2.2）
- **系统控制块：**处理器的编程模型接口，提供系统实现信息和系统控制，包括配置、控制和报告系统异常（请参阅节 4.2.3）。处理器的关键控制和状态功能在系统控制空间 (SCS) 内的 SCB 中集中管理。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.4 编程模型

本节介绍 Arm® Cortex®-M4F 的编程模型。有关软件执行和堆栈的处理器模式与特权级别的更多信息，以及各个内核寄存器的说明，请参阅节 2.5。

2.4.1 软件执行的处理器模式和特权等级

Arm® Cortex®-M4F 处理器具有两种运行模式：

- 线程模式执行应用软件。处理器在退出复位后进入线程模式。
- 处理程序模式处理异常。当处理器完成异常处理时，它返回到线程模式。

此外，Arm® Cortex®-M4F 处理器有两个特权等级：非特权级和特权级。

- 在非特权模式下，软件有以下限制：
  - 对 MSR 和 MRS 指令的访问受限，不能使用 CPS 指令
  - 无法访问系统计时器、NVIC 或 SCB
- 在特权模式下，软件可以使用所有指令，并可以访问处理器中的所有资源。

在线程模式下，CONTROL 寄存器（请参阅表 2-24）控制软件执行是特权级还是非特权级。在处理程序模式下，软件执行始终是特权级的。

只有特权软件可以写入 CONTROL 寄存器，以更改线程模式下软件执行的特权等级。非特权软件可以使用 SVC 指令进行监管调用，将控制权转移给特权软件。

2.4.2 栈

Arm® Cortex®-M4F 处理器使用一个完整的降序栈，这意味着栈指针指示存储器中的最后一个堆叠项。当处理器将新项推入栈时，它会递减栈指针，然后将该项写入新的存储器位置。处理器实现了两个栈，即主栈和进程栈，每个栈都有一个指针保存在独立的寄存器中（参见表 2-16 中的 SP 寄存器）。

在线程模式下，CONTROL 寄存器（请参阅表 2-24）控制处理器是使用主栈还是进程栈。在处理程序模式下，处理器始终使用主栈。表 2-1 列出了处理器运行的选项。

表 2-1. 处理器模式、特权级别和栈使用汇总

处理器模式	使用	权限级别	已使用的栈
Thread	应用	特权级或非特权 (1)	主栈或进程栈
处理程序	异常处理程序	始终具有特权	主栈

(1) 请参阅表 2-24 中的 CONTROL 寄存器。

2.4.3 异常和中断

异常会改变软件控制的正常流程。对中断和系统异常的支持是通过使用内置 NVIC 来实现，NVIC 支持多达 240 个外部中断输入。除外部中断外，Arm® Cortex®-M4F 还会处理 16 个预定义的异常源，包括复位、NMI 等。处理器和 NVIC 对所有异常进行优先级排序后处理。处理器使用处理程序模式来处理除复位之外的所有异常。软件通过寄存器配置分配给 NVIC 外部中断输入的实际优先级。

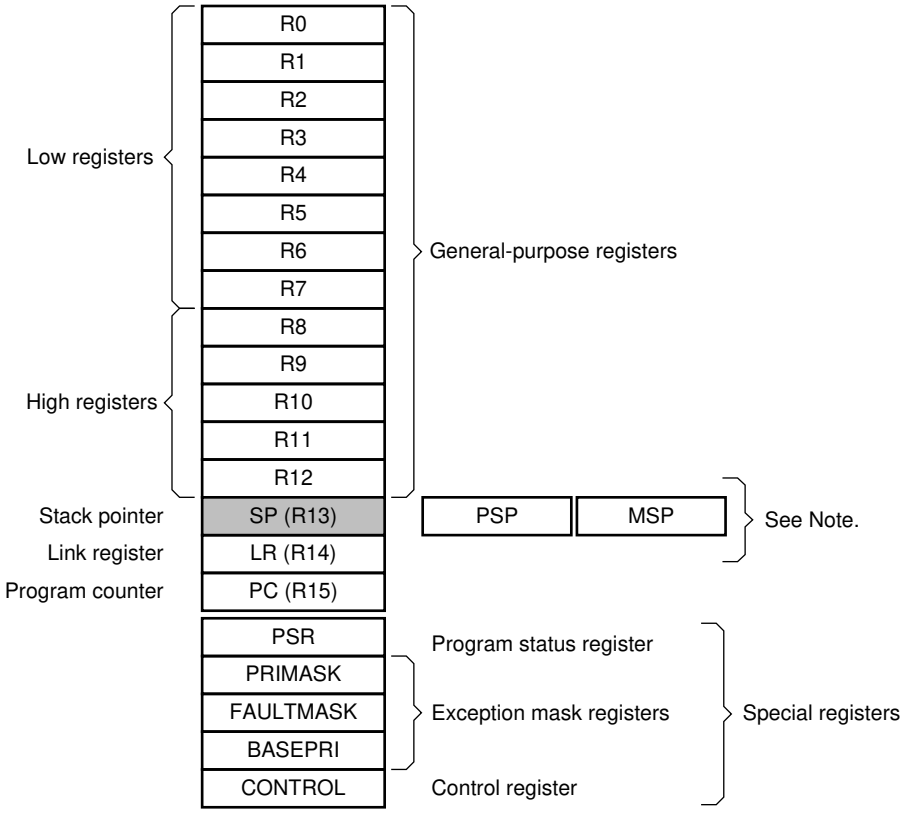
2.4.4 数据类型

Arm® Cortex®-M4F 处理器支持 32 位字、16 位半字及 8 位字节。该处理器还支持 64 位数据传输指令。所有指令及数据存储器访问都是小端字节序。更多信息，请参阅 *Cortex-M3/M4F 指令集技术用户手册*。

2.5 Arm® Cortex®-M4F 内核寄存器

图 2-3 展示了 Arm® Cortex®-M4F 寄存器组。表 2-2 列出了这些内核寄存器。内核寄存器未映射到存储器，需通过寄存器名称访问，因此基址为 N/A（不适用），且不存在偏移。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated



注意：SP 的分组版本

图 2-3. Arm® Cortex®-M4F 寄存器组

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.5.1 内核寄存器映射

表 2-2. 处理器寄存器映射

名称	类型	复位	说明	链接
R0	R/W	—	Cortex 通用数据寄存器 0	请参阅 节 2.5.2.1。
R1	R/W	—	Cortex 通用数据寄存器 1	请参阅 节 2.5.2.2。
R2	R/W	—	Cortex 通用数据寄存器 2	请参阅 节 2.5.2.3。
R3	R/W	—	Cortex 通用数据寄存器 3	请参阅 节 2.5.2.4。
R4	R/W	—	Cortex 通用数据寄存器 4	请参阅 节 2.5.2.5。
R5	R/W	—	Cortex 通用数据寄存器 5	请参阅 节 2.5.2.6。
R6	R/W	—	Cortex 通用数据寄存器 6	请参阅 节 2.5.2.7。
R7	R/W	—	Cortex 通用数据寄存器 7	请参阅 节 2.5.2.8。
R8	R/W	—	Cortex 通用数据寄存器 8	请参阅 节 2.5.2.9。
R9	R/W	—	Cortex 通用数据寄存器 9	请参阅 节 2.5.2.10。
R10	R/W	—	Cortex 通用数据寄存器 10	请参阅 节 2.5.2.11。
R11	R/W	—	Cortex 通用数据寄存器 11	请参阅 节 2.5.2.12。
R12	R/W	—	Cortex 通用数据寄存器 12	请参阅 节 2.5.2.13。
SP	R/W	-	栈指针	请参阅 节 2.5.2.14。
LR	R/W	0xFFFF FFFF	链接寄存器	请参阅 节 2.5.2.15。
PC	R/W	—	程序计数器	请参阅 节 2.5.2.16。
PSR	R/W	0x0100 0000	程序状态寄存器	请参阅 节 2.5.2.17。
PRIMASK	R/W	0x0000 0000	优先级屏蔽寄存器	请参阅 节 2.5.2.18。
FAULTMASK	R/W	0x0000 0000	故障屏蔽寄存器	请参阅 节 2.5.2.19。
BASEPRI	R/W	0x0000 0000	基址优先级屏蔽寄存器	请参阅 节 2.5.2.20。
控制	R/W	0x0000 0000	控制寄存器	请参阅 节 2.5.2.21。

2.5.2 内核寄存器描述

本节按图 2-3 中列出的顺序列出并描述 Arm® Cortex®-M4F 寄存器。内核寄存器不进行存储器映射，需通过寄存器名称而不是偏移进行访问。

备注

寄存器描述中所示的寄存器类型是指线程模式和处理程序模式下程序执行期间的类型。调试访问可能有所不同。

2.5.2.1 Cortex® 通用寄存器 0 (R0)

表 2-3. Cortex® 通用寄存器 0 (R0)

地址偏移量		**复位**	-
物理地址		**实例**	
说明	R0 寄存器是用于数据操作的 32 位通用寄存器，可在特权模式或非特权模式下进行访问。		
类型	R/W		

31	30	29	28	27	26	25	24	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																															

位	字段名称	说明	类型	复位
31-0	DATA	寄存器数据	R/W	—

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.5.2.2 Cortex® 通用寄存器 1 (R1)

表 2-4. Cortex® 通用寄存器 1 (R1)

地址偏移量		复位	-
物理地址		实例	
说明	R1 寄存器是用于数据操作的 32 位通用寄存器，可从特权模式或非特权模式进行访问。		
类型	R/W		

31	30	29	28	27	26	25	24	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																															

位	字段名称	说明	类型	复位
31-0	DATA	寄存器数据	R/W	—

2.5.2.3 Cortex® 通用寄存器 2 (R2)

表 2-5. Cortex® 通用寄存器 2 (R2)

地址偏移量		复位	-
物理地址		实例	
说明	R2 寄存器是用于数据操作的 32 位通用寄存器，可从特权模式或非特权模式进行访问。		
类型	R/W		

31	30	29	28	27	26	25	24	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																															

位	字段名称	说明	类型	复位
31-0	DATA	寄存器数据	R/W	—

2.5.2.4 Cortex® 通用寄存器 3 (R3)

表 2-6. Cortex® 通用寄存器 3 (R3)

地址偏移量		复位	-
物理地址		实例	
说明	R3 寄存器是用于数据操作的 32 位通用寄存器，可从特权模式或非特权模式进行访问。		
类型	R/W		

31	30	29	28	27	26	25	24	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																															

位	字段名称	说明	类型	复位
31-0	DATA	寄存器数据	R/W	—

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.5.2.5 Cortex® 通用寄存器 4 (R4)

表 2-7. Cortex® 通用寄存器 4 (R4)

地址偏移量		复位	-
物理地址		实例	
说明	R4 寄存器是用于数据操作的 32 位通用寄存器，可从特权模式或非特权模式进行访问。		
类型	R/W		

31	30	29	28	27	26	25	24	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																															

位	字段名称	说明	类型	复位
31-0	DATA	寄存器数据	R/W	—

2.5.2.6 Cortex® 通用寄存器 5 (R5)

表 2-8. Cortex® 通用寄存器 5 (R5)

地址偏移量		复位	-
物理地址		实例	
说明	R5 寄存器是用于数据操作的 32 位通用寄存器，可从特权模式或非特权模式进行访问。		
类型	R/W		

31	30	29	28	27	26	25	24	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																															

位	字段名称	说明	类型	复位
31-0	DATA	寄存器数据	R/W	—

2.5.2.7 Cortex® 通用寄存器 6 (R6)

表 2-9. Cortex® 通用寄存器 6 (R6)

地址偏移量		复位	-
物理地址		实例	
说明	R6 寄存器是用于数据操作的 32 位通用寄存器，可在特权模式或非特权模式下进行访问。		
类型	R/W		

31	30	29	28	27	26	25	24	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																															

位	字段名称	说明	类型	复位
31-0	DATA	寄存器数据	R/W	—

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.5.2.8 Cortex® 通用寄存器 7 (R7)

表 2-10. Cortex® 通用寄存器 7 (R7)

地址偏移量		复位	-
物理地址		实例	
说明	R7 寄存器是用于数据操作的 32 位通用寄存器，可从特权模式或非特权模式进行访问。		
类型	R/W		

31	30	29	28	27	26	25	24	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																															

位	字段名称	说明	类型	复位
31-0	DATA	寄存器数据	R/W	—

2.5.2.9 Cortex® 通用寄存器 8 (R8)

表 2-11. Cortex® 通用寄存器 8 (R8)

地址偏移量		复位	-
物理地址		实例	
说明	R8 寄存器是用于数据操作的 32 位通用寄存器，可从特权模式或非特权模式进行访问。		
类型	R/W		

31	30	29	28	27	26	25	24	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																															

位	字段名称	说明	类型	复位
31-0	DATA	寄存器数据	R/W	—

2.5.2.10 Cortex® 通用寄存器 9 (R9)

表 2-12. Cortex® 通用寄存器 9 (R9)

地址偏移量		复位	-
物理地址		实例	
说明	R9 寄存器是用于数据操作的 32 位通用寄存器，可从特权模式或非特权模式进行访问。		
类型	R/W		

31	30	29	28	27	26	25	24	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																															

位	字段名称	说明	类型	复位
31-0	DATA	寄存器数据	R/W	—

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.5.2.11 Cortex® 通用寄存器 10 (R10)

表 2-13. Cortex® 通用寄存器 10 (R10)

地址偏移量		复位	-
物理地址		实例	
说明	R10 寄存器是用于数据操作的 32 位通用寄存器，可在特权模式或非特权模式下进行访问。		
类型	R/W		

31	30	29	28	27	26	25	24	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																															

位	字段名称	说明	类型	复位
31-0	DATA	寄存器数据	R/W	—

2.5.2.12 Cortex® 通用寄存器 11 (R11)

表 2-14. Cortex® 通用寄存器 11 (R11)

地址偏移量		复位	-
物理地址		实例	
说明	R11 寄存器是用于数据操作的 32 位通用寄存器，可从特权模式或非特权模式进行访问。		
类型	R/W		

31	30	29	28	27	26	25	24	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																															

位	字段名称	说明	类型	复位
31-0	DATA	寄存器数据	R/W	—

2.5.2.13 Cortex® 通用寄存器 12 (R12)

表 2-15. Cortex® 通用寄存器 12 (R12)

地址偏移量		复位	-
物理地址		实例	
说明	R12 寄存器是用于数据操作的 32 位通用寄存器，可从特权模式或非特权模式进行访问。		
类型	R/W		

31	30	29	28	27	26	25	24	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																															

位	字段名称	说明	类型	复位
31-0	DATA	寄存器数据	R/W	—

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.5.2.14 栈指针 (SP)

表 2-16. 栈指针 (SP)

地址偏移量		复位	-
物理地址		实例	
说明			
栈指针 (SP) 为寄存器 R13。在线程模式下，该寄存器的功能取决于控制寄存器 (CONTROL) 中的 ASP 位。当 ASP 位清零时，该寄存器为主栈指针 (MSP)。当 ASP 位置位时，该寄存器为进程栈指针 (PSP)。复位时，ASP 位清零，处理器从地址 0x0000 0000 加载 MSP 的值。MSP 只能在特权模式下访问；PSP 可以在特权或非特权模式下访问。			
类型	R/W		

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

位	字段名称	说明	类型	复位
31-0	SP	该字段是栈指针的地址。	R/W	—

2.5.2.15 链接寄存器 (LR)

表 2-17. 链接寄存器 (LR)

地址偏移量		复位	0xFFFF FFFF
物理地址		实例	
说明			
链接寄存器 (LR) 是寄存器 R14，可存储子例程、函数调用和异常的返回信息。可从特权模式或非特权模式访问 LR。			
EXC_RETURN 在进入异常时加载至 LR 中。有关值和说明，请参阅表 5-2。			
类型	R/W		

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

位	字段名称	说明	类型	复位
31-0	LINK	该字段是返回地址。	R/W	0xFFFF FFFF

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.5.2.16 程序计数器 (PC)

表 2-18. 程序计数器 (PC)

地址偏移量		复位	-
物理地址		实例	
说明			
程序计数器 (PC) 是寄存器 R15，包含当前程序地址。复位时，处理器将复位矢量的值 (地址为 0x0000 0004) 加载到 PC。复位矢量的位 0 在复位时加载到 EPSR 寄存器的 THUMB 位中，并且必须为 1。可以在特权模式或非特权模式下访问 PC 寄存器			
类型	R/W		

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

位	字段名称	说明	类型	复位
31-0	PC	此字段是当前程序地址。	R/W	—

2.5.2.17 程序状态寄存器 (PSR)

表 2-19. PSR 组合

寄存器	类型	组合
PSR	R/W (2) (1)	APSR、EPSR 和 IPSR
IEPSR	RO	EPSR 和 IPSR
IAPSR	R/W	APSR 和 IPSR
EAPSR	R/W	APSR 和 EPSR

(1) 处理器忽略对 IPSR 位的写入。
(2) 使用 MSR 指令返回 0 直接读取 EPSR 位，并且处理器忽略对这些位的写入。

表 2-20. 程序状态寄存器 (PSR) 或 (xPSR)

地址偏移量		复位	0x0100 0000
物理地址		实例	
说明			
也称为 xPSR			
程序状态寄存器 (PSR) 有三项功能，并且寄存器位分配给不同的功能： • 应用程序程序状态寄存器 (APSR)，位 31 - 27 • 执行程序状态寄存器 (EPSR)，位 26 - 24、15 - 10 • 中断程序状态寄存器 (IPSR)，位 6 - 0			
PSR、IPSR 和 EPSR 寄存器只能在特权模式下被访问；APSR 寄存器可以在特权模式或非特权模式下被访问。			
APSR 包含之前指令执行中条件标志的当前状态。			
EPSR 包含 Thumb 状态位和 if-then (IT) 指令的执行状态位，或用于中断加载多条指令或存储多条指令的可中断连续指令 (ICI) 字段。使用 MSR 指令通过应用软件直接读取 EPSR 的尝试始终返回 0。使用应用软件中的 MSR 指令写入 EPSR 的尝试始终会被忽略。故障处理程序可以检查堆叠 PSR 中的 EPSR 值，以确定发生故障的操作 (请参阅节 5.1.7) 。			
IPSR 包含当前 ISR 的异常类型编号。			
可以单独访问这些寄存器，也可以作为任意两个或全部三个寄存器的组合访问，并将寄存器名称用作 MSR 或 MRS 指令的参数。例如，可以使用 PSR 和 MRS 指令读取所有寄存器，或者只能使用 APSR 和 MSR 指令写入 APSR。表 2-20 显示了 PSR 的可能寄存器组合。有关如何访问程序状态寄存器的更多信息，请参阅 *Cortex-M3/M4F 指令集技术用户手册* 中的 MRS 和 MSR 指令说明。			
类型	R/W		

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

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

N	Z	C	V	Q	ICI / IT	THUMB	RESERVED	ICI / IT	RESERVED	ISRNUM

位	字段名称	说明		类型	复位
31	N	APSR 负或更小标志		R/W	0
		值	**说明**		
		1	之前的操作结果为负或者小于。		
		0	先前的运算结果为正、零、大于或者等于		
		仅当访问 PSR 或 APSR 时，该位的值才有意义。			
30	Z	APSR 零标志		R/W	0
		值	**说明**		
		1	先前的运算结果为零。		
		0	之前的运算结果为非零。		
		仅当访问 PSR 或 APSR 时，该位的值才有意义。			
29	C	APSR 进位或借位标志		R/W	0
		值	**说明**		
		1	前一个加法操作产生进位位，或者前一个减法操作没有产生借位位。		
		0	前一个加法操作不会产生进位位，或者前一个减法操作产生借位位。		
		仅当访问 PSR 或 APSR 时，该位的值才有意义。			
28	V	APSR 溢出标志		R/W	0
		值	**说明**		
		1	上一个操作导致溢出。		
		0	上一个操作没有导致溢出。		
		仅当访问 PSR 或 APSR 时，该位的值才有意义。			
27	Q	APSR 粘滞溢出及饱和标志		R/W	0
		值	**说明**		
		1	发生了溢出或饱和。（通过 SSAT 或 USAT 指令设置）。		
		0	自复位以来或者自上次清除该位以来未发生溢出或饱和。		
		仅当访问 PSR 或 APSR 时，该位的值才有意义。 此标志是粘滞标志，因为当由指令设置时，它会保持设置状态，直到使用 MSR 指令显式清除。			
26-25	ICI / IT	EPSR ICI / IT 状态 这些位与位 15:10 一起包含用于中断加载多次或存储多条指令或 IT 指令的执行状态位的 ICI 字段。当 EPSR 保持 ICI 执行状态时，位 26:25 为 0。If-Then 块在 IT 指令之后最多包含四条指令。块中的每条指令都是有条件的。指令的条件要么全部相同，要么某些条件与其他条件相反。如需更多信息，请参考 *Cortex-M3/M4F 指令集技术用户手册*。仅当访问 PSR 或 EPSR 时，该字段的值才有意义。		R/O	0x0

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

位	字段名称	说明		类型	复位
24	THUMB	EPSR Thumb 状态 此位表示 Thumb 状态，必须始终设置该位。以下指令可以清除 THUMB 位： • BLX、BX 和 POP{PC} 指令 • 在返回异常时从堆叠的 xPSR 值恢复 • 异常条目或复位时矢量值的位 0 在该位清除时尝试执行指令会导致故障或者锁定。有关更多信息，请参阅 节 5.2.4。仅当访问 PSR 或 EPSR 时，该位的值才有意义。		R/O	1
23-16	RESERVED	保留		R/O	0x00
15-10	ICI / IT	EPSR ICI / IT 状态 这些位与位 26:25 一起包含用于中断加载多次或存储多条指令或 IT 指令的执行状态位的 ICI 字段。在执行 LDM、STM、PUSH 或 POP 指令期间发生中断时，处理器会暂时停止加载多个指令或存储多个指令操作，并将多次操作中的下一个寄存器操作数存储到位 15:12。为中断提供服务后，处理器返回到位 15:12 指向的寄存器，并恢复多个加载或存储指令的执行。当 EPSR 保持 ICI 执行状态时，位 11:10 为 0。If-Then 块在 16 位 IT 指令之后最多包含四条指令。块中的每条指令都是有条件的。指令的条件要么全部相同，要么某些条件与其他条件相反。如需更多信息，请参考 *Cortex-M3/M4F 指令集技术用户手册*。仅当访问 PSR 或 EPSR 时，该字段的值才有意义。		R/O	0x0
9-7	RESERVED	软件不得依赖保留位的值。为了提供与未来产品的兼容性，必须在读-修改-写操作中留存保留位的值。		R/O	0x0
6-0	ISRNUM	IPSR ISR 编号 该字段包含当前 ISR 的异常类型编号。		R/O	0x00
		值	**说明**		
		0x00	线程模式		
		0x01	保留		
		0x02	NMI		
		0x03	硬故障		
		0x04	内存管理故障		
		0x05	总线故障		
		0x06	使用故障		
		0x07 – 0x0A	保留		
		0x0B	SVCall		
		0x0C	保留用于调试		
		0x0D	保留		
		0x0E	PendSV		
		0x0F	SysTick		
		0x10	中断矢量 0		
		0x11	中断矢量 1		
		...	...		
		0x31	中断矢量 33		
		0x32-0x7F	保留		
		有关更多信息，请参阅 节 5.1.2。 仅当访问 PSR 或 IPSR 时，该字段的值才有意义。			

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.5.2.18 优先级屏蔽寄存器 (PRIMASK)

表 2-21. 优先级屏蔽寄存器 (PRIMASK)

地址偏移量		复位	0x0000 0000
物理地址		实例	
说明			
优先级屏蔽 (PRIMASK) 寄存器可防止激活具有可编程优先级的全部异常。复位、不可屏蔽中断 (NMI) 及硬故障是具有固定优先级的唯一例外情况。当异常可能影响关键任务的时间时，必须禁用这些异常。该寄存器仅在特权模式下可访问。MSR 和 MRS 指令用于访问 PRIMASK 寄存器，并且 CPS 指令可用于更改 PRIMASK 寄存器的值。有关这些指令更多信息，请参阅 *Cortex-M3/M4F 指令集技术用户手册*。有关异常优先级的更多信息、请参阅节 5.1.2。			
类型	R/W		

31	30	29	28	27	26	25	24	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																															PRIMASK

位	字段名称	说明		类型	复位
31-1	RESERVED	软件不得依赖保留位的值。为了提供与未来产品的兼容性，必须在读-修改-写操作中留存保留位的值。		R/O	0x0000 000
0	PRIMASK	优先级屏蔽		R/W	0
		值	**说明**		
		1	阻止激活所有具有可配置优先级的异常		
		0	没有影响		

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.5.2.19 故障屏蔽寄存器 (FAULTMASK)

表 2-22. 故障屏蔽寄存器 (FAULTMASK)

地址偏移量		复位	0x0000 0000
物理地址		实例	
说明			
故障屏蔽 FAULTMASK 寄存器可防止激活除 NMI 之外的所有异常。当异常可能影响关键任务的时序时，必须将其禁用。该寄存器只能在特权模式下访问。MSR 和 MRS 指令用于访问 FAULTMASK 寄存器，而 CPS 指令可用于更改 FAULTMASK 寄存器的值。有关这些指令的更多信息，请参阅 *Cortex-M3/M4F 指令集技术用户手册*。有关异常优先级数值的更多信息，请参阅节 5.1.2。			
类型	R/W		

31	30	29	28	27	26	25	24	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																															FAULTMASK

位	字段名称	说明		类型	复位
31-1	RESERVED	保留		R/O	0x0000 000
0	FAULTMASK	故障屏蔽		R/W	0
		值	**说明**		
		1	防止激活除 NMI 之外的所有异常		
		0	没有影响		
		处理器在退出除 NMI 处理程序之外的任何异常处理程序时，会清除 FAULTMASK 位。			

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.5.2.20 基址优先级屏蔽寄存器 (BASEPRI)

表 2-23. 基址优先级屏蔽寄存器 (BASEPRI)

地址偏移量		复位	0x0000 0000
物理地址		实例	
说明			
基址优先级屏蔽寄存器 BASEPRI 定义了异常处理的最低优先级。将 BASEPRI 设置为非零值时，它会阻止激活所有优先级数值与 BASEPRI 值相同或更低的异常。当异常可能影响关键任务的时序时，必须将其禁用。该寄存器只能在特权模式下访问。有关异常优先级数值的更多信息，请参阅节 5.1.2。			
类型	R/W		

31	30	29	28	27	26	25	24	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																								BASEPRI			RESERVED				

位	字段名称	说明		类型	复位
31-8	RESERVED	保留		R/O	0x0000 00
7-5	BASEPRI	基址优先级 任何具有可编程优先级且优先级数值与此字段值相同或更低的异常都将被屏蔽。PRIMASK 寄存器可用于屏蔽所有具有可编程优先级数值的异常。异常的优先级越高，其优先级数值越低。		R/W	0x0
		值	**说明**		
		0x0	所有异常均未屏蔽。		
		0x1	优先级数值为 1-7 的所有异常都被屏蔽。		
		0x2	优先级数值为 2-7 的所有异常都被屏蔽。		
		0x3	优先级数值为 3-7 的所有异常都被屏蔽。		
		0x4	优先级数值为 4-7 的所有异常都被屏蔽。		
		0x5	优先级数值为 5-7 的所有异常都被屏蔽。		
		0x6	优先级数值为 6 和 7 的所有异常都被屏蔽。		
		0x7	优先级数值为 7 的所有异常都被屏蔽。		
4-0	RESERVED	保留		R/O	0x0

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.5.2.21 控制寄存器 (CONTROL)

表 2-24. 控制寄存器 (CONTROL)

地址偏移量		复位	0x0000 0000
物理地址		实例	
说明			
当处理器处于线程模式时，CONTROL 寄存器控制所使用的栈和软件执行的权限级别。该寄存器仅在特权模式下可访问。			
处理程序模式始终使用 MSP，因此在处理程序模式下处理器会忽略对 CONTROL 寄存器 ASP 位的显式写入。异常进入和返回机制会根据 EXC_RETURN 值自动更新 CONTROL 寄存器（请参阅 表 5-2）。在 OS 环境中，以线程模式运行的线程必须使用进程栈，内核和异常处理程序必须使用主栈。默认情况下，线程模式使用 MSP。要将线程模式中使用的栈指针切换到 PSP，可以使用 MSR 指令设置 ASP 位（详见 *Cortex-M3/M4F 指令集技术用户手册*），也可以使用相应的 EXC_RETURN 值执行异常返回线程模式，如表 5-2 中所示。			
更改栈指针时，软件必须在 MSR 指令之后立即使用 ISB 指令，以确保 ISB 指令执行后的指令使用新的栈指针。请参考 *Cortex-M3/M4F 指令集技术用户手册*。			
类型	R/W		

31	30	29	28	27	26	25	24	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																														ASP	TMPL

位	字段名称	说明		类型	复位
31-2	RESERVED	保留		R/O	0x0000 000
1	ASP	活动栈指针		R/W	0
		值	**说明**		
		1	PSP 是当前的栈指针。		
		0	MSP 是当前栈指针。		
		在处理程序模式下，此位读取为零并忽略写入。Arm Cortex-M4F 在异常返回时自动更新此位。			
0	TMPL	线程模式特权级别		R/W	0
		值	**说明**		
		1	非特权软件可在线程模式下执行。		
		0	在线程模式下只能执行特权软件。		

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.6 指令集概要

本节提供以下信息：

- Arm® Cortex®-M4F 指令（请参阅节 2.6.1）
- 加载与存储时序（请参阅节 2.6.2）
- 与其他 Cortex® 处理器的二进制兼容性（请参阅节 2.6.3）

2.6.1 Arm® Cortex®-M4F 指令

处理器实现 ARMv7-M Thumb 指令集。表 2-25 列出了 Arm® Cortex®-M4F 指令及其周期数。周期计数基于具有零等待状态的系统。

在汇编器语法中，根据操作的不同，<op2> 字段可替换为以下选项之一：

- 例如简单的寄存器说明符

```
Rm
```

- 例如立即移位寄存器

```
Rm, LSL #4
```

- 例如寄存器移位寄存器

```
Rm, LSL Rs
```

- 例如一个直接值

```
#0xE000E000
```

为简洁起见，并未显示所有加载和存储寻址模式。有关更多信息，请参阅 *ARMv7-M 架构参考手册*。

表 2-25 在*周期*列中使用以下缩写：

P 流水线补充所需的周期数。这一范围为 1 至 3，具体取决于目标指令的对齐方式和宽度，以及处理器是否设法提前推测地址。

B 执行屏障操作所需的循环数。对于 DSB 和 DMB，最小周期数为零。对于 ISB，最小周期数等于流水线补充所需的数量。

N 要加载或存储的寄存器列表中的寄存器数量，包括 PC 或 LR。

W 等待相应事件所花费的周期数。

表 2-25. Arm® Cortex®-M4F 指令集摘要

运行	说明	汇编器	周期
移动	寄存器	`MOV Rd, <op2>`	1
	16 位立即	`MOVW Rd, #<imm>`	1
	立即进入顶部	`MOVT Rd, #<imm>`	1
	至 PC	`MOV PC, Rm`	1 + P
向	向	`ADD Rd, Rn, <op2>`	1
	添加到 PC	`ADD PC, PC, Rm`	1 + P
	随携带添加	`ADC Rd, Rn, <op2>`	1
	表单地址	`ADR Rd, <label>`	1

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 2-25. Arm® Cortex®-M4F 指令集摘要（续）

运行	说明	汇编器	周期
减法	减法	SUB Rd, Rn, <op2>	1
	用借位减去	SBC Rd, Rn, <op2>	1
	反向	RSB Rd, Rn, <op2>	1
乘法	乘法	MUL Rd, Rn, Rm	1
	乘累加	MLA Rd, Rn, Rm	1
	乘减	MLS Rd, Rn, Rm	1
	有符号长整型	SMULL RdLo, RdHi, Rn, Rm	1
	无符号长整型	UMULL RdLo, RdHi, Rn, Rm	1
	有符号长整型累加器	SMLAL RdLo, RdHi, Rn, Rm	1
	无符号长整型累加器	UMLAL RdLo, RdHi, Rn, Rm	1
分频	有符号	SDIV Rd, Rn, Rm	2 至 12(1)
	无符号	UDIV Rd, Rn, Rm	2 至 12(1)
饱和	有符号	SSAT Rd, #<imm>, <op2>	1
	无符号	USAT Rd, #<imm>, <op2>	1
比较	比较	CMP Rn, <op2>	1
	负	CMN Rn, <op2>	1
逻辑	和	AND Rd, Rn, <op2>	1
	异或	EOR Rd, Rn, <op2>	1
	或	ORR Rd, Rn, <op2>	1
	或非	ORN Rd, Rn, <op2>	1
	位清零	BIC Rd, Rn, <op2>	1
	不移动	MVN Rd, <op2>	1
	和测试	TST Rn, <op2>	1
	异或测试	TEQ Rn, <op1>	1
移位	逻辑左移	LSL Rd, Rn, #<imm>	1
	逻辑左移	LSL Rd, Rn, Rs	1
	逻辑右移	LSR Rd, Rn, #<imm>	1
	逻辑右移	LSR Rd, Rn, Rs	1
	算术右移	ASR Rd, Rn, #<imm>	1
	算术右移	ASR Rd, Rn, Rs	1
旋转	向右旋转	ROR Rd, Rn, #<imm>	1
	向右旋转	ROR Rd, Rn, Rs	1
	带延长	RRX Rd, Rn	1
数量	前导零	CLZ Rd, Rn	1

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 2-25. Arm® Cortex®-M4F 指令集摘要 （续）

运行	说明	汇编器	周期
负载	字	LDR Rd, [Rn, <op2>]	2[2]
	至 PC	LDR PC, [Rn, <op2>]	2[2] + P
	半字	LDRH Rd, [Rn, <op2>]	2[2]
	字节	LDRB Rd, [Rn, <op2>]	2[2]
	有符号半字	LDRSH Rd, [Rn, <op2>]	2[2]
	有符号字节	LDRSB Rd, [Rn, <op2>]	2[2]
	用户字	LDRT Rd, [Rn, #<imm>]	2[2]
	用户半字	LDRHT Rd, [Rn, #<imm>]	2[2]
	用户字节	LDRBT Rd, [Rn, #<imm>]	2[2]
	用户签名的半字	LDRSHT Rd, [Rn, #<imm>]	2[2]
	用户签名字节	LDRSBT Rd, [Rn, #<imm>]	2[2]
	PC 相对	LDR Rd,[PC, #<imm>]	2[2]
	双字	LDRD Rd, Rd, [Rn, #<imm>]	1 + N
	多个	LDM Rn, {<reglist>}	1 + N
	多个，包括 PC	LDM Rn, {<reglist>, PC}	1 + N + P
存储	字	STR Rd, [Rn, <op2>]	2[2]
	半字	STRH Rd, [Rn, <op2>]	2[2]
	字节	STRB Rd, [Rn, <op2>]	2[2]
	有符号半字	STRSH Rd, [Rn, <op2>]	2[2]
	有符号字节	STRSB Rd, [Rn, <op2>]	2[2]
	用户字	STRT Rd, [Rn, #<imm>]	2[2]
	用户半字	STRHT Rd, [Rn, #<imm>]	2[2]
	用户字节	STRBT Rd, [Rn, #<imm>]	2[2]
	用户签名的半字	STRSHT Rd, [Rn, #<imm>]	2[2]
	用户签名字节	STRSBT Rd, [Rn, #<imm>]	2[2]
	双字	STRD Rd, Rd, [Rn, #<imm>]	1 + N
	多个	STM Rn, {<reglist>}	1 + N
推入	推入	PUSH {<reglist>}	1 + N
	带连接寄存器的推入	PUSH {<reglist>, LR}	1 + N
弹出	弹出	POP {<reglist>}	1 + N
	弹出并返回	POP {<reglist>, PC}	1 + N + P

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 2-25. Arm® Cortex®-M4F 指令集摘要（续）

运行	说明	汇编器	周期
信标	加载独占项	`LDREX Rd, [Rn, #<imm>]`	2
	加载独占半字	`LDREXH Rd, [Rn]`	2
	加载独占字节	`LDREXB Rd, [Rn]`	2
	存储独占	`STREX Rd, Rt, [Rn,#<imm>]`	2
	存储独占半字	`STREXH Rd, Rt, [Rn]`	2
	存储独占字节	`STREXB Rd, Rt, [Rn]`	2
	清除独占监视器	`CLREX`	1
转移	条件	`B<cc> <label>`	1 或 1 + P(3)
	无条件	`B <label>`	1 + P
	带链接	`BL <label>`	1 + P
	带交换	`BX Rm`	1 + P
	带链路和交换	`BLX Rm`	1 + P
	如果为零，则转移	`CBZ Rn, <label>`	1 或 1 + P(3)
	如果不为零，则转移	`CBNZ Rn, <label>`	1 或 1 + P(3)
	字节表转移	`TBB [Rn, Rm]`	2 + P
	半字表转移	`TBH [Rn, Rm, LSL#1]`	2 + P
状态变化	管理程序调用	`SVC #<imm>`	-
	If-then-else	`IT... <cond>`	1(4)
	禁用中断	`CPSID <flags>`	1 或 2
	启用中断	`CPSIE <flags>`	1 或 2
	读取特殊寄存器	`MRS Rd, <specreg>`	1 或 2
	写入特殊寄存器	`MSR <specreg>, Rn`	1 或 2
	Breakpoint	`BKPT #<imm>`	-
扩展	有符号半字到字	`SXTH Rd, <op2>`	1
	有符号字节到字	`SXTB Rd, <op2>`	1
	无符号半字	`UXTH Rd, <op2>`	1
	无符号字节	`UXTB Rd, <op2>`	1
位字段	提取无符号	`UBFX Rd, Rn, #<imm>, #<imm>`	1
	提取有符号	`SBFX Rd, Rn, #<imm>, #<imm>`	1
	清除	`BFC Rd, Rn, #<imm>, #<imm>`	1
	插入	`BFI Rd, Rn, #<imm>, #<imm>`	1
反向	字中的字节	`REV Rd, Rm`	1
	两个半字中的字节	`REV16 Rd, Rm`	1
	有符号底部半字	`REVSH Rd, Rm`	1
	字中的位	`RBIT Rd, Rm`	1

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 2-25. Arm® Cortex®-M4F 指令集摘要（续）

运行	说明	汇编器	周期
提示	发送事件	SEV	1
	等待事件	WFE	1 + W
	等待中断	WFI	1 + W
	无操作	NOP	1
栅	指令同步	ISB	1 + B
	数据存储器	DMB	1 + B
	数据同步	DSB <flags>	1 + B

(1) 除法运算在除法计算完成时终止，所需的周期数取决于输入操作数的值。除法运算是可中断的，这意味着当发生中断时，操作可以被放弃，最坏情况下的延迟为一个周期，并在中断完成时重新启动。
(2) 相邻的加载和存储单个指令可以对其地址和数据阶段进行流水线处理，但在某些情况下（例如在奇半字边界上对齐的 32 位操作码），它们可能无法以最佳方式进行流水线处理。
(3) 如果未采用转移，条件转移将在单个周期内完成。
(4) IT 指令可以折叠到前面的 16 位 Thumb 指令上，从而能够在零周期内执行。

表 2-26 列出了 Arm® Cortex®-M4F 处理器实现的 DSP 指令。

表 2-26. Arm® Cortex®-M4F DSP 指令集摘要

运行	说明	汇编器	周期
乘法	32 位乘法加 32 个最高有效位累加	SMMLA	1
	32 位乘法与 32 个最高有效位相减	SMMLS	1
	32 位乘法会返回 32 个最高有效位	SMMUL	1
	32 位乘法，其中四舍五入的 32 个最高有效位累加	SMMLAR	1
	32 位乘法，其中四舍五入的 32 个最高有效位减法	SMMLSR	1
	32 位乘法返回四舍五入的 32 个最高有效位	SMMULR	1

Copyright © 2026 Texas Instruments Incorporated

表 2-26. Arm® Cortex®-M4F DSP 指令集摘要（续）

运行	说明	汇编器	周期
有符号的乘法	Q 设置 16 位有符号乘法，并进行 32 位累加，低乘低	SMLABB	1
	Q 设置 16 位有符号乘法与 32 位累加，低乘高	SMLABT	1
	16 位有符号乘法，并进行 64 位累加，低乘低	SMLALBB	1
	16 位有符号乘法与 64 位累加，低乘高	SMLALBT	1
	双 16 位有符号乘法，采用单个 64 位累加器	SMLALD{X}	1
	16 位有符号乘法，并进行 64 位累加，高乘低	SMLALTB	1
	16 位有符号乘法与 64 位累加，高乘高	SMLALTT	1
	16 位有符号乘法，生成 32 位结果，低乘低	SMULBB	1
	16 位有符号乘法，生成 32 位结果，低乘高	SMULBT	1
	16 位有符号乘法，生成 32 位结果，高乘低	SMULTB	1
	16 位有符号乘法，生成 32 位结果，高乘高	SMULTT	1
	16 位乘以 32 位有符号乘法，返回 32 个最高有效位，低	SMULWB	1
	16 位乘以 32 位有符号乘法，返回 32 个最高有效位，高	SMULWT	1
	双通道 16 位有符号乘法返回差异	SMUSD{X}	1
	Q 设置 16 位有符号乘法，累加 32 位，高乘低	SMLATB	1
	Q 设置 16 位有符号乘法，累加 32 位，高乘高	SMLATT	1
	Q 设置双 16 位有符号乘法，采用单个 32 位累加器	SMLAD{X}	1
	Q 设置 16 位乘 32 位有符号乘法，累加 32 位，低	SMLAWB	1
	Q 设置 16 位乘 32 位有符号乘法，累加 32 位，高	SMLAWT	1
	Q 设置双 16 位有符号乘减，累加 32 位	SMLSD{X}	1
	Q 设置双 16 位有符号乘减，采用 64 位累加器	SMLSLD{X}	1
	Q 设置双路 16 位有符号乘法的总和	SMUAD{X}	1
无符号的乘法	32 位无符号乘法，双精度 32 位累加，生成 64 位结果	UMAAL	1
饱和	Q 设置双 16 位饱和	SSAT16	1
	Q 设置双 16 位无符号饱和	USAT16	1

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 2-26. Arm® Cortex®-M4F DSP 指令集摘要（续）

运行	说明	汇编器	周期
紧凑和非紧凑	将移位后的低半字打包至高半字位置	PKHTB	
	将移位后的高半字打包至低半字位置	PKHBT	1
	提取 8 位，并将符号扩展到 32 位	SXTB	1
	双提取 8 位，并将每个符号扩展到 16 位	SXTB16	1
	提取 16 位，并将符号扩展到 32 位	SXTH	1
	提取 8 位，并将零扩展到 32 位	UXTB	1
	双路提取 8 位，并将零扩展到 16 位	UXTB16	1
	提取 16 位，并将零扩展到 32 位	UXTH	1
	提取 8 位至 32 位无符号加法	UXTAB	1
	双提取 8 位至 16 位无符号加法	UXTAB16	1
	提取的 16 位至 32 位无符号加法	UXTAH	1
	提取的 8 位至 32 位有符号加法	SXTAB	1
	双路提取 8 位至 16 位有符号加法	SXTAB16	1
	提取的 16 位至 32 位有符号加法	SXTAH	1
其他数据处理	根据 GE 位选择字节	SEL	1
	四通道 8 位无符号绝对差值的无符号和	USAD8	1
	32 位累加时四路 8 位无符号绝对差的无符号和	USADA8	1
加法	两路 16 位无符号饱和加法	UQADD16	1
	四路 8 位无符号饱和加法	UQADD8	1
	Q 设置饱和加法	QADD	1
	Q 设置两路 16 位饱和加法	QADD16	1
	Q 设置四路 8 位饱和加法	QADD8	1
	Q 设置饱和倍乘和加法	QDADD	1
	GE 设置四路 8 位有符号加法	SADD8	1
	GE 设置双 16 位有符号加法	SADD16	1
	两路 16 位有符号加法，结果减半	SHADD16	1
	四路 8 位有符号加法，结果减半	SHADD8	1
	GE 设置双 16 位无符号加法	UADD16	1
	GE 设置四路 8 位无符号加法	UADD8	1
	双 16 位无符号加法，结果减半	UHADD16	1
	四路 8 位无符号加法，结果减半	UHADD8	1

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 2-26. Arm® Cortex®-M4F DSP 指令集摘要（续）

运行	说明	汇编器	周期
减法	Q 设置饱和倍减	QDSUB	1
	两路 16 位无符号饱和减法	UQSUB16	1
	四路 8 位无符号饱和减法	UQSUB8	1
	Q 设置饱和减法	QSUB	1
	Q 设置两路 16 位饱和减法	QSUB16	1
	Q 设置四路 8 位饱和减法	QSUB8	1
	两路 16 位有符号减法，结果减半	SHSUB16	1
	四路 8 位有符号减法，结果减半	SHSUB8	1
	GE 设置两路 16 位有符号减法	SSUB16	1
	GE 设置四路 8 位有符号减法	SSUB8	1
	两路 16 位无符号减法，结果减半	UHSUB16	1
	四路 8 位无符号减法，结果减半	UHSUB8	1
	GE 设置两路 16 位无符号减法	USUB16	1
	GE 设置四路 8 位无符号减法	USUB8	1
并行加法和减法	具有交换的双路 16 位无符号饱和加法和减法	UQASX	1
	具有交换的双路 16 位无符号饱和减法和加法	UQSAX	1
	GE 设置带交换的双路 16 位加法和减法	SASX	1
	Q 设置双路 16 位加法并使用交换	QASX	1
	Q 设置双路 16 位减法与加法（通过交换）	QSAX	1
	两路 16 位有符号加法和减法，结果减半	SHASX	1
	两路 16 位有符号减法与加法，结果减半	SHSAX	1
	GE 设置具有交换功能的双 16 位有符号减法和加法	SSAX	1
	GE 设置带交换的双 16 位无符号加法和减法	UASX	1
	两路 16 位无符号加法和减法，可实现结果减半和交换	UHASX	1
	双 16 位无符号减法与加法，可得到一半的结果和交换	UHSAX	1
	GE 设置双路 16 位无符号减法与加法（通过交换）	USAX	1

2.6.2 加载和存储时序

本节介绍如何以最佳方式配对指令，以进一步减少时序开销。

- STR Rx,[Ry,#imm] 始终为一个周期。这是因为地址生成在初始周期中执行，而数据存储操作是与下一条指令的执行同时进行。如果存储操作的目标是写入缓冲区，且写入缓冲区已满或未使能，则下一条指令将被延迟，直至该存储操作完成。如果存储操作的目标不是写入缓冲区（例如存储到代码段），并且该传输发生阻塞，则只有在本次存储操作完成之前执行另一个加载或存储操作时，才会对时序产生明显影响。
- LDR PC,[any] 始终是一个阻塞操作。这意味着加载至少需要两个周期，流水线重新加载至少需要三个周期。因此，该操作至少需要五个周期，如果在加载或获取时停止，则需要更多周期。
- 如果任何加载或存储产生的地址取决于先前数据处理操作的结果，则在寄存器组更新期间，流水线会额外暂停一个周期。这种情况下，没有转发路径。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

- LDR Rx,[PC,#IMM] 可能会由于与获取单元的争用而增加一个周期。
- TBB 和 TBH 也是阻塞操作。加载至少需要两个周期，加法至少需要一个周期，流水线重新加载至少需要三个周期。这意味着如果在加载或获取时停止，则至少需要六个或更多周期。
- 如果可能，LDR [any] 会以流水线方式执行。这意味着，如果下一条指令是 LDR 或 STR，并且第一条 LDR 指令的目标未用于计算下一条指令的地址，则从下一条指令的开销中删除一个周期。因此，LDR 后面可能跟一条 STR，以便 STR 将 LDR 加载的数据写入内存。多个 LDR 指令可以一起以流水线方式执行。以下是一些经优化的示例：
  - LDR R0,[R1]；LDR R1,[R2]：通常总共三个周期
  - LDR R0,[R1,R2]；STR R0,[R3,#20]：通常总共三个周期
  - LDR R0,[R1,R2]；STR R1,[R3,R2]：通常总共三个周期
  - LDR R0,[R1,R5]；LDR R1,[R2]；LDR R2,[R3,#4]：通常总共四个周期
- 在带寄存器偏移的 STR 指令执行之后，其他指令无法以流水线方式执行。STR 指令只有在紧跟在 LDR 指令之后时才能以流水线方式执行，但在该存储操作之后无法再流水线执行任何指令。因为存在写入缓冲区，即使发生阻塞，STR 通常也只需要两个周期。
- LDREX 和 STREX 可像 LDR 一样以流水线方式执行。由于 STREX 的处理方式更接近于 LDR，因此可以按照 LDR 的说明进行流水线处理。同样，LDREX 被视为与 LDR 完全相同，因此也可以流水线言方式处理。
- LDRD 和 STRD 无法与前面或后面的指令一起进行流水线处理。但是，这两个字可以一起以流水线方式处理。因此，在未发生阻塞时，该操作需要三个周期。
- LDM 和 STM 无法与前面或后面的指令一起进行流水线处理。但是，第一个元素之后的所有元素都是一起流水线执行。因此，在未发生阻塞时，三个元素的 LDM 需要 2 + 1 + 1 或 5 个周期。同样，在未发生阻塞的情况下，八个元素的存储指令需要九个周期。中断时，LDM 和 STM 指令从返回时中断的位置继续。当启动时，继续操作会为第一个元素添加一个或两个循环。
- 未对齐的字或半字加载或存储会增加罚分周期。按字节对齐的半字加载或存储操作会额外增加一个周期，以作为两个字节的操作来执行。按半字对齐的字加载或存储会额外增加一个周期，以作为两个半字的操作来执行。按字节对齐的字加载或存储会额外增加两个周期，以作为一个字节、一个半字和一个字的操作来执行。如果内存发生阻塞，这些数字还会增加。STR 或 STRH 不能由于写入缓冲区的原因而延迟处理器。

2.6.3 与其他 Cortex® 处理器实现二进制兼容

该处理器实现了 ARMv7-M 架构配置文件提供的指令集和特性的子集，并且与在其他 Cortex®-M 配置文件处理器中实现的指令集和特性实现了二进制兼容。您可以将软件（包括系统级软件）从 Arm® Cortex®-M4F 处理器移动到其他 Cortex®-M 配置文件处理器。

为了确保平稳过渡，Arm® 建议设计用于在其他 Cortex®-M 配置文件处理器架构上运行的代码遵循以下规则，并适当地设定配置和控制寄存器 (CCR)：

- 只能使用字传输来访问 NVIC 及系统控制空间 (SCS) 中的寄存器。
- 将处理器上所有未使用的 SCS 寄存器及寄存器字段视为“不可修改”。
- 在 CCR 中配置以下字段：
  - 将 STKALIGN 位设置为 1
  - 将 UNALIGN_TRP 位设置为 1
  - 将 CCR 寄存器中所有其他位保留为其原始值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.7 浮点单元 (FPU)

本节介绍浮点单元 (FPU) 的编程器模型，包含有关以下内容的信息：

- 关于 FPU (请参阅 节 2.7.1)
- FPU 功能说明 (请参阅 节 2.7.2)
- FPU 编程器模型 (请参阅 节 2.7.3)

2.7.1 关于 FPU

Arm® Cortex®-M4F FPU 是 ARMv7-M 浮点扩展 (FPv4-SP) 单精度型号的一种实现。它提供了符合 ANSI/IEEE Std 754-2008《IEEE 二进制浮点运算标准》 (简称 IEEE 754 标准) 的浮点计算功能。该 FPU 支持 *ARMv7-M 架构参考手册*中描述的所有单精度数据处理指令和数据类型。

2.7.2 FPU 功能说明

FPU 完全支持单精度加、减、乘、除、乘累加及平方根运算。它还提供定点与浮点数据格式之间的转换，以及浮点常数指令。

FPU 功能说明包括以下主题：

- 寄存器组的 FPU 视图 (请参阅节 2.7.2.1)
- 工作模式 (请参阅节 2.7.2.2)
- FPU 指令集 (请参阅节 2.7.2.3)
- IEEE 754 标准合规性 (请参阅节 2.7.2.4)
- IEEE 754 标准的完整实现 (请参阅节 2.7.2.5)
- IEEE 754 标准实现选择 (请参阅节 2.7.2.6)
- 异常 (请参阅节 2.7.2.7)
- 启用 FPU (请参阅节 2.7.3.1)

2.7.2.1 FPU 寄存器组的视图

FPU 提供了一个包含 32 个单精度寄存器的扩展寄存器文件 (请参见图 2-4)。这些寄存器可以看作：

- 十六个 64 位双字寄存器，D0 – D15。
- 三十二个 32 位单字寄存器，S0 – S31。
- 这些视图中的寄存器的组合：

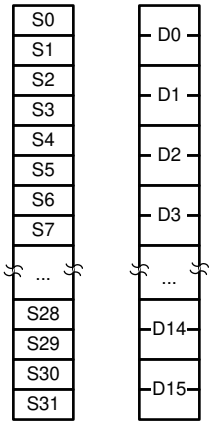


图 2-4. FPU 寄存器组

寄存器之间的映射关系如下：

- S<2n> 映射到 D<n> 的最低有效半部分。
- S<2n+1> 映射到 D<n> 的最高有效半部分。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

例如，您可以通过访问 S12 来访问 D6 中值的最低有效半部分，并通过访问 S13 来访问其最高有效半部分。

2.7.2.2 运行模式

FPU 提供三种运行模式来满足各种应用的需求：

- 完全合规模式
- 平齐到零模式
- 默认 NaN 模式

2.7.2.2.1 完全合规模式

在完全合规模式下，FPU 将在硬件中按照 IEEE 754 标准处理所有操作。

2.7.2.2.2 清零模式

设置浮点状态和控制寄存器 FPSCR[24] 的 FZ 位会启用清零模式。在该模式下，FPU 会将算术 CDP 操作的所有次正规输入操作数当作零处理。会针对零操作数产生的异常相应地发出信号。VABS、VNEG 和 VMOV 不被视为算术 CDP 运算，不受清零模式影响。根据 IEEE 754 标准，对于极小的结果（在四舍五入前，目标精度的幅值小于最小规范值），该结果将被替换为零。IDC 标志 FPSCR[7] 指示何时发生输入刷新。UFC 标志 FPSCR[3] 指示何时发生结果刷新。

2.7.2.2.3 默认 NaN 模式

设置 DN 位 FPSCR[25] 将启用默认 NaN 模式。在该模式中，涉及输入 NaN 或生成 NaN 结果的任何算术数据处理操作的结果都会返回默认 NaN。小数位的传播仅通过 VABS、VNEG 及 VMOV 运算来保持。所有其他 CDP 操作都会忽略输入 NaN 的小数位中的任何信息。

2.7.2.3 FPU 指令集

表 2-27 列出了 FPU 的指令集。

表 2-27. FPU 指令集

运行	说明	汇编器	周期
浮点数的	绝对值	VABS.F32	1
加法	浮点	VADD.F32	1
比较	浮点数与寄存器或零	VCMP.F32	1
	浮点数与寄存器或零	VCMPE.F32	1
转换	整数、定点数、半精度浮点数和浮点数之间	VCVT.F32	1
除法	浮点	VDIV.F32	14
负载	多个双精度浮点数	VLDM.64	1 + 2 × N，其中 N 是双精度浮点数的数量。
	多个浮点数	VLDM.32	1 + N，其中 N 是浮点数量。
	单个双精度浮点数	VLDR.64	3
	单个浮点数	VLDR.32	2
移动	双精度浮点数的上半部分/下半部分与内核寄存器之间	VMOV	1
	立即数/浮点数到浮点寄存器	VMOV	1
	两个浮点数/一个双精度浮点数与两个内核寄存器之间，或一个浮点数与一个内核寄存器之间	VMOV	2
	浮点控制/状态到内核寄存器	VMRS	1
	内核寄存器到浮点控制/状态	VMSR	1

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 2-27. FPU 指令集（续）

运行	说明	汇编器	周期
乘法	浮点数	VMUL.F32	1
	然后累加浮点数	VMLA.F32	3
	然后减去浮点数	VMLS.F32	3
	然后累加，再取浮点数的相反数	VNMLA.F32	3
	然后减去浮点数，再取浮点数的相反数	VNMLS.F32	3
乘法（融合）	然后累加浮点数	VFMA.F32	3
	然后减去浮点数	VFMS.F32	3
	然后累加，再取浮点数的相反数	VFNMA.F32	3
	然后减去浮点数，再取浮点数的相反数	VFNMS.F32	3
取	浮点数的相反数	VNEG.F32	1
	然后乘浮点数	VNMUL.F32	1
弹出	从堆栈中弹出双精度寄存器	VPOP.64	1 + 2 × N，其中 N 是双精度寄存器的数量
	从堆栈中弹出浮点寄存器	VPOP.32	1 + N，其中 N 是寄存器的数量
推入	双精度寄存器至堆栈	VPUSH.64	1 + 2 × N，其中 N 是双精度寄存器的数量
	浮点寄存器至堆栈	VPUSH.32	1 + N，其中 N 是寄存器的数量
浮点数的	平方根	VSQRT.F32	14
存储	多个双精度寄存器	VSTM.64	1 + 2 × N，其中 N 是双精度寄存器的数量
	多个浮点寄存器	VSTM.32	1 + N，其中 N 是浮点数量
	单个双精度寄存器	VSTR.64	3
	单个浮点寄存器	VSTR.32	2
减法	浮点数	VSUB.F32	1

备注

- 紧随 VDIVR 或 VSQRT 指令之后的纯整数指令以乱序方式完成。如果没有更多浮点指令需要执行，VDIV 和 VSQRT 指令仅需一个周期。
- 浮点算术数据处理指令（如加法、减法、乘法、除法、平方根、所有形式的乘法与累加以及各种类型的转换）的结果如果被以下指令使用，则需要多花费一个周期。
- 融合乘加和链式乘加指令都会晚一个周期才使用其加数，因此，若算术指令后面紧跟着一条乘加指令，则该算术指令的运算结果将作为该 MAC 指令的加数使用。

2.7.2.4 符合 IEEE 754 标准

禁用默认 NaN (DN) 和清空到零 (FZ) 模式时，FPv4 功能在硬件中符合 IEEE 754 标准。无需支持代码即可实现此合规性。

有关 FP 架构符合 IEEE 754 标准的信息，请参阅 *ARMv7-M 架构参考手册*。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.7.2.5 完全实现 IEEE 754 标准

浮点指令集不支持 IEEE 754-2008 标准中定义的全部运算。不支持的运算包括但不限于：

- 余数
- 将浮点数舍入为整数值浮点数
- 二进制到十进制转换
- 十进制到二进制转换
- 直接比较单精度值和双精度值

Arm® Cortex®-M4F FPU 支持 IEEE 标准中所述的融合 MAC 操作。为了完全实现 IEEE 754-2008 标准，必须通过库函数增强浮点功能。

2.7.2.6 IEEE 754 标准实现选择

*ARMv7-M 架构参考手册*中介绍了 IEEE 754-2008 标准允许并在 FPv4 架构中使用的一些实现选择。

2.7.2.6.1 NaN 处理

所有具有最大指数字段值和非零分数字段值的单精度值都是有效的 NaN。最高有效分数位为零表示信令 NaN (SNaN)，为一则表示静默 NaN (QNaN)。如果两个 NaN 值在任何位上都不同，则被视为不同的 NaN。表 2-28 显示了默认的 NaN 值。

表 2-28. 默认 NaN 值

符号	分数	分数
0	0xFF	位 [22] = 1，位 [21:0] 全为零

对于 Arm® 浮点功能和库，输入 NaN 的处理定义如下：

- 在完全合规模式下，NaN 的处理如 *ARMv7-M 架构参考手册* 所述。硬件直接为算术 CDP 指令处理 NaN。对于数据传输操作，会传输 NaN 而引发无效操作异常。对于非算术 CDP 指令（VABS、VNEG 和 VMOV），会复制 NaN（如果指令指定了符号改变则同时改变符号），而不会引起无效操作异常。
- 在默认 NaN 模式下，无论任何 NaN 操作数的分数如何，涉及 NaN 操作数的算术 CDP 指令都返回默认 NaN。算术 CDP 操作中的 SNaN 会设置 IOC 标志 FPSCR[0]。数据传输和非算术 CDP 指令对 NaN 的处理与完全合规模式相同。

表 2-29 总结了 NaN 操作数对指令执行的影响。

表 2-29. QNaN 和 SNaN 处理

指令类型	默认 NaN 模式	使用 QNaN 操作数	使用 SNaN 操作数
算术 CDP	关闭	根据 *ARMv7-M 架构参考手册* 中的规则，返回 QNaN 或其中一个 QNaN 操作数（如果存在多个）。	设置 IOC(1)。SNaN 被静默，结果 NaN 由 *ARMv7-M 架构参考手册* 中的规则决定。
	开启	返回默认 NaN。	设置 IOC(1)。返回默认 NaN。
非算术 CDP	关闭	NaN 传递到目标，符号按需改变。	
	开启		
FCMP(Z)	-	无序比较。	设置 IOC。无序比较。
	-	设置 IOC。无序比较。	设置 IOC。无序比较。
加载/存储	关闭	传输所有 NaN。	
	开启		

(1) IOC 是无效操作异常标志 FPSCR[0]。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.7.2.6.2 对比

比较结果会修改 FPSCR 中的标志。您可以使用 MVRS APSR_nzcv 指令（以前称为 FMSTAT）将当前标志从 FPSCR 传输至 APSR。请参阅 *ARMv7-M 架构参考手册*，了解 IEEE 754-2008 标准谓词到 Arm® 条件的映射。选择使用的标志，以便随后 Arm® 指令的条件执行可以测试 IEEE 标准中定义的谓词。

2.7.2.6.3 下溢

FPU 使用 IEEE 754-2008 标准中描述的舍入前微小值形式和非精确结果精度损失形式来产生下溢异常。

在迫零 (flush-to-zero) 模式下，按照 IEEE 标准所述，舍入前的微小值结果将被直接置为零，并置位 UFC 标志 FPSCR[3]。有关迫零模式的信息，请参阅 *ARMv7-M 架构参考手册*。

当 FPU 未处于迫零模式时，将对非规格化操作数执行运算。如果运算产生的结果不是微小值，则返回计算结果，且不置位 UFC 标志 FPSCR[3]。如果运算非精确，则置位 IXC 标志 FPSCR[4]。如果运算产生微小值结果，则该结果为非规格化值或零值；若该结果同时为非精确结果，则置位 UFC 标志 FPSCR[3]。

2.7.2.7 异常

FPU 根据 FPv4 架构，按照每条指令的要求设置 FPSCR 寄存器中的累积异常状态标志。FPU 不支持异常陷阱。该处理器还具有以下六个输出引脚，每个引脚反映一个累积异常标志的状态：

- FPIXC
- FPUFC
- FPOFC
- FPDZC
- FPIDC
- FPIOC

有关这六个输出的说明，请参阅 *Cortex-M4 集成与实施手册*。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

该处理器可以通过使用惰性压栈来减少异常延迟。这意味着该处理器会在栈上为 FP 状态预留空间，但除非处理器在当前异常处理程序中执行了 FPU 指令，否则不会将该状态信息保存到栈中。

FP 状态的惰性保存可被更高优先级的异常中断。从该异常恢复后，FP 状态保存操作将重新开始。

有关更多信息，请参阅 *ARMv7-M 架构参考手册*。

2.7.3 FPU 编程器模型

节 2.7.3.1 显示了 Arm® Cortex®-M4F 处理器中的 FP 系统寄存器（如果您的实现包含 FPU）。

表 2-30. Arm® Cortex®-M4F 浮点系统寄存器

地址	名称	类型	复位	说明
0xE000 EF34	FPCCR	R/W	0xC000 0000	FP 上下文控制寄存器
0xE000 EF38	FPCAR	R/W	-	FP 上下文地址寄存器
0xE000 EF3C	FPDSCR	R/W	0x0000 0000	FP 默认状态控制寄存器
0xE000 EF40	MVFR0	R/O	0x1011 0021	媒体和 VFP 功能寄存器 0，MVFR0
0xE000 EF44	MVFR1	R/O	0x1100 0011	媒体和 VFP 功能寄存器 1，MVFR1

所有 Arm® Cortex®-M4F FPU 寄存器都在 *ARMv7-M 架构参考手册* 中进行了介绍。

2.7.3.1 使能 FPU

示例 2-1 显示了在特权模式和用户模式下使能 FPU 的示例代码序列。处理器必须处于特权模式才能读写 CPACR。

示例 2-1. 启用 FPU

```
; CPACR is located at address 0xE000ED88
LDR.W R0, =0xE000ED88
; Read CPACR
LDR R1, [R0]
; Set bits 20-23 to enable CP10 and CP11 coprocessors
ORR R1, R1, #(0xF << 20)
; Write back the modified value to the CPACR
STR R1, [R0]
```


English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.8 存储器保护单元 (MPU)

本章介绍了处理器存储器保护单元 (MPU)。其中包含以下小节：

- 关于 MPU (请参阅节 2.8.1)
- MPU 功能说明 (请参阅节 2.8.2)
- MPU 编程器模型 (请参阅节 2.8.3)

2.8.1 关于 MPU

MPU 是用于存储器保护的可选元件。该处理器支持标准的 ARMv7 保护存储器系统架构模型。MPU 完全支持以下功能：

- 保护区域
- 重叠保护区域，具有升序的区域优先级：
  - 7 = 最高优先级
  - 0 = 最低优先级
- 访问权限
- 将存储器属性导出到系统

MPU 不匹配和权限违规会调用可编程优先级的 MemManage 故障处理程序。有关更多信息，请参阅 *ARMv7-M 架构参考手册*。

使用 MPU 可以实现以下目的：

- 强制执行特权规则
- 隔离进程
- 强制执行访问规则

2.8.2 MPU 功能说明

访问权限位，即区域访问控制寄存器的 TEX、C、B、AP 和 XN 控制对相应存储器区域的访问。如果在没有所需权限的情况下访问内存区域，则会引发权限故障。有关更多信息，请参阅 *ARMv7-M 架构参考手册*。

2.8.3 MPU 编程器模型

表 2-31 显示了 MPU 寄存器。这些寄存器在 *ARMv7-M 架构参考手册* 中进行了说明。

表 2-31. MPU 寄存器

地址	名称	类型	复位	说明
0xE000 ED90	MPU_TYPE	R/O	0x0000 0800	MPU 类型寄存器
0xE000 ED94	MPU_CTRL	R/W	0x0000 0000	MPU 控制寄存器
0xE000 ED98	MPU_RNR	R/W	0x0000 0000	MPU 区域编号寄存器
0xE000 ED9C	MPU_RBAR	R/W	0x0000 0000	MPU 区域基址寄存器
0xE000 EDA0	MPU_RASR	R/W	0x0000 0000	MPU 区域属性和大小寄存器
0xE000 EDA4	MPU_RBAR_A1		0x0000 0000	MPU 别名寄存器
0xE000 EDA8	MPU_RASR_A1		0x0000 0000	
0xE000 EDAC	MPU_RBAR_A2		0x0000 0000	
0xE000 EDB0	MPU_RASR_A2		0x0000 0000	
0xE000 EDB4	MPU_RBAR_A3		0x0000 0000	
0xE000 EDB8	MPU_RASR_A3		0x0000 0000	

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9 Arm® Cortex®-M4F 处理器寄存器

表 2-32 列出了 Arm® Cortex®-M4F 处理器中的寄存器。

表 2-32. Arm® Cortex®-M4F 处理器寄存器

模块	名称	部分
CPU_DWT	核心数据观察点和跟踪	节 2.9.1
CPU_FPB	内核闪存补丁及断点	节 2.9.2
CPU_ITM	内核仪器跟踪宏单元	节 2.9.3
CPU_SCS	内核系统控制空间	节 2.9.4
CPU_TPIU	内核跟踪端口接口单元	节 2.9.5

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.1 CPU_DWT 寄存器

表 2-33 列出了 CPU_DWT 寄存器的存储器映射寄存器。表 2-33 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

表 2-33. CPU_DWT 寄存器

偏移	首字母缩写词	寄存器名称	部分
0h	CTRL	控制	CTRL 寄存器（偏移 = 0h）[复位 = 40000000h]
4h	CYCCNT	当前 PC 采样器周期计数	CYCCNT 寄存器（偏移 = 4h）[复位 = 00000000h]
8h	CPICNT	CPI 数目	CPICNT 寄存器（偏移 = 8h）[复位 = 00000000h]
Ch	EXCCNT	异常开销计数	EXCCNT 寄存器（偏移 = Ch）[复位 = 00000000h]
10h	SLEEPCNT	睡眠计数	SLEEPCNT 寄存器（偏移 =10h）[复位 = 00000000h]
14h	LSUCNT	LSU 数目	LSUCNT 寄存器（偏移 = 14h）[复位 = 00000000h]
18h	FOLDCNT	折叠计数	FOLDCNT 寄存器（偏移 =18h）[复位 = 00000000h]
1Ch	PCSR	程序计数器样本	PCSR 寄存器（偏移 = 1Ch）[复位 = 00000000h]
20h	COMP0	比较器 0	COMP0 寄存器（偏移 = 20h）[复位 = 00000000h]
24h	MASK0	屏蔽 0	MASK0 寄存器（偏移 = 24h）[复位 = 00000000h]
28h	FUNCTION0	功能 0	FUNCTION0 寄存器（偏移 = 28h）[复位 = 00000000h]
30h	COMP1	比较器 1	COMP1 寄存器（偏移 = 30h）[复位 = 00000000h]
34h	MASK1	屏蔽 1	MASK1 寄存器（偏移 = 34h）[复位 = 00000000h]
38h	FUNCTION1	功能 1	FUNCTION1 寄存器（偏移 = 38h）[复位 = 00000200h]
40h	COMP2	比较器 2	COMP2 寄存器（偏移 = 40h）[复位 = 00000000h]
44h	MASK2	屏蔽 2	MASK2 寄存器（偏移 = 44h）[复位 = 00000000h]
48h	FUNCTION2	功能 2	FUNCTION2 寄存器（偏移 = 48h）[复位 = 00000000h]
50h	COMP3	比较器 3	COMP3 寄存器（偏移 = 50h）[复位 = 00000000h]
54h	MASK3	屏蔽 3	MASK3 寄存器（偏移 = 54h）[复位 = 00000000h]
58h	FUNCTION3	功能 3	FUNCTION3 寄存器（偏移 = 58h）[复位 = 00000000h]

复杂的位访问类型经过编码可适应小型表单元。表 2-34 展示了适用于此部分中访问类型的代码。

表 2-34. CPU_DWT 访问类型代码

访问类型	代码	说明
读取类型		
R	R	读取

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 2-34. CPU_DWT 访问类型代码（续）

访问类型	代码	说明
写入类型		
W	W	写入
复位或默认值		
-n		复位后的值或默认值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.1.1 CTRL 寄存器（偏移 = 0h）[复位 = 40000000h]

图 2-5 展示了 CTRL，表 2-35 中对此进行了介绍。

返回到汇总表。

控制
使用 DWT 控制寄存器启用 DWT 单元。

图 2-5. CTRL 寄存器

31	30	29	28	27	26	25	24
RESERVED						NOCYCCNT	NOPRFCNT
R/W-10h						R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	CYCEVTENA	FOLDEVTENA	LSUEVTENA	SLEEPEVTENA	EXCEVTENA	CPIEVTENA	EXCTRCENA
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			PCSAMPLEENA	SYNCTAP		CYCTAP	POSTCNT
R/W-0h			R/W-0h	R/W-0h		R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
POSTCNT			POSTPRESET				CYCCNTENA
R/W-0h			R/W-0h				R/W-0h

表 2-35. CTRL 寄存器字段说明

位	字段	类型	复位	说明
31-26	RESERVED	R/W	10h	软件不应依赖保留位的值。写入除复位值外的任何其他值都可能导致未定义的行为。
25	NOCYCCNT	R/W	0h	置位后，不支持 CYCCNT。
24	NOPRFCNT	R/W	0h	置位后，不支持 FOLDCNT、LSUCNT、SLEEPCNT、EXCCNT 和 CPICNT。
23	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值外的任何其他值都可能导致未定义的行为。
22	CYCEVTENA	R/W	0h	启用周期计数事件。在 POSTCNT 计数器触发时发出事件。有关详细信息，请参阅 CYCTAP 和 POSTPRESET。仅当 PCSAMPLEENA 被禁用时才会发出此事件。PCSAMPLEENA 会覆盖此位的设置。 0：禁用周期计数事件 1：启用周期计数事件
21	FOLDEVTENA	R/W	0h	启用折叠指令计数事件。当 FOLDCNT 溢出时（每 256 个折叠指令周期）发出事件。折叠指令是指执行时甚至用不到一个周期的指令。例如，IT 指令被折叠，因此不会用完一个周期。 0：禁用折叠指令计数事件。 1：启用折叠指令计数事件。
20	LSUEVTENA	R/W	0h	启用 LSU 计数事件。当 LSUCNT 溢出时（每 256 个 LSU 操作周期）发出事件。LSU 计数包括指令初始周期后的所有 LSU 成本。 0：禁用 LSU 计数事件。 1：启用 LSU 计数事件。
19	SLEEPEVTENA	R/W	0h	启用睡眠计数事件。当 SLEEPCNT 溢出时（处理器处于睡眠状态每 256 个周期）发出事件。 0：禁用睡眠计数事件。 1：启用睡眠计数事件。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 2-35. CTRL 寄存器字段说明（续）

位	字段	类型	复位	说明
18	EXCEVTENA	R/W	0h	启用中断开销事件。在 EXCCNT 溢出时（每 256 个中断开销周期）发出事件。 0x0：禁用中断开销事件。 0x1：启用中断开销事件。
17	CPIEVTENA	R/W	0h	启用 CPI 计数事件。当 CPICNT 溢出时（每 256 个多周期指令周期）发出事件。 0：禁用 CPI 计数器事件。 1：启用 CPI 计数器事件。
16	EXCTRCENA	R/W	0h	启用中断事件跟踪。 0：禁用中断事件跟踪。 1：启用中断事件跟踪。
15-13	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值外的任何其他值都可能导致未定义的行为。
12	PCSAMPLEENA	R/W	0h	启用 PC 采样事件。当 POSTCNT 计数器触发时发出 PC 采样事件。有关详细信息，请参阅 CYCTAP 和 POSTPRESET。启用该位会覆盖 CYCEVTENA。 0：禁用 PC 采样事件。 1：启用采样事件。
11-10	SYNCTAP	R/W	0h	选择同步数据包速率。还必须为此特性启用 CYCCNTENA 和 CPU_ITM:TCR.SYNCENA。 在抽头切换（0 到 1 或 1 到 0）时生成同步数据包（如果已启用）。 0h = 禁用。无同步数据包 1h = 在 CYCCNT 的 位 24 进行抽头 2h = 在 CYCCNT 的 位 26 进行抽头 3h = 在 CYCCNT 的 位 28 进行抽头
9	CYCTAP	R/W	0h	选择 CYCCNT 上的抽头位。这些位置分别位于位 [6] 和 [10]。当 CYCCNT 中选择的位从 0 变为 1 或从 1 变为 0 时，会向 POSTCNT 后分频计数器发出信号。然后该计数器递减计数。当后分频计数器为 0 时，如果位发生变化，则会触发 PC 采样或周期计数事件（请参阅 CYCEVTENA 中的详细信息）。 0h = 选择要进行抽头的位 [6] 1h = 选择要进行抽头的位 [10]
8-5	POSTCNT	R/W	0h	用于 CYCTAP 的后分频计数器。当选定的抽头位从 0 变为 1 或从 1 变为 0 时，如果后分频计数器非零，则进行递减计数。如果为 0，则触发一个事件供 PCSAMPLEENA 或 CYCEVTENA 使用。同时，它会从 POSTPRESET 重新加载该值。
4-1	POSTPRESET	R/W	0h	后分频计数器 POSTCNT 的重新加载值。为 0 时，每次抽头变化都会触发事件（相当于 2 的幂次分频）。如果该字段具有非 0 值，则形成一个递减计数值，每次计数到 0 时就会将该值重新加载到 POSTCNT。例如，该寄存器中的值 1 表示每隔一次抽头变化形成一个事件。
0	CYCCNTENA	R/W	0h	启用 CYCCNT，允许其递增并生成同步和计数事件。如果 NOCYCCNT = 1，则该位读为零，并忽略写入。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.1.2 CYCCNT 寄存器（偏移 = 4h）[复位 = 00000000h]

图 2-6 展示了 CYCCNT，表 2-36 中对此进行了介绍。

返回到汇总表。

当前 PC 采样器周期计数
该寄存器用于计算内核周期数。该计数器可以测量经过的执行时间。这是一个自由运行的计数器（在 CPU 的自由运行时钟停止的电源模式下，该计数器不会前进）。该计数器有三个功能：
1：当 CTRL.PCSAMPLEENA = 1 时，在选定的抽头位改变值（0 到 1 或 1 到 0）并且任意后分频值计数到 0 时，对 PC 进行采样并发出。
2：当 CTRL.CYCEVTENA = 1（且 CTRL.PCSAMPLEENA = 0）时，在选定的抽头位改变值（0 到 1 或 1 到 0）并且任意后分频值计数到 0 时，发出事件。
3：应用程序和调试器可以使用该计数器测量经过的执行时间。通过减去开始时间和结束时间，应用程序可以测量内核时钟的间隔时间（在调试期间中止时除外）。这在最多 2^{32} 个内核时钟周期内有效（例如，在 48MHz 下约为 89.5 秒）。

图 2-6. CYCCNT 寄存器

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

表 2-36. CYCCNT 寄存器字段说明

位	字段	类型	复位	说明
31-0	CYCCNT	R/W	0h	当前 PC 采样器周期计数器计数值。启用后，该计数器对内核周期数进行计数，但内核中止时除外。周期计数器是一个向上计数的自由运行计数器（在 CPU 的自由运行时钟停止的电源模式下，该计数器不会前进）。它在溢出时绕回到 0。首次启用时，调试器必须将其初始化为 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.1.3 CPICNT 寄存器（偏移 = 8h）[复位 = 00000000h]

图 2-7 展示了 CPICNT，表 2-37 中对此进行了介绍。

返回到汇总表。

CPI 计数
该寄存器用于计算第一个周期之后的指令周期总数。

图 2-7. CPICNT 寄存器

31	30	29	28	27	26	25	24	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																								CPICNT							
R/W-0h																								R/W-X							

表 2-37. CPICNT 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值外的任何其他值都可能导致未定义的行为。
7-0	CPICNT	R/W	X	当前 CPI 计数器值。在执行除 LSUCNT 记录的指令之外的所有指令所需的额外周期（第一个周期不计入）上递增。该计数器还会在所有指令提取停滞时递增。如果 CTRL.CPIEVTENA 置位，则在计数器溢出时发出事件。该计数器在通过 CTRL.CPIEVTENA 启用时初始化为 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.1.4 EXCCNT 寄存器（偏移 = Ch）[复位 = 00000000h]

图 2-8 展示了 EXCCNT，表 2-38 中对此进行了介绍。

返回到汇总表。

异常开销计数
该寄存器用于计算中断处理中花费的总周期数。

图 2-8. EXCCNT 寄存器

31	30	29	28	27	26	25	24	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																								EXCCNT							
R/W-0h																								R/W-X							

表 2-38. EXCCNT 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值外的任何其他值都可能导致未定义的行为。
7-0	EXCCNT	R/W	X	当前中断开销计数器值。计算中断处理中花费的总周期数（例如进入时的压栈、返回时的出栈、抢占等）。在计数器溢出（每 256 个周期）时发出事件。该计数器在通过 CTRL.EXCEVTENA 启用时初始化为 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.1.5 SLEEPCNT 寄存器（偏移 =10h）[复位 = 00000000h]

图 2-9 展示了 SLEEPCNT，表 2-39 中对此进行了介绍。

返回到汇总表。

睡眠计数
该寄存器用于计算处理器处于睡眠状态的总周期数。

图 2-9. SLEEPCNT 寄存器

31	30	29	28	27	26	25	24	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																								SLEEPCNT							
R/W-0h																								R/W-X							

表 2-39. SLEEPCNT 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值外的任何其他值都可能导致未定义的行为。
7-0	SLEEPCNT	R/W	X	睡眠计数器。计算处理器处于睡眠状态的周期数。在计数器溢出（每 256 个周期）时发出事件。该计数器在通过 CTRL.SLEEPEVTENA 启用时初始化为 0。请注意，睡眠计数器使用 CPU 的自由运行时钟计时。在某些电源模式下，CPU 的自由运行时钟被门控以更大限度降低功耗。这意味着在这些电源模式下，睡眠计数器将无效。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.1.6 LSUCNT 寄存器（偏移 = 14h）[复位 = 00000000h]

图 2-10 展示了 LSUCNT，表 2-40 中对此进行了介绍。

返回到汇总表。

LSU 计数
该寄存器用于计算处理器在第一个周期之后处理 LSU 操作的总周期数。

图 2-10. LSUCNT 寄存器

31	30	29	28	27	26	25	24	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																								LSUCNT							
R/W-0h																								R/W-X							

表 2-40. LSUCNT 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值外的任何其他值都可能导致未定义的行为。
7-0	LSUCNT	R/W	X	LSU 计数器。计算处理器处理 LSU 操作的总周期数。指令的初始执行成本不计入。例如，需要两个周期才能完成的 LDR 会将此计数器递增一个周期。同样，停顿了两个周期的 LDR（即需要四个周期来执行）会将此计数器递增三次。在计数器溢出（每 256 个周期）时发出事件。该计数器在通过 CTRL.LSUEVTENA 启用时初始化为 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.1.7 FOLDCNT 寄存器（偏移 =18h）[复位 = 00000000h]

图 2-11 展示了 FOLDCNT，表 2-41 中对此进行了介绍。

返回到汇总表。

折叠计数
该寄存器用于计算折叠指令的总数。该计数器在每个占用 0 个周期的指令上递增。

图 2-11. FOLDCNT 寄存器

31	30	29	28	27	26	25	24	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																								FOLDCNT							
R/W-0h																								R/W-X							

表 2-41. FOLDCNT 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值外的任何其他值都可能导致未定义的行为。
7-0	FOLDCNT	R/W	X	计算折叠指令的总数。该计数器在通过 CTRL.FOLDEVTENA 启用时初始化为 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.1.8 PCSR 寄存器（偏移 = 1Ch）[复位 = 00000000h]

图 2-12 展示了 PCSR，表 2-42 中对此进行了介绍。

返回到汇总表。

程序计数器采样
该寄存器用于通过调试代理启用粗粒度软件分析，而无需更改当前执行的代码。如果内核未处于调试状态，返回的值为最近执行指令的指令地址。如果内核处于调试状态，返回的值为 0xFFFFFFFF。

图 2-12. PCSR 寄存器

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

表 2-42. PCSR 寄存器字段说明

位	字段	类型	复位	说明
31-0	EIASAMPLE	R	X	执行指令地址采样，如果内核中止，则为 0xFFFFFFFF。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.1.9 COMP0 寄存器（偏移 = 20h）[复位 = 00000000h]

图 2-13 展示了 COMP0，表 2-43 对其进行了介绍。

返回到汇总表。

比较器 0
该寄存器用于写入比较器 0 的基准值。

图 2-13. COMP0 寄存器

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

表 2-43. COMP0 寄存器字段说明

位	字段	类型	复位	说明
31-0	COMP	R/W	X	用于与 PC 或 FUNCTION0 给出的数据地址进行比较的基准值。比较器 0 也可以与 PC 采样器计数器 (CYCCNT) 的值进行比较。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.1.10 MASK0 寄存器（偏移 = 24h）[复位 = 00000000h]

图 2-14 展示了 MASK0，表 2-44 对其进行了介绍。

返回到汇总表。

掩码 0
使用 DWT 掩码寄存器 0 在与 COMP0 匹配时对数据地址应用掩码。

图 2-14. MASK0 寄存器

31	30	29	28	27	26	25	24	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/W-0h																												R/W-X			

表 2-44. MASK0 寄存器字段说明

位	字段	类型	复位	说明
31-4	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值外的任何其他值都可能导致未定义的行为。
3-0	MASK	R/W	X	与 COMP0 匹配时应用于数据地址的掩码。这是忽略掩码的大小。也就是说，DWT 匹配的执行方式如下：（ADDR 和（0xFFFF 左移 MASK 位）的逻辑“与”结果）== COMP0。然而，实际比较稍微复杂一些，以便能匹配在总线上任何位置出现的地址。因此，如果 COMP0 为 3，将与字访问 0 匹配，因为 3 包含在该字中。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.1.11 FUNCTION0 寄存器（偏移 = 28h）[复位 = 00000000h]

图 2-15 展示了 FUNCTION0，表 2-45 对其进行了介绍。

返回到汇总表。

功能 0
使用 DWT 功能寄存器 0 控制比较器 0 的操作。该比较器可以：
1.与 PC 或数据地址匹配。这由 CYCMATCH 控制。该功能仅适用于比较器 0 (COMP0)。
2.根据 FUNCTION 定义的操作，发出数据或 PC 对，触发 ETM，或生成观察点。

图 2-15. FUNCTION0 寄存器

31	30	29	28	27	26	25	24
RESERVED							MATCHED
R-0h							R/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
CYCMATCH	RESERVED	EMITRANGE	RESERVED	功能			
R/W-0h	R-0h	R/W-0h	R-0h	R/W-0h			

表 2-45. FUNCTION0 寄存器字段说明

位	字段	类型	复位	说明
31-25	RESERVED	R	0h	软件不应依赖保留位的值。写入除复位值外的任何其他值都可能导致未定义的行为。
24	MATCHED	R/W	0h	当比较器匹配时，此位置位，并指示自上次读取此位以来已发生 FUNCTION 所定义的操作。该位在读取后清零除。
23-8	RESERVED	R	0h	软件不应依赖保留位的值。写入除复位值外的任何其他值都可能导致未定义的行为。
7	CYCMATCH	R/W	0h	该位仅在比较器 0 中可用。置位时，COMP0 将与周期计数器 (CYCCNT) 进行比较。
6	RESERVED	R	0h	软件不应依赖保留位的值。写入除复位值外的任何其他值都可能导致未定义的行为。
5	EMITRANGE	R/W	0h	发出范围字段。该位允许在范围匹配发生时发出偏移量。启用发出范围时，不支持 PC 采样。 该字段仅适用于：FUNCTION = 1、2、3、12、13、14 和 15。
4	RESERVED	R	0h	软件不应依赖保留位的值。写入除复位值外的任何其他值都可能导致未定义的行为。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 2-45. FUNCTION0 寄存器字段说明（续）

位	字段	类型	复位	说明
3-0	功能	R/W	0h	功能设置。 0x0: 禁用 0x1：EMITRANGE = 0，通过 ITM 对 PC 进行采样并发出。EMITRANGE = 1，通过 ITM 发出地址偏移 0x2：EMITRANGE = 0，在读和写时通过 ITM 发出数据。EMITRANGE = 1，在读或写时通过 ITM 发出数据和地址偏移。 0x3：EMITRANGE = 0，在读或写时通过 ITM 对 PC 和数据值进行采样。EMITRANGE = 1，在读或写时通过 ITM 发出地址偏移和数据值。 0x4：PC 匹配时的观察点。 0x5：读取时的观察点。 0x6：写入时的观察点。 0x7：读取或写入时的观察点。 0x8：PC 匹配时的 ETM 触发条件 0x9：读取时的 ETM 触发条件 0xA：写入时的 ETM 触发条件 0xB：读取或写入时的 ETM 触发条件 0xC：EMITRANGE = 0，读传输时采样数据。EMITRANGE = 1，读传输时采样 Daddr（低 16 位） 0xD：EMITRANGE = 0，写传输时采样数据。EMITRANGE = 1，写传输时采样 Daddr（低 16 位） 0xE：EMITRANGE = 0，读传输时采样 PC + 数据。EMITRANGE = 1，读传输时采样 Daddr（低 16 位）+ 数据 0xF：EMITRANGE = 0，写传输时采样 PC + 数据。EMITRANGE = 1，写传输时采样 Daddr（低 16 位）+ 数据 注释 1：如果未配备 ETM，则无法实现 ETM 触发。 注释 2：数据值仅针对不会引起故障（MPU 或总线故障）的访问进行采样。无论是否存在故障，PC 都会进行采样。PC 仅针对突发访问的第一个地址进行采样。 注释 3：不建议将 PC 匹配用于观察点，因为它会在指令之后停止。它主要用于保护和触发 ETM。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.1.12 COMP1 寄存器（偏移 = 30h）[复位 = 00000000h]

图 2-16 展示了 COMP1，表 2-46 对其进行了介绍。

返回到汇总表。

比较器 1
该寄存器用于写入比较器 1 的基准值。

图 2-16. COMP1 寄存器

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

表 2-46. COMP1 寄存器字段说明

位	字段	类型	复位	说明
31-0	COMP	R/W	X	用于与 PC 或 FUNCTION1 给出的数据地址进行比较的基准值。 比较器 1 还可以比较数据值。因此该寄存器可以包含用于数据匹配的基准值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.1.13 MASK1 寄存器（偏移 = 34h）[复位 = 00000000h]

图 2-17 展示了 MASK1，表 2-47 对其进行了介绍。

返回到汇总表。

掩码 1
使用 DWT 掩码寄存器 1 在与 COMP1 匹配时对数据地址应用掩码。

图 2-17. MASK1 寄存器

31	30	29	28	27	26	25	24	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/W-0h																												R/W-X			

表 2-47. MASK1 寄存器字段说明

位	字段	类型	复位	说明
31-4	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值外的任何其他值都可能导致未定义的行为。
3-0	MASK	R/W	X	与 COMP1 匹配时应用于数据地址的掩码。这是忽略掩码的大小。也就是说，DWT 匹配的执行方式如下：（ADDR 和（0xFFFF 左移 MASK 位）的逻辑“与”结果）== COMP1。然而，实际比较稍微复杂一些，以便能匹配在总线上任何位置出现的地址。因此，如果 COMP1 为 3，将与字访问 0 匹配，因为 3 包含在该字中。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.1.14 FUNCTION1 寄存器（偏移 = 38h）[复位 = 00000200h]

图 2-18 展示了 FUNCTION1，表 2-48 对其进行了介绍。

返回到汇总表。

功能 1
使用 DWT 功能寄存器 1 控制比较器 1 的操作。该比较器可以：
1.如果关联的地址比较器已执行地址匹配，则执行数据值比较。该功能仅适用于比较器 1 (COMP1)。
2.根据 FUNCTION 定义的操作，发出数据或 PC 对，触发 ETM，或生成观察点。

图 2-18. FUNCTION1 寄存器

31	30	29	28	27	26	25	24
RESERVED							MATCHED
R-0h							R/W-0h
23	22	21	20	19	18	17	16
RESERVED				DATAVADDR1			
R-0h				R/W-0h			
15	14	13	12	11	10	9	8
DATAVADDR0				DATAVSIZE		LNK1ENA	DATAVMATCH
R/W-0h				R/W-0h		R-1h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED		EMITRANGE	RESERVED	功能			
R-0h		R/W-0h	R-0h	R/W-0h			

表 2-48. FUNCTION1 寄存器字段说明

位	字段	类型	复位	说明
31-25	RESERVED	R	0h	软件不应依赖保留位的值。写入除复位值外的任何其他值都可能导致未定义的行为。
24	MATCHED	R/W	0h	当比较器匹配时，此位置位，并指示自上次读取此位以来已发生 FUNCTION 所定义的操作。该位在读取后清零除。
23-20	RESERVED	R	0h	软件不应依赖保留位的值。写入除复位值外的任何其他值都可能导致未定义的行为。
19-16	DATAVADDR1	R/W	0h	当 DATAVMATCH =1 且 LNK1ENA =1 时，用于数据值匹配的第二个链接地址比较器的标识。
15-12	DATAVADDR0	R/W	0h	DATAVMATCH =1 时用于数据值匹配的链接地址比较器的标识。
11-10	DATAVSIZE	R/W	0h	定义 COMP1 寄存器中要匹配的数据大小： 0x0：字节 0x1：半字 0x2：字 0x3：不可预测。
9	LNK1ENA	R	1h	仅在比较器 1 中支持只读位字段。 0：DATAVADDR1 不支持 1：DATAVADDR1 支持（已启用）
8	DATAVMATCH	R/W	0h	数据匹配特性： 0：执行地址比较 1：执行数据值比较。DATAVADDR0 和 DATAVADDR1 给出的比较器提供了用于数据比较的地址。DATAVADDR0 和 DATAVADDR1 给出的比较器的 FUNCTION 设置会被覆盖，这些比较器仅为数据比较提供地址匹配。 该位仅在比较器 1 中可用。
7-6	RESERVED	R	0h	软件不应依赖保留位的值。写入除复位值外的任何其他值都可能导致未定义的行为。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 2-48. FUNCTION1 寄存器字段说明（续）

位	字段	类型	复位	说明
5	EMITRANGE	R/W	0h	发出范围字段。该位允许在范围匹配发生时发出偏移量。启用发出范围时，不支持 PC 采样。 该字段仅适用于：FUNCTION = 1、2、3、12、13、14 和 15。
4	RESERVED	R	0h	软件不应依赖保留位的值。写入除复位值外的任何其他值都可能导致未定义的行为。
3-0	功能	R/W	0h	功能设置： 0x0：禁用 0x1：EMITRANGE = 0，通过 ITM 对 PC 进行采样并发出。EMITRANGE = 1，通过 ITM 发出地址偏移 0x2：EMITRANGE = 0，在读和写时通过 ITM 发出数据。EMITRANGE = 1，在读或写时通过 ITM 发出数据和地址偏移。 0x3：EMITRANGE = 0，在读或写时通过 ITM 对 PC 和数据值进行采样。EMITRANGE = 1，在读或写时通过 ITM 发出地址偏移和数据值。 0x4：PC 匹配时的观察点。 0x5：读取时的观察点。 0x6：写入时的观察点。 0x7：读取或写入时的观察点。 0x8：PC 匹配时的 ETM 触发条件 0x9：读取时的 ETM 触发条件 0xA：写入时的 ETM 触发条件 0xB：读取或写入时的 ETM 触发条件 0xC：EMITRANGE = 0，读传输时采样数据。EMITRANGE = 1，读传输时采样 Daddr（低 16 位） 0xD：EMITRANGE = 0，写传输时采样数据。EMITRANGE = 1，写传输时采样 Daddr（低 16 位） 0xE：EMITRANGE = 0，读传输时采样 PC + 数据。EMITRANGE = 1，读传输时采样 Daddr（低 16 位）+ 数据 0xF：EMITRANGE = 0，写传输时采样 PC + 数据。EMITRANGE = 1，写传输时采样 Daddr（低 16 位）+ 数据 注释 1：如果未配备 ETM，则无法实现 ETM 触发。 注释 2：数据值仅针对不会引起故障（MPU 或总线故障）的访问进行采样。无论是否存在故障，PC 都会进行采样。PC 仅针对突发访问的第一个地址进行采样。 注释 3：如果同时置位 DATAVMATCH，则会覆盖 DATAVADDR0 和 DATAVADDR1 所给出比较器的 FUNCTION。然后，DATAVADDR0 和 DATAVADDR1 给出的比较器只能对比较器 1 数据匹配执行地址比较器匹配。 注释 4：如果实现过程中未包含数据匹配功能，则无法置位 DATAVADDR0、DATAVADDR1 或 DATAVMATCH。这意味着数据匹配功能在实现方案中不可用。可通过写入和读取 DATAVMATCH 来测试数据匹配的可用性。如果无法置位，则数据匹配不可用。 注释 5：不建议将 PC 匹配用于观察点，因为它会在指令之后停止。它主要用于保护和触发 ETM。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.1.15 COMP2 寄存器（偏移 = 40h）[复位 = 00000000h]

图 2-19 展示了 COMP2，表 2-49 对其进行了介绍。

返回到汇总表。

比较器 2
该寄存器用于写入比较器 2 的基准值。

图 2-19. COMP2 寄存器

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

表 2-49. COMP2 寄存器字段说明

位	字段	类型	复位	说明
31-0	COMP	R/W	X	用于与 PC 或 FUNCTION2 给出的数据地址进行比较的基准值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.1.16 MASK2 寄存器（偏移 = 44h）[复位 = 00000000h]

图 2-20 展示了 MASK2，表 2-50 对其进行了介绍。

返回到汇总表。

掩码 2
使用 DWT 掩码寄存器 2 在与 COMP2 匹配时对数据地址应用掩码。

图 2-20. MASK2 寄存器

31	30	29	28	27	26	25	24	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/W-0h																												R/W-X			

表 2-50. MASK2 寄存器字段说明

位	字段	类型	复位	说明
31-4	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值外的任何其他值都可能导致未定义的行为。
3-0	MASK	R/W	X	与 COMP2 匹配时应用于数据地址的掩码。这是忽略掩码的大小。也就是说，DWT 匹配的执行方式如下：（ADDR 和（0xFFFF 左移 MASK 位）的逻辑“与”结果）== COMP2。然而，实际比较稍微复杂一些，以便能匹配在总线上任何位置出现的地址。因此，如果 COMP2 为 3，将与字访问 0 匹配，因为 3 包含在该字中。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.1.17 FUNCTION2 寄存器（偏移 = 48h）[复位 = 00000000h]

图 2-21 展示了 FUNCTION2，表 2-51 对其进行了介绍。

返回到汇总表。

功能 2
使用 DWT 功能寄存器 2 控制比较器 2 的操作。根据 FUNCTION 定义的操作，该比较器可以发出数据或 PC 对，触发 ETM，或生成观察点。

图 2-21. FUNCTION2 寄存器

31	30	29	28	27	26	25	24
RESERVED							MATCHED
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
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED		EMITRANGE	RESERVED	功能			
R-0h		R/W-0h	R-0h	R/W-0h			

表 2-51. FUNCTION2 寄存器字段说明

位	字段	类型	复位	说明
31-25	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值外的任何其他值都可能导致未定义的行为。
24	MATCHED	R/W	0h	当比较器匹配时，此位置位，并指示自上次读取此位以来已发生 FUNCTION 所定义的操作。该位在读取后清零除。
23-6	RESERVED	R	0h	软件不应依赖保留位的值。写入除复位值外的任何其他值都可能导致未定义的行为。
5	EMITRANGE	R/W	0h	发出范围字段。该位允许在范围匹配发生时发出偏移量。启用发出范围时，不支持 PC 采样。 该字段仅适用于：FUNCTION = 1、2、3、12、13、14 和 15。
4	RESERVED	R	0h	软件不应依赖保留位的值。写入除复位值外的任何其他值都可能导致未定义的行为。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 2-51. FUNCTION2 寄存器字段说明（续）

位	字段	类型	复位	说明
3-0	功能	R/W	0h	功能设置。 0x0: 禁用 0x1：EMITRANGE = 0，通过 ITM 对 PC 进行采样并发出。EMITRANGE = 1，通过 ITM 发出地址偏移 0x2：EMITRANGE = 0，在读和写时通过 ITM 发出数据。EMITRANGE = 1，在读或写时通过 ITM 发出数据和地址偏移。 0x3：EMITRANGE = 0，在读或写时通过 ITM 对 PC 和数据值进行采样。EMITRANGE = 1，在读或写时通过 ITM 发出地址偏移和数据值。 0x4：PC 匹配时的观察点。 0x5：读取时的观察点。 0x6：写入时的观察点。 0x7：读取或写入时的观察点。 0x8：PC 匹配时的 ETM 触发条件 0x9：读取时的 ETM 触发条件 0xA：写入时的 ETM 触发条件 0xB：读取或写入时的 ETM 触发条件 0xC：EMITRANGE = 0，读传输时采样数据。EMITRANGE = 1，读传输时采样 Daddr（低 16 位） 0xD：EMITRANGE = 0，写传输时采样数据。EMITRANGE = 1，写传输时采样 Daddr（低 16 位） 0xE：EMITRANGE = 0，读传输时采样 PC + 数据。EMITRANGE = 1，读传输时采样 Daddr（低 16 位）+ 数据 0xF：EMITRANGE = 0，写传输时采样 PC + 数据。EMITRANGE = 1，写传输时采样 Daddr（低 16 位）+ 数据 注释 1：如果未配备 ETM，则无法实现 ETM 触发。 注释 2：数据值仅针对不会引起故障（MPU 或总线故障）的访问进行采样。无论是否存在故障，PC 都会进行采样。PC 仅针对突发访问的第一个地址进行采样。 注释 3：不建议将 PC 匹配用于观察点，因为它会在指令之后停止。它主要用于保护和触发 ETM。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.1.18 COMP3 寄存器（偏移 = 50h）[复位 = 00000000h]

图 2-22 展示了 COMP3，表 2-52 对其进行了介绍。

返回到汇总表。

比较器 3
该寄存器用于写入比较器 3 的基准值。

图 2-22. COMP3 寄存器

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

表 2-52. COMP3 寄存器字段说明

位	字段	类型	复位	说明
31-0	COMP	R/W	X	用于与 PC 或 FUNCTION3 给出的数据地址进行比较的基准值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.1.19 MASK3 寄存器（偏移 = 54h）[复位 = 00000000h]

图 2-23 展示了 MASK3，表 2-53 对其进行了介绍。

返回到汇总表。

掩码 3
使用 DWT 掩码寄存器 3 在与 COMP3 匹配时对数据地址应用掩码。

图 2-23. MASK3 寄存器

31	30	29	28	27	26	25	24	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/W-0h																												R/W-X			

表 2-53. MASK3 寄存器字段说明

位	字段	类型	复位	说明
31-4	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值外的任何其他值都可能导致未定义的行为。
3-0	MASK	R/W	X	与 COMP3 匹配时应用于数据地址的掩码。这是忽略掩码的大小。也就是说，DWT 匹配的执行方式如下：（ADDR 和（0xFFFF 左移 MASK 位）的逻辑“与”结果）== COMP3。然而，实际比较稍微复杂一些，以便能匹配在总线上任何位置出现的地址。因此，如果 COMP3 为 3，将与字访问 0 匹配，因为 3 包含在该字中。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.1.20 FUNCTION3 寄存器（偏移 = 58h）[复位 = 00000000h]

图 2-24 展示了 FUNCTION3，表 2-54 对其进行了介绍。

返回到汇总表。

功能 3
使用 DWT 功能寄存器 3 控制比较器 3 的操作。根据 FUNCTION 定义的操作，该比较器可以发出数据或 PC 对，触发 ETM，或生成观察点。

图 2-24. FUNCTION3 寄存器

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

表 2-54. FUNCTION3 寄存器字段说明

位	字段	类型	复位	说明
31-25	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值外的任何其他值都可能导致未定义的行为。
24	MATCHED	R/W	0h	当比较器匹配时，此位置位，并指示自上次读取此位以来已发生 FUNCTION 所定义的操作。该位在读取后清零除。
23-6	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值外的任何其他值都可能导致未定义的行为。
5	EMITRANGE	R/W	0h	发出范围字段。该位允许在范围匹配发生时发出偏移量。启用发出范围时，不支持 PC 采样。 该字段仅适用于：FUNCTION = 1、2、3、12、13、14 和 15。
4	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值外的任何其他值都可能导致未定义的行为。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 2-54. FUNCTION3 寄存器字段说明（续）

位	字段	类型	复位	说明
3-0	功能	R/W	0h	功能设置。 0x0: 禁用 0x1：EMITRANGE = 0，通过 ITM 对 PC 进行采样并发出。EMITRANGE = 1，通过 ITM 发出地址偏移 0x2：EMITRANGE = 0，在读和写时通过 ITM 发出数据。EMITRANGE = 1，在读或写时通过 ITM 发出数据和地址偏移。 0x3：EMITRANGE = 0，在读或写时通过 ITM 对 PC 和数据值进行采样。EMITRANGE = 1，在读或写时通过 ITM 发出地址偏移和数据值。 0x4：PC 匹配时的观察点。 0x5：读取时的观察点。 0x6：写入时的观察点。 0x7：读取或写入时的观察点。 0x8：PC 匹配时的 ETM 触发条件 0x9：读取时的 ETM 触发条件 0xA：写入时的 ETM 触发条件 0xB：读取或写入时的 ETM 触发条件 0xC：EMITRANGE = 0，读传输时采样数据。EMITRANGE = 1，读传输时采样 Daddr（低 16 位） 0xD：EMITRANGE = 0，写传输时采样数据。EMITRANGE = 1，写传输时采样 Daddr（低 16 位） 0xE：EMITRANGE = 0，读传输时采样 PC + 数据。EMITRANGE = 1，读传输时采样 Daddr（低 16 位）+ 数据 0xF：EMITRANGE = 0，写传输时采样 PC + 数据。EMITRANGE = 1，写传输时采样 Daddr（低 16 位）+ 数据 注释 1：如果未配备 ETM，则无法实现 ETM 触发。 注释 2：数据值仅针对不会引起故障（MPU 或总线故障）的访问进行采样。无论是否存在故障，PC 都会进行采样。PC 仅针对突发访问的第一个地址进行采样。 注释 3：不建议将 PC 匹配用于观察点，因为它会在指令之后停止。它主要用于保护和触发 ETM。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.2 CPU_FPB 寄存器

表 2-55 列出了 CPU_FPB 寄存器的存储器映射寄存器。表 2-55 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

表 2-55. CPU_FPB 寄存器

偏移	首字母缩写词	寄存器名称	部分
0h	CTRL	控制	CTRL 寄存器（偏移 = 0h）[复位 = 00000260h]
4h	重映射	重映射	REMAP 寄存器（偏移 = 4h）[复位 = 20000000h]
8h	COMP0	比较器 0	COMP0 寄存器（偏移 = 8h）[复位 = 00000000h]
Ch	COMP1	比较器 1	COMP1 寄存器（偏移 = Ch）[复位 = 00000000h]
10h	COMP2	比较器 2	COMP2 寄存器（偏移 = 10h）[复位 = 00000000h]
14h	COMP3	比较器 3	COMP3 寄存器（偏移 = 14h）[复位 = 00000000h]
18h	COMP4	比较器 4	COMP4 寄存器（偏移 = 18h）[复位 = 00000000h]
1Ch	COMP5	比较器 5	COMP5 寄存器（偏移 = 1Ch）[复位 = 00000000h]
20h	COMP6	比较器 6	COMP6 寄存器（偏移 = 20h）[复位 = 00000000h]
24h	COMP7	比较器 7	COMP7 寄存器（偏移 = 24h）[复位 = 00000000h]

复杂的位访问类型经过编码可适应小型表单元。表 2-56 展示了适用于此部分中访问类型的代码。

表 2-56. CPU_FPB 访问类型代码

访问类型	代码	说明
读取类型		
R	R	读取
写入类型		
W	W	写入
复位或默认值		
-n		复位后的值或默认值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.2.1 CTRL 寄存器（偏移 = 0h）[复位 = 00000260h]

图 2-25 展示了 CTRL，表 2-57 中对此进行了介绍。

返回到汇总表。

控制
此寄存器用于启用闪存补丁块。

图 2-25. CTRL 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED		NUM_CODE2		NUM_LIT			
R-0h		R-0h		R-2h			
7	6	5	4	3	2	1	0
NUM_CODE1				RESERVED		KEY	ENABLE
R-6h				R-0h		W-0h	R/W-0h

表 2-57. CTRL 寄存器字段说明

位	字段	类型	复位	说明
31-14	RESERVED	R	0h	软件不应依赖保留的值。写入除复位值之外的任何其他值都可能导致未定义行为。
13-12	NUM_CODE2	R	0h	完整代码比较器组的数量，每组十六个比较器。如果提供的代码比较器少于十六个，则组计数为零，存在的数字由 NUM_CODE1 指示。此只读字段包含 3'b000，指示 Cortex-M 处理器的 0 个存储体。
11-8	NUM_LIT	R	2h	文字槽数目字段。 0x0：无文字槽 0x2：两个文字槽
7-4	NUM_CODE1	R	6h	代码槽数量字段。 0x0：无代码槽 0x2：两个代码槽 0x6：六个代码槽
3-2	RESERVED	R	0h	软件不应依赖保留的值。写入除复位值之外的任何其他值都可能导致未定义行为。
1	KEY	W	0h	关键字段。为了写入该寄存器，必须对该位字段写入“1”。该位始终读取 0。
0	ENABLE	R/W	0h	闪存补丁单元使能位 0x0：禁用闪存补丁单元 0x1：已启用闪存补丁单元

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.2.2 REMAP 寄存器（偏移 = 4h）[复位 = 20000000h]

图 2-26 展示了 REMAP，表 2-58 中对此进行了介绍。

返回到汇总表。

重映射
此寄存器提供了重映射匹配地址的基址位置。重映射基址的三个最高有效位和五个最低有效位分别硬编码为 3'b001 和 5'b00000。重映射基地址必须位于系统空间中，并且需要 8 字对齐，并将一个字分配给八个 FPB 比较器中的每一个。

图 2-26. REMAP 寄存器

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

表 2-58. REMAP 寄存器字段说明

位	字段	类型	复位	说明
31-29	RESERVED	R	1h	该字段的读数始终是 3'b001。写入该字段将被忽略。
28-5	重映射	R/W	X	重映射基址字段。
4-0	RESERVED	R	0h	该字段的读数始终是 0。写入该字段将被忽略。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.2.3 COMP0 寄存器（偏移 = 8h）[复位 = 00000000h]

图 2-27 展示了 COMP0，表 2-59 对其进行了介绍。

返回到汇总表。

比较器 0

图 2-27. COMP0 寄存器

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

表 2-59. COMP0 寄存器字段说明

位	字段	类型	复位	说明
31-30	REPLACE	R/W	0h	这用于选择 COMP 地址匹配时发生的情况。地址重映射仅在 0x0 设置中发生。 0x0：重映射到重映射地址。请参阅 REMAP.REMAP 0x1：在下半字上设置 BKPT，上限不受影响 0x2：在上半字上设置 BKPT，下半字不受影响 0x3：在下半字及上半字上设置 BKPT。
29	RESERVED	R/W	0h	软件不应依赖保留的值。写入除复位值之外的任何其他值都可能导致未定义行为。
28-2	COMP	R/W	0h	比较地址。
1	RESERVED	R/W	0h	软件不应依赖保留的值。写入除复位值之外的任何其他值都可能导致未定义行为。
0	ENABLE	R/W	0h	比较和重映射启用比较器 0。还必须设置 CTRL.ENABLE 以启用比较。 0x0：禁用比较器 0 的比较和重映射 0x1：比较器 0 的比较和重映射已启用

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.2.4 COMP1 寄存器（偏移 = Ch）[复位 = 00000000h]

图 2-28 展示了 COMP1，表 2-60 对其进行了介绍。

返回到汇总表。

比较器 1

图 2-28. COMP1 寄存器

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

表 2-60. COMP1 寄存器字段说明

位	字段	类型	复位	说明
31-30	REPLACE	R/W	0h	这用于选择 COMP 地址匹配时发生的情况。地址重映射仅在 0x0 设置中发生。 0x0：重映射到重映射地址。请参阅 REMAP.REMAP 0x1：在下半字上设置 BKPT，上限不受影响 0x2：在上半字上设置 BKPT，下半字不受影响 0x3：在下半字及上半字上设置 BKPT。
29	RESERVED	R/W	0h	软件不应依赖保留的值。写入除复位值之外的任何其他值都可能导致未定义行为。
28-2	COMP	R/W	0h	比较地址。
1	RESERVED	R/W	0h	软件不应依赖保留的值。写入除复位值之外的任何其他值都可能导致未定义行为。
0	ENABLE	R/W	0h	比较和重映射启用比较器 1。还必须设置 CTRL.ENABLE 以启用比较。 0x0：禁用比较器 1 的比较和重映射 0x1：比较器 1 的比较和重映射已启用

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.2.5 COMP2 寄存器（偏移 = 10h）[复位 = 00000000h]

图 2-29 展示了 COMP2，表 2-61 对其进行了介绍。

返回到汇总表。

比较器 2

图 2-29. COMP2 寄存器

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

表 2-61. COMP2 寄存器字段说明

位	字段	类型	复位	说明
31-30	REPLACE	R/W	0h	这用于选择 COMP 地址匹配时发生的情况。地址重映射仅在 0x0 设置中发生。 0x0：重映射到重映射地址。请参阅 REMAP.REMAP 0x1：在下半字上设置 BKPT，上限不受影响 0x2：在上半字上设置 BKPT，下半字不受影响 0x3：在下半字及上半字上设置 BKPT。
29	RESERVED	R/W	0h	软件不应依赖保留的值。写入除复位值之外的任何其他值都可能导致未定义行为。
28-2	COMP	R/W	0h	比较地址。
1	RESERVED	R/W	0h	软件不应依赖保留的值。写入除复位值之外的任何其他值都可能导致未定义行为。
0	ENABLE	R/W	0h	比较和重映射启用比较器 2。还必须设置 CTRL.ENABLE 以启用比较。 0x0：禁用比较器 2 的比较和重映射 0x1：比较器 2 的比较和重映射已启用

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.2.6 COMP3 寄存器（偏移 = 14h）[复位 = 00000000h]

图 2-30 展示了 COMP3，表 2-62 对其进行了介绍。

返回到汇总表。

比较器 3

图 2-30. COMP3 寄存器

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

表 2-62. COMP3 寄存器字段说明

位	字段	类型	复位	说明
31-30	REPLACE	R/W	0h	这用于选择 COMP 地址匹配时发生的情况。地址重映射仅在 0x0 设置中发生。 0x0：重映射到重映射地址。请参阅 REMAP.REMAP 0x1：在下半字上设置 BKPT，上限不受影响 0x2：在上半字上设置 BKPT，下半字不受影响 0x3：在下半字及上半字上设置 BKPT。
29	RESERVED	R/W	0h	软件不应依赖保留的值。写入除复位值之外的任何其他值都可能导致未定义行为。
28-2	COMP	R/W	0h	比较地址。
1	RESERVED	R/W	0h	软件不应依赖保留的值。写入除复位值之外的任何其他值都可能导致未定义行为。
0	ENABLE	R/W	0h	比较和重映射启用比较器 3。还必须设置 CTRL.ENABLE 以启用比较。 0x0：禁用比较器 3 的比较和重映射 0x1：比较器 3 的比较和重映射已启用

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.2.7 COMP4 寄存器（偏移 = 18h）[复位 = 00000000h]

图 2-31 展示了 COMP4，表 2-63 对其进行了介绍。

返回到汇总表。

比较器 4

图 2-31. COMP4 寄存器

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

表 2-63. COMP4 寄存器字段说明

位	字段	类型	复位	说明
31-30	REPLACE	R/W	0h	这用于选择 COMP 地址匹配时发生的情况。地址重映射仅在 0x0 设置中发生。 0x0：重映射到重映射地址。请参阅 REMAP.REMAP 0x1：在下半字上设置 BKPT，上限不受影响 0x2：在上半字上设置 BKPT，下半字不受影响 0x3：在下半字及上半字上设置 BKPT。
29	RESERVED	R/W	0h	软件不应依赖保留的值。写入除复位值之外的任何其他值都可能导致未定义行为。
28-2	COMP	R/W	0h	比较地址。
1	RESERVED	R/W	0h	软件不应依赖保留的值。写入除复位值之外的任何其他值都可能导致未定义行为。
0	ENABLE	R/W	0h	比较和重映射启用比较器 4。还必须设置 CTRL.ENABLE 以启用比较。 0x0：禁用比较器 4 的比较和重映射 0x1：比较器 4 的比较和重映射已启用

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.2.8 COMP5 寄存器（偏移 = 1Ch）[复位 = 00000000h]

图 2-32 展示了 COMP5，表 2-64 对其进行了介绍。

返回到汇总表。

比较器 5

图 2-32. COMP5 寄存器

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

表 2-64. COMP5 寄存器字段说明

位	字段	类型	复位	说明
31-30	REPLACE	R/W	0h	这用于选择 COMP 地址匹配时发生的情况。地址重映射仅在 0x0 设置中发生。 0x0：重映射到重映射地址。请参阅 REMAP.REMAP 0x1：在下半字上设置 BKPT，上限不受影响 0x2：在上半字上设置 BKPT，下半字不受影响 0x3：在下半字及上半字上设置 BKPT。
29	RESERVED	R/W	0h	软件不应依赖保留的值。写入除复位值之外的任何其他值都可能导致未定义行为。
28-2	COMP	R/W	0h	比较地址。
1	RESERVED	R/W	0h	软件不应依赖保留的值。写入除复位值之外的任何其他值都可能导致未定义行为。
0	ENABLE	R/W	0h	比较和重映射启用比较器 5。还必须设置 CTRL.ENABLE 以启用比较。 0x0：禁用比较器 5 的比较和重映射 0x1：比较器 5 的比较和重映射已启用

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.2.9 COMP6 寄存器（偏移 = 20h）[复位 = 00000000h]

图 2-33 展示了 COMP6，表 2-65 对其进行了介绍。

返回到汇总表。

比较器 6

图 2-33. COMP6 寄存器

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

表 2-65. COMP6 寄存器字段说明

位	字段	类型	复位	说明
31-30	REPLACE	R/W	0h	这用于选择 COMP 地址匹配时发生的情况。比较器 6 是字面比较器，唯一支持的设置是 0x0。其他设置将被忽略。 0x0：重映射到重映射地址。请参阅 REMAP.REMAP 0x1：在下半字上设置 BKPT，上限不受影响 0x2：在上半字上设置 BKPT，下半字不受影响 0x3：在下半字及上半字上设置 BKPT。
29	RESERVED	R/W	0h	软件不应依赖保留的值。写入除复位值之外的任何其他值都可能导致未定义行为。
28-2	COMP	R/W	0h	比较地址。
1	RESERVED	R/W	0h	软件不应依赖保留的值。写入除复位值之外的任何其他值都可能导致未定义行为。
0	ENABLE	R/W	0h	比较和重映射启用比较器 6。还必须设置 CTRL.ENABLE 以启用比较。 0x0：禁用比较器 6 的比较和重映射 0x1：比较器 6 的比较和重映射已启用

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.2.10 COMP7 寄存器（偏移 = 24h）[复位 = 00000000h]

图 2-34 展示了 COMP7，表 2-66 对其进行了介绍。

返回到汇总表。

比较器 7

图 2-34. COMP7 寄存器

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

表 2-66. COMP7 寄存器字段说明

位	字段	类型	复位	说明
31-30	REPLACE	R/W	0h	这用于选择 COMP 地址匹配时发生的情况。比较器 7 是字面比较器，唯一支持的设置是 0x0。其他设置将被忽略。 0x0：重映射到重映射地址。请参阅 REMAP.REMAP 0x1：在下半字上设置 BKPT，上限不受影响 0x2：在上半字上设置 BKPT，下半字不受影响 0x3：在下半字及上半字上设置 BKPT。
29	RESERVED	R/W	0h	软件不应依赖保留的值。写入除复位值之外的任何其他值都可能导致未定义行为。
28-2	COMP	R/W	0h	比较地址。
1	RESERVED	R/W	0h	软件不应依赖保留的值。写入除复位值之外的任何其他值都可能导致未定义行为。
0	ENABLE	R/W	0h	比较和重映射启用比较器 7。还必须设置 CTRL.ENABLE 以启用比较。 0x0：禁用比较器 7 的比较和重映射 0x1：比较器 7 的比较和重映射已启用

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.3 CPU_ITM 寄存器

表 2-67 列出了 CPU_ITM 寄存器的存储器映射寄存器。表 2-67 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

表 2-67. CPU_ITM 寄存器

偏移	首字母缩写词	寄存器名称	部分
0h	STIM0	激励端口 0	STIM0 寄存器 (偏移 = 0h) [复位 = 00000000h]
4h	STIM1	激励端口 1	STIM1 寄存器 (偏移 = 4h) [复位 = 00000000h]
8h	STIM2	激励端口 2	STIM2 寄存器 (偏移 = 8h) [复位 = 00000000h]
Ch	STIM3	激励端口 3	STIM3 寄存器 (偏移 = Ch) [复位 = 00000000h]
10h	STIM4	激励端口 4	STIM4 寄存器 (偏移 = 10h) [复位 = 00000000h]
14h	STIM5	激励端口 5	STIM5 寄存器 (偏移 = 14h) [复位 = 00000000h]
18h	STIM6	激励端口 6	STIM6 寄存器 (偏移 = 18h) [复位 = 00000000h]
1Ch	STIM7	激励端口 7	STIM7 寄存器 (偏移 = 1Ch) [复位 = 00000000h]
20h	STIM8	激励端口 8	STIM8 寄存器 (偏移 = 20h) [复位 = 00000000h]
24h	STIM9	激励端口 9	STIM9 寄存器 (偏移 = 24h) [复位 = 00000000h]
28h	STIM10	激励端口 10	STIM10 寄存器 (偏移 = 28h) [复位 = 00000000h]
2Ch	STIM11	激励端口 11	STIM11 寄存器 (偏移 = 2Ch) [复位 = 00000000h]
30h	STIM12	激励端口 12	STIM12 寄存器 (偏移 = 30h) [复位 = 00000000h]
34h	STIM13	激励端口 13	STIM13 寄存器 (偏移 = 34h) [复位 = 00000000h]
38h	STIM14	激励端口 14	STIM14 寄存器 (偏移 = 38h) [复位 = 00000000h]
3Ch	STIM15	激励端口 15	STIM15 寄存器 (偏移 = 3Ch) [复位 = 00000000h]
40h	STIM16	激励端口 16	STIM16 寄存器 (偏移 = 40h) [复位 = 00000000h]
44h	STIM17	激励端口 17	STIM17 寄存器 (偏移 = 44h) [复位 = 00000000h]
48h	STIM18	激励端口 18	STIM18 寄存器 (偏移 = 48h) [复位 = 00000000h]
4Ch	STIM19	激励端口 19	STIM19 寄存器 (偏移 = 4Ch) [复位 = 00000000h]
50h	STIM20	激励端口 20	STIM20 寄存器 (偏移 = 50h) [复位 = 00000000h]
54h	STIM21	激励端口 21	STIM21 寄存器 (偏移 = 54h) [复位 = 00000000h]
58h	STIM22	激励端口 22	STIM22 寄存器 (偏移 = 58h) [复位 = 00000000h]

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 2-67. CPU_ITM 寄存器（续）

偏移	首字母缩写词	寄存器名称	部分
5Ch	STIM23	激励端口 23	STIM23 寄存器（偏移 = 5Ch）[复位 = 00000000h]
60h	STIM24	激励端口 24	STIM24 寄存器（偏移 = 60h）[复位 = 00000000h]
64h	STIM25	激励端口 25	STIM25 寄存器（偏移 = 64h）[复位 = 00000000h]
68h	STIM26	激励端口 26	STIM26 寄存器（偏移 = 68h）[复位 = 00000000h]
6Ch	STIM27	激励端口 27	STIM27 寄存器（偏移 = 6Ch）[复位 = 00000000h]
70h	STIM28	激励端口 28	STIM28 寄存器（偏移 = 70h）[复位 = 00000000h]
74h	STIM29	激励端口 29	STIM29 寄存器（偏移 = 74h）[复位 = 00000000h]
78h	STIM30	激励端口 30	STIM30 寄存器（偏移 = 78h）[复位 = 00000000h]
7Ch	STIM31	激励端口 31	STIM31 寄存器（偏移 = 7Ch）[复位 = 00000000h]
E00h	TER	跟踪使能	TER 寄存器（偏移 = E00h）[复位 = 00000000h]
E40h	TPR	跟踪权限	TPR 寄存器（偏移 = E40h）[复位 = 00000000h]
E80h	TCR	跟踪控制	TCR 寄存器（偏移 = E80h）[复位 = 00000000h]
FB0h	LAR	锁定访问	LAR 寄存器（偏移 = FB0h）[复位 = 00000000h]
FB4h	LSR	锁定状态	LSR 寄存器（偏移 = FB4h）[复位 = 00000003h]

复杂的位访问类型经过编码可适应小型表单元。表 2-68 展示了适用于此部分中访问类型的代码。

表 2-68. CPU_ITM 访问类型代码

访问类型	代码	说明
读取类型		
R	R	读取
写入类型		
W	W	写入
复位或默认值		
-n		复位后的值或默认值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.3.1 STIM0 寄存器（偏移 = 0h）[复位 = 00000000h]

图 2-35 展示了 STIM0，表 2-69 对其进行了介绍。

返回到汇总表。

激励端口 0

图 2-35. STIM0 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STIM0																															
R/W-X																															

表 2-69. STIM0 寄存器字段说明

位	字段	类型	复位	说明
31-0	STIM0	R/W	X	如果设置了 TER.STIMENA0，则写入此位置会导致数据写入 FIFO。从激励端口读取数据时，返回值的位 [0] 表示 FIFO 状态：0 = 已满，1 = 未满。轮询 FIFO 接口不支持原子读-改-写操作。因此，当该 ITM 端口同时被中断服务程序或其他线程使用时，用户必须自行确保读-改-写操作具有独占性。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.3.2 STIM1 寄存器（偏移 = 4h）[复位 = 00000000h]

图 2-36 展示了 STIM1，表 2-70 对其进行了介绍。

返回到汇总表。

激励端口 1

图 2-36. STIM1 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STIM1																															
R/W-X																															

表 2-70. STIM1 寄存器字段说明

位	字段	类型	复位	说明
31-0	STIM1	R/W	X	如果设置了 TER.STIMENA1，则写入此位置会导致数据写入 FIFO。从激励端口读取数据时，返回值的位 [0] 表示 FIFO 状态：0 = 已满，1 = 未满。轮询 FIFO 接口不支持原子读-改-写操作。因此，当该 ITM 端口同时被中断服务程序或其他线程使用时，用户必须自行确保读-改-写操作具有独占性。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.3.3 STIM2 寄存器（偏移 = 8h）[复位 = 00000000h]

图 2-37 展示了 STIM2，表 2-71 对其进行了介绍。

返回到汇总表。

激励端口 2

图 2-37. STIM2 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STIM2																															
R/W-X																															

表 2-71. STIM2 寄存器字段说明

位	字段	类型	复位	说明
31-0	STIM2	R/W	X	如果设置了 TER.STIMENA2，则写入此位置会导致数据写入 FIFO。从激励端口读取数据时，返回值的位 [0] 表示 FIFO 状态：0 = 已满，1 = 未满。轮询 FIFO 接口不支持原子读-改-写操作。因此，当该 ITM 端口同时被中断服务程序或其他线程使用时，用户必须自行确保读-改-写操作具有独占性。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.3.4 STIM3 寄存器（偏移 = Ch）[复位 = 00000000h]

图 2-38 展示了 STIM3，表 2-72 对其进行了介绍。

返回到汇总表。

激励端口 3

图 2-38. STIM3 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STIM3																															
R/W-X																															

表 2-72. STIM3 寄存器字段说明

位	字段	类型	复位	说明
31-0	STIM3	R/W	X	如果设置了 TER.STIMENA3，则写入此位置会导致数据写入 FIFO。从激励端口读取数据时，返回值的位 [0] 表示 FIFO 状态：0 = 已满，1 = 未满。轮询 FIFO 接口不支持原子读-改-写操作。因此，当该 ITM 端口同时被中断服务程序或其他线程使用时，用户必须自行确保读-改-写操作具有独占性。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.3.5 STIM4 寄存器（偏移 = 10h）[复位 = 00000000h]

图 2-39 展示了 STIM4，表 2-73 对其进行了介绍。

返回到汇总表。

激励端口 4

图 2-39. STIM4 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STIM4																															
R/W-X																															

表 2-73. STIM4 寄存器字段说明

位	字段	类型	复位	说明
31-0	STIM4	R/W	X	如果设置了 TER.STIMENA4，则写入此位置会导致数据写入 FIFO。从激励端口读取数据时，返回值的位 [0] 表示 FIFO 状态：0 = 已满，1 = 未满。轮询 FIFO 接口不支持原子读-改-写操作。因此，当该 ITM 端口同时被中断服务程序或其他线程使用时，用户必须自行确保读-改-写操作具有独占性。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.3.6 STIM5 寄存器（偏移 = 14h）[复位 = 00000000h]

图 2-40 展示了 STIM5，表 2-74 对其进行了介绍。

返回到汇总表。

激励端口 5

图 2-40. STIM5 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STIM5																															
R/W-X																															

表 2-74. STIM5 寄存器字段说明

位	字段	类型	复位	说明
31-0	STIM5	R/W	X	如果设置了 TER.STIMENA5，则写入此位置会导致数据写入 FIFO。从激励端口读取数据时，返回值的位 [0] 表示 FIFO 状态：0 = 已满，1 = 未满。轮询 FIFO 接口不支持原子读-改-写操作。因此，当该 ITM 端口同时被中断服务程序或其他线程使用时，用户必须自行确保读-改-写操作具有独占性。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.3.7 STIM6 寄存器（偏移 = 18h）[复位 = 00000000h]

图 2-41 展示了 STIM6，表 2-75 对其进行了介绍。

返回到汇总表。

激励端口 6

图 2-41. STIM6 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STIM6																															
R/W-X																															

表 2-75. STIM6 寄存器字段说明

位	字段	类型	复位	说明
31-0	STIM6	R/W	X	如果设置了 TER.STIMENA6，则写入此位置会导致数据写入 FIFO。从激励端口读取数据时，返回值的位 [0] 表示 FIFO 状态：0 = 已满，1 = 未满。轮询 FIFO 接口不支持原子读-改-写操作。因此，当该 ITM 端口同时被中断服务程序或其他线程使用时，用户必须自行确保读-改-写操作具有独占性。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.3.8 STIM7 寄存器（偏移 = 1Ch）[复位 = 00000000h]

图 2-42 展示了 STIM7，表 2-76 对其进行了介绍。

返回到汇总表。

激励端口 7

图 2-42. STIM7 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STIM7																															
R/W-X																															

表 2-76. STIM7 寄存器字段说明

位	字段	类型	复位	说明
31-0	STIM7	R/W	X	如果设置了 TER.STIMENA7，则写入此位置会导致数据写入 FIFO。从激励端口读取数据时，返回值的位 [0] 表示 FIFO 状态：0 = 已满，1 = 未满。轮询 FIFO 接口不支持原子读-改-写操作。因此，当该 ITM 端口同时被中断服务程序或其他线程使用时，用户必须自行确保读-改-写操作具有独占性。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.3.9 STIM8 寄存器（偏移 = 20h）[复位 = 00000000h]

图 2-43 展示了 STIM8，表 2-77 对其进行了介绍。

返回到汇总表。

激励端口 8

图 2-43. STIM8 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STIM8																															
R/W-X																															

表 2-77. STIM8 寄存器字段说明

位	字段	类型	复位	说明
31-0	STIM8	R/W	X	如果设置了 TER.STIMENA8，则写入此位置会导致数据写入 FIFO。从激励端口读取数据时，返回值的位 [0] 表示 FIFO 状态：0 = 已满，1 = 未满。轮询 FIFO 接口不支持原子读-改-写操作。因此，当该 ITM 端口同时被中断服务程序或其他线程使用时，用户必须自行确保读-改-写操作具有独占性。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.3.10 STIM9 寄存器（偏移 = 24h）[复位 = 00000000h]

图 2-44 展示了 STIM9，表 2-78 对其进行了介绍。

返回到汇总表。

激励端口 9

图 2-44. STIM9 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STIM9																															
R/W-X																															

表 2-78. STIM9 寄存器字段说明

位	字段	类型	复位	说明
31-0	STIM9	R/W	X	如果设置了 TER.STIMENA9，则写入此位置会导致数据写入 FIFO。从激励端口读取数据时，返回值的位 [0] 表示 FIFO 状态：0 = 已满，1 = 未满。轮询 FIFO 接口不支持原子读-改-写操作。因此，当该 ITM 端口同时被中断服务程序或其他线程使用时，用户必须自行确保读-改-写操作具有独占性。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.3.11 STIM10 寄存器（偏移 = 28h）[复位 = 00000000h]

图 2-45 展示了 STIM10，表 2-79 对其进行了介绍。

返回到汇总表。

激励端口 10

图 2-45. STIM10 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STIM10																															
R/W-X																															

表 2-79. STIM10 寄存器字段说明

位	字段	类型	复位	说明
31-0	STIM10	R/W	X	如果设置了 TER.STIMENA10，则写入此位置会导致数据写入 FIFO。从激励端口读取数据时，返回值的位 [0] 表示 FIFO 状态：0 = 已满，1 = 未满。轮询 FIFO 接口不支持原子读-改-写操作。因此，当该 ITM 端口同时被中断服务程序或其他线程使用时，用户必须自行确保读-改-写操作具有独占性。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.3.12 STIM11 寄存器（偏移 = 2Ch）[复位 = 00000000h]

图 2-46 展示了 STIM11，表 2-80 对其进行了介绍。

返回到汇总表。

激励端口 11

图 2-46. STIM11 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STIM11																															
R/W-X																															

表 2-80. STIM11 寄存器字段说明

位	字段	类型	复位	说明
31-0	STIM11	R/W	X	如果设置了 TER.STIMENA11，则写入此位置会导致数据写入 FIFO。从激励端口读取数据时，返回值的位 [0] 表示 FIFO 状态：0 = 已满，1 = 未满。轮询 FIFO 接口不支持原子读-改-写操作。因此，当该 ITM 端口同时被中断服务程序或其他线程使用时，用户必须自行确保读-改-写操作具有独占性。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.3.13 STIM12 寄存器（偏移 = 30h）[复位 = 00000000h]

图 2-47 展示了 STIM12，表 2-81 对其进行了介绍。

返回到汇总表。

激励端口 12

图 2-47. STIM12 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STIM12																															
R/W-X																															

表 2-81. STIM12 寄存器字段说明

位	字段	类型	复位	说明
31-0	STIM12	R/W	X	如果设置了 TER.STIMENA12，则写入此位置会导致数据写入 FIFO。从激励端口读取数据时，返回值的位 [0] 表示 FIFO 状态：0 = 已满，1 = 未满。轮询 FIFO 接口不支持原子读-改-写操作。因此，当该 ITM 端口同时被中断服务程序或其他线程使用时，用户必须自行确保读-改-写操作具有独占性。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.3.14 STIM13 寄存器（偏移 = 34h）[复位 = 00000000h]

图 2-48 展示了 STIM13，表 2-82 对其进行了介绍。

返回到汇总表。

激励端口 13

图 2-48. STIM13 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STIM13																															
R/W-X																															

表 2-82. STIM13 寄存器字段说明

位	字段	类型	复位	说明
31-0	STIM13	R/W	X	如果设置了 TER.STIMENA13，则写入此位置会导致数据写入 FIFO。从激励端口读取数据时，返回值的位 [0] 表示 FIFO 状态：0 = 已满，1 = 未满。轮询 FIFO 接口不支持原子读-改-写操作。因此，当该 ITM 端口同时被中断服务程序或其他线程使用时，用户必须自行确保读-改-写操作具有独占性。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.3.15 STIM14 寄存器（偏移 = 38h）[复位 = 00000000h]

图 2-49 展示了 STIM14，表 2-83 对其进行了介绍。

返回到汇总表。

激励端口 14

图 2-49. STIM14 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STIM14																															
R/W-X																															

表 2-83. STIM14 寄存器字段说明

位	字段	类型	复位	说明
31-0	STIM14	R/W	X	如果设置了 TER.STIMENA14，则写入此位置会导致数据写入 FIFO。从激励端口读取数据时，返回值的位 [0] 表示 FIFO 状态：0 = 已满，1 = 未满。轮询 FIFO 接口不支持原子读-改-写操作。因此，当该 ITM 端口同时被中断服务程序或其他线程使用时，用户必须自行确保读-改-写操作具有独占性。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.3.16 STIM15 寄存器（偏移 = 3Ch）[复位 = 00000000h]

图 2-50 展示了 STIM15，表 2-84 对其进行了介绍。

返回到汇总表。

激励端口 15

图 2-50. STIM15 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STIM15																															
R/W-X																															

表 2-84. STIM15 寄存器字段说明

位	字段	类型	复位	说明
31-0	STIM15	R/W	X	如果设置了 TER.STIMENA15，则写入此位置会导致数据写入 FIFO。从激励端口读取数据时，返回值的位 [0] 表示 FIFO 状态：0 = 已满，1 = 未满。轮询 FIFO 接口不支持原子读-改-写操作。因此，当该 ITM 端口同时被中断服务程序或其他线程使用时，用户必须自行确保读-改-写操作具有独占性。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.3.17 STIM16 寄存器（偏移 = 40h）[复位 = 00000000h]

图 2-51 展示了 STIM16，表 2-85 对其进行了介绍。

返回到汇总表。

激励端口 16

图 2-51. STIM16 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STIM16																															
R/W-X																															

表 2-85. STIM16 寄存器字段说明

位	字段	类型	复位	说明
31-0	STIM16	R/W	X	如果设置了 TER.STIMENA16，则写入此位置会导致数据写入 FIFO。从激励端口读取数据时，返回值的位 [0] 表示 FIFO 状态：0 = 已满，1 = 未满。轮询 FIFO 接口不支持原子读-改-写操作。因此，当该 ITM 端口同时被中断服务程序或其他线程使用时，用户必须自行确保读-改-写操作具有独占性。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.3.18 STIM17 寄存器（偏移 = 44h）[复位 = 00000000h]

图 2-52 展示了 STIM17，表 2-86 对其进行了介绍。

返回到汇总表。

激励端口 17

图 2-52. STIM17 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STIM17																															
R/W-X																															

表 2-86. STIM17 寄存器字段说明

位	字段	类型	复位	说明
31-0	STIM17	R/W	X	如果设置了 TER.STIMENA17，则写入此位置会导致数据写入 FIFO。从激励端口读取数据时，返回值的位 [0] 表示 FIFO 状态：0 = 已满，1 = 未满。轮询 FIFO 接口不支持原子读-改-写操作。因此，当该 ITM 端口同时被中断服务程序或其他线程使用时，用户必须自行确保读-改-写操作具有独占性。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.3.19 STIM18 寄存器（偏移 = 48h）[复位 = 00000000h]

图 2-53 展示了 STIM18，表 2-87 对其进行了介绍。

返回到汇总表。

激励端口 18

图 2-53. STIM18 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STIM18																															
R/W-X																															

表 2-87. STIM18 寄存器字段说明

位	字段	类型	复位	说明
31-0	STIM18	R/W	X	如果设置了 TER.STIMENA18，则写入此位置会导致数据写入 FIFO。从激励端口读取数据时，返回值的位 [0] 表示 FIFO 状态：0 = 已满，1 = 未满。轮询 FIFO 接口不支持原子读-改-写操作。因此，当该 ITM 端口同时被中断服务程序或其他线程使用时，用户必须自行确保读-改-写操作具有独占性。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.3.20 STIM19 寄存器（偏移 = 4Ch）[复位 = 00000000h]

图 2-54 展示了 STIM19，表 2-88 对其进行了介绍。

返回到汇总表。

激励端口 19

图 2-54. STIM19 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STIM19																															
R/W-X																															

表 2-88. STIM19 寄存器字段说明

位	字段	类型	复位	说明
31-0	STIM19	R/W	X	如果设置了 TER.STIMENA19，则写入此位置会导致数据写入 FIFO。从激励端口读取数据时，返回值的位 [0] 表示 FIFO 状态：0 = 已满，1 = 未满。轮询 FIFO 接口不支持原子读-改-写操作。因此，当该 ITM 端口同时被中断服务程序或其他线程使用时，用户必须自行确保读-改-写操作具有独占性。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.3.21 STIM20 寄存器（偏移 = 50h）[复位 = 00000000h]

图 2-55 展示了 STIM20，表 2-89 对其进行了介绍。

返回到汇总表。

激励端口 20

图 2-55. STIM20 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STIM20																															
R/W-X																															

表 2-89. STIM20 寄存器字段说明

位	字段	类型	复位	说明
31-0	STIM20	R/W	X	如果设置了 TER.STIMENA20，则写入此位置会导致数据写入 FIFO。从激励端口读取数据时，返回值的位 [0] 表示 FIFO 状态：0 = 已满，1 = 未满。轮询 FIFO 接口不支持原子读-改-写操作。因此，当该 ITM 端口同时被中断服务程序或其他线程使用时，用户必须自行确保读-改-写操作具有独占性。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.3.22 STIM21 寄存器（偏移 = 54h）[复位 = 00000000h]

图 2-56 展示了 STIM21，表 2-90 对其进行了介绍。

返回到汇总表。

激励端口 21

图 2-56. STIM21 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STIM21																															
R/W-X																															

表 2-90. STIM21 寄存器字段说明

位	字段	类型	复位	说明
31-0	STIM21	R/W	X	如果设置了 TER.STIMENA21，则写入此位置会导致数据写入 FIFO。从激励端口读取数据时，返回值的位 [0] 表示 FIFO 状态：0 = 已满，1 = 未满。轮询 FIFO 接口不支持原子读-改-写操作。因此，当该 ITM 端口同时被中断服务程序或其他线程使用时，用户必须自行确保读-改-写操作具有独占性。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.3.23 STIM22 寄存器（偏移 = 58h）[复位 = 00000000h]

图 2-57 展示了 STIM22，表 2-91 对其进行了介绍。

返回到汇总表。

激励端口 22

图 2-57. STIM22 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STIM22																															
R/W-X																															

表 2-91. STIM22 寄存器字段说明

位	字段	类型	复位	说明
31-0	STIM22	R/W	X	如果设置了 TER.STIMENA22，则写入此位置会导致数据写入 FIFO。从激励端口读取数据时，返回值的位 [0] 表示 FIFO 状态：0 = 已满，1 = 未满。轮询 FIFO 接口不支持原子读-改-写操作。因此，当该 ITM 端口同时被中断服务程序或其他线程使用时，用户必须自行确保读-改-写操作具有独占性。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.3.24 STIM23 寄存器（偏移 = 5Ch）[复位 = 00000000h]

图 2-58 展示了 STIM23，表 2-92 对其进行了介绍。

返回到汇总表。

激励端口 23

图 2-58. STIM23 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STIM23																															
R/W-X																															

表 2-92. STIM23 寄存器字段说明

位	字段	类型	复位	说明
31-0	STIM23	R/W	X	如果设置了 TER.STIMENA23，则写入此位置会导致数据写入 FIFO。从激励端口读取数据时，返回值的位 [0] 表示 FIFO 状态：0 = 已满，1 = 未满。轮询 FIFO 接口不支持原子读-改-写操作。因此，当该 ITM 端口同时被中断服务程序或其他线程使用时，用户必须自行确保读-改-写操作具有独占性。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.3.25 STIM24 寄存器（偏移 = 60h）[复位 = 00000000h]

图 2-59 展示了 STIM24，表 2-93 对其进行了介绍。

返回到汇总表。

激励端口 24

图 2-59. STIM24 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STIM24																															
R/W-X																															

表 2-93. STIM24 寄存器字段说明

位	字段	类型	复位	说明
31-0	STIM24	R/W	X	如果设置了 TER.STIMENA24，则写入此位置会导致数据写入 FIFO。从激励端口读取数据时，返回值的位 [0] 表示 FIFO 状态：0 = 已满，1 = 未满。轮询 FIFO 接口不支持原子读-改-写操作。因此，当该 ITM 端口同时被中断服务程序或其他线程使用时，用户必须自行确保读-改-写操作具有独占性。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.3.26 STIM25 寄存器（偏移 = 64h）[复位 = 00000000h]

图 2-60 展示了 STIM25，表 2-94 对其进行了介绍。

返回到汇总表。

激励端口 25

图 2-60. STIM25 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STIM25																															
R/W-X																															

表 2-94. STIM25 寄存器字段说明

位	字段	类型	复位	说明
31-0	STIM25	R/W	X	如果设置了 TER.STIMENA25，则写入此位置会导致数据写入 FIFO。从激励端口读取数据时，返回值的位 [0] 表示 FIFO 状态：0 = 已满，1 = 未满。轮询 FIFO 接口不支持原子读-改-写操作。因此，当该 ITM 端口同时被中断服务程序或其他线程使用时，用户必须自行确保读-改-写操作具有独占性。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.3.27 STIM26 寄存器（偏移 = 68h）[复位 = 00000000h]

图 2-61 展示了 STIM26，表 2-95 对其进行了介绍。

返回到汇总表。

激励端口 26

图 2-61. STIM26 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STIM26																															
R/W-X																															

表 2-95. STIM26 寄存器字段说明

位	字段	类型	复位	说明
31-0	STIM26	R/W	X	如果设置了 TER.STIMENA26，则写入此位置会导致数据写入 FIFO。从激励端口读取数据时，返回值的位 [0] 表示 FIFO 状态：0 = 已满，1 = 未满。轮询 FIFO 接口不支持原子读-改-写操作。因此，当该 ITM 端口同时被中断服务程序或其他线程使用时，用户必须自行确保读-改-写操作具有独占性。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.3.28 STIM27 寄存器（偏移 = 6Ch）[复位 = 00000000h]

图 2-62 展示了 STIM27，表 2-96 对其进行了介绍。

返回到汇总表。

激励端口 27

图 2-62. STIM27 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STIM27																															
R/W-X																															

表 2-96. STIM27 寄存器字段说明

位	字段	类型	复位	说明
31-0	STIM27	R/W	X	如果设置了 TER.STIMENA27，则写入此位置会导致数据写入 FIFO。从激励端口读取数据时，返回值的位 [0] 表示 FIFO 状态：0 = 已满，1 = 未满。轮询 FIFO 接口不支持原子读-改-写操作。因此，当该 ITM 端口同时被中断服务程序或其他线程使用时，用户必须自行确保读-改-写操作具有独占性。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.3.29 STIM28 寄存器（偏移 = 70h）[复位 = 00000000h]

图 2-63 展示了 STIM28，表 2-97 对其进行了介绍。

返回到汇总表。

激励端口 28

图 2-63. STIM28 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STIM28																															
R/W-X																															

表 2-97. STIM28 寄存器字段说明

位	字段	类型	复位	说明
31-0	STIM28	R/W	X	如果设置了 TER.STIMENA28，则写入此位置会导致数据写入 FIFO。从激励端口读取数据时，返回值的位 [0] 表示 FIFO 状态：0 = 已满，1 = 未满。轮询 FIFO 接口不支持原子读-改-写操作。因此，当该 ITM 端口同时被中断服务程序或其他线程使用时，用户必须自行确保读-改-写操作具有独占性。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.3.30 STIM29 寄存器（偏移 = 74h）[复位 = 00000000h]

图 2-64 展示了 STIM29，表 2-98 对其进行了介绍。

返回到汇总表。

激励端口 29

图 2-64. STIM29 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STIM29																															
R/W-X																															

表 2-98. STIM29 寄存器字段说明

位	字段	类型	复位	说明
31-0	STIM29	R/W	X	如果设置了 TER.STIMENA29，则写入此位置会导致数据写入 FIFO。从激励端口读取数据时，返回值的位 [0] 表示 FIFO 状态：0 = 已满，1 = 未满。轮询 FIFO 接口不支持原子读-改-写操作。因此，当该 ITM 端口同时被中断服务程序或其他线程使用时，用户必须自行确保读-改-写操作具有独占性。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.3.31 STIM30 寄存器（偏移 = 78h）[复位 = 00000000h]

图 2-65 展示了 STIM30，表 2-99 对其进行了介绍。

返回到汇总表。

激励端口 30

图 2-65. STIM30 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STIM30																															
R/W-X																															

表 2-99. STIM30 寄存器字段说明

位	字段	类型	复位	说明
31-0	STIM30	R/W	X	如果设置了 TER.STIMENA30，则写入此位置会导致数据写入 FIFO。从激励端口读取数据时，返回值的位 [0] 表示 FIFO 状态：0 = 已满，1 = 未满。轮询 FIFO 接口不支持原子读-改-写操作。因此，当该 ITM 端口同时被中断服务程序或其他线程使用时，用户必须自行确保读-改-写操作具有独占性。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.3.32 STIM31 寄存器（偏移 = 7Ch）[复位 = 00000000h]

图 2-66 展示了 STIM31，表 2-100 对其进行了介绍。

返回到汇总表。

激励端口 31

图 2-66. STIM31 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STIM31																															
R/W-X																															

表 2-100. STIM31 寄存器字段说明

位	字段	类型	复位	说明
31-0	STIM31	R/W	X	如果设置了 TER.STIMENA31，则写入此位置会导致数据写入 FIFO。从激励端口读取数据时，返回值的位 [0] 表示 FIFO 状态：0 = 已满，1 = 未满。轮询 FIFO 接口不支持原子读-改-写操作。因此，当该 ITM 端口同时被中断服务程序或其他线程使用时，用户必须自行确保读-改-写操作具有独占性。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.3.33 TER 寄存器（偏移 = E00h）[复位 = 00000000h]

图 2-67 展示了 TER，表 2-101 中对此进行了介绍。

返回到汇总表。

跟踪使能
通过写入相应的激励端口，使用跟踪使能寄存器生成跟踪数据。注意：如果设置了 TCR.ITMENA，则接受对该寄存器的特权写入。如果设置了 TCR.ITMENA 并清除了相应的权限掩码，则接受用户对该寄存器的写入。对激励端口的特权访问使 RTOS 内核能够根据需要保障调试信息通道或带宽资源分配。

图 2-67. TER 寄存器

31	30	29	28	27	26	25	24
STIMENA31	STIMENA30	STIMENA29	STIMENA28	STIMENA27	STIMENA26	STIMENA25	STIMENA24
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
STIMENA23	STIMENA22	STIMENA21	STIMENA20	STIMENA19	STIMENA18	STIMENA17	STIMENA16
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
STIMENA15	STIMENA14	STIMENA13	STIMENA12	STIMENA11	STIMENA10	STIMENA9	STIMENA8
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
STIMENA7	STIMENA6	STIMENA5	STIMENA4	STIMENA3	STIMENA2	STIMENA1	STIMENA0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 2-101. TER 寄存器字段说明

位	字段	类型	复位	说明
31	STIMENA31	R/W	0h	用于启用 ITM 激励端口 31 跟踪功能的位掩码。
30	STIMENA30	R/W	0h	用于启用 ITM 激励端口 30 跟踪功能的位掩码。
29	STIMENA29	R/W	0h	用于启用 ITM 激励端口 29 跟踪功能的位掩码。
28	STIMENA28	R/W	0h	用于启用 ITM 激励端口 28 跟踪功能的位掩码。
27	STIMENA27	R/W	0h	用于启用 ITM 激励端口 27 跟踪功能的位掩码。
26	STIMENA26	R/W	0h	用于启用 ITM 激励端口 26 跟踪功能的位掩码。
25	STIMENA25	R/W	0h	用于启用 ITM 激励端口 25 跟踪功能的位掩码。
24	STIMENA24	R/W	0h	用于启用 ITM 激励端口 24 跟踪功能的位掩码。
23	STIMENA23	R/W	0h	用于启用 ITM 激励端口 23 跟踪功能的位掩码。
22	STIMENA22	R/W	0h	用于启用 ITM 激励端口 22 跟踪功能的位掩码。
21	STIMENA21	R/W	0h	用于启用 ITM 激励端口 21 跟踪功能的位掩码。
20	STIMENA20	R/W	0h	用于启用 ITM 激励端口 20 跟踪功能的位掩码。
19	STIMENA19	R/W	0h	用于启用 ITM 激励端口 19 跟踪功能的位掩码。
18	STIMENA18	R/W	0h	用于启用 ITM 激励端口 18 跟踪功能的位掩码。
17	STIMENA17	R/W	0h	用于启用 ITM 激励端口 17 跟踪功能的位掩码。
16	STIMENA16	R/W	0h	用于启用 ITM 激励端口 16 跟踪功能的位掩码。
15	STIMENA15	R/W	0h	用于启用 ITM 激励端口 15 跟踪功能的位掩码。
14	STIMENA14	R/W	0h	用于启用 ITM 激励端口 14 跟踪功能的位掩码。
13	STIMENA13	R/W	0h	用于启用 ITM 激励端口 13 跟踪功能的位掩码。
12	STIMENA12	R/W	0h	用于启用 ITM 激励端口 12 跟踪功能的位掩码。
11	STIMENA11	R/W	0h	用于启用 ITM 激励端口 11 跟踪功能的位掩码。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 2-101. TER 寄存器字段说明（续）

位	字段	类型	复位	说明
10	STIMENA10	R/W	0h	用于启用 ITM 激励端口 10 跟踪功能的位掩码。
9	STIMENA9	R/W	0h	用于启用 ITM 激励端口 9 跟踪功能的位掩码。
8	STIMENA8	R/W	0h	用于启用 ITM 激励端口 8 跟踪功能的位掩码。
7	STIMENA7	R/W	0h	用于启用 ITM 激励端口 7 跟踪功能的位掩码。
6	STIMENA6	R/W	0h	用于启用 ITM 激励端口 6 跟踪功能的位掩码。
5	STIMENA5	R/W	0h	用于启用 ITM 激励端口 5 跟踪功能的位掩码。
4	STIMENA4	R/W	0h	用于启用 ITM 激励端口 4 跟踪功能的位掩码。
3	STIMENA3	R/W	0h	用于启用 ITM 激励端口 3 跟踪功能的位掩码。
2	STIMENA2	R/W	0h	用于启用 ITM 激励端口 2 跟踪功能的位掩码。
1	STIMENA1	R/W	0h	用于启用 ITM 激励端口 1 跟踪功能的位掩码。
0	STIMENA0	R/W	0h	用于启用 ITM 激励端口 0 跟踪功能的位掩码。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.3.34 TPR 寄存器（偏移 = E40h）[复位 = 00000000h]

图 2-68 展示了 TPR，表 2-102 中对此进行了介绍。

返回到汇总表。

跟踪特权
该寄存器用于使操作系统能够控制用户代码可访问的激励端口。该寄存器只能在特权模式下使用。

图 2-68. TPR 寄存器

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												PRIVMASK			
R/W-0h												R/W-0h			

表 2-102. TPR 寄存器字段说明

位	字段	类型	复位	说明
31-4	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
3-0	PRIVMASK	R/W	0h	用于使能 ITM 激励端口的非特权（用户）访问的位掩码： 位 [0] 使能激励端口 0、1、...和 7。 位 [1] 使能激励端口 8、9、...和 15。 位 [2] 使能激励端口 16、17、...和 23。 位 [3] 使能激励端口 24、25、...和 31。 0：允许用户访问激励端口 1：仅允许特权访问激励端口

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.3.35 TCR 寄存器（偏移 = E80h）[复位 = 00000000h]

图 2-69 展示了 TCR，表 2-103 中对此进行了介绍。

返回到汇总表。

跟踪控制
使用该寄存器配置和控制 ITM 传输。该寄存器只能在特权模式下写入。在 ITM 块中未启用 DWT。然而，DWTENA 控制 DWT 激励数据进入 FIFO。如果 DWT 需要时间戳功能，则必须设置 TSENA 位。

图 2-69. TCR 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
BUSY	ATBID						
R/W-0h	R/W-0h						
15	14	13	12	11	10	9	8
RESERVED						TSPRESCALE	
R/W-0h						R/W-0h	
7	6	5	4	3	2	1	0
RESERVED			SWOENA	DWTENA	SYNCENA	TSENA	ITMENA
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 2-103. TCR 寄存器字段说明

位	字段	类型	复位	说明
31-24	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
23	BUSY	R/W	0h	当 ITM 事件存在且正在被清空时设置。
22-16	ATBID	R/W	0h	CoreSight 系统的跟踪总线 ID。用于多源跟踪流格式化的可选标识符。如果使用多源跟踪，则该字段必须写入非零值。
15-10	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
9-8	TSPRESCALE	R/W	0h	时间戳预分频器 0h = 不进行预分频 1h = 4 分频 2h = 16 分频 3h = 64 分频
7-5	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
4	SWOENA	R/W	0h	启用时间戳计数器的异步时钟功能（当 TSENA = 1 时）。如果 TSENA = 0，将该位写为 1 不会启用时间戳计数器的异步时钟功能。 0x0：模式已禁用。时间戳计数器使用来自核心的系统时钟并连续计数。 0x1：时间戳计数器使用来自 TPIU 接口的线路输出（与数据相关）时钟。当输出线路空闲时，时间戳计数器保持在复位状态。
3	DWTENA	R/W	0h	启用 DWT 激励（从 DWT 向 TPIU 发送硬件事件数据包）
2	SYNCENA	R/W	0h	为同步 TPIU 启用同步数据包传输。 必须将 CPU_DWT:CTRL.SYNCTAP 配置为正确的同步速度。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 2-103. TCR 寄存器字段说明（续）

位	字段	类型	复位	说明
1	TSENA	R/W	0h	启用差分时间戳。当向 FIFO 写入带非零时间戳计数器的数据包，以及当时间戳计数器溢出时，会发送差分时间戳。时间戳会在空闲时发送，间隔固定为二百万个周期。这为数据包及数据包间间隔提供时间参考。如果设置了 SWOENA（位 [4]），时间戳仅由内部跟踪总线活动触发。在这种情况下，当 ITM 处于空闲状态时，不会有常规时间戳输出。
0	ITMENA	R/W	0h	启用 ITM。这是主使能位，必须先设置此位，才能写入 ITM 激励寄存器和跟踪使能寄存器。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.3.36 LAR 寄存器（偏移 = FB0h）[复位 = 00000000h]

图 2-70 展示了 LAR，表 2-104 中对此进行了介绍。

返回到汇总表。

锁定访问
该寄存器用于防止对以下控制寄存器的写入访问：TER、TPR 和 TCR。

图 2-70. LAR 寄存器

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

表 2-104. LAR 寄存器字段说明

位	字段	类型	复位	说明
31-0	LOCK_ACCESS	W	0h	对 0xC5ACCE55 的特权写入支持对控制寄存器 TER、TPR 和 TCR 进行更多写入访问。无效写入会移除写入访问权限。

Copyright © 2026 Texas Instruments Incorporated

2.9.3.37 LSR 寄存器（偏移 = FB4h）[复位 = 00000003h]

图 2-71 展示了 LSR，表 2-105 中对此进行了介绍。

返回到汇总表。

锁定状态
使用此寄存器启用对控制寄存器的写入访问。

图 2-71. LSR 寄存器

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

表 2-105. LSR 寄存器字段说明

位	字段	类型	复位	说明
31-3	RESERVED	R	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
2	BYTEACC	R	0h	读值为 0，表示未实现 8 位锁定访问。
1	ACCESS	R	1h	对元件的写入访问被阻止。所有写入都被忽略，允许读取。
0	PRESENT	R	1h	表示此元件存在锁定机制。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4 CPU_SCS 寄存器

表 2-106 列出了 CPU_SCS 寄存器的存储器映射寄存器。表 2-106 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

表 2-106. CPU_SCS 寄存器

偏移	首字母缩写词	寄存器名称	部分
4h	ICTR	中断控制类型	ICTR 寄存器 (偏移 = 4h) [复位 = 00000001h]
8h	ACTLR	辅助控制	ACTLR 寄存器 (偏移 = 8h) [复位 = 00000000h]
10h	STCSR	SysTick 控制和状态	STCSR 寄存器 (偏移 = 10h) [复位 = 00000004h]
14h	STRVR	SysTick 重载值	STRVR 寄存器 (偏移 = 14h) [复位 = 00000000h]
18h	STCVR	SysTick 当前值	STCVR 寄存器 (偏移 = 18h) [复位 = 00000000h]
1Ch	STCR	SysTick 校准值	STCR 寄存器 (偏移 = 1Ch) [复位 = C0075300h]
100h	NVIC_ISER0	Irq 0 至 31 设置使能	NVIC_ISER0 寄存器 (偏移 = 100h) [复位 = 00000000h]
104h	NVIC_ISER1	Irq 32 至 63 设置使能	NVIC_ISER1 寄存器 (偏移 = 104h) [复位 = 00000000h]
180h	NVIC_ICER0	Irq 0 至 31 清除使能	NVIC_ICER0 寄存器 (偏移 = 180h) [复位 = 00000000h]
184h	NVIC_ICER1	Irq 32 至 63 清除使能	NVIC_ICER1 寄存器 (偏移 = 184h) [复位 = 00000000h]
200h	NVIC_ISPR0	Irq 0 至 31 设置挂起	NVIC_ISPR0 寄存器 (偏移 = 200h) [复位 = 00000000h]
204h	NVIC_ISPR1	Irq 32 至 63 设置挂起	NVIC_ISPR1 寄存器 (偏移 = 204h) [复位 = 00000000h]
280h	NVIC_ICPR0	Irq 0 至 31 清除挂起	NVIC_ICPR0 寄存器 (偏移 = 280h) [复位 = 00000000h]
284h	NVIC_ICPR1	Irq 32 至 63 清除挂起	NVIC_ICPR1 寄存器 (偏移 = 284h) [复位 = 00000000h]
300h	NVIC_IABR0	Irq 0 至 31 活动位	NVIC_IABR0 寄存器 (偏移 = 300h) [复位 = 00000000h]
304h	NVIC_IABR1	Irq 32 至 63 活动位	NVIC_IABR1 寄存器 (偏移 = 304h) [复位 = 00000000h]
400h	NVIC_IPR0	Irq 0 至 3 的优先级	NVIC_IPR0 寄存器 (偏移 = 400h) [复位 = 00000000h]
404h	NVIC_IPR1	Irq 4 至 7 的优先级	NVIC_IPR1 寄存器 (偏移 = 404h) [复位 = 00000000h]
408h	NVIC_IPR2	Irq 8 至 11 的优先级	NVIC_IPR2 寄存器 (偏移 = 408h) [复位 = 00000000h]
40Ch	NVIC_IPR3	Irq 12 至 15 的优先级	NVIC_IPR3 寄存器 (偏移 = 40Ch) [复位 = 00000000h]
410h	NVIC_IPR4	Irq 16 至 19 的优先级	NVIC_IPR4 寄存器 (偏移 = 410h) [复位 = 00000000h]
414h	NVIC_IPR5	Irq 20 至 23 的优先级	NVIC_IPR5 寄存器 (偏移 = 414h) [复位 = 00000000h]
418h	NVIC_IPR6	Irq 24 至 27 的优先级	NVIC_IPR6 寄存器 (偏移 = 418h) [复位 = 00000000h]

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 2-106. CPU_SCS 寄存器（续）

偏移	首字母缩写词	寄存器名称	部分
41Ch	NVIC_IPR7	Irq 28 至 31 的优先级	NVIC_IPR7 寄存器（偏移 = 41Ch）[复位 = 00000000h]
420h	NVIC_IPR8	Irq 32 至 35 的优先级	NVIC_IPR8 寄存器（偏移 = 420h）[复位 = 00000000h]
424h	NVIC_IPR9	Irq 32 至 35 的优先级	NVIC_IPR9 寄存器（偏移 = 424h）[复位 = 00000000h]
D00h	CPUID	CPUID 基地址	CPUID 寄存器（偏移 = D00h）[复位 = 410FC241h]
D04h	ICSR	中断控制状态	ICSR 寄存器（偏移 = D04h）[复位 = 00000000h]
D08h	VTOR	矢量表偏移	VTOR 寄存器（偏移 = D08h）[复位 = 00000000h]
D0Ch	AIRCR	应用中断/复位控制	AIRCR 寄存器（偏移 = D0Ch）[复位 = FA050000h]
D10h	SCR	系统控制	SCR 寄存器（偏移 = D10h）[复位 = 00000000h]
D14h	CCR	配置控制	CCR 寄存器（偏移 = D14h）[复位 = 00000200h]
D18h	SHPR1	系统处理程序 4-7 的优先级	SHPR1 寄存器（偏移 = D18h）[复位 = 00000000h]
D1Ch	SHPR2	系统处理程序 8-11 的优先级	SHPR2 寄存器（偏移 = D1Ch）[复位 = 00000000h]
D20h	SHPR3	系统处理程序 12-15 的优先级	SHPR3 寄存器（偏移 = D20h）[复位 = 00000000h]
D24h	SHCSR	系统处理程序控制和状态	SHCSR 寄存器（偏移 = D24h）[复位 = 00000000h]
D28h	CFSR	可配置故障状态	CFSR 寄存器（偏移 = D28h）[复位 = 00000000h]
D2Ch	HFSR	硬故障状态	HFSR 寄存器（偏移 = D2Ch）[复位 = 00000000h]
D30h	DFSR	调试故障状态	DFSR 寄存器（偏移 = D30h）[复位 = 00000000h]
D34h	MMFAR	存储器管理故障地址	MMFAR 寄存器（偏移 = D34h）[复位 = 00000000h]
D38h	BFAR	总线故障地址	BFAR 寄存器（偏移 = D38h）[复位 = 00000000h]
D3Ch	AFSR	辅助故障状态	AFSR 寄存器（偏移 = D3Ch）[复位 = 00000000h]
D40h	ID_PFR0	处理器功能 0	ID_PFR0 寄存器（偏移 = D40h）[复位 = 00000030h]
D44h	ID_PFR1	处理器功能 1	ID_PFR1 寄存器（偏移 = D44h）[复位 = 00000200h]
D48h	ID_DFR0	调试功能 0	ID_DFR0 寄存器（偏移 = D48h）[复位 = 00100000h]
D4Ch	ID_AFR0	辅助功能 0	ID_AFR0 寄存器（偏移 = D4Ch）[复位 = 00000000h]
D50h	ID_MMFR0	存储器模型功能 0	ID_MMFR0 寄存器（偏移 = D50h）[复位 = 00100030h]
D54h	ID_MMFR1	存储器模型功能 1	ID_MMFR1 寄存器（偏移 = D54h）[复位 = 00000000h]
D58h	ID_MMFR2	存储器模型功能 2	ID_MMFR2 寄存器（偏移 = D58h）[复位 = 01000000h]

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 2-106. CPU_SCS 寄存器（续）

偏移	首字母缩写词	寄存器名称	部分
D5Ch	ID_MMFR3	存储器模型功能 3	ID_MMFR3 寄存器（偏移 = D5Ch）[复位 = 00000000h]
D60h	ID_ISAR0	ISA 功能 0	ID_ISAR0 寄存器（偏移 = D60h）[复位 = 01101110h]
D64h	ID_ISAR1	ISA 功能 1	ID_ISAR1 寄存器（偏移 = D64h）[复位 = 02112000h]
D68h	ID_ISAR2	ISA 功能 2	ID_ISAR2 寄存器（偏移 = D68h）[复位 = 21232231h]
D6Ch	ID_ISAR3	ISA 功能 3	ID_ISAR3 寄存器（偏移 = D6Ch）[复位 = 01111131h]
D70h	ID_ISAR4	ISA 功能 4	ID_ISAR4 寄存器（偏移 = D70h）[复位 = 01310132h]
D88h	CPACR	协处理器访问控制	CPACR 寄存器（偏移 = D88h）[复位 = 00000000h]
D90h	MPU_TYPE	MPU 类型	MPU_TYPE 寄存器（偏移 = D90h）[复位 = 00000800h]
D94h	MPU_CTRL	MPU 控制	MPU_CTRL 寄存器（偏移 = D94h）[复位 = 00000000h]
D98h	MPU_RNR	MPU 区域编号	MPU_RNR 寄存器（偏移 = D98h）[复位 = 00000000h]
D9Ch	MPU_RBAR	MPU 区域基地址	MPU_RBAR 寄存器（偏移 = D9Ch）[复位 = 00000000h]
DA0h	MPU_RASR	MPU 区域属性和大小	MPU_RASR 寄存器（偏移 = DA0h）[复位 = 00000000h]
DA4h	MPU_RBAR_A1	MPU 别名 1 区域基地址	MPU_RBAR_A1 寄存器（偏移 = DA4h）[复位 = 00000000h]
DA8h	MPU_RASR_A1	MPU 别名 1 区域属性和大小	MPU_RASR_A1 寄存器（偏移 = DA8h）[复位 = 00000000h]
DAChh	MPU_RBAR_A2	MPU 别名 2 区域基地址	MPU_RBAR_A2 寄存器（偏移 = DACh）[复位 = 00000000h]
DB0h	MPU_RASR_A2	MPU 别名 2 区域属性和大小	MPU_RASR_A2 寄存器（偏移 = DB0h）[复位 = 00000000h]
DB4h	MPU_RBAR_A3	MPU 别名 3 区域基地址	MPU_RBAR_A3 寄存器（偏移 = DB4h）[复位 = 00000000h]
DB8h	MPU_RASR_A3	MPU 别名 3 区域属性和大小	MPU_RASR_A3 寄存器（偏移 = DB8h）[复位 = 00000000h]
DF0h	DHCSR	调试暂停控制和状态	DHCSR 寄存器（偏移 = DF0h）[复位 = 00000000h]
DF4h	DCRSR	调试内核寄存器选择器	DCRSR 寄存器（偏移 = DF4h）[复位 = 00000000h]
DF8h	DCRDR	调试内核寄存器数据	DCRDR 寄存器（偏移 = DF8h）[复位 = 00000000h]
DFCh	DEMCR	调试异常和监视器控制	DEMCR 寄存器（偏移 = DFCh）[复位 = 00000000h]
F00h	STIR	软件触发中断	STIR 寄存器（偏移 = F00h）[复位 = 00000000h]
F34h	FPCCR	浮点上下文控制	FPCCR 寄存器（偏移 = F34h）[复位 = C0000000h]
F38h	FPCAR	浮点上下文地址	FPCAR 寄存器（偏移 = F38h）[复位 = 00000000h]

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 2-106. CPU_SCS 寄存器（续）

偏移	首字母缩写词	寄存器名称	部分
F3Ch	FPDSCR	浮点默认状态控制	FPDSCR 寄存器（偏移 = F3Ch）[复位 = 00000000h]
F40h	MVFR0	媒体和浮点功能 0	MVFR0 寄存器（偏移 = F40h）[复位 = 10110021h]
F44h	MVFR1	媒体和浮点功能 1	MVFR1 寄存器（偏移 = F44h）[复位 = 11000011h]

复杂的位访问类型经过编码可适应小型表单元。表 2-107 展示了适用于此部分中访问类型的代码。

表 2-107. CPU_SCS 访问类型代码

访问类型	代码	说明
读取类型		
R	R	读取
写入类型		
W	W	写入
W1C	W 1C	写入 1 以清零
复位或默认值		
-n		复位后的值或默认值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.1 ICTR 寄存器（偏移 = 4h）[复位 = 00000001h]

图 2-72 展示了 ICTR，表 2-108 中对此进行了介绍。

返回到汇总表。

中断控制类型
读取该寄存器可查看 NVIC 支持的中断线数量。

图 2-72. ICTR 寄存器

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

表 2-108. ICTR 寄存器字段说明

位	字段	类型	复位	说明
31-3	RESERVED	R	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
2-0	INTLINESNUM	R	1h	以 32 为一组的中断线总数量。 0：0 至 32 1：33 至 64 2：65 至 96 3：97 至 128 4：129 至 160 5：161 至 192 6：193 至 224 7：225...256

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.2 ACTLR 寄存器（偏移 = 8h）[复位 = 00000000h]

图 2-73 展示了 ACTLR，表 2-109 中对此进行了介绍。

返回到汇总表。

辅助控制
该寄存器用于禁用处理器内部的某些功能

图 2-73. ACTLR 寄存器

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						DISOOFP	DISFPCA
R/W-0h						R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED					DISFOLD	DISDEFWBUF	DISMCYCINT
R/W-0h					R/W-0h	R/W-0h	R/W-0h

表 2-109. ACTLR 寄存器字段说明

位	字段	类型	复位	说明
31-10	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
9	DISOOFP	R/W	0h	禁用浮点指令相对于整数指令错序完成。
8	DISFPCA	R/W	0h	禁用 CONTROL.FPCA 的自动更新
7-3	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
2	DISFOLD	R/W	0h	禁用 IT 指令的折叠。
1	DISDEFWBUF	R/W	0h	在默认存储器映射访问期间禁用写缓冲区使用。这会使所有总线故障都成为精确总线故障，但会降低处理器的性能，因为存储器的存储操作必须在执行下一条指令之前完成。
0	DISMCYCINT	R/W	0h	禁用多周期指令的中断。这会增加处理器的中断延迟，因为 LDM/STM 会在中断压栈发生之前完成。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.3 STCSR 寄存器（偏移 = 10h）[复位 = 00000004h]

图 2-74 展示了 STCSR，表 2-110 中对此进行了介绍。

返回到汇总表。

SysTick 控制和状态
该寄存器可启用 SysTick 功能并返回与 SysTick 相关的状态标志。

图 2-74. STCSR 寄存器

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

表 2-110. STCSR 寄存器字段说明

位	字段	类型	复位	说明
31-17	RESERVED	R	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
16	COUNTFLAG	R	0h	如果自上次读取以来计时器已计时到 0，则返回 1。在应用程序读取 SysTick 控制和状态寄存器的任何部分时清除。如果调试器使用 DAP 读取该位，则仅在 **AHB-AP** 控制寄存器中的 MasterType 位设置为 0 时，会在读取时清除该位。否则，调试器读取不会更改 COUNTFLAG。
15-3	RESERVED	R	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
2	CLKSOURCE	R	1h	时钟源： 0：外部基准时钟。 1：内核时钟 该器件不支持外部时钟。对该字段的写入将被忽略。
1	TICKINT	R/W	0h	0：倒数到零不会使 SysTick 处理程序挂起。软件可以使用 COUNTFLAG 来确定 SysTick 处理程序是否已倒数到零。 1：倒数到零会使 SysTick 处理程序挂起。
0	ENABLE	R/W	0h	使能 SysTick 计数器 0：计数器禁用 1：计数器以周期性触发方式运行。也就是说，计数器使用重载值寄存器 STRVR.RELOAD 进行加载，然后开始倒数。达到 0 时，会将 COUNTFLAG 设置为 1，并根据 TICKINT 选择是否挂起 SysTick 处理程序。然后再次加载 STRVR.RELOAD，并开始倒数。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.4 STRVR 寄存器（偏移 = 14h）[复位 = 00000000h]

图 2-75 展示了 STRVR，表 2-111 中对此进行了介绍。

返回到汇总表。

SysTick 重载值
该寄存器用于指定在计数器达到 0 时加载到当前值寄存器 STCVR.CURRENT 中的起始值。可以是 1 到 0x00FFFFFF 之间的任何值。起始值可以为 0，但无效，因为从 1 倒数到 0 会激活 SysTick 中断和 STCSR.COUNTFLAG。

图 2-75. STRVR 寄存器

31	30	29	28	27	26	25	24	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								重新加载																							
R/W-0h								R/W-X																							

表 2-111. STRVR 寄存器字段说明

位	字段	类型	复位	说明
31-24	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
23-0	重新加载	R/W	X	计数器达到 0 时加载到 SysTick 当前值寄存器 STCVR.CURRENT 中的值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.5 STCVR 寄存器（偏移 = 18h）[复位 = 00000000h]

图 2-76 展示了 STCVR，表 2-112 中对此进行了介绍。

返回到汇总表。

SysTick 当前值
读取该寄存器会返回 SysTick 计数器的当前值。写入该寄存器会复位 SysTick 计数器（以及 STCSR.COUNTFLAG）。

图 2-76. STCVR 寄存器

31	30	29	28	27	26	25	24	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								电流																							
R/W-0h								R/W-X																							

表 2-112. STCVR 寄存器字段说明

位	字段	类型	复位	说明
31-24	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
23-0	电流	R/W	X	访问寄存器时的当前值。不提供读-改-写保护，因此请谨慎修改。写入任何值都会将该寄存器清零。清除该寄存器的同时，也会清除 STCSR.COUNTFLAG。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.6 STCR 寄存器（偏移 = 1Ch）[复位 = C0075300h]

图 2-77 展示了 STCR，表 2-113 中对此进行了介绍。

返回到汇总表。

SysTick 校准值
用于使软件能够使用除法和乘法扩展到任何所需的速度。

图 2-77. STCR 寄存器

31	30	29	28	27	26	25	24
NOREF	SKEW	RESERVED					
R-1h	R-1h	R-0h					
23	22	21	20	19	18	17	16
TENMS							
R-00075300h							
15	14	13	12	11	10	9	8
TENMS							
R-00075300h							
7	6	5	4	3	2	1	0
TENMS							
R-00075300h							

表 2-113. STCR 寄存器字段说明

位	字段	类型	复位	说明
31	NOREF	R	1h	读为 1。表示未提供独立的基准时钟。
30	SKEW	R	1h	读为 1。由于时钟频率的原因，校准值并非恰好为 10ms。这可能会影响其作为软件实时时钟的适用性。
29-24	RESERVED	R	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
23-0	TENMS	R	00075300h	用于 10ms (100Hz) 时序的可选重载值，受系统时钟偏斜错误的影响。仅当内核时钟为 48MHz 时，读取的值才有效。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.7 NVIC_ISER0 寄存器（偏移 = 100h）[复位 = 00000000h]

图 2-78 展示了 NVIC_ISER0，表 2-114 对其进行了介绍。

返回到汇总表。

Irq 0 至 31 设置使能
该寄存器用于启用中断并确定当前哪些中断处于启用状态。

图 2-78. NVIC_ISER0 寄存器

31	30	29	28	27	26	25	24
SETENA31	SETENA30	SETENA29	SETENA28	SETENA27	SETENA26	SETENA25	SETENA24
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
SETENA23	SETENA22	SETENA21	SETENA20	SETENA19	SETENA18	SETENA17	SETENA16
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
SETENA15	SETENA14	SETENA13	SETENA12	SETENA11	SETENA10	SETENA9	SETENA8
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
SETENA7	SETENA6	SETENA5	SETENA4	SETENA3	SETENA2	SETENA1	SETENA0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 2-114. NVIC_ISER0 寄存器字段说明

位	字段	类型	复位	说明
31	SETENA31	R/W	0h	向该位写入 0 无效，向该位写入 1 将启用 31 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL31.EV）。读取该位将返回其当前使能值。
30	SETENA30	R/W	0h	向该位写入 0 无效，向该位写入 1 将启用 30 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL30.EV）。读取该位将返回其当前使能值。
29	SETENA29	R/W	0h	向该位写入 0 无效，向该位写入 1 将启用 29 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL29.EV）。读取该位将返回其当前使能值。
28	SETENA28	R/W	0h	向该位写入 0 无效，向该位写入 1 将启用 28 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL28.EV）。读取该位将返回其当前使能值。
27	SETENA27	R/W	0h	向该位写入 0 无效，向该位写入 1 将启用 27 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL27.EV）。读取该位将返回其当前使能值。
26	SETENA26	R/W	0h	向该位写入 0 无效，向该位写入 1 将启用 26 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL26.EV）。读取该位将返回其当前使能值。
25	SETENA25	R/W	0h	向该位写入 0 无效，向该位写入 1 将启用 25 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL25.EV）。读取该位将返回其当前使能值。
24	SETENA24	R/W	0h	向该位写入 0 无效，向该位写入 1 将启用 24 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL24.EV）。读取该位将返回其当前使能值。
23	SETENA23	R/W	0h	向该位写入 0 无效，向该位写入 1 将启用 23 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL23.EV）。读取该位将返回其当前使能值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 2-114. NVIC_ISER0 寄存器字段说明（续）

位	字段	类型	复位	说明
22	SETENA22	R/W	0h	向该位写入 0 无效，向该位写入 1 将启用 22 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL22.EV）。读取该位将返回其当前使能值。
21	SETENA21	R/W	0h	向该位写入 0 无效，向该位写入 1 将启用 21 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL21.EV）。读取该位将返回其当前使能值。
20	SETENA20	R/W	0h	向该位写入 0 无效，向该位写入 1 将启用 20 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL20.EV）。读取该位将返回其当前使能值。
19	SETENA19	R/W	0h	向该位写入 0 无效，向该位写入 1 将启用 19 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL19.EV）。读取该位将返回其当前使能值。
18	SETENA18	R/W	0h	向该位写入 0 无效，向该位写入 1 将启用 18 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL18.EV）。读取该位将返回其当前使能值。
17	SETENA17	R/W	0h	向该位写入 0 无效，向该位写入 1 将启用 17 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL17.EV）。读取该位将返回其当前使能值。
16	SETENA16	R/W	0h	向该位写入 0 无效，向该位写入 1 将启用 16 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL16.EV）。读取该位将返回其当前使能值。
15	SETENA15	R/W	0h	向该位写入 0 无效，向该位写入 1 将启用 15 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL15.EV）。读取该位将返回其当前使能值。
14	SETENA14	R/W	0h	向该位写入 0 无效，向该位写入 1 将启用 14 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL14.EV）。读取该位将返回其当前使能值。
13	SETENA13	R/W	0h	向该位写入 0 无效，向该位写入 1 将启用 13 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL13.EV）。读取该位将返回其当前使能值。
12	SETENA12	R/W	0h	向该位写入 0 无效，向该位写入 1 将启用 12 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL12.EV）。读取该位将返回其当前使能值。
11	SETENA11	R/W	0h	向该位写入 0 无效，向该位写入 1 将启用 11 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL11.EV）。读取该位将返回其当前使能值。
10	SETENA10	R/W	0h	向该位写入 0 无效，向该位写入 1 将启用 10 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL10.EV）。读取该位将返回其当前使能值。
9	SETENA9	R/W	0h	向该位写入 0 无效，向该位写入 1 将启用 9 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL9.EV）。读取该位将返回其当前使能值。
8	SETENA8	R/W	0h	向该位写入 0 无效，向该位写入 1 将启用 8 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL8.EV）。读取该位将返回其当前使能值。
7	SETENA7	R/W	0h	向该位写入 0 无效，向该位写入 1 将启用 7 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL7.EV）。读取该位将返回其当前使能值。
6	SETENA6	R/W	0h	向该位写入 0 无效，向该位写入 1 将启用 6 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL6.EV）。读取该位将返回其当前使能值。
5	SETENA5	R/W	0h	向该位写入 0 无效，向该位写入 1 将启用 5 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL5.EV）。读取该位将返回其当前使能值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 2-114. NVIC_ISER0 寄存器字段说明（续）

位	字段	类型	复位	说明
4	SETENA4	R/W	0h	向该位写入 0 无效，向该位写入 1 将启用 4 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL4.EV）。读取该位将返回其当前使能值。
3	SETENA3	R/W	0h	向该位写入 0 无效，向该位写入 1 将启用 3 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL3.EV）。读取该位将返回其当前使能值。
2	SETENA2	R/W	0h	向该位写入 0 无效，向该位写入 1 将启用 2 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL2.EV）。读取该位将返回其当前使能值。
1	SETENA1	R/W	0h	向该位写入 0 无效，向该位写入 1 将启用 1 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL1.EV）。读取该位将返回其当前使能值。
0	SETENA0	R/W	0h	向该位写入 0 无效，向该位写入 1 将启用 0 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL0.EV）。读取该位将返回其当前使能值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.8 NVIC_ISER1 寄存器（偏移 = 104h）[复位 = 00000000h]

图 2-79 展示了 NVIC_ISER1，表 2-115 对其进行了介绍。

返回到汇总表。

Irq 32 至 63 设置使能
该寄存器用于启用中断并确定当前哪些中断处于启用状态。

图 2-79. NVIC_ISER1 寄存器

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		SETENA37	SETENA36	SETENA35	SETENA34	SETENA33	SETENA32
R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 2-115. NVIC_ISER1 寄存器字段说明

位	字段	类型	复位	说明
31-6	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
5	SETENA37	R/W	0h	向该位写入 0 无效，向该位写入 1 将启用 37 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL37.EV）。读取该位将返回其当前使能值。
4	SETENA36	R/W	0h	向该位写入 0 无效，向该位写入 1 将启用 36 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL36.EV）。读取该位将返回其当前使能值。
3	SETENA35	R/W	0h	向该位写入 0 无效，向该位写入 1 将启用 35 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL35.EV）。读取该位将返回其当前使能值。
2	SETENA34	R/W	0h	向该位写入 0 无效，向该位写入 1 将启用 34 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL34.EV）。读取该位将返回其当前使能值。
1	SETENA33	R/W	0h	向该位写入 0 无效，向该位写入 1 将启用 33 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL33.EV）。读取该位将返回其当前使能值。
0	SETENA32	R/W	0h	向该位写入 0 无效，向该位写入 1 将启用 32 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL32.EV）。读取该位将返回其当前使能值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.9 NVIC_ICER0 寄存器（偏移 = 180h）[复位 = 00000000h]

图 2-80 展示了 NVIC_ICER0，表 2-116 对其进行了介绍。

返回到汇总表。

Irq 0 至 31 清除使能
该寄存器用于禁用中断并确定当前哪些中断处于启用状态。

图 2-80. NVIC_ICER0 寄存器

31	30	29	28	27	26	25	24
CLRENA31	CLRENA30	CLRENA29	CLRENA28	CLRENA27	CLRENA26	CLRENA25	CLRENA24
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
CLRENA23	CLRENA22	CLRENA21	CLRENA20	CLRENA19	CLRENA18	CLRENA17	CLRENA16
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
CLRENA15	CLRENA14	CLRENA13	CLRENA12	CLRENA11	CLRENA10	CLRENA9	CLRENA8
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
CLRENA7	CLRENA6	CLRENA5	CLRENA4	CLRENA3	CLRENA2	CLRENA1	CLRENA0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 2-116. NVIC_ICER0 寄存器字段说明

位	字段	类型	复位	说明
31	CLRENA31	R/W	0h	向该位写入 0 无效，向该位写入 1 将禁用 31 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL31.EV）。读取该位将返回其当前使能值。
30	CLRENA30	R/W	0h	向该位写入 0 无效，向该位写入 1 将禁用 30 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL30.EV）。读取该位将返回其当前使能值。
29	CLRENA29	R/W	0h	向该位写入 0 无效，向该位写入 1 将禁用 29 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL29.EV）。读取该位将返回其当前使能值。
28	CLRENA28	R/W	0h	向该位写入 0 无效，向该位写入 1 将禁用 28 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL28.EV）。读取该位将返回其当前使能值。
27	CLRENA27	R/W	0h	向该位写入 0 无效，向该位写入 1 将禁用 27 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL27.EV）。读取该位将返回其当前使能值。
26	CLRENA26	R/W	0h	向该位写入 0 无效，向该位写入 1 将禁用 26 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL26.EV）。读取该位将返回其当前使能值。
25	CLRENA25	R/W	0h	向该位写入 0 无效，向该位写入 1 将禁用 25 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL25.EV）。读取该位将返回其当前使能值。
24	CLRENA24	R/W	0h	向该位写入 0 无效，向该位写入 1 将禁用 24 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL24.EV）。读取该位将返回其当前使能值。
23	CLRENA23	R/W	0h	向该位写入 0 无效，向该位写入 1 将禁用 23 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL23.EV）。读取该位将返回其当前使能值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 2-116. NVIC_ICER0 寄存器字段说明（续）

位	字段	类型	复位	说明
22	CLRENA22	R/W	0h	向该位写入 0 无效，向该位写入 1 将禁用 22 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL22.EV）。读取该位将返回其当前使能值。
21	CLRENA21	R/W	0h	向该位写入 0 无效，向该位写入 1 将禁用 21 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL21.EV）。读取该位将返回其当前使能值。
20	CLRENA20	R/W	0h	向该位写入 0 无效，向该位写入 1 将禁用 20 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL20.EV）。读取该位将返回其当前使能值。
19	CLRENA19	R/W	0h	向该位写入 0 无效，向该位写入 1 将禁用 19 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL19.EV）。读取该位将返回其当前使能值。
18	CLRENA18	R/W	0h	向该位写入 0 无效，向该位写入 1 将禁用 18 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL18.EV）。读取该位将返回其当前使能值。
17	CLRENA17	R/W	0h	向该位写入 0 无效，向该位写入 1 将禁用 17 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL17.EV）。读取该位将返回其当前使能值。
16	CLRENA16	R/W	0h	向该位写入 0 无效，向该位写入 1 将禁用 16 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL16.EV）。读取该位将返回其当前使能值。
15	CLRENA15	R/W	0h	向该位写入 0 无效，向该位写入 1 将禁用 15 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL15.EV）。读取该位将返回其当前使能值。
14	CLRENA14	R/W	0h	向该位写入 0 无效，向该位写入 1 将禁用 14 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL14.EV）。读取该位将返回其当前使能值。
13	CLRENA13	R/W	0h	向该位写入 0 无效，向该位写入 1 将禁用 13 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL13.EV）。读取该位将返回其当前使能值。
12	CLRENA12	R/W	0h	向该位写入 0 无效，向该位写入 1 将禁用 12 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL12.EV）。读取该位将返回其当前使能值。
11	CLRENA11	R/W	0h	向该位写入 0 无效，向该位写入 1 将禁用 11 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL11.EV）。读取该位将返回其当前使能值。
10	CLRENA10	R/W	0h	向该位写入 0 无效，向该位写入 1 将禁用 10 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL10.EV）。读取该位将返回其当前使能值。
9	CLRENA9	R/W	0h	向该位写入 0 无效，向该位写入 1 将禁用 9 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL9.EV）。读取该位将返回其当前使能值。
8	CLRENA8	R/W	0h	向该位写入 0 无效，向该位写入 1 将禁用 8 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL8.EV）。读取该位将返回其当前使能值。
7	CLRENA7	R/W	0h	向该位写入 0 无效，向该位写入 1 将禁用 7 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL7.EV）。读取该位将返回其当前使能值。
6	CLRENA6	R/W	0h	向该位写入 0 无效，向该位写入 1 将禁用 6 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL6.EV）。读取该位将返回其当前使能值。
5	CLRENA5	R/W	0h	向该位写入 0 无效，向该位写入 1 将禁用 5 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL5.EV）。读取该位将返回其当前使能值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 2-116. NVIC_ICER0 寄存器字段说明（续）

位	字段	类型	复位	说明
4	CLRENA4	R/W	0h	向该位写入 0 无效，向该位写入 1 将禁用 4 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL4.EV）。读取该位将返回其当前使能值。
3	CLRENA3	R/W	0h	向该位写入 0 无效，向该位写入 1 将禁用 3 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL3.EV）。读取该位将返回其当前使能值。
2	CLRENA2	R/W	0h	向该位写入 0 无效，向该位写入 1 将禁用 2 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL2.EV）。读取该位将返回其当前使能值。
1	CLRENA1	R/W	0h	向该位写入 0 无效，向该位写入 1 将禁用 1 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL1.EV）。读取该位将返回其当前使能值。
0	CLRENA0	R/W	0h	向该位写入 0 无效，向该位写入 1 将禁用 0 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL0.EV）。读取该位将返回其当前使能值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.10 NVIC_ICER1 寄存器（偏移 = 184h）[复位 = 00000000h]

图 2-81 展示了 NVIC_ICER1，表 2-117 对其进行了介绍。

返回到汇总表。

Irq 32 至 63 清除使能
该寄存器用于禁用中断并确定当前哪些中断处于启用状态。

图 2-81. NVIC_ICER1 寄存器

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		CLRENA37	CLRENA36	CLRENA35	CLRENA34	CLRENA33	CLRENA32
R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 2-117. NVIC_ICER1 寄存器字段说明

位	字段	类型	复位	说明
31-6	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
5	CLRENA37	R/W	0h	向该位写入 0 无效，向该位写入 1 将禁用 37 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL37.EV）。读取该位将返回其当前使能值。
4	CLRENA36	R/W	0h	向该位写入 0 无效，向该位写入 1 将禁用 36 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL36.EV）。读取该位将返回其当前使能值。
3	CLRENA35	R/W	0h	向该位写入 0 无效，向该位写入 1 将禁用 35 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL35.EV）。读取该位将返回其当前使能值。
2	CLRENA34	R/W	0h	向该位写入 0 无效，向该位写入 1 将禁用 34 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL34.EV）。读取该位将返回其当前使能值。
1	CLRENA33	R/W	0h	向该位写入 0 无效，向该位写入 1 将禁用 33 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL33.EV）。读取该位将返回其当前使能值。
0	CLRENA32	R/W	0h	向该位写入 0 无效，向该位写入 1 将禁用 32 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL32.EV）。读取该位将返回其当前使能值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.11 NVIC_ISPR0 寄存器（偏移 = 200h）[复位 = 00000000h]

图 2-82 展示了 NVIC_ISPR0，表 2-118 对其进行了介绍。

返回到汇总表。

Irq 0 至 31 设置挂起
该寄存器用于强制中断进入挂起状态并确定当前哪些中断处于挂起状态。

图 2-82. NVIC_ISPR0 寄存器

31	30	29	28	27	26	25	24
SETPEND31	SETPEND30	SETPEND29	SETPEND28	SETPEND27	SETPEND26	SETPEND25	SETPEND24
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
SETPEND23	SETPEND22	SETPEND21	SETPEND20	SETPEND19	SETPEND18	SETPEND17	SETPEND16
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
SETPEND15	SETPEND14	SETPEND13	SETPEND12	SETPEND11	SETPEND10	SETPEND9	SETPEND8
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
SETPEND7	SETPEND6	SETPEND5	SETPEND4	SETPEND3	SETPEND2	SETPEND1	SETPEND0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 2-118. NVIC_ISPR0 寄存器字段说明

位	字段	类型	复位	说明
31	SETPEND31	R/W	0h	向该位写入 0 无效，向该位写入 1 将挂起 31 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL31.EV）。读取该位将返回其当前状态。
30	SETPEND30	R/W	0h	向该位写入 0 无效，向该位写入 1 将挂起 30 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL30.EV）。读取该位将返回其当前状态。
29	SETPEND29	R/W	0h	向该位写入 0 无效，向该位写入 1 将挂起 29 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL29.EV）。读取该位将返回其当前状态。
28	SETPEND28	R/W	0h	向该位写入 0 无效，向该位写入 1 将挂起 28 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL28.EV）。读取该位将返回其当前状态。
27	SETPEND27	R/W	0h	向该位写入 0 无效，向该位写入 1 将挂起 27 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL27.EV）。读取该位将返回其当前状态。
26	SETPEND26	R/W	0h	向该位写入 0 无效，向该位写入 1 将挂起 26 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL26.EV）。读取该位将返回其当前状态。
25	SETPEND25	R/W	0h	向该位写入 0 无效，向该位写入 1 将挂起 25 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL25.EV）。读取该位将返回其当前状态。
24	SETPEND24	R/W	0h	向该位写入 0 无效，向该位写入 1 将挂起 24 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL24.EV）。读取该位将返回其当前状态。
23	SETPEND23	R/W	0h	向该位写入 0 无效，向该位写入 1 将挂起 23 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL23.EV）。读取该位将返回其当前状态。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 2-118. NVIC_ISPR0 寄存器字段说明（续）

位	字段	类型	复位	说明
22	SETPEND22	R/W	0h	向该位写入 0 无效，向该位写入 1 将挂起 22 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL22.EV）。读取该位将返回其当前状态。
21	SETPEND21	R/W	0h	向该位写入 0 无效，向该位写入 1 将挂起 21 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL21.EV）。读取该位将返回其当前状态。
20	SETPEND20	R/W	0h	向该位写入 0 无效，向该位写入 1 将挂起 20 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL20.EV）。读取该位将返回其当前状态。
19	SETPEND19	R/W	0h	向该位写入 0 无效，向该位写入 1 将挂起 19 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL19.EV）。读取该位将返回其当前状态。
18	SETPEND18	R/W	0h	向该位写入 0 无效，向该位写入 1 将挂起 18 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL18.EV）。读取该位将返回其当前状态。
17	SETPEND17	R/W	0h	向该位写入 0 无效，向该位写入 1 将挂起 17 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL17.EV）。读取该位将返回其当前状态。
16	SETPEND16	R/W	0h	向该位写入 0 无效，向该位写入 1 将挂起 16 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL16.EV）。读取该位将返回其当前状态。
15	SETPEND15	R/W	0h	向该位写入 0 无效，向该位写入 1 将挂起 15 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL15.EV）。读取该位将返回其当前状态。
14	SETPEND14	R/W	0h	向该位写入 0 无效，向该位写入 1 将挂起 14 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL14.EV）。读取该位将返回其当前状态。
13	SETPEND13	R/W	0h	向该位写入 0 无效，向该位写入 1 将挂起 13 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL13.EV）。读取该位将返回其当前状态。
12	SETPEND12	R/W	0h	向该位写入 0 无效，向该位写入 1 将挂起 12 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL12.EV）。读取该位将返回其当前状态。
11	SETPEND11	R/W	0h	向该位写入 0 无效，向该位写入 1 将挂起 11 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL11.EV）。读取该位将返回其当前状态。
10	SETPEND10	R/W	0h	向该位写入 0 无效，向该位写入 1 将挂起 10 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL10.EV）。读取该位将返回其当前状态。
9	SETPEND9	R/W	0h	向该位写入 0 无效，向该位写入 1 将挂起 9 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL9.EV）。读取该位将返回其当前状态。
8	SETPEND8	R/W	0h	向该位写入 0 无效，向该位写入 1 将挂起 8 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL8.EV）。读取该位将返回其当前状态。
7	SETPEND7	R/W	0h	向该位写入 0 无效，向该位写入 1 将挂起 7 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL7.EV）。读取该位将返回其当前状态。
6	SETPEND6	R/W	0h	向该位写入 0 无效，向该位写入 1 将挂起 6 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL6.EV）。读取该位将返回其当前状态。
5	SETPEND5	R/W	0h	向该位写入 0 无效，向该位写入 1 将挂起 5 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL5.EV）。读取该位将返回其当前状态。
4	SETPEND4	R/W	0h	向该位写入 0 无效，向该位写入 1 将挂起 4 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL4.EV）。读取该位将返回其当前状态。
3	SETPEND3	R/W	0h	向该位写入 0 无效，向该位写入 1 将挂起 3 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL3.EV）。读取该位将返回其当前状态。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 2-118. NVIC_ISPR0 寄存器字段说明（续）

位	字段	类型	复位	说明
2	SETPEND2	R/W	0h	向该位写入 0 无效，向该位写入 1 将挂起 2 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL2.EV）。读取该位将返回其当前状态。
1	SETPEND1	R/W	0h	向该位写入 0 无效，向该位写入 1 将挂起 1 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL1.EV）。读取该位将返回其当前状态。
0	SETPEND0	R/W	0h	向该位写入 0 无效，向该位写入 1 将挂起 0 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL0.EV）。读取该位将返回其当前状态。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.12 NVIC_ISPR1 寄存器（偏移 = 204h）[复位 = 00000000h]

图 2-83 展示了 NVIC_ISPR1，表 2-119 对其进行了介绍。

返回到汇总表。

Irq 32 至 63 设置挂起
该寄存器用于强制中断进入挂起状态并确定当前哪些中断处于挂起状态。

图 2-83. NVIC_ISPR1 寄存器

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		SETPEND37	SETPEND36	SETPEND35	SETPEND34	SETPEND33	SETPEND32
R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 2-119. NVIC_ISPR1 寄存器字段说明

位	字段	类型	复位	说明
31-6	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
5	SETPEND37	R/W	0h	向该位写入 0 无效，向该位写入 1 将挂起 37 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL37.EV）。读取该位将返回其当前状态。
4	SETPEND36	R/W	0h	向该位写入 0 无效，向该位写入 1 将挂起 36 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL36.EV）。读取该位将返回其当前状态。
3	SETPEND35	R/W	0h	向该位写入 0 无效，向该位写入 1 将挂起 35 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL35.EV）。读取该位将返回其当前状态。
2	SETPEND34	R/W	0h	向该位写入 0 无效，向该位写入 1 将挂起 34 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL34.EV）。读取该位将返回其当前状态。
1	SETPEND33	R/W	0h	向该位写入 0 无效，向该位写入 1 将挂起 33 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL33.EV）。读取该位将返回其当前状态。
0	SETPEND32	R/W	0h	向该位写入 0 无效，向该位写入 1 将挂起 32 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL32.EV）。读取该位将返回其当前状态。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.13 NVIC_ICPR0 寄存器（偏移 = 280h）[复位 = 00000000h]

图 2-84 展示了 NVIC_ICPR0，表 2-120 对其进行了介绍。

返回到汇总表。

Irq 0 至 31 清除挂起
该寄存器用于清除挂起的中断并确定当前哪些中断处于挂起状态。

图 2-84. NVIC_ICPR0 寄存器

31	30	29	28	27	26	25	24
CLRPEND31	CLRPEND30	CLRPEND29	CLRPEND28	CLRPEND27	CLRPEND26	CLRPEND25	CLRPEND24
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
CLRPEND23	CLRPEND22	CLRPEND21	CLRPEND20	CLRPEND19	CLRPEND18	CLRPEND17	CLRPEND16
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
CLRPEND15	CLRPEND14	CLRPEND13	CLRPEND12	CLRPEND11	CLRPEND10	CLRPEND9	CLRPEND8
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
CLRPEND7	CLRPEND6	CLRPEND5	CLRPEND4	CLRPEND3	CLRPEND2	CLRPEND1	CLRPEND0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 2-120. NVIC_ICPR0 寄存器字段说明

位	字段	类型	复位	说明
31	CLRPEND31	R/W	0h	向该位写入 0 无效，向该位写入 1 将清除相应挂起的 31 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL31.EV）。读取该位将返回其当前状态。
30	CLRPEND30	R/W	0h	向该位写入 0 无效，向该位写入 1 将清除相应挂起的 30 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL30.EV）。读取该位将返回其当前状态。
29	CLRPEND29	R/W	0h	向该位写入 0 无效，向该位写入 1 将清除相应挂起的 29 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL29.EV）。读取该位将返回其当前状态。
28	CLRPEND28	R/W	0h	向该位写入 0 无效，向该位写入 1 将清除相应挂起的 28 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL28.EV）。读取该位将返回其当前状态。
27	CLRPEND27	R/W	0h	向该位写入 0 无效，向该位写入 1 将清除相应挂起的 27 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL27.EV）。读取该位将返回其当前状态。
26	CLRPEND26	R/W	0h	向该位写入 0 无效，向该位写入 1 将清除相应挂起的 26 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL26.EV）。读取该位将返回其当前状态。
25	CLRPEND25	R/W	0h	向该位写入 0 无效，向该位写入 1 将清除相应挂起的 25 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL25.EV）。读取该位将返回其当前状态。
24	CLRPEND24	R/W	0h	向该位写入 0 无效，向该位写入 1 将清除相应挂起的 24 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL24.EV）。读取该位将返回其当前状态。
23	CLRPEND23	R/W	0h	向该位写入 0 无效，向该位写入 1 将清除相应挂起的 23 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL23.EV）。读取该位将返回其当前状态。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 2-120. NVIC_ICPR0 寄存器字段说明（续）

位	字段	类型	复位	说明
22	CLRPEND22	R/W	0h	向该位写入 0 无效，向该位写入 1 将清除相应挂起的 22 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL22.EV）。读取该位将返回其当前状态。
21	CLRPEND21	R/W	0h	向该位写入 0 无效，向该位写入 1 将清除相应挂起的 21 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL21.EV）。读取该位将返回其当前状态。
20	CLRPEND20	R/W	0h	向该位写入 0 无效，向该位写入 1 将清除相应挂起的 20 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL20.EV）。读取该位将返回其当前状态。
19	CLRPEND19	R/W	0h	向该位写入 0 无效，向该位写入 1 将清除相应挂起的 19 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL19.EV）。读取该位将返回其当前状态。
18	CLRPEND18	R/W	0h	向该位写入 0 无效，向该位写入 1 将清除相应挂起的 18 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL18.EV）。读取该位将返回其当前状态。
17	CLRPEND17	R/W	0h	向该位写入 0 无效，向该位写入 1 将清除相应挂起的 17 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL17.EV）。读取该位将返回其当前状态。
16	CLRPEND16	R/W	0h	向该位写入 0 无效，向该位写入 1 将清除相应挂起的 16 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL16.EV）。读取该位将返回其当前状态。
15	CLRPEND15	R/W	0h	向该位写入 0 无效，向该位写入 1 将清除相应挂起的 15 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL15.EV）。读取该位将返回其当前状态。
14	CLRPEND14	R/W	0h	向该位写入 0 无效，向该位写入 1 将清除相应挂起的 14 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL14.EV）。读取该位将返回其当前状态。
13	CLRPEND13	R/W	0h	向该位写入 0 无效，向该位写入 1 将清除相应挂起的 13 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL13.EV）。读取该位将返回其当前状态。
12	CLRPEND12	R/W	0h	向该位写入 0 无效，向该位写入 1 将清除相应挂起的 12 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL12.EV）。读取该位将返回其当前状态。
11	CLRPEND11	R/W	0h	向该位写入 0 无效，向该位写入 1 将清除相应挂起的 11 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL11.EV）。读取该位将返回其当前状态。
10	CLRPEND10	R/W	0h	向该位写入 0 无效，向该位写入 1 将清除相应挂起的 10 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL10.EV）。读取该位将返回其当前状态。
9	CLRPEND9	R/W	0h	向该位写入 0 无效，向该位写入 1 将清除相应挂起的 9 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL9.EV）。读取该位将返回其当前状态。
8	CLRPEND8	R/W	0h	向该位写入 0 无效，向该位写入 1 将清除相应挂起的 8 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL8.EV）。读取该位将返回其当前状态。
7	CLRPEND7	R/W	0h	向该位写入 0 无效，向该位写入 1 将清除相应挂起的 7 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL7.EV）。读取该位将返回其当前状态。
6	CLRPEND6	R/W	0h	向该位写入 0 无效，向该位写入 1 将清除相应挂起的 6 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL6.EV）。读取该位将返回其当前状态。
5	CLRPEND5	R/W	0h	向该位写入 0 无效，向该位写入 1 将清除相应挂起的 5 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL5.EV）。读取该位将返回其当前状态。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 2-120. NVIC_ICPR0 寄存器字段说明（续）

位	字段	类型	复位	说明
4	CLRPEND4	R/W	0h	向该位写入 0 无效，向该位写入 1 将清除相应挂起的 4 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL4.EV）。读取该位将返回其当前状态。
3	CLRPEND3	R/W	0h	向该位写入 0 无效，向该位写入 1 将清除相应挂起的 3 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL3.EV）。读取该位将返回其当前状态。
2	CLRPEND2	R/W	0h	向该位写入 0 无效，向该位写入 1 将清除相应挂起的 2 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL2.EV）。读取该位将返回其当前状态。
1	CLRPEND1	R/W	0h	向该位写入 0 无效，向该位写入 1 将清除相应挂起的 1 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL1.EV）。读取该位将返回其当前状态。
0	CLRPEND0	R/W	0h	向该位写入 0 无效，向该位写入 1 将清除相应挂起的 0 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL0.EV）。读取该位将返回其当前状态。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.14 NVIC_ICPR1 寄存器（偏移 = 284h）[复位 = 00000000h]

图 2-85 展示了 NVIC_ICPR1，表 2-121 对其进行了介绍。

返回到汇总表。

Irq 32 至 63 清除挂起
该寄存器用于清除挂起的中断并确定当前哪些中断处于挂起状态。

图 2-85. NVIC_ICPR1 寄存器

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		CLRPEND37	CLRPEND36	CLRPEND35	CLRPEND34	CLRPEND33	CLRPEND32
R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 2-121. NVIC_ICPR1 寄存器字段说明

位	字段	类型	复位	说明
31-6	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
5	CLRPEND37	R/W	0h	向该位写入 0 无效，向该位写入 1 将清除相应挂起的 37 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL37.EV）。读取该位将返回其当前状态。
4	CLRPEND36	R/W	0h	向该位写入 0 无效，向该位写入 1 将清除相应挂起的 36 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL36.EV）。读取该位将返回其当前状态。
3	CLRPEND35	R/W	0h	向该位写入 0 无效，向该位写入 1 将清除相应挂起的 35 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL35.EV）。读取该位将返回其当前状态。
2	CLRPEND34	R/W	0h	向该位写入 0 无效，向该位写入 1 将清除相应挂起的 34 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL34.EV）。读取该位将返回其当前状态。
1	CLRPEND33	R/W	0h	向该位写入 0 无效，向该位写入 1 将清除相应挂起的 33 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL33.EV）。读取该位将返回其当前状态。
0	CLRPEND32	R/W	0h	向该位写入 0 无效，向该位写入 1 将清除相应挂起的 32 号中断（有关详细信息，请参阅 EVENT:CPUIRQSEL32.EV）。读取该位将返回其当前状态。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.15 NVIC_IABR0 寄存器（偏移 = 300h）[复位 = 00000000h]

图 2-86 展示了 NVIC_IABR0，表 2-122 对其进行了介绍。

返回到汇总表。

Irq 0 至 31 活动位
该寄存器用于确定哪些中断处于活动状态。该寄存器中的每个标志对应一个中断。

图 2-86. NVIC_IABR0 寄存器

31	30	29	28	27	26	25	24
ACTIVE31	ACTIVE30	ACTIVE29	ACTIVE28	ACTIVE27	ACTIVE26	ACTIVE25	ACTIVE24
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
ACTIVE23	ACTIVE22	ACTIVE21	ACTIVE20	ACTIVE19	ACTIVE18	ACTIVE17	ACTIVE16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
ACTIVE15	ACTIVE14	ACTIVE13	ACTIVE12	ACTIVE11	ACTIVE10	ACTIVE9	ACTIVE8
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
ACTIVE7	ACTIVE6	ACTIVE5	ACTIVE4	ACTIVE3	ACTIVE2	ACTIVE1	ACTIVE0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

表 2-122. NVIC_IABR0 寄存器字段说明

位	字段	类型	复位	说明
31	ACTIVE31	R	0h	该位读值为 0 意味着中断线 31 处于非活动状态。该位读值为 1 意味着中断线 31 处于活动状态（有关详细信息，请参阅 EVENT:CPUIRQSEL31.EV）。
30	ACTIVE30	R	0h	该位读值为 0 意味着中断线 30 处于非活动状态。该位读值为 1 意味着中断线 30 处于活动状态（有关详细信息，请参阅 EVENT:CPUIRQSEL30.EV）。
29	ACTIVE29	R	0h	该位读值为 0 意味着中断线 29 处于非活动状态。该位读值为 1 意味着中断线 29 处于活动状态（有关详细信息，请参阅 EVENT:CPUIRQSEL29.EV）。
28	ACTIVE28	R	0h	该位读值为 0 意味着中断线 28 处于非活动状态。该位读值为 1 意味着中断线 28 处于活动状态（有关详细信息，请参阅 EVENT:CPUIRQSEL28.EV）。
27	ACTIVE27	R	0h	该位读值为 0 意味着中断线 27 处于非活动状态。该位读值为 1 意味着中断线 27 处于活动状态（有关详细信息，请参阅 EVENT:CPUIRQSEL27.EV）。
26	ACTIVE26	R	0h	该位读值为 0 意味着中断线 26 处于非活动状态。该位读值为 1 意味着中断线 26 处于活动状态（有关详细信息，请参阅 EVENT:CPUIRQSEL26.EV）。
25	ACTIVE25	R	0h	该位读值为 0 意味着中断线 25 处于非活动状态。该位读值为 1 意味着中断线 25 处于活动状态（有关详细信息，请参阅 EVENT:CPUIRQSEL25.EV）。
24	ACTIVE24	R	0h	该位读值为 0 意味着中断线 24 处于非活动状态。该位读值为 1 意味着中断线 24 处于活动状态（有关详细信息，请参阅 EVENT:CPUIRQSEL24.EV）。
23	ACTIVE23	R	0h	该位读值为 0 意味着中断线 23 处于非活动状态。该位读值为 1 意味着中断线 23 处于活动状态（有关详细信息，请参阅 EVENT:CPUIRQSEL23.EV）。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 2-122. NVIC_IABR0 寄存器字段说明（续）

位	字段	类型	复位	说明
22	ACTIVE22	R	0h	该位读值为 0 意味着中断线 22 处于非活动状态。该位读值为 1 意味着中断线 22 处于活动状态（有关详细信息，请参阅 EVENT:CPUIRQSEL22.EV）。
21	ACTIVE21	R	0h	该位读值为 0 意味着中断线 21 处于非活动状态。该位读值为 1 意味着中断线 21 处于活动状态（有关详细信息，请参阅 EVENT:CPUIRQSEL21.EV）。
20	ACTIVE20	R	0h	该位读值为 0 意味着中断线 20 处于非活动状态。该位读值为 1 意味着中断线 20 处于活动状态（有关详细信息，请参阅 EVENT:CPUIRQSEL20.EV）。
19	ACTIVE19	R	0h	该位读值为 0 意味着中断线 19 处于非活动状态。该位读值为 1 意味着中断线 19 处于活动状态（有关详细信息，请参阅 EVENT:CPUIRQSEL19.EV）。
18	ACTIVE18	R	0h	该位读值为 0 意味着中断线 18 处于非活动状态。该位读值为 1 意味着中断线 18 处于活动状态（有关详细信息，请参阅 EVENT:CPUIRQSEL18.EV）。
17	ACTIVE17	R	0h	该位读值为 0 意味着中断线 17 处于非活动状态。该位读值为 1 意味着中断线 17 处于活动状态（有关详细信息，请参阅 EVENT:CPUIRQSEL17.EV）。
16	ACTIVE16	R	0h	该位读值为 0 意味着中断线 16 处于非活动状态。该位读值为 1 意味着中断线 16 处于活动状态（有关详细信息，请参阅 EVENT:CPUIRQSEL16.EV）。
15	ACTIVE15	R	0h	该位读值为 0 意味着中断线 15 处于非活动状态。该位读值为 1 意味着中断线 15 处于活动状态（有关详细信息，请参阅 EVENT:CPUIRQSEL15.EV）。
14	ACTIVE14	R	0h	该位读值为 0 意味着中断线 14 处于非活动状态。该位读值为 1 意味着中断线 14 处于活动状态（有关详细信息，请参阅 EVENT:CPUIRQSEL14.EV）。
13	ACTIVE13	R	0h	该位读值为 0 意味着中断线 13 处于非活动状态。该位读值为 1 意味着中断线 13 处于活动状态（有关详细信息，请参阅 EVENT:CPUIRQSEL13.EV）。
12	ACTIVE12	R	0h	该位读值为 0 意味着中断线 12 处于非活动状态。该位读值为 1 意味着中断线 12 处于活动状态（有关详细信息，请参阅 EVENT:CPUIRQSEL12.EV）。
11	ACTIVE11	R	0h	该位读值为 0 意味着中断线 11 处于非活动状态。该位读值为 1 意味着中断线 11 处于活动状态（有关详细信息，请参阅 EVENT:CPUIRQSEL11.EV）。
10	ACTIVE10	R	0h	该位读值为 0 意味着中断线 10 处于非活动状态。该位读值为 1 意味着中断线 10 处于活动状态（有关详细信息，请参阅 EVENT:CPUIRQSEL10.EV）。
9	ACTIVE9	R	0h	该位读值为 0 意味着中断线 9 处于非活动状态。该位读值为 1 意味着中断线 9 处于活动状态（有关详细信息，请参阅 EVENT:CPUIRQSEL9.EV）。
8	ACTIVE8	R	0h	该位读值为 0 意味着中断线 8 处于非活动状态。该位读值为 1 意味着中断线 8 处于活动状态（有关详细信息，请参阅 EVENT:CPUIRQSEL8.EV）。
7	ACTIVE7	R	0h	该位读值为 0 意味着中断线 7 处于非活动状态。该位读值为 1 意味着中断线 7 处于活动状态（有关详细信息，请参阅 EVENT:CPUIRQSEL7.EV）。
6	ACTIVE6	R	0h	该位读值为 0 意味着中断线 6 处于非活动状态。该位读值为 1 意味着中断线 6 处于活动状态（有关详细信息，请参阅 EVENT:CPUIRQSEL6.EV）。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 2-122. NVIC_IABR0 寄存器字段说明（续）

位	字段	类型	复位	说明
5	ACTIVE5	R	0h	该位读值为 0 意味着中断线 5 处于非活动状态。该位读值为 1 意味着中断线 5 处于活动状态（有关详细信息，请参阅 EVENT:CPUIRQSEL5.EV）。
4	ACTIVE4	R	0h	该位读值为 0 意味着中断线 4 处于非活动状态。该位读值为 1 意味着中断线 4 处于活动状态（有关详细信息，请参阅 EVENT:CPUIRQSEL4.EV）。
3	ACTIVE3	R	0h	该位读值为 0 意味着中断线 3 处于非活动状态。该位读值为 1 意味着中断线 3 处于活动状态（有关详细信息，请参阅 EVENT:CPUIRQSEL3.EV）。
2	ACTIVE2	R	0h	该位读值为 0 意味着中断线 2 处于非活动状态。该位读值为 1 意味着中断线 2 处于活动状态（有关详细信息，请参阅 EVENT:CPUIRQSEL2.EV）。
1	ACTIVE1	R	0h	该位读值为 0 意味着中断线 1 处于非活动状态。该位读值为 1 意味着中断线 1 处于活动状态（有关详细信息，请参阅 EVENT:CPUIRQSEL1.EV）。
0	ACTIVE0	R	0h	该位读值为 0 意味着中断线 0 处于非活动状态。该位读值为 1 意味着中断线 0 处于活动状态（有关详细信息，请参阅 EVENT:CPUIRQSEL0.EV）。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.16 NVIC_IABR1 寄存器（偏移 = 304h）[复位 = 00000000h]

图 2-87 展示了 NVIC_IABR1，表 2-123 对其进行了介绍。

返回到汇总表。

Irq 32 至 63 活动位
该寄存器用于确定哪些中断处于活动状态。该寄存器中的每个标志对应一个中断。

图 2-87. NVIC_IABR1 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED		ACTIVE37	ACTIVE36	ACTIVE35	ACTIVE34	ACTIVE33	ACTIVE32
R-0h		R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

表 2-123. NVIC_IABR1 寄存器字段说明

位	字段	类型	复位	说明
31-6	RESERVED	R	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
5	ACTIVE37	R	0h	该位读值为 0 意味着中断线 37 处于非活动状态。该位读值为 1 意味着中断线 37 处于活动状态（有关详细信息，请参阅 EVENT:CPUIRQSEL37.EV）。
4	ACTIVE36	R	0h	该位读值为 0 意味着中断线 36 处于非活动状态。该位读值为 1 意味着中断线 36 处于活动状态（有关详细信息，请参阅 EVENT:CPUIRQSEL36.EV）。
3	ACTIVE35	R	0h	该位读值为 0 意味着中断线 35 处于非活动状态。该位读值为 1 意味着中断线 35 处于活动状态（有关详细信息，请参阅 EVENT:CPUIRQSEL35.EV）。
2	ACTIVE34	R	0h	该位读值为 0 意味着中断线 34 处于非活动状态。该位读值为 1 意味着中断线 34 处于活动状态（有关详细信息，请参阅 EVENT:CPUIRQSEL34.EV）。
1	ACTIVE33	R	0h	该位读值为 0 意味着中断线 33 处于非活动状态。该位读值为 1 意味着中断线 33 处于活动状态（有关详细信息，请参阅 EVENT:CPUIRQSEL33.EV）。
0	ACTIVE32	R	0h	该位读值为 0 意味着中断线 32 处于非活动状态。该位读值为 1 意味着中断线 32 处于活动状态（有关详细信息，请参阅 EVENT:CPUIRQSEL32.EV）。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.17 NVIC_IPR0 寄存器（偏移 = 400h）[复位 = 00000000h]

图 2-88 展示了 NVIC_IPR0，表 2-124 对其进行了介绍。

返回到汇总表。

Irq 0 至 3 的优先级
该寄存器用于为每个可用中断分配优先级，范围为 0 至 255。0 是最高优先级，255 是最低优先级。中断优先级寄存器的解释根据 AIRCR.PRIGROUP 中的设置而变化。

图 2-88. NVIC_IPR0 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRI_3								PRI_2								PRI_1								PRI_0							
R/W-0h								R/W-0h								R/W-0h								R/W-0h							

表 2-124. NVIC_IPR0 寄存器字段说明

位	字段	类型	复位	说明
31-24	PRI_3	R/W	0h	中断 3 的优先级（有关详细信息，请参阅 EVENT:CPUIRQSEL3.EV）。
23-16	PRI_2	R/W	0h	中断 2 的优先级（有关详细信息，请参阅 EVENT:CPUIRQSEL2.EV）。
15-8	PRI_1	R/W	0h	中断 1 的优先级（有关详细信息，请参阅 EVENT:CPUIRQSEL1.EV）。
7-0	PRI_0	R/W	0h	中断 0 的优先级（有关详细信息，请参阅 EVENT:CPUIRQSEL0.EV）。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.18 NVIC_IPR1 寄存器（偏移 = 404h）[复位 = 00000000h]

图 2-89 展示了 NVIC_IPR1，表 2-125 对其进行了介绍。

返回到汇总表。

Irq 4 至 7 的优先级
该寄存器用于为每个可用中断分配优先级，范围为 0 至 255。0 是最高优先级，255 是最低优先级。中断优先级寄存器的解释根据 AIRCR.PRIGROUP 中的设置而变化。

图 2-89. NVIC_IPR1 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRI_7								PRI_6								PRI_5								PRI_4							
R/W-0h								R/W-0h								R/W-0h								R/W-0h							

表 2-125. NVIC_IPR1 寄存器字段说明

位	字段	类型	复位	说明
31-24	PRI_7	R/W	0h	中断 7 的优先级（有关详细信息，请参阅 EVENT:CPUIRQSEL7.EV）。
23-16	PRI_6	R/W	0h	中断 6 的优先级（有关详细信息，请参阅 EVENT:CPUIRQSEL6.EV）。
15-8	PRI_5	R/W	0h	中断 5 的优先级（有关详细信息，请参阅 EVENT:CPUIRQSEL5.EV）。
7-0	PRI_4	R/W	0h	中断 4 的优先级（有关详细信息，请参阅 EVENT:CPUIRQSEL4.EV）。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.19 NVIC_IPR2 寄存器（偏移 = 408h）[复位 = 00000000h]

图 2-90 展示了 NVIC_IPR2，表 2-126 对其进行了介绍。

返回到汇总表。

Irq 8 至 11 的优先级
该寄存器用于为每个可用中断分配优先级，范围为 0 至 255。0 是最高优先级，255 是最低优先级。中断优先级寄存器的解释根据 AIRCR.PRIGROUP 中的设置而变化。

图 2-90. NVIC_IPR2 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRI_11								PRI_10								PRI_9								PRI_8							
R/W-0h								R/W-0h								R/W-0h								R/W-0h							

表 2-126. NVIC_IPR2 寄存器字段说明

位	字段	类型	复位	说明
31-24	PRI_11	R/W	0h	中断 11 的优先级（有关详细信息，请参阅 EVENT:CPUIRQSEL11.EV）。
23-16	PRI_10	R/W	0h	中断 10 的优先级（有关详细信息，请参阅 EVENT:CPUIRQSEL10.EV）。
15-8	PRI_9	R/W	0h	中断 9 的优先级（有关详细信息，请参阅 EVENT:CPUIRQSEL9.EV）。
7-0	PRI_8	R/W	0h	中断 8 的优先级（有关详细信息，请参阅 EVENT:CPUIRQSEL8.EV）。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.20 NVIC_IPR3 寄存器（偏移 = 40Ch）[复位 = 00000000h]

图 2-91 展示了 NVIC_IPR3，表 2-127 对其进行了介绍。

返回到汇总表。

Irq 12 至 15 的优先级
该寄存器用于为每个可用中断分配优先级，范围为 0 至 255。0 是最高优先级，255 是最低优先级。中断优先级寄存器的解释根据 AIRCR.PRIGROUP 中的设置而变化。

图 2-91. NVIC_IPR3 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRI_15								PRI_14								PRI_13								PRI_12							
R/W-0h								R/W-0h								R/W-0h								R/W-0h							

表 2-127. NVIC_IPR3 寄存器字段说明

位	字段	类型	复位	说明
31-24	PRI_15	R/W	0h	中断 15 的优先级（有关详细信息，请参阅 EVENT:CPUIRQSEL15.EV）。
23-16	PRI_14	R/W	0h	中断 14 的优先级（有关详细信息，请参阅 EVENT:CPUIRQSEL14.EV）。
15-8	PRI_13	R/W	0h	中断 13 的优先级（有关详细信息，请参阅 EVENT:CPUIRQSEL13.EV）。
7-0	PRI_12	R/W	0h	中断 12 的优先级（有关详细信息，请参阅 EVENT:CPUIRQSEL12.EV）。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.21 NVIC_IPR4 寄存器（偏移 = 410h）[复位 = 00000000h]

图 2-92 展示了 NVIC_IPR4，表 2-128 对其进行了介绍。

返回到汇总表。

Irq 16 至 19 的优先级
该寄存器用于为每个可用中断分配优先级，范围为 0 至 255。0 是最高优先级，255 是最低优先级。中断优先级寄存器的解释根据 AIRCR.PRIGROUP 中的设置而变化。

图 2-92. NVIC_IPR4 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRI_19								PRI_18								PRI_17								PRI_16							
R/W-0h								R/W-0h								R/W-0h								R/W-0h							

表 2-128. NVIC_IPR4 寄存器字段说明

位	字段	类型	复位	说明
31-24	PRI_19	R/W	0h	中断 19 的优先级（有关详细信息，请参阅 EVENT:CPUIRQSEL19.EV）。
23-16	PRI_18	R/W	0h	中断 18 的优先级（有关详细信息，请参阅 EVENT:CPUIRQSEL18.EV）。
15-8	PRI_17	R/W	0h	中断 17 的优先级（有关详细信息，请参阅 EVENT:CPUIRQSEL17.EV）。
7-0	PRI_16	R/W	0h	中断 16 的优先级（有关详细信息，请参阅 EVENT:CPUIRQSEL16.EV）。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.22 NVIC_IPR5 寄存器（偏移 = 414h）[复位 = 00000000h]

图 2-93 展示了 NVIC_IPR5，表 2-129 对其进行了介绍。

返回到汇总表。

Irq 20 至 23 的优先级
该寄存器用于为每个可用中断分配优先级，范围为 0 至 255。0 是最高优先级，255 是最低优先级。中断优先级寄存器的解释根据 AIRCR.PRIGROUP 中的设置而变化。

图 2-93. NVIC_IPR5 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRI_23								PRI_22								PRI_21								PRI_20							
R/W-0h								R/W-0h								R/W-0h								R/W-0h							

表 2-129. NVIC_IPR5 寄存器字段说明

位	字段	类型	复位	说明
31-24	PRI_23	R/W	0h	中断 23 的优先级（有关详细信息，请参阅 EVENT:CPUIRQSEL23.EV）。
23-16	PRI_22	R/W	0h	中断 22 的优先级（有关详细信息，请参阅 EVENT:CPUIRQSEL22.EV）。
15-8	PRI_21	R/W	0h	中断 21 的优先级（有关详细信息，请参阅 EVENT:CPUIRQSEL21.EV）。
7-0	PRI_20	R/W	0h	中断 20 的优先级（有关详细信息，请参阅 EVENT:CPUIRQSEL20.EV）。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.23 NVIC_IPR6 寄存器（偏移 = 418h）[复位 = 00000000h]

图 2-94 展示了 NVIC_IPR6，表 2-130 对其进行了介绍。

返回到汇总表。

Irq 24 至 27 的优先级
该寄存器用于为每个可用中断分配优先级，范围为 0 至 255。0 是最高优先级，255 是最低优先级。中断优先级寄存器的解释根据 AIRCR.PRIGROUP 中的设置而变化。

图 2-94. NVIC_IPR6 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRI_27								PRI_26								PRI_25								PRI_24							
R/W-0h								R/W-0h								R/W-0h								R/W-0h							

表 2-130. NVIC_IPR6 寄存器字段说明

位	字段	类型	复位	说明
31-24	PRI_27	R/W	0h	中断 27 的优先级（有关详细信息，请参阅 EVENT:CPUIRQSEL27.EV）。
23-16	PRI_26	R/W	0h	中断 26 的优先级（有关详细信息，请参阅 EVENT:CPUIRQSEL26.EV）。
15-8	PRI_25	R/W	0h	中断 25 的优先级（有关详细信息，请参阅 EVENT:CPUIRQSEL25.EV）。
7-0	PRI_24	R/W	0h	中断 24 的优先级（有关详细信息，请参阅 EVENT:CPUIRQSEL24.EV）。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.24 NVIC_IPR7 寄存器（偏移 = 41Ch）[复位 = 00000000h]

图 2-95 展示了 NVIC_IPR7，表 2-131 对其进行了介绍。

返回到汇总表。

Irq 28 至 31 的优先级
该寄存器用于为每个可用中断分配优先级，范围为 0 至 255。0 是最高优先级，255 是最低优先级。中断优先级寄存器的解释根据 AIRCR.PRIGROUP 中的设置而变化。

图 2-95. NVIC_IPR7 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRI_31								PRI_30								PRI_29								PRI_28							
R/W-0h								R/W-0h								R/W-0h								R/W-0h							

表 2-131. NVIC_IPR7 寄存器字段说明

位	字段	类型	复位	说明
31-24	PRI_31	R/W	0h	中断 31 的优先级（有关详细信息，请参阅 EVENT:CPUIRQSEL31.EV）。
23-16	PRI_30	R/W	0h	中断 30 的优先级（有关详细信息，请参阅 EVENT:CPUIRQSEL30.EV）。
15-8	PRI_29	R/W	0h	中断 29 的优先级（有关详细信息，请参阅 EVENT:CPUIRQSEL29.EV）。
7-0	PRI_28	R/W	0h	中断 28 的优先级（有关详细信息，请参阅 EVENT:CPUIRQSEL28.EV）。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.25 NVIC_IPR8 寄存器（偏移 = 420h）[复位 = 00000000h]

图 2-96 展示了 NVIC_IPR8，表 2-132 对其进行了介绍。

返回到汇总表。

Irq 32 至 35 的优先级
该寄存器用于为每个可用中断分配优先级，范围为 0 至 255。0 是最高优先级，255 是最低优先级。中断优先级寄存器的解释根据 AIRCR.PRIGROUP 中的设置而变化。

图 2-96. NVIC_IPR8 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRI_35								PRI_34								PRI_33								PRI_32							
R/W-0h								R/W-0h								R/W-0h								R/W-0h							

表 2-132. NVIC_IPR8 寄存器字段说明

位	字段	类型	复位	说明
31-24	PRI_35	R/W	0h	中断 35 的优先级（有关详细信息，请参阅 EVENT:CPUIRQSEL35.EV）。
23-16	PRI_34	R/W	0h	中断 34 的优先级（有关详细信息，请参阅 EVENT:CPUIRQSEL34.EV）。
15-8	PRI_33	R/W	0h	中断 33 的优先级（有关详细信息，请参阅 EVENT:CPUIRQSEL33.EV）。
7-0	PRI_32	R/W	0h	中断 32 的优先级（有关详细信息，请参阅 EVENT:CPUIRQSEL32.EV）。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.26 NVIC_IPR9 寄存器（偏移 = 424h）[复位 = 00000000h]

图 2-97 展示了 NVIC_IPR9，表 2-133 对其进行了介绍。

返回到汇总表。

Irq 32 至 35 的优先级
该寄存器用于为每个可用中断分配优先级，范围为 0 至 255。0 是最高优先级，255 是最低优先级。中断优先级寄存器的解释根据 AIRCR.PRIGROUP 中的设置而变化。

图 2-97. NVIC_IPR9 寄存器

31	30	29	28	27	26	25	24	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																PRI_37								PRI_36							
R/W-0h																R/W-0h								R/W-0h							

表 2-133. NVIC_IPR9 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
15-8	PRI_37	R/W	0h	中断 37 的优先级（有关详细信息，请参阅 EVENT:CPUIRQSEL37.EV）。
7-0	PRI_36	R/W	0h	中断 36 的优先级（有关详细信息，请参阅 EVENT:CPUIRQSEL36.EV）。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.27 CPUID 寄存器（偏移 = D00h）[复位 = 410FC241h]

图 2-98 展示了 CPUID，表 2-134 中对此进行了介绍。

返回到汇总表。

CPUID 基地址
该寄存器可确定处理器内核的 ID 号、处理器内核的版本号以及处理器内核的实现细节。

图 2-98. CPUID 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IMPLEMENTER								VARIANT				常量			
R-41h								R-0h				R-Fh			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PARTNO												修订版本			
R-C24h												R-1h			

表 2-134. CPUID 寄存器字段说明

位	字段	类型	复位	说明
31-24	IMPLEMENTER	R	41h	实现者代码。
23-20	型号	R	0h	实现定义的变体编号。
19-16	常量	R	Fh	读为 0xF
15-4	PARTNO	R	C24h	系列内部的处理器编号。
3-0	修订版本	R	1h	实现定义的版本号。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.28 ICSR 寄存器（偏移 = D04h）[复位 = 00000000h]

图 2-99 展示了 ICSR，表 2-135 中对此进行了介绍。

返回到汇总表。

中断控制状态
该寄存器用于设置挂起的不可屏蔽中断 (NMI)、设置或清除挂起的 SVC、设置或清除挂起的 SysTick、检查挂起的异常、检查具有最高优先级的挂起异常的矢量编号，以及检查活动异常的矢量编号。

图 2-99. ICSR 寄存器

31	30	29	28	27	26	25	24
NMIPENDSET	RESERVED		PENDSVSET	PENDSVCLR	PENDSTSET	PENDSTCLR	RESERVED
R/W-0h	R/W-0h		R/W-0h	W-X	R/W-0h	W-X	R-0h
23	22	21	20	19	18	17	16
ISRPREEMPT	ISRPENDING	RESERVED				VECTPENDING	
R-0h	R-0h	R-0h				R-0h	
15	14	13	12	11	10	9	8
VECTPENDING				RETTOBASE	RESERVED		VECTACTIVE
R-0h				R-0h	R-0h		R-0h
7	6	5	4	3	2	1	0
VECTACTIVE							
R-0h							

表 2-135. ICSR 寄存器字段说明

位	字段	类型	复位	说明
31	NMIPENDSET	R/W	0h	设置挂起的 NMI 位。设置该位会挂起并激活 NMI。由于 NMI 是具有最高优先级的中断，因此它会在注册后立即生效。 0：无操作 1：设置挂起的 NMI
30-29	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
28	PENDSVSET	R/W	0h	设置挂起的 pendSV 位。 0：无操作 1：设置挂起的 PendSV
27	PENDSVCLR	W	X	清除挂起的 pendSV 位 0：无操作 1：清除挂起的 pendSV
26	PENDSTSET	R/W	0h	设置挂起的 SysTick 位。 0：无操作 1：设置挂起的 SysTick
25	PENDSTCLR	W	X	清除挂起的 SysTick 位 0：无操作 1：清除挂起的 SysTick
24	RESERVED	R	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
23	ISRPREEMPT	R	0h	该字段只能在调试时使用。它表示将在下一个运行周期中响应挂起的中断。如果 DHCSR.C_MASKINTS = 0，则处理中断。 0：不处理挂起的异常。 1：在退出调试暂停状态时处理挂起的异常
22	ISRPENDING	R	0h	中断挂起标志。不包括 NMI 和故障。 0x0：中断未挂起 0x1：中断挂起

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 2-135. ICSR 寄存器字段说明（续）

位	字段	类型	复位	说明
21-18	RESERVED	R	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
17-12	VECTPENDING	R	0h	挂起的 ISR 编号字段。该字段包含具有最高优先级的挂起 ISR 的中断编号。
11	RETTOBASE	R	0h	指示是否有被抢占的活动异常： 0：有被抢占的活动异常待执行 1：没有活动异常，或者当前正在执行的异常是唯一的活动异常。
10-9	RESERVED	R	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
8-0	VECTACTIVE	R	0h	活动 ISR 编号字段。复位会清除该字段。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.29 VTOR 寄存器（偏移 = D08h）[复位 = 00000000h]

图 2-100 展示了 VTOR，表 2-136 中对此进行了介绍。

返回到汇总表。

矢量表偏移
该寄存器用于重定位矢量表基地址。矢量表基地址偏移决定了距存储器映射底部的偏移量。矢量表基地址偏移的两个最高有效位和七个最低有效位必须为 0。矢量表基地址偏移寄存器中允许更改的部分是 TBLOFF。

图 2-100. VTOR 寄存器

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

表 2-136. VTOR 寄存器字段说明

位	字段	类型	复位	说明
31-30	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
29-7	TBLOFF	R/W	0h	矢量表基地址偏移寄存器的位 29 到 7。
6-0	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.30 AIRCR 寄存器（偏移 = D0Ch）[复位 = FA050000h]

图 2-101 展示了 AIRCR，表 2-137 中对此进行了介绍。

返回到汇总表。

应用中断/复位控制
该寄存器用于确定数据字节序、清除所有活动状态信息以进行调试或从硬故障中恢复、执行系统复位，以及更改优先级分组位置（二进制点）。

图 2-101. AIRCR 寄存器

31	30	29	28	27	26	25	24
VECTKEY							
R/W-FA05h							
23	22	21	20	19	18	17	16
VECTKEY							
R/W-FA05h							
15	14	13	12	11	10	9	8
ENDIANESS	RESERVED				PRIGROUP		
R-0h	R-0h				R/W-0h		
7	6	5	4	3	2	1	0
RESERVED					SYSRESETREQ	VECTCLRACTIVE	VECTRESET
R/W-0h					W-0h	W-0h	W-0h

表 2-137. AIRCR 寄存器字段说明

位	字段	类型	复位	说明
31-16	VECTKEY	R/W	FA05h	寄存器密钥。写入该寄存器 (AIRCR) 需要 VECTKEY 字段为 0x05FA。否则写入值会被忽略。读取始终返回 0xFA05。
15	ENDIANESS	R	0h	数据字节序位 0h = 小端字节序 1h = 大端字节序
14-11	RESERVED	R	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
10-8	PRIGROUP	R/W	0h	中断优先级分组字段。该字段是一个二进制点位置指示符，用于为具有相同抢占级别的异常创建子优先级。它将中断优先级寄存器（NVIC_IPR0、NVIC_IPR1 至 NVIC_IPR8）中的 PRI_n 字段分为抢占级别和子优先级。二进制点位于值的左侧。这意味着 PRIGROUP 值表示一个从最低有效位 (LSB) 左侧起始的点。根据分配给优先级的位数以及实现选择，最小值可能不是 0。
7-3	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
2	SYSRESETREQ	W	0h	请求热复位。设置该位不会阻止暂停调试运行。
1	VECTCLRACTIVE	W	0h	清除活动 NMI、故障和中断的所有活动状态信息。应用程序负责重新初始化堆栈。该位用于在调试期间返回到已知状态。该位会自行清除。此操作不会清除 IPSR。因此，如果应用程序使用该位，则只能在激活的基础级别使用，或在可以设置活动位的系统处理程序内使用。
0	VECTRESET	W	0h	系统复位位。复位系统，调试组件除外。该位保留用于调试，仅当内核处于暂停状态时，才能写入 1。该位会自行清除。在内核未处于暂停状态时向该位写入 1 可能会导致不可预测的行为。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.31 SCR 寄存器（偏移 = D10h）[复位 = 00000000h]

图 2-102 展示了 SCR，表 2-138 中对此进行了介绍。

返回到汇总表。

系统控制
该寄存器用于电源管理功能，即当处理器可以进入低功耗状态时向系统发出信号，从而控制处理器进入和退出低功耗状态的方式。

图 2-102. SCR 寄存器

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			SEVONPEND	RESERVED	SLEEPDEEP	SLEEPONEXIT	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 2-138. SCR 寄存器字段说明

位	字段	类型	复位	说明
31-5	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
4	SEVONPEND	R/W	0h	挂起时发送事件位： 0：只有启用的中断或事件才能唤醒处理器，禁用的中断不可以 1：启用的事件和所有中断（包括禁用的中断）都可以唤醒处理器。 当事件或中断进入挂起状态时，事件信号会将处理器从 WFE 状态唤醒。 如果处理器没有等待事件，则会记录事件并影响下一个 WFE。 处理器还会在执行 SEV 指令时被唤醒。
3	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
2	SLEEPDEEP	R/W	0h	控制处理器使用睡眠模式还是深度睡眠模式作为其低功耗模式 0h = 睡眠模式 1h = 深度睡眠模式
1	SLEEPONEXIT	R/W	0h	从处理程序模式返回线程模式时，在退出时进入睡眠状态。使中断驱动的应用程序能够避免返回到空闲的主应用程序。 0：返回线程模式时不进入睡眠状态 1：在 ISR 退出时进入睡眠状态
0	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.32 CCR 寄存器（偏移 = D14h）[复位 = 00000200h]

图 2-103 展示了 CCR，表 2-139 中对此进行了介绍。

返回到汇总表。

配置控制
该寄存器用于使能 NMI、硬故障和 FAULTMASK 以忽略总线故障、除零陷阱和非对齐访问，使用户能够访问软件触发中断寄存器 (STIR)，并控制进入线程模式的行为。

图 2-103. CCR 寄存器

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						STKALIGN	BFHFNMIGN
R/W-0h						R/W-1h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED			DIV_0_TRP	UNALIGN_TRP	RESERVED	USERSETMPEND	NONBASETHREDENA
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 2-139. CCR 寄存器字段说明

位	字段	类型	复位	说明
31-10	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
9	STKALIGN	R/W	1h	栈对齐位。 0：在异常进入时，对于在异常发生之前使用的堆栈指针 (SP)，只能保证 4 字节对齐。 1：在异常进入时，在异常发生之前使用的 SP 会调整为 8 字节对齐，并保存用于恢复的上下文。在返回关联的异常时恢复 SP。
8	BFHFNMIGN	R/W	0h	使能优先级为 -1 或 -2 的处理程序，以忽略由加载和存储指令引起的数据总线故障。这适用于硬故障、NMI 和 FAULTMASK 升级处理程序： 0：加载和存储指令导致的数据总线故障会导致锁定 1：加载和存储指令导致的数据总线故障会被忽略。 仅当处理程序及其数据位于绝对安全的存储器中时，才将该位设置为 1。 该位的正常用途是探测系统器件和桥接器以检测问题。
7-5	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
4	DIV_0_TRP	R/W	0h	在处理器执行除数为 0 的 SDIV 或 UDIV 指令时使能故障触发或暂停： 0：不触发除零异常。在此模式下，除以零将返回商 0。 1：触发除零异常。相关的使用故障状态寄存器位为 CFSR.DIVBYZERO。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 2-139. CCR 寄存器字段说明（续）

位	字段	类型	复位	说明
3	UNALIGN_TRP	R/W	0h	使能非对齐访问陷阱： 0：不触发非对齐半字和字访问异常 1：触发非对齐半字和字访问异常。相关的使用故障状态寄存器位为 CFSR.UNALIGNED。 如果该位设置为 1，则非对齐访问会生成使用故障。 无论 UNALIGN_TRP 中的值如何，非对齐的 LDM、STM、LDRD 和 STRD 指令始终导致故障。
2	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
1	USERSETMPEND	R/W	0h	使能对 STIR 的非特权软件访问： 0：用户代码不允许写入软件触发中断寄存器 (STIR)。 1：用户代码可以写入软件触发中断寄存器 (STIR)，以触发（挂起）与主堆栈指针关联的主异常。
0	NONBASETHREDENA	R/W	0h	指示处理器进入线程模式的方式： 0：只有在没有活动异常时，处理器才能进入线程模式。 1：处理器可以使用适当的返回值 (EXC_RETURN)，从任何级别进入线程模式。 在处理程序模式下，当以下指令之一将值 0xFXXXXXXX 加载到 PC 时，会发生异常返回： - 包含加载 PC 操作的 POP/LDM。 - 以 PC 为目的地的 LDR。 - 使用任意寄存器的 BX。 写入 PC 的值会被截获，称为 EXC_RETURN 值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.33 SHPR1 寄存器（偏移 = D18h）[复位 = 00000000h]

图 2-104 展示了 SHPR1，表 2-140 对其进行了介绍。

返回到汇总表。

系统处理程序 4-7 的优先级
该寄存器用于为以下系统处理程序分配优先级：存储器管理、总线故障和使用故障。系统处理程序是一类特殊的异常处理程序，可以设置为任何优先级。其中大多数可以开启屏蔽（启用）或解除屏蔽（禁用）。禁用时，故障始终被视为硬故障。

图 2-104. SHPR1 寄存器

31	30	29	28	27	26	25	24	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								PRI_6								PRI_5								PRI_4							
R-0h								R/W-0h								R/W-0h								R/W-0h							

表 2-140. SHPR1 寄存器字段说明

位	字段	类型	复位	说明
31-24	RESERVED	R	0h	保留
23-16	PRI_6	R/W	0h	系统处理程序 6 的优先级：UsageFault
15-8	PRI_5	R/W	0h	系统处理程序 5 的优先级：BusFault
7-0	PRI_4	R/W	0h	系统处理程序 4 的优先级：MemManage

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.34 SHPR2 寄存器（偏移 = D1Ch）[复位 = 00000000h]

图 2-105 展示了 SHPR2，表 2-141 对其进行了介绍。

返回到汇总表。

系统处理程序 8-11 的优先级
该寄存器用于为 SVC 处理程序分配优先级。系统处理程序是一类特殊的异常处理程序，可以设置为任何优先级。其中大多数可以开启屏蔽（启用）或解除屏蔽（禁用）。禁用时，故障始终被视为硬故障。

图 2-105. SHPR2 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRI_11								RESERVED																							
R/W-0h								R-0h																							

表 2-141. SHPR2 寄存器字段说明

位	字段	类型	复位	说明
31-24	PRI_11	R/W	0h	系统处理程序 11 的优先级：SVCall
23-0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.35 SHPR3 寄存器（偏移 = D20h）[复位 = 00000000h]

图 2-106 展示了 SHPR3，表 2-142 对其进行了介绍。

返回到汇总表。

系统处理程序 12-15 的优先级
该寄存器用于为以下系统处理程序分配优先级：SysTick、PendSV 和调试监视器。系统处理程序是一类特殊的异常处理程序，可以设置为任何优先级。其中大多数可以开启屏蔽（启用）或解除屏蔽（禁用）。禁用时，故障始终被视为硬故障。

图 2-106. SHPR3 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRI_15								PRI_14								RESERVED								PRI_12							
R/W-0h								R/W-0h								R-0h								R/W-0h							

表 2-142. SHPR3 寄存器字段说明

位	字段	类型	复位	说明
31-24	PRI_15	R/W	0h	系统处理程序 15 的优先级：SysTick 异常
23-16	PRI_14	R/W	0h	系统处理程序 14 的优先级：PendSV
15-8	RESERVED	R	0h	保留
7-0	PRI_12	R/W	0h	系统处理程序 12 的优先级：调试监视器

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.36 SHCSR 寄存器（偏移 = D24h）[复位 = 00000000h]

图 2-107 展示了 SHCSR，表 2-143 中对此进行了介绍。

返回到汇总表。

系统处理程序控制和状态
该寄存器用于启用或禁用系统处理程序、确定总线故障、存储器管理故障和 SVC 的挂起状态，以及确定系统处理程序的活动状态。如果在其故障处理程序禁用时出现故障情况，则故障会升级为硬故障。

图 2-107. SHCSR 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED					USGFAULTENA	BUSFAULTENA	MEMFAULTENA
R/W-0h					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
SVCALLPENDED	BUSFAULTPENDED	MEMFAULTPENDED	USGFAULTPENDED	SYSTICKACT	PENDSVACT	RESERVED	MONITORACT
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
SVCALLACT	RESERVED			USGFAULTACT	RESERVED	BUSFAULTACT	MEMFAULTACT
R-0h	R-0h			R-0h	R-0h	R-0h	R-0h

表 2-143. SHCSR 寄存器字段说明

位	字段	类型	复位	说明
31-19	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
18	USGFAULTENA	R/W	0h	使用故障系统处理程序使能 0h = 禁用异常 1h = 启用异常
17	BUSFAULTENA	R/W	0h	总线故障系统处理程序使能 0h = 禁用异常 1h = 启用异常
16	MEMFAULTENA	R/W	0h	存储器管理故障系统处理程序使能 0h = 禁用异常 1h = 启用异常
15	SVCALLPENDED	R	0h	SVCall 挂起 0h = 异常处于非活动状态 1h = 异常挂起。
14	BUSFAULTPENDED	R	0h	总线故障挂起 0h = 异常处于非活动状态 1h = 异常挂起。
13	MEMFAULTPENDED	R	0h	存储器管理异常挂起 0h = 异常处于非活动状态 1h = 异常挂起。
12	USGFAULTPENDED	R	0h	使用故障挂起 0h = 异常处于非活动状态 1h = 异常挂起。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 2-143. SHCSR 寄存器字段说明（续）

位	字段	类型	复位	说明
11	SYSTICKACT	R	0h	SysTick 活动标志。 0x0：非活动 0x1：活动 0h = 异常处于非活动状态 1h = 异常处于活动状态
10	PENDSVACT	R	0h	PendSV 活动 0x0：非活动 0x1：有效
9	RESERVED	R	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
8	MONITORACT	R	0h	调试监视器活动 0h = 异常处于非活动状态 1h = 异常处于活动状态
7	SVCALLACT	R	0h	SVCall 活动 0h = 异常处于非活动状态 1h = 异常处于活动状态
6-4	RESERVED	R	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
3	USGFAULTACT	R	0h	使用故障异常活动 0h = 异常处于非活动状态 1h = 异常处于活动状态
2	RESERVED	R	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
1	BUSFAULTACT	R	0h	总线故障异常活动 0h = 异常处于非活动状态 1h = 异常处于活动状态
0	MEMFAULTACT	R	0h	存储器管理异常活动 0h = 异常处于非活动状态 1h = 异常处于活动状态

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.37 CFSR 寄存器（偏移 = D28h）[复位 = 00000000h]

图 2-108 展示了 CFSR，表 2-144 中对此进行了介绍。

返回到汇总表。

可配置故障状态
该寄存器用于获取有关本地故障的信息。这些寄存器包括三个子部分：第一个字节是存储器管理故障状态寄存器 (MMFSR)。第二个字节是总线故障状态寄存器 (BFSR)。高半字是使用故障状态寄存器 (UFSR)。这些寄存器中的标志指示本地故障出现的原因。如果发生多个故障，可以设置多个标志。这些寄存器为读/写清型寄存器。这意味着它们可以正常读取，但向任何位写入 1 都会清除该位。
CFSR 可按字节访问。CFSR 或其子寄存器可按如下方式访问：
可以对 CFSR 寄存器进行以下访问：
- 通过对 0xE000ED28 进行字访问来访问完整的寄存器。
- 通过对 0xE000ED28 进行字节访问来访问 MMFSR
- 通过对 0xE000ED28 进行半字访问来访问 MMFSR 和 BFSR
- 通过对 0xE000ED29 进行字节访问来访问 BFSR
- 通过对 0xE000ED2A 进行半字访问来访问 UFSR。

图 2-108. CFSR 寄存器

31	30	29	28	27	26	25	24
RESERVED						DIVBYZERO	UNALIGNED
R/W-0h						R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED				NOCP	INVPC	INVSTATE	UNDEFINSTR
R/W-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
BFARVALID	RESERVED		STKERR	UNSTKERR	IMPRECISERR	PRECISERR	IBUSERR
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
MMARVALID	RESERVED		MSTKERR	MUNSTKERR	RESERVED	DACCVIOL	IACCVIOL
R/W-0h	R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 2-144. CFSR 寄存器字段说明

位	字段	类型	复位	说明
31-26	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
25	DIVBYZERO	R/W	0h	CCR.DIV_0_TRP（请参阅第 8-26 页的“配置控制寄存器”）已启用且 SDIV 或 UDIV 指令使用的除数为 0 时，会发生该故障。指令将被执行，且返回的 PC 指向该指令。如果未设置 CCR.DIV_0_TRP，则除法运算将返回商 0。
24	UNALIGNED	R/W	0h	当 CCR.UNALIGN_TRP 已启用且尝试进行非对齐存储器访问时，会发生此故障。无论 CCR.UNALIGN_TRP 的设置如何，非对齐的 LDM/STM/LDRD/STRD 指令始终导致故障。
23-20	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
19	NOCP	R/W	0h	尝试使用协处理器指令。处理器不支持协处理器指令。
18	INVPC	R/W	0h	尝试非法将 EXC_RETURN 加载到 PC 中。无效指令、无效上下文、无效值。返回 PC 指向尝试设置 PC 的指令。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 2-144. CFSR 寄存器字段说明（续）

位	字段	类型	复位	说明
17	INVSTATE	R/W	0h	表示尝试在无效 EPSR 状态下执行（例如，在 BX 类型指令更改状态后）。这包括进入异常或从异常返回后的状态变化，以及互操作指令引起的状态变化。返回 PC 指向导致故障的指令，并且携带无效状态。
16	UNDEFINSTR	R/W	0h	当处理器尝试执行未定义的指令时，设置该位。未定义的指令是指处理器无法解码的指令。返回 PC 指向未定义的指令。
15	BFARVALID	R/W	0h	如果总线故障地址寄存器 (BFAR) 包含有效地址，则会设置该位。在地址已知的总线故障发生后，同样设置该位。其他故障则会清除该位，例如后发生的存储器管理故障。如果发生总线故障且该故障由于优先级而升级为硬故障，则硬故障处理程序必须清除该位。这可防止在返回已压栈的活动总线故障处理程序时，因其 BFAR 值已被覆写而出现问题。
14-13	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
12	STKERR	R/W	0h	异常压栈过程已导致一个或多个总线故障。仍会调整 SP，堆栈上下文区域中的值可能不正确。不写入 BFAR。
11	UNSTKERR	R/W	0h	异常返回时的出栈过程已导致一个或多个总线故障。这会链接到处理程序，以便保留原始返回栈。不会因故障返回而调整 SP，也不会执行新的保存。不写入 BFAR。
10	IMPRECISERR	R/W	0h	数据总线不精确错误。这是一个总线故障，但返回 PC 与导致该故障的指令无关。这不是同步故障。因此，如果在当前激活的优先级高于总线故障时检测到该故障，则只会挂起该故障。当返回到较低优先级的激活时，总线故障激活。如果在返回较低优先级的异常之前发生精确故障，处理程序会同时检测到 IMPRECISERR 置位以及某个精确故障状态位被置位。不写入 BFAR。
9	PRECISERR	R/W	0h	精确数据总线错误返回。
8	IBUSERR	R/W	0h	指令总线错误标志。该标志由预取错误设置。故障会在该指令处停止，因此如果错误在分支影子下发生，则不会发生故障。不写入 BFAR。
7	MMARVALID	R/W	0h	存储器管理地址寄存器 (MMFAR) 地址有效标志。后到达的故障（如总线故障）可以清除存储器管理故障。如果发生存储器管理故障且该故障由于优先级而升级为硬故障，则硬故障处理程序必须清除该位。这可防止在返回已压栈的活动存储器管理处理程序时，因其 MMFAR 值已被覆写而出现问题。
6-5	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
4	MSTKERR	R/W	0h	异常压栈过程已导致一个或多个访问违例。仍会调整 SP，堆栈上下文区域中的值可能不正确。不写入 MMFAR。
3	MUNSTKERR	R/W	0h	异常返回时的出栈过程已导致一个或多个访问违例。这会链接到处理程序，以便保留原始返回栈。不会因故障返回而调整 SP，也不会执行新的保存。不写入 MMFAR。
2	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
1	DACCVIOL	R/W	0h	数据访问违例标志。如果尝试在不允许操作的位置加载或存储，则会设置此标志。返回 PC 指向导致故障的指令。该错误会将所尝试访问的地址加载到 MMFAR 中。
0	IACCVIOL	R/W	0h	指令访问违例标志。如果尝试从不允许执行的位置获取指令，则会设置此标志。即使 MPU 已被禁用或不存在，但只要对 XN 区域进行任何访问，就会发生这种情况。返回 PC 指向导致故障的指令。不写入 MMFAR。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.38 HFSR 寄存器（偏移 = D2Ch）[复位 = 00000000h]

图 2-109 展示了 HFSR，表 2-145 中对此进行了介绍。

返回到汇总表。

硬故障状态
该寄存器用于获取有关激活硬故障处理程序的事件的信息。该寄存器是写清型寄存器。这意味着向某个位写入 1 会清除该位。

图 2-109. HFSR 寄存器

31	30	29	28	27	26	25	24
DEBUGEVT	FORCED	RESERVED					
R/W1C-0h	R/W1C-0h	R/W-0h					
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED						VECTTBL	RESERVED
R/W-0h						R/W1C-0h	R/W-0h

表 2-145. HFSR 寄存器字段说明

位	字段	类型	复位	说明
31	DEBUGEVT	R/W1C	0h	如果存在与调试相关的故障，则设置该位。仅当未启用暂停调试时，才会发生该故障。对于启用监视器的调试，仅当当前优先级高于监视器时，执行 BKPT 指令时才会发生该故障。当暂停调试和监视器调试都被禁用时，仅当调试事件未被忽略时，才会发生该故障（至少包含 BKPT 指令）。调试故障状态寄存器会被更新。
30	FORCED	R/W1C	0h	硬故障激活是因为接收到可配置故障，而无法激活是由于优先级原因或因为可配置故障被禁用。然后，硬故障处理程序必须读取其他故障状态寄存器以确定原因。
29-2	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
1	VECTTBL	R/W1C	0h	如果由于异常处理时读取的矢量表而导致故障（总线故障），则设置该位。这种情况下，故障始终是硬故障。返回 PC 指向被抢占的指令。
0	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.39 DFSR 寄存器（偏移 = D30h）[复位 = 00000000h]

图 2-110 展示了 DFSR，表 2-146 中对此进行了介绍。

返回到汇总表。

调试故障状态
该寄存器用于监视外部调试请求、矢量捕获、数据观察点匹配、BKPT 指令执行和暂停请求。当发生多种故障情况时，可以在调试故障状态寄存器中设置多个标志。该寄存器为读/写清型寄存器。这意味着它可以正常读取，而向某个位写入 1 则会清除该位。请注意，除非捕获到事件，否则不会设置这些位。这意味着它会导致某种停止。如果启用了暂停调试，这些事件会使处理器停止并进入调试状态。如果禁用了调试并启用了调试监视器，则在优先级允许的情况下，这将成为调试监视器处理程序调用。如果调试和监视器都被禁用，则其中一些事件为硬故障，而另一些事件将被忽略。

图 2-110. DFSR 寄存器

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			外部	VCATCH	DWTTRAP	BKPT	HALTED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 2-146. DFSR 寄存器字段说明

位	字段	类型	复位	说明
31-5	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
4	外部	R/W	0h	外部调试请求标志。处理器在下一指令边界处停止。 0x0：外部调试请求信号未置为有效 0x1：外部调试请求信号置为有效
3	VCATCH	R/W	0h	矢量捕获标志。设置该标志时，也会设置本地故障状态寄存器中的一个标志，以指示故障类型。 0x0：未发生矢量捕获 0x1：发生了矢量捕获
2	DWTTRAP	R/W	0h	数据观察点和跟踪 (DWT) 标志。处理器在当前指令或下一条指令处停止。 0x0：无 DWT 匹配 0x1：有 DWT 匹配
1	BKPT	R/W	0h	BKPT 标志。BKPT 标志由闪存补丁代码中的 BKPT 指令以及普通代码设置。返回 PC 指向包含断点的指令。 0x0：无 BKPT 指令执行 0x1：有 BKPT 指令执行
0	HALTED	R/W	0h	暂停请求标志。处理器在下一条指令处暂停。 0x0：无暂停请求 0x1：NVIC 请求暂停，包括步骤

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.40 MMFAR 寄存器（偏移 = D34h）[复位 = 00000000h]

图 2-111 展示了 MMFAR，表 2-147 中对此进行了介绍。

返回到汇总表。

存储器管理故障地址
该寄存器用于读取导致存储器管理故障的位置的地址。

图 2-111. MMFAR 寄存器

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

表 2-147. MMFAR 寄存器字段说明

位	字段	类型	复位	说明
31-0	地址	R/W	X	存储器管理故障地址字段。 该字段是故障加载或存储尝试的数据地址。发生非对齐访问故障时，地址是出现故障的实际地址。由于访问可以拆分为多个单独对齐的部分，因此该地址可以是请求大小范围内的任何偏移。标记 CFSR.IACCVIOL、CFSR.DACCVIOL、CFSR.MUNSTKERR 和 CFSR.MSTKERR 与 CFSR.MMARVALID 一起指示故障原因。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.41 BFAR 寄存器（偏移 = D38h）[复位 = 00000000h]

图 2-112 展示了 BFAR，表 2-148 中对此进行了介绍。

返回到汇总表。

总线故障地址
该寄存器用于读取生成总线故障的位置的地址。

图 2-112. BFAR 寄存器

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

表 2-148. BFAR 寄存器字段说明

位	字段	类型	复位	说明
31-0	地址	R/W	X	总线故障地址字段。该字段是故障加载或存储尝试的数据地址。发生非对齐访问故障时，地址是指令请求的地址，即使该地址不是出现故障的地址。 标记 CFSR.IBUSERR、CFSR.PRECISERR、CFSR.IMPRECISERR、CFSR.UNSTKERR 和 CFSR.STKERR 与 CFSR.BFARVALID 一起指示故障原因。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.42 AFSR 寄存器（偏移 = D3Ch）[复位 = 00000000h]

图 2-113 展示了 AFSR，表 2-149 中对此进行了介绍。

返回到汇总表。

辅助故障状态
该寄存器用于确定向软件提供的其他系统故障信息。辅助故障上的单周期高电平信号锁存为 1。只能通过向相应位写入 1 来清除该位。CPU 的辅助故障输入连接到 0。

图 2-113. AFSR 寄存器

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

表 2-149. AFSR 寄存器字段说明

位	字段	类型	复位	说明
31-0	IMPDEF	R/W	0h	实现定义。这些位直接映射到辅助故障输入的信号分配。连接到 0

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.43 ID_PFR0 寄存器（偏移 = D40h）[复位 = 00000030h]

图 2-114 展示了 ID_PFR0，表 2-150 对其进行了介绍。

返回到汇总表。

处理器功能 0

图 2-114. ID_PFR0 寄存器

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								STATE1				STATE0			
R-0h								R-3h				R-0h			

表 2-150. ID_PFR0 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
7-4	STATE1	R	3h	State1（T 位 = 1） 0x0：不适用 0x1：不适用 0x2：Thumb-2 编码，包含 16 位基本指令加 32 位 Buncond/BL，但不包含其他 32 位基本指令（请注意，可以使用适当的指令属性添加 32 位非基本指令，但不能添加其他 32 位基本指令。） 0x3：Thumb-2 编码，包含所有 Thumb-2 基本指令
3-0	STATE0	R	0h	State0（T 位 = 0） 0x0：无 ARM 编码 0x1：不适用

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.44 ID_PFR1 寄存器（偏移 = D44h）[复位 = 00000200h]

图 2-115 展示了 ID_PFR1，表 2-151 对其进行了介绍。

返回到汇总表。

处理器功能 1

图 2-115. ID_PFR1 寄存器

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

表 2-151. ID_PFR1 寄存器字段说明

位	字段	类型	复位	说明
31-12	RESERVED	R	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
11-8	MICROCONTROLLER_PROGRAMMERS_MODEL	R	2h	微控制器编程模型 0x0：不支持 0x2：双栈支持
7-0	RESERVED	R	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.45 ID_DFR0 寄存器（偏移 = D48h）[复位 = 00100000h]

图 2-116 展示了 ID_DFR0，表 2-152 对其进行了介绍。

返回到汇总表。

调试功能 0
该寄存器提供调试系统的概要视图。调试架构本身可提供更多详细信息。

图 2-116. ID_DFR0 寄存器

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

表 2-152. ID_DFR0 寄存器字段说明

位	字段	类型	复位	说明
31-24	RESERVED	R	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
23-20	MICROCONTROLLER_DEBUG_MODEL	R	1h	微控制器调试模型 - 存储器映射 0x0：不支持 0x1：微控制器调试 v1（ITMv1 和 DWTv1）
19-0	RESERVED	R	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.46 ID_AFR0 寄存器（偏移 = D4Ch）[复位 = 00000000h]

图 2-117 展示了 ID_AFR0，表 2-153 对其进行了介绍。

返回到汇总表。

辅助功能 0
该寄存器为实现定义的功能提供了一定的自由度，以便对其进行注册。Cortex-M 中未使用。

图 2-117. ID_AFR0 寄存器

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

表 2-153. ID_AFR0 寄存器字段说明

位	字段	类型	复位	说明
31-0	RESERVED	R	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.47 ID_MMFR0 寄存器（偏移 = D50h）[复位 = 00100030h]

图 2-118 展示了 ID_MMFR0，表 2-154 对其进行了介绍。

返回到汇总表。

存储器模型功能 0
关于存储器模型和存储器管理支持的一般信息。

图 2-118. ID_MMFR0 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-00100030h																															

表 2-154. ID_MMFR0 寄存器字段说明

位	字段	类型	复位	说明
31-0	RESERVED	R	00100030h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.48 ID_MMFR1 寄存器（偏移 = D54h）[复位 = 00000000h]

图 2-119 展示了 ID_MMFR1，表 2-155 对其进行了介绍。

返回到汇总表。

存储器模型功能 1
关于存储器模型和存储器管理支持的一般信息。

图 2-119. ID_MMFR1 寄存器

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

表 2-155. ID_MMFR1 寄存器字段说明

位	字段	类型	复位	说明
31-0	RESERVED	R	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.49 ID_MMFR2 寄存器（偏移 = D58h）[复位 = 01000000h]

图 2-120 展示了 ID_MMFR2，表 2-156 对其进行了介绍。

返回到汇总表。

存储器模型功能 2
关于存储器模型和存储器管理支持的一般信息。

图 2-120. ID_MMFR2 寄存器

31	30	29	28	27	26	25	24
RESERVED							WAIT_FOR_INTERRUPT_STALLING
R-0h							R-1h
23	22	21	20	19	18	17	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							

表 2-156. ID_MMFR2 寄存器字段说明

位	字段	类型	复位	说明
31-25	RESERVED	R	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
24	WAIT_FOR_INTERRUPT_STALLING	R	1h	等待中断停滞 0x0：不支持 0x1：支持等待中断
23-0	RESERVED	R	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.50 ID_MMFR3 寄存器（偏移 = D5Ch）[复位 = 00000000h]

图 2-121 展示了 ID_MMFR3，表 2-157 对其进行了介绍。

返回到汇总表。

存储器模型功能 3
关于存储器模型和存储器管理支持的一般信息。

图 2-121. ID_MMFR3 寄存器

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

表 2-157. ID_MMFR3 寄存器字段说明

位	字段	类型	复位	说明
31-0	RESERVED	R	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.51 ID_ISAR0 寄存器（偏移 = D60h）[复位 = 01101110h]

图 2-122 展示了 ID_ISAR0，表 2-158 对其进行了介绍。

返回到汇总表。

ISA 功能 0
有关指令集属性寄存器的信息

图 2-122. ID_ISAR0 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-01101110h																															

表 2-158. ID_ISAR0 寄存器字段说明

位	字段	类型	复位	说明
31-0	RESERVED	R	01101110h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.52 ID_ISAR1 寄存器（偏移 = D64h）[复位 = 02112000h]

图 2-123 展示了 ID_ISAR1，表 2-159 对其进行了介绍。

返回到汇总表。

ISA 功能 1
有关指令集属性寄存器的信息

图 2-123. ID_ISAR1 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-02112000h																															

表 2-159. ID_ISAR1 寄存器字段说明

位	字段	类型	复位	说明
31-0	RESERVED	R	02112000h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.53 ID_ISAR2 寄存器（偏移 = D68h）[复位 = 21232231h]

图 2-124 展示了 ID_ISAR2，表 2-160 对其进行了介绍。

返回到汇总表。

ISA 功能 2
有关指令集属性寄存器的信息

图 2-124. ID_ISAR2 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-21232231h																															

表 2-160. ID_ISAR2 寄存器字段说明

位	字段	类型	复位	说明
31-0	RESERVED	R	21232231h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.54 ID_ISAR3 寄存器（偏移 = D6Ch）[复位 = 01111131h]

图 2-125 展示了 ID_ISAR3，表 2-161 对其进行了介绍。

返回到汇总表。

ISA 功能 3
有关指令集属性寄存器的信息

图 2-125. ID_ISAR3 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-01111131h																															

表 2-161. ID_ISAR3 寄存器字段说明

位	字段	类型	复位	说明
31-0	RESERVED	R	01111131h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.55 ID_ISAR4 寄存器（偏移 = D70h）[复位 = 01310132h]

图 2-126 展示了 ID_ISAR4，表 2-162 对其进行了介绍。

返回到汇总表。

ISA 功能 4
有关指令集属性寄存器的信息

图 2-126. ID_ISAR4 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R-01310132h																															

表 2-162. ID_ISAR4 寄存器字段说明

位	字段	类型	复位	说明
31-0	RESERVED	R	01310132h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.56 CPACR 寄存器（偏移 = D88h）[复位 = 00000000h]

图 2-127 展示了 CPACR，表 2-163 中对此进行了介绍。

返回到汇总表。

协处理器访问控制
该寄存器指定协处理器的访问权限。

图 2-127. CPACR 寄存器

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

表 2-163. CPACR 寄存器字段说明

位	字段	类型	复位	说明
31-0	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.57 MPU_TYPE 寄存器（偏移 = D90h）[复位 = 00000800h]

图 2-128 展示了 MPU_TYPE，表 2-164 中对此进行了介绍。

返回到汇总表。

MPU 类型
该寄存器指示 MPU 支持的多个区域。

图 2-128. MPU_TYPE 寄存器

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

表 2-164. MPU_TYPE 寄存器字段说明

位	字段	类型	复位	说明
31-24	RESERVED	R	0h	读为 0。
23-16	IREGION	R	0h	处理器内核仅使用统一的 MPU，该字段的读值始终为 0x0。
15-8	DREGION	R	8h	受支持 MPU 区域的数量字段。该字段读值为 0x08，表示八个 MPU 区域。
7-1	RESERVED	R	0h	读为 0。
0	SEPARATE	R	0h	处理器内核仅使用统一的 MPU，因此该字段始终为 0。

Copyright © 2026 Texas Instruments Incorporated

2.9.4.58 MPU_CTRL 寄存器（偏移 = D94h）[复位 = 00000000h]

图 2-129 展示了 MPU_CTRL，表 2-165 中对此进行了介绍。

返回到汇总表。

MPU 控制
该寄存器用于使能 MPU、使能默认存储器映射（背景区域），以及在硬故障、不可屏蔽中断 (NMI) 和 FAULTMASK 升级处理程序中使能 MPU。除非设置了 PRIVDEFENA 位，否则启用 MPU 后，必须启用存储器映射的至少一个区域才能使 MPU 正常工作。如果设置了 PRIVDEFENA 位且未启用任何区域，则只有特权代码可以运行。禁用 MPU 后，将使用默认地址映射，就像不存在 MPU 一样。启用 MPU 后，只有系统分区和矢量表加载可以始终访问。根据区域以及是否启用了 PRIVDEFENA，可以访问其他区域。除非设置了 HFNMIENA，否则当异常优先级为 -1 或 -2 时，MPU 不会启用。仅当处于硬故障、NMI 或 FAULTMASK 已启用时，这些优先级才可行。HFNMIENA 位在以这两个优先级运行时启用 MPU。

图 2-129. MPU_CTRL 寄存器

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					PRIVDEFENA	HFNMIENA	ENABLE
R/W-0h					R/W-0h	R/W-0h	R/W-0h

表 2-165. MPU_CTRL 寄存器字段说明

位	字段	类型	复位	说明
31-3	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
2	PRIVDEFENA	R/W	0h	当 MPU 启用时，该位启用默认存储器映射作为背景区域，用于特权访问。背景区域的作用相当于任何可设置区域之前的 1 号区域。配置的任何区域都会覆盖该默认映射，并覆写该映射。如果未设置该位，则禁用默认存储器映射，并且未被区域覆盖的存储器会发生故障。这适用于存储器类型、从不执行 (XN)、缓存和可共享规则。但是，这仅适用于特权模式（获取和数据访问）。除非已为其代码和数据配置区域，否则用户模式代码会导致故障。MPU 禁用时，默认映射同时作用于特权模式和用户模式代码。无论是否设置了此使能，XN 和 SO 规则始终应用于系统分区。如果 MPU 禁用，则忽略该位。
1	HFNMIENA	R/W	0h	在处于硬故障、NMI 和 FAULTMASK 升级处理程序中时，该位会启用 MPU。如果设置了该位和 ENABLE 位，则在这些处理程序中 MPU 会被启用。如果未设置该位，则无论 ENABLE 位的值如何，在这些处理程序中都会禁用 MPU。如果设置了该位且未设置 ENABLE 位，则行为不可预测。
0	ENABLE	R/W	0h	使能 MPU 0：禁用 MPU 1：MPU 启用

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.59 MPU_RNR 寄存器（偏移 = D98h）[复位 = 00000000h]

图 2-130 展示了 MPU_RNR，表 2-166 中对此进行了介绍。

返回到汇总表。

MPU 区域编号
该寄存器用于选择访问哪个保护区域。对 MPU_RASR 或 MPU_RBAR 的以下写入会配置该寄存器选择的保护区域的特性。

图 2-130. MPU_RNR 寄存器

31	30	29	28	27	26	25	24	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																								区域							
R/W-0h																								R/W-0h							

表 2-166. MPU_RNR 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
7-0	区域	R/W	0h	区域选择字段。 使用 MPU_RASR 和 MPU_RBAR 时，该字段选择要在其上运行的区域。必须首先写入该位，除非同时写入地址 MPU_RBAR.VALID 和 MPU_RBAR.REGION 字段，此时会覆盖该位。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.60 MPU_RBAR 寄存器（偏移 = D9Ch）[复位 = 00000000h]

图 2-131 展示了 MPU_RBAR，表 2-167 中对此进行了介绍。

返回到汇总表。

MPU 区域基地址
该寄存器写入区域的基地址。它还包含一个 REGION 字段，如果设置了 VALID 位，则可以使用该字段来覆写 MPU_RNR.REGION。该寄存器设置区域的基地址。它按大小对齐。因此，64KB 大小的区域必须按 64KB 的倍数对齐，例如 0x00010000 或 0x00020000。区域始终回读为当前 MPU 区域编号。VALID 始终回读为 0。写入 VALID = 1 和 REGION = n 会将区域编号更改为 n。这是写入 MPU_RNR 的简短方法。如果不是作为字访问，则该寄存器的行为不可预测。

图 2-131. MPU_RBAR 寄存器

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

表 2-167. MPU_RBAR 寄存器字段说明

位	字段	类型	复位	说明
31-5	ADDR	R/W	0h	区域基地址字段。 LSB 的位置取决于区域大小，因此基地址会根据大小的偶数倍对齐。 MPU 区域属性和大小寄存器中 SZENABLE 字段指定的 2 的幂次大小决定了基地址中有多少位被使用。
4	有效	R/W	0h	MPU 区域编号有效位： 0：MPU_RNR 保持不变并被解释。 1：MPU_RNR 被 REGION 覆写。
3-0	区域	R/W	0h	MPU 区域覆写字段

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.61 MPU_RASR 寄存器（偏移 = DA0h）[复位 = 00000000h]

图 2-132 展示了 MPU_RASR，表 2-168 中对此进行了介绍。

返回到汇总表。

MPU 区域属性和大小
该寄存器控制 MPU 访问权限。该寄存器由两子寄存器组成，每个寄存器具有半字大小。可以使用半字大小来访问这些寄存器，也可以使用字操作同时访问这两个寄存器。32 字节、64 字节和 128 字节区域大小不支持子区域禁用位。使用这些区域大小时，子区域禁用位必须编程为 0。

图 2-132. MPU_RASR 寄存器

31	30	29	28	27	26	25	24
RESERVED			XN	RESERVED	AP		
R/W-0h			R/W-0h	R/W-0h	R/W-0h		
23	22	21	20	19	18	17	16
RESERVED		TEX			S	C	B
R/W-0h		R/W-0h			R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
SRD							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED		尺寸					ENABLE
R/W-0h		R/W-0h					R/W-0h

表 2-168. MPU_RASR 寄存器字段说明

位	字段	类型	复位	说明
31-29	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
28	XN	R/W	0h	指令访问禁用： 0：启用指令获取 1：禁用指令获取
27	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
26-24	AP	R/W	0h	数据访问权限： 0x0：特权权限：无访问。用户权限：无访问。 0x1：特权权限：读-写。用户权限：无访问。 0x2：特权权限：读-写。用户权限：只读。 0x3：特权权限：读-写。用户权限：读-写 0x4：保留 0x5：特权权限：只读。用户权限：无访问。 0x6：特权权限：只读。用户权限：只读。 0x7：特权权限：只读。用户权限：只读。
23-22	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
21-19	TEX	R/W	0h	类型扩展
18	S	R/W	0h	可共享位： 0：不可共享 1：可共享
17	C	R/W	0h	可缓存位： 0：不可缓存 1：可缓存

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 2-168. MPU_RASR 寄存器字段说明（续）

位	字段	类型	复位	说明
16	B	R/W	0h	可缓冲位： 0：不可缓冲 1：可缓冲
15-8	SRD	R/W	0h	子区域禁用字段： 设置该字段中的某个位会禁用相应的子区域。区域分为八个大小相同的子区域。128 字节及以下区域大小不支持子区域。
7-6	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
5-1	尺寸	R/W	0h	MPU 保护区域大小字段： 0x04：32B 0x05：64B 0x06：128B 0x07：256B 0x08：512B 0x09：1KB 0x0A：2KB 0x0B：4KB 0x0C：8KB 0x0D：16KB 0x0E：32KB 0x0F：64KB 0x10：128KB 0x11：256KB 0x12：512KB 0x13：1MB 0x14：2MB 0x15：4MB 0x16：8MB 0x17：16MB 0x18：32MB 0x19：64MB 0x1A：128MB 0x1B：256MB 0x1C：512MB 0x1D：1GB 0x1E：2GB 0x1F：4GB
0	ENABLE	R/W	0h	区域使能位： 0：禁用区域 1：启用区域

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.62 MPU_RBAR_A1 寄存器（偏移 = DA4h）[复位 = 00000000h]

图 2-133 展示了 MPU_RBAR_A1，表 2-169 对其进行了介绍。

返回到汇总表。

MPU 别名 1 区域基地址
MPU_RBAR 的别名

图 2-133. MPU_RBAR_A1 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MPU_RBAR_A1																															
R/W-0h																															

表 2-169. MPU_RBAR_A1 寄存器字段说明

位	字段	类型	复位	说明
31-0	MPU_RBAR_A1	R/W	0h	MPU_RBAR 的别名

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.63 MPU_RASR_A1 寄存器（偏移 = DA8h）[复位 = 00000000h]

图 2-134 展示了 MPU_RASR_A1，表 2-170 对其进行了介绍。

返回到汇总表。

MPU 别名 1 区域属性和大小
MPU_RASR 的别名

图 2-134. MPU_RASR_A1 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MPU_RASR_A1																															
R/W-0h																															

表 2-170. MPU_RASR_A1 寄存器字段说明

位	字段	类型	复位	说明
31-0	MPU_RASR_A1	R/W	0h	MPU_RASR 的别名

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.64 MPU_RBAR_A2 寄存器（偏移 = DACh）[复位 = 00000000h]

图 2-135 展示了 MPU_RBAR_A2，表 2-171 中对此进行了介绍。

返回到汇总表。

MPU 别名 2 区域基地址
MPU_RBAR 的别名

图 2-135. MPU_RBAR_A2 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MPU_RBAR_A2																															
R/W-0h																															

表 2-171. MPU_RBAR_A2 寄存器字段说明

位	字段	类型	复位	说明
31-0	MPU_RBAR_A2	R/W	0h	MPU_RBAR 的别名

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.65 MPU_RASR_A2 寄存器（偏移 = DB0h）[复位 = 00000000h]

图 2-136 展示了 MPU_RASR_A2，表 2-172 对其进行了介绍。

返回到汇总表。

MPU 别名 2 区域属性和大小
MPU_RASR 的别名

图 2-136. MPU_RASR_A2 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MPU_RASR_A2																															
R/W-0h																															

表 2-172. MPU_RASR_A2 寄存器字段说明

位	字段	类型	复位	说明
31-0	MPU_RASR_A2	R/W	0h	MPU_RASR 的别名

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.66 MPU_RBAR_A3 寄存器（偏移 = DB4h）[复位 = 00000000h]

图 2-137 展示了 MPU_RBAR_A3，表 2-173 对其进行了介绍。

返回到汇总表。

MPU 别名 3 区域基地址
MPU_RBAR 的别名

图 2-137. MPU_RBAR_A3 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MPU_RBAR_A3																															
R/W-0h																															

表 2-173. MPU_RBAR_A3 寄存器字段说明

位	字段	类型	复位	说明
31-0	MPU_RBAR_A3	R/W	0h	MPU_RBAR 的别名

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.67 MPU_RASR_A3 寄存器（偏移 = DB8h）[复位 = 00000000h]

图 2-138 展示了 MPU_RASR_A3，表 2-174 对其进行了介绍。

返回到汇总表。

MPU 别名 3 区域属性和大小
MPU_RASR 的别名

图 2-138. MPU_RASR_A3 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MPU_RASR_A3																															
R/W-0h																															

表 2-174. MPU_RASR_A3 寄存器字段说明

位	字段	类型	复位	说明
31-0	MPU_RASR_A3	R/W	0h	MPU_RASR 的别名

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.68 DHCSR 寄存器（偏移 = DF0h）[复位 = 00000000h]

图 2-139 展示了 DHCSR，表 2-175 中对此进行了介绍。

返回到汇总表。

调试暂停控制和状态
该寄存器的目的是提供有关处理器状态的状态信息、使能内核调试，以及暂停和逐步执行处理器。对于写操作，必须向该寄存器的高半字写入 0xA05F，否则写操作会被忽略，并且不会向寄存器写入任何位。如果未启用暂停模式，则 C_DEBUGEN = 1，所有其他字段均被禁用。该寄存器不会在内核复位时复位，而是由上电复位操作复位。但是，C_HALT 始终会在内核复位时清除。若要在复位时暂停，必须启用以下位：DEMCR.VC_CORERESET 和 C_DEBUGEN。请注意，对该寄存器进行非字大小的写入都会产生不可预测的结果。可以用任何大小读取该寄存器，这可用于避免或有意更改粘滞位。
在 CPU 处于暂停状态时（即 C_DEBUGEN = 1 且 S_HALT= 1）对该寄存器进行写操作时的系统行为：
C_HALT = 0、C_STEP = 0、C_MASKINTS = 0 退出调试状态并开始指令执行。异常根据异常配置规则激活。
C_HALT = 0、C_STEP = 0、C_MASKINTS = 1 退出调试状态并开始指令执行。PendSV、SysTick 和外部可配置中断均被禁用，否则异常会根据标准配置规则激活。
C_HALT = 0、C_STEP = 1、C_MASKINTS = 0 退出调试状态、逐步执行指令并暂停。异常根据异常配置规则激活。
C_HALT = 0、C_STEP = 1、C_MASKINTS = 1 退出调试状态、逐步执行指令并暂停。PendSV、SysTick 和外部可配置中断均被禁用，否则异常会根据标准配置规则激活。
C_HALT = 1、C_STEP = x、C_MASKINTS = x 保持调试状态

图 2-139. DHCSR 寄存器

31	30	29	28	27	26	25	24
RESERVED						S_RESET_ST	S_RETIRE_ST
R/W-0h						R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED				S_LOCKUP	S_SLEEP	S_HALT	S_REGRDY
R/W-0h				R/W-0h	R/W-0h	R/W-0h	R/W-X
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED		C_SNAPSTALL	RESERVED	C_MASKINTS	C_STEP	C_HALT	C_DEBUGEN
R-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 2-175. DHCSR 寄存器字段说明

位	字段	类型	复位	说明
31-26	RESERVED	R/W	0h	软件不应依赖保留位的值。 写入该寄存器时，必须向该位字段写入 0x28，否则写操作将被忽略，并且不会向寄存器写入任何位。
25	S_RESET_ST	R/W	0h	表示自上次读取该位以来内核已复位或现在正在复位。这是一个粘滞位，将在读取后清除。因此，读取两次并先变为 1 后变为 0 表示它已被复位。读取两次并两次变为 1 表示现在正在复位（仍保持在复位状态）。 写入该寄存器时，必须向该位字段写入 0，否则写操作将被忽略，并且不会向寄存器写入任何位。
24	S_RETIRE_ST	R/W	0h	表示自上次读取后有一条指令已完成。这是一个粘滞位，将在读取后清除。可以用于确定内核是在加载/存储还是获取时停滞。 写入该寄存器时，必须向该位字段写入 0，否则写操作将被忽略，并且不会向寄存器写入任何位。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 2-175. DHCSR 寄存器字段说明（续）

位	字段	类型	复位	说明
23-20	RESERVED	R/W	0h	软件不应依赖保留位的值。 写入该寄存器时，必须向该位字段写入 0x5，否则写操作将被忽略，并且不会向寄存器写入任何位。
19	S_LOCKUP	R/W	0h	如果内核正在运行（未暂停）并且存在锁定条件，则读为 1。 写入该寄存器时，必须向该位字段写入 1，否则写操作将被忽略，并且不会向寄存器写入任何位。
18	S_SLEEP	R/W	0h	表示内核正处于睡眠状态（WFI、WFE 或 **SLEEP-ON-EXIT**）。必须使用 C_HALT 来获得控制或等待中断唤醒。 写入该寄存器时，必须向该位字段写入 1，否则写操作将被忽略，并且不会向寄存器写入任何位。
17	S_HALT	R/W	0h	设置该位时，内核处于调试状态。 写入该寄存器时，必须向该位字段写入 1，否则写操作将被忽略，并且不会向寄存器写入任何位。
16	S_REGRDY	R/W	X	调试内核寄存器选择器寄存器上的寄存器读/写操作可用。上次传输已完成。 写入该寄存器时，必须向该位字段写入 1，否则写操作将被忽略，并且不会向寄存器写入任何位。
15-6	RESERVED	R	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
5	C_SNAPSTALL	R/W	0h	如果内核在加载/存储操作时停滞，则停滞结束，并且指令会被强制完成。这会使能暂停调试，以获得对内核的控制。只能在以下情况下设置：C_DEBUGEN = 1 且 C_HALT = 1。内核将 S_RETIRE_ST 读为 0。这表示没有指令推进，可以防止误用。使用该位时，总线状态不可预测。S_RETIRE_ST 可检测加载/存储操作中的内核停滞。
4	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
3	C_MASKINTS	R/W	0h	在逐步执行时或暂停调试模式下屏蔽中断。该屏蔽不会影响由指令执行引起的 NMI、故障异常和 SVC。仅当处理器处于暂停状态 (S_HALT = 1) 时，才能修改该位。在解除暂停之前，必须设置或清除 C_MASKINTS（即，设置或清除 C_MASKINTS 以及设置或清除 C_HALT 的写操作必须分开进行）。在系统运行且启用了暂停调试支持（C_DEBUGEN = 1、S_HALT = 0）时修改 C_MASKINTS 可能会导致不可预测的行为。
2	C_STEP	R/W	0h	在暂停调试模式下逐步执行内核。在 C_DEBUGEN = 0 时，该位无影响。只能在处理器处于暂停状态 (S_HALT = 1) 时修改。 在系统运行且启用了暂停调试支持（C_DEBUGEN = 1、S_HALT = 0）时修改 C_STEP 可能会导致不可预测的行为。
1	C_HALT	R/W	0h	暂停内核。当内核暂停时，会自动设置该位。例如断点。该位会在内核复位时清除。
0	C_DEBUGEN	R/W	0h	使能调试。该位只能由 AHB-AP 写入，而不能由内核写入。由内核写入时无法设置或清除，因此会被忽略。写入 C_HALT 以使自身暂停时，内核必须向该位写入 1。 C_DEBUGEN = 0 时，C_HALT、C_STEP 和 C_MASKINTS 的值会被硬件忽略。C_DEBUGEN = 0 时，C_HALT、C_STEP 和 C_MASKINTS 字段的读取值对于软件来说是未知的。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.69 DCRSR 寄存器（偏移 = DF4h）[复位 = 00000000h]

图 2-140 展示了 DCRSR，表 2-176 中对此进行了介绍。

返回到汇总表。

调试内核寄存器选择器
该寄存器的用途是选择要与其进行数据传输的处理器寄存器。该只写寄存器会生成与内核的握手，以便与调试内核寄存器数据寄存器和所选寄存器进行数据传输。在该内核事务完成之前，DHCSR.S_REGRDY 为 0。请注意，对该寄存器进行非字大小写操作会产生不可预测的结果。
请注意，通过这种方式可以完全访问 PSR 寄存器，而在使用 MRS 指令时某些寄存器会读为 0。请注意，所有位都可写入，但在恢复执行时，某些组合会导致故障。

图 2-140. DCRSR 寄存器

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

表 2-176. DCRSR 寄存器字段说明

位	字段	类型	复位	说明
31-17	RESERVED	W	X	软件不应依赖保留位的值。写入 0。
16	REGWNR	W	X	1：写入 0：读取
15-5	RESERVED	W	X	软件不应依赖保留位的值。写入 0。
4-0	REGSEL	W	X	寄存器选择 0x00：R0 0x01：R1 0x02：R2 0x03：R3 0x04：R4 0x05：R5 0x06：R6 0x07：R7 0x08：R8 0x09：R9 0x0A：R10 0x0B：R11 0x0C：R12 0x0D：当前 SP 0x0E：LR 0x0F：DebugReturnAddress 0x10：XPSR/标志、执行状态信息和异常编号 0x11：MSP（主 SP） 0x12：PSP（进程 SP） 0x14： CONTROL<<24 | FAULTMASK<<16 | BASEPRI<<8 | PRIMASK

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.70 DCRDR 寄存器（偏移 = DF8h）[复位 = 00000000h]

图 2-141 展示了 DCRDR，表 2-177 中对此进行了介绍。

返回到汇总表。

调试内核寄存器数据

图 2-141. DCRDR 寄存器

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

表 2-177. DCRDR 寄存器字段说明

位	字段	类型	复位	说明
31-0	DCRDR	R/W	X	该寄存器保存与处理器之间进行寄存器读写操作的数据。这是写入 DCRSR 所选寄存器的数据值。当处理器接收到来自 DCRSR 的请求时，处理器会使用正常的负载存储单元操作读取或写入该寄存器。如果未执行内核寄存器传输，则基于软件的调试监视器可以使用该寄存器在非暂停调试中进行通信。这使得标志和位能够确认状态，并指示命令是否已被接收、已被回复，或已被接收并回复。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.71 DEMCR 寄存器（偏移 = DFCh）[复位 = 00000000h]

图 2-142 展示了 DEMCR，表 2-178 中对此进行了介绍。

返回到汇总表。

调试异常和监视器控制
该寄存器的用途是矢量捕获和调试监视器控制。该寄存器在调试时管理异常行为。矢量捕获仅可用于暂停调试。高半字用于监视器控制，低半字用于暂停异常支持。该寄存器不会在系统复位时复位，而是通过上电复位操作复位。在内核复位时，始终清除字段 MON_EN、MON_PEND、MON_STEP 和 MON_REQ。调试监视器由软件在复位处理程序中或之后启用，或通过 **AHB-AP** 端口启用。矢量捕获是半同步的。检测到匹配的事件时，会请求暂停。由于处理器只能在指令边界处暂停，因此必须等到下一个指令边界。因此，它会在异常处理程序的第一条指令处停止。但是，触发矢量捕获后，存在两种特殊情况：1.如果在矢量读取或堆栈推送错误期间出现故障，则暂停发生在矢量错误或栈推送的相应故障处理程序上。2.如果在矢量读取或堆栈推送错误期间检测到延迟到达的中断，则不会响应该中断。也就是说，在这种情况下，支持延迟到达优化的实现方案必须抑制该中断。

图 2-142. DEMCR 寄存器

31	30	29	28	27	26	25	24
RESERVED							TRCENA
R/W-0h							R/W-0h
23	22	21	20	19	18	17	16
RESERVED				MON_REQ	MON_STEP	MON_PEND	MON_EN
R/W-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED					VC_HARDERR	VC_INTERR	VC_BUSERR
R/W-0h					R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
VC_STATERR	VC_CHKERR	VC_NOCPERR	VC_MMERR	RESERVED			VC_CORERESET
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h			R/W-0h

表 2-178. DEMCR 寄存器字段说明

位	字段	类型	复位	说明
31-25	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
24	TRCENA	R/W	0h	该位必须设置为 1 才能启用跟踪和调试块：DWT、ITM、ETM 和 TPIU。除非需要跟踪，否则该位使能对电源使用情况的控制。应用中可以启用该位，用于 ITM 或由调试器使用。
23-20	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
19	MON_REQ	R/W	0h	该位使能监视器，以识别唤醒方式。内核复位时会清除该位。 0x0：由调试异常唤醒。 0x1：由 MON_PEND 唤醒
18	MON_STEP	R/W	0h	MON_EN = 1 时，逐步执行内核。MON_EN = 0 时，忽略该位。该位与 DHCSR.C_STEP 等效。中断仅根据监视器的优先级以及 PRIMASK、FAULTMASK 或 BASEPRI 的设置来逐步执行。
17	MON_PEND	R/W	0h	挂起监视器，待优先级允许时再激活。可以通过 AHB-AP 端口唤醒监视器。该位等效于用于监视器调试的 DHCSR.C_HALT。该寄存器不会在系统复位时复位，而只能通过上电复位操作复位。调试监视器必须由软件在复位处理程序中或之后启用，或由 DAP 启用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 2-178. DEMCR 寄存器字段说明（续）

位	字段	类型	复位	说明
16	MON_EN	R/W	0h	使能调试监视器。 启用后，系统处理程序优先级寄存器控制其优先级。如果禁用，则所有调试事件都进入硬故障状态。DHCSR.C_DEBUGEN 覆写该位。矢量捕获是半同步的。检测到匹配的事件时，会请求暂停。由于处理器只能在指令边界处暂停，因此必须等到下一个指令边界。因此，它会在异常处理程序的第一条指令处停止。但是，触发矢量捕获后，存在两种特殊情况：1.如果在矢量查找、矢量读取或堆栈推送错误期间发生故障，则会在相应的故障处理程序上针对矢量错误或堆栈推送发生暂停。2.如果在矢量查找过程中出现延迟到达的中断，则不会响应该中断。也就是说，在这种情况下，支持延迟到达优化的实现方案必须抑制该中断。
15-11	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
10	VC_HARDERR	R/W	0h	硬故障时的调试陷阱。DHCSR.C_DEBUGEN 被清除时忽略。
9	VC_INTERR	R/W	0h	异常进入或返回序列期间发生故障时的调试陷阱。 DHCSR.C_DEBUGEN 被清除时忽略。
8	VC_BUSERR	R/W	0h	正常总线错误时的调试陷阱。DHCSR.C_DEBUGEN 被清除时忽略。
7	VC_STATERR	R/W	0h	使用故障状态错误时的调试陷阱。DHCSR.C_DEBUGEN 被清除时忽略。
6	VC_CHKERR	R/W	0h	使用故障启用检查错误时的调试陷阱。DHCSR.C_DEBUGEN 被清除时忽略。
5	VC_NOCPERR	R/W	0h	对协处理器进行使用故障访问时的调试陷阱。DHCSR.C_DEBUGEN 被清除时忽略。
4	VC_MMERR	R/W	0h	存储器管理故障时的调试陷阱。DHCSR.C_DEBUGEN 被清除时忽略。
3-1	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
0	VC_CORERESET	R/W	0h	重置矢量捕捉。如果发生内核复位，则暂停正在运行的系统。 DHCSR.C_DEBUGEN 被清除时忽略。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.72 STIR 寄存器（偏移 = F00h）[复位 = 00000000h]

图 2-143 展示了 STIR，表 2-179 中对此进行了介绍。

返回到汇总表。

软件触发中断

图 2-143. STIR 寄存器

31	30	29	28	27	26	25	24	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																							INTID								
W-0h																							W-X								

表 2-179. STIR 寄存器字段说明

位	字段	类型	复位	说明
31-9	RESERVED	W	0h	软件不应依赖保留位的值。写入 0。
8-0	INTID	W	X	中断 ID 字段。向该位字段写入值，等同于通过在 NVIC_ISPR0 或 NVIC_ISPR1 中的中断设置挂起寄存器中设置相应的中断位来手动挂起中断。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.73 FPCCR 寄存器（偏移 = F34h）[复位 = C0000000h]

图 2-144 展示了 FPCCR，表 2-180 中对此进行了介绍。

返回到汇总表。

浮点上下文控制
该寄存器保存用于浮点单元的控制数据。只能由特权软件访问。

图 2-144. FPCCR 寄存器

31	30	29	28	27	26	25	24
ASPEN	LSPEN	RESERVED					
R/W-1h	R/W-1h	R/W-0h					
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							MONRDY
R/W-0h							R/W-0h
7	6	5	4	3	2	1	0
RESERVED	BFRDY	MMRDY	HFRDY	THREAD	RESERVED	USER	LSPACT
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 2-180. FPCCR 寄存器字段说明

位	字段	类型	复位	说明
31	ASPEN	R/W	1h	自动状态保留使能。 设置该位将导致在执行浮点指令时设置特殊控制寄存器 (FPCA) 的位 [2]，从而导致在异常进入时自动保留浮点状态。
30	LSPEN	R/W	1h	惰性状态保留使能。 惰性状态保留是指处理器执行上下文保存时，保留堆栈上的空间用于浮点状态，但直到新的上下文执行浮点运算后才会压栈。 0：禁用浮点上下文的自动惰性状态保留。 1：启用浮点上下文的自动惰性状态保留。
29-9	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
8	MONRDY	R/W	0h	指示在处理器分配了浮点栈帧时，执行的软件是否能够将调试监视器异常设置为挂起。 0：分配了浮点栈帧时，调试监视器禁用或优先级不允许设置 DEMCR.MON_PEND。 1：分配了浮点栈帧时，调试监视器启用且优先级允许设置 DEMCR.MON_PEND。
7	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
6	BFRDY	R/W	0h	指示在处理器分配了浮点栈帧时，执行的软件是否能够将总线故障异常设置为挂起。 0：分配了浮点栈帧时，总线故障禁用或优先级不允许将总线故障处理程序设置为挂起状态。 1：分配了浮点栈帧时，总线故障启用且优先级允许将总线故障处理程序设置为挂起状态。
5	MMRDY	R/W	0h	指示在处理器分配了浮点栈帧时，执行的软件是否能够将存储器管理异常设置为挂起。 0：分配了浮点栈帧时，存储器管理禁用或优先级不允许将存储器管理处理程序设置为挂起状态。 1: 分配了浮点栈帧时，存储器管理启用且优先级允许将存储器管理处理程序设置为挂起状态。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 2-180. FPCCR 寄存器字段说明（续）

位	字段	类型	复位	说明
4	HFRDY	R/W	0h	指示在处理器分配了浮点栈帧时，执行的软件是否能够将硬故障异常设置为挂起。 0：分配了浮点栈帧时，优先级不允许将硬故障处理程序设置为挂起状态。 1: 分配了浮点栈帧时，优先级允许将硬故障处理程序设置为挂起状态。
3	THREAD	R/W	0h	指示分配了浮点栈帧时的处理器模式为线程模式。 0：分配了浮点栈帧时，模式不是线程模式。 1：分配了浮点栈帧时，模式是线程模式。
2	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
1	USER	R/W	0h	指示在处理器分配了浮点栈帧时，执行软件的权限级别为用户（非特权）： 0：分配了浮点栈帧时，权限级别不是用户。 1：分配了浮点栈帧时，权限级别是用户。
0	LSPACT	R/W	0h	指示浮点的惰性状态保留是否处于活动状态： 0：惰性状态保留处于非活动状态。 1：惰性状态保留处于活动状态。浮点栈帧已分配，但其保存状态已延迟。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.74 FPCAR 寄存器（偏移 = F38h）[复位 = 00000000h]

图 2-145 展示了 FPCAR，表 2-181 中对此进行了介绍。

返回到汇总表。

浮点上下文地址
该寄存器保存在异常栈帧上分配的未填充浮点寄存器空间的位置。

图 2-145. FPCAR 寄存器

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

表 2-181. FPCAR 寄存器字段说明

位	字段	类型	复位	说明
31-2	地址	R/W	0h	保存在异常栈帧上分配的未填充浮点寄存器空间的（双字对齐）位置。
1-0	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.75 FPDSCR 寄存器（偏移 = F3Ch）[复位 = 00000000h]

图 2-146 展示了 FPDSCR，表 2-182 中对此进行了介绍。

返回到汇总表。

浮点默认状态控制
该寄存器保存浮点状态控制数据的默认值，该数据是处理器在创建新的浮点上下文时分配给 FPSCR 的。只能由特权软件访问。

图 2-146. FPDSCR 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					AHP	DN	FZ	RMODE		RESERVED					
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
RESERVED															
R/W-0h															

表 2-182. FPDSCR 寄存器字段说明

位	字段	类型	复位	说明
31-27	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
26	AHP	R/W	0h	备用半精度位的默认值。（如果该位设置为 1，则选择备用半精度格式）。
25	DN	R/W	0h	默认 NaN 模式位的默认值。（如果该位设置为 1，则任何涉及一个或多个 NaN 的操作都会返回默认 NaN）。
24	FZ	R/W	0h	刷新到零模式位的默认值。（如果该位设置为 1，则刷新到零模式启用）。
23-22	RMODE	R/W	0h	舍入模式控制字段的默认值。（该字段的编码为： 0b00 四舍五入 (RN) 模式 0b01 舍入到正无穷 (RP) 模式 0b10 舍入到负无穷 (RM) 模式 0b11 舍入到零 (RZ) 模式。 几乎所有浮点指令都会使用指定的舍入模式）。
21-0	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.76 MVFR0 寄存器（偏移 = F40h）[复位 = 10110021h]

图 2-147 展示了 MVFR0，表 2-183 对其进行了介绍。

返回到汇总表。

媒体和浮点功能 0
描述了浮点扩展提供的功能。

图 2-147. MVFR0 寄存器

31	30	29	28	27	26	25	24
FP_ROUNDING_MODES				SHORT_VECTORS			
R-1h				R-0h			
23	22	21	20	19	18	17	16
SQUARE_ROOT				分频值			
R-1h				R-1h			
15	14	13	12	11	10	9	8
FP_EXCEPTION_TRAPPING				DOUBLE_PRECISION			
R-0h				R-0h			
7	6	5	4	3	2	1	0
SINGLE_PRECISION				A_SIMD			
R-2h				R-1h			

表 2-183. MVFR0 寄存器字段说明

位	字段	类型	复位	说明
31-28	FP_ROUNDING_MODES	R	1h	指示 FP 浮点硬件支持的舍入模式。该字段的值为：0b0001 - 支持所有舍入模式。
27-24	SHORT_VECTORS	R	0h	指示对 FP 短矢量的硬件支持。该字段的值为：0b0000 - 不支持。
23-20	SQUARE_ROOT	R	1h	指示对 FP 平方根运算的硬件支持。该字段的值为：0b0001 - 支持。
19-16	分频值	R	1h	指示对 FP 除法运算的硬件支持。该字段的值为：0b0001 - 支持。
15-12	FP_EXCEPTION_TRAPPING	R	0h	指示 FP 硬件实现是否支持异常触发。该字段的值为：0b0000 - 不支持。
11-8	DOUBLE_PRECISION	R	0h	指示对 FP 双精度运算的硬件支持。该字段的值为：0b0000 - 不支持。
7-4	SINGLE_PRECISION	R	2h	指示对 FP 单精度运算的硬件支持。该字段的值为：0b0010 - 支持。
3-0	A_SIMD	R	1h	指示 FP 寄存器组的大小。该字段的值为：0b0001 - 支持，16 x 64 位寄存器。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.4.77 MVFR1 寄存器（偏移 = F44h）[复位 = 11000011h]

图 2-148 展示了 MVFR1，表 2-184 对其进行了介绍。

返回到汇总表。

媒体和浮点功能 1
描述了浮点扩展提供的功能。

图 2-148. MVFR1 寄存器

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

表 2-184. MVFR1 寄存器字段说明

位	字段	类型	复位	说明
31-28	FP_FUSED_MAC	R	1h	指示 FP 是否支持融合乘加运算。该字段的值为：0b0001 - 支持。
27-24	FP_HPFP	R	1h	指示 FP 是否支持半精度浮点转换运算。该字段的值为：0b0001 - 支持。
23-8	RESERVED	R	0h	软件不应依赖保留位的值。
7-4	D_NAN_MODE	R	1h	指示 FP 硬件实现是否仅支持默认 NaN 模式。该字段的值为： 0b0001 - 硬件支持 NaN 值的传播。
3-0	FTZ_MODE	R	1h	指示 FP 硬件实现是否仅支持刷新到零模式运算。该字段的值为： 0b0001 - 硬件支持完整的去规范化数运算。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.5 CPU_TPIU 寄存器

表 2-185 列出了 CPU_TPIU 寄存器的存储器映射寄存器。表 2-185 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

表 2-185. CPU_TPIU 寄存器

偏移	首字母缩写词	寄存器名称	部分
0h	SSPSR	支持的同步端口大小	SSPSR 寄存器（偏移 = 0h）[复位 = 0000000Bh]
4h	CSPSR	当前同步端口大小	CSPSR 寄存器（偏移 = 4h）[复位 = 00000001h]
10h	ACPR	异步时钟预分频器	ACPR 寄存器（偏移 = 10h）[复位 = 00000000h]
F0h	SPPR	所选引脚协议	SPPR 寄存器（偏移 = F0h）[复位 = 00000001h]
300h	FFSR	格式化程序和刷新状态	FFSR 寄存器（偏移 = 300h）[复位 = 00000008h]
304h	FFCR	格式化程序和冲洗控制	FFCR 寄存器（偏移 = 304h）[复位 = 00000102h]
308h	FSCR	格式化程序同步计数器	FSCR 寄存器（偏移 = 308h）[复位 = 00000000h]
FA0h	CLAIMMASK	声明标签掩码	CLAIMMASK 寄存器（偏移 =FA0h）[复位 = 0000000Fh]
FA0h	CLAIMSET	声明标签集	CLAIMSET 寄存器（偏移 =FA0h）[复位 = 0000000Fh]
FA4h	CLAIMTAG	电流声明标签	CLAIMTAG 寄存器（偏移 =FA4h）[复位 = 00000000h]
FA4h	CLAIMCLR	声明标签清除	CLAIMCLR 寄存器（偏移 =FA4h）[复位 = 00000000h]
FC8h	DEVID	器件 ID	DEVID 寄存器（偏移 = FC8h）[复位 = 00000CA0h]

复杂的位访问类型经过编码可适应小型表单元。表 2-186 展示了适用于此部分中访问类型的代码。

表 2-186. CPU_TPIU 访问类型代码

访问类型	代码	说明
读取类型		
R	R	读取
写入类型		
W	W	写入
复位或默认值		
-n		复位后的值或默认值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.5.1 SSPSR 寄存器（偏移 = 0h）[复位 = 0000000Bh]

图 2-149 展示了 SSPSR，表 2-187 中对此进行了介绍。

返回到汇总表。

支持的同步端口大小
该寄存器表示器件支持的单端口大小，即 4、2 或 1。这是为了保证工具不会尝试选择连接的 TPA 无法捕获的端口宽度。

图 2-149. SSPSR 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				四	三	二	一
R-0h				R-1h	R-0h	R-1h	R-1h

表 2-187. SSPSR 寄存器字段说明

位	字段	类型	复位	说明
31-4	RESERVED	R	0h	软件不应依赖保留的值。写入除复位值之外的任何其他值都可能导致未定义行为。
3	四	R	1h	4 位端口大小支持 0x0：不支持 0x1：支持
2	三	R	0h	3 位端口大小支持 0x0：不支持 0x1：支持
1	二	R	1h	2 位端口大小支持 0x0：不支持 0x1：支持
0	一	R	1h	1 位端口大小支持 0x0：不支持 0x1：支持

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.5.2 CSPSR 寄存器（偏移 = 4h）[复位 = 00000001h]

图 2-150 展示了 CSPSR，表 2-188 中对此进行了介绍。

返回到汇总表。

当前同步端口大小
该寄存器与 SSPSR 的格式相同，但只能设置一个位，所有其他位都必须为零。写入设置了多个位的值，或设置未指示为支持的位可能会导致不可预测的行为。复位时，此默认值为可能的最小端口大小，1 位。

图 2-150. CSPSR 寄存器

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				四	三	二	一
R/W-0h				R/W-0h	R/W-0h	R/W-0h	R/W-1h

表 2-188. CSPSR 寄存器字段说明

位	字段	类型	复位	说明
31-4	RESERVED	R/W	0h	软件不应依赖保留的值。写入除复位值之外的任何其他值都可能导致未定义行为。
3	四	R/W	0h	4 位端口支持 写入在 CSPSR 中设置了多个位的值，或设置未在 SSPSR 中指示为支持的位，可能会导致不可预测的行为。
2	三	R/W	0h	3 位端口支持 写入在 CSPSR 中设置了多个位的值，或设置未在 SSPSR 中指示为支持的位，可能会导致不可预测的行为。
1	二	R/W	0h	2 位端口支持 写入在 CSPSR 中设置了多个位的值，或设置未在 SSPSR 中指示为支持的位，可能会导致不可预测的行为。
0	一	R/W	1h	1 位端口支持 写入在 CSPSR 中设置了多个位的值，或设置未在 SSPSR 中指示为支持的位，可能会导致不可预测的行为。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.5.3 ACPR 寄存器（偏移 = 10h）[复位 = 00000000h]

图 2-151 展示了 ACPR，表 2-189 中对此进行了介绍。

返回到汇总表。

异步时钟预分频器
该寄存器调节异步输出的波特率。

图 2-151. ACPR 寄存器

31	30	29	28	27	26	25	24	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																			预分频器												
R/W-0h																			R/W-0h											

表 2-189. ACPR 寄存器字段说明

位	字段	类型	复位	说明
31-13	RESERVED	R/W	0h	软件不应依赖保留的值。写入除复位值之外的任何其他值都可能导致未定义行为。
12-0	预分频器	R/W	0h	输入跟踪时钟的分频器为 (PRESCALER + 1)。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.5.4 SPPR 寄存器（偏移 = F0h）[复位 = 00000001h]

图 2-152 展示了 SPPR，表 2-190 中对此进行了介绍。

返回到汇总表。

选定引脚协议
该寄存器选择要用于跟踪输出的协议。
注意：如果在输出跟踪数据时更改该寄存器，则会发生数据损坏。

图 2-152. SPPR 寄存器

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						协议	
R/W-0h						R/W-1h	

表 2-190. SPPR 寄存器字段说明

位	字段	类型	复位	说明
31-2	RESERVED	R/W	0h	软件不应依赖保留的值。写入除复位值之外的任何其他值都可能导致未定义行为。
1-0	协议	R/W	1h	跟踪输出协议 0h = TracePort 模式 1h = SerialWire 输出 (Manchester)。这是复位值。 2h = SerialWire 输出 (NRZ)

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.5.5 FFSR 寄存器（偏移 = 300h）[复位 = 00000008h]

图 2-153 展示了 FFSR，表 2-191 中对此进行了介绍。

返回到汇总表。

格式化程序和刷新状态

图 2-153. FFSR 寄存器

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

表 2-191. FFSR 寄存器字段说明

位	字段	类型	复位	说明
31-4	RESERVED	R	0h	软件不应依赖保留的值。写入除复位值之外的任何其他值都可能导致未定义行为。
3	FTNONSTOP	R	1h	0：可停止格式化程序 1：无法停止格式化程序
2-0	RESERVED	R	0h	该字段始终读取为零

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.5.6 FFCR 寄存器（偏移 = 304h）[复位 = 00000102h]

图 2-154 展示了 FFCR，表 2-192 中对此进行了介绍。

返回到汇总表。

格式化程序和刷新控制
当选择两种单线输出 (SWO) 模式之一时，ENFCONT 将绕过格式化程序。如果格式化程序被旁路，则只有 ITM/DWT 跟踪源 (ATDATA2) 会通过。TPIU 接受并且丢弃 ETM 端口 (ATDATA1) 上显示的数据。当需要将包含 ETM 的器件连接到只能捕获串行线输出 (SWO) 数据的跟踪捕获器件时，应使用此功能。启用或禁用格式化程序会导致短暂的数据损坏。
注意：如果所选引脚协议寄存器 (SPPR.PROTOCOL) 设置为 0x00（TracePort 模式），则该寄存器始终读取 0x102，因为格式化程序会自动启用。如果选择了其中一种串行线模式,则寄存器会恢复到其之前编程的值。

图 2-154. FFCR 寄存器

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							TRIGIN
R/W-0h							R/W-1h
7	6	5	4	3	2	1	0
RESERVED						ENFCONT	RESERVED
R/W-0h						R/W-1h	R/W-0h

表 2-192. FFCR 寄存器字段说明

位	字段	类型	复位	说明
31-9	RESERVED	R/W	0h	软件不应依赖保留的值。写入除复位值之外的任何其他值都可能导致未定义行为。
8	TRIGIN	R/W	1h	表示在触发引脚置为有效时插入了触发器。
7-2	RESERVED	R/W	0h	软件不应依赖保留的值。写入除复位值之外的任何其他值都可能导致未定义行为。
1	ENFCONT	R/W	1h	启用连续格式化： 0：禁用连续格式化 1：连续格式化已启用
0	RESERVED	R/W	0h	软件不应依赖保留的值。写入除复位值之外的任何其他值都可能导致未定义行为。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.5.7 FSCR 寄存器（偏移 = 308h）[复位 = 00000000h]

图 2-155 展示了 FSCR，表 2-193 中对此进行了介绍。

返回到汇总表。

格式化程序同步计数器

图 2-155. FSCR 寄存器

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

表 2-193. FSCR 寄存器字段说明

位	字段	类型	复位	说明
31-0	FSCR	R	0h	全局同步触发由程序计数器 (PC) 采样器块生成。这意味着 TPIU 中没有同步计数器。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.5.8 CLAIMMASK 寄存器（偏移 =FA0h）[复位 = 0000000Fh]

图 2-156 展示了 CLAIMMASK，表 2-194 中对此进行了介绍。

返回到汇总表。

声明标签掩码

图 2-156. CLAIMMASK 寄存器

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

表 2-194. CLAIMMASK 寄存器字段说明

位	字段	类型	复位	说明
31-0	CLAIMMASK	R	Fh	此寄存器构成声明标签值的一半。读取该寄存器时会返回可设置的位数（单独考虑每个位）： 0：未实现此声明标签位 1：实现此声明标签位 CLAIMSET 中描述了写入此寄存器时的行为。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.5.9 CLAIMSET 寄存器（偏移 =FA0h）[复位 = 0000000Fh]

图 2-157 展示了 CLAIMSET，表 2-195 中对此进行了介绍。

返回到汇总表。

声明标签集

图 2-157. CLAIMSET 寄存器

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

表 2-195. CLAIMSET 寄存器字段说明

位	字段	类型	复位	说明
31-0	CLAIMSET	W	Fh	此寄存器构成声明标签值的一半。写入此位置允许设置单独的位（会单独考虑每个位）： 0：无效 1：在声明标签中设置此位 从该位置读取时的行为如 CLAIMMASK 中所述。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.5.10 CLAIMTAG 寄存器（偏移 =FA4h）[复位 = 00000000h]

图 2-158 展示了 CLAIMTAG，表 2-196 中对此进行了介绍。

返回到汇总表。

电流声明标签

图 2-158. CLAIMTAG 寄存器

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

表 2-196. CLAIMTAG 寄存器字段说明

位	字段	类型	复位	说明
31-0	CLAIMTAG	R	0h	此寄存器构成声明标签值的一半。读取此寄存器将返回当前声明标签值。 读取 CLAIMMASK 将确定必须使用的该寄存器中的位数。 CLAIMCLR 中说明了写入该寄存器时的行为。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.5.11 CLAIMCLR 寄存器（偏移 =FA4h）[复位 = 00000000h]

图 2-159 展示了 CLAIMCLR，表 2-197 中对此进行了介绍。

返回到汇总表。

声明标签清除

图 2-159. CLAIMCLR 寄存器

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

表 2-197. CLAIMCLR 寄存器字段说明

位	字段	类型	复位	说明
31-0	CLAIMCLR	W	0h	此寄存器构成声明标签值的一半。写入此位置可以清除各个位（单独考虑每个位）： 0：无效 1：清除声明标签中的此位。 在 CLAIMTAG 中说明了从该位置读取时的行为。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2.9.5.12 DEVID 寄存器（偏移 = FC8h）[复位 = 00000CA0h]

图 2-160 展示了 DEVID，表 2-198 中对此进行了介绍。

返回到汇总表。

器件 ID

图 2-160. DEVID 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEVID																															
R-CA0h																															

表 2-198. DEVID 寄存器字段说明

位	字段	类型	复位	说明
31-0	DEVID	R	CA0h	该字段返回：如果存在 ETM，则为 0xCA1。如果不存在 ETM，则为 0xCA0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

3.1 存储器映射

表 3-1. 存储器映射

模块	模块名称	基址
FLASHMEM	程序闪存存储器	0x0000 0000
SRAM	静态 RAM	0x2000 0000
RFC_RAM	射频内核无线电 RAM	0x2100 0000
RFC_ULLRAM	射频内核命令和数据包引擎 RAM	0x2100 4000
SSI0	同步串行接口 (SSI) 0	0x4000 0000
UART0	通用异步接收器/传输器 (UART) 0	0x4000 1000
I2C0	内部集成电路 (I^2C) 0	0x4000 2000
SSI1	同步串行接口 (SSI) 1	0x4000 8000
UART1	通用异步接收器/传输器 (UART) 1	0x4000 B000
GPT0	通用计时器 0	0x4001 0000
GPT1	通用计时器 1	0x4001 1000
GPT2	通用计时器 2	0x4001 2000
GPT3	通用计时器 3	0x4001 3000
UDMA0	微型直接存储器存取 (µDMA)	0x4002 0000
I2S0	IC 间音频 (I^2S) 0	0x4002 1000
GPIO	通用 I/O	0x4002 2000
CRYPTO	加密引擎	0x4002 4000
PKA	公钥加速	0x4002 5000
PKA_RAM	公钥加速 RAM	0x4002 6000
PKA_INT	公钥加速选项	0x4002 7000
TRNG	真随机数生成器	0x4002 8000
闪存	闪存子系统控制器	0x4003 0000
VIMS	多功能指令存储器系统	0x4003 4000
SRAM_MMR	SRAM 寄存器	0x4003 5000
RFC_PWR	射频内核无线电功率控制器	0x4004 0000
RFC_DBELL	射频内核无线电门铃	0x4004 1000
RFC_RAT	射频内核无线电计时器	0x4004 3000
WDT	看门狗计时器	0x4008 0000
IOC	I/O 控制器	0x4008 1000
PRCM	电源、时钟和复位管理	0x4008 2000
事件	事件结构	0x4008 3000
SMPH	系统 CPU 信标	0x4008 4000
AON_PMCTL	AON 电源管理控制器	0x4009 0000

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 3-1. 存储器映射（续）

模块	模块名称	基址
AON_RTC	AON 实时时钟	0x4009 2000
AON_EVENT	AON 事件结构	0x4009 3000
AON_IOC	AON I/O 控制器	0x4009 4000
AON_BATMON	AON 电池监测器和温度传感器	0x4009 5000
AUX_SPIM	AUX 域 SPI 主器件	0x400C 1000
AUX_MAC	传感器控制器数学加速器	0x400C 2000
AUX_TIMER2	AUX 域计时器 2	0x400C 3000
AUX_TDC	AUX 域时数转换器	0x400C 4000
AUX_EVCTL	AUX 域事件控制器	0x400C 5000
AUX_SYSIF	AUX 域系统接口	0x400C 6000
AUX_TIMER01	AUX 域计时器 1	0x400C 7000
AUX_SMPH	AUX 域信标	0x400C 8000
AUX_ANAIF	AUX 域模拟接口	0x400C 9000
AUX_DDI0_OSC	AUX 域时钟控制	0x400C A000
AUX_ADI4	AUX 域模拟数字接口主器件	0x400C B000
AUX_AIODIO0	AUX 域模拟数字 I/O 0	0x400C C000
AUX_AIODIO1	AUX 域模拟数字 I/O 1	0x400C D000
AUX_AIODIO2	AUX 域模拟数字 I/O 2	0x400C E000
AUX_AIODIO3	AUX 域模拟数字 I/O 3	0x400C F000
AUX_RAM	传感器控制器 AUX RAM	0x400E 0000
AUX_SCE	传感器控制器引擎	0x400E 1000
FCFG1	出厂配置	0x5000 1000
CCFG	自定义配置	0x5000 3000
CPU_ITM	内核仪表跟踪宏单元	0xE000 0000
CPU_DWT	内核数据观察点和跟踪	0xE000 1000
CPU_FPB	内核闪存补丁和断点	0xE000 2000
CPU_SCS	内核系统控制空间	0xE000 E000
CPU_TPIU	内核跟踪端口接口单元	0xE004 0000

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

章节 4

Arm® Cortex®-M4F 外设

本章介绍了 Arm® Cortex®-M4F 外设。

4.1 Arm® Cortex®-M4F 外设介绍 313
4.2 功能说明 313

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

4.1 Arm® Cortex®-M4F 外设介绍

本章介绍 CC13x2 和 CC26x2 器件平台的 Arm® Cortex®-M4F 处理器外设实现方案，包括：

- 系统计时器 (SysTick)（请参阅节 4.2.1）：提供一个简单的 24 位写时清零、递减、归零回绕计数器，并具有灵活的控制机制。
- 嵌套向量中断控制器 (NVIC)（请参阅节 4.2.2）：
  - 有助于实现低延迟的异常和中断处理
  - 与系统控制器配合使用（请参阅章节 7）以控制电源管理
  - 实现系统控制寄存器
- Arm® Cortex®-M4F 系统控制块 (SCB)（请参阅节 4.2.3）：提供系统实现信息和系统控制，包括系统异常的配置、控制和报告。

表 4-1 列出了专用外设总线 (PPB) 的地址映射。某些外设寄存器区域分割成两个地址区域，如所列的两个地址所示。

表 4-1. 内核外设寄存器区域

地址	内核外设	链接
0xE000 E010 至 0xE000 E01C	系统计时器 (SysTick)	请参阅 节 4.2.1
0xE000 E100 至 0xE000 E420 0xE000 EF00 至 0xE000 EF00	嵌套向量中断控制器 (NVIC)	请参阅 节 4.2.2
0xE000 E008 至 0xE000 E00F 0xE000 ED00 至 0xE000 ED3F	系统控制块 (SCB)	请参阅 节 4.2.3
0xE000 1000 至 0xE000 1FFC	数据观察点和跟踪 (DWT)	请参阅 节 4.2.7
0xE000 2000 至 0xE000 2FFC	闪存补丁和断点 (FPB)	请参阅 节 4.2.5
0xE000 0000 至 0xE000 0FFC	仪表跟踪宏单元 (ITM)	请参阅 节 4.2.4
0xE00F F000 至 0xE00F FFFC	ROM 表	
0xE004 0000 至 0xE004 0FFC	跟踪端口接口单元 (TPIU)	请参阅 节 4.2.6
0xE00F EFF8 至 0xE00F EFFC	TIPROP	

4.2 功能说明

本章提供了有关 Arm® Cortex®-M4F 处理器外设的 CC13x2 和 CC26x2 器件平台实现的信息：

- 系统计时器 (SysTick)
- 嵌套矢量中断控制器 (NVIC)
- 系统控制块 (SCB)
- 数据观察点和跟踪 (DWT)
- 闪存补丁和断点 (FPB)
- 仪表跟踪宏单元 (ITM)
- 跟踪端口接口单元 (TPIU)

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

4.2.1 SysTick

Arm® Cortex®-M4F 处理器包括一个集成系统计时器 SysTick，该计时器可提供一个具有灵活控制机制的简单 24 位写入清除，递减归零绕回计数器。该计数器可采用数种使用方式。例如，该计数器可以是：

- RTOS 节拍计时器，它以可编程的速率（例如 100Hz）触发并调用 SysTick 例程
- 使用系统时钟的高速警报计时器
- 可变速率警报或信号计时器 - 持续时间取决于所使用的基准时钟和计数器的动态范围
- 一个简单的计数器，用于测量完成时间和使用的时间
- 基于缺失和/或会议持续时间的内部时钟源控制 — 控制和状态寄存器 (STCSR) COUNTFLAG 位可用于确定操作是否在设定的持续时间内完成（作为动态时钟管理控制环路的一部分）

计时器包含三个寄存器：

- SysTick 控制及状态寄存器 (STCSR)：控制和状态计数器，用于配置时钟、启用计数器、启用 SysTick 中断以及确定计数器状态（请参阅节 2.9.4）
- SysTick 重加载值寄存器 (STRVR)：计数器的重新加载值，用于提供计数器的绕回值（请参阅节 2.9.4）
- SysTick 电流值寄存器 (STCVR)：计数器的当前值（请参阅节 2.9.4）

启用后，计时器在每个时钟上从重载值递减计数到 0，在下一个时钟沿重新加载（绕回）到 STRVR 寄存器中的值，然后在后续时钟上递减。清除 STRVR 寄存器会在下一次绕回时禁用计数器。当计数器达到 0 时，设置 COUNTFLAG 状态位。COUNTFLAG 位在读取时清零。

写入 STCVR 寄存器会清除该寄存器及 COUNTFLAG 状态位。写入不会触发 SysTick 异常逻辑。读取时，当前值是访问寄存器时的寄存器值。

SysTick 计数器在系统时钟上运行。如果该时钟信号在低功耗模式下停止，SysTick 计数器会停止。确保软件使用对齐字访问来访问 SysTick 寄存器。

备注

当暂停处理器以进行调试时，计数器不会递减。

4.2.2 NVIC

本节介绍 NVIC 及其使用的寄存器。NVIC 支持：

- 34 条中断线路
- 每个中断均可编程优先级，取值范围 0 至 7。优先级数字越大则其优先级越低，也就是说 0 代表最高优先级。
- 低延迟异常和中断处理
- 中断信号的电平和脉冲检测
- 动态重设中断优先级
- 优先级值可分组为组优先级域和子优先级字段
- 咬尾中断
- 外部不可屏蔽中断 (NMI)

处理器在进入异常时自动将其状态压入堆栈，并在退出异常时自动将其弹出堆栈，无需任何指令开销，从而提供低延迟异常处理。

4.2.2.1 电平敏感中断和脉冲中断

处理器支持电平敏感中断和脉冲中断。脉冲中断也被描述为边沿触发中断。电平敏感中断会保持有效状态，直到外设将中断信号置为无效。CC13x2 和 CC26x2 器件平台器件中的中断源通常是电平敏感的。也就是说，它们会保持活动状态，直到外设中的中断源被清除。这通常是因为中断服务例程 (ISR) 访问了该外设，从而使其清除中断请求。为确保 NVIC 能够检测到中断，外设必须将中断信号保持有效至少一个时钟周期。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

处理器进入 ISR 后，将自动清除该中断的挂起状态（请参阅节 4.2.2.2）。对于电平敏感中断，如果在处理器从 ISR 返回之前信号未置为无效，则该中断将再次变为挂起状态，处理器必须再次执行其 ISR。因此，外设可以保持中断信号为有效状态，直到它不再需要服务为止。

4.2.2.2 中断的硬件和软件控制

Arm® Cortex®-M4F 处理器会锁存所有中断。外设中断会由于以下原因之一变为挂起状态：

- NVIC 检测到中断信号被置为有效，且该中断未处于活动状态。
- NVIC 在中断信号上检测到上升沿。
- 软件写入相应的中断设置挂起寄存器位，或写入软件触发中断寄存器 (STIR) 来生成软件中断挂起状态（请参阅节 2.9.4 中的 NVIC_ISPR0 SETPENDn 寄存器位或 STIR INTID 寄存器字段）。

挂起中断一直保持挂起，直到发生以下情况之一：

- 处理器进入中断的 ISR，将中断状态从挂起变为活动状态。然后：
  - 对于电平敏感型中断，当处理器从 ISR 返回时，NVIC 采样中断信号。如果该信号被置为有效，中断状态将变为挂起，这可能导致处理器立即重新进入 ISR。否则，中断变为非活动状态。
  - 对于脉冲中断，NVIC 会持续监测中断信号，如果该信号出现脉冲，中断状态将变为挂起且活动状态。在此情况下，当处理器从 ISR 返回时，中断状态将变为挂起，这可能导致处理器立即重新进入 ISR。如果在处理器处于 ISR 期间中断信号未出现脉冲，当处理器从 ISR 返回时，中断状态将变为非活动。
- 软件写入相应的中断清除挂起寄存器位：
  - 对于电平敏感型中断，如果中断信号仍然置为有效，中断状态不会改变。否则，中断变为非活动状态。
  - 对于脉冲中断，如果状态为挂起，则中断状态变为非活动；如果状态为活动且挂起，则变为活动。

4.2.3 SCB

SCB 提供系统实现信息和系统控制功能，以及系统异常配置、控制和报告。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

4.2.4 ITM

ITM 是一个应用程序的跟踪源，它支持 printf() 样式调试以跟踪操作系统和应用事件，并生成诊断系统信息。ITM 以数据包形式生成跟踪信息。如果多个源同时生成数据包，则 ITM 会对数据包的输出顺序进行仲裁。这些来源按优先级以降序排列如下：

- **软件跟踪：**软件可直接写入 ITM 激励寄存器以生成数据包。
- **硬件跟踪：**DWT 生成这些数据包，并且 ITM 输出这些数据包。
- **时间戳：**生成的时间戳是相对于数据包的。ITM 含有一个 21 位计数器来生成时间戳。Arm® Cortex®-M4F 时钟或串行有线查看器 (SWV) 输出的位时钟速率为计数器计时。

备注

在特权模式下，可以完全访问 ITM 寄存器。在用户模式下，可以读取所有寄存器，但只能写入激励寄存器和跟踪使能寄存器，并且仅在设置了相应的跟踪特权寄存器位时才可写入。ITM 寄存器的无效用户模式写入将被丢弃。

4.2.5 FPB

FPB 实现了硬件断点，并将代码和数据从代码空间修补到系统空间。

完整的 FPB 装置包含：

- 两个字面量比较器，用于与从代码空间加载的字面量进行匹配，并重映射到系统空间中对应的区域。
- 六个指令比较器，用于与从代码空间获取的指令进行匹配，并重新映射到系统空间中的相应区域。或者，可以单独配置比较器，使其在硬件断点功能匹配的情况下将断点 (BKPT) 指令返回到处理器内核。

精简的 FPB 装置包含：

- 两个指令比较器可单独配置，以在匹配时将 BKPT 指令返回到处理器并提供硬件断点功能

FPB 包含八个比较器的全局使能和单独使能端。如果条目的比较匹配，则地址为：

- 重新映射到重新映射寄存器中设置的地址加上和匹配的比较器相对应的偏移量

  或
- 如果启用了该功能，则重新映射到 BKPT 指令

比较是动态发生的，但比较结果发生得太晚，无法停止从代码空间进行原始指令提取或字面加载。但是，处理器会忽略此事务，仅使用重新映射的事务。

如果 FPB 仅支持两个断点，则仅使用比较器 0 和 1，并且 FPB 不支持闪存修补。

4.2.6 TPIU

Arm® Cortex®-M4F TPIU 充当来自嵌入式跟踪宏单元 (ETM) 和具有单独 ID 的仪表跟踪宏单元 (ITM) 的片上跟踪数据与数据流之间的桥梁。TPIU 在需要的地方封装 ID，然后由跟踪端口分析器 (TPA) 捕获数据流。

TPIU 提供了两个配置：

- 支持 ITM 调试跟踪的配置
- 同时支持 ITM 及 ETM 调试跟踪的配置

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

4.2.7 DWT

DWT 为处理器提供观察点、数据跟踪及系统配置。一个完整的 DWT 包含四个比较器，可以配置为以下任何一个：

- 硬件观察点
- ETM 触发器
- PC 采样器事件触发器
- 数据地址采样器事件触发器

第一个比较器 DWT_COMP0 也可与时钟周期计数器 CYCCNT 进行比较。第二个比较器 DWT_COMP1 可用作数据比较器。

降低的 DWT 包含一个比较器，该比较器可用作观察点或触发器。减少的 DWT 不支持数据匹配。

DWT 包含以下项的计数器：

- 时钟周期 (CYCCNT)
- 折叠指令
- 负载存储单元 (LSU) 操作
- 睡眠周期
- CPI (即除第一个周期外的所有指令周期)
- 中断开销

DWT 以定义的间隔生成 PC 样本和中断事件信息。DWT 还可以向 ITM 和 TPIU 提供协议同步的定期请求。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

章节 5

中断和事件

本章介绍 CC13x2 和 CC26x2 器件平台的中断和事件。

5.1 异常模型 319
5.2 故障处理 326
5.3 事件结构 328
5.4 AON 事件结构 330
5.5 MCU 事件结构 331
5.6 AON 事件 336
5.7 中断和事件寄存器 337

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.1 异常模型

Arm® Cortex®-M4F 处理器和嵌套向量中断控制器 (NVIC) 在处理程序模式下对所有异常进行优先级排序和处理。发生异常时，处理器的状态会自动保存到堆栈中，并在中断服务例程 (ISR) 结束时自动从堆栈恢复。向量的获取与状态保存并行进行，从而实现高效的中断进入。处理器支持尾链，能够执行背对背中断，而没有状态保存和恢复开销。

表 5-1 列出了所有异常类型。软件可以为其中七个异常（系统处理程序）以及 CC13x2 和 CC26x2 器件平台的中断（在表 5-7 中列出）设置八个优先级数值。

系统处理程序的优先级通过 NVIC 系统处理程序优先级 n 寄存器（节 2.9.4 中的 CPU_SCS:SHPRn）设置。中断通过 NVIC 中断设置使能 n 寄存器（节 2.9.4 中的 CPU_SCS:NVIC_ISERn）启用，并通过 NVIC 中断优先级 n 寄存器（节 2.9.4 中的 CPU_SCS:NVIC_IPRn）设置优先级。优先级可以通过将优先级数值划分为抢占优先级和子优先级来进行分组。节 4.2.2 介绍了所有中断寄存器。

在内部，最高用户可编程优先级 (0) 被视为第三优先级，仅次于复位和硬件故障（按此顺序）。

备注

所有可编程优先级的默认优先级均为 0。

小心

在写入以清除中断源之后，由于写缓冲的存在，NVIC 可能需要几个处理器周期才能检测到中断源被置为无效。因此，如果在中断处理程序中将中断清除作为最后一项操作，则可能发生中断处理程序完成时 NVIC 仍将中断检测为有效状态，从而导致错误地重新进入中断处理程序。要避免这种情况，可以在中断处理程序开始时清除中断源，或在写入以清除中断源（并刷新写缓冲区）后从同一地址执行读取操作。

有关异常和中断的更多信息，请参阅节 4.2.2。

5.1.1 异常状态

每个异常都处于以下状态之一：

- **非活动：** 异常不活动且未挂起。
- **待定：** 异常正在等待处理器处理。来自外设或软件的中断请求可以将相应中断的状态更改成挂起。
- **激活：** 处理器正在处理一个异常，但该处理尚未完成。异常处理程序可中断另一个异常处理程序的执行。在这种情况下，两个异常都处于活动状态。
- **活动和挂起：** 异常由处理器处理，并且同一来源存在一个挂起异常。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.1.2 异常类型

异常类型如下：

- **复位：**在加电或热复位时调用复位。异常模型将复位视为一种特殊形式的异常。当复位置为有效时，处理器停止工作（可能停止在指令执行的任何位置）。当复位置为无效时，处理器从矢量表中复位条目所指定的地址重新开始执行。在线程模式下，执行以特权级执行方式重新启动。
- **硬故障：**硬故障是由于异常处理期间发生错误或某个异常无法通过任何其他异常机制管理而引发的异常。硬故障的优先级固定为 –1，这意味着它们的优先级高于具有可配置优先级的任何异常。
- **总线故障：**总线故障是由于指令或数据存储器事务的内存相关故障（例如预取故障或内存访问故障）而发生的异常。这种故障可启用或禁用。
- **使用故障：**使用故障是由于与指令执行相关的故障而发生的异常，例如：
  - 未定义的指令
  - 非法的未对齐访问
  - 指令执行中的无效状态
  - 返回异常时出错

  当内核配置正确时，对字或半字内存访问中的未对齐地址或除以零的操作都会触发使用故障。
- **SVCall：**监管调用 (SVC) 是由 SVC 指令触发的异常。在操作系统环境中，应用程序可以使用 SVC 指令访问操作系统内核功能和设备驱动程序。
- **调试监控器：**该异常是由调试监控器（未停止时）引起。该异常仅在启用时有效。如果该异常的优先级低于当前激活的优先级，则该异常不会激活。
- **PendSV：**PendSV 是一种可挂起、由中断驱动的系统级服务请求。在操作系统环境中，当没有其他异常处于活跃状态时，使用 PendSV 执行上下文切换。PendSV 使用中断控制和状态 CPU_SCS:ICSR 寄存器来触发。
- **SysTick：**当系统计时器达到 0 且允许生成中断时，会生成 SysTick 异常。软件也可以使用中断控制和状态寄存器 CPU_SCS:ICSR 来生成 SysTick 异常。在操作系统环境中，处理器可以使用此异常作为系统节拍。
- **中断 (IRQ)：**中断或 IRQ 是由外设发出信号或由软件请求生成并通过 NVIC（具有优先级）馈送的异常。所有中断都与指令执行异步。在系统中，外设使用中断与处理器进行通信。表 5-7 列出了 CC13x2 和 CC26x2 器件平台的控制器上的中断。

对于除复位以外的异步异常，处理器可以在触发异常和处理器进入异常处理程序之间执行另一条指令。

特权软件可以禁用 表 5-1 显示为具有可配置优先级的异常（请参阅 节 2.9.4 中的 CPU_SCS:SHCSR 寄存器和 CPU_SCS:NVIC_ICER0 寄存器）。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

有关硬故障、、总线故障和使用故障的更多信息，请参阅 节 5.2。

表 5-1. 异常类型

异常类型	矢量编号	优先级 (1)	矢量地址或偏移量 (2)	激活
—	0	—	0x0000 0000	栈顶在复位时从矢量表的第一个条目加载。
复位	1	-3（最高）	0x0000 0004	异步
—	—	—	—	—
硬故障	3	-1	0x0000 000C	—
总线故障	5	可编程 (3)	0x0000 0014	精确时同步，不精确时异步
使用故障	6	可编程	0x0000 0018	同步
—	7 至 10	—	—	保留
SVCall	11	可编程	0x0000 002C	同步
调试监控器	12	可编程	0x0000 0030	同步
—	13	—	—	保留
PendSV	14	可编程	0x0000 0038	异步
SysTick	15	可编程	0x0000 003C	异步
中断	16 及以上	可编程 (4)	0x0000 0040 及更高	异步

(1) 0 是所有可编程优先级的默认优先级。
(2) 请参阅 节 5.1.4。
(3) 请参阅 节 2.9.4 中的 CPU_SCS:SHPR 1。
(4) 请参阅 节 2.9.4 中的 IPRn 寄存器。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.1.3 异常处理程序

处理器使用以下命令来处理异常：

- **中断服务例程 (ISR)：**中断 (IRQx) 是通过 ISR 处理的异常。
- **故障处理程序：**硬故障、使用故障和总线故障是由故障处理程序处理的故障异常。
- **系统处理程序：**PendSV、SVCall、SysTick 及故障异常都是由系统处理程序处理的系统异常。

5.1.4 矢量表

图 5-1 包含所有异常处理程序的堆栈指针和起始地址（也称作异常矢量）的复位值。矢量表是使用表 5-1 中列出的矢量地址或偏移量构建的。图 5-1 显示了矢量表中异常矢量的顺序。每个矢量的最低有效位 (LSB) 必须为 1，表示异常处理程序为 Thumb 代码。

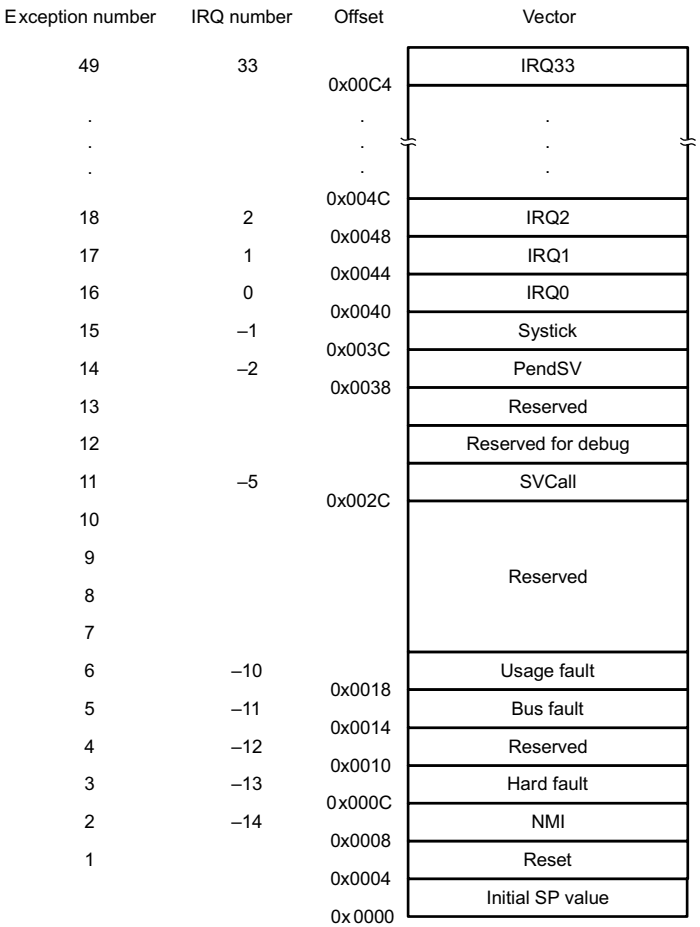


图 5-1. 矢量表

在系统复位时，矢量表固定在地址 0x0000 0000 处。特权软件可以写入矢量表偏移寄存器 (CPU_SCS:VTOR)，以将矢量表起始地址重定位到不同的存储器位置，范围为 0x0000 0200 至 0x3FFF FE00。配置 CPU_SCS:VTOR 寄存器时，偏移必须在 512 字节边界上对齐。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.1.5 异常优先级

如表 5-1 所示，所有异常都具有关联的优先级，其较低的优先级值表示除复位和硬故障之外的所有异常的较高优先级和可配置的优先级。如果软件未配置任何优先级，则所有具有可配置优先级的异常的优先级都为 0。有关配置异常优先级的信息，请参阅节 2.9.4 中的系统处理程序优先级寄存器 (CPU_SCS:SHPRn) 和中断优先级寄存器 (CPU_SCS:NVIC_IPRn)。

备注

CC13x2 和 CC26x2 器件平台实现的可配置优先级值介于 0 到 7 之间。这意味着，具有固定负优先级值的复位和硬故障异常的优先级始终高于任何其他异常。

例如，为 IRQ[0]分配一个较高的优先级值，为 IRQ[1] 分配一个较低的优先级值，意味着 IRQ[1] 具有比 IRQ[0] 更高的优先级。如果 IRQ[1] 和 IRQ[0] 有效，则在 IRQ[0] 之前处理 IRQ[1]。

如果多个挂起异常具有相同的优先级，则具有最低异常编号的挂起异常优先。例如，如果 IRQ[0] 和 IRQ[1] 挂起并具有相同的优先级，则在 IRQ[1] 之前处理 IRQ[0]。

当处理器正在执行异常处理程序时，如果发生更高优先级的异常，异常处理程序将被抢占。如果发生异常的优先级与正在处理的异常相同，则无论异常编号如何，处理程序都不会被抢占。但是，新中断的状态更改为挂起。

5.1.6 中断优先级分组

为了在具有中断的系统中增强优先级控制，NVIC 支持优先级分组。这种分组将每个中断优先级寄存器条目划分为两个字段：

- 一个高位字段，用于定义组优先级
- 一个低位字段，用于定义组内的子优先级

只有组优先级决定中断异常的抢占。当处理器正在执行一个中断异常处理程序时，另一个与该正在处理的中断具有相同组优先级的中断不会抢占该处理程序。

如果多个挂起的中断具有相同的组优先级，则子优先级字段决定它们的处理顺序。如果多个挂起的中断具有相同的组优先级和子优先级，将首先处理 IRQ 编号最小的中断。

有关将中断优先级字段拆分为组优先级和子优先级的信息，请参阅节 2.9.4 中的应用中断/复位控制寄存器 (CPU_SCS:AIRCR)。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.1.7 异常进入和返回

异常处理的说明使用以下术语：

- **抢占式：**当处理器执行异常处理程序时，如果异常的优先级高于正在处理的异常的优先级，则前者可以抢占异常处理程序。有关中断进行抢占的更多信息，请参阅节 5.1.6。当一个异常抢占另一个异常的资源时，这些异常称为嵌套异常。有关更多信息，请参阅 节 5.1.7.1。
- **返回：**当异常处理程序完成，并且没有具有足够优先级的挂起异常可供处理，而且完成的异常处理程序没有处理延迟到达的异常时，便会发生返回。处理器会弹出栈并将处理器状态恢复成中断发生之前的状态。有关更多信息，请参阅 节 5.1.7.2。
- **尾链：**该机制可以加快异常服务。当异常处理程序完成时，如果挂起的异常满足异常进入的要求，则会跳过栈弹出并将控制传输到新的异常处理程序。
- **延迟抵达：**这种机制会加快抢占速度。如果在保存先前异常的状态期间发生高优先级异常，处理器会切换为处理高优先级异常，并启动该异常的矢量获取。状态保存不受延迟到达的影响，因为保存的状态对于这两个例外都是相同的。因此，状态保存不会中断。处理器可以接受延迟到达异常，直到原始异常的异常处理程序的第一条指令进入处理器的执行阶段。当延迟到达的异常从异常处理程序返回时，将应用正常的尾链规则。

5.1.7.1 异常进入

当存在具有足够优先级的挂起异常，且处理器处于线程模式或新异常的优先级高于正在处理的异常时，就会发生异常进入。在后一种情况下，新异常会抢占原异常。

当一个异常抢占另一个异常时，就会形成异常嵌套。

*足够优先级*意味着该异常的优先级高于屏蔽寄存器所设定的任何限制（请参阅优先级屏蔽寄存器 PRIMASK、故障屏蔽寄存器 FAULTMASK 和基址优先级寄存器 BASEPRI）。优先级低于此限制的异常将保持挂起状态，不会被处理器处理。

当处理器接受一个异常时，除非该异常是尾链异常或延迟到达异常，否则处理器会将信息压入当前栈（请参阅图 5-2）。该操作称为压栈，而八个数据字的结构称为栈帧。

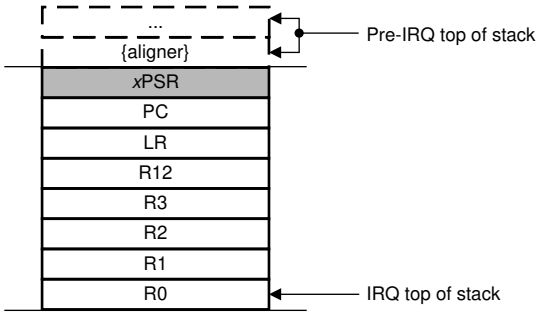


图 5-2. 异常栈帧

压栈后，栈指针立即指向栈帧中的最低地址。

栈帧中包含返回地址，即被中断程序中下一条指令的地址。该值在异常返回时会恢复到 PC 中，以便被中断程序恢复执行。

执行压栈操作的同时，处理器会执行向量获取，从向量表中读取异常处理程序的起始地址。压栈完成后，处理器开始执行异常处理程序。同时，处理器会将一个 EXC_RETURN 值写入 LR，指示哪个栈指针对应于该栈帧，以及进入异常前处理器处于哪种工作模式。

如果在异常进入期间没有发生更高优先级的异常，则处理器开始执行异常处理程序，并自动将相应挂起中断更改为活动状态。

如果在异常进入期间发生了另一个更高优先级的异常（称为延迟到达异常），则处理器开始执行该异常的异常处理程序，且不会更改前一个异常的挂起状态。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.1.7.2 异常返回

当处理器处于处理程序模式，并执行以下任一指令将 EXC_RETURN 值加载到 PC 时，便会发生异常返回：

- 加载 PC 的 LDM 或 POP 指令
- 使用任意寄存器的 BX 指令
- 以 PC 为目标的 LDR 指令

EXC_RETURN 是在异常进入时加载到 LR 的值。异常机制依靠该值来检测处理器何时完成异常处理程序。该值的最低 4 位提供有关返回栈和处理器模式的信息。表 5-2 列出了 EXC_RETURN 值以及异常返回行为说明。

EXC_RETURN 位 31 – 4 全部置位。当该值加载到 PC 时，即向处理器表明异常已完成，处理器随即启动相应的异常返回序列。

表 5-2. 异常返回行为

EXC_RETURN[31:0]	说明
0xFFFF FFF0	保留
0xFFFF FFF1	返回处理程序模式 异常返回使用来自 MSP 的状态 返回后继续使用 MSP 执行。
0xFFFF FFF2 至 0xFFFF FFF8	保留
0xFFFF FFF9	返回线程模式：VTOR 异常返回使用来自 MSP 的状态 返回后继续使用 MSP 执行。
0xFFFF FFFA 至 0xFFFF FFFC	保留
0xFFFF FFFD	返回线程模式 异常返回使用来自 PSP 的状态。 返回后继续使用 PSP 执行
0xFFFF FFFE 至 0xFFFF FFFF	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.2 故障处理

故障是异常的一个子集（请参阅节 5.1）。以下情况会产生故障：

- 获取指令、加载向量表或访问数据时发生总线错误
- 在内部检测到错误，如未定义指令或试图通过 BX 指令更改状态

5.2.1 故障类型

表 5-3 列出了故障类型、用于故障的处理程序、相应的故障状态寄存器，以及指示发生了故障的寄存器位。有关故障状态寄存器的更多信息，请参阅节 2.9 中的 CPU_SCS:CFSR。

表 5-3. 故障

故障	处理程序	故障状态寄存器	位名称
引导程序读取时出现总线错误	硬故障	硬故障状态 (HFSR)	VECTTBL
故障升级为硬故障	硬故障	硬故障状态 (HFSR)	FORCED
异常堆叠期间出现总线错误	总线故障	总线故障状态 (BFSR)	STKERR
异常取消堆叠期间出现总线错误	总线故障	总线故障状态 (BFSR)	UNSTEKRR
指令预取期间出现总线错误	总线故障	总线故障状态 (BFSR)	IBUSERR
数据总线精确误差	总线故障	总线故障状态 (BFSR)	PRECISERR
数据总线不精确误差	总线故障	总线故障状态 (BFSR)	IMPRECISERR
尝试访问协处理器	使用故障	使用故障状态 (UFSR)	NOCP
未定义的指令	使用故障	使用故障状态 (UFSR)	UNDEFINSTR
尝试进入无效指令集状态 (1)	使用故障	使用故障状态 (UFSR)	INVSTATE
EXC_RETURN 值无效	使用故障	使用故障状态 (UFSR)	INVPC
非法的未对齐加载或存储	使用故障	使用故障状态 (UFSR)	UNALIGNED
除以 0	使用故障	使用故障状态 (UFSR)	DIVBYZERO

(1) 尝试使用 Thumb 指令集以外的指令集，或返回到非加载存储多指令并继续执行 ICI。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.2.2 故障升级和硬故障

除硬故障以外的所有故障异常都具有可配置的异常优先级（参见节 2.9.4 中的 CPU_SCS:SHPR1）。软件可以禁用这些故障的处理程序的执行（参见节 2.9.4 中的 CPU_SCS:SHCSR）。

通常，异常优先级与异常屏蔽寄存器的值一起确定处理器是否进入故障处理程序，以及故障处理程序是否可以优先于另一个故障处理程序，如节 2.9.4 中所述。

在某些情况下，具有可配置优先级的故障会被视为硬故障。此流程称为优先级升级，故障被描述为升级的硬故障。在以下情况下升级至硬故障：

- 故障处理程序导致的故障类型与它正在维护的故障类型相同。这种升级到硬故障的原因是故障处理程序无法自行抢占，因为它必须具有与当前优先级相同的优先级。
- 故障处理程序会导致优先级与其处理的故障相同或比之更低的故障。发生这种情况是因为新故障的处理程序无法优先于当前正在执行的故障处理程序。
- 异常处理程序会导致优先级等于或低于当前正执行的异常的故障。
- 发生故障，且该故障的处理程序未启用。

备注

如果进入总线故障处理程序时在栈推送过程中发生总线故障，总线故障不会升级为硬故障。因此，如果栈损坏导致故障，即使处理程序的栈推入失败，也会执行故障处理程序。故障处理程序会运行，但栈内容已损坏。

5.2.3 故障状态寄存器和故障地址寄存器

故障状态寄存器可以指示故障原因。对于总线故障，故障地址寄存器指示导致故障的操作所访问的地址，如 表 5-4 所示。

表 5-4. 故障状态和故障地址寄存器

处理程序	状态寄存器名称	地址寄存器名称	寄存器说明
硬故障	硬故障状态 (HFSR)	—	请参阅 节 2.9.4
总线故障	总线故障状态 (BFSR)	总线故障地址 (BFAR)	请参阅 节 2.9.4
使用故障	使用故障状态 (UFSR)	—	—

5.2.4 锁定

如果在执行硬故障处理程序时发生硬故障，处理器会进入锁定状态。锁定状态会将 CC13x2 和 CC26x2 器件平台中的系统复位。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.3 事件结构

5.3.1 简介

事件结构 (event fabric) 是事件源与事件订阅者之间的组合路由器。事件输入被路由到中央事件总线，订阅者可以在总线中选择适当的事件，并输出它们以作为外设的输入。图 5-3 展示了事件结构的一般概念。事件结构是严格的组合逻辑。本章仅提供事件结构以及系统 CPU、NMI 和 Freeze 订阅者的总体概述，有关如何为不同订阅者和外设使用和配置事件，请参阅本用户指南中具体的外设章节。

大多数事件（信号）采用静态路由，这意味着仅有少量可配置的输出线路连接到事件订阅者。来自订阅者的可配置输出线路可以从该特定订阅者可用的多个输入事件中进行选择。订阅者输出的事件信号与输入信号完全相同。也就是说，事件只是按输入端口上呈现的原样通过事件结构传递。可能的事件类型包括系统硬件中断、软件可编程中断以及 DMA 触发条件。所有事件输入均视为高电平有效的电平触发事件。DMA 触发条件等事件可能是电平类型信号，也可能不是。

图 5-3 展示了事件结构概念的简单示意图。显然，事件结构本身并非外设，而是外设及其他模块之间的路由块。具有可配置输入的线路由连接到 MUX 的选择寄存器控制，该 MUX 将订阅者中的选定输入转发给外设。

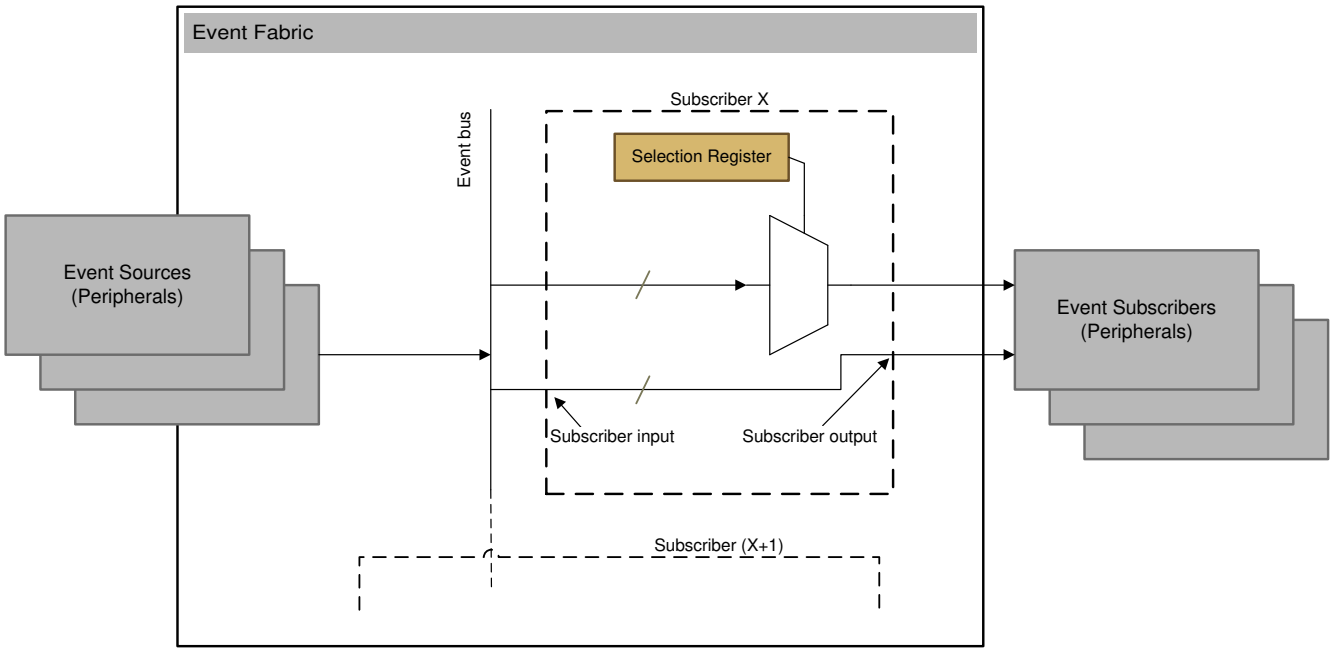


图 5-3. 事件结构概念

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.3.2 事件结构概述

CC13x2 和 CC26x2 器件中有两个主要的事件结构模块。一个位于 MCU 电源域（MCU 事件结构），另一个位于 AON 电源域（AON 事件结构）。图 5-4 显示了这两个模块的简化概述。MCU 事件结构是 AON 事件结构的订阅者之一。

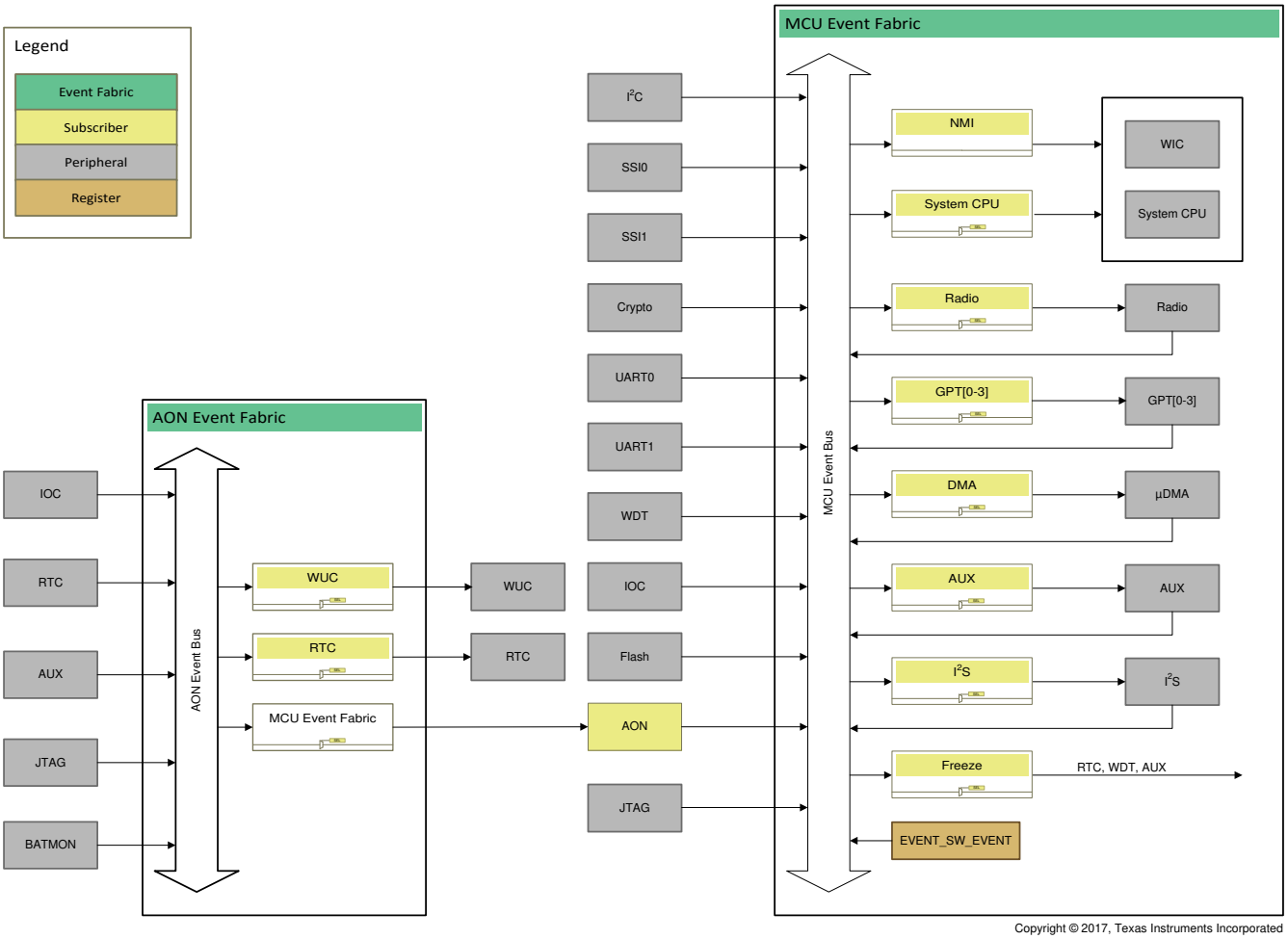


图 5-4. 事件结构概述（简化）

5.3.2.1 寄存器

事件结构有两种寄存器类型。第一种类型是配置寄存器，用于控制和报告订阅者输出的选择设置。对于每个订阅者输出，会映射一个读取寄存器的地址，其中包含代表当前为该订阅者输出所设置的输入事件选择。对于不可配置的输出，仅实现一个只读寄存器。读取该地址会返回静态的预定义值。事件结构中第二种类型的寄存器只有一个，即名为 SWEV 的操作寄存器。该寄存器可置位和清除四个软件事件中的任何一个。

AON 事件结构通过驻留在 MCU 电源域中的一系列寄存器进行控制。AON 域中的 AON-MCU 接口模块会镜像这些寄存器，从而在 MCU 处于掉电状态时为 AON 模块提供这些寄存器数据。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.4 AON 事件结构

AON 事件结构位于 AON 电源域中，该电源域中包含唤醒控制器、调试子系统、AUX 域和实时时钟 (RTC)。

5.4.1 通用输入事件列表

节 5.5.1 列出了 AON 事件结构的输入事件。这些事件的源被视为电平触发的高电平有效。

5.4.2 事件订阅者

AON 事件结构中有三个订阅者，如 图 5-4 中所示。第一个订阅者是位于 MCU 电源域中的 MCU 事件结构。其他两个订阅者（WUC 和 RTC）都位于 AON 电源域中，将在以下小节中介绍。

5.4.2.1 唤醒控制器 (WUC)

WUC 接收来自 AON 事件结构中 WUC 订阅者的输出信号，在该信号中上电序列可由来自 JTAG、AUX 或 MCU 的配置输入事件触发。JTAG 有一个进入 WUC 的唤醒事件，而 AUX 域有三个可编程输入事件，MCU 域有四个可编程输入事件。这些特定的输入事件在一起进行 OR 运算，形成 WUC 的单个输入，一个来自 MCU，一个来自 AUX。图 5-5 显示了该配置。可以在两个选择寄存器中配置输入，即 AON_EVENT:AUXWUSEL 和 AON_EVENT:MCUWUSEL。通过选择相应的事件 ID，可以选择表 5-5 中列出的任何事件作为输入。默认情况下，这些 ID 设置为 63（NULL，无事件），其中线路始终保持逻辑低电平。

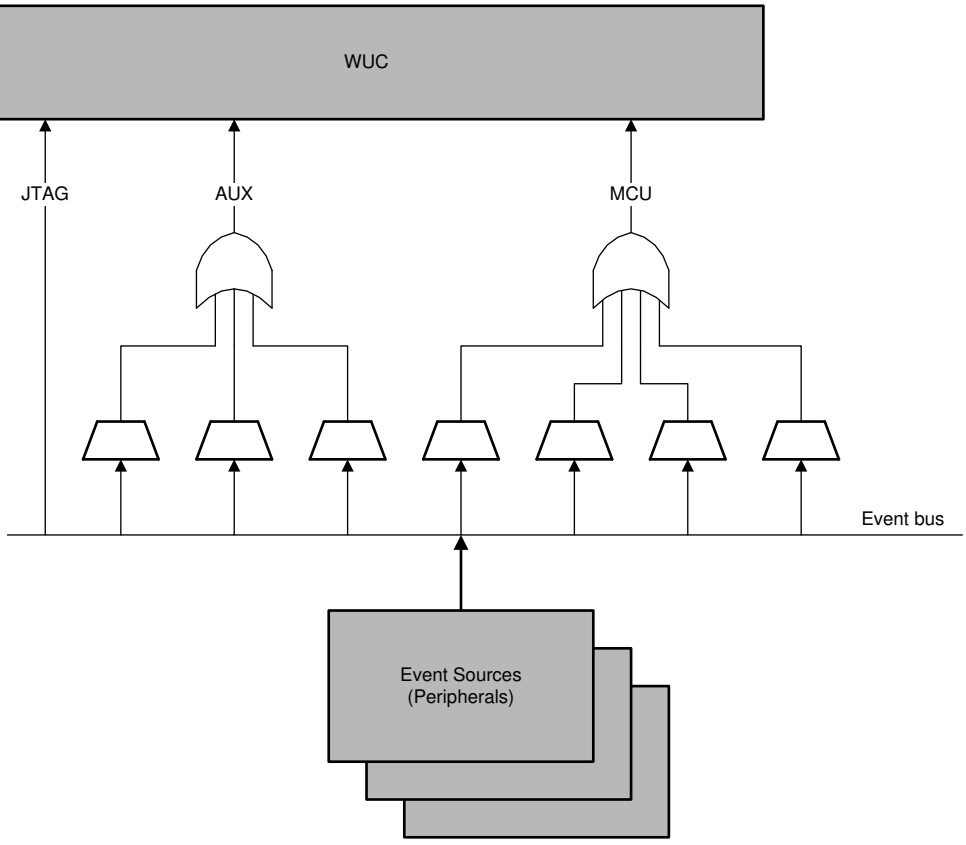


图 5-5. AON 事件结构中的 WUC 订阅者

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.4.2.2 实时时钟

RTC 有一个可在 RTCSEL 寄存器中配置的可编程事件以及一个 ID 为 46 的固定事件（通道 2 清除 — 来自 AUX）。

5.4.2.3 MCU 事件结构

来自 AON 事件结构的七个输出事件作为输入路由到 MCU 事件结构。这些事件包括：

- AON 可编程 0
- AON 可编程 1
- AON 可编程 2
- AON 边沿检测
- AON RTC

有三条可编程线路，可以为其选择表 5-5 中的任何输入事件。这可以在 CTRL_EVENT:MCU 寄存器中设置。

5.5 MCU 事件结构

MCU 事件结构位于 MCU 电源域中，在大多数外设和不同的内部模块之间路由信号。本节仅介绍 MCU 事件结构中的少数订阅者。有关其余订阅者的更多信息，请参阅具体订阅者的相应消费者（外设）的特定外设章节。

5.5.1 通用输入事件列表

表 5-5 列出了 MCU 事件结构的输入事件。这些事件的源被视为电平触发的高电平有效。

表 5-5. MCU 事件结构输入事件

事件编号	事件枚举	说明
0x0	无	始终无效（低电平）
0x1	AON_PROG0	通过 AON_EVENT MCU 事件选择器选择的事件，AON_EVENT:EVTOMCUSEL.AON_PROG0_EV
0x2	AON_PROG1	通过 AON_EVENT MCU 事件选择器选择的事件，AON_EVENT:EVTOMCUSEL.AON_PROG1_EV
0x3	AON_PROG2	通过 AON_EVENT MCU 事件选择器选择的事件，AON_EVENT:EVTOMCUSEL.AON_PROG2_EV
0x4	AON_GPIO_EDGE	来自 IOC 的边沿检测事件。通过 IOC:IOCFGn.EDGE_IRQ_EN 和 IOC:IOCFGn.EDGE_DET 设置进行配置
0x5	RESERVED	
0x6	RESERVED	
0x7	AON_RTC_COMB	来自 AON_RTC 的事件，通过 AON_RTC:CTL.COMB_EV_MASK 设置来控制
0x8	I2S_IRQ	来自 I^2S 的中断事件
0x9	I2C_IRQ	来自 I^2C 的中断事件
0xA	AON_AUX_SWEV0	AUX 软件事件 0，AUX_EVCTL:SWEVSET.SWEV0
0xB	AUX_COMB	AUX 组合事件，相应的标志寄存器在这里为 AUX_EVCTL:EVTOMCUFLAGS.*
0xC	GPT2A	GPT2A 中断事件，由 GPT2:TAMR.* 控制
0xD	GPT2B	GPT2B 中断事件，由 GPT2:TBMR.* 控制
0xE	GPT3A	GPT3A 中断事件，由 GPT3:TAMR.* 控制
0xF	GPT3B	GPT3B 中断事件，由 GPT3:TBMR.* 控制
0x10	GPT0A	GPT0A 中断事件，由 GPT0:TAMR.* 控制
0x11	GPT0B	GPT0B 中断事件，由 GPT0:TBMR.* 控制
0x12	GPT1A	GPT1A 中断事件，由 GPT1:TAMR.* 控制
0x13	GPT1B	GPT1B 中断事件，由 GPT1:TBMR.* 控制

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 5-5. MCU 事件结构输入事件（续）

事件编号	事件枚举	说明
0x14	DMA_CH0_DONE	软件触发的 UDMA 通道 0 的 DMA 完成，请参阅节 14.5.1 中的 UDMA0:SOFTREQ.*
0x15	闪存	闪存控制器错误事件，状态标志为 FLASH:FEDACSTAT.FSM_DONE 和 FLASH:FEDACSTAT.RVF_INT
0x16	DMA_CH18_DONE	软件触发的 UDMA 通道 18 的 DMA 完成，请参阅节 14.5.1 中的 UDMA0:SOFTREQ.*
0x17	RESERVED	
0x18	WDT_IRQ	看门狗中断事件，由 WDT:CTL.INTEN 控制
0x19	RFC_CMD_ACK	RFC 门铃命令确认中断，相当于 RFC_DBELL:RFACKIFG.ACKFLAG。
0x1A	RFC_HW_COMB	组合的 RCF 硬件中断，相应的标志在此处为 RFC_DBELL:RFHWIFG.*
0x1B	RFC_CPE_0	CPE 生成的事件的组合中断。相应的标志在这里是 RFC_DBELL:RFCPEIFG.*。只有在 RFC_DBELL:RFCPEIFG.* 中通过 CPE0 选择的中断才能触发 RFC_CPE_0 事件。
0x1C	AUX_SWEV0	AUX 软件事件 0，由 AUX_EVCTL:SWEVSET.SWEV0 触发，也可用作 AUX_EVENT0 AON 唤醒事件。 MCU 域唤醒控制 AON_EVENT:MCUWUSEL.* AUX 域唤醒控制 AON_EVENT:AUXWUSEL.*
0x1D	AUX_SWEV1	AUX 软件事件 1，由 AUX_EVCTL:SWEVSET.SWEV1 触发，也可用作 AUX_EVENT2 AON 唤醒事件。 MCU 域唤醒控制 AON_EVENT:MCUWUSEL.* AUX 域唤醒控制 AON_EVENT:AUXWUSEL.*
0x1E	RFC_CPE_1	CPE 生成的事件的组合中断。相应的标志在这里是 RFC_DBELL:RFCPEIFG.*。只有在 RFC_DBELL:RFCPEIFG.* 中通过 CPE1 选择的中断才能触发 RFC_CPE_1 事件。
0x1F 至 0x21	RESERVED	
0x22	SSI0_COMB	SSI0 组合中断，在此处找到的中断标志为 SSI0:MIS.*。
0x23	SSI1_COMB	SSI0 组合中断，中断标志位于此处 SSI1:MIS.*。
0x24	UART0_COMB	UART0 组合中断，在此处找到的中断标志为 UART0:MIS.*。
0x25	RESERVED	始终为 0（低电平）
0x26	DMA_ERR	DMA 总线错误，对应于 UDMA0:ERROR.STATUS
0x27	DMA_DONE_COMB	组合式 DMA 完成，此处相应的标志为 UDMA0:REQDONE.*
0x28	SSI0_RX_DMABREQ	SSI0 RX DMA 突发请求，通过 SSI0:DMACR.RXDMAE 控制
0x29	SSI0_RX_DMASREQ	SSI0 RX DMA 单请求，通过 SSI0:DMACR.RXDMAE 控制
0x2A	SSI0_TX_DMABREQ	SSI0 TX DMA 突发请求，通过 SSI0:DMACR.TXDMAE 控制
0x2B	SSI0_TX_DMASREQ	SSI0 TX DMA 单请求，通过 SSI0:DMACR.TXDMAE 控制
0x2C	SSI1_RX_DMABREQ	SSI1 RX DMA 突发请求，通过 SSI0:DMACR.RXDMAE 控制
0x2D	SSI1_RX_DMASREQ	SSI1 RX DMA 单请求，通过 SSI0:DMACR.RXDMAE 控制
0x2E	SSI1_TX_DMABREQ	SSI1 TX DMA 突发请求，通过 SSI0:DMACR.TXDMAE 控制
0x2F	SSI1_TX_DMASREQ	SSI1 TX DMA 单请求，通过 SSI0:DMACR.TXDMAE 控制
0x30	UART0_RX_DMABREQ	UART0 RX DMA 突发请求，通过 UART0:DMACTL.RXDMAE 控制
0x31	UART0_RX_DMASREQ	UART0 RX DMA 单请求，通过 UART0:DMACTL.RXDMAE 控制
0x32	UART0_TX_DMABREQ	UART0 TX DMA 突发请求，通过 UART0:DMACTL.TXDMAE 控制
0x33	UART0_TX_DMASREQ	UART0 TX DMA 单请求，通过 UART0:DMACTL.TXDMAE 控制
0x34 至 0x37	RESERVED	始终为 0
0x38	RESERVED	
0x39	RESERVED	
0x3A	RESERVED	

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 5-5. MCU 事件结构输入事件（续）

事件编号	事件枚举	说明
0x3B	RESERVED	
0x3C	RESERVED	
0x3D	GPT0A_CMP	GPT0A 比较事件。通过 GPT0:TAMR.TCACT 配置。
0x3E	GPT0B_CMP	GPT0B 比较事件。通过 GPT0:TBMR.TCACT 配置。
0x3F	GPT1A_CMP	GPT1A 比较事件。通过 GPT1:TAMR.TCACT 配置。
0x40	GPT1B_CMP	GPT1B 比较事件。通过 GPT1:TBMR.TCACT 配置。
0x41	GPT2A_CMP	GPT2A 比较事件。通过 GPT2:TAMR.TCACT 配置。
0x42	GPT2B_CMP	GPT2B 比较事件。通过 GPT2:TBMR.TCACT 配置。
0x43	GPT3A_CMP	GPT3A 比较事件。通过 GPT3:TAMR.TCACT 配置。
0x44	GPT3B_CMP	GPT3B 比较事件。通过 GPT3:TBMR.TCACT 配置。
0x45 至 0x4C	TIE_LOW	未使用；连接至 0（低电平）
0x4D	GPT0A_DMABREQ	GPT 0A DMA 触发事件。通过 GPT0:DMAEV.* 进行配置。
0x4E	GPT0B_DMABREQ	GPT 0B DMA 触发事件。通过 GPT0:DMAEV.* 进行配置。
0x4F	GPT1A_DMABREQ	GPT 1A DMA 触发事件。通过 GPT1:DMAEV.* 进行配置。
0x50	GPT1B_DMABREQ	GPT 1B DMA 触发事件。通过 GPT1:DMAEV.* 进行配置。
0x51	GPT2A_DMABREQ	GPT 2A DMA 触发事件。通过 GPT2:DMAEV.* 进行配置。
0x52	GPT2B_DMABREQ	GPT 2B DMA 触发事件。通过 GPT2:DMAEV.* 进行配置。
0x53	GPT3A_DMABREQ	GPT 3A DMA 触发事件。通过 GPT3:DMAEV.* 进行配置。
0x54	GPT3B_DMABREQ	GPT 3B DMA 触发事件。通过 GPT3:DMAEV.* 进行配置。
0x55	PORT_EVENT0	来自 IOC 的端口捕获事件，通过 IOC:IOCFGn.PORT_ID 进行配置。通过 **ENUM** **PORT_EVENT0** 配置的端口上的事件将在此处路由。
0x56	PORT_EVENT1	来自 IOC 的端口捕获事件，通过 IOC:IOCFGn.PORT_ID 进行配置。通过 PORT_EVENT1 配置的端口上的事件会在此处路由。
0x57	PORT_EVENT2	来自 IOC 的端口捕获事件，通过 IOC:IOCFGn.PORT_ID 进行配置。通过 PORT_EVENT2 配置的端口上的事件会在此处路由。
0x58	PORT_EVENT3	来自 IOC 的端口捕获事件，通过 IOC:IOCFGn.PORT_ID 进行配置。通过 PORT_EVENT3 配置的端口上的事件会在此处路由。
0x59	PORT_EVENT4	来自 IOC 的端口捕获事件，通过 IOC:IOCFGn.PORT_ID 进行配置。通过 ENUM PORT_EVENT4 配置的端口上的事件将在此处路由。
0x5A	PORT_EVENT5	来自 IOC 的端口捕获事件，通过 IOC:IOCFGn.PORT_ID 进行配置。通过 PORT_EVENT4 配置的端口上的事件会在此处路由。
0x5B	PORT_EVENT6	来自 IOC 的端口捕获事件，通过 IOC:IOCFGn.PORT_ID 进行配置。通过 PORT_EVENT6 配置的端口上的事件会在此处路由。
0x5C	PORT_EVENT7	来自 IOC 的端口捕获事件，通过 IOC:IOCFGn.PORT_ID 进行配置。通过 PORT_EVENT7 配置的端口上的事件会在此处路由。
0x5D	CRYPTO_RESULT_AVAIL_IRQ	加密结果可用中断事件，在此处找到相应的标志 CRYPTO:IRQSTAT.RESULT_AVAIL。由 CRYPTO:IRQSTAT.RESULT_AVAIL 控制
0x5E	CRYPTO_DMA_DONE_IRQ	CRYPTO DMA 输入完成事件，相应的标志为 CRYPTO:IRQSTAT.DMA_IN_DONE。通过 CRYPTO:IRQEN.DMA_IN_DONE 控制
0x5F	RFC_IN_EV4	RFC RAT 事件 4，通过 RFC_RAT:RATEV.OEVT4 进行配置
0x60	RFC_IN_EV5	RFC RAT 事件 5，通过 RFC_RAT:RATEV.OEVT5 进行配置
0x61	RFC_IN_EV6	RFC RAT 事件 6，通过 RFC_RAT:RATEV.OEVT6 进行配置
0x62	RFC_IN_EV7	RFC RAT 事件 7，通过 RFC_RAT:RATEV.OEVT7 进行配置
0x63	WDT_NMI	WATCHDOG 不可屏蔽中断事件，由 WDT:CTL.INTTYPE 控制

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 5-5. MCU 事件结构输入事件（续）

事件编号	事件枚举	说明
0x64	SWEV0	软件事件 0，由 SWEV.SWEV0 触发
0x65	SWEV1	软件事件 1，由 SWEV.SWEV1 触发
0x66	SWEV2	软件事件 2，由 SWEV.SWEV2 触发
0x67	SWEV3	软件事件 3，由 SWEV.SWEV3 触发
0x68	TRNG_IRQ	TRNG 中断事件，通过 TRNG:IRQEN.EN 控制
0x69	AUX_AON_WU_EV	AON 唤醒事件，对应的标志在这里为 AUX_EVCTL:EVTOMCUFLAGS.AON_WU_EV。
0x6A	AUX_COMPA	AUX COMP A 事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_COMPA。
0x6B	AUX_COMPB	AUX COMP B 事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_COMPB。
0x6C	AUX_TDC_DONE	AUX TDC 测量完成事件，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.TDC_DONE 和 AUX_TDC 状态 AUX_TDC:STAT.DONE。
0x6D	AUX_TIMER0_EV	AUX 计时器 0 事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.TIMER0_EV。
0x6E	AUX_TIMER1_EV	AUX 计时器 1 事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.TIMER1_EV。
0x6F	AUX_SMPH_AUTOTAKE_DONE	来自 AUX 信标的自动获取事件，通过 AUX_SMPH:AUTOTAKE.* 配置。
0x70	AUX_ADC_DONE	AUX ADC 完成，对应于 AUX_EVCTL:EVTOMCUFLAGS.ADC_DONE
0x71	AUX_ADC_FIFO_ALMOST_FULL	AUX ADC FIFO 水印事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.ADC_FIFO_ALMOST_FULL
0x72	AUX_OBSMUX0	OBSMUX0 通过 AUX 环回，对应于 AUX_EVCTL:EVTOMCUFLAGS.OBSMUX0
0x73	AUX_ADC_IRQ	AUX ADC 中断事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.ADC_IRQ。在这里找到的状态标志为 AUX_EVCTL:EVTOMCUFLAGS.ADC*
0x74	AUX_SW_DMABREQ	AUX 观察回送
0x75	AUX_DMASREQ	来自 AUX 的 DMA 单请求事件，通过 AUX_EVCTL:DMACTL.* 进行配置
0x76	AUX_DMABREQ	来自 AUX 的 DMA 突发请求事件，通过 AUX_EVCTL:DMACTL.* 进行配置
0x77	AON_RTC_UPD	通过 AON_RTC:CTL.RTC_UPD_EN 控制的 RTC 周期性事件
0x78	CPU_HALTED	CPU 暂停
0x79	ALWAYS_ACTIVE	始终置为有效（高电平）

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.5.2 事件订阅者

MCU 事件结构有 11 个订阅者。这些订阅者大多是不同的外设，必须根据各自用途进行不同的配置。以下五个订阅者未在本章中介绍，而会各自对应的外设章节中介绍：

- 微型直接存储器存取 (µDMA)（请参阅章节 14）
- 通用计时器（请参阅章节 15）
- 带数字和模拟外设的传感器控制器 (AUX)（请参阅章节 19）
- IC 间音频 (I^2S)（请参阅章节 24）
- 无线电（请参阅章节 25）

这里介绍了以下三个订阅者，因为它们与 CPU 和 CPU 中断相关：

- 系统 CPU
- 系统 CPU 的不可屏蔽中断 (NMI)
- 冻结

5.5.2.1 系统 CPU

表 5-7 展示了矢量数为 16 至 49 的中断源自在 MCU 事件结构中路由至系统 CPU 的事件。事件结构将所有级别的中断事件路由至系统 CPU。名为 *AON 可编程 0* 的事件/中断可在 AON 事件结构中配置。EVENT:CPUIRQSEL29 是一个只读寄存器，用于在 MCU 事件结构内进行路由且无法配置，但进入该线路的 AON 事件结构内的输入事件可以进行配置。一个称为动态 *可编程事件* 的动态事件/中断具有表 5-7 中所示的有效选择。EVENT:CPUIRQSEL29 寄存器用于配置输入（请参阅节 5.7.2）。

请参阅节 5.7.2 中的 EVENT:CPUIRQSEL30 寄存器。

5.5.2.2 NMI

NMI 订阅者有一个来自 WDT 的不可配置输入。只读寄存器 (CM3NMISEL0) 仅显示唯一有效的输入事件。

5.5.2.3 冻结

在 CC13x2 和 CC26x2 器件平台中，冻结订阅者将暂停的调试信号传递给通用计时器、具有数字及模拟外设 (AUX) 的传感器控制器、无线电及 RTC 等外设。当系统 CPU 停止时，已连接且启用了冻结功能的外设也会停止。可编程输出可设置为静态值 0 或 1，也可设置为传递暂停的信号。表 5-6 中列出的可能事件可以在 FRZSEL0 寄存器中选择。

表 5-6. 冻结订阅者事件选择

事件编号	事件枚举
0x0	无
0x78	CPU_HALTED
0x79	ALWAYS_ACTIVE

备注

当冻结置为有效时，RTC 停止递增主计数器，但来自 RTC 的更新事件（进入 RF 内核和 AON 事件结构）不会停止。更新事件是 SCLK_LF 的下分频，不依赖于主计数器。因此在实践中，当您暂停 CPU 进行调试时，无法停止 RFC 的这些更新事件。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.6 AON 事件

表 5-7. AON 事件

事件编号	名称	说明
0x0 至 0x1F	DIO0 至 DIO31	DIOn 上边沿检测，n = 0..31
0x20	DIO	任何 DIO 上的边沿检测
0x23 至 0x25	RTC_CH0 至 RTC_CH2	RTC 通道 n 事件，n = 0..2
0x26 至 0x28	RTC_CH0_DLY 至 RTC_CH2_DLY	RTC 通道 n – 延迟事件，n = 0..2
0x29	RTC_COMB_DLY	RTC 组合延迟事件
0x2A	RTC_UPD	RTC 更新节拍
0x2B	JTAG	JTAG 生成的事件
0x2C 至 0x2E	AUX_SWEV0 至 AUX_SWEV2	AUX 软件触发事件 #n，n = 0..2
0x2F	AUX_COMPA	已触发比较器 A
0x30	AUX_COMPB	已触发比较器 B
0x31	AUX_ADC_DONE	ADC 转换完成
0x32	AUX_TDC_DONE	TDC 已完成或超时
0x33 至 0x34	AUX_TIMER0_EV 至 AUX_TIMER1_EV	AUX 计时器 n 事件，n = 0..1
0x35	BATMON_TEMP	BATMON 温度更新事件
0x36	BATMON_VOLT	BATMON 电压更新事件
0x37	AUX_COMPB_ASYNC	已触发比较器 B
0x38	AUX_COMPB_ASYNC_N	比较器 B 未触发
0x3F	无	无事件

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7 中断和事件寄存器

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.1 AON_EVENT 寄存器

表 5-8 列出了 AON_EVENT 寄存器的存储器映射寄存器。表 5-8 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

表 5-8. AON_EVENT 寄存器

偏移	首字母缩写词	寄存器名称	部分
0h	MCUWUSEL	MCU 的唤醒选择器	MCUWUSEL 寄存器 (偏移 =0h) [复位 = 3F3F3F3Fh]
4h	MCUWUSEL1	MCU 的唤醒选择器	MCUWUSEL1 寄存器 (偏移 = 4h) [复位 = 3F3F3F3Fh]
8h	EVTOMCUSEL	MCU 事件结构的事件选择器	EVTOMCUSEL 寄存器 (偏移 = 8h) [复位 = 002B2B2Bh]
Ch	RTCSEL	AON_RTC 的 RTC 捕获事件选择器	RTCSEL 寄存器 (偏移 = Ch) [复位 = 0000003Fh]

复杂的位访问类型经过编码可适应小型表单元。表 5-9 展示了适用于此部分中访问类型的代码。

表 5-9. AON_EVENT 访问类型代码

访问类型	代码	说明
读取类型		
R	R	读取
写入类型		
W	W	写入
复位或默认值		
-n		复位后的值或默认值

Copyright © 2026 Texas Instruments Incorporated

5.7.1.1 MCUWUSEL 寄存器（偏移 =0h）[复位 = 3F3F3F3Fh]

图 5-6 展示了 MCUWUSEL，表 5-10 中对此进行了介绍。

返回到汇总表。

MCU 唤醒选择器
此寄存器包含指向 8 个事件（事件 0 至 3）中 4 个的指针，这些事件路由到 AON_PMCTRL 作为 MCU 的唤醒源。当 8 个所选事件中的任何一个生效时，AON_PMCTRL 将启动 MCU 域的唤醒序列。唤醒序列将保证 MCU 电源开关打开、LDO 资源可用、SCLK_HF 可用并且被选为 MCU 的时钟源。
注意：在 MCU 请求断电之前，需要在 AON_EVENT 中设置唤醒事件（PRCM 请求 uLDO，请参阅 PRCM:VDCTL.ULDO 中的条件）。

图 5-6. MCUWUSEL 寄存器

31	30	29	28	27	26	25	24
RESERVED		WU3_EV					
R-0h		R/W-3Fh					
23	22	21	20	19	18	17	16
RESERVED		WU2_EV					
R-0h		R/W-3Fh					
15	14	13	12	11	10	9	8
RESERVED		WU1_EV					
R-0h		R/W-3Fh					
7	6	5	4	3	2	1	0
RESERVED		WU0_EV					
R-0h		R/W-3Fh					

表 5-10. MCUWUSEL 寄存器字段说明

位	字段	类型	复位	说明
31-30	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 5-10. MCUWUSEL 寄存器字段说明（续）

位	字段	类型	复位	说明
29-24	WU3_EV	R/W	3Fh	MCU 唤醒源#3 AON 事件源选择路由到 AON_PMCTRL 的 8 个事件中的 1 个，以从关断或断电状态唤醒 MCU 域。 注意： 0h = 来自 DIO 的边沿检测 IO 事件，其在 [MCU_IOC:IOCFGx.IOEV_MCU_WU_EN] 中启用了对 IOEV_MCU_WU 的贡献 1h = 来自 AUX TImer2 的事件 0 2h = 来自 AUX TImer2 的事件 1 3h = 来自 AUX TImer2 的事件 2 4h = 来自 AUX TImer2 的事件 3 5h = BATMON 事件：电池电平高于上限 6h = BATMON 事件：电池电平低于下限 7h = BATMON 事件：温度水平高于上限 8h = BATMON 事件： 温度低于下限 9h = 来自 BATMON 的组合事件 20h = 任何 PAD 上的边缘检测 23h = RTC 通道 0 事件 24h = RTC 通道 1 事件 25h = RTC 通道 2 事件 26h = RTC 通道 0 - 显示的事件 27h = RTC 通道 1 - 显示的事件 28h = RTC 通道 2 - 显示的事件 29h = RTC 组合的显示事件 2Ah = RTC 更新时钟周期（16 kHz 信号，即事件线每 32 kHz 时钟周期切换一次值） 2Bh = JTAG 生成的事件 2Ch = AUX 软件触发事件 #0. 由 AUX_EVCTL:SWEVSET.SWEV0 触发 2Dh = AUX 软件触发事件 #1。 由 AUX_EVCTL:SWEVSET.SWEV1 触发 2Eh = AUX 软件触发事件 #2。 由 AUX_EVCTL:SWEVSET.SWEV2 触发 2Fh = 触发比较器 A 30h = 触发比较器 B 31h = 已完成 ADC 转换 32h = TDC 已完成或超时 33h = AUX 定时器 0 事件 34h = AUX 定时器 1 事件 35h = BATMON 温度更新事件 36h = BATMON 电压更新事件 37h = 触发比较器 B。 直接来自 AUX 比较器 B 的异步信号，与在 AUX 中同步的 AUX_COMPB 相反 38h = 不触发比较器 B。 直接来自 AUX 比较器 B（反相）的异步信号，与在 AUX 中同步的 AUX_COMPB 相反 3Fh = 无事件，总为低电平
23-22	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 5-10. MCUWUSEL 寄存器字段说明（续）

位	字段	类型	复位	说明
21-16	WU2_EV	R/W	3Fh	MCU 唤醒源#2 AON 事件源选择路由到 AON_PMCTRL 的 8 个事件中的 1 个，以从关断或断电状态唤醒 MCU 域。 注意： 0h = 来自 DIO 的边沿检测 IO 事件，其在 [MCU_IOC:IOCFGx.IOEV_MCU_WU_EN] 中启用了对 IOEV_MCU_WU 的贡献 1h = 来自 AUX TImer2 的事件 0 2h = 来自 AUX TImer2 的事件 1 3h = 来自 AUX TImer2 的事件 2 4h = 来自 AUX TImer2 的事件 3 5h = BATMON 事件：电池电平高于上限 6h = BATMON 事件：电池电平低于下限 7h = BATMON 事件：温度水平高于上限 8h = BATMON 事件： 温度低于下限 9h = 来自 BATMON 的组合事件 20h = 任何 PAD 上的边缘检测 23h = RTC 通道 0 事件 24h = RTC 通道 1 事件 25h = RTC 通道 2 事件 26h = RTC 通道 0 - 显示的事件 27h = RTC 通道 1 - 显示的事件 28h = RTC 通道 2 - 显示的事件 29h = RTC 组合的显示事件 2Ah = RTC 更新时钟周期（16 kHz 信号，即事件线每 32 kHz 时钟周期切换一次值） 2Bh = JTAG 生成的事件 2Ch = AUX 软件触发事件 #0. 由 AUX_EVCTL:SWEVSET.SWEV0 触发 2Dh = AUX 软件触发事件 #1。 由 AUX_EVCTL:SWEVSET.SWEV1 触发 2Eh = AUX 软件触发事件 #2。 由 AUX_EVCTL:SWEVSET.SWEV2 触发 2Fh = 触发比较器 A 30h = 触发比较器 B 31h = 已完成 ADC 转换 32h = TDC 已完成或超时 33h = AUX 定时器 0 事件 34h = AUX 定时器 1 事件 35h = BATMON 温度更新事件 36h = BATMON 电压更新事件 37h = 触发比较器 B。 直接来自 AUX 比较器 B 的异步信号，与在 AUX 中同步的 AUX_COMPB 相反 38h = 不触发比较器 B。 直接来自 AUX 比较器 B（反相）的异步信号，与在 AUX 中同步的 AUX_COMPB 相反 3Fh = 无事件，总为低电平
15-14	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 5-10. MCUWUSEL 寄存器字段说明（续）

位	字段	类型	复位	说明
13-8	WU1_EV	R/W	3Fh	MCU 唤醒源#1 AON 事件源选择路由到 AON_PMCTRL 的 8 个事件中的 1 个，以从关断或断电状态唤醒 MCU 域。 注意： 0h = 来自 DIO 的边沿检测 IO 事件，其在 [MCU_IOC:IOCFGx.IOEV_MCU_WU_EN] 中启用了对 IOEV_MCU_WU 的贡献 1h = 来自 AUX TImer2 的事件 0 2h = 来自 AUX TImer2 的事件 1 3h = 来自 AUX TImer2 的事件 2 4h = 来自 AUX TImer2 的事件 3 5h = BATMON 事件：电池电平高于上限 6h = BATMON 事件：电池电平低于下限 7h = BATMON 事件：温度水平高于上限 8h = BATMON 事件： 温度低于下限 9h = 来自 BATMON 的组合事件 20h = 任何 PAD 上的边缘检测 23h = RTC 通道 0 事件 24h = RTC 通道 1 事件 25h = RTC 通道 2 事件 26h = RTC 通道 0 - 显示的事件 27h = RTC 通道 1 - 显示的事件 28h = RTC 通道 2 - 显示的事件 29h = RTC 组合的显示事件 2Ah = RTC 更新时钟周期（16 kHz 信号，即事件线每 32 kHz 时钟周期切换一次值） 2Bh = JTAG 生成的事件 2Ch = AUX 软件触发事件 #0. 由 AUX_EVCTL:SWEVSET.SWEV0 触发 2Dh = AUX 软件触发事件 #1。 由 AUX_EVCTL:SWEVSET.SWEV1 触发 2Eh = AUX 软件触发事件 #2。 由 AUX_EVCTL:SWEVSET.SWEV2 触发 2Fh = 触发比较器 A 30h = 触发比较器 B 31h = 已完成 ADC 转换 32h = TDC 已完成或超时 33h = AUX 定时器 0 事件 34h = AUX 定时器 1 事件 35h = BATMON 温度更新事件 36h = BATMON 电压更新事件 37h = 触发比较器 B。 直接来自 AUX 比较器 B 的异步信号，与在 AUX 中同步的 AUX_COMPB 相反 38h = 不触发比较器 B。 直接来自 AUX 比较器 B（反相）的异步信号，与在 AUX 中同步的 AUX_COMPB 相反 3Fh = 无事件，总为低电平
7-6	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 5-10. MCUWUSEL 寄存器字段说明（续）

位	字段	类型	复位	说明
5-0	WU0_EV	R/W	3Fh	MCU 唤醒源#0 AON 事件源选择路由到 AON_PMCTRL 的 8 个事件中的 1 个，以从关断或断电状态唤醒 MCU 域。 注意： 0h = 来自 DIO 的边沿检测 IO 事件，其在 [MCU_IOC:IOCFGx.IOEV_MCU_WU_EN] 中启用了对 IOEV_MCU_WU 的贡献 1h = 来自 AUX TImer2 的事件 0 2h = 来自 AUX TImer2 的事件 1 3h = 来自 AUX TImer2 的事件 2 4h = 来自 AUX TImer2 的事件 3 5h = BATMON 事件：电池电平高于上限 6h = BATMON 事件：电池电平低于下限 7h = BATMON 事件：温度水平高于上限 8h = BATMON 事件： 温度低于下限 9h = 来自 BATMON 的组合事件 20h = 任何 PAD 上的边缘检测 23h = RTC 通道 0 事件 24h = RTC 通道 1 事件 25h = RTC 通道 2 事件 26h = RTC 通道 0 - 显示的事件 27h = RTC 通道 1 - 显示的事件 28h = RTC 通道 2 - 显示的事件 29h = RTC 组合的显示事件 2Ah = RTC 更新时钟周期（16 kHz 信号，即事件线每 32 kHz 时钟周期切换一次值） 2Bh = JTAG 生成的事件 2Ch = AUX 软件触发事件 #0. 由 AUX_EVCTL:SWEVSET.SWEV0 触发 2Dh = AUX 软件触发事件 #1。 由 AUX_EVCTL:SWEVSET.SWEV1 触发 2Eh = AUX 软件触发事件 #2。 由 AUX_EVCTL:SWEVSET.SWEV2 触发 2Fh = 触发比较器 A 30h = 触发比较器 B 31h = 已完成 ADC 转换 32h = TDC 已完成或超时 33h = AUX 定时器 0 事件 34h = AUX 定时器 1 事件 35h = BATMON 温度更新事件 36h = BATMON 电压更新事件 37h = 触发比较器 B。 直接来自 AUX 比较器 B 的异步信号，与在 AUX 中同步的 AUX_COMPB 相反 38h = 不触发比较器 B。 直接来自 AUX 比较器 B（反相）的异步信号，与在 AUX 中同步的 AUX_COMPB 相反 3Fh = 无事件，总为低电平

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.1.2 MCUWUSEL1 寄存器（偏移 = 4h）[复位 = 3F3F3F3Fh]

图 5-7 展示了 MCUWUSEL1，表 5-11 对其进行了介绍。

返回到汇总表。

MCU 唤醒选择器
此寄存器包含指向 8 个事件（事件 4 至 7）中 4 个的指针，这些事件路由到 AON_PMCTRL 作为 MCU 的唤醒源。当 8 个所选事件中的任何一个生效时，AON_PMCTRL 将启动 MCU 域的唤醒序列。唤醒序列将保证 MCU 电源开关打开、LDO 资源可用、SCLK_HF 可用并且被选为 MCU 的时钟源。
注意：在 MCU 请求断电之前，需要在 AON_EVENT 中设置唤醒事件（PRCM 请求 uLDO，请参阅 PRCM:VDCTL.ULDO 中的条件）。

图 5-7. MCUWUSEL1 寄存器

31	30	29	28	27	26	25	24
RESERVED		WU7_EV					
R-0h		R/W-3Fh					
23	22	21	20	19	18	17	16
RESERVED		WU6_EV					
R-0h		R/W-3Fh					
15	14	13	12	11	10	9	8
RESERVED		WU5_EV					
R-0h		R/W-3Fh					
7	6	5	4	3	2	1	0
RESERVED		WU4_EV					
R-0h		R/W-3Fh					

表 5-11. MCUWUSEL1 寄存器字段说明

位	字段	类型	复位	说明
31-30	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 5-11. MCUWUSEL1 寄存器字段说明（续）

位	字段	类型	复位	说明
29-24	WU7_EV	R/W	3Fh	MCU 唤醒源#7 AON 事件源选择路由到 AON_PMCTRL 的 8 个事件中的 1 个，以从关断或断电状态唤醒 MCU 域。 注意： 0h = 来自 DIO 的边沿检测 IO 事件，其在 [MCU_IOC:IOCFGx.IOEV_MCU_WU_EN] 中启用了对 IOEV_MCU_WU 的贡献 1h = 来自 AUX TImer2 的事件 0 2h = 来自 AUX TImer2 的事件 1 3h = 来自 AUX TImer2 的事件 2 4h = 来自 AUX TImer2 的事件 3 5h = BATMON 事件：电池电平高于上限 6h = BATMON 事件：电池电平低于下限 7h = BATMON 事件：温度水平高于上限 8h = BATMON 事件： 温度低于下限 9h = 来自 BATMON 的组合事件 20h = 任何 PAD 上的边缘检测 23h = RTC 通道 0 事件 24h = RTC 通道 1 事件 25h = RTC 通道 2 事件 26h = RTC 通道 0 - 显示的事件 27h = RTC 通道 1 - 显示的事件 28h = RTC 通道 2 - 显示的事件 29h = RTC 组合的显示事件 2Ah = RTC 更新时钟周期（16 kHz 信号，即事件线每 32 kHz 时钟周期切换一次值） 2Bh = JTAG 生成的事件 2Ch = AUX 软件触发事件 #0. 由 AUX_EVCTL:SWEVSET.SWEV0 触发 2Dh = AUX 软件触发事件 #1。 由 AUX_EVCTL:SWEVSET.SWEV1 触发 2Eh = AUX 软件触发事件 #2。 由 AUX_EVCTL:SWEVSET.SWEV2 触发 2Fh = 触发比较器 A 30h = 触发比较器 B 31h = 已完成 ADC 转换 32h = TDC 已完成或超时 33h = AUX 定时器 0 事件 34h = AUX 定时器 1 事件 35h = BATMON 温度更新事件 36h = BATMON 电压更新事件 37h = 触发比较器 B。 直接来自 AUX 比较器 B 的异步信号，与在 AUX 中同步的 AUX_COMPB 相反 38h = 不触发比较器 B。 直接来自 AUX 比较器 B（反相）的异步信号，与在 AUX 中同步的 AUX_COMPB 相反 3Fh = 无事件，总为低电平
23-22	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 5-11. MCUWUSEL1 寄存器字段说明（续）

位	字段	类型	复位	说明
21-16	WU6_EV	R/W	3Fh	MCU 唤醒源#6 AON 事件源选择路由到 AON_PMCTRL 的 8 个事件中的 1 个，以从关断或断电状态唤醒 MCU 域。 注意： 0h = 来自 DIO 的边沿检测 IO 事件，其在 [MCU_IOC:IOCFGx.IOEV_MCU_WU_EN] 中启用了对 IOEV_MCU_WU 的贡献 1h = 来自 AUX TImer2 的事件 0 2h = 来自 AUX TImer2 的事件 1 3h = 来自 AUX TImer2 的事件 2 4h = 来自 AUX TImer2 的事件 3 5h = BATMON 事件：电池电平高于上限 6h = BATMON 事件：电池电平低于下限 7h = BATMON 事件：温度水平高于上限 8h = BATMON 事件： 温度低于下限 9h = 来自 BATMON 的组合事件 20h = 任何 PAD 上的边缘检测 23h = RTC 通道 0 事件 24h = RTC 通道 1 事件 25h = RTC 通道 2 事件 26h = RTC 通道 0 - 显示的事件 27h = RTC 通道 1 - 显示的事件 28h = RTC 通道 2 - 显示的事件 29h = RTC 组合的显示事件 2Ah = RTC 更新时钟周期（16 kHz 信号，即事件线每 32 kHz 时钟周期切换一次值） 2Bh = JTAG 生成的事件 2Ch = AUX 软件触发事件 #0. 由 AUX_EVCTL:SWEVSET.SWEV0 触发 2Dh = AUX 软件触发事件 #1。 由 AUX_EVCTL:SWEVSET.SWEV1 触发 2Eh = AUX 软件触发事件 #2。 由 AUX_EVCTL:SWEVSET.SWEV2 触发 2Fh = 触发比较器 A 30h = 触发比较器 B 31h = 已完成 ADC 转换 32h = TDC 已完成或超时 33h = AUX 定时器 0 事件 34h = AUX 定时器 1 事件 35h = BATMON 温度更新事件 36h = BATMON 电压更新事件 37h = 触发比较器 B。 直接来自 AUX 比较器 B 的异步信号，与在 AUX 中同步的 AUX_COMPB 相反 38h = 不触发比较器 B。 直接来自 AUX 比较器 B（反相）的异步信号，与在 AUX 中同步的 AUX_COMPB 相反 3Fh = 无事件，总为低电平
15-14	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 5-11. MCUWUSEL1 寄存器字段说明（续）

位	字段	类型	复位	说明
13-8	WU5_EV	R/W	3Fh	MCU 唤醒源#5 AON 事件源选择路由到 AON_PMCTRL 的 8 个事件中的 1 个，以从关断或断电状态唤醒 MCU 域。 注意： 0h = 来自 DIO 的边沿检测 IO 事件，其在 [MCU_IOC:IOCFGx.IOEV_MCU_WU_EN] 中启用了对 IOEV_MCU_WU 的贡献 1h = 来自 AUX TImer2 的事件 0 2h = 来自 AUX TImer2 的事件 1 3h = 来自 AUX TImer2 的事件 2 4h = 来自 AUX TImer2 的事件 3 5h = BATMON 事件：电池电平高于上限 6h = BATMON 事件：电池电平低于下限 7h = BATMON 事件：温度水平高于上限 8h = BATMON 事件： 温度低于下限 9h = 来自 BATMON 的组合事件 20h = 任何 PAD 上的边缘检测 23h = RTC 通道 0 事件 24h = RTC 通道 1 事件 25h = RTC 通道 2 事件 26h = RTC 通道 0 - 显示的事件 27h = RTC 通道 1 - 显示的事件 28h = RTC 通道 2 - 显示的事件 29h = RTC 组合的显示事件 2Ah = RTC 更新时钟周期（16 kHz 信号，即事件线每 32 kHz 时钟周期切换一次值） 2Bh = JTAG 生成的事件 2Ch = AUX 软件触发事件 #0. 由 AUX_EVCTL:SWEVSET.SWEV0 触发 2Dh = AUX 软件触发事件 #1。 由 AUX_EVCTL:SWEVSET.SWEV1 触发 2Eh = AUX 软件触发事件 #2。 由 AUX_EVCTL:SWEVSET.SWEV2 触发 2Fh = 触发比较器 A 30h = 触发比较器 B 31h = 已完成 ADC 转换 32h = TDC 已完成或超时 33h = AUX 定时器 0 事件 34h = AUX 定时器 1 事件 35h = BATMON 温度更新事件 36h = BATMON 电压更新事件 37h = 触发比较器 B。 直接来自 AUX 比较器 B 的异步信号，与在 AUX 中同步的 AUX_COMPB 相反 38h = 不触发比较器 B。 直接来自 AUX 比较器 B（反相）的异步信号，与在 AUX 中同步的 AUX_COMPB 相反 3Fh = 无事件，总为低电平
7-6	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 5-11. MCUWUSEL1 寄存器字段说明（续）

位	字段	类型	复位	说明
5-0	WU4_EV	R/W	3Fh	MCU 唤醒源#4 AON 事件源选择路由到 AON_PMCTRL 的 8 个事件中的 1 个，以从关断或断电状态唤醒 MCU 域。 注意： 0h = 来自 DIO 的边沿检测 IO 事件，其在 [MCU_IOC:IOCFGx.IOEV_MCU_WU_EN] 中启用了对 IOEV_MCU_WU 的贡献 1h = 来自 AUX TImer2 的事件 0 2h = 来自 AUX TImer2 的事件 1 3h = 来自 AUX TImer2 的事件 2 4h = 来自 AUX TImer2 的事件 3 5h = BATMON 事件：电池电平高于上限 6h = BATMON 事件：电池电平低于下限 7h = BATMON 事件：温度水平高于上限 8h = BATMON 事件： 温度低于下限 9h = 来自 BATMON 的组合事件 20h = 任何 PAD 上的边缘检测 23h = RTC 通道 0 事件 24h = RTC 通道 1 事件 25h = RTC 通道 2 事件 26h = RTC 通道 0 - 显示的事件 27h = RTC 通道 1 - 显示的事件 28h = RTC 通道 2 - 显示的事件 29h = RTC 组合的显示事件 2Ah = RTC 更新时钟周期（16 kHz 信号，即事件线每 32 kHz 时钟周期切换一次值） 2Bh = JTAG 生成的事件 2Ch = AUX 软件触发事件 #0. 由 AUX_EVCTL:SWEVSET.SWEV0 触发 2Dh = AUX 软件触发事件 #1。 由 AUX_EVCTL:SWEVSET.SWEV1 触发 2Eh = AUX 软件触发事件 #2。 由 AUX_EVCTL:SWEVSET.SWEV2 触发 2Fh = 触发比较器 A 30h = 触发比较器 B 31h = 已完成 ADC 转换 32h = TDC 已完成或超时 33h = AUX 定时器 0 事件 34h = AUX 定时器 1 事件 35h = BATMON 温度更新事件 36h = BATMON 电压更新事件 37h = 触发比较器 B。 直接来自 AUX 比较器 B 的异步信号，与在 AUX 中同步的 AUX_COMPB 相反 38h = 不触发比较器 B。 直接来自 AUX 比较器 B（反相）的异步信号，与在 AUX 中同步的 AUX_COMPB 相反 3Fh = 无事件，总为低电平

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.1.3 EVTOMCUSEL 寄存器（偏移 = 8h）[复位 = 002B2B2Bh]

图 5-8 展示了 EVTOMCUSEL，表 5-12 中对此进行了介绍。

返回到汇总表。

MCU 事件结构的事件选择器
此寄存器含有路由到 MCU 事件结构 EVENT 的 3 个 AON 事件的指针

图 5-8. EVTOMCUSEL 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED		AON_PROG2_EV					
R-0h		R/W-2Bh					
15	14	13	12	11	10	9	8
RESERVED		AON_PROG1_EV					
R-0h		R/W-2Bh					
7	6	5	4	3	2	1	0
RESERVED		AON_PROG0_EV					
R-0h		R/W-2Bh					

表 5-12. EVTOMCUSEL 寄存器字段说明

位	字段	类型	复位	说明
31-22	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 5-12. EVTOMCUSEL 寄存器字段说明（续）

位	字段	类型	复位	说明
21-16	AON_PROG2_EV	R/W	2Bh	AON_PROG2 事件的事件选择器。 AON 事件源 id#选择作为 AON_PROG2 事件路由到 EVENT 的事件。 0h = 来自 DIO 的边沿检测 IO 事件，其在 [MCU_IOC:IOCFGx.IOEV_AON_PROG2_EN] 中启用了对 IOEV_AON_PROG2 的贡献 1h = 来自 AUX TImer2 的事件 0 2h = 来自 AUX TImer2 的事件 1 3h = 来自 AUX TImer2 的事件 2 4h = 来自 AUX TImer2 的事件 3 5h = BATMON 事件：电池电平高于上限 6h = BATMON 事件：电池电平低于下限 7h = BATMON 事件：温度水平高于上限 8h = BATMON 事件： 温度低于下限 9h = 来自 BATMON 的组合事件 20h = 任何 PAD 上的边缘检测 23h = RTC 通道 0 事件 24h = RTC 通道 1 事件 25h = RTC 通道 2 事件 26h = RTC 通道 0 - 显示的事件 27h = RTC 通道 1 - 显示的事件 28h = RTC 通道 2 - 显示的事件 29h = RTC 组合的显示事件 2Ah = RTC 更新时钟周期（16 kHz 信号，即事件线每 32 kHz 时钟周期切换一次值） 2Bh = JTAG 生成的事件 2Ch = AUX 软件触发事件 #0. 由 AUX_EVCTL:SWEVSET.SWEV0 触发 2Dh = AUX 软件触发事件 #1。 由 AUX_EVCTL:SWEVSET.SWEV1 触发 2Eh = AUX 软件触发事件 #2。 由 AUX_EVCTL:SWEVSET.SWEV2 触发 2Fh = 触发比较器 A 30h = 触发比较器 B 31h = 已完成 ADC 转换 32h = TDC 已完成或超时 33h = AUX 定时器 0 事件 34h = AUX 定时器 1 事件 35h = BATMON 温度更新事件 36h = BATMON 电压更新事件 37h = 触发比较器 B。 直接来自 AUX 比较器 B 的异步信号，与在 AUX 中同步的 AUX_COMPB 相反 38h = 不触发比较器 B。 直接来自 AUX 比较器 B（反相）的异步信号，与在 AUX 中同步的 AUX_COMPB 相反 3Fh = 无事件，总为低电平
15-14	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 5-12. EVTOMCUSEL 寄存器字段说明（续）

位	字段	类型	复位	说明
13-8	AON_PROG1_EV	R/W	2Bh	AON_PROG1 事件的事件选择器。 AON 事件源 id#选择作为 AON_PROG1 事件路由到 EVENT 的事件。 0h = 来自 DIO 的边沿检测 IO 事件，其在 [MCU_IOC:IOCFGx.IOEV_AON_PROG1_EN] 中启用了对 IOEV_AON_PROG1 的贡献 1h = 来自 AUX TImer2 的事件 0 2h = 来自 AUX TImer2 的事件 1 3h = 来自 AUX TImer2 的事件 2 4h = 来自 AUX TImer2 的事件 3 5h = BATMON 事件：电池电平高于上限 6h = BATMON 事件：电池电平低于下限 7h = BATMON 事件：温度水平高于上限 8h = BATMON 事件： 温度水平低于下限 9h = 来自 BATMON 的组合事件 20h = 任何 PAD 上的边缘检测 23h = RTC 通道 0 事件 24h = RTC 通道 1 事件 25h = RTC 通道 2 事件 26h = RTC 通道 0 - 延迟事件 27h = 0 28h = 0 29h = 0 2Ah = 0 2Bh = 0 2Ch = 0 2Dh = 0 2Eh = 0 2Fh = 0 30h = 0 31h = 0 32h = 0 33h = 0 34h = 0 35h = 0 36h = 0 37h = 0 38h = 0 3Fh = 0
7-6	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 5-12. EVTOMCUSEL 寄存器字段说明（续）

位	字段	类型	复位	说明
5-0	AON_PROG0_EV	R/W	2Bh	AON_PROG0 事件的事件选择器。 AON 事件源 id#选择作为 AON_PROG0 事件路由到 EVENT 的事件。 0h = 来自 DIO 的边沿检测 IO 事件，其在 [MCU_IOC:IOCFGx.IOEV_AON_PROG0_EN] 中启用了对 IOEV_AON_PROG0 的贡献 1h = 0 2h = 0 3h = 0 4h = 0 5h = 0 6h = 0 7h = 0 8h = 0 9h = 0 20h = 0 23h = 0 24h = 0 25h = 0 26h = 0 27h = 0 28h = 0 29h = 0 2Ah = 0 2Bh = 0 2Ch = 0 2Dh = 0 2Eh = 0 2Fh = 0 30h = 0 31h = 0 32h = 0 33h = 0 34h = 0 35h = 0 36h = 0 37h = 0 38h = 0 3Fh = 0

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.1.4 RTCSEL 寄存器（偏移 = Ch）[复位 = 0000003Fh]

图 5-9 展示了 RTCSEL，表 5-13 中对此进行了介绍。

返回到汇总表。

AON_RTC 的 RTC 捕获事件选择器
此寄存器含有一个指针，用来选择用于 RTC 捕获的 AON 事件。请参阅 AON_RTC:CH1CAPT

图 5-9. RTCSEL 寄存器

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										RTC_CH1_CAPT_EV					
R-0h										R/W-3Fh					

表 5-13. RTCSEL 寄存器字段说明

位	字段	类型	复位	说明
31-6	RESERVED	R	0h	保留
5-0	RTC_CH1_CAPT_EV	R/W	3Fh	馈送到 AON_RTC 的 RTCSEL 事件的 AON 事件源 id#。请参阅 AON_RTC:CH1CAPT 0h = 来自 DIO 的边沿检测 IO 事件，其在 [MCU_IOC:IOCFGx.IOEV_RTC_EN] 中启用了对 IOEV_RTC 的贡献 1h = 0 2h = 0 3h = 0 4h = 0 5h = 0 6h = 0 7h = 0 8h = 0 9h = 0 20h = 0 23h = 0 24h = 0 25h = 0 26h = 0 27h = 0 28h = 0 29h = 0 2Ah = 0 2Bh = 0 2Ch = 0 2Dh = 0 2Eh = 0 2Fh = 0 30h = 0 31h = 0 32h = 0 33h = 0 34h = 0 35h = 0 36h = 0 37h = 0 38h = 0 3Fh = 0

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2 EVENT 寄存器

表 5-14 列出了 EVENT 寄存器的存储器映射寄存器。表 5-14 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

表 5-14. EVENT 寄存器

偏移	首字母缩写词	寄存器名称	部分
0h	CPUIRQSEL0	CPU 中断 0 的输出选择	CPUIRQSEL0 寄存器（偏移 = 0h）[复位 = 00000004h]
4h	CPUIRQSEL1	CPU 中断 1 的输出选择	CPUIRQSEL1 寄存器（偏移 = 4h）[复位 = 00000009h]
8h	CPUIRQSEL2	CPU 中断 2 的输出选择	CPUIRQSEL2 寄存器（偏移 = 8h）[复位 = 0000001Eh]
Ch	CPUIRQSEL3	CPU 中断 3 的输出选择	CPUIRQSEL3 寄存器（偏移 = Ch）[复位 = 0000001Fh]
10h	CPUIRQSEL4	CPU 中断 4 的输出选择	CPUIRQSEL4 寄存器（偏移 = 10h）[复位 = 00000007h]
14h	CPUIRQSEL5	CPU 中断 5 的输出选择	CPUIRQSEL5 寄存器（偏移 = 14h）[复位 = 00000024h]
18h	CPUIRQSEL6	CPU 中断 6 的输出选择	CPUIRQSEL6 寄存器（偏移 = 18h）[复位 = 0000001Ch]
1Ch	CPUIRQSEL7	CPU 中断 7 的输出选择	CPUIRQSEL7 寄存器（偏移 = 1Ch）[复位 = 00000022h]
20h	CPUIRQSEL8	CPU 中断 8 的输出选择	CPUIRQSEL8 寄存器（偏移 = 20h）[复位 = 00000023h]
24h	CPUIRQSEL9	CPU 中断 9 的输出选择	CPUIRQSEL9 寄存器（偏移 = 24h）[复位 = 0000001Bh]
28h	CPUIRQSEL10	CPU 中断 10 的输出选择	CPUIRQSEL10 寄存器（偏移 = 28h）[复位 = 0000001Ah]
2Ch	CPUIRQSEL11	CPU 中断 11 的输出选择	CPUIRQSEL11 寄存器（偏移 = 2Ch）[复位 = 00000019h]
30h	CPUIRQSEL12	CPU 中断 12 的输出选择	CPUIRQSEL12 寄存器（偏移 = 30h）[复位 = 00000008h]
34h	CPUIRQSEL13	CPU 中断 13 的输出选择	CPUIRQSEL13 寄存器（偏移 = 34h）[复位 = 0000001Dh]
38h	CPUIRQSEL14	CPU 中断 14 的输出选择	CPUIRQSEL14 寄存器（偏移 = 38h）[复位 = 00000018h]
3Ch	CPUIRQSEL15	CPU 中断 15 的输出选择	CPUIRQSEL15 寄存器（偏移 = 3Ch）[复位 = 00000010h]
40h	CPUIRQSEL16	CPU 中断 16 的输出选择	CPUIRQSEL16 寄存器（偏移 = 40h）[复位 = 00000011h]
44h	CPUIRQSEL17	CPU 中断 17 的输出选择	CPUIRQSEL17 寄存器（偏移 = 44h）[复位 = 00000012h]
48h	CPUIRQSEL18	CPU 中断 18 的输出选择	CPUIRQSEL18 寄存器（偏移 = 48h）[复位 = 00000013h]
4Ch	CPUIRQSEL19	CPU 中断 19 的输出选择	CPUIRQSEL19 寄存器（偏移 = 4Ch）[复位 = 0000000Ch]
50h	CPUIRQSEL20	CPU 中断 20 的输出选择	CPUIRQSEL20 寄存器（偏移 = 50h）[复位 = 0000000Dh]
54h	CPUIRQSEL21	CPU 中断 21 的输出选择	CPUIRQSEL21 寄存器（偏移 = 54h）[复位 = 0000000Eh]
58h	CPUIRQSEL22	CPU 中断 22 的输出选择	CPUIRQSEL22 寄存器（偏移 = 58h）[复位 = 0000000Fh]

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 5-14. EVENT 寄存器（续）

偏移	首字母缩写词	寄存器名称	部分
5Ch	CPUIRQSEL23	CPU 中断 23 的输出选择	CPUIRQSEL23 寄存器（偏移 = 5Ch）[复位 = 0000005Dh]
60h	CPUIRQSEL24	CPU 中断 24 的输出选择	CPUIRQSEL24 寄存器（偏移 = 60h）[复位 = 00000027h]
64h	CPUIRQSEL25	CPU 中断 25 的输出选择	CPUIRQSEL25 寄存器（偏移 = 64h）[复位 = 00000026h]
68h	CPUIRQSEL26	CPU 中断 26 的输出选择	CPUIRQSEL26 寄存器（偏移 = 68h）[复位 = 00000015h]
6Ch	CPUIRQSEL27	CPU 中断 27 的输出选择	CPUIRQSEL27 寄存器（偏移 = 6Ch）[复位 = 00000064h]
70h	CPUIRQSEL28	CPU 中断 28 的输出选择	CPUIRQSEL28 寄存器（偏移 = 70h）[复位 = 0000000Bh]
74h	CPUIRQSEL29	CPU 中断 29 的输出选择	CPUIRQSEL29 寄存器（偏移 = 74h）[复位 = 00000001h]
78h	CPUIRQSEL30	CPU 中断 30 的输出选择	CPUIRQSEL30 寄存器（偏移 = 78h）[复位 = 00000000h]
7Ch	CPUIRQSEL31	CPU 中断 31 的输出选择	CPUIRQSEL31 寄存器（偏移 = 7Ch）[复位 = 0000006Ah]
80h	CPUIRQSEL32	CPU 中断 32 的输出选择	CPUIRQSEL32 寄存器（偏移 = 80h）[复位 = 00000073h]
84h	CPUIRQSEL33	CPU 中断 33 的输出选择	CPUIRQSEL33 寄存器（偏移 = 84h）[复位 = 00000068h]
88h	CPUIRQSEL34	CPU 中断 34 的输出选择	CPUIRQSEL34 寄存器（偏移 = 88h）[复位 = 00000006h]
8Ch	CPUIRQSEL35	CPU 中断 35 的输出选择	CPUIRQSEL35 寄存器（偏移 = 8Ch）[复位 = 00000038h]
90h	CPUIRQSEL36	CPU 中断 36 的输出选择	CPUIRQSEL36 寄存器（偏移 = 90h）[复位 = 00000025h]
94h	CPUIRQSEL37	CPU 中断 37 的输出选择	CPUIRQSEL37 寄存器（偏移 = 94h）[复位 = 00000005h]
100h	RFCSEL0	RFC 事件 0 的输出选择	RFCSEL0 寄存器（偏移 = 100h）[复位 = 0000003Dh]
104h	RFCSEL1	RFC 事件 1 的输出选择	RFCSEL1 寄存器（偏移 = 104h）[复位 = 0000003Eh]
108h	RFCSEL2	RFC 事件 2 的输出选择	RFCSEL2 寄存器（偏移 = 108h）[复位 = 0000003Fh]
10Ch	RFCSEL3	RFC 事件 3 的输出选择	RFCSEL3 寄存器（偏移 = 10Ch）[复位 = 00000040h]
110h	RFCSEL4	RFC 事件 4 的输出选择	RFCSEL4 寄存器（偏移 = 110h）[复位 = 00000041h]
114h	RFCSEL5	RFC 事件 5 的输出选择	RFCSEL5 寄存器（偏移 = 114h）[复位 = 00000042h]
118h	RFCSEL6	RFC 事件 6 的输出选择	RFCSEL6 寄存器（偏移 = 118h）[复位 = 00000043h]
11Ch	RFCSEL7	RFC 事件 7 的输出选择	RFCSEL7 寄存器（偏移 = 11Ch）[复位 = 00000044h]
120h	RFCSEL8	RFC 事件 8 的输出选择	RFCSEL8 寄存器（偏移 = 120h）[复位 = 00000077h]
124h	RFCSEL9	RFC 事件 9 的输出选择	RFCSEL9 寄存器（偏移 = 124h）[复位 = 00000002h]

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 5-14. EVENT 寄存器（续）

偏移	首字母缩写词	寄存器名称	部分
200h	GPT0ACAPTSEL	GPT0 0 的输出选择	GPT0ACAPTSEL 寄存器（偏移 = 200h）[复位 = 00000055h]
204h	GPT0BCAPTSEL	GPT0 1 的输出选择	GPT0BCAPTSEL 寄存器（偏移 = 204h）[复位 = 00000056h]
300h	GPT1ACAPTSEL	GPT1 0 的输出选择	GPT1ACAPTSEL 寄存器（偏移 = 300h）[复位 = 00000057h]
304h	GPT1BCAPTSEL	GPT1 1 的输出选择	GPT1BCAPTSEL 寄存器（偏移 = 304h）[复位 = 00000058h]
400h	GPT2ACAPTSEL	GPT2 0 的输出选择	GPT2ACAPTSEL 寄存器（偏移 = 400h）[复位 = 00000059h]
404h	GPT2BCAPTSEL	GPT2 1 的输出选择	GPT2BCAPTSEL 寄存器（偏移 = 404h）[复位 = 0000005Ah]
508h	UDMACH1SSEL	DMA 通道 1 SREQ 的输出选择	UDMACH1SSEL 寄存器（偏移 = 508h）[复位 = 00000031h]
50Ch	UDMACH1BSEL	DMA 通道 1 REQ 的输出选择	UDMACH1BSEL 寄存器（偏移 = 50Ch）[复位 = 00000030h]
510h	UDMACH2SSEL	DMA 通道 2 SREQ 的输出选择	UDMACH2SSEL 寄存器（偏移 = 510h）[复位 = 00000033h]
514h	UDMACH2BSEL	DMA 通道 2 REQ 的输出选择	UDMACH2BSEL 寄存器（偏移 = 514h）[复位 = 00000032h]
518h	UDMACH3SSEL	DMA 通道 3 SREQ 的输出选择	UDMACH3SSEL 寄存器（偏移 = 518h）[复位 = 00000029h]
51Ch	UDMACH3BSEL	DMA 通道 3 REQ 的输出选择	UDMACH3BSEL 寄存器（偏移 = 51Ch）[复位 = 00000028h]
520h	UDMACH4SSEL	DMA 通道 4 SREQ 的输出选择	UDMACH4SSEL 寄存器（偏移 = 520h）[复位 = 0000002Bh]
524h	UDMACH4BSEL	DMA 通道 4 REQ 的输出选择	UDMACH4BSEL 寄存器（偏移 = 524h）[复位 = 0000002Ah]
528h	UDMACH5SSEL	DMA 通道 5 SREQ 的输出选择	UDMACH5SSEL 寄存器（偏移 = 528h）[复位 = 00000035h]
52Ch	UDMACH5BSEL	DMA 通道 5 REQ 的输出选择	UDMACH5BSEL 寄存器（偏移 = 52Ch）[复位 = 00000034h]
530h	UDMACH6SSEL	DMA 通道 6 SREQ 的输出选择	UDMACH6SSEL 寄存器（偏移 = 530h）[复位 = 00000037h]
534h	UDMACH6BSEL	DMA 通道 6 REQ 的输出选择	UDMACH6BSEL 寄存器（偏移 = 534h）[复位 = 00000036h]
538h	UDMACH7SSEL	DMA 通道 7 SREQ 的输出选择	UDMACH7SSEL 寄存器（偏移 = 538h）[复位 = 00000075h]
53Ch	UDMACH7BSEL	DMA 通道 7 REQ 的输出选择	UDMACH7BSEL 寄存器（偏移 = 53Ch）[复位 = 00000076h]
540h	UDMACH8SSEL	DMA 通道 8 SREQ 的输出选择	UDMACH8SSEL 寄存器（偏移 = 540h）[复位 = 00000074h]
544h	UDMACH8BSEL	DMA 通道 8 REQ 的输出选择	UDMACH8BSEL 寄存器（偏移 = 544h）[复位 = 00000074h]
548h	UDMACH9SSEL	DMA 通道 9 SREQ 的输出选择	UDMACH9SSEL 寄存器（偏移 = 548h）[复位 = 00000045h]
54Ch	UDMACH9BSEL	DMA 通道 9 REQ 的输出选择	UDMACH9BSEL 寄存器（偏移 = 54Ch）[复位 = 0000004Dh]
550h	UDMACH10SSEL	DMA 通道 10 SREQ 的输出选择	UDMACH10SSEL 寄存器（偏移 = 550h）[复位 = 00000046h]

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 5-14. EVENT 寄存器（续）

偏移	首字母缩写词	寄存器名称	部分
554h	UDMACH10BSEL	DMA 通道 10 REQ 的输出选择	UDMACH10BSEL 寄存器（偏移 = 554h）[复位 = 0000004Eh]
558h	UDMACH11SSEL	DMA 通道 11 SREQ 的输出选择	UDMACH11SSEL 寄存器（偏移 = 558h）[复位 = 00000047h]
55Ch	UDMACH11BSEL	DMA 通道 11 REQ 的输出选择	UDMACH11BSEL 寄存器（偏移 = 55Ch）[复位 = 0000004Fh]
560h	UDMACH12SSEL	DMA 通道 12 SREQ 的输出选择	UDMACH12SSEL 寄存器（偏移 = 560h）[复位 = 00000048h]
564h	UDMACH12BSEL	DMA 通道 12 REQ 的输出选择	UDMACH12BSEL 寄存器（偏移 = 564h）[复位 = 00000050h]
56Ch	UDMACH13BSEL	DMA 通道 13 REQ 的输出选择	UDMACH13BSEL 寄存器（偏移 = 56Ch）[复位 = 00000003h]
574h	UDMACH14BSEL	DMA 通道 14 REQ 的输出选择	UDMACH14BSEL 寄存器（偏移 = 574h）[复位 = 00000001h]
57Ch	UDMACH15BSEL	DMA 通道 15 REQ 的输出选择	UDMACH15BSEL 寄存器（偏移 = 57Ch）[复位 = 00000007h]
580h	UDMACH16SSEL	DMA 通道 16 SREQ 的输出选择	UDMACH16SSEL 寄存器（偏移 = 580h）[复位 = 0000002Dh]
584h	UDMACH16BSEL	DMA 通道 16 REQ 的输出选择	UDMACH16BSEL 寄存器（偏移 = 584h）[复位 = 0000002Ch]
588h	UDMACH17SSEL	DMA 通道 17 SREQ 的输出选择	UDMACH17SSEL 寄存器（偏移 = 588h）[复位 = 0000002Fh]
58Ch	UDMACH17BSEL	DMA 通道 17 REQ 的输出选择	UDMACH17BSEL 寄存器（偏移 = 58Ch）[复位 = 0000002Eh]
5A8h	UDMACH21SSEL	DMA 通道 21 SREQ 的输出选择	UDMACH21SSEL 寄存器（偏移 = 5A8h）[复位 = 00000064h]
5ACh	UDMACH21BSEL	DMA 通道 21 REQ 的输出选择	UDMACH21BSEL 寄存器（偏移 = 5ACh）[复位 = 00000064h]
5B0h	UDMACH22SSEL	DMA 通道 22 SREQ 的输出选择	UDMACH22SSEL 寄存器（偏移 = 5B0h）[复位 = 00000065h]
5B4h	UDMACH22BSEL	DMA 通道 22 REQ 的输出选择	UDMACH22BSEL 寄存器（偏移 = 5B4h）[复位 = 00000065h]
5B8h	UDMACH23SSEL	DMA 通道 23 SREQ 的输出选择	UDMACH23SSEL 寄存器（偏移 = 5B8h）[复位 = 00000066h]
5BCh	UDMACH23BSEL	DMA 通道 23 REQ 的输出选择	UDMACH23BSEL 寄存器（偏移 = 5BCh）[复位 = 00000066h]
5C0h	UDMACH24SSEL	DMA 通道 24 SREQ 的输出选择	UDMACH24SSEL 寄存器（偏移 = 5C0h）[复位 = 00000067h]
5C4h	UDMACH24BSEL	DMA 通道 24 REQ 的输出选择	UDMACH24BSEL 寄存器（偏移 = 5C4h）[复位 = 00000067h]
600h	GPT3ACAPTSEL	GPT3 0 的输出选择	GPT3ACAPTSEL 寄存器（偏移 = 600h）[复位 = 0000005Bh]
604h	GPT3BCAPTSEL	GPT3 1 的输出选择	GPT3BCAPTSEL 寄存器（偏移 = 604h）[复位 = 0000005Ch]
700h	AUXSEL0	AUX 订阅者 0 的输出选择	AUXSEL0 寄存器（偏移 = 700h）[复位 = 00000010h]
800h	CM3NMISEL0	NMI 订阅者 0 的输出选择	CM3NMISEL0 寄存器（偏移 = 800h）[复位 = 00000063h]
900h	I2SSTMPSEL0	I2S 订阅者 0 的输出选择	I2SSTMPSEL0 寄存器（偏移 = 900h）[复位 = 0000005Fh]

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 5-14. EVENT 寄存器（续）

偏移	首字母缩写词	寄存器名称	部分
A00h	FRZSEL0	FRZ 订阅者的输出选择	FRZSEL0 寄存器（偏移 = A00h）[复位 = 00000078h]
F00h	SWEV	设置或清除软件事件	SWEV 寄存器（偏移 = F00h）[复位 = 00000000h]

复杂的位访问类型经过编码可适应小型表单元。表 5-15 展示了适用于此部分中访问类型的代码。

表 5-15. EVENT 访问类型代码

访问类型	代码	说明
读取类型		
R	R	读取
写入类型		
W	W	写入
复位或默认值		
-n		复位后的值或默认值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.1 CPUIRQSEL0 寄存器（偏移 = 0h）[复位 = 00000004h]

图 5-10 展示了 CPUIRQSEL0，表 5-16 对其进行了介绍。

返回到汇总表。

CPU 中断 0 的输出选择

图 5-10. CPUIRQSEL0 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-4h						

表 5-16. CPUIRQSEL0 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	4h	只读选择值 4h = 来自 IOC 的边沿检测事件。通过 IOC:IOCFGn.EDGE_IRQ_EN 和 IOC:IOCFGn.EDGE_DET 设置进行配置

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.2 CPUIRQSEL1 寄存器（偏移 = 4h）[复位 = 00000009h]

图 5-11 展示了 CPUIRQSEL1，表 5-17 对其进行了介绍。

返回到汇总表。

CPU 中断 1 的输出选择

图 5-11. CPUIRQSEL1 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-9h						

表 5-17. CPUIRQSEL1 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	9h	只读选择值 9h = 来自 I2C 的中断事件

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.3 CPUIRQSEL2 寄存器（偏移 = 8h）[复位 = 0000001Eh]

图 5-12 展示了 CPUIRQSEL2，表 5-18 对其进行了介绍。

返回到汇总表。

CPU 中断 2 的输出选择

图 5-12. CPUIRQSEL2 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-1Eh						

表 5-18. CPUIRQSEL2 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	1Eh	只读选择值 1Eh = CPE 生成的事件的组合中断。在这里对应的标志是 RFC_DBELL:RFCPEIFG。只有在 RFC_DBELL:RFCPEIFG 中通过 CPE1 选择的中断才能触发 RFC_CPE_1 事件

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.4 CPUIRQSEL3 寄存器（偏移 = Ch）[复位 = 0000001Fh]

图 5-13 展示了 CPUIRQSEL3，表 5-19 对其进行了介绍。

返回到汇总表。

CPU 中断 3 的输出选择

图 5-13. CPUIRQSEL3 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-1Fh						

表 5-19. CPUIRQSEL3 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	1Fh	只读选择值 1Fh = PKA 中断事件

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.5 CPUIRQSEL4 寄存器（偏移 = 10h）[复位 = 00000007h]

图 5-14 展示了 CPUIRQSEL4，表 5-20 对其进行了介绍。

返回到汇总表。

CPU 中断 4 的输出选择

图 5-14. CPUIRQSEL4 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-7h						

表 5-20. CPUIRQSEL4 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	7h	只读选择值 7h = 来自 AON_RTC 的事件，通过 AON_RTC:CTL.COMB_EV_MASK 设置来控制

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.6 CPUIRQSEL5 寄存器（偏移 = 14h）[复位 = 00000024h]

图 5-15 展示了 CPUIRQSEL5，表 5-21 对其进行了介绍。

返回到汇总表。

CPU 中断 5 的输出选择

图 5-15. CPUIRQSEL5 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-24h						

表 5-21. CPUIRQSEL5 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	24h	只读选择值 24h = UART0 组合中断，在此处找到的中断标志为 UART0:MIS

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.7 CPUIRQSEL6 寄存器（偏移 = 18h）[复位 = 0000001Ch]

图 5-16 展示了 CPUIRQSEL6，表 5-22 对其进行了介绍。

返回到汇总表。

CPU 中断 6 的输出选择

图 5-16. CPUIRQSEL6 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-1Ch						

表 5-22. CPUIRQSEL6 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	1Ch	只读选择值 1Ch = AUX 软件事件 0，由 AUX_EVCTL:SWEVSET.SWEV0 触发，也可用作 AUX_EVENT0 AON 唤醒事件。 MCU 域唤醒控制 AON_EVENT:MCUWUSEL

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.8 CPUIRQSEL7 寄存器（偏移 = 1Ch）[复位 = 00000022h]

图 5-17 展示了 CPUIRQSEL7，表 5-23 对其进行了介绍。

返回到汇总表。

CPU 中断 7 的输出选择

图 5-17. CPUIRQSEL7 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-22h						

表 5-23. CPUIRQSEL7 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	22h	只读选择值 22h = SSI0 组合中断，在此处找到的中断标志为 SSI0:MIS

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.9 CPUIRQSEL8 寄存器（偏移 = 20h）[复位 = 00000023h]

图 5-18 展示了 CPUIRQSEL8，表 5-24 对其进行了介绍。

返回到汇总表。

CPU 中断 8 的输出选择

图 5-18. CPUIRQSEL8 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-23h						

表 5-24. CPUIRQSEL8 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	23h	只读选择值 23h = SSI1 组合中断，在此处找到的中断标志为 SSI1:MIS

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.10 CPUIRQSEL9 寄存器（偏移 = 24h）[复位 = 0000001Bh]

图 5-19 展示了 CPUIRQSEL9，表 5-25 对其进行了介绍。

返回到汇总表。

CPU 中断 9 的输出选择

图 5-19. CPUIRQSEL9 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-1Bh						

表 5-25. CPUIRQSEL9 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	1Bh	只读选择值 1Bh = CPE 生成的事件的组合中断。在这里对应的标志是 RFC_DBELL:RFCPEIFG。只有在 RFC_DBELL:RFCPEIFG 中通过 CPE0 选择的中断才能触发 RFC_CPE_0 事件

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.11 CPUIRQSEL10 寄存器（偏移 = 28h）[复位 = 0000001Ah]

图 5-20 展示了 CPUIRQSEL10，表 5-26 对其进行了介绍。

返回到汇总表。

CPU 中断 10 的输出选择

图 5-20. CPUIRQSEL10 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-1Ah						

表 5-26. CPUIRQSEL10 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	1Ah	只读选择值 1Ah = 组合的 RFC 硬件中断，相应的标志在此处为 RFC_DBELL:RFHWIFG

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.12 CPUIRQSEL11 寄存器（偏移 = 2Ch）[复位 = 00000019h]

图 5-21 展示了 CPUIRQSEL11，表 5-27 对其进行了介绍。

返回到汇总表。

CPU 中断 11 的输出选择

图 5-21. CPUIRQSEL11 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-19h						

表 5-27. CPUIRQSEL11 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	19h	只读选择值 19h = RFC 门铃命令确认中断，相当于 RFC_DBELL:RFACKIFG.ACKFLAG

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.13 CPUIRQSEL12 寄存器（偏移 = 30h）[复位 = 00000008h]

图 5-22 展示了 CPUIRQSEL12，表 5-28 对其进行了介绍。

返回到汇总表。

CPU 中断 12 的输出选择

图 5-22. CPUIRQSEL12 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-8h						

表 5-28. CPUIRQSEL12 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	8h	只读选择值 8h = 来自 I2S 的中断事件

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.14 CPUIRQSEL13 寄存器（偏移 = 34h）[复位 = 0000001Dh]

图 5-23 展示了 CPUIRQSEL13，表 5-29 对其进行了介绍。

返回到汇总表。

CPU 中断 13 的输出选择

图 5-23. CPUIRQSEL13 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-1Dh						

表 5-29. CPUIRQSEL13 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	1Dh	只读选择值 1Dh = AUX 软件事件 1，由 AUX_EVCTL:SWEVSET.SWEV1 触发，也可用作 AUX_EVENT2 AON 唤醒事件。 MCU 域唤醒控制 AON_EVENT:MCUWUSEL

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.15 CPUIRQSEL14 寄存器（偏移 = 38h）[复位 = 00000018h]

图 5-24 展示了 CPUIRQSEL14，表 5-30 对其进行了介绍。

返回到汇总表。

CPU 中断 14 的输出选择

图 5-24. CPUIRQSEL14 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-18h						

表 5-30. CPUIRQSEL14 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	18h	只读选择值 18h = 看门狗中断事件，由 WDT:CTL.INTEN 控制

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.16 CPUIRQSEL15 寄存器（偏移 = 3Ch）[复位 = 00000010h]

图 5-25 展示了 CPUIRQSEL15，表 5-31 对其进行了介绍。

返回到汇总表。

CPU 中断 15 的输出选择

图 5-25. CPUIRQSEL15 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-10h						

表 5-31. CPUIRQSEL15 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	10h	只读选择值 10h = GPT0A 中断事件，由 GPT0:TAMR 控制

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.17 CPUIRQSEL16 寄存器（偏移 = 40h）[复位 = 00000011h]

图 5-26 展示了 CPUIRQSEL16，表 5-32 对其进行了介绍。

返回到汇总表。

CPU 中断 16 的输出选择

图 5-26. CPUIRQSEL16 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-11h						

表 5-32. CPUIRQSEL16 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	11h	只读选择值 11h = GPT0B 中断事件，由 GPT0:TBMR 控制

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.18 CPUIRQSEL17 寄存器（偏移 = 44h）[复位 = 00000012h]

图 5-27 展示了 CPUIRQSEL17，表 5-33 对其进行了介绍。

返回到汇总表。

CPU 中断 17 的输出选择

图 5-27. CPUIRQSEL17 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-12h						

表 5-33. CPUIRQSEL17 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	12h	只读选择值 12h = GPT1A 中断事件，由 GPT1:TAMR 控制

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.19 CPUIRQSEL18 寄存器（偏移 = 48h）[复位 = 00000013h]

图 5-28 展示了 CPUIRQSEL18，表 5-34 对其进行了介绍。

返回到汇总表。

CPU 中断 18 的输出选择

图 5-28. CPUIRQSEL18 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-13h						

表 5-34. CPUIRQSEL18 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	13h	只读选择值 13h = GPT1B 中断事件，由 GPT1:TBMR 控制

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.20 CPUIRQSEL19 寄存器（偏移 = 4Ch）[复位 = 0000000Ch]

图 5-29 展示了 CPUIRQSEL19，表 5-35 对其进行了介绍。

返回到汇总表。

CPU 中断 19 的输出选择

图 5-29. CPUIRQSEL19 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-Ch						

表 5-35. CPUIRQSEL19 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	Ch	只读选择值 Ch = GPT2A 中断事件，由 GPT2:TAMR 控制

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.21 CPUIRQSEL20 寄存器（偏移 = 50h）[复位 = 0000000Dh]

图 5-30 展示了 CPUIRQSEL20，表 5-36 对其进行了介绍。

返回到汇总表。

CPU 中断 20 的输出选择

图 5-30. CPUIRQSEL20 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-Dh						

表 5-36. CPUIRQSEL20 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	Dh	只读选择值 Dh = GPT2B 中断事件，由 GPT2:TBMR 控制

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.22 CPUIRQSEL21 寄存器（偏移 = 54h）[复位 = 0000000Eh]

图 5-31 展示了 CPUIRQSEL21，表 5-37 对其进行了介绍。

返回到汇总表。

CPU 中断 21 的输出选择

图 5-31. CPUIRQSEL21 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-Eh						

表 5-37. CPUIRQSEL21 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	Eh	只读选择值 Eh = GPT3A 中断事件，由 GPT3:TAMR 控制

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.23 CPUIRQSEL22 寄存器（偏移 = 58h）[复位 = 0000000Fh]

图 5-32 展示了 CPUIRQSEL22，表 5-38 对其进行了介绍。

返回到汇总表。

CPU 中断 22 的输出选择

图 5-32. CPUIRQSEL22 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-Fh						

表 5-38. CPUIRQSEL22 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	Fh	只读选择值 Fh = GPT3B 中断事件，由 GPT3:TBMR 控制

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.24 CPUIRQSEL23 寄存器（偏移 = 5Ch）[复位 = 0000005Dh]

图 5-33 展示了 CPUIRQSEL23，表 5-39 对其进行了介绍。

返回到汇总表。

CPU 中断 23 的输出选择

图 5-33. CPUIRQSEL23 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-5Dh						

表 5-39. CPUIRQSEL23 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	5Dh	只读选择值 5Dh = 加密结果可用中断事件，在此处找到相应的标志 CRYPTO:IRQSTAT.RESULT_AVAIL。由 CRYPTO:IRQSTAT.RESULT_AVAIL 控制

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.25 CPUIRQSEL24 寄存器（偏移 = 60h）[复位 = 00000027h]

图 5-34 展示了 CPUIRQSEL24，表 5-40 对其进行了介绍。

返回到汇总表。

CPU 中断 24 的输出选择

图 5-34. CPUIRQSEL24 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-27h						

表 5-40. CPUIRQSEL24 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	27h	只读选择值 27h = 组合式 DMA 完成，此处相应的标志为 UDMA0:REQDONE

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.26 CPUIRQSEL25 寄存器（偏移 = 64h）[复位 = 00000026h]

图 5-35 展示了 CPUIRQSEL25，表 5-41 对其进行了介绍。

返回到汇总表。

CPU 中断 25 的输出选择

图 5-35. CPUIRQSEL25 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-26h						

表 5-41. CPUIRQSEL25 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	26h	只读选择值 26h = DMA 总线错误，对应于 UDMA0:ERROR.STATUS

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.27 CPUIRQSEL26 寄存器（偏移 = 68h）[复位 = 00000015h]

图 5-36 展示了 CPUIRQSEL26，表 5-42 对其进行了介绍。

返回到汇总表。

CPU 中断 26 的输出选择

图 5-36. CPUIRQSEL26 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-15h						

表 5-42. CPUIRQSEL26 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	15h	只读选择值 15h = 闪存控制器错误事件，状态标志为 FLASH:FEDACSTAT.FSM_DONE 和 FLASH:FEDACSTAT.RVF_INT

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.28 CPUIRQSEL27 寄存器（偏移 = 6Ch）[复位 = 00000064h]

图 5-37 展示了 CPUIRQSEL27，表 5-43 对其进行了介绍。

返回到汇总表。

CPU 中断 27 的输出选择

图 5-37. CPUIRQSEL27 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-64h						

表 5-43. CPUIRQSEL27 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	64h	只读选择值 64h =软件事件 0，由 SWEV.SWEV0 触发

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.29 CPUIRQSEL28 寄存器（偏移 = 70h）[复位 = 0000000Bh]

图 5-38 展示了 CPUIRQSEL28，表 5-44 对其进行了介绍。

返回到汇总表。

CPU 中断 28 的输出选择

图 5-38. CPUIRQSEL28 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-Bh						

表 5-44. CPUIRQSEL28 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	Bh	只读选择值 Bh = AUX 组合事件，相应的标志寄存器在这里为 AUX_EVCTL:EVTOMCUFLAGS

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.30 CPUIRQSEL29 寄存器（偏移 = 74h）[复位 = 00000001h]

图 5-39 展示了 CPUIRQSEL29，表 5-45 对其进行了介绍。

返回到汇总表。

CPU 中断 29 的输出选择

图 5-39. CPUIRQSEL29 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-1h						

表 5-45. CPUIRQSEL29 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	1h	只读选择值 1h = AON 可编程事件 0。通过 AON_EVENT MCU 事件选择器选择的事件，AON_EVENT:EVTOMCUSEL.AON_PROG0_EV

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.31 CPUIRQSEL30 寄存器（偏移 = 78h）[复位 = 00000000h]

图 5-40 展示了 CPUIRQSEL30，表 5-46 对其进行了介绍。

返回到汇总表。

CPU 中断 30 的输出选择

图 5-40. CPUIRQSEL30 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R/W-0h						

表 5-46. CPUIRQSEL30 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 5-46. CPUIRQSEL30 寄存器字段说明（续）

位	字段	类型	复位	说明
6-0	EV	R/W	0h	读取/写入选择值 写入由 ENUM 定义的值以外的任何其他值可能会导致未定义的行为。 0h = 始终无效 2h = AON 可编程事件 1。通过 AON_EVENT MCU 事件选择器选择的事件，AON_EVENT:EVTOMCUSEL.AON_PROG1_EV 3h = AON 可编程事件 2。通过 AON_EVENT MCU 事件选择器选择的事件，AON_EVENT:EVTOMCUSEL.AON_PROG2_EV 8h = 来自 I2S 的中断事件 Ah = AUX 软件事件 0，AUX_EVCTL:SWEVSET.SWEV0 14h = 软件触发的 UDMA 通道 0 的 DMA 完成，请参阅 UDMA0:SOFTREQ 16h = 软件触发的 UDMA 通道 18 的 DMA 完成，请参阅 UDMA0:SOFTREQ 38h = AUX Timer2 事件 0，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_EV0 39h = AUX Timer2 事件 1，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_EV1 3Ah = AUX Timer2 事件 2，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_EV2 3Bh = AUX Timer2 事件 3，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_EV3 3Ch = AUX Timer2 脉冲，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_PULSE 5Eh = CRYPTO DMA 输入完成事件，相应的标志为 CRYPTO:IRQSTAT.DMA_IN_DONE。通过 CRYPTO:IRQEN.DMA_IN_DONE 控制 69h = AON 唤醒事件，相应的标志在这里为 AUX_EVCTL:EVTOMCUFLAGS.AUX_WU_EV 6Bh = AUX 比较 B 事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_COMPB 6Ch = AUX TDC 测量完成事件，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TDC_DONE and the AUX_TDC 状态 AUX_TDC:STAT.DONE 6Dh = AUX 计时器 0 事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER0_EV 6Eh = AUX 计时器 1 事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER1_EV 6Fh = 来自 AUX 信标的自动获取事件，通过 AUX_SMPH:AUTOTAKE 配置 70h = AUX ADC 完成，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_ADC_DONE 71h = AUX ADC FIFO 水印事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_ADC_FIFO_ALMOST_FULL 72h = OBSMUX0 通过 AUX 环回，对应于 AUX_EVCTL:EVTOMCUFLAGS.MCU_OBSMUX0 77h = 通过 AON_RTC:CTL.RTC_UPD_EN 控制的 RTC 周期性事件 79h = 始终置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.32 CPUIRQSEL31 寄存器（偏移 = 7Ch）[复位 = 0000006Ah]

图 5-41 展示了 CPUIRQSEL31，表 5-47 对其进行了介绍。

返回到汇总表。

CPU 中断 31 的输出选择

图 5-41. CPUIRQSEL31 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-6Ah						

表 5-47. CPUIRQSEL31 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	6Ah	只读选择值 6Ah = AUX 比较 A 事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_COMPA

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.33 CPUIRQSEL32 寄存器（偏移 = 80h）[复位 = 00000073h]

图 5-42 展示了 CPUIRQSEL32，表 5-48 对其进行了介绍。

返回到汇总表。

CPU 中断 32 的输出选择

图 5-42. CPUIRQSEL32 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-73h						

表 5-48. CPUIRQSEL32 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	73h	只读选择值 73h = AUX ADC 中断事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_ADC_IRQ。在这里找到的状态标志为 AUX_EVCTL:EVTOMCUFLAGS

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.34 CPUIRQSEL33 寄存器（偏移 = 84h）[复位 = 00000068h]

图 5-43 展示了 CPUIRQSEL33，表 5-49 对其进行了介绍。

返回到汇总表。

CPU 中断 33 的输出选择

图 5-43. CPUIRQSEL33 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-68h						

表 5-49. CPUIRQSEL33 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	68h	只读选择值 68h = TRNG 中断事件，通过 TRNG:IRQEN.EN 控制

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.35 CPUIRQSEL34 寄存器（偏移 = 88h）[复位 = 00000006h]

图 5-44 展示了 CPUIRQSEL34，表 5-50 对其进行了介绍。

返回到汇总表。

CPU 中断 34 的输出选择

图 5-44. CPUIRQSEL34 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-6h						

表 5-50. CPUIRQSEL34 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	6h	只读选择值 6h = 通过振荡器控制的组合事件

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.36 CPUIRQSEL35 寄存器（偏移 = 8Ch）[复位 = 00000038h]

图 5-45 展示了 CPUIRQSEL35，表 5-51 对其进行了介绍。

返回到汇总表。

CPU 中断 35 的输出选择

图 5-45. CPUIRQSEL35 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-38h						

表 5-51. CPUIRQSEL35 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	38h	只读选择值 38h = AUX Timer2 事件 0，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_EV0

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.37 CPUIRQSEL36 寄存器（偏移 = 90h）[复位 = 00000025h]

图 5-46 展示了 CPUIRQSEL36，表 5-52 对其进行了介绍。

返回到汇总表。

CPU 中断 36 的输出选择

图 5-46. CPUIRQSEL36 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-25h						

表 5-52. CPUIRQSEL36 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	25h	只读选择值 25h = UART1 组合中断，在此处找到的中断标志为 UART1:MIS

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.38 CPUIRQSEL37 寄存器（偏移 = 94h）[复位 = 00000005h]

图 5-47 展示了 CPUIRQSEL37，表 5-53 对其进行了介绍。

返回到汇总表。

CPU 中断 37 的输出选择

图 5-47. CPUIRQSEL37 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-5h						

表 5-53. CPUIRQSEL37 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	5h	只读选择值 5h = 电池监测器的组合事件

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.39 RFCSEL0 寄存器（偏移 = 100h）[复位 = 0000003Dh]

图 5-48 展示了 RFCSEL0，表 5-54 对其进行了介绍。

返回到汇总表。

RFC 事件 0 的输出选择

图 5-48. RFCSEL0 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-3Dh						

表 5-54. RFCSEL0 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	3Dh	只读选择值 3Dh = GPT0A 比较事件。通过 GPT0:TAMR.TCACT 进行配置

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.40 RFCSEL1 寄存器（偏移 = 104h）[复位 = 0000003Eh]

图 5-49 展示了 RFCSEL1，表 5-55 对其进行了介绍。

返回到汇总表。

RFC 事件 1 的输出选择

图 5-49. RFCSEL1 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-3Eh						

表 5-55. RFCSEL1 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	3Eh	只读选择值 3Eh = GPT0B 比较事件。通过 GPT0:TBMR.TCACT 进行配置

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.41 RFCSEL2 寄存器（偏移 = 108h）[复位 = 0000003Fh]

图 5-50 展示了 RFCSEL2，表 5-56 对其进行了介绍。

返回到汇总表。

RFC 事件 2 的输出选择

图 5-50. RFCSEL2 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-3Fh						

表 5-56. RFCSEL2 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	3Fh	只读选择值 3Fh = GPT1A 比较事件。通过 GPT1:TAMR.TCACT 进行配置

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.42 RFCSEL3 寄存器（偏移 = 10Ch）[复位 = 00000040h]

图 5-51 展示了 RFCSEL3，表 5-57 对其进行了介绍。

返回到汇总表。

RFC 事件 3 的输出选择

图 5-51. RFCSEL3 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-40h						

表 5-57. RFCSEL3 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	40h	只读选择值 40h = GPT1B 比较事件。通过 GPT1:TBMR.TCACT 进行配置

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.43 RFCSEL4 寄存器（偏移 = 110h）[复位 = 00000041h]

图 5-52 展示了 RFCSEL4，表 5-58 对其进行了介绍。

返回到汇总表。

RFC 事件 4 的输出选择

图 5-52. RFCSEL4 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-41h						

表 5-58. RFCSEL4 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	41h	只读选择值 41h = GPT2A 比较事件。通过 GPT2:TAMR.TCACT 进行配置

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.44 RFCSEL5 寄存器（偏移 = 114h）[复位 = 00000042h]

图 5-53 展示了 RFCSEL5，表 5-59 对其进行了介绍。

返回到汇总表。

RFC 事件 5 的输出选择

图 5-53. RFCSEL5 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-42h						

表 5-59. RFCSEL5 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	42h	只读选择值 42h = GPT2B 比较事件。通过 GPT2:TBMR.TCACT 进行配置

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.45 RFCSEL6 寄存器（偏移 = 118h）[复位 = 00000043h]

图 5-54 展示了 RFCSEL6，表 5-60 对其进行了介绍。

返回到汇总表。

RFC 事件 6 的输出选择

图 5-54. RFCSEL6 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-43h						

表 5-60. RFCSEL6 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	43h	只读选择值 43h = GPT3A 比较事件。通过 GPT3:TAMR.TCACT 进行配置

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.46 RFCSEL7 寄存器（偏移 = 11Ch）[复位 = 00000044h]

图 5-55 展示了 RFCSEL7，表 5-61 对其进行了介绍。

返回到汇总表。

RFC 事件 7 的输出选择

图 5-55. RFCSEL7 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-44h						

表 5-61. RFCSEL7 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	44h	只读选择值 44h = GPT3B 比较事件。通过 GPT3:TBMR.TCACT 进行配置

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.47 RFCSEL8 寄存器（偏移 = 120h）[复位 = 00000077h]

图 5-56 展示了 RFCSEL8，表 5-62 对其进行了介绍。

返回到汇总表。

RFC 事件 8 的输出选择

图 5-56. RFCSEL8 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-77h						

表 5-62. RFCSEL8 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	77h	只读选择值 77h = 通过 AON_RTC:CTL.RTC_UPD_EN 控制的 RTC 周期性事件

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.48 RFCSEL9 寄存器（偏移 = 124h）[复位 = 00000002h]

图 5-57 展示了 RFCSEL9，表 5-63 对其进行了介绍。

返回到汇总表。

RFC 事件 9 的输出选择

图 5-57. RFCSEL9 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R/W-2h						

表 5-63. RFCSEL9 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 5-63. RFCSEL9 寄存器字段说明（续）

位	字段	类型	复位	说明
6-0	EV	R/W	2h	读取/写入选择值 写入由 ENUM 定义的值以外的任何其他值可能会导致未定义的行为。 0h = 始终无效 1h = AON 可编程事件 0。通过 AON_EVENT MCU 事件选择器选择的事件，AON_EVENT:EVTOMCUSEL.AON_PROG0_EV 2h = AON 可编程事件 1。通过 AON_EVENT MCU 事件选择器选择的事件，AON_EVENT:EVTOMCUSEL.AON_PROG1_EV 8h = 来自 I2S 的中断事件 Ah = AUX 软件事件 0，AUX_EVCTL:SWEVSET.SWEV0 18h = 看门狗中断事件，由 WDT:CTL.INTEN 控制 22h = SSI0 组合中断，在此处找到的中断标志为 SSI0:MIS 23h = SSI1 组合中断，在此处找到的中断标志为 SSI1:MIS 24h = UART0 组合中断，在此处找到的中断标志为 UART0:MIS 25h = UART1 组合中断，在此处找到的中断标志为 UART1:MIS 27h = 组合式 DMA 完成，此处相应的标志为 UDMA0:REQDONE 38h = AUX Timer2 事件 0，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_EV0 39h = AUX Timer2 事件 1，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_EV1 3Ah = AUX Timer2 事件 2，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_EV2 3Bh = AUX Timer2 事件 3，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_EV3 3Ch = AUX Timer2 脉冲，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_PULSE 5Dh = 加密结果可用中断事件，在此处找到相应的标志 CRYPTO:IRQSTAT.RESULT_AVAIL。由 CRYPTO:IRQSTAT.RESULT_AVAIL 控制 64h =软件事件 0，由 SWEV.SWEV0 触发 65h =软件事件 1，由 SWEV.SWEV1 触发 69h = AON 唤醒事件，相应的标志在这里为 AUX_EVCTL:EVTOMCUFLAGS.AUX_WU_EV 6Ah = AUX 比较 A 事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_COMPA 6Bh = AUX 比较 B 事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_COMPB 6Ch = AUX TDC 测量完成事件，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TDC_DONE and the AUX_TDC 状态 AUX_TDC:STAT.DONE 6Dh = AUX 计时器 0 事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER0_EV 6Eh = AUX 计时器 1 事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER1_EV 6Fh = 来自 AUX 信标的自动获取事件，通过 AUX_SMPH:AUTOTAKE 配置 70h = AUX ADC 完成，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_ADC_DONE 71h = AUX ADC FIFO 水印事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_ADC_FIFO_ALMOST_FULL 72h = OBSMUX0 通过 AUX 环回，对应于 AUX_EVCTL:EVTOMCUFLAGS.MCU_OBSMUX0 73h = AUX ADC 中断事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_ADC_IRQ。在这里找到的状态标志为 AUX_EVCTL:EVTOMCUFLAGS 79h = 始终置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.49 GPT0ACAPTSEL 寄存器（偏移 = 200h）[复位 = 00000055h]

图 5-58 展示了 GPT0ACAPTSEL，表 5-64 对其进行了介绍。

返回到汇总表。

GPT0 0 的输出选择

图 5-58. GPT0ACAPTSEL 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R/W-55h						

表 5-64. GPT0ACAPTSEL 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 5-64. GPT0ACAPTSEL 寄存器字段说明（续）

位	字段	类型	复位	说明
6-0	EV	R/W	55h	读取/写入选择值 写入由 ENUM 定义的值以外的任何其他值可能会导致未定义的行为。 0h = 始终无效 4h = 来自 IOC 的边沿检测事件。通过 IOC:IOCFGn.EDGE_IRQ_EN 和 IOC:IOCFGn.EDGE_DET 设置进行配置 5h = 电池监测器的组合事件 6h = 通过振荡器控制的组合事件 7h = 来自 AON_RTC 的事件，通过 AON_RTC:CTL.COMB_EV_MASK 设置来控制 9h = 来自 I2C 的中断事件 Bh = AUX 组合事件，相应的标志寄存器在这里为 AUX_EVCTL:EVTOMCUFLAGS 15h = 闪存控制器错误事件，状态标志为 FLASH:FEDACSTAT.FSM_DONE 和 FLASH:FEDACSTAT.RVF_INT 19h = RFC 门铃命令确认中断，相当于 RFC_DBELL:RFACKIFG.ACKFLAG 1Ah = 组合的 RFC 硬件中断，相应的标志在此处为 RFC_DBELL:RFHWIFG 1Bh = CPE 生成的事件的组合中断。在这里对应的标志是 RFC_DBELL:RFCPEIFG。只有在 RFC_DBELL:RFCPEIFG 中通过 CPE0 选择的中断才能触发 RFC_CPE_0 事件 1Eh = CPE 生成的事件的组合中断。在这里对应的标志是 RFC_DBELL:RFCPEIFG。只有在 RFC_DBELL:RFCPEIFG 中通过 CPE1 选择的中断才能触发 RFC_CPE_1 事件 22h = SSI0 组合中断，在此处找到的中断标志为 SSI0:MIS 23h = SSI1 组合中断，在此处找到的中断标志为 SSI1:MIS 24h = UART0 组合中断，在此处找到的中断标志为 UART0:MIS 25h = UART1 组合中断，在此处找到的中断标志为 UART1:MIS 38h = AUX Timer2 事件 0，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_EV0 39h = AUX Timer2 事件 1，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_EV1 3Ah = AUX Timer2 事件 2，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_EV2 3Bh = AUX Timer2 事件 3，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_EV3 3Ch = AUX Timer2 脉冲，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_PULSE 3Dh = GPT0A 比较事件。通过 GPT0:TAMR.TCACT 进行配置 3Eh = GPT0B 比较事件。通过 GPT0:TBMR.TCACT 进行配置 3Fh = GPT1A 比较事件。通过 GPT1:TAMR.TCACT 进行配置 40h = GPT1B 比较事件。通过 GPT1:TBMR.TCACT 进行配置 41h = GPT2A 比较事件。通过 GPT2:TAMR.TCACT 进行配置 42h = GPT2B 比较事件。通过 GPT2:TBMR.TCACT 进行配置 43h = GPT3A 比较事件。通过 GPT3:TAMR.TCACT 进行配置 44h = GPT3B 比较事件。通过 GPT3:TBMR.TCACT 进行配置 55h = 来自 IOC 的端口捕获事件，通过 IOC:IOCFGn.PORT_ID 进行配置。通过 ENUM PORT_EVENT0 配置的端口上的事件将在此处路由。 56h = 来自 IOC 的端口捕获事件，通过 IOC:IOCFGn.PORT_ID 进行配置。通过 ENUM PORT_EVENT1 配置的端口上的事件将在此处路由。 69h = AON 唤醒事件，相应的标志在这里为 AUX_EVCTL:EVTOMCUFLAGS.AUX_WU_EV 6Ah = AUX 比较 A 事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_COMPA 6Bh = AUX 比较 B 事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_COMPB 6Ch = AUX TDC 测量完成事件，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TDC_DONE and the AUX_TDC 状态 AUX_TDC:STAT.DONE

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 5-64. GPT0ACAPTSEL 寄存器字段说明（续）

位	字段	类型	复位	说明
				6Dh = AUX 计时器 0 事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER0_EV 6Eh = AUX 计时器 1 事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER1_EV 6Fh = 来自 AUX 信标的自动获取事件，通过 AUX_SMPH:AUTOTAKE 配置 70h = AUX ADC 完成，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_ADC_DONE 71h = AUX ADC FIFO 水印事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_ADC_FIFO_ALMOST_FULL 72h = OBSMUX0 通过 AUX 环回，对应于 AUX_EVCTL:EVTOMCUFLAGS.MCU_OBSMUX0 73h = AUX ADC 中断事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_ADC_IRQ。在这里找到的状态标志为 AUX_EVCTL:EVTOMCUFLAGS 77h = 通过 AON_RTC:CTL.RTC_UPD_EN 控制的 RTC 周期性事件 79h = 始终置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.50 GPT0BCAPTSEL 寄存器（偏移 = 204h）[复位 = 00000056h]

图 5-59 展示了 GPT0BCAPTSEL，表 5-65 对其进行了介绍。

返回到汇总表。

GPT0 1 的输出选择

图 5-59. GPT0BCAPTSEL 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R/W-56h						

表 5-65. GPT0BCAPTSEL 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 5-65. GPT0BCAPTSEL 寄存器字段说明（续）

位	字段	类型	复位	说明
6-0	EV	R/W	56h	读取/写入选择值 写入由 ENUM 定义的值以外的任何其他值可能会导致未定义的行为。 0h = 始终无效 4h = 来自 IOC 的边沿检测事件。通过 IOC:IOCFGn.EDGE_IRQ_EN 和 IOC:IOCFGn.EDGE_DET 设置进行配置 5h = 电池监测器的组合事件 6h = 通过振荡器控制的组合事件 7h = 来自 AON_RTC 的事件，通过 AON_RTC:CTL.COMB_EV_MASK 设置来控制 9h = 来自 I2C 的中断事件 Bh = AUX 组合事件，相应的标志寄存器在这里为 AUX_EVCTL:EVTOMCUFLAGS 15h = 闪存控制器错误事件，状态标志为 FLASH:FEDACSTAT.FSM_DONE 和 FLASH:FEDACSTAT.RVF_INT 19h = RFC 门铃命令确认中断，相当于 RFC_DBELL:RFACKIFG.ACKFLAG 1Ah = 组合的 RFC 硬件中断，相应的标志在此处为 RFC_DBELL:RFHWIFG 1Bh = CPE 生成的事件的组合中断。在这里对应的标志是 RFC_DBELL:RFCPEIFG。只有在 RFC_DBELL:RFCPEIFG 中通过 CPE0 选择的中断才能触发 RFC_CPE_0 事件 1Eh = CPE 生成的事件的组合中断。在这里对应的标志是 RFC_DBELL:RFCPEIFG。只有在 RFC_DBELL:RFCPEIFG 中通过 CPE1 选择的中断才能触发 RFC_CPE_1 事件 22h = SSI0 组合中断，在此处找到的中断标志为 SSI0:MIS 23h = SSI1 组合中断，在此处找到的中断标志为 SSI1:MIS 24h = UART0 组合中断，在此处找到的中断标志为 UART0:MIS 25h = UART1 组合中断，在此处找到的中断标志为 UART1:MIS 38h = AUX Timer2 事件 0，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_EV0 39h = AUX Timer2 事件 1，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_EV1 3Ah = AUX Timer2 事件 2，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_EV2 3Bh = AUX Timer2 事件 3，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_EV3 3Ch = AUX Timer2 脉冲，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_PULSE 3Dh = GPT0A 比较事件。通过 GPT0:TAMR.TCACT 进行配置 3Eh = GPT0B 比较事件。通过 GPT0:TBMR.TCACT 进行配置 3Fh = GPT1A 比较事件。通过 GPT1:TAMR.TCACT 进行配置 40h = GPT1B 比较事件。通过 GPT1:TBMR.TCACT 进行配置 41h = GPT2A 比较事件。通过 GPT2:TAMR.TCACT 进行配置 42h = GPT2B 比较事件。通过 GPT2:TBMR.TCACT 进行配置 43h = GPT3A 比较事件。通过 GPT3:TAMR.TCACT 进行配置 44h = GPT3B 比较事件。通过 GPT3:TBMR.TCACT 进行配置 55h = 来自 IOC 的端口捕获事件，通过 IOC:IOCFGn.PORT_ID 进行配置。通过 ENUM PORT_EVENT0 配置的端口上的事件将在此处路由。 56h = 来自 IOC 的端口捕获事件，通过 IOC:IOCFGn.PORT_ID 进行配置。通过 ENUM PORT_EVENT1 配置的端口上的事件将在此处路由。 69h = AON 唤醒事件，相应的标志在这里为 AUX_EVCTL:EVTOMCUFLAGS.AUX_WU_EV 6Ah = AUX 比较 A 事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_COMPA 6Bh = AUX 比较 B 事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_COMPB 6Ch = AUX TDC 测量完成事件，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TDC_DONE and the AUX_TDC 状态 AUX_TDC:STAT.DONE

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 5-65. GPT0BCAPTSEL 寄存器字段说明（续）

位	字段	类型	复位	说明
				6Dh = AUX 计时器 0 事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER0_EV 6Eh = AUX 计时器 1 事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER1_EV 6Fh = 来自 AUX 信标的自动获取事件，通过 AUX_SMPH:AUTOTAKE 配置 70h = AUX ADC 完成，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_ADC_DONE 71h = AUX ADC FIFO 水印事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_ADC_FIFO_ALMOST_FULL 72h = OBSMUX0 通过 AUX 环回，对应于 AUX_EVCTL:EVTOMCUFLAGS.MCU_OBSMUX0 73h = AUX ADC 中断事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_ADC_IRQ。在这里找到的状态标志为 AUX_EVCTL:EVTOMCUFLAGS 77h = 通过 AON_RTC:CTL.RTC_UPD_EN 控制的 RTC 周期性事件 79h = 始终置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.51 GPT1ACAPTSEL 寄存器（偏移 = 300h）[复位 = 00000057h]

图 5-60 展示了 GPT1ACAPTSEL，表 5-66 对其进行了介绍。

返回到汇总表。

GPT1 0 的输出选择

图 5-60. GPT1ACAPTSEL 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R/W-57h						

表 5-66. GPT1ACAPTSEL 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 5-66. GPT1ACAPTSEL 寄存器字段说明（续）

位	字段	类型	复位	说明
6-0	EV	R/W	57h	读取/写入选择值 写入由 ENUM 定义的值以外的任何其他值可能会导致未定义的行为。 0h = 始终无效 4h = 来自 IOC 的边沿检测事件。通过 IOC:IOCFGn.EDGE_IRQ_EN 和 IOC:IOCFGn.EDGE_DET 设置进行配置 5h = 电池监测器的组合事件 6h = 通过振荡器控制的组合事件 7h = 来自 AON_RTC 的事件，通过 AON_RTC:CTL.COMB_EV_MASK 设置来控制 9h = 来自 I2C 的中断事件 Bh = AUX 组合事件，相应的标志寄存器在这里为 AUX_EVCTL:EVTOMCUFLAGS 15h = 闪存控制器错误事件，状态标志为 FLASH:FEDACSTAT.FSM_DONE 和 FLASH:FEDACSTAT.RVF_INT 19h = RFC 门铃命令确认中断，相当于 RFC_DBELL:RFACKIFG.ACKFLAG 1Ah = 组合的 RFC 硬件中断，相应的标志在此处为 RFC_DBELL:RFHWIFG 1Bh = CPE 生成的事件的组合中断。在这里对应的标志是 RFC_DBELL:RFCPEIFG。只有在 RFC_DBELL:RFCPEIFG 中通过 CPE0 选择的中断才能触发 RFC_CPE_0 事件 1Eh = CPE 生成的事件的组合中断。在这里对应的标志是 RFC_DBELL:RFCPEIFG。只有在 RFC_DBELL:RFCPEIFG 中通过 CPE1 选择的中断才能触发 RFC_CPE_1 事件 22h = SSI0 组合中断，在此处找到的中断标志为 SSI0:MIS 23h = SSI1 组合中断，在此处找到的中断标志为 SSI1:MIS 24h = UART0 组合中断，在此处找到的中断标志为 UART0:MIS 25h = UART1 组合中断，在此处找到的中断标志为 UART1:MIS 38h = AUX Timer2 事件 0，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_EV0 39h = AUX Timer2 事件 1，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_EV1 3Ah = AUX Timer2 事件 2，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_EV2 3Bh = AUX Timer2 事件 3，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_EV3 3Ch = AUX Timer2 脉冲，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_PULSE 3Dh = GPT0A 比较事件。通过 GPT0:TAMR.TCACT 进行配置 3Eh = GPT0B 比较事件。通过 GPT0:TBMR.TCACT 进行配置 3Fh = GPT1A 比较事件。通过 GPT1:TAMR.TCACT 进行配置 40h = GPT1B 比较事件。通过 GPT1:TBMR.TCACT 进行配置 41h = GPT2A 比较事件。通过 GPT2:TAMR.TCACT 进行配置 42h = GPT2B 比较事件。通过 GPT2:TBMR.TCACT 进行配置 43h = GPT3A 比较事件。通过 GPT3:TAMR.TCACT 进行配置 44h = GPT3B 比较事件。通过 GPT3:TBMR.TCACT 进行配置 57h = 来自 IOC 的端口捕获事件，通过 IOC:IOCFGn.PORT_ID 进行配置。通过 ENUM PORT_EVENT2 配置的端口上的事件将在此处路由。 58h = 来自 IOC 的端口捕获事件，通过 IOC:IOCFGn.PORT_ID 进行配置。通过 ENUM PORT_EVENT3 配置的端口上的事件将在此处路由。 69h = AON 唤醒事件，相应的标志在这里为 AUX_EVCTL:EVTOMCUFLAGS.AUX_WU_EV 6Ah = AUX 比较 A 事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_COMPA 6Bh = AUX 比较 B 事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_COMPB 6Ch = AUX TDC 测量完成事件，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TDC_DONE and the AUX_TDC 状态 AUX_TDC:STAT.DONE

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 5-66. GPT1ACAPTSEL 寄存器字段说明（续）

位	字段	类型	复位	说明
				6Dh = AUX 计时器 0 事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER0_EV 6Eh = AUX 计时器 1 事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER1_EV 6Fh = 来自 AUX 信标的自动获取事件，通过 AUX_SMPH:AUTOTAKE 配置 70h = AUX ADC 完成，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_ADC_DONE 71h = AUX ADC FIFO 水印事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_ADC_FIFO_ALMOST_FULL 72h = OBSMUX0 通过 AUX 环回，对应于 AUX_EVCTL:EVTOMCUFLAGS.MCU_OBSMUX0 73h = AUX ADC 中断事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_ADC_IRQ。在这里找到的状态标志为 AUX_EVCTL:EVTOMCUFLAGS 77h = 通过 AON_RTC:CTL.RTC_UPD_EN 控制的 RTC 周期性事件 79h = 始终置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.52 GPT1BCAPTSEL 寄存器（偏移 = 304h）[复位 = 00000058h]

图 5-61 展示了 GPT1BCAPTSEL，表 5-67 对其进行了介绍。

返回到汇总表。

GPT1 1 的输出选择

图 5-61. GPT1BCAPTSEL 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R/W-58h						

表 5-67. GPT1BCAPTSEL 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 5-67. GPT1BCAPTSEL 寄存器字段说明（续）

位	字段	类型	复位	说明
6-0	EV	R/W	58h	读取/写入选择值 写入由 ENUM 定义的值以外的任何其他值可能会导致未定义的行为。 0h = 始终无效 4h = 来自 IOC 的边沿检测事件。通过 IOC:IOCFGn.EDGE_IRQ_EN 和 IOC:IOCFGn.EDGE_DET 设置进行配置 5h = 电池监测器的组合事件 6h = 通过振荡器控制的组合事件 7h = 来自 AON_RTC 的事件，通过 AON_RTC:CTL.COMB_EV_MASK 设置来控制 9h = 来自 I2C 的中断事件 Bh = AUX 组合事件，相应的标志寄存器在这里为 AUX_EVCTL:EVTOMCUFLAGS 15h = 闪存控制器错误事件，状态标志为 FLASH:FEDACSTAT.FSM_DONE 和 FLASH:FEDACSTAT.RVF_INT 19h = RFC 门铃命令确认中断，相当于 RFC_DBELL:RFACKIFG.ACKFLAG 1Ah = 组合的 RFC 硬件中断，相应的标志在此处为 RFC_DBELL:RFHWIFG 1Bh = CPE 生成的事件的组合中断。在这里对应的标志是 RFC_DBELL:RFCPEIFG。只有在 RFC_DBELL:RFCPEIFG 中通过 CPE0 选择的中断才能触发 RFC_CPE_0 事件 1Eh = CPE 生成的事件的组合中断。在这里对应的标志是 RFC_DBELL:RFCPEIFG。只有在 RFC_DBELL:RFCPEIFG 中通过 CPE1 选择的中断才能触发 RFC_CPE_1 事件 22h = SSI0 组合中断，在此处找到的中断标志为 SSI0:MIS 23h = SSI1 组合中断，在此处找到的中断标志为 SSI1:MIS 24h = UART0 组合中断，在此处找到的中断标志为 UART0:MIS 25h = UART1 组合中断，在此处找到的中断标志为 UART1:MIS 38h = AUX Timer2 事件 0，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_EV0 39h = AUX Timer2 事件 1，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_EV1 3Ah = AUX Timer2 事件 2，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_EV2 3Bh = AUX Timer2 事件 3，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_EV3 3Ch = AUX Timer2 脉冲，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_PULSE 3Dh = GPT0A 比较事件。通过 GPT0:TAMR.TCACT 进行配置 3Eh = GPT0B 比较事件。通过 GPT0:TBMR.TCACT 进行配置 3Fh = GPT1A 比较事件。通过 GPT1:TAMR.TCACT 进行配置 40h = GPT1B 比较事件。通过 GPT1:TBMR.TCACT 进行配置 41h = GPT2A 比较事件。通过 GPT2:TAMR.TCACT 进行配置 42h = GPT2B 比较事件。通过 GPT2:TBMR.TCACT 进行配置 43h = GPT3A 比较事件。通过 GPT3:TAMR.TCACT 进行配置 44h = GPT3B 比较事件。通过 GPT3:TBMR.TCACT 进行配置 57h = 来自 IOC 的端口捕获事件，通过 IOC:IOCFGn.PORT_ID 进行配置。通过 ENUM PORT_EVENT2 配置的端口上的事件将在此处路由。 58h = 来自 IOC 的端口捕获事件，通过 IOC:IOCFGn.PORT_ID 进行配置。通过 ENUM PORT_EVENT3 配置的端口上的事件将在此处路由。 69h = AON 唤醒事件，相应的标志在这里为 AUX_EVCTL:EVTOMCUFLAGS.AUX_WU_EV 6Ah = AUX 比较 A 事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_COMPA 6Bh = AUX 比较 B 事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_COMPB 6Ch = AUX TDC 测量完成事件，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TDC_DONE and the AUX_TDC 状态 AUX_TDC:STAT.DONE

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 5-67. GPT1BCAPTSEL 寄存器字段说明（续）

位	字段	类型	复位	说明
				6Dh = AUX 计时器 0 事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER0_EV 6Eh = AUX 计时器 1 事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER1_EV 6Fh = 来自 AUX 信标的自动获取事件，通过 AUX_SMPH:AUTOTAKE 配置 70h = AUX ADC 完成，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_ADC_DONE 71h = AUX ADC FIFO 水印事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_ADC_FIFO_ALMOST_FULL 72h = OBSMUX0 通过 AUX 环回，对应于 AUX_EVCTL:EVTOMCUFLAGS.MCU_OBSMUX0 73h = AUX ADC 中断事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_ADC_IRQ。在这里找到的状态标志为 AUX_EVCTL:EVTOMCUFLAGS 77h = 通过 AON_RTC:CTL.RTC_UPD_EN 控制的 RTC 周期性事件 79h = 始终置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.53 GPT2ACAPTSEL 寄存器（偏移 = 400h）[复位 = 00000059h]

图 5-62 展示了 GPT2ACAPTSEL，表 5-68 对其进行了介绍。

返回到汇总表。

GPT2 0 的输出选择

图 5-62. GPT2ACAPTSEL 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R/W-59h						

表 5-68. GPT2ACAPTSEL 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 5-68. GPT2ACAPTSEL 寄存器字段说明（续）

位	字段	类型	复位	说明
6-0	EV	R/W	59h	读取/写入选择值 写入由 ENUM 定义的值以外的任何其他值可能会导致未定义的行为。 0h = 始终无效 4h = 来自 IOC 的边沿检测事件。通过 IOC:IOCFGn.EDGE_IRQ_EN 和 IOC:IOCFGn.EDGE_DET 设置进行配置 5h = 电池监测器的组合事件 6h = 通过振荡器控制的组合事件 7h = 来自 AON_RTC 的事件，通过 AON_RTC:CTL.COMB_EV_MASK 设置来控制 9h = 来自 I2C 的中断事件 Bh = AUX 组合事件，相应的标志寄存器在这里为 AUX_EVCTL:EVTOMCUFLAGS 15h = 闪存控制器错误事件，状态标志为 FLASH:FEDACSTAT.FSM_DONE 和 FLASH:FEDACSTAT.RVF_INT 19h = RFC 门铃命令确认中断，相当于 RFC_DBELL:RFACKIFG.ACKFLAG 1Ah = 组合的 RFC 硬件中断，相应的标志在此处为 RFC_DBELL:RFHWIFG 1Bh = CPE 生成的事件的组合中断。在这里对应的标志是 RFC_DBELL:RFCPEIFG。只有在 RFC_DBELL:RFCPEIFG 中通过 CPE0 选择的中断才能触发 RFC_CPE_0 事件 1Eh = CPE 生成的事件的组合中断。在这里对应的标志是 RFC_DBELL:RFCPEIFG。只有在 RFC_DBELL:RFCPEIFG 中通过 CPE1 选择的中断才能触发 RFC_CPE_1 事件 22h = SSI0 组合中断，在此处找到的中断标志为 SSI0:MIS 23h = SSI1 组合中断，在此处找到的中断标志为 SSI1:MIS 24h = UART0 组合中断，在此处找到的中断标志为 UART0:MIS 25h = UART1 组合中断，在此处找到的中断标志为 UART1:MIS 38h = AUX Timer2 事件 0，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_EV0 39h = AUX Timer2 事件 1，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_EV1 3Ah = AUX Timer2 事件 2，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_EV2 3Bh = AUX Timer2 事件 3，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_EV3 3Ch = AUX Timer2 脉冲，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_PULSE 3Dh = GPT0A 比较事件。通过 GPT0:TAMR.TCACT 进行配置 3Eh = GPT0B 比较事件。通过 GPT0:TBMR.TCACT 进行配置 3Fh = GPT1A 比较事件。通过 GPT1:TAMR.TCACT 进行配置 40h = GPT1B 比较事件。通过 GPT1:TBMR.TCACT 进行配置 41h = GPT2A 比较事件。通过 GPT2:TAMR.TCACT 进行配置 42h = GPT2B 比较事件。通过 GPT2:TBMR.TCACT 进行配置 43h = GPT3A 比较事件。通过 GPT3:TAMR.TCACT 进行配置 44h = GPT3B 比较事件。通过 GPT3:TBMR.TCACT 进行配置 59h = 来自 IOC 的端口捕获事件，通过 IOC:IOCFGn.PORT_ID 进行配置。通过 ENUM PORT_EVENT4 配置的端口上的事件将在此处路由。 5Ah = 来自 IOC 的端口捕获事件，通过 IOC:IOCFGn.PORT_ID 进行配置。通过 ENUM PORT_EVENT4 配置的端口上的事件将在此处路由。 69h = AON 唤醒事件，相应的标志在这里为 AUX_EVCTL:EVTOMCUFLAGS.AUX_WU_EV 6Ah = AUX 比较 A 事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_COMPA 6Bh = AUX 比较 B 事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_COMPB 6Ch = AUX TDC 测量完成事件，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TDC_DONE and the AUX_TDC 状态 AUX_TDC:STAT.DONE

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 5-68. GPT2ACAPTSEL 寄存器字段说明（续）

位	字段	类型	复位	说明
				6Dh = AUX 计时器 0 事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER0_EV 6Eh = AUX 计时器 1 事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER1_EV 6Fh = 来自 AUX 信标的自动获取事件，通过 AUX_SMPH:AUTOTAKE 配置 70h = AUX ADC 完成，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_ADC_DONE 71h = AUX ADC FIFO 水印事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_ADC_FIFO_ALMOST_FULL 72h = OBSMUX0 通过 AUX 环回，对应于 AUX_EVCTL:EVTOMCUFLAGS.MCU_OBSMUX0 73h = AUX ADC 中断事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_ADC_IRQ。在这里找到的状态标志为 AUX_EVCTL:EVTOMCUFLAGS 77h = 通过 AON_RTC:CTL.RTC_UPD_EN 控制的 RTC 周期性事件 79h = 始终置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.54 GPT2BCAPTSEL 寄存器（偏移 = 404h）[复位 = 0000005Ah]

图 5-63 展示了 GPT2BCAPTSEL，表 5-69 对其进行了介绍。

返回到汇总表。

GPT2 1 的输出选择

图 5-63. GPT2BCAPTSEL 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R/W-5Ah						

表 5-69. GPT2BCAPTSEL 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 5-69. GPT2BCAPTSEL 寄存器字段说明（续）

位	字段	类型	复位	说明
6-0	EV	R/W	5Ah	读取/写入选择值 写入由 ENUM 定义的值以外的任何其他值可能会导致未定义的行为。 0h = 始终无效 4h = 来自 IOC 的边沿检测事件。通过 IOC:IOCFGn.EDGE_IRQ_EN 和 IOC:IOCFGn.EDGE_DET 设置进行配置 5h = 电池监测器的组合事件 6h = 通过振荡器控制的组合事件 7h = 来自 AON_RTC 的事件，通过 AON_RTC:CTL.COMB_EV_MASK 设置来控制 9h = 来自 I2C 的中断事件 Bh = AUX 组合事件，相应的标志寄存器在这里为 AUX_EVCTL:EVTOMCUFLAGS 15h = 闪存控制器错误事件，状态标志为 FLASH:FEDACSTAT.FSM_DONE 和 FLASH:FEDACSTAT.RVF_INT 19h = RFC 门铃命令确认中断，相当于 RFC_DBELL:RFACKIFG.ACKFLAG 1Ah = 组合的 RFC 硬件中断，相应的标志在此处为 RFC_DBELL:RFHWIFG 1Bh = CPE 生成的事件的组合中断。在这里对应的标志是 RFC_DBELL:RFCPEIFG。只有在 RFC_DBELL:RFCPEIFG 中通过 CPE0 选择的中断才能触发 RFC_CPE_0 事件 1Eh = CPE 生成的事件的组合中断。在这里对应的标志是 RFC_DBELL:RFCPEIFG。只有在 RFC_DBELL:RFCPEIFG 中通过 CPE1 选择的中断才能触发 RFC_CPE_1 事件 22h = SSI0 组合中断，在此处找到的中断标志为 SSI0:MIS 23h = SSI1 组合中断，在此处找到的中断标志为 SSI1:MIS 24h = UART0 组合中断，在此处找到的中断标志为 UART0:MIS 25h = UART1 组合中断，在此处找到的中断标志为 UART1:MIS 38h = AUX Timer2 事件 0，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_EV0 39h = AUX Timer2 事件 1，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_EV1 3Ah = AUX Timer2 事件 2，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_EV2 3Bh = AUX Timer2 事件 3，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_EV3 3Ch = AUX Timer2 脉冲，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_PULSE 3Dh = GPT0A 比较事件。通过 GPT0:TAMR.TCACT 进行配置 3Eh = GPT0B 比较事件。通过 GPT0:TBMR.TCACT 进行配置 3Fh = GPT1A 比较事件。通过 GPT1:TAMR.TCACT 进行配置 40h = GPT1B 比较事件。通过 GPT1:TBMR.TCACT 进行配置 41h = GPT2A 比较事件。通过 GPT2:TAMR.TCACT 进行配置 42h = GPT2B 比较事件。通过 GPT2:TBMR.TCACT 进行配置 43h = GPT3A 比较事件。通过 GPT3:TAMR.TCACT 进行配置 44h = GPT3B 比较事件。通过 GPT3:TBMR.TCACT 进行配置 59h = 来自 IOC 的端口捕获事件，通过 IOC:IOCFGn.PORT_ID 进行配置。通过 ENUM PORT_EVENT4 配置的端口上的事件将在此处路由。 5Ah = 来自 IOC 的端口捕获事件，通过 IOC:IOCFGn.PORT_ID 进行配置。通过 ENUM PORT_EVENT4 配置的端口上的事件将在此处路由。 69h = AON 唤醒事件，相应的标志在这里为 AUX_EVCTL:EVTOMCUFLAGS.AUX_WU_EV 6Ah = AUX 比较 A 事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_COMPA 6Bh = AUX 比较 B 事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_COMPB 6Ch = AUX TDC 测量完成事件，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TDC_DONE and the AUX_TDC 状态 AUX_TDC:STAT.DONE

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 5-69. GPT2BCAPTSEL 寄存器字段说明（续）

位	字段	类型	复位	说明
				6Dh = AUX 计时器 0 事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER0_EV 6Eh = AUX 计时器 1 事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER1_EV 6Fh = 来自 AUX 信标的自动获取事件，通过 AUX_SMPH:AUTOTAKE 配置 70h = AUX ADC 完成，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_ADC_DONE 71h = AUX ADC FIFO 水印事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_ADC_FIFO_ALMOST_FULL 72h = OBSMUX0 通过 AUX 环回，对应于 AUX_EVCTL:EVTOMCUFLAGS.MCU_OBSMUX0 73h = AUX ADC 中断事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_ADC_IRQ。在这里找到的状态标志为 AUX_EVCTL:EVTOMCUFLAGS 77h = 通过 AON_RTC:CTL.RTC_UPD_EN 控制的 RTC 周期性事件 79h = 始终置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.55 UDMACH1SSEL 寄存器（偏移 = 508h）[复位 = 00000031h]

图 5-64 展示了 UDMACH1SSEL，表 5-70 对其进行了介绍。

返回到汇总表。

DMA 通道 1 SREQ 的输出选择

图 5-64. UDMACH1SSEL 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-31h						

表 5-70. UDMACH1SSEL 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	31h	只读选择值 31h = UART0 RX DMA 单请求，通过 UART0:DMACTL.RXDMAE 控制

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.56 UDMACH1BSEL 寄存器（偏移 = 50Ch）[复位 = 00000030h]

图 5-65 展示了 UDMACH1BSEL，表 5-71 对其进行了介绍。

返回到汇总表。

DMA 通道 1 REQ 的输出选择

图 5-65. UDMACH1BSEL 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-30h						

表 5-71. UDMACH1BSEL 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	30h	只读选择值 30h = UART0 RX DMA 突发请求，通过 UART0:DMACTL.RXDMAE 控制

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.57 UDMACH2SSEL 寄存器（偏移 = 510h）[复位 = 00000033h]

图 5-66 展示了 UDMACH2SSEL，表 5-72 对其进行了介绍。

返回到汇总表。

DMA 通道 2 SREQ 的输出选择

图 5-66. UDMACH2SSEL 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-33h						

表 5-72. UDMACH2SSEL 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	33h	只读选择值 33h = UART0 TX DMA 单请求，通过 UART0:DMACTL.TXDMAE 控制

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.58 UDMACH2BSEL 寄存器（偏移 = 514h）[复位 = 00000032h]

图 5-67 展示了 UDMACH2BSEL，表 5-73 对其进行了介绍。

返回到汇总表。

DMA 通道 2 REQ 的输出选择

图 5-67. UDMACH2BSEL 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-32h						

表 5-73. UDMACH2BSEL 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	32h	只读选择值 32h = UART0 TX DMA 突发请求，通过 UART0:DMACTL.TXDMAE 控制

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.59 UDMACH3SSEL 寄存器（偏移 = 518h）[复位 = 00000029h]

图 5-68 展示了 UDMACH3SSEL，表 5-74 对其进行了介绍。

返回到汇总表。

DMA 通道 3 SREQ 的输出选择

图 5-68. UDMACH3SSEL 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-29h						

表 5-74. UDMACH3SSEL 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	29h	只读选择值 29h = SSI0 RX DMA 单请求，通过 SSI0:DMACR.RXDMAE 控制

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.60 UDMACH3BSEL 寄存器（偏移 = 51Ch）[复位 = 00000028h]

图 5-69 展示了 UDMACH3BSEL，表 5-75 对其进行了介绍。

返回到汇总表。

DMA 通道 3 REQ 的输出选择

图 5-69. UDMACH3BSEL 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-28h						

表 5-75. UDMACH3BSEL 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	28h	只读选择值 28h = SSI0 RX DMA 突发请求，通过 SSI0:DMACR.RXDMAE 控制

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.61 UDMACH4SSEL 寄存器（偏移 = 520h）[复位 = 0000002Bh]

图 5-70 展示了 UDMACH4SSEL，表 5-76 对其进行了介绍。

返回到汇总表。

DMA 通道 4 SREQ 的输出选择

图 5-70. UDMACH4SSEL 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-2Bh						

表 5-76. UDMACH4SSEL 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	2Bh	只读选择值 2Bh = SSI0 TX DMA 单请求，通过 SSI0:DMACR.TXDMAE 控制

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.62 UDMACH4BSEL 寄存器（偏移 = 524h）[复位 = 0000002Ah]

图 5-71 展示了 UDMACH4BSEL，表 5-77 对其进行了介绍。

返回到汇总表。

DMA 通道 4 REQ 的输出选择

图 5-71. UDMACH4BSEL 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-2Ah						

表 5-77. UDMACH4BSEL 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	2Ah	只读选择值 2Ah = SSI0 TX DMA 突发请求，通过 SSI0:DMACR.TXDMAE 控制

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.63 UDMACH5SSEL 寄存器（偏移 = 528h）[复位 = 00000035h]

图 5-72 展示了 UDMACH5SSEL，表 5-78 对其进行了介绍。

返回到汇总表。

DMA 通道 5 SREQ 的输出选择

图 5-72. UDMACH5SSEL 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-35h						

表 5-78. UDMACH5SSEL 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	35h	只读选择值 35h = UART1 RX DMA 单请求，通过 UART1:DMACTL.RXDMAE 控制

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.64 UDMACH5BSEL 寄存器（偏移 = 52Ch）[复位 = 00000034h]

图 5-73 展示了 UDMACH5BSEL，表 5-79 对其进行了介绍。

返回到汇总表。

DMA 通道 5 REQ 的输出选择

图 5-73. UDMACH5BSEL 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-34h						

表 5-79. UDMACH5BSEL 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	34h	只读选择值 34h = UART1 RX DMA 突发请求，通过 UART1:DMACTL.RXDMAE 控制

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.65 UDMACH6SSEL 寄存器（偏移 = 530h）[复位 = 00000037h]

图 5-74 展示了 UDMACH6SSEL，表 5-80 对其进行了介绍。

返回到汇总表。

DMA 通道 6 SREQ 的输出选择

图 5-74. UDMACH6SSEL 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-37h						

表 5-80. UDMACH6SSEL 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	37h	只读选择值 37h = UART1 TX DMA 单请求，通过 UART1:DMACTL.TXDMAE 控制

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.66 UDMACH6BSEL 寄存器（偏移 = 534h）[复位 = 00000036h]

图 5-75 展示了 UDMACH6BSEL，表 5-81 对其进行了介绍。

返回到汇总表。

DMA 通道 6 REQ 的输出选择

图 5-75. UDMACH6BSEL 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-36h						

表 5-81. UDMACH6BSEL 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	36h	只读选择值 36h = UART1 TX DMA 突发请求，通过 UART1:DMACTL.TXDMAE 控制

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.67 UDMACH7SSEL 寄存器（偏移 = 538h）[复位 = 00000075h]

图 5-76 展示了 UDMACH7SSEL，表 5-82 对其进行了介绍。

返回到汇总表。

DMA 通道 7 SREQ 的输出选择

图 5-76. UDMACH7SSEL 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-75h						

表 5-82. UDMACH7SSEL 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	75h	只读选择值 75h = 来自 AUX 的 DMA 单请求事件，通过 AUX_EVCTL:DMACTL 进行配置

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.68 UDMACH7BSEL 寄存器（偏移 = 53Ch）[复位 = 00000076h]

图 5-77 展示了 UDMACH7BSEL，表 5-83 对其进行了介绍。

返回到汇总表。

DMA 通道 7 REQ 的输出选择

图 5-77. UDMACH7BSEL 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-76h						

表 5-83. UDMACH7BSEL 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	76h	只读选择值 76h = 来自 AUX 的 DMA 突发请求事件，通过 AUX_EVCTL:DMACTL 进行配置

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.69 UDMACH8SSEL 寄存器（偏移 = 540h）[复位 = 00000074h]

图 5-78 展示了 UDMACH8SSEL，表 5-84 对其进行了介绍。

返回到汇总表。

DMA 通道 8 SREQ 的输出选择
此通道的单请求将被忽略

图 5-78. UDMACH8SSEL 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-74h						

表 5-84. UDMACH8SSEL 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	74h	只读选择值 74h = 从 AUX 触发 DMA 软件，由 AUX_EVCTL:DMASWREQ.START 触发

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.70 UDMACH8BSEL 寄存器（偏移 = 544h）[复位 = 00000074h]

图 5-79 展示了 UDMACH8BSEL，表 5-85 对其进行了介绍。

返回到汇总表。

DMA 通道 8 REQ 的输出选择

图 5-79. UDMACH8BSEL 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-74h						

表 5-85. UDMACH8BSEL 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	74h	只读选择值 74h = 从 AUX 触发 DMA 软件，由 AUX_EVCTL:DMASWREQ.START 触发

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.71 UDMACH9SSEL 寄存器（偏移 = 548h）[复位 = 00000045h]

图 5-80 展示了 UDMACH9SSEL，表 5-86 对其进行了介绍。

返回到汇总表。

DMA 通道 9 SREQ 的输出选择
相应 DMA 通道的 DMA_DONE 作为 GPT0 上的中断可用作 GPT0:RIS.DMAARIS

图 5-80. UDMACH9SSEL 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R/W-45h						

表 5-86. UDMACH9SSEL 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R/W	45h	读取/写入选择值 写入由 ENUM 定义的值以外的任何其他值可能会导致未定义的行为。 0h = 始终无效 45h = 未使用并连接至 0 4Dh = GPT0A DMA 触发事件。通过 GPT0:DMAEV 进行配置 4Eh = GPT0B DMA 触发事件。通过 GPT0:DMAEV 进行配置 4Fh = GPT1A DMA 触发事件。通过 GPT1:DMAEV 进行配置 50h = GPT1B DMA 触发事件。通过 GPT1:DMAEV 进行配置 51h = GPT2A DMA 触发事件。通过 GPT2:DMAEV 进行配置 52h = GPT2B DMA 触发事件。通过 GPT2:DMAEV 进行配置 53h = GPT3A DMA 触发事件。通过 GPT3:DMAEV 进行配置 54h = GPT3B DMA 触发事件。通过 GPT3:DMAEV 进行配置 79h = 始终置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.72 UDMACH9BSEL 寄存器（偏移 = 54Ch）[复位 = 0000004Dh]

图 5-81 展示了 UDMACH9BSEL，表 5-87 对其进行了介绍。

返回到汇总表。

DMA 通道 9 REQ 的输出选择
相应 DMA 通道的 DMA_DONE 作为 GPT0 上的中断可用作 GPT0:RIS.DMAARIS

图 5-81. UDMACH9BSEL 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R/W-4Dh						

表 5-87. UDMACH9BSEL 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R/W	4Dh	读取/写入选择值 写入由 ENUM 定义的值以外的任何其他值可能会导致未定义的行为。 0h = 始终无效 4Dh = GPT0A DMA 触发事件。通过 GPT0:DMAEV 进行配置 4Eh = GPT0B DMA 触发事件。通过 GPT0:DMAEV 进行配置 4Fh = GPT1A DMA 触发事件。通过 GPT1:DMAEV 进行配置 50h = GPT1B DMA 触发事件。通过 GPT1:DMAEV 进行配置 51h = GPT2A DMA 触发事件。通过 GPT2:DMAEV 进行配置 52h = GPT2B DMA 触发事件。通过 GPT2:DMAEV 进行配置 53h = GPT3A DMA 触发事件。通过 GPT3:DMAEV 进行配置 54h = GPT3B DMA 触发事件。通过 GPT3:DMAEV 进行配置 79h = 始终置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.73 UDMACH10SSEL 寄存器（偏移 = 550h）[复位 = 00000046h]

图 5-82 展示了 UDMACH10SSEL，表 5-88 对其进行了介绍。

返回到汇总表。

DMA 通道 10 SREQ 的输出选择
相应 DMA 通道的 DMA_DONE 作为 GPT0 上的中断可用作 GPT0:RIS.DMABRIS

图 5-82. UDMACH10SSEL 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R/W-46h						

表 5-88. UDMACH10SSEL 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R/W	46h	读取/写入选择值 写入由 ENUM 定义的值以外的任何其他值可能会导致未定义的行为。 0h = 始终无效 46h = 未使用并连接至 0 4Dh = GPT0A DMA 触发事件。通过 GPT0:DMAEV 进行配置 4Eh = GPT0B DMA 触发事件。通过 GPT0:DMAEV 进行配置 4Fh = GPT1A DMA 触发事件。通过 GPT1:DMAEV 进行配置 50h = GPT1B DMA 触发事件。通过 GPT1:DMAEV 进行配置 51h = GPT2A DMA 触发事件。通过 GPT2:DMAEV 进行配置 52h = GPT2B DMA 触发事件。通过 GPT2:DMAEV 进行配置 53h = GPT3A DMA 触发事件。通过 GPT3:DMAEV 进行配置 54h = GPT3B DMA 触发事件。通过 GPT3:DMAEV 进行配置 79h = 始终置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.74 UDMACH10BSEL 寄存器（偏移 = 554h）[复位 = 0000004Eh]

图 5-83 展示了 UDMACH10BSEL，表 5-89 对其进行了介绍。

返回到汇总表。

DMA 通道 10 REQ 的输出选择
相应 DMA 通道的 DMA_DONE 作为 GPT0 上的中断可用作 GPT0:RIS.DMABRIS

图 5-83. UDMACH10BSEL 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R/W-4Eh						

表 5-89. UDMACH10BSEL 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R/W	4Eh	读取/写入选择值 写入由 ENUM 定义的值以外的任何其他值可能会导致未定义的行为。 0h = 始终无效 4Dh = GPT0A DMA 触发事件。通过 GPT0:DMAEV 进行配置 4Eh = GPT0B DMA 触发事件。通过 GPT0:DMAEV 进行配置 4Fh = GPT1A DMA 触发事件。通过 GPT1:DMAEV 进行配置 50h = GPT1B DMA 触发事件。通过 GPT1:DMAEV 进行配置 51h = GPT2A DMA 触发事件。通过 GPT2:DMAEV 进行配置 52h = GPT2B DMA 触发事件。通过 GPT2:DMAEV 进行配置 53h = GPT3A DMA 触发事件。通过 GPT3:DMAEV 进行配置 54h = GPT3B DMA 触发事件。通过 GPT3:DMAEV 进行配置 79h = 始终置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.75 UDMACH11SSEL 寄存器（偏移 = 558h）[复位 = 00000047h]

图 5-84 展示了 UDMACH11SSEL，表 5-90 对其进行了介绍。

返回到汇总表。

DMA 通道 11 SREQ 的输出选择
相应 DMA 通道的 DMA_DONE 作为 GPT1 上的中断可用作 GPT1:RIS.DMAARIS

图 5-84. UDMACH11SSEL 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R/W-47h						

表 5-90. UDMACH11SSEL 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R/W	47h	读取/写入选择值 写入由 ENUM 定义的值以外的任何其他值可能会导致未定义的行为。 0h = 始终无效 47h = 未使用并连接至 0 4Dh = GPT0A DMA 触发事件。通过 GPT0:DMAEV 进行配置 4Eh = GPT0B DMA 触发事件。通过 GPT0:DMAEV 进行配置 4Fh = GPT1A DMA 触发事件。通过 GPT1:DMAEV 进行配置 50h = GPT1B DMA 触发事件。通过 GPT1:DMAEV 进行配置 51h = GPT2A DMA 触发事件。通过 GPT2:DMAEV 进行配置 52h = GPT2B DMA 触发事件。通过 GPT2:DMAEV 进行配置 53h = GPT3A DMA 触发事件。通过 GPT3:DMAEV 进行配置 54h = GPT3B DMA 触发事件。通过 GPT3:DMAEV 进行配置 79h = 始终置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.76 UDMACH11BSEL 寄存器（偏移 = 55Ch）[复位 = 0000004Fh]

图 5-85 展示了 UDMACH11BSEL，表 5-91 对其进行了介绍。

返回到汇总表。

DMA 通道 11 REQ 的输出选择
相应 DMA 通道的 DMA_DONE 作为 GPT1 上的中断可用作 GPT1:RIS.DMAARIS

图 5-85. UDMACH11BSEL 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R/W-4Fh						

表 5-91. UDMACH11BSEL 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R/W	4Fh	读取/写入选择值 写入由 ENUM 定义的值以外的任何其他值可能会导致未定义的行为。 0h = 始终无效 4Dh = GPT0A DMA 触发事件。通过 GPT0:DMAEV 进行配置 4Eh = GPT0B DMA 触发事件。通过 GPT0:DMAEV 进行配置 4Fh = GPT1A DMA 触发事件。通过 GPT1:DMAEV 进行配置 50h = GPT1B DMA 触发事件。通过 GPT1:DMAEV 进行配置 51h = GPT2A DMA 触发事件。通过 GPT2:DMAEV 进行配置 52h = GPT2B DMA 触发事件。通过 GPT2:DMAEV 进行配置 53h = GPT3A DMA 触发事件。通过 GPT3:DMAEV 进行配置 54h = GPT3B DMA 触发事件。通过 GPT3:DMAEV 进行配置 79h = 始终置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.77 UDMACH12SSEL 寄存器（偏移 = 560h）[复位 = 00000048h]

图 5-86 展示了 UDMACH12SSEL，表 5-92 对其进行了介绍。

返回到汇总表。

DMA 通道 12 SREQ 的输出选择
相应 DMA 通道的 DMA_DONE 作为 GPT1 上的中断可用作 GPT1:RIS.DMABRIS

图 5-86. UDMACH12SSEL 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R/W-48h						

表 5-92. UDMACH12SSEL 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R/W	48h	读取/写入选择值 写入由 ENUM 定义的值以外的任何其他值可能会导致未定义的行为。 0h = 始终无效 48h = 未使用并连接至 0 4Dh = GPT0A DMA 触发事件。通过 GPT0:DMAEV 进行配置 4Eh = GPT0B DMA 触发事件。通过 GPT0:DMAEV 进行配置 4Fh = GPT1A DMA 触发事件。通过 GPT1:DMAEV 进行配置 50h = GPT1B DMA 触发事件。通过 GPT1:DMAEV 进行配置 51h = GPT2A DMA 触发事件。通过 GPT2:DMAEV 进行配置 52h = GPT2B DMA 触发事件。通过 GPT2:DMAEV 进行配置 53h = GPT3A DMA 触发事件。通过 GPT3:DMAEV 进行配置 54h = GPT3B DMA 触发事件。通过 GPT3:DMAEV 进行配置 79h = 始终置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.78 UDMACH12BSEL 寄存器（偏移 = 564h）[复位 = 00000050h]

图 5-87 展示了 UDMACH12BSEL，表 5-93 对其进行了介绍。

返回到汇总表。

DMA 通道 12 REQ 的输出选择
相应 DMA 通道的 DMA_DONE 作为 GPT1 上的中断可用作 GPT1:RIS.DMABRIS

图 5-87. UDMACH12BSEL 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R/W-50h						

表 5-93. UDMACH12BSEL 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R/W	50h	读取/写入选择值 写入由 ENUM 定义的值以外的任何其他值可能会导致未定义的行为。 0h = 始终无效 4Dh = GPT0A DMA 触发事件。通过 GPT0:DMAEV 进行配置 4Eh = GPT0B DMA 触发事件。通过 GPT0:DMAEV 进行配置 4Fh = GPT1A DMA 触发事件。通过 GPT1:DMAEV 进行配置 50h = GPT1B DMA 触发事件。通过 GPT1:DMAEV 进行配置 51h = GPT2A DMA 触发事件。通过 GPT2:DMAEV 进行配置 52h = GPT2B DMA 触发事件。通过 GPT2:DMAEV 进行配置 53h = GPT3A DMA 触发事件。通过 GPT3:DMAEV 进行配置 54h = GPT3B DMA 触发事件。通过 GPT3:DMAEV 进行配置 79h = 始终置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.79 UDMACH13BSEL 寄存器（偏移 = 56Ch）[复位 = 00000003h]

图 5-88 展示了 UDMACH13BSEL，表 5-94 对其进行了介绍。

返回到汇总表。

DMA 通道 13 REQ 的输出选择

图 5-88. UDMACH13BSEL 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-3h						

表 5-94. UDMACH13BSEL 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	3h	只读选择值 3h = AON 可编程事件 2。通过 AON_EVENT MCU 事件选择器选择的事件，AON_EVENT:EVTOMCUSEL.AON_PROG2_EV

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.80 UDMACH14BSEL 寄存器（偏移 = 574h）[复位 = 00000001h]

图 5-89 展示了 UDMACH14BSEL，表 5-95 对其进行了介绍。

返回到汇总表。

DMA 通道 14 REQ 的输出选择

图 5-89. UDMACH14BSEL 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R/W-1h						

表 5-95. UDMACH14BSEL 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 5-95. UDMACH14BSEL 寄存器字段说明（续）

位	字段	类型	复位	说明
6-0	EV	R/W	1h	读取/写入选择值 写入由 ENUM 定义的值以外的任何其他值可能会导致未定义的行为。 0h = 始终无效 1h = AON 可编程事件 0。通过 AON_EVENT MCU 事件选择器选择的事件，AON_EVENT:EVTOMCUSEL.AON_PROG0_EV 2h = AON 可编程事件 1。通过 AON_EVENT MCU 事件选择器选择的事件，AON_EVENT:EVTOMCUSEL.AON_PROG1_EV 3h = AON 可编程事件 2。通过 AON_EVENT MCU 事件选择器选择的事件，AON_EVENT:EVTOMCUSEL.AON_PROG2_EV 4h = 来自 IOC 的边沿检测事件。通过 IOC:IOCFGn.EDGE_IRQ_EN 和 IOC:IOCFGn.EDGE_DET 设置进行配置 5h = 电池监测器的组合事件 6h = 通过振荡器控制的组合事件 7h = 来自 AON_RTC 的事件，通过 AON_RTC:CTL.COMB_EV_MASK 设置来控制 8h = 来自 I2S 的中断事件 9h = 来自 I2C 的中断事件 Ah = AUX 软件事件 0，AUX_EVCTL:SWEVSET.SWEV0 Bh = AUX 组合事件，相应的标志寄存器在这里为 AUX_EVCTL:EVTOMCUFLAGS Ch = GPT2A 中断事件，由 GPT2:TAMR 控制 Dh = GPT2B 中断事件，由 GPT2:TBMR 控制 Eh = GPT3A 中断事件，由 GPT3:TAMR 控制 Fh = GPT3B 中断事件，由 GPT3:TBMR 控制 10h = GPT0A 中断事件，由 GPT0:TAMR 控制 11h = GPT0B 中断事件，由 GPT0:TBMR 控制 12h = GPT1A 中断事件，由 GPT1:TAMR 控制 13h = GPT1B 中断事件，由 GPT1:TBMR 控制 14h = 软件触发的 UDMA 通道 0 的 DMA 完成，请参阅 UDMA0:SOFTREQ 15h = 闪存控制器错误事件，状态标志为 FLASH:FEDACSTAT.FSM_DONE 和 FLASH:FEDACSTAT.RVF_INT 16h = 软件触发的 UDMA 通道 18 的 DMA 完成，请参阅 UDMA0:SOFTREQ 18h = 看门狗中断事件，由 WDT:CTL.INTEN 控制 19h = RFC 门铃命令确认中断，相当于 RFC_DBELL:RFACKIFG.ACKFLAG 1Ah = 组合的 RFC 硬件中断，相应的标志在此处为 RFC_DBELL:RFHWIFG 1Bh = CPE 生成的事件的组合中断。在这里对应的标志是 RFC_DBELL:RFCPEIFG。只有在 RFC_DBELL:RFCPEIFG 中通过 CPE0 选择的中断才能触发 RFC_CPE_0 事件 1Dh = AUX 软件事件 1，由 AUX_EVCTL:SWEVSET.SWEV1 触发，也可用作 AUX_EVENT2 AON 唤醒事件。 MCU 域唤醒控制 AON_EVENT:MCUWUSEL 1Eh = CPE 生成的事件的组合中断。在这里对应的标志是 RFC_DBELL:RFCPEIFG。只有在 RFC_DBELL:RFCPEIFG 中通过 CPE1 选择的中断才能触发 RFC_CPE_1 事件 1Fh = PKA 中断事件 22h = SSI0 组合中断，在此处找到的中断标志为 SSI0:MIS 23h = SSI1 组合中断，在此处找到的中断标志为 SSI1:MIS 24h = UART0 组合中断，在此处找到的中断标志为 UART0:MIS 25h = UART1 组合中断，在此处找到的中断标志为 UART1:MIS 26h = DMA 总线错误，对应于 UDMA0:ERROR.STATUS 27h = 组合式 DMA 完成，此处相应的标志为 UDMA0:REQDONE 28h = SSI0 RX DMA 突发请求，通过 SSI0:DMACR.RXDMAE 控制 29h = SSI0 RX DMA 单请求，通过 SSI0:DMACR.RXDMAE 控制 2Ah = SSI0 TX DMA 突发请求，通过 SSI0:DMACR.TXDMAE 控制 2Bh = SSI0 TX DMA 单请求，通过 SSI0:DMACR.TXDMAE 控制 2Ch = SSI1 RX DMA 突发请求，通过 SSI0:DMACR.RXDMAE 控制

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 5-95. UDMACH14BSEL 寄存器字段说明（续）

位	字段	类型	复位	说明
				2Dh = SSI1 RX DMA 单请求，通过 SSI0:DMACR.RXDMAE 控制 2Eh = SSI1 TX DMA 突发请求，通过 SSI0:DMACR.TXDMAE 控制 2Fh = SSI1 TX DMA 单请求，通过 SSI0:DMACR.TXDMAE 控制 30h = UART0 RX DMA 突发请求，通过 UART0:DMACTL.RXDMAE 控制 31h = UART0 RX DMA 单请求，通过 UART0:DMACTL.RXDMAE 控制 32h = UART0 TX DMA 突发请求，通过 UART0:DMACTL.TXDMAE 控制 33h = UART0 TX DMA 单请求，通过 UART0:DMACTL.TXDMAE 控制 34h = UART1 RX DMA 突发请求，通过 UART1:DMACTL.RXDMAE 控制 35h = UART1 RX DMA 单请求，通过 UART1:DMACTL.RXDMAE 控制 36h = UART1 TX DMA 突发请求，通过 UART1:DMACTL.TXDMAE 控制 37h = UART1 TX DMA 单请求，通过 UART1:DMACTL.TXDMAE 控制 38h = AUX Timer2 事件 0，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_EV0 39h = AUX Timer2 事件 1，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_EV1 3Ah = AUX Timer2 事件 2，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_EV2 3Bh = AUX Timer2 事件 3，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_EV3 3Ch = AUX Timer2 脉冲，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_PULSE 3Dh = GPT0A 比较事件。通过 GPT0:TAMR.TCACT 进行配置 3Eh = GPT0B 比较事件。通过 GPT0:TBMR.TCACT 进行配置 3Fh = GPT1A 比较事件。通过 GPT1:TAMR.TCACT 进行配置 40h = GPT1B 比较事件。通过 GPT1:TBMR.TCACT 进行配置 41h = GPT2A 比较事件。通过 GPT2:TAMR.TCACT 进行配置 42h = GPT2B 比较事件。通过 GPT2:TBMR.TCACT 进行配置 43h = GPT3A 比较事件。通过 GPT3:TAMR.TCACT 进行配置 44h = GPT3B 比较事件。通过 GPT3:TBMR.TCACT 进行配置 4Dh = GPT0A DMA 触发事件。通过 GPT0:DMAEV 进行配置 4Eh = GPT0B DMA 触发事件。通过 GPT0:DMAEV 进行配置 4Fh = GPT1A DMA 触发事件。通过 GPT1:DMAEV 进行配置 50h = GPT1B DMA 触发事件。通过 GPT1:DMAEV 进行配置 51h = GPT2A DMA 触发事件。通过 GPT2:DMAEV 进行配置 52h = GPT2B DMA 触发事件。通过 GPT2:DMAEV 进行配置 53h = GPT3A DMA 触发事件。通过 GPT3:DMAEV 进行配置 54h = GPT3B DMA 触发事件。通过 GPT3:DMAEV 进行配置 55h = 来自 IOC 的端口捕获事件，通过 IOC:IOCFGn.PORT_ID 进行配置。通过 ENUM PORT_EVENT0 配置的端口上的事件将在此处路由。 56h = 来自 IOC 的端口捕获事件，通过 IOC:IOCFGn.PORT_ID 进行配置。通过 ENUM PORT_EVENT1 配置的端口上的事件将在此处路由。 57h = 来自 IOC 的端口捕获事件，通过 IOC:IOCFGn.PORT_ID 进行配置。通过 ENUM PORT_EVENT2 配置的端口上的事件将在此处路由。 58h = 来自 IOC 的端口捕获事件，通过 IOC:IOCFGn.PORT_ID 进行配置。通过 ENUM PORT_EVENT3 配置的端口上的事件将在此处路由。 59h = 来自 IOC 的端口捕获事件，通过 IOC:IOCFGn.PORT_ID 进行配置。通过 ENUM PORT_EVENT4 配置的端口上的事件将在此处路由。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 5-95. UDMACH14BSEL 寄存器字段说明（续）

位	字段	类型	复位	说明
				5Ah = 来自 IOC 的端口捕获事件，通过 IOC:IOCFGn.PORT_ID 进行配置。通过 ENUM PORT_EVENT4 配置的端口上的事件将在此处路由。 5Bh = 来自 IOC 的端口捕获事件，通过 IOC:IOCFGn.PORT_ID 进行配置。通过 ENUM PORT_EVENT6 配置的端口上的事件将在此处路由。 5Ch = 来自 IOC 的端口捕获事件，通过 IOC:IOCFGn.PORT_ID 进行配置。通过 ENUM PORT_EVENT7 配置的端口上的事件将在此处路由。 5Dh = 加密结果可用中断事件，在此处找到相应的标志 CRYPTO:IRQSTAT.RESULT_AVAIL。由 CRYPTO:IRQSTAT.RESULT_AVAIL 控制 5Eh = CRYPTO DMA 输入完成事件，相应的标志为 CRYPTO:IRQSTAT.DMA_IN_DONE。通过 CRYPTO:IRQEN.DMA_IN_DONE 控制 63h =看门狗不可屏蔽中断事件，通过 WDT:CTL.INTTYPE 控制 64h =软件事件 0，由 SWEV.SWEV0 触发 65h =软件事件 1，由 SWEV.SWEV1 触发 66h =软件事件 2，由 SWEV.SWEV2 触发 67h =软件事件 3，由 SWEV.SWEV3 触发 68h = TRNG 中断事件，通过 TRNG:IRQEN.EN 控制 69h = AON 唤醒事件，相应的标志在这里为 AUX_EVCTL:EVTOMCUFLAGS.AUX_WU_EV 6Ah = AUX 比较 A 事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_COMPA 6Bh = AUX 比较 B 事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_COMPB 6Ch = AUX TDC 测量完成事件，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TDC_DONE and the AUX_TDC 状态 AUX_TDC:STAT.DONE 6Dh = AUX 计时器 0 事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER0_EV 6Eh = AUX 计时器 1 事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER1_EV 6Fh = 来自 AUX 信标的自动获取事件，通过 AUX_SMPH:AUTOTAKE 配置 70h = AUX ADC 完成，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_ADC_DONE 71h = AUX ADC FIFO 水印事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_ADC_FIFO_ALMOST_FULL 72h = OBSMUX0 通过 AUX 环回，对应于 AUX_EVCTL:EVTOMCUFLAGS.MCU_OBSMUX0 73h = AUX ADC 中断事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_ADC_IRQ。在这里找到的状态标志为 AUX_EVCTL:EVTOMCUFLAGS 74h = 从 AUX 触发 DMA 软件，由 AUX_EVCTL:DMASWREQ.START 触发 75h = 来自 AUX 的 DMA 单请求事件，通过 AUX_EVCTL:DMACTL 进行配置 76h = 来自 AUX 的 DMA 突发请求事件，通过 AUX_EVCTL:DMACTL 进行配置 77h = 通过 AON_RTC:CTL.RTC_UPD_EN 控制的 RTC 周期性事件 78h = CPU 停止 79h = 始终置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.81 UDMACH15BSEL 寄存器（偏移 = 57Ch）[复位 = 00000007h]

图 5-90 展示了 UDMACH15BSEL，表 5-96 对其进行了介绍。

返回到汇总表。

DMA 通道 15 REQ 的输出选择

图 5-90. UDMACH15BSEL 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-7h						

表 5-96. UDMACH15BSEL 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	7h	只读选择值 7h = 来自 AON_RTC 的事件，通过 AON_RTC:CTL.COMB_EV_MASK 设置来控制

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.82 UDMACH16SSEL 寄存器（偏移 = 580h）[复位 = 0000002Dh]

图 5-91 展示了 UDMACH16SSEL，表 5-97 对其进行了介绍。

返回到汇总表。

DMA 通道 16 SREQ 的输出选择

图 5-91. UDMACH16SSEL 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-2Dh						

表 5-97. UDMACH16SSEL 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	2Dh	只读选择值 2Dh = SSI1 RX DMA 单请求，通过 SSI0:DMACR.RXDMAE 控制

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.83 UDMACH16BSEL 寄存器（偏移 = 584h）[复位 = 0000002Ch]

图 5-92 展示了 UDMACH16BSEL，表 5-98 对其进行了介绍。

返回到汇总表。

DMA 通道 16 REQ 的输出选择

图 5-92. UDMACH16BSEL 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-2Ch						

表 5-98. UDMACH16BSEL 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	2Ch	只读选择值 2Ch = SSI1 RX DMA 突发请求，通过 SSI0:DMACR.RXDMAE 控制

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.84 UDMACH17SSEL 寄存器（偏移 = 588h）[复位 = 0000002Fh]

图 5-93 展示了 UDMACH17SSEL，表 5-99 对其进行了介绍。

返回到汇总表。

DMA 通道 17 SREQ 的输出选择

图 5-93. UDMACH17SSEL 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-2Fh						

表 5-99. UDMACH17SSEL 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	2Fh	只读选择值 2Fh = SSI1 TX DMA 单请求，通过 SSI0:DMACR.TXDMAE 控制

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.85 UDMACH17BSEL 寄存器（偏移 = 58Ch）[复位 = 0000002Eh]

图 5-94 展示了 UDMACH17BSEL，表 5-100 对其进行了介绍。

返回到汇总表。

DMA 通道 17 REQ 的输出选择

图 5-94. UDMACH17BSEL 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-2Eh						

表 5-100. UDMACH17BSEL 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	2Eh	只读选择值 2Eh = SSI1 TX DMA 突发请求，通过 SSI0:DMACR.TXDMAE 控制

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.86 UDMACH21SSEL 寄存器（偏移 = 5A8h）[复位 = 00000064h]

图 5-95 展示了 UDMACH21SSEL，表 5-101 对其进行了介绍。

返回到汇总表。

DMA 通道 21 SREQ 的输出选择

图 5-95. UDMACH21SSEL 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-64h						

表 5-101. UDMACH21SSEL 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	64h	只读选择值 64h =软件事件 0，由 SWEV.SWEV0 触发

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.87 UDMACH21BSEL 寄存器（偏移 = 5ACh）[复位 = 00000064h]

图 5-96 展示了 UDMACH21BSEL，表 5-102 对其进行了介绍。

返回到汇总表。

DMA 通道 21 REQ 的输出选择

图 5-96. UDMACH21BSEL 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-64h						

表 5-102. UDMACH21BSEL 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	64h	只读选择值 64h =软件事件 0，由 SWEV.SWEV0 触发

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.88 UDMACH22SSEL 寄存器（偏移 = 5B0h）[复位 = 00000065h]

图 5-97 展示了 UDMACH22SSEL，表 5-103 对其进行了介绍。

返回到汇总表。

DMA 通道 22 SREQ 的输出选择

图 5-97. UDMACH22SSEL 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-65h						

表 5-103. UDMACH22SSEL 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	65h	只读选择值 65h =软件事件 1，由 SWEV.SWEV1 触发

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.89 UDMACH22BSEL 寄存器（偏移 = 5B4h）[复位 = 00000065h]

图 5-98 展示了 UDMACH22BSEL，表 5-104 对其进行了介绍。

返回到汇总表。

DMA 通道 22 REQ 的输出选择

图 5-98. UDMACH22BSEL 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-65h						

表 5-104. UDMACH22BSEL 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	65h	只读选择值 65h =软件事件 1，由 SWEV.SWEV1 触发

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.90 UDMACH23SSEL 寄存器（偏移 = 5B8h）[复位 = 00000066h]

图 5-99 展示了 UDMACH23SSEL，表 5-105 对其进行了介绍。

返回到汇总表。

DMA 通道 23 SREQ 的输出选择

图 5-99. UDMACH23SSEL 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-66h						

表 5-105. UDMACH23SSEL 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	66h	只读选择值 66h =软件事件 2，由 SWEV.SWEV2 触发

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.91 UDMACH23BSEL 寄存器（偏移 = 5BCh）[复位 = 00000066h]

图 5-100 展示了 UDMACH23BSEL，表 5-106 对其进行了介绍。

返回到汇总表。

DMA 通道 23 REQ 的输出选择

图 5-100. UDMACH23BSEL 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-66h						

表 5-106. UDMACH23BSEL 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	66h	只读选择值 66h =软件事件 2，由 SWEV.SWEV2 触发

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.92 UDMACH24SSEL 寄存器（偏移 = 5C0h）[复位 = 00000067h]

图 5-101 展示了 UDMACH24SSEL，表 5-107 对其进行了介绍。

返回到汇总表。

DMA 通道 24 SREQ 的输出选择

图 5-101. UDMACH24SSEL 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-67h						

表 5-107. UDMACH24SSEL 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	67h	只读选择值 67h =软件事件 3，由 SWEV.SWEV3 触发

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.93 UDMACH24BSEL 寄存器（偏移 = 5C4h）[复位 = 00000067h]

图 5-102 展示了 UDMACH24BSEL，表 5-108 对其进行了介绍。

返回到汇总表。

DMA 通道 24 REQ 的输出选择

图 5-102. UDMACH24BSEL 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-67h						

表 5-108. UDMACH24BSEL 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	67h	只读选择值 67h =软件事件 3，由 SWEV.SWEV3 触发

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.94 GPT3ACAPTSEL 寄存器（偏移 = 600h）[复位 = 0000005Bh]

图 5-103 展示了 GPT3ACAPTSEL，表 5-109 对其进行了介绍。

返回到汇总表。

GPT3 0 的输出选择

图 5-103. GPT3ACAPTSEL 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R/W-5Bh						

表 5-109. GPT3ACAPTSEL 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 5-109. GPT3ACAPTSEL 寄存器字段说明（续）

位	字段	类型	复位	说明
6-0	EV	R/W	5Bh	读取/写入选择值 写入由 ENUM 定义的值以外的任何其他值可能会导致未定义的行为。 0h = 始终无效 4h = 来自 IOC 的边沿检测事件。通过 IOC:IOCFGn.EDGE_IRQ_EN 和 IOC:IOCFGn.EDGE_DET 设置进行配置 5h = 电池监测器的组合事件 6h = 通过振荡器控制的组合事件 7h = 来自 AON_RTC 的事件，通过 AON_RTC:CTL.COMB_EV_MASK 设置来控制 9h = 来自 I2C 的中断事件 Bh = AUX 组合事件，相应的标志寄存器在这里为 AUX_EVCTL:EVTOMCUFLAGS 15h = 闪存控制器错误事件，状态标志为 FLASH:FEDACSTAT.FSM_DONE 和 FLASH:FEDACSTAT.RVF_INT 19h = RFC 门铃命令确认中断，相当于 RFC_DBELL:RFACKIFG.ACKFLAG 1Ah = 组合的 RFC 硬件中断，相应的标志在此处为 RFC_DBELL:RFHWIFG 1Bh = CPE 生成的事件的组合中断。在这里对应的标志是 RFC_DBELL:RFCPEIFG。只有在 RFC_DBELL:RFCPEIFG 中通过 CPE0 选择的中断才能触发 RFC_CPE_0 事件 1Eh = CPE 生成的事件的组合中断。在这里对应的标志是 RFC_DBELL:RFCPEIFG。只有在 RFC_DBELL:RFCPEIFG 中通过 CPE1 选择的中断才能触发 RFC_CPE_1 事件 22h = SSI0 组合中断，在此处找到的中断标志为 SSI0:MIS 23h = SSI1 组合中断，在此处找到的中断标志为 SSI1:MIS 24h = UART0 组合中断，在此处找到的中断标志为 UART0:MIS 25h = UART1 组合中断，在此处找到的中断标志为 UART1:MIS 38h = AUX Timer2 事件 0，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_EV0 39h = AUX Timer2 事件 1，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_EV1 3Ah = AUX Timer2 事件 2，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_EV2 3Bh = AUX Timer2 事件 3，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_EV3 3Ch = AUX Timer2 脉冲，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_PULSE 3Dh = GPT0A 比较事件。通过 GPT0:TAMR.TCACT 进行配置 3Eh = GPT0B 比较事件。通过 GPT0:TBMR.TCACT 进行配置 3Fh = GPT1A 比较事件。通过 GPT1:TAMR.TCACT 进行配置 40h = GPT1B 比较事件。通过 GPT1:TBMR.TCACT 进行配置 41h = GPT2A 比较事件。通过 GPT2:TAMR.TCACT 进行配置 42h = GPT2B 比较事件。通过 GPT2:TBMR.TCACT 进行配置 43h = GPT3A 比较事件。通过 GPT3:TAMR.TCACT 进行配置 44h = GPT3B 比较事件。通过 GPT3:TBMR.TCACT 进行配置 5Bh = 来自 IOC 的端口捕获事件，通过 IOC:IOCFGn.PORT_ID 进行配置。通过 ENUM PORT_EVENT6 配置的端口上的事件将在此处路由。 5Ch = 来自 IOC 的端口捕获事件，通过 IOC:IOCFGn.PORT_ID 进行配置。通过 ENUM PORT_EVENT7 配置的端口上的事件将在此处路由。 69h = AON 唤醒事件，相应的标志在这里为 AUX_EVCTL:EVTOMCUFLAGS.AUX_WU_EV 6Ah = AUX 比较 A 事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_COMPA 6Bh = AUX 比较 B 事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_COMPB 6Ch = AUX TDC 测量完成事件，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TDC_DONE and the AUX_TDC 状态 AUX_TDC:STAT.DONE

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 5-109. GPT3ACAPTSEL 寄存器字段说明（续）

位	字段	类型	复位	说明
				6Dh = AUX 计时器 0 事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER0_EV 6Eh = AUX 计时器 1 事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER1_EV 6Fh = 来自 AUX 信标的自动获取事件，通过 AUX_SMPH:AUTOTAKE 配置 70h = AUX ADC 完成，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_ADC_DONE 71h = AUX ADC FIFO 水印事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_ADC_FIFO_ALMOST_FULL 72h = OBSMUX0 通过 AUX 环回，对应于 AUX_EVCTL:EVTOMCUFLAGS.MCU_OBSMUX0 73h = AUX ADC 中断事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_ADC_IRQ。在这里找到的状态标志为 AUX_EVCTL:EVTOMCUFLAGS 77h = 通过 AON_RTC:CTL.RTC_UPD_EN 控制的 RTC 周期性事件 79h = 始终置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.95 GPT3BCAPTSEL 寄存器（偏移 = 604h）[复位 = 0000005Ch]

图 5-104 展示了 GPT3BCAPTSEL，表 5-110 对其进行了介绍。

返回到汇总表。

GPT3 1 的输出选择

图 5-104. GPT3BCAPTSEL 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R/W-5Ch						

表 5-110. GPT3BCAPTSEL 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 5-110. GPT3BCAPTSEL 寄存器字段说明（续）

位	字段	类型	复位	说明
6-0	EV	R/W	5Ch	读取/写入选择值 写入由 ENUM 定义的值以外的任何其他值可能会导致未定义的行为。 0h = 始终无效 4h = 来自 IOC 的边沿检测事件。通过 IOC:IOCFGn.EDGE_IRQ_EN 和 IOC:IOCFGn.EDGE_DET 设置进行配置 5h = 电池监测器的组合事件 6h = 通过振荡器控制的组合事件 7h = 来自 AON_RTC 的事件，通过 AON_RTC:CTL.COMB_EV_MASK 设置来控制 9h = 来自 I2C 的中断事件 Bh = AUX 组合事件，相应的标志寄存器在这里为 AUX_EVCTL:EVTOMCUFLAGS 15h = 闪存控制器错误事件，状态标志为 FLASH:FEDACSTAT.FSM_DONE 和 FLASH:FEDACSTAT.RVF_INT 19h = RFC 门铃命令确认中断，相当于 RFC_DBELL:RFACKIFG.ACKFLAG 1Ah = 组合的 RFC 硬件中断，相应的标志在此处为 RFC_DBELL:RFHWIFG 1Bh = CPE 生成的事件的组合中断。在这里对应的标志是 RFC_DBELL:RFCPEIFG。只有在 RFC_DBELL:RFCPEIFG 中通过 CPE0 选择的中断才能触发 RFC_CPE_0 事件 1Eh = CPE 生成的事件的组合中断。在这里对应的标志是 RFC_DBELL:RFCPEIFG。只有在 RFC_DBELL:RFCPEIFG 中通过 CPE1 选择的中断才能触发 RFC_CPE_1 事件 22h = SSI0 组合中断，在此处找到的中断标志为 SSI0:MIS 23h = SSI1 组合中断，在此处找到的中断标志为 SSI1:MIS 24h = UART0 组合中断，在此处找到的中断标志为 UART0:MIS 25h = UART1 组合中断，在此处找到的中断标志为 UART1:MIS 38h = AUX Timer2 事件 0，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_EV0 39h = AUX Timer2 事件 1，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_EV1 3Ah = AUX Timer2 事件 2，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_EV2 3Bh = AUX Timer2 事件 3，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_EV3 3Ch = AUX Timer2 脉冲，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER2_PULSE 3Dh = GPT0A 比较事件。通过 GPT0:TAMR.TCACT 进行配置 3Eh = GPT0B 比较事件。通过 GPT0:TBMR.TCACT 进行配置 3Fh = GPT1A 比较事件。通过 GPT1:TAMR.TCACT 进行配置 40h = GPT1B 比较事件。通过 GPT1:TBMR.TCACT 进行配置 41h = GPT2A 比较事件。通过 GPT2:TAMR.TCACT 进行配置 42h = GPT2B 比较事件。通过 GPT2:TBMR.TCACT 进行配置 43h = GPT3A 比较事件。通过 GPT3:TAMR.TCACT 进行配置 44h = GPT3B 比较事件。通过 GPT3:TBMR.TCACT 进行配置 5Bh = 来自 IOC 的端口捕获事件，通过 IOC:IOCFGn.PORT_ID 进行配置。通过 ENUM PORT_EVENT6 配置的端口上的事件将在此处路由。 5Ch = 来自 IOC 的端口捕获事件，通过 IOC:IOCFGn.PORT_ID 进行配置。通过 ENUM PORT_EVENT7 配置的端口上的事件将在此处路由。 69h = AON 唤醒事件，相应的标志在这里为 AUX_EVCTL:EVTOMCUFLAGS.AUX_WU_EV 6Ah = AUX 比较 A 事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_COMPA 6Bh = AUX 比较 B 事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_COMPB 6Ch = AUX TDC 测量完成事件，对应于标志 AUX_EVCTL:EVTOMCUFLAGS.AUX_TDC_DONE and the AUX_TDC 状态 AUX_TDC:STAT.DONE

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 5-110. GPT3BCAPTSEL 寄存器字段说明（续）

位	字段	类型	复位	说明
				6Dh = AUX 计时器 0 事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER0_EV 6Eh = AUX 计时器 1 事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_TIMER1_EV 6Fh = 来自 AUX 信标的自动获取事件，通过 AUX_SMPH:AUTOTAKE 配置 70h = AUX ADC 完成，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_ADC_DONE 71h = AUX ADC FIFO 水印事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_ADC_FIFO_ALMOST_FULL 72h = OBSMUX0 通过 AUX 环回，对应于 AUX_EVCTL:EVTOMCUFLAGS.MCU_OBSMUX0 73h = AUX ADC 中断事件，对应于 AUX_EVCTL:EVTOMCUFLAGS.AUX_ADC_IRQ。在这里找到的状态标志为 AUX_EVCTL:EVTOMCUFLAGS 77h = 通过 AON_RTC:CTL.RTC_UPD_EN 控制的 RTC 周期性事件 79h = 始终置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.96 AUXSEL0 寄存器（偏移 = 700h）[复位 = 00000010h]

图 5-105 展示了 AUXSEL0，表 5-111 对其进行了介绍。

返回到汇总表。

AUX 订阅者 0 的输出选择

图 5-105. AUXSEL0 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R/W-10h						

表 5-111. AUXSEL0 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R/W	10h	读取/写入选择值 写入由 ENUM 定义的值以外的任何其他值可能会导致未定义的行为。 0h = 始终无效 Ch = GPT2A 中断事件，由 GPT2:TAMR 控制 Dh = GPT2B 中断事件，由 GPT2:TBMR 控制 Eh = GPT3A 中断事件，由 GPT3:TAMR 控制 Fh = GPT3B 中断事件，由 GPT3:TBMR 控制 10h = GPT0A 中断事件，由 GPT0:TAMR 控制 11h = GPT0B 中断事件，由 GPT0:TBMR 控制 12h = GPT1A 中断事件，由 GPT1:TAMR 控制 13h = GPT1B 中断事件，由 GPT1:TBMR 控制 3Dh = GPT0A 比较事件。通过 GPT0:TAMR.TCACT 进行配置 3Eh = GPT0B 比较事件。通过 GPT0:TBMR.TCACT 进行配置 3Fh = GPT1A 比较事件。通过 GPT1:TAMR.TCACT 进行配置 40h = GPT1B 比较事件。通过 GPT1:TBMR.TCACT 进行配置 41h = GPT2A 比较事件。通过 GPT2:TAMR.TCACT 进行配置 42h = GPT2B 比较事件。通过 GPT2:TBMR.TCACT 进行配置 43h = GPT3A 比较事件。通过 GPT3:TAMR.TCACT 进行配置 44h = GPT3B 比较事件。通过 GPT3:TBMR.TCACT 进行配置 79h = 始终置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.97 CM3NMISEL0 寄存器（偏移 = 800h）[复位 = 00000063h]

图 5-106 展示了 CM3NMISEL0，表 5-112 对其进行了介绍。

返回到汇总表。

NMI 订阅者 0 的输出选择

图 5-106. CM3NMISEL0 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R-63h						

表 5-112. CM3NMISEL0 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R	63h	只读选择值 63h =看门狗不可屏蔽中断事件，通过 WDT:CTL.INTTYPE 控制

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.98 I2SSTMPSEL0 寄存器（偏移 = 900h）[复位 = 0000005Fh]

图 5-107 展示了 I2SSTMPSEL0，表 5-113 对其进行了介绍。

返回到汇总表。

I2S 订阅者 0 的输出选择

图 5-107. I2SSTMPSEL0 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R/W-5Fh						

表 5-113. I2SSTMPSEL0 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R/W	5Fh	读取/写入选择值 写入由 ENUM 定义的值以外的任何其他值可能会导致未定义的行为。 0h = 始终无效 79h = 始终置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.99 FRZSEL0 寄存器（偏移 = A00h）[复位 = 00000078h]

图 5-108 展示了 FRZSEL0，表 5-114 对其进行了介绍。

返回到汇总表。

FRZ 订阅者的输出选择
暂停的调试信号会传递到通用计时器、具有数字及模拟外设 (AUX) 的传感器控制器、无线电及 RTC 等外设。当系统 CPU 停止时，已连接且启用了冻结功能的外设也会停止。可编程输出可设置为静态值 0 或 1，也可设置为传递暂停的信号。

图 5-108. FRZSEL0 寄存器

31	30	29	28	27	26	25	24	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																									EV						
R-0h																									R/W-78h						

表 5-114. FRZSEL0 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	EV	R/W	78h	读取/写入选择值 写入由 ENUM 定义的值以外的任何其他值可能会导致未定义的行为。 0h = 始终无效 78h = CPU 停止 79h = 始终置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5.7.2.100 SWEV 寄存器（偏移 = F00h）[复位 = 00000000h]

图 5-109 展示了 SWEV，表 5-115 中对此进行了介绍。

返回到汇总表。

设置或清除软件事件

图 5-109. SWEV 寄存器

31	30	29	28	27	26	25	24
RESERVED							SWEV3
R-0h							R/W-0h
23	22	21	20	19	18	17	16
RESERVED							SWEV2
R-0h							R/W-0h
15	14	13	12	11	10	9	8
RESERVED							SWEV1
R-0h							R/W-0h
7	6	5	4	3	2	1	0
RESERVED							SWEV0
R-0h							R/W-0h

表 5-115. SWEV 寄存器字段说明

位	字段	类型	复位	说明
31-25	RESERVED	R	0h	保留
24	SWEV3	R/W	0h	当值为“0”时，向该位写入“1”会触发软件 3 事件。
23-17	RESERVED	R	0h	保留
16	SWEV2	R/W	0h	当值为“0”时，向该位写入“1”会触发软件 2 事件。
15-9	RESERVED	R	0h	保留
8	SWEV1	R/W	0h	当值为“0”时，向该位写入“1”会触发软件 1 事件。
7-1	RESERVED	R	0h	保留
0	SWEV0	R/W	0h	当值为“0”时，向该位写入“1”会触发软件 0 事件。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

章节 6

JTAG 接口

本章介绍了用于片上调试支持的 cJTAG 和 JTAG 接口。

表 6-1. 参考资料

ID	说明
[JTAG 1]	IEEE 标准测试接入端口和边界扫描架构、IEEE 标准 1149.1a 1993 和补充标准 1149.1b 1994、电气和电子工程师学会
[JTAG 2]	IEEE 1149.7 减少引脚和增强功能测试接入端口和边界扫描架构标准

6.1 顶层调试系统 481
6.2 cJTAG 483
6.3 ICEPick 489
6.4 ICEMelter 499
6.5 串行线查看器 (SWV) 500
6.6 引导中暂停 (HIB) 500
6.7 调试与关断 500
6.8 通过 WUC TAP 支持的调试功能 501
6.9 分析器寄存器 501
6.10 边界扫描 503

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

6.1 顶层调试系统

在 CC13x2 和 CC26x2 器件平台中，调试子系统采用两种用于调试和测试的 IEEE 标准：

- IEEE 标准 1149.1：标准测试接入端口和边界扫描体系结构测试接入端口 (TAP) [JTAG 1]。这个标准以首字母缩略词 JTAG 为名。
- 4 级 IEEE 1149.7：缩减引脚和增强功能测试接入端口和边界扫描架构的标准 [JTAG 2]。此项以首字母缩略词 cJTAG（紧凑型 JTAG）为名。此标准使用各种压缩格式对 IEEE 1149.1 事务进行串行化，以减少实现 JTAG 调试端口所需的引脚数。

调试子系统还实现了防火墙，用于防止未经授权访问调试/测试端口。图 6-1 显示了调试子系统的框图。

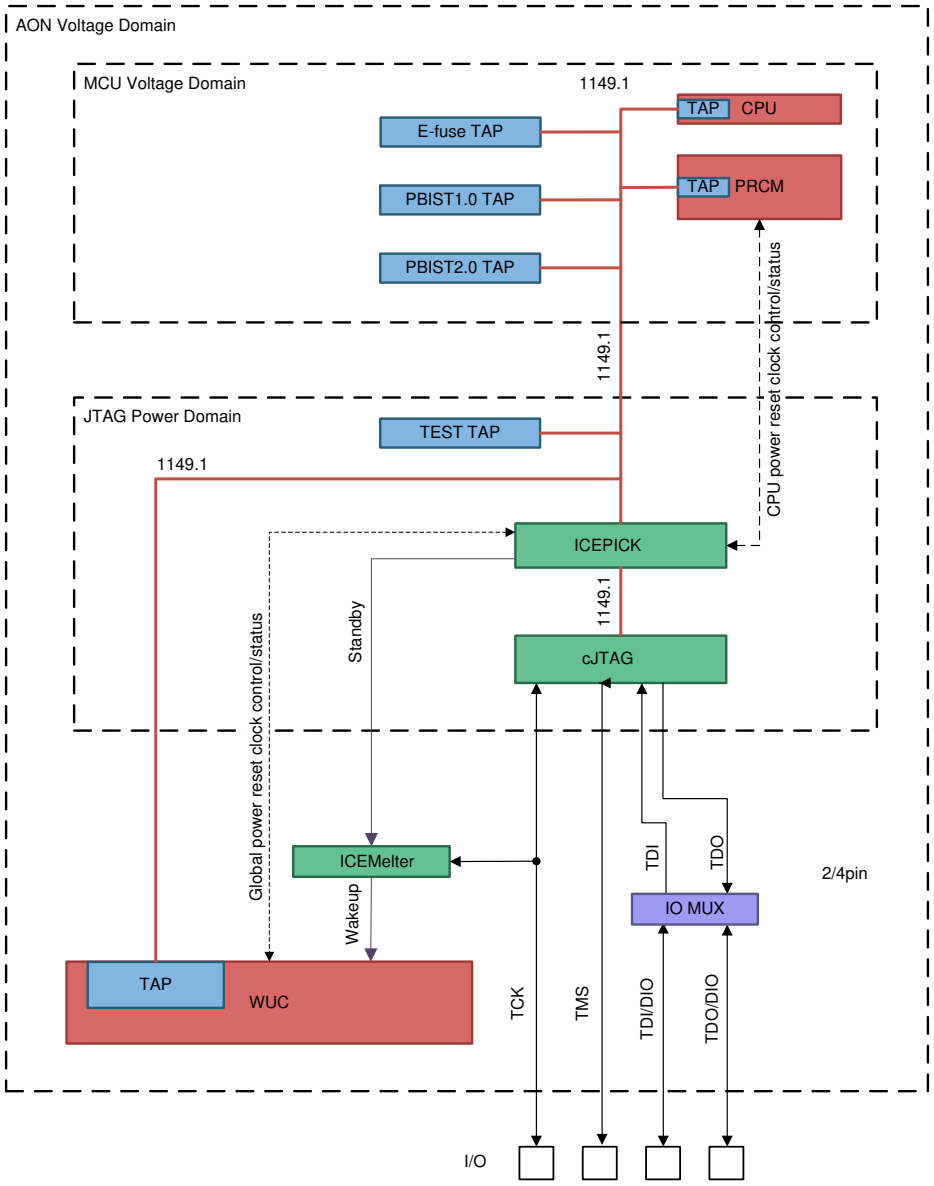


图 6-1. 顶层调试系统

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

IEEE 1149.1 TAP 使用以下信号来支持操作：

- **TCK（测试时钟）：**该信号同步内部状态机运行。
- **TMS（测试模式选择）：**此信号在 TCK 的上升沿进行采样，以确定下一个状态。
- **TDI（测试数据输入）：**该信号表示移入器件测试或编程逻辑的数据。当内部状态机处于正确状态时，将在 TCK 的上升沿对 TDI 进行采样。
- **TDO（测试数据输出）：**此信号表示从器件的测试或编程逻辑移出的数据，当内部状态机处于正确状态时，在 TCK 的下降沿有效。

TRST 没有专用的 I/O 引脚。调试子系统通过系统级复位和上电复位来进行复位。

TAP 控制器是一种状态机，它的转换由 TMS 信号控制，用于控制 JTAG 系统的行为。图 6-2 展示了 JTAG 的状态转换图。

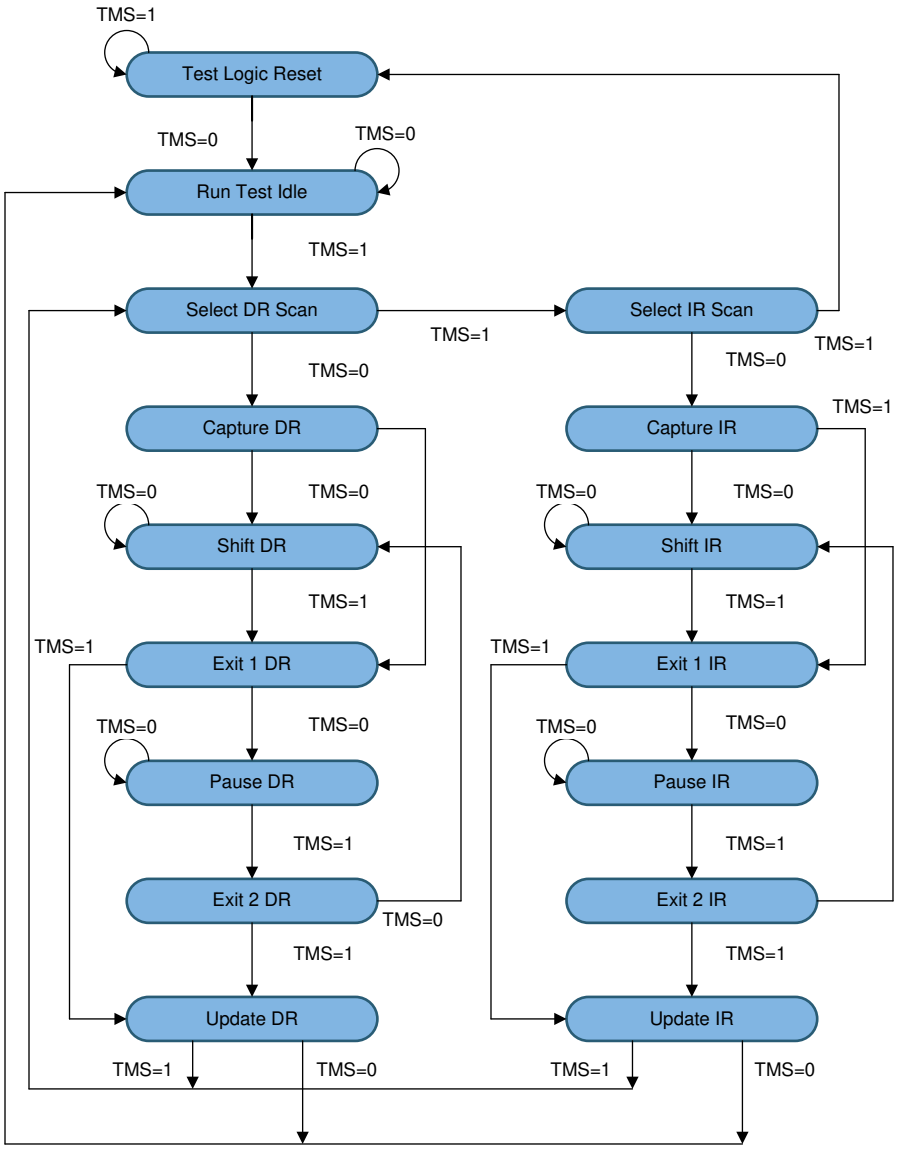


图 6-2. JTAG 状态机

每个状态都有两个退出条件，因此所有转换都可由在 TCK 上采样的单个 TMS 信号进行控制。两条主路径允许进行设置或从器件的数据寄存器 (DR) 或指令寄存器 (IR) 检索信息。数据寄存器取决于加载到指令寄存器中的值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

6.2 cJTAG

此模块实现了符合 IEEE 1149.7 标准的紧凑型 JTAG (cJTAG) 适配器，该适配器在 IEEE 1149.1 JTAG 测试接入端口 (TAP) 上运行一个 2 引脚通信协议。上电后，仅使用 TCK 和 TMS 的 2 引脚 JTAG 模式是默认配置。CC13x2 及 CC26x2 器件平台中的 cJTAG 配置实现了等级 4 功能扫描模式的子集。等级 4 继承等级 0、1、2 和 3 的功能（表 6-2 中提到的功能除外）。图 6-3 显示了 cJTAG 模块的概念图。

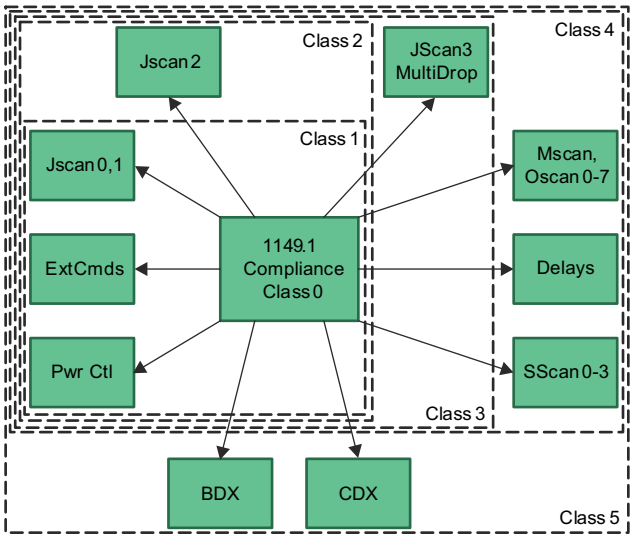


图 6-3. cJTAG 概念图

- **等级 0：**严格符合 IEEE 1149.1 规范，具有内部 TAP 选择功能
- **等级 1：**添加了 cJTAG 命令协议，一些可选的离散命令
- **等级 2：**增加串行选择功能
- **等级 3：**添加了 JTAG 星型配置、控制器 ID 及扫描选择指令
- **等级 4：**添加高级扫描协议

表 6-2 列出了 CC13x2 和 CC26x2 器件平台中支持的 IEEE 1149.7 中的功能。CC13x2 及 CC26x2 器件平台中的 cJTAG 模块支持 12 种扫描格式。扫描格式使用多种压缩协议（每位 1 到 4 个时钟）对每个数据包进行序列化。

表 6-2. IEEE 1149.7 特性子集

	IEEE 1149.7 特性	通过 cJTAG 提供器件支持	注释
配置	4 级 TAP	是	支持 2 引脚操作
	5 级 TAP	否	用于后台数据传输的数据和自定义通道
可选元件	FRST	否	功能复位
	TRST	否	测试复位
	RDBK 能力	否	读回寄存器数据
	辅助引脚功能	是	重复使用 TDI 及 TDO 引脚
	TCKWID	否	可编程 TCK 宽度
功率控制	断电逻辑	否	关断 cJTAG 模块的逻辑功能

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 6-2. IEEE 1149.7 特性子集（续）

	IEEE 1149.7 特性	通过 cJTAG 提供器件支持	注释
扫描格式	JScan0	是	并行模式
	JScan1	是	与防火墙并行
	JScan2	是	并联超级旁路选择
	JScan3	是	并联寄存器选择
	MScan	否	多器件模式，支持失速
	OScan0	是	支持失速
	OScan1	是	非失速模式
	OScan2	是	双向传输，流水线型
	OScan3	是	仅主机到目标，流水线型
	OScan4	是	支持失速
	OScan5	是	流水线型
	OScan6	是	双向传输，流水线型
	OScan7	是	仅主机到目标，流水线型
	SScan0	否	分段扫描
	SScan1	否	分段扫描，支持失速
	SScan2	否	分段扫描
	SScan3	否	分段扫描，支持失速

表 6-3. OScan 扫描数据包内容

扫描格式	非移位状态				移位状态			
OScan0	nTDI	TMS	RDY	TDO	nTDI	TMS	RDY	TDO
OScan1	nTDI	TMS		TDO	nTDI	TMS		TDO
OScan2		TMS			nTDI	TMS		TDO
OScan3		TMS			nTDI	TMS		
OScan4	nTDI	TMS	RDY	TDO	nTDI		RDY	TDO
OScan5	nTDI	TMS		TDO	nTDI			TDO
OScan6		TMS			nTDI	TMS(1)		TDO
OScan7		TMS			nTDI			

(1) 移位的第一个数据包存在 TMS。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

6.2.1 cJTAG 命令

cJTAG 命令通过良性的 JTAG 扫描活动传输。

以下是三个基本步骤：

1. 加载惰性操作码
2. 设置控制级别 2
3. 发出命令

在发出 cJTAG 命令之前，控制器必须确保扫描活动不会在器件中引发任何意外操作。为此，必须将 BYPASS 或 IDCODE 等惰性操作码加载到指令寄存器中。通常会使用旁路，因为其值（全 1）由 IEEE 1149.1 规范规定。

通过执行两次零位扫描 (ZBS)，然后进行一次 1 位移位，可以启用命令检测。ZBS 定义为一种扫描序列，它遍历“捕获 DR”状态并最终到达“更新 DR”状态，绝不会进入“移位 DR”状态。扫描序列可以进入“暂停 DR”状态并持续任意多个时钟周期，或者完全跳过“暂停 DR”状态。每个连续的 ZBS 都会使控制级别递增。当第一个“移位 DR”状态发生时，控制级别被锁定。

控制级别锁定后，命令通过成对 DR 扫描发出，有时还需要第三次 DR 扫描。每次扫描都会对在移位 DR 状态中花费的时钟周期数进行计数（从 0 到 31 个时钟）。第一次 DR 扫描，即命令部分 0 (CP0)，形成命令的操作码。第二次 DR 扫描，即命令部分 1 (CP1)，提供有关命令的附加信息。这些信息可能是更多的操作码位或数据字段，具体取决于操作码。

有三个命令（SCNB、SCNS 和 CIDA）需要第三次 DR 扫描，即命令部分 2 (CP2)，以将数据传入或传出器件。表 6-4 列出了这些命令。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 6-4. cJTAG 命令

操作码	指令		
00000	STMC 存储其他控制 操作数：**bbbxy**		
	bbb		
	0	状态控制	
		xy	
		0	NOP
		1	ExitCmdLev (ECL)
		2	退出/暂停 (SUSPEND = 1)
		3	ZBS 禁止 (ZBSINH = 1)
	1	扫描控制	
		x	
		0	扫描组候选 (SGC) SGC = y
		1	条件组成员 (CGM) CGM = y
	2	就绪控制 RDYC = **xy** 使用非 MScan 的扫描格式时，SP 有效载荷最后一位之前的逻辑 1 RDY 位数量为 **xy** + 1	
	3	延迟控制 (DLYC) DLYC = **xy**	
		xy	
		0	未添加 DTS 延迟。
		1	添加一个 TCKC 信号周期。
		2	添加两个 TCKC 信号周期。
		3	添加可变的 TCKC 信号周期数量。
	4-7	保留	
00001	STC1 存储条件 1 位 操作数：**cbbbv**		
	bbb		
	0	采样边沿 (SEDGE) 定义用于对 TMSC 信号输入进行采样的 TCKC 信号边沿 SEDGE == 0：在 TCKC 信号下降沿对 TMSC 信号进行采样 SEDGE == 1：在 TCKC 信号上升沿对 TMSC 信号进行采样	
		c	
		0	SEDGE = **v**
		1	如果 CGM == 1，则 SEDGE = **v**
	1-7	保留	

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 6-4. cJTAG 命令（续）

操作码	指令		
00010	STC2 存储条件 2 位 操作数：**cbbvv**		
	bb		
	0-1	保留	
	2	辅助引脚功能控制 (APFC)	
		APFC==00： APFC==01： APFC== 1x：	默认引脚功能无变化。 引脚功能变成标准引脚功能。 引脚功能变成辅助引脚功能。
		C	
		0	APFC = **vv**
		1	如果 CGM == 1，则 APFC = **vv**
	3	保留	
00011	STFMT 存储扫描格式 操作数：**nnnnn**		
	nnnnn		
	0	JSCAN0	
	1	JSCAN1	
	2	JSCAN2	
	3	JSCAN3	
	4-7	保留	
	8	OSCAN0	
	9	OSCAN1	
	10	OSCAN2	
	11	OSCAN3	
	12	OSCAN4	
	13	OSCAN5	
	14	OSCAN6	
	15	OSCAN7	
	16	保留	
00100	MSS 做出扫描选择 操作数：**miiii**		
	m		
	0	置位目标控制器的 SGC 位 清除非目标控制器的 SGC 位	
	1	置位目标控制器的 SGC 位 非目标控制器的 SGC 位不受影响	
00101-00110	保留		
00111	CCE 条件命令使能 操作数：**miiii**		
	m		
	0	置位目标控制器的 CGM 位 清除非目标控制器的 CGM 位	
	1	置位目标控制器的 CGM 位 非目标控制器的 CGM 位不受影响	

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 6-4. cJTAG 命令（续）

操作码	指令	
01000	SCNB 扫描位 操作数：**yyyyy** + CR 扫描	
	yyyyy	
	00	SGC，扫描组候选、写入
	01	CGM，条件组成员，写入
	02-05	CNFG0-3，TAP.7 控制器类，读取
	06-31	保留
01001-11111	保留	

6.2.1.1 必需命令

有三个强制命令用于管理命令处理。这些命令是 STMC 的子命令，包括 Exit Command Level、Suspend 和 ZBSINH。如果器件将 ZBS 用于其自己的用途，则可以使用最后两个命令。

- 退出命令级别将终止命令处理。
- 暂停禁止命令检测，直到检测到特殊序列。
- 在复位之前，ZBSINH 会抑制命令检测。

有三个强制命令可用于管理命令处理。

6.2.2 编程序列

6.2.2.1 打开命令窗口

在 cJTAG 模块接受任何命令之前，必须将控制级别设置为 2 并锁定。

1. 扫描 IR（旁路，结束于 Pause DR）：将良性操作码加载到指令寄存器中。
2. 转到扫描（通过 Update DR，结束于 Pause DR）：这是第一次 ZBS。
3. 转到扫描（通过 Update DR，结束于 Pause DR）：这是第二次 ZBS。
4. 扫描 DR（1 位，结束于 Pause DR）：这将控制级别锁定为 2。

打开命令窗口会解耦器件 TAP；解耦发生在第二次 ZBS 发生时。

6.2.2.2 更改为 4 引脚模式

当命令窗口打开时，便可以发出命令。要更改为 4 引脚模式，必须将 APFC 写为 1（使用 STC2 命令），前提是 TAP 状态从 Pause DR 开始。

1. Scan DR（2 位 1，以 Pause DR 结束）：将 2 加载到 CP0 中。
2. Goto Scan（经由 Update DR 至 Pause DR）：通过更新完成 CP0。
3. Scan DR（9 位 1，以 Pause DR 结束）：将 9 加载到 CP1 中。
4. Goto Scan（经由 Update DR 至 Pause DR）：通过更新完成 CP1。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

6.2.2.3 关闭命令窗口

可通过执行 IR 扫描、测试逻辑复位或 ECL 命令来关闭命令窗口。ECL 命令是 STMC（操作码 0）命令的子命令。ECL 命令假定 TAP 状态从 Pause DR 开始。

1. 转到扫描（通过更新 DR 来暂停 DR）：执行零位扫描以使用 0 加载 CP0。
2. 扫描 DR（1 位，于 Pause DR 结束）：使用 1 加载 CP1。
3. 转到扫描（通过更新 DR 来暂停 DR）：通过更新完成 CP1。

备注

命令窗口关闭时，器件 TAP 会进行耦合，因此会向器件 TAP 发出任何后续扫描（IR 或 DR）。

6.3 ICEPick

ICEPick 是芯片中的主要 TAP。它充当芯片上符合 IEEE 1149.1 JTAG 标准的顶级路由器。从概念上讲，ICEPick 可以被视为一组开关，它们可以将模块级 TAP 与更高级别芯片 TAP 连接或隔离。模块级 TAP 称为次级 TAP，而主要 TAP 和外部 JTAG 信号称为主扫描路径。ICEPick TAP 显示为第一个 TAP，并且仅在开机后显示为扫描路径中唯一 TAP。主扫描路径中没有已选或可见的次要 TAP。从外部 JTAG 接口的角度来看，未选择的次级 TAP 似乎不存在。ICEPick TAP 具有多个自身的扫描路径，以支持次要 TAP 选择、控制和状态。ICEPick 启用动态扫描链管理，可以选择一个或多个从 TAP 并将它们链接到扫描链中。

ICEPick 内的每个次级 TAP 都有多个相关联的控制位。其中一些位仅适用于 ICEPick 管理的 TAP，而其他位则适用于次要 TAP 所在的整个子系统或电源域。这些控制位处理包含在扫描路径中的 TAP 选择、次级 TAP 测试复位管理和所需的调试关注点。

ICEPick 内的每个次级 TAP 都有多个相关联的状态位。这些状态位报告可访问性、可见性、功率及时钟状态。

在调试应用和使用 2 引脚模式的片上 cJTAG TAP 之间建立连接后，可以将通信协议更改为 4 引脚配置。当 cJTAG 切换到 4 引脚模式时，TDI 和 TDO 通过 IOC 自动映射到引脚，并且这优先于在切换发生之前映射到相应 DIO 的任何其他功能。还支持从 4 引脚切换至 2 引脚模式。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

6.3.1 次级 TAP

每个次级 TAP 都分配了一个编号。TAP 编号为线性，并以 0 开头。分配给次级 TAP 的编号与它在 ICEPick 中的次级控制和状态寄存器中的位置相对应。第一个选定的 TAP 是数值最低的 TAP，而最后一个选定的 TAP 是数字最高的 TAP。ICEPick 模块有一个防火墙，用于防范未经授权访问从属分路器。表 6-5 列出了可用的 TAP，对应的顺序以及这些 TAP 对于最终用户的可用性。通过写入客户配置区域中的相应字段，可以锁定打开的 TAP。

表 6-5. 从 TAP 顺序

编号	测试 TAP 名称	说明	最终用户的可用性
测试组			
0	测试	DFT 功能和分析器	请参阅 (1)
1	PBIST1.0	RAM BIST 控制器接口	已锁定
2	PBIST2.0	ROM BIST 控制器接口	已锁定
3	电子保险丝	用于 SRAM 修复的电子保险丝接口	已锁定
4	保留	保留	保留
5	AON WUC	VD 覆盖控制/状态	请参阅 (2)
调试组			
0	CM4F	用于 Cortex-M4F 调试的 DAP	请参阅 (2) (3)

(1) 除了功耗分析器寄存器外，TEST TAP 中的所有功能都针对最终用户锁定。通过写入客户配置区域中的相应字段，可以阻止对功耗分析器寄存器的访问（请参阅节 6.9）。
(2) AON WUC TAP 中的一些功能是为最终用户打开的。这包括用于请求芯片擦除、系统复位及 MCU 复位的寄存器。
(3) 通过写入客户配置区域中的相应字段，可以阻止对 CPU 调试端口的访问（请参阅节 6.9）。

6.3.1.1 从器件 DAP (CPU DAP)

调试子系统只有一个从 DAP (CPU DAP)。该调试端口实现了串行线 JTAG 调试端口 (SWJ-DP) 接口，允许外部访问高级高性能总线访问端口 (AHB-AP) 接口，从而在 CPU 中进行调试访问。

SWJ-DP 是一个标准 Arm CoreSight™ 调试端口，它结合了 JTAG-DP 和串行线调试端口 (SW-DP)。尽管 SWJ-DP 支持 SW-DP 接口，但 CC13x2 和 CC26x2 器件平台不使用此模式。主要原因是，在存在 2 引脚 JTAG (1149.7) 模式的情况下，SW-DP 会在设计上变得冗余。

6.3.1.2 从属 TAP 和 DAP 的排序

- 当仅选择一个次级 TAP 时，它实际上连接到 ICEPick TAP 的 TDO。
- 当选择一个或多个次级 TAP 时，它们从编号最低的 TAP 到编号最高的 TAP 依次链接。
- 所选的编号最低的 TAP 连接位置最靠近器件级 TDI（ICEPick 除外），而编号最高的 TAP 连接位置最靠近器件级 TDO。
- 测试组内的任何选定 TAP 都先于调试组内的任何 TAP（例如 DAP）链接。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

6.3.2 ICEPick 寄存器

表 6-6 列出了 ICEPick 中的控制和状态寄存器。

表 6-6. 寄存器汇总

寄存器	缩写	宽度	编号	说明
数据移位寄存器	DSR	32	1	TAP 数据寄存器
指令寄存器	IR	6	1	TAP 指令寄存器
旁路寄存器	旁路	1	1	由 BYPASS 指令使用
器件标识寄存器	TAPID	32	1	与 IDCODE 一起使用的器件 ID
用户代码寄存器	UC	32	1	与 USERCODE 一起使用的用户代码
ICEPick 标识	IPID	32	1	ICEPick 的版本
连接	连接	7	1	连接代码
次级调试 TAP 寄存器 (SDTR)	SDTR	24	1	每个实例化的调试 TAP 都存在一个寄存器。它用于控制选择、电源、复位以及与每个 TAP 关联的时钟。
次级测试 TAP 寄存器 (STTR)	STTR	24	6	每个实例化的测试 TAP 存在一个寄存器。它用于控制每个 TAP 的选择。
保留	SUTR	24	1	保留
链接模式	LMR	24	1	指定 ICEPick 如何管理 TAP 选择。
ICEPick 控制	IPCR	24	1	常规 ICEPick 控制

6.3.2.1 IR 指令

ICEPick TAP 支持表 6-7 中列出的说明。所有未使用的 TAP 控制器指令都默认设置在旁路寄存器中。保留了多条指令，用于扩展 ICEPick 操作码。有关器件标识寄存器说明，请参阅节 6.3.2.5。

表 6-7. 指令寄存器操作码

IR	ICEPick 说明	访问
000000、111111	BYPASS	始终打开
10	ROUTER	已连接
100	IDCODE	始终打开
101	ICEPICKCODE	始终打开
111	CONNECT	始终打开
1000	USERCODE	始终打开
000001、000011、 000110、 001001 – 111110	保留	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

6.3.2.2 数据移位寄存器

图 6-4 是用于在 ICEPick TDI 和 TDO 之间移位的寄存器。该寄存器为 32 位宽。数据移位寄存器具有多个点移位，有助于在指令路径和几个数据路径上移位。

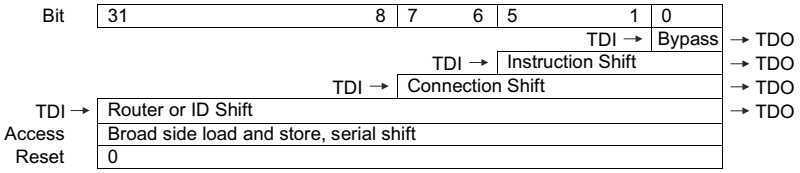


图 6-4. 数据移位寄存器

当请求移位时，每个位将 1 个位移入下一个低位。当最低有效位移出到 TDO 时，一个新值从 TDI 移入。移位寄存器根据当前 TAP 状态或指令寄存器中的值有几个插入点。

6.3.2.3 指令寄存器

此寄存器包含当前 TAP 指令。ICEPick IR 的宽度为 6 位（请参阅图 6-5）。

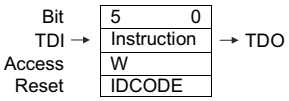


图 6-5. 指令寄存器

有关有效的 IR 操作码，请参阅表 6-7。

6.3.2.4 旁路寄存器

该寄存器是一个 1 位寄存器（请参阅图 6-6）。在 TDI 中扫描的值将保留并在一个 TCK 周期后从 TDO 中扫描出来。

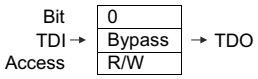


图 6-6. 旁路寄存器

6.3.2.5 器件标识寄存器

该寄存器允许通过 TAP 确定器件的制造商、器件型号和版本（请参阅图 6-7）。响应 IDCODE 指令时，会扫描器件标识寄存器。

IDCODE 有三个字段：版本、器件型号和制造商。

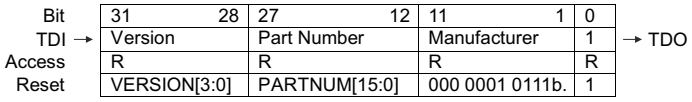


图 6-7. 器件标识寄存器

该寄存器的内容会复制到存储器所映射到的器件配置区域。有关该寄存器的详细信息，请参阅节 11.4.1 中的 FCFG1:ICEPICK_DEVICE_ID。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 6-8. 器件标识寄存器说明

字段	宽度	说明
版本	4	器件的修订版本
器件型号	16	器件型号
制造商	11	TI 的 JEDEC 库和公司代码：00000010111b
0	1	该位始终为 1。

6.3.2.6 用户代码寄存器

用户代码寄存器有助于区分由同一芯片构建的不同器件。用户代码寄存器的值通过电子保险丝设置。每个型号通过其特性集或引出的接口进行唯一标识。

用户代码寄存器是一个 32 位寄存器，用于指定组件的版本和器件型号。该寄存器的内容会复制到存储器所映射到的器件配置区域。有关该寄存器的详细信息，请参阅图 6-8 和表 6-9。

Bit	31 28	27 12	11 1	0	
TDI →	Version	Variant Number	Reserved	1	→ TDO
Access	R	R	R	R	
Reset	VERSION[3:0]	VARIANT[15:0]	0	1	

图 6-8. 用户代码寄存器

表 6-9. 用户代码寄存器说明

字段	宽度	说明
版本	4	器件的修订版本。每次修改器件的逻辑或屏蔽集时，此字段都必须更改。初始值为 0。
型号	16	芯片型号。此字段的解码结果显示在 FCFG1:USER_ID 中（请参阅节 11.4.1）。
保留	11	0
0	1	位 0 始终为 1

6.3.2.7 ICEPick 标识寄存器

该寄存器指示 ICEPick 模块的特性和版本（不要将 ICEPick IR 同器件 IR 混淆）。

ID 寄存器是一个 32 位寄存器，用于指定 ICEPick 模块的版本和特性。有关 ICEPick IR 的说明，请参见图 6-9 和表 6-10。

Bit	31 24	23 20	19 16	15 4	3 0	
TDI →	Version	Test TAPs	EMU TAPs	ICEPick Type	Capabilities	→ TDO
Access	R	R	R	R	R	
Reset	0x41	6	1	0x1CC	*	

图 6-9. ICEPick 标识寄存器

表 6-10. ICEPick 标识寄存器说明

字段	宽度	说明
版本	8	ICEPick 的修订版本
测试 TAP	4	测试 TAP 的数量
EMU TAP	4	EMU TAP 的数量
ICEpick 类型	12	ICEpick 类型的标识符 此字段设置为 0x1CC，对应于类型 C。
功能	4	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

6.3.2.8 连接寄存器

该寄存器可保护器件免受噪声、仿真器电缆热插拔或配置错误的扫描控制器意外扫描的影响。该寄存器可降低因噪声或意外扫描而意外触发调试功能的可能性。有关详细信息，请参阅图 6-10 和表 6-11。

Bit	7	6 4	3 0	
TDI →	Write	Reserved	ConnectKey	→ TDO
Access	W	R	R/W	
Reset	0	0	b0110	

图 6-10. 连接寄存器

表 6-11. 连接寄存器描述

位	字段	宽度	类型	复位	说明
7	写入启用	1	W	0	必须为 1 才能写入连接密钥。值为 0 表示读取。读取时，返回值为 0。
6-4	保留	3	R	0	保留
3-0	ConnectKey	4	R/W	0110	当该字段保持密钥代码 1001 时，扫描控制器被视为已连接。所有其他值均处于未连接状态。在此状态下，仅有限数量的 IR 指令有效。

6.3.3 路由器扫描链

该寄存器可访问所有 TAP 链接及控制寄存器。扫描链长度为 32 位。有关更多信息，请参阅图 6-11 和表 6-12。

Bit	31	30 28	27 24	23 0	
TDI →	Write Enable / Write Failure	Block Select	Register Number	Register Value	→TDO
Access	R/W	W	W	R/W	

图 6-11. ROUTER DR 扫描链

表 6-12. ROUTER DR 扫描链说明

位	字段	宽度	类型	复位	说明
31	写入启用	1	W	0	扫描输入时： 0：仅执行读取。 1：执行对指定寄存器的写入。 扫描输出时： 如果上一次扫描导致写入 ROUTER 寻址的寄存器，则当在下一次跳闸期间通过移位 DR 状态扫描出位 31 时，它会指示上一次写入是否成功。如果为 1，则上一次写入失败。如果为 0，则上一次写入成功。 写入调试或测试次级 TAP 控制和状态寄存器可能会因为以下几个原因而失败： • ICEPick 处于断开状态。 • TapPresent 位为 0，表示该位置不存在 TAP。 • TapEnable 位为 0，表示安全性或其他原因当前阻止访问该 TAP。 • 尚未处理先前对 ResetControl 或者 ReleaseFromWIR 位的编程。
30-28	块选择	3	R/W	000	块选择： 000：ICEPick 控制（请参阅节 6.3.4.1） 001：测试 TAP 链接控制块（请参阅节 6.3.4.2） 010：调试 TAP 链接控制块（请参阅节 6.3.4.3） 011 – 111：保留
27-24	寄存器编号	4	R/W	0000	此字段指定所选块中的寄存器（请参阅表 6-13、表 6-15 和表 6-20）。
23-0	选定的寄存器内容	24	–	–	根据块选择及寄存器编号字段中的值；相应的寄存器映射到此字段。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

在捕获 DR 状态期间，会检查数据移位寄存器。读取由块和寄存器字段指定的寄存器，值被放置在数据移位寄存器的低 24 位中。

备注

数据移位寄存器的当前内容是上次扫描所加载的内容。

在扫描 n 期间读取 DR 扫描 n 1 中指定的寄存器。当然，如果发生中间的 IR 扫描，则数据移位寄存器的内容是不可预测的，因此不会读取 DR 扫描 n 1 中指示的寄存器。

有时，当达到更新 DR 状态时，目标寄存器上的操作仍待处理。在操作挂起时，目标寄存器的一些位可能无法更改，例如复位控制信号已被写入但尚未生效。因此，此写入所指示的新值可能不会应用于寄存器。如果发生这种情况，则会抑制对 ICEPick 寄存器的写入，并且写失败标志会设置为 1。写入失败位在第 31 位被捕获到数据移位寄存器。捕获到该值后，会清除 WF 标志。

如果第 31 位表示必须执行读取，则此时不触碰指定的 ICEPick 寄存器。ICEPick 寄存器内容保持不受干扰。

如果数据移位寄存器的内容在下一个捕获 DR 状态之前保持不变，则此时会读取指定的寄存器。中间的 IR 扫描会干扰数据移位寄存器内容，因此无法保证将读取指定的寄存器。

除数据移位寄存器外，ICEPick 中没有针对读取块和寄存器的地址缓冲。当遵循正确的扫描序列时，不需要额外存储。有关序列的信息，请参阅节 6.5。

6.3.4 TAP 路由寄存器

本部分介绍可使用路由器扫描访问的 TAP 路由寄存器。

6.3.4.1 ICEPick 控制块

ICEPick 控制块实现表 6-13 中的功能。读取未使用的寄存器将返回全 0。

表 6-13. 控制块寄存器

寄存器	寄存器名称
0x0	All0s
0x1	控制
0x2	链接模式
0x3 至 0xF	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

6.3.4.1.1 All0s 寄存器

该寄存器是一个虚拟寄存器，读取时返回 0。忽略写入。写入或者读取该寄存器没有任何副作用。

表 6-14. All0s 寄存器

位	字段	宽度	类型	复位	说明
23-0	零	24	R	0	读取零

6.3.4.1.2 ICEPick 控制寄存器

表 6-15. ICEPick 控制寄存器

位	字段	宽度	类型	复位	说明
23-8	保留	16	R/W	0	保留
7	KeepPoweredinTLR	1	R/W	0	为 1 时，即使处于测试逻辑复位 (TLR) 状态，JTAG 电源域仍保持通电。为 0 时，如果 ICEPick 可见且 TMS 为 1，则 JTAG 电源域将在测试逻辑复位 (TLR) 状态下断电。
6	BlockSysReset	1	R/W	0	为 1 时，器件系统复位信号将被阻止。
5-1	保留	5	R/W	0	保留
0	SystemReset	1	R/W	0	仿真器控制的系统复位 该信号为扫描控制器提供将系统热复位置为有效的能力。写入 1 时，此行为等同于外部芯片热复位信号已短暂置为有效。该信号不会复位任何仿真逻辑。这是一个自行清除位。该位通过将所请求的复位置为有效进行清除。 写入 0 不起作用。

6.3.4.1.3 链接模式寄存器

表 6-16. ICEPick 链接模式寄存器

位	字段	宽度	类型	复位	说明
23-4	保留	20	R/W	0x0	保留
3-1	TAPLinkMode	3	R/W	000	请参阅 表 6-17
0	ActivateMode	1	R/W	0	向该位写入 1 时，当前选择的 TAPLinkMode 将激活。当 ICEPick TAP 在 IR 中以任意操作码前进到运行-测试/空闲状态时，ICEPick 会根据这些设置链接 TAP。

表 6-17. ICEPick TAP 链接模式

值	模式	行为
000	始终优先	ICEPick TAP 始终存在，并链接作为最靠近 TDI 的 TAP。
011	永远消失	激活后，ICEPick TAP 在器件 TDI 和 TDO 之间不再可见。只有上电复位才能使该 TAP 再次可见。
001-010、100 – 111	保留	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

6.3.4.2 测试 TAP 链接块

测试 TAP 链接块包含控制和状态寄存器，如表 6-18 所示。这些寄存器用于选择主扫描路径中的次级 TAP。每个 TAP 都有自己的测试 TAP 控制和状态寄存器。

表 6-18. 测试 TAP 链接寄存器

寄存器	寄存器名称
0x0	次级测试 TAP 0 寄存器
0x1	次级测试 TAP 1 寄存器
0x2	次级测试 TAP 2 寄存器
0x3	次级测试 TAP 3 寄存器
0x4	次级测试 TAP 4 寄存器
0x5	次级测试 TAP 5 寄存器
0x6 – 0xF	保留

6.3.4.2.1 次级测试 TAP 寄存器

表 6-19. 次级测试 TAP 寄存器 (STTR)

位	字段	宽度	类型	复位	说明
23-10	保留	14	R/W	0	保留
9	VisibleTAP	1	R	—	请参阅表 6-21。
8	SelectTAP	1	R/W	0	请参阅表 6-21。
7-2	保留	6	R	0	
1	TapAccessible	1	R	—	请参阅表 6-21。
0	TapPresent	1	R	—	请参阅表 6-21。

6.3.4.3 调试 TAP 链接块

调试 TAP 链接块包含在选择主扫描路径中的次级 TAP 时使用的控制和状态寄存器。辅助调试抽头具有自己的调试 TAP 控制和状态寄存器。如需了解更多详情，请参阅表 6-20。

表 6-20. 调试 TAP 链接寄存器

寄存器	寄存器名称
0x0	次级调试 TAP 0 寄存器
0x1 – 0xF	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

6.3.4.3.1 次级调试 TAP 寄存器

表 6-21 列出了次级调试 TAP 寄存器 (SDTR)。表 6-22 列出了复位控制。

表 6-21. 次级调试 TAP 寄存器 (SDTR)

位	字段	宽度	类型	复位	说明
23-21	保留	3	R/W	0	保留
20	InhibitSleep	1	W	0	为 0 时。该位不影响模块的时钟和电源设置。 当此位为 1 时，一旦开启，就不允许关闭 TAP 模块的电源或时钟。 如果设置该位时目标没有功耗或时钟，则 InhibitSleep 在目标器件上电并再次计时之前不会更改功耗/时钟状态。
			R	—	读取的值不会反映写入的值，直到电源和时钟控制器对写入值的变化采取行动为止。
19-18	保留	2	R	—	保留
17	InReset	1	R	—	InReset 状态及 ReleaseFromWIR 控制共享相同的位。 当为 1 时，次级 TAP 控制的一个或多个模块处于复位状态。 为 0 时，模块未处于复位状态。
	ReleaseFromWIR		W	0	InReset 状态及 ReleaseFromWIR 控制共享相同的位。 当向该位写入 1 并且模块由于 WaitInReset 位而保持复位状态时，模块复位被释放。仅当 WaitInReset 为 1 并且这是使得模块保持复位状态的唯一原因时才会发生这种情况。这是一个自行清除位。 写入 0 不起作用。
16-14	ResetControl	3	R/W	0	覆盖模块功能热复位的应用程序控制。请参阅表 6-22。
13-10	保留	4	R/W	0	保留
9	VisibleTAP	1	R	—	如果为 1，则当前选择了 TAP，并且在活动扫描链中可见。 VisibleTap 位表示之前使用 SelectTap 位选择的 TAP 现在是器件主扫描路径的一部分。当达到 Run-Test-Idle 状态时，ICEPick 会设置 VisibleTap 位。
8	SelectTAP	1	R/W	0	SelectTap 位允许扫描控制器软件更改器件级主扫描路径中包含的次级 TAP。当此位设置为 1 时，选择的 TAP 应包含在主扫描路径中，此时 TAP 状态会前进到运行-测试-空闲状态。当此位更改为 0 时，如果 TAP 状态前进到运行-测试-空闲，从主扫描路径取消选择 TAP。无论当前 IR 指令如何，选择或取消选择都会在运行-测试-空闲状态下发生。 如果 TapPresent 为 0，则阻止对 SelectTap 位的写入，该位保持为 0。
7-4	保留	4	R/W	0	保留
3	ForceActive (ForcePowerAndClock)	1	W	—	当 ForceActive 为 0 时，模块的时钟和电源设置遵循正常的应用程序设置，除非其他仿真控制之一影响该状态。设置 ForceActive 位会使电源及时钟保持开启并在必要时开启。从这个意义上讲，ForceActive 位可以命名为 ForcePowerAndClock。 清除 ForceActive 位会向应用程序返回对电源及时钟设置的控制。如果应用控制指示电源和时钟必须关闭，则模块的电源和时钟将关闭。
			R	—	读取的值不会反映写入的值，直到电源和时钟控制器对写入值的变化采取行动为止。
2	保留	1	R	—	保留
1	TapAccessible	1	R	—	为 0 时，由于安全原因，无法访问 TAP。 如果为 1，则可以访问 TAP。
0	TapPresent	1	R	—	当为 0 时，没有为该点分配 TAP。 如果为 1，则器件中存在该 TAP。 如果不存在 TAP，则此寄存器中的其余控件和状态位被视为不工作。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 6-22. 复位控制

值	命令	说明
000	正常运行	复位在应用或器件控制的正常控制下运行。
001	复位等待 （延长复位时间）	当复位被置位时，由该次级 TAP 控制的模块保持复位状态。单独使用该位不会复位处理器。
010	保留	保留
011	保留	保留
1xx	取消	取消复位命令锁定

6.4 ICEMelter

ICEMelter 唤醒包含 ICEPick 和 cJTAG 模块的 JTAG 电源域，并监控 TCK 引脚上的活动。当 ICEMelter 在 TCK 引脚上检测到流量（TCK 上的 8 个上升沿和 8 个下降沿）时，会向 AON WUC 发送为 JTAG 电源域加电的请求。器件内置超时机制，如果从第三个上升沿到第八个上升沿的时间窗口超过 4ms，则会复位 ICEMelter。在向 JTAG 接口发送剩余命令之前，仿真器必须为 JTAG 电源域留出至少 200µs 的加电时间。

TI 建议注意避免 TCK 引脚上出现意外流量。例如，如果 TCK 引脚通过一个经常插拔的连接器引出，或位于日常使用中可能被触碰到的排针上，就可能发生这种情况。TCK 引脚上的意外流量会导致 ICEMelter 为 JTAG 域加电，这会在任何器件模式（包括待机模式）下增加约 400µA 的电流，并置位引导中暂停 (HIB) 标志。HIB 标志会导致暂停器件的后续启动（例如在任何除引脚复位或 POR 之外的系统复位之后，或进入关断模式时）。退出引导中暂停状态，清除 HIB 标志以及禁用 JTAG 域，都只能通过引脚复位、POR 或使用 JTAG 接口本身来完成。

TCK 引脚具有内部上拉电阻，旨在避免因噪声导致的意外流量，但如果存在因意外触碰或引脚短路导致 TCK 引脚出现外部活动的风险，则此内部上拉电阻是不够的。如果无法保证不发生此类活动，建议使用强外部上拉电阻，甚至通过零欧姆电阻器将该引脚短接至电源电压。上拉电阻的大小，或是否将该引脚短接至电源电压，是在抗意外外部活动的稳健性与对可访问调试端口的需求之间进行权衡的结果。如果通过短接至电源电压来禁用 TCK 引脚，则必须使用引导加载程序进行闪存编程。

可以通过 [AON_IOC:TCKCTL] 禁用 TCK 引脚的输入驱动器。禁用 TCK 引脚会损害调试会话，因此，如果希望保留代码调试支持，在禁用 TCK 引脚时需要特别小心。TI 强烈建议将 TCK 引脚的禁用条件设为 [FLASH:FWFLAG][2] 为 1。[FLASH:FWFLAG][2] 指示在上一次启动中是否发生了 HIB（0：发生了 HIB，1：未发生 HIB）。如果 TCK 引脚被禁用，将无法调试正在运行的目标。但是，如果将 TCK 引脚的禁用条件设为 [FLASH:FWFLAG][2] 为 1，则仍然可以调试器件。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

6.5 串行线查看器 (SWV)

CPU 使用处理器内部的 TPIU 宏单元来支持串行线查看器 (SWV) 接口（一种单线接口）。按照以下顺序启用 CPU 上的 SWV 输出：

1. 通过设置 CPU_SCS:DEMCR.TRCENA 来启用跟踪系统（请参阅节 2.9.4）。
2. 通过写入锁访问寄存器 CPU_ITM:LAR 来解锁 ITM 配置（请参阅节 2.9.3）。
3. 通过设置 CPU_ITM:TCR.ITMENA 来启用 ITM（请参阅节 2.9.3）。
4. 在 CPU_ITM:TER 中启用所需的激励端口（0 到 31）（请参阅节 2.9.3）。
5. 如果需要，更改 CPU_TPIU:FFCR 中的格式化器配置（请参阅节 2.9.5）。
6. 如果需要，更改 CPU_TPIU:SPPR 中的引脚协议（请参阅节 2.9.5）。
7. 在 CPU_TPIU:ACPR 中设置波特率（请参阅节 2.9.5）。
8. 通过向 IOC:IOCFGn 写入相应的端口 ID，可将 SWV 映射到 DIOn（请参阅节 13.10.3）。如需了解更多详情，请参阅章节 13。

如果 FIFO 未满，对 CPU_ITM:STIMn 寄存器的写入（假设它们已启用）会触发 SWV 输出上的传输。

6.6 引导中暂停 (HIB)

CC13x2 和 CC26x2 器件平台实现了一种机制，以确保外部仿真器可在器件执行任何应用程序代码之前获取对器件的控制。这种机制称为引导中暂停 (HIB)。当 HIB 检测到调试活动时，引导代码会在执行结束后到跳转到闪存中的应用程序代码之前，停在等待中断指令 (WFI) 处。

检测到 TCK 引脚上的活动（该引脚会为 JTAG 电源域加电）是下次引导时进入 HIB 的条件。如果在系统复位发生前通过进入测试逻辑复位 (TLR) 状态关闭了 JTAG 电源域，则可以清除 HIB 条件。如果在写入 AON_PMCTL:SHUTDOWN.EN = 1 之后 JTAG 电源域才关闭，则 HIB 条件不会清除。

要退出 HIB，外部仿真器必须连接到器件，并首先通过 DAP 使 CPU 暂停，然后恢复。暂停 CPU 是一个调试事件，会将 CPU 从 WFI 指令唤醒。恢复后，程序将从应用程序代码继续执行。

6.7 调试与关断

在关断模式下，调试器无法保持连接，因为调试子系统的电源在该模式下会关闭。这意味着进入关断模式会导致突然断开与仿真器的连接。为了方便调试关断场景，CC13x2 和 CC26x2 器件平台进行了如下考虑：

- 如果器件处于关断模式，TCK 上的活动会导致立即唤醒。
- 如果在进入关断模式时满足 HIB 条件，则器件一旦达到关断状态就会立即唤醒。如果在唤醒后发生的引导过程中满足 HIB 条件，引导代码将循环运行，直到发生 I/O 唤醒事件。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

6.8 通过 WUC TAP 支持的调试功能

表 6-23. 通过 WUC TAP 支持的调试功能

命令	控制位	功能
CHIP_ERASE_REQ	IR 0x01， DR[7:0] 中的位 1	设置该位（如果后跟通过 WUC TAP 发出的 MCU VD 复位请求）将启动芯片擦除。
MCU_VD_RESET_REQ	IR 0x01， DR[7:0] 中的位 5	设置此位会请求复位整个 MCU VD。
SHUTDOWN_W_JTAG	IR 0x01， DR[7:0] 中的位 6	1：进入关断模式会延迟，直到 JTAG 断开连接。 0：允许器件在不等待与 JTAG 断开连接的情况下进入关断模式。进入关断模式会导致与仿真器突然断开连接。
SYS_RESET_REQ	IR 0x01， DR[7:0] 中的位 7	设置此位会请求复位整个芯片。DEBUGEN 位在复位后保持有效，用于在下次启动后确保 HIB。
TMS_PAD_CFG	IR 0x0C， DR[6:0] 中的位 [5:0]	TMS 引脚的强度和压摆控制设置。
MCU_VD_FORCE_ACTIVE	IR 0x0C， DR[6:0] 中的位 6	1：如果 MCU VD 关闭，则强制激活会为 MCU VD 上电。 0：应用程序将控制 MCU VD。
JTAG_DO_NOT_PU	IR 0x04， DR[6:0] 中的位 0	1：防止 JTAG 电源域从 ICEMelter 上电。 0：当满足唤醒条件时，ICEMelter 为 JTAG 电源域上电。
JTAG_DO_NOT_RESET	IR 0x04， DR[6:0] 中的位 4	1：请勿在 JTAG 电源域断电时复位 WUC tap。 0：JTAG 电源域断电时，WUC 将复位。

6.9 分析器寄存器

此寄存器可用于从芯片中提取运行时信息，而不会干扰代码执行。此寄存器位于 TEST TAP 中（分析器寄存器 IR 编号为 0x06）。如需了解更多详情，请参阅表 6-24。

图 6-12. 分析器寄存器

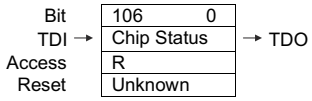


表 6-24. 分析器寄存器字段

位	宽度	说明
81-80	2	AON_PMCTL:PWRSTAT.SW
79	1	AUX_SYSIF:ADCCLKCTL.REQ
78	1	AUX_SYSIF:TDCCLKCTL.REQ
77	1	AUX_ANAIF:DACSMPLCTL.EN
76-74	3	AUX_SYSIF:SWPWRPROF.STAT
73-72	2	AUX_SYSIF:OPMODESTAT.STATE
71	1	AUX_SCE:CTL.RUN 而非 AUX_SCE:CPUSTAT.SLEEP
70	1	SYSBUS 的时钟状态(1)： 0：时钟未运行。 1：时钟正在运行。
69	1	闪存的时钟状态(1)： 0：时钟未运行。 1：时钟正在运行。
68	1	RFCORE 的时钟状态(1)： 0：时钟未运行。 1：时钟正在运行。
67	1	GPIO 的时钟状态(1)： 0：时钟未运行。 1：时钟正在运行。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 6-24. 分析器寄存器字段（续）

位	宽度	说明
66	1	GPTM0 的时钟信号(1)： 0：时钟未运行。 1：时钟正在运行。
65	1	GPTM1 的时钟状态(1)： 0：时钟未运行。 1：时钟正在运行。
64	1	GPTM2 的时钟状态(1)： 0：时钟未运行。 1：时钟正在运行。
63	1	GPTM3 的时钟状态(1)： 0：时钟未运行。 1：时钟正在运行。
62	1	I^2C 的时钟状态(1)： 0：时钟未运行。 1：时钟正在运行。
61	1	I^2S 的时钟状态(1)： 0：时钟未运行。 1：时钟正在运行。
60	1	UDMA 的时钟状态(1)： 0：时钟未运行。 1：时钟正在运行。
59	1	TRNG 的时钟状态(1)： 0：时钟未运行。 1：时钟正在运行。
58	1	CRYPTO 的时钟状态(1)： 0：时钟未运行。 1：时钟正在运行。
57	1	PKA 的时钟状态(1)： 0：时钟未运行。 1：时钟正在运行。
56	1	SSI0 的时钟状态(1)： 0：时钟未运行。 1：时钟正在运行。
55	1	SSI1 的时钟信号(1)： 0：时钟未运行。 1：时钟正在运行。
54	1	UART0 的时钟状态(1)： 0：时钟未运行。 1：时钟正在运行。
53	1	UART1 的时钟状态(1)： 0：时钟未运行。 1：时钟正在运行。
52	1	来自 CPU 的睡眠请求
51	1	来自 CPU 的深度睡眠请求，由 WIC 限定
50	1	保留
49	1	CPU PD 状态： 0：电源域关闭。 1：电源域开启。
48	1	SERIAL PD 状态： 0：电源域关闭。 1：电源域开启。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 6-24. 分析器寄存器字段（续）

位	宽度	说明
47	1	PERIPH PD 状态： 0：电源域关闭。 1：电源域开启。
46	1	RFCORE PD 状态： 0：电源域关闭。 1：电源域开启。
45	1	VIMS PD 状态： 0：电源域关闭。 1：电源域开启。
44	1	MCU 电源状态： 0：STANDBY 1： ACTIVE
43	1	DDI_0_OSC:STAT0.XOSC_HF_EN
42	1	DDI_0_OSC:STAT0.SCLK_HF_SRC
41-40	2	DDI_0_OSC:STAT0.SCLK_LF_SRC
39-36	4	RF_CORE_STATE： b0000 NA 尚无可用信息或 RF 核心已关闭。 b0001 IDLE RF 内核已通电但处于空闲状态；无 RF。 b0010 RFACTIVE RF 合成器处于活动状态。 b0110 RXWAIT RF 内核正在等待同步。 b1110 RXPACKET RF 内核正在接收数据包。 b1010 TX RF 内核正在传输数据包 其他保留
35-28	8	PRCM:PWRPROFSTAT
27	1	压缩程序计数器的值错误： 1：位 26 – 6 中返回的值不可信。 0：位 26 – 6 中返回的值可信。
26-6	21	压缩程序计数器（删除预计为 0 的 32 位程序计数器中的 11 个） 程序计数器 {PC[29:28],PC[24], PC[18:1]} 的有效位
5-0	6	指示 CPU 中当前执行上下文的中断号

(1) 如果 MCU 电源状态处于 STANDBY 状态，则该字段可能无效。

6.10 边界扫描

CC13x2 及 CC26x2 器件使用 IEEE 1149.1 JTAG 标准下定义的标准边界扫描。每个 DIO 都有一个专用的边界扫描单元，其中包含六个寄存器（其中各有两个用于输出引脚，双向控制引脚和输入引脚），如图 6-13 所示。

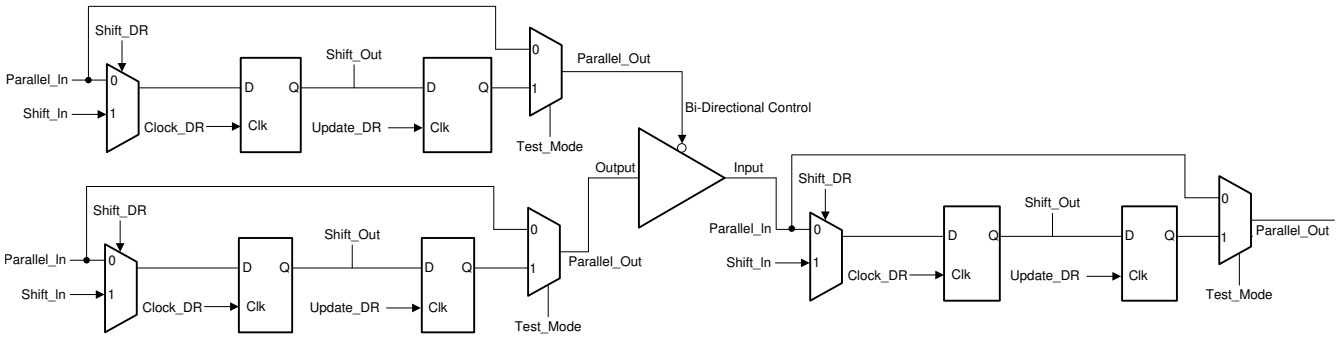


图 6-13. 边界扫描单元

在六个寄存器中，三个是移位寄存器，其他三个是更新寄存器。移位寄存器位于扫描链中，连接在所有 DIO 之间。每个边界扫描寄存器单元内都有几个多路复用器，用于在测试数据和功能数据之间进行选择。边界扫描实现

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

旨在涵盖直流参数测试及 IODFT 测试。CC13x2 和 CC26x2 器件中提供的 32 个 DIO 根据测试引脚多路复用以及测试仪接触式和非接触式 I/O 分为不同的组。有关跨不同软件包的边界扫描 I/O 的详细信息，请参阅表 6-25。

表 6-25. 不同器件的边界扫描 I/O

引脚名称	Tap 接口	组
TCK	TAP_SCAN_CLOCK	
TMS	TAP_SCAN_MODE	
DIO0		grp1
DIO1		grp1
DIO2		grp1
DIO3		grp1
DIO4		grp1
DIO5		grp1
DIO6		grp1
DIO7		grp1
DIO8		grp1
DIO9		grp1
DIO10		grp1
DIO11		grp1
DIO12		grp1
DIO13		grp1
DIO14		grp1
DIO15		grp1
DIO16	TAP_SCAN_OUT	grp3
DIO17	TAP_SCAN_IN	grp3
DIO18		grp1
DIO19		grp1
DIO20		grp1
DIO21		grp1
DIO22		grp1
DIO23		grp2
DIO24		grp2
DIO25		grp1
DIO26		grp1
DIO27		grp1
DIO28		grp1
DIO29		grp1
DIO30		grp1

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

使用对每个 DIO 的边界扫描访问来测试直流参数。

1. 对于输入型直流参数测试：
   - 通过将双向控制引脚驱动为 1，将所有 I/O（DIO2 除外）配置为输入模式。
   - 通过向状态输出引脚 (DIO2) 以外的所有 BIDI 应用低电压来测试 V_{IL}。
   - 在输入 BSR 寄存器上捕获响应，并使用 INTEST 指令移出 DIO2 上捕获的数据。
   - 任何不符合 V_{IL} 规范的引脚都将导致状态输出变成 1。
   - 通过向除状态输出引脚 (DIO2) 之外的所有其他双向引脚施加高电压，对 V_{IH} 重复相同的操作。

     要测试 I_{IH} 和 I_{IL}，需要执行前面讨论的步骤。但是，必须测量电流，而不是电压。要测试状态输出的 V_{IL} 和 V_{IH}，请选择另一个 DIO 作为状态输出，屏蔽所有其他引脚，并重复测试。该功能在边界扫描实现中未包含，但可以使用 DIO 的存储器映射控制来实现。
2. 对于输出直流参数测试：
   - 通过将双向控制引脚驱动为 0，将所有 I/O（DIO1 除外）配置为输出模式。
   - 通过 DIO1 向所有 I/O 移入 0。
   - 使用 EXTEST 指令测量所有引脚上的 V_{OL}。
   - 通过 DIO1 将 1 移位到所有 I/O 来重复相同操作。
   - 测量所有引脚上的 V_{OH}。

   要测试 I_{OH} 和 I_{OL}，需要执行前面讨论的相同步骤。但是，必须测量电流，而不是电压。要测试 V_{OL} 和 V_{OH}，请测试状态输出，选择另一个 DIO 并重复测试。该功能在边界扫描实现中未包含，但可以使用 DIO 的存储器映射控制来实现。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

章节 7

电源、复位和时钟管理 (PRCM)

本章详细介绍了 CC13x2 和 CC26x2 器件平台的灵活电源管理和时钟控制 (PRCM)。

7.1 简介.....507
7.2 系统 CPU 模式.....508
7.3 电源系统.....508
7.4 数字电源分区.....509
7.5 时钟管理.....511
7.6 电源模式.....517
7.7 复位.....520
7.8 PRCM 寄存器.....521

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.1 简介

CC13x2 和 CC26x2 器件平台中的电源和时钟管理 (PRCM) 非常灵活，有助于实现低功耗应用。除介绍复位特性之外，以下各部分还介绍了时钟和电源控制的详细信息。

本章中的特性在 TI-RTOS 中嵌入并且优化。TI-RTOS 用户可认为本章仅供参考。

图 7-1 展示了 CC13x2 和 CC26x2 器件平台中节能特性的层次结构。节能模式的低功耗及循环时间成反比。具有最低功耗的节能模式从启动到节能模式需要最长时间，恢复工作模式的唤醒时间也是如此。表 7-1 总结了节能特性。

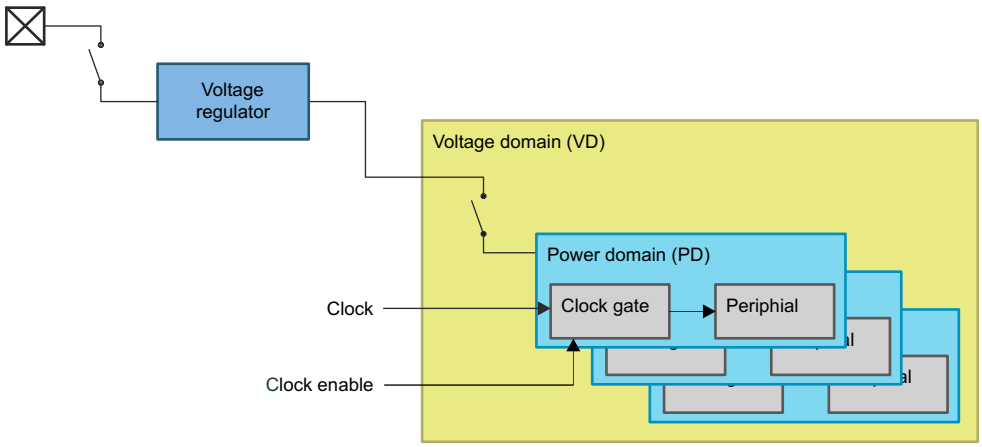


图 7-1. 节能特性的层次结构

表 7-1. 省电特性

省电特性	说明
时钟门控	立即响应 — 无延迟。此特性可节省的电量最少
电源域关闭 （覆盖时钟门控）	与时钟门控相比，下电和上电所需的时间更长。在恢复功能之前，必须重新初始化电源域中没有保留功能的模块。
稳压器关闭	与电源域循环相比，下电和上电所需的时间更长。CC13x2 和 CC26x2 器件平台会丢失所有配置，并会在唤醒时启动。该特性可提供尽可能低的耗电。

表 7-2 列出了为表 7-1 中所列的 TI-RTOS 中的节能特性定义的四种电源模式。节 7.6 详细讨论了电源模式。

表 7-2. TI-RTOS 中的电源模式

电源模式	说明
工作模式	系统 CPU 正在运行。
空闲模式	CPU 电源域已断电。
待机模式	所有电源域都断电，并且 AUX 域处于低功耗或断电模式。电压域由微 LDO 提供。
关断模式	仅 I/O 维持其运行。全部稳压器、电压域和电源域均关闭。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.2 系统 CPU 模式

节 7.3 指的是系统 CPU 模式，因此了解这意味着什么很重要。

系统 CPU 有三种不同的操作模式：运行、睡眠和深度睡眠（请参阅表 7-3）。除了可以根据当前系统 CPU 模式选通的外设时钟之外，每个模式还用于选通系统 CPU 中的内部时钟。在某些情况下，深度睡眠模式是关断电压和电源域断电的多项要求之一。

表 7-3. 系统 CPU 模式

系统 CPU 模式	说明
运行模式	WFI 和 WFE 都处于非活动状态，CPU_SCS:SCR.SLEEPDEEP 不用考虑
睡眠模式	WFI 或 WFE 处于活动状态且 CPU_SCS:SCR.SLEEPDEEP = 0
深度睡眠模式	WFI 或 WFE 处于活动状态且 CPU_SCS:SCR.SLEEPDEEP = 1

7.3 电源系统

CC13x2 和 CC26x2 器件平台的电源系统很复杂，且由硬件控制。图 7-2 展示了一个简化的方案，重点突出可由软件控制的器件。图 7-2 中突出显示了影响不同电源域的寄存器。

有关电压域和电源域的详细信息，请参阅图 7-3。

7.3.1 内部 DC/DC 转换器和全局 LDO

通常，CC13x2 和 CC26x2 器件平台的 VDDS 电源引脚由 1.8V 至 3.8V 的电源（例如电池）供电，而 VDDR 电源引脚由内部 DC/DC 稳压器供电。

或者，也可以使用内部全局 LDO 代替 DC/DC 稳压器，但这会增加器件的电流消耗。在该模式下，断开 DCDC_SW 并将 VDDS_DCDC 连接到 VDDS 电源。全局 LDO 在内部连接到 VDDR 引脚，该引脚必须从外部连接到 VDDR_RF 引脚。必须在 VDDR 网络上使用一个 µF 级的电容器对全局 LDO 进行去耦。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.4 数字电源分区

CC13x2 和 CC26x2 器件平台有两个电压域：MCU_VD 和 AON_VD。两个电压域均包含多个电源域：*_PD。每个电源域包含数字模块。图 7-3 展示了 CC13x2 和 CC26x2 器件平台中的电源分区的详细信息。

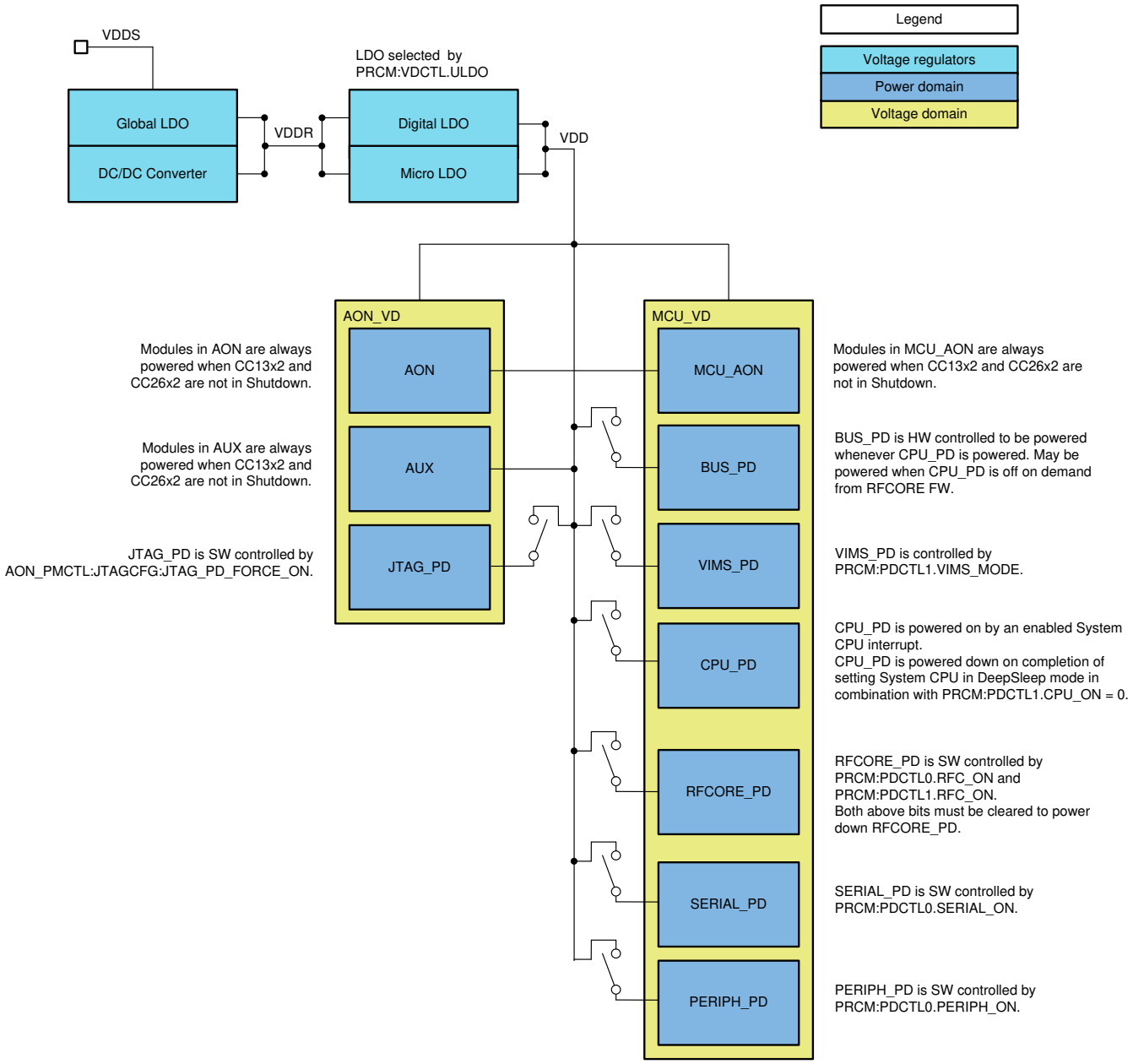
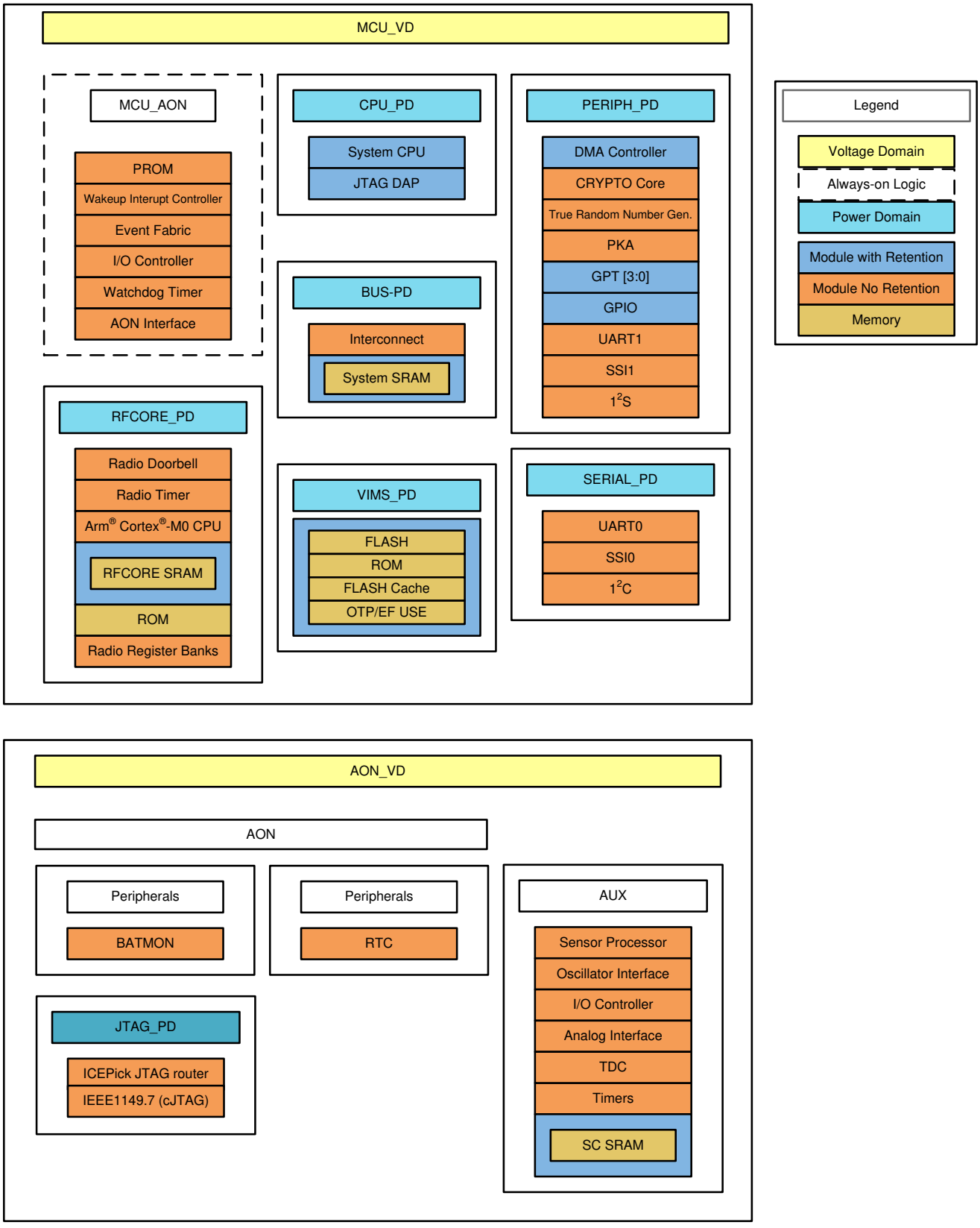


图 7-2. CC13x2 和 CC26x2 电源系统

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated



Copyright © 2017, Texas Instruments Incorporated

图 7-3. CC13x2 和 CC26x2 中的数字电源分区

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.4.1 MCU_VD

图 7-3 显示 MCU 电压域包含划分为多个电源域的 CPU 系统。MCU_VD 还包括未封装在电源域中的常开逻辑，只要 MCU_VD 电压稳压器开启，该逻辑就会上电。当 CC13x2 和 CC26x2 器件平台未处于关断或复位模式时，该电压稳压器将开启且 MCU_AON 将通电。图 7-3 将此逻辑显示为 MCU_AON。

任何已启用的唤醒源都可以对 MCU_VD 加电。

关闭 MCU_VD 的要求可在 PRCM:VDCTL.MCU_VD 的寄存器描述中找到（请参阅节 7.8.2）。

7.4.1.1 MCU_VD 电源域

图 7-2 显示了 MCU_VD 电源域的控制，并提供寄存器的说明。

7.4.2 AON_VD

AON_VD 包含一个电源域和在图 7-3 中标记为 AON 的常开逻辑。

当 CC13x2 和 CC26x2 器件平台未处于关断模式时，AON 中的逻辑始终通电。

7.4.2.1 AON_VD 电源域

图 7-2 展示了 AON_VD 电源域的控制并对各个寄存器进行了说明。

7.5 时钟管理

7.5.1 系统时钟

图 7-4 和表 7-4 显示，CC13x2 和 CC26x2 器件平台具有灵活的时钟多路复用器，系统时钟可从多个来源获得。

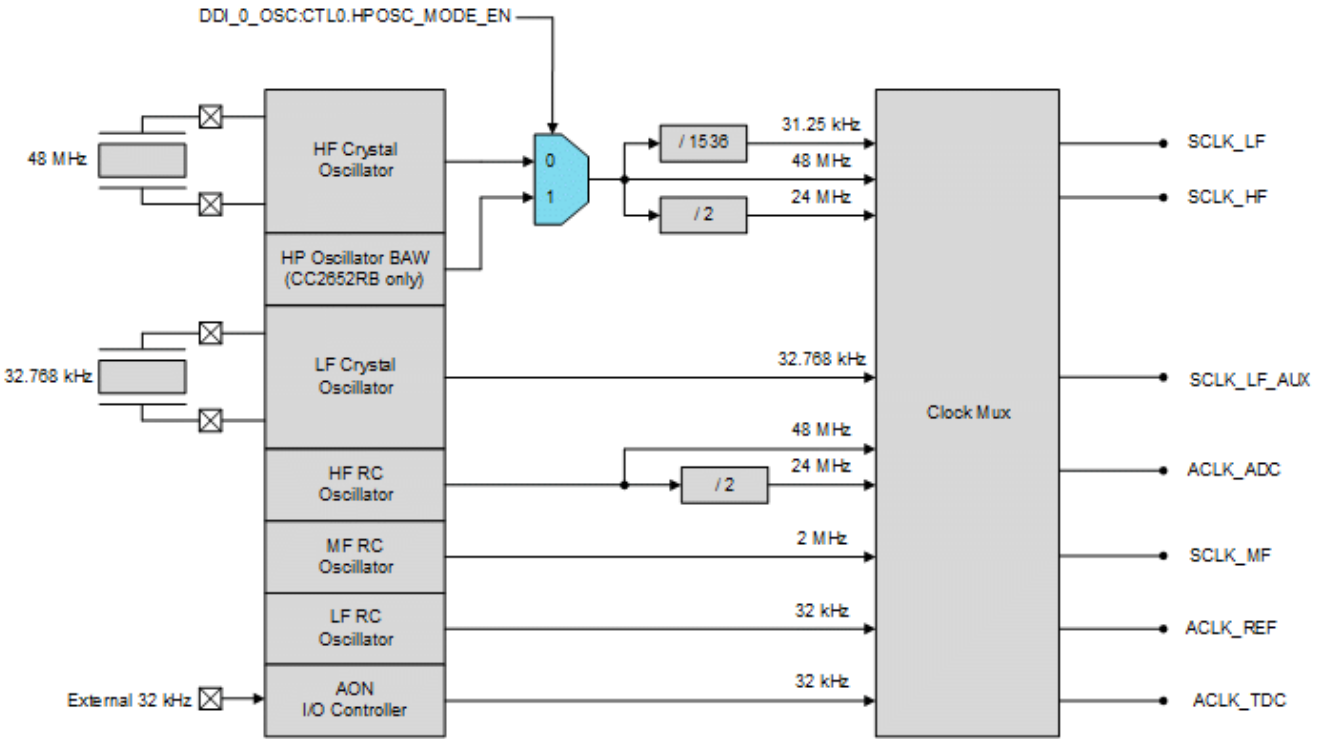


图 7-4. 时钟源

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.5.1.1 控制振荡器

表 7-4 列出了系统时钟说明及可能的源。

表 7-4. 系统时钟

时钟	说明	可能的来源
SCLK_LF	低频时钟 • 始终由 AON 使用 • 在工作模式和待机模式下可用于 AUX	• 31.25kHz 源自 48MHz RC 振荡器 • 31.25kHz 源自 48MHz 晶体振荡器 • 31.25kHz 源自 HP 振荡器 (BAW [1]) • 32 kHz RC 振荡器 • 32.768kHz 晶体振荡器 可在 [DDI_0_OSC:CTL0.SCLK_LF_SRC_SEL] 中选择
SCLK_MF	中频时钟 • PMCTL 始终在工作和空闲模式下使用 • 工作、空闲和待机模式下可用于 AUX	2 MHz RC 振荡器
SCLK_HF	高频时钟 • 由 MCU_VD 在工作和空闲模式下使用 • 在工作模式和空闲模式下可用于 AUX	• 48MHz 源自 48MHz RC 振荡器 • 48MHz 源自 48MHz 晶体振荡器 • 48MHz 源自 HP 振荡器 (BAW [1]) 可在 [DDI_0_OSC:CTL0.SCLK_HF_SRC_SEL] 中选择
SCLK_LF_AUX	用于 AUX_PD (COMP_B) 中的低功耗比较器并用作 REFSYS 中充电比较器的时钟	与 SCLK_LF 相同
ACLK_ADC	用作 ADC 的时钟源	与 SCLK_HF 相同
ACLK_REF	用作时间数字转换器 (TDC) 的起始及停止源	• 31.25kHz 源自 48MHz RC 振荡器 • 31.25kHz 源自 48MHz 晶体振荡器 • 31.25kHz 源自 HP 振荡器 (BAW [1]) • 32 kHz RC 振荡器 • 32.768kHz 晶体振荡器 • 2 MHz RC 振荡器 可在 [DDI_0_OSC:CTL0.ACLK_REF_SRC_SEL] 中选择
ACLK_TDC	用作 TDC 的时钟	• 来自 RC 振荡器的 48MHz • 来自 RC 振荡器的 24MHz • 来自晶体振荡器的 24MHz 可在 [DDI_0_OSC:CTL0.ACLK_TDC_SRC_SEL] 中选择

1. 仅适用于 CC2652RB 器件

备注

启用 48MHz 晶体振荡器时（通过选择 XOSCHF 作为 SCLK_HF 的源），在 XOSCHF 报告为稳定并切换至该源之前，不得关闭 XOSCHF，也不得将 SCLK_HF 源更改为另一个源。当在启动晶体后使 DDI_0_OSC:STAT0.PENDINGSCLKHFSWITCHING 置为有效时，XOSCHF 是稳定的。DriverLib API 应当用于切换 SCLK_HF 源，并且执行此操作时必须禁用中断。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

小心

如果选择来自 48MHz 晶体振荡器的 31.25kHz 作为 SCLK_LF 源，则不得允许器件进入待机模式。如果器件在 48MHz 晶体振荡器运行的情况下进入待机状态，振荡器可能会停止，使器件处于无响应状态。如果还使用直流/直流稳压器，则停止 48MHz 稳压器可能会对器件造成永久损坏。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

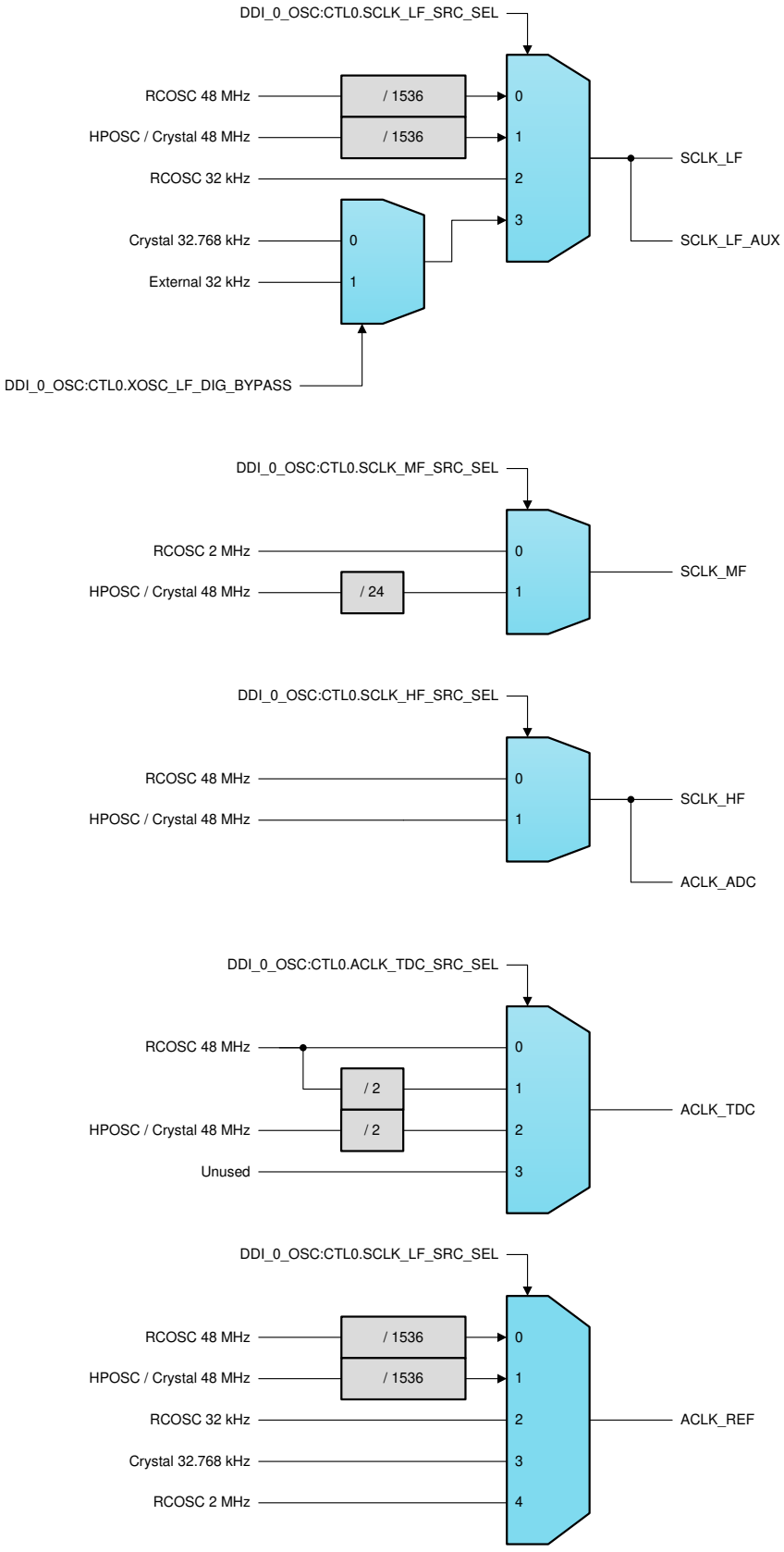


图 7-5. 系统时钟多路复用

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.5.2 MCU_VD 中的时钟

AON_PMCTL 支持 MCU_VD，该时钟在分配到 MCU_VD 中的所有模块之前由 PRCM 进行分频和门控。图 7-6 展示了 PRCM 中用于定义所有模块时钟的分频与门控制的寄存器。当不会发生总线事务时，硬件会自动门控 SYSBUS 时钟。
以下条件必须符合，才能确定 SYSBUS：

- 系统 CPU 处于深度睡眠模式
- PRCM:SECDMACLKGDS.DMA_CLK_EN = 0
- PRCM:SECDMACLKGR.DMA_AM_CLK_EN = 0
- PRCM:SECDMACLKGDS.SEC_CLK_EN = 0
- PRCM:SECDMACLKGR.SEC_AM_CLK_EN = 0
- PRCM:I2SCLKGDS.CLK_EN = 0
- PRCM:I2SCLKGR.AM_CLK_EN = 0
- RFCORE FW 不需要总线访问

当 DMA、SEC、I^{2}S 或 RFCORE 需要活动互连时，即使系统 CPU 处于深度睡眠模式，SYSBUS 时钟也可能运行。

MCU_AON 有两个时钟，一个是始终运行的 INFRASTRUCTURE 时钟，一个是 PERBUSULL 时钟（只要 SYSBUS 时钟运行，该时钟就与 INFRASTRUCTURE 时钟相同）。当 SYSBUS 时钟门控时，会自动门控 PERBUSULL 时钟。无论 PRCM:INFCLKDIVR.RATIO、PRCM:INFCLKDIVS.RATIO 或 PRCM:INFCLKDIVDS.RATIO 中的设置如何，INFRASTRUCTURE 和 PERBUSULL 时钟都会自动门控为最大以 SCLK_HF 时钟频率的一半运行。

7.5.2.1 时钟门控

如图 7-6 中所示，外围模块具有取决于系统 CPU 模式的条件时钟门。当系统 CPU 模式改变时，可以启用或禁用模块时钟。

示例：

- PRCM:I2CCLKGR.CLK_EN = 1
- PRCM:I2CCLKGS.CLK_EN = 0
- PRCM:I2CCLKGDS.CLK_EN = 1

这些设置导致 I^{2}C 时钟在以下情况运行：系统 CPU 处于运行模式和深度睡眠模式时，I^{2}C 时钟被禁用时，以及系统 CPU 处于睡眠模式时。

备注

当设置为深度睡眠模式时，系统 CPU 在转换期间保持处于睡眠模式几个时钟周期。当系统 CPU 更改模式时，需要连续模块时钟的应用会在转换期间启用模块的所有时钟门寄存器。

通过对所有模式使用模块时钟控制 AM_CLK_EN，可以独立于系统 CPU 模式来控制时钟操作。

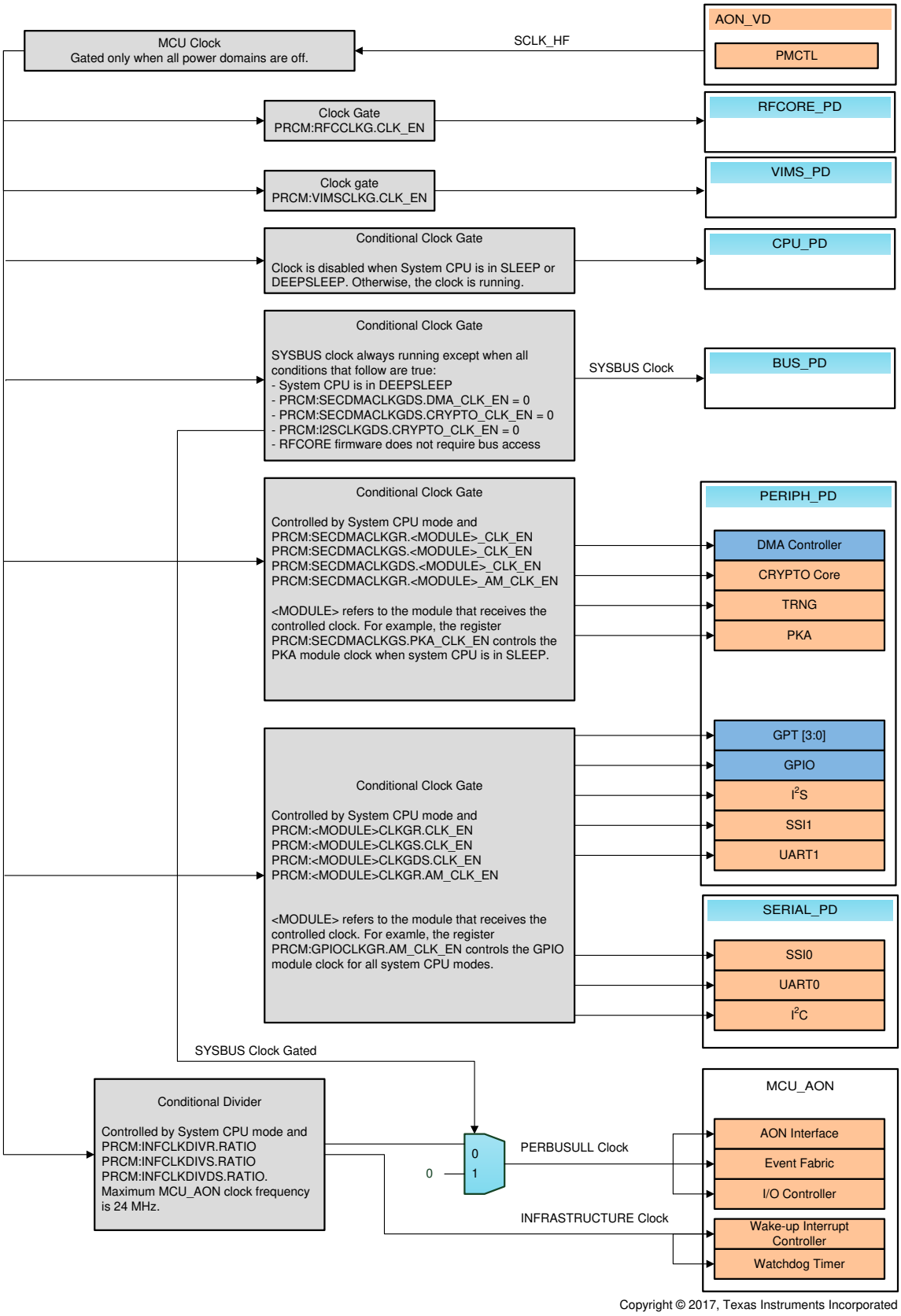
示例：

PRCM:I2CCLKGR.AM_CLK_EN = 1

此设置会使 I^{2}C 时钟独立于系统 CPU 模式运行。

MCU_VD 中的时钟在下电上电期间由硬件控制，因此在电源域上电或断电时无需控制模块时钟。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated



Copyright © 2017, Texas Instruments Incorporated

图 7-6. MCU_VD 中的时钟

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.5.2.2 定标器到 GPT

GPT 的定标器可用于使 GPT 以比 SYSBUS 时钟更慢的频率进行计数。PRCM:GPTCLKDIV 寄存器中的设置对系统中的所有 GPT 有效。

7.5.2.3 WDT 分频器

有一个固定分频比为 32 的 MCU 时钟分频器。无论 PRCM:INFCLKDIVR、PRCM:INFCLKDIVS 和 PRCM:INFCLKDIVDS 寄存器的设置如何，看门狗都以恒定的速度计数。

7.5.3 AON_VD 中的时钟

AON_VD 中的所有模块均在 SCLK_LF 和 SCLK_MF 上运行，AUX 除外。AUX 的时钟可由用户配置。

7.6 电源模式

CC13x2 和 CC26x2 器件平台具有灵活的电源管理功能，支持使用多种不同的配置来实现低功耗应用。本节介绍了 TI-RTOS 定义的电源模式，这些模式涵盖了一系列省电模式（从快速循环时间下的低功耗省电模式到长循环时间下的高功耗省电模式）。

表 7-5 概述了 TI-RTOS 中定义的电源模式。

表 7-5. TI-RTOS 中定义的电源模式

模式	可通过软件配置的电源模式				复位引脚保持
	有效	空闲	待机	关断	
系统 CPU	有效	关闭	关闭	关闭	关闭
系统 SRAM	开启	开启	保持	关闭	关闭
寄存器保持(1)	全双工	全双工	部分	否	否
VIMS_PD（闪存）	开启	可用	关闭	关闭	关闭
RFCORE_PD（无线电）	可用	可用	关闭	关闭	关闭
SERIAL_PD	可用	可用	关闭	关闭	关闭
PERIPH_PD	可用	可用	关闭	关闭	关闭
传感器控制器	可用	可用	可用	关闭	关闭
电源系统	开启	开启	占空比	关闭	关闭
高速时钟	XOSC_HF、HPOSC 或 RCOSC_HF	XOSC_HF、HPOSC 或 RCOSC_HF	关闭	关闭	关闭
中速时钟	RCOSC_MF	RCOSC_MF	RCOSC_MF 可用	关闭	关闭
低速时钟	XOSC_LF 或 RCOSC_LF	XOSC_LF 或 RCOSC_LF	XOSC_LF 或 RCOSC_LF	关闭	关闭
RTC 唤醒	可用	可用	可用	关闭	关闭
在引脚边沿唤醒	可用	可用	可用	可用	关闭
在复位引脚上唤醒	可用	可用	可用	可用	可用
欠压检测 (BOD)	有效	有效	部分(2)	关闭	不适用
上电复位 (POR)	有效	有效	有效	有效	不适用

(1) 有关具备保持能力的模块，请参阅图 7-3。
(2) 在待机模式下的再充电周期之间禁用欠压检测器。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.6.1 启动状态

在系统复位、上电或从关断模式唤醒后，CC13x2 和 CC26x2 器件平台的状态如下：

- 全局 LDO 处于活动状态。
- 数字 LDO 处于活动状态。
- AON_VD 通电。
  - AON 通电。
  - JTAG_PD 断电（默认开启，但如果未连接调试器，BOOT 会将其关闭）。
- MCU_VD 通电。
  - MCU_AON 通电。
  - CPU_PD 通电。
    - 系统 CPU 处于运行模式。
  - BUS_PD 通电。
    - SYSBUS 时钟正在运行。
  - VIMS_PD 通电。
    - VIMS 时钟正在运行。
  - 所有其他电源域均关闭。
  - 所有数字模块时钟均禁用。

7.6.2 活动模式

*活动模式*定义为 CPU_PD 已通电的任何可能的芯片状态，包括 BUS_PD 和 VIMS_PD（请参阅图 7-2）。

在活动模式下，所有模块均可用，功耗高度依赖于应用。
省电特性包括：

- 启用 DC/DC 转换器
- 仅为必要的电源域供电
- 仅启用必要的模块时钟

备注

CC13x2 和 CC26x2 器件平台中电源域的唤醒时间大约需要 11µs。由于 CC13x2 和 CC26x2 器件平台中的时钟门控非常高效，禁用一个电源域中的所有时钟并保持该域上电，可能比频繁对其上下电更省电。

7.6.3 空闲模式

*空闲模式*定义为 CPU_PD 断电而任何其他模块均可通电的任何可能的芯片状态。在空闲模式下，所有模块均可用，功耗高度依赖于应用。

CC13x2 和 CC26x2 器件平台进入空闲模式需满足以下要求：

- PRCM:PDCTL1.CPU_ON = 0
- CPU_SCS:SCR.SLEEPDEEP = 1
- WFI 或 WFE 处于活动状态

CC13x2 和 CC26x2 器件平台可从任何启用的唤醒源唤醒。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.6.4 待机模式

待机模式 定义为 MCU_VD 中的所有电源域均已断电，并且微 LDO 为 AON_VD 和 MCU_AON 供电（请参阅图 7-2）。待机模式是最低功耗模式，其中 CC13x2 和 CC26x2 器件平台除了保留 I/O 输出引脚之外仍然具有其他功能（请参阅
表 7-6）

MCU_AON 中的所有器件在待机模式下保留。MCU_VD 中具有保留功能的所有模块（如图 7-3 所示）都保留在待机模式下。从待机模式唤醒后，MCU_VD 中的所有其他逻辑都必须重新配置。

当 CC13x2 和 CC26x2 器件平台处于待机模式时，传感器控制器在自主模式下可用。

可能的唤醒源包括来自 I/O、JTAG、RTC 和传感器处理器的事件。

以下是 CC13x2 及 CC26x2 器件平台进入待机模式的先决条件：

- AUX 处于低功耗或者断电模式。
- JTAG_PD 断电。
- SCLK_HF 时钟源自 48 MHz RC 振荡器。
- SCLK_LF 时钟源自以下时钟源之一：
  - 32kHz RC 振荡器
  - 32.768kHz 晶体振荡器
- 请求 micro LDO 提供数字器件（请参阅图 7-2）。

表 7-6. 在待机模式下设置 CC13x2 及 CC26x2 的示例序列

说明	寄存器	所需步骤
启用直流/直流转换器以降低功耗	AON_PMCTL:PWRCTL.DCDC_ACTIVE 和 AON_PMCTL:PWRCTL.DCDC_EN	否 （默认值：Global LDO）
将 HF 时钟设置为正确的源	DDI_0_OSC:CTL0.SCLK_HF_SRC_SEL	是
将 LF 时钟设置为正确的源	DDI_0_OSC:CTL0.SCLK_LF_SRC_SEL	是
配置充电配置	AON_PMCTL:RECHARGECFG	是
为 MCU 配置一个或多个唤醒源	AON_EVENT:MCUWUSEL	是
配置系统 SRAM 保留	AON_PMCTL:MCUCFG.SRAM_RET_EN	否 （默认值：启用保持功能）
关闭 JTAG	AON_PMCTL:JTAGCFG:JTAG_PD_FORCE_ON	是
配置唤醒源以生成事件	IOC:IOCFG / AON_RTC / AUX	是
请求 AUX_PD 断电或低功耗模式	AUX_WUC:AUX_SYSIF.OPMODEREQ.REQ	是
锁存 I/O 状态	AON_IOC:IOCLATCH.EN	是
关闭电源域并验证它们是否已关闭	PRCM.PDCTL0 PRCM.PDCTL1 PRCM.PDSTAT0 PRCM.PDSTAT1	是
请求数字电源为微 LDO	PRCM:VDCTL.ULDO	是
将事务同步到 AON 域	AON_RTC.SYNC.WBUSY	是 （读取寄存器）
设置系统 CPU SLEEPDEEP 位	CPU_SCS:SCR.SLEEPDEEP	是
停止系统 CPU 以启动断电序列	WFI 或 WFE	是

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.6.5 关断模式

关断模式 是指 CC13x2 和 CC26x2 器件平台中没有活动的电源稳压器。不建议将关断用作功耗模式，以便在器件正常运行期间节省电量。

在将 CC13x2 和 CC26x2 器件平台置于关断模式之前，I/O 引脚会被锁存以将其输出值保持在关断模式 - 这是使用复位引脚使器件保持复位状态和使用关断模式之间的唯一区别。

只有启用的引脚中断或复位引脚才能将 CC13x2 及 CC26x2 器件平台从关断模式唤醒。

备注

在器件达到关断状态之前，不会检测到从关断状态唤醒的唤醒事件。启动关断后但在实际关断前发生的唤醒事件不会被捕获，因此不会导致器件唤醒。

表 7-7. 进入关断状态的序列示例

说明	寄存器	所需步骤
配置唤醒引脚	IOC:IOCFGxx.WU_CFG	是
将事务同步到 AON 域	AON_RTC.SYNC	是（读取寄存器）
启用关断和锁存 I/O	AON_PMCTL:SHUTDOWN.EN	是
停止系统 CPU 以启动断电序列	WFI 或 WFE	是

当配置为从关断模式唤醒 (IOC:IOFGxx.WU_CFG) 时，IO 焊盘是**电平敏感的**。当输入电平与设定电平匹配时，焊盘会产生内部信号（WU_CFG - 10：低电平，- 11：高电平）。

如果多个引脚配置为从关断状态唤醒，则必须仔细考虑以允许正确触发唤醒事件。由于来自 IO 焊盘的信号在内部一起进行 OR 运算以形成一个路由到内部边沿触发块的信号，因此当其中一个输入信号保持有效时，在释放有效输入之前，内部边沿检测块上不会有边沿信号。换句话说，如果一个引脚保持有效（高电平或低电平，具体取决于配置），则其他引脚在第一个引脚被释放或变为无效之前不会生成唤醒事件。

7.7 复位

CC13x2 和 CC26x2 器件平台具有多种复位源；有些复位源是由于错误或意外行为而触发，而其他复位源是用户启动的。
复位可能会导致以下项复位：

- 整个芯片
- 电源域
- 一个用于调试的数字模块

7.7.1 系统复位

产生完整加电序列和系统 CPU 引导序列的复位被定义为*系统复位*。AON_PMCTL:RESETCTL.RESET_SRC 寄存器是可读的，且始终会显示导致系统复位的最后一个复位源。

以下复位不能禁用，当触发时，一律会导致系统复位：

- 上电复位
- 引脚复位
- VDDS 失效
- VDDR 失效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.7.1.1 时钟丢失检测

当通过 DDI_0_OSC:CTL0.CLK_LOSS_EN 和 AON_PMCTL:RESETCTL.CLK_LOSS_EN 寄存器启用时钟丢失特性时，如果检测到 SCLK_LF 丢失，将导致系统复位。恢复后，AON_PMCTL:RESETCTL.RESET_SRC 寄存器会将时钟丢失显示为复位源。

备注

应用程序必须同时置位 DDI_0_OSC:CTL0.CLK_LOSS_EN 和 AON_PMCTL:RESETCTL.CLK_LOSS_EN 才能启用时钟丢失检测，该特性不会在引导后启用。

7.7.1.2 软件启动的系统复位

写入 AON_PMCTL:RESETCTL.SYSRESET 寄存器会导致系统复位。恢复后，AON_PMCTL:RESETCTL.RESET_SRC 寄存器会将 SYSRESET 显示为复位源。

7.7.1.3 热复位转换为系统复位

来自 MCU 系统的复位请求默认设置成在触发任何热复位源时导致系统复位。恢复后，AON_PMCTL:RESETCTL.RESET_SRC 寄存器会将 WARMRESET 显示为复位源。

7.7.2 MCU_VD 电源域及模块的复位

电源域中逻辑的复位由硬件来控制。当封装电源域进行下电上电时，不保留的模块将复位。具有保持功能的模块会在 MCU_VD 下电上电或者复位时复位。

7.7.3 AON_VD 复位

AON_VD 通过系统复位来进行复位。相关详细信息，请参阅节 7.7.1。

7.8 PRCM 寄存器

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.1 DDI_0_OSC 寄存器

表 7-8 列出了 DDI_0_OSC 寄存器的存储器映射寄存器。表 7-8 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

表 7-8. DDI_0_OSC 寄存器

偏移	首字母缩写词	寄存器名称	部分
0h	CTL0	控制 0	CTL0 寄存器 (偏移 = 0h) [复位 = 00000000h]
4h	CTL1	控制 1	CTL1 寄存器 (偏移 = 4h) [复位 = 00000000h]
8h	RADCEXTCFG	RADC 外部配置	RADCEXTCFG 寄存器 (偏移 = 8h) [复位 = 00000000h]
Ch	AMPCOMPCTL	振幅补偿控制	AMPCOMPCTL 寄存器 (偏移 = Ch) [复位 = 00000000h]
10h	AMPCOMPTH1	振幅补偿阈值 1	AMPCOMPTH1 寄存器 (偏移 = 10h) [复位 = 00000000h]
14h	AMPCOMPTH2	振幅补偿阈值 2	AMPCOMPTH2 寄存器 (偏移 = 14h) [复位 = 00000000h]
18h	ANABYPASSVAL1	模拟旁路值 1	ANABYPASSVAL1 寄存器 (偏移 = 18h) [复位 = 00000000h]
1Ch	ANABYPASSVAL2	内部	ANABYPASSVAL2 寄存器 (偏移 = 1Ch) [复位 = 00000000h]
20h	ATESTCTL	模拟测试控制	ATESTCTL 寄存器 (偏移 = 20h) [复位 = 00000000h]
24h	ADCDOUBLERNANOAMPCTL	ADC 倍频器纳安控制	ADCDOUBLERNANOAMPCTL 寄存器 (偏移 = 24h) [复位 = 00000000h]
28h	XOSCHFCTL	XOSCHF 控制	XOSCHFCTL 寄存器 (偏移 = 28h) [复位 = 00000000h]
2Ch	LFOSCCTL	低频振荡器控制	LFOSCCTL 寄存器 (偏移 = 2Ch) [复位 = 00000000h]
30h	RCOSCHFCTL	RCOSCHF 控制	RCOSCHFCTL 寄存器 (偏移 = 30h) [复位 = 00000000h]
34h	RCOSCMFCTL	RCOSC_MF 控制	RCOSCMFCTL 寄存器 (偏移 = 34h) [复位 = 00000000h]
3Ch	STAT0	状态 0	STAT0 寄存器 (偏移 = 3Ch) [复位 = 00000000h]
40h	STAT1	状态 1	STAT1 寄存器 (偏移 = 40h) [复位 = 00000000h]
44h	STAT2	状态 2	STAT2 寄存器 (偏移 = 44h) [复位 = 00000000h]

复杂的位访问类型经过编码可适应小型表单元。表 7-9 展示了适用于此部分中访问类型的代码。

表 7-9. DDI_0_OSC 访问类型代码

访问类型	代码	说明
读取类型		
R	R	读取
写入类型		
W	W	写入
复位或默认值		
-n		复位后的值或默认值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.1.1 CTL0 寄存器（偏移 = 0h）[复位 = 00000000h]

图 7-7 展示了 CTL0，表 7-10 对其进行了介绍。

返回到汇总表。

控制 0
控制时钟源选择

图 7-7. CTL0 寄存器

31	30	29	28	27	26	25	24
XTAL_IS_24M	RESERVED	BYPASS_XOSC_LF_CLK_QUAL	BYPASS_RCOSC_LF_CLK_QUAL	DOUBLER_START_DURATION		DOUBLER_RESET_DURATION	CLK_DCDC_SRC_SEL
R/W-0h	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
RESERVED	HPOSC_MODE_EN	RESERVED	RCOSC_LF_TRIMMED	XOSC_HF_POWER_MODE	XOSC_LF_DIG_BYPASS	CLK_LOSS_EN	ACLK_TDC_SRC_SEL
R-0h	R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
ACLK_TDC_SRC_SEL	ACLK_REF_SRC_SEL			SCLK_LF_SRC_SEL		RESERVED	SCLK_HF_SRC_SEL
R/W-0h	R/W-0h			R/W-0h		R-0h	R/W-0h

表 7-10. CTL0 寄存器字段说明

位	字段	类型	复位	说明
31	XTAL_IS_24M	R/W	0h	根据精确的高频率 XTAL 进行设置。
30	RESERVED	R	0h	保留
29	BYPASS_XOSC_LF_CLK_QUAL	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
28	BYPASS_RCOSC_LF_CLK_QUAL	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
27-26	DOUBLER_START_DURATION	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
25	DOUBLER_RESET_DURATION	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
24	CLK_DCDC_SRC_SEL	R/W	0h	选择 DCDC 时钟源。 0：CLK_DCDC 是来自 RCOSC 或 XOSC/HPOSC 的 48MHz 时钟 1：CLK_DCDC 始终为来自 RCOSC 的 48MHz 时钟
23-15	RESERVED	R	0h	保留
14	HPOSC_MODE_EN	R/W	0h	0：HPOSC 模式未启用。无线电运行需要 48MHz 晶体。 1：启用 HPOSC 模式。内部 HPOSC 可用作 HF 系统时钟，且可用于无线电操作。
13	RESERVED	R	0h	保留
12	RCOSC_LF_TRIMMED	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
11	XOSC_HF_POWER_MODE	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 7-10. CTL0 寄存器字段说明（续）

位	字段	类型	复位	说明
10	XOSC_LF_DIG_BYPASS	R/W	0h	旁路 XOSC_LF 并将来自 AON 的数字输入时钟用于 xosc_lf 时钟。 0：使用 32kHz XOSC 作为 xosc_lf 时钟源 1：使用来自 AON 的数字输入作为 xosc_lf 时钟源。 仅当 SCLK_LF_SRC_SEL 选择 xosc_lf 作为 sclk_lf 时钟源时，该位才会生效。此位执行的多路复用不是无干扰的。更改该字段时，必须遵循以下过程，以避免 sclk_lf 上出现干扰。 1) 设置 SCLK_LF_SRC_SEL 以选择 xosc_lf 时钟源以外的任何时钟源。 2) 设置或清除该位以对 xosc_lf 进行（或不进行）旁路。 3) 设置 SCLK_LF_SRC_SEL 以使用 xosc_lf。 建议在上述步骤 1 中选择 rcosc_hf 或 xosc_hf（以当前有效的为准）作为时钟源。这可以更快更改时钟。
9	CLK_LOSS_EN	R/W	0h	启用时钟丢失检测，从而为系统控制器提供指示信号。检查 SCLK_HF、SCLK_MF 和 SCLK_LF 时钟丢失指示信号。 0：禁用 1：启用 更改 sclk_lf 时钟源时，必须禁用时钟丢失检测。可以轮询 STAT0.SCLK_LF_SRC 以确定对新 sclk_lf 时钟源的更改何时完成。
8-7	ACLK_TDC_SRC_SEL	R/W	0h	aclk_tdc 的时钟源选择。 00：RCOSC_HF (48MHz) 01：RCOSC_HF (24MHz) 10：XOSC_HF (24MHz) 11：未使用
6-4	ACLK_REF_SRC_SEL	R/W	0h	aclk_ref 的时钟源选择 000：源自 RCOSC_HF (31.25kHz) 001：源自 XOSC_HF (31.25kHz) 010：RCOSC_LF (32kHz) 011：XOSC_LF (32.768kHz) 100：RCOSC_MF (2MHz) 101-111：未使用
3-2	SCLK_LF_SRC_SEL	R/W	0h	sclk_lf 的时钟源选择 0h = 源自高频率 RCOSC 的低频率时钟 1h = 源自高频率 XOSC 或 HPOSC 时钟的低频率时钟（当 HPOSC_MODE_EN = 1 时使用 HPOSC） 2h =低频率 RCOSC 3h = 低频率 XOSC
1	RESERVED	R	0h	保留
0	SCLK_HF_SRC_SEL	R/W	0h	sclk_hf 的时钟源选择。 0h = 高频率 RCOSC 时钟 1h = 高频率 XOSC 或 HPOSC 时钟（当 HPOSC_MODE_EN = 1 时使用 HPOSC）

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.1.2 CTL1 寄存器（偏移 = 4h）[复位 = 00000000h]

图 7-8 展示了 CTL1，表 7-11 对其进行了介绍。

返回到汇总表。

控制 1
此寄存器包含 OSC_DIG 配置

图 7-8. CTL1 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED	RCOSCHFCTRIMFRACT					RCOSCHFCTRIMFRACT_EN	RESERVED
R-0h	R/W-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						XOSC_HF_FAST_START	
R-0h						R/W-0h	

表 7-11. CTL1 寄存器字段说明

位	字段	类型	复位	说明
31-23	RESERVED	R	0h	保留
22-18	RCOSCHFCTRIMFRACT	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
17	RCOSCHFCTRIMFRACT_EN	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
16-2	RESERVED	R	0h	保留
1-0	XOSC_HF_FAST_START	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.1.3 RADCEXTCFG 寄存器（偏移 = 8h）[复位 = 00000000h]

图 7-9 展示了 RADCEXTCFG，表 7-12 对其进行了介绍。

返回到汇总表。

RADC 外部配置

图 7-9. RADCEXTCFG 寄存器

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

表 7-12. RADCEXTCFG 寄存器字段说明

位	字段	类型	复位	说明
31-22	HPM_IBIAS_WAIT_CNT	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
21-16	LPM_IBIAS_WAIT_CNT	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
15-12	IDAC_STEP	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
11-6	RADC_DAC_TH	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
5	RADC_MODE_IS_SAR	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
4-0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.1.4 AMPCOMPCTL 寄存器（偏移 = Ch）[复位 = 00000000h]

图 7-10 展示了 AMPCOMPCTL，表 7-13 对其进行了介绍。

返回到汇总表。

振幅补偿控制

图 7-10. AMPCOMPCTL 寄存器

31	30	29	28	27	26	25	24
RESERVED	AMPCOMP_REQ_MODE	AMPCOMP_FSM_UPDATE_RATE		AMPCOMP_SW_CTRL	AMPCOMP_SW_EN	RESERVED	
R-0h	R/W-0h	R/W-0h		R/W-0h	R/W-0h	R-0h	
23	22	21	20	19	18	17	16
IBIAS_OFFSET				IBIAS_INIT			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
LPM_IBIAS_WAIT_CNT_FINAL							
R/W-0h							
7	6	5	4	3	2	1	0
CAP_STEP				IBIASCAP_HPTOLP_OL_CNT			
R/W-0h				R/W-0h			

表 7-13. AMPCOMPCTL 寄存器字段说明

位	字段	类型	复位	说明
31	RESERVED	R	0h	保留
30	AMPCOMP_REQ_MODE	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
29-28	AMPCOMP_FSM_UPDATE_RATE	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
27	AMPCOMP_SW_CTRL	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
26	AMPCOMP_SW_EN	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
25-24	RESERVED	R	0h	保留
23-20	IBIAS_OFFSET	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
19-16	IBIAS_INIT	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
15-8	LPM_IBIAS_WAIT_CNT_FINAL	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
7-4	CAP_STEP	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
3-0	IBIASCAP_HPTOLP_OL_CNT	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.1.5 AMPCOMPTH1 寄存器（偏移 = 10h）[复位 = 00000000h]

图 7-11 展示了 AMPCOMPTH1，表 7-14 对其进行了介绍。

返回到汇总表。

振幅补偿阈值 1
此寄存器包含振幅补偿算法的阈值

图 7-11. AMPCOMPTH1 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
HPMRAMP3_LTH						RESERVED	
R/W-0h						R-0h	
15	14	13	12	11	10	9	8
HPMRAMP3_HTH						IBIASCAP_LPTOHP_OL_CNT	
R/W-0h						R/W-0h	
7	6	5	4	3	2	1	0
IBIASCAP_LPTOHP_OL_CNT		HPMRAMP1_TH					
R/W-0h		R/W-0h					

表 7-14. AMPCOMPTH1 寄存器字段说明

位	字段	类型	复位	说明
31-24	RESERVED	R	0h	保留
23-18	HPMRAMP3_LTH	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
17-16	RESERVED	R	0h	保留
15-10	HPMRAMP3_HTH	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
9-6	IBIASCAP_LPTOHP_OL_CNT	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
5-0	HPMRAMP1_TH	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.1.6 AMPCOMPTH2 寄存器（偏移 = 14h）[复位 = 00000000h]

图 7-12 展示了 AMPCOMPTH2，表 7-15 对其进行了介绍。

返回到汇总表。

振幅补偿阈值 2
此寄存器包含振幅补偿算法的阈值。

图 7-12. AMPCOMPTH2 寄存器

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

表 7-15. AMPCOMPTH2 寄存器字段说明

位	字段	类型	复位	说明
31-26	LPMUPDATE_LTH	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
25-24	RESERVED	R	0h	保留
23-18	LPMUPDATE_HTH	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
17-16	RESERVED	R	0h	保留
15-10	ADC_COMP_AMPTH_LPM	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
9-8	RESERVED	R	0h	保留
7-2	ADC_COMP_AMPTH_HPM	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
1-0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.1.7 ANABYPASSVAL1 寄存器（偏移 = 18h）[复位 = 00000000h]

图 7-13 展示了 ANABYPASSVAL1，表 7-16 对其进行了介绍。

返回到汇总表。

模拟旁路值 1

图 7-13. ANABYPASSVAL1 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED				XOSC_HF_ROW_Q12			
R-0h				R/W-0h			
15	14	13	12	11	10	9	8
XOSC_HF_COLUMN_Q12							
R/W-0h							
7	6	5	4	3	2	1	0
XOSC_HF_COLUMN_Q12							
R/W-0h							

表 7-16. ANABYPASSVAL1 寄存器字段说明

位	字段	类型	复位	说明
31-20	RESERVED	R	0h	保留
19-16	XOSC_HF_ROW_Q12	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
15-0	XOSC_HF_COLUMN_Q12	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.1.8 ANABYPASSVAL2 寄存器（偏移 = 1Ch）[复位 = 00000000h]

图 7-14 展示了 ANABYPASSVAL2，表 7-17 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 7-14. ANABYPASSVAL2 寄存器

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		XOSC_HF_IBIASTHERM													
R-0h		R/W-0h													

表 7-17. ANABYPASSVAL2 寄存器字段说明

位	字段	类型	复位	说明
31-14	RESERVED	R	0h	保留
13-0	XOSC_HF_IBIASTHERM	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.1.9 ATESTCTL 寄存器（偏移 = 20h）[复位 = 00000000h]

图 7-15 展示了 ATESTCTL，表 7-18 对其进行了介绍。

返回到汇总表。

模拟测试控制

图 7-15. ATESTCTL 寄存器

31	30	29	28	27	26	25	24
SCLK_LF_AUX_EN	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
TEST_RCOSCMF		ATEST_RCOSCMF		RESERVED			
R/W-0h		R/W-0h		R-0h			
7	6	5	4	3	2	1	0
RESERVED							
R-0h							

表 7-18. ATESTCTL 寄存器字段说明

位	字段	类型	复位	说明
31	SCLK_LF_AUX_EN	R/W	0h	为 AUX_COMPB 启用 32kHz 时钟。
30-16	RESERVED	R	0h	保留
15-14	TEST_RCOSCMF	R/W	0h	RCOSC_MF 的测试模式控制 0x0：测试模式已禁用 0x1：向自偏置反相器注入增强偏置电流 0x2：时钟鉴定已禁用 0x3：向自偏置反相器注入增强偏置电流 + 时钟鉴定已禁用
13-12	ATEST_RCOSCMF	R/W	0h	RCOSC_MF 的 ATEST 控制 0x0：ATEST 禁用 0x1：ATEST 已启用，VDD_LOCAL 已连接，**RCOSC_MF* 内部的通路 ATEST 已启用，用于发送 2MHz 时钟信号。 0x2: ATEST 禁用 0x3：ATEST 已启用，偏置电流已连接，**RCOSC_MF* 内部的通路 ATEST 已启用，用于发送 2MHz 时钟信号。
11-0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.1.10 ADCDOUBLERNANOAMPCTL 寄存器（偏移 = 24h）[复位 = 00000000h]

图 7-16 展示了 ADCDOUBLERNANOAMPCTL，表 7-19 对其进行了介绍。

返回到汇总表。

ADC 倍频器纳安控制

图 7-16. ADCDOUBLERNANOAMPCTL 寄存器

31	30	29	28	27	26	25	24
RESERVED							NANOAMP_BIAS_ENABLE
R-0h							R/W-0h
23	22	21	20	19	18	17	16
SPARE23	RESERVED						
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		ADC_SH_MODE_EN	ADC_SH_VBUF_EN	RESERVED		ADC_IREF_CTRL	
R-0h		R/W-0h	R/W-0h	R-0h		R/W-0h	

表 7-19. ADCDOUBLERNANOAMPCTL 寄存器字段说明

位	字段	类型	复位	说明
31-25	RESERVED	R	0h	保留
24	NANOAMP_BIAS_ENABLE	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
23	SPARE23	R/W	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为
22-6	RESERVED	R	0h	保留
5	ADC_SH_MODE_EN	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
4	ADC_SH_VBUF_EN	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
3-2	RESERVED	R	0h	保留
1-0	ADC_IREF_CTRL	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.1.11 XOSCHFCTL 寄存器（偏移 = 28h）[复位 = 00000000h]

图 7-17 展示了 XOSCHFCTL，表 7-20 对其进行了介绍。

返回到汇总表。

XOSCHF 控制

图 7-17. XOSCHFCTL 寄存器

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

表 7-20. XOSCHFCTL 寄存器字段说明

位	字段	类型	复位	说明
31-14	RESERVED	R	0h	保留
13	TCXO_MODE_XOSC_HF_EN	R/W	0h	如果当 TCXO_MODE = 1 时该寄存器为 1，则 XOSC_HF 会启用，从而打开 XOSC_HF 偏置电流，允许向外部输入端的削波正弦波时钟信号提供直流偏置点。
12	TCXO_MODE	R/W	0h	如果在 BYPASS = 1 时该寄存器为 1，则会在外部输入的 TCXO 时钟上启用时钟鉴定。当 BYPASS 为 0 时，该寄存器无效。
11-10	RESERVED	R	0h	保留
9-8	PEAK_DET_ITRIM	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
7	RESERVED	R	0h	保留
6	BYPASS	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
5	RESERVED	R	0h	保留
4-2	HP_BUF_ITRIM	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
1-0	LP_BUF_ITRIM	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.1.12 LFOSCCTL 寄存器（偏移 = 2Ch）[复位 = 00000000h]

图 7-18 展示了 LFOSCCTL，表 7-21 对其进行了介绍。

返回到汇总表。

低频振荡器控制

图 7-18. LFOSCCTL 寄存器

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

表 7-21. LFOSCCTL 寄存器字段说明

位	字段	类型	复位	说明
31-24	RESERVED	R	0h	保留
23-22	XOSCLF_REGULATOR_TRIM	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
21-18	XOSCLF_CMIRRWR_RATIO	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
17-10	RESERVED	R	0h	保留
9-8	RCOSCLF_RTUNE_TRIM	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
7-0	RCOSCLF_CTUNE_TRIM	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.1.13 RCOSCHFCTL 寄存器（偏移 = 30h）[复位 = 00000000h]

图 7-19 展示了 RCOSCHFCTL，表 7-22 对其进行了介绍。

返回到汇总表。

RCOSCHF 控制

图 7-19. RCOSCHFCTL 寄存器

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
RCOSCHF_CTRIM								RESERVED							
R/W-0h								R-0h							

表 7-22. RCOSCHFCTL 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-8	RCOSCHF_CTRIM	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
7-0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.1.14 RCOSCMFCTL 寄存器（偏移 = 34h）[复位 = 00000000h]

图 7-20 展示了 RCOSCMFCTL，表 7-23 对其进行了介绍。

返回到汇总表。

RCOSC_MF 控制

图 7-20. RCOSCMFCTL 寄存器

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

表 7-23. RCOSCMFCTL 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-9	RCOSC_MF_CAP_ARRAY	R/W	0h	调整 RCOSC_MF 电容器阵列。 0x0：标称频率，0.625pF 0x40：最高频率，0.125pF 0x3F：最低频率，1.125pF
8	RCOSC_MF_REG_SEL	R/W	0h	选择稳压器类型。 0：默认 1：备用
7-6	RCOSC_MF_RES_COARSE	R/W	0h	选择用于频率调节的粗略电阻。 0x0：400kΩs，默认频率 0x1：300kΩs，最小频率 0x2：600kΩs，最大频率 0x3：500kΩs
5-4	RCOSC_MF_RES_FINE	R/W	0h	选择用于频率调节的精细电阻。 0x0：11kΩ，最小电阻，最大频率 0x1：13kΩs 0x2：16kΩs 0x3：20kΩs，最大电阻，最小频率
3-0	RCOSC_MF_BIAS_ADJ	R/W	0h	将偏置电流调节为 RCOSC_MF。 0x8 最小电流 0x0 默认电流 0x7 最大电流

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.1.15 STAT0 寄存器（偏移 = 3Ch）[复位 = 00000000h]

图 7-21 展示了 STAT0，表 7-24 对其进行了介绍。

返回到汇总表。

状态 0
此寄存器包含来自 OSC_DIG 的状态信号

图 7-21. STAT0 寄存器

31	30	29	28	27	26	25	24
RESERVED	SCLK_LF_SRC		SCLK_HF_SRC	RESERVED			
R-0h	R-0h		R-0h	R-0h			
23	22	21	20	19	18	17	16
RESERVED	RCOSC_HF_EN	RCOSC_LF_EN	XOSC_LF_EN	CLK_DCDC_RDY	CLK_DCDC_RDY_ACK	SCLK_HF_LOSS	SCLK_LF_LOSS
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
XOSC_HF_EN	RESERVED	XB_48M_CLK_EN	RESERVED	XOSC_HF_LP_BUF_EN	XOSC_HF_HP_BUF_EN	RESERVED	ADC_THMET
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
ADC_DATA_READY	ADC_DATA						PENDINGSCLKHFSWITCHING
R-0h	R-0h						R-0h

表 7-24. STAT0 寄存器字段说明

位	字段	类型	复位	说明
31	RESERVED	R	0h	保留
30-29	SCLK_LF_SRC	R	0h	指示 sclk_lf 的时钟源 0h = 低频率时钟，源自高频率 RCOSC 1h = 低频率时钟，源自高频率 XOSC 2h = 低频率 RCOSC 3h = 低频率 XOSC
28	SCLK_HF_SRC	R	0h	指示 sclk_hf 的时钟源 0h = 高频率 RCOSC 时钟 1h = 高频率 XOSC
27-23	RESERVED	R	0h	保留
22	RCOSC_HF_EN	R	0h	RCOSC_HF_EN
21	RCOSC_LF_EN	R	0h	RCOSC_LF_EN
20	XOSC_LF_EN	R	0h	XOSC_LF_EN
19	CLK_DCDC_RDY	R	0h	CLK_DCDC_RDY
18	CLK_DCDC_RDY_ACK	R	0h	CLK_DCDC_RDY_ACK
17	SCLK_HF_LOSS	R	0h	指示 sclk_hf 丢失
16	SCLK_LF_LOSS	R	0h	指示 sclk_lf 丢失
15	XOSC_HF_EN	R	0h	指示 XOSC_HF 已启用。
14	RESERVED	R	0h	保留
13	XB_48M_CLK_EN	R	0h	表示来自 DOUBLER 的 48 MHz 时钟已启用。 使用 24 或 48MHz 晶体时将启用（对于 48 MHz 晶体，在倍频器旁路模式下启用）。
12	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 7-24. STAT0 寄存器字段说明（续）

位	字段	类型	复位	说明
11	XOSC_HF_LP_BUF_EN	R	0h	XOSC_HF_LP_BUF_EN
10	XOSC_HF_HP_BUF_EN	R	0h	XOSC_HF_HP_BUF_EN
9	RESERVED	R	0h	保留
8	ADC_THMET	R	0h	ADC_THMET
7	ADC_DATA_READY	R	0h	指示 adc_data 就绪时间。
6-1	ADC_DATA	R	0h	adc_data
0	PENDINGSCLKHFSWITCHING	R	0h	指示 SCLK_HF 时钟源何时准备好进行切换

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.1.16 STAT1 寄存器（偏移 = 40h）[复位 = 00000000h]

图 7-22 展示了 STAT1，表 7-25 对其进行了介绍。

返回到汇总表。

状态 1
此寄存器包含来自 OSC_DIG 的状态信号

图 7-22. STAT1 寄存器

31	30	29	28	27	26	25	24
RAMPSTATE				HPM_UPDATE_AMP			
R-0h				R-0h			
23	22	21	20	19	18	17	16
HPM_UPDATE_AMP		LPM_UPDATE_AMP					
R-0h		R-0h					
15	14	13	12	11	10	9	8
FORCE_RCOSC_HF	SCLK_HF_EN	SCLK_MF_EN	ACLK_ADC_EN	ACLK_TDC_EN	ACLK_REF_EN	CLK_CHP_EN	CLK_DCDC_EN
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
SCLK_HF_GOOD	SCLK_MF_GOOD	SCLK_LF_GOOD	ACLK_ADC_GOOD	ACLK_TDC_GOOD	ACLK_REF_GOOD	CLK_CHP_GOOD	CLK_DCDC_GOOD
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

表 7-25. STAT1 寄存器字段说明

位	字段	类型	复位	说明
31-28	RAMPSTATE	R	0h	AMPCOMP FSM 状态 0h = RESET 1h = INITIALIZATION 2h = HPM_RAMP1 3h = HPM_RAMP2 4h = HPM_RAMP3 5h = HPM_UPDATE 6h = IDAC_INCREMENT 7h = IBIAS_CAP_UPDATE 8h = IBIAS_DECREMENT_WITH_MEASURE 9h = LPM_UPDATE Ah = IBIAS_INCREMENT Bh = IDAC_DECREMENT_WITH_MEASURE Ch = DUMMY_TO_INIT_1 Dh = FAST_START Eh = FAST_START_SETTLE
27-22	HPM_UPDATE_AMP	R	0h	HPM_UPDATE 状态期间的 XOSC_HF 振幅。 在高性能模式下启用 XOSC_HF 振幅补偿时，该值为片上振荡器 ADC 测得的晶体振荡振幅除以 15mV 得出。例如，值为 0x20 表示晶体的振幅约为 480mV。要启用振幅补偿，必须将 AON_WUC OSCCFG 设置为非零值。
21-16	LPM_UPDATE_AMP	R	0h	LPM_UPDATE 状态期间的 XOSC_HF 振幅 在低功耗模式下启用 XOSC_HF 振幅补偿时，该值为片上振荡器 ADC 测得的晶体振荡振幅除以 15mV 得出。例如，值为 0x20 表示晶体的振幅约为 480mV。要启用振幅补偿，必须将 AON_WUC OSCCFG 设置为非零值。
15	FORCE_RCOSC_HF	R	0h	force_rcosc_hf
14	SCLK_HF_EN	R	0h	SCLK_HF_EN
13	SCLK_MF_EN	R	0h	SCLK_MF_EN

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 7-25. STAT1 寄存器字段说明（续）

位	字段	类型	复位	说明
12	ACLK_ADC_EN	R	0h	ACLK_ADC_EN
11	ACLK_TDC_EN	R	0h	ACLK_TDC_EN
10	ACLK_REF_EN	R	0h	ACLK_REF_EN
9	CLK_CHP_EN	R	0h	CLK_CHP_EN
8	CLK_DCDC_EN	R	0h	CLK_DCDC_EN
7	SCLK_HF_GOOD	R	0h	SCLK_HF_GOOD
6	SCLK_MF_GOOD	R	0h	SCLK_MF_GOOD
5	SCLK_LF_GOOD	R	0h	SCLK_LF_GOOD
4	ACLK_ADC_GOOD	R	0h	ACLK_ADC_GOOD
3	ACLK_TDC_GOOD	R	0h	ACLK_TDC_GOOD
2	ACLK_REF_GOOD	R	0h	ACLK_REF_GOOD。
1	CLK_CHP_GOOD	R	0h	CLK_CHP_GOOD
0	CLK_DCDC_GOOD	R	0h	CLK_DCDC_GOOD

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.1.17 STAT2 寄存器（偏移 = 44h）[复位 = 00000000h]

图 7-23 展示了 STAT2，表 7-26 对其进行了介绍。

返回到汇总表。

状态 2
此寄存器包含来自 AMPCOMP FSM 的状态信号

图 7-23. STAT2 寄存器

31	30	29	28	27	26	25	24
ADC_DCBIAS						HPM_RAMP1_THMET	HPM_RAMP2_THMET
R-0h						R-0h	R-0h
23	22	21	20	19	18	17	16
HPM_RAMP3_THMET	RESERVED						
R-0h	R-0h						
15	14	13	12	11	10	9	8
RAMPSTATE				RESERVED			
R-0h				R-0h			
7	6	5	4	3	2	1	0
RESERVED				AMPCOMP_REQ	XOSC_HF_AMPGOOD	XOSC_HF_FREQGOOD	XOSC_HF_RF_FREQGOOD
R-0h				R-0h	R-0h	R-0h	R-0h

表 7-26. STAT2 寄存器字段说明

位	字段	类型	复位	说明
31-26	ADC_DCBIAS	R	0h	SAR 模式下由 RADC 读取的直流偏置值 该值为无符号整数。它仅用于调试目的。
25	HPM_RAMP1_THMET	R	0h	指示 hpm_ramp1 已达到阈值
24	HPM_RAMP2_THMET	R	0h	指示 hpm_ramp2 已达到阈值
23	HPM_RAMP3_THMET	R	0h	指示 hpm_ramp3 已达到阈值
22-16	RESERVED	R	0h	保留
15-12	RAMPSTATE	R	0h	xosc_hf 振幅补偿 FSM 与 STAT1.RAMPSTATE 相同。请参阅相关说明，了解编码信息。
11-4	RESERVED	R	0h	保留
3	AMPCOMP_REQ	R	0h	ampcomp_req
2	XOSC_HF_AMPGOOD	R	0h	xosc_hf 振幅在要求的阈值（由 DDI 设置）范围内。不用于调试/状态监控之外的任何目的
1	XOSC_HF_FREQGOOD	R	0h	xosc_hf 的频率适合用于数字时钟
0	XOSC_HF_RF_FREQGOOD	R	0h	xosc_hf 的频率在 +/- 20ppm 范围内，且 xosc_hf 适合用于无线电操作。供软件用于启动频率合成器。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2 PRCM 寄存器

表 7-27 列出了 PRCM 寄存器的存储器映射寄存器。表 7-27 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

表 7-27. PRCM 寄存器

偏移	首字母缩写词	寄存器名称	部分
0h	INFRCLKDIVR	运行模式的基础设施时钟分频系数	INFRCLKDIVR 寄存器（偏移 = 0h）[复位 = 00000000h]
4h	INFRCLKDIVS	睡眠模式的基础设施时钟分频系数	INFRCLKDIVS 寄存器（偏移 = 4h）[复位 = 00000000h]
8h	INFRCLKDIVDS	深度睡眠模式的基础设施时钟分频系数	INFRCLKDIVDS 寄存器（偏移 = 8h）[复位 = 00000000h]
Ch	VDCTL	MCU 电压域控制	VDCTL 寄存器（偏移 = Ch）[复位 = 00000000h]
28h	CLKLOADCTL	将 PRCM 设置加载到 CLKCTRL 电源域	CLKLOADCTL 寄存器（偏移 = 28h）[复位 = 00000002h]
2Ch	RFCCLKG	RFC 时钟选通	RFCCLKG 寄存器（偏移 = 2Ch）[复位 = 00000001h]
30h	VIMSCLKG	VIMS 时钟选通	VIMSCLKG 寄存器（偏移 = 30h）[复位 = 00000003h]
3Ch	SECDMACLKGR	运行及所有模式下 SEC（PKA、TRNG 和 CRYPTO）与 UDMA 的时钟门控	SECDMACLKGR 寄存器（偏移 = 3Ch）[复位 = 00000000h]
40h	SECDMACLKGS	睡眠模式下 SEC（PKA、TRNG 和 CRYPTO）与 UDMA 的时钟门控	SECDMACLKGS 寄存器（偏移 = 40h）[复位 = 00000000h]
44h	SECDMACLKGDS	深度睡眠模式下 SEC（PKA、TRNG 和 CRYPTO）与 UDMA 的时钟门控	SECDMACLKGDS 寄存器（偏移 = 44h）[复位 = 00000000h]
48h	GPIOCLKGR	运行及所有模式下的 GPIO 时钟门控	GPIOCLKGR 寄存器（偏移 = 48h）[复位 = 00000000h]
4Ch	GPIOCLKGS	睡眠模式下的 GPIO 时钟门控	GPIOCLKGS 寄存器（偏移 = 4Ch）[复位 = 00000000h]
50h	GPIOCLKGDS	深度睡眠模式下的 GPIO 时钟门控	GPIOCLKGDS 寄存器（偏移 = 50h）[复位 = 00000000h]
54h	GPTCLKGR	运行及所有模式下的 GPT 时钟门控	GPTCLKGR 寄存器（偏移 = 54h）[复位 = 00000000h]
58h	GPTCLKGS	睡眠模式下的 GPT 时钟门控	GPTCLKGS 寄存器（偏移 = 58h）[复位 = 00000000h]
5Ch	GPTCLKGDS	深度睡眠模式下的 GPT 时钟门控	GPTCLKGDS 寄存器（偏移 = 5Ch）[复位 = 00000000h]
60h	I2CCLKGR	运行及所有模式下的 I2C 时钟门控	I2CCLKGR 寄存器（偏移 = 60h）[复位 = 00000000h]
64h	I2CCLKGS	睡眠模式下的 I2C 时钟门控	I2CCLKGS 寄存器（偏移 = 64h）[复位 = 00000000h]
68h	I2CCLKGDS	深度睡眠模式下的 I2C 时钟门控	I2CCLKGDS 寄存器（偏移 = 68h）[复位 = 00000000h]
6Ch	UARTCLKGR	运行及所有模式下的 UART 时钟门控	UARTCLKGR 寄存器（偏移 = 6Ch）[复位 = 00000000h]
70h	UARTCLKGS	睡眠模式下的 UART 时钟门控	UARTCLKGS 寄存器（偏移 = 70h）[复位 = 00000000h]
74h	UARTCLKGDS	深度睡眠模式下的 UART 时钟门控	UARTCLKGDS 寄存器（偏移 = 74h）[复位 = 00000000h]
78h	SSICLKGR	运行及所有模式下的 SSI 时钟门控	SSICLKGR 寄存器（偏移 = 78h）[复位 = 00000000h]

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 7-27. PRCM 寄存器（续）

偏移	首字母缩写词	寄存器名称	部分
7Ch	SSICLKGS	睡眠模式下的 SSI 时钟门控	SSICLKGS 寄存器（偏移 = 7Ch）[复位 = 00000000h]
80h	SSICLKGDS	深度睡眠模式下的 SSI 时钟门控	SSICLKGDS 寄存器（偏移 = 80h）[复位 = 00000000h]
84h	I2SCLKGR	运行及所有模式下的 I2S 时钟门控	I2SCLKGR 寄存器（偏移 = 84h）[复位 = 00000000h]
88h	I2SCLKGS	睡眠模式下的 I2S 时钟门控	I2SCLKGS 寄存器（偏移 = 88h）[复位 = 00000000h]
8Ch	I2SCLKGDS	深度睡眠模式下的 I2S 时钟门控	I2SCLKGDS 寄存器（偏移 = 8Ch）[复位 = 00000000h]
B4h	SYSBUSCLKDIV	内部	SYSBUSCLKDIV 寄存器（偏移 = B4h）[复位 = 00000000h]
B8h	CPUCLKDIV	内部	CPUCLKDIV 寄存器（偏移 = B8h）[复位 = 00000000h]
BCh	PERBUSCPUCLKDIV	内部	PERBUSCPUCLKDIV 寄存器（偏移 = BCh）[复位 = 00000000h]
C4h	PERDMACLKDIV	内部	PERDMACLKDIV 寄存器（偏移 = C4h）[复位 = 00000000h]
C8h	I2SBCLKSEL	I2S 时钟控制	I2SBCLKSEL 寄存器（偏移 = C8h）[复位 = 00000000h]
CCh	GPTCLKDIV	GPT 分频器	GPTCLKDIV 寄存器（偏移 = CCh）[复位 = 00000000h]
D0h	I2SCLKCTL	I2S 时钟控制	I2SCLKCTL 寄存器（偏移 = D0h）[复位 = 00000000h]
D4h	I2SMCLKDIV	MCLK 分频比	I2SMCLKDIV 寄存器（偏移 = D4h）[复位 = 00000000h]
D8h	I2SBCLKDIV	BCLK 分压比	I2SBCLKDIV 寄存器（偏移 = D8h）[复位 = 00000000h]
DCh	I2SWCLKDIV	WCLK 分频比	I2SWCLKDIV 寄存器（偏移 = DCh）[复位 = 00000000h]
F0h	RESETSECDMA	SEC（PKA、TRNG 和 CRYPTO）与 UDMA 的复位	RESETSECDMA 寄存器（偏移 = F0h）[复位 = 00000000h]
F4h	RESETGPIO	GPIO IP 的复位	RESETGPIO 寄存器（偏移 = F4h）[复位 = 00000000h]
F8h	RESETGPT	GPT IP 的复位	RESETGPT 寄存器（偏移 = F8h）[复位 = 00000000h]
FCh	RESETI2C	I2C IP 的复位	RESETI2C 寄存器（偏移 = FCh）[复位 = 00000000h]
100h	RESETUART	UART IP 的复位	RESETUART 寄存器（偏移 = 100h）[复位 = 00000000h]
104h	RESETSSI	SSI IP 的复位	RESETSSI 寄存器（偏移 = 104h）[复位 = 00000000h]
108h	RESETI2S	I2S IP 的复位	RESETI2S 寄存器（偏移 = 108h）[复位 = 00000000h]
12Ch	PDCTL0	电源域控制	PDCTL0 寄存器（偏移 = 12Ch）[复位 = 00000000h]
130h	PDCTL0RFC	RFC 电源域控制	PDCTL0RFC 寄存器（偏移 = 130h）[复位 = 00000000h]
134h	PDCTL0SERIAL	SERIAL 电源域控制	PDCTL0SERIAL 寄存器（偏移 = 134h）[复位 = 00000000h]

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 7-27. PRCM 寄存器（续）

偏移	首字母缩写词	寄存器名称	部分
138h	PDCTL0PERIPH	PERIPH 电源域控制	PDCTL0PERIPH 寄存器（偏移 = 138h）[复位 = 00000000h]
140h	PDSTAT0	电源域状态	PDSTAT0 寄存器（偏移 = 140h）[复位 = 00000000h]
144h	PDSTAT0RFC	RFC 电源域状态	PDSTAT0RFC 寄存器（偏移 = 144h）[复位 = 00000000h]
148h	PDSTAT0SERIAL	SERIAL 电源域状态	PDSTAT0SERIAL 寄存器（偏移 = 148h）[复位 = 00000000h]
14Ch	PDSTAT0PERIPH	PERIPH 电源域状态	PDSTAT0PERIPH 寄存器（偏移 = 14Ch）[复位 = 00000000h]
17Ch	PDCTL1	电源域控制	PDCTL1 寄存器（偏移 = 17Ch）[复位 = 0000000Ah]
184h	PDCTL1CPU	CPU 电源域直接控制	PDCTL1CPU 寄存器（偏移 = 184h）[复位 = 00000001h]
188h	PDCTL1RFC	RFC 电源域直接控制	PDCTL1RFC 寄存器（偏移 = 188h）[复位 = 00000000h]
18Ch	PDCTL1VIMS	VIMS 模式直接控制	PDCTL1VIMS 寄存器（偏移 = 18Ch）[复位 = 00000001h]
194h	PDSTAT1	电源管理器状态	PDSTAT1 寄存器（偏移 = 194h）[复位 = 0000001Ah]
198h	PDSTAT1BUS	BUS 电源域直接读取状态	PDSTAT1BUS 寄存器（偏移 = 198h）[复位 = 00000001h]
19Ch	PDSTAT1RFC	RFC 电源域直接读取状态	PDSTAT1RFC 寄存器（偏移 = 19Ch）[复位 = 00000000h]
1A0h	PDSTAT1CPU	CPU 电源域直接读取状态	PDSTAT1CPU 寄存器（偏移 = 1A0h）[复位 = 00000001h]
1A4h	PDSTAT1VIMS	VIMS 模式直接读取状态	PDSTAT1VIMS 寄存器（偏移 = 1A4h）[复位 = 00000001h]
1CCh	RFCBITS	RFC 控制	RFCBITS 寄存器（偏移 = 1CCh）[复位 = 00000000h]
1D0h	RFCMODESEL	所选 RFC 模式	RFCMODESEL 寄存器（偏移 = 1D0h）[复位 = 00000000h]
1D4h	RFCMODEHWOPT	允许的 RFC 模式	RFCMODEHWOPT 寄存器（偏移 = 1D4h）[复位 = 00000000h]
1E0h	PWRPROFSTAT	功耗分析器寄存器	PWRPROFSTAT 寄存器（偏移 = 1E0h）[复位 = 00000001h]
21Ch	MCUSRAMCFG	MCU SRAM 配置	MCUSRAMCFG 寄存器（偏移 = 21Ch）[复位 = 00000020h]
224h	RAMRETEN	存储器保持控制	RAMRETEN 寄存器（偏移 = 224h）[复位 = 0000000Bh]
290h	OSCIMSC	振荡器中断屏蔽控制	OSCIMSC 寄存器（偏移 = 290h）[复位 = 00000036h]
294h	OSCRIS	振荡器原始中断状态	OSCRIS 寄存器（偏移 = 294h）[复位 = 00000000h]
298h	OSCICR	振荡器原始中断清除	OSCICR 寄存器（偏移 = 298h）[复位 = 00000000h]

复杂的位访问类型经过编码可适应小型表单元。表 7-28 展示了适用于此部分中访问类型的代码。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 7-28. PRCM 访问类型代码

访问类型	代码	说明
读取类型		
R	R	读取
写入类型		
W	W	写入
复位或默认值		
-n		复位后的值或默认值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.1 INFRCLKDIVR 寄存器（偏移 = 0h）[复位 = 00000000h]

图 7-24 展示了 INFRCLKDIVR，表 7-29 中对此进行了介绍。

返回到汇总表。

运行模式的基础设施时钟分频系数

图 7-24. INFRCLKDIVR 寄存器

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	

表 7-29. INFRCLKDIVR 寄存器字段说明

位	字段	类型	复位	说明
31-2	RESERVED	R	0h	保留
1-0	RATIO	R/W	0h	系统 CPU 处于运行模式时，驱动 MCU_AON 域中模块的时钟分频比。分频比同时影响基础设施时钟和 perbusull 时钟。 0h = 1 分频 1h = 2 分频 2h = 8 分频 3h = 32 分频

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.2 INFRCLKDIVS 寄存器（偏移 = 4h）[复位 = 00000000h]

图 7-25 展示了 INFRCLKDIVS，表 7-30 中对此进行了介绍。

返回到汇总表。

睡眠模式的基础设施时钟分频系数

图 7-25. INFRCLKDIVS 寄存器

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	

表 7-30. INFRCLKDIVS 寄存器字段说明

位	字段	类型	复位	说明
31-2	RESERVED	R	0h	保留
1-0	RATIO	R/W	0h	系统 CPU 处于睡眠模式时，驱动 MCU_AON 域中模块的时钟分频比。分频比同时影响基础设施时钟和 perbusull 时钟。 0h = 1 分频 1h = 2 分频 2h = 8 分频 3h = 32 分频

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.3 INFRCLKDIVDS 寄存器（偏移 = 8h）[复位 = 00000000h]

图 7-26 展示了 INFRCLKDIVDS，表 7-31 中对此进行了介绍。

返回到汇总表。

深度睡眠模式的基础设施时钟分频系数

图 7-26. INFRCLKDIVDS 寄存器

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	

表 7-31. INFRCLKDIVDS 寄存器字段说明

位	字段	类型	复位	说明
31-2	RESERVED	R	0h	保留
1-0	RATIO	R/W	0h	系统 CPU 处于深度睡眠模式时，驱动 MCU_AON 域中模块的时钟分频比。分频比同时影响基础设施时钟和 perbusull 时钟。 0h = 1 分频 1h = 2 分频 2h = 8 分频 3h = 32 分频

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.4 VDCTL 寄存器（偏移 = Ch）[复位 = 00000000h]

图 7-27 展示了 VDCTL，表 7-32 中对此进行了介绍。

返回到汇总表。

MCU 电压域控制

图 7-27. VDCTL 寄存器

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

表 7-32. VDCTL 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	ULDO	R/W	0h	请求 PMCTL 切换到 uLDO。 0：无请求 1：尽可能发出请求 在满足以下要求之前，该位将不起作用： 1. PDCTL1.CPU_ON = 0 2. PDCTL1.VIMS_MODE = x0 3. SECDMACLKGDS.DMA_CLK_EN = 0 且 SECDMACLKGR.DMA_AM_CLK_EN = 0（注意：必须使用 CLKLOADCTL.LOAD 加载设置） 4.SECDMACLKGDS.CRYPTO_CLK_EN = 0 且 SECDMACLKGR.CRYPTO_AM_CLK_EN = 0（注意：必须使用 CLKLOADCTL.LOAD 加载设置） 5.I2SCLKGDS.CLK_EN = 0 且 I2SCLKGR.AM_CLK_EN = 0（注意：必须使用 CLKLOADCTL.LOAD 加载设置） 6.RFC 未请求访问 BUS 7.系统 CPU 处于深度睡眠状态

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.5 CLKLOADCTL 寄存器（偏移 = 28h）[复位 = 00000002h]

图 7-28 展示了 CLKLOADCTL，表 7-33 中对此进行了介绍。

返回到汇总表。

将 PRCM 设置加载到 CLKCTRL 电源域

图 7-28. CLKLOADCTL 寄存器

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

表 7-33. CLKLOADCTL 寄存器字段说明

位	字段	类型	复位	说明
31-2	RESERVED	R	0h	保留
1	LOAD_DONE	R	1h	LOAD 的状态。 当任何需要 LOAD 的寄存器被写入时清零，当 LOAD 完成时设置为 1。 请注意，向寄存器写入无变化的值也会导致 LOAD_DONE 清零。 0：上次 LOAD 后，一个或多个寄存器已被写入访问 1：上次 LOAD 后，没有寄存器被写入访问

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 7-33. CLKLOADCTL 寄存器字段说明 （续）

位	字段	类型	复位	说明
0	LOAD	W	0h	0：无操作 1：将设置加载到 CLKCTRL。该位被硬件清零。 可以在更改多项设置后写入 LOAD 一次，以便所有更改同时生效。也可以在一项设置更新后执行 LOAD。 需要在其设置被应用之前执行 LOAD 的寄存器包括： - SYSBUSCLKDIV - CPUCLKDIV - PERBUSCPUCLKDIV - PERDMACLKDIV - PERBUSCPUCLKG - RFCCLKG - VIMSCLKG - SECDMACLKGR - SECDMACLKGS - SECDMACLKGDS - GPIOCLKGR - GPIOCLKGS - GPIOCLKGDS - GPTCLKGR - GPTCLKGS - GPTCLKGDS - GPTCLKDIV - I2CCLKGR - I2CCLKGS - I2CCLKGDS - SSICLKGR - SSICLKGS - SSICLKGDS - UARTCLKGR - UARTCLKGS - UARTCLKGDS - I2SCLKGR - I2SCLKGS - I2SCLKGDS - I2SBCLKSEL - I2SCLKCTL - I2SMCLKDIV - I2SBCLKDIV - I2SWCLKDIV

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.6 RFCCLKG 寄存器（偏移 = 2Ch）[复位 = 00000001h]

图 7-29 展示了 RFCCLKG，表 7-34 中对此进行了介绍。

返回到汇总表。

RFC 时钟选通

图 7-29. RFCCLKG 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	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_EN
R-0h							R/W-1h

表 7-34. RFCCLKG 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	CLK_EN	R/W	1h	0：禁用时钟 1：如果 RFC 电源域已开启，则启用时钟 要使更改生效，需要写入 CLKLOADCTL.LOAD

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.7 VIMSCLKG 寄存器（偏移 = 30h）[复位 = 00000003h]

图 7-30 展示了 VIMSCLKG，表 7-35 中对此进行了介绍。

返回到汇总表。

VIMS 时钟选通

图 7-30. VIMSCLKG 寄存器

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														CLK_EN	
R-0h														R/W-3h	

表 7-35. VIMSCLKG 寄存器字段说明

位	字段	类型	复位	说明
31-2	RESERVED	R	0h	保留
1-0	CLK_EN	R/W	3h	00：禁用时钟 01：当 SYSBUS 时钟被禁用时禁用时钟 11：启用时钟 要使更改生效，需要写入 CLKLOADCTL.LOAD

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.8 SECDMACLKGR 寄存器（偏移 = 3Ch）[复位 = 00000000h]

图 7-31 展示了 SECDMACLKGR，表 7-36 中对此进行了介绍。

返回到汇总表。

运行及所有模式下 SEC（PKA、TRNG 和 CRYPTO）与 UDMA 的时钟门控

图 7-31. SECDMACLKGR 寄存器

31	30	29	28	27	26	25	24
RESERVED							DMA_AM_CLK_EN
R-0h							R/W-0h
23	22	21	20	19	18	17	16
RESERVED				PKA_ZERIOZE_RESET_N	PKA_AM_CLK_EN	TRNG_AM_CLK_EN	CRYPTO_AM_CLK_EN
R-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED							DMA_CLK_EN
R-0h							R/W-0h
7	6	5	4	3	2	1	0
RESERVED					PKA_CLK_EN	TRNG_CLK_EN	CRYPTO_CLK_EN
R-0h					R/W-0h	R/W-0h	R/W-0h

表 7-36. SECDMACLKGR 寄存器字段说明

位	字段	类型	复位	说明
31-25	RESERVED	R	0h	保留
24	DMA_AM_CLK_EN	R/W	0h	0：不强制 1：在所有模式（运行、睡眠和深度睡眠）下强制开启时钟 启用时覆盖 DMA_CLK_EN、SECDMACLKGS.DMA_CLK_EN 和 SECDMACLKGDS.DMA_CLK_EN。 启用时，SYSBUS 时钟将始终运行 要使更改生效，需要写入 CLKLOADCTL.LOAD
23-20	RESERVED	R	0h	保留
19	PKA_ZERIOZE_RESET_N	R/W	0h	清零逻辑硬件复位。 0：pka_zeroize 逻辑无效。 1：启用存储器的 pka_zeroize。 该寄存器必须保持活动状态，直到存储器完全清零，这需要 256 个系统总线时钟周期。
18	PKA_AM_CLK_EN	R/W	0h	0：不强制 1：在所有模式（运行、睡眠和深度睡眠）下强制开启时钟 启用时覆盖 PKA_CLK_EN、SECDMACLKGS.PKA_CLK_EN 和 SECDMACLKGDS.PKA_CLK_EN。 要使更改生效，需要写入 CLKLOADCTL.LOAD
17	TRNG_AM_CLK_EN	R/W	0h	0：不强制 1：在所有模式（运行、睡眠和深度睡眠）下强制开启时钟 启用时覆盖 TRNG_CLK_EN、SECDMACLKGS.TRNG_CLK_EN 和 SECDMACLKGDS.TRNG_CLK_EN。 要使更改生效，需要写入 CLKLOADCTL.LOAD

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 7-36. SECDMACLKGR 寄存器字段说明（续）

位	字段	类型	复位	说明
16	CRYPTO_AM_CLK_EN	R/W	0h	0：不强制 1：在所有模式（运行、睡眠和深度睡眠）下强制开启时钟 启用时覆盖 CRYPTO_CLK_EN、SECDMACLKGS.CRYPTO_CLK_EN 和 SECDMACLKGDS.CRYPTO_CLK_EN。 启用时，SYSBUS 时钟将始终运行 要使更改生效，需要写入 CLKLOADCTL.LOAD
15-9	RESERVED	R	0h	保留
8	DMA_CLK_EN	R/W	0h	0：禁用时钟 1：启用时钟 可以通过 DMA_AM_CLK_EN 强制开启 要使更改生效，需要写入 CLKLOADCTL.LOAD
7-3	RESERVED	R	0h	保留
2	PKA_CLK_EN	R/W	0h	0：禁用时钟 1：启用时钟 可以通过 PKA_AM_CLK_EN 强制开启 要使更改生效，需要写入 CLKLOADCTL.LOAD
1	TRNG_CLK_EN	R/W	0h	0：禁用时钟 1：启用时钟 可以通过 TRNG_AM_CLK_EN 强制开启 要使更改生效，需要写入 CLKLOADCTL.LOAD
0	CRYPTO_CLK_EN	R/W	0h	0：禁用时钟 1：启用时钟 可以通过 CRYPTO_AM_CLK_EN 强制开启 要使更改生效，需要写入 CLKLOADCTL.LOAD

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.9 SECDMACLKGS 寄存器（偏移 = 40h）[复位 = 00000000h]

图 7-32 展示了 SECDMACLKGS，表 7-37 中对此进行了介绍。

返回到汇总表。

睡眠模式下 SEC（PKA、TRNG 和 CRYPTO）与 UDMA 的时钟门控

图 7-32. SECDMACLKGS 寄存器

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

表 7-37. SECDMACLKGS 寄存器字段说明

位	字段	类型	复位	说明
31-9	RESERVED	R	0h	保留
8	DMA_CLK_EN	R/W	0h	0：禁用时钟 1：启用时钟 可以通过 SECDMACLKGR.DMA_AM_CLK_EN 强制开启 要使更改生效，需要写入 CLKLOADCTL.LOAD
7-3	RESERVED	R	0h	保留
2	PKA_CLK_EN	R/W	0h	0：禁用时钟 1：启用时钟 可以通过 SECDMACLKGR.PKA_AM_CLK_EN 强制开启 要使更改生效，需要写入 CLKLOADCTL.LOAD
1	TRNG_CLK_EN	R/W	0h	0：禁用时钟 1：启用时钟 可以通过 SECDMACLKGR.TRNG_AM_CLK_EN 强制开启 要使更改生效，需要写入 CLKLOADCTL.LOAD
0	CRYPTO_CLK_EN	R/W	0h	0：禁用时钟 1：启用时钟 可以通过 SECDMACLKGR.CRYPTO_AM_CLK_EN 强制开启 要使更改生效，需要写入 CLKLOADCTL.LOAD

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.10 SECDMACLKGDS 寄存器（偏移 = 44h）[复位 = 00000000h]

图 7-33 展示了 SECDMACLKGDS，表 7-38 中对此进行了介绍。

返回到汇总表。

深度睡眠模式下 SEC（PKA、TRNG 和 CRYPTO）与 UDMA 的时钟门控

图 7-33. SECDMACLKGDS 寄存器

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

表 7-38. SECDMACLKGDS 寄存器字段说明

位	字段	类型	复位	说明
31-9	RESERVED	R	0h	保留
8	DMA_CLK_EN	R/W	0h	0：禁用时钟 1：启用时钟 可以通过 SECDMACLKGR.DMA_AM_CLK_EN 强制开启 要使更改生效，需要写入 CLKLOADCTL.LOAD
7-3	RESERVED	R	0h	保留
2	PKA_CLK_EN	R/W	0h	0：禁用时钟 1：启用时钟 可以通过 SECDMACLKGR.PKA_AM_CLK_EN 强制开启 要使更改生效，需要写入 CLKLOADCTL.LOAD
1	TRNG_CLK_EN	R/W	0h	0：禁用时钟 1：启用时钟 启用时 SYSBUS 时钟将始终运行 可以通过 SECDMACLKGR.TRNG_AM_CLK_EN 强制开启 要使更改生效，需要写入 CLKLOADCTL.LOAD
0	CRYPTO_CLK_EN	R/W	0h	0：禁用时钟 1：启用时钟 启用时 SYSBUS 时钟将始终运行 可以通过 SECDMACLKGR.CRYPTO_AM_CLK_EN 强制开启 要使更改生效，需要写入 CLKLOADCTL.LOAD

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.11 GPIOCLKGR 寄存器 (偏移 = 48h) [复位 = 00000000h]

图 7-34 展示了 GPIOCLKGR，表 7-39 中对此进行了介绍。

返回到汇总表。

运行及所有模式下的 GPIO 时钟门控

图 7-34. GPIOCLKGR 寄存器

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

表 7-39. GPIOCLKGR 寄存器字段说明

位	字段	类型	复位	说明
31-9	RESERVED	R	0h	保留
8	AM_CLK_EN	R/W	0h	0：不强制 1：在所有模式 (运行、睡眠和深度睡眠) 下强制开启时钟启用时覆盖 CLK_EN、GPIOCLKGS.CLK_EN 和 GPIOCLKGDS.CLK_EN。 要使更改生效，需要写入 CLKLOADCTL.LOAD
7-1	RESERVED	R	0h	保留
0	CLK_EN	R/W	0h	0：禁用时钟 1：启用时钟 可以通过 AM_CLK_EN 强制开启 要使更改生效，需要写入 CLKLOADCTL.LOAD

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.12 GPIOCLKGS 寄存器（偏移 = 4Ch）[复位 = 00000000h]

图 7-35 展示了 GPIOCLKGS，表 7-40 中对此进行了介绍。

返回到汇总表。

睡眠模式下的 GPIO 时钟门控

图 7-35. GPIOCLKGS 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	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_EN
R-0h							R/W-0h

表 7-40. GPIOCLKGS 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	CLK_EN	R/W	0h	0：禁用时钟 1：启用时钟 可以通过 GPIOCLKGR.AM_CLK_EN 强制开启 要使更改生效，需要写入 CLKLOADCTL.LOAD

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.13 GPIOCLKGDS 寄存器（偏移 = 50h）[复位 = 00000000h]

图 7-36 展示了 GPIOCLKGDS，表 7-41 中对此进行了介绍。

返回到汇总表。

深度睡眠模式下的 GPIO 时钟门控

图 7-36. GPIOCLKGDS 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	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_EN
R-0h							R/W-0h

表 7-41. GPIOCLKGDS 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	CLK_EN	R/W	0h	0：禁用时钟 1：启用时钟 可以通过 GPIOCLKGR.AM_CLK_EN 强制开启 要使更改生效，需要写入 CLKLOADCTL.LOAD

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.14 GPTCLKGR 寄存器（偏移 = 54h）[复位 = 00000000h]

图 7-37 展示了 GPTCLKGR，表 7-42 中对此进行了介绍。

返回到汇总表。

运行及所有模式下的 GPT 时钟门控

图 7-37. GPTCLKGR 寄存器

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				AM_CLK_EN				RESERVED				CLK_EN			
R-0h				R/W-0h				R-0h				R/W-0h			

表 7-42. GPTCLKGR 寄存器字段说明

位	字段	类型	复位	说明
31-12	RESERVED	R	0h	保留
11-8	AM_CLK_EN	R/W	0h	以下每位具有以下含义： 0：不强制 1：在所有模式（运行、睡眠和深度睡眠）下强制开启时钟 启用时覆盖 CLK_EN、GPTCLKGS.CLK_EN 和 GPTCLKGDS.CLK_EN。 ENUM 可以进行组合 要使更改生效，需要写入 CLKLOADCTL.LOAD 1h = 在所有模式下为 GPT0 启用时钟 2h = 在所有模式下为 GPT1 启用时钟 4h = 在所有模式下为 GPT2 启用时钟 8h = 在所有模式下为 GPT3 启用时钟
7-4	RESERVED	R	0h	保留
3-0	CLK_EN	R/W	0h	以下每位具有以下含义： 0：禁用时钟 1：启用时钟 可以通过 AM_CLK_EN 强制开启 ENUM 可以进行组合 要使更改生效，需要写入 CLKLOADCTL.LOAD 1h = 为 GPT0 启用时钟 2h = 为 GPT1 启用时钟 4h = 为 GPT2 启用时钟 8h = 为 GPT3 启用时钟

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.15 GPTCLKGS 寄存器（偏移 = 58h）[复位 = 00000000h]

图 7-38 展示了 GPTCLKGS，表 7-43 中对此进行了介绍。

返回到汇总表。

睡眠模式下的 GPT 时钟门控

图 7-38. GPTCLKGS 寄存器

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												CLK_EN			
R-0h												R/W-0h			

表 7-43. GPTCLKGS 寄存器字段说明

位	字段	类型	复位	说明
31-4	RESERVED	R	0h	保留
3-0	CLK_EN	R/W	0h	以下每位具有以下含义： 0：禁用时钟 1：启用时钟 可以通过 GPTCLKGR.AM_CLK_EN 强制开启 ENUM 可以进行组合 要使更改生效，需要写入 CLKLOADCTL.LOAD 1h = 为 GPT0 启用时钟 2h = 为 GPT1 启用时钟 4h = 为 GPT2 启用时钟 8h = 为 GPT3 启用时钟

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.16 GPTCLKGDS 寄存器（偏移 = 5Ch）[复位 = 00000000h]

图 7-39 展示了 GPTCLKGDS，表 7-44 中对此进行了介绍。

返回到汇总表。

深度睡眠模式下的 GPT 时钟门控

图 7-39. GPTCLKGDS 寄存器

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												CLK_EN			
R-0h												R/W-0h			

表 7-44. GPTCLKGDS 寄存器字段说明

位	字段	类型	复位	说明
31-4	RESERVED	R	0h	保留
3-0	CLK_EN	R/W	0h	以下每位具有以下含义： 0：禁用时钟 1：启用时钟 可以通过 GPTCLKGR.AM_CLK_EN 强制开启 ENUM 可以进行组合 要使更改生效，需要写入 CLKLOADCTL.LOAD 1h = 为 GPT0 启用时钟 2h = 为 GPT1 启用时钟 4h = 为 GPT2 启用时钟 8h = 为 GPT3 启用时钟

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.17 I2CCLKGR 寄存器（偏移 = 60h）[复位 = 00000000h]

图 7-40 展示了 I2CCLKGR，表 7-45 对其进行了介绍。

返回到汇总表。

运行及所有模式下的 I2C 时钟门控

图 7-40. I2CCLKGR 寄存器

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

表 7-45. I2CCLKGR 寄存器字段说明

位	字段	类型	复位	说明
31-9	RESERVED	R	0h	保留
8	AM_CLK_EN	R/W	0h	0：不强制 1：在所有模式（运行、睡眠和深度睡眠）下强制开启时钟启用时覆盖 CLK_EN、I2CCLKGS.CLK_EN 和 I2CCLKGDS.CLK_EN。 要使更改生效，需要写入 CLKLOADCTL.LOAD
7-1	RESERVED	R	0h	保留
0	CLK_EN	R/W	0h	0：禁用时钟 1：启用时钟 可以通过 AM_CLK_EN 强制开启 要使更改生效，需要写入 CLKLOADCTL.LOAD

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.18 I2CCLKGS 寄存器（偏移 = 64h）[复位 = 00000000h]

图 7-41 展示了 I2CCLKGS，表 7-46 对其进行了介绍。

返回到汇总表。

睡眠模式下的 I2C 时钟门控

图 7-41. I2CCLKGS 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	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_EN
R-0h							R/W-0h

表 7-46. I2CCLKGS 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	CLK_EN	R/W	0h	0：禁用时钟 1：启用时钟 可以通过 I2CCLKGR.AM_CLK_EN 强制开启 要使更改生效，需要写入 CLKLOADCTL.LOAD

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.19 I2CCLKGDS 寄存器（偏移 = 68h）[复位 = 00000000h]

图 7-42 展示了 I2CCLKGDS，表 7-47 对其进行了介绍。

返回到汇总表。

深度睡眠模式下的 I2C 时钟门控

图 7-42. I2CCLKGDS 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	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_EN
R-0h							R/W-0h

表 7-47. I2CCLKGDS 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	CLK_EN	R/W	0h	0：禁用时钟 1：启用时钟 可以通过 I2CCLKGR.AM_CLK_EN 强制开启 要使更改生效，需要写入 CLKLOADCTL.LOAD

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.20 UARTCLKGR 寄存器（偏移 = 6Ch）[复位 = 00000000h]

图 7-43 展示了 UARTCLKGR，表 7-48 中对此进行了介绍。

返回到汇总表。

运行及所有模式下的 UART 时钟门控

图 7-43. UARTCLKGR 寄存器

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

表 7-48. UARTCLKGR 寄存器字段说明

位	字段	类型	复位	说明
31-10	RESERVED	R	0h	保留
9-8	AM_CLK_EN	R/W	0h	0：不强制 1：在所有模式（运行、睡眠和深度睡眠）下强制开启时钟 启用时覆盖 CLK_EN、UARTCLKGS.CLK_EN 和 UARTCLKGDS.CLK_EN。 要使更改生效，需要写入 CLKLOADCTL.LOAD 1h = 为 UART0 启用时钟 2h = 为 UART1 启用时钟
7-2	RESERVED	R	0h	保留
1-0	CLK_EN	R/W	0h	0：禁用时钟 1：启用时钟 可以通过 AM_CLK_EN 强制开启 要使更改生效，需要写入 CLKLOADCTL.LOAD 1h = 为 UART0 启用时钟 2h = 为 UART1 启用时钟

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.21 UARTCLKGS 寄存器（偏移 = 70h）[复位 = 00000000h]

图 7-44 展示了 UARTCLKGS，表 7-49 中对此进行了介绍。

返回到汇总表。

睡眠模式下的 UART 时钟门控

图 7-44. UARTCLKGS 寄存器

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															CLK_EN
R-0h															R/W-0h

表 7-49. UARTCLKGS 寄存器字段说明

位	字段	类型	复位	说明
31-2	RESERVED	R	0h	保留
1-0	CLK_EN	R/W	0h	0：禁用时钟 1：启用时钟 可以通过 UARTCLKGR.AM_CLK_EN 强制开启 要使更改生效，需要写入 CLKLOADCTL.LOAD 1h = 为 UART0 启用时钟 2h = 为 UART1 启用时钟

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.22 UARTCLKGDS 寄存器（偏移 = 74h）[复位 = 00000000h]

图 7-45 展示了 UARTCLKGDS，表 7-50 中对此进行了介绍。

返回到汇总表。

深度睡眠模式下的 UART 时钟门控

图 7-45. UARTCLKGDS 寄存器

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														CLK_EN	
R-0h														R/W-0h	

表 7-50. UARTCLKGDS 寄存器字段说明

位	字段	类型	复位	说明
31-2	RESERVED	R	0h	保留
1-0	CLK_EN	R/W	0h	0：禁用时钟 1：启用时钟 可以通过 UARTCLKGR.AM_CLK_EN 强制开启 要使更改生效，需要写入 CLKLOADCTL.LOAD 1h = 为 UART0 启用时钟 2h = 为 UART1 启用时钟

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.23 SSICLKGR 寄存器（偏移 = 78h）[复位 = 00000000h]

图 7-46 展示了 SSICLKGR，表 7-51 中对此进行了介绍。

返回到汇总表。

运行及所有模式下的 SSI 时钟门控

图 7-46. SSICLKGR 寄存器

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

表 7-51. SSICLKGR 寄存器字段说明

位	字段	类型	复位	说明
31-10	RESERVED	R	0h	保留
9-8	AM_CLK_EN	R/W	0h	0：不强制 1：在所有模式（运行、睡眠和深度睡眠）下强制开启时钟 启用时覆盖 CLK_EN、SSICLKGS.CLK_EN 和 SSICLKGDS.CLK_EN。 要使更改生效，需要写入 CLKLOADCTL.LOAD 1h = 为 SSI0 启用时钟 2h = 为 SSI1 启用时钟
7-2	RESERVED	R	0h	保留
1-0	CLK_EN	R/W	0h	0：禁用时钟 1：启用时钟 可以通过 AM_CLK_EN 强制开启 要使更改生效，需要写入 CLKLOADCTL.LOAD 1h = 为 SSI0 启用时钟 2h = 为 SSI1 启用时钟

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.24 SSICLKGS 寄存器（偏移 = 7Ch）[复位 = 00000000h]

图 7-47 展示了 SSICLKGS，表 7-52 中对此进行了介绍。

返回到汇总表。

睡眠模式下的 SSI 时钟门控

图 7-47. SSICLKGS 寄存器

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														CLK_EN	
R-0h														R/W-0h	

表 7-52. SSICLKGS 寄存器字段说明

位	字段	类型	复位	说明
31-2	RESERVED	R	0h	保留
1-0	CLK_EN	R/W	0h	0：禁用时钟 1：启用时钟 可以通过 SSICLKGR.AM_CLK_EN 强制开启 要使更改生效，需要写入 CLKLOADCTL.LOAD 1h = 为 SSI0 启用时钟 2h = 为 SSI1 启用时钟

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.25 SSICLKGDS 寄存器（偏移 = 80h）[复位 = 00000000h]

图 7-48 展示了 SSICLKGDS，表 7-53 中对此进行了介绍。

返回到汇总表。

深度睡眠模式下的 SSI 时钟门控

图 7-48. SSICLKGDS 寄存器

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														CLK_EN	
R-0h														R/W-0h	

表 7-53. SSICLKGDS 寄存器字段说明

位	字段	类型	复位	说明
31-2	RESERVED	R	0h	保留
1-0	CLK_EN	R/W	0h	0：禁用时钟 1：启用时钟 可以通过 SSICLKGR.AM_CLK_EN 强制开启 要使更改生效，需要写入 CLKLOADCTL.LOAD 1h = 为 SSI0 启用时钟 2h = 为 SSI1 启用时钟

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.26 I2SCLKGR 寄存器（偏移 = 84h）[复位 = 00000000h]

图 7-49 展示了 I2SCLKGR，表 7-54 对其进行了介绍。

返回到汇总表。

运行及所有模式下的 I2S 时钟门控

图 7-49. I2SCLKGR 寄存器

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

表 7-54. I2SCLKGR 寄存器字段说明

位	字段	类型	复位	说明
31-9	RESERVED	R	0h	保留
8	AM_CLK_EN	R/W	0h	0：不强制 1：在所有模式（运行、睡眠和深度睡眠）下强制开启时钟 启用时覆盖 CLK_EN、I2SCLKGS.CLK_EN 和 I2SCLKGDS.CLK_EN。 启用时，SYSBUS 时钟将始终运行 要使更改生效，需要写入 CLKLOADCTL.LOAD
7-1	RESERVED	R	0h	保留
0	CLK_EN	R/W	0h	0：禁用时钟 1：启用时钟 可以通过 AM_CLK_EN 强制开启 要使更改生效，需要写入 CLKLOADCTL.LOAD

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.27 I2SCLKGS 寄存器（偏移 = 88h）[复位 = 00000000h]

图 7-50 展示了 I2SCLKGS，表 7-55 对其进行了介绍。

返回到汇总表。

睡眠模式下的 I2S 时钟门控

图 7-50. I2SCLKGS 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	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_EN
R-0h							R/W-0h

表 7-55. I2SCLKGS 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	CLK_EN	R/W	0h	0：禁用时钟 1：启用时钟 可以通过 I2SCLKGR.AM_CLK_EN 强制开启 要使更改生效，需要写入 CLKLOADCTL.LOAD

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.28 I2SCLKGDS 寄存器（偏移 = 8Ch）[复位 = 00000000h]

图 7-51 展示了 I2SCLKGDS，表 7-56 对其进行了介绍。

返回到汇总表。

深度睡眠模式下的 I2S 时钟门控

图 7-51. I2SCLKGDS 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	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_EN
R-0h							R/W-0h

表 7-56. I2SCLKGDS 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	CLK_EN	R/W	0h	0：禁用时钟 1：启用时钟 启用时 SYSBUS 时钟将始终运行 可以通过 I2SCLKGR.AM_CLK_EN 强制开启 要使更改生效，需要写入 CLKLOADCTL.LOAD

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.29 SYSBUSCLKDIV 寄存器（偏移 = B4h）[复位 = 00000000h]

图 7-52 展示了 SYSBUSCLKDIV，表 7-57 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 7-52. SYSBUSCLKDIV 寄存器

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		

表 7-57. SYSBUSCLKDIV 寄存器字段说明

位	字段	类型	复位	说明
31-3	RESERVED	R	0h	保留
2-0	RATIO	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.30 CPUCLKDIV 寄存器（偏移 = B8h）[复位 = 00000000h]

图 7-53 展示了 CPUCLKDIV，表 7-58 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 7-53. CPUCLKDIV 寄存器

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

表 7-58. CPUCLKDIV 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	RATIO	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.31 PERBUSCPUCLKDIV 寄存器（偏移 = BCh）[复位 = 00000000h]

图 7-54 展示了 PERBUSCPUCLKDIV，表 7-59 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 7-54. PERBUSCPUCLKDIV 寄存器

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			

表 7-59. PERBUSCPUCLKDIV 寄存器字段说明

位	字段	类型	复位	说明
31-4	RESERVED	R	0h	保留
3-0	RATIO	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.32 PERDMACLKDIV 寄存器（偏移 = C4h）[复位 = 00000000h]

图 7-55 展示了 PERDMACLKDIV，表 7-60 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 7-55. PERDMACLKDIV 寄存器

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			

表 7-60. PERDMACLKDIV 寄存器字段说明

位	字段	类型	复位	说明
31-4	RESERVED	R	0h	保留
3-0	RATIO	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.33 I2SBCLKSEL 寄存器（偏移 = C8h）[复位 = 00000000h]

图 7-56 展示了 I2SBCLKSEL，表 7-61 对其进行了介绍。

返回到汇总表。

I2S 时钟控制

图 7-56. I2SBCLKSEL 寄存器

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															SRC
R-0h															R/W-0h

表 7-61. I2SBCLKSEL 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	SRC	R/W	0h	BCLK 源选择器 0：使用外部 BCLK 1：使用内部生成的时钟 要使更改生效，需要写入 CLKLOADCTL.LOAD

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.34 GPTCLKDIV 寄存器（偏移 = CCh）[复位 = 00000000h]

图 7-57 展示了 GPTCLKDIV，表 7-62 中对此进行了介绍。

返回到汇总表。

GPT 分频器

图 7-57. GPTCLKDIV 寄存器

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			

表 7-62. GPTCLKDIV 寄存器字段说明

位	字段	类型	复位	说明
31-4	RESERVED	R	0h	保留
3-0	RATIO	R/W	0h	用于 GPT 的分频器。在运行/睡眠/深度睡眠模式下，分频比将是恒定的且不受门控。要使更改生效，需要写入 CLKLOADCTL.LOAD。不支持其他值。 0h = 1 分频 1h = 2 分频 2h = 4 分频 3h = 8 分频 4h = 16 分频 5h = 32 分频 6h = 64 分频 7h = 128 分频 8h = 256 分频

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.35 I2SCLKCTL 寄存器（偏移 = D0h）[复位 = 00000000h]

图 7-58 展示了 I2SCLKCTL，表 7-63 对其进行了介绍。

返回到汇总表。

I2S 时钟控制

图 7-58. I2SCLKCTL 寄存器

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

表 7-63. I2SCLKCTL 寄存器字段说明

位	字段	类型	复位	说明
31-4	RESERVED	R	0h	保留
3	SMPL_ON_POSEDGE	R/W	0h	在 I2S 串行接口上，在 BCLK 的相反边沿对数据和 WCLK 进行采样并按时钟输出。 0 - 数据和 WCLK 在负边沿采样，在正边沿按时钟输出。 1 - 数据和 WCLK 在正边沿采样，在负边沿按时钟输出。 要使更改生效，需要写入 CLKLOADCTL.LOAD
2-1	WCLK_PHASE	R/W	0h	决定 WCLK 分频比的计算方式，并用于生成不同的占空比（请参阅 I2SWCLKDIV.WDIV）。 0：单相 1：双相 2：用户定义 3：保留/未定义 要使更改生效，需要写入 CLKLOADCTL.LOAD
0	EN	R/W	0h	0：MCLK、BCLK 和 WCLK 将为静态低电平 1：启用 MCLK、BCLK 和 WCLK 的生成 要使更改生效，需要写入 CLKLOADCTL.LOAD

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.36 I2SMCLKDIV 寄存器（偏移 = D4h）[复位 = 00000000h]

图 7-59 展示了 I2SMCLKDIV，表 7-64 对其进行了介绍。

返回到汇总表。

MCLK 分频比

图 7-59. I2SMCLKDIV 寄存器

31	30	29	28	27	26	25	24	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									

表 7-64. I2SMCLKDIV 寄存器字段说明

位	字段	类型	复位	说明
31-10	RESERVED	R	0h	保留
9-0	MDIV	R/W	0h	用于生成 MCLK 的分频比的无符号系数 [2-1024]： MCLK = MCUCLK/MDIV[Hz] MCUCLK 为 48MHz。 值 0 解释为 1024。 值 1 无效。 如果 MDIV 为奇数，则时钟的低电平相位比高电平相位长一个 MCUCLK 周期。 要使更改生效，需要写入 CLKLOADCTL.LOAD

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.37 I2SBCLKDIV 寄存器（偏移 = D8h）[复位 = 00000000h]

图 7-60 展示了 I2SBCLKDIV，表 7-65 对其进行了介绍。

返回到汇总表。

BCLK 分压比

图 7-60. I2SBCLKDIV 寄存器

31	30	29	28	27	26	25	24	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									

表 7-65. I2SBCLKDIV 寄存器字段说明

位	字段	类型	复位	说明
31-10	RESERVED	R	0h	保留
9-0	BDIV	R/W	0h	用于生成 I2S BCLK 的分频比的无符号系数 [2-1024]： BCLK = MCUCLK/BDIV[Hz] MCUCLK 为 48MHz。 值 0 解释为 1024。 值 1 无效。 如果 BDIV 为奇数且 I2SCLKCTL.SMPL_ON_POSEDGE = 0，则时钟的低电平相位比高电平相位长一个 MCUCLK 周期。 如果 BDIV 为奇数且 I2SCLKCTL.SMPL_ON_POSEDGE = 1，则时钟的高电平相位比低电平相位长一个 MCUCLK 周期。 要使更改生效，需要写入 CLKLOADCTL.LOAD

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.38 I2SWCLKDIV 寄存器（偏移 = DCh）[复位 = 00000000h]

图 7-61 展示了 I2SWCLKDIV，表 7-66 中对此进行了介绍。

返回到汇总表。

WCLK 分频比

图 7-61. I2SWCLKDIV 寄存器

31	30	29	28	27	26	25	24	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															

表 7-66. I2SWCLKDIV 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	WDIV	R/W	0h	如果 I2SCLKCTL.WCLK_PHASE = 0，单相。 WCLK 高电平持续一个 BCLK 周期，低电平持续 WDIV[9:0]（无符号，[1-1023]）个 BCLK 周期。 WCLK = MCUCLK / BDIV*(WDIV[9:0] + 1) [Hz] MCUCLK 为 48MHz。 如果 I2SCLKCTL.WCLK_PHASE = 1，双相。 WCLK 的每个相位（50% 占空比）持续 WDIV[9:0]（无符号，[1-1023]）个 BCLK 周期。 WCLK = MCUCLK / BDIV*(2*WDIV[9:0]) [Hz] 如果 I2SCLKCTL.WCLK_PHASE = 2，用户定义。 WCLK 高电平持续 WDIV[7:0]（无符号，[1-255]）个 BCLK 周期，低电平持续 WDIV[15:8]（无符号，[1-255]）个 BCLK 周期。 WCLK = MCUCLK / (BDIV*(WDIV[7:0] + WDIV[15:8]) [Hz] 要使更改生效，需要写入 CLKLOADCTL.LOAD

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.39 RESETSECDMA 寄存器（偏移 = F0h）[复位 = 00000000h]

图 7-62 展示了 RESETSECDMA，表 7-67 中对此进行了介绍。

返回到汇总表。

SEC（PKA、TRNG 和 CRYPTO）与 UDMA 的复位

图 7-62. RESETSECDMA 寄存器

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

表 7-67. RESETSECDMA 寄存器字段说明

位	字段	类型	复位	说明
31-9	RESERVED	R	0h	保留
8	DMA	W	0h	写入 1 复位。硬件清除。 仅在 PERIPH 电源域开启 (PDSTAT0.PERIPH_ON = 1) 时访问才有效 写入前设置 FLASH:CFG.DIS_READACCESS = 1，以确保在从闪存执行期间不会激活复位。这意味着使用软件复位时不能从闪存执行代码。
7-3	RESERVED	R	0h	保留
2	PKA	W	0h	写入 1 复位。硬件清除。 仅在 PERIPH 电源域开启 (PDSTAT0.PERIPH_ON = 1) 时访问才有效 写入前设置 FLASH:CFG.DIS_READACCESS = 1，以确保在从闪存执行期间不会激活复位。这意味着使用软件复位时不能从闪存执行代码。
1	TRNG	W	0h	写入 1 复位。硬件清除。 仅在 PERIPH 电源域开启 (PDSTAT0.PERIPH_ON = 1) 时访问才有效 写入前设置 FLASH:CFG.DIS_READACCESS = 1，以确保在从闪存执行期间不会激活复位。这意味着使用软件复位时不能从闪存执行代码。
0	CRYPTO	W	0h	写入 1 复位。硬件清除。 仅在 PERIPH 电源域开启 (PDSTAT0.PERIPH_ON = 1) 时访问才有效 写入前设置 FLASH:CFG.DIS_READACCESS = 1，以确保在从闪存执行期间不会激活复位。这意味着使用软件复位时不能从闪存执行代码。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.40 RESETGPIO 寄存器（偏移 = F4h）[复位 = 00000000h]

图 7-63 展示了 RESETGPIO，表 7-68 中对此进行了介绍。

返回到汇总表。

GPIO IP 的复位

图 7-63. RESETGPIO 寄存器

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

表 7-68. RESETGPIO 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	GPIO	W	0h	0：无操作 1：复位 GPIO。硬件清除。 仅在 PERIPH 电源域开启 (PDSTAT0.PERIPH_ON = 1) 时访问才有效 写入前设置 FLASH:CFG.DIS_READACCESS = 1，以确保在从闪存执行期间不会激活复位。这意味着使用软件复位时不能从闪存执行代码。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.41 RESETGPT 寄存器（偏移 = F8h）[复位 = 00000000h]

图 7-64 展示了 RESETGPT，表 7-69 中对此进行了介绍。

返回到汇总表。

GPT IP 的复位

图 7-64. RESETGPT 寄存器

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															GPT
R-0h															W-0h

表 7-69. RESETGPT 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	GPT	W	0h	0：无操作 1：复位所有 GPT。硬件清除。 仅在 PERIPH 电源域开启 (PDSTAT0.PERIPH_ON = 1) 时访问才有效 写入前设置 FLASH:CFG.DIS_READACCESS = 1，以确保在从闪存执行期间不会激活复位。这意味着使用软件复位时不能从闪存执行代码。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.42 RESETI2C 寄存器（偏移 = FCh）[复位 = 00000000h]

图 7-65 展示了 RESETI2C，表 7-70 中对此进行了介绍。

返回到汇总表。

I2C IP 的复位

图 7-65. RESETI2C 寄存器

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															I2C
R-0h															W-0h

表 7-70. RESETI2C 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	I2C	W	0h	0：无操作 1：复位 I2C。硬件清除。 仅在 SERIAL 电源域开启 (PDSTAT0.SERIAL_ON = 1) 时访问才有效 写入前设置 FLASH:CFG.DIS_READACCESS = 1，以确保在从闪存执行期间不会激活复位。这意味着使用软件复位时不能从闪存执行代码。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.43 RESETUART 寄存器（偏移 = 100h）[复位 = 00000000h]

图 7-66 展示了 RESETUART，表 7-71 中对此进行了介绍。

返回到汇总表。

UART IP 的复位

图 7-66. RESETUART 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						UART1	UART0
R-0h						W-0h	W-0h

表 7-71. RESETUART 寄存器字段说明

位	字段	类型	复位	说明
31-2	RESERVED	R	0h	保留
1	UART1	W	0h	0：无操作 1：复位 UART1。硬件清除。 仅在 PERIPH 电源域开启 (PDSTAT0.PERIPH_ON = 1) 时访问才有效 写入前设置 FLASH:CFG.DIS_READACCESS = 1，以确保在从闪存执行期间不会激活复位。这意味着使用软件复位时不能从闪存执行代码。
0	UART0	W	0h	0：无操作 1：复位 UART0。硬件清除。 仅在 SERIAL 电源域开启 (PDSTAT0.SERIAL_ON = 1) 时访问才有效 写入前设置 FLASH:CFG.DIS_READACCESS = 1，以确保在从闪存执行期间不会激活复位。这意味着使用软件复位时不能从闪存执行代码。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.44 RESETSSI 寄存器（偏移 = 104h）[复位 = 00000000h]

图 7-67 展示了 RESETSSI，表 7-72 中对此进行了介绍。

返回到汇总表。

SSI IP 的复位

图 7-67. RESETSSI 寄存器

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														SSI	
R-0h														W-0h	

表 7-72. RESETSSI 寄存器字段说明

位	字段	类型	复位	说明
31-2	RESERVED	R	0h	保留
1-0	SSI	W	0h	SSI 0： 0：无操作 1：复位 SSI。硬件清除。 仅在 SERIAL 电源域开启 (PDSTAT0.SERIAL_ON = 1) 时访问才有效 写入前设置 FLASH:CFG.DIS_READACCESS = 1，以确保在从闪存执行期间不会激活复位。这意味着使用软件复位时不能从闪存执行代码。 SSI 1： 0：无操作 1：复位 SSI。硬件清除。 仅在 PERIPH 电源域开启 (PDSTAT0.PERIPH_ON = 1) 时访问才有效 写入前设置 FLASH:CFG.DIS_READACCESS = 1，以确保在从闪存执行期间不会激活复位。这意味着使用软件复位时不能从闪存执行代码。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.45 RESETI2S 寄存器（偏移 = 108h）[复位 = 00000000h]

图 7-68 展示了 RESETI2S，表 7-73 对其进行了介绍。

返回到汇总表。

I2S IP 的复位

图 7-68. RESETI2S 寄存器

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															I2S
R-0h															W-0h

表 7-73. RESETI2S 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	I2S	W	0h	0：无操作 1：复位模块。硬件清除。 仅在 PERIPH 电源域开启 (PDSTAT0.PERIPH_ON = 1) 时访问才有效 写入前设置 FLASH:CFG.DIS_READACCESS = 1，以确保在从闪存执行期间不会激活复位。这意味着使用软件复位时不能从闪存执行代码。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.46 PDCTL0 寄存器（偏移 = 12Ch）[复位 = 00000000h]

图 7-69 展示了 PDCTL0，表 7-74 对其进行了介绍。

返回到汇总表。

电源域控制

图 7-69. PDCTL0 寄存器

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

表 7-74. PDCTL0 寄存器字段说明

位	字段	类型	复位	说明
31-3	RESERVED	R	0h	保留
2	PERIPH_ON	R/W	0h	PERIPH 电源域。 0：PERIPH 电源域断电 1：PERIPH 电源域加电
1	SERIAL_ON	R/W	0h	SERIAL 电源域。 0：SERIAL 电源域断电 1：SERIAL 电源域加电
0	RFC_ON	R/W	0h	0：如果 PDCTL1.RFC_ON = 0，则 RFC 电源域断电 1：RFC 电源域上电

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.47 PDCTL0RFC 寄存器（偏移 = 130h）[复位 = 00000000h]

图 7-70 展示了 PDCTL0RFC，表 7-75 对其进行了介绍。

返回到汇总表。

RFC 电源域控制

图 7-70. PDCTL0RFC 寄存器

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															ON
R-0h															R/W-0h

表 7-75. PDCTL0RFC 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	ON	R/W	0h	PDCTL0.RFC_ON 的别名

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.48 PDCTL0SERIAL 寄存器（偏移 = 134h）[复位 = 00000000h]

图 7-71 展示了 PDCTL0SERIAL，表 7-76 对其进行了介绍。

返回到汇总表。

SERIAL 电源域控制

图 7-71. PDCTL0SERIAL 寄存器

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															ON
R-0h															R/W-0h

表 7-76. PDCTL0SERIAL 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	ON	R/W	0h	PDCTL0.SERIAL_ON 的别名

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.49 PDCTL0PERIPH 寄存器（偏移 = 138h）[复位 = 00000000h]

图 7-72 展示了 PDCTL0PERIPH，表 7-77 对其进行了介绍。

返回到汇总表。

PERIPH 电源域控制

图 7-72. PDCTL0PERIPH 寄存器

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															ON
R-0h															R/W-0h

表 7-77. PDCTL0PERIPH 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	ON	R/W	0h	PDCTL0.PERIPH_ON 的别名

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.50 PDSTAT0 寄存器（偏移 = 140h）[复位 = 00000000h]

图 7-73 展示了 PDSTAT0，表 7-78 对其进行了介绍。

返回到汇总表。

电源域状态

图 7-73. PDSTAT0 寄存器

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

表 7-78. PDSTAT0 寄存器字段说明

位	字段	类型	复位	说明
31-3	RESERVED	R	0h	保留
2	PERIPH_ON	R	0h	PERIPH 电源域。 0：域可能已断电 1：域加电（保证）
1	SERIAL_ON	R	0h	SERIAL 电源域。 0：域可能已断电 1：域加电（保证）
0	RFC_ON	R	0h	RFC 电源域 0：域可能已断电 1：域加电（保证）

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.51 PDSTAT0RFC 寄存器（偏移 = 144h）[复位 = 00000000h]

图 7-74 展示了 PDSTAT0RFC，表 7-79 对其进行了介绍。

返回到汇总表。

RFC 电源域状态

图 7-74. PDSTAT0RFC 寄存器

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															ON
R-0h															R-0h

表 7-79. PDSTAT0RFC 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	ON	R	0h	PDSTAT0.RFC_ON 的别名

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.52 PDSTAT0SERIAL 寄存器（偏移 = 148h）[复位 = 00000000h]

图 7-75 展示了 PDSTAT0SERIAL，表 7-80 对其进行了介绍。

返回到汇总表。

SERIAL 电源域状态

图 7-75. PDSTAT0SERIAL 寄存器

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															ON
R-0h															R-0h

表 7-80. PDSTAT0SERIAL 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	ON	R	0h	PDSTAT0.SERIAL_ON 的别名

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.53 PDSTAT0PERIPH 寄存器（偏移 = 14Ch）[复位 = 00000000h]

图 7-76 展示了 PDSTAT0PERIPH，表 7-81 对其进行了介绍。

返回到汇总表。

PERIPH 电源域状态

图 7-76. PDSTAT0PERIPH 寄存器

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															ON
R-0h															R-0h

表 7-81. PDSTAT0PERIPH 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	ON	R	0h	PDSTAT0.PERIPH_ON 的别名

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.54 PDCTL1 寄存器（偏移 = 17Ch）[复位 = 0000000Ah]

图 7-77 展示了 PDCTL1，表 7-82 对其进行了介绍。

返回到汇总表。

电源域控制

图 7-77. PDCTL1 寄存器

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

表 7-82. PDCTL1 寄存器字段说明

位	字段	类型	复位	说明
31-5	RESERVED	R	0h	保留
4-3	VIMS_MODE	R/W	1h	00：VIMS 电源域仅在 CPU 电源域通电时通电。 01：只要 BUS 电源域通电，VIMS 电源域就会通电。 1X: 下次唤醒时阻止 VIMS 电源域加电。仅当 VIMS 电源域未通电时，此模式才有效。用于自主射频内核。
2	RFC_ON	R/W	0h	0：如果 PDCTL0.RFC_ON = 0，则 RFC 电源域断电。1：RFC 电源域上电。该位应由 RFC 在自主模式下使用，但没有硬件限制阻止系统 CPU 访问该位。
1	CPU_ON	R/W	1h	0：当系统 CPU 指示其空闲时，会导致 CPU 电源域断电。 1：启动 CPU 电源域加电。 该位由 WIC 上电事件自动设置。
0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.55 PDCTL1CPU 寄存器（偏移 = 184h）[复位 = 00000001h]

图 7-78 展示了 PDCTL1CPU，表 7-83 对其进行了介绍。

返回到汇总表。

CPU 电源域直接控制

图 7-78. PDCTL1CPU 寄存器

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															ON
R-0h															R/W-1h

表 7-83. PDCTL1CPU 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	ON	R/W	1h	这是 PDCTL1.CPU_ON 的别名

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.56 PDCTL1RFC 寄存器（偏移 = 188h）[复位 = 00000000h]

图 7-79 展示了 PDCTL1RFC，表 7-84 对其进行了介绍。

返回到汇总表。

RFC 电源域直接控制

图 7-79. PDCTL1RFC 寄存器

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															ON
R-0h															R/W-0h

表 7-84. PDCTL1RFC 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	ON	R/W	0h	这是 PDCTL1.RFC_ON 的别名

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.57 PDCTL1VIMS 寄存器（偏移 = 18Ch）[复位 = 00000001h]

图 7-80 展示了 PDCTL1VIMS，表 7-85 对其进行了介绍。

返回到汇总表。

VIMS 模式直接控制

图 7-80. PDCTL1VIMS 寄存器

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														模式	
R-0h														R/W-1h	

表 7-85. PDCTL1VIMS 寄存器字段说明

位	字段	类型	复位	说明
31-2	RESERVED	R	0h	保留
1-0	模式	R/W	1h	这是 PDCTL1.VIMS_MODE 的别名

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.58 PDSTAT1 寄存器（偏移 = 194h）[复位 = 0000001Ah]

图 7-81 展示了 PDSTAT1，表 7-86 对其进行了介绍。

返回到汇总表。

电源管理器状态

图 7-81. PDSTAT1 寄存器

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

表 7-86. PDSTAT1 寄存器字段说明

位	字段	类型	复位	说明
31-5	RESERVED	R	0h	保留
4	BUS_ON	R	1h	0：BUS 域不可访问 1：BUS 域当前可访问
3	VIMS_ON	R	1h	0：VIMS 域不可访问 1：VIMS 域当前可访问
2	RFC_ON	R	0h	0：RFC 域不可访问 1：RFC 域当前可访问
1	CPU_ON	R	1h	0：CPU 和 BUS 域不可访问 1：CPU 和 BUS 域当前可访问
0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.59 PDSTAT1BUS 寄存器（偏移 = 198h）[复位 = 00000001h]

图 7-82 展示了 PDSTAT1BUS，表 7-87 对其进行了介绍。

返回到汇总表。

BUS 电源域直接读取状态

图 7-82. PDSTAT1BUS 寄存器

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															ON
R-0h															R-1h

表 7-87. PDSTAT1BUS 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	ON	R	1h	这是 PDSTAT1.BUS_ON 的别名

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.60 PDSTAT1RFC 寄存器（偏移 = 19Ch）[复位 = 00000000h]

图 7-83 展示了 PDSTAT1RFC，表 7-88 对其进行了介绍。

返回到汇总表。

RFC 电源域直接读取状态

图 7-83. PDSTAT1RFC 寄存器

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															ON
R-0h															R-0h

表 7-88. PDSTAT1RFC 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	ON	R	0h	这是 PDSTAT1.RFC_ON 的别名

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.61 PDSTAT1CPU 寄存器（偏移 = 1A0h）[复位 = 00000001h]

图 7-84 展示了 PDSTAT1CPU，表 7-89 对其进行了介绍。

返回到汇总表。

CPU 电源域直接读取状态

图 7-84. PDSTAT1CPU 寄存器

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															ON
R-0h															R-1h

表 7-89. PDSTAT1CPU 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	ON	R	1h	这是 PDSTAT1.CPU_ON 的别名

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.62 PDSTAT1VIMS 寄存器（偏移 = 1A4h）[复位 = 00000001h]

图 7-85 展示了 PDSTAT1VIMS，表 7-90 对其进行了介绍。

返回到汇总表。

VIMS 模式直接读取状态

图 7-85. PDSTAT1VIMS 寄存器

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															ON
R-0h															R-1h

表 7-90. PDSTAT1VIMS 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	ON	R	1h	这是 PDSTAT1.VIMS_ON 的别名

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.63 RFCBITS 寄存器（偏移 = 1CCh）[复位 = 00000000h]

图 7-86 展示了 RFCBITS，表 7-91 中对此进行了介绍。

返回到汇总表。

RFC 控制

图 7-86. RFCBITS 寄存器

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

表 7-91. RFCBITS 寄存器字段说明

位	字段	类型	复位	说明
31-0	读取	R/W	0h	RFC 的控制位。RF 内核 CPE 处理器在启动时会自动检查此寄存器，可用于立即指示 CPE 在启动时执行某些任务。支持的功能由 ROM 定义，可能会有所不同。有关更多详细信息，请参阅技术参考手册。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.64 RFCMODESEL 寄存器（偏移 = 1D0h）[复位 = 00000000h]

图 7-87 展示了 RFCMODESEL，表 7-92 中对此进行了介绍。

返回到汇总表。

所选 RFC 模式

图 7-87. RFCMODESEL 寄存器

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													CURR		
R-0h													R/W-0h		

表 7-92. RFCMODESEL 寄存器字段说明

位	字段	类型	复位	说明
31-3	RESERVED	R	0h	保留
2-0	CURR	R/W	0h	选择 RFC 将接受的命令集。只有 RFCMODEHWOPT.AVAIL 允许的模式是可写的。有关详细信息，请参阅技术参考手册。 0h = 选择模式 0 1h = 选择模式 1 2h = 选择模式 2 3h = 选择模式 3 4h = 选择模式 4 5h = 选择模式 5 6h = 选择模式 6 7h = 选择模式 7

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.65 RFCMODEHWOPT 寄存器（偏移 = 1D4h）[复位 = 00000000h]

图 7-88 展示了 RFCMODEHWOPT，表 7-93 中对此进行了介绍。

返回到汇总表。

允许的 RFC 模式

图 7-88. RFCMODEHWOPT 寄存器

31	30	29	28	27	26	25	24	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																								AVAIL							
R-0h																								R-0h							

表 7-93. RFCMODEHWOPT 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-0	AVAIL	R	0h	允许的 RFC 模式。可以允许多个属性。 1h = 允许模式 0 2h = 允许模式 1 4h = 允许模式 2 8h = 允许模式 3 10h = 允许模式 4 20h = 允许模式 5 40h = 允许模式 6 80h = 允许模式 7

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.66 PWRPROFSTAT 寄存器（偏移 = 1E0h）[复位 = 00000001h]

图 7-89 展示了 PWRPROFSTAT，表 7-94 中对此进行了介绍。

返回到汇总表。

功耗分析器寄存器

图 7-89. PWRPROFSTAT 寄存器

31	30	29	28	27	26	25	24	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																								值							
R-0h																								R/W-1h							

表 7-94. PWRPROFSTAT 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-0	值	R/W	1h	软件可以使用这些位来为应用程序添加时间戳。这些位也可通过 testtap 获取，进而使仿真器可以使用它们进行实时分析。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.67 MCUSRAMCFG 寄存器（偏移 = 21Ch）[复位 = 00000020h]

图 7-90 展示了 MCUSRAMCFG，表 7-95 中对此进行了介绍。

返回到汇总表。

MCU SRAM 配置

图 7-90. MCUSRAMCFG 寄存器

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

表 7-95. MCUSRAMCFG 寄存器字段说明

位	字段	类型	复位	说明
31-6	RESERVED	R	0h	保留
5	BM_OFF	R/W	1h	禁用突发模式 0：启用突发模式。 1：关闭突发模式。
4	PAGE	R/W	0h	页面模式选择 0：禁用页面模式。存储器在标准模式下工作 1：启用页面模式。仅选择蝶形阵列的一半。页面模式将根据 PGS 设置选择字的 LSB 半部分或 MSB 半部分。 该模式可用于额外节省电能
3	PGS	R/W	0h	0：页面模式期间 (PAGE = 1) 选择字的 LSB 半部分 1：页面模式期间 (PAGE = 1) 选择字的 MSB 半部分
2	BM	R/W	0h	突发模式启用 0：突发模式禁用。存储器在标准模式下工作。 1：突发模式启用 处于突发模式时，位线预充和字线触发取决于 PCH_F 和 PCH_L。 突发模式可降低有功功耗。
1	PCH_F	R/W	0h	0：在周期的后半部分无位线预充 1：处于突发模式时 (BM = 1)，在周期的后半部分进行位线预充
0	PCH_L	R/W	0h	0：在周期的前半部分无位线预充 1：处于突发模式时 (BM = 1)，在周期的前半部分进行位线预充

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.68 RAMRETEN 寄存器（偏移 = 224h）[复位 = 0000000Bh]

图 7-91 展示了 RAMRETEN，表 7-96 中对此进行了介绍。

返回到汇总表。

存储器保持控制

图 7-91. RAMRETEN 寄存器

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

表 7-96. RAMRETEN 寄存器字段说明

位	字段	类型	复位	说明
31-4	RESERVED	R	0h	保留
3	RFCULL	R/W	1h	0：禁用 RFC ULL SRAM 保持 1：启用 RFC ULL SRAM 保持 存储器受控： CPEULLRAM
2	RFC	R/W	0h	0：禁用 RFC SRAM 保持 1：启用 RFC SRAM 保持 存储器受控： CPERAM MCERAM RFERAM DSBRAM
1-0	VIMS	R/W	3h	0：禁用存储器保持 1：启用存储器保持 位 0：VIMS_TRAM 位 1：VIMS_CRAM 合法模式取决于 VIMS:CTL.MODE 中的设置 00：在 DEEPSLEEP 被置为有效之前，VIMS:CTL.MODE 必须为 OFF - 再次唤醒后必须设置回 CACHE 或 SPLIT 模式。 01：在 DEEPSLEEP 被置为有效之前，VIMS:CTL.MODE 必须为 GPRAM。唤醒后必须保持在 GPRAM 模式，或者先选择 OFF 模式，然后再选择 CACHE 或 SPILT 模式。 10：非法模式 11：无限制

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.69 OSCIMSC 寄存器（偏移 = 290h）[复位 = 00000036h]

图 7-92 展示了 OSCIMSC，表 7-97 中对此进行了介绍。

返回到汇总表。

振荡器中断屏蔽控制

图 7-92. OSCIMSC 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
HFSRCPENDIM	LFSRCDONEIM	XOSCDLFIM	XOSCLFIM	RCOSCDLFIM	RCOSCLFIM	XOSCHFIM	RCOSCHFIM
R/W-0h	R/W-0h	R/W-1h	R/W-1h	R/W-0h	R/W-1h	R/W-1h	R/W-0h

表 7-97. OSCIMSC 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7	HFSRCPENDIM	R/W	0h	0：当 HFSRCPEND 符合条件时禁用中断生成 1：当 HFSRCPEND 符合条件时启用中断生成
6	LFSRCDONEIM	R/W	0h	0：当 LFSRCDONE 符合条件时禁用中断生成 1：当 LFSRCDONE 符合条件时启用中断生成
5	XOSCDLFIM	R/W	1h	0：当 XOSCDLF 符合条件时禁用中断生成 1：当 XOSCDLF 符合条件时启用中断生成
4	XOSCLFIM	R/W	1h	0：当 XOSCLF 符合条件时禁用中断生成 1：当 XOSCLF 符合条件时启用中断生成
3	RCOSCDLFIM	R/W	0h	0：当 RCOSCDLF 符合条件时禁用中断生成 1：当 RCOSCDLF 符合条件时启用中断生成
2	RCOSCLFIM	R/W	1h	0：当 RCOSCLF 符合条件时禁用中断生成 1：当 RCOSCLF 符合条件时启用中断生成
1	XOSCHFIM	R/W	1h	0：当 XOSCHF 符合条件时禁用中断生成 1：当 XOSCHF 符合条件时启用中断生成
0	RCOSCHFIM	R/W	0h	0：当 RCOSCHF 符合条件时禁用中断生成 1：当 RCOSCHF 符合条件时启用中断生成

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.70 OSCRIS 寄存器（偏移 = 294h）[复位 = 00000000h]

图 7-93 展示了 OSCRIS，表 7-98 中对此进行了介绍。

返回到汇总表。

振荡器原始中断状态

图 7-93. OSCRIS 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
HFSRCPENDRIS	LFSRCDONERIS	XOSCDLFRIS	XOSCLFRIS	RCOSCDLFRIS	RCOSCLFRIS	XOSCHFRIS	RCOSCHFRIS
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

表 7-98. OSCRIS 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7	HFSRCPENDRIS	R	0h	SCLK_HF 源切换挂起中断。 在写入 DDI_0_OSC:CTL0.SCLK_HF_SRC_SEL 导致 SCLK_HF 源变更请求后，所请求的 SCLK_HF 源将启用并符合条件。当新源准备好用作时钟源时，中断 HSSRCPENDRIS 将变为高电平。当闪存允许在闪存读取访问被禁用后进行 SCLK_HF 源切换时。此时将执行实际的 SCLK_HF 时钟源切换，并且中断状态 HSSRCPENDRIS 将变为低电平。 0：指示 SCLK_HF 源未准备好进行切换 1：指示 SCLK_HF 源已准备好进行切换 无论 OSCIMSC.HFSRCPENDIM 设置如何，中断都符合条件。验证原始中断和使能中断屏蔽的顺序对于生成 OSC 中断无关紧要。 由硬件置位。通过写入 OSCICR.HFSRCPENDC 清除
6	LFSRCDONERIS	R	0h	SCLK_LF 源切换完成。 DDI_0_OSC:CTL0.SCLK_LF_SRC_SEL 寄存器字段用于请求更改 SCLK_LF 源。在请求更改 SCLK_LF 时钟源后，新源可能需要先启用并符合条件，然后才能进行时钟源切换。中断 LFRSRCDONERIS 变为高电平，指示已执行 SCLK_LF 时钟源切换。当通过写入 DDI_0_OSC:CTL0.SCLK_LF_SRC_SEL 请求下一次时钟源更改时，LFRSRCDONERIS 将再次变为低电平。 0：指示 SCLK_LF 源切换尚未完成 1：指示 SCLK_LF 源切换已完成 无论 OSCIMSC.LFSRCDONEIM 设置如何，中断都符合条件。验证原始中断和使能中断屏蔽的顺序对于生成 OSC 中断无关紧要。 由硬件置位。通过写入 OSCICR.LFSRCDONEC 清除

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 7-98. OSCRIS 寄存器字段说明（续）

位	字段	类型	复位	说明
5	XOSCDLFRIS	R	0h	XOSCDLFRIS 中断指示 XOSC_HF 振荡器何时准备好用作 SCLK_LF 或 ACLK_REF 的派生低频时钟源。当 XOSCDLFRIS 为高电平时，XOSC_HF 被选择后将用作 SCLK_LF 的源。当没有任何系统时钟选择 XOSC_HF 作为时钟源时，XOSC_HF 源会自动禁用，且 XOSCDLFRIS 中断状态将变为低电平。 0：XOSCDLF 尚不符合条件 1：XOSCDLF 已符合条件 无论 OSCIMSC.XOSCDLFIM 设置如何，中断都符合条件。验证原始中断和使能中断屏蔽的顺序对于生成 OSC 中断无关紧要。 由硬件置位。通过写入 OSCICR.XOSCDLFC 清除
4	XOSCLFRIS	R	0h	XOSCLFRIS 中断指示 XOSC_LF 振荡器的输出何时在频率上符合条件。当 XOSC_LF 振荡器准备好用作时钟源时，XOSCLFRIS 中断状态变为高电平。 时钟成功通过合格性验证后，XOSCLFRIS 中断状态保持高电平，并且会关闭进一步的合格性验证，直到 XOSC_LF 振荡器被禁用。仅在初始上电时或在 XOSC_LF 振荡器因被取消选择为时钟源而被禁用后，XOSCLFRIS 中断状态才会变为低电平。 0：XOSCLF 尚不符合条件 1：XOSCLF 已符合条件 无论 OSCIMSC.XOSCLFIM 设置如何，中断都符合条件。验证原始中断和使能中断屏蔽的顺序对于生成 OSC 中断无关紧要。 由硬件置位。通过写入 OSCICR.XOSCLFC 清除
3	RCOSCDLFRIS	R	0h	RCOSCDLFRIS 中断指示 RCOSC_HF 振荡器何时准备好用作 SCLK_LF 或 ACLK_REF 的派生低频时钟源。当 RCOSCDLFRIS 为高电平时，RCOSC_HF 被选择后将用作 SCLK_LF 的源。当没有任何系统时钟选择 RCOSC_HF 作为时钟源时，RCOSC_HF 源会自动禁用，且 RCOSCDLFRIS 中断状态将变为低电平。 如果 SCLK_LF 或 ACLK_REF 源从 RCOSC_HF 派生时钟更改为 XOSC_HF 派生低频时钟，且新源尚不符合条件，则时钟将继续使用原始源运行。然后，RCOSCDLFRIS 中断将保持高电平。 0：RCOSCDLF 尚不符合条件 1：RCOSCDLF 已符合条件 无论 OSCIMSC.RCOSCDLFIM 设置如何，中断都符合条件。验证原始中断和使能中断屏蔽的顺序对于生成 OSC 中断无关紧要。 由硬件置位。通过写入 OSCICR.RCOSCDLFC 清除
2	RCOSCLFRIS	R	0h	RCOSCLFRIS 中断指示 RCOSC_LF 振荡器的输出何时在频率上符合条件。当 RCOSC_LF 振荡器准备好用作时钟源时，RCOSCLFRIS 中断状态变为高电平。 时钟成功通过合格性验证后，RCOSCLFRIS 中断状态保持高电平，并且会关闭进一步的合格性验证，直到 RCOSC_LF 振荡器被禁用。仅在初始上电时或在 RCOSC_LF 振荡器因被取消选择为时钟源而被禁用后，RCOSCLFRIS 中断状态才会变为低电平。 0：RCOSCLF 尚不符合条件 1：RCOSCLF 已符合条件 无论 OSCIMSC.RCOSCLFIM 设置如何，中断都符合条件。验证原始中断和使能中断屏蔽的顺序对于生成 OSC 中断无关紧要。 由硬件置位。通过写入 OSCICR.RCOSCLFC 清除
1	XOSCHFRIS	R	0h	XOSCHFRIS 中断指示 XOSC_HF 振荡器何时符合用作时钟源的条件。XOSCHFRIS 也在 TCXO 模式下（当 DDI_0_OSC：XOSCHFCTL.TCXO_MODE 为 1 时）使用。 当 XOSCHFRIS 中断为高电平时，表示振荡器符合条件，在被选择后将用作时钟源。当振荡器在被取消选择为时钟源后被禁用时，XOSCHFRIS 中断变为低电平。 0：XOSC_HF 尚不符合条件 1：XOSC_HF 已符合条件 无论 OSCIMSC.XOSCHFIM 设置如何，中断都符合条件。验证原始中断和使能中断屏蔽的顺序对于生成 OSC 中断无关紧要。 由硬件置位。通过写入 OSCICR.XOSCHFC 清除

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 7-98. OSCRIS 寄存器字段说明（续）

位	字段	类型	复位	说明
0	RCOSCHFRIS	R	0h	RCOSCHFRIS 中断指示 RCOSC_HF 振荡器何时符合用作时钟源的条件。当 RCOSCHFRIS 中断为高电平时，表示振荡器符合条件，在被选择后将用作时钟源。当振荡器在被取消选择为时钟源后被禁用时，RCOSCHFRIS 中断变为低电平。 0：RCOSC_HF 尚不符合条件 1：RCOSC_HF 已符合条件 无论 OSCIMSC.RCOSCHFIM 设置如何，中断都符合条件。验证原始中断和使能中断屏蔽的顺序对于生成 OSC 中断无关紧要。 由硬件置位。通过写入 OSCICR.RCOSCHFC 清除

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.2.71 OSCICR 寄存器（偏移 = 298h）[复位 = 00000000h]

图 7-94 展示了 OSCICR，表 7-99 中对此进行了介绍。

返回到汇总表。

振荡器原始中断清除

图 7-94. OSCICR 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
HFSRCPENDC	LFSRCDONEC	XOSCDLFC	XOSCLFC	RCOSCDLFC	RCOSCLFC	XOSCHFC	RCOSCHFC
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

表 7-99. OSCICR 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7	HFSRCPENDC	W	0h	向该字段写入 1 会清除 HFSRCPEND 原始中断状态。写入 0 无效。
6	LFSRCDONEC	W	0h	向该字段写入 1 会清除 LFSRCDONE 原始中断状态。写入 0 无效。
5	XOSCDLFC	W	0h	向该字段写入 1 会清除 XOSCDLF 原始中断状态。写入 0 无效。
4	XOSCLFC	W	0h	向该字段写入 1 会清除 XOSCLF 原始中断状态。写入 0 无效。
3	RCOSCDLFC	W	0h	向该字段写入 1 会清除 RCOSCDLF 原始中断状态。写入 0 无效。
2	RCOSCLFC	W	0h	向该字段写入 1 会清除 RCOSCLF 原始中断状态。写入 0 无效。
1	XOSCHFC	W	0h	向该字段写入 1 会清除 XOSCHF 原始中断状态。写入 0 无效。
0	RCOSCHFC	W	0h	向该字段写入 1 会清除 RCOSCHF 原始中断状态。写入 0 无效。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.3 AON_PMCTL 寄存器

表 7-100 列出了 AON_PMCTL 寄存器的存储器映射寄存器。表 7-100 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

表 7-100. AON_PMCTL 寄存器

偏移	首字母缩写词	寄存器名称	部分
4h	AUXSCECLK	AUX SCE 时钟管理	AUXSCECLK 寄存器 (偏移 = 4h) [复位 = 00000000h]
8h	RAMCFG	RAM 配置	RAMCFG 寄存器 (偏移 = 8h) [复位 = 0001000Fh]
10h	PWRCTL	电源管理控制	PWRCTL 寄存器 (偏移 = 10h) [复位 = 00000000h]
14h	PWRSTAT	AON 电源和复位状态	PWRSTAT 寄存器 (偏移 = 14h) [复位 = 03C00003h]
18h	SHUTDOWN	关断控制	SHUTDOWN 寄存器 (偏移 = 18h) [复位 = 00000000h]
1Ch	RECHARGECFG	充电控制器配置	RECHARGECFG 寄存器 (偏移 = 1Ch) [复位 = C0000000h]
20h	RECHARGESTAT	充电控制器状态	RECHARGESTAT 寄存器 (偏移 = 20h) [复位 = 00000000h]
24h	OSCCFG	振荡器配置	OSCCFG 寄存器 (偏移 = 24h) [复位 = 00000000h]
28h	RESETCTL	复位管理	RESETCTL 寄存器 (偏移 = 28h) [复位 = 000001C0h]
2Ch	SLEEPCTL	睡眠控制	SLEEPCTL 寄存器 (偏移 =2Ch) [复位 = 00000000h]
34h	JTAGCFG	JTAG 配置	JTAGCFG 寄存器 (偏移 = 34h) [复位 = 00000100h]
3Ch	JTAGUSERCODE	JTAG USERCODE	JTAGUSERCODE 寄存器 (偏移 = 3Ch) [复位 = 0B99A02Fh]

复杂的位访问类型经过编码可适应小型表单元。表 7-101 展示了适用于此部分中访问类型的代码。

表 7-101. AON_PMCTL 访问类型代码

访问类型	代码	说明
读取类型		
R	R	读取
写入类型		
W	W	写入
W1C	W 1C	写入 1 以清零
复位或默认值		
-n		复位后的值或默认值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.3.1 AUXSCECLK 寄存器（偏移 = 4h）[复位 = 00000000h]

图 7-95 展示了 AUXSCECLK，表 7-102 对其进行了介绍。

返回到汇总表。

AUX SCE 时钟管理
该寄存器包含用于设置 AUX 域时钟的相关位字段。

图 7-95. AUXSCECLK 寄存器

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

表 7-102. AUXSCECLK 寄存器字段说明

位	字段	类型	复位	说明
31-9	RESERVED	R	0h	保留
8	PD_SRC	R/W	0h	当 AUX 处于掉电模式时，选择 AUX 域的时钟源。 注意：切换时钟源保证无干扰 0h = 无时钟 1h = LF 时钟 (SCLK_LF)
7-1	RESERVED	R	0h	保留
0	SRC	R/W	0h	当 AUX 处于工作模式时，选择 AUX 域的时钟源。 注意：切换时钟源保证无干扰 0h = HF 时钟除以 2 (SCLK_HFDIV2) 1h = MF 时钟 (SCLK_MF)

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.3.2 RAMCFG 寄存器（偏移 = 8h）[复位 = 0001000Fh]

图 7-96 展示了 RAMCFG，表 7-103 中对此进行了介绍。

返回到汇总表。

RAM 配置
该寄存器包含与 MCU 和 AUX 域中 SRAM 的电源管理相关配置。

图 7-96. RAMCFG 寄存器

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

表 7-103. RAMCFG 寄存器字段说明

位	字段	类型	复位	说明
31-18	RESERVED	R	0h	保留
17	AUX_SRAM_PWR_OFF	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
16	AUX_SRAM_RET_EN	R/W	1h	内部。仅可通过 TI 提供的 API 使用。
15-4	RESERVED	R	0h	保留
3-0	BUS_SRAM_RET_EN	R/W	Fh	MCU SRAM 被划分为 5 个存储体。该寄存器控制在 MCU 总线域掉电期间哪些存储体保留数据 0h = 数据保留已禁用 1h = SRAM:BANK0 和 SRAM:BANK1 开启数据保留 3h = SRAM:BANK0、SRAM:BANK1 和 SRAM:BANK2 开启数据保留 7h = SRAM:BANK0、SRAM:BANK1、SRAM:BANK2 和 SRAM:BANK3 开启数据保留 Fh = 所有存储体（SRAM:BANK0、SRAM:BANK1、SRAM:BANK2、SRAM:BANK3 和 SRAM:BANK4）均开启数据保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.3.3 PWRCTL 寄存器（偏移 = 10h）[复位 = 00000000h]

图 7-97 展示了 PWRCTL，表 7-104 中对此进行了介绍。

返回到汇总表。

电源管理控制
该寄存器控制用于设置底层电源管理特性的位字段，例如为 VDDR 电源选择稳压器，以及控制 IO 环中特定段的启用或禁用。

图 7-97. PWRCTL 寄存器

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

表 7-104. PWRCTL 寄存器字段说明

位	字段	类型	复位	说明
31-3	RESERVED	R	0h	保留
2	DCDC_ACTIVE	R/W	0h	选择在工作模式下为 VDDR 使用 DCDC 稳压器 0：在工作模式下使用 GLDO 对 VDDR 进行调节。 1：在工作模式下使用 DCDC 对 VDDR 进行调节。 要在工作模式下使用 DCDC 作为 VDDR 的稳压器，还须设置 DCDC_EN
1	EXT_REG_MODE	R	0h	VDDRsupply 的源状态： 0：DCDC 或 GLDO 正在生成 VDDR 1：旁路 DCDC 和 GLDO，并由外部稳压器为 VDDR 供电
0	DCDC_EN	R/W	0h	选择在 VDDR 充电期间使用 DCDC 稳压器 0：使用 GLDO 为 VDDR 充电 1：使用 DCDC 为 VDDR 充电 注意：此位字段应设置为与 DCDC_ACTIVE 相同

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.3.4 PWRSTAT 寄存器（偏移 = 14h）[复位 = 03C00003h]

图 7-98 展示了 PWRSTAT，表 7-105 对其进行了介绍。

返回到汇总表。

AON 电源和复位状态
该寄存器用于监控 AON 中的各种电源管理相关信号。除 JTAG_PD_ON、AUX_BUS_RESET_DONE 和 AUX_RESET_DONE 之外的所有其他信号仅用于测试、校准和调试目的。

图 7-98. PWRSTAT 寄存器

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

表 7-105. PWRSTAT 寄存器字段说明

位	字段	类型	复位	说明
31-3	RESERVED	R	0h	保留
2	JTAG_PD_ON	R	0h	表示 JTAG 电源状态： 0：JTAG 已掉电 1：JTAG 已通电
1	AUX_BUS_RESET_DONE	R	1h	指示 AUX 总线复位完成： 0：AUX 总线正在复位 1：AUX 总线复位已解除
0	AUX_RESET_DONE	R	1h	指示 AUX 复位完成： 0：AUX 正在复位 1：AUX 复位已解除

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.3.5 SHUTDOWN 寄存器（偏移 = 18h）[复位 = 00000000h]

图 7-99 展示了 SHUTDOWN，表 7-106 对其进行了介绍。

返回到汇总表。

关断控制
此寄存器包含进入关断模式所需的位字段

图 7-99. SHUTDOWN 寄存器

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															EN
R-0h															R/W-0h

表 7-106. SHUTDOWN 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	EN	R/W	0h	关断控制。 0：请勿为该位写入 0。 1：立即启动进入关断模式的流程

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.3.6 RECHARGECFG 寄存器（偏移 = 1Ch）[复位 = C0000000h]

图 7-100 展示了 RECHARGECFG，表 7-107 对其进行了介绍。

返回到汇总表。

充电控制器配置
该寄存器设置用于控制充电算法的所有相关参数。

图 7-100. RECHARGECFG 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
模式		RESERVED						C2				C1			
R/W-3h		R-0h						R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAX_PER_M					MAX_PER_E			PER_M				PER_E			
R/W-0h					R/W-0h			R/W-0h				R/W-0h			

表 7-107. RECHARGECFG 寄存器字段说明

位	字段	类型	复位	说明
31-30	模式	R/W	3h	当系统在 uLDO 上运行时，为 VDDR 选择充电算法 0h = 充电已禁用 1h = 静态计时器 2h = 自适应计时器 3h = 外部充电比较器。 请注意，在选择该充电算法之前，必须启用充电比较器的时钟 [ANATOP_MMAP:ADI_3_REFSYS:CTL_RECHARGE_CMP0:COMP_CLK_DISABLE]。
29-24	RESERVED	R	0h	保留
23-20	C2	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
19-16	C1	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
15-11	MAX_PER_M	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
10-8	MAX_PER_E	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
7-3	PER_M	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
2-0	PER_E	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.3.7 RECHARGESTAT 寄存器（偏移 = 20h）[复位 = 00000000h]

图 7-101 展示了 RECHARGESTAT，表 7-108 对其进行了介绍。

返回到汇总表。

充电控制器状态
该寄存器控制在充电期间更新的各种状态寄存器。该寄存器主要用于测试和调试。

图 7-101. RECHARGESTAT 寄存器

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

表 7-108. RECHARGESTAT 寄存器字段说明

位	字段	类型	复位	说明
31-20	RESERVED	R	0h	保留
19-16	VDDR_SMPLS	R	0h	最后 4 个 VDDR 采样值。 对于每个位： 0：开始充电时，VDDR 低于 VDDR_OK 阈值 1：开始充电时，VDDR 高于 VDDR_OK 阈值 该寄存器会在每个充电周期之前更新并左移，位 0 会更新为最后一个 VDDR 采样值。
15-0	MAX_USED_PER	R/W	0h	显示两个充电周期之间所经历的最大 32kHz 周期数，且在后一次充电开始时 VDDR 仍高于 VDDR_OK 阈值。该寄存器可用于指示待机期间的泄漏电流。 写入此寄存器时，此位字段将清零。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.3.8 OSCCFG 寄存器（偏移 = 24h）[复位 = 00000000h]

图 7-102 展示了 OSCCFG，表 7-109 中对此进行了介绍。

返回到汇总表。

振荡器配置
该寄存器设置发送到振荡器控制系统的振幅补偿请求的周期。振幅补偿仅在 XOSC_HF 在低功耗模式下运行时适用。

图 7-102. OSCCFG 寄存器

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								PER_M					PER_E		
R-0h								R/W-0h					R/W-0h		

表 7-109. OSCCFG 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-3	PER_M	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
2-0	PER_E	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.3.9 RESETCTL 寄存器（偏移 = 28h）[复位 = 000001C0h]

图 7-103 展示了 RESETCTL，表 7-110 对其进行了介绍。

返回到汇总表。

复位管理
此寄存器包含与系统复位相关的位字段，例如复位源和复位请求以及欠压复位控制。

图 7-103. RESETCTL 寄存器

31	30	29	28	27	26	25	24
SYSRESET	RESERVED					BOOT_DET_1_CLR	BOOT_DET_0_CLR
W-0h	R-0h					W-0h	W-0h
23	22	21	20	19	18	17	16
RESERVED						BOOT_DET_1_SET	BOOT_DET_0_SET
R-0h						W-0h	W-0h
15	14	13	12	11	10	9	8
WU_FROM_SD	GPIO_WU_FROM_SD	BOOT_DET_1	BOOT_DET_0	RESERVED			VDDS_LOSS_EN
R-0h	R-0h	R-0h	R-0h	R-0h			R/W-1h
7	6	5	4	3	2	1	0
VDDR_LOSS_EN	VDD_LOSS_EN	CLK_LOSS_EN	MCU_WARM_RESET	RESET_SRC			RESERVED
R/W-1h	R/W-1h	R/W-0h	R/W1C-0h	R-0h			R-0h

表 7-110. RESETCTL 寄存器字段说明

位	字段	类型	复位	说明
31	SYSRESET	W	0h	冷复位寄存器。向此位字段写入 1 将复位整个芯片并导致引导代码再次运行。 0：无效 1：生成系统复位。在 RESET_SRC 中显示为 SYSRESET
30-26	RESERVED	R	0h	保留
25	BOOT_DET_1_CLR	W	0h	内部。仅可通过 TI 提供的 API 使用。
24	BOOT_DET_0_CLR	W	0h	内部。仅可通过 TI 提供的 API 使用。
23-18	RESERVED	R	0h	保留
17	BOOT_DET_1_SET	W	0h	内部。仅可通过 TI 提供的 API 使用。
16	BOOT_DET_0_SET	W	0h	内部。仅可通过 TI 提供的 API 使用。
15	WU_FROM_SD	R	0h	发生 IO 事件时，或由于连接了调试器，从 SHUTDOWN 状态唤醒已发生。（TCK 引脚被强制为低电平） 请参阅 IOC:IOCFGn.WU_CFG 以将 IO 配置为唤醒源。 0：如 RESET_SRC 所示，从冷复位或欠压状态唤醒已发生 1：从 SHUTDOWN 状态唤醒已发生 注意：当 SLEEPCTL.IO_PAD_SLEEP_DIS 置为有效时，该标志将被清除。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 7-110. RESETCTL 寄存器字段说明 （续）

位	字段	类型	复位	说明
14	GPIO_WU_FROM_SD	R	0h	发生 IO 事件时，从 SHUTDOWN 状态唤醒已发生 请参阅 IOC:IOCFGn.WU_CFG 以将 IO 配置为唤醒源。 0：发生 IO 事件时，从 SHUTDOWN 状态唤醒未发生 1：由于 IO 事件，从 SHUTDOWN 状态唤醒已发生 WU_FROM_SD 置为有效但此位字段未置为有效的情况仅会在调试会话中发生。在此位字段也置为有效之前，引导代码不会继续执行从 SHUTDOWN 状态唤醒过程。 注意：当 SLEEPCTL.IO_PAD_SLEEP_DIS 置为有效时，该标志将被清除。
13	BOOT_DET_1	R	0h	内部。仅可通过 TI 提供的 API 使用。
12	BOOT_DET_0	R	0h	内部。仅可通过 TI 提供的 API 使用。
11-9	RESERVED	R	0h	保留
8	VDDS_LOSS_EN	R/W	1h	在 VDDS 丢失的情况下控制复位生成 0：除非 VDDS_LOSS_EN_OVR=1，否则会忽略 VDDS 的欠压检测 1：VDDS 的欠压检测会生成系统复位
7	VDDR_LOSS_EN	R/W	1h	在 VDDR 丢失的情况下控制复位生成 0：除非 VDDR_LOSS_EN_OVR=1，否则会忽略 VDDR 的欠压检测 1：VDDR 的欠压检测会生成系统复位
6	VDD_LOSS_EN	R/W	1h	在 VDD 丢失的情况下控制复位生成 0：除非 VDD_LOSS_EN_OVR=1，否则会忽略 VDD 的欠压检测 1：VDD 的欠压检测会生成系统复位
5	CLK_LOSS_EN	R/W	0h	通过 [ANATOP_MMAP:DDI_0_OSC:CTL0.CLK_LOSS_EN] 启用时钟丢失检测后，如果 SCLK_LF、SCLK_MF 或 SCLK_HF 丢失，则控制复位生成 0：忽略时钟丢失 1：时钟丢失会生成系统复位 注意：更改 SCLK_LF 的时钟源时，必须禁用时钟丢失复位生成。否则可能导致伪波系统复位。时钟丢失复位生成由 [ANATOP_MMAP:DDI_0_OSC:CTL0.CLK_LOSS_EN] 控制
4	MCU_WARM_RESET	R/W1C	0h	内部。仅可通过 TI 提供的 API 使用。
3-1	RESET_SRC	R	0h	显示了上次系统复位的根本原因。在上次系统复位期间，超过报告的复位源可能已处于活动状态，但仅报告了根本原因。 在释放所有可能的复位源并且结果复制到 AON_PMCTL 之前，不会重新布设捕获功能。在复制和重新布设过程中，这是一个 2MHz 周期，在该周期内，无论根本原因如何，最终的新系统复位都将报告为上电复位。 0h = 上电复位 1h = 复位引脚 2h = VDDS 上的欠压检测 4h = VDDR 上的欠压检测 5h = SCLK_LF、SCLK_MF 或 SCLK_HF 时钟丢失检测 6h = 通过 SYSRESET 进行的软件复位或硬件电源管理超时检测。 注意： 硬件电源管理超时电路始终处于启用状态。 7h = 通过 PRCM 热复位请求进行软件复位
0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.3.10 SLEEPCTL 寄存器（偏移 =2Ch）[复位 = 00000000h]

图 7-104 展示了 SLEEPCTL，表 7-111 对其进行了介绍。

返回到汇总表。

睡眠控制
该寄存器用于从 SHUTDOWN 状态唤醒后解冻 IO 焊盘环

图 7-104. SLEEPCTL 寄存器

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

表 7-111. SLEEPCTL 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	IO_PAD_SLEEP_DIS	R/W	0h	控制 I/O 焊盘睡眠模式。除非从 SHUTDOWN 状态唤醒，否则引导代码将自动设置此位字段（设置了 RESETCTL.WU_FROM_SD）。 0：I/O 焊盘睡眠模式已启用，这意味着所有输出和焊盘配置都会被锁存。如果在 IO_PAD_SLEEP_DIS 设置为 1 之前将焊盘配置为输入，则输入为透明状态 1：I/O 焊盘睡眠模式已禁用 从 SHUTDOWN 状态唤醒后，应用软件必须先重新配置所有 IO 的状态，然后才能设置此位字段，以避免引脚干扰。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.3.11 JTAGCFG 寄存器（偏移 = 34h）[复位 = 00000100h]

图 7-105 展示了 JTAGCFG，表 7-112 对其进行了介绍。

返回到汇总表。

JTAG 配置
该寄存器包含对 JTAG 域配置的控制。这包括每个 TAP 的权限。

图 7-105. JTAGCFG 寄存器

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

表 7-112. JTAGCFG 寄存器字段说明

位	字段	类型	复位	说明
31-9	RESERVED	R	0h	保留
8	JTAG_PD_FORCE_ON	R/W	1h	控制 JTAG 电源域的电源状态： 0：只能由调试子系统控制。（除非连接了调试器，否则 JTAG 电源域将断电） 1：JTAG 电源域被强制启动，与调试子系统无关。 注意：默认情况下，复位值会导致 JTAG 电源域上电。软件必须清除该位，才能关闭 JTAG 电源域
7-0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

7.8.3.12 JTAGUSERCODE 寄存器（偏移 = 3Ch）[复位 = 0B99A02Fh]

图 7-106 展示了 JTAGUSERCODE，表 7-113 中对此进行了介绍。

返回到汇总表。

JTAG USERCODE
引导代码将 JTAG USERCODE 复制到此寄存器，并从此处转发到调试子系统。

图 7-106. JTAGUSERCODE 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USER_CODE																															
R/W-0B99A02Fh																															

表 7-113. JTAGUSERCODE 寄存器字段说明

位	字段	类型	复位	说明
31-0	USER_CODE	R/W	0B99A02Fh	32 位 JTAG USERCODE 寄存器向主 JTAG TAP 馈电 注意：此字段可通过 LOCKCFG.LOCK 锁定

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

章节 8

多功能指令存储器系统 (VIMS)

本章探讨了 CC13x2 和 CC26x2 器件平台的多功能指令存储器系统 (VIMS)。

8.1 简介.....637
8.2 VIMS 配置.....638
8.3 VIMS 软件注意事项.....640
8.4 ROM.....642
8.5 闪存.....642
8.6 ROM 功能.....645
8.7 VIMS 寄存器.....645

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.1 简介

主指令存储器封装在多功能指令存储器系统 (VIMS) 模块中，其中包括以下存储器：

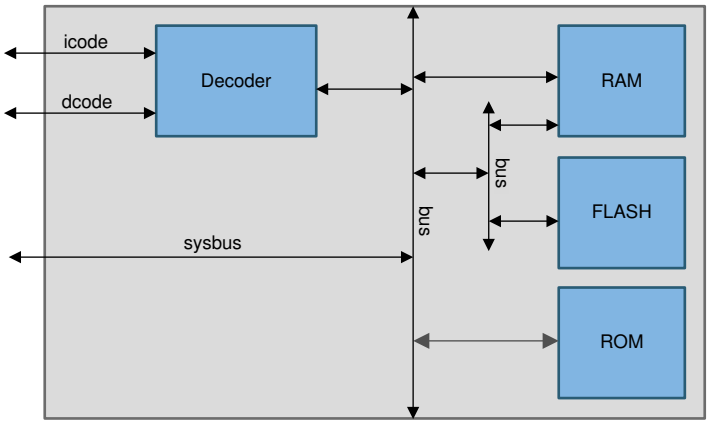
- 352kB 闪存
- 8kB 或 RAM 高速缓存或通用 RAM (GPRAM)
- 255kB 引导 ROM

图 8-1 展示了 VIMS 模块的概况。

VIMS 模块将 CPU 访问 (icode/dcode) 和系统总线访问转发到已寻址的存储器。VIMS 模块还对 CPU 和系统总线之间的访问进行仲裁。

VIMS 模块在 48MHz 系统时钟上运行。

可通过用户软件、调试接口和 ROM 引导加载程序对闪存进行编程。RAM 块可用作闪存块的高速缓存或 GPRAM。



Copyright © 2017, Texas Instruments Incorporated

图 8-1. VIMS 概述

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.2 VIMS 配置

8.2.1 VIMS 模式

RAM 块的工作方式如下：

- GPRAM
- CACHE
- OFF

当前模式显示在 VIMS:STAT.MODE 寄存器中，模式切换通过 VIMS:CTL.MODE 寄存器进行控制。图 8-2 展示了模式的转换。所有模式更改均通过软件启动。*无效* 状态是由硬件控制的过渡状态。无效化会初始化 RAM 块的整个内容并需要 1029 个时钟周期来执行。

启动模式更改（如 VIMS:STATUS.MODE_CHANGING 寄存器中所示）后，必须先完成模式更改，然后才能启动另一个模式更改。在模式更改期间，VIMS:CTL.MODE 寄存器被阻止进行更新。

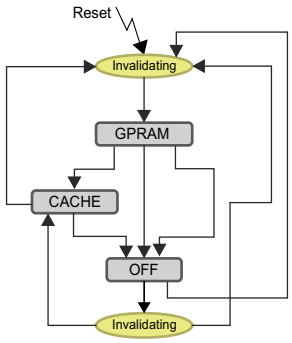


图 8-2. VIMS 模式切换流程图

8.2.1.1 GPRAM 模式

在 GPRAM 模式下，RAM 块用作通用 RAM（请参阅图 8-3）。闪存块不支持缓存，对闪存的所有访问都直接路由到闪存块。

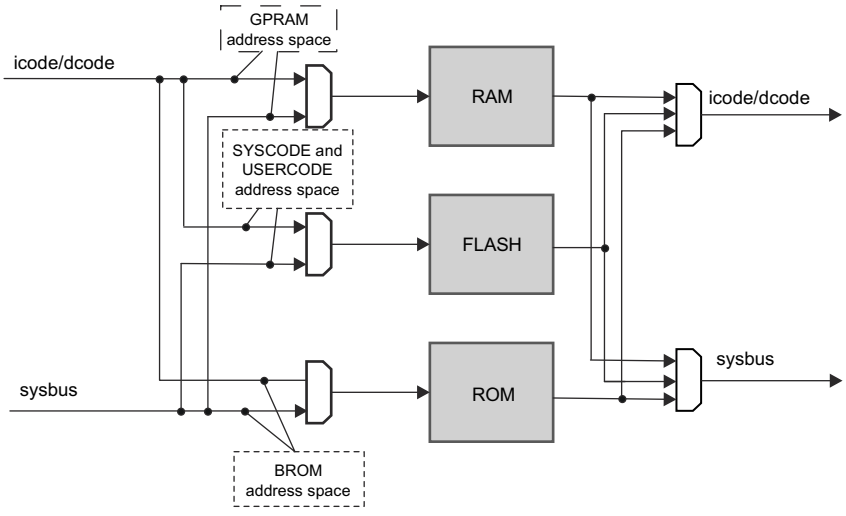


图 8-3. GPRAM 模式下的 VIMS 模块

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.2.1.2 关闭模式

在关闭模式下，RAM 块被禁用，CPU 或系统总线无法访问（请参阅 图 8-4）。在关闭模式下，GPRAM 空间不可用。

闪存块不支持高速缓存，对闪存的所有访问都直接路由到闪存块。

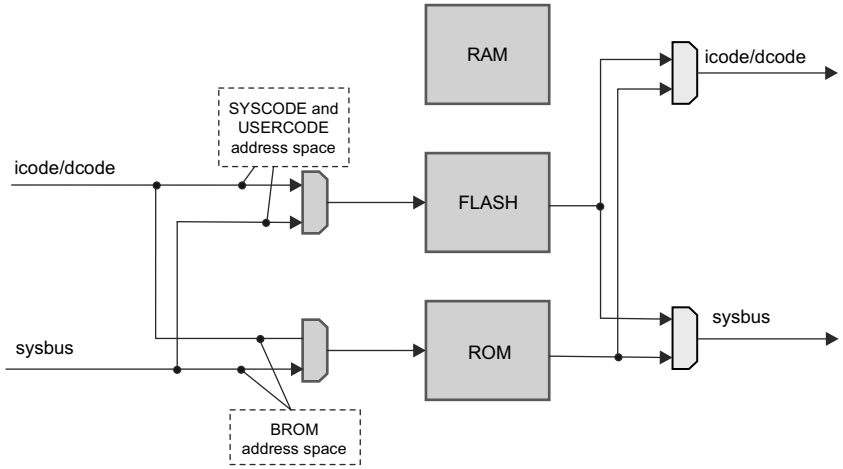


图 8-4. VIMS 模块处于关闭模式

8.2.1.3 缓存模式

在缓存模式下，RAM 块用作闪存块的 8K 4 路随机替换缓存（请参阅图 8-5）。缓存模式下 GPRAM 空间不可用。

缓存支持仅适用于 CPU 对闪存 SYSCODE 地址空间的访问。系统总线对闪存块的访问以及 CPU 对闪存 USERCODE 地址空间的访问会直接路由到闪存块。

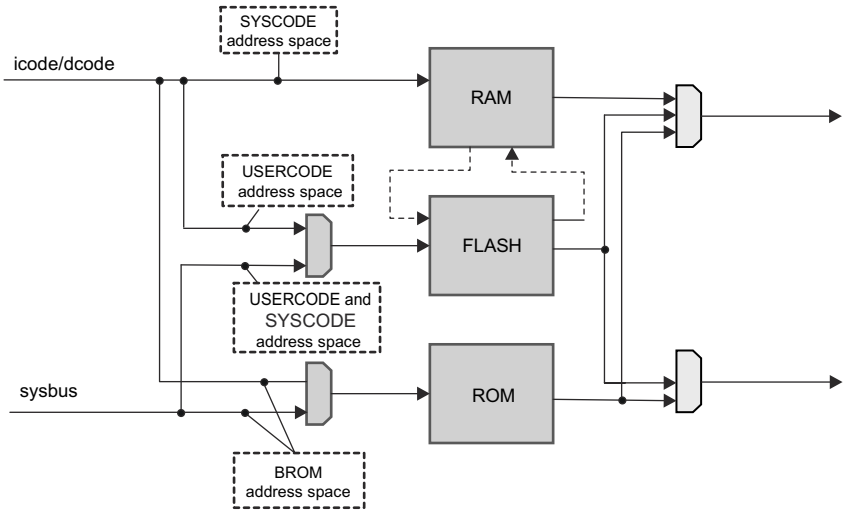


图 8-5. 缓存模式下的 VIMS 模块

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

在缓存模式下，CPU 对闪存 SYSCODE 地址空间的所有访问首先被导向缓存。缓存在内部标签 RAM 中查找输入地址，以确定该访问是缓存命中还是缓存未命中。

如果缓存未命中，访问将被转发到闪存块。来自闪存块的响应路由回缓存，随后缓存得到更新。

如果缓存命中，则直接从缓存 RAM 中获取数据。

由于缓存 RAM 字长为 64 位，缓存还包含一个行缓冲区。行缓冲区的作用是避免重新获取在前一次访问中已获取（但未使用）的 32 位数据部分。如果数据已在行缓冲区中，行缓冲区会阻止进行 TAG 和 CACHE 查找。

作为无效化方案的一部分，将会清除缓存行缓冲区。

8.2.2 VIMS 闪存线路缓冲器

VIMS 模块包含两个闪存线路缓冲器，因为闪存字大小为 128 位。

- 线路缓冲器放置在由 VIMS:CTL.IDCODE_LB_DIS 寄存器控制的闪存 CPU 总线路径中。
- 线路缓冲器放置在由 VIMS:CTL.SYSBUS_LB_DIS 寄存器控制的闪存系统总线路径中。

这些缓冲器的目的是防止重新获取上一周期中已提取（但未使用）的数据的 32 位部分。线路缓冲器的状态可在 VIMS:STATUS.IDCODE_LB_DIS 寄存器和 VIMS:STATUS.SYSBUS_LB_DIS 寄存器中找到。

8.2.3 VIMS 仲裁

VIMS 提供 CPU 和系统总线之间的仲裁。仲裁可通过 VIMS:CTL.ARB_CFG 寄存器在*循环*和*静态*之间进行配置。默认情况下启用静态仲裁，并赋予 CPU 优先级，而非系统总线。

系统仲裁器允许同时进行访问，前提是 CPU 和系统总线具有不同的目标存储器。例如，如果 CPU 访问导致缓存命中，进行系统总线访问时可以同时访问闪存。

8.2.4 VIMS 高速缓存 TAG 预取

缓存包含一个 TAG 预取系统，该系统会自动预取下一个 64 位地址的 TAG 数据。此功能通过 VIMS:CTL.PREF_EN 寄存器控制，只有在 VIMS 模式设置为缓存模式时才启用。任何使用预取 TAG 的访问都会在访问中节省一个 CLK 周期，因为可以跳过标签查找。*预取命中*被定义为使用预取的 TAG 数据和缓存中可用的数据进行的访问。

TAG 预取主要用于 CPU 全速运行时的性能优化。如果 CPU 未全速运行，则不会进行性能优化；因此，应禁用 TAG 预取系统以尽可能降低功耗。

8.3 VIMS 软件注意事项

当 FLASH 被编程或更新时，或当 VIMS 域进入断电状态时，软件方面必须特别注意。

使用内置 ROM 函数和标准 API 函数时，以下注意事项会自动得到处理。但是，自定义代码必须考虑以下注意事项。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.3.1 闪存程序或者更新

当 VIMS 处于缓存模式时，请勿编程或更新闪存。

在对闪存进行编程或更新之前，请完成以下任务：

- 如果当前模式为高速缓存模式，则将 VIMS 设置为关闭模式。
- 禁用两个 VIMS 闪存线路缓冲器。

这些操作可防止在闪存程序或更新后从缓存或行缓冲器中获取旧数据或指令。

闪存编程或更新后，VIMS 可设回高速缓存模式。当 VIMS 设置回缓存模式时，硬件将使缓存无效。

8.3.2 VIMS 保持

如果需要，当域进入断电状态时，可以保留 VIMS 域。保留控件可选择结合 VIMS 逻辑来指定保留哪些存储器（内部 TAG RAM 或高速缓存 RAM）。

表 8-1 指定 VIMS 存储器的有效保留组合。

表 8-1. VIMS 存储器的有效保留组合

模式	已启用保留			注释
	TAG-RAM	CACHE-RAM	VIMS 逻辑	
1	否	否	是	软件必须补偿 RAM 内的数据丢失
2	否	是	是	在 GPRAM 模式下工作，无需软件干预
3	是	是	是	

8.3.2.1 模式 1

模式 1 适用于 VIMS 处于关闭模式或者高速缓存模式时。

如果 VIMS 处于高速缓存模式，请先将 VIMS 模式更改为关闭模式，然后再关闭 VIMS 域。当系统关机后，您可以将 VIMS 模式重新设置为高速缓存模式，这将使高速缓存内存无效（初始化）（请参阅图 8-6）。

当系统处于 GPRAM 模式时，也可使用模式 1，但当 VIMS 域设置为断电状态时，GPRAM 中的所有数据都会丢失。

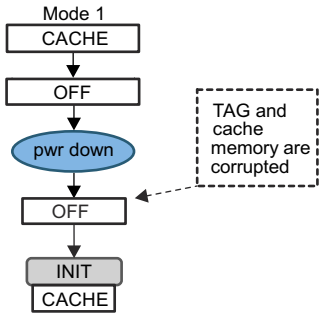


图 8-6. 无 RAM 保留的软件预防措施

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.3.2.2 模式 2

模式 2 适用于缓存处于 GPRAM 模式的系统。VIMS 保留在 GPRAM 的保留电源中。

备注

如果软件在保留后尝试将 VIMS 置于高速缓存模式，则系统会因为 TAG 存储器损坏而失败。

正确的程序是在将 VIMS 置于缓存模式*之前*，将 VIMS 置于关闭模式。如需了解更多详情，请参阅图 8-7。

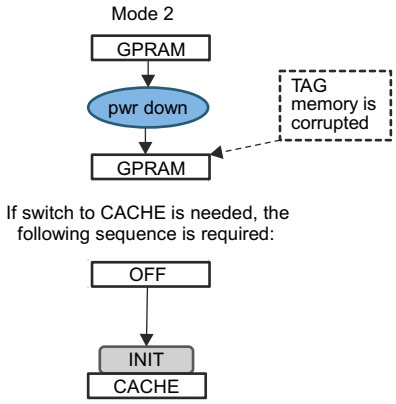


图 8-7. GPRAM 保持

8.3.2.3 模式 3

模式 3 适合在 VIMS 处于高速缓存模式时使用。在模式 3 中，当 VIMS 域断电时，VIMS 可以处于高速缓存模式。

8.4 ROM

ROM 包含一个支持 SPI 和 UART 的串行引导加载程序（请参阅章节 10），以及驱动程序库和射频协议栈支持。相关详细信息，请参阅节 5.6。

8.5 闪存

闪存被组织为一组 8kB 的块，可以单独擦除。可以对一个单独的 64 位字进行编程，将位从 1 更改为 0。擦除一个块会使该块的全部内容均重置为 1。8kB 块与 8kB 块组配对，可通过标记为只读来单独保护这些块。无法对只读块进行擦除或编程，从而保护这些块的内容不被修改。只读锁定位的位置是在 CCFG 中。因此，批量擦除或擦除最后一个闪存页面（使用 CCFG）将禁用只读锁定。

对于擦除之间允许针对闪存行进行多少次写入操作存在限制。一行由 2048 位（或 256 字节）组成。闪存被均匀划分成物理行。在两次擦除之间的一行中最多可以进行 83 次写入操作。如果在重新擦除之前执行了 83 个以上的写入操作，则可能会看到被擦除的行中未写入的位（处于逻辑 1 状态）被编程（更改为逻辑 0 状态）。用户软件必须注意此限制，没有硬件检查并通知是否违反此限制。

闪存块主要由 48MHz 系统时钟计时。

8.5.1 闪存保护

通过配置 CCFG，可在 8kB 扇区中对闪存进行读/写保护。

8.5.2 存储器编程

使用 TI 提供的 API 来完成存储器编程。调用 API 函数时，应禁用所有中断，以防止在执行这些函数期间任何读取闪存的尝试。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 8-2. CC13x2 和 CC26x2 存储器写入/擦除保护 (1) (2) (3)

存储器区域 CC13x2 和 CC26x2 状态	FCFG0（电子熔丝）	FCFG1 (ENGR)	CCFG	TI 锁定扇区	客户已锁定	客户未锁定
非紧凑裸片	写入 1s（无法返回）	未锁定	未锁定	无	无	全部
封装裸片	已锁定	未锁定	未锁定	无	无	全部
工程样片	已锁定	未锁定	未锁定	无	无	全部
客户开发	已锁定	已锁定	未锁定	固定	无	TI 锁定扇区之外
客户交货案例 1	已锁定	已锁定	可写（不可擦除）(4)	固定	可以添加锁定扇区 (4)	可能会降低(4)
客户交货案例 2	已锁定	已锁定	已锁定(4)	固定	固定(4)	固定(4)

(1) 已锁定：不可写入和不可擦除。
(2) 未锁定：可写和可擦除。
(3) 固定：此类型的数量是固定的。
(4) 芯片擦除函数会擦除 TI 未锁定的所有扇区。

8.5.3 闪存编程

在闪存写入或擦除操作期间，不得读取闪存。如果在闪存操作期间需要执行指令，则在闪存操作进行时，执行代码必须放置在 SRAM 中（并从 SRAM 执行）。

8.5.4 电源管理要求

该模块实现了以下节能功能（请参阅图 8-8）：

- **电压关闭：**模块逻辑 VDD 已关闭。泵和气缸组保持深度睡眠状态。此模式需要复位和软件配置才能激活。
- **电源关闭：**这与“关断电压”的状态相同，唯一的区别是模块逻辑在所有寄存器上都保留。在断电模式下，该模块无需任何软件配置即可变为活动状态。
- **深度待机：**内部电路部分断电。激活不需要内部配置，但由于电压斜升等原因会有一些延迟。
- **空闲读数：**在不进行活动读数时，使用泵和存储体中的高级功率降低功能来节省功耗。在此模式下，切换到活动读取时不会降低读取延迟。
- **阅读：**闪存正在主动读取，而不会发生任何功耗降低。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

更改功耗模式的方法：

- 只有通过系统电源管理才能达到电压关闭或断电。电压关闭类似于初始上电。断电需要恢复保持状态、内部序列发生器必须上电并且对组和电荷泵进行配置。
- 退出深度待机可以通过以下方式开始：
  - PRCM
  - 写入 MMR 中的寄存器
  - 启动对闪存的读取访问
- 当读取结束时，会自动在空闲读取和读取之间切换。可通过寄存器设置禁用开关。
- 通过为目标功耗模式设置一个寄存器，可从空闲读数切换到任何其他模式。一段时间后没有读取访问权限，该模块进入或准备所选模式。实现关闭电源或电压的最后一步由系统电源管理完成。

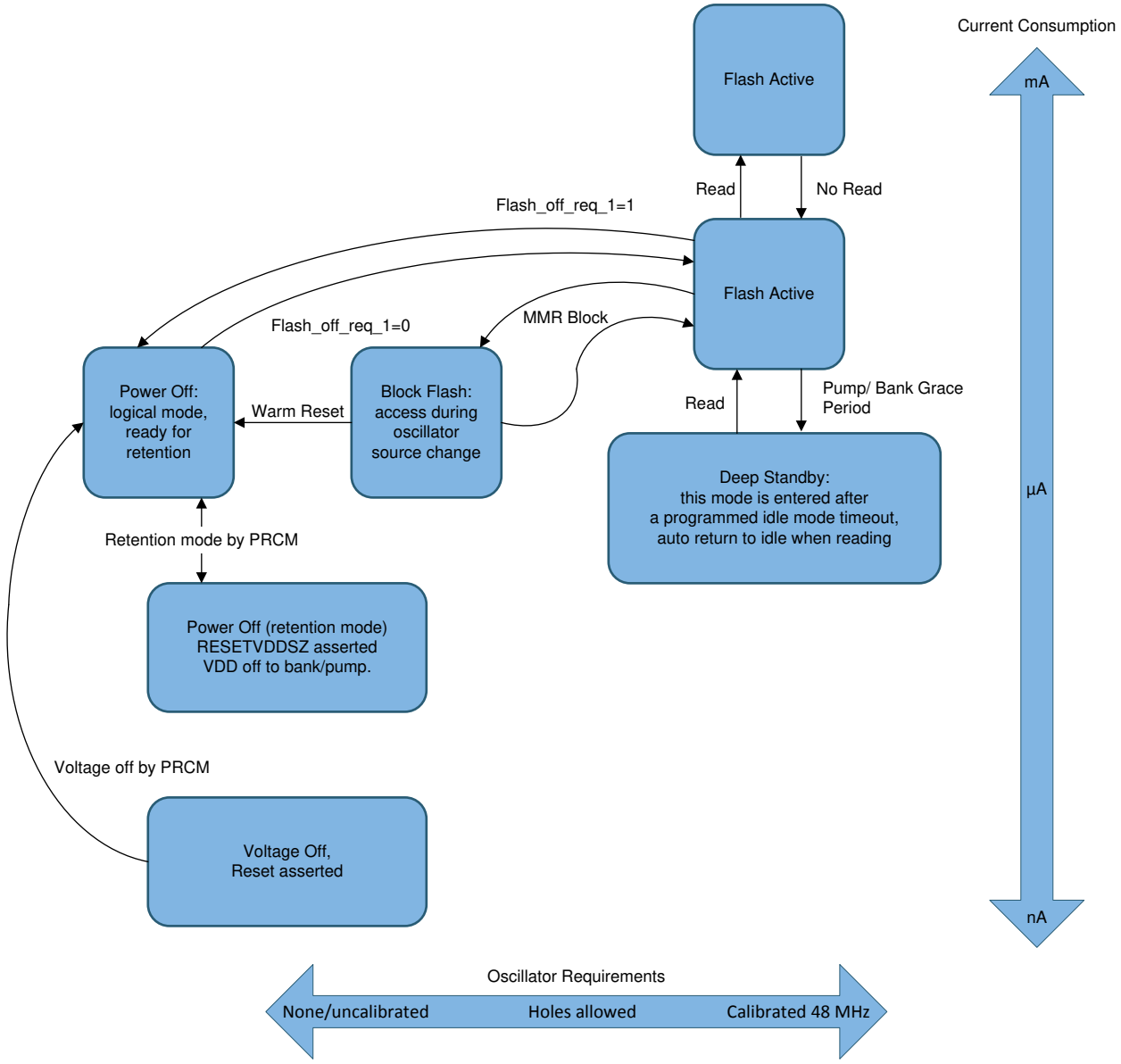


图 8-8. 闪存电源状态

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.6 ROM 功能

存储器内容概述：

- 电子保险丝
  - 包含引导加载程序启动前所需的大部分关键芯片修整项
  - 通过数字内核中的闪存模块进行接口连接
    - FLASH 修整
    - RAM 修复
    - 模拟修整（带隙、欠压、所选稳压器、内部 48MHz RC 振荡器）
    - JTAG TAP/DAP 锁定
    - CRC 校验（8 位）
  - 此处唯一的关键项是默认锁定（如果保险丝熔断）的 JTAG TAP/DAP 锁。
- FCFG：
  - 当前为一个单独的闪存块
  - 所有修整项以及整个器件配置
    - 支持擦除/写入的闪存修整
    - 模块修整（模拟、RF+++）
    - 芯片配置（ID、器件类型、封装尺寸、引脚排列++、量产测试数据）
    - 引导加载程序配置
    - 安全性
      - TI FA 分析选项
      - JTAG TAP/DAP 锁定覆盖
      - 启用引导加载程序
- 客户配置（闪存中的最后一页）：
  - 禁用引导加载程序
  - 禁用 JTAG DAP/TAP
  - 禁用 TI FA 分析
  - 客户配置区域写入或擦除保护
  - 其他与安全无关的配置

配置存储器：

- RO
  - OTP、1KB 读取接口（通过 FMC 写入）
- RO
  - ENGR 1KB 读取接口（通过 FMC 写入）
- CCFG
  - 闪存扇区 4KB 读取接口（通过 FMC 写入）
- RO
  - EFUSE，仅可通过 MMR 接口访问

ROM 预编程了一个串行引导加载程序（SPI 或 UART）。对于需要现场可编程性的应用程序，免专利费的引导加载程序可作为应用程序加载程序，支持现场固件更新。如果未将有效映像写入闪存，引导加载程序将自动执行；也可以通过可配置的 GPIO 后门启动引导加载程序。引导加载程序可能无法从应用程序代码中调用。

8.7 VIMS 寄存器

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1 FLASH 寄存器

表 8-3 列出了 FLASH 寄存器的存储器映射寄存器。表 8-3 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

表 8-3. FLASH 寄存器

偏移	首字母缩写词	寄存器名称	部分
1Ch	STAT	FMC 和电子保险丝状态	STAT 寄存器 (偏移 = 1Ch) [复位 = 00000000h]
24h	CFG	内部	CFG 寄存器 (偏移 = 24h) [复位 = 00000000h]
28h	SYSCODE_START	内部	SYSCODE_START 寄存器 (偏移 = 28h) [复位 = 00000000h]
2Ch	FLASH_SIZE	内部	FLASH_SIZE 寄存器 (偏移 = 2Ch) [复位 = 00000000h]
3Ch	FWLOCK	内部	FWLOCK 寄存器 (偏移 = 3Ch) [复位 = 00000000h]
40h	FWFLAG	内部	FWFLAG 寄存器 (偏移 = 40h) [复位 = 00000000h]
1000h	EFUSE	内部	EFUSE 寄存器 (偏移 = 1000h) [复位 = 00000000h]
1004h	EFUSEADDR	内部	EFUSEADDR 寄存器 (偏移 = 1004h) [复位 = 00000000h]
1008h	DATAUPPER	内部	DATAUPPER 寄存器 (偏移 = 1008h) [复位 = 00000000h]
100Ch	DATALOWER	内部	DATALOWER 寄存器 (偏移 = 100Ch) [复位 = 00000000h]
1010h	EFUSECFG	内部	EFUSECFG 寄存器 (偏移 = 1010h) [复位 = 00000001h]
1014h	EFUSESTAT	内部	EFUSESTAT 寄存器 (偏移 = 1014h) [复位 = 00000001h]
1018h	ACC	内部	ACC 寄存器 (偏移 = 1018h) [复位 = 00000000h]
101Ch	BOUNDARY	内部	BOUNDARY 寄存器 (偏移 = 101Ch) [复位 = 00000000h]
1020h	EFUSEFLAG	内部	EFUSEFLAG 寄存器 (偏移 = 1020h) [复位 = 00000000h]
1024h	EFUSEKEY	内部	EFUSEKEY 寄存器 (偏移 = 1024h) [复位 = 00000000h]
1028h	EFUSERELEASE	内部	EFUSERELEASE 寄存器 (偏移 = 1028h) [复位 = 00000000h]
102Ch	EFUSEPINS	内部	EFUSEPINS 寄存器 (偏移 = 102Ch) [复位 = 00000000h]
1030h	EFUSECRA	内部	EFUSECRA 寄存器 (偏移 = 1030h) [复位 = 00000000h]
1034h	EFUSEREAD	内部	EFUSEREAD 寄存器 (偏移 = 1034h) [复位 = 00000000h]
1038h	EFUSEPROGRAM	内部	EFUSEPROGRAM 寄存器 (偏移 = 1038h) [复位 = 00000000h]
103Ch	EFUSEERROR	内部	EFUSEERROR 寄存器 (偏移 = 103Ch) [复位 = 00000000h]
1040h	SINGLEBIT	内部	SINGLEBIT 寄存器 (偏移 = 1040h) [复位 = 00000000h]

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 8-3. FLASH 寄存器（续）

偏移	首字母缩写词	寄存器名称	部分
1044h	TWOBIT	内部	TWOBIT 寄存器（偏移 = 1044h）[复位 = 00000000h]
1048h	SELFTESTCYC	内部	SELFTESTCYC 寄存器（偏移 = 1048h）[复位 = 00000000h]
104Ch	SELFTESTSIGN	内部	SELFTESTSIGN 寄存器（偏移 = 104Ch）[复位 = 00000000h]
2000h	FRDCTL	内部	FRDCTL 寄存器（偏移 = 2000h）[复位 = 00000200h]
2004h	FSPRD	内部	FSPRD 寄存器（偏移 = 2004h）[复位 = 00000000h]
2008h	FEDACCTL1	内部	FEDACCTL1 寄存器（偏移 = 2008h）[复位 = 00000000h]
201Ch	FEDACSTAT	内部	FEDACSTAT 寄存器（偏移 = 201Ch）[复位 = 00000000h]
2030h	FBPROT	内部	FBPROT 寄存器（偏移 = 2030h）[复位 = 00000000h]
2034h	FBSE	内部	FBSE 寄存器（偏移 = 2034h）[复位 = 00000000h]
2038h	FBBUSY	内部	FBBUSY 寄存器（偏移 = 2038h）[复位 = 000000FEh]
203Ch	FBAC	内部	FBAC 寄存器（偏移 = 203Ch）[复位 = 0000000Fh]
2040h	FBFALLBACK	内部	FBFALLBACK 寄存器（偏移 = 2040h）[复位 = 0505FFFFh]
2044h	FBPRDY	内部	FBPRDY 寄存器（偏移 = 2044h）[复位 = 00FF00FEh]
2048h	FPAC1	内部	FPAC1 寄存器（偏移 = 2048h）[复位 = 02082081h]
204Ch	FPAC2	内部	FPAC2 寄存器（偏移 = 204Ch）[复位 = 00000000h]
2050h	FMAC	内部	FMAC 寄存器（偏移 = 2050h）[复位 = 00000000h]
2054h	FMSTAT	内部	FMSTAT 寄存器（偏移 = 2054h）[复位 = 00000000h]
2064h	FLOCK	内部	FLOCK 寄存器（偏移 = 2064h）[复位 = 000055AAh]
2080h	FVREADCT	内部	FVREADCT 寄存器（偏移 = 2080h）[复位 = 00000008h]
2084h	FVHVCT1	内部	FVHVCT1 寄存器（偏移 = 2084h）[复位 = 00840088h]
2088h	FVHVCT2	内部	FVHVCT2 寄存器（偏移 = 2088h）[复位 = 00A20000h]
208Ch	FVHVCT3	内部	FVHVCT3 寄存器（偏移 = 208Ch）[复位 = 000F0000h]
2090h	FVNVCT	内部	FVNVCT 寄存器（偏移 = 2090h）[复位 = 00000800h]
2094h	FVSLP	内部	FVSLP 寄存器（偏移 = 2094h）[复位 = 00008000h]
2098h	FVWLCT	内部	FVWLCT 寄存器（偏移 = 2098h）[复位 = 00000008h]

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 8-3. FLASH 寄存器（续）

偏移	首字母缩写词	寄存器名称	部分
209Ch	FEFUSECTL	内部	FEFUSECTL 寄存器（偏移 = 209Ch）[复位 = 0701010Ah]
20A0h	FEFUSESTAT	内部	FEFUSESTAT 寄存器（偏移 = 20A0h）[复位 = 00000000h]
20A4h	FEFUSEDATA	内部	FEFUSEDATA 寄存器（偏移 = 20A4h）[复位 = 00000000h]
20A8h	FSEQPMP	内部	FSEQPMP 寄存器（偏移 = 20A8h）[复位 = 85080000h]
2100h	FBSTROBES	内部	FBSTROBES 寄存器（偏移 = 2100h）[复位 = 00000104h]
2104h	FPSTROBES	内部	FPSTROBES 寄存器（偏移 = 2104h）[复位 = 00000103h]
2108h	FBMODE	内部	FBMODE 寄存器（偏移 = 2108h）[复位 = 00000000h]
210Ch	FTCR	内部	FTCR 寄存器（偏移 = 210Ch）[复位 = 00000000h]
2110h	FADDR	内部	FADDR 寄存器（偏移 = 2110h）[复位 = 00000000h]
211Ch	FTCTL	内部	FTCTL 寄存器（偏移 = 211Ch）[复位 = 00000000h]
2120h	FWPWRITE0	内部	FWPWRITE0 寄存器（偏移 = 2120h）[复位 = FFFFFFFFh]
2124h	FWPWRITE1	内部	FWPWRITE1 寄存器（偏移 = 2124h）[复位 = FFFFFFFFh]
2128h	FWPWRITE2	内部	FWPWRITE2 寄存器（偏移 = 2128h）[复位 = FFFFFFFFh]
212Ch	FWPWRITE3	内部	FWPWRITE3 寄存器（偏移 = 212Ch）[复位 = FFFFFFFFh]
2130h	FWPWRITE4	内部	FWPWRITE4 寄存器（偏移 = 2130h）[复位 = FFFFFFFFh]
2134h	FWPWRITE5	内部	FWPWRITE5 寄存器（偏移 = 2134h）[复位 = FFFFFFFFh]
2138h	FWPWRITE6	内部	FWPWRITE6 寄存器（偏移 = 2138h）[复位 = FFFFFFFFh]
213Ch	FWPWRITE7	内部	FWPWRITE7 寄存器（偏移 = 213Ch）[复位 = FFFFFFFFh]
2140h	FWPWRITE_ECC	内部	FWPWRITE_ECC 寄存器（偏移 = 2140h）[复位 = FFFFFFFFh]
2144h	FSWSTAT	内部	FSWSTAT 寄存器（偏移 = 2144h）[复位 = 00000001h]
2200h	FSM_GLBCTL	内部	FSM_GLBCTL 寄存器（偏移 = 2200h）[复位 = 00000001h]
2204h	FSM_STATE	内部	FSM_STATE 寄存器（偏移 = 2204h）[复位 = 00000C00h]
2208h	FSM_STAT	内部	FSM_STAT 寄存器（偏移 = 2208h）[复位 = 00000004h]
220Ch	FSM_CMD	内部	FSM_CMD 寄存器（偏移 = 220Ch）[复位 = 00000000h]
2210h	FSM_PE_OSU	内部	FSM_PE_OSU 寄存器（偏移 = 2210h）[复位 = 00000000h]

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 8-3. FLASH 寄存器（续）

偏移	首字母缩写词	寄存器名称	部分
2214h	FSM_VSTAT	内部	FSM_VSTAT 寄存器（偏移 = 2214h）[复位 = 00003000h]
2218h	FSM_PE_VSU	内部	FSM_PE_VSU 寄存器（偏移 = 2218h）[复位 = 00000000h]
221Ch	FSM_CMP_VSU	内部	FSM_CMP_VSU 寄存器（偏移 = 221Ch）[复位 = 00000000h]
2220h	FSM_EX_VAL	内部	FSM_EX_VAL 寄存器（偏移 = 2220h）[复位 = 00000301h]
2224h	FSM_RD_H	内部	FSM_RD_H 寄存器（偏移 = 2224h）[复位 = 0000005Ah]
2228h	FSM_P_OH	内部	FSM_P_OH 寄存器（偏移 = 2228h）[复位 = 00000100h]
222Ch	FSM_ERA_OH	内部	FSM_ERA_OH 寄存器（偏移 = 222Ch）[复位 = 00000001h]
2230h	FSM_SAV_PPUL	内部	FSM_SAV_PPUL 寄存器（偏移 = 2230h）[复位 = 00000000h]
2234h	FSM_PE_VH	内部	FSM_PE_VH 寄存器（偏移 = 2234h）[复位 = 00000100h]
2240h	FSM_PRG_PW	内部	FSM_PRG_PW 寄存器（偏移 = 2240h）[复位 = 00000000h]
2244h	FSM_ERA_PW	内部	FSM_ERA_PW 寄存器（偏移 = 2244h）[复位 = 00000000h]
2254h	FSM_SAV_ERA_PUL	内部	FSM_SAV_ERA_PUL 寄存器（偏移 = 2254h）[复位 = 00000000h]
2258h	FSM_TIMER	内部	FSM_TIMER 寄存器（偏移 = 2258h）[复位 = 00000000h]
225Ch	FSM_MODE	内部	FSM_MODE 寄存器（偏移 = 225Ch）[复位 = 00000000h]
2260h	FSM_PGM	内部	FSM_PGM 寄存器（偏移 = 2260h）[复位 = 00000000h]
2264h	FSM_ERA	内部	FSM_ERA 寄存器（偏移 = 2264h）[复位 = 00000000h]
2268h	FSM_PRG_PUL	内部	FSM_PRG_PUL 寄存器（偏移 = 2268h）[复位 = 00040032h]
226Ch	FSM_ERA_PUL	内部	FSM_ERA_PUL 寄存器（偏移 = 226Ch）[复位 = 00040BB8h]
2270h	FSM_STEP_SIZE	内部	FSM_STEP_SIZE 寄存器（偏移 = 2270h）[复位 = 00000000h]
2274h	FSM_PUL_CNTR	内部	FSM_PUL_CNTR 寄存器（偏移 = 2274h）[复位 = 00000000h]
2278h	FSM_EC_STEP_HEIGHT	内部	FSM_EC_STEP_HEIGHT 寄存器（偏移 = 2278h）[复位 = 00000000h]
227Ch	FSM_ST_MACHINE	内部	FSM_ST_MACHINE 寄存器（偏移 = 227Ch）[复位 = 00800500h]
2280h	FSM_FLES	内部	FSM_FLES 寄存器（偏移 = 2280h）[复位 = 00000000h]
2288h	FSM_WR_ENA	内部	FSM_WR_ENA 寄存器（偏移 = 2288h）[复位 = 00000002h]
228Ch	FSM_ACC_PP	内部	FSM_ACC_PP 寄存器（偏移 = 228Ch）[复位 = 00000000h]

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 8-3. FLASH 寄存器（续）

偏移	首字母缩写词	寄存器名称	部分
2290h	FSM_ACC_EP	内部	FSM_ACC_EP 寄存器（偏移 = 2290h）[复位 = 00000000h]
22A0h	FSM_ADDR	内部	FSM_ADDR 寄存器（偏移 = 22A0h）[复位 = 00000000h]
22A4h	FSM_SECTOR	内部	FSM_SECTOR 寄存器（偏移 = 22A4h）[复位 = FFFF0000h]
22A8h	FMC_REV_ID	内部	FMC_REV_ID 寄存器（偏移 = 22A8h）[复位 = 00000000h]
22ACh	FSM_ERR_ADDR	内部	FSM_ERR_ADDR 寄存器（偏移 = 22ACh）[复位 = 00000000h]
22B0h	FSM_PGM_MAXPUL	内部	FSM_PGM_MAXPUL 寄存器（偏移 = 22B0h）[复位 = 00000000h]
22B4h	FSM_EXECUTE	内部	FSM_EXECUTE 寄存器（偏移 = 22B4h）[复位 = 000A000Ah]
22C0h	FSM_SECTOR1	内部	FSM_SECTOR1 寄存器（偏移 = 22C0h）[复位 = FFFFFFFFh]
22C4h	FSM_SECTOR2	内部	FSM_SECTOR2 寄存器（偏移 = 22C4h）[复位 = 00000FFFh]
22E0h	FSM_BSLE0	内部	FSM_BSLE0 寄存器（偏移 = 22E0h）[复位 = 00000000h]
22E4h	FSM_BSLE1	内部	FSM_BSLE1 寄存器（偏移 = 22E4h）[复位 = 00000000h]
22F0h	FSM_BSLP0	内部	FSM_BSLP0 寄存器（偏移 = 22F0h）[复位 = 00000000h]
22F4h	FSM_BSLP1	内部	FSM_BSLP1 寄存器（偏移 = 22F4h）[复位 = 00000000h]
22F8h	FSM_PGM128	FMC FSM 使能 128 位宽编程	FSM_PGM128 寄存器（偏移 = 22F8h）[复位 = 00000000h]
2400h	FCFG_BANK	内部	FCFG_BANK 寄存器（偏移 = 2400h）[复位 = 00000801h]
2404h	FCFG_WRAPPER	内部	FCFG_WRAPPER 寄存器（偏移 = 2404h）[复位 = 50009007h]
2408h	FCFG_BNK_TYPE	内部	FCFG_BNK_TYPE 寄存器（偏移 = 2408h）[复位 = 00000004h]
2410h	FCFG_B0_START	内部	FCFG_B0_START 寄存器（偏移 = 2410h）[复位 = 02000000h]
2414h	FCFG_B1_START	内部	FCFG_B1_START 寄存器（偏移 = 2414h）[复位 = 00000000h]
2418h	FCFG_B2_START	内部	FCFG_B2_START 寄存器（偏移 = 2418h）[复位 = 00000000h]
241Ch	FCFG_B3_START	内部	FCFG_B3_START 寄存器（偏移 = 241Ch）[复位 = 00000000h]
2420h	FCFG_B4_START	内部	FCFG_B4_START 寄存器（偏移 = 2420h）[复位 = 00000000h]
2424h	FCFG_B5_START	内部	FCFG_B5_START 寄存器（偏移 = 2424h）[复位 = 00000000h]
2428h	FCFG_B6_START	内部	FCFG_B6_START 寄存器（偏移 = 2428h）[复位 = 00000000h]
242Ch	FCFG_B7_START	内部	FCFG_B7_START 寄存器（偏移 = 242Ch）[复位 = 00000000h]

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 8-3. FLASH 寄存器（续）

偏移	首字母缩写词	寄存器名称	部分
2430h	FCFG_B0_SSIZE0	内部	FCFG_B0_SSIZE0 寄存器（偏移 = 2430h）[复位 = 002C0008h]

复杂的位访问类型经过编码可适应小型表单元。表 8-4 展示了适用于此部分中访问类型的代码。

表 8-4. FLASH 访问类型代码

访问类型	代码	说明
读取类型		
R	R	读取
写入类型		
W	W	写入
W1C	W 1C	写入 1 以清零
复位或默认值		
-n		复位后的值或默认值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.1 STAT 寄存器（偏移 = 1Ch）[复位 = 00000000h]

图 8-9 展示了 STAT，表 8-5 中对此进行了介绍。

返回到汇总表。

FMC 和电子保险丝状态

图 8-9. STAT 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
EFUSE_BLANK	EFUSE_TIMEOUT	SPRS_BYTE_NOT_OK	EFUSE_ERRCODE				
R-0h	R-0h	R-0h	R-0h				
7	6	5	4	3	2	1	0
RESERVED					SAMHOLD_DIS	BUSY	POWER_MODE
R-0h					R-0h	R-0h	R-0h

表 8-5. STAT 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15	EFUSE_BLANK	R	0h	电子保险丝扫描检测保险丝 ROM 是否空白： 0：非空白 1：空白
14	EFUSE_TIMEOUT	R	0h	电子保险丝扫描导致超时错误。 0：无超时错误 1：有超时错误
13	SPRS_BYTE_NOT_OK	R	0h	电子保险丝扫描导致扫描链稀疏字节错误。 0：无稀疏错误 1：有稀疏错误
12-8	EFUSE_ERRCODE	R	0h	与 EFUSEERROR.CODE 相同
7-3	RESERVED	R	0h	保留
2	SAMHOLD_DIS	R	0h	闪存采样保持时序逻辑的状态指示器。在 CFG.DIS_IDLE 设置为 1 后，该位会在延迟一段时间后变为 1。 0：未禁用 1：采样保持已禁用且稳定
1	BUSY	R	0h	FMC FMSTAT.BUSY 位的快速版本。 该标志在操作设置后立即有效（FMSTAT.BUSY 延迟一定周期） 0：非繁忙 1：繁忙
0	POWER_MODE	R	0h	闪存子系统的电源状态。 0：活动 1：低功耗

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.2 CFG 寄存器（偏移 = 24h）[复位 = 00000000h]

图 8-10 展示了 CFG，表 8-6 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-10. CFG 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							STANDBY_MODE_SEL
R-0h							R/W-0h
7	6	5	4	3	2	1	0
STANDBY_PW_SEL		DIS_EFUSECLK	DIS_READACCESS	ENABLE_SWINTF	RESERVED	DIS_STANDBY	DIS_IDLE
R/W-0h		R/W-0h	R/W-0h	R/W-0h	R-0h	R/W-0h	R/W-0h

表 8-6. CFG 寄存器字段说明

位	字段	类型	复位	说明
31-9	RESERVED	R	0h	保留
8	STANDBY_MODE_SEL	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
7-6	STANDBY_PW_SEL	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
5	DIS_EFUSECLK	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
4	DIS_READACCESS	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
3	ENABLE_SWINTF	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
2	RESERVED	R	0h	保留
1	DIS_STANDBY	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
0	DIS_IDLE	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.3 SYSCODE_START 寄存器（偏移 = 28h）[复位 = 00000000h]

图 8-11 展示了 SYSCODE_START，表 8-7 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-11. SYSCODE_START 寄存器

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										SYSCODE_START					
R-0h										R/W-0h					

表 8-7. SYSCODE_START 寄存器字段说明

位	字段	类型	复位	说明
31-6	RESERVED	R	0h	保留
5-0	SYSCODE_START	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.4 FLASH_SIZE 寄存器（偏移 = 2Ch）[复位 = 00000000h]

图 8-12 展示了 FLASH_SIZE，表 8-8 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-12. FLASH_SIZE 寄存器

31	30	29	28	27	26	25	24	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																								SECTORS							
R-0h																								R/W-0h							

表 8-8. FLASH_SIZE 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-0	SECTORS	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.5 FWLOCK 寄存器（偏移 = 3Ch）[复位 = 00000000h]

图 8-13 展示了 FWLOCK，表 8-9 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-13. FWLOCK 寄存器

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													FWLOCK		
R-0h													R/W-0h		

表 8-9. FWLOCK 寄存器字段说明

位	字段	类型	复位	说明
31-3	RESERVED	R	0h	保留
2-0	FWLOCK	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.6 FWFLAG 寄存器（偏移 = 40h）[复位 = 00000000h]

图 8-14 展示了 FWFLAG，表 8-10 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-14. FWFLAG 寄存器

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													FWFLAG		
R-0h													R/W-0h		

表 8-10. FWFLAG 寄存器字段说明

位	字段	类型	复位	说明
31-3	RESERVED	R	0h	保留
2-0	FWFLAG	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.7 EFUSE 寄存器（偏移 = 1000h）[复位 = 00000000h]

图 8-15 展示了 EFUSE，表 8-11 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-15. EFUSE 寄存器

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

表 8-11. EFUSE 寄存器字段说明

位	字段	类型	复位	说明
31-29	RESERVED	R	0h	保留
28-24	指令	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
23-16	RESERVED	R	0h	保留
15-0	DUMPWORD	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.8 EFUSEADDR 寄存器（偏移 = 1004h）[复位 = 00000000h]

图 8-16 展示了 EFUSEADDR，表 8-12 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-16. EFUSEADDR 寄存器

31	30	29	28	27	26	25	24	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																块					行										
R-0h																R/W-0h					R/W-0h									

表 8-12. EFUSEADDR 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-11	块	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
10-0	行	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.9 DATAUPPER 寄存器（偏移 = 1008h）[复位 = 00000000h]

图 8-17 展示了 DATAUPPER，表 8-13 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-17. DATAUPPER 寄存器

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								SPARE					P	R	EEN
R-0h								R/W-0h					R/W-0h	R/W-0h	R/W-0h

表 8-13. DATAUPPER 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-3	SPARE	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
2	P	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
1	R	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
0	EEN	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.10 DATALOWER 寄存器（偏移 = 100Ch）[复位 = 00000000h]

图 8-18 展示了 DATALOWER，表 8-14 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-18. DATALOWER 寄存器

31	30	29	28	27	26	25	24	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																															

表 8-14. DATALOWER 寄存器字段说明

位	字段	类型	复位	说明
31-0	DATA	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.11 EFUSECFG 寄存器（偏移 = 1010h）[复位 = 00000001h]

图 8-19 展示了 EFUSECFG，表 8-15 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-19. EFUSECFG 寄存器

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

表 8-15. EFUSECFG 寄存器字段说明

位	字段	类型	复位	说明
31-9	RESERVED	R	0h	保留
8	IDLEGATING	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
7-5	RESERVED	R	0h	保留
4-3	SLAVEPOWER	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
2-1	RESERVED	R	0h	保留
0	GATING	R/W	1h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.12 EFUSESTAT 寄存器（偏移 = 1014h）[复位 = 00000001h]

图 8-20 展示了 EFUSESTAT，表 8-16 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-20. EFUSESTAT 寄存器

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

表 8-16. EFUSESTAT 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	RESETDONE	R	1h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.13 ACC 寄存器（偏移 = 1018h）[复位 = 00000000h]

图 8-21 展示了 ACC，表 8-17 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-21. ACC 寄存器

31	30	29	28	27	26	25	24	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								ACCUMULATOR																							
R-0h								R-0h																							

表 8-17. ACC 寄存器字段说明

位	字段	类型	复位	说明
31-24	RESERVED	R	0h	保留
23-0	ACCUMULATOR	R	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.14 BOUNDARY 寄存器（偏移 = 101Ch）[复位 = 00000000h]

图 8-22 展示了 BOUNDARY，表 8-18 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-22. BOUNDARY 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
DISROW0	SPARE	EFC_SELF_TEST_ERROR	EFC_INSTRUCTION_INFO	EFC_INSTRUCTION_ERROR	EFC_AUTOLOAD_ERROR	OUTPUTENABLE	
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
OUTPUTENABLE		SYS_ECC_SELF_TEST_EN	SYS_ECC_OVERRIDE_EN	EFC_FDI	SYS_DIEID_AUTOLOAD_EN	SYS_REPAIR_EN	
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
SYS_WS_READ_STATES				INPUTENABLE			
R/W-0h				R/W-0h			

表 8-18. BOUNDARY 寄存器字段说明

位	字段	类型	复位	说明
31-24	RESERVED	R	0h	保留
23	DISROW0	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
22	SPARE	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
21	EFC_SELF_TEST_ERROR	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
20	EFC_INSTRUCTION_INFO	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
19	EFC_INSTRUCTION_ERROR	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
18	EFC_AUTOLOAD_ERROR	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
17-14	OUTPUTENABLE	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
13	SYS_ECC_SELF_TEST_EN	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
12	SYS_ECC_OVERRIDE_EN	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
11	EFC_FDI	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
10	SYS_DIEID_AUTOLOAD_EN	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
9-8	SYS_REPAIR_EN	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
7-4	SYS_WS_READ_STATES	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
3-0	INPUTENABLE	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.15 EFUSEFLAG 寄存器（偏移 = 1020h）[复位 = 00000000h]

图 8-23 展示了 EFUSEFLAG，表 8-19 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-23. EFUSEFLAG 寄存器

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															KEY
R-0h															R-0h

表 8-19. EFUSEFLAG 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	KEY	R	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.16 EFUSEKEY 寄存器（偏移 = 1024h）[复位 = 00000000h]

图 8-24 展示了 EFUSEKEY，表 8-20 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-24. EFUSEKEY 寄存器

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

表 8-20. EFUSEKEY 寄存器字段说明

位	字段	类型	复位	说明
31-0	代码	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.17 EFUSERELEASE 寄存器（偏移 = 1028h）[复位 = 00000000h]

图 8-25 展示了 EFUSERELEASE，表 8-21 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-25. EFUSERELEASE 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ODPYEAR							ODPMONTH				ODPDAY				
R-X							R-X				R-X				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EFUSEYEAR							EFUSEMONTH				EFUSEDAY				
R-X							R-X				R-X				

表 8-21. EFUSERELEASE 寄存器字段说明

位	字段	类型	复位	说明
31-25	ODPYEAR	R	X	内部。仅可通过 TI 提供的 API 使用。
24-21	ODPMONTH	R	X	内部。仅可通过 TI 提供的 API 使用。
20-16	ODPDAY	R	X	内部。仅可通过 TI 提供的 API 使用。
15-9	EFUSEYEAR	R	X	内部。仅可通过 TI 提供的 API 使用。
8-5	EFUSEMONTH	R	X	内部。仅可通过 TI 提供的 API 使用。
4-0	EFUSEDAY	R	X	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.18 EFUSEPINS 寄存器（偏移 = 102Ch）[复位 = 00000000h]

图 8-26 展示了 EFUSEPINS，表 8-22 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-26. EFUSEPINS 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
EFC_SELF_TEST_DONE	EFC_SELF_TEST_ERROR	SYS_ECC_SELF_TEST_EN	EFC_INSTRUCTION_INFO	EFC_INSTRUCTION_ERROR	EFC_AUTOLOAD_ERROR	SYS_ECC_OVERRIDE_EN	EFC_READY
R-X	R-X	R-X	R-X	R-X	R-X	R-X	R-X
7	6	5	4	3	2	1	0
EFC_FCLRZ	SYS_DIEID_AUTOLOAD_EN	SYS_REPAIR_EN		SYS_WS_READ_STATES			
R-X	R-X	R-X		R-X			

表 8-22. EFUSEPINS 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	内部。仅可通过 TI 提供的 API 使用。
15	EFC_SELF_TEST_DONE	R	X	内部。仅可通过 TI 提供的 API 使用。
14	EFC_SELF_TEST_ERROR	R	X	内部。仅可通过 TI 提供的 API 使用。
13	SYS_ECC_SELF_TEST_EN	R	X	内部。仅可通过 TI 提供的 API 使用。
12	EFC_INSTRUCTION_INFO	R	X	内部。仅可通过 TI 提供的 API 使用。
11	EFC_INSTRUCTION_ERROR	R	X	内部。仅可通过 TI 提供的 API 使用。
10	EFC_AUTOLOAD_ERROR	R	X	内部。仅可通过 TI 提供的 API 使用。
9	SYS_ECC_OVERRIDE_EN	R	X	内部。仅可通过 TI 提供的 API 使用。
8	EFC_READY	R	X	内部。仅可通过 TI 提供的 API 使用。
7	EFC_FCLRZ	R	X	内部。仅可通过 TI 提供的 API 使用。
6	SYS_DIEID_AUTOLOAD_EN	R	X	内部。仅可通过 TI 提供的 API 使用。
5-4	SYS_REPAIR_EN	R	X	内部。仅可通过 TI 提供的 API 使用。
3-0	SYS_WS_READ_STATES	R	X	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.19 EFUSECRA 寄存器（偏移 = 1030h）[复位 = 00000000h]

图 8-27 展示了 EFUSECRA，表 8-23 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-27. EFUSECRA 寄存器

31	30	29	28	27	26	25	24	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					

表 8-23. EFUSECRA 寄存器字段说明

位	字段	类型	复位	说明
31-6	RESERVED	R	0h	保留
5-0	DATA	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.20 EFUSEREAD 寄存器（偏移 = 1034h）[复位 = 00000000h]

图 8-28 展示了 EFUSEREAD，表 8-24 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-28. EFUSEREAD 寄存器

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

表 8-24. EFUSEREAD 寄存器字段说明

位	字段	类型	复位	说明
31-10	RESERVED	R	0h	保留
9-8	DATABIT	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
7-4	READCLOCK	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
3	DEBUG	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
2	SPARE	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
1-0	MARGIN	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.21 EFUSEPROGRAM 寄存器（偏移 = 1038h）[复位 = 00000000h]

图 8-29 展示了 EFUSEPROGRAM，表 8-25 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-29. EFUSEPROGRAM 寄存器

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

表 8-25. EFUSEPROGRAM 寄存器字段说明

位	字段	类型	复位	说明
31	RESERVED	R	0h	保留
30	COMPAREDISABLE	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
29-14	CLOCKSTALL	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
13	VPPTOVDD	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
12-9	ITERATIONS	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
8-0	WRITECLOCK	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.22 EFUSEERROR 寄存器（偏移 = 103Ch）[复位 = 00000000h]

图 8-30 展示了 EFUSEERROR，表 8-26 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-30. EFUSEERROR 寄存器

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

表 8-26. EFUSEERROR 寄存器字段说明

位	字段	类型	复位	说明
31-6	RESERVED	R	0h	保留
5	DONE	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
4-0	代码	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.23 SINGLEBIT 寄存器（偏移 = 1040h）[复位 = 00000000h]

图 8-31 展示了 SINGLEBIT，表 8-27 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-31. SINGLEBIT 寄存器

31	30	29	28	27	26	25	24
FROMN							
R-0h							
23	22	21	20	19	18	17	16
FROMN							
R-0h							
15	14	13	12	11	10	9	8
FROMN							
R-0h							
7	6	5	4	3	2	1	0
FROMN							FROM0
R-0h							R-0h

表 8-27. SINGLEBIT 寄存器字段说明

位	字段	类型	复位	说明
31-1	FROMN	R	0h	内部。仅可通过 TI 提供的 API 使用。
0	FROM0	R	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.24 TWOBIT 寄存器（偏移 = 1044h）[复位 = 00000000h]

图 8-32 展示了 TWOBIT，表 8-28 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-32. TWOBIT 寄存器

31	30	29	28	27	26	25	24
FROMN							
R-0h							
23	22	21	20	19	18	17	16
FROMN							
R-0h							
15	14	13	12	11	10	9	8
FROMN							
R-0h							
7	6	5	4	3	2	1	0
FROMN							FROM0
R-0h							R-0h

表 8-28. TWOBIT 寄存器字段说明

位	字段	类型	复位	说明
31-1	FROMN	R	0h	内部。仅可通过 TI 提供的 API 使用。
0	FROM0	R	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.25 SELFTESTCYC 寄存器（偏移 = 1048h）[复位 = 00000000h]

图 8-33 展示了 SELFTESTCYC，表 8-29 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-33. SELFTESTCYC 寄存器

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

表 8-29. SELFTESTCYC 寄存器字段说明

位	字段	类型	复位	说明
31-0	周期	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.26 SELFTESTSIGN 寄存器（偏移 = 104Ch）[复位 = 00000000h]

图 8-34 展示了 SELFTESTSIGN，表 8-30 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-34. SELFTESTSIGN 寄存器

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

表 8-30. SELFTESTSIGN 寄存器字段说明

位	字段	类型	复位	说明
31-0	SIGNATURE	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.27 FRDCTL 寄存器（偏移 = 2000h）[复位 = 00000200h]

图 8-35 展示了 FRDCTL，表 8-31 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-35. FRDCTL 寄存器

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				RWAIT				RESERVED							
R-0h				R/W-2h				R-0h							

表 8-31. FRDCTL 寄存器字段说明

位	字段	类型	复位	说明
31-12	RESERVED	R	0h	保留
11-8	RWAIT	R/W	2h	内部。仅可通过 TI 提供的 API 使用。
7-0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.28 FSPRD 寄存器 (偏移 = 2004h) [复位 = 00000000h]

图 8-36 展示了 FSPRD，表 8-32 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-36. FSPRD 寄存器

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
RMBSEM								RESERVED						RM1	RM0
R/W-0h								R-0h						R/W-0h	R/W-0h

表 8-32. FSPRD 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-8	RMBSEM	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
7-2	RESERVED	R	0h	保留
1	RM1	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
0	RM0	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.29 FEDACCTL1 寄存器（偏移 = 2008h）[复位 = 00000000h]

图 8-37 展示了 FEDACCTL1，表 8-33 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-37. FEDACCTL1 寄存器

31	30	29	28	27	26	25	24
RESERVED							SUSP_IGNR
R-0h							R/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							
R-0h							

表 8-33. FEDACCTL1 寄存器字段说明

位	字段	类型	复位	说明
31-25	RESERVED	R	0h	保留
24	SUSP_IGNR	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
23-0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.30 FEDACSTAT 寄存器（偏移 = 201Ch）[复位 = 00000000h]

图 8-38 展示了 FEDACSTAT，表 8-34 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-38. FEDACSTAT 寄存器

31	30	29	28	27	26	25	24
RESERVED						RVF_INT	FSM_DONE
R-0h						R/W1C-0h	R/W1C-0h
23	22	21	20	19	18	17	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							

表 8-34. FEDACSTAT 寄存器字段说明

位	字段	类型	复位	说明
31-26	RESERVED	R	0h	保留
25	RVF_INT	R/W1C	0h	内部。仅可通过 TI 提供的 API 使用。
24	FSM_DONE	R/W1C	0h	内部。仅可通过 TI 提供的 API 使用。
23-0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.31 FBPROT 寄存器（偏移 = 2030h）[复位 = 00000000h]

图 8-39 展示了 FBPROT，表 8-35 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-39. FBPROT 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							PROTL1DIS
R-0h							R/W-0h

表 8-35. FBPROT 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	PROTL1DIS	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.32 FBSE 寄存器（偏移 = 2034h）[复位 = 00000000h]

图 8-40 展示了 FBSE，表 8-36 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-40. FBSE 寄存器

31	30	29	28	27	26	25	24	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																BSE															
R-0h																R/W-0h															

表 8-36. FBSE 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	BSE	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.33 FBBUSY 寄存器（偏移 = 2038h）[复位 = 000000FEh]

图 8-41 展示了 FBBUSY，表 8-37 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-41. FBBUSY 寄存器

31	30	29	28	27	26	25	24	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																								BUSY							
R-0h																								R-FEh							

表 8-37. FBBUSY 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-0	BUSY	R	FEh	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.34 FBAC 寄存器（偏移 = 203Ch）[复位 = 0000000Fh]

图 8-42 展示了 FBAC，表 8-38 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-42. FBAC 寄存器

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

表 8-38. FBAC 寄存器字段说明

位	字段	类型	复位	说明
31-17	RESERVED	R	0h	保留
16	OTPPROTDIS	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
15-8	BAGP	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
7-0	VREADS	R/W	Fh	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.35 FBFALLBACK 寄存器（偏移 = 2040h）[复位 = 0505FFFFh]

图 8-43 展示了 FBFALLBACK，表 8-39 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-43. FBFALLBACK 寄存器

31	30	29	28	27	26	25	24
RESERVED				FSM_PWRSAV			
R-0h				R/W-5h			
23	22	21	20	19	18	17	16
RESERVED				REG_PWRSAV			
R-0h				R/W-5h			
15	14	13	12	11	10	9	8
BANKPWR7		BANKPWR6		BANKPWR5		BANKPWR4	
R/W-3h		R/W-3h		R/W-3h		R/W-3h	
7	6	5	4	3	2	1	0
BANKPWR3		BANKPWR2		BANKPWR1		BANKPWR0	
R/W-3h		R/W-3h		R/W-3h		R/W-3h	

表 8-39. FBFALLBACK 寄存器字段说明

位	字段	类型	复位	说明
31-28	RESERVED	R	0h	保留
27-24	FSM_PWRSAV	R/W	5h	内部。仅可通过 TI 提供的 API 使用。
23-20	RESERVED	R	0h	保留
19-16	REG_PWRSAV	R/W	5h	内部。仅可通过 TI 提供的 API 使用。
15-14	BANKPWR7	R/W	3h	内部。仅可通过 TI 提供的 API 使用。
13-12	BANKPWR6	R/W	3h	内部。仅可通过 TI 提供的 API 使用。
11-10	BANKPWR5	R/W	3h	内部。仅可通过 TI 提供的 API 使用。
9-8	BANKPWR4	R/W	3h	内部。仅可通过 TI 提供的 API 使用。
7-6	BANKPWR3	R/W	3h	内部。仅可通过 TI 提供的 API 使用。
5-4	BANKPWR2	R/W	3h	内部。仅可通过 TI 提供的 API 使用。
3-2	BANKPWR1	R/W	3h	内部。仅可通过 TI 提供的 API 使用。
1-0	BANKPWR0	R/W	3h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.36 FBPRDY 寄存器（偏移 = 2044h）[复位 = 00FF00FEh]

图 8-44 展示了 FBPRDY，表 8-40 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-44. FBPRDY 寄存器

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

表 8-40. FBPRDY 寄存器字段说明

位	字段	类型	复位	说明
31-17	RESERVED	R	0h	保留
16	BANKBUSY	R	1h	内部。仅可通过 TI 提供的 API 使用。
15	PUMPRDY	R	0h	内部。仅可通过 TI 提供的 API 使用。
14-1	RESERVED	R	0h	保留
0	BANKRDY	R	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.37 FPAC1 寄存器（偏移 = 2048h）[复位 = 02082081h]

图 8-45 展示了 FPAC1，表 8-41 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-45. FPAC1 寄存器

31	30	29	28	27	26	25	24
RESERVED				PSLEEPTDIS			
R-0h				R/W-208h			
23	22	21	20	19	18	17	16
PSLEEPTDIS							
R/W-208h							
15	14	13	12	11	10	9	8
PUMPRESET_PW							
R/W-208h							
7	6	5	4	3	2	1	0
PUMPRESET_PW				RESERVED		PUMPPWR	
R/W-208h				R-0h		R/W-1h	

表 8-41. FPAC1 寄存器字段说明

位	字段	类型	复位	说明
31-28	RESERVED	R	0h	保留
27-16	PSLEEPTDIS	R/W	208h	内部。仅可通过 TI 提供的 API 使用。
15-4	PUMPRESET_PW	R/W	208h	内部。仅可通过 TI 提供的 API 使用。
3-2	RESERVED	R	0h	保留
1-0	PUMPPWR	R/W	1h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.38 FPAC2 寄存器（偏移 = 204Ch）[复位 = 00000000h]

图 8-46 展示了 FPAC2，表 8-42 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-46. FPAC2 寄存器

31	30	29	28	27	26	25	24	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																PAGP															
R-0h																R/W-0h															

表 8-42. FPAC2 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	PAGP	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.39 FMAC 寄存器（偏移 = 2050h）[复位 = 00000000h]

图 8-47 展示了 FMAC，表 8-43 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-47. FMAC 寄存器

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													BANK		
R-0h													R/W-0h		

表 8-43. FMAC 寄存器字段说明

位	字段	类型	复位	说明
31-3	RESERVED	R	0h	保留
2-0	BANK	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.40 FMSTAT 寄存器（偏移 = 2054h）[复位 = 00000000h]

图 8-48 展示了 FMSTAT，表 8-44 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-48. FMSTAT 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED						RVSUSP	RDVER
R-0h						R-0h	R-0h
15	14	13	12	11	10	9	8
RVF	ILA	DBF	PGV	PCV	EV	CV	BUSY
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
ERS	PGM	INVDAT	CSTAT	VOLSTAT	ESUSP	PSUSP	SLOCK
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

表 8-44. FMSTAT 寄存器字段说明

位	字段	类型	复位	说明
31-18	RESERVED	R	0h	保留
17	RVSUSP	R	0h	内部。仅可通过 TI 提供的 API 使用。
16	RDVER	R	0h	内部。仅可通过 TI 提供的 API 使用。
15	RVF	R	0h	内部。仅可通过 TI 提供的 API 使用。
14	ILA	R	0h	内部。仅可通过 TI 提供的 API 使用。
13	DBF	R	0h	内部。仅可通过 TI 提供的 API 使用。
12	PGV	R	0h	内部。仅可通过 TI 提供的 API 使用。
11	PCV	R	0h	内部。仅可通过 TI 提供的 API 使用。
10	EV	R	0h	内部。仅可通过 TI 提供的 API 使用。
9	CV	R	0h	内部。仅可通过 TI 提供的 API 使用。
8	BUSY	R	0h	内部。仅可通过 TI 提供的 API 使用。
7	ERS	R	0h	内部。仅可通过 TI 提供的 API 使用。
6	PGM	R	0h	内部。仅可通过 TI 提供的 API 使用。
5	INVDAT	R	0h	内部。仅可通过 TI 提供的 API 使用。
4	CSTAT	R	0h	内部。仅可通过 TI 提供的 API 使用。
3	VOLSTAT	R	0h	内部。仅可通过 TI 提供的 API 使用。
2	ESUSP	R	0h	内部。仅可通过 TI 提供的 API 使用。
1	PSUSP	R	0h	内部。仅可通过 TI 提供的 API 使用。
0	SLOCK	R	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.41 FLOCK 寄存器（偏移 = 2064h）[复位 = 000055AAh]

图 8-49 展示了 FLOCK，表 8-45 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-49. FLOCK 寄存器

31	30	29	28	27	26	25	24	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																ENCOM															
R-0h																R/W-55AAh															

表 8-45. FLOCK 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	ENCOM	R/W	55AAh	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.42 FVREADCT 寄存器（偏移 = 2080h）[复位 = 00000008h]

图 8-50 展示了 FVREADCT，表 8-46 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-50. FVREADCT 寄存器

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												VREADCT			
R-0h												R/W-8h			

表 8-46. FVREADCT 寄存器字段说明

位	字段	类型	复位	说明
31-4	RESERVED	R	0h	保留
3-0	VREADCT	R/W	8h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.43 FVHVCT1 寄存器（偏移 = 2084h）[复位 = 00840088h]

图 8-51 展示了 FVHVCT1，表 8-47 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-51. FVHVCT1 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED								TRIM13_E				VHVCT_E			
R-0h								R/W-8h				R/W-4h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								TRIM13_PV				VHVCT_PV			
R-0h								R/W-8h				R/W-8h			

表 8-47. FVHVCT1 寄存器字段说明

位	字段	类型	复位	说明
31-24	RESERVED	R	0h	保留
23-20	TRIM13_E	R/W	8h	内部。仅可通过 TI 提供的 API 使用。
19-16	VHVCT_E	R/W	4h	内部。仅可通过 TI 提供的 API 使用。
15-8	RESERVED	R	0h	保留
7-4	TRIM13_PV	R/W	8h	内部。仅可通过 TI 提供的 API 使用。
3-0	VHVCT_PV	R/W	8h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.44 FVHVCT2 寄存器（偏移 = 2088h）[复位 = 00A20000h]

图 8-52 展示了 FVHVCT2，表 8-48 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-52. FVHVCT2 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED								TRIM13_P				VHVCT_P			
R-0h								R/W-Ah				R/W-2h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															
R-0h															

表 8-48. FVHVCT2 寄存器字段说明

位	字段	类型	复位	说明
31-24	RESERVED	R	0h	保留
23-20	TRIM13_P	R/W	Ah	内部。仅可通过 TI 提供的 API 使用。
19-16	VHVCT_P	R/W	2h	内部。仅可通过 TI 提供的 API 使用。
15-0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.45 FVHVCT3 寄存器（偏移 = 208Ch）[复位 = 000F0000h]

图 8-53 展示了 FVHVCT3，表 8-49 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-53. FVHVCT3 寄存器

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

表 8-49. FVHVCT3 寄存器字段说明

位	字段	类型	复位	说明
31-20	RESERVED	R	0h	保留
19-16	WCT	R/W	Fh	内部。仅可通过 TI 提供的 API 使用。
15-4	RESERVED	R	0h	保留
3-0	VHVCT_READ	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.46 FVNVCT 寄存器（偏移 = 2090h）[复位 = 00000800h]

图 8-54 展示了 FVNVCT，表 8-50 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-54. FVNVCT 寄存器

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			VCG2P5CT					RESERVED			VIN_CT				
R-0h			R/W-8h					R-0h			R/W-0h				

表 8-50. FVNVCT 寄存器字段说明

位	字段	类型	复位	说明
31-13	RESERVED	R	0h	保留
12-8	VCG2P5CT	R/W	8h	内部。仅可通过 TI 提供的 API 使用。
7-5	RESERVED	R	0h	保留
4-0	VIN_CT	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.47 FVSLP 寄存器（偏移 = 2094h）[复位 = 00008000h]

图 8-55 展示了 FVSLP，表 8-51 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-55. FVSLP 寄存器

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
VSL_P				RESERVED											
R/W-8h				R-0h											

表 8-51. FVSLP 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-12	VSL_P	R/W	8h	内部。仅可通过 TI 提供的 API 使用。
11-0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.48 FVWLCT 寄存器（偏移 = 2098h）[复位 = 00000008h]

图 8-56 展示了 FVWLCT，表 8-52 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-56. FVWLCT 寄存器

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											VWLCT_P				
R-0h											R/W-8h				

表 8-52. FVWLCT 寄存器字段说明

位	字段	类型	复位	说明
31-5	RESERVED	R	0h	保留
4-0	VWLCT_P	R/W	8h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.49 FEFUSECTL 寄存器（偏移 = 209Ch）[复位 = 0701010Ah]

图 8-57 展示了 FEFUSECTL，表 8-53 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-57. FEFUSECTL 寄存器

31	30	29	28	27	26	25	24
RESERVED						CHAIN_SEL	
R-0h						R/W-7h	
23	22	21	20	19	18	17	16
RESERVED						WRITE_EN	BP_SEL
R-0h						R/W-0h	R/W-1h
15	14	13	12	11	10	9	8
RESERVED							EF_CLRZ
R-0h							R/W-1h
7	6	5	4	3	2	1	0
RESERVED			EF_TEST	EFUSE_EN			
R-0h			R/W-0h	R/W-Ah			

表 8-53. FEFUSECTL 寄存器字段说明

位	字段	类型	复位	说明
31-27	RESERVED	R	0h	保留
26-24	CHAIN_SEL	R/W	7h	内部。仅可通过 TI 提供的 API 使用。
23-18	RESERVED	R	0h	保留
17	WRITE_EN	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
16	BP_SEL	R/W	1h	内部。仅可通过 TI 提供的 API 使用。
15-9	RESERVED	R	0h	保留
8	EF_CLRZ	R/W	1h	内部。仅可通过 TI 提供的 API 使用。
7-5	RESERVED	R	0h	保留
4	EF_TEST	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
3-0	EFUSE_EN	R/W	Ah	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.50 FEFUSESTAT 寄存器（偏移 = 20A0h）[复位 = 00000000h]

图 8-58 展示了 FEFUSESTAT，表 8-54 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-58. FEFUSESTAT 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							SHIFT_DONE
R-0h							R/W1C-0h

表 8-54. FEFUSESTAT 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	SHIFT_DONE	R/W1C	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.51 FEFUSEDATA 寄存器（偏移 = 20A4h）[复位 = 00000000h]

图 8-59 展示了 FEFUSEDATA，表 8-55 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-59. FEFUSEDATA 寄存器

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

表 8-55. FEFUSEDATA 寄存器字段说明

位	字段	类型	复位	说明
31-0	FEFUSEDATA	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.52 FSEQPMP 寄存器（偏移 = 20A8h）[复位 = 85080000h]

图 8-60 展示了 FSEQPMP，表 8-56 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-60. FSEQPMP 寄存器

31	30	29	28	27	26	25	24
RESERVED				TRIM_3P4			
R/W-8h				R/W-5h			
23	22	21	20	19	18	17	16
RESERVED		TRIM_1P7		TRIM_0P8			
R-0h		R/W-0h		R/W-8h			
15	14	13	12	11	10	9	8
RESERVED	VIN_AT_X			RESERVED			VIN_BY_PASS
R-0h	R/W-0h			R-0h			R/W-0h
7	6	5	4	3	2	1	0
RESERVED							
R-0h							

表 8-56. FSEQPMP 寄存器字段说明

位	字段	类型	复位	说明
31-28	RESERVED	R/W	8h	内部。仅可通过 TI 提供的 API 使用。
27-24	TRIM_3P4	R/W	5h	内部。仅可通过 TI 提供的 API 使用。
23-22	RESERVED	R	0h	保留
21-20	TRIM_1P7	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
19-16	TRIM_0P8	R/W	8h	内部。仅可通过 TI 提供的 API 使用。
15	RESERVED	R	0h	保留
14-12	VIN_AT_X	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
11-9	RESERVED	R	0h	保留
8	VIN_BY_PASS	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
7-0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.53 FBSTROBES 寄存器（偏移 = 2100h）[复位 = 00000104h]

图 8-61 展示了 FBSTROBES，表 8-57 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-61. FBSTROBES 寄存器

31	30	29	28	27	26	25	24
RESERVED							ECBIT
R-0h							R/W-0h
23	22	21	20	19	18	17	16
RESERVED					RWAIT2_FLCLK	RWAIT_FLCLK	FLCLKEN
R-0h					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED							CTRLENZ
R-0h							R/W-1h
7	6	5	4	3	2	1	0
RESERVED	NOCOLRED	PRECOL	TI_OTP	OTP	TEZ	RESERVED	
R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-1h	R-0h	

表 8-57. FBSTROBES 寄存器字段说明

位	字段	类型	复位	说明
31-25	RESERVED	R	0h	保留
24	ECBIT	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
23-19	RESERVED	R	0h	保留
18	RWAIT2_FLCLK	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
17	RWAIT_FLCLK	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
16	FLCLKEN	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
15-9	RESERVED	R	0h	保留
8	CTRLENZ	R/W	1h	内部。仅可通过 TI 提供的 API 使用。
7	RESERVED	R	0h	保留
6	NOCOLRED	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
5	PRECOL	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
4	TI_OTP	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
3	OTP	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
2	TEZ	R/W	1h	内部。仅可通过 TI 提供的 API 使用。
1-0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.54 FPSTROBES 寄存器（偏移 = 2104h）[复位 = 00000103h]

图 8-62 展示了 FPSTROBES，表 8-58 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-62. FPSTROBES 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							EXECUTEZ
R-0h							R/W-1h
7	6	5	4	3	2	1	0
RESERVED						V3PWRDNZ	V5PWRDNZ
R-0h						R/W-1h	R/W-1h

表 8-58. FPSTROBES 寄存器字段说明

位	字段	类型	复位	说明
31-9	RESERVED	R	0h	保留
8	EXECUTEZ	R/W	1h	内部。仅可通过 TI 提供的 API 使用。
7-2	RESERVED	R	0h	保留
1	V3PWRDNZ	R/W	1h	内部。仅可通过 TI 提供的 API 使用。
0	V5PWRDNZ	R/W	1h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.55 FBMODE 寄存器（偏移 = 2108h）[复位 = 00000000h]

图 8-63 展示了 FBMODE，表 8-59 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-63. FBMODE 寄存器

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														模式	
R-0h														R/W-0h	

表 8-59. FBMODE 寄存器字段说明

位	字段	类型	复位	说明
31-3	RESERVED	R	0h	保留
2-0	模式	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.56 FTCR 寄存器（偏移 = 210Ch）[复位 = 00000000h]

图 8-64 展示了 FTCR，表 8-60 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-64. FTCR 寄存器

31	30	29	28	27	26	25	24	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																									TCR						
R-0h																									R/W-0h						

表 8-60. FTCR 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	TCR	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.57 FADDR 寄存器（偏移 = 2110h）[复位 = 00000000h]

图 8-65 展示了 FADDR，表 8-61 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-65. FADDR 寄存器

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

表 8-61. FADDR 寄存器字段说明

位	字段	类型	复位	说明
31-0	FADDR	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.58 FTCTL 寄存器（偏移 = 211Ch）[复位 = 00000000h]

图 8-66 展示了 FTCTL，表 8-62 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-66. FTCTL 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							WDATA_BLK_CLR
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						TEST_EN	RESERVED
R-0h						R/W-0h	R-0h

表 8-62. FTCTL 寄存器字段说明

位	字段	类型	复位	说明
31-17	RESERVED	R	0h	保留
16	WDATA_BLK_CLR	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
15-2	RESERVED	R	0h	保留
1	TEST_EN	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.59 FWPWRITE0 寄存器（偏移 = 2120h）[复位 = FFFFFFFFh]

图 8-67 展示了 FWPWRITE0，表 8-63 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-67. FWPWRITE0 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FWPWRITE0																															
R/W-FFFFFFFFh																															

表 8-63. FWPWRITE0 寄存器字段说明

位	字段	类型	复位	说明
31-0	FWPWRITE0	R/W	FFFFFFFFh	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.60 FWPWRITE1 寄存器 (偏移 = 2124h) [复位 = FFFFFFFFh]

图 8-68 展示了 FWPWRITE1，表 8-64 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-68. FWPWRITE1 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FWPWRITE1																															
R/W-FFFFFFFFh																															

表 8-64. FWPWRITE1 寄存器字段说明

位	字段	类型	复位	说明
31-0	FWPWRITE1	R/W	FFFFFFFFh	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.61 FWPWRITE2 寄存器 (偏移 = 2128h) [复位 = FFFFFFFFh]

图 8-69 展示了 FWPWRITE2，表 8-65 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-69. FWPWRITE2 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FWPWRITE2																															
R/W-FFFFFFFFh																															

表 8-65. FWPWRITE2 寄存器字段说明

位	字段	类型	复位	说明
31-0	FWPWRITE2	R/W	FFFFFFFFh	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.62 FWPWRITE3 寄存器（偏移 = 212Ch）[复位 = FFFFFFFFh]

图 8-70 展示了 FWPWRITE3，表 8-66 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-70. FWPWRITE3 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FWPWRITE3																															
R/W-FFFFFFFFh																															

表 8-66. FWPWRITE3 寄存器字段说明

位	字段	类型	复位	说明
31-0	FWPWRITE3	R/W	FFFFFFFFh	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.63 FWPWRITE4 寄存器（偏移 = 2130h）[复位 = FFFFFFFFh]

图 8-71 展示了 FWPWRITE4，表 8-67 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-71. FWPWRITE4 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FWPWRITE4																															
R/W-FFFFFFFFh																															

表 8-67. FWPWRITE4 寄存器字段说明

位	字段	类型	复位	说明
31-0	FWPWRITE4	R/W	FFFFFFFFh	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.64 FWPWRITE5 寄存器（偏移 = 2134h）[复位 = FFFFFFFFh]

图 8-72 展示了 FWPWRITE5，表 8-68 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-72. FWPWRITE5 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FWPWRITE5																															
R/W-FFFFFFFFh																															

表 8-68. FWPWRITE5 寄存器字段说明

位	字段	类型	复位	说明
31-0	FWPWRITE5	R/W	FFFFFFFFh	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.65 FWPWRITE6 寄存器（偏移 = 2138h）[复位 = FFFFFFFFh]

图 8-73 展示了 FWPWRITE6，表 8-69 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-73. FWPWRITE6 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FWPWRITE6																															
R/W-FFFFFFFFh																															

表 8-69. FWPWRITE6 寄存器字段说明

位	字段	类型	复位	说明
31-0	FWPWRITE6	R/W	FFFFFFFFh	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.66 FWPWRITE7 寄存器（偏移 = 213Ch）[复位 = FFFFFFFFh]

图 8-74 展示了 FWPWRITE7，表 8-70 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-74. FWPWRITE7 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FWPWRITE7																															
R/W-FFFFFFFFh																															

表 8-70. FWPWRITE7 寄存器字段说明

位	字段	类型	复位	说明
31-0	FWPWRITE7	R/W	FFFFFFFFh	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.67 FWPWRITE_ECC 寄存器（偏移 = 2140h）[复位 = FFFFFFFFh]

图 8-75 展示了 FWPWRITE_ECC，表 8-71 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-75. FWPWRITE_ECC 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ECCBYTES07_00								ECCBYTES15_08							
R/W-FFh								R/W-FFh							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCBYTES23_16								ECCBYTES31_24							
R/W-FFh								R/W-FFh							

表 8-71. FWPWRITE_ECC 寄存器字段说明

位	字段	类型	复位	说明
31-24	ECCBYTES07_00	R/W	FFh	内部。仅可通过 TI 提供的 API 使用。
23-16	ECCBYTES15_08	R/W	FFh	内部。仅可通过 TI 提供的 API 使用。
15-8	ECCBYTES23_16	R/W	FFh	内部。仅可通过 TI 提供的 API 使用。
7-0	ECCBYTES31_24	R/W	FFh	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.68 FSWSTAT 寄存器（偏移 = 2144h）[复位 = 00000001h]

图 8-76 展示了 FSWSTAT，表 8-72 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-76. FSWSTAT 寄存器

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

表 8-72. FSWSTAT 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	SAFELV	R	1h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.69 FSM_GLBCTL 寄存器（偏移 = 2200h）[复位 = 00000001h]

图 8-77 展示了 FSM_GLBCTL，表 8-73 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-77. FSM_GLBCTL 寄存器

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

表 8-73. FSM_GLBCTL 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	CLKSEL	R	1h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.70 FSM_STATE 寄存器（偏移 = 2204h）[复位 = 00000C00h]

图 8-78 展示了 FSM_STATE，表 8-74 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-78. FSM_STATE 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED				CTRLENZ	EXECUTEZ	RESERVED	FSM_ACT
R-0h				R-1h	R-1h	R-0h	R-0h
7	6	5	4	3	2	1	0
TIOTP_ACT	OTP_ACT	RESERVED					
R-0h	R-0h	R-0h					

表 8-74. FSM_STATE 寄存器字段说明

位	字段	类型	复位	说明
31-12	RESERVED	R	0h	保留
11	CTRLENZ	R	1h	内部。仅可通过 TI 提供的 API 使用。
10	EXECUTEZ	R	1h	内部。仅可通过 TI 提供的 API 使用。
9	RESERVED	R	0h	保留
8	FSM_ACT	R	0h	内部。仅可通过 TI 提供的 API 使用。
7	TIOTP_ACT	R	0h	内部。仅可通过 TI 提供的 API 使用。
6	OTP_ACT	R	0h	内部。仅可通过 TI 提供的 API 使用。
5-0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.71 FSM_STAT 寄存器（偏移 = 2208h）[复位 = 00000004h]

图 8-79 展示了 FSM_STAT，表 8-75 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-79. FSM_STAT 寄存器

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

表 8-75. FSM_STAT 寄存器字段说明

位	字段	类型	复位	说明
31-3	RESERVED	R	0h	保留
2	NON_OP	R	1h	内部。仅可通过 TI 提供的 API 使用。
1	OVR_PUL_CNT	R	0h	内部。仅可通过 TI 提供的 API 使用。
0	INV_DAT	R	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.72 FSM_CMD 寄存器（偏移 = 220Ch）[复位 = 00000000h]

图 8-80 展示了 FSM_CMD，表 8-76 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-80. FSM_CMD 寄存器

31	30	29	28	27	26	25	24	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																										FSMCMD					
R-0h																										R/W-0h					

表 8-76. FSM_CMD 寄存器字段说明

位	字段	类型	复位	说明
31-6	RESERVED	R	0h	保留
5-0	FSMCMD	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.73 FSM_PE_OSU 寄存器（偏移 = 2210h）[复位 = 00000000h]

图 8-81 展示了 FSM_PE_OSU，表 8-77 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-81. FSM_PE_OSU 寄存器

31	30	29	28	27	26	25	24	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																PGM_OSU								ERA_OSU							
R-0h																R/W-0h								R/W-0h							

表 8-77. FSM_PE_OSU 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-8	PGM_OSU	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
7-0	ERA_OSU	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.74 FSM_VSTAT 寄存器（偏移 = 2214h）[复位 = 00003000h]

图 8-82 展示了 FSM_VSTAT，表 8-78 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-82. FSM_VSTAT 寄存器

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
VSTAT_CNT				RESERVED											
R/W-3h				R-0h											

表 8-78. FSM_VSTAT 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-12	VSTAT_CNT	R/W	3h	内部。仅可通过 TI 提供的 API 使用。
11-0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.75 FSM_PE_VSU 寄存器（偏移 = 2218h）[复位 = 00000000h]

图 8-83 展示了 FSM_PE_VSU，表 8-79 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-83. FSM_PE_VSU 寄存器

31	30	29	28	27	26	25	24	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																PGM_VSU								ERA_VSU							
R-0h																R/W-0h								R/W-0h							

表 8-79. FSM_PE_VSU 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-8	PGM_VSU	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
7-0	ERA_VSU	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.76 FSM_CMP_VSU 寄存器（偏移 = 221Ch）[复位 = 00000000h]

图 8-84 展示了 FSM_CMP_VSU，表 8-80 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-84. FSM_CMP_VSU 寄存器

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
ADD_EXZ				RESERVED											
R/W-0h				R-0h											

表 8-80. FSM_CMP_VSU 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-12	ADD_EXZ	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
11-0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.77 FSM_EX_VAL 寄存器（偏移 = 2220h）[复位 = 00000301h]

图 8-85 展示了 FSM_EX_VAL，表 8-81 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-85. FSM_EX_VAL 寄存器

31	30	29	28	27	26	25	24	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																REP_VSU								EXE_VALD							
R-0h																R/W-3h								R/W-1h							

表 8-81. FSM_EX_VAL 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-8	REP_VSU	R/W	3h	内部。仅可通过 TI 提供的 API 使用。
7-0	EXE_VALD	R/W	1h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.78 FSM_RD_H 寄存器（偏移 = 2224h）[复位 = 0000005Ah]

图 8-86 展示了 FSM_RD_H，表 8-82 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-86. FSM_RD_H 寄存器

31	30	29	28	27	26	25	24	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																								RD_H							
R-0h																								R/W-5Ah							

表 8-82. FSM_RD_H 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-0	RD_H	R/W	5Ah	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.79 FSM_P_OH 寄存器（偏移 = 2228h）[复位 = 00000100h]

图 8-87 展示了 FSM_P_OH，表 8-83 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-87. FSM_P_OH 寄存器

31	30	29	28	27	26	25	24	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																PGM_OH								RESERVED							
R-0h																R/W-1h								R-0h							

表 8-83. FSM_P_OH 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-8	PGM_OH	R/W	1h	内部。仅可通过 TI 提供的 API 使用。
7-0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.80 FSM_ERA_OH 寄存器（偏移 = 222Ch）[复位 = 00000001h]

图 8-88 展示了 FSM_ERA_OH，表 8-84 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-88. FSM_ERA_OH 寄存器

31	30	29	28	27	26	25	24	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																ERA_OH															
R-0h																R/W-1h															

表 8-84. FSM_ERA_OH 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	ERA_OH	R/W	1h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.81 FSM_SAV_PPUL 寄存器（偏移 = 2230h）[复位 = 00000000h]

图 8-89 展示了 FSM_SAV_PPUL，表 8-85 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-89. FSM_SAV_PPUL 寄存器

31	30	29	28	27	26	25	24	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																				SAV_P_PUL											
R-0h																				R-0h											

表 8-85. FSM_SAV_PPUL 寄存器字段说明

位	字段	类型	复位	说明
31-12	RESERVED	R	0h	保留
11-0	SAV_P_PUL	R	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.82 FSM_PE_VH 寄存器（偏移 = 2234h）[复位 = 00000100h]

图 8-90 展示了 FSM_PE_VH，表 8-86 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-90. FSM_PE_VH 寄存器

31	30	29	28	27	26	25	24	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																PGM_VH								RESERVED							
R-0h																R/W-1h								R-0h							

表 8-86. FSM_PE_VH 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-8	PGM_VH	R/W	1h	内部。仅可通过 TI 提供的 API 使用。
7-0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.83 FSM_PRG_PW 寄存器（偏移 = 2240h）[复位 = 00000000h]

图 8-91 展示了 FSM_PRG_PW，表 8-87 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-91. FSM_PRG_PW 寄存器

31	30	29	28	27	26	25	24	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																PROG_PUL_WIDTH															
R-0h																R/W-0h															

表 8-87. FSM_PRG_PW 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	PROG_PUL_WIDTH	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.84 FSM_ERA_PW 寄存器（偏移 = 2244h）[复位 = 00000000h]

图 8-92 展示了 FSM_ERA_PW，表 8-88 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-92. FSM_ERA_PW 寄存器

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

表 8-88. FSM_ERA_PW 寄存器字段说明

位	字段	类型	复位	说明
31-0	FSM_ERA_PW	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.85 FSM_SAV_ERA_PUL 寄存器（偏移 = 2254h）[复位 = 00000000h]

图 8-93 展示了 FSM_SAV_ERA_PUL，表 8-89 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-93. FSM_SAV_ERA_PUL 寄存器

31	30	29	28	27	26	25	24	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																				SAV_ERA_PUL											
R-0h																				R-0h											

表 8-89. FSM_SAV_ERA_PUL 寄存器字段说明

位	字段	类型	复位	说明
31-12	RESERVED	R	0h	内部。仅可通过 TI 提供的 API 使用。
11-0	SAV_ERA_PUL	R	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.86 FSM_TIMER 寄存器（偏移 = 2258h）[复位 = 00000000h]

图 8-94 展示了 FSM_TIMER，表 8-90 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-94. FSM_TIMER 寄存器

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

表 8-90. FSM_TIMER 寄存器字段说明

位	字段	类型	复位	说明
31-0	FSM_TIMER	R	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.87 FSM_MODE 寄存器（偏移 = 225Ch）[复位 = 00000000h]

图 8-95 展示了 FSM_MODE，表 8-91 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-95. FSM_MODE 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED				RDV_SUBMODE		PGM_SUBMODE	
R-0h				R-0h		R-0h	
15	14	13	12	11	10	9	8
ERA_SUBMODE		SUBMODE		SAV_PGM_CMD			SAV_ERA_MODE
R-0h		R-0h		R-0h			R-0h
7	6	5	4	3	2	1	0
SAV_ERA_MODE		模式			CMD		
R-0h		R-0h			R-0h		

表 8-91. FSM_MODE 寄存器字段说明

位	字段	类型	复位	说明
31-20	RESERVED	R	0h	保留
19-18	RDV_SUBMODE	R	0h	内部。仅可通过 TI 提供的 API 使用。
17-16	PGM_SUBMODE	R	0h	内部。仅可通过 TI 提供的 API 使用。
15-14	ERA_SUBMODE	R	0h	内部。仅可通过 TI 提供的 API 使用。
13-12	SUBMODE	R	0h	内部。仅可通过 TI 提供的 API 使用。
11-9	SAV_PGM_CMD	R	0h	内部。仅可通过 TI 提供的 API 使用。
8-6	SAV_ERA_MODE	R	0h	内部。仅可通过 TI 提供的 API 使用。
5-3	模式	R	0h	内部。仅可通过 TI 提供的 API 使用。
2-0	CMD	R	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.88 FSM_PGM 寄存器（偏移 = 2260h）[复位 = 00000000h]

图 8-96 展示了 FSM_PGM，表 8-92 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-96. FSM_PGM 寄存器

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

表 8-92. FSM_PGM 寄存器字段说明

位	字段	类型	复位	说明
31-26	RESERVED	R	0h	保留
25-23	PGM_BANK	R	0h	内部。仅可通过 TI 提供的 API 使用。
22-0	PGM_ADDR	R	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.89 FSM_ERA 寄存器（偏移 = 2264h）[复位 = 00000000h]

图 8-97 展示了 FSM_ERA，表 8-93 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-97. FSM_ERA 寄存器

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

表 8-93. FSM_ERA 寄存器字段说明

位	字段	类型	复位	说明
31-26	RESERVED	R	0h	保留
25-23	ERA_BANK	R	0h	内部。仅可通过 TI 提供的 API 使用。
22-0	ERA_ADDR	R	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.90 FSM_PRG_PUL 寄存器（偏移 = 2268h）[复位 = 00040032h]

图 8-98 展示了 FSM_PRG_PUL，表 8-94 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-98. FSM_PRG_PUL 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED												BEG_EC_LEVEL			
R-0h												R/W-4h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				MAX_PRG_PUL											
R-0h				R/W-32h											

表 8-94. FSM_PRG_PUL 寄存器字段说明

位	字段	类型	复位	说明
31-20	RESERVED	R	0h	保留
19-16	BEG_EC_LEVEL	R/W	4h	内部。仅可通过 TI 提供的 API 使用。
15-12	RESERVED	R	0h	保留
11-0	MAX_PRG_PUL	R/W	32h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.91 FSM_ERA_PUL 寄存器（偏移 = 226Ch）[复位 = 00040BB8h]

图 8-99 展示了 FSM_ERA_PUL，表 8-95 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-99. FSM_ERA_PUL 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED												MAX_EC_LEVEL			
R-0h												R/W-4h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				MAX_ERA_PUL											
R-0h				R/W-BB8h											

表 8-95. FSM_ERA_PUL 寄存器字段说明

位	字段	类型	复位	说明
31-20	RESERVED	R	0h	保留
19-16	MAX_EC_LEVEL	R/W	4h	内部。仅可通过 TI 提供的 API 使用。
15-12	RESERVED	R	0h	保留
11-0	MAX_ERA_PUL	R/W	BB8h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.92 FSM_STEP_SIZE 寄存器（偏移 = 2270h）[复位 = 00000000h]

图 8-100 展示了 FSM_STEP_SIZE，表 8-96 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-100. FSM_STEP_SIZE 寄存器

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

表 8-96. FSM_STEP_SIZE 寄存器字段说明

位	字段	类型	复位	说明
31-25	RESERVED	R	0h	保留
24-16	EC_STEP_SIZE	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
15-0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.93 FSM_PUL_CNTR 寄存器（偏移 = 2274h）[复位 = 00000000h]

图 8-101 展示了 FSM_PUL_CNTR，表 8-97 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-101. FSM_PUL_CNTR 寄存器

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

表 8-97. FSM_PUL_CNTR 寄存器字段说明

位	字段	类型	复位	说明
31-25	RESERVED	R	0h	保留
24-16	CUR_EC_LEVEL	R	0h	内部。仅可通过 TI 提供的 API 使用。
15-12	RESERVED	R	0h	保留
11-0	PUL_CNTR	R	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.94 FSM_EC_STEP_HEIGHT 寄存器（偏移 = 2278h）[复位 = 00000000h]

图 8-102 展示了 FSM_EC_STEP_HEIGHT，表 8-98 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-102. FSM_EC_STEP_HEIGHT 寄存器

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

表 8-98. FSM_EC_STEP_HEIGHT 寄存器字段说明

位	字段	类型	复位	说明
31-4	RESERVED	R	0h	保留
3-0	EC_STEP_HEIGHT	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.95 FSM_ST_MACHINE 寄存器（偏移 = 227Ch）[复位 = 00800500h]

图 8-103 展示了 FSM_ST_MACHINE，表 8-99 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-103. FSM_ST_MACHINE 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
DO_PRECOND	FSM_INT_EN	ALL_BANKS	CMPV_ALLOWED	RANDOM	RV_SEC_EN	RV_RES	RV_INT_EN
R/W-1h	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	ONE_TIME_GOOD	RESERVED		DO_REDU_COL	DBG_SHORT_ROW		
R-0h	R/W-0h	R-0h		R/W-0h	R/W-Ah		
7	6	5	4	3	2	1	0
DBG_SHORT_ROW	RESERVED	PGM_SEC_COF_EN	PREC_STOP_EN	DIS_TST_EN	CMD_EN	INV_DATA	OVERRIDE
R/W-Ah	R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 8-99. FSM_ST_MACHINE 寄存器字段说明

位	字段	类型	复位	说明
31-24	RESERVED	R	0h	保留
23	DO_PRECOND	R/W	1h	内部。仅可通过 TI 提供的 API 使用。
22	FSM_INT_EN	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
21	ALL_BANKS	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
20	CMPV_ALLOWED	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
19	RANDOM	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
18	RV_SEC_EN	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
17	RV_RES	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
16	RV_INT_EN	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
15	RESERVED	R	0h	保留
14	ONE_TIME_GOOD	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
13-12	RESERVED	R	0h	保留
11	DO_REDU_COL	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
10-7	DBG_SHORT_ROW	R/W	Ah	内部。仅可通过 TI 提供的 API 使用。
6	RESERVED	R	0h	保留
5	PGM_SEC_COF_EN	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
4	PREC_STOP_EN	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
3	DIS_TST_EN	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
2	CMD_EN	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
1	INV_DATA	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
0	OVERRIDE	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.96 FSM_FLES 寄存器（偏移 = 2280h）[复位 = 00000000h]

图 8-104 展示了 FSM_FLES，表 8-100 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-104. FSM_FLES 寄存器

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				BLK_TIOTP				BLK_OTP							
R-0h				R/W-0h				R/W-0h							

表 8-100. FSM_FLES 寄存器字段说明

位	字段	类型	复位	说明
31-12	RESERVED	R	0h	保留
11-8	BLK_TIOTP	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
7-0	BLK_OTP	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.97 FSM_WR_ENA 寄存器（偏移 = 2288h）[复位 = 00000002h]

图 8-105 展示了 FSM_WR_ENA，表 8-101 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-105. FSM_WR_ENA 寄存器

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														WR_ENA	
R-0h														R/W-2h	

表 8-101. FSM_WR_ENA 寄存器字段说明

位	字段	类型	复位	说明
31-3	RESERVED	R	0h	保留
2-0	WR_ENA	R/W	2h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.98 FSM_ACC_PP 寄存器（偏移 = 228Ch）[复位 = 00000000h]

图 8-106 展示了 FSM_ACC_PP，表 8-102 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-106. FSM_ACC_PP 寄存器

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

表 8-102. FSM_ACC_PP 寄存器字段说明

位	字段	类型	复位	说明
31-0	FSM_ACC_PP	R	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.99 FSM_ACC_EP 寄存器（偏移 = 2290h）[复位 = 00000000h]

图 8-107 展示了 FSM_ACC_EP，表 8-103 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-107. FSM_ACC_EP 寄存器

31	30	29	28	27	26	25	24	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																ACC_EP															
R-0h																R-0h															

表 8-103. FSM_ACC_EP 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	ACC_EP	R	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.100 FSM_ADDR 寄存器（偏移 = 22A0h）[复位 = 00000000h]

图 8-108 展示了 FSM_ADDR，表 8-104 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-108. FSM_ADDR 寄存器

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

表 8-104. FSM_ADDR 寄存器字段说明

位	字段	类型	复位	说明
31	RESERVED	R	0h	保留
30-28	BANK	R	0h	内部。仅可通过 TI 提供的 API 使用。
27-0	CUR_ADDR	R	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.101 FSM_SECTOR 寄存器（偏移 = 22A4h）[复位 = FFFF0000h]

图 8-109 展示了 FSM_SECTOR，表 8-105 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-109. FSM_SECTOR 寄存器

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

表 8-105. FSM_SECTOR 寄存器字段说明

位	字段	类型	复位	说明
31-16	SECT_ERASED	R/W	FFFFh	内部。仅可通过 TI 提供的 API 使用。
15-8	FSM_SECTOR_EXTENSION	R	0h	内部。仅可通过 TI 提供的 API 使用。
7-4	扇区	R	0h	内部。仅可通过 TI 提供的 API 使用。
3-0	SEC_OUT	R	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.102 FMC_REV_ID 寄存器（偏移 = 22A8h）[复位 = 00000000h]

图 8-110 展示了 FMC_REV_ID，表 8-106 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-110. FMC_REV_ID 寄存器

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

表 8-106. FMC_REV_ID 寄存器字段说明

位	字段	类型	复位	说明
31-12	MOD_VERSION	R	X	内部。仅可通过 TI 提供的 API 使用。
11-0	CONFIG_CRC	R	X	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.103 FSM_ERR_ADDR 寄存器（偏移 = 22ACh）[复位 = 00000000h]

图 8-111 展示了 FSM_ERR_ADDR，表 8-107 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-111. FSM_ERR_ADDR 寄存器

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

表 8-107. FSM_ERR_ADDR 寄存器字段说明

位	字段	类型	复位	说明
31-8	FSM_ERR_ADDR	R	0h	内部。仅可通过 TI 提供的 API 使用。
7-4	RESERVED	R	0h	保留
3-0	FSM_ERR_BANK	R	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.104 FSM_PGM_MAXPUL 寄存器（偏移 = 22B0h）[复位 = 00000000h]

图 8-112 展示了 FSM_PGM_MAXPUL，表 8-108 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-112. FSM_PGM_MAXPUL 寄存器

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				FSM_PGM_MAXPUL											
R-0h				R-0h											

表 8-108. FSM_PGM_MAXPUL 寄存器字段说明

位	字段	类型	复位	说明
31-12	RESERVED	R	0h	保留
11-0	FSM_PGM_MAXPUL	R	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.105 FSM_EXECUTE 寄存器（偏移 = 22B4h）[复位 = 000A000Ah]

图 8-113 展示了 FSM_EXECUTE，表 8-109 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-113. FSM_EXECUTE 寄存器

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

表 8-109. FSM_EXECUTE 寄存器字段说明

位	字段	类型	复位	说明
31-20	RESERVED	R	0h	保留
19-16	SUSPEND_NOW	R/W	Ah	内部。仅可通过 TI 提供的 API 使用。
15-5	RESERVED	R	0h	保留
4-0	FSMEXECUTE	R/W	Ah	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.106 FSM_SECTOR1 寄存器（偏移 = 22C0h）[复位 = FFFFFFFFh]

图 8-114 展示了 FSM_SECTOR1，表 8-110 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-114. FSM_SECTOR1 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FSM_SECTOR1																															
R/W-FFFFFFFFh																															

表 8-110. FSM_SECTOR1 寄存器字段说明

位	字段	类型	复位	说明
31-0	FSM_SECTOR1	R/W	FFFFFFFFh	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.107 FSM_SECTOR2 寄存器（偏移 = 22C4h）[复位 = 00000FFFh]

图 8-115 展示了 FSM_SECTOR2，表 8-111 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-115. FSM_SECTOR2 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FSM_SECTOR2																															
R/W-FFFh																															

表 8-111. FSM_SECTOR2 寄存器字段说明

位	字段	类型	复位	说明
31-0	FSM_SECTOR2	R/W	FFFh	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.108 FSM_BSLE0 寄存器（偏移 = 22E0h）[复位 = 00000000h]

图 8-116 展示了 FSM_BSLE0，表 8-112 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-116. FSM_BSLE0 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FSM_BSLE0																															
R/W-0h																															

表 8-112. FSM_BSLE0 寄存器字段说明

位	字段	类型	复位	说明
31-0	FSM_BSLE0	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.109 FSM_BSLE1 寄存器（偏移 = 22E4h）[复位 = 00000000h]

图 8-117 展示了 FSM_BSLE1，表 8-113 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-117. FSM_BSLE1 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FSM_BSL1																															
R/W-0h																															

表 8-113. FSM_BSLE1 寄存器字段说明

位	字段	类型	复位	说明
31-0	FSM_BSL1	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.110 FSM_BSLP0 寄存器（偏移 = 22F0h）[复位 = 00000000h]

图 8-118 展示了 FSM_BSLP0，表 8-114 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-118. FSM_BSLP0 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FSM_BSLP0																															
R/W-0h																															

表 8-114. FSM_BSLP0 寄存器字段说明

位	字段	类型	复位	说明
31-0	FSM_BSLP0	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.111 FSM_BSLP1 寄存器（偏移 = 22F4h）[复位 = 00000000h]

图 8-119 展示了 FSM_BSLP1，表 8-115 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-119. FSM_BSLP1 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FSM_BSL1																															
R/W-0h																															

表 8-115. FSM_BSLP1 寄存器字段说明

位	字段	类型	复位	说明
31-0	FSM_BSL1	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.112 FSM_PGM128 寄存器（偏移 = 22F8h）[复位 = 00000000h]

图 8-120 展示了 FSM_PGM128，表 8-116 对其进行了介绍。

返回到汇总表。

FMC FSM 使能 128 位宽编程

图 8-120. FSM_PGM128 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							EN_PGM128
R-0h							R/W-0h

表 8-116. FSM_PGM128 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	EN_PGM128	R/W	0h	1：使能 128 位宽编程。该模式要求闪存泵处的编程电源电压大于 2.5V。该模式的主要用例是制造测试，可用于缩短测试时间。 0：64 位宽编程。在任何编程电压下都有效。一个 128 位字被分为两个 64 位字进行编程。【默认】 该寄存器由 FSM_WR_ENA 寄存器进行写保护。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.113 FCFG_BANK 寄存器（偏移 = 2400h）[复位 = 00000801h]

图 8-121 展示了 FCFG_BANK，表 8-117 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-121. FCFG_BANK 寄存器

31	30	29	28	27	26	25	24
EE_BANK_WIDTH							
R-0h							
23	22	21	20	19	18	17	16
EE_BANK_WIDTH				EE_NUM_BANK			
R-0h				R-0h			
15	14	13	12	11	10	9	8
MAIN_BANK_WIDTH							
R-80h							
7	6	5	4	3	2	1	0
MAIN_BANK_WIDTH				MAIN_NUM_BANK			
R-80h				R-1h			

表 8-117. FCFG_BANK 寄存器字段说明

位	字段	类型	复位	说明
31-20	EE_BANK_WIDTH	R	0h	内部。仅可通过 TI 提供的 API 使用。
19-16	EE_NUM_BANK	R	0h	内部。仅可通过 TI 提供的 API 使用。
15-4	MAIN_BANK_WIDTH	R	80h	内部。仅可通过 TI 提供的 API 使用。
3-0	MAIN_NUM_BANK	R	1h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.114 FCFG_WRAPPER 寄存器（偏移 = 2404h）[复位 = 50009007h]

图 8-122 展示了 FCFG_WRAPPER，表 8-118 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-122. FCFG_WRAPPER 寄存器

31	30	29	28	27	26	25	24
FAMILY_TYPE							
R-50h							
23	22	21	20	19	18	17	16
RESERVED			MEM_MAP	CPU2			
R-0h			R-0h	R-0h			
15	14	13	12	11	10	9	8
EE_IN_MAIN				ROM	IFLUSH	SIL3	ECCA
R-9h				R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
AUTO_SUSP		UERR		CPU_TYPE1			
R-0h		R-0h		R-7h			

表 8-118. FCFG_WRAPPER 寄存器字段说明

位	字段	类型	复位	说明
31-24	FAMILY_TYPE	R	50h	内部。仅可通过 TI 提供的 API 使用。
23-21	RESERVED	R	0h	保留
20	MEM_MAP	R	0h	内部。仅可通过 TI 提供的 API 使用。
19-16	CPU2	R	0h	内部。仅可通过 TI 提供的 API 使用。
15-12	EE_IN_MAIN	R	9h	内部。仅可通过 TI 提供的 API 使用。
11	ROM	R	0h	内部。仅可通过 TI 提供的 API 使用。
10	IFLUSH	R	0h	内部。仅可通过 TI 提供的 API 使用。
9	SIL3	R	0h	内部。仅可通过 TI 提供的 API 使用。
8	ECCA	R	0h	内部。仅可通过 TI 提供的 API 使用。
7-6	AUTO_SUSP	R	0h	内部。仅可通过 TI 提供的 API 使用。
5-4	UERR	R	0h	内部。仅可通过 TI 提供的 API 使用。
3-0	CPU_TYPE1	R	7h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.115 FCFG_BNK_TYPE 寄存器（偏移 = 2408h）[复位 = 00000004h]

图 8-123 展示了 FCFG_BNK_TYPE，表 8-119 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-123. FCFG_BNK_TYPE 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
B7_TYPE				B6_TYPE				B5_TYPE				B4_TYPE			
R-0h				R-0h				R-0h				R-0h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
B3_TYPE				B2_TYPE				B1_TYPE				B0_TYPE			
R-0h				R-0h				R-0h				R-4h			

表 8-119. FCFG_BNK_TYPE 寄存器字段说明

位	字段	类型	复位	说明
31-28	B7_TYPE	R	0h	内部。仅可通过 TI 提供的 API 使用。
27-24	B6_TYPE	R	0h	内部。仅可通过 TI 提供的 API 使用。
23-20	B5_TYPE	R	0h	内部。仅可通过 TI 提供的 API 使用。
19-16	B4_TYPE	R	0h	内部。仅可通过 TI 提供的 API 使用。
15-12	B3_TYPE	R	0h	内部。仅可通过 TI 提供的 API 使用。
11-8	B2_TYPE	R	0h	内部。仅可通过 TI 提供的 API 使用。
7-4	B1_TYPE	R	0h	内部。仅可通过 TI 提供的 API 使用。
3-0	B0_TYPE	R	4h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.116 FCFG_B0_START 寄存器（偏移 = 2410h）[复位 = 02000000h]

图 8-124 展示了 FCFG_B0_START，表 8-120 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-124. FCFG_B0_START 寄存器

31	30	29	28	27	26	25	24
B0_MAX_SECTOR				B0_MUX_FACTOR			
R-0h				R-2h			
23	22	21	20	19	18	17	16
B0_START_ADDR							
R-0h							
15	14	13	12	11	10	9	8
B0_START_ADDR							
R-0h							
7	6	5	4	3	2	1	0
B0_START_ADDR							
R-0h							

表 8-120. FCFG_B0_START 寄存器字段说明

位	字段	类型	复位	说明
31-28	B0_MAX_SECTOR	R	0h	内部。仅可通过 TI 提供的 API 使用。
27-24	B0_MUX_FACTOR	R	2h	内部。仅可通过 TI 提供的 API 使用。
23-0	B0_START_ADDR	R	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.117 FCFG_B1_START 寄存器（偏移 = 2414h）[复位 = 00000000h]

图 8-125 展示了 FCFG_B1_START，表 8-121 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-125. FCFG_B1_START 寄存器

31	30	29	28	27	26	25	24
B1_MAX_SECTOR				B1_MUX_FACTOR			
R-0h				R-0h			
23	22	21	20	19	18	17	16
B1_START_ADDR							
R-0h							
15	14	13	12	11	10	9	8
B1_START_ADDR							
R-0h							
7	6	5	4	3	2	1	0
B1_START_ADDR							
R-0h							

表 8-121. FCFG_B1_START 寄存器字段说明

位	字段	类型	复位	说明
31-28	B1_MAX_SECTOR	R	0h	内部。仅可通过 TI 提供的 API 使用。
27-24	B1_MUX_FACTOR	R	0h	内部。仅可通过 TI 提供的 API 使用。
23-0	B1_START_ADDR	R	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.118 FCFG_B2_START 寄存器（偏移 = 2418h）[复位 = 00000000h]

图 8-126 展示了 FCFG_B2_START，表 8-122 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-126. FCFG_B2_START 寄存器

31	30	29	28	27	26	25	24
B2_MAX_SECTOR				B2_MUX_FACTOR			
R-0h				R-0h			
23	22	21	20	19	18	17	16
B2_START_ADDR							
R-0h							
15	14	13	12	11	10	9	8
B2_START_ADDR							
R-0h							
7	6	5	4	3	2	1	0
B2_START_ADDR							
R-0h							

表 8-122. FCFG_B2_START 寄存器字段说明

位	字段	类型	复位	说明
31-28	B2_MAX_SECTOR	R	0h	内部。仅可通过 TI 提供的 API 使用。
27-24	B2_MUX_FACTOR	R	0h	内部。仅可通过 TI 提供的 API 使用。
23-0	B2_START_ADDR	R	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.119 FCFG_B3_START 寄存器（偏移 = 241Ch）[复位 = 00000000h]

图 8-127 展示了 FCFG_B3_START，表 8-123 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-127. FCFG_B3_START 寄存器

31	30	29	28	27	26	25	24
B3_MAX_SECTOR				B3_MUX_FACTOR			
R-0h				R-0h			
23	22	21	20	19	18	17	16
B3_START_ADDR							
R-0h							
15	14	13	12	11	10	9	8
B3_START_ADDR							
R-0h							
7	6	5	4	3	2	1	0
B3_START_ADDR							
R-0h							

表 8-123. FCFG_B3_START 寄存器字段说明

位	字段	类型	复位	说明
31-28	B3_MAX_SECTOR	R	0h	内部。仅可通过 TI 提供的 API 使用。
27-24	B3_MUX_FACTOR	R	0h	内部。仅可通过 TI 提供的 API 使用。
23-0	B3_START_ADDR	R	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.120 FCFG_B4_START 寄存器（偏移 = 2420h）[复位 = 00000000h]

图 8-128 展示了 FCFG_B4_START，表 8-124 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-128. FCFG_B4_START 寄存器

31	30	29	28	27	26	25	24
B4_MAX_SECTOR				B4_MUX_FACTOR			
R-0h				R-0h			
23	22	21	20	19	18	17	16
B4_START_ADDR							
R-0h							
15	14	13	12	11	10	9	8
B4_START_ADDR							
R-0h							
7	6	5	4	3	2	1	0
B4_START_ADDR							
R-0h							

表 8-124. FCFG_B4_START 寄存器字段说明

位	字段	类型	复位	说明
31-28	B4_MAX_SECTOR	R	0h	内部。仅可通过 TI 提供的 API 使用。
27-24	B4_MUX_FACTOR	R	0h	内部。仅可通过 TI 提供的 API 使用。
23-0	B4_START_ADDR	R	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.121 FCFG_B5_START 寄存器（偏移 = 2424h）[复位 = 00000000h]

图 8-129 展示了 FCFG_B5_START，表 8-125 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-129. FCFG_B5_START 寄存器

31	30	29	28	27	26	25	24
B5_MAX_SECTOR				B5_MUX_FACTOR			
R-0h				R-0h			
23	22	21	20	19	18	17	16
B5_START_ADDR							
R-0h							
15	14	13	12	11	10	9	8
B5_START_ADDR							
R-0h							
7	6	5	4	3	2	1	0
B5_START_ADDR							
R-0h							

表 8-125. FCFG_B5_START 寄存器字段说明

位	字段	类型	复位	说明
31-28	B5_MAX_SECTOR	R	0h	内部。仅可通过 TI 提供的 API 使用。
27-24	B5_MUX_FACTOR	R	0h	内部。仅可通过 TI 提供的 API 使用。
23-0	B5_START_ADDR	R	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.122 FCFG_B6_START 寄存器（偏移 = 2428h）[复位 = 00000000h]

图 8-130 展示了 FCFG_B6_START，表 8-126 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-130. FCFG_B6_START 寄存器

31	30	29	28	27	26	25	24
B6_MAX_SECTOR				B6_MUX_FACTOR			
R-0h				R-0h			
23	22	21	20	19	18	17	16
B6_START_ADDR							
R-0h							
15	14	13	12	11	10	9	8
B6_START_ADDR							
R-0h							
7	6	5	4	3	2	1	0
B6_START_ADDR							
R-0h							

表 8-126. FCFG_B6_START 寄存器字段说明

位	字段	类型	复位	说明
31-28	B6_MAX_SECTOR	R	0h	内部。仅可通过 TI 提供的 API 使用。
27-24	B6_MUX_FACTOR	R	0h	内部。仅可通过 TI 提供的 API 使用。
23-0	B6_START_ADDR	R	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.123 FCFG_B7_START 寄存器（偏移 = 242Ch）[复位 = 00000000h]

图 8-131 展示了 FCFG_B7_START，表 8-127 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-131. FCFG_B7_START 寄存器

31	30	29	28	27	26	25	24
B7_MAX_SECTOR				B7_MUX_FACTOR			
R-0h				R-0h			
23	22	21	20	19	18	17	16
B7_START_ADDR							
R-0h							
15	14	13	12	11	10	9	8
B7_START_ADDR							
R-0h							
7	6	5	4	3	2	1	0
B7_START_ADDR							
R-0h							

表 8-127. FCFG_B7_START 寄存器字段说明

位	字段	类型	复位	说明
31-28	B7_MAX_SECTOR	R	0h	内部。仅可通过 TI 提供的 API 使用。
27-24	B7_MUX_FACTOR	R	0h	内部。仅可通过 TI 提供的 API 使用。
23-0	B7_START_ADDR	R	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.1.124 FCFG_B0_SSIZE0 寄存器（偏移 = 2430h）[复位 = 002C0008h]

图 8-132 展示了 FCFG_B0_SSIZE0，表 8-128 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 8-132. FCFG_B0_SSIZE0 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED				B0_NUM_SECTORS											
R-0h				R-2Ch											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												B0_SECT_SIZE			
R-0h												R-8h			

表 8-128. FCFG_B0_SSIZE0 寄存器字段说明

位	字段	类型	复位	说明
31-28	RESERVED	R	0h	保留
27-16	B0_NUM_SECTORS	R	2Ch	内部。仅可通过 TI 提供的 API 使用。
15-4	RESERVED	R	0h	保留
3-0	B0_SECT_SIZE	R	8h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.2 VIMS 寄存器

表 8-129 列出了 VIMS 寄存器的存储器映射寄存器。表 8-129 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

表 8-129. VIMS 寄存器

偏移	首字母缩写词	寄存器名称	部分
0h	STAT	状态	STAT 寄存器 (偏移 = 0h) [复位 = 00000000h]
4h	CTL	控制	CTL 寄存器 (偏移 = 4h) [复位 = 00000000h]

复杂的位访问类型经过编码可适应小型表单元。表 8-130 展示了适用于此部分中访问类型的代码。

表 8-130. VIMS 访问类型代码

访问类型	代码	说明
读取类型		
R	R	读取
写入类型		
W	W	写入
复位或默认值		
-n		复位后的值或默认值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.2.1 STAT 寄存器（偏移 = 0h）[复位 = 00000000h]

图 8-133 展示了 STAT，表 8-131 中对此进行了介绍。

返回到汇总表。

状态
显示当前 VIMS 模式及线路缓冲状态

图 8-133. STAT 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED		IDCODE_LB_DIS	SYSBUS_LB_DIS	MODE_CHANGING	INV	模式	
R-0h		R-0h	R-0h	R-0h	R-0h	R-0h	

表 8-131. STAT 寄存器字段说明

位	字段	类型	复位	说明
31-6	RESERVED	R	0h	保留
5	IDCODE_LB_DIS	R	0h	Icode/Dcode 闪存行缓冲器状态 0：启用或者转换为禁用 1：已禁用并刷新
4	SYSBUS_LB_DIS	R	0h	Sysbus 闪存行缓冲器控制 0：启用或者转换为禁用 1：已禁用并刷新
3	MODE_CHANGING	R	0h	VIMS 模式更改状态 0：VIMS 处于 MODE 定义的模式 1：VIMS 正在切换到 CTL.MODE 中给出的模式
2	INV	R	0h	当缓存内存无效处于活动状态/正在进行时，设置此位
1-0	模式	R	0h	当前 VIMS 模式 0h = GPRAM：VIMS GPRAM 模式 1h = CACHE： VIMS 缓存模式 3h = VIMS 关闭模式

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

8.7.2.2 CTL 寄存器（偏移 = 4h）[复位 = 00000000h]

图 8-134 展示了 CTL，表 8-132 中对此进行了介绍。

返回到汇总表。

控制
配置 VIMS 模式和线路缓冲设置

图 8-134. CTL 寄存器

31	30	29	28	27	26	25	24
STATS_CLR	STATS_EN	DYN_CG_EN	RESERVED				
R/W-0h	R/W-0h	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		IDCODE_LB_DIS	SYSBUS_LB_DIS	ARB_CFG	PREF_EN	模式	
R-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	

表 8-132. CTL 寄存器字段说明

位	字段	类型	复位	说明
31	STATS_CLR	R/W	0h	设置此位以清除统计数据计数器。
30	STATS_EN	R/W	0h	设置该位以启用统计数据计数器。
29	DYN_CG_EN	R/W	0h	0：内置时钟门控功能将被旁路。 1：内置时钟门控功能已启用，可在不需要时自动门控时钟。
28-6	RESERVED	R	0h	保留
5	IDCODE_LB_DIS	R/W	0h	Icode/Dcode 闪存行缓冲器控制 0：启用 1：禁用
4	SYSBUS_LB_DIS	R/W	0h	Sysbus 闪存行缓冲器控制 0：启用 1：禁用
3	ARB_CFG	R/W	0h	Icode/Dcode 和 sysbus 仲裁方案 0：静态仲裁 (icode/docde > sysbus) 1：轮询仲裁
2	PREF_EN	R/W	0h	标签预取控制 0：禁用 1：启用
1-0	模式	R/W	0h	VIMS 模式请求。 当 STAT.MODE_CHANGING 设置为 1 时，将阻止对该字段的写入访问。 0h = GPRAM：VIMS GPRAM 模式 1h = CACHE： VIMS 缓存模式 3h = VIMS 关闭模式

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

章节 9

SRAM

本章探讨了 CC13x2 和 CC26x2 器件平台的 RAM 系统。

9.1 简介.....780
9.2 主要特性.....780
9.3 数据保留.....780
9.4 奇偶校验和 SRAM 错误支持.....780
9.5 SRAM 自动初始化.....780
9.6 奇偶校验调试行为.....780
9.7 SRAM 寄存器.....780

Copyright © 2026 Texas Instruments Incorporated

9.1 简介

CC13x2 和 CC26x2 器件平台提供 80kB 的系统 RAM，由单周期片上 SRAM 组成。该 SRAM 在待机电源模式下支持完全数据保持。

支持位带操作，以减少对存储器的读取-修改-写入 (RMW) 操作的执行时间。通过位带操作，存储器映射中的某些区域（SRAM 和外设空间）可以使用地址别名在一个原子操作中访问单个位。

9.2 主要特性

SRAM 的主要特性为：

- 待机功耗模式下的可配置数据保留
- 奇偶校验错误检测
- 自动初始化函数

9.3 数据保留

在待机电源模式下，存储在 SRAM 中的数据会被保留。可通过 AON_PMCTL:RAM_CFG 寄存器按存储体配置 SRAM 保留（请参阅章节 7）。默认情况下会保留所有存储器。

9.4 奇偶校验和 SRAM 错误支持

在 SRAM 写入期间，会为每个写入的字节计算并存储一个奇偶校验位。奇偶校验错误检测在 SRAM 读取操作期间按字节进行。这意味着存储器读取操作中任何字节上的奇偶校验错误都会导致检测到存储器数据错误。奇偶校验错误会导致处理器总线故障异常（请参阅节 5.1.2）。

在奇偶校验错误检测期间，读取的 SRAM 地址会捕获到 SRAM_MMR:PER_CHK 寄存器中。该奇偶校验错误地址存储为相对于 SRAM 存储器基址的偏移量。

9.5 SRAM 自动初始化

SRAM 可通过专用自动初始化硬件进行初始化。所有存储器位置均初始化为零，并且奇偶校验信息根据需要进行初始化。存储器初始化可确保不会因为从尚未由软件初始化的位置进行读取而生成奇偶校验错误。使用 SRAM_MMR:MEM_CTL 寄存器执行自动初始化。

备注

SRAM 在系统复位后的系统 CPU 引导序列期间自动初始化。

9.6 奇偶校验调试行为

提供了以下特性，以便调试和测试对因奇偶校验错误导致的奇偶校验错误检测和总线故障异常的处理。

- 可通过使用 SRAM_MMR:PER_CTL 寄存器强制生成 SRAM 奇偶校验错误。在从写入 SRAM_MMR:PER_DBG 寄存器的地址偏移处读取数据后，将观察到所生成的奇偶校验错误。
- 可通过设置 SRAM_MMR:PER_CTL.PER_DISABLE 位来禁用 SRAM_MMR:PER_CHK 寄存器中的状态更新。如果已知调试器软件读取未初始化的存储器位置会生成奇偶校验错误，调试器软件可以使用该特性。

9.7 SRAM 寄存器

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

9.7.1 SRAM_MMR 寄存器

表 9-1 列出了 SRAM_MMR 寄存器的存储器映射寄存器。表 9-1 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

表 9-1. SRAM_MMR 寄存器

偏移	首字母缩写词	寄存器名称	部分
0h	PER_CTL	奇偶校验错误控制	PER_CTL 寄存器（偏移 = 0h）[复位 = 00000000h]
4h	PER_CHK	奇偶校验错误检查	PER_CHK 寄存器（偏移 = 4h）[复位 = 00000000h]
8h	PER_DBG	奇偶校验错误调试	PER_DBG 寄存器（偏移 = 8h）[复位 = 00000000h]
Ch	MEM_CTL	存储器控制	MEM_CTL 寄存器（偏移 = Ch）[复位 = 00000000h]

复杂的位访问类型经过编码可适应小型表单元。表 9-2 展示了适用于此部分中访问类型的代码。

表 9-2. SRAM_MMR 访问类型代码

访问类型	代码	说明
读取类型		
R	R	读取
写入类型		
W	W	写入
复位或默认值		
-n		复位后的值或默认值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

9.7.1.1 PER_CTL 寄存器（偏移 = 0h）[复位 = 00000000h]

图 9-1 展示了 PER_CTL，表 9-3 中对此进行了介绍。

返回到汇总表。

奇偶校验错误控制
奇偶校验错误检查控制

图 9-1. PER_CTL 寄存器

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

表 9-3. PER_CTL 寄存器字段说明

位	字段	类型	复位	说明
31-9	RESERVED	R	0h	保留
8	PER_DISABLE	R/W	0h	奇偶校验状态禁用 0：奇偶校验错误将更新 PER_CHK.PER_ADDR 字段 1：奇偶校验错误不更新 PER_CHK.PER_ADDR 字段
7-1	RESERVED	R/W	0h	软件不应依赖保留位的值。写入除复位值外的任何其他值都可能导致未定义的行为。
0	PER_DEBUG_ENABLE	R/W	0h	奇偶校验错误调试启用 0：正常运行 1：可以向 PER_DBG.PER_DEBUG_ADDR 写入一个地址偏移，对该偏移范围内的读取操作将产生奇偶校验错误

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

9.7.1.2 PER_CHK 寄存器（偏移 = 4h）[复位 = 00000000h]

图 9-2 展示了 PER_CHK，表 9-4 中对此进行了介绍。

返回到汇总表。

奇偶校验错误检查
奇偶校验错误检查结果

图 9-2. PER_CHK 寄存器

31	30	29	28	27	26	25	24	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								PER_ADDR																							
R-0h								R-0h																							

表 9-4. PER_CHK 寄存器字段说明

位	字段	类型	复位	说明
31-24	RESERVED	R	0h	保留
23-0	PER_ADDR	R	0h	奇偶校验错误地址偏移 返回在 SRAM 读取期间导致奇偶校验错误的最后一个地址偏移。返回的地址偏移始终是包含奇偶校验错误位置的字对齐地址。对于非字对齐访问的奇偶校验故障，CPU_SCS:BFAR.ADDRESS 将保存导致奇偶校验错误的位置地址。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

9.7.1.3 PER_DBG 寄存器（偏移 = 8h）[复位 = 00000000h]

图 9-3 展示了 PER_DBG，表 9-5 中对此进行了介绍。

返回到汇总表。

奇偶校验错误调试
奇偶校验错误检查调试地址设置

图 9-3. PER_DBG 寄存器

31	30	29	28	27	26	25	24	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								PER_DEBUG_ADDR																							
R-0h								R/W-0h																							

表 9-5. PER_DBG 寄存器字段说明

位	字段	类型	复位	说明
31-24	RESERVED	R	0h	保留
23-0	PER_DEBUG_ADDR	R/W	0h	调试奇偶校验错误地址偏移 当 PER_CTL.PER_DEBUG 为 1 时，该字段用于设置调试奇偶校验地址偏移。该地址偏移必须是一个字对齐地址。在该地址偏移范围内写入，会强制将不正确的奇偶校验位与写入的数据一起存储。随后在同一地址偏移范围内的读取操作将因此产生奇偶校验错误。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

9.7.1.4 MEM_CTL 寄存器（偏移 = Ch）[复位 = 00000000h]

图 9-4 展示了 MEM_CTL，表 9-6 中对此进行了介绍。

返回到汇总表。

存储器控制
控制存储器初始化

图 9-4. MEM_CTL 寄存器

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

表 9-6. MEM_CTL 寄存器字段说明

位	字段	类型	复位	说明
31-2	RESERVED	R	0h	保留
1	MEM_BUSY	R	0h	存储器忙碌状态 0：存储器接受传输 1：存储器控制器在初始化期间忙碌。未执行读写传输。
0	MEM_CLR_EN	R/W	0h	存储器内容初始化启用 向 MEM_CLR_EN 写入 1 将启动存储器初始化。所有字节位置的内容都将初始化为 0x00。存储器初始化完成之前，MEM_BUSY 将为 1。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

9.7.2 SRAM 寄存器

表 9-7 列出了 SRAM 寄存器的存储器映射寄存器。表 9-7 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

表 9-7. SRAM 寄存器

偏移	首字母缩写词	寄存器名称	部分
0h + 公式	BANK0_y	16k SRAM	BANK0_y 寄存器 (偏移 = 0h + 公式) [复位 = 00000000h]
4000h + 公式	BANK1_y	16k SRAM	BANK1_y 寄存器 (偏移 = 4000h + 公式) [复位 = 00000000h]
8000h + 公式	BANK2_y	16k SRAM	BANK2_y 寄存器 (偏移 = 8000h + 公式) [复位 = 00000000h]
C000h + 公式	BANK3_y	16k SRAM	BANK3_y 寄存器 (偏移 = C000h + 公式) [复位 = 00000000h]
00010000h + 公式	BANK4_y	16k SRAM	BANK4_y 寄存器 (偏移 = 00010000h + 公式) [复位 = 00000000h]

复杂的位访问类型经过编码可适应小型表单元。表 9-8 展示了适用于此部分中访问类型的代码。

表 9-8. SRAM 访问类型代码

访问类型	代码	说明
读取类型		
R	R	读取
写入类型		
W	W	写入
复位或默认值		
-n		复位后的值或默认值
寄存器数组变量		
i、j、k、l、m、n		当这些变量用于寄存器名称、偏移或地址时，它们指的是寄存器数组的值，其中寄存器是一组重复寄存器的一部分。寄存器组构成分层结构，数组用公式表示。
y		当该变量用于寄存器名称、偏移或地址时，它指的是寄存器数组的值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

9.7.2.1 BANK0_y 寄存器（偏移 = 0h + 公式）[复位 = 00000000h]

图 9-5 展示了 BANK0_y，表 9-9 对其进行了介绍。

返回到汇总表。

16k SRAM

偏移 = 0h + (y * 4h)；其中 y = 0h 至 FFFh

图 9-5. BANK0_y 寄存器

31	30	29	28	27	26	25	24	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-X																															

表 9-9. BANK0_y 寄存器字段说明

位	字段	类型	复位	说明
31-0	DATA	R/W	X	SRAM 数据

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

9.7.2.2 BANK1_y 寄存器（偏移 = 4000h + 公式）[复位 = 00000000h]

图 9-6 展示了 BANK1_y，表 9-10 对其进行了介绍。

返回到汇总表。

16k SRAM

偏移 = 4000h + (y * 4h)；其中 y = 0h 至 FFFh

图 9-6. BANK1_y 寄存器

31	30	29	28	27	26	25	24	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-X																															

表 9-10. BANK1_y 寄存器字段说明

位	字段	类型	复位	说明
31-0	DATA	R/W	X	SRAM 数据

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

9.7.2.3 BANK2_y 寄存器（偏移 = 8000h + 公式）[复位 = 00000000h]

图 9-7 展示了 BANK2_y，表 9-11 对其进行了介绍。

返回到汇总表。

16k SRAM

偏移 = 8000h + (y * 4h)；其中 y = 0h 至 FFFh

图 9-7. BANK2_y 寄存器

31	30	29	28	27	26	25	24	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-X																															

表 9-11. BANK2_y 寄存器字段说明

位	字段	类型	复位	说明
31-0	DATA	R/W	X	SRAM 数据

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

9.7.2.4 BANK3_y 寄存器（偏移 = C000h + 公式）[复位 = 00000000h]

图 9-8 展示了 BANK3_y，表 9-12 对其进行了介绍。

返回到汇总表。

16k SRAM

偏移 = C000h + (y * 4h)；其中 y = 0h 至 FFFh

图 9-8. BANK3_y 寄存器

31	30	29	28	27	26	25	24	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-X																															

表 9-12. BANK3_y 寄存器字段说明

位	字段	类型	复位	说明
31-0	DATA	R/W	X	SRAM 数据

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

9.7.2.5 BANK4_y 寄存器（偏移 = 00010000h + 公式）[复位 = 00000000h]

图 9-9 展示了 BANK4_y，表 9-13 对其进行了介绍。

返回到汇总表。

16k SRAM

偏移 = 00010000h + (y * 4h)；其中 y = 0h 至 FFFh

图 9-9. BANK4_y 寄存器

31	30	29	28	27	26	25	24	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-X																															

表 9-13. BANK4_y 寄存器字段说明

位	字段	类型	复位	说明
31-0	DATA	R/W	X	SRAM 数据

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

章节 10

引导加载程序

本节介绍 CC13x2 和 CC26x2 器件平台中包含的引导加载程序。

10.1 引导加载程序功能 793
10.2 引导加载程序接口 793

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

10.1 引导加载程序功能

CC13x2 和 CC26x2 器件平台包含一个简单的、基于 ROM 的引导加载程序，可通过 UART0 和 SSI0 外设上的串行接口与外部器件通信。两个串行接口使用相同的通信协议。这些外设是来自 Arm®的 IP。

ROM 引导加载程序的主要目的是为下载闪存映像的功能提供支持。

10.1.1 引导加载程序禁用

ROM 引导加载程序支持可读取闪存映像的命令。由于这种读取功能，实施了一种禁用引导加载程序的安全措施。如果使用 CCFG BOOTLOADER_ENABLE 参数禁用引导加载程序，则引导加载程序无法执行任何命令，这可以防止攻击者在 Arm® Cortex®-M4F 处理器的程序计数器 (PC) 被迫从引导加载程序代码执行时使用引导加载程序。

在 TI 分布式软件中，CCFG 参数在编译时在 ccfg.c 文件中设置。CCFG BOOTLOADER_ENABLE 参数通过 SET_CCFG_BL_CONFIG_BOOTLOADER_ENABLE（位于 ccfg.c 中）的值配置。

备注

即使引导加载程序被禁用，它仍然可以执行 CMD_GET_STATUS 命令。这样可验证 CMD_DOWNLOAD_CRC 命令是否已正确执行（即使下载的闪存映像包含禁用引导加载程序的 CCFG 数据）。如果在禁用引导加载程序时向器件发送了 CMD_GET_STATUS 以外的任何命令，即使是以下 CMD_GET_STATUS 命令，引导加载程序也会停止响应。

10.1.2 引导加载程序后门

为了在闪存中存在有效镜像时也能进入 ROM 引导加载程序，实现了一个引导加载程序后门。CCFG 参数 BL_ENABLE 可以启用此后门。后门功能使用一个可配置的 I/O 引脚（CCFG 参数 BL_PIN_NO）和一个可配置的 I/O 引脚电平（CCFG 参数 BL_LEVEL）。

如果启用了后门功能，复位时在可配置的 I/O 引脚上从外部施加一个可配置的信号电平，可以强制进入 ROM 引导加载程序。如果后门被启用并且存在有效的闪存镜像，启动代码会检查该 I/O 引脚的电平。如果配置的 I/O 引脚电平与配置的信号电平匹配，则 ROM 引导加载程序不会将控制权转交给闪存映像。

如果后门引脚配置与 UART0 或 SSI0 引脚之一匹配，则外部用户必须在通过 UART0 或 SSI0 接口传输数据之前，将后门信号置为无效。

在 TI 分发的软件中，称为 BL_ENABLE、BL_PIN_NO 和 BL_LEVEL 的 CCFG 参数由 ccfg.c 文件中以下宏定义的值进行配置：

- SET_CCFG_BL_CONFIG_BL_ENABLE
- SET_CCFG_BL_CONFIG_BL_PIN_NO
- SET_CCFG_BL_CONFIG_BL_LEVEL

节 11.2.1 展示了 BL_BACKDOOR_CONFIG 参数在 CCFG 中的布局。

备注

当使用引导加载程序后门功能时，在检查后门电平期间，配置为后门 (BL_PIN_NO) 的引脚将被配置为使能一个与所配置的 I/O 引脚电平相反的上拉/下拉电平。

10.2 引导加载程序接口

引导加载程序通过 2 引脚 UART 或 4 引脚 SSI 接口与外部器件进行通信。以下各部分介绍了通信协议层和传输层。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

10.2.1 数据包处理

引导加载程序使用定义明确的数据包来确保与外部通信程序可靠地进行通信。所有通信（UART 自动波特率除外 [请参阅节 10.2.2.1]）都使用这些定义明确的数据包。数据包始终由通信器件通过定义的 ACK 或 NACK 字节进行确认或者未确认。

在接收和发送数据包时使用相同的格式。此格式包括确认成功接收或未成功接收数据包的方式。

虽然串行端口上的实际信号不同，但对于支持的 UART 和 SSI 接口，数据包格式保持不变。

数据包发送和数据包接收必须遵守如图 10-1 所示的简单协议。

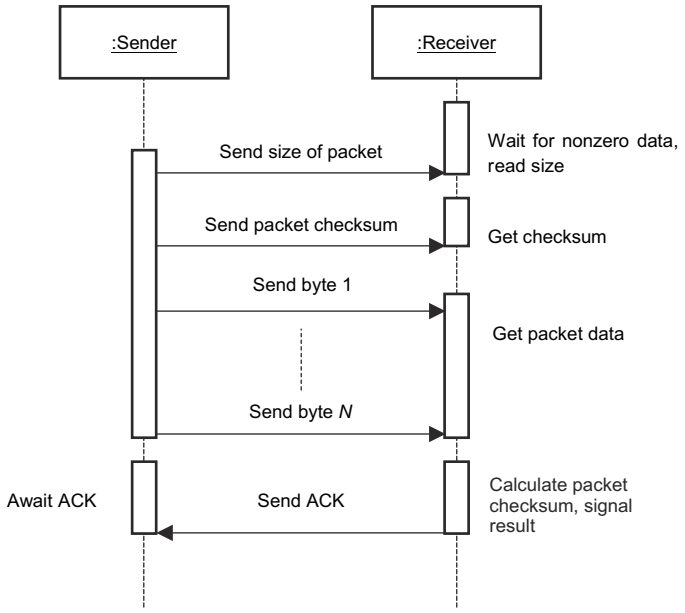


图 10-1. 发送和接收协议的序列图表

执行以下步骤以成功发送数据包：

1. 发送要发送到器件的数据包大小。大小始终为数据的大小 + 2，截断为 8 位。
2. 发送数据缓冲器校验和以确保命令正确传输。校验和算法是数据字节的总和。
3. 发送实际数据字节。
4. 等待器件发出单字节确认，确认已正确接收数据或检测到传输错误。

请执行以下步骤来成功接收数据包：

1. 等待从器件返回非零数据。这一点很重要，因为器件可能会在发送的数据包和接收的数据包之间发送零字节。接收到的第一个非零字节是接收到的数据包大小。
2. 读取下一个字节，即数据包的校验和。
3. 从器件读取数据字节。在数据阶段，发送数据包大小减去 2 个字节。例如，如果数据包大小为 3，则只接收 1 个字节的数据。
4. 计算数据字节的校验和并验证其是否同数据包中接收到的校验和匹配。
5. 向器件发送一个确认字节或一个非确认字节，以指示成功接收到数据包或未成功接收到数据包。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

每当接收器成功接收并验证数据包时，就会发送确认 (ACK) 字节。每当检测到发送的数据包存在错误时，就会发送一个非确认 (NAK) 字节，该错误通常是由于校验和错误或仅是数据包中格式错误的数据导致的，这允许发送方重新传输之前的数据包。

为了说明数据包处理，基本数据包格式如图 10-2 所示。

在图 10-2 中，最上面一行显示正在传输数据的设备；最下面一行是来自另一台设备的响应。

在这种情况下，发送一个 6 字节的数据包，数据如图 10-2 中所示。该数据产生校验和 0x48+0x6f+0x6c+0x61，当截断为 8 位时，该校验和为 0x84。传输的第一个字节保存数据包的大小（以字节数表示）。然后传输校验和字节。要输出的下一个字节是该数据包中的 4 个数据字节。发送器可以发送零，直到接收到非零响应，这是 SSI 所必需的、并由 UART 允许。接收器被允许返回零，直到它准备好对正在发送的数据包进行 ACK 或 NAK 操作。在传输数据包后接收到响应之前，两个器件都不会传输非零字节。

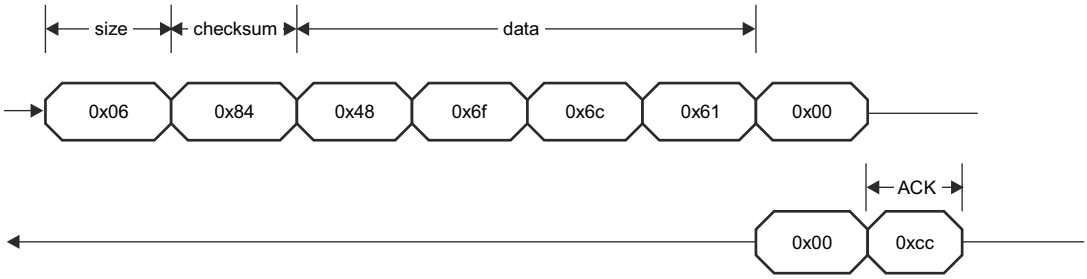


图 10-2. 串行总线数据包格式

10.2.1.1 数据包确认和非确认字节

表 10-1 展示了数据包确认 (ACK) 和非确认 (NAK) 字节的定义值。

表 10-1. 协议确认和非确认字节

协议字节	值
ACK	0xCC
NACK	0x33

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

10.2.2 传输层

引导加载程序支持通过 UART0 和 SSI0 端口进行更新，这些端口在 CC13x2 和 CC26x2 器件平台上提供。SSI0 端口的优点是支持更高、更灵活的数据速率，但它也要求与 CC13x2 和 CC26x2 器件平台之间具有更多的连接。UART0 的缺点是速率略低且可能不太灵活。然而，UART0 需要的引脚较少，并且可以通过任何标准 UART 接口轻松实现。

表 10-2 指定将哪些串行接口信号配置为特定 DIO。这些引脚是固定的，无法重新配置。

表 10-2. 串行接口的配置

信号	CC26x2R	CC1312R	CC1352x
UART0_RX	DIO2	DIO2	DIO12
UART0_TX	DIO3	DIO3	DIO13
SSI0_CLK	DIO10	DIO10	DIO10
SSI0_FSS	DIO11	DIO11	DIO11
SSI0_RX	DIO9	DIO9	DIO9
SSI0_TX	DIO8	DIO8	DIO8

引导加载程序最初仅配置两个串行接口上的输入引脚。默认情况下，所有 I/O 引脚都禁用它们的输入缓冲器，因此引导加载程序将所需的引脚配置为输入引脚，以便在此时之前无法从主机访问引导加载程序接口。对于输入引脚的这种初始配置，固件会配置 IOC，以将表 10-2 中列出的输入信号路由到其相应的外设信号。

引导加载程序会选择第一个由外部器件访问的接口。选择后，配置所选接口的 TX 输出引脚；禁用非活动接口（UART0 或 SSI0）上的模块。若要切换到另一个接口，必须复位 CC13x2 和 CC26x2 器件平台。TX 引脚的延迟配置在 SSI0 主器件上对第一个数据包的第一个字节的传输有特别的考虑（请参阅节 10.2.2.2）。

10.2.2.1 UART 传输

使用 UART 端口所需的连接具有以下两个引脚：UART0 TX 及 UART0 RX。与引导加载程序通信的器件会驱动 CC13x2 和 CC26x2 器件平台上的 UART0 RX 引脚，而 CC13x2 和 CC26x2 器件平台会驱动 UART0 TX 引脚。

虽然波特率很灵活，但 UART 串行格式固定为 8 个数据位，无奇偶校验位和 1 个停止位。引导加载程序会自动检测通信的波特率。

10.2.2.1.1 UART 波特率自动检测

引导加载程序提供了一种自动检测用于与其通信的 UART 波特率的方法。

若要与主机同步，引导加载程序必须接收值为 0x55 的 2 个字节。如果同步成功，引导加载程序将返回一个由 2 个字节组成的确认，其值为 0x00 和 0xCC。

如果同步失败，引导加载程序会等待同步尝试。

在自动检测功能中，使用 GPIO 中断来监测 UART0 RX 引脚是否存在边沿。当检测到足够的边沿时，引导加载程序会确定对 UART 进行编程所需的波特率和频率之比。

UART 模块系统时钟必须至少为波特率的 16 倍；因此，最大波特率不能高于 3Mbaud（48MHz 除以 16）。由于固件功能能够检测主机的传输速率，因此最大波特率限制为 1.6Mbaud。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

10.2.2.2 SSI 传输

使用 SSI 端口所需的连接为以下四个引脚：

- SSI0_TX
- SSI0_RX
- SSI0_CLK
- SSI0_FSS

与引导加载程序通信的器件驱动 FSS 引脚（SSI0_RX、SSI0_CLK 和 SSI0），而 CC13x2 和 CC26x2 器件平台驱动 SSI0_TX 引脚。

SSI 通信使用的格式是 Motorola 格式，其中 SPH 设置为 1，SPO 设置为 1（有关此格式的更多信息，请参阅图 22-9）。SSI 接口有一项硬件要求，将 SSI 时钟的最大速率限制为 SSI 模块时钟频率的最多 1/12 (48MHz / 12 = 4MHz)。

必须特别注意主器件（关于 SSI0 接口的使用），因为在外部主器件选择串行接口之前，不允许配置任何输出引脚。

备注

在主器件传输的第一个数据包上，当引导加载程序输出数据包第一个字节中的位时，不会从引导加载程序接收到数据。

当引导加载程序检测到在 SSI0_RX 上接收到 1 个字节时，引导加载程序会配置 SSI0_TX 输出引脚。

在传输第一个数据包中的下一个字节之前，主器件必须存在较小延迟，以确保引导加载程序已完成 SSI0_TX 输出引脚的配置。

10.2.3 串行总线命令

表 10-3 列出了 UART0 和 SSI0 引导加载程序接口上的自定义协议所支持的命令。

每条命令在协议数据包内传输。包内的前 2 个字节是大小字节，其后是校验和字节。第三个字节保存用于标识命令的命令值；所有支持命令的命令值列于表 10-3 的*命令值*列中。包中剩余的字节是命令参数。有关每条命令的命令字节和参数字节的完整说明，请参阅节 10.2.3.1 至节 10.2.3.12。

以下小节指定了每条命令在协议包内的各个字节。

表 10-3. 支持的引导加载程序命令

命令	命令值	包中字节数	说明
COMMAND_PING	0x20	3	从引导加载程序接收确认信号，表示通信已建立。
COMMAND_DOWNLOAD	0x21	11	准备闪存编程。指定将数据编程到闪存中的起始位置，以及后续 COMMAND_SEND_DATA 命令将要发送的字节数。
COMMAND_GET_STATUS	0x23	3	返回最后发出的命令的状态。通常，引导加载程序必须在每条命令发送后收到此命令，才能确保前一条命令已成功。有关已定义的状态值，请参阅表 10-4。状态在一个 3 字节的协议数据包中返回。
COMMAND_SEND_DATA	0x24	4 至 255	传输数据并编程闪存。传输编程到闪存中 COMMAND_DOWNLOAD 命令或另一条 COMMAND_SEND_DATA 命令之后的数据。 要编程到闪存中的数据字节数可为 1 至 252（数据包中的最大数据负载）。 如果 COMMAND_SEND_DATA 命令下载的数据量超过了 COMMAND_DOWNLOAD 命令指定的字节数，将生成错误状态。
COMMAND_RESET	0x25	3	执行系统复位。相关详细信息，请参阅节 7.7.1.2。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 10-3. 支持的引导加载程序命令（续）

命令	命令值	包中字节数	说明
COMMAND_SECTOR_ERASE	0x26	7	擦除主闪存组中的一个扇区。要擦除的扇区由扇区起始地址指定。仅擦除未被 FCFG1 和 CCFG 中的写保护位保护的闪存扇区。 如果选择了顶部扇区（包含 CCFG），CCFG 的内容将复位为 TI 出厂器件时的值。
COMMAND_CRC32	0x27	15	计算指定存储器区域上的 CRC32。指定每个存储器位置的读取次数。
COMMAND_GET_CHIP_ID	0x28	3	从 AON_PMCTL JTAGUSERCODE 寄存器中返回 32 位的 USER CODE，MSB 在前。ID 在协议数据包中返回。
COMMAND_MEMORY_READ	0x2A	9	从指定的存储器映射起始地址，以指定的访问宽度（8 位或 32 位）读取指定数量的元素。请求的数据量必须小于通信数据包的最大大小。
COMMAND_MEMORY_WRITE	0x2B	9 至 255	从指定的存储器映射起始地址，以指定的宽度（8 位或 32 位）写入收到的数据。要写入的数据必须与命令包含在同一个数据包中。
COMMAND_BANK_ERASE	0x2C	3	对所有不受 FCFG1 和 CCFG 写保护位保护的客户可访问闪存扇区执行擦除。如果 CCFG 参数 BANK_ERASE_DIS 被清除，则不执行擦除操作。 由于顶部扇区（包含 CCFG）可能被擦除，CCFG 的内容将复位为 TI 出厂器件时的值。
COMMAND_SET_CCFG	0x2D	11	使用此命令数据字节中收到的值，将 CC13x2 和 CC26x2 定义的 CCFG 字段写入闪存 CCFG 区域。 此命令使用户无需详细了解闪存 CCFG 内保存所定义 CCFG 字段的物理地址。
COMMAND_DOWNLOAD_CRC	0x2F	15	使用指定的 CRC32 值准备闪存编程。指定将数据编程到主闪存组中的起始位置、后续 COMMAND_SEND_DATA 命令将要发送的字节数，以及覆盖待编程总字节数的 32 位 CRC 值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

10.2.3.1 COMMAND_PING

COMMAND_PING 命令接收来自引导加载程序的确认，表明已建立通信。此命令是单字节。

包括该命令在内的数据包格式如下：

```
unsigned char ucCommand[3];
     ucCommand[0] = <size=3>;
     ucCommand[1] = <checksum>;
     ucCommand[2] = COMMAND_PING;
```

10.2.3.2 COMMAND_DOWNLOAD

COMMAND_DOWNLOAD 命令被发送到引导加载程序，以指示闪存中存储数据的位置以及随后的 COMMAND_SEND_DATA 命令将发送多少个字节。该命令包含两个 32 位值，这两个值都以 MSB 优先的方式传输。第一个 32 位值是在其中开始编程数据的地址，而第二个是将发送的数据的 32 位大小。此命令必须后跟 COMMAND_GET_STATUS 命令，以确保程序地址和程序大小对器件有效。在 CC13x2 和 CC26x2 器件上，闪存从地址 0x0000 0000 开始。该命令不执行任何类型的擦除操作；它仅准备由 COMMAND_SEND_DATA 命令执行的以下闪存编程。可以通过 COMMAND_BANK_ERASE 和 COMMAND_SECTOR_ERASE 命令来完成所需的闪存擦除。

包括该命令在内的数据包格式如下：

```
unsigned char ucCommand[11];
     ucCommand[0] = <size=11>;
     ucCommand[1] = <checksum>;
     ucCommand[2] = COMMAND_DOWNLOAD;
     ucCommand[3] = Program Address [31:24];
     ucCommand[4] = Program Address [23:16];
     ucCommand[5] = Program Address [15:8];
     ucCommand[6] = Program Address [7:0];
     ucCommand[7] = Program Size [31:24];
     ucCommand[8] = Program Size [23:16];
     ucCommand[9] = Program Size [15:8];
     ucCommand[10] = Program Size [7:0];
```


English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

10.2.3.3 COMMAND_SEND_DATA

COMMAND_SEND_DATA 命令必须仅在 COMMAND_DOWNLOAD 命令之后发出，如果需要更多数据则在另一个 COMMAND_SEND_DATA 命令之后发出。连续的 COMMAND_SEND_DATA 命令会自动递增地址并从先前的位置继续编程。

当接收到由 COMMAND_DOWNLOAD 命令指示的字节数时，该命令终止编程。

引导加载程序在实际编程完成后发送 ACK 以响应该命令。每次调用此函数时，都应输入一条 COMMAND_GET_STATUS 命令，以确保数据已成功编程到闪存中。如果引导加载程序对该命令发送 NAK 信号，则引导加载程序不会递增当前地址，这允许重新传输先前的数据。

包含该命令的数据包格式如下：

```
unsigned char ucCommand[4-255];
     ucCommand[0] = <size>;
     ucCommand[1] = <checksum>;
     ucCommand[2] = COMMAND_SEND_DATA;
     ucCommand[3] = Data byte to be programmed[0];
     ucCommand[4] = Data byte to be programmed[1];
     ucCommand[5] = Data byte to be programmed[2];
     ucCommand[6] = Data byte to be programmed[3];
     ucCommand[7] = Data byte to be programmed[4];
     ucCommand[<size-1>] = Data byte to be programmed[<size-4>];
```

10.2.3.4 COMMAND_SECTOR_ERASE

COMMAND_SECTOR_ERASE 命令用于擦除指定的闪存扇区。一个闪存扇区的大小为 8kB。

该命令包含一个 32 位值，该值首先传输 MSB。32 位值是要擦除的闪存扇区的起始地址。

在执行实际擦除操作后，引导加载程序会向主机器件发送 ACK 信号。

在 CC13x2 和 CC26x2 器件平台上，闪存从地址 0x0000 0000 开始，每个扇区为 8kB。

备注

不会擦除 FCFG1 及 CCFG 中由写保护位保护的扇区。

如果指定了顶部扇区（包括 CCFG 区域）的扇区地址，则实际擦除之后会将 CCFG 值编程为与器件从 TI 交付时所具备值相同的值。

包括该命令在内的数据包格式如下：

```
unsigned char ucCommand[7];
     ucCommand[0]= <size=7>;
     ucCommand[1]= <checksum>;
     ucCommand[2]= COMMAND_ERASE;
     ucCommand[3]= Sector Address [31:24];
     ucCommand[4]= Sector Address [23:16];
     ucCommand[5]= Sector Address [15: 8];
     ucCommand[6]= Sector Address [ 7: 0];
```

10.2.3.5 COMMAND_GET_STATUS

COMMAND_GET_STATUS 命令会返回最后发出的命令的状态。通常，该命令会在每发送一条其他命令后被接收，以确保上一条命令成功；或者，如果命令失败，则正确响应失败。引导加载程序通过发送一个 3 字节数据包进行响应，其中包含当前状态值的大小字节、校验和字节和 1 字节。

然后，引导加载程序等待来自主机的 ACK，以确认已接收到数据包。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

包括该命令在内的数据包格式如下：

```
unsigned char ucCommand[3];
     ucCommand[0] = <size=3>;
     ucCommand[1] = <checksum>;
     ucCommand[2] = COMMAND_GET_STATUS;
```

表 10-4 列出了将 COMMAND_GET_STATUS 命令发送到引导加载程序时，可从引导加载程序返回的可能状态值的定义

表 10-4. 定义的状态值

状态定义	值	说明
COMMAND_RET_SUCCESS	0x40	成功执行的命令的状态
COMMAND_RET_UNKNOWN_CMD	0x41	未知命令的状态
COMMAND_RET_INVALID_CMD	0x42	无效命令的状态（换而言之，即数据包大小不正确）
COMMAND_RET_INVALID_ADR	0x43	无效输入地址的状态
COMMAND_RET_FLASH_FAIL	0x44	未完成闪存擦除或编程操作的状态

10.2.3.6 COMMAND_RESET

COMMAND_RESET 命令指示引导加载程序执行系统复位。将新闪存映像下载到 CC13x2 和 CC26x2 器件后，可使用此命令让新应用程序从复位状态开始运行。将发生正常的启动序列，闪存映像的运行效果就如同由硬件复位触发一样。此外，如果发生严重错误，并且主机器件希望重新启动与引导加载程序的通信，也可以使用此命令来复位引导加载程序。

在实际执行系统复位之前，引导加载程序会向主机器件回复一个 ACK 信号。此 ACK 信号通知更新应用程序已成功收到命令，随后 CC13x2 和 CC26x2 器件将复位。

包含该命令的数据包格式如下：

```
unsigned char ucCommand[3];
     ucCommand[0] = <size=3>;
     ucCommand[1] = <checksum>;
     ucCommand[2] = COMMAND_RESET;
```


English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

10.2.3.7 COMMAND_GET_CHIP_ID

COMMAND_GET_CHIP_ID 命令使引导加载程序从 AON_PMCTL JTAGUSERCODE 寄存器返回 32 位用户 ID 的值。引导加载程序首先通过发送 ACK 信号来响应命令；然后引导加载程序发送一个包含大小字节、校验和字节以及 4 个字节（MSB 优先）用户 ID 的数据包，共 6 个字节。

然后，引导加载程序等待来自主机的 ACK 信号，作为对已收到数据包的确认。

命令格式如下：

```
unsigned char ucCommand[3];
     ucCommand[0] = <size=3>;
     ucCommand[1] = <checksum>;
     ucCommand[2] = COMMAND_GET_CHIP_ID;
```

10.2.3.8 COMMAND_CRC32

COMMAND_CRC32 命令使用 CRC32 校验闪存区域。该命令由三个 32 位值组成，所有值均以 MSB 在前的方式传输。第一个 32 位值是存储器中的 CRC32 计算起始地址，第二个 32 位值是 CRC32 计算所包含的字节数，第三个 32 位值是每个数据位置的读取重复次数。读取重复计数为 0x0000 0000 时，将仅通过一次读取所有数据位置来生成校验和。在实际 CRC32 计算之后，该命令会发送 ACK 信号作为对命令的响应。结果最终以 4 字节（MSB 在前）的形式通过 6 字节数据包返回。然后，引导加载程序等待来自主机的 ACK 信号，作为对已收到数据包的确认。保存字节数的第二个参数必须大于 8。否则，返回的校验和将为 0xFFFF FFFF。

包含该命令的数据包格式如下：

```
unsigned char ucCommand[15];
     ucCommand[0] = <size=15>;
     ucCommand[1] = <checksum>;
     ucCommand[2]= COMMAND_CRC32;
     ucCommand[3]= Data Address [31:24];
     ucCommand[4]= Data Address [23:16];
     ucCommand[5]= Data Address [15: 8];
     ucCommand[6]= Data Address [ 7: 0];
     ucCommand[7]= Data Size [31:24];
     ucCommand[8]= Data Size [23:16];
     ucCommand[9]= Data Size [15: 8];
     ucCommand[10]= Data Size [7: 0];
     ucCommand[11]= Read Repeat Count [31:24];
     ucCommand[12]= Read Repeat Count [23:16];
     ucCommand[13]= Read Repeat Count [15: 8];
     ucCommand[14]= Read Repeat Count [7: 0];
```


English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

10.2.3.9 COMMAND_BANK_ERASE

如果 CCFG 参数 BANK_ERASE_DIS 被清除，则 COMMAND_BANK_ERASE 命令不会执行任何擦除操作。如果未清除 COMMAND_BANK_ERASE，此命令将擦除所有主存储体闪存扇区，包括未受 FCFG1 和 CCFG 中写保护位保护的 CCFG。

在执行实际擦除操作后，该命令会发送 ACK 以响应命令。实际擦除之后会将 CCFG 值编程为与器件从 TI 交付时一样的值。

包括该命令在内的数据包格式如下：

```
unsigned char ucCommand[3];
     ucCommand[0] = <size=3>;
     ucCommand[1] = <checksum>;
     ucCommand[2] = COMMAND_BANK_ERASE;
```

10.2.3.10 COMMAND_MEMORY_READ

该命令从指定的存储器映射起始地址，以指定的访问类型（8 位或 32 位）读取指定数量的元素，并在单独的通信数据包中返回读取的数据。请求的数据量必须小于通信数据包的最大大小。指定的访问类型必须为 0 或 1。值 0 强制进行 8 位读取访问。值 1 强制进行 32 位读取访问。指定的访问次数给出 8 位或 32 位读取访问的次数。当访问类型 = 0 时，访问次数的最大值为 253。当访问类型 = 1 时，访问次数的最大值为 63。包含该命令的数据包格式如下：

```
unsigned char ucCommand[9];
     ucCommand[0] = <size=9>;
     ucCommand[1] = <checksum>;
     ucCommand[2] = COMMAND_MEMORY_READ;
     ucCommand[3] = Memory Map Address [31:24];
     ucCommand[4] = Memory Map Address [23:16];
     ucCommand[5] = Memory Map Address [15:8];
     ucCommand[6] = Memory Map Address [7:0];
     ucCommand[7] = Access Type [7:0];
     ucCommand[8] = Number of Accesses [7:0];
```


English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

10.2.3.11 COMMAND_MEMORY_WRITE

此命令用于从指定的存储器映射起始地址写入具有指定宽度（8 位或者 32 位）的访问中所接收数据。要写入的数据必须与命令包含在同一个数据包中。访问宽度由指定的访问类型给出。访问类型必须为 0 或者 1。值为 0 时强制进行 8 位写入访问。值为 1 时强制进行 32 位写入访问。接收到的数据字节数由数据包大小字节给出。对于访问宽度 0，最大数据字节数为 247，对于访问宽度 1 为 244。

不得使用 COMMAND_MEMORY_WRITE 命令写入特定存储器映射区域。存储器写入以下存储器映射区域可能会导致串行引导加载程序最终进入非功能状态，因为引导加载程序会使用这些区域。这对以下存储器映射区域有效：

- 较低的 4kB SRAM（0x2000 0000 到 0x2000 0FFF）
- 控制串行引导加载程序当前使用的串行接口（UART 或 SSI）功能的任何硬件寄存器。

备注

不能使用 COMMAND_MEMORY_WRITE 命令来写入闪存。

包括该命令在内的数据包格式如下：

```
unsigned char ucCommand[(from 9 to 255)];
     ucCommand[0] = <size=(from 9 to 255)>;
     ucCommand[1] = <checksum>;
     ucCommand[2] = COMMAND_MEMORY_WRITE;
     ucCommand[3] = Memory Map Address [31:24];
     ucCommand[4] = Memory Map Address [23:16];
     ucCommand[5] = Memory Map Address [15:8];
     ucCommand[6] = Memory Map Address [7:0];
     ucCommand[7] = Access Type [7:0];
     ucCommand[8] = Data [7:0];
     ...
     ...
     ucCommand[9 + (packet size - 9)] = Data [7:0] or Data[31:24];
```


English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

10.2.3.12 COMMAND_SET_CCFG

将 COMMAND_SET_CCFG 命令发送到引导加载程序，以配置闪存 CCFG 区域中由 ROM 引导固件读取的定义字段。在执行实际的闪存编程操作之后，该命令会发送 ACK 信号以响应命令。该命令在写入操作之前不执行任何擦除操作。

该命令包含两个 32 位值，这两个值都以 MSB 优先的方式传输。第一个 32 位值是 CCFG 字段 ID，用于标识要写入的 CCFG 参数，第二个 32 位值是要编程的字段值。命令处理程序会屏蔽掉与 CCFG 参数大小不对应的字段值位。

备注

COMMAND_SET_CCFG 命令只能将 CCFG 参数值位从 1 更改成 0。

尝试将任何位从 0 更改为 1 会导致错误状态，可通过以下 COMMAND_GET_STATUS 命令观察到该状态。

仅有在 ROM 引导期间控制器件配置的 CCFG 字段才能通过该命令写入。（从 BL_CONFIG 到结束的字段）。

要将 CCFG 参数值位从 0 更改为 1，唯一方法是擦除整个 CCFG 闪存扇区。在实际闪存编程终止后，该命令会发送 ACK 信号以响应命令。

如果 CCFG 区域（闪存顶部扇区）由 FCFG1 或 CCFG 中的保护位进行写保护，则编程操作会失败。

包括该命令在内的数据包格式如下：

```
unsigned char ucCommand[11];
     ucCommand[0] = <size=11>;
     ucCommand[1] = <checksum>;
     ucCommand[2] = COMMAND_SET_CCFG;
     ucCommand[3] = Field Id[31:24];
     ucCommand[4] = Field Id[23:16];
     ucCommand[5] = Field Id[15:8];
     ucCommand[6] = Field Id[7:0];
     ucCommand[7] = Field Value[31:24];
     ucCommand[8] = Field Value[23:16];
     ucCommand[9] = Field Value[15:8];
     ucCommand[10] = Field Value[7:0];
```


English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 10-5 中介绍了定义的 CCFG 字段 ID 以及相应的字段值。

表 10-5. 定义了 CCFG 字段 ID 和字段值

字段 ID	字段值	说明
0： ID_SECTOR_PROT	Bit[31:0] - 用来防止编程和擦除的闪存扇区的数量	CCFG 中对应于指定扇区编号的扇区写保护位设置为 0。这样可以保护该扇区不被编程及擦除。第一个闪存扇区的扇区编号为 0。 此命令还用于设置闪存包装程序寄存器中的粘滞扇区保护位。 请注意是否保护扇区 31，即 CCFG 扇区。如果扇区 31 受到保护，则无法设置其他 CCFG 参数。
1： ID_IMAGE_VALID	Bit[31:0] - 0x0000 0000	为了将执行控制传输到闪存映像的引导序列，此字段值必须设置为闪存映像矢量表的起始地址。
2： ID_TEST_TAP_LCK	Bit[31:8] – 不用考虑 Bit[7:0] – 0xC5 = TAP 已解锁	0xC5 以外的任何其他值都会在执行以下引导序列后强制执行锁定 TAP
3： ID_PWRPROF_TAP_LCK	Bit[31:8] – 不用考虑 Bit[7:0] – 0xC5 = TAP 已解锁	0xC5 以外的任何其他值都会在执行以下引导序列后强制执行锁定 TAP。
4： ID_CPU_DAP_LCK	Bit[31:8] – 不用考虑 Bit[7:0] – 0xC5 = DAP 已解锁	0xC5 以外的任何其他值都会在执行以下引导序列后强制执行锁定 DAP。
5： ID_AON_TAP_LCK	Bit[31:8] – 不用考虑 Bit[7:0] – 0xC5 = TAP 已解锁	0xC5 以外的任何其他值都会在执行以下引导序列后强制执行锁定 TAP。
6：ID_PBIST1_TAP_LCK	Bit[31:8] – 不用考虑 Bit[7:0] – 0xC5 = TAP 已解锁	0xC5 以外的任何其他值都会在执行以下引导序列后强制执行锁定 TAP。
7：ID_PBIST2_TAP_LCK	Bit[31:8] – 不用考虑 Bit[7:0] – 0xC5 = TAP 已解锁	0xC5 以外的任何其他值都会在执行以下引导序列后强制执行锁定 TAP。
8： ID_BANK_ERASE_DIS	Bit[31:1] – 不用考虑 Bit[0] – 0 = 组擦除禁用	如果设置为 0，COMMAND_BANK_ERASE 引导加载程序命令不会强制执行任何擦除操作。
9： ID_CHIP_ERASE_DIS	Bit[31:1] – 不用考虑 Bit[0] – 0 = 芯片擦除禁用	如果设置为 0，则无论任何芯片擦除请求如何，启动序列都不会执行任何芯片擦除操作。
10： ID_TI_FA_ENABLE	Bit[31:8] – 不用考虑 Bit[7:0] – 0xC5 = TI FA 启用	除 0xC5 以外的任何值都会在后续引导中禁用 TI FA 启用功能。
11： ID_BL_BACKDOOR_EN	Bit[31:8] – 不用考虑 Bit[7:0] – 0xC5 = 引导加载程序后门启用	除 0xC5 以外的任何其他值都会强制在后续引导序列中禁用引导加载程序后门。
12： ID_BL_BACKDOOR_PIN	Bit[31:8] – 不用考虑 Bit[7:0] – 引导加载程序后门 I/O 引脚编号	如果引脚编号超过器件上的 I/O 引脚数量，则会选择器件上最高的 I/O 引脚编号。
13： ID_BL_BACKDOOR_LEVEL	Bit[31:1] – 不用考虑 Bit[0] – 引导加载程序后门引脚有效电平	0 = 低电平有效
14： ID_BL_ENABLE	Bit[31:8] – 不用考虑 Bit[7:0] – 引导加载程序启用	除 0xC5 以外的任何值都会强制引导加载程序忽略任何接收到的命令。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

10.2.3.13 COMMAND_DOWNLOAD_CRC

COMMAND_DOWNLOAD_CRC 命令被发送到引导加载程序，以指示将数据编程到闪存中的位置、随后由 COMMAND_SEND_DATA 命令发送的字节数，以及覆盖要存储到 NVM 中的总数据字节数的预期 32 位 CRC 值。该命令由三个 32 位值组成，所有值都以 MSB 优先传输。第一个 32 位值是要开始编程数据的地址，第二个是要发送的数据的 32 位大小，第三个是覆盖要编程到闪存中的总字节数的 32 位 CRC 值。

此命令后应跟随 COMMAND_GET_STATUS 命令，以确保编程地址和编程大小对该器件有效。

闪存起始地址为 0x0000 0000。

COMMAND_DOWNLOAD_CRC 命令不执行任何类型的擦除操作；它仅准备后续由 COMMAND_SEND_DATA 命令执行的 NVM 编程。所需的闪存擦除可以通过 COMMAND_BANK_ERASE 和 COMMAND_SECTOR_ERASE 命令完成。对 COMMAND_DOWNLOAD_CRC 命令指定的总字节数的最后一个字节进行编程的 COMMAND_SEND_DATA 命令（在编程之后）将计算覆盖 COMMAND_DOWNLOAD_CRC 命令指定的闪存区域的 CRC32 值，并检查计算出的 CRC32 值是否等于指定的 32 位 CRC 值。如果这些值不相等，则在 COMMAND_SEND_DATA 命令响应 ACK 之前，整个已编程的闪存区域将被擦除。在这种情况下，后续的 COMMAND_GET_STATUS 命令将报告 COMMAND_RET_FLASH_FAIL 状态。

备注

由于在执行最后一个 COMMAND_SEND_DATA 命令期间需要计算 CRC 并可能执行闪存擦除操作，因此 COMMAND_DOWNLOAD_CRC 命令的 ACK 响应将出现延迟。

包含该命令的数据包格式如下：

```
unsigned char ucCommand[15];
     ucCommand[0] = <size=15>;
     ucCommand[1] = <checksum>;
     ucCommand[2] = COMMAND_DOWNLOAD_CRC;
     ucCommand[3] = Program Address [31:24];
     ucCommand[4] = Program Address [23:16];
     ucCommand[5] = Program Address [15:8];
     ucCommand[6] = Program Address [7:0];
     ucCommand[7] = Program Size [31:24];
     ucCommand[8] = Program Size [23:16];
     ucCommand[9] = Program Size [15:8];
     ucCommand[10] = Program Size [7:0];
     ucCommand[11] = Crc [31:24];
     ucCommand[12] = Crc [23:16];
     ucCommand[13] = Crc [15:8];
     ucCommand[14] = Crc [7:0];
```


English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

章节 11

器件配置

本章介绍了器件配置区域。出厂配置 (FCFG) 和客户配置 (CCFG) 区域位于闪存中。FCFG 由德州仪器 (TI) 在器件生产期间中设置，包含器件特定的校准值和配置。CCFG 必须由应用程序进行设置，其中包含 ROM 引导代码、器件硬件以及器件固件的配置参数。

11.1 客户配置 (CCFG) 809
11.2 CCFG 寄存器 809
11.3 出厂配置 (FCFG) 840
11.4 FCFG 寄存器 840

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.1 客户配置 (CCFG)

- 图像有效参数（通常由编程工具设置）
- 故障分析访问配置
- 自定义 MAC 地址
- 引导加载程序配置
- TAP 和 DAP 访问配置
- CC13x2 器件：将输出功率配置为 +14 dBm

在 TI 分布式软件中，CCFG 参数在编译时在 ccfg.c 文件中设置。CCFG 设置是默认设置的，以便对器件进行全面调试。不建议在生产中使用 CCFG 设置。

仅适用于 CC13x2 器件：

要启用 +14dBm 的输出功率，必须将 TI 在 cc13xxware 中分发的 ccfg.c 中的 CCFG_FORCE_VDDR_HH 定义设置为 1。如果 CCFG_FORCE_VDDR_HH 设置为 0，则可能的最大输出功率为 +12.5dBm。

TI 建议使用以下步骤配置器件以用于最终生产：

1. 必须将 ccfg.c 中的以下定义设置为 0x00，以禁止通过引导加载程序接口访问闪存内容。
   - SET_CCFG_BL_CONFIG_BOOTLOADER_ENABLE
   - SET_CCFG_BL_CONFIG_BL_ENABLE
2. 必须将 ccfg.c 中的 SET_CCFG_CCFG_TI_OPTIONS_TI_FA_ENABLE 定义设置为 0x00，以禁止 TI 进行失效分析访问。
3. ccfg.c 中的以下定义必须设置成 0x00 以单独禁止访问 JTAG 访问端口：
   - SET_CCFG_CCFG_TAP_DAP_0_PWRPROF_TAP_ENABLE
   - SET_CCFG_CCFG_TAP_DAP_0_TEST_TAP_ENABLE
   - SET_CCFG_CCFG_TAP_DAP_0_CPU_DAP_ENABLE
   - SET_CCFG_CCFG_TAP_DAP_1_AON_TAP_ENABLE
   - SET_CCFG_CCFG_TAP_DAP_1_PBIST1_TAP_ENABLE
   - SET_CCFG_CCFG_TAP_DAP_1_PBIST2_TAP_ENABLE
4. 若要使用 EnergyTrace™ 软件启用功耗性能分析，必须将 cfg.c 中的 SET_CCFG_CCFG_TAP_DAP_0_PWRPROF_TAP_ENABLE 定义设置为 0xC5。
5. ccfg.c 中的 SET_CCFG_IMAGE_VALID_CONF_IMAGE_VALID 定义必须设置为闪存映像矢量表的地址、以在启动时将控制权传递到闪存中编程的映像。大多数标准闪存映像的矢量表位于地址 0x0000 0000。
6. 或者，可以将 ccfg.c 中的 SET_CCFG_ERASE_CONF_CHIP_ERASE_DIS_N 定义设置为 0x0，以在 JTAG 请求芯片擦除时禁止擦除闪存。
7. 使用 ccfg.c 中的 SET_CCFG_CCFG_PROT_n 定义，对闪存中未设计为由最终产品在系统中更新的扇区进行编程和擦除保护。如果定义中的任何位设置为 0，则都会强制对相应的闪存扇区编号进行编程和擦除保护。

备注

在 CCFG 中的 ENABLE 字段中启用某些功能取决于 FCFG 中相应的 ENABLE 字段，该字段已由 TI 生产测试设置为启用。例如，TEST_TAP 访问端口的访问配置就是这种情况。在未在 FCFG 中启用 TEST_TAP 访问的产品中，由 ccfg.c 中的 SET_CCFG_CCFG_TAP_DAP_0_TEST_TAP_ENABLE 定义设置的相应 CCFG 字段中的值将被忽略，且功能将被禁用。

11.2 CCFG 寄存器

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.2.1 CCFG 寄存器

表 11-1 列出了 CCFG 寄存器的存储器映射寄存器。表 11-1 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

表 11-1. CCFG 寄存器

偏移	首字母缩写词	寄存器名称	部分
1FA8h	EXT_LF_CLK	外部 LF 时钟配置	EXT_LF_CLK 寄存器（偏移 = 1FA8h）[复位 = FFFFFFFFh]
1FACh	MODE_CONF_1	模式配置 1	MODE_CONF_1 寄存器（偏移 = 1FACh）[复位 = FFFFFFFFh]
1FB0h	SIZE_AND_DIS_FLAGS	CCFG 大小和禁用标志	SIZE_AND_DIS_FLAGS 寄存器（偏移 = 1FB0h）[复位 = FFFFFFFFh]
1FB4h	MODE_CONF	模式配置 0	MODE_CONF 寄存器（偏移 = 1FB4h）[复位 = FFFFFFFFh]
1FB8h	VOLT_LOAD_0	电压负载 0	VOLT_LOAD_0 寄存器（偏移 = 1FB8h）[复位 = FFFFFFFFh]
1FBCh	VOLT_LOAD_1	电压负载 1	VOLT_LOAD_1 寄存器（偏移 = 1FBCh）[复位 = FFFFFFFFh]
1FC0h	RTC_OFFSET	实时时钟偏移	RTC_OFFSET 寄存器（偏移 = 1FC0h）[复位 = FFFFFFFFh]
1FC4h	FREQ_OFFSET	频率偏移	FREQ_OFFSET 寄存器（偏移 = 1FC4h）[复位 = FFFFFFFFh]
1FC8h	IEEE_MAC_0	IEEE MAC 地址 0	IEEE_MAC_0 寄存器（偏移 = 1FC8h）[复位 = FFFFFFFFh]
1FCCh	IEEE_MAC_1	IEEE MAC 地址 1	IEEE_MAC_1 寄存器（偏移 = 1FCCh）[复位 = FFFFFFFFh]
1FD0h	IEEE_BLE_0	IEEE BLE 地址 0	IEEE_BLE_0 寄存器（偏移 = 1FD0h）[复位 = FFFFFFFFh]
1FD4h	IEEE_BLE_1	IEEE BLE 地址 1	IEEE_BLE_1 寄存器（偏移 = 1FD4h）[复位 = FFFFFFFFh]
1FD8h	BL_CONFIG	引导加载程序配置	BL_CONFIG 寄存器（偏移 =1FD8h）[复位 = C5FFFFFFh]
1FDCh	ERASE_CONF	擦除配置	ERASE_CONF 寄存器（偏移 = 1FDCh）[复位 = FFFFFFFFh]
1FE0h	CCFG_TI_OPTIONS	TI 选项	CCFG_TI_OPTIONS 寄存器（偏移 = 1FE0h）[复位 = FFFFFFC5h]
1FE4h	CCFG_TAP_DAP_0	测试接入点使能 0	CCFG_TAP_DAP_0 寄存器（偏移 = 1FE4h）[复位 = FFC5C5C5h]
1FE8h	CCFG_TAP_DAP_1	测试接入点使能 1	CCFG_TAP_DAP_1 寄存器（偏移 = 1FE8h）[复位 = FFC5C5C5h]
1FECh	IMAGE_VALID_CONF	映像有效	IMAGE_VALID_CONF 寄存器（偏移 = 1FECh）[复位 = FFFFFFFFh]
1FF0h	CCFG_PROT_31_0	保护扇区 0-31	CCFG_PROT_31_0 寄存器（偏移 = 1FF0h）[复位 = FFFFFFFFh]
1FF4h	CCFG_PROT_63_32	保护扇区 32-63	CCFG_PROT_63_32 寄存器（偏移 = 1FF4h）[复位 = FFFFFFFFh]
1FF8h	CCFG_PROT_95_64	保护扇区 64-95	CCFG_PROT_95_64 寄存器（偏移 = 1FF8h）[复位 = FFFFFFFFh]
1FFCh	CCFG_PROT_127_96	保护扇区 96-127	CCFG_PROT_127_96 寄存器（偏移 = 1FFCh）[复位 = FFFFFFFFh]

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

复杂的位访问类型经过编码可适应小型表单元。表 11-2 展示了适用于此部分中访问类型的代码。

表 11-2. CCFG 访问类型代码

访问类型	代码	说明
读取类型		
R	R	读取
复位或默认值		
-n		复位后的值或默认值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.2.1.1 EXT_LF_CLK 寄存器（偏移 = 1FA8h）[复位 = FFFFFFFFh]

图 11-1 展示了 EXT_LF_CLK，表 11-3 对其进行了介绍。

返回到汇总表。

外部 LF 时钟配置

图 11-1. EXT_LF_CLK 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIO								RTC_INCREMENT																							
R-FFh								R-00FFFFFFh																							

表 11-3. EXT_LF_CLK 寄存器字段说明

位	字段	类型	复位	说明
31-24	DIO	R	FFh	无符号整数，当 MODE_CONF.SCLK_LF_OPTION 设置为 EXTERNAL 时，选择 DIO 以提供外部 32kHz 时钟作为 SCLK_LF。所选 DIO 将被引脚驱动程序（TI-RTOS 环境）标记为保留，因此不可用于其他用途。
23-0	RTC_INCREMENT	R	00FFFFFFh	无符号整数，定义外部时钟的输入频率并写入 AON_RTC:SUBSECINC.VALUEINC。定义如下：EXT_LF_CLK.RTC_INCREMENT = 2^{38}/InputClockFrequency（以赫兹为单位）（例如：InputClockFrequency=32768Hz 时为 RTC_INCREMENT=0x800000）

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.2.1.2 MODE_CONF_1 寄存器（偏移 = 1FACh）[复位 = FFFFFFFFh]

图 11-2 展示了 MODE_CONF_1，表 11-4 对其进行了介绍。

返回到汇总表。

模式配置 1

图 11-2. MODE_CONF_1 寄存器

31	30	29	28	27	26	25	24
TCXO_TYPE	TCXO_MAX_START						
R-1h	R-7Fh						
23	22	21	20	19	18	17	16
ALT_DCDC_VMIN				ALT_DCDC_DITHER_EN	ALT_DCDC_IPEAK		
R-Fh				R-1h	R-7h		
15	14	13	12	11	10	9	8
DELTA_IBIAS_INIT				DELTA_IBIAS_OFFSET			
R-Fh				R-Fh			
7	6	5	4	3	2	1	0
XOSC_MAX_START							
R-FFh							

表 11-4. MODE_CONF_1 寄存器字段说明

位	字段	类型	复位	说明
31	TCXO_TYPE	R	1h	选择 TCXO 类型。 0：CMOS 类型。不会启用内部共模偏置。 1：削波正弦型。使用 TCXO 时将启用内部共模偏置。 仅当 MODE_CONF.XOSC_FREQ=0 时，位字段值才有效。
30-24	TCXO_MAX_START	R	7Fh	最大 TCXO 启动时间，以 100us 为单位。 仅当 MODE_CONF.XOSC_FREQ=0 时，位字段值才有效。
23-20	ALT_DCDC_VMIN	R	Fh	如果启用了备用 DC/DC 设置 (SIZE_AND_DIS_FLAGS.DIS_ALT_DCDC_SETTING=0)，则应使用 DC/DC 时的最低电压。 电压 = (28 + ALT_DCDC_VMIN) / 16。 0：1.75V 1：1.8125V ... 14：2.625V 15：2.6875V 注意！必须定期调用 DriverLib 函数 SysCtrl_DCDC_VoltageConditionalControl()，才能应用此字段（如果使用 TI RTOS，则会自动处理）。
19	ALT_DCDC_DITHER_EN	R	1h	如果启用了备用 DC/DC 设置 (SIZE_AND_DIS_FLAGS.DIS_ALT_DCDC_SETTING=0)，则启用 DC/DC 抖动。 0：抖动禁用 1：抖动启用
18-16	ALT_DCDC_IPEAK	R	7h	启用备用 DC/DC 设置 (SIZE_AND_DIS_FLAGS.DIS_ALT_DCDC_SETTING=0) 时的电感器峰值电流。假设使用 10uH 外部电感器！ 0：46mA (min) ... 4：70mA ... 7：87mA（最大值）

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 11-4. MODE_CONF_1 寄存器字段说明（续）

位	字段	类型	复位	说明
15-12	DELTA_IBIAS_INIT	R	Fh	IBIAS_INIT 的有符号增量值。增量值仅在 SIZE_AND_DIS_FLAGS.DIS_XOSC_OVR=0 时适用。 请参阅 FCFG1:AMPCOMP_CTRL1.IBIAS_INIT
11-8	DELTA_IBIAS_OFFSET	R	Fh	IBIAS_OFFSET 的有符号增量值。增量值仅在 SIZE_AND_DIS_FLAGS.DIS_XOSC_OVR=0 时适用。 请参阅 FCFG1:AMPCOMP_CTRL1.IBIAS_OFFSET
7-0	XOSC_MAX_START	R	FFh	最长 XOSC 启动时间（最坏情况）的无符号值，以 100us 为单位。值仅在 SIZE_AND_DIS_FLAGS.DIS_XOSC_OVR=0 时适用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.2.1.3 SIZE_AND_DIS_FLAGS 寄存器（偏移 = 1FB0h）[复位 = FFFFFFFFh]

图 11-3 展示了 SIZE_AND_DIS_FLAGS，表 11-5 对其进行了介绍。

返回到汇总表。

CCFG 大小和禁用标志

图 11-3. SIZE_AND_DIS_FLAGS 寄存器

31	30	29	28	27	26	25	24
SIZE_OF_CCFG							
R-FFFFh							
23	22	21	20	19	18	17	16
SIZE_OF_CCFG							
R-FFFFh							
15	14	13	12	11	10	9	8
DISABLE_FLAGS							
R-7FFh							
7	6	5	4	3	2	1	0
DISABLE_FLAGS			DIS_LINEAR_CAPARRAY_DELTA_WORKAROUND	DIS_TCXO	DIS_GPRAM	DIS_ALT_DCDC_SETTING	DIS_XOSC_OVR
R-7FFh			R-1h	R-1h	R-1h	R-1h	R-1h

表 11-5. SIZE_AND_DIS_FLAGS 寄存器字段说明

位	字段	类型	复位	说明
31-16	SIZE_OF_CCFG	R	FFFFh	CCFG 的总大小（以字节为单位）。
15-5	DISABLE_FLAGS	R	7FFh	保留供将来使用。软件不应依赖保留位的值。写入除复位/默认值之外的任何其他值都可能导致未定义的行为。
4	DIS_LINEAR_CAPARRAY_DELTA_WORKAROUND	R	1h	只要未添加 CAPARRAY_DELTA 调整，默认的 CAPARRAY 设置就处于正常状态，但如果未启用权变措施，CAPARRAY 设置将产生非线性行为。默认情况下，该权变措施处于禁用状态，以避免在软件更新时发生意外更改。 0：CAPARRAY_DELTA 权变措施已启用。 1：CAPARRAY_DELTA 权变措施已禁用。
3	DIS_TCXO	R	1h	已弃用。必须设置为 1。
2	DIS_GPRAM	R	1h	禁用 GPRAM（或使用 8K VIMS RAM 作为高速缓存 RAM）。 0：GPRAM 已启用，因此高速缓存已禁用。 1：GPRAM 已禁用，而缓存已启用（默认）。 注意： - 禁用高速缓存将降低 CPU 执行速度（最降低多 60%）。 - GPRAM 大小为 8kb，如果启用，则位于 0x11000000-0x11001FFF 处。 请参阅： VIMS:CTL.MODE
1	DIS_ALT_DCDC_SETTING	R	1h	禁用备用 DC/DC 设置。 0：启用备用 DC/DC 设置。 1：禁用备用 DC/DC 设置。 请参阅： MODE_CONF_1.ALT_DCDC_VMIN MODE_CONF_1.ALT_DCDC_DITHER_EN MODE_CONF_1.ALT_DCDC_IPEAK 注释！必须定期调用 DriverLib 函数 SysCtrl_DCDC_VoltageConditionalControl()，才能应用此字段（如果使用 TI RTOS，则会自动处理）。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 11-5. SIZE_AND_DIS_FLAGS 寄存器字段说明 （续）

位	字段	类型	复位	说明
0	DIS_XOSC_OVR	R	1h	禁用 XOSC 覆盖功能。 0：启用 XOSC 覆盖功能。 1：禁用 XOSC 覆盖功能。 请参阅： MODE_CONF_1.DELTA_IBIAS_INIT MODE_CONF_1.DELTA_IBIAS_OFFSET MODE_CONF_1.XOSC_MAX_START

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.2.1.4 MODE_CONF 寄存器（偏移 = 1FB4h）[复位 = FFFFFFFFh]

图 11-4 展示了 MODE_CONF，表 11-6 对其进行了介绍。

返回到汇总表。

模式配置 0

图 11-4. MODE_CONF 寄存器

31	30	29	28	27	26	25	24
VDDR_TRIM_SLEEP_DELTA				DCDC_RECHARGE	DCDC_ACTIVE	VDDR_EXT_LOAD	VDDS_BOD_LEVEL
R-Fh				R-1h	R-1h	R-1h	R-1h
23	22	21	20	19	18	17	16
SCLK_LF_OPTION		VDDR_TRIM_SLEEP_TC	RTC_COMP	XOSC_FREQ		XOSC_CAP_MOD	HF_COMP
R-3h		R-1h	R-1h	R-3h		R-1h	R-1h
15	14	13	12	11	10	9	8
XOSC_CAPARRAY_DELTA							
R-FFh							
7	6	5	4	3	2	1	0
VDDR_CAP							
R-FFh							

表 11-6. MODE_CONF 寄存器字段说明

位	字段	类型	复位	说明
31-28	VDDR_TRIM_SLEEP_DELTA	R	Fh	应用于 VDDR_TRIM_SLEEP 目标的 有符号增量值（减一）。请参阅 FCFG1:VOLT_TRIM.VDDR_TRIM_SLEEP_H. 0x8 (-8)：Δ = -7 ... 0xF (-1)：Δ = 0 0x0 (0)：Δ = +1 ... 0x7 (7)：Δ = +8
27	DCDC_RECHARGE	R	1h	掉电状态下，充电过程中的 DC/DC。 0：掉电状态下，在充电过程中使用 DC/DC。 1：掉电状态下，在充电过程中请勿使用 DC/DC（默认）。 注意！必须定期调用 DriverLib 函数 SysCtrl_DCDC_VoltageConditionalControl()，才能应用此字段（如果使用 TI RTOS，则会自动处理）。
26	DCDC_ACTIVE	R	1h	DC/DC 处于工作模式。 0：在工作模式下使用 DC/DC。 1：在工作模式下，请勿使用 DC/DC（默认）。 注意！必须定期调用 DriverLib 函数 SysCtrl_DCDC_VoltageConditionalControl()，才能应用此字段（如果使用 TI RTOS，则会自动处理）。
25	VDDR_EXT_LOAD	R	1h	保留供将来使用。软件不应依赖保留位的值。写入除复位/默认值之外的任何其他值都可能导致未定义的行为。
24	VDDS_BOD_LEVEL	R	1h	VDDS BOD 电平。 0：VDDS BOD 电平为 2.0V（对于外部负载模式或 CC13xx 的最大 PA 输出功率而言是必需的）。 1：VDDS BOD 电平为 1.8V（或外部稳压器模式下为 1.65V）（默认）。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 11-6. MODE_CONF 寄存器字段说明（续）

位	字段	类型	复位	说明
23-22	SCLK_LF_OPTION	R	3h	选择 SCLK_LF 的时钟源。 0h = 31.25kHz 时钟源自 48MHz XOSC 或 HPOSC。RTC 滴答速度 AON_RTC:SUBSECINC 更新为 0x8637BD，对应 31.25kHz 时钟（在 SetupTrimDevice() driverlib 引导函数中完成）。使用此时钟源时，必须防止器件进入待机模式。 1h = DIO 的外部低频率时钟由 EXT_LF_CLK.DIO 定义。RTC 滴答速度 AON_RTC:SUBSECINC 更新为 EXT_LF_CLK.RTC_INCREMENT（在 SetupTrimDevice() driverlib 引导函数中完成）。在芯片处于待机状态以进行 VDDR 充电定时时，外部时钟必须一直处于运行状态。 2h = 32.768kHz 低频率 XOSC 3h = 低频率 RCOSC（默认）
21	VDDR_TRIM_SLEEP_TC	R	1h	0x1：VDDR_TRIM_SLEEP_DELTA 未进行温度补偿 0x0：每次进入待机模式时，RTOS/驱动器都会对 VDDR_TRIM_SLEEP_DELTA 进行温度补偿。这样可以提高待机模式下的低温 RCOSC_LF 频率稳定性。 执行温度补偿时，按照以下方式计算增量： Δ = max (Δ, min(8, floor(62-temp)/8)) 这里，Δ 由 VDDR_TRIM_SLEEP_DELTA 给出，temp 是当前温度，单位为 °C。
20	RTC_COMP	R	1h	保留供将来使用。软件不应依赖保留位的值。写入除复位/默认值之外的任何其他值都可能导致未定义的行为。
19-18	XOSC_FREQ	R	3h	选择所使用的高频率振荡器（无线电应用必需）。 0h = 外部 48MHz TCXO。 有关 TCXO 的其他配置，请参阅 MODE_CONF_1.TCXO_MAX_START 和 MODE_CONF_1.TCXO_TYPE 位字段。 1h = 内部高精度振荡器。 2h = 48M：48 MHz XOSC_HF 3h = 24M： 24 MHz XOSC_HF。不支持。
17	XOSC_CAP_MOD	R	1h	启用对 XOSC 电容器阵列的修改 (Δ)。XOSC_CAPARRAY_DELTA 中指定的值。 0：应用电容器阵列增量 1：请勿应用电容器阵列增量（默认）
16	HF_COMP	R	1h	保留供将来使用。软件不应依赖保留位的值。写入除复位/默认值之外的任何其他值都可能导致未定义的行为。
15-8	XOSC_CAPARRAY_DELTA	R	FFh	有符号 8 位值，直接修改修整后的 XOSC 电容器阵列步长值。由 XOSC_CAP_MOD 启用。
7-0	VDDR_CAP	R	FFh	无符号 8 位整数，表示 VDDR 上的最小去耦电容（最坏情况），以 100nF 为单位。这应考虑电容器容差和与电压相关的电容变化。当进入掉电或待机状态时，该位会影响充电周期计算。 注意！如果使用以下函数，则必须配置此字段（由 TI RTOS 使用）： SysCtrlSetRechargeBeforePowerDown() SysCtrlAdjustRechargeAfterPowerDown()

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.2.1.5 VOLT_LOAD_0 寄存器（偏移 = 1FB8h）[复位 = FFFFFFFFh]

图 11-5 展示了 VOLT_LOAD_0，表 11-7 对其进行了介绍。

返回到汇总表。

电压负载 0
由 MODE_CONF.VDDR_EXT_LOAD 启用。

图 11-5. VOLT_LOAD_0 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VDDR_EXT_TP45								VDDR_EXT_TP25							
R-FFh								R-FFh							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VDDR_EXT_TP5								VDDR_EXT_TM15							
R-FFh								R-FFh							

表 11-7. VOLT_LOAD_0 寄存器字段说明

位	字段	类型	复位	说明
31-24	VDDR_EXT_TP45	R	FFh	保留供将来使用。软件不应依赖保留位的值。写入除复位/默认值之外的任何其他值都可能导致未定义的行为。
23-16	VDDR_EXT_TP25	R	FFh	保留供将来使用。软件不应依赖保留位的值。写入除复位/默认值之外的任何其他值都可能导致未定义的行为。
15-8	VDDR_EXT_TP5	R	FFh	保留供将来使用。软件不应依赖保留位的值。写入除复位/默认值之外的任何其他值都可能导致未定义的行为。
7-0	VDDR_EXT_TM15	R	FFh	保留供将来使用。软件不应依赖保留位的值。写入除复位/默认值之外的任何其他值都可能导致未定义的行为。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.2.1.6 VOLT_LOAD_1 寄存器（偏移 = 1FBCh）[复位 = FFFFFFFFh]

图 11-6 展示了 VOLT_LOAD_1，表 11-8 对其进行了介绍。

返回到汇总表。

电压负载 1
由 MODE_CONF.VDDR_EXT_LOAD 启用。

图 11-6. VOLT_LOAD_1 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VDDR_EXT_TP125								VDDR_EXT_TP105							
R-FFh								R-FFh							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VDDR_EXT_TP85								VDDR_EXT_TP65							
R-FFh								R-FFh							

表 11-8. VOLT_LOAD_1 寄存器字段说明

位	字段	类型	复位	说明
31-24	VDDR_EXT_TP125	R	FFh	保留供将来使用。软件不应依赖保留位的值。写入除复位/默认值之外的任何其他值都可能导致未定义的行为。
23-16	VDDR_EXT_TP105	R	FFh	保留供将来使用。软件不应依赖保留位的值。写入除复位/默认值之外的任何其他值都可能导致未定义的行为。
15-8	VDDR_EXT_TP85	R	FFh	保留供将来使用。软件不应依赖保留位的值。写入除复位/默认值之外的任何其他值都可能导致未定义的行为。
7-0	VDDR_EXT_TP65	R	FFh	保留供将来使用。软件不应依赖保留位的值。写入除复位/默认值之外的任何其他值都可能导致未定义的行为。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.2.1.7 RTC_OFFSET 寄存器（偏移 = 1FC0h）[复位 = FFFFFFFFh]

图 11-7 展示了 RTC_OFFSET，表 11-9 对其进行了介绍。

返回到汇总表。

实时时钟偏移
由 MODE_CONF.RTC_COMP 启用。

图 11-7. RTC_OFFSET 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RTC_COMP_P0															
R-FFFFh															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTC_COMP_P1								RTC_COMP_P2							
R-FFh								R-FFh							

表 11-9. RTC_OFFSET 寄存器字段说明

位	字段	类型	复位	说明
31-16	RTC_COMP_P0	R	FFFFh	保留供将来使用。软件不应依赖保留位的值。写入除复位/默认值之外的任何其他值都可能导致未定义的行为。
15-8	RTC_COMP_P1	R	FFh	保留供将来使用。软件不应依赖保留位的值。写入除复位/默认值之外的任何其他值都可能导致未定义的行为。
7-0	RTC_COMP_P2	R	FFh	保留供将来使用。软件不应依赖保留位的值。写入除复位/默认值之外的任何其他值都可能导致未定义的行为。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.2.1.8 FREQ_OFFSET 寄存器（偏移 = 1FC4h）[复位 = FFFFFFFFh]

图 11-8 展示了 FREQ_OFFSET，表 11-10 对其进行了介绍。

返回到汇总表。

频率偏移

图 11-8. FREQ_OFFSET 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HF_COMP_P0															
R-FFFFh															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HF_COMP_P1								HF_COMP_P2							
R-FFh								R-FFh							

表 11-10. FREQ_OFFSET 寄存器字段说明

位	字段	类型	复位	说明
31-16	HF_COMP_P0	R	FFFFh	保留供将来使用。软件不应依赖保留位的值。写入除复位/默认值之外的任何其他值都可能导致未定义的行为。
15-8	HF_COMP_P1	R	FFh	保留供将来使用。软件不应依赖保留位的值。写入除复位/默认值之外的任何其他值都可能导致未定义的行为。
7-0	HF_COMP_P2	R	FFh	保留供将来使用。软件不应依赖保留位的值。写入除复位/默认值之外的任何其他值都可能导致未定义的行为。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.2.1.9 IEEE_MAC_0 寄存器（偏移 = 1FC8h）[复位 = FFFFFFFFh]

图 11-9 展示了 IEEE_MAC_0，表 11-11 对其进行了介绍。

返回到汇总表。

IEEE MAC 地址 0

图 11-9. IEEE_MAC_0 寄存器

31	30	29	28	27	26	25	24	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-FFFFFFFFh																															

表 11-11. IEEE_MAC_0 寄存器字段说明

位	字段	类型	复位	说明
31-0	ADDR	R	FFFFFFFFh	64 位自定义 IEEE MAC 地址的位 [31:0]。 如果不是 0xFFFFFFFF，则应用此字段的值， 否则使用 FCFG 中的值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.2.1.10 IEEE_MAC_1 寄存器（偏移 = 1FCCh）[复位 = FFFFFFFFh]

图 11-10 展示了 IEEE_MAC_1，表 11-12 对其进行了介绍。

返回到汇总表。

IEEE MAC 地址 1

图 11-10. IEEE_MAC_1 寄存器

31	30	29	28	27	26	25	24	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-FFFFFFFFh																															

表 11-12. IEEE_MAC_1 寄存器字段说明

位	字段	类型	复位	说明
31-0	ADDR	R	FFFFFFFFh	64 位自定义 IEEE MAC 地址的位 [63:32]。 如果不是 0xFFFFFFFF，则应用此字段的值， 否则使用 FCFG 中的值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.2.1.11 IEEE_BLE_0 寄存器（偏移 = 1FD0h）[复位 = FFFFFFFFh]

图 11-11 展示了 IEEE_BLE_0，表 11-13 对其进行了介绍。

返回到汇总表。

IEEE BLE 地址 0

图 11-11. IEEE_BLE_0 寄存器

31	30	29	28	27	26	25	24	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-FFFFFFFFh																															

表 11-13. IEEE_BLE_0 寄存器字段说明

位	字段	类型	复位	说明
31-0	ADDR	R	FFFFFFFFh	64 位自定义 IEEE BLE 地址的位 [31:0]。 如果不是 0xFFFFFFFF，则应用此字段的值， 否则使用 FCFG 中的值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.2.1.12 IEEE_BLE_1 寄存器（偏移 = 1FD4h）[复位 = FFFFFFFFh]

图 11-12 展示了 IEEE_BLE_1，表 11-14 对其进行了介绍。

返回到汇总表。

IEEE BLE 地址 1

图 11-12. IEEE_BLE_1 寄存器

31	30	29	28	27	26	25	24	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-FFFFFFFFh																															

表 11-14. IEEE_BLE_1 寄存器字段说明

位	字段	类型	复位	说明
31-0	ADDR	R	FFFFFFFFh	64 位自定义 IEEE BLE 地址的位 [63:32]。 如果不是 0xFFFFFFFF，则应用此字段的值， 否则使用 FCFG 中的值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.2.1.13 BL_CONFIG 寄存器（偏移 =1FD8h）[复位 = C5FFFFFFh]

图 11-13 展示了 BL_CONFIG，表 11-15 对其进行了介绍。

返回到汇总表。

引导加载程序配置
配置 ROM 引导加载程序的功能。
如果引导加载程序通过 BOOTLOADER_ENABLE 字段启用，并且引导加载程序后门由 BL_ENABLE 字段启用，则即使闪存中存在有效映像，也可以强制进入 ROM 引导加载程序。

图 11-13. BL_CONFIG 寄存器

31	30	29	28	27	26	25	24
BOOTLOADER_ENABLE							
R-C5h							
23	22	21	20	19	18	17	16
RESERVED							BL_LEVEL
R-0h							R-1h
15	14	13	12	11	10	9	8
BL_PIN_NUMBER							
R-FFh							
7	6	5	4	3	2	1	0
BL_ENABLE							
R-FFh							

表 11-15. BL_CONFIG 寄存器字段说明

位	字段	类型	复位	说明
31-24	BOOTLOADER_ENABLE	R	C5h	引导加载程序启用。如果 IMAGE_VALID_CONF.IMAGE_VALID 为非零或启用了 BL_ENABLE（并且满足引导加载程序后门的条件），则可以访问引导加载程序。 0xC5：引导加载程序已启用。 任何其他值：引导加载程序已禁用。
23-17	RESERVED	R	0h	保留
16	BL_LEVEL	R	1h	如果引导加载程序后门由 BL_ENABLE 字段启用，则设置所选 DIO 编号 BL_PIN_NUMBER 的有效电平。 0：低电平有效。 1：高电平有效。
15-8	BL_PIN_NUMBER	R	FFh	当 BL_ENABLE 字段启用引导加载程序后门时，用于进行电平检查的 DIO 编号。
7-0	BL_ENABLE	R	FFh	启用引导加载程序后门。 0xC5：启动加载程序后门已启用。 任何其他值：禁用引导加载程序后门。 注意！如果启用了引导加载程序后门，则必须启用引导加载程序（请参阅 BOOTLOADER_ENABLE）。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.2.1.14 ERASE_CONF 寄存器（偏移 = 1FDCh）[复位 = FFFFFFFFh]

图 11-14 展示了 ERASE_CONF，表 11-16 对其进行了介绍。

返回到汇总表。

擦除配置

图 11-14. ERASE_CONF 寄存器

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

表 11-16. ERASE_CONF 寄存器字段说明

位	字段	类型	复位	说明
31-9	RESERVED	R	0h	保留
8	CHIP_ERASE_DIS_N	R	1h	芯片擦除。 该位控制通过 JTAG WUC TAP 请求的芯片擦除是否将在由 MCU VD 复位引起的后续引导中被忽略。 成功的芯片擦除操作将强制使闪存主存储体的内容恢复到 TI 发送时的状态。 0：禁用。引导期间检测到的所有芯片擦除请求都将被忽略。 1：使能。引导期间检测到的所有芯片擦除请求都将由引导固件执行。
7-1	RESERVED	R	0h	保留
0	BANK_ERASE_DIS_N	R	1h	存储体擦除。 该位控制 ROM 串行引导加载程序是否将接受收到的存储体擦除命令 (COMMAND_BANK_ERASE)。 成功执行的存储体擦除操作将擦除 CCFG 中写保护配置位未保护的所有主存储体扇区。 0：禁用引导加载程序存储体擦除功能。 1：启用引导加载程序存储体擦除功能。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.2.1.15 CCFG_TI_OPTIONS 寄存器（偏移 = 1FE0h）[复位 = FFFFFFC5h]

图 11-15 展示了 CCFG_TI_OPTIONS，表 11-17 对其进行了介绍。

返回到汇总表。

TI 选项

图 11-15. CCFG_TI_OPTIONS 寄存器

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								TI_FA_ENABLE							
R-0h								R-C5h							

表 11-17. CCFG_TI_OPTIONS 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-0	TI_FA_ENABLE	R	C5h	TI 故障分析。 0xC5：启用使用解锁代码解锁 TI FA（TI 失效分析）选项的功能。 所有其他值：禁用使用解锁代码解锁 TI FA 选项的功能。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.2.1.16 CCFG_TAP_DAP_0 寄存器（偏移 = 1FE4h）[复位 = FFC5C5C5h]

图 11-16 展示了 CCFG_TAP_DAP_0，表 11-18 对其进行了介绍。

返回到汇总表。

测试接入点使能 0

图 11-16. CCFG_TAP_DAP_0 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED								CPU_DAP_ENABLE							
R-0h								R-C5h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWRPROF_TAP_ENABLE								TEST_TAP_ENABLE							
R-C5h								R-C5h							

表 11-18. CCFG_TAP_DAP_0 寄存器字段说明

位	字段	类型	复位	说明
31-24	RESERVED	R	0h	保留
23-16	CPU_DAP_ENABLE	R	C5h	启用 CPU DAP。 0xC5：在上电/系统复位期间，ROM 引导固件会启用主 CPU DAP 访问。 任何其他值：在上电/系统复位后，主 CPU DAP 访问将保持禁用状态。
15-8	PWRPROF_TAP_ENABLE	R	C5h	启用 PWRPROF TAP。 0xC5：如果由 TI 定义的 FCFG1 中的相应配置值启用，则在上电/系统复位期间，ROM 引导固件会启用 PWRPROF TAP 访问。 任何其他值：在上电/系统复位后，PWRPROF TAP 访问将保持禁用状态。
7-0	TEST_TAP_ENABLE	R	C5h	启用 Test TAP。 0xC5：如果由 TI 定义的 FCFG1 中的相应配置值启用，则在上电/系统复位期间，ROM 引导固件会启用 TEST TAP 访问。 任何其他值：在上电/系统复位后，TEST TAP 访问将保持禁用状态。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.2.1.17 CCFG_TAP_DAP_1 寄存器（偏移 = 1FE8h）[复位 = FFC5C5C5h]

图 11-17 展示了 CCFG_TAP_DAP_1，表 11-19 对其进行了介绍。

返回到汇总表。

测试接入点使能 1

图 11-17. CCFG_TAP_DAP_1 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED								PBIST2_TAP_ENABLE							
R-0h								R-C5h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PBIST1_TAP_ENABLE								AON_TAP_ENABLE							
R-C5h								R-C5h							

表 11-19. CCFG_TAP_DAP_1 寄存器字段说明

位	字段	类型	复位	说明
31-24	RESERVED	R	0h	保留
23-16	PBIST2_TAP_ENABLE	R	C5h	启用 PBIST2 TAP。 0xC5：如果由 TI 定义的 FCFG1 中的相应配置值启用，则在上电/系统复位期间，ROM 引导固件会启用 PBIST2 TAP 访问。 任何其他值：在上电/系统复位后，PBIST2 TAP 访问将保持禁用状态。
15-8	PBIST1_TAP_ENABLE	R	C5h	启用 PBIST1 TAP。 0xC5：如果由 TI 定义的 FCFG1 中的相应配置值启用，则在上电/系统复位期间，ROM 引导固件会启用 PBIST1 TAP 访问。 任何其他值：在上电/系统复位后，PBIST1 TAP 访问将保持禁用状态。
7-0	AON_TAP_ENABLE	R	C5h	启用 AON TAP。 0xC5：如果由 TI 定义的 FCFG1 中的相应配置值启用，则在上电/系统复位期间，ROM 引导固件会启用 AON TAP 访问。 任何其他值：在上电/系统复位后，AON TAP 访问将保持禁用状态。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.2.1.18 IMAGE_VALID_CONF 寄存器（偏移 = 1FECh）[复位 = FFFFFFFFh]

图 11-18 展示了 IMAGE_VALID_CONF，表 11-20 对其进行了介绍。

返回到汇总表。

映像有效

图 11-18. IMAGE_VALID_CONF 寄存器

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

表 11-20. IMAGE_VALID_CONF 寄存器字段说明

位	字段	类型	复位	说明
31-0	IMAGE_VALID	R	FFFFFFFFh	该字段必须填入闪存向量表的起始地址值，以便使 ROM 中的引导固件能够将控制权转交至闪存映像。 任何非法的向量表起始地址值都将迫使 ROM 中的引导固件将控制权转交至 ROM 内的串行引导加载程序。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.2.1.19 CCFG_PROT_31_0 寄存器（偏移 = 1FF0h）[复位 = FFFFFFFFh]

图 11-19 展示了 CCFG_PROT_31_0，表 11-21 对其进行了介绍。

返回到汇总表。

保护扇区 0-31
每一个位都会对一个 8KB 闪存扇区进行写入保护，防止被编程和擦除。为了启用扇区写入保护功能，位必须设置为 0。

图 11-19. CCFG_PROT_31_0 寄存器

31	30	29	28	27	26	25	24
WRT_PROT_SEC_31	WRT_PROT_SEC_30	WRT_PROT_SEC_29	WRT_PROT_SEC_28	WRT_PROT_SEC_27	WRT_PROT_SEC_26	WRT_PROT_SEC_25	WRT_PROT_SEC_24
R-1h	R-1h	R-1h	R-1h	R-1h	R-1h	R-1h	R-1h
23	22	21	20	19	18	17	16
WRT_PROT_SEC_23	WRT_PROT_SEC_22	WRT_PROT_SEC_21	WRT_PROT_SEC_20	WRT_PROT_SEC_19	WRT_PROT_SEC_18	WRT_PROT_SEC_17	WRT_PROT_SEC_16
R-1h	R-1h	R-1h	R-1h	R-1h	R-1h	R-1h	R-1h
15	14	13	12	11	10	9	8
WRT_PROT_SEC_15	WRT_PROT_SEC_14	WRT_PROT_SEC_13	WRT_PROT_SEC_12	WRT_PROT_SEC_11	WRT_PROT_SEC_10	WRT_PROT_SEC_9	WRT_PROT_SEC_8
R-1h	R-1h	R-1h	R-1h	R-1h	R-1h	R-1h	R-1h
7	6	5	4	3	2	1	0
WRT_PROT_SEC_7	WRT_PROT_SEC_6	WRT_PROT_SEC_5	WRT_PROT_SEC_4	WRT_PROT_SEC_3	WRT_PROT_SEC_2	WRT_PROT_SEC_1	WRT_PROT_SEC_0
R-1h	R-1h	R-1h	R-1h	R-1h	R-1h	R-1h	R-1h

表 11-21. CCFG_PROT_31_0 寄存器字段说明

位	字段	类型	复位	说明
31	WRT_PROT_SEC_31	R	1h	0：扇区受保护
30	WRT_PROT_SEC_30	R	1h	0：扇区受保护
29	WRT_PROT_SEC_29	R	1h	0：扇区受保护
28	WRT_PROT_SEC_28	R	1h	0：扇区受保护
27	WRT_PROT_SEC_27	R	1h	0：扇区受保护
26	WRT_PROT_SEC_26	R	1h	0：扇区受保护
25	WRT_PROT_SEC_25	R	1h	0：扇区受保护
24	WRT_PROT_SEC_24	R	1h	0：扇区受保护
23	WRT_PROT_SEC_23	R	1h	0：扇区受保护
22	WRT_PROT_SEC_22	R	1h	0：扇区受保护
21	WRT_PROT_SEC_21	R	1h	0：扇区受保护
20	WRT_PROT_SEC_20	R	1h	0：扇区受保护
19	WRT_PROT_SEC_19	R	1h	0：扇区受保护
18	WRT_PROT_SEC_18	R	1h	0：扇区受保护
17	WRT_PROT_SEC_17	R	1h	0：扇区受保护
16	WRT_PROT_SEC_16	R	1h	0：扇区受保护
15	WRT_PROT_SEC_15	R	1h	0：扇区受保护
14	WRT_PROT_SEC_14	R	1h	0：扇区受保护
13	WRT_PROT_SEC_13	R	1h	0：扇区受保护

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 11-21. CCFG_PROT_31_0 寄存器字段说明（续）

位	字段	类型	复位	说明
12	WRT_PROT_SEC_12	R	1h	0：扇区受保护
11	WRT_PROT_SEC_11	R	1h	0：扇区受保护
10	WRT_PROT_SEC_10	R	1h	0：扇区受保护
9	WRT_PROT_SEC_9	R	1h	0：扇区受保护
8	WRT_PROT_SEC_8	R	1h	0：扇区受保护
7	WRT_PROT_SEC_7	R	1h	0：扇区受保护
6	WRT_PROT_SEC_6	R	1h	0：扇区受保护
5	WRT_PROT_SEC_5	R	1h	0：扇区受保护
4	WRT_PROT_SEC_4	R	1h	0：扇区受保护
3	WRT_PROT_SEC_3	R	1h	0：扇区受保护
2	WRT_PROT_SEC_2	R	1h	0：扇区受保护
1	WRT_PROT_SEC_1	R	1h	0：扇区受保护
0	WRT_PROT_SEC_0	R	1h	0：扇区受保护

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.2.1.20 CCFG_PROT_63_32 寄存器（偏移 = 1FF4h）[复位 = FFFFFFFFh]

图 11-20 展示了 CCFG_PROT_63_32，表 11-22 对其进行了介绍。

返回到汇总表。

保护扇区 32-63
每一个位都会对一个 8KB 闪存扇区进行写入保护，防止被编程和擦除。为了启用扇区写入保护功能，位必须设置为 0。

图 11-20. CCFG_PROT_63_32 寄存器

31	30	29	28	27	26	25	24
WRT_PROT_SEC_63	WRT_PROT_SEC_62	WRT_PROT_SEC_61	WRT_PROT_SEC_60	WRT_PROT_SEC_59	WRT_PROT_SEC_58	WRT_PROT_SEC_57	WRT_PROT_SEC_56
R-1h	R-1h	R-1h	R-1h	R-1h	R-1h	R-1h	R-1h
23	22	21	20	19	18	17	16
WRT_PROT_SEC_55	WRT_PROT_SEC_54	WRT_PROT_SEC_53	WRT_PROT_SEC_52	WRT_PROT_SEC_51	WRT_PROT_SEC_50	WRT_PROT_SEC_49	WRT_PROT_SEC_48
R-1h	R-1h	R-1h	R-1h	R-1h	R-1h	R-1h	R-1h
15	14	13	12	11	10	9	8
WRT_PROT_SEC_47	WRT_PROT_SEC_46	WRT_PROT_SEC_45	WRT_PROT_SEC_44	WRT_PROT_SEC_43	WRT_PROT_SEC_42	WRT_PROT_SEC_41	WRT_PROT_SEC_40
R-1h	R-1h	R-1h	R-1h	R-1h	R-1h	R-1h	R-1h
7	6	5	4	3	2	1	0
WRT_PROT_SEC_39	WRT_PROT_SEC_38	WRT_PROT_SEC_37	WRT_PROT_SEC_36	WRT_PROT_SEC_35	WRT_PROT_SEC_34	WRT_PROT_SEC_33	WRT_PROT_SEC_32
R-1h	R-1h	R-1h	R-1h	R-1h	R-1h	R-1h	R-1h

表 11-22. CCFG_PROT_63_32 寄存器字段说明

位	字段	类型	复位	说明
31	WRT_PROT_SEC_63	R	1h	0：扇区受保护
30	WRT_PROT_SEC_62	R	1h	0：扇区受保护
29	WRT_PROT_SEC_61	R	1h	0：扇区受保护
28	WRT_PROT_SEC_60	R	1h	0：扇区受保护
27	WRT_PROT_SEC_59	R	1h	0：扇区受保护
26	WRT_PROT_SEC_58	R	1h	0：扇区受保护
25	WRT_PROT_SEC_57	R	1h	0：扇区受保护
24	WRT_PROT_SEC_56	R	1h	0：扇区受保护
23	WRT_PROT_SEC_55	R	1h	0：扇区受保护
22	WRT_PROT_SEC_54	R	1h	0：扇区受保护
21	WRT_PROT_SEC_53	R	1h	0：扇区受保护
20	WRT_PROT_SEC_52	R	1h	0：扇区受保护
19	WRT_PROT_SEC_51	R	1h	0：扇区受保护
18	WRT_PROT_SEC_50	R	1h	0：扇区受保护
17	WRT_PROT_SEC_49	R	1h	0：扇区受保护
16	WRT_PROT_SEC_48	R	1h	0：扇区受保护
15	WRT_PROT_SEC_47	R	1h	0：扇区受保护
14	WRT_PROT_SEC_46	R	1h	0：扇区受保护
13	WRT_PROT_SEC_45	R	1h	0：扇区受保护

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 11-22. CCFG_PROT_63_32 寄存器字段说明（续）

位	字段	类型	复位	说明
12	WRT_PROT_SEC_44	R	1h	0：扇区受保护
11	WRT_PROT_SEC_43	R	1h	0：扇区受保护
10	WRT_PROT_SEC_42	R	1h	0：扇区受保护
9	WRT_PROT_SEC_41	R	1h	0：扇区受保护
8	WRT_PROT_SEC_40	R	1h	0：扇区受保护
7	WRT_PROT_SEC_39	R	1h	0：扇区受保护
6	WRT_PROT_SEC_38	R	1h	0：扇区受保护
5	WRT_PROT_SEC_37	R	1h	0：扇区受保护
4	WRT_PROT_SEC_36	R	1h	0：扇区受保护
3	WRT_PROT_SEC_35	R	1h	0：扇区受保护
2	WRT_PROT_SEC_34	R	1h	0：扇区受保护
1	WRT_PROT_SEC_33	R	1h	0：扇区受保护
0	WRT_PROT_SEC_32	R	1h	0：扇区受保护

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.2.1.21 CCFG_PROT_95_64 寄存器（偏移 = 1FF8h）[复位 = FFFFFFFFh]

图 11-21 展示了 CCFG_PROT_95_64，表 11-23 对其进行了介绍。

返回到汇总表。

保护扇区 64-95
每一个位都会对一个闪存扇区进行写入保护，防止被编程和擦除。为了启用扇区写入保护功能，位必须设置为 0。
未使用。

图 11-21. CCFG_PROT_95_64 寄存器

31	30	29	28	27	26	25	24
WRT_PROT_SEC_95	WRT_PROT_SEC_94	WRT_PROT_SEC_93	WRT_PROT_SEC_92	WRT_PROT_SEC_91	WRT_PROT_SEC_90	WRT_PROT_SEC_89	WRT_PROT_SEC_88
R-1h	R-1h	R-1h	R-1h	R-1h	R-1h	R-1h	R-1h
23	22	21	20	19	18	17	16
WRT_PROT_SEC_87	WRT_PROT_SEC_86	WRT_PROT_SEC_85	WRT_PROT_SEC_84	WRT_PROT_SEC_83	WRT_PROT_SEC_82	WRT_PROT_SEC_81	WRT_PROT_SEC_80
R-1h	R-1h	R-1h	R-1h	R-1h	R-1h	R-1h	R-1h
15	14	13	12	11	10	9	8
WRT_PROT_SEC_79	WRT_PROT_SEC_78	WRT_PROT_SEC_77	WRT_PROT_SEC_76	WRT_PROT_SEC_75	WRT_PROT_SEC_74	WRT_PROT_SEC_73	WRT_PROT_SEC_72
R-1h	R-1h	R-1h	R-1h	R-1h	R-1h	R-1h	R-1h
7	6	5	4	3	2	1	0
WRT_PROT_SEC_71	WRT_PROT_SEC_70	WRT_PROT_SEC_69	WRT_PROT_SEC_68	WRT_PROT_SEC_67	WRT_PROT_SEC_66	WRT_PROT_SEC_65	WRT_PROT_SEC_64
R-1h	R-1h	R-1h	R-1h	R-1h	R-1h	R-1h	R-1h

表 11-23. CCFG_PROT_95_64 寄存器字段说明

位	字段	类型	复位	说明
31	WRT_PROT_SEC_95	R	1h	0：扇区受保护
30	WRT_PROT_SEC_94	R	1h	0：扇区受保护
29	WRT_PROT_SEC_93	R	1h	0：扇区受保护
28	WRT_PROT_SEC_92	R	1h	0：扇区受保护
27	WRT_PROT_SEC_91	R	1h	0：扇区受保护
26	WRT_PROT_SEC_90	R	1h	0：扇区受保护
25	WRT_PROT_SEC_89	R	1h	0：扇区受保护
24	WRT_PROT_SEC_88	R	1h	0：扇区受保护
23	WRT_PROT_SEC_87	R	1h	0：扇区受保护
22	WRT_PROT_SEC_86	R	1h	0：扇区受保护
21	WRT_PROT_SEC_85	R	1h	0：扇区受保护
20	WRT_PROT_SEC_84	R	1h	0：扇区受保护
19	WRT_PROT_SEC_83	R	1h	0：扇区受保护
18	WRT_PROT_SEC_82	R	1h	0：扇区受保护
17	WRT_PROT_SEC_81	R	1h	0：扇区受保护
16	WRT_PROT_SEC_80	R	1h	0：扇区受保护
15	WRT_PROT_SEC_79	R	1h	0：扇区受保护
14	WRT_PROT_SEC_78	R	1h	0：扇区受保护
13	WRT_PROT_SEC_77	R	1h	0：扇区受保护

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 11-23. CCFG_PROT_95_64 寄存器字段说明（续）

位	字段	类型	复位	说明
12	WRT_PROT_SEC_76	R	1h	0：扇区受保护
11	WRT_PROT_SEC_75	R	1h	0：扇区受保护
10	WRT_PROT_SEC_74	R	1h	0：扇区受保护
9	WRT_PROT_SEC_73	R	1h	0：扇区受保护
8	WRT_PROT_SEC_72	R	1h	0：扇区受保护
7	WRT_PROT_SEC_71	R	1h	0：扇区受保护
6	WRT_PROT_SEC_70	R	1h	0：扇区受保护
5	WRT_PROT_SEC_69	R	1h	0：扇区受保护
4	WRT_PROT_SEC_68	R	1h	0：扇区受保护
3	WRT_PROT_SEC_67	R	1h	0：扇区受保护
2	WRT_PROT_SEC_66	R	1h	0：扇区受保护
1	WRT_PROT_SEC_65	R	1h	0：扇区受保护
0	WRT_PROT_SEC_64	R	1h	0：扇区受保护

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.2.1.22 CCFG_PROT_127_96 寄存器（偏移 = 1FFCh）[复位 = FFFFFFFFh]

图 11-22 展示了 CCFG_PROT_127_96，表 11-24 对其进行了介绍。

返回到汇总表。

保护扇区 96-127
每一个位都会对一个闪存扇区进行写入保护，防止被编程和擦除。为了启用扇区写入保护功能，位必须设置为 0。
未使用。

图 11-22. CCFG_PROT_127_96 寄存器

31	30	29	28	27	26	25	24
WRT_PROT_SEC_127	WRT_PROT_SEC_126	WRT_PROT_SEC_125	WRT_PROT_SEC_124	WRT_PROT_SEC_123	WRT_PROT_SEC_122	WRT_PROT_SEC_121	WRT_PROT_SEC_120
R-1h	R-1h	R-1h	R-1h	R-1h	R-1h	R-1h	R-1h
23	22	21	20	19	18	17	16
WRT_PROT_SEC_119	WRT_PROT_SEC_118	WRT_PROT_SEC_117	WRT_PROT_SEC_116	WRT_PROT_SEC_115	WRT_PROT_SEC_114	WRT_PROT_SEC_113	WRT_PROT_SEC_112
R-1h	R-1h	R-1h	R-1h	R-1h	R-1h	R-1h	R-1h
15	14	13	12	11	10	9	8
WRT_PROT_SEC_111	WRT_PROT_SEC_110	WRT_PROT_SEC_109	WRT_PROT_SEC_108	WRT_PROT_SEC_107	WRT_PROT_SEC_106	WRT_PROT_SEC_105	WRT_PROT_SEC_104
R-1h	R-1h	R-1h	R-1h	R-1h	R-1h	R-1h	R-1h
7	6	5	4	3	2	1	0
WRT_PROT_SEC_103	WRT_PROT_SEC_102	WRT_PROT_SEC_101	WRT_PROT_SEC_100	WRT_PROT_SEC_99	WRT_PROT_SEC_98	WRT_PROT_SEC_97	WRT_PROT_SEC_96
R-1h	R-1h	R-1h	R-1h	R-1h	R-1h	R-1h	R-1h

表 11-24. CCFG_PROT_127_96 寄存器字段说明

位	字段	类型	复位	说明
31	WRT_PROT_SEC_127	R	1h	0：扇区受保护
30	WRT_PROT_SEC_126	R	1h	0：扇区受保护
29	WRT_PROT_SEC_125	R	1h	0：扇区受保护
28	WRT_PROT_SEC_124	R	1h	0：扇区受保护
27	WRT_PROT_SEC_123	R	1h	0：扇区受保护
26	WRT_PROT_SEC_122	R	1h	0：扇区受保护
25	WRT_PROT_SEC_121	R	1h	0：扇区受保护
24	WRT_PROT_SEC_120	R	1h	0：扇区受保护
23	WRT_PROT_SEC_119	R	1h	0：扇区受保护
22	WRT_PROT_SEC_118	R	1h	0：扇区受保护
21	WRT_PROT_SEC_117	R	1h	0：扇区受保护
20	WRT_PROT_SEC_116	R	1h	0：扇区受保护
19	WRT_PROT_SEC_115	R	1h	0：扇区受保护
18	WRT_PROT_SEC_114	R	1h	0：扇区受保护
17	WRT_PROT_SEC_113	R	1h	0：扇区受保护
16	WRT_PROT_SEC_112	R	1h	0：扇区受保护
15	WRT_PROT_SEC_111	R	1h	0：扇区受保护
14	WRT_PROT_SEC_110	R	1h	0：扇区受保护
13	WRT_PROT_SEC_109	R	1h	0：扇区受保护

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 11-24. CCFG_PROT_127_96 寄存器字段说明（续）

位	字段	类型	复位	说明
12	WRT_PROT_SEC_108	R	1h	0：扇区受保护
11	WRT_PROT_SEC_107	R	1h	0：扇区受保护
10	WRT_PROT_SEC_106	R	1h	0：扇区受保护
9	WRT_PROT_SEC_105	R	1h	0：扇区受保护
8	WRT_PROT_SEC_104	R	1h	0：扇区受保护
7	WRT_PROT_SEC_103	R	1h	0：扇区受保护
6	WRT_PROT_SEC_102	R	1h	0：扇区受保护
5	WRT_PROT_SEC_101	R	1h	0：扇区受保护
4	WRT_PROT_SEC_100	R	1h	0：扇区受保护
3	WRT_PROT_SEC_99	R	1h	0：扇区受保护
2	WRT_PROT_SEC_98	R	1h	0：扇区受保护
1	WRT_PROT_SEC_97	R	1h	0：扇区受保护
0	WRT_PROT_SEC_96	R	1h	0：扇区受保护

11.3 出厂配置 (FCFG)

通过 TI 生产测试为每个器件对 FCFG 进行编程。FCFG 包含特定于器件的修整值及配置。大多数修整值由 TI 引导代码、RF 内核、ROM 代码使用，或者由 TI 软件自动提供。

FCFG 中一些更有用的字段为：

- MAC_15_4_n 字段，这些字段提供芯片组的预编程 IEEE 地址
- MAC_BLE_n 字段，这些字段提供芯片组的低功耗蓝牙地址

11.4 FCFG 寄存器

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1 FCFG1 寄存器

表 11-25 列出了 FCFG1 寄存器的存储器映射寄存器。表 11-25 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

表 11-25. FCFG1 寄存器

偏移	首字母缩写词	寄存器名称	部分
A0h	MISC_CONF_1	其他配置	MISC_CONF_1 寄存器 (偏移 = A0h) [复位 = FFFFFF00h]
A4h	MISC_CONF_2	内部	MISC_CONF_2 寄存器 (偏移 = A4h) [复位 = FFFFFF00h]
B0h	HPOSC_MEAS_5	内部	HPOSC_MEAS_5 寄存器 (偏移 = B0h) [复位 = 00000000h]
B4h	HPOSC_MEAS_4	内部	HPOSC_MEAS_4 寄存器 (偏移 = B4h) [复位 = 00000000h]
B8h	HPOSC_MEAS_3	内部	HPOSC_MEAS_3 寄存器 (偏移 = B8h) [复位 = 00000000h]
BCh	HPOSC_MEAS_2	内部	HPOSC_MEAS_2 寄存器 (偏移 = BCh) [复位 = 00000000h]
C0h	HPOSC_MEAS_1	内部	HPOSC_MEAS_1 寄存器 (偏移 = C0h) [复位 = 00000000h]
C4h	CONFIG_CC26_FE	内部	CONFIG_CC26_FE 寄存器 (偏移 = C4h) [复位 = 70000F00h]
C8h	CONFIG_CC13_FE	内部	CONFIG_CC13_FE 寄存器 (偏移 = C8h) [复位 = 70000F00h]
CCh	CONFIG_RF_COMMON	内部	CONFIG_RF_COMMON 寄存器 (偏移 = CCh) [复位 = 81C0014Dh]
D0h	CONFIG_SYNTH_DIV2_CC26_2G4	内部	CONFIG_SYNTH_DIV2_CC26_2G4 寄存器 (偏移 = D0h) [复位 = 0000001Fh]
D4h	CONFIG_SYNTH_DIV2_CC13_2G4	内部	CONFIG_SYNTH_DIV2_CC13_2G4 寄存器 (偏移 = D4h) [复位 = 0000001Fh]
D8h	CONFIG_SYNTH_DIV2_CC26_1G	内部	CONFIG_SYNTH_DIV2_CC26_1G 寄存器 (偏移 = D8h) [复位 = 0000001Fh]
DCh	CONFIG_SYNTH_DIV2_CC13_1G	内部	CONFIG_SYNTH_DIV2_CC13_1G 寄存器 (偏移 = DCh) [复位 = 0000001Fh]
E0h	CONFIG_SYNTH_DIV4_CC26	内部	CONFIG_SYNTH_DIV4_CC26 寄存器 (偏移 = E0h) [复位 = 0000001Fh]
E4h	CONFIG_SYNTH_DIV4_CC13	内部	CONFIG_SYNTH_DIV4_CC13 寄存器 (偏移 = E4h) [复位 = 0000001Fh]
E8h	CONFIG_SYNTH_DIV5	内部	CONFIG_SYNTH_DIV5 寄存器 (偏移 = E8h) [复位 = 0000001Fh]
ECh	CONFIG_SYNTH_DIV6_CC26	内部	CONFIG_SYNTH_DIV6_CC26 寄存器 (偏移 = ECh) [复位 = 0000001Fh]
F0h	CONFIG_SYNTH_DIV6_CC13	内部	CONFIG_SYNTH_DIV6_CC13 寄存器 (偏移 = F0h) [复位 = 0000001Fh]
F4h	CONFIG_SYNTH_DIV10	内部	CONFIG_SYNTH_DIV10 寄存器 (偏移 = F4h) [复位 = 0000001Fh]
F8h	CONFIG_SYNTH_DIV12_CC26	内部	CONFIG_SYNTH_DIV12_CC26 寄存器 (偏移 = F8h) [复位 = 0000001Fh]
FCh	CONFIG_SYNTH_DIV12_CC13	内部	CONFIG_SYNTH_DIV12_CC13 寄存器 (偏移 = FCh) [复位 = 0000001Fh]
100h	CONFIG_SYNTH_DIV15	内部	CONFIG_SYNTH_DIV15 寄存器 (偏移 = 100h) [复位 = 0000001Fh]

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 11-25. FCFG1 寄存器（续）

偏移	首字母缩写词	寄存器名称	部分
104h	CONFIG_SYNTH_DIV30	内部	CONFIG_SYNTH_DIV30 寄存器（偏移 = 104h）[复位 = 0000001Fh]
164h	FLASH_NUMBER	闪存信息	FLASH_NUMBER 寄存器（偏移 = 164h）[复位 = 00000000h]
16Ch	FLASH_COORDINATE	闪存信息	FLASH_COORDINATE 寄存器（偏移 = 16Ch）[复位 = 00000000h]
170h	FLASH_E_P	内部	FLASH_E_P 寄存器（偏移 = 170h）[复位 = 4C644C64h]
174h	FLASH_C_E_P_R	内部	FLASH_C_E_P_R 寄存器（偏移 = 174h）[复位 = 0A0A2000h]
178h	FLASH_P_R_PV	内部	FLASH_P_R_PV 寄存器（偏移 = 178h）[复位 = 02C10200h]
17Ch	FLASH_EH_SEQ	内部	FLASH_EH_SEQ 寄存器（偏移 = 17Ch）[复位 = 0200F000h]
180h	FLASH_VHV_E	内部	FLASH_VHV_E 寄存器（偏移 = 180h）[复位 = 00000001h]
184h	FLASH_PP	内部	FLASH_PP 寄存器（偏移 = 184h）[复位 = 00000014h]
188h	FLASH_PROG_EP	内部	FLASH_PROG_EP 寄存器（偏移 = 188h）[复位 = 0FA00010h]
18Ch	FLASH_ERA_PW	内部	FLASH_ERA_PW 寄存器（偏移 = 18Ch）[复位 = 00000FA0h]
190h	FLASH_VHV	内部	FLASH_VHV 寄存器（偏移 = 190h）[复位 = 00000004h]
194h	FLASH_VHV_PV	内部	FLASH_VHV_PV 寄存器（偏移 = 194h）[复位 = 00080001h]
198h	FLASH_V	内部	FLASH_V 寄存器（偏移 = 198h）[复位 = 00000000h]
294h	USER_ID	用户标识。	USER_ID 寄存器（偏移 = 294h）[复位 = 30000000h]
2B0h	FLASH_OTP_DATA3	内部	FLASH_OTP_DATA3 寄存器（偏移 = 2B0h）[复位 = 00110003h]
2B4h	ANA2_TRIM	内部	ANA2_TRIM 寄存器（偏移 = 2B4h）[复位 = 8240787Fh]
2B8h	LDO_TRIM	内部	LDO_TRIM 寄存器（偏移 = 2B8h）[复位 = E0F8E0FBh]
2E8h	MAC_BLE_0	MAC BLE 地址 0	MAC_BLE_0 寄存器（偏移 = 2E8h）[复位 = 00000000h]
2ECh	MAC_BLE_1	MAC BLE 地址 1	MAC_BLE_1 寄存器（偏移 = 2ECh）[复位 = 00000000h]
2F0h	MAC_15_4_0	MAC IEEE 802.15.4 地址 0	MAC_15_4_0 寄存器（偏移 = 2F0h）[复位 = 00000000h]
2F4h	MAC_15_4_1	MAC IEEE 802.15.4 地址 1	MAC_15_4_1 寄存器（偏移 = 2F4h）[复位 = 00000000h]
308h	FLASH_OTP_DATA4	内部	FLASH_OTP_DATA4 寄存器（偏移 = 308h）[复位 = 98989F9Fh]
30Ch	MISC_TRIM	其他修整参数	MISC_TRIM 寄存器（偏移 = 30Ch）[复位 = FFFE003Bh]
310h	RCOSC_HF_TEMPCOMP	内部	RCOSC_HF_TEMPCOMP 寄存器（偏移 = 310h）[复位 = 00000003h]

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 11-25. FCFG1 寄存器（续）

偏移	首字母缩写词	寄存器名称	部分
318h	ICEPICK_DEVICE_ID	IcePick 器件标识	ICEPICK_DEVICE_ID 寄存器（偏移 = 318h）[复位 = 3BB4102Fh]
31Ch	FCFG1_REVISION	出厂配置 (FCFG1) 版本	FCFG1_REVISION 寄存器（偏移 = 31Ch）[复位 = 0000002Ah]
320h	MISC_OTP_DATA	其他 OTP 数据	MISC_OTP_DATA 寄存器（偏移 = 320h）[复位 = 0000CFFFh]
344h	IOCONF	IO 配置	IOCONF 寄存器（偏移 = 344h）[复位 = FFFFFF00h]
34Ch	CONFIG_IF_ADC	内部	CONFIG_IF_ADC 寄存器（偏移 = 34Ch）[复位 = 3460F400h]
350h	CONFIG_OSC_TOP	内部	CONFIG_OSC_TOP 寄存器（偏移 = 350h）[复位 = DC07FC00h]
35Ch	SOC_ADC_ABS_GAIN	绝对基准模式下的 AUX_ADC 增益	SOC_ADC_ABS_GAIN 寄存器（偏移 = 35Ch）[复位 = 00000000h]
360h	SOC_ADC_REL_GAIN	相对基准模式下的 AUX_ADC 增益	SOC_ADC_REL_GAIN 寄存器（偏移 = 360h）[复位 = 00000000h]
368h	SOC_ADC_OFFSET_INT	绝对基准模式下的 AUX_ADC 温度偏移	SOC_ADC_OFFSET_INT 寄存器（偏移 = 368h）[复位 = 00000000h]
36Ch	SOC_ADC_REF_TRIM_AND_OFFSET_EXT	内部	SOC_ADC_REF_TRIM_AND_OFFSET_EXT 寄存器（偏移 = 36Ch）[复位 = 0000C080h]
370h	AMPCOMP_TH1	内部	AMPCOMP_TH1 寄存器（偏移 = 370h）[复位 = FF7B828Eh]
374h	AMPCOMP_TH2	内部	AMPCOMP_TH2 寄存器（偏移 = 374h）[复位 = 6B8B0303h]
378h	AMPCOMP_CTRL1	内部	AMPCOMP_CTRL1 寄存器（偏移 = 378h）[复位 = FF483F47h]
37Ch	ANABYPASS_VALUE2	内部	ANABYPASS_VALUE2 寄存器（偏移 = 37Ch）[复位 = FFFFC3FFh]
388h	VOLT_TRIM	内部	VOLT_TRIM 寄存器（偏移 = 388h）[复位 = E0E0E0E0h]
38Ch	OSC_CONF	OSC 配置	OSC_CONF 寄存器（偏移 = 38Ch）[复位 = F00900E6h]
390h	FREQ_OFFSET	内部	FREQ_OFFSET 寄存器（偏移 = 390h）[复位 = 00000000h]
398h	MISC_OTP_DATA_1	内部	MISC_OTP_DATA_1 寄存器（偏移 = 398h）[复位 = E08403F8h]
3D0h	SHDW_DIE_ID_0	EFUSE:DIE_ID_0 寄存器的影子寄存器	SHDW_DIE_ID_0 寄存器（偏移 = 3D0h）[复位 = 00000000h]
3D4h	SHDW_DIE_ID_1	EFUSE:DIE_ID_1 寄存器的影子寄存器	SHDW_DIE_ID_1 寄存器（偏移 = 3D4h）[复位 = 00000000h]
3D8h	SHDW_DIE_ID_2	EFUSE:DIE_ID_2 寄存器的影子寄存器	SHDW_DIE_ID_2 寄存器（偏移 = 3D8h）[复位 = 00000000h]
3DCh	SHDW_DIE_ID_3	EFUSE:DIE_ID_3 寄存器的影子寄存器	SHDW_DIE_ID_3 寄存器（偏移 = 3DCh）[复位 = 00000000h]
3F8h	SHDW_OSC_BIAS_LDO_TRIM	内部	SHDW_OSC_BIAS_LDO_TRIM 寄存器（偏移 = 3F8h）[复位 = 00000000h]
3FCh	SHDW_ANA_TRIM	内部	SHDW_ANA_TRIM 寄存器（偏移 = 3FCh）[复位 = 00000000h]
408h	OSC_CONF1	振荡器配置	OSC_CONF1 寄存器（偏移 = 408h）[复位 = FFFF0000h]

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 11-25. FCFG1 寄存器（续）

偏移	首字母缩写词	寄存器名称	部分
40Ch	DAC_BIAS_CNF	内部	DAC_BIAS_CNF 寄存器（偏移 = 40Ch）[复位 = FFFC00FFh]
418h	TFW_PROBE	内部	TFW_PROBE 寄存器（偏移 = 418h）[复位 = 00000000h]
41Ch	TFW_FT	内部	TFW_FT 寄存器（偏移 = 41Ch）[复位 = 00000000h]
420h	DAC_CAL0	内部	DAC_CAL0 寄存器（偏移 = 420h）[复位 = 00000000h]
424h	DAC_CAL1	内部	DAC_CAL1 寄存器（偏移 = 424h）[复位 = 00000000h]
428h	DAC_CAL2	内部	DAC_CAL2 寄存器（偏移 = 428h）[复位 = 00000000h]
42Ch	DAC_CAL3	内部	DAC_CAL3 寄存器（偏移 = 42Ch）[复位 = 00000000h]

复杂的位访问类型经过编码可适应小型表单元。表 11-26 展示了适用于此部分中访问类型的代码。

表 11-26. FCFG1 访问类型代码

访问类型	代码	说明
读取类型		
R	R	读取
复位或默认值		
-n		复位后的值或默认值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.1 MISC_CONF_1 寄存器（偏移 = A0h）[复位 = FFFFFF00h]

图 11-23 展示了 MISC_CONF_1，表 11-27 对其进行了介绍。

返回到汇总表。

其他配置

图 11-23. MISC_CONF_1 寄存器

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								DEVICE_MINOR_REV							
R-0h								R-X							

表 11-27. MISC_CONF_1 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-0	DEVICE_MINOR_REV	R	X	硬件次要版本号（值 0xFF 应等同于 0x00）。 软件对该字段的任何测试都应作为有符号整数的“大于或等于”比较方式来实现。 值可能会在没有警告的情况下发生变化。 默认值保存来自生产测试的日志信息。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.2 MISC_CONF_2 寄存器（偏移 = A4h）[复位 = FFFFFF00h]

图 11-24 展示了 MISC_CONF_2，表 11-28 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-24. MISC_CONF_2 寄存器

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								HPOSC_COMP_P3							
R-0h								R-X							

表 11-28. MISC_CONF_2 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-0	HPOSC_COMP_P3	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.3 HPOSC_MEAS_5 寄存器（偏移 = B0h）[复位 = 00000000h]

图 11-25 展示了 HPOSC_MEAS_5，表 11-29 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-25. HPOSC_MEAS_5 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HPOSC_D5															
R-X															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HPOSC_T5								HPOSC_DT5							
R-X								R-X							

表 11-29. HPOSC_MEAS_5 寄存器字段说明

位	字段	类型	复位	说明
31-16	HPOSC_D5	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的日志信息。
15-8	HPOSC_T5	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的日志信息。
7-0	HPOSC_DT5	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的日志信息。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.4 HPOSC_MEAS_4 寄存器（偏移 = B4h）[复位 = 00000000h]

图 11-26 展示了 HPOSC_MEAS_4，表 11-30 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-26. HPOSC_MEAS_4 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HPOSC_D4															
R-X															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HPOSC_T4								HPOSC_DT4							
R-X								R-X							

表 11-30. HPOSC_MEAS_4 寄存器字段说明

位	字段	类型	复位	说明
31-16	HPOSC_D4	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的日志信息。
15-8	HPOSC_T4	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的日志信息。
7-0	HPOSC_DT4	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的日志信息。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.5 HPOSC_MEAS_3 寄存器（偏移 = B8h）[复位 = 00000000h]

图 11-27 展示了 HPOSC_MEAS_3，表 11-31 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-27. HPOSC_MEAS_3 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HPOSC_D3															
R-X															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HPOSC_T3								HPOSC_DT3							
R-X								R-X							

表 11-31. HPOSC_MEAS_3 寄存器字段说明

位	字段	类型	复位	说明
31-16	HPOSC_D3	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的日志信息。
15-8	HPOSC_T3	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的日志信息。
7-0	HPOSC_DT3	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的日志信息。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.6 HPOSC_MEAS_2 寄存器（偏移 = BCh）[复位 = 00000000h]

图 11-28 展示了 HPOSC_MEAS_2，表 11-32 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-28. HPOSC_MEAS_2 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HPOSC_D2															
R-X															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HPOSC_T2								HPOSC_DT2							
R-X								R-X							

表 11-32. HPOSC_MEAS_2 寄存器字段说明

位	字段	类型	复位	说明
31-16	HPOSC_D2	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的日志信息。
15-8	HPOSC_T2	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的日志信息。
7-0	HPOSC_DT2	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的日志信息。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.7 HPOSC_MEAS_1 寄存器（偏移 = C0h）[复位 = 00000000h]

图 11-29 展示了 HPOSC_MEAS_1，表 11-33 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-29. HPOSC_MEAS_1 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HPOSC_D1															
R-X															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HPOSC_T1								HPOSC_DT1							
R-X								R-X							

表 11-33. HPOSC_MEAS_1 寄存器字段说明

位	字段	类型	复位	说明
31-16	HPOSC_D1	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的日志信息。
15-8	HPOSC_T1	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的日志信息。
7-0	HPOSC_DT1	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的日志信息。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.8 CONFIG_CC26_FE 寄存器（偏移 = C4h）[复位 = 70000F00h]

图 11-30 展示了 CONFIG_CC26_FE，表 11-34 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-30. CONFIG_CC26_FE 寄存器

31	30	29	28	27	26	25	24
IFAMP_IB				LNA_IB			
R-7h				R-X			
23	22	21	20	19	18	17	16
IFAMP_TRIM					CTL_PA0_TRIM		
R-0h					R-X		
15	14	13	12	11	10	9	8
CTL_PA0_TRIM		PATRIMCOMPLETE_N	RSSITRIMCOMPLETE_N	RESERVED			
R-X		R-X	R-X	R-0h			
7	6	5	4	3	2	1	0
RSSI_OFFSET							
R-X							

表 11-34. CONFIG_CC26_FE 寄存器字段说明

位	字段	类型	复位	说明
31-28	IFAMP_IB	R	7h	内部。仅可通过 TI 提供的 API 使用。
27-24	LNA_IB	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
23-19	IFAMP_TRIM	R	0h	内部。仅可通过 TI 提供的 API 使用。
18-14	CTL_PA0_TRIM	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
13	PATRIMCOMPLETE_N	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
12	RSSITRIMCOMPLETE_N	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
11-8	RESERVED	R	0h	保留
7-0	RSSI_OFFSET	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.9 CONFIG_CC13_FE 寄存器（偏移 = C8h）[复位 = 70000F00h]

图 11-31 展示了 CONFIG_CC13_FE，表 11-35 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-31. CONFIG_CC13_FE 寄存器

31	30	29	28	27	26	25	24
IFAMP_IB				LNA_IB			
R-7h				R-X			
23	22	21	20	19	18	17	16
IFAMP_TRIM					CTL_PA0_TRIM		
R-0h					R-X		
15	14	13	12	11	10	9	8
CTL_PA0_TRIM		PATRIMCOMPLETE_N	RSSITRIMCOMPLETE_N	RESERVED			
R-X		R-X	R-X	R-0h			
7	6	5	4	3	2	1	0
RSSI_OFFSET							
R-X							

表 11-35. CONFIG_CC13_FE 寄存器字段说明

位	字段	类型	复位	说明
31-28	IFAMP_IB	R	7h	内部。仅可通过 TI 提供的 API 使用。
27-24	LNA_IB	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
23-19	IFAMP_TRIM	R	0h	内部。仅可通过 TI 提供的 API 使用。
18-14	CTL_PA0_TRIM	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
13	PATRIMCOMPLETE_N	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
12	RSSITRIMCOMPLETE_N	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
11-8	RESERVED	R	0h	保留
7-0	RSSI_OFFSET	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.10 CONFIG_RF_COMMON 寄存器（偏移 = CCh）[复位 = 81C0014Dh]

图 11-32 展示了 CONFIG_RF_COMMON，表 11-36 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-32. CONFIG_RF_COMMON 寄存器

31	30	29	28	27	26	25	24
DISABLE_CORNER_CAP	SLDO_TRIM_OUTPUT						RESERVED
R-1h	R-X						R-0h
23	22	21	20	19	18	17	16
RESERVED		PA20DBMTRIMCOMPLETE_N	CTL_PA_20DBM_TRIM				
R-0h		R-X	R-X				
15	14	13	12	11	10	9	8
RFLDO_TRIM_OUTPUT							QUANTCTLTHRES
R-X							R-5h
7	6	5	4	3	2	1	0
QUANTCTLTHRES		DACTRIM					
R-5h		R-Dh					

表 11-36. CONFIG_RF_COMMON 寄存器字段说明

位	字段	类型	复位	说明
31	DISABLE_CORNER_CAP	R	1h	内部。仅可通过 TI 提供的 API 使用。
30-25	SLDO_TRIM_OUTPUT	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
24-22	RESERVED	R	0h	保留
21	PA20DBMTRIMCOMPLETE_N	R	X	内部。仅可通过 TI 提供的 API 使用。
20-16	CTL_PA_20DBM_TRIM	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
15-9	RFLDO_TRIM_OUTPUT	R	X	内部。仅可通过 TI 提供的 API 使用。
8-6	QUANTCTLTHRES	R	5h	内部。仅可通过 TI 提供的 API 使用。
5-0	DACTRIM	R	Dh	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.11 CONFIG_SYNTH_DIV2_CC26_2G4 寄存器（偏移 = D0h）[复位 = 0000001Fh]

图 11-33 展示了 CONFIG_SYNTH_DIV2_CC26_2G4，表 11-37 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-33. CONFIG_SYNTH_DIV2_CC26_2G4 寄存器

31	30	29	28	27	26	25	24
MIN_ALLOWED_RTRIM				RFC_MDM_DEMIQMC0			
R-X				R-X			
23	22	21	20	19	18	17	16
RFC_MDM_DEMIQMC0							
R-X							
15	14	13	12	11	10	9	8
RFC_MDM_DEMIQMC0				LDOVCO_TRIM_OUTPUT			
R-X				R-X			
7	6	5	4	3	2	1	0
LDOVCO_TRIM_OUTPUT		RFC_MDM_DEMIQMC0_TRIMCOMPLETE_N	RESERVED				
R-X		R-X	R-0h				

表 11-37. CONFIG_SYNTH_DIV2_CC26_2G4 寄存器字段说明

位	字段	类型	复位	说明
31-28	MIN_ALLOWED_RTRIM	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
27-12	RFC_MDM_DEMIQMC0	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
11-6	LDOVCO_TRIM_OUTPUT	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
5	RFC_MDM_DEMIQMC0_TRIMCOMPLETE_N	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
4-0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.12 CONFIG_SYNTH_DIV2_CC13_2G4 寄存器（偏移 = D4h）[复位 = 0000001Fh]

图 11-34 展示了 CONFIG_SYNTH_DIV2_CC13_2G4，表 11-38 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-34. CONFIG_SYNTH_DIV2_CC13_2G4 寄存器

31	30	29	28	27	26	25	24
MIN_ALLOWED_RTRIM				RFC_MDM_DEMIQMC0			
R-X				R-X			
23	22	21	20	19	18	17	16
RFC_MDM_DEMIQMC0							
R-X							
15	14	13	12	11	10	9	8
RFC_MDM_DEMIQMC0				LDOVCO_TRIM_OUTPUT			
R-X				R-X			
7	6	5	4	3	2	1	0
LDOVCO_TRIM_OUTPUT		RFC_MDM_DEMIQMC0_TRIMCOMPLETE_N	RESERVED				
R-X		R-X	R-0h				

表 11-38. CONFIG_SYNTH_DIV2_CC13_2G4 寄存器字段说明

位	字段	类型	复位	说明
31-28	MIN_ALLOWED_RTRIM	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
27-12	RFC_MDM_DEMIQMC0	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
11-6	LDOVCO_TRIM_OUTPUT	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
5	RFC_MDM_DEMIQMC0_TRIMCOMPLETE_N	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
4-0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.13 CONFIG_SYNTH_DIV2_CC26_1G 寄存器（偏移 = D8h）[复位 = 0000001Fh]

图 11-35 展示了 CONFIG_SYNTH_DIV2_CC26_1G，表 11-39 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-35. CONFIG_SYNTH_DIV2_CC26_1G 寄存器

31	30	29	28	27	26	25	24
MIN_ALLOWED_RTRIM				RFC_MDM_DEMIQMC0			
R-X				R-X			
23	22	21	20	19	18	17	16
RFC_MDM_DEMIQMC0							
R-X							
15	14	13	12	11	10	9	8
RFC_MDM_DEMIQMC0				LDOVCO_TRIM_OUTPUT			
R-X				R-X			
7	6	5	4	3	2	1	0
LDOVCO_TRIM_OUTPUT		RFC_MDM_DEMIQMC0_TRIMCOMPLETE_N	RESERVED				
R-X		R-X	R-0h				

表 11-39. CONFIG_SYNTH_DIV2_CC26_1G 寄存器字段说明

位	字段	类型	复位	说明
31-28	MIN_ALLOWED_RTRIM	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
27-12	RFC_MDM_DEMIQMC0	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
11-6	LDOVCO_TRIM_OUTPUT	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
5	RFC_MDM_DEMIQMC0_TRIMCOMPLETE_N	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
4-0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.14 CONFIG_SYNTH_DIV2_CC13_1G 寄存器（偏移 = DCh）[复位 = 0000001Fh]

图 11-36 展示了 CONFIG_SYNTH_DIV2_CC13_1G，表 11-40 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-36. CONFIG_SYNTH_DIV2_CC13_1G 寄存器

31	30	29	28	27	26	25	24
MIN_ALLOWED_RTRIM				RFC_MDM_DEMIQMC0			
R-X				R-X			
23	22	21	20	19	18	17	16
RFC_MDM_DEMIQMC0							
R-X							
15	14	13	12	11	10	9	8
RFC_MDM_DEMIQMC0				LDOVCO_TRIM_OUTPUT			
R-X				R-X			
7	6	5	4	3	2	1	0
LDOVCO_TRIM_OUTPUT		RFC_MDM_DEMIQMC0_TRIMCOMPLETE_N	RESERVED				
R-X		R-X	R-0h				

表 11-40. CONFIG_SYNTH_DIV2_CC13_1G 寄存器字段说明

位	字段	类型	复位	说明
31-28	MIN_ALLOWED_RTRIM	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
27-12	RFC_MDM_DEMIQMC0	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
11-6	LDOVCO_TRIM_OUTPUT	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
5	RFC_MDM_DEMIQMC0_TRIMCOMPLETE_N	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
4-0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.15 CONFIG_SYNTH_DIV4_CC26 寄存器（偏移 = E0h）[复位 = 0000001Fh]

图 11-37 展示了 CONFIG_SYNTH_DIV4_CC26，表 11-41 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-37. CONFIG_SYNTH_DIV4_CC26 寄存器

31	30	29	28	27	26	25	24
MIN_ALLOWED_RTRIM				RFC_MDM_DEMIQMC0			
R-X				R-X			
23	22	21	20	19	18	17	16
RFC_MDM_DEMIQMC0							
R-X							
15	14	13	12	11	10	9	8
RFC_MDM_DEMIQMC0				LDOVCO_TRIM_OUTPUT			
R-X				R-X			
7	6	5	4	3	2	1	0
LDOVCO_TRIM_OUTPUT		RFC_MDM_DEMIQMC0_TRIMCOMPLETE_N	RESERVED				
R-X		R-X	R-0h				

表 11-41. CONFIG_SYNTH_DIV4_CC26 寄存器字段说明

位	字段	类型	复位	说明
31-28	MIN_ALLOWED_RTRIM	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
27-12	RFC_MDM_DEMIQMC0	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
11-6	LDOVCO_TRIM_OUTPUT	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
5	RFC_MDM_DEMIQMC0_TRIMCOMPLETE_N	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
4-0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.16 CONFIG_SYNTH_DIV4_CC13 寄存器（偏移 = E4h）[复位 = 0000001Fh]

图 11-38 展示了 CONFIG_SYNTH_DIV4_CC13，表 11-42 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-38. CONFIG_SYNTH_DIV4_CC13 寄存器

31	30	29	28	27	26	25	24
MIN_ALLOWED_RTRIM				RFC_MDM_DEMIQMC0			
R-X				R-X			
23	22	21	20	19	18	17	16
RFC_MDM_DEMIQMC0							
R-X							
15	14	13	12	11	10	9	8
RFC_MDM_DEMIQMC0				LDOVCO_TRIM_OUTPUT			
R-X				R-X			
7	6	5	4	3	2	1	0
LDOVCO_TRIM_OUTPUT		RFC_MDM_DEMIQMC0_TRIMCOMPLETE_N	RESERVED				
R-X		R-X	R-0h				

表 11-42. CONFIG_SYNTH_DIV4_CC13 寄存器字段说明

位	字段	类型	复位	说明
31-28	MIN_ALLOWED_RTRIM	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
27-12	RFC_MDM_DEMIQMC0	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
11-6	LDOVCO_TRIM_OUTPUT	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
5	RFC_MDM_DEMIQMC0_TRIMCOMPLETE_N	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
4-0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.17 CONFIG_SYNTH_DIV5 寄存器（偏移 = E8h）[复位 = 0000001Fh]

图 11-39 展示了 CONFIG_SYNTH_DIV5，表 11-43 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-39. CONFIG_SYNTH_DIV5 寄存器

31	30	29	28	27	26	25	24
MIN_ALLOWED_RTRIM				RFC_MDM_DEMIQMC0			
R-X				R-X			
23	22	21	20	19	18	17	16
RFC_MDM_DEMIQMC0							
R-X							
15	14	13	12	11	10	9	8
RFC_MDM_DEMIQMC0				LDOVCO_TRIM_OUTPUT			
R-X				R-X			
7	6	5	4	3	2	1	0
LDOVCO_TRIM_OUTPUT		RFC_MDM_DEMIQMC0_TRIMCOMPLETE_N	RESERVED				
R-X		R-X	R-0h				

表 11-43. CONFIG_SYNTH_DIV5 寄存器字段说明

位	字段	类型	复位	说明
31-28	MIN_ALLOWED_RTRIM	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
27-12	RFC_MDM_DEMIQMC0	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
11-6	LDOVCO_TRIM_OUTPUT	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
5	RFC_MDM_DEMIQMC0_TRIMCOMPLETE_N	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
4-0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.18 CONFIG_SYNTH_DIV6_CC26 寄存器（偏移 = ECh）[复位 = 0000001Fh]

图 11-40 展示了 CONFIG_SYNTH_DIV6_CC26，表 11-44 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-40. CONFIG_SYNTH_DIV6_CC26 寄存器

31	30	29	28	27	26	25	24
MIN_ALLOWED_RTRIM				RFC_MDM_DEMIQMC0			
R-X				R-X			
23	22	21	20	19	18	17	16
RFC_MDM_DEMIQMC0							
R-X							
15	14	13	12	11	10	9	8
RFC_MDM_DEMIQMC0				LDOVCO_TRIM_OUTPUT			
R-X				R-X			
7	6	5	4	3	2	1	0
LDOVCO_TRIM_OUTPUT		RFC_MDM_DEMIQMC0_TRIMCOMPLETE_N	RESERVED				
R-X		R-X	R-0h				

表 11-44. CONFIG_SYNTH_DIV6_CC26 寄存器字段说明

位	字段	类型	复位	说明
31-28	MIN_ALLOWED_RTRIM	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
27-12	RFC_MDM_DEMIQMC0	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
11-6	LDOVCO_TRIM_OUTPUT	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
5	RFC_MDM_DEMIQMC0_TRIMCOMPLETE_N	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
4-0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.19 CONFIG_SYNTH_DIV6_CC13 寄存器（偏移 = F0h）[复位 = 0000001Fh]

图 11-41 展示了 CONFIG_SYNTH_DIV6_CC13，表 11-45 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-41. CONFIG_SYNTH_DIV6_CC13 寄存器

31	30	29	28	27	26	25	24
MIN_ALLOWED_RTRIM				RFC_MDM_DEMIQMC0			
R-X				R-X			
23	22	21	20	19	18	17	16
RFC_MDM_DEMIQMC0							
R-X							
15	14	13	12	11	10	9	8
RFC_MDM_DEMIQMC0				LDOVCO_TRIM_OUTPUT			
R-X				R-X			
7	6	5	4	3	2	1	0
LDOVCO_TRIM_OUTPUT		RFC_MDM_DEMIQMC0_TRIMCOMPLETE_N	RESERVED				
R-X		R-X	R-0h				

表 11-45. CONFIG_SYNTH_DIV6_CC13 寄存器字段说明

位	字段	类型	复位	说明
31-28	MIN_ALLOWED_RTRIM	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
27-12	RFC_MDM_DEMIQMC0	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
11-6	LDOVCO_TRIM_OUTPUT	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
5	RFC_MDM_DEMIQMC0_TRIMCOMPLETE_N	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
4-0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.20 CONFIG_SYNTH_DIV10 寄存器（偏移 = F4h）[复位 = 0000001Fh]

图 11-42 展示了 CONFIG_SYNTH_DIV10，表 11-46 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-42. CONFIG_SYNTH_DIV10 寄存器

31	30	29	28	27	26	25	24
MIN_ALLOWED_RTRIM				RFC_MDM_DEMIQMC0			
R-X				R-X			
23	22	21	20	19	18	17	16
RFC_MDM_DEMIQMC0							
R-X							
15	14	13	12	11	10	9	8
RFC_MDM_DEMIQMC0				LDOVCO_TRIM_OUTPUT			
R-X				R-X			
7	6	5	4	3	2	1	0
LDOVCO_TRIM_OUTPUT		RFC_MDM_DEMIQMC0_TRIMCOMPLETE_N	RESERVED				
R-X		R-X	R-0h				

表 11-46. CONFIG_SYNTH_DIV10 寄存器字段说明

位	字段	类型	复位	说明
31-28	MIN_ALLOWED_RTRIM	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
27-12	RFC_MDM_DEMIQMC0	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
11-6	LDOVCO_TRIM_OUTPUT	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
5	RFC_MDM_DEMIQMC0_TRIMCOMPLETE_N	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
4-0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.21 CONFIG_SYNTH_DIV12_CC26 寄存器（偏移 = F8h）[复位 = 0000001Fh]

图 11-43 展示了 CONFIG_SYNTH_DIV12_CC26，表 11-47 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-43. CONFIG_SYNTH_DIV12_CC26 寄存器

31	30	29	28	27	26	25	24
MIN_ALLOWED_RTRIM				RFC_MDM_DEMIQMC0			
R-X				R-X			
23	22	21	20	19	18	17	16
RFC_MDM_DEMIQMC0							
R-X							
15	14	13	12	11	10	9	8
RFC_MDM_DEMIQMC0				LDOVCO_TRIM_OUTPUT			
R-X				R-X			
7	6	5	4	3	2	1	0
LDOVCO_TRIM_OUTPUT		RFC_MDM_DEMIQMC0_TRIMCOMPLETE_N	RESERVED				
R-X		R-X	R-0h				

表 11-47. CONFIG_SYNTH_DIV12_CC26 寄存器字段说明

位	字段	类型	复位	说明
31-28	MIN_ALLOWED_RTRIM	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
27-12	RFC_MDM_DEMIQMC0	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
11-6	LDOVCO_TRIM_OUTPUT	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
5	RFC_MDM_DEMIQMC0_TRIMCOMPLETE_N	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
4-0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.22 CONFIG_SYNTH_DIV12_CC13 寄存器（偏移 = FCh）[复位 = 0000001Fh]

图 11-44 展示了 CONFIG_SYNTH_DIV12_CC13，表 11-48 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-44. CONFIG_SYNTH_DIV12_CC13 寄存器

31	30	29	28	27	26	25	24
MIN_ALLOWED_RTRIM				RFC_MDM_DEMIQMC0			
R-X				R-X			
23	22	21	20	19	18	17	16
RFC_MDM_DEMIQMC0							
R-X							
15	14	13	12	11	10	9	8
RFC_MDM_DEMIQMC0				LDOVCO_TRIM_OUTPUT			
R-X				R-X			
7	6	5	4	3	2	1	0
LDOVCO_TRIM_OUTPUT		RFC_MDM_DEMIQMC0_TRIMCOMPLETE_N	RESERVED				
R-X		R-X	R-0h				

表 11-48. CONFIG_SYNTH_DIV12_CC13 寄存器字段说明

位	字段	类型	复位	说明
31-28	MIN_ALLOWED_RTRIM	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
27-12	RFC_MDM_DEMIQMC0	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
11-6	LDOVCO_TRIM_OUTPUT	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
5	RFC_MDM_DEMIQMC0_TRIMCOMPLETE_N	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
4-0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.23 CONFIG_SYNTH_DIV15 寄存器（偏移 = 100h）[复位 = 0000001Fh]

图 11-45 展示了 CONFIG_SYNTH_DIV15，表 11-49 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-45. CONFIG_SYNTH_DIV15 寄存器

31	30	29	28	27	26	25	24
MIN_ALLOWED_RTRIM				RFC_MDM_DEMIQMC0			
R-X				R-X			
23	22	21	20	19	18	17	16
RFC_MDM_DEMIQMC0							
R-X							
15	14	13	12	11	10	9	8
RFC_MDM_DEMIQMC0				LDOVCO_TRIM_OUTPUT			
R-X				R-X			
7	6	5	4	3	2	1	0
LDOVCO_TRIM_OUTPUT		RFC_MDM_DEMIQMC0_TRIMCOMPLETE_N	RESERVED				
R-X		R-X	R-0h				

表 11-49. CONFIG_SYNTH_DIV15 寄存器字段说明

位	字段	类型	复位	说明
31-28	MIN_ALLOWED_RTRIM	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
27-12	RFC_MDM_DEMIQMC0	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
11-6	LDOVCO_TRIM_OUTPUT	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
5	RFC_MDM_DEMIQMC0_TRIMCOMPLETE_N	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
4-0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.24 CONFIG_SYNTH_DIV30 寄存器（偏移 = 104h）[复位 = 0000001Fh]

图 11-46 展示了 CONFIG_SYNTH_DIV30，表 11-50 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-46. CONFIG_SYNTH_DIV30 寄存器

31	30	29	28	27	26	25	24
MIN_ALLOWED_RTRIM				RFC_MDM_DEMIQMC0			
R-X				R-X			
23	22	21	20	19	18	17	16
RFC_MDM_DEMIQMC0							
R-X							
15	14	13	12	11	10	9	8
RFC_MDM_DEMIQMC0				LDOVCO_TRIM_OUTPUT			
R-X				R-X			
7	6	5	4	3	2	1	0
LDOVCO_TRIM_OUTPUT		RFC_MDM_DEMIQMC0_TRIMCOMPLETE_N	RESERVED				
R-X		R-X	R-0h				

表 11-50. CONFIG_SYNTH_DIV30 寄存器字段说明

位	字段	类型	复位	说明
31-28	MIN_ALLOWED_RTRIM	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
27-12	RFC_MDM_DEMIQMC0	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
11-6	LDOVCO_TRIM_OUTPUT	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
5	RFC_MDM_DEMIQMC0_TRIMCOMPLETE_N	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
4-0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.25 FLASH_NUMBER 寄存器（偏移 = 164h）[复位 = 00000000h]

图 11-47 展示了 FLASH_NUMBER，表 11-51 对其进行了介绍。

返回到汇总表。

闪存信息

图 11-47. FLASH_NUMBER 寄存器

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

表 11-51. FLASH_NUMBER 寄存器字段说明

位	字段	类型	复位	说明
31-0	LOT_NUMBER	R	X	生产该器件的制造批次号。 默认值保存来自生产测试的日志信息。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.26 FLASH_COORDINATE 寄存器（偏移 = 16Ch）[复位 = 00000000h]

图 11-48 展示了 FLASH_COORDINATE，表 11-52 对其进行了介绍。

返回到汇总表。

闪存信息

图 11-48. FLASH_COORDINATE 寄存器

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

表 11-52. FLASH_COORDINATE 寄存器字段说明

位	字段	类型	复位	说明
31-16	XCOORDINATE	R	X	该器件在晶圆上的 X 坐标。 默认值保存来自生产测试的日志信息。
15-0	YCOORDINATE	R	X	该器件在晶圆上的 Y 坐标。 默认值保存来自生产测试的日志信息。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.27 FLASH_E_P 寄存器（偏移 = 170h）[复位 = 4C644C64h]

图 11-49 展示了 FLASH_E_P，表 11-53 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-49. FLASH_E_P 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSU								ESU								PVSU								EVSU							
R-4Ch								R-64h								R-4Ch								R-64h							

表 11-53. FLASH_E_P 寄存器字段说明

位	字段	类型	复位	说明
31-24	PSU	R	4Ch	内部。仅可通过 TI 提供的 API 使用。
23-16	ESU	R	64h	内部。仅可通过 TI 提供的 API 使用。
15-8	PVSU	R	4Ch	内部。仅可通过 TI 提供的 API 使用。
7-0	EVSU	R	64h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.28 FLASH_C_E_P_R 寄存器（偏移 = 174h）[复位 = 0A0A2000h]

图 11-50 展示了 FLASH_C_E_P_R，表 11-54 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-50. FLASH_C_E_P_R 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RVSU								PV_ACCESS							
R-Ah								R-Ah							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A_EXEZ_SETUP				CVSU											
R-2h				R-0h											

表 11-54. FLASH_C_E_P_R 寄存器字段说明

位	字段	类型	复位	说明
31-24	RVSU	R	Ah	内部。仅可通过 TI 提供的 API 使用。
23-16	PV_ACCESS	R	Ah	内部。仅可通过 TI 提供的 API 使用。
15-12	A_EXEZ_SETUP	R	2h	内部。仅可通过 TI 提供的 API 使用。
11-0	CVSU	R	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.29 FLASH_P_R_PV 寄存器（偏移 = 178h）[复位 = 02C10200h]

图 11-51 展示了 FLASH_P_R_PV，表 11-55 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-51. FLASH_P_R_PV 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PH								RH								PVH								PVH2							
R-2h								R-C1h								R-2h								R-0h							

表 11-55. FLASH_P_R_PV 寄存器字段说明

位	字段	类型	复位	说明
31-24	PH	R	2h	内部。仅可通过 TI 提供的 API 使用。
23-16	RH	R	C1h	内部。仅可通过 TI 提供的 API 使用。
15-8	PVH	R	2h	内部。仅可通过 TI 提供的 API 使用。
7-0	PVH2	R	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.30 FLASH_EH_SEQ 寄存器（偏移 = 17Ch）[复位 = 0200F000h]

图 11-52 展示了 FLASH_EH_SEQ，表 11-56 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-52. FLASH_EH_SEQ 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EH								SEQ							
R-2h								R-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VSTAT				SM_FREQUENCY											
R-Fh				R-0h											

表 11-56. FLASH_EH_SEQ 寄存器字段说明

位	字段	类型	复位	说明
31-24	EH	R	2h	内部。仅可通过 TI 提供的 API 使用。
23-16	SEQ	R	0h	内部。仅可通过 TI 提供的 API 使用。
15-12	VSTAT	R	Fh	内部。仅可通过 TI 提供的 API 使用。
11-0	SM_FREQUENCY	R	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.31 FLASH_VHV_E 寄存器（偏移 = 180h）[复位 = 00000001h]

图 11-53 展示了 FLASH_VHV_E，表 11-57 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-53. FLASH_VHV_E 寄存器

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

表 11-57. FLASH_VHV_E 寄存器字段说明

位	字段	类型	复位	说明
31-16	VHV_E_START	R	0h	内部。仅可通过 TI 提供的 API 使用。
15-0	VHV_E_STEP_HIGHT	R	1h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.32 FLASH_PP 寄存器（偏移 = 184h）[复位 = 00000014h]

图 11-54 展示了 FLASH_PP，表 11-58 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-54. FLASH_PP 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PUMP_SU								TRIM3P4								MAX_PP															
R-0h								R-X								R-14h															

表 11-58. FLASH_PP 寄存器字段说明

位	字段	类型	复位	说明
31-24	PUMP_SU	R	0h	内部。仅可通过 TI 提供的 API 使用。
23-16	TRIM3P4	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
15-0	MAX_PP	R	14h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.33 FLASH_PROG_EP 寄存器（偏移 = 188h）[复位 = 0FA00010h]

图 11-55 展示了 FLASH_PROG_EP，表 11-59 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-55. FLASH_PROG_EP 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAX_EP																PROGRAM_PW															
R-FA0h																R-10h															

表 11-59. FLASH_PROG_EP 寄存器字段说明

位	字段	类型	复位	说明
31-16	MAX_EP	R	FA0h	内部。仅可通过 TI 提供的 API 使用。
15-0	PROGRAM_PW	R	10h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.34 FLASH_ERA_PW 寄存器（偏移 = 18Ch）[复位 = 00000FA0h]

图 11-56 展示了 FLASH_ERA_PW，表 11-60 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-56. FLASH_ERA_PW 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERASE_PW																															
R-FA0h																															

表 11-60. FLASH_ERA_PW 寄存器字段说明

位	字段	类型	复位	说明
31-0	ERASE_PW	R	FA0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.35 FLASH_VHV 寄存器（偏移 = 190h）[复位 = 00000004h]

图 11-57 展示了 FLASH_VHV，表 11-61 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-57. FLASH_VHV 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED				TRIM13_P				RESERVED				VHV_P			
R-0h				R-X				R-0h				R-X			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				TRIM13_E				RESERVED				VHV_E			
R-0h				R-X				R-0h				R-4h			

表 11-61. FLASH_VHV 寄存器字段说明

位	字段	类型	复位	说明
31-28	RESERVED	R	0h	保留
27-24	TRIM13_P	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
23-20	RESERVED	R	0h	保留
19-16	VHV_P	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
15-12	RESERVED	R	0h	保留
11-8	TRIM13_E	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
7-4	RESERVED	R	0h	保留
3-0	VHV_E	R	4h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.36 FLASH_VHV_PV 寄存器（偏移 = 194h）[复位 = 00080001h]

图 11-58 展示了 FLASH_VHV_PV，表 11-62 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-58. FLASH_VHV_PV 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED				TRIM13_PV				RESERVED				VHV_PV			
R-0h				R-X				R-0h				R-8h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VCG2P5								VINH							
R-X								R-1h							

表 11-62. FLASH_VHV_PV 寄存器字段说明

位	字段	类型	复位	说明
31-28	RESERVED	R	0h	保留
27-24	TRIM13_PV	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
23-20	RESERVED	R	0h	保留
19-16	VHV_PV	R	8h	内部。仅可通过 TI 提供的 API 使用。
15-8	VCG2P5	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
7-0	VINH	R	1h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.37 FLASH_V 寄存器（偏移 = 198h）[复位 = 00000000h]

图 11-59 展示了 FLASH_V，表 11-63 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-59. FLASH_V 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VSL_P								VWL_P								V_READ								TRIM0P8							
R-X								R-X								R-X								R-X							

表 11-63. FLASH_V 寄存器字段说明

位	字段	类型	复位	说明
31-24	VSL_P	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
23-16	VWL_P	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
15-8	V_READ	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
7-0	TRIM0P8	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.38 USER_ID 寄存器（偏移 = 294h）[复位 = 30000000h]

图 11-60 展示了 USER_ID，表 11-64 对其进行了介绍。

返回到汇总表。

用户标识。
读取此寄存器和 FCFG1:ICEPICK_DEVICE_ID 寄存器是唯一支持的器件识别方法。
在安全区中时，该寄存器的值将由引导固件写入 AON_PMCTL:JTAGUSERCODE。

图 11-60. USER_ID 寄存器

31	30	29	28	27	26	25	24
PG_REV				VER		PA	RESERVED
R-3h				R-X		R-X	R-0h
23	22	21	20	19	18	17	16
CC13	序列					封装	
R-X	R-X					R-X	
15	14	13	12	11	10	9	8
协议				RESERVED			
R-X				R-0h			
7	6	5	4	3	2	1	0
RESERVED							
R-0h							

表 11-64. USER_ID 寄存器字段说明

位	字段	类型	复位	说明
31-28	PG_REV	R	3h	用于区分器件版本的字段
27-26	VER	R	X	版本号。 0x0：此寄存器的位 [25:12] 具有前述含义。 任何其他设置均表示这些位采用不同的编码方式。 默认值因器件型号而异。
25	PA	R	X	0：不支持 20dBm PA 1：支持 20dBM PA 默认值因器件型号而异。
24	RESERVED	R	0h	保留
23	CC13	R	X	0：CC26xx 器件类型 1：CC13xx 器件类型 默认值因器件型号而异。
22-19	序列	R	X	序列。 用于区分营销型号/可订购产品中其他寄存器字段（如温度范围、闪存容量、电压范围等）均相同的不同产品 默认值因器件型号而异。
18-16	封装	R	X	封装类型。 0x0：4x4mm QFN (RHB) 封装 0x1：5x5mm QFN (RSM) 封装 0x2：7x7mm QFN (RGZ) 封装 0x3：晶圆销售封装（裸片） 0x4：WCSP (YFV) 0x5：具有可湿性侧面的 7x7mm QFN 封装 其他值保留供将来使用。 器件数据表中展示了适用于特定器件的封装。 默认值因器件型号而异。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 11-64. USER_ID 寄存器字段说明（续）

位	字段	类型	复位	说明
15-12	协议	R	X	支持的协议。 0x1：BLE 0x2：RF4CE 0x4：Zigbee/6lowpan 0x8：专有 同一器件可支持多个协议，此时应将上述取值组合使用。 默认值因器件型号而异。
11-0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.39 FLASH_OTP_DATA3 寄存器（偏移 = 2B0h）[复位 = 00110003h]

图 11-61 展示了 FLASH_OTP_DATA3，表 11-65 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-61. FLASH_OTP_DATA3 寄存器

31	30	29	28	27	26	25	24
EC_STEP_SIZE							
R-0h							
23	22	21	20	19	18	17	16
EC_STEP_SIZE	DO_PRECOND	MAX_EC_LEVEL				TRIM_1P7	
R-0h	R-0h	R-4h				R-1h	
15	14	13	12	11	10	9	8
FLASH_SIZE							
R-X							
7	6	5	4	3	2	1	0
WAIT_SYSCODE							
R-3h							

表 11-65. FLASH_OTP_DATA3 寄存器字段说明

位	字段	类型	复位	说明
31-23	EC_STEP_SIZE	R	0h	内部。仅可通过 TI 提供的 API 使用。
22	DO_PRECOND	R	0h	内部。仅可通过 TI 提供的 API 使用。
21-18	MAX_EC_LEVEL	R	4h	内部。仅可通过 TI 提供的 API 使用。
17-16	TRIM_1P7	R	1h	内部。仅可通过 TI 提供的 API 使用。
15-8	FLASH_SIZE	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值因器件型号而异。
7-0	WAIT_SYSCODE	R	3h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.40 ANA2_TRIM 寄存器（偏移 = 2B4h）[复位 = 8240787Fh]

图 11-62 展示了 ANA2_TRIM，表 11-66 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-62. ANA2_TRIM 寄存器

31	30	29	28	27	26	25	24
RCOSCHFCTRIMFRACT_EN	RCOSCHFCTRIMFRACT					RESERVED	SET_RCOSC_HF_FINE_RESISTOR
R-1h	R-X					R-0h	R-X
23	22	21	20	19	18	17	16
SET_RCOSC_HF_FINE_RESISTOR	ATESTLF_UDIGLDO_IBIAS_TRIM	NANOAMP_RES_TRIM					
R-X	R-1h	R-X					
15	14	13	12	11	10	9	8
NANOAMP_RES_TRIM	RESERVED			DITHER_EN	DCDC_IPEAK		
R-X	R-0h			R-1h	R-0h		
7	6	5	4	3	2	1	0
DEAD_TIME_TRIM		DCDC_LOW_EN_SEL			DCDC_HIGH_EN_SEL		
R-1h		R-7h			R-7h		

表 11-66. ANA2_TRIM 寄存器字段说明

位	字段	类型	复位	说明
31	RCOSCHFCTRIMFRACT_EN	R	1h	内部。仅可通过 TI 提供的 API 使用。
30-26	RCOSCHFCTRIMFRACT	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
25	RESERVED	R	0h	保留
24-23	SET_RCOSC_HF_FINE_RESISTOR	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
22	ATESTLF_UDIGLDO_IBIAS_TRIM	R	1h	内部。仅可通过 TI 提供的 API 使用。
21-15	NANOAMP_RES_TRIM	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
14-12	RESERVED	R	0h	保留
11	DITHER_EN	R	1h	内部。仅可通过 TI 提供的 API 使用。
10-8	DCDC_IPEAK	R	0h	内部。仅可通过 TI 提供的 API 使用。
7-6	DEAD_TIME_TRIM	R	1h	内部。仅可通过 TI 提供的 API 使用。
5-3	DCDC_LOW_EN_SEL	R	7h	内部。仅可通过 TI 提供的 API 使用。
2-0	DCDC_HIGH_EN_SEL	R	7h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.41 LDO_TRIM 寄存器（偏移 = 2B8h）[复位 = E0F8E0FBh]

图 11-63 展示了 LDO_TRIM，表 11-67 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-63. LDO_TRIM 寄存器

31	30	29	28	27	26	25	24
RESERVED			VDDR_TRIM_SLEEP				
R-0h			R-X				
23	22	21	20	19	18	17	16
RESERVED					GLDO_CURSRC		
R-0h					R-0h		
15	14	13	12	11	10	9	8
RESERVED			ITRIM_DIGLDO_LOAD		ITRIM_UDIGLDO		
R-0h			R-0h		R-0h		
7	6	5	4	3	2	1	0
RESERVED					VTRIM_DELTA		
R-0h					R-3h		

表 11-67. LDO_TRIM 寄存器字段说明

位	字段	类型	复位	说明
31-29	RESERVED	R	0h	保留
28-24	VDDR_TRIM_SLEEP	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
23-19	RESERVED	R	0h	保留
18-16	GLDO_CURSRC	R	0h	内部。仅可通过 TI 提供的 API 使用。
15-13	RESERVED	R	0h	保留
12-11	ITRIM_DIGLDO_LOAD	R	0h	内部。仅可通过 TI 提供的 API 使用。
10-8	ITRIM_UDIGLDO	R	0h	内部。仅可通过 TI 提供的 API 使用。
7-3	RESERVED	R	0h	保留
2-0	VTRIM_DELTA	R	3h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.42 MAC_BLE_0 寄存器（偏移 = 2E8h）[复位 = 00000000h]

图 11-64 展示了 MAC_BLE_0，表 11-68 对其进行了介绍。

返回到汇总表。

MAC BLE 地址 0

图 11-64. MAC_BLE_0 寄存器

31	30	29	28	27	26	25	24	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_0_31																															
R-X																															

表 11-68. MAC_BLE_0 寄存器字段说明

位	字段	类型	复位	说明
31-0	ADDR_0_31	R	X	64 位 MAC BLE 地址的前 32 位 默认值保存来自生产测试的修整值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.43 MAC_BLE_1 寄存器（偏移 = 2ECh）[复位 = 00000000h]

图 11-65 展示了 MAC_BLE_1，表 11-69 对其进行了介绍。

返回到汇总表。

MAC BLE 地址 1

图 11-65. MAC_BLE_1 寄存器

31	30	29	28	27	26	25	24	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_32_63																															
R-X																															

表 11-69. MAC_BLE_1 寄存器字段说明

位	字段	类型	复位	说明
31-0	ADDR_32_63	R	X	64 位 MAC BLE 地址的最后 32 位 默认值保存来自生产测试的修整值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.44 MAC_15_4_0 寄存器（偏移 = 2F0h）[复位 = 00000000h]

图 11-66 展示了 MAC_15_4_0，表 11-70 对其进行了介绍。

返回到汇总表。

MAC IEEE 802.15.4 地址 0

图 11-66. MAC_15_4_0 寄存器

31	30	29	28	27	26	25	24	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_0_31																															
R-X																															

表 11-70. MAC_15_4_0 寄存器字段说明

位	字段	类型	复位	说明
31-0	ADDR_0_31	R	X	64 位 MAC 15.4 地址的前 32 位 默认值保存来自生产测试的修整值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.45 MAC_15_4_1 寄存器（偏移 = 2F4h）[复位 = 00000000h]

图 11-67 展示了 MAC_15_4_1，表 11-71 对其进行了介绍。

返回到汇总表。

MAC IEEE 802.15.4 地址 1

图 11-67. MAC_15_4_1 寄存器

31	30	29	28	27	26	25	24	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_32_63																															
R-X																															

表 11-71. MAC_15_4_1 寄存器字段说明

位	字段	类型	复位	说明
31-0	ADDR_32_63	R	X	64 位 MAC 15.4 地址的最后 32 位 默认值保存来自生产测试的修整值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.46 FLASH_OTP_DATA4 寄存器（偏移 = 308h）[复位 = 98989F9Fh]

图 11-68 展示了 FLASH_OTP_DATA4，表 11-72 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-68. FLASH_OTP_DATA4 寄存器

31	30	29	28	27	26	25	24
STANDBY_MODE_SEL_INT_WRT	STANDBY_PW_SEL_INT_WRT		DIS_STANDBY_INT_WRT	DIS_IDLE_INT_WRT	VIN_AT_X_INT_WRT		
R-1h	R-0h		R-1h	R-1h	R-0h		
23	22	21	20	19	18	17	16
STANDBY_MODE_SEL_EXT_WRT	STANDBY_PW_SEL_EXT_WRT		DIS_STANDBY_EXT_WRT	DIS_IDLE_EXT_WRT	VIN_AT_X_EXT_WRT		
R-1h	R-0h		R-1h	R-1h	R-0h		
15	14	13	12	11	10	9	8
STANDBY_MODE_SEL_INT_RD	STANDBY_PW_SEL_INT_RD		DIS_STANDBY_INT_RD	DIS_IDLE_INT_RD	VIN_AT_X_INT_RD		
R-1h	R-0h		R-1h	R-1h	R-7h		
7	6	5	4	3	2	1	0
STANDBY_MODE_SEL_EXT_RD	STANDBY_PW_SEL_EXT_RD		DIS_STANDBY_EXT_RD	DIS_IDLE_EXT_RD	VIN_AT_X_EXT_RD		
R-1h	R-0h		R-1h	R-1h	R-7h		

表 11-72. FLASH_OTP_DATA4 寄存器字段说明

位	字段	类型	复位	说明
31	STANDBY_MODE_SEL_INT_WRT	R	1h	内部。仅可通过 TI 提供的 API 使用。
30-29	STANDBY_PW_SEL_INT_WRT	R	0h	内部。仅可通过 TI 提供的 API 使用。
28	DIS_STANDBY_INT_WRT	R	1h	内部。仅可通过 TI 提供的 API 使用。
27	DIS_IDLE_INT_WRT	R	1h	内部。仅可通过 TI 提供的 API 使用。
26-24	VIN_AT_X_INT_WRT	R	0h	内部。仅可通过 TI 提供的 API 使用。
23	STANDBY_MODE_SEL_EXT_WRT	R	1h	内部。仅可通过 TI 提供的 API 使用。
22-21	STANDBY_PW_SEL_EXT_WRT	R	0h	内部。仅可通过 TI 提供的 API 使用。
20	DIS_STANDBY_EXT_WRT	R	1h	内部。仅可通过 TI 提供的 API 使用。
19	DIS_IDLE_EXT_WRT	R	1h	内部。仅可通过 TI 提供的 API 使用。
18-16	VIN_AT_X_EXT_WRT	R	0h	内部。仅可通过 TI 提供的 API 使用。
15	STANDBY_MODE_SEL_INT_RD	R	1h	内部。仅可通过 TI 提供的 API 使用。
14-13	STANDBY_PW_SEL_INT_RD	R	0h	内部。仅可通过 TI 提供的 API 使用。
12	DIS_STANDBY_INT_RD	R	1h	内部。仅可通过 TI 提供的 API 使用。
11	DIS_IDLE_INT_RD	R	1h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 11-72. FLASH_OTP_DATA4 寄存器字段说明（续）

位	字段	类型	复位	说明
10-8	VIN_AT_X_INT_RD	R	7h	内部。仅可通过 TI 提供的 API 使用。
7	STANDBY_MODE_SEL_EXT_RD	R	1h	内部。仅可通过 TI 提供的 API 使用。
6-5	STANDBY_PW_SEL_EXT_RD	R	0h	内部。仅可通过 TI 提供的 API 使用。
4	DIS_STANDBY_EXT_RD	R	1h	内部。仅可通过 TI 提供的 API 使用。
3	DIS_IDLE_EXT_RD	R	1h	内部。仅可通过 TI 提供的 API 使用。
2-0	VIN_AT_X_EXT_RD	R	7h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.47 MISC_TRIM 寄存器（偏移 = 30Ch）[复位 = FFFE003Bh]

图 11-69 展示了 MISC_TRIM，表 11-73 对其进行了介绍。

返回到汇总表。

其他修整参数

图 11-69. MISC_TRIM 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							TRIM_RECHARGE_COMP_OFFSET
R-0h							R-X
15	14	13	12	11	10	9	8
TRIM_RECHARGE_COMP_OFFSET				TRIM_RECHARGE_COMP_REFLEVEL			
R-X				R-X			
7	6	5	4	3	2	1	0
TEMPVSLOPE							
R-3Bh							

表 11-73. MISC_TRIM 寄存器字段说明

位	字段	类型	复位	说明
31-17	RESERVED	R	0h	保留
16-12	TRIM_RECHARGE_COMP_OFFSET	R	X	内部。仅可通过 TI 提供的 API 使用。
11-8	TRIM_RECHARGE_COMP_REFLEVEL	R	X	内部。仅可通过 TI 提供的 API 使用。
7-0	TEMPVSLOPE	R	3Bh	有符号字节值，表示电池电压的 TEMP 斜率，单位为 °C/V，并包含四位小数。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.48 RCOSC_HF_TEMPCOMP 寄存器（偏移 = 310h）[复位 = 00000003h]

图 11-70 展示了 RCOSC_HF_TEMPCOMP，表 11-74 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-70. RCOSC_HF_TEMPCOMP 寄存器

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

表 11-74. RCOSC_HF_TEMPCOMP 寄存器字段说明

位	字段	类型	复位	说明
31-24	FINE_RESISTOR	R	0h	内部。仅可通过 TI 提供的 API 使用。
23-16	CTRIM	R	0h	内部。仅可通过 TI 提供的 API 使用。
15-8	CTRIMFRACT_QUAD	R	0h	内部。仅可通过 TI 提供的 API 使用。
7-0	CTRIMFRACT_SLOPE	R	3h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.49 ICEPICK_DEVICE_ID 寄存器（偏移 = 318h）[复位 = 3BB4102Fh]

图 11-71 展示了 ICEPICK_DEVICE_ID，表 11-75 对其进行了介绍。

返回到汇总表。

IcePick 器件标识
读取此寄存器和 FCFG1:USER_ID 寄存器是唯一支持的器件识别方法。

图 11-71. ICEPICK_DEVICE_ID 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PG_REV				WAFER_ID											
R-3h				R-BB41h											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WAFER_ID				MANUFACTURER_ID											
R-BB41h				R-2Fh											

表 11-75. ICEPICK_DEVICE_ID 寄存器字段说明

位	字段	类型	复位	说明
31-28	PG_REV	R	3h	用于区分器件版本的字段。
27-12	WAFER_ID	R	BB41h	用于标识硅片的字段。
11-0	MANUFACTURER_ID	R	2Fh	制造商代码。 0x02F：德州仪器 (TI)

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.50 FCFG1_REVISION 寄存器（偏移 = 31Ch）[复位 = 0000002Ah]

图 11-72 展示了 FCFG1_REVISION，表 11-76 对其进行了介绍。

返回到汇总表。

出厂配置 (FCFG1) 版本

图 11-72. FCFG1_REVISION 寄存器

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

表 11-76. FCFG1_REVISION 寄存器字段说明

位	字段	类型	复位	说明
31-0	版本	R	2Ah	FCFG1 布局的版本号。应用软件将读取此值，以确定哪些 FCFG1 参数具有有效值。如果 FCFG1 布局自上一批器件生产以来发生变化，则在生产任何器件之前，该版本号必须加 1。 值可能在没有警告的情况下发生变化。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.51 MISC_OTP_DATA 寄存器（偏移 = 320h）[复位 = 0000CFFFh]

图 11-73 展示了 MISC_OTP_DATA，表 11-77 对其进行了介绍。

返回到汇总表。

其他 OTP 数据

图 11-73. MISC_OTP_DATA 寄存器

31	30	29	28	27	26	25	24
RCOSC_HF_ITUNE				RCOSC_HF_CRIM			
R-X				R-X			
23	22	21	20	19	18	17	16
RCOSC_HF_CRIM				PER_M			
R-X				R-1h			
15	14	13	12	11	10	9	8
PER_M	PER_E			RESERVED			
R-1h	R-4h			R-0h			
7	6	5	4	3	2	1	0
RESERVED							
R-0h							

表 11-77. MISC_OTP_DATA 寄存器字段说明

位	字段	类型	复位	说明
31-28	RCOSC_HF_ITUNE	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
27-20	RCOSC_HF_CRIM	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
19-15	PER_M	R	1h	内部。仅可通过 TI 提供的 API 使用。
14-12	PER_E	R	4h	内部。仅可通过 TI 提供的 API 使用。
11-0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.52 IOCONF 寄存器（偏移 = 344h）[复位 = FFFFFF00h]

图 11-74 展示了 IOCONF，表 11-78 中对此进行了介绍。

返回到汇总表。

IO 配置

图 11-74. IOCONF 寄存器

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									GPIO_CNT						
R-0h									R-X						

表 11-78. IOCONF 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	GPIO_CNT	R	X	可用 DIO 的数量。 默认值因器件型号而异。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.53 CONFIG_IF_ADC 寄存器（偏移 = 34Ch）[复位 = 3460F400h]

图 11-75 展示了 CONFIG_IF_ADC，表 11-79 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-75. CONFIG_IF_ADC 寄存器

31	30	29	28	27	26	25	24
FF2ADJ				FF3ADJ			
R-3h				R-4h			
23	22	21	20	19	18	17	16
INT3ADJ				FF1ADJ			
R-6h				R-0h			
15	14	13	12	11	10	9	8
AAFCAP		INT2ADJ				IFDIGLDO_TRIM_OUTPUT	
R-3h		R-Dh				R-X	
7	6	5	4	3	2	1	0
IFDIGLDO_TRIM_OUTPUT			IFANALDO_TRIM_OUTPUT				
R-X			R-X				

表 11-79. CONFIG_IF_ADC 寄存器字段说明

位	字段	类型	复位	说明
31-28	FF2ADJ	R	3h	内部。仅可通过 TI 提供的 API 使用。
27-24	FF3ADJ	R	4h	内部。仅可通过 TI 提供的 API 使用。
23-20	INT3ADJ	R	6h	内部。仅可通过 TI 提供的 API 使用。
19-16	FF1ADJ	R	0h	内部。仅可通过 TI 提供的 API 使用。
15-14	AAFCAP	R	3h	内部。仅可通过 TI 提供的 API 使用。
13-10	INT2ADJ	R	Dh	内部。仅可通过 TI 提供的 API 使用。
9-5	IFDIGLDO_TRIM_OUTPUT	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
4-0	IFANALDO_TRIM_OUTPUT	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.54 CONFIG_OSC_TOP 寄存器（偏移 = 350h）[复位 = DC07FC00h]

图 11-76 展示了 CONFIG_OSC_TOP，表 11-80 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-76. CONFIG_OSC_TOP 寄存器

31	30	29	28	27	26	25	24
RESERVED		XOSC_HF_ROW_Q12				XOSC_HF_COLUMN_Q12	
R-0h		R-7h				R-1FFh	
23	22	21	20	19	18	17	16
XOSC_HF_COLUMN_Q12							
R-1FFh							
15	14	13	12	11	10	9	8
XOSC_HF_COLUMN_Q12						RCOSCLF_CTUNE_TRIM	
R-1FFh						R-X	
7	6	5	4	3	2	1	0
RCOSCLF_CTUNE_TRIM						RCOSCLF_RTUNE_TRIM	
R-X						R-0h	

表 11-80. CONFIG_OSC_TOP 寄存器字段说明

位	字段	类型	复位	说明
31-30	RESERVED	R	0h	保留
29-26	XOSC_HF_ROW_Q12	R	7h	内部。仅可通过 TI 提供的 API 使用。
25-10	XOSC_HF_COLUMN_Q12	R	1FFh	内部。仅可通过 TI 提供的 API 使用。
9-2	RCOSCLF_CTUNE_TRIM	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
1-0	RCOSCLF_RTUNE_TRIM	R	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.55 SOC_ADC_ABS_GAIN 寄存器（偏移 = 35Ch）[复位 = 00000000h]

图 11-77 展示了 SOC_ADC_ABS_GAIN，表 11-81 对其进行了介绍。

返回到汇总表。

绝对基准模式下的 AUX_ADC 增益

图 11-77. SOC_ADC_ABS_GAIN 寄存器

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
SOC_ADC_ABS_GAIN_TEMP1															
R-X															

表 11-81. SOC_ADC_ABS_GAIN 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	SOC_ADC_ABS_GAIN_TEMP1	R	X	温度为 1 (30C) 时，绝对基准模式下的 SOC_ADC 增益。在生产测试中计算得出。 默认值保存来自生产测试的日志信息。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.56 SOC_ADC_REL_GAIN 寄存器（偏移 = 360h）[复位 = 00000000h]

图 11-78 展示了 SOC_ADC_REL_GAIN，表 11-82 对其进行了介绍。

返回到汇总表。

相对基准模式下的 AUX_ADC 增益

图 11-78. SOC_ADC_REL_GAIN 寄存器

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
SOC_ADC_REL_GAIN_TEMP1															
R-X															

表 11-82. SOC_ADC_REL_GAIN 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	SOC_ADC_REL_GAIN_TEMP1	R	X	温度为 1 (30C) 时，相对基准模式下的 SOC_ADC 增益。在生产测试中计算得出。 默认值保存来自生产测试的修整值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.57 SOC_ADC_OFFSET_INT 寄存器（偏移 = 368h）[复位 = 00000000h]

图 11-79 展示了 SOC_ADC_OFFSET_INT，表 11-83 对其进行了介绍。

返回到汇总表。

绝对基准模式下的 AUX_ADC 温度偏移

图 11-79. SOC_ADC_OFFSET_INT 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED								SOC_ADC_REL_OFFSET_TEMP1							
R-0h								R-X							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								SOC_ADC_ABS_OFFSET_TEMP1							
R-0h								R-X							

表 11-83. SOC_ADC_OFFSET_INT 寄存器字段说明

位	字段	类型	复位	说明
31-24	RESERVED	R	0h	保留
23-16	SOC_ADC_REL_OFFSET_TEMP1	R	X	温度为 1 (30C) 时，相对基准模式下的 SOC_ADC 偏移。有符号 8 位数字。在生产测试中计算得出。 默认值保存来自生产测试的修整值。
15-8	RESERVED	R	0h	保留
7-0	SOC_ADC_ABS_OFFSET_TEMP1	R	X	温度为 1 (30C) 时，绝对基准模式下的 SOC_ADC 偏移。有符号 8 位数字。在生产测试中计算得出。 默认值保存来自生产测试的修整值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.58 SOC_ADC_REF_TRIM_AND_OFFSET_EXT 寄存器（偏移 = 36Ch）[复位 = 0000C080h]

图 11-80 展示了 SOC_ADC_REF_TRIM_AND_OFFSET_EXT，表 11-84 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-80. SOC_ADC_REF_TRIM_AND_OFFSET_EXT 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED		SOC_ADC_REF_VOLTAGE_TRIM_TEMP1					
R-0h		R-X					

表 11-84. SOC_ADC_REF_TRIM_AND_OFFSET_EXT 寄存器字段说明

位	字段	类型	复位	说明
31-6	RESERVED	R	0h	保留
5-0	SOC_ADC_REF_VOLTAGE_TRIM_TEMP1	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.59 AMPCOMP_TH1 寄存器（偏移 = 370h）[复位 = FF7B828Eh]

图 11-81 展示了 AMPCOMP_TH1，表 11-85 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-81. AMPCOMP_TH1 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
HPMRAMP3_LTH						RESERVED	
R-1Eh						R-0h	
15	14	13	12	11	10	9	8
HPMRAMP3_HTH						IBIASCAP_LPTOHP_OL_CNT	
R-20h						R-Ah	
7	6	5	4	3	2	1	0
IBIASCAP_LPTOHP_OL_CNT		HPMRAMP1_TH					
R-Ah		R-Eh					

表 11-85. AMPCOMP_TH1 寄存器字段说明

位	字段	类型	复位	说明
31-24	RESERVED	R	0h	保留
23-18	HPMRAMP3_LTH	R	1Eh	内部。仅可通过 TI 提供的 API 使用。
17-16	RESERVED	R	0h	保留
15-10	HPMRAMP3_HTH	R	20h	内部。仅可通过 TI 提供的 API 使用。
9-6	IBIASCAP_LPTOHP_OL_CNT	R	Ah	内部。仅可通过 TI 提供的 API 使用。
5-0	HPMRAMP1_TH	R	Eh	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.60 AMPCOMP_TH2 寄存器（偏移 = 374h）[复位 = 6B8B0303h]

图 11-82 展示了 AMPCOMP_TH2，表 11-86 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-82. AMPCOMP_TH2 寄存器

31	30	29	28	27	26	25	24
LPMUPDATE_LTH						RESERVED	
R-1Ah						R-0h	
23	22	21	20	19	18	17	16
LPMUPDATE_HTM						RESERVED	
R-22h						R-0h	
15	14	13	12	11	10	9	8
ADC_COMP_AMPTH_LPM						RESERVED	
R-0h						R-0h	
7	6	5	4	3	2	1	0
ADC_COMP_AMPTH_HPM						RESERVED	
R-0h						R-0h	

表 11-86. AMPCOMP_TH2 寄存器字段说明

位	字段	类型	复位	说明
31-26	LPMUPDATE_LTH	R	1Ah	内部。仅可通过 TI 提供的 API 使用。
25-24	RESERVED	R	0h	保留
23-18	LPMUPDATE_HTM	R	22h	内部。仅可通过 TI 提供的 API 使用。
17-16	RESERVED	R	0h	保留
15-10	ADC_COMP_AMPTH_LPM	R	0h	内部。仅可通过 TI 提供的 API 使用。
9-8	RESERVED	R	0h	保留
7-2	ADC_COMP_AMPTH_HPM	R	0h	内部。仅可通过 TI 提供的 API 使用。
1-0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.61 AMPCOMP_CTRL1 寄存器（偏移 = 378h）[复位 = FF483F47h]

图 11-83 展示了 AMPCOMP_CTRL1，表 11-87 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-83. AMPCOMP_CTRL1 寄存器

31	30	29	28	27	26	25	24
RESERVED	AMPCOMP_REQ_MODE	RESERVED					
R-0h	R-1h	R-0h					
23	22	21	20	19	18	17	16
IBIAS_OFFSET				IBIAS_INIT			
R-4h				R-8h			
15	14	13	12	11	10	9	8
LPM_IBIAS_WAIT_CNT_FINAL							
R-3Fh							
7	6	5	4	3	2	1	0
CAP_STEP				IBIASCAP_HPTOLP_OL_CNT			
R-4h				R-7h			

表 11-87. AMPCOMP_CTRL1 寄存器字段说明

位	字段	类型	复位	说明
31	RESERVED	R	0h	保留
30	AMPCOMP_REQ_MODE	R	1h	内部。仅可通过 TI 提供的 API 使用。
29-24	RESERVED	R	0h	保留
23-20	IBIAS_OFFSET	R	4h	内部。仅可通过 TI 提供的 API 使用。
19-16	IBIAS_INIT	R	8h	内部。仅可通过 TI 提供的 API 使用。
15-8	LPM_IBIAS_WAIT_CNT_FINAL	R	3Fh	内部。仅可通过 TI 提供的 API 使用。
7-4	CAP_STEP	R	4h	内部。仅可通过 TI 提供的 API 使用。
3-0	IBIASCAP_HPTOLP_OL_CNT	R	7h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.62 ANABYPASS_VALUE2 寄存器（偏移 = 37Ch）[复位 = FFFFC3FFh]

图 11-84 展示了 ANABYPASS_VALUE2，表 11-88 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-84. ANABYPASS_VALUE2 寄存器

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		XOSC_HF_IBIASTHERM													
R-0h		R-3FFh													

表 11-88. ANABYPASS_VALUE2 寄存器字段说明

位	字段	类型	复位	说明
31-14	RESERVED	R	0h	保留
13-0	XOSC_HF_IBIASTHERM	R	3FFh	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.63 VOLT_TRIM 寄存器（偏移 = 388h）[复位 = E0E0E0E0h]

图 11-85 展示了 VOLT_TRIM，表 11-89 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-85. VOLT_TRIM 寄存器

31	30	29	28	27	26	25	24
RESERVED			VDDR_TRIM_HH				
R-0h			R-X				
23	22	21	20	19	18	17	16
RESERVED			VDDR_TRIM_H				
R-0h			R-X				
15	14	13	12	11	10	9	8
RESERVED			VDDR_TRIM_SLEEP_H				
R-0h			R-X				
7	6	5	4	3	2	1	0
RESERVED			TRIMBOD_H				
R-0h			R-X				

表 11-89. VOLT_TRIM 寄存器字段说明

位	字段	类型	复位	说明
31-29	RESERVED	R	0h	保留
28-24	VDDR_TRIM_HH	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
23-21	RESERVED	R	0h	保留
20-16	VDDR_TRIM_H	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
15-13	RESERVED	R	0h	保留
12-8	VDDR_TRIM_SLEEP_H	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
7-5	RESERVED	R	0h	保留
4-0	TRIMBOD_H	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.64 OSC_CONF 寄存器（偏移 = 38Ch）[复位 = F00900E6h]

图 11-86 展示了 OSC_CONF，表 11-90 对其进行了介绍。

返回到汇总表。

OSC 配置

图 11-86. OSC_CONF 寄存器

31	30	29	28	27	26	25	24
RESERVED		ADC_SH_VBUF_EN	ADC_SH_MODE_EN	ATESTLF_RCOSCLF_IBIAS_TRIM	XOSCLF_REGULATOR_TRIM		XOSCLF_CMIRRWR_RATIO
R-0h		R-1h	R-1h	R-0h	R-0h		R-0h
23	22	21	20	19	18	17	16
XOSCLF_CMIRRWR_RATIO			XOSC_HF_FAST_START		XOSC_OPTION	HPOSC_OPTION	HPOSC_BIAS_HOLD_MODE_EN
R-0h			R-1h		R-X	R-X	R-1h
15	14	13	12	11	10	9	8
HPOSC_CURRMIRR_RATIO				HPOSC_BIAS_RES_SET			
R-X				R-X			
7	6	5	4	3	2	1	0
HPOSC_FILTER_EN	HPOSC_BIAS_RECHARGE_DELAY		RESERVED		HPOSC_SERIES_CAP		HPOSC_DIV3_BYPASS
R-1h	R-3h		R-0h		R-3h		R-0h

表 11-90. OSC_CONF 寄存器字段说明

位	字段	类型	复位	说明
31-30	RESERVED	R	0h	保留
29	ADC_SH_VBUF_EN	R	1h	DDI_0_OSC:ADCDOUBLERNANOAMPCTL.ADC_SH_VBUF_EN 的修整值。
28	ADC_SH_MODE_EN	R	1h	DDI_0_OSC:ADCDOUBLERNANOAMPCTL.ADC_SH_MODE_EN 的修整值。
27	ATESTLF_RCOSCLF_IBIAS_TRIM	R	0h	DDI_0_OSC:ATESTCTL.ATESTLF_RCOSCLF_IBIAS_TRIM 的修整值。
26-25	XOSCLF_REGULATOR_TRIM	R	0h	DDI_0_OSC:LFOSCCTL.XOSCLF_REGULATOR_TRIM 的修整值。
24-21	XOSCLF_CMIRRWR_RATIO	R	0h	DDI_0_OSC:LFOSCCTL.XOSCLF_CMIRRWR_RATIO 的修整值。
20-19	XOSC_HF_FAST_START	R	1h	DDI_0_OSC:CTL1.XOSC_HF_FAST_START 的修整值。
18	XOSC_OPTION	R	X	0：XOSC_HF 不可用（可能无法进行外部键合） 1：XOSC_HF 可用（默认） 默认值因器件型号而异。
17	HPOSC_OPTION	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值因器件型号而异。
16	HPOSC_BIAS_HOLD_MODE_EN	R	1h	内部。仅可通过 TI 提供的 API 使用。
15-12	HPOSC_CURRMIRR_RATIO	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
11-8	HPOSC_BIAS_RES_SET	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
7	HPOSC_FILTER_EN	R	1h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 11-90. OSC_CONF 寄存器字段说明（续）

位	字段	类型	复位	说明
6-5	HPOSC_BIAS_RECHARGE_DELAY	R	3h	内部。仅可通过 TI 提供的 API 使用。
4-3	RESERVED	R	0h	保留
2-1	HPOSC_SERIES_CAP	R	3h	内部。仅可通过 TI 提供的 API 使用。
0	HPOSC_DIV3_BYPASS	R	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.65 FREQ_OFFSET 寄存器（偏移 = 390h）[复位 = 00000000h]

图 11-87 展示了 FREQ_OFFSET，表 11-91 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-87. FREQ_OFFSET 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HPOSC_COMP_P0															
R-X															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HPOSC_COMP_P1								HPOSC_COMP_P2							
R-X								R-X							

表 11-91. FREQ_OFFSET 寄存器字段说明

位	字段	类型	复位	说明
31-16	HPOSC_COMP_P0	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
15-8	HPOSC_COMP_P1	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
7-0	HPOSC_COMP_P2	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.66 MISC_OTP_DATA_1 寄存器（偏移 = 398h）[复位 = E08403F8h]

图 11-88 展示了 MISC_OTP_DATA_1，表 11-92 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-88. MISC_OTP_DATA_1 寄存器

31	30	29	28	27	26	25	24
RESERVED			PEAK_DET_ITRIM		HP_BUF_ITRIM		
R-0h			R-0h		R-0h		
23	22	21	20	19	18	17	16
LP_BUF_ITRIM		DBLR_LOOP_FILTER_RESET_VOLTAGE		HPM_IBIAS_WAIT_CNT			
R-2h		R-0h		R-100h			
15	14	13	12	11	10	9	8
HPM_IBIAS_WAIT_CNT						LPM_IBIAS_WAIT_CNT	
R-100h						R-3Fh	
7	6	5	4	3	2	1	0
LPM_IBIAS_WAIT_CNT				IDAC_STEP			
R-3Fh				R-8h			

表 11-92. MISC_OTP_DATA_1 寄存器字段说明

位	字段	类型	复位	说明
31-29	RESERVED	R	0h	保留
28-27	PEAK_DET_ITRIM	R	0h	内部。仅可通过 TI 提供的 API 使用。
26-24	HP_BUF_ITRIM	R	0h	内部。仅可通过 TI 提供的 API 使用。
23-22	LP_BUF_ITRIM	R	2h	内部。仅可通过 TI 提供的 API 使用。
21-20	DBLR_LOOP_FILTER_RESET_VOLTAGE	R	0h	内部。仅可通过 TI 提供的 API 使用。
19-10	HPM_IBIAS_WAIT_CNT	R	100h	内部。仅可通过 TI 提供的 API 使用。
9-4	LPM_IBIAS_WAIT_CNT	R	3Fh	内部。仅可通过 TI 提供的 API 使用。
3-0	IDAC_STEP	R	8h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.67 SHDW_DIE_ID_0 寄存器（偏移 = 3D0h）[复位 = 00000000h]

图 11-89 展示了 SHDW_DIE_ID_0，表 11-93 对其进行了介绍。

返回到汇总表。

eFuse 中 DIE_ID_0 寄存器的影子寄存器

图 11-89. SHDW_DIE_ID_0 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID_31_0																															
R-X																															

表 11-93. SHDW_DIE_ID_0 寄存器字段说明

位	字段	类型	复位	说明
31-0	ID_31_0	R	X	eFuse 行编号 5 中 DIE_ID_0 寄存器的影子寄存器 默认值取决于 eFuse 值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.68 SHDW_DIE_ID_1 寄存器（偏移 = 3D4h）[复位 = 00000000h]

图 11-90 展示了 SHDW_DIE_ID_1，表 11-94 对其进行了介绍。

返回到汇总表。

eFuse 中 DIE_ID_1 寄存器的影子寄存器

图 11-90. SHDW_DIE_ID_1 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID_63_32																															
R-X																															

表 11-94. SHDW_DIE_ID_1 寄存器字段说明

位	字段	类型	复位	说明
31-0	ID_63_32	R	X	eFuse 行编号 6 中 DIE_ID_1 寄存器的影子寄存器 默认值取决于 eFuse 值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.69 SHDW_DIE_ID_2 寄存器（偏移 = 3D8h）[复位 = 00000000h]

图 11-91 展示了 SHDW_DIE_ID_2，表 11-95 对其进行了介绍。

返回到汇总表。

eFuse 中 DIE_ID_2 寄存器的影子寄存器

图 11-91. SHDW_DIE_ID_2 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID_95_64																															
R-X																															

表 11-95. SHDW_DIE_ID_2 寄存器字段说明

位	字段	类型	复位	说明
31-0	ID_95_64	R	X	eFuse 行编号 7 中 DIE_ID_2 寄存器的影子寄存器 默认值取决于 eFuse 值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.70 SHDW_DIE_ID_3 寄存器（偏移 = 3DCh）[复位 = 00000000h]

图 11-92 展示了 SHDW_DIE_ID_3，表 11-96 对其进行了介绍。

返回到汇总表。

eFuse 中 DIE_ID_3 寄存器的影子寄存器

图 11-92. SHDW_DIE_ID_3 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID_127_96																															
R-X																															

表 11-96. SHDW_DIE_ID_3 寄存器字段说明

位	字段	类型	复位	说明
31-0	ID_127_96	R	X	eFuse 行编号 8 中 DIE_ID_3 寄存器的影子寄存器 默认值取决于 eFuse 值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.71 SHDW_OSC_BIAS_LDO_TRIM 寄存器（偏移 = 3F8h）[复位 = 00000000h]

图 11-93 展示了 SHDW_OSC_BIAS_LDO_TRIM，表 11-97 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-93. SHDW_OSC_BIAS_LDO_TRIM 寄存器

31	30	29	28	27	26	25	24
RESERVED						TRIMMAG	
R-0h						R-X	
23	22	21	20	19	18	17	16
TRIMMAG	TRIMIREF					ITRIM_DIG_LDO	
R-X	R-X					R-X	
15	14	13	12	11	10	9	8
VTRIM_DIG				VTRIM_COARSE			
R-X				R-X			
7	6	5	4	3	2	1	0
RCOSCHF_CTRIM							
R-X							

表 11-97. SHDW_OSC_BIAS_LDO_TRIM 寄存器字段说明

位	字段	类型	复位	说明
31-27	RESERVED	R	0h	保留
26-23	TRIMMAG	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值取决于 eFuse 值。
22-18	TRIMIREF	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值取决于 eFuse 值。
17-16	ITRIM_DIG_LDO	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值取决于 eFuse 值。
15-12	VTRIM_DIG	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值取决于 eFuse 值。
11-8	VTRIM_COARSE	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值取决于 eFuse 值。
7-0	RCOSCHF_CTRIM	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值取决于 eFuse 值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.72 SHDW_ANA_TRIM 寄存器（偏移 = 3FCh）[复位 = 00000000h]

图 11-94 展示了 SHDW_ANA_TRIM，表 11-98 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-94. SHDW_ANA_TRIM 寄存器

31	30	29	28	27	26	25	24
RESERVED	ALT_VDDR_TRIM	DET_LOGIC_DIS	BOD_BANDGAP_TRIM_CNF_EXT		BOD_BANDGAP_TRIM_CNF		VDDR_ENABLE_PG1
R-0h	R-X	R-X	R-X		R-X		R-X
23	22	21	20	19	18	17	16
VDDR_OK_HYS	IPTAT_TRIM		VDDR_TRIM				
R-X	R-X		R-X				
15	14	13	12	11	10	9	8
TRIMBOD_INTMODE					TRIMBOD_EXTMODE		
R-X					R-X		
7	6	5	4	3	2	1	0
TRIMBOD_EXTMODE		TRIMTEMP					
R-X		R-X					

表 11-98. SHDW_ANA_TRIM 寄存器字段说明

位	字段	类型	复位	说明
31	RESERVED	R	0h	保留
30	ALT_VDDR_TRIM	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值取决于 eFuse 值。
29	DET_LOGIC_DIS	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值取决于 eFuse 值。
28-27	BOD_BANDGAP_TRIM_CNF_EXT	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值取决于 eFuse 值。
26-25	BOD_BANDGAP_TRIM_CNF	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值取决于 eFuse 值。
24	VDDR_ENABLE_PG1	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值取决于 eFuse 值。
23	VDDR_OK_HYS	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值取决于 eFuse 值。
22-21	IPTAT_TRIM	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值取决于 eFuse 值。
20-16	VDDR_TRIM	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值取决于 eFuse 值。
15-11	TRIMBOD_INTMODE	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值取决于 eFuse 值。
10-6	TRIMBOD_EXTMODE	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值取决于 eFuse 值。
5-0	TRIMTEMP	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值取决于 eFuse 值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.73 OSC_CONF1 寄存器（偏移 = 408h）[复位 = FFFF0000h]

图 11-95 展示了 OSC_CONF1，表 11-99 对其进行了介绍。

返回到汇总表。

振荡器配置

图 11-95. OSC_CONF1 寄存器

31	30	29	28	27	26	25	24
RCOSC_MF_BIAS_HTEMP				RCOSC_MF_TEMP_DEPEND_MODE	RCOSC_MF_SINGLE_TRIM_METHOD	RESERVED	
R-X				R-X	R-X	R-0h	
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				RCOSC_MF_BIAS_ADJ			
R-0h				R-X			

表 11-99. OSC_CONF1 寄存器字段说明

位	字段	类型	复位	说明
31-28	RCOSC_MF_BIAS_HTEMP	R	X	内部。仅可通过 TI 提供的 API 使用。
27	RCOSC_MF_TEMP_DEPEND_MODE	R	X	内部。仅可通过 TI 提供的 API 使用。
26	RCOSC_MF_SINGLE_TRIM_METHOD	R	X	内部。仅可通过 TI 提供的 API 使用。
25-4	RESERVED	R	0h	保留
3-0	RCOSC_MF_BIAS_ADJ	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.74 DAC_BIAS_CNF 寄存器（偏移 = 40Ch）[复位 = FFFC00FFh]

图 11-96 展示了 DAC_BIAS_CNF，表 11-100 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-96. DAC_BIAS_CNF 寄存器

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

表 11-100. DAC_BIAS_CNF 寄存器字段说明

位	字段	类型	复位	说明
31-18	RESERVED	R	0h	保留
17-12	LPM_TRIM_IOUT	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
11-9	LPM_BIAS_WIDTH_TRIM	R	0h	内部。仅可通过 TI 提供的 API 使用。
8	LPM_BIAS_BACKUP_EN	R	0h	内部。仅可通过 TI 提供的 API 使用。
7-0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.75 TFW_PROBE 寄存器（偏移 = 418h）[复位 = 00000000h]

图 11-97 展示了 TFW_PROBE，表 11-101 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-97. TFW_PROBE 寄存器

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

表 11-101. TFW_PROBE 寄存器字段说明

位	字段	类型	复位	说明
31-0	版本	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.76 TFW_FT 寄存器（偏移 = 41Ch）[复位 = 00000000h]

图 11-98 展示了 TFW_FT，表 11-102 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-98. TFW_FT 寄存器

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

表 11-102. TFW_FT 寄存器字段说明

位	字段	类型	复位	说明
31-0	版本	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.77 DAC_CAL0 寄存器（偏移 = 420h）[复位 = 00000000h]

图 11-99 展示了 DAC_CAL0，表 11-103 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-99. DAC_CAL0 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SOC_DAC_VOUT_CAL_DECOUPLE_C2															
R-X															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOC_DAC_VOUT_CAL_DECOUPLE_C1															
R-X															

表 11-103. DAC_CAL0 寄存器字段说明

位	字段	类型	复位	说明
31-16	SOC_DAC_VOUT_CAL_DECOUPLE_C2	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
15-0	SOC_DAC_VOUT_CAL_DECOUPLE_C1	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.78 DAC_CAL1 寄存器（偏移 = 424h）[复位 = 00000000h]

图 11-100 展示了 DAC_CAL1，表 11-104 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-100. DAC_CAL1 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SOC_DAC_VOUT_CAL_PRECH_C2															
R-X															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOC_DAC_VOUT_CAL_PRECH_C1															
R-X															

表 11-104. DAC_CAL1 寄存器字段说明

位	字段	类型	复位	说明
31-16	SOC_DAC_VOUT_CAL_PRECH_C2	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
15-0	SOC_DAC_VOUT_CAL_PRECH_C1	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.79 DAC_CAL2 寄存器（偏移 = 428h）[复位 = 00000000h]

图 11-101 展示了 DAC_CAL2，表 11-105 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-101. DAC_CAL2 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SOC_DAC_VOUT_CAL_ADCREF_C2															
R-X															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOC_DAC_VOUT_CAL_ADCREF_C1															
R-X															

表 11-105. DAC_CAL2 寄存器字段说明

位	字段	类型	复位	说明
31-16	SOC_DAC_VOUT_CAL_ADCREF_C2	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
15-0	SOC_DAC_VOUT_CAL_ADCREF_C1	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

11.4.1.80 DAC_CAL3 寄存器（偏移 = 42Ch）[复位 = 00000000h]

图 11-102 展示了 DAC_CAL3，表 11-106 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 11-102. DAC_CAL3 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SOC_DAC_VOUT_CAL_VDDS_C2															
R-X															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOC_DAC_VOUT_CAL_VDDS_C1															
R-X															

表 11-106. DAC_CAL3 寄存器字段说明

位	字段	类型	复位	说明
31-16	SOC_DAC_VOUT_CAL_VDDS_C2	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。
15-0	SOC_DAC_VOUT_CAL_VDDS_C1	R	X	内部。仅可通过 TI 提供的 API 使用。 默认值保存来自生产测试的修整值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

章节 12

加密

CC13x2 和 CC26x2 器件平台的安全内核配备了具有本地密钥库和 DMA 功能的高级加密标准 (AES) 模块。

CC13x2 和 CC26x2 器件平台还配备了支持 256 位密钥的 AES 模块、一个哈希引擎，并且安全特性支持 PKA 引擎。本章介绍有关配置 AES 和哈希引擎的信息。

12.1 AES 和哈希加密处理器简介 929
12.2 功能说明 929
12.3 电源管理和睡眠模式 930
12.4 硬件说明 931
12.5 模块说明 932
12.6 AES 模块性能 943
12.7 编程指南 945
12.8 约定与合规性 982
12.9 加密寄存器 983

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.1 AES 和哈希加密处理器简介

AES 安全模块可基于二进制密钥提供硬件加速的数据加密和解密操作。该模块在硬件中支持用于加密和解密的 256 位密钥，并使用对称算法，这意味着加密和解密密钥是相同的。加密可将明文数据转换为难以理解的名为*密文*的形式。密文解密是将之前加密的数据转换回其原始形式，即明文。AES 模块的主要特性为：

- 支持和提供以下工作模式：
  - 电子源码书模式 (ECB)
  - 密码块链接 (CBC)
  - 密码块链消息认证码 (CBC-MAC)
  - 计数器模式 (CTR)
  - 带 CBC-MAC 的计数器模式 (CCM)
  - 带 CBC-GCM 的计数器模式 (GCM)
- 密钥大小：128 位、192 位和 256 位
- 支持 CBC-MAC 身份验证模式
- 硬件中的密钥调度
- 支持 µDMA 传输
- 全同步设计
- 支持哈希算法（SHA-224、SHA-256、SHA-384 和 SHA-512）

ECB、CBC、CTR、GCM 和 CCM 模式需要读取及写入数据。CBC-MAC 仅需要从外部源读取数据。CCM 运行模式会返回身份验证结果。该结果可以通过单独的 DMA 运行读取，也可以通过从器件接口读取。对于所有模式，一个选项通过从器件接口而不是使用 DMA 提供（一部分）数据。AES 引擎被强制使用密钥存储区模块中的密钥进行操作。通过触发密钥存储区模块，向 AES 引擎提供一个密钥，以便从密钥存储区存储器中读取 AES 密钥并将其写入 AES 密钥寄存器。对于 AES CBC-MAC、GCM 和 CCM（包括未对齐的 AAD 数据），AES 引擎会自动用零填充或屏蔽未对齐的最后数据块。对于 AES CTR 模式，未对齐的最后一个数据块在内部被屏蔽，以支持非块大小输入数据。

12.2 功能说明

AES 引擎直接连接到上下文和数据寄存器，以便它可以在所有数据可用时立即开始处理。AES 引擎还与 I/O 控制 FSM 和 µDMA 请求接口连接。AES 包含以下主要功能块：

- 全局控制 FSM 和 µDMA 接口
- 寄存器接口模块
- AES 引擎

AES 引擎是主要的顶级组件，包含以下功能块：

- 模式控制 FSM：管理进出 AES 引擎的数据流，并启动每个加密和解密操作
- 反馈模式：实现 AES 支持的各种反馈模式的逻辑
- AES 密钥调度程序：生成 AES 加密及解密（轮次）密钥
- AES 加密内核：AES 加密算法
- AES 解密内核：AES 解密算法
- 替换框 (S-box)：包含 AES S-Box GF(2^8) 实现。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

对于 128 位的密钥长度，需要 10 轮或 32 个时钟周期，因为 {时钟周期数} = 2 + 3 × {轮数}。当处理一个数据块时，可立即预加载下一个数据块。预加载一个数据块后，只有上一个数据块完成后才能加载更多数据。因此，一旦流水线已满，就可以每 32 个时钟周期传递一次顺序数据块。

哈希引擎支持基本的 SHA-256、SHA-224、SHA-512 及 SHA-384 操作。它只需要从外部源读取输入数据，该数据通过 DMAC 模块传输。可使用 DMA 运行或通过从接口读取哈希结果来传输哈希结果。该模块还支持带密钥的哈希操作（如 HMAC），其中部分数据可由 AHB 从接口提供，而不是使用 DMA

12.2.1 调试功能

AES 模块提供以下状态寄存器来监控引擎的运行：

- DMA 状态和端口错误状态寄存器
- 主控制模块内的中断状态寄存器
- 密钥库模块状态寄存器

12.2.2 异常处理

AES 模块可以检测 AHB 主总线错误并中止 DMA 操作。AES 密钥存储模块可以检测密钥加载错误，并且在这种情况下不会存储*错误*的密钥。在这两种情况下，主控制模块中的状态寄存器都会指示该错误。

12.3 电源管理和睡眠模式

加密寄存器没有保持逻辑。时钟可以通过以下 PRCM 寄存器使能或门控：

- 运行模式下，SECDMACLKGR.CRYPTO_CLK_EN 位
- 睡眠模式下，SECDMASCLKG.CRYPTO_CLK_EN 位
- 深度睡眠模式下，SECDMACLKGDS.CRYPTO_CLK_EN 位

加密模块通过 SECDMAHWOPT.CRYPTO_EN 位进行使能和禁用。

为节省功耗，应用程序可以在 AES 模块不使用时禁用其时钟。在睡眠模式下，通过设置 SECDMACLKGS 寄存器的 CRYPTO_CLK_EN 位来门控 AES 时钟。在运行模式下，也可以通过设置 SECDMACLKGR 寄存器的 CRYPTO_CLK_EN 位来门控 AES 时钟。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.4 硬件说明

12.4.1 AHB 从总线

AES 模块的内部寄存器由从接口访问。AHB 从器件接口接受 8 位、16 位及 32 位传输。但是，AES 模块仅接受 32 位单次访问。

由于根据传输的地址、大小和类型检查每次传输是否存在多个错误条件，因此会对寄存的信号执行这些检查，以改善输入信号的时序。因此，每次传输都必须插入一个等待周期。如果发生错误响应，则必须在接收到地址后的一个周期内将 h_ready_out 置为低电平。这就产生了如下时序：

- 写入传输需要两个时钟周期。
- 读取传输需要三个时钟周期。

AHB 从器件仅处理小端字节序传输，并且对于寄存器访问，仅允许 32 位单次访问。

12.4.2 AHB 主总线

该模块通过 DMA 配置 DMABUSCFG 寄存器（请参阅节 12.9.1）进行配置，默认情况下执行单次 8 位或 32 位非顺序单次传输。DMA 传输的传输地址及长度参数采用字节对齐方式。

当 AES 模块请求 DMA 传输时，AHB 主器件置为有效并发出信号以向仲裁器指示它需要总线。在 DMA 最后一次传输的地址阶段之前，此信号一直有效，并且没有请求新的 DMA 传输。

当没有请求 DMA 传输时，AHB 主器件会执行 IDLE 传输。如果 AHB 主器件已被授予并获得 DMA 请求，则第一次写入传输是 IDLE 传输。最后一个传输始终是 IDLE 传输。

如果 AHB_MST1_LOCK_EN 位被置为有效，则 AHB 主器件会断言锁定信号，以指示 AHB 正在执行多个不可分割的传输。当锁定传输序列的第一次传输开始时，仲裁器不授予任何其他 AHB 主器件对总线的访问权限。AHB 主器件在每个块序列之后插入一个 IDLE 传输。

AHB 主器件可以处理大端序及小端序传输。AES 模块在内部为小端序导向。但是，当连接到大端序 AHB 系统时，可以在 AHB 主接口中完成大端序到小端序的转换。默认情况下，假定采用小端序导向的 AHB 主机系统。当 AHB 系统为大端序定向时，AHB_MST1_BIGEND 位必须设置为 1。

备注

CC13x2 和 CC26x2 器件平台不支持通过内部互连进行突发或者非顺序传输。不得更改 DMABUSCFG 寄存器以确保正常运行。

12.4.3 中断

AES 模块有两个中断输出；二者均由主控制模块驱动，并受相应寄存器控制（请参阅节 12.5.4.3）。

要启用 AES 引擎的中断，必须置位 IRQTYPE.EN 位，并且在 IRQEN 寄存器中配置中断源。

可使用 IRQCLR 寄存器清除中断输出和错误状态位。IRQSET 寄存器为软件提供了一种测试中断连接的方法，并且只能用于调试。

IRQSTAT 寄存器提供两个中断的状态和错误状态消息。错误状态位在检测到时置位，并且通常在置位 RESULT_AVAIL 位后，DMA_BUS_ERR 和 KEY_ST_WR_ERR 信号的值即有效。在触发密钥库模块从存储器中读取密钥并提供给 AES 引擎后，KEY_ST_RD_ERR 位即有效。

当使用 DMA 的操作完成时，会激活中断 RESULT_AVAIL。当 DMA 和内部模块都处于 IDLE 状态时，该信号会置为有效。

另一个中断 DMA_IN_DONE 仅在输入 DMA 完成时激活，用于调试。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

备注

对于不使用 DMA 的操作，不会触发中断输出。

12.5 模块说明

12.5.1 简介

本节介绍一些可访问的寄存器、内部接口及模块功能。将分别讨论每个子模块的寄存器和功能。有关模块寄存器的完整信息，请参阅节 12.9.1。

12.5.2 模块存储器映射

表 12-1 列出了存储器映射详情。有关说明，请参阅节 12.9.1。

表 12-1. 详细的存储器映射 (1)

物理地址	寄存器名称	类型	复位值	注释
DMA 控制器寄存器				
0x4002 4000	DMACH0CTL	R/W	0x0000 0000	通道 0 控制寄存器
0x4002 4004	DMACH0EXTADDR	R/W	0x0000 0000	通道 0 外部地址
0x4002 400C	DMACH0LEN	R/W	0x0000 0000	通道 0 DMA 长度
0x4002 4018	DMASTAT	R	0x0000 0000	DMAC 状态
0x4002 401C	DMASWRESET	W	0x0000 0000	DMAC 软件复位
0x4002 4020	DMACH1CTL	R/W	0x0000 0000	通道 1 控制寄存器
0x4002 4024	DMACH1EXTADDR	R/W	0x0000 0000	通道 1 外部地址
0x4002 402C	DMACH1LEN	R/W	0x0000 0000	通道 1 DMA 长度
0x4002 4078	DMABUSCFG	R/W	0x0000 6000	主控运行时参数
0x4002 407C	DMAPORTERR	R	0x0000 0000	端口错误原始状态寄存器
0x4002 40F8	DMAHWOPT	R	0x0000 0202	DMAC 选项寄存器
0x4002 40FC	DMAHWVER	R	0x0101 2ED1	DMAC 版本寄存器
密钥存储寄存器				
0x4002 4400	KEYWRITEAREA	R/W	0x0000 0000	写入端区域寄存器
0x4002 4404	KEYWRITTENAREA	R/W	0x0000 0000	写入区域寄存器
0x4002 4408	KEYSIZE	R/W	0x0000 0001	密钥大小寄存器
0x4002 440C	KEYREADAREA	R/W	0x0000 0008	读取区域寄存器

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 12-1. 详细的存储器映射 (1) （续）

物理地址	寄存器名称	类型	复位值	注释
AES 引擎寄存器				
0x4002 4500 至 0x4002 450C	AESKEY2_0 至 AESKEY2_3	W	0x0000 0000	清除/擦除 AESKEY2__0 至 AESKEY2__3 寄存器
0x4002 4510 至 0x4002 451C	AESKEY3_0 至 AESKEY3_3	W	0x0000 0000	清除/擦除 AESKEY3__0 至 AESKEY3__3 寄存器
0x4002 4540 至 0x4002 454C	AESIV_0 至 AESIV_3	R/W	0x0000 0000	AES IV (LSW)
0x4002 4550	AESCTL	R/W	0x8000 0000	I/O 和控制模式
0x4002 4554	AESDATALEN0	W	0x0000 0000	加密数据长度 (LSW)
0x4002 4558	AESDATALEN1	W	0x0000 0000	加密数据长度 (MSW)
0x4002 455C	AESAUTHLEN	W	0x0000 0000	AAD 数据长度
0x4002 4560	AESDATAOUT0	R	0x0000 0000	数据输出 (LSW)
0x4002 4560	AESDATAIN0	W	0x0000 0000	数据输入 (LSW)
0x4002 4564	AESDATAOUT1	R	0x0000 0000	数据输出
0x4002 4564	AESDATAIN1	W	0x0000 0000	数据输入
0x4002 4568	AESDATAOUT2	R	0x0000 0000	数据输出
0x4002 4568	AESDATAIN2	W	0x0000 0000	数据输入
0x4002 456C	AESDATAOUT3	R	0x0000 0000	数据输出 (MSW)
0x4002 456C	AESDATAIN3	W	0x0000 0000	数据输入 (MSW)
0x4002 4570 至 0x4002 4057C	AESTAGOUT_0 至 AESTAGOUT_3	W	0x0000 0000	标签输出 (LSW)
哈希引擎寄存器				
0x600	HASH_DATA_IN_0	W	0x0000 0000	数据输入位 [31:0] (LSW)
0x604	HASHDATAIN1	W	0x0000 0000	数据输入位 [63:32]
0x608	HASHDATAIN2	W	0x0000 0000	数据输入位 [95:64]
0x60C	HASHDATAIN3	W	0x0000 0000	数据输入位 [127:96]
0x610	HASHDATAIN4	W	0x0000 0000	数据输入位 [159:128]
0x614	HASHDATAIN5	W	0x0000 0000	数据输入位 [191:160]
0x618	HASHDATAIN6	W	0x0000 0000	数据输入位 [223:192]
0x61C	HASHDATAIN7	W	0x0000 0000	数据输入位 [255:224]
0x620	HASHDATAIN8	W	0x0000 0000	数据输入位 [287:256]
0x624	HASHDATAIN9	W	0x0000 0000	数据输入位 [319:288]
0x628	HASHDATAIN10	W	0x0000 0000	数据输入位 [351:320]
0x62C	HASHDATAIN11	W	0x0000 0000	数据输入位 [383:352]
0x630	HASHDATAIN12	W	0x0000 0000	数据输入位 [415:384]
0x634	HASHDATAIN13	W	0x0000 0000	数据输入位 [447:416]
0x638	HASHDATAIN14	W	0x0000 0000	数据输入位 [479:448]
0x63C	HASHDATAIN15	W	0x0000 0000	数据输入位 [511:480]
0x640	HASHDATAIN16	W	0x0000 0000	数据输入位 [543:512]
0x644	HASHDATAIN17	W	0x0000 0000	数据输入位 [575:544]
0x648	HASHDATAIN18	W	0x0000 0000	数据输入位 [607:576]
0x64C	HASHDATAIN19	W	0x0000 0000	数据输入位 [639:608]
0x650	HASHDATAIN20	W	0x0000 0000	数据输入位 [671:640]
0x654	HASHDATAIN21	W	0x0000 0000	数据输入位 [703:672]

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 12-1. 详细的存储器映射 (1) （续）

物理地址	寄存器名称	类型	复位值	注释
0x658	HASHDATAIN22	W	0x0000 0000	数据输入位 [735:704]
0x65C	HASHDATAIN23	W	0x0000 0000	数据输入位 [767:736]
0x660	HASHDATAIN24	W	0x0000 0000	数据输入位 [799:768]
0x664	HASHDATAIN25	W	0x0000 0000	数据输入位 [831:800]
0x668	HASHDATAIN26	W	0x0000 0000	数据输入位 [863:832]
0x66C	HASHDATAIN27	W	0x0000 0000	数据输入位 [895:864]
0x670	HASHDATAIN28	W	0x0000 0000	数据输入位 [927:896]
0x674	HASHDATAIN29	W	0x0000 0000	数据输入位 [959:928]
0x678	HASHDATAIN30	W	0x0000 0000	数据输入位 [991:960]
0x67C	HASHDATAIN31	W	0x0000 0000	数据输入位 [1023:992] (MSW)
0x680	HASHIOBUFCTRL	W	0x0000 0000	I/O 缓冲器控制
0x680	HASH_IO_BUF_STAT	R	0x0000 0004	I/O 缓冲器状态
0x684	HASHMODE	W	0x0000 0000	模式输入寄存器
0x688	HASHINLENL	W	0x0000 0000	长度输入位 [31:0] (LSW)
0x68C	HASHINLENH	W	0x0000 0000	长度输入位 [63:32] (MSW)
0x6C0	HASHDIGESTA	R/W	0x0000 0000	哈希摘要位 [31:0] (LSW)
0x6C4	HASHDIGESTB	R/W	0x0000 0000	哈希摘要位 [63:32]
0x6C8	HASHDIGESTC	R/W	0x0000 0000	哈希摘要位 [95:64]
0x6CC	HASHDIGESTD	R/W	0x0000 0000	哈希摘要位 [127:96]
0x6D0	HASHDIGESTE	R/W	0x0000 0000	哈希摘要位 [159:128]
0x6D4	HASHDIGESTF	R/W	0x0000 0000	哈希摘要位 [191:160]
0x6D8	HASHDIGESTG	R/W	0x0000 0000	哈希摘要位 [223:192]
0x6DC	HASHDIGESTH	R/W	0x0000 0000	哈希摘要位 [255:224]
0x6E0	HASHDIGESTI	R/W	0x0000 0000	哈希摘要位 [287:256]
0x6E4	HASHDIGESTJ	R/W	0x0000 0000	哈希摘要位 [319:288]
0x6E8	HASHDIGESTK	R/W	0x0000 0000	哈希摘要位 [351:320]
0x6EC	HASHDIGESTL	R/W	0x0000 0000	哈希摘要位 [383:352]
0x6F0	HASHDIGESTM	R/W	0x0000 0000	哈希摘要位 [415:384]
0x6F4	HASHDIGESTN	R/W	0x0000 0000	哈希摘要位 [447:416]
0x6F8	HASHDIGESTO	R/W	0x0000 0000	哈希摘要位 [479:448]
0x6FC	HASHDIGESTP	R/W	0x0000 0000	哈希摘要位 [511:480] (MSW)
主控制寄存器				
0x4002 4700	ALGSEL	R/W	0x0000 0000	算法选择
0x4002 4704	DMAPROTCTL	R/W	0x0000 0000	使能主控上的特权访问
0x4002 4740	SWRESET	W	0x0000 0000	主控制软件复位
0x4002 4780	IRQTYPE	R/W	0x0000 0000	中断配置寄存器
0x4002 4784	IRQEN	R/W	0x0000 0000	中断使能寄存器
0x4002 4788	IRQCLR	W	0x0000 0000	中断清除寄存器
0x4002 478C	IRQSET	W	0x0000 0000	中断置位寄存器
0x4002 4790	IRQSTAT	R	0x0000 0000	中断状态寄存器
0x4002 47F8	HWOPT	R	0x0101 01F7	类型和选项寄存器

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 12-1. 详细的存储器映射 (1) （续）

物理地址	寄存器名称	类型	复位值	注释
0x4002 47FC	HWVER	R	0x9200 8778	版本寄存器

(1) 未指定的地址为保留地址，禁止写入且在读取时应忽略。

12.5.3 DMA 控制器

图 12-1 显示了 DMA 控制器 (DMAC) 及其在 AES 模块中的集成。

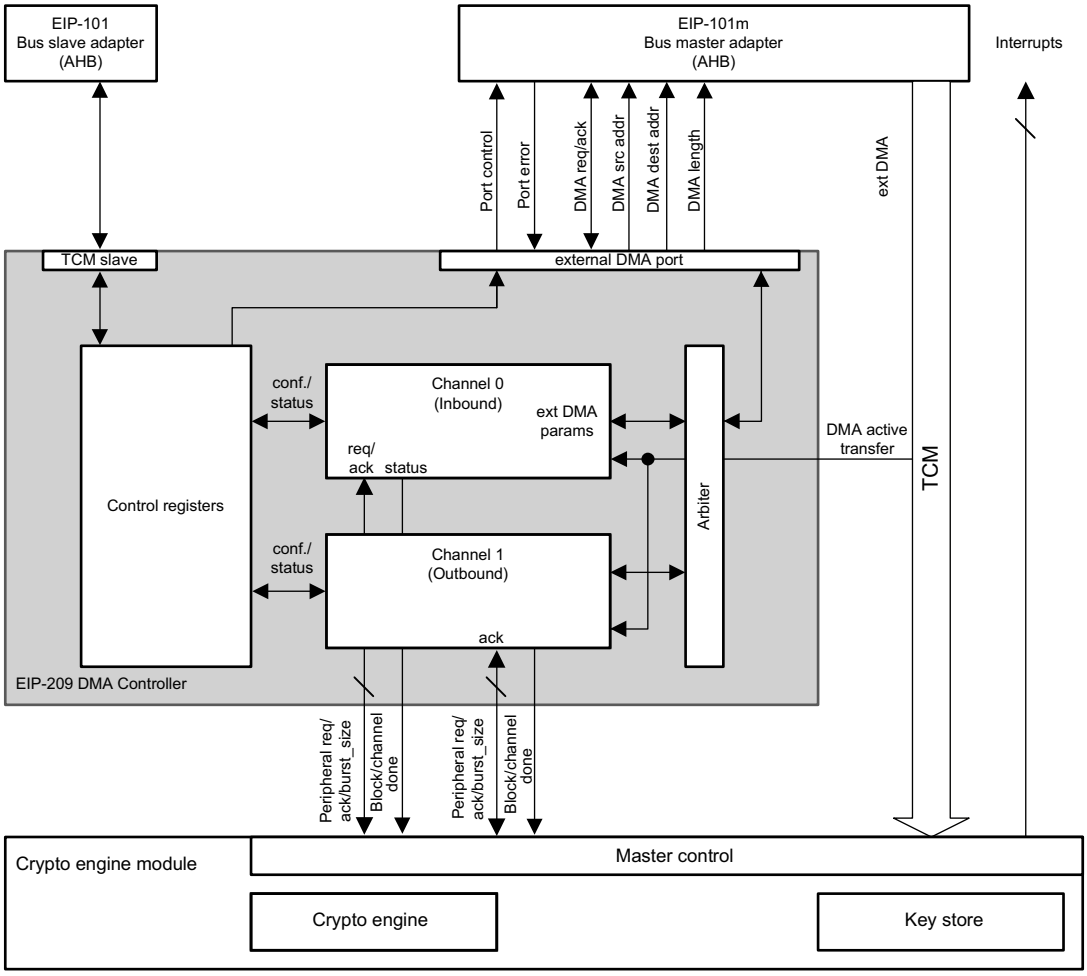


图 12-1. DMA 控制器及集成

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

AES 模块的 DMAC 控制发送到 AHB 主适配器的数据传输请求，该适配器在 AES 引擎和密钥存储区域之间进行数据传输。

AHB 主器件接口端口正常运行所需的参数在 DMABUSCFG 寄存器中定义。该寄存器的默认配置会配置固定长度传输及 4 字节的最大突发大小。因此，在 AHB 总线上仅执行非顺序单次传输。

DMASTAT 和 DMAPORTERR 寄存器提供每个 DMA 通道的实际状态和各个 AHB 端口错误。端口错误会中止所有服务通道上的操作，并防止使用该端口进行进一步的传输，直到通过写入 DMASWRESET 寄存器来清除错误。

如果地址和长度为 32 位对齐，则主器件只进行大小为 4 字节的 NONSEQ 类型和 SINGLE 类型传输。

DMAC 将通道 DMA 运行拆分成小型 DMA 传输。小型 DMA 传输的大小由目标内部模块决定，等于加密操作的块大小。

该 DMAC 具有以下特性：

- 可同时启用的两个通道（一个入站通道和一个出站通道）
- 由 16 位长寄存器控制 DMA 运行的最大大小
- 具有一个仲裁器，用于安排对外部 AHB 端口的通道访问
- 具有捕获外部总线错误的功能

DMAC 由两个优先级可编程的 DMA 通道组成：一个可编程通道用于将输入数据和密钥从外部存储器移动到 AES 模块，另一个可编程通道用于将结果数据从 AES 模块移动到外部存储器。对 AHB 主端口通道的访问由仲裁器模块处理。

通道控制寄存器用于通道启用及优先级选择。当某个通道被禁用时，只有当所有正在进行的请求都完成时，该通道才会变为非活动状态。

备注

主机必须对所有通道控制寄存器（DMACHxCTL、DMACHxEXTADDR 和 DMACHxLEN）进行编程才能启动新的 DMA 运行。

DMAC 在源地址和目的地址之间传输数据。从非字对齐边界开始，生成字节传输，直至到达字边界。然后，只要数据可用，就会生成字传输。如果传输未在字对齐地址完成，则剩余的传输将再次进行字节传输。

备注

不生成半字传输。

当 AHB_MST1_INCR_EN 位设置为 1 时，认情况下会生成定义长度的突发和单次传输。最大大小取决于编程的突发大小。

DMAC 寄存器映射到外部寄存器映射。要开始操作，主机必须对 DMAC 的模式和操作参数进行编程。这些参数涉及方向（读取、写入或读取和写入）、长度（1 至 65,535 字节）、外部源地址（用于读取）和外部目标地址（用于写入）。有关寄存器的详细信息，请参阅节 12.9.1。

备注

内部目标使用主控制模块中的专用算法选择寄存器进行编程。根据该寄存器的设置将突发大小提供给 DMAC。

12.5.3.1 内部操作

DMAC 使用具有两个端口的 AHB 主适配器工作。一个端口是外部 AHB 端口，用于对外部 AHB 子系统执行读取和写入操作。该端口可对完整的 32 位地址范围进行寻址。第二个端口是内部 TCM 端口（主 TCM），用于对加密

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

内核 AES 引擎和密钥存储区的内部模块执行读写操作。必须在主控制模块中选择分配用于 DMA 运行的内部模块（请参阅节 12.5.4.1.1）；因此，DMAC 中不需要内部地址。

从 AHB 主模块的 TCM 端口到内部模块的数据路径位于 DMAC 外部。DMAC 仅观察传输的字数，以确定相应通道所请求的 DMA 运行何时完成。

密钥存储区是一个 32 位存储器块，深度为 32 个字，整体采用了控制逻辑。当 AES 模块配置为通过 DMA 将密钥写入该密钥存储区模块时，密钥存储区会根据其寄存器设置在内部管理对密钥存储区 RAM 的访问（包括生成密钥库 RAM 地址）。AES 模块仅支持对密钥存储区进行 DMA 写入操作。

AES 引擎具有一个用于加密或解密输入数据的 32 位写入接口，和一个用于结果数据和标签的 32 位读取接口。AES 模块的写入接口将 32 位数据收集至 128 位输入块中（AES 块大小）。当接收到一个完整块时，接收到的块的 AES 计算开始。当接收到最后一个块的最后一个字时，DMAC 和主控制器会向加密引擎生成一个“数据完成”信号。使用目标接口，对模式、消息长度和可选参数进行编程。

在 TCM 端，密钥库模块立即接受所有数据，无延迟周期，而加密模块在数据块边界上运行。在 TCM 端，密钥库模块立即接受所有数据，无延迟周期，而加密模块在数据块边界上运行（其处理需要多个时钟周期）。DMAC 和加密模块之间使用了特殊的握手信号：

- 当加密模块可以接受下一个数据块时，数据输入请求将发送到 DMA 入站通道（通道 0）。
- 在进行处理或哈希模块有可用摘要后，当加密模块的下一个数据块或标签可用时，数据输出请求会发送到 DMA 输出通道（通道 1）。
- 在 DMA 运行开始、通道传输完成、发送了块且通道传输完成时，或发送了所有数据时，两个通道都发送确认。

12.5.3.2 支持的 DMA 运行

对于来自加密引擎的每个数据请求，DMAC 都会请求来自 AHB 主器件的传输。传输大小最多为相应算法的块大小。该块大小取决于在主控制模块中选择的算法。

表 12-2 提供支持的 DMAC 操作的摘要。该模块是指主控制模块中选定的模块。在表 12-2 中，*TAG 已启用* 标签表示已在主控配置寄存器中设置 TAG 位。

表 12-2. 支持的 DMAC 操作

模块	传入数据流（用于通道 0）		输出数据流（用于通道 1）	
	源	目标	源	目标
密钥存储区	外部存储器位置	密钥存储区 RAM	-	-
Crypto	RAM（仅限身份验证数据）	AES	请参阅 (1)	请参阅 (1)
	外部存储器位置	AES	AES	外部存储器位置
	请参阅 (2)	请参阅 (2)	AES（TAG 已启用）	外部存储器位置
哈希值	外部存储器位置	SHA-2（TAG 禁用）	请参阅 (1)	请参阅 (1)
	请参阅 (2)	请参阅 (2)	SHA-2（TAG 已启用）	外部存储器位置
	外部存储器位置	SHA-2（TAG 禁用）	SHA-2（TAG 已启用）	外部存储器位置

(1) TAG 通过从器件接口传输或使用单独的 DMA 传输。
(2) 数据通过之前已执行的另一个 DMA 传输。

12.5.4 主控制和选择模块

主控制模块可同步 DMA 操作及加密模块握手信号。在此模块中，将选择加密算法并定义 DMA 突发大小。完整的加密操作完成后，会将中断置为有效。

备注

对于身份验证操作，仅当身份验证结果可用时才将中断置为有效。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

AES 模块还提供一个中断来指示输入 DMA 传输已完成。该中断主要用于确定 AAD 数据 DMA 传输 (AES-CCM) 的结束，该传输通常设置为单独的输入数据传输。

12.5.4.1 算法选择寄存器

算法选择寄存器配置 DMAC 的内部目标。

12.5.4.1.1 算法选择

表 12-3 总结了 ALGSEL 寄存器允许的位组合。

表 12-3. ALGSEL 标志的有效组合

运行	标志			
	KEY STORE	AES	哈希值	TAG
哈希数据通过 DMA 加载；结果摘要由从器件接口读取。	0	0	1	0
哈希数据通过 DMA 加载；结果摘要由 DMA 接口读取。	0	0	1	1
密钥存储区通过 DMA 加载。	1	0	0	0
AES 数据通过 DMA 加载，加密和解密的数据通过 DMA（加密和解密）读取。 或者 AES 数据通过 DMA 加载，然后结果标签通过从器件接口读取（仅身份验证操作）。	0	1	0	0
AES 数据通过 DMA 加载；结果标签通过 DMA 读取（仅身份验证操作）。	0	1	0	1

12.5.4.2 主器件 PROT 启用

12.5.4.2.1 主 PROT 特权访问启用

DMAPORTCTL 寄存器为 DMA 传输选择 AHB 传输保护控制，使用密钥库模块作为目标。

12.5.4.3 软件复位

有关软复位程序的更多详细信息，请参阅节 12.7.5.1。

要对 AES 模块执行软件复位，请向 SWRESET 寄存器中的 RESET 位写入 1。当软件复位完成时，SWRESET 寄存器中的 RESET 位会自动复位。软件必须确保软件复位完成后才能开始任何操作。

在 DMA 控制模块中，软件复位用于复位 DMAC，以停止所有传输并将 DMAPORTERR 寄存器清零。执行软件复位后，所有通道都被禁用，并且通道不会执行新的请求。DMAC 等待现有（活动的）请求完成，然后设置 DMAC 状态寄存器。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.5.5 AES 引擎

AES 内核的组成如下：

- 主数据路径在输入块上运行，执行所需的替代、移位和混合操作。
- 密钥调度程序生成轮次密钥。每轮生成一个新的子密钥，并用数据进行 XOR 运算。

AES 密钥调度程序会生成轮次密钥。在每个轮次，输入密钥将生成一个新的子密钥，用数据进行 XOR 运算。轮次密钥是动态生成的，与数据处理并行，可最大限度地降低寄存器要求。对于加密操作，密钥序列发生器会将初始密钥数据传输到 AES 内核。对于解密操作，密钥调度程序必须向 AES 内核提供最终子密钥，以便其能够以相反的顺序生成子密钥。

AES 内核在输入块上运行并执行所需的替换、移位及混合运算。对于每个轮次，加密内核都会从 AES 密钥调度程序接收适当的轮次密钥。AES 算法的一个基本组成部分是 S-box。S-box 为每个 8 位输入
提供唯一的 8 位输出。

AES 解密内核的架构通常与加密内核的架构相同。一个区别是，用于解密的轮次密钥生成需要将输入密钥（始终由主机以加密密钥的形式提供）初始转换为相应的解密密钥。这种转换通过执行虚拟加密操作并将最终轮次密钥存储为解密密钥来完成。然后，反转密钥调度程序以生成用于解密操作的轮次密钥。因此，对于同一密钥下的每个解密操作序列，会发生与加密单个块时间相对应的单个吞吐量降低。生成解密密钥时，使用相同密钥的后续解密操作会直接使用生成的解密密钥。

12.5.5.1 第二密钥寄存器（内部，但可清除）

表 12-4 和 表 12-5 中列出的寄存器无法通过主机进行读取和写入访问。这些寄存器用于存储内部计算的密钥信息和中间结果。但是，当主机分别对相应的 AESKEY2__0 至 AESKEY2__3 或 AESKEY3__0 至 AESKEY3__3 地址执行写入操作时，整个 128 位 AESKEY2__0 至 AESKEY2__3 或 AESKEY3__0 至 AESKEY3__3 寄存器会被清零。

AES_GHASH_IN_n 寄存器（GHASH 所必需，而 GHASH 是 GCM 的一部分的）映射到 AES_KEY2_n 寄存器。GCM 和 CCM 的中间身份验证结果存储在 AESKEY3_n 寄存器中。

表 12-4. AES_KEY

AESKEY2__0 至 AESKEY2__3（只写），32 位地址偏移：0x500 至 0x50C，以 0x4 字节为增量																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AESKEY2__0 至 AESKEY2__3[31:0] AESKEY2__0 至 AESKEY2__3[63:32] AESKEY2__0 至 AESKEY2__3[95:64] AESKEY2__0 至 AESKEY2__3[127:96]																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 12-5. AES_KEY

AESKEY3__0 至 AESKEY3__3（只写），32 位地址偏移：0x510 至 0x51C，以 0x4 字节为增量																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AESKEY3__0 至 AESKEY3__3[31:0] / AESKEY2__0 至 AESKEY2__3[159:128] AESKEY3__0 至 AESKEY3__3[63:32] / AESKEY2__0 至 AESKEY2__3[191:160] AESKEY3__0 至 AESKEY3__3[95:64] / AESKEY2__0 至 AESKEY2__3[223:192] AESKEY3__0 至 AESKEY3__3[127:96] / AESKEY2__0 至 AESKEY2__3[255:224]																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 12-6. 对于 CCM

位	字段名称	功能
255-0	-	该寄存器用于存储中间值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 12-7. 对于 CBC-MAC

位	字段名称	功能
255-0	零点	该寄存器必须保持为零。

表 12-8. 对于 GCM

位	名称	功能
127-0	GHASH_H	内部计算出的 GHASH 密钥存储在这些寄存器中。仅用于使用 GHASH 函数 (GCM) 的模式。
255-128	-	该寄存器用于存储中间值，并在加载新密钥时初始化为零。

可重复使用 AES_KEYn 寄存器来进行顺序操作；但对于 CBC-MAC，在对相应的模式和长度参数进行编程时，必须清除中间值。

在不加载新密钥 (通过密钥存储区) 的情况下执行 GCM 操作时，需要对其中一个 AES_KEY3 寄存器位置进行写入以清除寄存器。

如果在不加载新密钥 (通过密钥存储区) 的情况下启动 CBC-MAC 操作，并且之前的操作不是 CBC-MAC 操作，则在开始 CBC-MAC 操作之前，必须先执行 AESKEY2__0 至 AESKEY2__3 以及 AESKEY3__0 至 AESKEY3__3 寄存器位置的写入操作，这是清除这两个密钥寄存器所必需的操作。

12.5.5.2 AES 初始化矢量 (IV) 寄存器

表 12-9 展示了用于提供和从 AES 引擎读取 IV 的 AES 初始化矢量寄存器。

表 12-9. AES 初始化矢量寄存器

AES_IV_0 (读取/写入) ，32 位地址偏移：0x540
AES_IV_1，(读取/写入) ，32 位地址偏移：0x544
AES_IV_2，(读取/写入) ，32 位地址偏移：0x548
AES_IV_3，(读取/写入) ，32 位地址偏移：0x54C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AES_IV[31:0] AES_IV[63:32] AES_IV[95:64] AES_IV[127:96]																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 12-10. 初始化矢量，用于常规非 ECB 模式 (CBC/ECB CTR)

位	名称	说明
127-0	AES_IV	对于常规 AES 操作 (CBC 和 CTR) ，这些寄存器必须使用新的 128 位 IV 进行写入。 运算后，这些寄存器包含由 EIP-120t 生成的最新 128 位结果 IV。 如果选择了 CTR 模式，则当新的数据块提交到引擎时，此值会以 0x1 (首次使用后) 递增。

表 12-11. 对于 GCM

位	名称	说明
127-0	AES_IV	对于 GCM 操作，必须使用新的 128 位 IV 写入这些寄存器。 运算后，这些寄存器包含由 EIP-120t 生成的已更新 128 位结果 IV。 IV 的位 [127 - 96] 表示初始计数器值 (对于 GCM，该值为 1) 、因此必须初始化为 0x0100 0000。 当新的数据块提交到引擎时，此值以 0x1 递增 (首次使用后) 。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 12-12. 初始化矢量，用于 CCM

位	名称	说明
127-0	A0	对于 CCM，此字段必须写入值 A0。此值是以下各项的连接： A0 标志（5 位为 0，3 位为 L），随机数和计数器值。 L 必须是 AESCTL 寄存器 L 值的副本。此 L 指示随机数和计数器的宽度。 加载的计数器必须初始化为零。 A0 的总宽度为 128 位。

表 12-13. 初始化矢量，用于 CBC-MAC

位	名称	说明
127-0	零	对于 CBC-MAC，每次操作开始时，该寄存器必须写入零。操作之后，这些寄存器包含由加密内核生成的 128 位 TAG 输出。

12.5.5.3 AES I/O 缓冲器控制、模式和长度寄存器

I/O 缓冲器和模式控制寄存器 (AESCTL) 指定 AES 引擎的运行模式。

备注

通过将所有模式位设置为 0 并向长度寄存器（AESDATALEN0、AESDATALEN1 和 AESAUTHLEN [请参阅节 12.9.1]）写入零，可以中断 AES 模块的内部运行。

长度寄存器将长度值写入 AES 模块。处理期间，长度值递减至 0。如果两个长度均为 0，则表示数据流完成，并会请求新的上下文。对于基本 AES 模式（ECB、CBC 和 CTR），如果必须使用同一密钥处理多个流，则可以写入加密长度 0。写入 0 长度将导致连续的数据请求，直到写入新的上下文。对于其他模式（CBC-MAC、GCM 和 CCM），如果长度递减至 0 或等于 0，则不会发出新的数据请求。

TI 建议为每个数据包写入一个新长度。如果长度寄存器递减至 0，则在写入新的上下文或长度值之前不会处理任何新数据。

如果写入新模式时未写入长度寄存器，则会重复使用先前上下文的长度寄存器值。

12.5.5.4 数据输入和输出寄存器

AESDATAINn 和 AESDATAOUTn 数据寄存器通常通过 DMA 访问，而不是通过主机写入和读取来访问。然而，出于调试目的，可通过主机写入和读取操作来访问数据输入和输出寄存器。这些寄存器用于缓冲进出加密内核的输入和输出数据块（请参阅表 12-14）。

备注

数据输入缓冲器 AESDATAINn 和数据输出缓冲器 AESDATAOUTn 映射到相同的地址位置。

向这些地址写入值（DMA 和主机）将加载输入缓冲器，而读取操作将从输出缓冲器拉取值。因此，对于写入访问，写入的是数据输入缓冲器；对于读取访问，读取的是数据输出缓冲器。在开始操作之前必须向数据输入缓冲器写入值。数据输出缓冲器在操作完成时包含有效数据。因此，任何 128 位数据块都可以分割成多次 32 位字传输；这些字传输可以与外部接口上的其他传输混合进行。

对于正常操作，不使用此寄存器，因为数据的输入和输出通过 DMA 进出 AES 内核。对于主机写入操作，必须使用 128 位输入块写入这些寄存器，以便进行下一次 AES 操作。在此地址范围内以字对齐偏移量写入数据，会将一个字（4 字节）的数据存储到 4 字深（16 字节 = 128 位 AES 块）数据输入缓冲区的相应位置。此缓冲器用于下一次 AES 操作。如果最后一个数据块未完全填充有效数据，则只能写入包含有效数据的字。最后，通过写入 AESCTL.INPUT_RDY 位来触发 AES 操作。

对于主机读取操作该寄存器包含来自最新 AES 操作的 128 位输出块。从该地址范围内的字对齐偏移量读取数据，将从 4 字深（16 字节 = 128 位 AES 块）的数据输出缓冲区中读取一个字（4 字节）的数据。必须先读取这些字

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

（4 个字，一个完整的块），内核才会将下一个块移动到数据输出缓冲区。要清空数据输出缓冲区，必须写入 AESCTL.OUTPUT_RDY 位。

对于带认证的模式（CBC-MAC 和 CCM），无效（消息）字节或字可以写入任何数据。

备注

AES 通常对具有多个输入数据的 128 位块进行操作。CTR、GCM 和 CCM 模式属于例外情况。CTR 模式消息的最后一个块可能包含少于 128 位的数据（请参阅美国国家标准与技术研究院的 NIST 800-38A 文档）：0 < n < = 128 位。对于 GCM 和 CCM，AAD 和消息数据的最后一个块都可能包含少于 128 位的数据（请参阅美国国家标准与技术研究院的 NIST 800-38D）。AES 模块会自动对 CCM 和 CBC-MAC 的未对齐结束数据块进行零填充或掩码处理。对于 CTR 模式，未对齐的数据块中的剩余数据将被忽略。AAD 或仅认证数据不会复制到输出缓冲区，而仅用于认证。

表 12-14. 每种工作模式的输入/输出块格式

运行	数据输入缓冲区	数据输出缓冲区
ECB/CBC 加密	128 位明文块	128 位密文块
ECB/CBC 解密	128 位密文块	128 位明文块
CTR 加密	n 位明文块	n 位密文块
CTR 解密	n 位密文块	n 位明文块
GCM/CCM AAD 数据	n 位明文块	无输出数据
GCM/CCM 加密数据	n 位明文块	n 位密文块
GCM/CCM 解密数据	n 位密文块	n 位明文块
CBC-MAC 数据	n 位明文块	无输出数据

12.5.5.5 TAG 寄存器

表 12-15 展示了用于缓存来自 AES 模块的 TAG 的 TAG 寄存器，可通过 DMA 或主机直接读取进行访问。TAG 寄存器与中间身份验证结果寄存器共享存储空间，但在处理完成之前无法读取。处理过程中，从这些寄存器读取将返回零。如果操作未返回 TAG，则从这些寄存器读取将返回初始化向量 (IV)。如果操作返回一个 TAG 加一个 IV，且两者都必须由主机读取，则主机必须先读取 TAG，然后再读取 IV。以相反顺序读取这些值会返回 IV 两次。

对于主机读取操作，这些寄存器包含 AES 内核最后的 128 位 TAG 输出。在写入下一个上下文之前，TAG 一直可用。仅当 TAG 可用且 AESCTL 寄存器中的 SAVED_CONTEXT_RDY 位被置位时，该寄存器才包含有效数据。在处理过程中或者对于不返回 TAG 的操作/模式，从该寄存器读取将返回 IV 寄存器中的数据。

表 12-15. AES 标签输出寄存器

AESTAGOUT__0 至 AESTAGOUT__3，（只读），32 位地址偏移：0x570 至 0x57C，按 0x4 字节递增

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AES_TAG[31:0] AES_TAG[63:32] AES_TAG[95:64] AES_TAG[127:96]																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 12-16. 适用于 GCM、CCM 和 CBC-MAC

位	字段名称	说明
127-0	TAG	该寄存器包含用于组合模式和仅身份验证模式的身份验证 TAG。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.5.6 键区寄存器

本地密钥存储模块直接连接至 1kB 存储器上。该模块最多可存储 8 个 AES 密钥并有 8 个 128 位条目。密钥大小在密钥存储区模块中进行编程。无法通过 AHB 主器件和从器件接口的读取操作访问密钥存储区中的密钥材料。

密钥只能通过 DMA 写入密钥存储区。启动密钥读取的 DMA 运行后，所有接收到的数据都将写入密钥存储区模块。存储在密钥存储区存储器中的密钥只能传输到 AES 密钥寄存器，无法用于任何其他目的。

12.5.6.1 密钥写入区域寄存器

密钥写入区域寄存器定义必须在密钥存储区 RAM 中写入密钥的位置。写入密钥写入区域寄存器后，密钥库模块就可以通过 DMA 运行接收密钥。如果密钥数据传输触发了密钥存储区中的错误，则在 DMA 完成后，中断状态寄存器 IRQSTAT 中会显示该错误。当编程或所选区域未完全写入时，密钥存储区写入错误 KEY_ST_WR_ERR 将置为有效。当 DMA 运行写入未选择的 RAM 区域时，此错误也会生效。

如需了解更多详情，请参阅节 12.9.1。

12.5.6.2 密钥写入区域寄存器

密钥写入区域寄存器会显示密钥存储区 RAM 的哪些区域包含有效的写入密钥。

当必须在已被有效密钥占用的位置将新密钥写入密钥存储区时，必须首先清除该密钥区域。在将新密钥写入密钥存储区存储器之前，通过写入该寄存器来清除密钥区域。

不允许尝试写入已包含有效密钥的密钥区域，这会导致错误。

如需了解更多详情，请参阅节 12.9.1。

12.5.6.3 密钥大小寄存器

密钥大小寄存器定义通过 DMA 写入的密钥大小。必须在写入 KEYWRITEAREA 寄存器之前配置密钥大小寄存器。

如需了解更多详情，请参阅节 12.9.1。

12.5.6.4 密钥存储区读取区域寄存器

密钥存储区读取区域寄存器选择密钥存储 RAM 区域，必须从该区域读取用于 AES 操作的密钥。写入该寄存器后直接开始操作。操作完成后，IRQSTAT 中断状态寄存器中提供密钥存储区读取操作的状态。当选择的 RAM 区域不包含有效的写入密钥时，密钥存储区读取错误将置为有效。

如需了解更多详情，请参阅节 12.9.1。

12.5.6.5 哈希引擎

SHA-2 模块通过包装器模块连接到寄存器及 DMA 接口；一次只有一个处于活动状态。除 I/O 控制寄存器外，所有哈希模块寄存器都驱动哈希引擎的接口。仅 I/O 控制寄存器具有与之关联的握手逻辑。

12.6 AES 模块性能

12.6.1 简介

AES 模块的处理步骤为计算性能的基础。对于使用 DMA 的加密操作，确定了以下三个主要步骤：

1. 初始化（引擎、DMA 等的设置和初始化）
2. 完整消息的数据处理
3. 最终确定（读取结果、状态检查）

图 12-2 的橙色部分（完全处理）与步骤 1 和步骤 3 相对应。步骤 1 和步骤 3 由主机 CPU 控制，因此取决于主机的性能。步骤 2 对应于图 12-2 中的绿色部分（数据处理），并完全由硬件处理，这不依赖于主机 CPU 的性能。

每个处理命令需要一次完整的处理部分，并且是在处理第一个数据块之前。数据处理块取决于命令要处理的数据量。当操作生成结果摘要或 TAG 时，需要最终确定。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

所需块数由命令所选算法的块大小要求决定。AES 块大小为 128 位。

SHA-256 和 SHA-224 的哈希块大小为 512 位，SHA-512 和 SHA-384 的块大小为 1024 位。

对于较长的数据流，数据处理时间接近理论上的最大吞吐量。对于使用从接口作为 DMA 替代的操作，性能取决于主机 CPU 的性能。

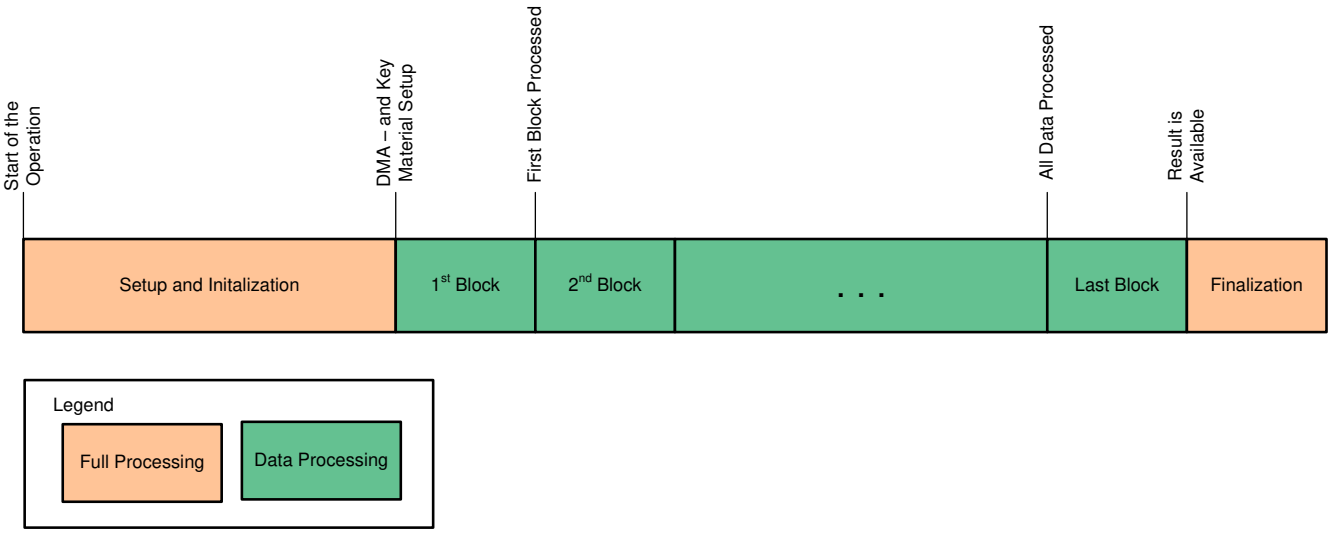


图 12-2. 对称加密处理步骤

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.6.2 基于 DMA 的操作的性能

表 12-17 展示了在 200MHz 下运行的 AES 模块执行基于 DMA 的加密操作的性能。

表 12-17. 基于 DMA 的操作的性能表

200MHz 下的性能（以 Mbps 为单位）				
加密模式	原始引擎性能	1 块数据包性能(1)	20 块性能(1)	100 块性能(1)
AES-128（1 块= 128 位）				
AES-128-ECB	492	111	420	476
AES-128-CBC	483	104	408	466
AES-128-CTR	492	104	415	474
AES-128-GCM(2)	492	77	382	459
AES-192（1 块= 128 位）				
AES-192-ECB	412	107	361	401
AES-192-CBC	406	100	352	394
AES-192-CTR	412	100	357	400
AES-192-GCM(2)	412	73	330	388
AES-256（1 块= 128 位）				
AES-256-ECB	355	102	316	347
AES-256-CBC	350	96	309	341
AES-256-CTR	355	96	313	346
AES-256-GCM(2)	355	63	291	336
哈希（SHA-256：1 块 = 512 位；SHA-512：1 块= 1024 位）				
SHA-256	1575	375	1358	1526
SHA-512	2528	671	2221	2460

(1) 该性能假设对引擎进行了完整编程，加载了密钥，并通过 DMA 从器件设置了 DMA 引擎。如果重用了上下文（模式或密钥），性能会提高。对于各种模式和算法，每个数据包的最大周期开销为 100 到 150 个周期。
(2) AES-GCM 原始性能数据不包括创建 TAG 的最终操作；块性能数据包含此开销。

引擎性能在很大程度上取决于每次操作处理的块数。处理单个块的引擎性能最低；在这种情况下，配置开销最为显著（假设引擎每次操作都完全重新配置）。因此，每次操作处理多个块可带来显著更高的性能。

12.7 编程指南

本节介绍了用于针对支持的用例配置和使用 AES 模块的底层编程序列。

12.7.1 复位后的一次性初始化

初始化的目的是将 AES 模块设置为所有操作通用的初始模式。硬件复位后执行以下初始化步骤：

1. 读出并检查 AES 模块的版本和配置是否与预期的硬件配置匹配。
2. 将 DMABUSCFG 寄存器中的 DMAC 运行时参数编程为对所有 DMA 操作通用的所需值。
3. 初始化所需的中断类型（电平），并启用主器件控制模块中的中断输出信号 RESULT_AVAIL。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.7.2 DMAC 及主控

本部分包含有关如何对 DMAC 进行编程以执行特定操作的一般准则。

12.7.2.1 经常使用

对以下寄存器进行编程以配置 DMA 通道：

- 如果可能，清除所有未处理的中断和错误标志（参见节 12.9.1 中的 IRQCLR）。
- 必须对主控制模块算法选择寄存器进行编程，以允许在所需的内部模块上进行 DMA 运行，从而启用 DMA/AHB 主时钟，并使其保持启用状态，直至主机禁用时钟（请参阅节 12.9.1 中的 ALGSEL）。
- 通道 n 控制寄存器启用了通道位（请参阅节 12.9.1 中的 DMACH0CTL 和 DMACH1CTL）。
- 通道 n 外部地址寄存器（请参阅节 12.9.1 中的 DMACH0EXTADDR 和 DMACH1EXTADDR）。
- 通道 DMA n 长度寄存器。写入该寄存器将启动相应通道上的 DMA 运行（请参阅节 12.9.1 中的 DMACH0LEN 和 DMACH1LEN）。
- 结果可用的中断输出或相应的状态寄存器指示完成操作。处理中断后清除中断（请参阅节 12.9.1 中的 IRQSTAT 和 IRQCLR）。
- 主控制模块算法选择寄存器必须清除至 0 才能关闭 DMA/AHB 主时钟（参见节 12.9.1 中的 ALGSEL）。

备注

如果系统中或在 DMA 运行期间可能发生总线错误的系统中可能发生总线错误（在调试阶段通常有效），则必须检查 IRQSTAT 寄存器以了解可能的错误。

12.7.2.2 中断 DMA 传输

为了停止 DMA 传输并中止操作，主机使用 DMACHnCTL 寄存器禁用通道。当该寄存器的 EN 位设置为 0 时，该通道不会请求新的 DMA 传输，当前活动传输完成。或者，可以通过使用 DMASWRESET 寄存器激活 DMAC 软复位来停止所有活动通道。

备注

停止 DMAC 时，主机必须停止所有活动通道。

必须使用 DMASTAT 寄存器来检查 DMAC 通道的状态。当此寄存器中已禁用通道的 CHx_ACTIVE 位设置为 0 时，DMAC 通道停止。

要结合 AES 引擎停止 DMAC，必须首先将 AES 引擎设置为空闲模式，方法是向长度寄存器写入零，然后禁用 AESCTL 寄存器中的所有模式。

停止 DMAC 通道可能会使主控制模块处于未完成状态，因为来自引擎的未决事件永远不会发生。因此，为了正确恢复引擎，SWRESET 寄存器必须在所有活动的 DMAC 通道停止后发出主控制软复位。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.7.2.3 中断、硬件和软件同步

本部分介绍 RESULT_AVAIL 中断激活和加密内核内 DMAC 数据写入完成之间的重要关系。

当 AHB 主器件完成来自加密内核的数据写入传输并完成内部操作时，RESULT_AVAIL 中断将激活。但是，由于从 AHB 主器件到目标（通常是存储器）的延迟，这并不能确保数据已写入外部存储器。这种延迟可能发生在加密内核之外的 AHB 总线子系统中，因为该系统可能包含桥接器。

备注

如果可能发生此延迟，主机必须确保（使用超时或其他同步机制）只有在所有存储器写入操作完成后才执行外部存储器读取。

12.7.3 哈希

哈希引擎具有以下接口：

- 哈希引擎接受来自两个来源的数据：AHB 从机接口和 DMA。在一次操作中可以组合来自这两个源的数据以完成哈希运算：来自从机接口的写入数据和来自 DMA 接口的写入数据。
- 必须通过寄存器接口对输入摘要（用于恢复哈希）和长度进行编程。
- 结果摘要可通过从机接口或 DMA 读取。

12.7.3.1 数据格式和字节顺序

在大多数系统中，消息数据以小端字节序存储在主机存储器中。哈希引擎设计为小端字节序内核，以防止系统中发生数据交换。因此，消息数据必须以小端字节序提供给内核。

以下代码示例展示了如何根据 FIPS 180-2 和 RFC 1321 规范向哈希引擎提供数据。

备注

以红色突出显示的字节是数据或摘要块的第一个字节。

FIPS 180-2，第 B.2 章：

```
Data in:"abcdbcdecdefdefgefghfghighijhi
jkijkljklmklmnlmnomnopnopq"
"61626364 62636465 63646566 64656667
 65666768 66676869 6768696a 696a6b6c
 68696a6b 6a6b6c6d 6b6c6d6e 6c6d6e6f
 6d6e6f70 6e6f7071
Digest out:"248d6a61 d20638b8 e5c02693 0ce6039
```

外部 RAM 中的哈希数据，通过 DMAC 加载：

```
Word_0 [31:0]6463626 1Word_1 [31:0]65646362
Word_2 [31:0]66656463
...
```

输出摘要，通过从器件接口或 DMA 读取：

```
HASH_DIGEST_A [31:0]:616a8d24HASH_DIGEST_B [31:0]:b83806d2
HASH_DIGEST_C [31:0]:9326c0e5
```


English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.7.3.2 带有来自 DMA 的数据的基本哈希

哈希引擎对通过 DMA 从外部存储器接收到的数据进行哈希处理，并可通过对其进行编程以使用 DMA 运行将结果摘要存储到外部存储器中。使用 EIP-120t 进行操作的典型顺序如下：

- 通过从接口对哈希引擎模式进行编程。
- 对于恢复的哈希操作，通过从器件对上一会话中哈希引擎的中间摘要进行编程。如果恢复的操作是上一个（未完成的）哈希操作的继续，哈希引擎将保持其当前状态。在这种情况下，可以将其重复用于下一个操作，并且不需要对初始摘要进行编程。
- 对主控制器进行编程，以将数据移动到哈希引擎。如果结果摘要必须写入外部存储器，则应相应地对主控制器进行编程。
- 对 DMAC 通道 0 进行编程，以便从外部存储器读取数据并将其复制到哈希引擎的数据输入端口。
- 如果结果摘要必须写入外部存储器，则会对 DMAC 通道 1 进行编程，以将结果摘要存储到外部存储器中的预分配区域。
- 观察中断状态以检查哈希引擎何时完成操作。
- 如果结果摘要必须由主机准备就绪，则可通过从接口读取。如果结果摘要通过 DMA 写入外部存储器，则可在外部存储器中使用它（请参阅节 12.7.2.3）。

12.7.3.2.1 通过从器件读取摘要的新哈希会话

伪代码中的以下软件示例描述了通常由主机软件执行的操作。该示例使用一个新的哈希会话启动哈希引擎，该会话通过 DMA 接口接收输入数据。最后，中间摘要（非最终哈希操作）或最终的哈希摘要（最终哈希操作）通过从接口作为结果摘要被读取。

```
// configure master control module
write CTRL_ALG_SEL 0x0000_0008 // enable DMA path to the SHA-2 engine
write CTRL_INT_CLR 0x0000_0001 // clear any outstanding events
// configure hash engine
write HASH_MODE = 0x0000_0021 // indicate the start of a new hash session and SHA512
write HASH_LENGTH_L// write the length of the message (lo)
write HASH_LENGTH_H// write the length of the message (hi)
// if the final digest is required (pad the input DMA data), write the following register
write HASH_IO_BUF_CTRL = 0x80 // pad the data that is transferred via DMA
// configure DMAC
write DMAC_CH0_CTRL 0x0000_00001 // enable DMA channel 0 for message data
write DMAC_CH0_EXTADDR <ext_memory_address> // base address of the data in ext. memory
write DMAC_CH0_DMALENGTH <length>           // input data in bytes, equal to
                                            // the message length
wait CTRL_INT_STAT[0] = '1'   // wait for operation done (hash and DMAC are ready)
check CTRL_INT_STAT[31] == '0' // check for the absence of errors
// read digest
read HASH_DIGEST_A
...
read HASH_DIGEST_P
// acknowledge result and clear interrupts
write HASH_IO_BUF_CTRL = 0x0000_0001// acknowledge reading of the digest
write CTRL_INT_CLR 0x0000_0001// clear the interrupt
write CTRL_ALG_SEL 0x0000_0000// disable the master control/DMA clock
// end of algorithm
```


English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.7.3.2.2 新建哈希会话并将摘要输出到外部存储器

下面的伪代码示例描述了主机软件通常执行的操作。该示例启动哈希引擎，新建一个哈希会话以通过 DMA 接口接收输入数据。最终，通过 DMA 读取中间摘要（非最终哈希运算结果）或最终的哈希摘要（最终哈希运算结果）作为结果摘要。

```
// configure master control module
write CTRL_ALG_SEL 0x8000_0008 // enable DMA path to the SHA-512 engine + Digest readout
write CTRL_INT_CLR 0x8000_0001 // clear any outstanding events
// configure hash engine
write HASH_MODE = 0x0000_0021 // indicate the start of a new hash session and SHA512
write HASH_LENGTH_L// write the length of the message (lo)
write HASH_LENGTH_H// write the length of the message (hi)
// if the final digest is required (pad the input DMA data), write the following register
write HASH_IO_BUF_CTRL = 0x80 // pad the data that is transferred via DMA
// configure DMAC
write DMAC_CH0_CTRL 0x0000_00001 // enable DMA channel 0 for message data
write DMAC_CH0_EXTADDR <ext_memory_address>// base address of the data in ext. memory
write DMAC_CH0_DMALENGTH <length> // input data in bytes, equal to the message
                                  // length
write DMAC_CH1_CTRL 0x0000_00001 // enable DMA channel 1 for result digest
write DMAC_CH1_EXTADDR <ext_memory_address>// base address of the digest buffer
write DMAC_CH1_DMALENGTH <length> // length of the result digest
// wait for completion and acknowledge the interrupt
wait CTRL_INT_STAT[0] = '1' // wait for operation done (hash and DMAC are ready)
check CTRL_INT_STAT[31] == '0' // check for the absence of errors
write CTRL_INT_CLR 0x0000_0001 // clear the interrupt
write CTRL_ALG_SEL 0x0000_0000 // disable the master control/DMA clock
// the digest can now be ready from the external memory (see note in section 6.2.3)
// end of algorithm
```

12.7.3.2.3 恢复的哈希会话

下面的软件伪代码示例描述了主机软件通常执行的操作。该示例启动哈希引擎，新建一个哈希会话以通过 DMA 接口接收输入数据。最终，通过从器件接口读取中间摘要（非最终哈希运算结果）或最终的哈希摘要（最终哈希运算结果）作为结果摘要。

```
// configure master control module
write CTRL_ALG_SEL 0x0000_0008 // enable DMA path to the SHA-512 engine
write CTRL_INT_CLR 0x0000_0001 // clear any outstanding events
// configure hash engine
write HASH_MODE = 0x0000_0020 // indicate the start of a resumed hash session and SHA512
write HASH_LENGTH_L// write the length of the total message (lo)
write HASH_LENGTH_H// write the length of the total message (hi)
// write the initial digest
write HASH_DIGEST_A
...
write HASH_DIGEST_P
// if the final digest is required (pad the input DMA data), write the following register
write HASH_IO_BUF_CTRL = 0x80 // pad the data that is transferred via DMA
// configure DMAC
write DMAC_CH0_CTRL 0x0000_00001            // enable DMA channel 0
write DMAC_CH0_EXTADDR <ext_memory_address>// base address of the data in ext. memory
write DMAC_CH0_DMALENGTH <length>        // input data in bytes
wait CTRL_INT_STAT[0] = '1'   // wait for operation done (hash and DMAC are ready)
check CTRL_INT_STAT[31] == '0' //check for the absence of errors
// read digest
read HASH_DIGEST_A
...
read HASH_DIGEST_P
// acknowledge result and clear interrupts
write HASH_IO_BUF_CTRL = 0x01 // acknowledge reading of the digest
write CTRL_INT_CLR 0x0000_0001// clear the interrupt
write CTRL_ALG_SEL 0x0000_0000// disable the master control/DMA clock
// end of algorithm
```


English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.7.3.3 HMAC

EIP-120T 根据 RFC2104 中定义的算法，分两步支持 HMAC 操作（不包括可选预处理）。

让：

- H(●) 是一个加密散列函数
- k 是一个密钥，用加零的哈希函数块大小填充到右侧
- m 是要进行验证的消息
- || 表示串联
- ⊕ 表示排他性或者 (XOR)
- opad 是外部填充（0x5c5c5c...5c5c，一块长的十六进制常量）
- ipad 是内部填充（0x363636...3636，一块长的十六进制常量）

然后，HMAC (k,m) 通过 方程式 1 用数学方法定义。

$$HMAC\ (K,m) = H((K \oplus opad) \,||\, H((K \oplus ipad) \,||\, m)) \quad (1)$$

其中：

- H(K ⊕ ipad) 是内摘要。
- H(K ⊕ opad) 是外摘要。
- H(K ⊕ ipad) || m 是中间摘要。

12.7.3.3.1 安全 HMAC

EIP-120T 中的安全 HMAC 操作使用多个基本哈希运算执行，并假设所有安全敏感参数（哈希密钥，中间产品和消息数据）都存储在主机的 DMA 可访问存储器中。

安全 HMAC 操作的实施基于以下要求：

- XOR 型密钥在外部存储器中准备就绪并通过 DMA 读取（或者，可以通过从器件接口写入）。如果哈希密钥长于哈希块大小（SHA-512 和 SHA-384 为 128 字节，SHA-256 和 SHA224 为 64 字节），主机必须使用基本哈希操作来压缩密钥，这可以通过使用基本哈希操作和 EIP-120t 来执行。
- 输入消息位于外部存储器中，通过 DMA 进行读取。
- 中间摘要状态存储在 DMA 可访问的存储器中，随后通过 DA 读回最终哈希值（状态值可选择通过主机接口读取和提供）。
- 结果摘要通过从器件接口读取。

图 12-3 显示了使用 EIP-120t 实现安全 HMAC 所必须执行的步骤。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

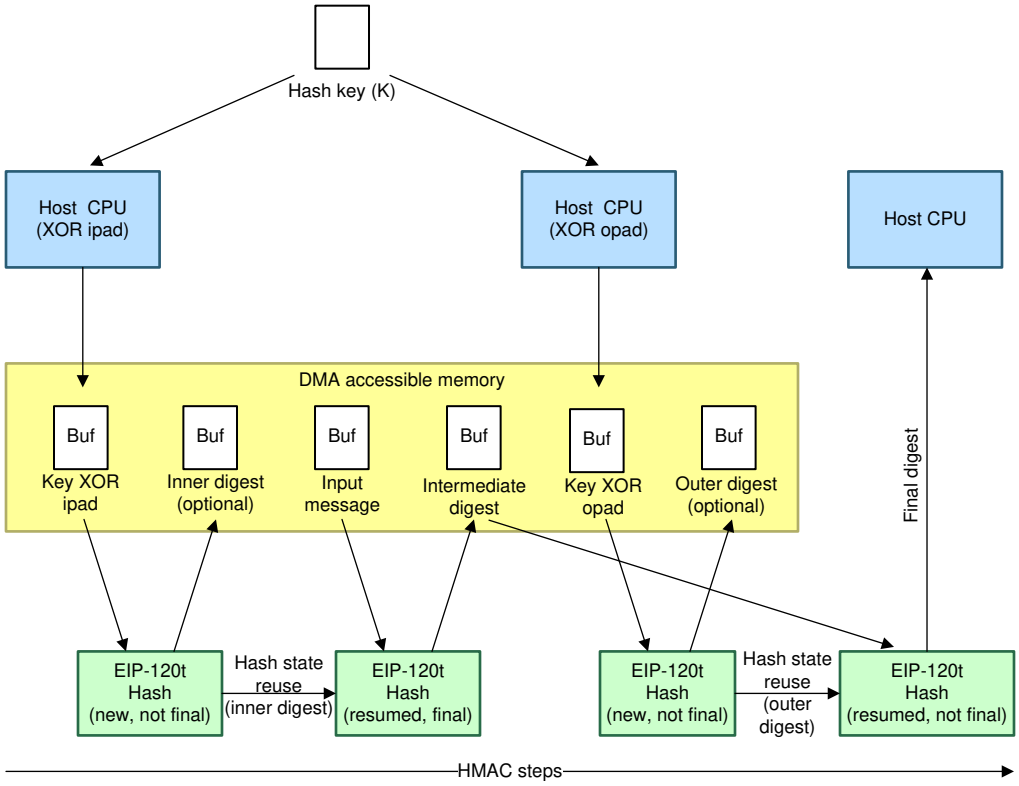


图 12-3. HMAC 步骤

安全 HMAC 对 EIP-120t 使用以下基本哈希运算：

1. 建立了一个新的哈希操作来对填充的密钥 (ipad) 进行哈希处理，该密钥通过 DMA 读取并生成内部摘要。内部摘要保持哈希引擎的内部状态，并用于下一个恢复的哈希。内部摘要可以恢复到外部存储器中的预分配区域，并且可用于使用同一密钥的其他 HMAC 操作。

备注

内部摘要计算需要一个新的哈希操作，该操作尚未完成，因为必须在下一步中恢复。

2. 将建立恢复的哈希操作来对实际消息进行哈希处理。初始摘要在上一操作中生成，在内部状态下仍可用 — 若内部摘要在外部存储器中准备，则主机可以读取该摘要并通过从接口对其进行编程。此操作的结果通过 DMA 存储在预分配的外部存储器中。
3. 建立了一个新的哈希操作来对填充的密钥 (opad) 进行哈希处理，该密钥通过 DMA 读取并生成外部摘要。外部摘要保持哈希引擎的内部状态，并可用于下一个恢复的哈希。外部摘要可以恢复到外部存储器中的预分配区域，并可用于下一个恢复的哈希。外部摘要可以恢复到外部存储器中的预分配区域，以便其可用于使用同一密钥的其他 HMAC 操作。

备注

外部摘要计算需要一个新的哈希运算，该运算未完成，因为必须在下一步中恢复。

4. 已建立恢复的哈希操作，以对步骤 2 的结果进行哈希处理并生成最终的 HMAC 摘要。初始摘要在上一操作中生成，在内部状态下仍可用 — 如果外部摘要在外部存储器中准备，则主机可读取此摘要并通过从接口对其进行编程。最终 HMAC 摘要通过从器件接口读取。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.7.3.4 数据源自从器件接口的备用基本哈希

12.7.3.4.1 新哈希会话

以下伪代码形式的软件示例描述了主机软件通常执行的操作。该示例启动哈希引擎，新建一个哈希会话以通过从机接口接收输入数据。最后，中间摘要（非最终哈希操作）或最终确定的哈希摘要（最终哈希操作）作为结果摘要通过从机接口读取。

```
// disable the DMA path
write CTRL_ALG_SEL 0x00000000
// wait until the input buffer is available for writing by the host
wait HASH_IO_BUF_STAT[2]=='1'
write HASH_MODE = 0x0000_0021 // indicate the start of a new hash session and SHA-512
write HASH_LENGTH_L // the length may be written at any time during session;
write HASH_LENGTH_H // the length must be written when last data has been written
// first block: write and hand-off the block of data
write HASH_DATA_IN_0
...
write HASH_DATA_IN_31
write HASH_IO_BUF_CTRL[6:0]= 0x02 // indicate that a block of data is available
loop: // intermediate blocks// wait until the input buffer available for writing by host
        wait HASH_IO_BUF_STAT[2]=='1'
        // write the intermediate block and hand off the data
        write HASH_DATA_IN_0
        ...
        write HASH_DATA_IN_31
        write HASH_IO_BUF_CTRL[6:0]= 0x02 // indicate that a block of data is available
         if more than one input block, go to loop
endloop
wait HASH_IO_BUF_STAT[2]=='1'
// write the last block and hand-off the data// Note: if last block is misaligned, the last 32-bit
word must be padded (any data is
//       accepted
write HASH_DATA_IN_0
...
write HASH_DATA_IN_31
// if an intermediate digest is required (input data is hash block size aligned)
write HASH_IO_BUF_CTRL[6:0]= 0x42 // indicate that data is available and get the
                                  // intermediate digest (no internal padding)
// else if final digest is required (input data padded by hash engine)
write HASH_IO_BUF_CTRL[6:0]= 0x22 // indicate that data is available and get the final
                                  // digest of the padded data
// wait until output data ready
wait HASH_IO_BUF_STAT[0] == '1'
// read digest
read HASH_DIGEST_A
...
read HASH_DIGEST_P
write HASH_IO_BUF_CTRL = 0x01 // acknowledge that the digest is read
// eng of algorithm
```


English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.7.3.4.2 恢复的哈希会话

以下伪代码形式的软件示例描述了主机软件通常执行的操作。该示例启动哈希引擎，新建一个哈希会话以通过从机接口接收输入数据。最后，中间摘要（非最终哈希操作）或最终确定的哈希摘要（最终哈希操作）作为结果摘要通过从机接口读取。

```
// disable the DMA path
write CTRL_ALG_SEL 0x00000000
// wait until the input buffer is available for writing by the Host
wait HASH_IO_BUF_STAT[2]=='1'
write HASH_MODE = 0x0000_0020 // indicate the start of the resumed hash session and SHA256
//write initial digest
write HASH_DIGEST_A
...
write HASH_DIGEST_P
write HASH_LENGTH_L // the length may be written at any time during session;
write HASH_LENGTH_H // the length must be written when last data has been written
// first block: write and hand-off the block of data
write HASH_DATA_IN_0
...
write HASH_DATA_IN_31
write HASH_IO_BUF_CTRL[6:0]= 0x02 // indicate that a block of data is available
loop: // intermediate blocks// wait until the input buffer available for writing by Host
      wait HASH_IO_BUF_STAT[2]=='1'
      // write the intermediate block and hand off the data
      write HASH_DATA_IN_0
      ...
      write HASH_DATA_IN_31
      write HASH_IO_BUF_CTRL[6:0]= 0x02 // indicate that a block of data is available
      if more than one input block, go to loop
endloop
wait HASH_IO_BUF_STAT[2]=='1'
// write the last block and hand-off the data// Note: if last block is misaligned, the last 32-bit
word must be padded (any data is accepted)
write HASH_DATA_IN_0
...
write HASH_DATA_IN_31
// if an intermediate digest is required (input data is hash block size aligned)
write HASH_IO_BUF_CTRL[6:0]= 0x42 // indicate that data is available and get the
                                  // intermediate digest
//else if final digest is required (input data padded by hash engine)
write HASH_IO_BUF_CTRL[6:0]= 0x22 // indicate that data is available and get the final
                                  // digest of the padded data
// wait until output data ready
wait HASH_IO_BUF_STAT[0] == '1'
// read digest
read HASH_DIGEST_A
...
read HASH_DIGEST_P
write HASH_IO_BUF_CTRL = 0x01 // acknowledge reading of the digest
// eng of algorithm
```


English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.7.4 加密和解密

加密引擎 (AES) 通过以下接口传输数据：

- AES 接受来自两个源的输入数据：AHB 从机接口和 DMA。在一次操作中，可以组合来自这两个源的数据：通过从机接口写入数据，并通过 DMA 写入数据以完成操作。
- 输入 IV 和长度必须通过 AHB 从机接口提供。输出 IV 只能通过从机接口读取。
- 结果数据必须使用与输入数据相同的接口读取：使用从机接口或 DMA。
- 带身份验证操作的结果标签可以使用从机接口或 DMA 读取。

12.7.4.1 数据格式和字节顺序

以下示例显示了如何必须将数据提交到 AES 引擎。由于 AHB 从机接口对数据大多是透明的（不修改数据），因此寄存器接口的对齐方式与以下示例中所述相同。

备注

粗体字节表示密钥或消息的第一个字节。

NIST SP 800-38a，AES-ECB 256 位加密：

```
AES Key In:      603deb10 15ca71be 2b73aef0 857d7781
                 1f352c07 3b6108d7 2d9810a3 0914dff4
AES Data In:     6bc1bee2 2e409f96 e93d7e11 7393172a
AES Data Out:    f3eed1bd b5d2a03c 064b5a7e 3db181f8
```

外部 RAM 中的 AES 密钥，通过密钥存储模块加载：

```
Word_0[31:0]: 10eb3d60
Word_1[31:0]: be71ca15
Word_2[31:0]: f0ae732b
Word_3[31:0]: 81777d85
Word_4[31:0]: 072c351f
Word_5[31:0]: d708613b
Word_6[31:0]: a310982d
Word_7[31:0]: f4df1409
```

使用从机接口或 DMA 输入数据：

```
AESDATAIN0[31:0]: e2bec16b
AESDATAIN1[31:0]: 969f402e
AESDATAIN2[31:0]: 117e3de9
AESDATAIN3[31:0]: 2a179373
```

使用从机接口或 DMA 输出数据：

```
AESDATAOUT0[31:0]: bdd1eef3
AESDATAOUT1[31:0]: 3ca0d2b5
AESDATAOUT2[31:0]: 7e5a4b06
AESDATAOUT3[31:0]: f881b13d
```


Copyright © 2026 Texas Instruments Incorporated

12.7.4.2 密钥存储

在任何加密或解密操作开始之前，密钥存储模块必须至少加载一个密钥，并可供加密操作使用。密钥只能通过 DMA 操作从外部存储器加载。此用途使用 DMAC 通道 0（入站）来实现。

12.7.4.2.1 从外部存储器加载密钥

以下伪代码形式的软件示例描述了主机软件通常为下述用途而执行的操作：将一个或多个密钥加载到密钥存储模块中。

```
// configure master control module
write ALGSEL 0x0000_0001 // enable DMA path to the key store module
write IRQCLR 0x0000_0001 // clear any outstanding events
// configure key store module (area, size)
write KEYSIZE 0x0000_0001 // 128-bit key size
write KEYWRITEAREA 0x0000_0001 // enable keys to write (e.g. Key 0)
// configure DMAC
write DMACH0CTL 0x0000_00001 // enable DMA channel 0
write DMACH0EXTADDR <ext_memory_address> // base address of the key in ext. memory
write DMACH0LEN <length> // total key length in bytes (e.g. 16 for 1 x 128-bit
       // key)
// wait for completion
wait IRQSTAT[0]==’1’ // wait for operation completed
check IRQSTAT[31:30] == ‘00’ // check for absence of errors in DMA and key store
write IRQCLR 0x0000_0001 // acknowledge the interrupt
write ALGSEL 0x0000_0000 // disable master control/DMA clock
// check status
check KEYWRITTENAREA 0x0000_00001 // check that Key 0 was written
// end of algorithm
```


English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.7.4.3 基本 AES 模式

12.7.4.3.1 AES-ECB

对于 AES-ECB 操作，需要以下配置参数：

- 来自密钥存储模块的密钥
- 控制寄存器设置（模式、方向、密钥大小）
- 数据长度

长度字段可以有任何值。如果完成了一个数据流，而下一个数据流使用相同的密钥和控制方式，则只能写入长度字段的新值。长度字段也可以为 0，用于继续处理。

12.7.4.3.2 AES-CBC

对于 AES-CBC 操作，需要以下配置参数：

- 来自密钥存储模块的密钥
- 来自从机接口的 IV
- 控制寄存器设置（模式、方向、密钥大小）
- 数据长度

长度字段可以有任何值。如果完成了一个数据流，而下一个数据流使用相同的密钥和控制方式，则允许仅使用新值写入 IV 和长度字段。长度字段也可以为 0，用于继续处理。

如果主机必须读取结果 IV，则在处理完编程的字节数后，必须将 SAVE_CONTEXT 位设置为 1。

12.7.4.3.3 AES-CTR

对于 AES-CTR 操作，需要以下配置参数：

- 来自密钥存储模块的密钥
- 来自从机接口的 IV，包括初始计数器值（通常为 0x0000 0001）
- 控制寄存器设置（模式、方向、密钥大小）
- 数据长度（可能非块大小对齐）

长度字段可以有任何值。如果完成了一个数据流，而下一个数据流使用相同的密钥和控制方式，则只能写入 IV 和长度字段的新值。长度字段可以为 0，这将导致继续处理。

如果主机必须读取结果 IV，则在处理完编程的字节数后，必须将 save_context 位设置为 1。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.7.4.3.4 使用 DMA 数据的编程顺序

以下伪代码形式的软件示例描述了主机软件通常为下述用途而执行的操作：加密存储在外部存储器中的消息（使用基本 AES 模式）并将加密结果放入外部存储器中的预分配区域。

```
// configure the master control module
write ALGSEL 0x0000_0002 // enable the DMA path to the AES engine
write IRQCLR 0x0000_0001 // clear any outstanding events
// configure the key store to provide pre-loaded AES key
write KEYREADAREA 0x0000_0000 // load the key from ram area 0 (NOTE: The key
              // must be pre-loaded to this area)
wait KEYREADAREA[31]=='0' // wait until the key is loaded to the AES module
check IRQSTAT[29] = '0' // check that the key is loaded without errors
// Write the IV for non-ECB modes// The IV must be written with the same conventions as the data
(refer to
6.4.1 in IP docs)if ((not ECB mode) and (not IV reuse)) then:// write the initialization vector
when a new IV is required
write AESIV_0 ... write AESIV_3
endif// configure AES engine
writeAESCTL = 0b0010_0000_0000_0000_
             0000_0000_0010_1100 // program AES-CBC-128 encryption and save IV
write AESDATALEN0 // write length of the message (lo)
write AESDATALEN1 // write length of the message (hi)
write DMACH0CTL 0x0000_00001 // enable DMA channel 0// configure DMAC
write DMACH0EXTADDR <address> // base address of the input data in ext.memory
write DMACH0LEN <length> // input data length in bytes, equal to the message
         // length (may be non-block size aligned)
write DMACH1CTL 0x0000_00001 // enable DMA channel 1
writeDMACH1EXTADDR <address> // base address of the output data buffer
write DMACH1LEN<length> // output data length in bytes, equal to the result
         // data length (may be non-blocksize aligned)
// wait for completion
wait IRQSTAT[0]=='1' // waitfor operation completed
check IRQSTAT[31] == '0' // check for absence of errors
write AESALGSEL0x0000_0000 // disable master control/DMA clock
if (not ECB mode) then: //only if the IV needs to be re-used/read
wait AESCTL[30]=='1' // wait forSAVED_CONTEXT_RDY bit [30]
read AESIV_0
...
read AESIV_3 // this read clears the SAVED_CONTEXT_RDYflag
endif// end of algorithm
```

12.7.4.4 CBC-MAC

对于 CBC-MAC 操作，需要以下配置参数：

- 来自密钥存储模块的密钥
- IV 必须写入零
- 控制寄存器设置（模式、方向、密钥大小）
- 经过身份验证的数据的长度（可能非块大小对齐）

对于 CBC-MAC 操作，输入数据可能以非对齐方式结束。如果是这种情况，加密内核会在内部对最后一个输入数据块进行填充。

长度字段可以有任何值。如果一个数据流完成，并且下一个数据流使用相同的密钥和控制，则不允许仅写入下一个上下文的一部分。新的数据流必须始终写入完整的上下文。长度字段绝不能写入零。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.7.4.4.1 CBC-MAC 编程顺序

以下伪代码形式的软件示例描述了主机软件通常为下述用途而执行的操作：使用 AES-CBC-MAC 模式对存储在外部存储器中的消息进行身份验证。结果 TAG 通过从机接口读取。

以下顺序处理至少包含 1 个输入数据字节的数据包。

```
// configure the master control module
write ALGSEL 0x0000_0002 // enable the DMA path to the AES engine
write IRQCLR 0x0000_0001 // clear any outstanding events
// configure the key store to provide a pre-loaded AES key
write KEYREADAREA 0x0000_0000 // load the key from ram area 0 (NOTE: The key
              // must be pre-loaded to this area)
wait KEYREADAREA[31]==’0’ // wait until the key is loaded to the AES module
check IRQSTAT[29] = ‘0’// check that the key is loaded without errors
// write the initialization vector
write AESIV_0
...
write AESIV_3
// configure the AES engine
write AESCTL = 0b0010_0000_0000_0000_
              1000_0000_0100_1100 // program AES-CBC-MAC-128 authentication
write AESDATALEN0 // write length of the crypto block (lo)
write AESDATALEN1 // write the length of the crypto block (hi)
              // (may be non-block size aligned)
/write DMACH0CTL 0x0000_00001 // enable DMA channel 0/ configure DMAC
write DMACH0EXTADDR <address> // base address of the input data in ext. memory
write DMACH0LEN <length> // input data length in bytes, equal to the message
              // length len({aad data, pad, crypto_data, pad})
              // (may be non-block size aligned)
// wait for completion
wait IRQSTAT[0]==’1’ // wait for operation completed
check IRQSTAT[31]==‘0’ // check for the absence of errors
write ALGSEL 0x0000_0000 // disable master control/DMA clock
// read tag
wait AESCTL[30]==’1’ // wait for the SAVED_CONTEXT_RDY bit [30]
read AESTAGOUT__0 - AESTAGOUT__3  // this read clears the SAVED_CONTEXT_RDY flag
// end of algorithm
```


English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.7.4.5 AES-CCM

对于 AES-CCM 操作，需要以下配置参数：

- 来自密钥存储模块的密钥
- 必须为 IV 写入加密操作的标志和用于身份验证和加密的 NONCE 字节（请参阅节 12.5.5.2）
- 控制寄存器设置（模式、方向、密钥大小）
- 加密数据的长度（可能非块大小对齐）
- AAD 数据的长度；必须小于 $2^{16} - 2^{8}$ 字节（可以非块大小对齐）

CCM-L 必须为 001、011 或 111，分别表示加密数据长度字段为 2、4 或 8 字节。

CCM-M 可以设置为任何值，但对处理没有影响。主机必须从 128 位 TAG 中选择有效的 TAG 字节。

AAD 和加密数据可能在结尾出现错位。在这种情况下，加密内核用零将两种数据类型填充到 128 位边界。填充方式如下：AAD 和加密数据填充满足位字符串 0n（其中 0 ≤ n ≤ 127），使得包括填充在内的输入数据块长度为 128 位对齐。AAD 数据必须通过单独的 DMA 操作（不能与有效载荷数据合并）或使用从机传输来传输到 AES 引擎。

上下文长度字段可以有任何值。如果完成了一个数据流，而下一个数据流使用相同的密钥和控制方式，则只能写入 IV 和长度字段的新值。用户不能将两个长度字段都写入零。

结果 TAG 通常使用从机接口读取，但也可以使用单独的 DMA 操作写入外部存储器位置。

12.7.4.5.1 AES-CCM 编程顺序

以下伪代码形式的软件示例描述了主机软件通常为下述用途而执行的操作：使用 AES-CCM 模式对存储在外部存储器中的消息（AAD 和有效载荷数据）进行加密和身份验证。加密结果放置在外部存储器中预分配的区域中。结果 TAG 通过从机接口读取。

以下顺序处理至少包含 1 字节 AAD 数据和至少 1 字节加密数据的数据包。

```
// configure the master control module
write ALGSEL 0x0000_0002 // enable the DMA path to the AES engine
write IRQCLR 0x0000_0001 // clear any outstanding events
// configure the key store to provide pre-loaded AES key
write KEYREADAREA 0x0000_0000 // load the key from ram area 0 (NOTE: The key
             // must be pre-loaded to this area)
wait KEYREADAREA[31]=='0' // wait until the key is loaded to the AES module
check IRQSTAT[29] = '0'// check that the key is loaded without errors
// write the initialization vector
write AESIV_0
...
write AESIV_3
// configure the AES engine
write AESCTL = 0b0010_0000_0101_1100_
             0000_0000_0100_1100 // program AES-CCM-128 encryption (M=1, L=3)
write AESDATALEN0// write the length of the crypto block (lo)
write AESDATALEN1// write the length of the crypto block (hi)
             // (may be non-block size aligned)
write AESAUTHLEN // write the length of the AAD data block
             // (may be non-block size aligned)
// configure DMAC to fetch the AAD data
write DMACH0CTL 0x0000_00001 // enable DMA channel 0
write DMACH0EXTADDR <address> // base address of the AAD input data in ext. memory
write DMACH0LEN <length> // AAD data length in bytes, equal to the AAD
             // length len({aad data})
             // (may be non-block size aligned)
// wait for completion of the AAD data transfer
wait IRQSTAT[1]=='1'// wait for DMA_IN_DONE
check IRQSTAT[31]=='0'// check for the absence of errors
// configure DMAC
write DMACH0CTL 0x0000_00001 // enable DMA channel 0
write DMACH0EXTADDR <address> // base address of the payload data in ext. memory
write DMACH0LEN <length> // payload data length in bytes, equal to the message
             // length len({crypto_data})
```


English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

```
write DMACH1CTL 0x0000_00001 // enable DMA channel 1
write DMACH1EXTADDR <address> // base address of the output data buffer
write DMACH1LEN <length> // output data length in bytes, equal to the result
                // data length len({crypto data})
// wait for completion
wait IRQSTAT[0]=='1'// wait for operation completed
check IRQSTAT[31]=='0'// check for the absence of errors
write ALGSEL 0x0000_0000 // disable the master control/DMA clock
// read tag
wait AESCTL[30]=='1' // wait for the SAVED_CONTEXT_RDY bit [30]
read AESTAGOUT 0 -
  AESTAGOUT 3 // this read clears the 'saved_context_ready' flag
// end of algorithm
```

12.7.4.6 AES-GCM

对于 AES-GCM 操作，需要以下配置参数：

- 来自密钥存储模块的密钥
- 来自从机接口的 IV，包括初始计数器值 1
- 控制寄存器设置（模式、方向、密钥大小）
- 密码数据的长度（可能非块大小对齐）
- AAD 数据的长度（可能非块大小对齐）

AES 模块使用的模式是：它基于来自密钥存储模块的密码密钥，在内部计算加密后的 Y0 和 H（哈希密钥）。

AAD 和加密数据可能在结尾出现错位。在这种情况下，模块内部用零将两组数据填充到 128 位边界。填充方式如下：AAD 和加密数据填充满足位字符串 0n（其中 0 ≤ n ≤ 127），使得包括填充在内的输入 AAD 和数据块长度为 128 位对齐。AAD 数据必须通过单独的 DMA 操作（不能与有效载荷数据合并）或使用从机传输来传输到 AES 引擎。

长度字段可以有任何值。如果完成了一个数据流，而下一个数据流使用相同的密钥和控制方式，则只能写入 IV 和长度字段的新值。不允许将两个长度字段都写入零。GCM 操作不能被中断。结果 TAG 通常通过从机接口读取，但也可以通过单独的 DMA 操作写入外部存储器位置。

12.7.4.6.1 AES-GCM 编程顺序

以下伪代码形式的软件示例描述了主机软件通常为下述用途而执行的操作：使用 AES-GCM 模式对消息进行加密和身份验证。消息（AAD 和有效载荷数据）从外部存储器获取，加密结果放置在外部存储器中预分配的区域中。

结果 TAG 通过从机接口读取。以下顺序处理至少包含 1 字节 AAD 数据和至少 1 字节加密数据的数据包。

```
// configure the master control module
write CTRL_ALG_SEL 0x0000_0002 // enable the DMA path to the AES engine
write CTRL_INT_CLR 0x0000_0001 // clear any outstanding events
// configure the key store to provide a pre-loaded AES key
write KEY_STORE_READ_AREA 0x0000_0000 // load the key from ram area 0 (NOTE: The key
// must be pre-loaded to this area)
wait KEY_STORE_READ_AREA[31]=='0' // wait until the key is loaded to the AES module
check CTRL_INT_STAT[29] = '0' // check that the key is loaded without errors
// write the initialization vector
write AES_IV_0
...
write AES_IV_3
// configure the AES engine
write AES_CTRL = 0b0010_0000_0000_0011_
0000_0000_0100_1100 // program AES-GCM-128 encryption (autonomous)
write AES_C_LENGTH_0 // write the length of the crypto block (lo)
write AES_C_LENGTH_1 // write the length of the crypto block (hi)
                     // (may be non-block size aligned)
write AES_AUTH_LENGTH // write the length of the AAD data block
// (may be non-block size aligned)
// configure DMAC to fetch the AAD data
write DMAC_CH0_CTRL 0x0000_00001 // enable DMA channel 0
write DMAC_CH0_EXTADDR <address> // base address of the AAD data in ext. memory
write DMAC_CH0_DMALENGTH <length> // AAD data length in bytes, equal to the aad
```


English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

```
// length len({aad data})
// (may be non-block size aligned)
// wait for completion of the AAD data transfer
wait CTRL_INT_STAT[1]=='1' // wait for DMA_IN_DONE
check CTRL_INT_STAT[31]=='0' // check for the absence of errors
// configure DMAC to process the payload data
write DMAC_CH0_CTRL 0x0000_0001 // enable DMA channel 0
write DMAC_CH0_EXTADDR <address> // base address of the payload data in ext. memory
write DMAC_CH0_DMALENGTH <length> // payload data length in bytes, equal to the payload
// length len({crypto_data})
// (may be non-block size aligned)
write DMAC_CH1_CTRL 0x0000_0001 // enable DMA channel 1
write DMAC_CH1_EXTADDR <address> // base address of the output data buffer
write DMAC_CH1_DMALENGTH <length> // output data length in bytes, equal to the result
// data length len({crypto data})
// (may be non-block size aligned)
```

```
// wait for completion
wait CTRL_INT_STAT[0]=='1' // wait for operation completed
check CTRL_INT_STAT[31]=='0' // check for the absence of errors
write CTRL_ALG_SEL 0x0000_0000 // disable the master control/DMA clock
// read tag
wait AES_CTRL[30]=='1' // wait for the context ready bit [30]
read AES_TAG_OUT_0
...
read AES_TAG_OUT_3 // this read clears the 'saved_context_ready' flag
// end of algorithm
```

12.7.5 异常处理

12.7.5.1 软复位

如果需要，可以使用软复位强制 AES 模块中止当前正在执行的操作并进入 IDLE 状态。

IDLE 状态意味着以下情况：

- DMAC 未主动执行 DMA 操作。
- 加密模块处于 IDLE 状态。
- 密钥存储模块没有加载任何密钥。
- 主控制模块处于 IDLE 状态。
- 软复位必须按以下顺序执行：
  - 如果 DMA 被使用且正在运行，则必须将其停止。
  - 主控制模块必须通过 SWRESET 寄存器进行复位。
- 向加密内核的模式寄存器和长度寄存器写入零。
  这些模式寄存器和长度寄存器包括：
  - AESCTL
  - AESDATALEN0
  - AESDATALEN1
  - AESAUTHLEN

12.7.5.2 外部端口错误

加密内核内部的 AHB 主接口和 DMAC 可检测通过 AHB_ERR 信号接收到的 AHB 端口错误。

在这种情况下，DMAC 会禁用所有通道，以便不再请求新的传输，同时错误信息被捕获到状态寄存器中。DMAPORTERR 寄存器包含发生 AHB 端口错误时活动通道的信息。DMAC 会向主控制模块指示通道完成。恢复过程如下：

1. 使用 DMASWRESET 寄存器对 DMAC 发出软复位，以清除 DMAPORTERR 寄存器并将通道初始化到其默认状态。
2. 对主控制模块发出软复位，以清除其中间状态。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.7.5.3 密钥存储错误

密钥存储错误生成功能是出于调试目的而实现的。在正常或指定操作中，加密内核密钥存储的写入和读取不得触发任何错误。总线错误是唯一可能导致密钥存储写入错误的异常情况。

密钥存储模块会检查密钥是否正确写入密钥存储 RAM。当发生密钥写入错误时，IRQSTAT 寄存器中的 KEY_ST_WR_ERR 标志将置为有效。在这种情况下，密钥不会被存储。主机必须检查 KEY_ST_WR_ERR 标志的状态，并确保相应的 RAM 区域不用于 AES 操作。

如果主机（由于软件故障）尝试从未写入的 RAM 区域使用密钥，密钥存储模块会生成读取错误。在这种情况下，IRQSTAT 寄存器中的 KEY_ST_RD_ERR 标志将置为有效。主机必须检查此标志的状态，并确保不执行 AES 操作的其余所有步骤。

备注

如果发生读取错误，密钥存储会将一个所有字节均设置为 0 的密钥写入 AES 引擎。

12.7.5.3.1 PKA 引擎

PKA 引擎是一个公钥加速 (PKA) 模块，支持中国剩余定理 (CRT)。

PKA 引擎提供以下基本操作：

- 大矢量加法、减法以及组合加法及减法
- 向左或向右移动大矢量
- 大矢量乘法和除法（带或者不带商）
- 大矢量比较和复制

PKA 引擎提供以下复杂操作：

- 大矢量无符号值模块化幂运算
- 使用带预先计算的 Q 逆向量的 CRT 方法的大矢量无符号值模块化幂
- 模逆：给定 A 和 M，计算 B，使得 [(A × B) MOD M] = 1
- 椭圆曲线 $y^2 = x^3 + ax + b \pmod p$ 上的 ECC 点加/倍点，其中：
  - p 是素数。
  - a 和 b 是操作的输入值。

  添加两个相同的点会自动执行倍点操作。
- 椭圆曲线 $y^2 = x^3 + ax + b \pmod p$ 上的 ECC 点乘，其中：
  - p 是素数。
  - a 和 b 是操作的输入值。

  使用 *Montgomery 阶梯* 算法的一个版本以提供侧信道攻击防护。

PKA 模块支持可编程 ROM（PKA 固件是片上固定的）以及用于包含敏感数据的存储器的硬件归零逻辑。硬件归零逻辑独立于 PKA IP 的复位运行，并由 <PRCM.SECDMACLKGR.PKA_ZEROIZATION_RESET_N> 控制。为了电源效率，模块使用仅在需要时才激活的动态控制时钟开关。部分 RAM 使用这些动态控制的时钟进行时钟控制，在启用归零逻辑之前，外部系统必须启用这些时钟。

此外，外部系统应确保在模块活动操作期间不启用归零逻辑。因此，TI 建议在将模块置入硬件复位模式后激活归零逻辑。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.7.5.3.2 功能说明

12.7.5.3.2.1 模块架构

PKA 引擎（请参阅图 12-4）包含以下内部模块：

- PKCP 模块 (EIP-27)：可以执行公钥加密应用中常见的一系列大数（向量）运算。参数和结果都存储在 PKA RAM（PKA 引擎及其主机共享的存储器块）中。
- LNME 模块（可选）：基于可扩展的脉动处理单元阵列的蒙哥马利乘法和幂运算单元。需要访问 PKA RAM（通过单独的端口）和专用的 LNME FIFO RAM。
- 序列发生器模块 (EIP-83)：控制 PKA RAM 中的大数的模幂运算、椭圆曲线密码学和模逆运算。序列发生器模块的主要任务之一是隐藏这些操作中的大多数都是用蒙哥马利形式的数字完成的。该模块需要一个程序 ROM 或 RAM 作为代码存储。
- 寄存器接口：用于控制 PKCP、LNME 和序列发生器模块。

此配置中不存在 LNME 和 LNME FIFO RAM。PKA 引擎是一种具有单个时钟的完全同步设计，并具有单个低电平有效异步复位输入。

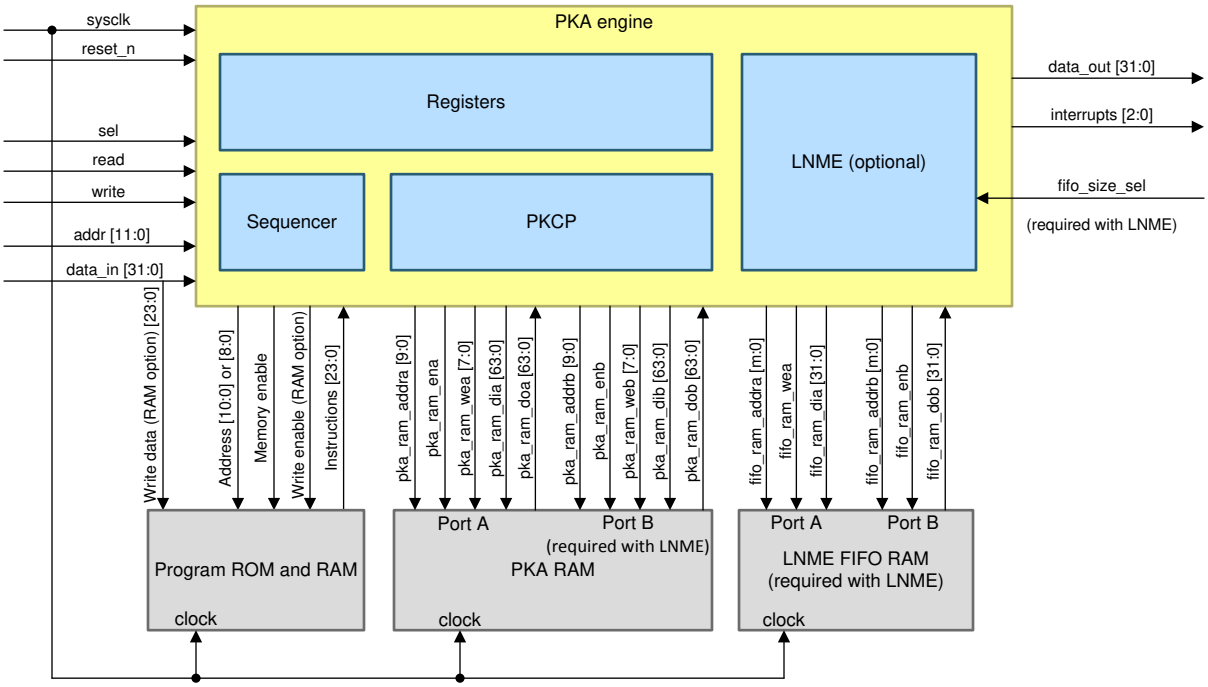


图 12-4. PKA 引擎

12.7.5.3.3 PKA RAM

作为 PKA 操作输入和输出的向量（大数）存储在 PKA RAM 中。每个向量由一系列 32 位字组成，这些字存储在一个连续的存储器块中，LSB 位于该存储器块的最低地址。所有输入和输出向量都必须从偶数的 32 位字地址开始。

根据性能要求和向量长度需求，我们选择了一个 2kB（256 × 64 位）RAM。

备注

PKA RAM 始终可以从主机接口进行 32 位访问（所有输入和输出向量都必须从偶数的 32 位字地址开始；也就是说，它们必须在 PKA RAM 中按 8 字节边界对齐）。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.7.5.3.3.1 PKCP 操作

表 12-18 列出了每个 PKCP 操作的参数和结果。

表 12-18. PKCP 操作 (1)

功能	数学运算	向量 A	向量 B	向量 C	向量 D
乘法	A × B → C	被乘数	乘数	产品	不适用
向	A + B → C	加数	加数	总和	不适用
减法	A – B → C	被减数	减数	差值	不适用
AddSub	A + C – B → C	加数	减数	加数	结果
右移	A >> Shift → C	输入	不适用	结果	不适用
左移	A << Shift → C	输入	不适用	结果	不适用
除法	A mod B → C, A div B → D	被除数	除数	余数	商
取模	A mod B → C	被除数	除数	余数	不适用
比较	A = B, A < B, A > B	Input1	Input2	不适用	不适用
复制	A → C	输入	不适用	结果	不适用

(1) N/A = 不可用

要获得正确的结果，输入向量必须满足中表 12-19 所示的要求。

备注

- PKCP 不检查输入限制。
- *A_Len* 和 *B_Len* 以 32 位字表示向量 A 和 B 的大小。
- Max_{Len} 等于 128 个（32 位）字（即，标准最大向量大小为 4096 位）。

表 12-19. PKCP 操作的输入向量限制

操作限制	
功能	**要求**
乘法	0 < A_Len B_Len ≤ Max_Len
向	0 < A_Len B_Len ≤ Max_Len
减法	0 < A_Len B_Len ≤ Max_Len 结果必须为正 (A ≥ B)。
AddSub	0 < A_Len ≤ Max_Len（B 和 C 操作数的长度为 A_Len，忽略 B_Len） 结果必须为正 [(A + C) ≥ B]。
右移	0 < A_Len ≤ Max_Len
左移	0 < A_Len ≤ Max_Len
除法、取模	1 < B_Len ≤ A_Len ≤ Max_Len B 操作数的最高有效 32 位字不能为 0。
比较	0 < A_Len ≤ Max_Len（B 操作数的长度为 A_Len，忽略 B_Len）
复制	0 < A_Len ≤ Max_Len

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

主机负责在 PKA RAM 中为结果向量分配一个连续内存块。表 12-20 指示应为结果向量分配多少内存。

表 12-20. 结果向量内存分配

功能	结果向量	结果向量长度（以 32 位字为单位）
乘法	C	A_Len + B_Len + 6（应丢弃 6 个暂存字）
向	C	Max(A_Len, B_Len) + 1
减法	C	Max(A_Len, B_Len)
AddSub	D	A_Len + 1
右移	C	A_Len
左移	C	A_Len + 1 （当移位值非零时）
		A_Len （当移位值为零时）
除法	C	余数 → B_Len + 1（应丢弃 1 个暂存字）
	D	商 → A_Len – B_Len + 1
取模	C	余数 → B_Len + 1（应丢弃 1 个暂存字）
比较	无	更新 PKA_COMPARE 寄存器
复制	C	A_Len

操作的输入向量始终允许在内存中重叠（部分或完全）。
表 12-21 列出了操作中输出向量与输入向量重叠的限制。

表 12-21. PKCP 结果向量与向量重叠限制

功能	结果向量	限制
乘法	C	不允许与 A 或 B 向量重叠。
加法、减法	C	如果 C 向量的起始地址不高于其重叠的向量的起始地址，则允许与 A 和（或）B 向量重叠。
AddSub	D	如果 D 向量的起始地址不高于其重叠的向量的起始地址，则允许与 A、B 和（或）C 向量重叠。
右移、左移	C	如果 C 向量的起始地址不高于 A 向量的起始地址，则允许与 A 向量重叠。
除法	C	不允许与 A、B 或 D 向量重叠。
	D	不允许与 A、B 或 C 向量重叠。
取模	C	不允许与 A 或 B 向量重叠。
比较	无	比较操作不写入结果向量。
复制	C	复制的限制与右移和左移相同；即使源和目标重叠，始终允许将向量复制到较低的地址(1)。

(1) *复制*操作可用于通过打破重叠限制来填充内存，但这需要设置两个初始的（32 位）字：

- 要将一个存储器块归零
- 将 A 向量指针设置为块起始地址
- 将 C 向量指针设置为高两个字的位置
- 将 A 向量长度设置为块长度减二（字）
- 用常量零填充块的前两个字，并执行 PKCP 复制操作以将块的其余部分归零。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.7.5.3.3.2 序列发生器操作

12.7.5.3.3.2.1 模幂运算

序列发生器控制模幂运算。表 12-22 总结了模幂 (ExpMod) 运算操作。

表 12-22. ExpMod 操作

功能	数学运算	向量 A	向量 B	向量 C	向量 D
ExpMod-ACT2	$C^A \bmod B \rightarrow D$	指数，长度 = A_Len	模数，长度 = B_Len	底数，长度 = B_Len	结果和工作区
ExpMod-ACT4					
ExpMod-variable					
ExpMod-CRT	—（请参阅以下列表中的计算步骤。）	指数 P，后跟指数 Q（位于下一个最高偶字地址）(1)，两者长度均为 A_Len	模数 P + 缓冲字，后跟模数 Q（位于下一个最高偶字地址）(2)，两者长度均为 B_Len	Q 的逆，长度 = B_Len	输入、结果（长度均为 2 × B_Len）和工作空间

(1) 如果 A_Len 为偶数，指数 Q 紧跟在指数 P 之后；如果 A_Len 为奇数，则指数 Q 和指数 P 之间有一个空字。
(2) 如果 B_Len 为偶数，模数 P 和模数 Q 之间有两个空字；如果 B_Len 为奇数，则模数 Q 和模数 P 之间有一个空（缓冲）字。注意，模数 P 和模数 Q 后面的字可能被*序列发生器*固件清零。

ExpMod-CRT 操作执行以下计算步骤：

1. X ← (输入 mod 模数 P)$^{Exp\ P}$ mod 模数 P
2. Y ← (输入 mod 模数 Q)$^{Exp\ Q}$ mod 模数 Q
3. Z ← ({[(X – Y) mod 模数 P] × Q 的逆} mod 模数 P) × 模数 Q
4. 结果 ← Y + Z

ExpMod-ACT2、ExpMod-ACT4 和 ExpMod-variable 函数实现相同的数学运算，但预计算的*奇数幂*表大小不同。ExpMod-ACT2 函数使用包含两个条目的表，而 ExpMod-ACT4 函数使用包含八个条目的表。ACT4 版本性能更好，但需要更多内存。

ExpMod-variable 和 ExpMod-CRT 允许通过通常用于指定移位操作位数的寄存器，选择可变数量（从 1 到 16）的奇数幂。

对于 PKA 引擎的用户，模幂运算函数似乎是 PKCP 函数集的扩展。输入和结果向量的传递方式与基本 PKCP 操作相同。

表 12-23 列出了 ExpMod 操作的输入向量限制。

表 12-23. 操作限制

功能	要求
ExpMod-ACT2	1. 0 < A_Len ≤ Max_Len
ExpMod-ACT4	2. 1 < B_Len ≤ Max_Len
ExpMod-variable	3. 模数 B 必须为奇数（即 LSB 必须为 1）。 4. 模数 B > 2^{32} 5. 底数 C < 模数 B 6. 矢量 B 和 C 后必须跟一个空的 32 位*缓冲*字。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 12-23. 操作限制（续）

功能	要求
ExpMod-CRT	1. 0 < A_Len ≤ Max_Len 2. 1 < B_Len ≤ Max_Len 3. 模数 P 和模数 Q 必须为奇数（即 LSB 必须为 1）。 4. 模数 P > 模数 Q > 2^{32} (1) 5. 模数 P 和模数 Q 必须为互质（其 GCD 必须为 1）。 6. 0 < 指数 P < (模数 P – 1) 7. 0 < 指数 Q < (模数 Q – 1) 8. (Q 的逆 × 模数 Q) = 1 (mod 模数 P) 9. 输入 < (模数 P × 模数 Q) 10. 模数 P 和模数 Q 后必须跟一个空的 32 位*缓冲*字。

(1) 模数 P 必须大于模数 Q

表 12-24 列出了模幂运算所需的暂存区大小。表中的 M_{Len} 是实际模数长度（对于 ExpMod-CRT 操作，指模数 P 的长度；对于其他操作，指模数 B 的长度），
以 32 位字为单位（即不包含末尾的零字）。如果给定的模量向量的最后一个字为非零，则 M_{Len} 等于 B_{Len}。

表 12-24. 结果向量和暂存区内存分配（起始于 PKA_DPTR）

功能	PKA 引擎类型	暂存区大小（32 位字）(1)
ExpMod-ACT2	使用 *LNME*	3 × [M_Len + 2 – (M_Len MOD 2)] + 10
	仅 *PKCP*	5 × (M_Len + 2)
ExpMod-ACT4	使用 *LNME*	9 × [M_Len + 2 – (M_Len MOD 2)]
	仅 *PKCP*	11 × (M_Len + 2)
ExpMod-variable	使用 *LNME*	max(3 × [M_Len + 2 – (M_Len MOD 2)] + 10, (奇数幂数 + 1) × [M_Len + 2 – (M_Len MOD 2)])
	仅 *PKCP*	(奇数幂数 + 3) × (M_Len + 2)
ExpMod-CRT	使用 *LNME*	max(4 × [M_Len + 2 – (M_Len MOD 2)] + 10, (奇数幂数 + 2) × [M_Len + 2 – (M_Len MOD 2)])
	仅 *PKCP*	(奇数幂数 + 3) × (M_Len + 2) + [M_Len + 2 – (M_Len MOD 2)]

(1) 结果向量长度为 M_Len 或 2 × M_Len 个 32 位字。

表 12-25 列出了结果向量与输入向量的重叠限制。

表 12-25. 结果向量与输入向量的重叠限制

功能	结果向量	限制
ExpMod-ACT2	D	起始于 D 的暂存区不得与任何其他向量重叠，但底数 C 可与结果向量 D 共位以节省空间（即允许 PKA_CPTR = PKA_CPTR）。
ExpMod-ACT4		
ExpMod-variable		
ExpMod-CRT	D	起始于 D 的暂存区不得与任何其他向量重叠；此处也是主输入向量的位置（长度为 2 × B_Len）。

对于模幂运算，PKA RAM 的最小大小取决于最大模数长度和奇数幂的数量。此外，在执行模幂运算期间，序列发生器固件需要固定数量的字节作为暂存区；此暂存区必须位于 PKA RAM 的末尾。

- PKA RAM 的大小应确保最常执行的模幂运算可以使用 4 个奇数幂完成。
- A 和 B 引擎需要 4kB 的 PKA RAM。尽管运行 2kB 的模幂运算是可以接受的，但这些 PKA 引擎很少运行 4kB 的模幂运算，因为这需要大量时间。当最常用的模数长度为 1kB 和 2kB 且不需要快速运行时，2kB 的 PKA RAM 就足够了（2kB RAM 大小无法进行 4kB 操作）。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 12-26 列出了模幂运算所需的 RAM 大小。

表 12-26. 所需 RAM 大小

模数大小（非 CRT）	1 个奇数幂		> 1 个奇数幂(1)
	PKA_CPTR = PKA_DPTR	PKA_CPTR ≠ PKA_DPTR	
1024 位	808 字节	944 字节	+ 每个额外奇数幂 136 字节
2048 位	1576 字节	1840 字节	+ 每个额外奇数幂 256 字节
4096 位	3112 字节	3632 字节	+ 每个额外奇数幂 520 字节
高速缓存	+ 34 字节（固定，位于 PKA RAM 末尾）		
CRT 模数	1 个奇数幂		> 1 个奇数幂(1)
2 × 512 位	696 字节		+ 每个额外奇数幂 72 字节
2 × 1024 位	1336 字节		+ 每个额外奇数幂 136 字节
2 × 2048 位	2616 字节		+ 每个额外奇数幂 264 字节
高速缓存	+ 72 字节（固定，位于 PKA RAM 末尾）		

(1) 在 1 个奇数幂大小的基础上增加。

表 12-27 列出了不同标准 PKA RAM 大小和 PKA 引擎类型下可使用的最大奇数幂数量（非 CRT 操作，使用 PKA_CPTR = PKA_DPTR）。

表 12-27. 最大奇数幂数量

PKA 引擎类型	运行	模数和指数大小	PKA RAM 大小对应的最大奇数幂数量			
			1kB	2kB	4kB	8kB
使用 LNME	非 CRT	1024 位	4	11	16	16
		2048 位	不适用	4	12	16
		4096 位	不适用	不适用	4	12
	CRT	2 × 512 位	6	16	16	16
		2 × 1024 位	不适用	7	16	16
		2 × 2048 位	不适用	不适用	8	16
仅限 PKCP	非 CRT	1024 位	2	9	16	16
		2048 位	不适用	2	10	16
		4096 位	不适用	不适用	2	10
	CRT	2 × 512 位	4	16	16	16
		2 × 1024 位	不适用	5	16	16
		2 × 2048 位	不适用	不适用	6	16

Copyright © 2026 Texas Instruments Incorporated

表 12-28 列出了使用和不使用 CRT 的模幂运算操作的 PKA RAM 向量分配示例。空闲空间起始地址是矢量工作区之后的第一个空闲字节。序列发生器执行暂存区（非 CRT 为 34 字节，使用 CRT 为 72 字节）必须位于空闲空间起始地址与 PKA RAM 末尾之间。

备注

非 CRT 操作使用 PKA_CPTR = PKA_DPTR 以节省空间。

表 12-28. 示例 PKA RAM 向量分配

引擎与操作	（子）向量	起始地址字节偏移	大小（字）	缓冲（字）
使用 LNME (-PExx)，非 CRT (ALENGTH = 0x040 BLENGTH = 0x040 4 个奇数幂)	指数	0x000 (APTR = 0x000)	64	0
	取模	0x100 (BPTR = 0x040)	64	2
	进制	0x208 (CPTR = 0x082)	64	2
	结果	0x208 (DPTR = 0x082)	64	2
	向量工作区	0x208 (= 结果)	5 × (62 + 2 – 0) = 330	0
	空闲空间	0x730 （已用 1840 字节）	–	–
仅 PKCP（-A 和 -B），非 CRT (ALENGTH = 0x040 BLENGTH = 0x040 4 个奇数幂)	指数	0x000 (APTR = 0x000)	64	0
	取模	0x100 (BPTR = 0x040)	64	2
	进制	0x208 (CPTR = 0x082)	64	2
	结果	0x208 (DPTR = 0x082)	64	2
	向量工作区	0x208 (= 结果)	7 × (64 + 2) = 462	0
	空闲空间	0x940 （已用 2368 字节）	–	–
使用 LNME (-PExx)， 使用 CRT (ALENGTH = 0x020 BLENGTH = 0x020 4 个奇数幂)	指数 P	0x000 (APTR = 0x000)	32	0
	指数 Q	0x080	32	0
	模数 P	0x100 (BPTR = 0x040)	32	2
	模数 Q	0x188	32	2
	Q 的逆	0x210 (CPTR = 0x084)	32	0
	输入、结果	0x290 (DPTR = 0x0A4)	64	0
	向量工作区	0x290 (= 结果)	6 × (32 + 2 – 0) = 204	0
	空闲空间	0x5C0 （已用 1472 字节）	–	–
仅 PKCP（-A 和 -B），使用 CRT (ALENGTH = 0x020 BLENGTH = 0x020 4 个奇数幂)	指数 P	0x000 (APTR = 0x000)	32	0
	指数 Q	0x080	32	0
	模数 P	0x100 (BPTR = 0x040)	32	2
	模数 Q	0x188	32	2
	Q 的逆	0x210 (CPTR = 0x084)	32	0
	输入、结果	0x290 (DPTR = 0x0A4)	64	0
	向量工作区	0x290 (= 结果)	7 × (32 + 2) + 32 + 2 – 0 = 272	0
	空闲空间	0x6D0 （已用 1744 字节）	–	–

12.7.5.3.3.2.2 模逆运算

序列发生器控制模逆运算。表 12-29 列出了作为基本 PKCP 功能扩展的功能，但有以下例外：

- 向量 D 指向结果和工作区。
- PKA_SHIFT 寄存器字段用于返回有关运算结果的信息。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 12-29. 基本 PKCP 功能的扩展

功能	数学运算	向量 A	向量 B	向量 C	向量 D
ModInv	$A^{-1} \bmod B \rightarrow D$	待求逆数 长度 = A_Len	模数 长度 = B_Len	未使用	结果和工作区

表 12-30 列出了 PKA_SHIFT 的结果值。

表 12-30. PKA_SHIFT 结果值

功能	结束时 PKA 移位寄存器字段值
ModInv	0 → 成功；向量 D 保存结果 7 → 逆不存在 [GCD (A, B) ≠ 1（即 A 和 B 有公因子）；结果未定义] 31 → 错误；模数为偶数，结果未定义 其他值保留。

表 12-31 和表 12-32 列出了 ModInv 运算对输入和结果向量的限制。

表 12-31. 操作限制

功能	要求
ModInv	0 < A_Len ≤ Max_Len 0 < B_Len ≤ Max_Len 模数 B 必须为奇数（即 LSB 必须为 1）。 模数 B 的值不能为 1（结果未定义，不指示错误）。 模数向量的最高位字（由 B_Len 指示）不能为 0。

表 12-32. 结果向量与输入向量的重叠限制

功能	结果向量	限制
ModInv	D	从 D 开始的暂存区不能与任何其他向量重叠。

表 12-33 列出了 ModInv 运算所需的暂存区大小。

表 12-33. 结果向量和暂存区内存分配 (1)

功能	暂存区大小（以 32 位字为单位），结果向量长度为 B_Len 个 32 位字
ModInv	5 × [M + ε (M)]，其中 M = Max(A_Len, B_Len)

(1) 均从 PKA_DPTR 开始。ε (*n*) = 2 + (*n* MOD 2)；即 2（*n* 为偶数）或 3（*n* 为奇数）。

ModInv 运算要求模数为奇数；这似乎使得该运算在 RSA 密钥生成中无用，因为私钥指数 ***d*** 是根据选定的公钥指数 ***e*** 通过方程式 2 推导出来的：

$$d = \text{ModInv}(e, \Phi) \tag{2}$$

其中：

- $\Phi = (p - 1) \times (q - 1)$
- *p* 和 *q* 均为素数

备注

在方程式 2 中，Φ 为偶数。然而，由于 ***e*** 必须为奇数（否则逆不存在），***d*** 可以通过方程式 3 计算。

$$d = 1 + \{\Phi \times [e - \text{ModInv}(\Phi, e)]\} / e \tag{3}$$

通过四个基本的 PKCP 运算，ModInv 可以在模数为偶数时用于求逆。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

如果模数值是素数，模逆运算可以通过模幂运算，使用模数值减 2 作为指数来执行；这是因为 (在 M 为素数的约束下) ：

- $(A^{M}) \bmod M = A \rightarrow$
- $(A^{M-1}) \bmod M = 1 \rightarrow$
- $(A^{M-2}) \bmod M = A^{-1} \pmod M$

对于包含 LNME 的大型 PKA 引擎，有必要检查此方法是否比直接使用 ModInv 运算更快。此 PKA 引擎支持的 ECC 曲线模数值必须为素数，因此该方法可用于 ECDSA 运算。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.7.5.3.3.2.3 性能

表 12-34 列出了有关执行基本 PKCP 操作所需时钟数的信息。

表 12-34. PKCP 操作所需时钟

操作 （向量长度以位为单位）	32 位 PKCP 的时钟数	16 位 PKCP 的时钟数	调整至 其他向量大小的比例
1K + 1K 加法	99	195	Max (A_Len, B_Len)
1K – 1K 减法	98	194	Max (A_Len, B_Len)
1K × 1K 乘法	1080	4210	A_Len × B_Len
2K/1K 除法或取模	6570(1)	19241(1)	(A_Len – B_Len) × B _Len
1K + 1K – 1K 加或减	131	259	A_Len
左移或右移 1K	69	133	A_Len
1K 复制	64	128	A_Len
1K = 1K 比较	69(2)	133(2)	A_Len

(1) 由于可能需要校正周期，时序略有波动。
(2) 数据相关 — 仅当向量具有相同值或仅在最低有效 16 位或 32 位字上不同时，才需要给出的最大时间。

12.7.5.3.3.2.4 ECC 操作

序列发生器还控制 ECC 操作（有关摘要，请参阅表 12-35）。

表 12-35. ECC 操作

功能	数学运算	向量 A	向量 B	向量 C	向量 D
ECC-ADD	椭圆曲线上的点加/倍点(1)： $y^2 = x^3 + ax + b \pmod p$ pntA + pntC → pntD	pntA.x 后跟(2) pntA.y 长度均为 B_Len（未使用 A_Len）	曲线参数 *p* 后跟(2) *a*（不需要 *b*） 长度均为 B_Len	pntC.x 后跟(2) pntC.y 长度均为 B_Len	结果（即 pntD.x 后跟(2) pntD.y 和工作区）
ECC-MUL	椭圆曲线上的点乘： $y^2 = x^3 + ax + b \pmod p$ *k* × pntC → pntD	标量 *k* 长度为 A_Len	曲线参数 *p* 后跟(2) *a* 和 *b* 长度均为 B_Len	pntC.x 后跟(2) pntC.y 长度均为 B_Len	结果（即 pntD.x 后跟(2) pntD.y 和工作区）

(1) 如果 pntA = pntC，则会自动执行点加倍操作。
(2) 所有输入分量必须位于 64 位边界上，并且必须在其最高有效字之后具有 ε 个额外的缓冲字（每个 32 位）。
ε 必须为 3（B_Len 为奇数）或 2（B_Len 为偶数）。每个结果分量（即 pntD.x、pntD.y）也将后跟 ε 个缓冲（零）字。

对于 PKA 引擎的用户，表 12-35 中的函数似乎是节 12.7.5.3.3.1 中描述的 PKCP 函数集的扩展，但有以下例外：

- 输入向量和结果向量可以是复合向量（即，它们可以由两个或三个相等大小的子向量组成）。
- 向量 D 指向结果和工作区。

PKA_SHIFT 寄存器返回有关操作结果的信息。

表 12-36 列出了 PKA_SHIFT 的结果值。

表 12-36. PKA_SHIFT 结果值

功能	结束时的 PKA_SHIFT 寄存器字段值
ECC-ADD	0 → 成功；向量 D 保存结果点。
ECC-MUL	7 → 结果为*无穷远点*；向量 D 结果点未定义。 31 → 错误（*p* 不是奇数、太短等）；向量 D 结果点未定义。 其他值保留。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 12-37 列出了操作限制。

表 12-37. 操作限制

功能	要求
ECC-ADD	1 < B_Len ≤ 24（最大向量长度为 768 位） 模数 *p* 必须是 > 2^{63} 的素数。 有效模数大小（以位为单位）必须是 32 的倍数。(1) 模数向量的最高位字（由 B_Len 指示）不能为 0。 *a* < *p* 且 *b* < *p* pntA 和 pntC 必须在曲线上（不检查此条件）。 pntA 和 pntC 都不能是*无穷远点*（尽管 ECC-ADD 可以返回此点作为结果）。
ECC-MUL	0 < A_Len ≤ 24（最大向量长度为 768 位） 1 < B_Len ≤ 24（最大向量长度为 768 位） 模数 *p* 必须是 > 2^{63} 的素数。 有效模数大小（以位为单位）必须是 32 的倍数。(1) 模数向量的最高位字（由 B_Len 指示）不能为 0。 *a* < *p* 且 *b* < *p* pntC 必须在曲线上（不检查此条件）。 pntC 不能是*无穷远点*（尽管 ECC-MUL 可以返回此点作为结果）。

(1) 根据引擎类型，一些不符合此规则的模数长度将导致错误结果 — *标准*的 112 位和 521 位模数长度在所有引擎上都能工作，因此这些长度是此规则的例外。

表 12-38 和表 12-39 分别列出了输入向量的重叠限制和暂存区的存储器分配详细信息。

表 12-38. 输入和输出向量的重叠限制

功能	结果向量	限制
ECC-ADD ECC-MUL	D	从 D 开始的暂存区不能与任何其他向量重叠。

表 12-39. 结果向量和暂存区的存储器分配 (1) (2)

功能	暂存区大小 结果向量长度为 2 × (B_Len + ε (B_Len)) 个 32 位字
ECC-ADD	2 × L + 5 × M 其中： • L = B_Len + ε (B_Len) • M = B_Len + 1 + ε (B_Len + 1)
ECC-MUL	18 × L + Max(8, L) 其中： • L = B_Len + ε (B_Len)

(1) 结果向量和暂存区存储器分配都从 PKA_DPTR 开始。
(2) ε (*n*) = 2 + (*n* MOD 2)，即 2（*n* 为偶数）或 3（*n* 为奇数）。

在执行 ECC-ADD 或 ECC-MUL 操作期间，PKA RAM 的最后 72 个字节用作序列发生器程序执行的通用暂存区。在执行期间，这些区域不得与任何输入向量或 D 向量暂存区重叠。

表 12-40 列出了用于 ECC 点乘运算的 PKA RAM 向量分配示例。*空闲空间*起始地址是向量暂存区之后的第一个空闲字节（72 字节的序列发生器执行暂存区必须位于该地址和 PKA RAM 末尾之间）；因此，无法对 2kB PKA RAM 执行 521 位 ECC 点乘。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 12-40. PKA RAM 向量分配

模数长度	（子）向量	起始地址字节偏移	大小（字）	缓冲（字）
192 位 （= 6 个字， ALENGTH = 0x006， BLENGTH = 0x006）	标量 *k*	0x000 (APTR = 0x000)	6	0
	p	0x018 (BPTR = 0x006)	6	2
	a	0x038	6	2
	b	0x058	6	2
	pntC.x（基点）	0x078 (CPTR = 0x01E)	6	2
	pntC.y（基点）	0x098	6	2
	pntD.x（结果）	0x0B8 (DPTR = 0x02E)	6	2
	pntD.y（结果）	0x0D8	6	0
	向量暂存区	0x0B8 (= pntD.x)	(18 × 8) + 8 = 152	0
	空闲空间	0x318 （已用 792 字节）	-	-
384 位 （= 12 个字， ALENGTH = 0x00C， BLENGTH = 0x00C）	标量 *k*	0x000 (APTR = 0x000)	12	0
	p	0x030 (BPTR = 0x00C)	12	2
	a	0x068	12	2
	b	0x0A0	12	2
	pntC.x（基点）	0x0D8 (CPTR = 0x036)	12	2
	pntC.y（基点）	0x110	12	2
	pntD.x（结果）	0x148 (DPTR = 0x052)	12	2
	pntD.y（结果）	0x180	12	0
	向量暂存区	0x148 (= pntD.x)	(18 × 14) + 14 = 266	0
	空闲空间	0x570 （已用 1392 字节）	-	-
521 位 （= 17 个字， ALENGTH = 0x011， BLENGTH = 0x011）	标量 *k*	0x000 (APTR = 0x000)	17	1（对齐 *p*）
	p	0x048 (BPTR = 0x012)	17	3
	a	0x098	17	3
	b	0x0E8	17	3
	pntC.x（基点）	0x138 (CPTR = 0x04E)	17	3
	pntC.y（基点）	0x188	17	3
	pntD.x（结果）	0x1D8 (DPTR = 0x076)	17	3
	pntD.y（结果）	0x228	17	0
	向量暂存区	0x1D8 (= pntD.x)	(18 × 20) + 20 = 380	0
	空闲空间	0x7C8 （已用 1992 字节）	-	-

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.7.5.3.3.2.5 性能

表 12-41 列出了有关执行基本 PKCP 操作所需时钟数的信息。

表 12-41. PKCP 操作所需时钟

运行	32 位 PKCP 的时钟数	16 位 PKCP 的时钟数	调整至其他矢量大小
1K + 1K 加法	99	195	最大值（A_Len、B_Len）
1K – 1K 减法	98	194	最大值（A_Len、B_Len）
1K × 1K 乘法	1080	4210	A_Len × B_Len
2K/1K 除法或取模	6570(1)	19241(1)	(A_Len – B_Len) × B _Len
1K + 1K – 1K 加或减	131	259	A_Len
左移或右移 1K	69	133	A_Len
1K 拷贝	64	128	A_Len
1K = 1K 比较	69(2)	133(2)	A_Len

12.7.5.3.3.2.6 ExpMod 性能

以一个奇数幂作为 100% 的基线性能：

- 两个奇数幂（由 ExpMod-ACT2 使用）提供大约 112% 的性能。
- 四个奇数幂提供大约 121% 的性能。
- 八个奇数幂（由 ExpMod-ACT4 使用）提供大约 125% 的性能。
- 原来的 EIP-23 ACT2 幂运算具有相同的 100% 基线性能，但其时序与指数值无关（对于单个奇数幂操作则不成立）。
- 原来的 EIP-23 ACT5 幂运算提供 120% 的性能。

这些性能数据假设预处理所需的时间可以忽略不计（对于较长的指数长度成立），并且使用 50% 比特位为 1 的随机指数向量。使用超过八个奇数幂不会带来显著的提速，因此不建议这样做。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.7.5.3.3.2.7 模逆性能

模逆运算使用 PKCP 引擎进行计算。性能取决于 PKCP 引擎的类型，并且略微依赖于所使用的数据值（预计有几个百分点的波动）。表 12-42 列出了五次不同模拟的平均性能值。

表 12-42. 五次模拟的性能值

模数和输入的向量长度		32 位 PKCP		16 位 PKCP	
		时钟数	400MHz 下 ops/秒	时钟数	400MHz 下 ops/秒
128	位	26,021	15,372	29,226	13,686
256		60,133	6651	75,483	5299
512		146,406	2372	220,647	1812
1024		426,966	936	719,587	555
2048		1,429,127	279	2,594,516	154
4096		5,167,943	77	9,812,581	40

12.7.5.3.3.2.8 ECC 操作性能

ECC-ADD 仅使用 PKCP 引擎进行计算。性能取决于 PKCP 引擎的类型，并且略微依赖于所使用的数据值（预计有几个百分点的波动）。表 12-43 列出了性能值。

表 12-43. ECC 操作性能值

模数和输入的向量长度			32 位 PKCP		16 位 PKCP	
			时钟数	400MHz 下 ops/秒	时钟数	400MHz 下 ops/秒
128	点加	位	29,731	13,453	33,714	11,855
	倍点		30,623	13,062	34,749	11,511
160	点加		36,689	10,902	43,266	9245
	倍点		38,953	10,268	46,246	8649
192	点加		46,062	8683	56,403	7091
	倍点		47,757	8375	58,229	6869
224	点加		55,276	7236	68,969	5799
	倍点		55,586	7196	70,169	5700
256	点加		65,661	6091	84,140	4753
	倍点		64,179	6232	82,935	4823
320	点加		81,660	4898	110,963	3604
	倍点		84,722	4721	115,581	3460
384	点加		102,271	3911	145,655	2746
	倍点		108,127	3699	154,698	2585
512	点加		154,132	2595	236,331	1692
	倍点		155,896	2565	241,818	1654
521	点加		155,609	2570	243,138	1645
	倍点		163,727	2443	257,409	1553

ECC-MUL 操作并行使用 PKCP 和 LNME 引擎（如果两者都可用）。由于使用了 ECC 阶梯算法的一个版本，性能与标量乘法向量 *k* 无关。由于在算法末尾执行了三次模逆，预计会有轻微波动（小于 2%）（请参阅表 12-44）。

Copyright © 2026 Texas Instruments Incorporated

表 12-44. ECC-MUL 操作性能值

所有输入值的向量长度（位）	性能：时钟数及 400MHz 下 ops/秒	PKA 引擎类型							
		16 位 PKCP EIP-28-A	32 位 PKCP EIP-28-B	16 位 PKCP 和 4 个 PE LNME EIP-28-PE4	16 位 PKCP 和 6 个 PE LNME EIP-28-PE6	16 位 PKCP 和 8 个 PE LNME EIP-28-PE8	16 位 PKCP 和 12 个 PE LNME EIP-28-PE12	16 位 PKCP 和 17 个 PE LNME EIP-28-PE17	16 位 PKCP 和 33 个 PE LNME EIP-28-PE33
128	时钟数	1.36×10^6	593,198	279,360	260,900	261,204	261,147	273,271	262,939
	ops/秒	294	674	1433	1538	1532	1532	1465	1526
160	时钟数	1.90×10^6	805,751	403,005	362,210	362,103	362,052	363,446	359,373
	ops/秒	210	496	992	1104	1104	1104	1101	1114
192	时钟数	3.00×10^6	1.35×10^6	523,642	483,576	471,758	468,428	470,356	474,056
	ops/秒	133	296	764	828	849	854	851	843
224	时钟数	4.67×10^6	1.70×10^6	759,184	600,342	602,421	600,341	602,500	601,374
	ops/秒	85	235	527	666	664	666	664	665
256	时钟数	5.51×10^6	2.06×10^6	932,685	757,287	731,397	733,945	732,391	786,247
	ops/秒	72	194	429	528	547	545	546	508
320	时钟数	10.3×10^6	3.58×10^6	1.52×10^6	1.19×10^6	1.10×10^6	1.08×10^6	1.07×10^6	1.08×10^6
	ops/秒	38	111	263	336	363	370	373	370
384	时钟数	15.5×10^6	5.67×10^6	2.30×10^6	1.84×10^6	1.61×10^6	1.46×10^6	1.46×10^6	1.47×10^6
	ops/秒	25	70	173	217	248	273	273	272
512	时钟数	33.2×10^6	10.6×10^6	4.58×10^6	3.40×10^6	3.00×10^6	2.42×10^6	2.43×10^6	2.41×10^6
	ops/秒	12	37	87	117	133	165	164	165
521	时钟数	35.5×10^6	11.2×10^6	5.02×10^6	3.92×10^6	3.26×10^6	2.61×10^6	2.53×10^6	2.55×10^6
	ops/秒	11	35	79	102	122	153	158	156

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.7.5.3.3.3 序列发生器 ROM 行为和接口

序列发生器程序 ROM 具有高达 2048 位（每字 24 位）的同步只读存储器，其接口如表 12-45 所示（方向以 PKA 引擎为参考；数据总线方向命名以序列器程序 ROM 为参考）。

表 12-45. 序列发生器 ROM 接口信息

信号	方向	功能
rom_me	OUT	高电平有效存储器使能；最大输出延迟为时钟周期的 50%。
rom_addr[10:0]	OUT	地址输出总线；最大输出延迟时间为时钟周期的 50%。
rom_rdata[23:0]	IN	数据输入总线；必须在时钟周期的 70% 之前有效。

12.7.5.3.3.4 寄存器配置

表 12-46 提供了 PKA 引擎（包括 PKA 和 TRNG 引擎）的内部寄存器到外部地址总线的映射。

备注

TRNG 不是 CC13x2 和 CC26x2 器件平台所用的 PKA 引擎的一部分。

可以使用地址总线的 [15:14] 位来选择寄存器空间。

表 12-46. 寄存器地址映射

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
在 PLB 配置中被屏蔽，在 AGB 配置中被跳过																寄存器空间选择		子模块地址（32 位对齐）													
EIP-150 内部寄存器和 AIC 寄存器																															
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0	A	A	A	A	A	A	A	A	A	A	A	A	0	0
PKA 引擎																															
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	1	0	A	A	A	A	A	A	A	A	A	A	A	0	0
PKA RAM（程序 RAM）																															
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	1	1	A	A	A	A	A	A	A	A	A	A	A	0	0
TRNG 引擎																															
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0	0	0	0	0	0	0	A	A	A	A	A	0	0

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

完整的控制寄存器集支持对 PKCP 和序列器控制/状态寄存器的访问。在这些寄存器中，表 12-47 中仅列出 PKCP 控制寄存器、一个序列发生器控制与状态寄存器以及一个硬件和固件版本寄存器。

表 12-47. PKCP 控制、序列发生器和状态、硬件和固件版本寄存器

6 位字偏移	名称	访问	大小（位）	复位值	说明
0x00	PKA_APTR	R/W	11	000h	A 操作数地址偏移量
0x01	PKA_BPTR	R/W	11	000h	B 操作数地址偏移量
0x02	PKA_CPTR	R/W	11	000h	C 操作数和结果地址偏移量
0x03	PKA_DPTR	R/W	11	000h	D 操作数和结果地址偏移量
0x04	PKA_ALENGTH	R/W	9	000h	A 操作数的长度
0x05	PKA_BLENGTH	R/W	9	000h	B 操作数的长度
0x06	PKA_SHIFT	R/W	5	00h	要移位的位
0x07	PKA_FUNCTION	R/W	15 + 1 + 1	0000h 0b 0b	功能代码、运行控制和状态以及停顿结果控制
0x08	PKA_COMPARE	R	3	001b	比较结果
0x09	PKA_MSW	R	11 + 1	000h 1b	MS 非零字地址
0x0A	PKA_DIVMSW	R	11 + 1	000h 1b	余数的 MS 非零字地址（MOD 和 DIV 操作）
0x32	PKA_SEQ_CTRL	R/W	8 + 8 + 1	00h 00h	序列发生器控制、状态和复位
0x3D	PKA_OPTIONS	R	32	-	硬件配置选项
0x3E	PKA_SW_REV	R	16	0000h	固件版本号和功能
0x3F	PKA_REVISION	R	28	-	硬件版本号和 EIP 代码

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.7.5.3.3.5 运行时序

图 12-5、图 12-6 和图 12-7 显示了可用于操作 PKA 引擎的几个操作序列。接口总线事务和主中断输出（中断输出总线的位 [1]）的行为以程式化的方式进行描述。

图 12-5 所示序列的左侧显示了启动行为。中断在模块复位期间处于无效状态（低电平），在启动序列发生器程序后的 10 个模块时钟周期内处于有效状态（高电平）。在使用程序 ROM 的情况下，当释放模块复位时，序列发生器会立即启动。对于配备了程序 RAM 的引擎，必须首先加载序列发生器固件，然后将序列发生器退出复位状态（向 PKA_SEQ_CTRL 寄存器的位 [31] 写入 *0b*），以启动序列发生器程序。

图 12-5 的右侧显示了正常操作序列。正常操作序列首先在 PKA 数据 RAM 中写入输入向量，并将向量指针和长度值写入 PKA 引擎控制寄存器（可以按任意顺序完成）。操作通过对 PKA_FUNCTION 寄存器的写入来启动，这会在设置 RUN 位后的两个时钟周期内使主中断输出变为无效。当 PKA 引擎完成执行请求的操作后，主中断将再次激活，并可以从状态寄存器中读取结果状态（如果需要）。结果向量可以从 PKA 数据 RAM 中读取。

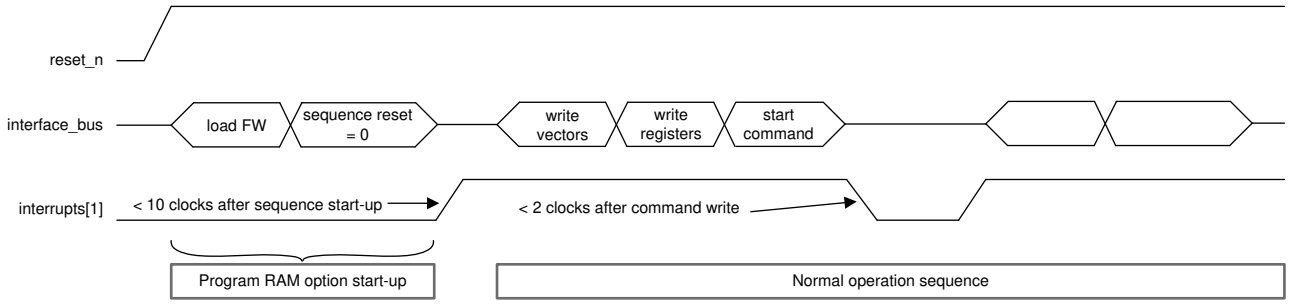


图 12-5. 运行时序

图 12-6 显示了更优化的交错操作序列。当有足够的 PKA 数据 RAM 可用时，可以使用 RAM 中的不同区域来交错写入输入向量和读取结果向量。第一个操作执行期间，写入第二个操作的输入向量。指针和长度寄存器的写入以及第二个命令的实际启动，都在从 PKA 数据 RAM 读取第一个命令的结果向量之前完成。第三个操作的输入向量可以在读取第一个结果后立即写入，所有这些都在第二个操作执行期间完成。

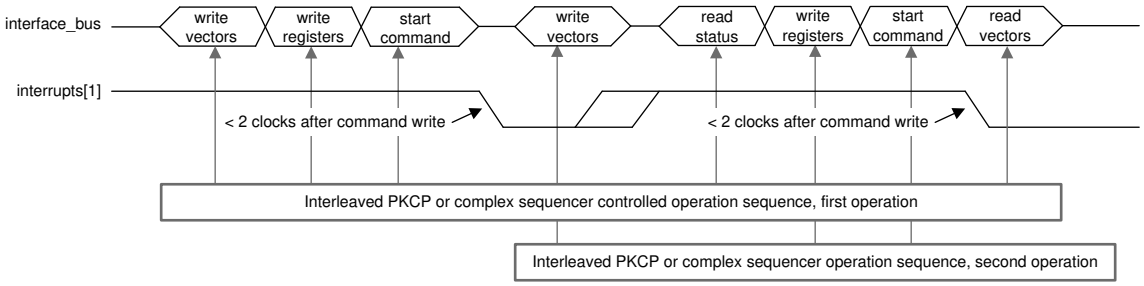


图 12-6. 交错操作序列

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

图 12-7 显示了一个高度优化的基本 PKCP 操作序列。对于基本的 PKCP 操作，向量指针和长度寄存器是双缓冲的；它们可以在操作进行期间被写入。有了这种能力，在第一个操作执行期间，对第二个操作的向量、指针和长度寄存器的写入可以实现更优化的总线访问交错。当中断激活发生时，只需要第二个操作的命令即可启动。

图 12-7 还显示第一个操作的状态是在第二个命令启动之后读取的；为确保此状态不会因第二个操作的提前完成而改变，第二个操作必须在 Stall Result 位（PKA_FUNCTION 寄存器的位 [24]）写入为 1b 的情况下启动。读取状态后，必须复位 Stall Result 位以允许状态更新。如果第一个操作没有需要检查的状态，则不需要设置 Stall Result 位。

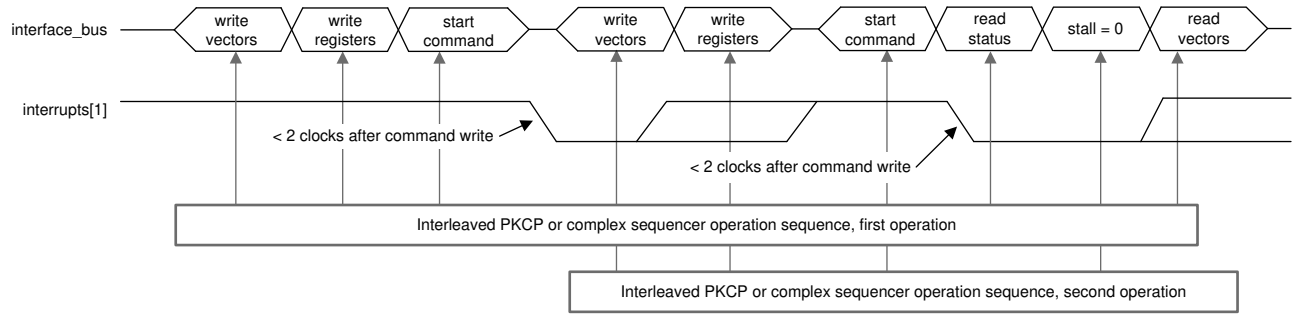


图 12-7. 基本 PKCP 操作序列

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.8 约定与合规性

12.8.1 本手册中使用的约定

表 12-48 列出了本文档中使用的缩写词。

表 12-48. 首字母缩写词

首字母缩写词	全称
ACT2	具有 2 个地址位的加法链式表（4 个条目）
ACT4	具有 4 个地址位的加法链式表（16 个条目）
AES	高级加密标准
AES-CCM	使用 CBC-MAC 的 AES 计数器模式
AHB	高级高速总线
AMBA	高级微控制器总线架构
CBC	密码分组链接
CCM	使用 CBC-MAC 的计数器模式
CM	加密模块
CRT	中国剩余定理
CTR	计数器模式
DMAC	DMA 控制器
DPRAM	双端口随机存取存储器
ECB	电码本
ECC	椭圆曲线加密
EIP	嵌入式知识产权
FIFO	先入先出
FIPS	联邦信息处理标准
GB	千兆字节
Gbit	千兆位
Gbps	千兆位/秒
HMAC	哈希 MAC
HW	硬件
ICM	整数计数器模式
IETF	互联网工程任务组
IP	互联网协议 知识产权
IV	初始化矢量
LNME	大数乘法和幂运算单元
MMM	蒙哥马利模乘
PE	处理单元
PKA	公钥加速器
PKCP	公钥协处理器
RAM	随机存取存储器
ROM	只读存储器
TRNG	真随机数生成器

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.8.1.1 术语

本手册频繁使用某些术语。这些术语指代加密内核用于操作的结构。

外部存储器：一个在外部连接到加密内核 AHB 主端口且只能通过 DMAC 操作进行访问的存储器

从机接口（主机处理器总线）：加密内核的接口，主机处理器通过该接口读取或写入引擎的寄存器。

标签或摘要：两个可互换的术语，指示身份验证操作的结果。术语*摘要*用于常规哈希操作，而*标签*用于身份验证加密操作 (AES-CCM)。

加密上下文：定义加密操作的参数集合：模式、密钥、IV 等

12.8.1.2 公式和命名规则

本文档包含适不同数据类型的公式和命名规则。表 12-49 列出了语法的表示形式。

表 12-49. 公式和命名规则

形式	定义
0x00 或 0h	十六进制值
0b	二进制值
0d	十进制值
0	数字逻辑 0 或低电平
1	数字逻辑 1 或高电平
位	二进制位
8 位	1 字节
16 位	半字
32 位	字
64 位	双字
128 位	四字
MOD	取模
REM	余数
A & B	A 逻辑与 B
A OR B	A 逻辑或 B
NOR	逻辑或非
NOT A	逻辑非
A NOR B	A 逻辑或非 B
AB	A 逻辑异或 B 或 XOR
异或非	逻辑异或非
NAND	逻辑与非
DIV	整数除法
||	连接
[n:m]	寄存器或信号的大小（以位为单位），其中 n > m[2] (1)

(1) 31:0 表示大小为 32 位，最高有效位为 31，LSB 为 0。11:3 表示大小为 9 位，最高有效位为 11，LSB 为 3。

12.8.2 合规性

ECB 和 CBC 模式下的 AES 加密符合 FIPS-197 标准。

12.9 加密寄存器

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1 CRYPTO 寄存器

表 12-50 列出了 CRYPTO 寄存器的存储器映射寄存器。表 12-50 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

表 12-50. CRYPTO 寄存器

偏移	首字母缩写词	寄存器名称	部分
0h	DMACH0CTL	通道 0 控制	DMACH0CTL 寄存器（偏移 = 0h）[复位 = 00000000h]
4h	DMACH0EXTADDR	通道 0 外部地址	DMACH0EXTADDR 寄存器（偏移 = 4h）[复位 = 00000000h]
Ch	DMACH0LEN	通道 0 DMA 长度	DMACH0LEN 寄存器（偏移 = Ch）[复位 = 00000000h]
18h	DMASTAT	DMAC 状态	DMASTAT 寄存器（偏移 = 18h）[复位 = 00000000h]
1Ch	DMASWRESET	DMAC 软件复位	DMASWRESET 寄存器（偏移 = 1Ch）[复位 = 00000000h]
20h	DMACH1CTL	通道 1 控制	DMACH1CTL 寄存器（偏移 = 20h）[复位 = 00000000h]
24h	DMACH1EXTADDR	通道 1 外部地址	DMACH1EXTADDR 寄存器（偏移 = 24h）[复位 = 00000000h]
2Ch	DMACH1LEN	通道 1 DMA 长度	DMACH1LEN 寄存器（偏移 = 2Ch）[复位 = 00000000h]
78h	DMABUSCFG	DMAC 主器件运行时间参数	DMABUSCFG 寄存器（偏移 =78h）[复位 = 00002400h]
7Ch	DMAPORTERR	DMAC 端口错误原始状态	DMAPORTERR 寄存器（偏移 =7Ch）[复位 = 00000000h]
FCh	DMAHWVER	DMAC 版本	DMAHWVER 寄存器（偏移 = FCh）[复位 = 01012ED1h]
400h	KEYWRITEAREA	密钥存储区可写区域	KEYWRITEAREA 寄存器（偏移 =400h）[复位 = 00000000h]
404h	KEYWRITTENAREA	密钥存储区写入区域	KEYWRITTENAREA 寄存器（偏移 = 404h）[复位 = 00000000h]
408h	KEYSIZE	密钥存储区大小	KEYSIZE 寄存器（偏移 = 408h）[复位 = 00000001h]
40Ch	KEYREADAREA	密钥存储区读取区域	KEYREADAREA 寄存器（偏移 = 40Ch）[复位 = 00000008h]
500h + 公式	AESKEY2_y	AES_KEY2_0/AES_GHASH_H_IN_0	AESKEY2_y 寄存器（偏移 = 500h + 公式）[复位 = 00000000h]
510h + 公式	AESKEY3_y	AES_KEY3_0/AES_KEY2_4	AESKEY3_y 寄存器（偏移 = 510h + 公式）[复位 = 00000000h]
540h + 公式	AESIV_y	AES 初始化向量寄存器	AESIV_y 寄存器（偏移 = 540h + 公式）[复位 = 00000000h]
550h	AESCTL	AES 控制	AESCTL 寄存器（偏移 = 550h）[复位 = 80000000h]
554h	AESDATALEN0	AES 加密长度 0 (LSW)	AESDATALEN0 寄存器（偏移 = 554h）[复位 = 00000000h]
558h	AESDATALEN1	AES 加密长度 1 (MSW)	AESDATALEN1 寄存器（偏移 = 558h）[复位 = 00000000h]
55Ch	AESAUTHLEN	AES 身份验证长度	AESAUTHLEN 寄存器（偏移 = 55Ch）[复位 = 00000000h]
560h	AESDATAOUT0	数据输入/输出	AESDATAOUT0 寄存器（偏移 = 560h）[复位 = 00000000h]

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 12-50. CRYPTO 寄存器（续）

偏移	首字母缩写词	寄存器名称	部分
560h	AESDATAIN0	AES 数据 Input_Output 0	AESDATAIN0 寄存器（偏移 = 560h）[复位 = 00000000h]
564h	AESDATAOUT1	数据输入/输出	AESDATAOUT1 寄存器（偏移 = 564h）[复位 = 00000000h]
564h	AESDATAIN1	AES 数据 Input_Output 0	AESDATAIN1 寄存器（偏移 = 564h）[复位 = 00000000h]
568h	AESDATAOUT2	数据输入/输出	AESDATAOUT2 寄存器（偏移 = 568h）[复位 = 00000000h]
568h	AESDATAIN2	AES 数据 Input_Output 2	AESDATAIN2 寄存器（偏移 = 568h）[复位 = 00000000h]
56Ch	AESDATAOUT3	数据输入/输出	AESDATAOUT3 寄存器（偏移 = 56Ch）[复位 = 00000000h]
56Ch	AESDATAIN3	AES 数据 Input_Output 3	AESDATAIN3 寄存器（偏移 = 56Ch）[复位 = 00000000h]
570h + 公式	AESTAGOUT_y	AES 标签输出 0	AESTAGOUT_y 寄存器（偏移 = 570h + 公式）[复位 = 00000000h]
604h	HASHDATAIN1	HASH 数据输入 1	HASHDATAIN1 寄存器（偏移 = 604h）[复位 = 00000000h]
608h	HASHDATAIN2	HASH 数据输入 2	HASHDATAIN2 寄存器（偏移 = 608h）[复位 = 00000000h]
60Ch	HASHDATAIN3	HASH 数据输入 3	HASHDATAIN3 寄存器（偏移 = 60Ch）[复位 = 00000000h]
610h	HASHDATAIN4	HASH 数据输入 4	HASHDATAIN4 寄存器（偏移 = 610h）[复位 = 00000000h]
614h	HASHDATAIN5	HASH 数据输入 5	HASHDATAIN5 寄存器（偏移 = 614h）[复位 = 00000000h]
618h	HASHDATAIN6	HASH 数据输入 6	HASHDATAIN6 寄存器（偏移 = 618h）[复位 = 00000000h]
61Ch	HASHDATAIN7	HASH 数据输入 7	HASHDATAIN7 寄存器（偏移 = 61Ch）[复位 = 00000000h]
620h	HASHDATAIN8	HASH 数据输入 8	HASHDATAIN8 寄存器（偏移 = 620h）[复位 = 00000000h]
624h	HASHDATAIN9	HASH 数据输入 9	HASHDATAIN9 寄存器（偏移 = 624h）[复位 = 00000000h]
628h	HASHDATAIN10	HASH 数据输入 10	HASHDATAIN10 寄存器（偏移 = 628h）[复位 = 00000000h]
62Ch	HASHDATAIN11	HASH 数据输入 11	HASHDATAIN11 寄存器（偏移 = 62Ch）[复位 = 00000000h]
630h	HASHDATAIN12	HASH 数据输入 12	HASHDATAIN12 寄存器（偏移 = 630h）[复位 = 00000000h]
634h	HASHDATAIN13	HASH 数据输入 13	HASHDATAIN13 寄存器（偏移 = 634h）[复位 = 00000000h]
638h	HASHDATAIN14	HASH 数据输入 14	HASHDATAIN14 寄存器（偏移 = 638h）[复位 = 00000000h]
63Ch	HASHDATAIN15	HASH 数据输入 15	HASHDATAIN15 寄存器（偏移 = 63Ch）[复位 = 00000000h]
640h	HASHDATAIN16	HASH 数据输入 16	HASHDATAIN16 寄存器（偏移 = 640h）[复位 = 00000000h]
644h	HASHDATAIN17	HASH 数据输入 17	HASHDATAIN17 寄存器（偏移 = 644h）[复位 = 00000000h]

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 12-50. CRYPTO 寄存器（续）

偏移	首字母缩写词	寄存器名称	部分
648h	HASHDATAIN18	HASH 数据输入 18	HASHDATAIN18 寄存器（偏移 = 648h）[复位 = 00000000h]
64Ch	HASHDATAIN19	HASH 数据输入 19	HASHDATAIN19 寄存器（偏移 = 64Ch）[复位 = 00000000h]
650h	HASHDATAIN20	HASH 数据输入 20	HASHDATAIN20 寄存器（偏移 = 650h）[复位 = 00000000h]
654h	HASHDATAIN21	HASH 数据输入 21	HASHDATAIN21 寄存器（偏移 = 654h）[复位 = 00000000h]
658h	HASHDATAIN22	HASH 数据输入 22	HASHDATAIN22 寄存器（偏移 = 658h）[复位 = 00000000h]
65Ch	HASHDATAIN23	HASH 数据输入 23	HASHDATAIN23 寄存器（偏移 = 65Ch）[复位 = 00000000h]
660h	HASHDATAIN24	HASH 数据输入 24	HASHDATAIN24 寄存器（偏移 = 660h）[复位 = 00000000h]
664h	HASHDATAIN25	HASH 数据输入 25	HASHDATAIN25 寄存器（偏移 = 664h）[复位 = 00000000h]
668h	HASHDATAIN26	HASH 数据输入 26	HASHDATAIN26 寄存器（偏移 = 668h）[复位 = 00000000h]
66Ch	HASHDATAIN27	HASH 数据输入 27	HASHDATAIN27 寄存器（偏移 = 66Ch）[复位 = 00000000h]
670h	HASHDATAIN28	HASH 数据输入 28	HASHDATAIN28 寄存器（偏移 = 670h）[复位 = 00000000h]
674h	HASHDATAIN29	HASH 数据输入 29	HASHDATAIN29 寄存器（偏移 = 674h）[复位 = 00000000h]
678h	HASHDATAIN30	HASH 数据输入 30	HASHDATAIN30 寄存器（偏移 = 678h）[复位 = 00000000h]
67Ch	HASHDATAIN31	HASH 数据输入 31	HASHDATAIN31 寄存器（偏移 = 67Ch）[复位 = 00000000h]
680h	HASHIOBUFCTRL	HASH Input_Output 缓冲器控制	HASHIOBUFCTRL 寄存器（偏移 = 680h）[复位 = 00000004h]
684h	HASHMODE	HASH 模式	HASHMODE 寄存器（偏移 = 684h）[复位 = 00000000h]
688h	HASHINLENL	HASH 输入长度 LSB	HASHINLENL 寄存器（偏移 = 688h）[复位 = 00000000h]
68Ch	HASHINLENH	HASH 输入长度 MSB	HASHINLENH 寄存器（偏移 = 68Ch）[复位 = 00000000h]
6C0h	HASHDIGESTA	HASH 摘要 A	HASHDIGESTA 寄存器（偏移 = 6C0h）[复位 = 00000000h]
6C4h	HASHDIGESTB	HASH 摘要 B	HASHDIGESTB 寄存器（偏移 = 6C4h）[复位 = 00000000h]
6C8h	HASHDIGESTC	HASH 摘要 C	HASHDIGESTC 寄存器（偏移 = 6C8h）[复位 = 00000000h]
6CCh	HASHDIGESTD	HASH 摘要 D	HASHDIGESTD 寄存器（偏移 =6CCh）[复位 = 00000000h]
6D0h	HASHDIGESTE	HASH 摘要 E	HASHDIGESTE 寄存器（偏移 = 6D0h）[复位 = 00000000h]
6D4h	HASHDIGESTF	HASH 摘要 F	HASHDIGESTF 寄存器（偏移 = 6D4h）[复位 = 00000000h]
6D8h	HASHDIGESTG	HASH 摘要 G	HASHDIGESTG 寄存器（偏移 =6D8h）[复位 = 00000000h]

Copyright © 2026 Texas Instruments Incorporated

表 12-50. CRYPTO 寄存器（续）

偏移	首字母缩写词	寄存器名称	部分
6DCh	HASHDIGESTH	HASH 摘要 H	HASHDIGESTH 寄存器（偏移 = 6DCh）[复位 = 00000000h]
6E0h	HASHDIGESTI	HASH 摘要 I	HASHDIGESTI 寄存器（偏移 = 6E0h）[复位 = 00000000h]
6E4h	HASHDIGESTJ	HASH 摘要 J	HASHDIGESTJ 寄存器（偏移 = 6E4h）[复位 = 00000000h]
6E8h	HASHDIGESTK	HASH 摘要 K	HASHDIGESTK 寄存器（偏移 = 6E8h）[复位 = 00000000h]
6ECh	HASHDIGESTL	HASH 摘要 L	HASHDIGESTL 寄存器（偏移 = 6ECh）[复位 = 00000000h]
6F0h	HASHDIGESTM	HASH 摘要 M	HASHDIGESTM 寄存器（偏移 = 6F0h）[复位 = 00000000h]
6F4h	HASHDIGESTN	HASH 摘要 N	HASHDIGESTN 寄存器（偏移 = 6F4h）[复位 = 00000000h]
6F8h	HASHDIGESTO	HASH 摘要 0	HASHDIGESTO 寄存器（偏移 = 6F8h）[复位 = 00000000h]
6FCh	HASHDIGESTP	HASH 摘要 P	HASHDIGESTP 寄存器（偏移 = 6FCh）[复位 = 00000000h]
700h	ALGSEL	算法选择	ALGSEL 寄存器（偏移 = 700h）[复位 = 000000000h]
704h	DMAPROTCTL	DMA Protection Control	DMAPROTCTL 寄存器（偏移 = 704h）[复位 = 00000000h]
740h	SWRESET	软件复位	SWRESET 寄存器（偏移 =740h）[复位 = 00000000h]
780h	IRQTYPE	控制中断配置	IRQTYPE 寄存器（偏移 = 780h）[复位 = 00000000h]
784h	IRQEN	控制中断使能	IRQEN 寄存器（偏移 = 784h）[复位 = 00000000h]
788h	IRQCLR	控制中断清除	IRQCLR 寄存器（偏移 = 788h）[复位 = 00000000h]
78Ch	IRQSET	控制中断设置	IRQSET 寄存器（偏移 = 78Ch）[复位 = 00000000h]
790h	IRQSTAT	控制中断状态	IRQSTAT 寄存器（偏移 = 790h）[复位 = 00000000h]
7FCh	HWVER	硬件版本	HWVER 寄存器（偏移 = 7FCh）[复位 = 92008778h]

复杂的位访问类型经过编码可适应小型表单元。表 12-51 展示了适用于此部分中访问类型的代码。

表 12-51. CRYPTO 访问类型代码

访问类型	代码	说明
读取类型		
R	R	读取
写入类型		
W	W	写入
W1C	W 1C	写入 1 以清零
复位或默认值		
-n		复位后的值或默认值
寄存器数组变量		

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 12-51. CRYPTO 访问类型代码（续）

访问类型	代码	说明
i、j、k、l、m、n		当这些变量用于寄存器名称、偏移或地址时，它们指的是寄存器数组的值，其中寄存器是一组重复寄存器的一部分。寄存器组构成分层结构，数组用公式表示。
y		当该变量用于寄存器名称、偏移或地址时，它指的是寄存器数组的值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.1 DMACH0CTL 寄存器（偏移 = 0h）[复位 = 00000000h]

图 12-8 展示了 DMACH0CTL，表 12-52 对其进行了介绍。

返回到汇总表。

通道 0 控制
此寄存器用于通道启用和优先级选择。当某个通道被禁用时，只有当所有正在进行的请求都完成后，该通道才会变为非活动状态。

图 12-8. DMACH0CTL 寄存器

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

表 12-52. DMACH0CTL 寄存器字段说明

位	字段	类型	复位	说明
31-2	RESERVED	R	0h	保留
1	PRIO	R/W	0h	通道优先级 0：低 1：高 如果两个通道的优先级相同，则使用轮询方案来仲裁对外部端口的通道访问权限。如果一个通道具有高优先级，另一个通道具有低优先级，那么在同时出现同访问请求时，先为高优先级的通道提供服务。
0	EN	R/W	0h	通道使能 0：禁用 1：启用 注意：禁用活动通道会中断 DMA 操作。系统会完成正在进行的块传输，但不会请求新的传输。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.2 DMACH0EXTADDR 寄存器（偏移 = 4h）[复位 = 00000000h]

图 12-9 展示了 DMACH0EXTADDR，表 12-53 对其进行了介绍。

返回到汇总表。

通道 0 外部地址

图 12-9. DMACH0EXTADDR 寄存器

31	30	29	28	27	26	25	24	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																															

表 12-53. DMACH0EXTADDR 寄存器字段说明

位	字段	类型	复位	说明
31-0	ADDR	R/W	0h	通道外部地址值 在运行期间读取时，它会保存发送到主接口后最后更新的外部地址。 注意：加密 DMA 会最多复制 3 个字节，直到遇到字边界为止，因此该地址无需按字对齐。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.3 DMACH0LEN 寄存器（偏移 = Ch）[复位 = 00000000h]

图 12-10 展示了 DMACH0LEN，表 12-54 中对此进行了介绍。

返回到汇总表。

通道 0 DMA 长度

图 12-10. DMACH0LEN 寄存器

31	30	29	28	27	26	25	24	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																DMALEN															
R-0h																R/W-0h															

表 12-54. DMACH0LEN 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	DMALEN	R/W	0h	通道 DMA 长度（以字节为单位） 配置期间，此寄存器包含以字节为单位的 DMA 传输长度。运行期间，它包含发送到主接口后更新的 DMA 传输长度的最后一个值。 注意：如果通道已启用，则将此寄存器设置为非零值即可启动传输。因此，在设置 DMA 通道时，必须最后写入此寄存器。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.4 DMASTAT 寄存器（偏移 = 18h）[复位 = 00000000h]

图 12-11 展示了 DMASTAT，表 12-55 中对此进行了介绍。

返回到汇总表。

DMAC 状态
该寄存器提供每个 DMA 通道的实际状态。如果在数据传输过程中主接口模块收到端口错误，它也会报告这些错误。

图 12-11. DMASTAT 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED						PORT_ERR	RESERVED
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						CH1_ACT	CH0_ACT
R-0h						R-0h	R-0h

表 12-55. DMASTAT 寄存器字段说明

位	字段	类型	复位	说明
31-18	RESERVED	R	0h	保留
17	PORT_ERR	R	0h	反映 AHB 端口上可能存在的传输错误。
16-2	RESERVED	R	0h	保留
1	CH1_ACT	R	0h	值为 1 表示通道 1 处于活动状态（DMA 传输正在进行中）。
0	CH0_ACT	R	0h	值为 1 表示通道 0 处于活动状态（DMA 传输正在进行中）。

Copyright © 2026 Texas Instruments Incorporated

12.9.1.5 DMASWRESET 寄存器（偏移 = 1Ch）[复位 = 00000000h]

图 12-12 展示了 DMASWRESET，表 12-56 中对此进行了介绍。

返回到汇总表。

DMAC 软件复位
软件复位用于复位 DMAC，以停止所有传输并清除端口错误状态寄存器。软件复位后，所有通道将被禁用，并且通道不会发起任何新的请求。DMAC 会等待现有（活动）请求完成，并相应地设置 DMASTAT。

图 12-12. DMASWRESET 寄存器

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

表 12-56. DMASWRESET 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	SWRES	W	0h	软件复位启用 0：禁用 1：启用（自行清零） 必须通过 DMASTAT 检查软件复位是否完成

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.6 DMACH1CTL 寄存器（偏移 = 20h）[复位 = 00000000h]

图 12-13 展示了 DMACH1CTL，表 12-57 对其进行了介绍。

返回到汇总表。

通道 1 控制
此寄存器用于通道启用和优先级选择。当某个通道被禁用时，只有当所有正在进行的请求都完成后，该通道才会变为非活动状态。

图 12-13. DMACH1CTL 寄存器

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

表 12-57. DMACH1CTL 寄存器字段说明

位	字段	类型	复位	说明
31-2	RESERVED	R	0h	保留
1	PRIO	R/W	0h	通道优先级 0：低 1：高 如果两个通道的优先级相同，则使用轮询方案来仲裁对外部端口的通道访问权限。如果一个通道具有高优先级，另一个通道具有低优先级，那么在同时出现同访问请求时，先为高优先级的通道提供服务。
0	EN	R/W	0h	通道使能 0：禁用 1：启用 注意：禁用活动通道会中断 DMA 操作。系统会完成正在进行的块传输，但不会请求新的传输。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.7 DMACH1EXTADDR 寄存器（偏移 = 24h）[复位 = 00000000h]

图 12-14 展示了 DMACH1EXTADDR，表 12-58 对其进行了介绍。

返回到汇总表。

通道 1 外部地址

图 12-14. DMACH1EXTADDR 寄存器

31	30	29	28	27	26	25	24	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																															

表 12-58. DMACH1EXTADDR 寄存器字段说明

位	字段	类型	复位	说明
31-0	ADDR	R/W	0h	通道外部地址值。 在运行期间读取时，它会保存发送到主接口后最后更新的外部地址。 注意：加密 DMA 会最多复制 3 个字节，直到遇到字边界为止，因此该地址无需按字对齐。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.8 DMACH1LEN 寄存器（偏移 = 2Ch）[复位 = 00000000h]

图 12-15 展示了 DMACH1LEN，表 12-59 对其进行了介绍。

返回到汇总表。

通道 1 DMA 长度

图 12-15. DMACH1LEN 寄存器

31	30	29	28	27	26	25	24	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																DMALEN															
R-0h																R/W-0h															

表 12-59. DMACH1LEN 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	DMALEN	R/W	0h	通道 DMA 长度（以字节为单位）。 配置期间，此寄存器包含以字节为单位的 DMA 传输长度。运行期间，它包含发送到主接口后更新的 DMA 传输长度的最后一个值。 注意：如果通道已启用，则将此寄存器设置为非零值即可启动传输。因此，在设置 DMA 通道时，必须最后写入此寄存器。

Copyright © 2026 Texas Instruments Incorporated

12.9.1.9 DMABUSCFG 寄存器（偏移 =78h）[复位 = 00002400h]

图 12-16 展示了 DMABUSCFG，表 12-60 中对此进行了介绍。

返回到汇总表。

DMAC 主器件运行时参数
此寄存器定义 AHB 主器件接口端口的所有运行时参数。如要使 EIP-101m AHB 主器件适配器正常运行，这些参数为必要项。

图 12-16. DMABUSCFG 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
AHB_MST1_BURST_SIZE				AHB_MST1_IDLE_EN	AHB_MST1_INCR_EN	AHB_MST1_LOCK_EN	AHB_MST1_BIGEND
R/W-2h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED							
R-0h							

表 12-60. DMABUSCFG 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-12	AHB_MST1_BURST_SIZE	R/W	2h	可在 AHB 总线上执行的最大突发大小 2h = 4_BYTE：4 字节 3h = 8_BYTE：8 字节 4h = 16_BYTE：16 字节 5h = 32_BYTE：32 字节 6h = 64_BYTE：64 字节
11	AHB_MST1_IDLE_EN	R/W	0h	AHB 上的连续突发传输之间的空闲插入 0h = 不插入空闲传输。 1h = 启用空闲传输插入
10	AHB_MST1_INCR_EN	R/W	1h	AHB 传输的突发长度类型 0h = 未指定长度的突发传输 1h = 固定长度突发传输或单次传输
9	AHB_MST1_LOCK_EN	R/W	0h	AHB 上的锁定转换 0h = 传输未锁定 1h = 传输已锁定
8	AHB_MST1_BIGEND	R/W	0h	AHB 主器件的字节序 0h = 小端字节序 1h = 大端字节序
7-0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.10 DMAPORTERR 寄存器（偏移 =7Ch）[复位 = 00000000h]

图 12-17 展示了 DMAPORTERR，表 12-61 中对此进行了介绍。

返回到汇总表。

DMAC 端口错误原始状态
该寄存器提供单个端口错误的实际状态。它还指示由外部 AHB 端口服务的通道（该通道会在端口错误发生时遭到冻结）。端口错误会中止所有服务通道上的操作（通道启用位强制设为 0），并防止通过该端口进行进一步的传输，直到通过写入 DMASWRESET 寄存器清除错误为止。

图 12-17. DMAPORTERR 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED			PORT1_AHB_ERROR	RESERVED		PORT1_CHANNEL	RESERVED
R-0h			R-0h	R-0h		R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED							
R-0h							

表 12-61. DMAPORTERR 寄存器字段说明

位	字段	类型	复位	说明
31-13	RESERVED	R	0h	保留
12	PORT1_AHB_ERROR	R	0h	值 1 表示 EIP-101 检测到 AHB 总线错误
11-10	RESERVED	R	0h	保留
9	PORT1_CHANNEL	R	0h	指示 AHB 主端口最后服务的具体通道（通道 0 或通道 1）。
8-0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.11 DMAHWVER 寄存器（偏移 = FCh）[复位 = 01012ED1h]

图 12-18 展示了 DMAHWVER，表 12-62 中对此进行了介绍。

返回到汇总表。

DMAC 版本
该寄存器包含此 DMAC 的 EIP 类型指示（或签名），以及硬件版本/补丁编号。

图 12-18. DMAHWVER 寄存器

31	30	29	28	27	26	25	24
RESERVED				HW_MAJOR_VERSION			
R-0h				R-1h			
23	22	21	20	19	18	17	16
HW_MINOR_VERSION				HW_PATCH_LEVEL			
R-0h				R-1h			
15	14	13	12	11	10	9	8
EIP_NUMBER_COMPL							
R-2Eh							
7	6	5	4	3	2	1	0
EIP_NUMBER							
R-D1h							

表 12-62. DMAHWVER 寄存器字段说明

位	字段	类型	复位	说明
31-28	RESERVED	R	0h	保留
27-24	HW_MAJOR_VERSION	R	1h	主要版本号
23-20	HW_MINOR_VERSION	R	0h	次要版本号
19-16	HW_PATCH_LEVEL	R	1h	补丁级别 本版本首次推出时从 0 开始
15-8	EIP_NUMBER_COMPL	R	2Eh	EIP_NUMBER 字段位的逐位补码。
7-0	EIP_NUMBER	R	D1h	此 DMA 控制器 EIP 编号的二进制编码 (209)

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.12 KEYWRITEAREA 寄存器（偏移 =400h）[复位 = 00000000h]

图 12-19 展示了 KEYWRITEAREA，表 12-63 中对此进行了介绍。

返回到汇总表。

密钥存储区可写区域
此寄存器定义应在密钥存储 RAM 中写入密钥的位置。写入该寄存器后，密钥存储区模块即可通过 DMA 操作接收密钥。如果密钥数据传输触发了密钥存储区中的错误，则在 DMA 完成后，您可在中断状态寄存器中查看该错误。当已编程/所选区域未完全写入时，密钥存储区写入错误将被置为有效。如果 DMA 操作写入了未被选择的 RAM 区域，该错误同样会被置为有效。
密钥存储 RAM 被划分为 8 个区域，每个区域为 128 位。
写入密钥存储 RAM 的 192 位密钥应从 256 位边界开始。也就是说，将一段 192 位密钥写入密钥存储 RAM 时，必须通过写入 256 位数据来完成，并且这 256 位中的 64 个最高有效位应置为 0。AES 引擎会忽略这些位。

图 12-19. KEYWRITEAREA 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RAM_AREA7	RAM_AREA6	RAM_AREA5	RAM_AREA4	RAM_AREA3	RAM_AREA2	RAM_AREA1	RAM_AREA0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 12-63. KEYWRITEAREA 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7	RAM_AREA7	R/W	0h	每个 RAM_AREAx 代表一个 128 位的区域。 选择需要写入密钥的密钥存储 RAM 区域。 0：未选择写入 RAM_AREA7。 1：已选择写入 RAM_AREA7。 只有当选定的 RAM 区域是连续的时，才能写入多个 RAM 位置。 需要多个 RAM 位置的密钥（密钥大小为 192 或 256 位）必须从以下区域之一开始：RAM_AREA0、RAM_AREA2、RAM_AREA4 或 RAM_AREA6。 0h = 未选择写入该 RAM 区域 1h = 已选择写入该 RAM 区域
6	RAM_AREA6	R/W	0h	每个 RAM_AREAx 代表一个 128 位的区域。 选择需要写入密钥的密钥存储 RAM 区域。 0：未选择写入 RAM_AREA6. 1：已选择写入 RAM_AREA6。 只有当选定的 RAM 区域是连续的时，才能写入多个 RAM 位置。 需要多个 RAM 位置的密钥（密钥大小为 192 或 256 位）必须从以下区域之一开始：RAM_AREA0、RAM_AREA2、RAM_AREA4 或 RAM_AREA6。 0h = 未选择写入该 RAM 区域 1h = 已选择写入该 RAM 区域

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 12-63. KEYWRITEAREA 寄存器字段说明（续）

位	字段	类型	复位	说明
5	RAM_AREA5	R/W	0h	每个 RAM_AREAx 代表一个 128 位的区域。 选择需要写入密钥的密钥存储 RAM 区域。 0：未选择写入 RAM_AREA5. 1：已选择写入 RAM_AREA5。 只有当选定的 RAM 区域是连续的时，才能写入多个 RAM 位置。 需要多个 RAM 位置的密钥（密钥大小为 192 或 256 位）必须从以下区域之一开始：RAM_AREA0、RAM_AREA2、RAM_AREA4 或 RAM_AREA6。 0h = 未选择写入该 RAM 区域 1h = 已选择写入该 RAM 区域
4	RAM_AREA4	R/W	0h	每个 RAM_AREAx 代表一个 128 位的区域。 选择需要写入密钥的密钥存储 RAM 区域。 0：未选择写入 RAM_AREA4. 1：已选择写入 RAM_AREA4。 只有当选定的 RAM 区域是连续的时，才能写入多个 RAM 位置。 需要多个 RAM 位置的密钥（密钥大小为 192 或 256 位）必须从以下区域之一开始：RAM_AREA0、RAM_AREA2、RAM_AREA4 或 RAM_AREA6。 0h = 未选择写入该 RAM 区域 1h = 已选择写入该 RAM 区域
3	RAM_AREA3	R/W	0h	每个 RAM_AREAx 代表一个 128 位的区域。 选择需要写入密钥的密钥存储 RAM 区域。 0：未选择写入 RAM_AREA3. 1：已选择写入 RAM_AREA3。 只有当选定的 RAM 区域是连续的时，才能写入多个 RAM 位置。 需要多个 RAM 位置的密钥（密钥大小为 192 或 256 位）必须从以下区域之一开始：RAM_AREA0、RAM_AREA2、RAM_AREA4 或 RAM_AREA6。 0h = 未选择写入该 RAM 区域 1h = 已选择写入该 RAM 区域
2	RAM_AREA2	R/W	0h	每个 RAM_AREAx 代表一个 128 位的区域。 选择需要写入密钥的密钥存储 RAM 区域。 0：未选择写入 RAM_AREA2. 1：已选择写入 RAM_AREA2。 只有当选定的 RAM 区域是连续的时，才能写入多个 RAM 位置。 需要多个 RAM 位置的密钥（密钥大小为 192 或 256 位）必须从以下区域之一开始：RAM_AREA0、RAM_AREA2、RAM_AREA4 或 RAM_AREA6。 0h = 未选择写入该 RAM 区域 1h = 已选择写入该 RAM 区域
1	RAM_AREA1	R/W	0h	每个 RAM_AREAx 代表一个 128 位的区域。 选择需要写入密钥的密钥存储 RAM 区域。 0：未选择写入 RAM_AREA1. 1：已选择写入 RAM_AREA1。 只有当选定的 RAM 区域是连续的时，才能写入多个 RAM 位置。 需要多个 RAM 位置的密钥（密钥大小为 192 或 256 位）必须从以下区域之一开始：RAM_AREA0、RAM_AREA2、RAM_AREA4 或 RAM_AREA6。 0h = 未选择写入该 RAM 区域 1h = 已选择写入该 RAM 区域

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 12-63. KEYWRITEAREA 寄存器字段说明（续）

位	字段	类型	复位	说明
0	RAM_AREA0	R/W	0h	每个 RAM_AREAx 代表一个 128 位的区域。 选择需要写入密钥的密钥存储 RAM 区域。 0：未选择写入 RAM_AREA0. 1：已选择写入 RAM_AREA0。 只有当选定的 RAM 区域是连续的时，才能写入多个 RAM 位置。 需要多个 RAM 位置的密钥（密钥大小为 192 或 256 位）必须从以下区域之一开始：RAM_AREA0、RAM_AREA2、RAM_AREA4 或 RAM_AREA6。 0h = 未选择写入该 RAM 区域 1h = 已选择写入该 RAM 区域

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.13 KEYWRITTENAREA 寄存器（偏移 = 404h）[复位 = 00000000h]

图 12-20 展示了 KEYWRITTENAREA，表 12-64 中对此进行了介绍。

返回到汇总表。

密钥存储区写入区域
此寄存器指示密钥存储 RAM 中哪些区域包含已写入的有效密钥。
当需要将新密钥写入密钥存储区时，如果该位置已被有效密钥占用，则必须先清除该密钥区域。这可以通过在将新密钥写入密钥存储内存之前写入此寄存器来实现。
不得尝试写入已包含有效密钥的密钥区域，这一做法会导致错误。

图 12-20. KEYWRITTENAREA 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RAM_AREA_WRITTEN7	RAM_AREA_WRITTEN6	RAM_AREA_WRITTEN5	RAM_AREA_WRITTEN4	RAM_AREA_WRITTEN3	RAM_AREA_WRITTEN2	RAM_AREA_WRITTEN1	RAM_AREA_WRITTEN0
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h

表 12-64. KEYWRITTENAREA 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7	RAM_AREA_WRITTEN7	R/W1C	0h	读取时，该位返回密钥区域写入状态。 可通过写入 1 来复位该位。 注意：通过写入 DMASWRESET.SWRES 启动软复位时，该寄存器将复位。软复位后，必须重新将所有密钥写入密钥存储区。 0h = 此 RAM 区域未写入有效密钥信息 1h = 此 RAM 区域已写入有效密钥信息
6	RAM_AREA_WRITTEN6	R/W1C	0h	读取时，该位返回密钥区域写入状态。 可通过写入 1 来复位该位。 注意：通过写入 DMASWRESET.SWRES 启动软复位时，该寄存器将复位。软复位后，必须重新将所有密钥写入密钥存储区。 0h = 此 RAM 区域未写入有效密钥信息 1h = 此 RAM 区域已写入有效密钥信息
5	RAM_AREA_WRITTEN5	R/W1C	0h	读取时，该位返回密钥区域写入状态。 可通过写入 1 来复位该位。 注意：通过写入 DMASWRESET.SWRES 启动软复位时，该寄存器将复位。软复位后，必须重新将所有密钥写入密钥存储区。 0h = 此 RAM 区域未写入有效密钥信息 1h = 此 RAM 区域已写入有效密钥信息
4	RAM_AREA_WRITTEN4	R/W1C	0h	读取时，该位返回密钥区域写入状态。 可通过写入 1 来复位该位。 注意：通过写入 DMASWRESET.SWRES 启动软复位时，该寄存器将复位。软复位后，必须重新将所有密钥写入密钥存储区。 0h = 此 RAM 区域未写入有效密钥信息 1h = 此 RAM 区域已写入有效密钥信息

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 12-64. KEYWRITTENAREA 寄存器字段说明（续）

位	字段	类型	复位	说明
3	RAM_AREA_WRITTEN3	R/W1C	0h	读取时，该位返回密钥区域写入状态。 可通过写入 1 来复位该位。 注意：通过写入 DMASWRESET.SWRES 启动软复位时，该寄存器将复位。软复位后，必须重新将所有密钥写入密钥存储区。 0h = 此 RAM 区域未写入有效密钥信息 1h = 此 RAM 区域已写入有效密钥信息
2	RAM_AREA_WRITTEN2	R/W1C	0h	读取时，该位返回密钥区域写入状态。 可通过写入 1 来复位该位。 注意：通过写入 DMASWRESET.SWRES 启动软复位时，该寄存器将复位。软复位后，必须重新将所有密钥写入密钥存储区。 0h = 此 RAM 区域未写入有效密钥信息 1h = 此 RAM 区域已写入有效密钥信息
1	RAM_AREA_WRITTEN1	R/W1C	0h	读取时，该位返回密钥区域写入状态。 可通过写入 1 来复位该位。 注意：通过写入 DMASWRESET.SWRES 启动软复位时，该寄存器将复位。软复位后，必须重新将所有密钥写入密钥存储区。 0h = 此 RAM 区域未写入有效密钥信息 1h = 此 RAM 区域已写入有效密钥信息
0	RAM_AREA_WRITTEN0	R/W1C	0h	读取时，该位返回密钥区域写入状态。 可通过写入 1 来复位该位。 注意：通过写入 DMASWRESET.SWRES 启动软复位时，该寄存器将复位。软复位后，必须重新将所有密钥写入密钥存储区。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.14 KEYSIZE 寄存器（偏移 = 408h）[复位 = 00000001h]

图 12-21 展示了 KEYSIZE，表 12-65 中对此进行了介绍。

返回到汇总表。

密钥存储区大小
该寄存器定义通过 DMA 写入的密钥的大小。应在写入 KEY_STORE_WRITE_AREA 寄存器之前配置此寄存器。

图 12-21. KEYSIZE 寄存器

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														尺寸	
R-0h														R/W-1h	

表 12-65. KEYSIZE 寄存器字段说明

位	字段	类型	复位	说明
31-2	RESERVED	R	0h	保留
1-0	尺寸	R/W	1h	密钥大小： 00：保留 在将其写入该寄存器时，KEY_STORE_WRITTEN_AREA 寄存器会复位。 1h = 128_BIT：128 位 2h = 192_BIT：192 位 3h = 256_BIT：256 位

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.15 KEYREADAREA 寄存器（偏移 = 40Ch）[复位 = 00000008h]

图 12-22 展示了 KEYREADAREA，表 12-66 中对此进行了介绍。

返回到汇总表。

密钥存储区读取区域
该寄存器用于选择密钥存储 RAM 区域，从中读取将用于 AES 操作的密钥。写入该寄存器后，操作将立即启动。操作完成后，可以在中断状态寄存器中查看密钥存储区读取操作的状态。当选择的 RAM 区域不包含有效写入密钥时，密钥存储区读取错误将置为有效。

图 12-22. KEYREADAREA 寄存器

31	30	29	28	27	26	25	24
BUSY	RESERVED						
R-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				RAM_AREA			
R-0h				R/W-8h			

表 12-66. KEYREADAREA 寄存器字段说明

位	字段	类型	复位	说明
31	BUSY	R	0h	密钥存储区操作繁忙状态标志（只读）： 0：操作完成。 1：操作未完成，密钥存储区繁忙。
30-4	RESERVED	R	0h	保留
3-0	RAM_AREA	R/W	8h	选择密钥存储 RAM 中需要读取的密钥区域，该密钥将被写入 AES 引擎 RAM_AREA： RAM 区域 RAM_AREA0、RAM_AREA2、RAM_AREA4 和 RAM_AREA6 是 192 位和 256 位密钥长度的唯一有效读取区域。 仅可选择包含有效写入密钥的 RAM 区域。 0h = RAM 区域 0 1h = RAM 区域 1 2h = RAM 区域 2 3h = RAM 区域 3 4h = RAM 区域 4 5h = RAM 区域 5 6h = RAM 区域 6 7h = RAM 区域 7 8h = 无 RAM

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.16 AESKEY2_y 寄存器（偏移 = 500h + 公式）[复位 = 00000000h]

图 12-23 展示了 AESKEY2_y，表 12-67 中对此进行了介绍。

返回到汇总表。

AES_KEY2_0 / AES_GHASH_H_IN_0
第二密钥 / GHASH 密钥（内部，但可清除）
以下寄存器无法通过主机进行读写操作。这些寄存器用于存储内部计算的关键信息和中间结果。但是，当主机对相应的 AES_KEY2_n 或 AES_KEY3_n 地址执行写入操作时，整个 128 位 AES_KEY2_n 或 AES_KEY3_n 寄存器将被清零。
AES_GHASH_H_IN_n 寄存器（GHASH 是 GCM 的一部分，需要这些寄存器）映射到 AES_KEY2_n 寄存器。GCM 和 CCM 的（中间）认证结果存储在 AES_KEY3_n 寄存器中。

偏移 = 500h + (y * 4h)；其中 y = 0h 至 3h

图 12-23. AESKEY2_y 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AES_KEY2																															
W-0h																															

表 12-67. AESKEY2_y 寄存器字段说明

位	字段	类型	复位	说明
31-0	AES_KEY2	W	0h	AES_KEY2/AES_GHASH_H[31:0] 对于 GCM： -[127:0] - GHASH_H - 内部计算的 GHASH 密钥存储在这些寄存器中。仅用于使用 GHASH 函数的模式 (GCM)。 -[255:128] - 此寄存器用于存储中间值，并在加载新密钥时初始化为 0。 对于 CCM： -[255:0] - 此寄存器用于存储中间值。 对于 CBC-MAC： -[255:0] - 零 - 此寄存器必须保持为 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.17 AESKEY3_y 寄存器（偏移 = 510h + 公式）[复位 = 00000000h]

图 12-24 展示了 AESKEY3_y，表 12-68 中对此进行了介绍。

返回到汇总表。

AES_KEY3_0 / AES_KEY2_4
第三密钥 / 第二密钥（内部，但可清除）
以下寄存器无法通过主机进行读写操作。这些寄存器用于存储内部计算的关键信息和中间结果。但是，当主机对相应的 AES_KEY2_n 或 AES_KEY3_n 地址执行写入操作时，整个 128 位 AES_KEY2_n 或 AES_KEY3_n 寄存器将被清零。
AES_GHASH_H_IN_n 寄存器（GHASH 是 GCM 的一部分，需要这些寄存器）映射到 AES_KEY2_n 寄存器。GCM 和 CCM 的（中间）认证结果存储在 AES_KEY3_n 寄存器中。

偏移 = 510h + (y * 4h)；其中 y = 0h 至 3h

图 12-24. AESKEY3_y 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AES_KEY3																															
W-0h																															

表 12-68. AESKEY3_y 寄存器字段说明

位	字段	类型	复位	说明
31-0	AES_KEY3	W	0h	AES_KEY3[31:0]/AES_KEY2[159:128] 对于 GCM： -[127:0] - GHASH_H - 内部计算的 GHASH 密钥存储在这些寄存器中。仅用于使用 GHASH 函数的模式 (GCM)。 -[255:128] - 此寄存器用于存储中间值，并在加载新密钥时初始化为 0。 对于 CCM： -[255:0] - 此寄存器用于存储中间值。 对于 CBC-MAC： -[255:0] - 零 - 此寄存器必须保持为 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.18 AESIV_y 寄存器（偏移 = 540h + 公式）[复位 = 00000000h]

图 12-25 展示了 AESIV_y，表 12-69 中对此进行了介绍。

返回到汇总表。

AES 初始化向量寄存器
这些寄存器用于提供和读取 AES 引擎的 IV。

偏移 = 540h + (y * 4h)；其中 y = 0h 至 3h

图 12-25. AESIV_y 寄存器

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

表 12-69. AESIV_y 寄存器字段说明

位	字段	类型	复位	说明
31-0	AES_IV	R/W	0h	AES_IV[31:0] 初始化向量 用于常规非 ECB 模式 (CBC/CTR)： -[127:0] - AES_IV - 对于常规 AES 操作（CBC 和 CTR），这些寄存器必须写入新的 128 位 IV。操作结束后，这些寄存器包含由 EIP-120t 生成的最新 128 位结果初始化向量。如果选择 CTR 模式、则此值以 0x1 递增：首次使用后 - 当向引擎提交新的数据块时 对于 GCM： -[127:0] - AES_IV - 对于 GCM 操作，这些寄存器必须写入新的 128 位 IV。 操作结束后，这些寄存器包含由 EIP-120t 生成的更新后的 128 位结果 IV。请注意，IV 的 [127:96] 位表示初始计数器值（对于 GCM 为 1），因此必须初始化为 0x01000000。此值以 0x1 递增：首次使用后 - 当向引擎提交新的数据块时。 对于 CCM： -[127:0] - A0：对于 CCM，此字段必须写入值 A0，该值由以下部分拼接而成：A0 标志（5 位 0 和 3 位“L”）、随机数和计数器值。“L”必须与 AES_CTRL 寄存器中的“L”值相同。此“L”指示随机数和计数器的宽度。加载的计数器必须初始化为 0。A0 的总宽度为 128 位。 对于 CBC-MAC： -[127:0] - 零 - 对于 CBC-MAC 模式，此寄存器必须在每次操作开始时写入 0。操作结束后，这些寄存器包含由 EIP-120t 生成的 128 位 TAG 输出。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.19 AESCTL 寄存器（偏移 = 550h）[复位 = 80000000h]

图 12-26 展示了 AESCTL，表 12-70 中对此进行了介绍。

返回到汇总表。

AES 控制
AES 输入/输出缓冲器控制和模式寄存器
此寄存器指定 EIP-120t 的 AES 操作模式。
如果此寄存器的位 [28:5] 全部为 0，则自动选择电码本 (ECB) 模式。

图 12-26. AESCTL 寄存器

31	30	29	28	27	26	25	24
CONTEXT_READY	SAVED_CONTEXT_RDY	SAVE_CONTEXT	RESERVED				CCM_M
R-1h	R/W-0h	R/W-0h	R-0h				R/W-0h
23	22	21	20	19	18	17	16
CCM_M			CCM_L		CCM	GCM	
R/W-0h			R/W-0h		R/W-0h	R/W-0h	
15	14	13	12	11	10	9	8
CBC_MAC	RESERVED						CTR_WIDTH
R/W-0h	R-0h						R/W-0h
7	6	5	4	3	2	1	0
CTR_WIDTH	CTR	CBC	KEY_SIZE		DIR	INPUT_READY	OUTPUT_READY
R/W-0h	R/W-0h	R/W-0h	R-0h		R/W-0h	R/W-0h	R/W-0h

表 12-70. AESCTL 寄存器字段说明

位	字段	类型	复位	说明
31	CONTEXT_READY	R	1h	如果为“1”，该只读状态位表示上下文数据寄存器可以被覆盖并且允许主机写入下一个上下文。
30	SAVED_CONTEXT_RDY	R/W	0h	如果为“1”，该状态位表示有一个 AES 身份验证 TAG 和/或 IV 块可供主机检索。仅当“save_context”位设置为“1”时才会将该位置为有效。该位与 context_ready 位互斥。 写入 1 会将该位清除为 0，表示 AES 内核可以开始其下一个操作。当读取输出 TAG 和/或 IV 的第 4 个字时，此位也会被清除。 注意：当此模式位写入为 1 时，所有其他模式位写入操作都将被忽略。 注意：该位由 EIP-120t 自动控制，用于 TAG 读取 DMA 操作。
29	SAVE_CONTEXT	R/W	0h	此位指示需要将认证 TAG 或结果 IV 存储为结果上下文。 通常，对于返回 TAG 的认证模式（CBC-MAC、GCM 和 CCM），或者对于需要使用当前结果 IV 继续执行后续操作的基本加密模式，必须设置此位。 如果设置了此位，引擎将保留其完整上下文，直到读取 TAG 和/或 IV 寄存器为止。 AES 引擎必须先读取 TAG 或 IV，才能开始新的操作。
28-25	RESERVED	R	0h	保留
24-22	CCM_M	R/W	0h	定义“M”以指示用于 CCM 操作的身份验证字段的长度 身份验证字段长度等于两倍（CCM-M 的值加 1）。 注意：EIP-120t 始终返回一个 128 位身份验证字段，其中有 M 个最低有效字节有效。支持所有值。
21-19	CCM_L	R/W	0h	定义“L”以指示用于 CCM 操作的长度字段的宽度 长度字段（以字节为单位）等于 CMM-L 的值加 1。支持所有值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 12-70. AESCTL 寄存器字段说明（续）

位	字段	类型	复位	说明
18	CCM	R/W	0h	如果设置为“1”，则会选择 AES-CCM AES-CCM 是一种组合模式，使用 AES 进行身份验证和加密。 注意：选择 AES-CCM 模式需要在写入所有其他寄存器之后写入 AAD 长度寄存器。 注意：该寄存器中的 CTR 模式位也必须设置为 1 才能启用 AES-CTR 选择 CTR 模式以外的其他 AES 模式是无效的。
17-16	GCM	R/W	0h	将这些位设置为 11 以选择 AES-GCM 模式。 AES-GCM 是一种组合模式，使用伽罗瓦域乘法器 GF（2 的 128 次方）进行认证，并使用 AES-CTR 模式进行加密。 注意：此寄存器中的 CTR 模式位也必须设置为 1 才能启用 AES-CTR 位组合说明： 00 = 无 GCM 模式 01 = 保留，请勿选择 10 = 保留，请勿选择 11 = 自主 GHASH（H 和 Y0 加密均在内部计算） 注意：EIP-120t-1 配置仅支持模式 11（自主 GHASH），不允许其他 GCM 模式。
15	CBC_MAC	R/W	0h	设置为 1 可选择 AES-CBC MAC 模式。 对于此模式，方向位必须设置为 1。 选择此模式需要在写入所有其他寄存器之后写入长度寄存器。
14-9	RESERVED	R	0h	保留
8-7	CTR_WIDTH	R/W	0h	指定 AES-CTR 模式的计数器宽度 00 = 32 位计数器 01 = 64 位计数器 10 = 96 位计数器 11 = 128 位计数器 0h = 32_BIT：32 位 1h = 64_BIT：64 位 2h = 96_BIT：96 位 3h = 128_BIT：128 位
6	CTR	R/W	0h	如果设置为“1”，则会选择 AES 计数器模式 (CTR)。 注意：当需要加密/解密时，还必须为 GCM 和 CCM 设置该位。
5	CBC	R/W	0h	如果设置为“1”，则会选择密码分组链接 (CBC) 模式。
4-3	KEY_SIZE	R	0h	此只读字段用于指定密钥大小。 通过密钥存储区模块加载新密钥时，会自动配置密钥大小。 00 = 不适用 - 保留 01 = 128 位 10 = 192 位 11 = 256 位
2	DIR	R/W	0h	如果设置为 1，则执行加密操作。 如果设置为 0，则执行解密操作。 当选择 CBC-MAC 时、必须将该位写入 1。
1	INPUT_READY	R/W	0h	如果为 1，则此状态位表示 16 字节 AES 输入缓冲器为空。主机可以写入下一个数据块。 写入 0 会将该位清除为 0 并表示 AES 内核可以使用提供的输入数据块。 向该位写入 1 会被忽略。 注意：对于 DMA 操作，此位由 EIP-120T 自动控制。 复位后，该位为 0。写入上下文后，该位将变为“1”。
0	OUTPUT_READY	R/W	0h	如果为 1，则此状态位表示主机可以检索 AES 输出块。 写入 0 会将该位清除为 0，并表示主机已读取输出数据。AES 内核可以提供下一个输出数据块。 向该位写入 1 会被忽略。 注意：对于 DMA 操作，此位由 EIP-120T 自动控制。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.20 AESDATALEN0 寄存器（偏移 = 554h）[复位 = 00000000h]

图 12-27 展示了 AESDATALEN0，表 12-71 对其进行了介绍。

返回到汇总表。

AES 加密长度 0 (LSW)
这些寄存器用于将长度值写入 EIP-120t。处理期间，长度值递减至 0。如果两个长度均为 0，则表示数据流完成，并会请求新的上下文。对于基本 AES 模式（ECB、CBC 和 CTR），如果需要使用相同的密钥处理多个流，则可以写入长度为 0 的加密信息。写入长度为 0 会导致持续的数据请求，直到写入新的上下文为止。对于其他模式（CBC-MAC、GCM 和 CCM），如果长度递减至 0 或等于 0，则不会发出（新的）数据请求。
建议为每个数据包都写入一个新的长度。如果长度寄存器递减至 0，则在写入新的上下文或长度值之前，不会处理任何新数据。
当写入新模式而不写入长度寄存器时，将重用先前上下文中的长度寄存器值。

图 12-27. AESDATALEN0 寄存器

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

表 12-71. AESDATALEN0 寄存器字段说明

位	字段	类型	复位	说明
31-0	C_LENGTH	W	0h	C_LENGTH[31:0] 加密长度寄存器（LSW 和 MSW）的位 [60:0] 存储所有模式的加密数据长度（以字节为单位）。一旦开始使用此上下文进行处理，此长度将递减至 0。允许的数据长度最大为 (261: 1) 字节。 对于 GCM，可以使用不超过 236 - 32 字节的任何值。这是因为使用了 32 位计数器模式， 128 位块的最大数量为 232 - 2，导致最大字节数为 236 - 32。 写入此寄存器会触发引擎开始使用此上下文。这对除 GCM 和 CCM 之外的所有模式都有效。 注意：对于组合模式（GCM 和 CCM），此长度不包括仅用于认证的数据 身份验证长度在 AESAUTHLEN 寄存器中指定 所有模式的长度必须大于 0。对于组合模式，允许其中一个长度为 0。 对于基本加密模式（ECB、CBC 和 CTR），允许通过编程方式将长度字段设为零 这时长度被视为无限。 所有数据都必须按字节（8 位）对齐 EIP-120t 不支持按位对齐的数据流。对于分组密码模式，数据长度必须以分组密码大小（16 字节）的倍数进行编程。 对于主机读取操作，这些寄存器返回全 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.21 AESDATALEN1 寄存器（偏移 = 558h）[复位 = 00000000h]

图 12-28 展示了 AESDATALEN1，表 12-72 对其进行了介绍。

返回到汇总表。

AES 加密长度 1 (MSW)
这些寄存器用于将长度值写入 EIP-120t。处理期间，长度值递减至 0。如果两个长度均为 0，则表示数据流完成，并会请求新的上下文。对于基本 AES 模式（ECB、CBC 和 CTR），如果需要使用相同的密钥处理多个流，则可以写入长度为 0 的加密信息。写入长度为 0 会导致持续的数据请求，直到写入新的上下文为止。对于其他模式（CBC-MAC、GCM 和 CCM），如果长度递减至 0 或等于 0，则不会发出（新的）数据请求。
建议为每个数据包都写入一个新的长度。如果长度寄存器递减至 0，则在写入新的上下文或长度值之前，不会处理任何新数据。
当写入新模式而不写入长度寄存器时，将重用先前上下文中的长度寄存器值。

图 12-28. AESDATALEN1 寄存器

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

表 12-72. AESDATALEN1 寄存器字段说明

位	字段	类型	复位	说明
31-29	RESERVED	R	0h	保留
28-0	C_LENGTH	W	0h	C_LENGTH[60:32] 加密长度寄存器（LSW 和 MSW）的位 [60:0] 存储所有模式的加密数据长度（以字节为单位）。一旦开始使用此上下文进行处理，此长度将递减至 0。允许的数据长度最大为 (261: 1) 字节。 对于 GCM，可以使用不超过 236 - 32 字节的任何值。这是因为使用了 32 位计数器模式， 128 位块的最大数量为 232 - 2，导致最大字节数为 236 - 32。 写入此寄存器会触发引擎开始使用此上下文。这对除 GCM 和 CCM 之外的所有模式都有效。 注意：对于组合模式（GCM 和 CCM），此长度不包括仅用于认证的数据 身份验证长度在 AESAUTHLEN 寄存器中指定 所有模式的长度必须大于 0。对于组合模式，允许其中一个长度为 0。 对于基本加密模式（ECB、CBC 和 CTR），允许通过编程方式将长度字段设为零 这时长度被视为无限。 所有数据都必须按字节（8 位）对齐 EIP-120t 不支持按位对齐的数据流。对于分组密码模式，数据长度必须以分组密码大小（16 字节）的倍数进行编程。 对于主机读取操作，这些寄存器返回全 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.22 AESAUTHLEN 寄存器（偏移 = 55Ch）[复位 = 00000000h]

图 12-29 展示了 AESAUTHLEN，表 12-73 中对此进行了介绍。

返回到汇总表。

AES 身份验证长度

图 12-29. AESAUTHLEN 寄存器

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

表 12-73. AESAUTHLEN 寄存器字段说明

位	字段	类型	复位	说明
31-0	AUTH_LENGTH	W	0h	身份验证长度寄存器的位 [31:0] 仅存储组合模式（GCM 或 CCM）的身份验证数据长度（以字节为单位） 对于 CCM，支持的 AAD 长度为 0 至 (2^{16} - 2^{8}) 字节。对于 GCM，可以使用不超过 (2^{32} - 1) 字节的任何值。一旦开始使用此上下文进行处理，该长度将递减至 0。 对该寄存器执行一次写操作，会触发引擎开始使用该上下文来进行 GCM 和 CCM。 对于主机读取操作，这些寄存器返回全 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.23 AESDATAOUT0 寄存器（偏移 = 560h）[复位 = 00000000h]

图 12-30 展示了 AESDATAOUT0，表 12-74 对其进行了介绍。

返回到汇总表。

数据输入/输出

图 12-30. AESDATAOUT0 寄存器

31	30	29	28	27	26	25	24	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																															

表 12-74. AESDATAOUT0 寄存器字段说明

位	字段	类型	复位	说明
31-0	DATA	R	0h	数据寄存器 0：用于从加密外设输出分组数据。 这些位 =AES Output Data[31:0]，来源于 {127:0} 在正常操作下，不使用该寄存器，因为数据输入和输出通过 DMA 在 AES 引擎与外设之间传输。 对于主机的读取操作，这些寄存器将包含来自最近一次 AES 操作的 128 位输出分组。从该地址范围内的字对齐偏移量读取数据，将从 4 字深（16 字节 = 128 位 AES 块）的数据输出缓冲区中读取一个字（4 字节）的数据。必须先读取这些字（4 个字，一个完整的块），内核才会将下一个块移动到数据输出缓冲区。要清空数据输出缓冲区，必须写入 AESCTL.OUTPUT_READY。 对于带认证的模式（CBC-MAC、GCM 和 CCM），无效（消息）字节/字可以写入任何数据。 注意：AAD/仅认证数据不会复制到输出缓冲区，而仅用于认证。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.24 AESDATAIN0 寄存器（偏移 = 560h）[复位 = 00000000h]

图 12-31 展示了 AESDATAIN0，表 12-75 对其进行了介绍。

返回到汇总表。

AES 数据 Input_Output 0
数据寄存器通常通过 DMA 访问，而不是通过主机写入和/或读取来访问。然而，出于调试目的，可通过主机写入和读取操作来访问数据输入/输出寄存器。这些寄存器用于缓冲进出 EIP-120t 的输入/输出数据块。
注意：数据输入缓冲器 (AES_DATA_IN_n) 和数据输出缓冲器 (AES_DATA_OUT_n) 映射到相同的地址位置。
写入（DMA 和主机）到这些地址会将数据加载到输入缓冲区，而读取操作则会从输出缓冲区读取数据。因此，对于写入访问，写入的是数据输入缓冲器
对于读取访问，读取的是数据输出缓冲器。在开始操作之前必须向数据输入缓冲器写入值。数据输出缓冲器在操作完成时包含有效数据。因此，任何 128 位数据块都可以分割成多次 32 位字传输
这些字传输可以与外部接口上的其他主机传输混合进行。

图 12-31. AESDATAIN0 寄存器

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

表 12-75. AESDATAIN0 寄存器字段说明

位	字段	类型	复位	说明
31-0	AES_DATA_IN_OUT	W	0h	AES 输入数据[31:0] / AES 输出数据[31:0] 用于在 EIP-120t 与输入/输出模块之间传输数据的寄存器。 正常情况下，不使用此寄存器，因为数据输入和输出通过 DMA 与 AES 内核进行传输。对于主机写入操作，必须使用 128 位输入块写入这些寄存器，以便进行下一次 AES 操作。在此地址范围内以字对齐偏移量写入数据，会将一个字（4 字节）的数据存储到 4 字深（16 字节 = 128 位 AES 块）数据输入缓冲区的相应位置。此缓冲器用于下一次 AES 操作。如果最后一个数据块未完全填充有效数据（参见以下注释），则只允许写入包含有效数据的字。下一次 AES 操作通过写入 AES_CTRL 寄存器的 input_ready 标志来触发。 对于主机读取操作，这些寄存器包含来自最近一次 AES 操作的 128 位输出块。从该地址范围内的字对齐偏移量读取数据，将从 4 字深（16 字节 = 128 位 AES 块）的数据输出缓冲区中读取一个字（4 字节）的数据。必须先读取这些字（4 个字，一个完整的块），内核才会将下一个块移动到数据输出缓冲区。要清空数据输出缓冲区，必须写入 AES_CTRL 寄存器的 output_ready 标志。 对于带认证的模式（CBC-MAC、GCM 和 CCM），无效（消息）字节/字可以写入任何数据。 注意：AES 通常以 128 位块对输入数据进行操作。CTR、GCM 和 CCM 模式属于例外情况。CTR 模式消息的最后一个块可能包含少于 128 位的数据（参见 [NIST 800-38A]）。对于 GCM/CCM，AAD 和消息数据的最后一个块可能包含少于 128 位的数据（请参阅[NIST 800-38D]）。EIP-120t 会对在结束处不对齐的数据块进行自动填充或掩码处理：在 GCM、CCM 和 CBC-MAC 模式下，会用 0 对这些块进行填充或掩码处理。对于 CTR 模式，未对齐数据块中的剩余数据将被忽略。 注意：AAD/仅认证数据不会复制到输出缓冲区，而仅用于认证。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.25 AESDATAOUT1 寄存器（偏移 = 564h）[复位 = 00000000h]

图 12-32 展示了 AESDATAOUT1，表 12-76 对其进行了介绍。

返回到汇总表。

数据输入/输出

图 12-32. AESDATAOUT1 寄存器

31	30	29	28	27	26	25	24	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																															

表 12-76. AESDATAOUT1 寄存器字段说明

位	字段	类型	复位	说明
31-0	DATA	R	0h	数据寄存器 0：用于从加密外设输出分组数据。 这些位 =AES Output Data[31:0]，来源于 {127:0} 在正常操作下，不使用该寄存器，因为数据输入和输出通过 DMA 在 AES 引擎与外设之间传输。 对于主机的读取操作，这些寄存器将包含来自最近一次 AES 操作的 128 位输出分组。从该地址范围内的字对齐偏移量读取数据，将从 4 字深（16 字节 = 128 位 AES 块）的数据输出缓冲区中读取一个字（4 字节）的数据。必须先读取这些字（4 个字，一个完整的块），内核才会将下一个块移动到数据输出缓冲区。要清空数据输出缓冲区，必须写入 AESCTL.OUTPUT_READY。 对于带认证的模式（CBC-MAC、GCM 和 CCM），无效（消息）字节/字可以写入任何数据。 注意：AAD/仅认证数据不会复制到输出缓冲区，而仅用于认证。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.26 AESDATAIN1 寄存器（偏移 = 564h）[复位 = 00000000h]

图 12-33 展示了 AESDATAIN1，表 12-77 对其进行了介绍。

返回到汇总表。

AES 数据 Input_Output 0
数据寄存器通常通过 DMA 访问，而不是通过主机写入和/或读取来访问。然而，出于调试目的，可通过主机写入和读取操作来访问数据输入/输出寄存器。这些寄存器用于缓冲进出 EIP-120T 的输入/输出数据块。
注意：数据输入缓冲器 (AES_DATA_IN_n) 和数据输出缓冲器 (AES_DATA_OUT_n) 映射到相同的地址位置。
写入（DMA 和主机）到这些地址会将数据加载到输入缓冲区，而读取操作则会从输出缓冲区读取数据。因此，对于写入访问，写入的是数据输入缓冲器
对于读取访问，读取的是数据输出缓冲器。在开始操作之前必须向数据输入缓冲器写入值。数据输出缓冲器在操作完成时包含有效数据。因此，任何 128 位数据块都可以分割成多次 32 位字传输
这些字传输可以与外部接口上的其他主机传输混合进行。

图 12-33. AESDATAIN1 寄存器

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

表 12-77. AESDATAIN1 寄存器字段说明

位	字段	类型	复位	说明
31-0	AES_DATA_IN_OUT	W	0h	AES 输入数据[31:0] / AES 输出数据[63:32] 用于在 EIP-120t 与输入/输出模块之间传输数据的寄存器。 正常情况下，不使用此寄存器，因为数据输入和输出通过 DMA 与 AES 内核进行传输。对于主机写入操作，必须使用 128 位输入块写入这些寄存器，以便进行下一次 AES 操作。在此地址范围内以字对齐偏移量写入数据，会将一个字（4 字节）的数据存储到 4 字深（16 字节 = 128 位 AES 块）数据输入缓冲区的相应位置。此缓冲器用于下一次 AES 操作。如果最后一个数据块未完全填充有效数据（参见以下注释），则只允许写入包含有效数据的字。下一次 AES 操作通过写入 AES_CTRL 寄存器的 input_ready 标志来触发。 对于主机读取操作，这些寄存器包含来自最近一次 AES 操作的 128 位输出块。从该地址范围内的字对齐偏移量读取数据，将从 4 字深（16 字节 = 128 位 AES 块）的数据输出缓冲区中读取一个字（4 字节）的数据。必须先读取这些字（4 个字，一个完整的块），内核才会将下一个块移动到数据输出缓冲区。要清空数据输出缓冲区，必须写入 AES_CTRL 寄存器的 output_ready 标志。 对于带认证的模式（CBC-MAC、GCM 和 CCM），无效（消息）字节/字可以写入任何数据。 注意：AES 通常以 128 位块对输入数据进行操作。CTR、GCM 和 CCM 模式属于例外情况。CTR 模式消息的最后一个块可能包含少于 128 位的数据（参见 [NIST 800-38A]）。对于 GCM/CCM，AAD 和消息数据的最后一个块可能包含少于 128 位的数据（请参阅[NIST 800-38D]）。EIP-120t 会对在结束处不对齐的数据块进行自动填充或掩码处理：在 GCM、CCM 和 CBC-MAC 模式下，会用 0 对这些块进行填充或掩码处理。对于 CTR 模式，未对齐数据块中的剩余数据将被忽略。 注意：AAD/仅认证数据不会复制到输出缓冲区，而仅用于认证。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.27 AESDATAOUT2 寄存器（偏移 = 568h）[复位 = 00000000h]

图 12-34 展示了 AESDATAOUT2，表 12-78 对其进行了介绍。

返回到汇总表。

数据输入/输出

图 12-34. AESDATAOUT2 寄存器

31	30	29	28	27	26	25	24	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																															

表 12-78. AESDATAOUT2 寄存器字段说明

位	字段	类型	复位	说明
31-0	DATA	R	0h	数据寄存器 0：用于从加密外设输出分组数据。 这些位 =AES Output Data[31:0]，来源于 {127:0} 在正常操作下，不使用该寄存器，因为数据输入和输出通过 DMA 在 AES 引擎与外设之间传输。 对于主机的读取操作，这些寄存器将包含来自最近一次 AES 操作的 128 位输出分组。从该地址范围内的字对齐偏移量读取数据，将从 4 字深（16 字节 = 128 位 AES 块）的数据输出缓冲区中读取一个字（4 字节）的数据。必须先读取这些字（4 个字，一个完整的块），内核才会将下一个块移动到数据输出缓冲区。要清空数据输出缓冲区，必须写入 AESCTL.OUTPUT_READY。 对于带认证的模式（CBC-MAC、GCM 和 CCM），无效（消息）字节/字可以写入任何数据。 注意：AAD/仅认证数据不会复制到输出缓冲区，而仅用于认证。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.28 AESDATAIN2 寄存器（偏移 = 568h）[复位 = 00000000h]

图 12-35 展示了 AESDATAIN2，表 12-79 对其进行了介绍。

返回到汇总表。

AES 数据 Input_Output 2
数据寄存器通常通过 DMA 访问，而不是通过主机写入和/或读取来访问。然而，出于调试目的，可通过主机写入和读取操作来访问数据输入/输出寄存器。这些寄存器用于缓冲进出 EIP-120T 的输入/输出数据块。
注意：数据输入缓冲器 (AES_DATA_IN_n) 和数据输出缓冲器 (AES_DATA_OUT_n) 映射到相同的地址位置。
写入（DMA 和主机）到这些地址会将数据加载到输入缓冲区，而读取操作则会从输出缓冲区读取数据。因此，对于写入访问，写入的是数据输入缓冲器
对于读取访问，读取的是数据输出缓冲器。在开始操作之前必须向数据输入缓冲器写入值。数据输出缓冲器在操作完成时包含有效数据。因此，任何 128 位数据块都可以分割成多次 32 位字传输
这些字传输可以与外部接口上的其他主机传输混合进行。

图 12-35. AESDATAIN2 寄存器

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

表 12-79. AESDATAIN2 寄存器字段说明

位	字段	类型	复位	说明
31-0	AES_DATA_IN_OUT	W	0h	AES 输入数据[95:64] / AES 输出数据[95:64] 用于在 EIP-120t 与输入/输出模块之间传输数据的寄存器。 正常情况下，不使用此寄存器，因为数据输入和输出通过 DMA 与 AES 内核进行传输。对于主机写入操作，必须使用 128 位输入块写入这些寄存器，以便进行下一次 AES 操作。在此地址范围内以字对齐偏移量写入数据，会将一个字（4 字节）的数据存储到 4 字深（16 字节 = 128 位 AES 块）数据输入缓冲区的相应位置。此缓冲器用于下一次 AES 操作。如果最后一个数据块未完全填充有效数据（参见以下注释），则只允许写入包含有效数据的字。下一次 AES 操作通过写入 AES_CTRL 寄存器的 input_ready 标志来触发。 对于主机读取操作，这些寄存器包含来自最近一次 AES 操作的 128 位输出块。从该地址范围内的字对齐偏移量读取数据，将从 4 字深（16 字节 = 128 位 AES 块）的数据输出缓冲区中读取一个字（4 字节）的数据。必须先读取这些字（4 个字，一个完整的块），内核才会将下一个块移动到数据输出缓冲区。要清空数据输出缓冲区，必须写入 AES_CTRL 寄存器的 output_ready 标志。 对于带认证的模式（CBC-MAC、GCM 和 CCM），无效（消息）字节/字可以写入任何数据。 注意：AES 通常以 128 位块对输入数据进行操作。CTR、GCM 和 CCM 模式属于例外情况。CTR 模式消息的最后一个块可能包含少于 128 位的数据（参见 [NIST 800-38A]）。对于 GCM/CCM，AAD 和消息数据的最后一个块可能包含少于 128 位的数据（请参阅[NIST 800-38D]）。EIP-120t 会对在结束处不对齐的数据块进行自动填充或掩码处理：在 GCM、CCM 和 CBC-MAC 模式下，会用 0 对这些块进行填充或掩码处理。对于 CTR 模式，未对齐数据块中的剩余数据将被忽略。 注意：AAD/仅认证数据不会复制到输出缓冲区，而仅用于认证。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.29 AESDATAOUT3 寄存器（偏移 = 56Ch）[复位 = 00000000h]

图 12-36 展示了 AESDATAOUT3，表 12-80 对其进行了介绍。

返回到汇总表。

数据输入/输出

图 12-36. AESDATAOUT3 寄存器

31	30	29	28	27	26	25	24	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																															

表 12-80. AESDATAOUT3 寄存器字段说明

位	字段	类型	复位	说明
31-0	DATA	R	0h	数据寄存器 0：用于从加密外设输出分组数据。 这些位 =AES Output Data[31:0]，来源于 {127:0} 在正常操作下，不使用该寄存器，因为数据输入和输出通过 DMA 在 AES 引擎与外设之间传输。 对于主机的读取操作，这些寄存器将包含来自最近一次 AES 操作的 128 位输出分组。从该地址范围内的字对齐偏移量读取数据，将从 4 字深（16 字节 = 128 位 AES 块）的数据输出缓冲区中读取一个字（4 字节）的数据。必须先读取这些字（4 个字，一个完整的块），内核才会将下一个块移动到数据输出缓冲区。要清空数据输出缓冲区，必须写入 AESCTL.OUTPUT_READY。 对于带认证的模式（CBC-MAC、GCM 和 CCM），无效（消息）字节/字可以写入任何数据。 注意：AAD/仅认证数据不会复制到输出缓冲区，而仅用于认证。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.30 AESDATAIN3 寄存器（偏移 = 56Ch）[复位 = 00000000h]

图 12-37 展示了 AESDATAIN3，表 12-81 对其进行了介绍。

返回到汇总表。

AES 数据 Input_Output 3
数据寄存器通常通过 DMA 访问，而不是通过主机写入和/或读取来访问。然而，出于调试目的，可通过主机写入和读取操作来访问数据输入/输出寄存器。这些寄存器用于缓冲进出 EIP-120T 的输入/输出数据块。
注意：数据输入缓冲器 (AES_DATA_IN_n) 和数据输出缓冲器 (AES_DATA_OUT_n) 映射到相同的地址位置。
写入（DMA 和主机）到这些地址会将数据加载到输入缓冲区，而读取操作则会从输出缓冲区读取数据。因此，对于写入访问，写入的是数据输入缓冲器
对于读取访问，读取的是数据输出缓冲器。在开始操作之前必须向数据输入缓冲器写入值。数据输出缓冲器在操作完成时包含有效数据。因此，任何 128 位数据块都可以分割成多次 32 位字传输
这些字传输可以与外部接口上的其他主机传输混合进行。

图 12-37. AESDATAIN3 寄存器

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

表 12-81. AESDATAIN3 寄存器字段说明

位	字段	类型	复位	说明
31-0	AES_DATA_IN_OUT	W	0h	AES 输入数据[127:96] / AES 输出数据[127:96] 用于在 EIP-120t 与输入/输出模块之间传输数据的寄存器。 正常情况下，不使用此寄存器，因为数据输入和输出通过 DMA 与 AES 内核进行传输。对于主机写入操作，必须使用 128 位输入块写入这些寄存器，以便进行下一次 AES 操作。在此地址范围内以字对齐偏移量写入数据，会将一个字（4 字节）的数据存储到 4 字深（16 字节 = 128 位 AES 块）数据输入缓冲区的相应位置。此缓冲器用于下一次 AES 操作。如果最后一个数据块未完全填充有效数据（参见以下注释），则只允许写入包含有效数据的字。下一次 AES 操作通过写入 AES_CTRL 寄存器的 input_ready 标志来触发。 对于主机读取操作，这些寄存器包含来自最近一次 AES 操作的 128 位输出块。从该地址范围内的字对齐偏移量读取数据，将从 4 字深（16 字节 = 128 位 AES 块）的数据输出缓冲区中读取一个字（4 字节）的数据。必须先读取这些字（4 个字，一个完整的块），内核才会将下一个块移动到数据输出缓冲区。要清空数据输出缓冲区，必须写入 AES_CTRL 寄存器的 output_ready 标志。 对于带认证的模式（CBC-MAC、GCM 和 CCM），无效（消息）字节/字可以写入任何数据。 注意：AES 通常以 128 位块对输入数据进行操作。CTR、GCM 和 CCM 模式属于例外情况。CTR 模式消息的最后一个块可能包含少于 128 位的数据（参见 [NIST 800-38A]）。对于 GCM/CCM，AAD 和消息数据的最后一个块可能包含少于 128 位的数据（请参阅[NIST 800-38D]）。EIP-120t 会对在结束处不对齐的数据块进行自动填充或掩码处理：在 GCM、CCM 和 CBC-MAC 模式下，会用 0 对这些块进行填充或掩码处理。对于 CTR 模式，未对齐数据块中的剩余数据将被忽略。 注意：AAD/仅认证数据不会复制到输出缓冲区，而仅用于认证。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.31 AESTAGOUT_y 寄存器（偏移 = 570h + 公式）[复位 = 00000000h]

图 12-38 展示了 AESTAGOUT_y，表 12-82 中对此进行了介绍。

返回到汇总表。

AES 标记输出 0
这些标记寄存器可通过 DMA 访问，或由主机进行直接读取。
该寄存器用于缓冲来自 EIP-120t 的 TAG。这些寄存器与中间身份验证结果寄存器是共用的，但在处理完成之前无法读取。处理过程中，从这些寄存器读取会返回全 0。如果某次操作不返回 TAG，那么读取这些寄存器将返回 IV。如果某次操作同时返回 TAG 和 IV，并且主机需要读取两者，则主机必须先读取 TAG，再读取 IV。如果按相反的顺序读取，则会返回两次 IV。

偏移 = 570h + (y * 4h)；其中 y = 0h 至 3h

图 12-38. AESTAGOUT_y 寄存器

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

表 12-82. AESTAGOUT_y 寄存器字段说明

位	字段	类型	复位	说明
31-0	AES_TAG	R	0h	AES_TAG[31:0] 此寄存器的 [31:0] 位存储组合模式和仅认证模式下的认证值。 对于主机读取操作，这些寄存器包含 EIP-120t 的最后一个 128 位 TAG 输出 该 TAG 在下一个上下文写入之前一直可用。 仅当 TAG 可用且 AESCTL.SAVED_CONTEXT_RDY 寄存器已设置时，此寄存器才会包含有效数据。在处理过程中或者对于不返回 TAG 的操作/模式，对该寄存器的读取将返回 IV 寄存器中的数据。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.32 HASHDATAIN1 寄存器（偏移 = 604h）[复位 = 00000000h]

图 12-39 展示了 HASHDATAIN1，表 12-83 对其进行了介绍。

返回到汇总表。

HASH 数据输入 1
数据输入寄存器应该用于通过从机接口向哈希模块提供输入数据。

图 12-39. HASHDATAIN1 寄存器

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

表 12-83. HASHDATAIN1 寄存器字段说明

位	字段	类型	复位	说明
31-0	HASH_DATA_IN	W	0h	HASH_DATA_IN[63:32] 这些寄存器必须写入 512 位输入数据。数据线直接连接到哈希模块的数据输入，因此连接到引擎的内部数据缓冲区。对每个寄存器的写入都会触发内部缓冲器对应的 32 位写入使能信号。 注意：只有当 HASHIOBUFCTRL.RFD_IN 为 1 时，主机才能写入输入数据缓冲区。如果 HASHIOBUFCTRL.RFD_IN 为 0，则引擎正忙于处理。处理期间，不允许写入新的输入数据。 对于长度大于 64 字节的消息，会通过 HASHIOBUFCTRL 寄存器的标志进行握手，将多个数据块写入此输入缓冲区。除最后一个数据块外，所有数据块的大小都必须为 512 位。如果最后一个数据块的长度不是 512 位，则只能写入数据的最低有效位，但必须用 0 填充到下一个 32 位边界。 主机从这些寄存器地址读取数据的操作将返回 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.33 HASHDATAIN2 寄存器（偏移 = 608h）[复位 = 00000000h]

图 12-40 展示了 HASHDATAIN2，表 12-84 对其进行了介绍。

返回到汇总表。

HASH 数据输入 2
数据输入寄存器应该用于通过从机接口向哈希模块提供输入数据。

图 12-40. HASHDATAIN2 寄存器

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

表 12-84. HASHDATAIN2 寄存器字段说明

位	字段	类型	复位	说明
31-0	HASH_DATA_IN	W	0h	HASH_DATA_IN[95:64] 这些寄存器必须写入 512 位输入数据。数据线直接连接到哈希模块的数据输入，因此连接到引擎的内部数据缓冲区。对每个寄存器的写入都会触发内部缓冲器对应的 32 位写入使能信号。 注意：只有当 HASHIOBUFCTRL.RFD_IN 为 1 时，主机才能写入输入数据缓冲区。如果 HASHIOBUFCTRL.RFD_IN 为 0，则引擎正忙于处理。处理期间，不允许写入新的输入数据。 对于长度大于 64 字节的消息，会通过 HASHIOBUFCTRL 寄存器的标志进行握手，将多个数据块写入此输入缓冲区。除最后一个数据块外，所有数据块的大小都必须为 512 位。如果最后一个数据块的长度不是 512 位，则只能写入数据的最低有效位，但必须用 0 填充到下一个 32 位边界。 主机从这些寄存器地址读取数据的操作将返回 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.34 HASHDATAIN3 寄存器（偏移 = 60Ch）[复位 = 00000000h]

图 12-41 展示了 HASHDATAIN3，表 12-85 对其进行了介绍。

返回到汇总表。

HASH 数据输入 3
数据输入寄存器应该用于通过从机接口向哈希模块提供输入数据。

图 12-41. HASHDATAIN3 寄存器

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

表 12-85. HASHDATAIN3 寄存器字段说明

位	字段	类型	复位	说明
31-0	HASH_DATA_IN	W	0h	HASH_DATA_IN[127:96] 这些寄存器必须写入 512 位输入数据。数据线直接连接到哈希模块的数据输入，因此连接到引擎的内部数据缓冲区。对每个寄存器的写入都会触发内部缓冲器对应的 32 位写入使能信号。 注意：只有当 HASH_IO_BUF_CTRL 寄存器的 rfd_in 位为高电平时，主机才能写入输入数据缓冲区。如果 rfd_in 位为 0，则表示引擎正忙于处理。在处理过程中、不允许写入新的输入数据。 对于大于 64 字节的消息长度、使用通过 HASH_IO_BUF_CTRL 寄存器的标志进行握手的方式、将多个数据块写入该输入缓冲区。除最后一个数据块外，所有数据块的大小都必须为 512 位。如果最后一个数据块的长度不是 512 位，则只能写入数据的最低有效位，但必须用 0 填充到下一个 32 位边界。 主机从这些寄存器地址读取数据的操作将返回 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.35 HASHDATAIN4 寄存器（偏移 = 610h）[复位 = 00000000h]

图 12-42 展示了 HASHDATAIN4，表 12-86 对其进行了介绍。

返回到汇总表。

HASH 数据输入 4
数据输入寄存器应该用于通过从机接口向哈希模块提供输入数据。

图 12-42. HASHDATAIN4 寄存器

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

表 12-86. HASHDATAIN4 寄存器字段说明

位	字段	类型	复位	说明
31-0	HASH_DATA_IN	W	0h	HASH_DATA_IN[159:128] 这些寄存器必须写入 512 位输入数据。数据线直接连接到哈希模块的数据输入，因此连接到引擎的内部数据缓冲区。对每个寄存器的写入都会触发内部缓冲器对应的 32 位写入使能信号。 注意：只有当 HASHIOBUFCTRL.RFD_IN 为“1”时，主机才能写入输入数据缓冲区。如果 HASHIOBUFCTRL.RFD_IN 为 0，则引擎正忙于处理。处理期间，不允许写入新的输入数据。 对于长度大于 64 字节的消息，会通过 HASHIOBUFCTRL 寄存器的标志进行握手，将多个数据块写入此输入缓冲区。除最后一个数据块外，所有数据块的大小都必须为 512 位。如果最后一个数据块的长度不是 512 位，则只能写入数据的最低有效位，但必须用 0 填充到下一个 32 位边界。 主机从这些寄存器地址读取数据的操作将返回 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.36 HASHDATAIN5 寄存器（偏移 = 614h）[复位 = 00000000h]

图 12-43 展示了 HASHDATAIN5，表 12-87 对其进行了介绍。

返回到汇总表。

HASH 数据输入 5
数据输入寄存器应该用于通过从机接口向哈希模块提供输入数据。

图 12-43. HASHDATAIN5 寄存器

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

表 12-87. HASHDATAIN5 寄存器字段说明

位	字段	类型	复位	说明
31-0	HASH_DATA_IN	W	0h	HASH_DATA_IN[191:160] 这些寄存器必须写入 512 位输入数据。数据线直接连接到哈希模块的数据输入，因此连接到引擎的内部数据缓冲区。对每个寄存器的写入都会触发内部缓冲器对应的 32 位写入使能信号。 注意：只有当 HASHIOBUFCTRL.RFD_IN 为 1 时，主机才能写入输入数据缓冲区。如果 HASHIOBUFCTRL.RFD_IN 为 0，则引擎正忙于处理。处理期间，不允许写入新的输入数据。 对于长度大于 64 字节的消息，会通过 HASHIOBUFCTRL 寄存器的标志进行握手，将多个数据块写入此输入缓冲区。除最后一个数据块外，所有数据块的大小都必须为 512 位。如果最后一个数据块的长度不是 512 位，则只能写入数据的最低有效位，但必须用 0 填充到下一个 32 位边界。 主机从这些寄存器地址读取数据的操作将返回 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.37 HASHDATAIN6 寄存器（偏移 = 618h）[复位 = 00000000h]

图 12-44 展示了 HASHDATAIN6，表 12-88 对其进行了介绍。

返回到汇总表。

HASH 数据输入 6
数据输入寄存器应该用于通过从机接口向哈希模块提供输入数据。

图 12-44. HASHDATAIN6 寄存器

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

表 12-88. HASHDATAIN6 寄存器字段说明

位	字段	类型	复位	说明
31-0	HASH_DATA_IN	W	0h	HASH_DATA_IN[223:192] 这些寄存器必须写入 512 位输入数据。数据线直接连接到哈希模块的数据输入，因此连接到引擎的内部数据缓冲区。对每个寄存器的写入都会触发内部缓冲器对应的 32 位写入使能信号。 注意：只有当 HASHIOBUFCTRL.RFD_IN 为 1 时，主机才能写入输入数据缓冲区。如果 HASHIOBUFCTRL.RFD_IN 为 0，则引擎正忙于处理。处理期间，不允许写入新的输入数据。 对于长度大于 64 字节的消息，会通过 HASHIOBUFCTRL 寄存器的标志进行握手，将多个数据块写入此输入缓冲区。除最后一个数据块外，所有数据块的大小都必须为 512 位。如果最后一个数据块的长度不是 512 位，则只能写入数据的最低有效位，但必须用 0 填充到下一个 32 位边界。 主机从这些寄存器地址读取数据的操作将返回 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.38 HASHDATAIN7 寄存器（偏移 = 61Ch）[复位 = 00000000h]

图 12-45 展示了 HASHDATAIN7，表 12-89 对其进行了介绍。

返回到汇总表。

HASH 数据输入 7
数据输入寄存器应该用于通过从机接口向哈希模块提供输入数据。

图 12-45. HASHDATAIN7 寄存器

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

表 12-89. HASHDATAIN7 寄存器字段说明

位	字段	类型	复位	说明
31-0	HASH_DATA_IN	W	0h	HASH_DATA_IN[255:224] 这些寄存器必须写入 512 位输入数据。数据线直接连接到哈希模块的数据输入，因此连接到引擎的内部数据缓冲区。对每个寄存器的写入都会触发内部缓冲器对应的 32 位写入使能信号。 注意：只有当 HASHIOBUFCTRL.RFD_IN 为 1 时，主机才能写入输入数据缓冲区。如果 HASHIOBUFCTRL.RFD_IN 为 0，则引擎正忙于处理。处理期间，不允许写入新的输入数据。 对于长度大于 64 字节的消息，会通过 HASHIOBUFCTRL 寄存器的标志进行握手，将多个数据块写入此输入缓冲区。除最后一个数据块外，所有数据块的大小都必须为 512 位。如果最后一个数据块的长度不是 512 位，则只能写入数据的最低有效位，但必须用 0 填充到下一个 32 位边界。 主机从这些寄存器地址读取数据的操作将返回 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.39 HASHDATAIN8 寄存器（偏移 = 620h）[复位 = 00000000h]

图 12-46 展示了 HASHDATAIN8，表 12-90 对其进行了介绍。

返回到汇总表。

HASH 数据输入 8
数据输入寄存器应该用于通过从机接口向哈希模块提供输入数据。

图 12-46. HASHDATAIN8 寄存器

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

表 12-90. HASHDATAIN8 寄存器字段说明

位	字段	类型	复位	说明
31-0	HASH_DATA_IN	W	0h	HASH_DATA_IN[287:256] 这些寄存器必须写入 512 位输入数据。数据线直接连接到哈希模块的数据输入，因此连接到引擎的内部数据缓冲区。对每个寄存器的写入都会触发内部缓冲器对应的 32 位写入使能信号。 注意：只有当 HASHIOBUFCTRL.RFD_IN 为 1 时，主机才能写入输入数据缓冲区。如果 HASHIOBUFCTRL.RFD_IN 为 0，则引擎正忙于处理。处理期间，不允许写入新的输入数据。 对于长度大于 64 字节的消息，会通过 HASHIOBUFCTRL 寄存器的标志进行握手，将多个数据块写入此输入缓冲区。除最后一个数据块外，所有数据块的大小都必须为 512 位。如果最后一个数据块的长度不是 512 位，则只能写入数据的最低有效位，但必须用 0 填充到下一个 32 位边界。 主机从这些寄存器地址读取数据的操作将返回 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.40 HASHDATAIN9 寄存器（偏移 = 624h）[复位 = 00000000h]

图 12-47 展示了 HASHDATAIN9，表 12-91 对其进行了介绍。

返回到汇总表。

HASH 数据输入 9
数据输入寄存器应该用于通过从机接口向哈希模块提供输入数据。

图 12-47. HASHDATAIN9 寄存器

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

表 12-91. HASHDATAIN9 寄存器字段说明

位	字段	类型	复位	说明
31-0	HASH_DATA_IN	W	0h	HASH_DATA_IN[319:288] 这些寄存器必须写入 512 位输入数据。数据线直接连接到哈希模块的数据输入，因此连接到引擎的内部数据缓冲区。对每个寄存器的写入都会触发内部缓冲器对应的 32 位写入使能信号。 注意：只有当 HASHIOBUFCTRL.RFD_IN 为 1 时，主机才能写入输入数据缓冲区。如果 HASHIOBUFCTRL.RFD_IN 为 0，则引擎正忙于处理。处理期间，不允许写入新的输入数据。 对于长度大于 64 字节的消息，会通过 HASHIOBUFCTRL 寄存器的标志进行握手，将多个数据块写入此输入缓冲区。除最后一个数据块外，所有数据块的大小都必须为 512 位。如果最后一个数据块的长度不是 512 位，则只能写入数据的最低有效位，但必须用 0 填充到下一个 32 位边界。 主机从这些寄存器地址读取数据的操作将返回 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.41 HASHDATAIN10 寄存器（偏移 = 628h）[复位 = 00000000h]

图 12-48 展示了 HASHDATAIN10，表 12-92 对其进行了介绍。

返回到汇总表。

HASH 数据输入 10
数据输入寄存器应该用于通过从机接口向哈希模块提供输入数据。

图 12-48. HASHDATAIN10 寄存器

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

表 12-92. HASHDATAIN10 寄存器字段说明

位	字段	类型	复位	说明
31-0	HASH_DATA_IN	W	0h	HASH_DATA_IN[351:320] 这些寄存器必须写入 512 位输入数据。数据线直接连接到哈希模块的数据输入，因此连接到引擎的内部数据缓冲区。对每个寄存器的写入都会触发内部缓冲器对应的 32 位写入使能信号。 注意：只有当 HASHIOBUFCTRL.RFD_IN 为 1 时，主机才能写入输入数据缓冲区。如果 HASHIOBUFCTRL.RFD_IN 为 0，则引擎正忙于处理。处理期间，不允许写入新的输入数据。 对于长度大于 64 字节的消息，会通过 HASHIOBUFCTRL 寄存器的标志进行握手，将多个数据块写入此输入缓冲区。除最后一个数据块外，所有数据块的大小都必须为 512 位。如果最后一个数据块的长度不是 512 位，则只能写入数据的最低有效位，但必须用 0 填充到下一个 32 位边界。 主机从这些寄存器地址读取数据的操作将返回 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.42 HASHDATAIN11 寄存器（偏移 = 62Ch）[复位 = 00000000h]

图 12-49 展示了 HASHDATAIN11，表 12-93 对其进行了介绍。

返回到汇总表。

HASH 数据输入 11
数据输入寄存器应该用于通过从机接口向哈希模块提供输入数据。

图 12-49. HASHDATAIN11 寄存器

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

表 12-93. HASHDATAIN11 寄存器字段说明

位	字段	类型	复位	说明
31-0	HASH_DATA_IN	W	0h	HASH_DATA_IN[383:352] 这些寄存器必须写入 512 位输入数据。数据线直接连接到哈希模块的数据输入，因此连接到引擎的内部数据缓冲区。对每个寄存器的写入都会触发内部缓冲器对应的 32 位写入使能信号。 注意：只有当 HASHIOBUFCTRL.RFD_IN 为 1 时，主机才能写入输入数据缓冲区。如果 HASHIOBUFCTRL.RFD_IN 为 0，则引擎正忙于处理。处理期间，不允许写入新的输入数据。 对于长度大于 64 字节的消息，会通过 HASHIOBUFCTRL 寄存器的标志进行握手，将多个数据块写入此输入缓冲区。除最后一个数据块外，所有数据块的大小都必须为 512 位。如果最后一个数据块的长度不是 512 位，则只能写入数据的最低有效位，但必须用 0 填充到下一个 32 位边界。 主机从这些寄存器地址读取数据的操作将返回 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.43 HASHDATAIN12 寄存器（偏移 = 630h）[复位 = 00000000h]

图 12-50 展示了 HASHDATAIN12，表 12-94 对其进行了介绍。

返回到汇总表。

HASH 数据输入 12
数据输入寄存器应该用于通过从机接口向哈希模块提供输入数据。

图 12-50. HASHDATAIN12 寄存器

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

表 12-94. HASHDATAIN12 寄存器字段说明

位	字段	类型	复位	说明
31-0	HASH_DATA_IN	W	0h	HASH_DATA_IN[415:384] 这些寄存器必须写入 512 位输入数据。数据线直接连接到哈希模块的数据输入，因此连接到引擎的内部数据缓冲区。对每个寄存器的写入都会触发内部缓冲器对应的 32 位写入使能信号。 注意：只有当 HASHIOBUFCTRL.RFD_IN 为 1 时，主机才能写入输入数据缓冲区。如果 HASHIOBUFCTRL.RFD_IN 为 0，则引擎正忙于处理。处理期间，不允许写入新的输入数据。 对于长度大于 64 字节的消息，会通过 HASHIOBUFCTRL 寄存器的标志进行握手，将多个数据块写入此输入缓冲区。除最后一个数据块外，所有数据块的大小都必须为 512 位。如果最后一个数据块的长度不是 512 位，则只能写入数据的最低有效位，但必须用 0 填充到下一个 32 位边界。 主机从这些寄存器地址读取数据的操作将返回 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.44 HASHDATAIN13 寄存器（偏移 = 634h）[复位 = 00000000h]

图 12-51 展示了 HASHDATAIN13，表 12-95 对其进行了介绍。

返回到汇总表。

HASH 数据输入 13
数据输入寄存器应该用于通过从机接口向哈希模块提供输入数据。

图 12-51. HASHDATAIN13 寄存器

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

表 12-95. HASHDATAIN13 寄存器字段说明

位	字段	类型	复位	说明
31-0	HASH_DATA_IN	W	0h	HASH_DATA_IN[447:416] 这些寄存器必须写入 512 位输入数据。数据线直接连接到哈希模块的数据输入，因此连接到引擎的内部数据缓冲区。对每个寄存器的写入都会触发内部缓冲器对应的 32 位写入使能信号。 注意：只有当 HASHIOBUFCTRL.RFD_IN 为 1 时，主机才能写入输入数据缓冲区。如果 HASHIOBUFCTRL.RFD_IN 为 0，则引擎正忙于处理。处理期间，不允许写入新的输入数据。 对于长度大于 64 字节的消息，会通过 HASHIOBUFCTRL 寄存器的标志进行握手，将多个数据块写入此输入缓冲区。除最后一个数据块外，所有数据块的大小都必须为 512 位。如果最后一个数据块的长度不是 512 位，则只能写入数据的最低有效位，但必须用 0 填充到下一个 32 位边界。 主机从这些寄存器地址读取数据的操作将返回 0。

Copyright © 2026 Texas Instruments Incorporated

12.9.1.45 HASHDATAIN14 寄存器（偏移 = 638h）[复位 = 00000000h]

图 12-52 展示了 HASHDATAIN14，表 12-96 对其进行了介绍。

返回到汇总表。

HASH 数据输入 14
数据输入寄存器应该用于通过从机接口向哈希模块提供输入数据。

图 12-52. HASHDATAIN14 寄存器

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

表 12-96. HASHDATAIN14 寄存器字段说明

位	字段	类型	复位	说明
31-0	HASH_DATA_IN	W	0h	HASH_DATA_IN[479:448] 这些寄存器必须写入 512 位输入数据。数据线直接连接到哈希模块的数据输入，因此连接到引擎的内部数据缓冲区。对每个寄存器的写入都会触发内部缓冲器对应的 32 位写入使能信号。 注意：只有当 HASHIOBUFCTRL.RFD_IN 为 1 时，主机才能写入输入数据缓冲区。如果 HASHIOBUFCTRL.RFD_IN 为 0，则引擎正忙于处理。处理期间，不允许写入新的输入数据。 对于长度大于 64 字节的消息，会通过 HASHIOBUFCTRL 寄存器的标志进行握手，将多个数据块写入此输入缓冲区。除最后一个数据块外，所有数据块的大小都必须为 512 位。如果最后一个数据块的长度不是 512 位，则只能写入数据的最低有效位，但必须用 0 填充到下一个 32 位边界。 主机从这些寄存器地址读取数据的操作将返回 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.46 HASHDATAIN15 寄存器（偏移 = 63Ch）[复位 = 00000000h]

图 12-53 展示了 HASHDATAIN15，表 12-97 对其进行了介绍。

返回到汇总表。

HASH 数据输入 15
数据输入寄存器应该用于通过从机接口向哈希模块提供输入数据。

图 12-53. HASHDATAIN15 寄存器

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

表 12-97. HASHDATAIN15 寄存器字段说明

位	字段	类型	复位	说明
31-0	HASH_DATA_IN	W	0h	HASH_DATA_IN[511:480] 这些寄存器必须写入 512 位输入数据。数据线直接连接到哈希模块的数据输入，因此连接到引擎的内部数据缓冲区。对每个寄存器的写入都会触发内部缓冲器对应的 32 位写入使能信号。 注意：只有当 HASHIOBUFCTRL.RFD_IN 为 1 时，主机才能写入输入数据缓冲区。如果 HASHIOBUFCTRL.RFD_IN 为 0，则引擎正忙于处理。处理期间，不允许写入新的输入数据。 对于长度大于 64 字节的消息，会通过 HASHIOBUFCTRL 寄存器的标志进行握手，将多个数据块写入此输入缓冲区。除最后一个数据块外，所有数据块的大小都必须为 512 位。如果最后一个数据块的长度不是 512 位，则只能写入数据的最低有效位，但必须用 0 填充到下一个 32 位边界。 主机从这些寄存器地址读取数据的操作将返回 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.47 HASHDATAIN16 寄存器（偏移 = 640h）[复位 = 00000000h]

图 12-54 展示了 HASHDATAIN16，表 12-98 对其进行了介绍。

返回到汇总表。

HASH 数据输入 16
数据输入寄存器应该用于通过从机接口向哈希模块提供输入数据。

图 12-54. HASHDATAIN16 寄存器

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

表 12-98. HASHDATAIN16 寄存器字段说明

位	字段	类型	复位	说明
31-0	HASH_DATA_IN	W	0h	HASH_DATA_IN[543:512] 这些寄存器必须写入 512 位输入数据。数据线直接连接到哈希模块的数据输入，因此连接到引擎的内部数据缓冲区。对每个寄存器的写入都会触发内部缓冲器对应的 32 位写入使能信号。 注意：只有当 HASHIOBUFCTRL.RFD_IN 为 1 时，主机才能写入输入数据缓冲区。如果 HASHIOBUFCTRL.RFD_IN 为 0，则引擎正忙于处理。处理期间，不允许写入新的输入数据。 对于长度大于 64 字节的消息，会通过 HASHIOBUFCTRL 寄存器的标志进行握手，将多个数据块写入此输入缓冲区。除最后一个数据块外，所有数据块的大小都必须为 512 位。如果最后一个数据块的长度不是 512 位，则只能写入数据的最低有效位，但必须用 0 填充到下一个 32 位边界。 主机从这些寄存器地址读取数据的操作将返回 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.48 HASHDATAIN17 寄存器（偏移 = 644h）[复位 = 00000000h]

图 12-55 展示了 HASHDATAIN17，表 12-99 对其进行了介绍。

返回到汇总表。

HASH 数据输入 17
数据输入寄存器应该用于通过从机接口向哈希模块提供输入数据。

图 12-55. HASHDATAIN17 寄存器

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

表 12-99. HASHDATAIN17 寄存器字段说明

位	字段	类型	复位	说明
31-0	HASH_DATA_IN	W	0h	HASH_DATA_IN[575:544] 这些寄存器必须写入 512 位输入数据。数据线直接连接到哈希模块的数据输入，因此连接到引擎的内部数据缓冲区。对每个寄存器的写入都会触发内部缓冲器对应的 32 位写入使能信号。 注意：只有当 HASHIOBUFCTRL.RFD_IN 为 1 时，主机才能写入输入数据缓冲区。如果 HASHIOBUFCTRL.RFD_IN 为 0，则引擎正忙于处理。处理期间，不允许写入新的输入数据。 对于长度大于 64 字节的消息，会通过 HASHIOBUFCTRL 寄存器的标志进行握手，将多个数据块写入此输入缓冲区。除最后一个数据块外，所有数据块的大小都必须为 512 位。如果最后一个数据块的长度不是 512 位，则只能写入数据的最低有效位，但必须用 0 填充到下一个 32 位边界。 主机从这些寄存器地址读取数据的操作将返回 0。

Copyright © 2026 Texas Instruments Incorporated

12.9.1.49 HASHDATAIN18 寄存器（偏移 = 648h）[复位 = 00000000h]

图 12-56 展示了 HASHDATAIN18，表 12-100 对其进行了介绍。

返回到汇总表。

HASH 数据输入 18
数据输入寄存器应该用于通过从机接口向哈希模块提供输入数据。

图 12-56. HASHDATAIN18 寄存器

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

表 12-100. HASHDATAIN18 寄存器字段说明

位	字段	类型	复位	说明
31-0	HASH_DATA_IN	W	0h	HASH_DATA_IN[607:576] 这些寄存器必须写入 512 位输入数据。数据线直接连接到哈希模块的数据输入，因此连接到引擎的内部数据缓冲区。对每个寄存器的写入都会触发内部缓冲器对应的 32 位写入使能信号。 注意：只有当 HASHIOBUFCTRL.RFD_IN 为 1 时，主机才能写入输入数据缓冲区。如果 HASHIOBUFCTRL.RFD_IN 为 0，则引擎正忙于处理。处理期间，不允许写入新的输入数据。 对于长度大于 64 字节的消息，会通过 HASHIOBUFCTRL 寄存器的标志进行握手，将多个数据块写入此输入缓冲区。除最后一个数据块外，所有数据块的大小都必须为 512 位。如果最后一个数据块的长度不是 512 位，则只能写入数据的最低有效位，但必须用 0 填充到下一个 32 位边界。 主机从这些寄存器地址读取数据的操作将返回 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.50 HASHDATAIN19 寄存器（偏移 = 64Ch）[复位 = 00000000h]

图 12-57 展示了 HASHDATAIN19，表 12-101 对其进行了介绍。

返回到汇总表。

HASH 数据输入 19
数据输入寄存器应该用于通过从机接口向哈希模块提供输入数据。

图 12-57. HASHDATAIN19 寄存器

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

表 12-101. HASHDATAIN19 寄存器字段说明

位	字段	类型	复位	说明
31-0	HASH_DATA_IN	W	0h	HASH_DATA_IN[639:608] 这些寄存器必须写入 512 位输入数据。数据线直接连接到哈希模块的数据输入，因此连接到引擎的内部数据缓冲区。对每个寄存器的写入都会触发内部缓冲器对应的 32 位写入使能信号。 注意：只有当 HASHIOBUFCTRL.RFD_IN 为 1 时，主机才能写入输入数据缓冲区。如果 HASHIOBUFCTRL.RFD_IN 为 0，则引擎正忙于处理。处理期间，不允许写入新的输入数据。 对于长度大于 64 字节的消息，会通过 HASHIOBUFCTRL 寄存器的标志进行握手，将多个数据块写入此输入缓冲区。除最后一个数据块外，所有数据块的大小都必须为 512 位。如果最后一个数据块的长度不是 512 位，则只能写入数据的最低有效位，但必须用 0 填充到下一个 32 位边界。 主机从这些寄存器地址读取数据的操作将返回 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.51 HASHDATAIN20 寄存器（偏移 = 650h）[复位 = 00000000h]

图 12-58 展示了 HASHDATAIN20，表 12-102 对其进行了介绍。

返回到汇总表。

HASH 数据输入 20
数据输入寄存器应该用于通过从机接口向哈希模块提供输入数据。

图 12-58. HASHDATAIN20 寄存器

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

表 12-102. HASHDATAIN20 寄存器字段说明

位	字段	类型	复位	说明
31-0	HASH_DATA_IN	W	0h	HASH_DATA_IN[671:640] 这些寄存器必须写入 512 位输入数据。数据线直接连接到哈希模块的数据输入，因此连接到引擎的内部数据缓冲区。对每个寄存器的写入都会触发内部缓冲器对应的 32 位写入使能信号。 注意：只有当 HASHIOBUFCTRL.RFD_IN 为 1 时，主机才能写入输入数据缓冲区。如果 HASHIOBUFCTRL.RFD_IN 为 0，则引擎正忙于处理。处理期间，不允许写入新的输入数据。 对于长度大于 64 字节的消息，会通过 HASHIOBUFCTRL 寄存器的标志进行握手，将多个数据块写入此输入缓冲区。除最后一个数据块外，所有数据块的大小都必须为 512 位。如果最后一个数据块的长度不是 512 位，则只能写入数据的最低有效位，但必须用 0 填充到下一个 32 位边界。 主机从这些寄存器地址读取数据的操作将返回 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.52 HASHDATAIN21 寄存器（偏移 = 654h）[复位 = 00000000h]

图 12-59 展示了 HASHDATAIN21，表 12-103 对其进行了介绍。

返回到汇总表。

HASH 数据输入 21
数据输入寄存器应该用于通过从机接口向哈希模块提供输入数据。

图 12-59. HASHDATAIN21 寄存器

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

表 12-103. HASHDATAIN21 寄存器字段说明

位	字段	类型	复位	说明
31-0	HASH_DATA_IN	W	0h	HASH_DATA_IN[703:672] 这些寄存器必须写入 512 位输入数据。数据线直接连接到哈希模块的数据输入，因此连接到引擎的内部数据缓冲区。对每个寄存器的写入都会触发内部缓冲器对应的 32 位写入使能信号。 注意：只有当 HASHIOBUFCTRL.RFD_IN 为 1 时，主机才能写入输入数据缓冲区。如果 HASHIOBUFCTRL.RFD_IN 为 0，则引擎正忙于处理。处理期间，不允许写入新的输入数据。 对于长度大于 64 字节的消息，会通过 HASHIOBUFCTRL 寄存器的标志进行握手，将多个数据块写入此输入缓冲区。除最后一个数据块外，所有数据块的大小都必须为 512 位。如果最后一个数据块的长度不是 512 位，则只能写入数据的最低有效位，但必须用 0 填充到下一个 32 位边界。 主机从这些寄存器地址读取数据的操作将返回 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.53 HASHDATAIN22 寄存器（偏移 = 658h）[复位 = 00000000h]

图 12-60 展示了 HASHDATAIN22，表 12-104 对其进行了介绍。

返回到汇总表。

HASH 数据输入 22
数据输入寄存器应该用于通过从机接口向哈希模块提供输入数据。

图 12-60. HASHDATAIN22 寄存器

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

表 12-104. HASHDATAIN22 寄存器字段说明

位	字段	类型	复位	说明
31-0	HASH_DATA_IN	W	0h	HASH_DATA_IN[735:704] 这些寄存器必须写入 512 位输入数据。数据线直接连接到哈希模块的数据输入，因此连接到引擎的内部数据缓冲区。对每个寄存器的写入都会触发内部缓冲器对应的 32 位写入使能信号。 注意：只有当 HASHIOBUFCTRL.RFD_IN 为 1 时，主机才能写入输入数据缓冲区。如果 HASHIOBUFCTRL.RFD_IN 为 0，则引擎正忙于处理。处理期间，不允许写入新的输入数据。 对于长度大于 64 字节的消息，会通过 HASHIOBUFCTRL 寄存器的标志进行握手，将多个数据块写入此输入缓冲区。除最后一个数据块外，所有数据块的大小都必须为 512 位。如果最后一个数据块的长度不是 512 位，则只能写入数据的最低有效位，但必须用 0 填充到下一个 32 位边界。 主机从这些寄存器地址读取数据的操作将返回 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.54 HASHDATAIN23 寄存器（偏移 = 65Ch）[复位 = 00000000h]

图 12-61 展示了 HASHDATAIN23，表 12-105 对其进行了介绍。

返回到汇总表。

HASH 数据输入 23
数据输入寄存器应该用于通过从机接口向哈希模块提供输入数据。

图 12-61. HASHDATAIN23 寄存器

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

表 12-105. HASHDATAIN23 寄存器字段说明

位	字段	类型	复位	说明
31-0	HASH_DATA_IN	W	0h	HASH_DATA_IN[767:736] 这些寄存器必须写入 512 位输入数据。数据线直接连接到哈希模块的数据输入，因此连接到引擎的内部数据缓冲区。对每个寄存器的写入都会触发内部缓冲器对应的 32 位写入使能信号。 注意：只有当 HASHIOBUFCTRL.RFD_IN 为 1 时，主机才能写入输入数据缓冲区。如果 HASHIOBUFCTRL.RFD_IN 为 0，则引擎正忙于处理。处理期间，不允许写入新的输入数据。 对于长度大于 64 字节的消息，会通过 HASHIOBUFCTRL 寄存器的标志进行握手，将多个数据块写入此输入缓冲区。除最后一个数据块外，所有数据块的大小都必须为 512 位。如果最后一个数据块的长度不是 512 位，则只能写入数据的最低有效位，但必须用 0 填充到下一个 32 位边界。 主机从这些寄存器地址读取数据的操作将返回 0。

Copyright © 2026 Texas Instruments Incorporated

12.9.1.55 HASHDATAIN24 寄存器（偏移 = 660h）[复位 = 00000000h]

图 12-62 展示了 HASHDATAIN24，表 12-106 对其进行了介绍。

返回到汇总表。

HASH 数据输入 24
数据输入寄存器应该用于通过从机接口向哈希模块提供输入数据。

图 12-62. HASHDATAIN24 寄存器

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

表 12-106. HASHDATAIN24 寄存器字段说明

位	字段	类型	复位	说明
31-0	HASH_DATA_IN	W	0h	HASH_DATA_IN[799:768] 这些寄存器必须写入 512 位输入数据。数据线直接连接到哈希模块的数据输入，因此连接到引擎的内部数据缓冲区。对每个寄存器的写入都会触发内部缓冲器对应的 32 位写入使能信号。 注意：只有当 HASHIOBUFCTRL.RFD_IN 为 1 时，主机才能写入输入数据缓冲区。如果 HASHIOBUFCTRL.RFD_IN 为 0，则引擎正忙于处理。处理期间，不允许写入新的输入数据。 对于长度大于 64 字节的消息，会通过 HASHIOBUFCTRL 寄存器的标志进行握手，将多个数据块写入此输入缓冲区。除最后一个数据块外，所有数据块的大小都必须为 512 位。如果最后一个数据块的长度不是 512 位，则只能写入数据的最低有效位，但必须用 0 填充到下一个 32 位边界。 主机从这些寄存器地址读取数据的操作将返回 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.56 HASHDATAIN25 寄存器（偏移 = 664h）[复位 = 00000000h]

图 12-63 展示了 HASHDATAIN25，表 12-107 对其进行了介绍。

返回到汇总表。

HASH 数据输入 25
数据输入寄存器应该用于通过从机接口向哈希模块提供输入数据。

图 12-63. HASHDATAIN25 寄存器

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

表 12-107. HASHDATAIN25 寄存器字段说明

位	字段	类型	复位	说明
31-0	HASH_DATA_IN	W	0h	HASH_DATA_IN[831:800] 这些寄存器必须写入 512 位输入数据。数据线直接连接到哈希模块的数据输入，因此连接到引擎的内部数据缓冲区。对每个寄存器的写入都会触发内部缓冲器对应的 32 位写入使能信号。 注意：只有当 HASHIOBUFCTRL.RFD_IN 为 1 时，主机才能写入输入数据缓冲区。如果 HASHIOBUFCTRL.RFD_IN 为 0，则引擎正忙于处理。处理期间，不允许写入新的输入数据。 对于长度大于 64 字节的消息，会通过 HASHIOBUFCTRL 寄存器的标志进行握手，将多个数据块写入此输入缓冲区。除最后一个数据块外，所有数据块的大小都必须为 512 位。如果最后一个数据块的长度不是 512 位，则只能写入数据的最低有效位，但必须用 0 填充到下一个 32 位边界。 主机从这些寄存器地址读取数据的操作将返回 0。

Copyright © 2026 Texas Instruments Incorporated

12.9.1.57 HASHDATAIN26 寄存器（偏移 = 668h）[复位 = 00000000h]

图 12-64 展示了 HASHDATAIN26，表 12-108 对其进行了介绍。

返回到汇总表。

HASH 数据输入 26
数据输入寄存器应该用于通过从机接口向哈希模块提供输入数据。

图 12-64. HASHDATAIN26 寄存器

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

表 12-108. HASHDATAIN26 寄存器字段说明

位	字段	类型	复位	说明
31-0	HASH_DATA_IN	W	0h	HASH_DATA_IN[863:832] 这些寄存器必须写入 512 位输入数据。数据线直接连接到哈希模块的数据输入，因此连接到引擎的内部数据缓冲区。对每个寄存器的写入都会触发内部缓冲器对应的 32 位写入使能信号。 注意：只有当 HASHIOBUFCTRL.RFD_IN 为 1 时，主机才能写入输入数据缓冲区。如果 HASHIOBUFCTRL.RFD_IN 为 0，则引擎正忙于处理。处理期间，不允许写入新的输入数据。 对于长度大于 64 字节的消息，会通过 HASHIOBUFCTRL 寄存器的标志进行握手，将多个数据块写入此输入缓冲区。除最后一个数据块外，所有数据块的大小都必须为 512 位。如果最后一个数据块的长度不是 512 位，则只能写入数据的最低有效位，但必须用 0 填充到下一个 32 位边界。 主机从这些寄存器地址读取数据的操作将返回 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.58 HASHDATAIN27 寄存器（偏移 = 66Ch）[复位 = 00000000h]

图 12-65 展示了 HASHDATAIN27，表 12-109 对其进行了介绍。

返回到汇总表。

HASH 数据输入 27
数据输入寄存器应该用于通过从机接口向哈希模块提供输入数据。

图 12-65. HASHDATAIN27 寄存器

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

表 12-109. HASHDATAIN27 寄存器字段说明

位	字段	类型	复位	说明
31-0	HASH_DATA_IN	W	0h	HASH_DATA_IN[895:864] 这些寄存器必须写入 512 位输入数据。数据线直接连接到哈希模块的数据输入，因此连接到引擎的内部数据缓冲区。对每个寄存器的写入都会触发内部缓冲器对应的 32 位写入使能信号。 注意：只有当 HASHIOBUFCTRL.RFD_IN 为 1 时，主机才能写入输入数据缓冲区。如果 HASHIOBUFCTRL.RFD_IN 为 0，则引擎正忙于处理。处理期间，不允许写入新的输入数据。 对于长度大于 64 字节的消息，会通过 HASHIOBUFCTRL 寄存器的标志进行握手，将多个数据块写入此输入缓冲区。除最后一个数据块外，所有数据块的大小都必须为 512 位。如果最后一个数据块的长度不是 512 位，则只能写入数据的最低有效位，但必须用 0 填充到下一个 32 位边界。 主机从这些寄存器地址读取数据的操作将返回 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.59 HASHDATAIN28 寄存器（偏移 = 670h）[复位 = 00000000h]

图 12-66 展示了 HASHDATAIN28，表 12-110 对其进行了介绍。

返回到汇总表。

HASH 数据输入 28
数据输入寄存器应该用于通过从机接口向哈希模块提供输入数据。

图 12-66. HASHDATAIN28 寄存器

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

表 12-110. HASHDATAIN28 寄存器字段说明

位	字段	类型	复位	说明
31-0	HASH_DATA_IN	W	0h	HASH_DATA_IN[923:896] 这些寄存器必须写入 512 位输入数据。数据线直接连接到哈希模块的数据输入，因此连接到引擎的内部数据缓冲区。对每个寄存器的写入都会触发内部缓冲器对应的 32 位写入使能信号。 注意：只有当 HASHIOBUFCTRL.RFD_IN 为 1 时，主机才能写入输入数据缓冲区。如果 HASHIOBUFCTRL.RFD_IN 为 0，则引擎正忙于处理。处理期间，不允许写入新的输入数据。 对于长度大于 64 字节的消息，会通过 HASHIOBUFCTRL 寄存器的标志进行握手，将多个数据块写入此输入缓冲区。除最后一个数据块外，所有数据块的大小都必须为 512 位。如果最后一个数据块的长度不是 512 位，则只能写入数据的最低有效位，但必须用 0 填充到下一个 32 位边界。 主机从这些寄存器地址读取数据的操作将返回 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.60 HASHDATAIN29 寄存器（偏移 = 674h）[复位 = 00000000h]

图 12-67 展示了 HASHDATAIN29，表 12-111 对其进行了介绍。

返回到汇总表。

HASH 数据输入 29
数据输入寄存器应该用于通过从机接口向哈希模块提供输入数据。

图 12-67. HASHDATAIN29 寄存器

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

表 12-111. HASHDATAIN29 寄存器字段说明

位	字段	类型	复位	说明
31-0	HASH_DATA_IN	W	0h	HASH_DATA_IN[959:924] 这些寄存器必须写入 512 位输入数据。数据线直接连接到哈希模块的数据输入，因此连接到引擎的内部数据缓冲区。对每个寄存器的写入都会触发内部缓冲器对应的 32 位写入使能信号。 注意：只有当 HASHIOBUFCTRL.RFD_IN 为 1 时，主机才能写入输入数据缓冲区。如果 HASHIOBUFCTRL.RFD_IN 为 0，则引擎正忙于处理。处理期间，不允许写入新的输入数据。 对于长度大于 64 字节的消息，会通过 HASHIOBUFCTRL 寄存器的标志进行握手，将多个数据块写入此输入缓冲区。除最后一个数据块外，所有数据块的大小都必须为 512 位。如果最后一个数据块的长度不是 512 位，则只能写入数据的最低有效位，但必须用 0 填充到下一个 32 位边界。 主机从这些寄存器地址读取数据的操作将返回 0。

Copyright © 2026 Texas Instruments Incorporated

12.9.1.61 HASHDATAIN30 寄存器（偏移 = 678h）[复位 = 00000000h]

图 12-68 展示了 HASHDATAIN30，表 12-112 对其进行了介绍。

返回到汇总表。

HASH 数据输入 30
数据输入寄存器应该用于通过从机接口向哈希模块提供输入数据。

图 12-68. HASHDATAIN30 寄存器

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

表 12-112. HASHDATAIN30 寄存器字段说明

位	字段	类型	复位	说明
31-0	HASH_DATA_IN	W	0h	HASH_DATA_IN[991:960] 这些寄存器必须写入 512 位输入数据。数据线直接连接到哈希模块的数据输入，因此连接到引擎的内部数据缓冲区。对每个寄存器的写入都会触发内部缓冲器对应的 32 位写入使能信号。 注意：只有当 HASHIOBUFCTRL.RFD_IN 为 1 时，主机才能写入输入数据缓冲区。如果 HASHIOBUFCTRL.RFD_IN 为 0，则引擎正忙于处理。处理期间，不允许写入新的输入数据。 对于长度大于 64 字节的消息，会通过 HASHIOBUFCTRL 寄存器的标志进行握手，将多个数据块写入此输入缓冲区。除最后一个数据块外，所有数据块的大小都必须为 512 位。如果最后一个数据块的长度不是 512 位，则只能写入数据的最低有效位，但必须用 0 填充到下一个 32 位边界。 主机从这些寄存器地址读取数据的操作将返回 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.62 HASHDATAIN31 寄存器（偏移 = 67Ch）[复位 = 00000000h]

图 12-69 展示了 HASHDATAIN31，表 12-113 对其进行了介绍。

返回到汇总表。

HASH 数据输入 31
数据输入寄存器应该用于通过从机接口向哈希模块提供输入数据。

图 12-69. HASHDATAIN31 寄存器

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

表 12-113. HASHDATAIN31 寄存器字段说明

位	字段	类型	复位	说明
31-0	HASH_DATA_IN	W	0h	HASH_DATA_IN[1023:992] 这些寄存器必须写入 512 位输入数据。数据线直接连接到哈希模块的数据输入，因此连接到引擎的内部数据缓冲区。对每个寄存器的写入都会触发内部缓冲器对应的 32 位写入使能信号。 注意：只有当 HASHIOBUFCTRL.RFD_IN 为 1 时，主机才能写入输入数据缓冲区。如果 HASHIOBUFCTRL.RFD_IN 为 0，则引擎正忙于处理。处理期间，不允许写入新的输入数据。 对于长度大于 64 字节的消息，会通过 HASHIOBUFCTRL 寄存器的标志进行握手，将多个数据块写入此输入缓冲区。除最后一个数据块外，所有数据块的大小都必须为 512 位。如果最后一个数据块的长度不是 512 位，则只能写入数据的最低有效位，但必须用 0 填充到下一个 32 位边界。 主机从这些寄存器地址读取数据的操作将返回 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.63 HASHIOBUFCTRL 寄存器（偏移 = 680h）[复位 = 00000004h]

图 12-70 展示了 HASHIOBUFCTRL，表 12-114 中对此进行了介绍。

返回到汇总表。

HASH Input_Output 缓冲区控制
这对寄存器共享同一个地址，包含用于控制和监视主机与哈希引擎之间数据流的位。

图 12-70. HASHIOBUFCTRL 寄存器

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

表 12-114. HASHIOBUFCTRL 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7	PAD_DMA_MESSAGE	R/W	0h	注意：此位仅当数据通过 DMA 传输时使用。当数据通过从设备接口传输时，不应使用此位。 此位指示哈希引擎是否需要对通过 DMA 接收的消息进行填充并最终确定哈希值。 当设置为 1 时，哈希引擎会使用设定的长度填充最后一个数据块。填充完成后，计算最终的哈希结果。 当设置为 0 时，哈希引擎会将最后一个写入的数据块视为按块大小对齐，并计算中间摘要。 当最后一个 DMA 数据块到达哈希引擎时，系统会自动清除此位。
6	GET_DIGEST	R/W	0h	注意：以下位描述仅适用于通过从机接口发送数据的情况。当通过 DMA 接收数据时，此位必须设置为 0。 此位指示哈希引擎是否应提供哈希摘要。 如果与 data_in_av 同时提供，则在处理完 HASHDATAINn 寄存器中当前存储的数据后提供哈希摘要。如果未与 data_in_av 同时提供，则将当前内部摘要缓冲区的值复制到 HASHDIGESTn 寄存器。 主机必须将此位写入 1 才能使中间哈希摘要可用。 将此位写入 0 无效。 当哈希引擎处理完 HASHDATAINn 寄存器的内容后，此位会自动清除（即读数为 0）。在主机设置此位到实际处理 HASHDATAINn 寄存器之间的这段时间内，此位读数为 1。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 12-114. HASHIOBUFCTRL 寄存器字段说明（续）

位	字段	类型	复位	说明
5	PAD_MESSAGE	R/W	0h	注意：以下位描述仅适用于通过从机接口发送数据的情况。当通过 DMA 接收数据时，此位必须设置为 0。 此位表示 HASHDATAINn 寄存器中保存的是消息的最后数据，需要对该数据应用哈希填充。 主机必须将此位写入 1，以指示哈希引擎 HASHDATAINn 寄存器当前保存着消息的最后一个数据。当 pad_message 置为 1 时，哈希引擎会对 HASHDATAINn 寄存器中当前的数据添加填充位。 当最后一个消息块小于 512 位时，pad_message 位必须与 data_in_av 位一起置为 1。 当最后一个消息块等于 512 位时，pad_message 可以与 data_in_av 一起置为 1。在这种情况下，也可以在最后一个数据块写入哈希引擎之后再置位 pad_message（即在写完最后一个数据块后，当 rfd_in 位再次变为 1 时）。 向该位写入 0 无任何影响。 该位会被哈希引擎自动清零（即读取值为 0）。在主机设置此位之后，直到哈希引擎根据其值解释之前，此位的读取值为 1。
4-3	RESERVED	R/W	0h	写入 0 并在读取时忽略
2	RFD_IN	R/W	1h	注意：以下位描述仅适用于通过从机接口发送数据的情况。当数据通过 DMA 接收时，可以忽略该位。 该位为只读，用于表示哈希引擎输入缓冲区的状态。 当该位为 1 时，表示哈希引擎的输入缓冲区可以接收新的数据 因而可以安全地向 HASHDATAINn 寄存器填入新的数据。 当该位为 0 时，表示哈希引擎正在处理当前位于 HASHDATAINn 中的数据 此时不允许向这些寄存器写入新数据。
1	DATA_IN_AV	R/W	0h	注意：以下位描述仅适用于通过从机接口发送数据的情况。当数据通过 DMA 接收时，必须将此位置为 0。 此位表示 HASHDATAINn 寄存器中包含新的输入数据，等待哈希引擎进行处理。 主机必须将此位置为 1，以开始处理 HASHDATAINn 中的数据 哈希引擎将在其准备好接收时（rfd_in 位为 1）立即处理这些新数据。 向该位写入 0 无任何效果。 当哈希引擎开始处理 HASHDATAINn 的内容时，此位会被自动清零（即读取值为 0）。在主机设置此位之后，直到哈希引擎实际开始处理输入数据块之前，此位的读取值为 1。
0	OUTPUT_FULL	R/W	0h	表示输出缓冲寄存器 (HASHDIGESTn) 已可供主机读取。 当该位读数为 0 时，输出缓冲寄存器即被释放 允许哈希引擎向其写入新数据。在此情况下，主机不应读取这些寄存器。 当该位读数为 1 时，表明哈希引擎已将最新哈希操作的结果存储在输出缓冲寄存器中。只要该位读数为 1，主机就可以读取输出缓冲寄存器，并阻止哈希引擎将新数据写入输出缓冲区。 从输出缓冲区获取哈希结果数据后，主机必须向此位写入 1 以将其清除。这使得摘要输出缓冲区可供哈希引擎用于存储新的哈希结果。 向该位写入 0 无效。 注意：如果此位被置为有效 (1)，则在从哈希引擎检索到摘要且此位被清零 (0) 之前，不应启动任何新操作。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.64 HASHMODE 寄存器（偏移 = 684h）[复位 = 00000000h]

图 12-71 展示了 HASHMODE，表 12-115 中对此进行了介绍。

返回到汇总表。

HASH 模式

图 12-71. HASHMODE 寄存器

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	SHA384_MODE	SHA512_MODE	SHA224_MODE	SHA256_MODE	RESERVED		NEW_HASH
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h		W-0h

表 12-115. HASHMODE 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	W	0h	写入 0 并在读取时忽略
6	SHA384_MODE	W	0h	主机在处理 SHA 384 会话之前必须将此位写入 1。
5	SHA512_MODE	W	0h	主机在处理 SHA 512 会话之前必须将此位写入 1。
4	SHA224_MODE	W	0h	主机在处理 SHA 224 会话之前必须将此位写入 1。
3	SHA256_MODE	W	0h	主机在处理 SHA 256 会话之前必须将此位写入 1。
2-1	RESERVED	W	0h	写入 0 并在读取时忽略
0	NEW_HASH	W	0h	设置为 1 时，表示哈希引擎必须开始处理新的哈希会话。 [HASHDIGESTn.*] 寄存器将自动加载所选哈希算法的初始哈希算法常量。 当哈希处理开始时，如果此位为 0，则初始哈希算法常量不会加载到 HASHDIGESTn 寄存器中。哈希引擎将使用其内部 HASHDIGESTn 寄存器中当前存储的摘要开始处理。 哈希处理开始时，系统会自动清除此位。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.65 HASHINLENL 寄存器（偏移 = 688h）[复位 = 00000000h]

图 12-72 展示了 HASHINLENL，表 12-116 中对此进行了介绍。

返回到汇总表。

HASH 输入长度 LSB

图 12-72. HASHINLENL 寄存器

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

表 12-116. HASHINLENL 寄存器字段说明

位	字段	类型	复位	说明
31-0	LENGTH_IN	W	0h	LENGTH_IN[31:0] 消息长度寄存器。在哈希会话的消息填充阶段，哈希引擎会使用这些寄存器中的内容。这些寄存器的数据线直接连接到哈希引擎的接口。对于主机执行的写入操作，应向这些寄存器写入以位为单位的消息长度。 最终哈希操作： 对于需要进行最终化处理（即填充）的新哈希操作，必须配置输入数据的总长度。输入数据必须通过从机接口或 DMA 接口提供。 连续哈希操作（最终化处理）： 对于需要进行最终化处理的连续哈希操作，必须配置完整的消息总长度，其中应包含与当前写入的输入摘要相对应的、此前已完成哈希处理的数据长度。 非最终哈希操作： 对于无需进行最终化处理的哈希操作（即输入数据长度恰好是 512 位——即 SHA-256 算法的数据块大小——的整数倍），无需配置长度字段，因为此类操作并不使用该字段。 如果以位为单位的消息长度小于 (2^{32} - 1)，则仅需写入当前寄存器即可。在此情况下，硬件会自动将 HASH_LENGTH_IN_H 寄存器的值清零。 在哈希会话期间，只要 HASHIOBUFCTRL.RFD_IN 标志位处于高电平状态，主机便可随时向长度寄存器执行写入操作。在将当前活跃哈希会话的最后一段数据写入哈希引擎之前，必须先完成对长度寄存器的写入操作。 若主机尝试从这些寄存器地址读取数据，读取结果将始终为 0。 注意：从 DMA 获取数据时，必须在对 DMA 进行启动编程之前，对该寄存器进行编程。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.66 HASHINLENH 寄存器（偏移 = 68Ch）[复位 = 00000000h]

图 12-73 展示了 HASHINLENH，表 12-117 中对此进行了介绍。

返回到汇总表。

HASH 输入长度 MSB

图 12-73. HASHINLENH 寄存器

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

表 12-117. HASHINLENH 寄存器字段说明

位	字段	类型	复位	说明
31-0	LENGTH_IN	W	0h	LENGTH_IN[63:32] 消息长度寄存器。在哈希会话的消息填充阶段，哈希引擎会使用这些寄存器中的内容。这些寄存器的数据线直接连接到哈希引擎的接口。对于主机执行的写入操作，应向这些寄存器写入以位为单位的消息长度。 最终哈希操作： 对于需要进行最终化处理（即填充）的新哈希操作，必须配置输入数据的总长度。输入数据必须通过从机接口或 DMA 接口提供。 连续哈希操作（最终化处理）： 对于需要进行最终化处理的连续哈希操作，必须配置完整的消息总长度，其中应包含与当前写入的输入摘要相对应的、此前已完成哈希处理的数据长度。 非最终哈希操作： 对于无需进行最终化处理的哈希操作（即输入数据长度恰好是 512 位——即 SHA-256 算法的数据块大小——的整数倍），无需配置长度字段，因为此类操作并不使用该字段。 如果以位为单位的消息长度小于 (2^{32} - 1)，则仅需写入 HASHINLENL 寄存器即可。在此情况下，硬件会自动将 HASH_LENGTH_IN_H 寄存器的值清零。 在哈希会话期间，只要 HASHIOBUFCTRL.RFD_IN 标志位处于高电平状态，主机便可随时向长度寄存器执行写入操作。在将当前活跃哈希会话的最后一段数据写入哈希引擎之前，必须先完成对长度寄存器的写入操作。 若主机尝试从这些寄存器地址读取数据，读取结果将始终为 0。 注意：从 DMA 获取数据时，必须在对 DMA 进行启动编程之前，对该寄存器进行编程。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.67 HASHDIGESTA 寄存器（偏移 = 6C0h）[复位 = 00000000h]

图 12-74 展示了 HASHDIGESTA，表 12-118 中对此进行了介绍。

返回到汇总表。

哈希摘要 A
哈希摘要寄存器由八个 32 位寄存器组成，命名为 HASH_DIGEST_A 至 HASH_DIGEST_H。处理一条消息后，可以从这些寄存器中读取输出摘要。可向这些寄存器写入中间哈希结果，以进行后续的哈希操作。

图 12-74. HASHDIGESTA 寄存器

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

表 12-118. HASHDIGESTA 寄存器字段说明

位	字段	类型	复位	说明
31-0	HASH_DIGEST	R/W	0h	HASH_DIGEST[31:0] 哈希摘要寄存器 写入操作： 续接哈希： 在启动续接哈希会话（即启动会话时 HASHMODE.NEW_HASH 位为 0）之前，必须向这些寄存器写入上下文数据。 新建哈希： 在启动新建哈希会话（即 HASHMODE.NEW_HASH 位为 1）时，内部摘要寄存器会自动设置为 SHA-256 算法的预设常数，此时不应向这些寄存器进行写入操作。 读取这些寄存器，可在数据处理完成后，获取中间哈希结果（非最终哈希操作）或最终哈希结果（最终哈希操作）。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.68 HASHDIGESTB 寄存器（偏移 = 6C4h）[复位 = 00000000h]

图 12-75 展示了 HASHDIGESTB，表 12-119 中对此进行了介绍。

返回到汇总表。

哈希摘要 B
哈希摘要寄存器由八个 32 位寄存器组成，命名为 HASH_DIGEST_A 至 HASH_DIGEST_H。处理一条消息后，可以从这些寄存器中读取输出摘要。可向这些寄存器写入中间哈希结果，以进行后续的哈希操作。

图 12-75. HASHDIGESTB 寄存器

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

表 12-119. HASHDIGESTB 寄存器字段说明

位	字段	类型	复位	说明
31-0	HASH_DIGEST	R/W	0h	HASH_DIGEST[63:32] 哈希摘要寄存器 写入操作： 续接哈希： 在启动续接哈希会话（即启动会话时 HASHMODE.NEW_HASH 位为 0）之前，必须向这些寄存器写入上下文数据。 新建哈希： 在启动新建哈希会话（即 HASHMODE.NEW_HASH 位为 1）时，内部摘要寄存器会自动设置为 SHA-256 算法的预设常数，此时不应向这些寄存器进行写入操作。 读取这些寄存器，可在数据处理完成后，获取中间哈希结果（非最终哈希操作）或最终哈希结果（最终哈希操作）。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.69 HASHDIGESTC 寄存器（偏移 = 6C8h）[复位 = 00000000h]

图 12-76 展示了 HASHDIGESTC，表 12-120 中对此进行了介绍。

返回到汇总表。

哈希摘要 C
哈希摘要寄存器由八个 32 位寄存器组成，命名为 HASH_DIGEST_A 至 HASH_DIGEST_H。处理一条消息后，可以从这些寄存器中读取输出摘要。可向这些寄存器写入中间哈希结果，以进行后续的哈希操作。

图 12-76. HASHDIGESTC 寄存器

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

表 12-120. HASHDIGESTC 寄存器字段说明

位	字段	类型	复位	说明
31-0	HASH_DIGEST	R/W	0h	HASH_DIGEST[95:64] 哈希摘要寄存器 写入操作： 续接哈希： 在启动续接哈希会话（即启动会话时 HASHMODE.NEW_HASH 位为 0）之前，必须向这些寄存器写入上下文数据。 新建哈希： 在启动新建哈希会话（即 HASHMODE.NEW_HASH 位为 1）时，内部摘要寄存器会自动设置为 SHA-256 算法的预设常数，此时不应向这些寄存器进行写入操作。 读取这些寄存器，可在数据处理完成后，获取中间哈希结果（非最终哈希操作）或最终哈希结果（最终哈希操作）。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.70 HASHDIGESTD 寄存器（偏移 =6CCh）[复位 = 00000000h]

图 12-77 展示了 HASHDIGESTD，表 12-121 中对此进行了介绍。

返回到汇总表。

哈希摘要 D
哈希摘要寄存器由八个 32 位寄存器组成，命名为 HASH_DIGEST_A 至 HASH_DIGEST_H。处理一条消息后，可以从这些寄存器中读取输出摘要。可向这些寄存器写入中间哈希结果，以进行后续的哈希操作。

图 12-77. HASHDIGESTD 寄存器

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

表 12-121. HASHDIGESTD 寄存器字段说明

位	字段	类型	复位	说明
31-0	HASH_DIGEST	R/W	0h	HASH_DIGEST[127:96] 哈希摘要寄存器 写入操作： 续接哈希： 在启动续接哈希会话（即启动会话时 HASHMODE.NEW_HASH 位为 0）之前，必须向这些寄存器写入上下文数据。 新建哈希： 在启动新建哈希会话（即 HASHMODE.NEW_HASH 位为 1）时，内部摘要寄存器会自动设置为 SHA-256 算法的预设常数，此时不应向这些寄存器进行写入操作。 读取这些寄存器，可在数据处理完成后，获取中间哈希结果（非最终哈希操作）或最终哈希结果（最终哈希操作）。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.71 HASHDIGESTE 寄存器（偏移 = 6D0h）[复位 = 00000000h]

图 12-78 展示了 HASHDIGESTE，表 12-122 中对此进行了介绍。

返回到汇总表。

哈希摘要 E
哈希摘要寄存器由八个 32 位寄存器组成，命名为 HASH_DIGEST_A 至 HASH_DIGEST_H。处理一条消息后，可以从这些寄存器中读取输出摘要。可向这些寄存器写入中间哈希结果，以进行后续的哈希操作。

图 12-78. HASHDIGESTE 寄存器

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

表 12-122. HASHDIGESTE 寄存器字段说明

位	字段	类型	复位	说明
31-0	HASH_DIGEST	R/W	0h	HASH_DIGEST[159:128] 哈希摘要寄存器 写入操作： 续接哈希： 在启动续接哈希会话（即启动会话时 HASHMODE.NEW_HASH 位为 0）之前，必须向这些寄存器写入上下文数据。 新建哈希： 在启动新建哈希会话（即 HASHMODE.NEW_HASH 位为 1）时，内部摘要寄存器会自动设置为 SHA-256 算法的预设常数，此时不应向这些寄存器进行写入操作。 读取这些寄存器，可在数据处理完成后，获取中间哈希结果（非最终哈希操作）或最终哈希结果（最终哈希操作）。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.72 HASHDIGESTF 寄存器（偏移 = 6D4h）[复位 = 00000000h]

图 12-79 展示了 HASHDIGESTF，表 12-123 中对此进行了介绍。

返回到汇总表。

哈希摘要 F
哈希摘要寄存器由八个 32 位寄存器组成，命名为 HASH_DIGEST_A 至 HASH_DIGEST_H。处理一条消息后，可以从这些寄存器中读取输出摘要。可向这些寄存器写入中间哈希结果，以进行后续的哈希操作。

图 12-79. HASHDIGESTF 寄存器

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

表 12-123. HASHDIGESTF 寄存器字段说明

位	字段	类型	复位	说明
31-0	HASH_DIGEST	R/W	0h	HASH_DIGEST[191:160] 哈希摘要寄存器 写入操作： 续接哈希： 在启动续接哈希会话（即启动会话时 HASHMODE.NEW_HASH 位为 0）之前，必须向这些寄存器写入上下文数据。 新建哈希： 在启动新建哈希会话（即 HASHMODE.NEW_HASH 位为 1）时，内部摘要寄存器会自动设置为 SHA-256 算法的预设常数，此时不应向这些寄存器进行写入操作。 读取这些寄存器，可在数据处理完成后，获取中间哈希结果（非最终哈希操作）或最终哈希结果（最终哈希操作）。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.73 HASHDIGESTG 寄存器（偏移 =6D8h）[复位 = 00000000h]

图 12-80 展示了 HASHDIGESTG，表 12-124 中对此进行了介绍。

返回到汇总表。

哈希摘要 G
哈希摘要寄存器由八个 32 位寄存器组成，命名为 HASH_DIGEST_A 至 HASH_DIGEST_H。处理一条消息后，可以从这些寄存器中读取输出摘要。可向这些寄存器写入中间哈希结果，以进行后续的哈希操作。

图 12-80. HASHDIGESTG 寄存器

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

表 12-124. HASHDIGESTG 寄存器字段说明

位	字段	类型	复位	说明
31-0	HASH_DIGEST	R/W	0h	HASH_DIGEST[223:192] 哈希摘要寄存器 写入操作： 续接哈希： 在启动续接哈希会话（即启动会话时 HASHMODE.NEW_HASH 位为 0）之前，必须向这些寄存器写入上下文数据。 新建哈希： 在启动新建哈希会话（即 HASHMODE.NEW_HASH 位为 1）时，内部摘要寄存器会自动设置为 SHA-256 算法的预设常数，此时不应向这些寄存器进行写入操作。 读取这些寄存器，可在数据处理完成后，获取中间哈希结果（非最终哈希操作）或最终哈希结果（最终哈希操作）。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.74 HASHDIGESTH 寄存器（偏移 = 6DCh）[复位 = 00000000h]

图 12-81 展示了 HASHDIGESTH，表 12-125 中对此进行了介绍。

返回到汇总表。

哈希摘要 H
哈希摘要寄存器由八个 32 位寄存器组成，命名为 HASH_DIGEST_A 至 HASH_DIGEST_H。处理一条消息后，可以从这些寄存器中读取输出摘要。可向这些寄存器写入中间哈希结果，以进行后续的哈希操作。

图 12-81. HASHDIGESTH 寄存器

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

表 12-125. HASHDIGESTH 寄存器字段说明

位	字段	类型	复位	说明
31-0	HASH_DIGEST	R/W	0h	HASH_DIGEST[255:224] 哈希摘要寄存器 写入操作： 续接哈希： 在启动续接哈希会话（即启动会话时 HASHMODE.NEW_HASH 位为 0）之前，必须向这些寄存器写入上下文数据。 新建哈希： 在启动新建哈希会话（即 HASHMODE.NEW_HASH 位为 1）时，内部摘要寄存器会自动设置为 SHA-256 算法的预设常数，此时不应向这些寄存器进行写入操作。 读取这些寄存器，可在数据处理完成后，获取中间哈希结果（非最终哈希操作）或最终哈希结果（最终哈希操作）。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.75 HASHDIGESTI 寄存器（偏移 = 6E0h）[复位 = 00000000h]

图 12-82 展示了 HASHDIGESTI，表 12-126 中对此进行了介绍。

返回到汇总表。

哈希摘要 I
哈希摘要寄存器由八个 32 位寄存器组成，命名为 HASH_DIGEST_A 至 HASH_DIGEST_H。处理一条消息后，可以从这些寄存器中读取输出摘要。可向这些寄存器写入中间哈希结果，以进行后续的哈希操作。

图 12-82. HASHDIGESTI 寄存器

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

表 12-126. HASHDIGESTI 寄存器字段说明

位	字段	类型	复位	说明
31-0	HASH_DIGEST	R/W	0h	HASH_DIGEST[287:256] 哈希摘要寄存器 写入操作： 续接哈希： 在启动续接哈希会话（即启动会话时 HASHMODE.NEW_HASH 位为 0）之前，必须向这些寄存器写入上下文数据。 新建哈希： 在启动新建哈希会话（即 HASHMODE.NEW_HASH 位为 1）时，内部摘要寄存器会自动设置为 SHA-256 算法的预设常数，此时不应向这些寄存器进行写入操作。 读取这些寄存器，可在数据处理完成后，获取中间哈希结果（非最终哈希操作）或最终哈希结果（最终哈希操作）。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.76 HASHDIGESTJ 寄存器（偏移 = 6E4h）[复位 = 00000000h]

图 12-83 展示了 HASHDIGESTJ，表 12-127 中对此进行了介绍。

返回到汇总表。

哈希摘要 J
哈希摘要寄存器由八个 32 位寄存器组成，命名为 HASH_DIGEST_A 至 HASH_DIGEST_H。处理一条消息后，可以从这些寄存器中读取输出摘要。可向这些寄存器写入中间哈希结果，以进行后续的哈希操作。

图 12-83. HASHDIGESTJ 寄存器

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

表 12-127. HASHDIGESTJ 寄存器字段说明

位	字段	类型	复位	说明
31-0	HASH_DIGEST	R/W	0h	HASH_DIGEST[319:288] 哈希摘要寄存器 写入操作： 续接哈希： 在启动续接哈希会话（即启动会话时 HASHMODE.NEW_HASH 位为 0）之前，必须向这些寄存器写入上下文数据。 新建哈希： 在启动新建哈希会话（即 HASHMODE.NEW_HASH 位为 1）时，内部摘要寄存器会自动设置为 SHA-256 算法的预设常数，此时不应向这些寄存器进行写入操作。 读取这些寄存器，可在数据处理完成后，获取中间哈希结果（非最终哈希操作）或最终哈希结果（最终哈希操作）。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.77 HASHDIGESTK 寄存器（偏移 = 6E8h）[复位 = 00000000h]

图 12-84 展示了 HASHDIGESTK，表 12-128 中对此进行了介绍。

返回到汇总表。

哈希摘要 K
哈希摘要寄存器由八个 32 位寄存器组成，命名为 HASH_DIGEST_A 至 HASH_DIGEST_H。处理一条消息后，可以从这些寄存器中读取输出摘要。可向这些寄存器写入中间哈希结果，以进行后续的哈希操作。

图 12-84. HASHDIGESTK 寄存器

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

表 12-128. HASHDIGESTK 寄存器字段说明

位	字段	类型	复位	说明
31-0	HASH_DIGEST	R/W	0h	HASH_DIGEST[351:320] 哈希摘要寄存器 写入操作： 续接哈希： 在启动续接哈希会话（即启动会话时 HASHMODE.NEW_HASH 位为 0）之前，必须向这些寄存器写入上下文数据。 新建哈希： 在启动新建哈希会话（即 HASHMODE.NEW_HASH 位为 1）时，内部摘要寄存器会自动设置为 SHA-256 算法的预设常数，此时不应向这些寄存器进行写入操作。 读取这些寄存器，可在数据处理完成后，获取中间哈希结果（非最终哈希操作）或最终哈希结果（最终哈希操作）。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.78 HASHDIGESTL 寄存器（偏移 = 6ECh）[复位 = 00000000h]

图 12-85 展示了 HASHDIGESTL，表 12-129 中对此进行了介绍。

返回到汇总表。

哈希摘要 L
哈希摘要寄存器由八个 32 位寄存器组成，命名为 HASH_DIGEST_A 至 HASH_DIGEST_H。处理一条消息后，可以从这些寄存器中读取输出摘要。可向这些寄存器写入中间哈希结果，以进行后续的哈希操作。

图 12-85. HASHDIGESTL 寄存器

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

表 12-129. HASHDIGESTL 寄存器字段说明

位	字段	类型	复位	说明
31-0	HASH_DIGEST	R/W	0h	HASH_DIGEST[383:352] 哈希摘要寄存器 写入操作： 续接哈希： 在启动续接哈希会话（即启动会话时 HASHMODE.NEW_HASH 位为 0）之前，必须向这些寄存器写入上下文数据。 新建哈希： 在启动新建哈希会话（即 HASHMODE.NEW_HASH 位为 1）时，内部摘要寄存器会自动设置为 SHA-256 算法的预设常数，此时不应向这些寄存器进行写入操作。 读取这些寄存器，可在数据处理完成后，获取中间哈希结果（非最终哈希操作）或最终哈希结果（最终哈希操作）。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.79 HASHDIGESTM 寄存器（偏移 = 6F0h）[复位 = 00000000h]

图 12-86 展示了 HASHDIGESTM，表 12-130 中对此进行了介绍。

返回到汇总表。

哈希摘要 M
哈希摘要寄存器由八个 32 位寄存器组成，命名为 HASH_DIGEST_A 至 HASH_DIGEST_H。处理一条消息后，可以从这些寄存器中读取输出摘要。可向这些寄存器写入中间哈希结果，以进行后续的哈希操作。

图 12-86. HASHDIGESTM 寄存器

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

表 12-130. HASHDIGESTM 寄存器字段说明

位	字段	类型	复位	说明
31-0	HASH_DIGEST	R/W	0h	HASH_DIGEST[415:384] 哈希摘要寄存器 写入操作： 续接哈希： 在启动续接哈希会话（即启动会话时 HASHMODE.NEW_HASH 位为 0）之前，必须向这些寄存器写入上下文数据。 新建哈希： 在启动新建哈希会话（即 HASHMODE.NEW_HASH 位为 1）时，内部摘要寄存器会自动设置为 SHA-256 算法的预设常数，此时不应向这些寄存器进行写入操作。 读取这些寄存器，可在数据处理完成后，获取中间哈希结果（非最终哈希操作）或最终哈希结果（最终哈希操作）。

Copyright © 2026 Texas Instruments Incorporated

12.9.1.80 HASHDIGESTN 寄存器（偏移 = 6F4h）[复位 = 00000000h]

图 12-87 展示了 HASHDIGESTN，表 12-131 中对此进行了介绍。

返回到汇总表。

哈希摘要 N
哈希摘要寄存器由八个 32 位寄存器组成，命名为 HASH_DIGEST_A 至 HASH_DIGEST_H。处理一条消息后，可以从这些寄存器中读取输出摘要。可向这些寄存器写入中间哈希结果，以进行后续的哈希操作。

图 12-87. HASHDIGESTN 寄存器

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

表 12-131. HASHDIGESTN 寄存器字段说明

位	字段	类型	复位	说明
31-0	HASH_DIGEST	R/W	0h	HASH_DIGEST[447:416] 哈希摘要寄存器 写入操作： 续接哈希： 在启动续接哈希会话（即启动会话时 HASHMODE.NEW_HASH 位为 0）之前，必须向这些寄存器写入上下文数据。 新建哈希： 在启动新建哈希会话（即 HASHMODE.NEW_HASH 位为 1）时，内部摘要寄存器会自动设置为 SHA-256 算法的预设常数，此时不应向这些寄存器进行写入操作。 读取这些寄存器，可在数据处理完成后，获取中间哈希结果（非最终哈希操作）或最终哈希结果（最终哈希操作）。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.81 HASHDIGESTO 寄存器（偏移 = 6F8h）[复位 = 00000000h]

图 12-88 展示了 HASHDIGESTO，表 12-132 中对此进行了介绍。

返回到汇总表。

哈希摘要 0
哈希摘要寄存器由八个 32 位寄存器组成，命名为 HASH_DIGEST_A 至 HASH_DIGEST_H。处理一条消息后，可以从这些寄存器中读取输出摘要。可向这些寄存器写入中间哈希结果，以进行后续的哈希操作。

图 12-88. HASHDIGESTO 寄存器

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

表 12-132. HASHDIGESTO 寄存器字段说明

位	字段	类型	复位	说明
31-0	HASH_DIGEST	R/W	0h	HASH_DIGEST[479:448] 哈希摘要寄存器 写入操作： 续接哈希： 在启动续接哈希会话（即启动会话时 HASHMODE.NEW_HASH 位为 0）之前，必须向这些寄存器写入上下文数据。 新建哈希： 在启动新建哈希会话（即 HASHMODE.NEW_HASH 位为 1）时，内部摘要寄存器会自动设置为 SHA-256 算法的预设常数，此时不应向这些寄存器进行写入操作。 读取这些寄存器，可在数据处理完成后，获取中间哈希结果（非最终哈希操作）或最终哈希结果（最终哈希操作）。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.82 HASHDIGESTP 寄存器（偏移 = 6FCh）[复位 = 00000000h]

图 12-89 展示了 HASHDIGESTP，表 12-133 中对此进行了介绍。

返回到汇总表。

哈希摘要 P
哈希摘要寄存器由八个 32 位寄存器组成，命名为 HASH_DIGEST_A 至 HASH_DIGEST_H。处理一条消息后，可以从这些寄存器中读取输出摘要。可向这些寄存器写入中间哈希结果，以进行后续的哈希操作。

图 12-89. HASHDIGESTP 寄存器

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

表 12-133. HASHDIGESTP 寄存器字段说明

位	字段	类型	复位	说明
31-0	HASH_DIGEST	R/W	0h	HASH_DIGEST[511:480] 哈希摘要寄存器 写入操作： 续接哈希： 在启动续接哈希会话（即启动会话时 HASHMODE.NEW_HASH 位为 0）之前，必须向这些寄存器写入上下文数据。 新建哈希： 在启动新建哈希会话（即 HASHMODE.NEW_HASH 位为 1）时，内部摘要寄存器会自动设置为 SHA-256 算法的预设常数，此时不应向这些寄存器进行写入操作。 读取这些寄存器，可在数据处理完成后，获取中间哈希结果（非最终哈希操作）或最终哈希结果（最终哈希操作）。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.83 ALGSEL 寄存器（偏移 = 700h）[复位 = 000000000h]

表 12-134 中显示了 ALGSEL。

返回到汇总表。

算法选择
该算法选择寄存器用于配置 DMA 控制器的内部目标地址。

表 12-134. ALGSEL 寄存器字段说明

位	字段	类型	复位	说明
32	HASH_SHA_512	R/W	0h	若设置为 1，则将处于 512B 模式下的哈希引擎选定为 DMA 的目标 此时，DMA 引擎的最大传输大小设定为：读取操作 64 字节，写入操作 32 字节（后者仅适用于通过 DMA 输出哈希结果的情况）。
31	TAG	R/W	0h	若将此位清零，则 DMA 操作仅涉数据。 若将此位置位，则 DMA 操作将包含一个 TAG（即认证结果/摘要）。 对于 SHA-256 操作，必须针对输入数据和 TAG 分别配置 DMA。对于选定的其他任何模块，若将此位置位，则仅允许执行读取 TAG 的 DMA 操作。不允许通过 DMA 向该选定模块传输数据或从其接收数据。
30-4	RESERVED	R	0h	保留
3	RESERVED	R/W	0h	
2	HASH_SHA_256	R/W	0h	若设置为 1，则将处于 256B 模式下的哈希引擎选定为 DMA 的目标 此时，DMA 引擎的最大传输大小设定为：读取操作 64 字节，写入操作 32 字节（后者仅适用于通过 DMA 输出哈希结果的情况）。
1	AES	R/W	0h	若设置为 1，则将 AES 引擎选定为 DMA 的源或目标 向 DMA 引擎的读取和写入最大传输大小均设置为 16 字节。
0	KEY_STORE	R/W	0h	若设置为 1，则将密钥存储区选定为 DMA 的目标 向 DMA 引擎传输的最大大小设定为 32 字节（不过，允许进行 16、24 和 32 字节的传输）

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.84 DMAPROTCTL 寄存器（偏移 = 704h）[复位 = 00000000h]

图 12-90 展示了 DMAPROTCTL，表 12-135 中对此进行了介绍。

返回到汇总表。

DMA 保护控制
Master PROT 特权访问使能
当在 AHB 主接口上执行对密钥的读取操作，以便将密钥写入存储模块时，该寄存器会将 AHB 主接口的 AHB HPROT 总线第二位（位 [1]）置为有效。

图 12-90. DMAPROTCTL 寄存器

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

表 12-135. DMAPROTCTL 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	PROT_EN	R/W	0h	选择针对以密钥存储区为目的地的 DMA 传输所使用的 AHB 传输保护控制。 0：传输使用“USER”类型访问。 1：传输使用“PRIVILEGED”类型访问。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.85 SWRESET 寄存器（偏移 =740h）[复位 = 00000000h]

图 12-91 展示了 SWRESET，表 12-136 中对此进行了介绍。

返回到汇总表。

软件复位

图 12-91. SWRESET 寄存器

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

表 12-136. SWRESET 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	SW_RESET	R/W	0h	如果此位设置为 1，则以下模块将被复位： - 复位主控内部状态。其中包括中断、错误状态寄存器以及用于产生“结果可用”中断的 FSM。 - 复位密钥存储模块状态。这包括清除写入区域的标志 因此，必须将密钥重新加载到密钥存储模块中。 写入 0 无效。 复位后，系统会自动清除该位。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.86 IRQTYPE 寄存器（偏移 = 780h）[复位 = 00000000h]

图 12-92 展示了 IRQTYPE，表 12-137 中对此进行了介绍。

返回到汇总表。

控制中断配置

图 12-92. IRQTYPE 寄存器

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

表 12-137. IRQTYPE 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	等级	R/W	0h	若此位为 0，则中断输出为脉冲信号 若此位设为 1，则中断为电平中断，必须通过写入中断清除寄存器进行清除。 此位适用于两个中断输出信号。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.87 IRQEN 寄存器（偏移 = 784h）[复位 = 00000000h]

图 12-93 展示了 IRQEN，表 12-138 中对此进行了介绍。

返回到汇总表。

控制中断使能

图 12-93. IRQEN 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	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_IN_DONE	RESULT_AVAIL
R-0h						R/W-0h	R/W-0h

表 12-138. IRQEN 寄存器字段说明

位	字段	类型	复位	说明
31-2	RESERVED	R	0h	保留
1	DMA_IN_DONE	R/W	0h	如果此位设为 0，则 DMA 输入完成 (irq_dma_in_done) 中断输出将被禁用，并保持为 0。 如果此位设为 1，则 DMA 输入完成中断输出将被启用。
0	RESULT_AVAIL	R/W	0h	如果此位设为 0，则结果可用 (irq_result_av) 中断输出将被禁用，并保持为 0。 如果此位设为 1，则结果可用中断输出将被启用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.88 IRQCLR 寄存器（偏移 = 788h）[复位 = 00000000h]

图 12-94 展示了 IRQCLR，表 12-139 中对此进行了介绍。

返回到汇总表。

控制中断清除

图 12-94. IRQCLR 寄存器

31	30	29	28	27	26	25	24
DMA_BUS_ERR	KEY_ST_WR_ERR	KEY_ST_RD_ERR	RESERVED				
W-0h	W-0h	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						DMA_IN_DONE	RESULT_AVAIL
R-0h						W-0h	W-0h

表 12-139. IRQCLR 寄存器字段说明

位	字段	类型	复位	说明
31	DMA_BUS_ERR	W	0h	若向此位写入 1，则清除 DMA 总线错误状态。 写入 0 无效。
30	KEY_ST_WR_ERR	W	0h	若向此位写入 1，则清除密钥存储写入错误状态。 写入 0 无效。
29	KEY_ST_RD_ERR	W	0h	若向此位写入 1，则清除密钥存储区读取错误状态。 写入 0 无效。
28-2	RESERVED	R	0h	保留
1	DMA_IN_DONE	W	0h	若向此位写入 1，则 DMA 输入完成 (irq_dma_in_done) 中断输出将被清除。 写入 0 无效。 请注意，仅当该中断输出被配置为电平触发模式时，清除中断操作才具有实际意义（详见 IRQTYPE）。
0	RESULT_AVAIL	W	0h	若向此位写入 1，则结果可用 (irq_result_av) 中断输出将被清除。 写入 0 无效。 请注意，仅当该中断输出被配置为电平触发模式时，清除中断操作才具有实际意义（详见 IRQTYPE）。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.89 IRQSET 寄存器（偏移 = 78Ch）[复位 = 00000000h]

图 12-95 展示了 IRQSET，表 12-140 中对此进行了介绍。

返回到汇总表。

控制中断设置

图 12-95. IRQSET 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	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_IN_DONE	RESULT_AVAIL
R-0h						W-0h	W-0h

表 12-140. IRQSET 寄存器字段说明

位	字段	类型	复位	说明
31-2	RESERVED	R	0h	保留
1	DMA_IN_DONE	W	0h	若向此位写入 1，则 DMA 数据输入完成 (irq_dma_in_done) 中断输出将被置为 1。 写入 0 无效。 如果中断配置寄存器被设置为脉冲模式，则无需清除 DMA 数据输入完成 (irq_dma_in_done) 中断。如果配置为电平触发模式，应通过写入中断清除寄存器 (IRQCLR.DMA_IN_DONE) 来清除中断输出。
0	RESULT_AVAIL	W	0h	若向此位写入 1，则结果可用 (irq_result_av) 中断输出将被置为 1。 写入 0 无效。 如果中断配置寄存器被设置为脉冲模式，则无需清除结果可用 (irq_result_av) 中断。如果配置为电平触发模式，应通过写入中断清除寄存器 (IRQCLR.RESULT_AVAIL) 来清除中断输出。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.90 IRQSTAT 寄存器（偏移 = 790h）[复位 = 00000000h]

图 12-96 展示了 IRQSTAT，表 12-141 中对此进行了介绍。

返回到汇总表。

控制中断状态

图 12-96. IRQSTAT 寄存器

31	30	29	28	27	26	25	24
DMA_BUS_ERR	KEY_ST_WR_ERR	KEY_ST_RD_ERR	RESERVED				
R-0h	R-0h	R-0h	R-0h				
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						DMA_IN_DONE	RESULT_AVAIL
R-0h						R-0h	R-0h

表 12-141. IRQSTAT 寄存器字段说明

位	字段	类型	复位	说明
31	DMA_BUS_ERR	R	0h	当在 DMA 操作期间检测到 DMA 总线错误时，该位将被置位。该寄存器的数值将一直保持，直至通过 IRQCLR.DMA_BUS_ERR 将其清除。 注意：当在 DMA 操作期间检测到 AHB 主接口错误时，将触发此错误。
30	KEY_ST_WR_ERR	R	0h	在向密钥存储区执行 DMA 写入操作期间检测到写入错误时，该位将被置位。该寄存器的数值将一直保持，直至通过 IRQCLR.KEY_ST_WR_ERR 寄存器将其清除。 注意：如果 DMA 操作未覆盖完整的密钥区域或写入的区域多于预期，则将会此错误置为有效。
29	KEY_ST_RD_ERR	R	0h	在从密钥存储区读取密钥并将其复制至 AES 核心的过程中检测到读取错误时，该位将被置位。该寄存器的数值将一直保持，直至通过 IRQCLR.KEY_ST_RD_ERR 寄存器将其清除。 注意：如果在密钥存储区中选择了不可用的密钥位置，则会将此错误置为有效。
28-2	RESERVED	R	0h	保留
1	DMA_IN_DONE	R	0h	该只读位返回 DMA 数据输入完成 (irq_data_in_done) 中断状态的实际值，对应于 DMA 数据输入完成中断输出引脚 (irq_data_in_done) 的状态。
0	RESULT_AVAIL	R	0h	该只读位返回结果可用 (irq_result_av) 中断输出引脚的实际中断状态。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

12.9.1.91 HWVER 寄存器（偏移 = 7FCh）[复位 = 92008778h]

图 12-97 展示了 HWVER，表 12-142 中对此进行了介绍。

返回到汇总表。

硬件版本

图 12-97. HWVER 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED				HW_MAJOR_VER				HW_MINOR_VER				HW_PATCH_LVL			
R-0h				R-2h				R-0h				R-0h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VER_NUM_COMPL								VER_NUM							
R-87h								R-78h							

表 12-142. HWVER 寄存器字段说明

位	字段	类型	复位	说明
31-28	RESERVED	R	0h	保留
27-24	HW_MAJOR_VER	R	2h	主要版本号
23-20	HW_MINOR_VER	R	0h	次要版本号
19-16	HW_PATCH_LVL	R	0h	补丁级别 本版本首次推出时从 0 开始
15-8	VER_NUM_COMPL	R	87h	这些位仅包含位 [7:0] (0x87) 的补码，供驱动程序用于确认 EIP-120t 寄存器确实已被读取。
7-0	VER_NUM	R	78h	这些位对 EIP-120T 的 EIP 编号进行编码，该字段包含值 120（十进制）或 0x78。

Copyright © 2026 Texas Instruments Incorporated

章节 13

I/O 控制器 (IOC)

本章介绍了输入/输出控制器 (IOC) 和通用输入输出 (GPIO)。IOC 提供了灵活的配置，因为大多数外设端口都可以映射到任何物理 I/O 引脚。CC13x2 和 CC26x2 器件平台具有多达 31 个 I/O 引脚，这些引脚可配置为 GPIO 或外设 I/O 功能。RSIP/PSIP 模块支持多达 32 个 I/O 引脚，其中 31 个可配置为 GPIO，32 个为外设 I/O 功能。

13.1 简介 1086
13.2 IOC 概述 1086
13.3 I/O 映射和配置 1087
13.4 DIO 引脚上的边沿检测 1088
13.5 未使用的 I/O 引脚 1089
13.6 GPIO 1089
13.7 I/O 引脚功能 1090
13.8 外设 PORTID 1092
13.9 I/O 引脚 1093
13.10 IOC 寄存器 1094

Copyright © 2026 Texas Instruments Incorporated

13.1 简介

I/O 控制器可配置 I/O 引脚，并将外设信号映射到 CC13x2 和 CC26x2 器件封装和 RSIP/PSIP 模块上的物理引脚 (DIOx)。本章介绍了 I/O 控制器功能，并举例说明了如何将外设功能映射到用户选择的引脚。

几个类似的术语定义如下：

- PORTID 是外设功能的编号。
- GPIO 是 PORTID 为 0x0 的外设功能。
- DIOn（DIO0 至 DIO31）是特定封装上不同 I/O 引脚的逻辑名称。

  其中八个 DIO 还具备模拟功能。器件特定数据表提供了不同封装中 DIO 及引脚之间的映射。

13.2 IOC 概述

图 13-1 显示 IOC 的总体概览。

IOC 模块包含两个主要子模块：

- 微控制器单元 IOC (MCU IOC) 将外设端口配置为用户定义的引脚。
- 常开型 IOC (AON IOC) 模块可以处理 JTAG、32kHz 时钟、AON 外设及 AUX 域信号。

常开型外设（RTC，电池监测器和内部温度传感器）采用 32kHz 低频时钟（SCLK_LF，有关更多详细信息，请参阅节 7.5）来计时。这使得器件能够在非常低的功率级别下运行，同时仍保持这些外设功能的有效运行。正确配置后，AON IOC 确保在 AUX 电源域断电时所有 I/O 的输出电平保持不变

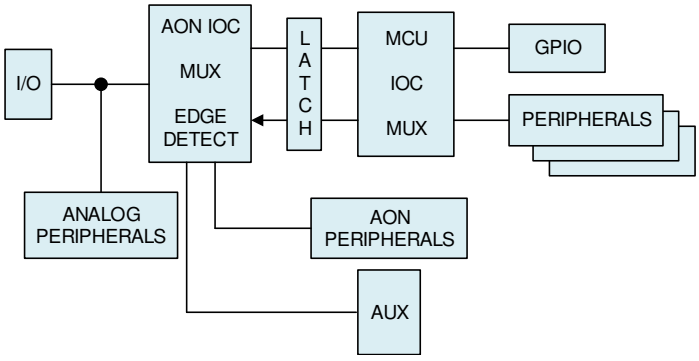


图 13-1. IOC 概述（简化版）

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.3 I/O 映射和配置

MCU IOC 可以将 GPIO、SSI、UART、I²C 和 I²S 等许多外设模块映射到任何可用的 I/O。JTAG 功能仅限于特定 I/O 引脚。每种类型的外设信号都有一个唯一的 PORTID，可以分配给所选的 I/O 引脚（称为 DIO）。表 13-3 列出了不同的 PORTID 信号。

13.3.1 基本 I/O 映射

要将外设功能映射到 DIOn（其中 n 的范围为 0 至最大值 31），必须在相应的 IOC:IOCFGn 寄存器中设置 PORTID 和引脚配置。要选择引脚必须路由的功能类型，请为所需外设功能选择 PORTID 编号，并将 PORTID 编号写入 IOC:IOCFGn.PORTID 位字段。

可使用以下驱动程序库函数设置该函数：

```
IOCPortConfigureSet(DIOn, PORTID, PIN-CONFIG);
```

请参阅节 13.5，了解可在 PIN-CONFIG 中设置的配置类型。

13.3.2 将 AUXIO 映射到 DIO 引脚

传感器控制器域（AUX 域）中有多达 32 个信号（AUXIO0 到 AUXIO31）。可以将这些信号路由至表 13-2 中给出的特定 DIO 引脚。信号 AUXIO19 至 AUXIO26 具有模拟功能，但也可以用作数字 I/O。所有其他 AUXIOn 信号都是数字信号。从 AUX 域路由的信号的配置与 GPIO 及其他外设功能不同。本节不涉及传感器控制器的所有功能的使用（有关更多详细信息，请参阅章节 19）。

在此示例中，AUXIO1 映射到 DIO17 并设置为数字输入。

1. 将 IOC:IOCFG17 PORTID 位字段设置为 0x08 (AUX_I/O)，以将 AUXIO1 路由到 DIO17。
2. AUX 域中的 I/O 信号具有自己的开源或开漏配置，必须在 AUX 域中的 AUX_AIODIO0:IOMODE 寄存器中设置这些配置。将 AUX_AIODIO0:IOMODE.IO1 设置为 0x01 以启用 AUXIO1 作为数字输入。
3. 通过将 AUX_AIODIO0:GPIODIE 寄存器中的 IO7_0 位字段设置为 0x02 来启用 AUXIO1 的数字输入缓冲器。

13.3.3 通过 I/O 控制外部 LNA/PA（范围扩展器）

有四个称为 *RF 内核数据输出 n* 的逻辑 RF 内核内部输出信号，其中 n 从 0 变为 3。这些信号可以映射至 DIO。默认情况下，当必须启用 LNA 时，RF 内核数据输出 0 设置为高电平，当必须启用 PA 时，RF 内核数据输出 1 设置为高电平。表 13-1 对这些信号进行了说明。通过在指定的 IOCFGn 寄存器中设置相关 PORTID，可以将信号映射到任何 DIO。

```
#include <driverlib/ioc.h>
// Map RFC_GPO0 to DIOx
IOCPortConfigureSet(IOID_x, IOC_PORT_RFC_GPO0, IOC_IOMODE_NORMAL);
```

表 13-1. PA 和 LNA 的 RF 内核数据信号

端口名称	PORTID	RF 内核信号	说明
RFC_GPO0	0x2F	RF 内核数据输出 0	LNA 使能
RFC_GPO1	0x30	RF 内核数据输出 1	PA 使能
RFC_GPO2	0x31	RF 内核数据输出 2	合成器校准正在运行
RFC_GPO3	0x32	RF 内核数据输出 3	TX 开始

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.3.4 将 32kHz 系统时钟（LF 时钟）映射至 DIO

AON IOC 包含 32kHz LF 系统时钟输出的输出使能控制，而该时钟信号具有自己的 PORTID，称为 AON_CLK32K (0x7)。这样可以轻松地将时钟信号输出至引脚。将时钟映射到所选的 DIO，并通过将 AON_IOC:CLK32KCTL.OE_N 寄存器字段设置为 0x0 来启用时钟输出。以下两个驱动程序库调用实现了相同的结果：

```
#include <driverlib/aon_ioc.h>
IOCPortConfigureSet(IOIDn, IOC_PORT_AON_CLK32K, IOC_STD_OUTPUT);
AONIOC32kHzOutputEnable();
```

这会在除关断外的所有功耗模式下输出 LF 系统时钟信号。

使用 DIO 作为 32kHz LF 系统时钟的输入源时，也会选择 AON_CLK32K PORTID 值。有关如何使用 DIO 作为时钟源的说明，请参阅章节 7。

备注

AON_CLK32K PORTID 必须仅用作单个时钟输出或仅用作时钟输入。

13.4 DIO 引脚上的边沿检测

AON IOC 支持检测数字 I/O 上的正边沿和（或）负边沿，并将结果事件以下面的事件形式提供给 AON 事件结构：

- IOEV_AON_PROG0
- IOEV_AON_PROG1
- IOEV_AON_PROG2
- IOEV_RTC
- IOEV_MCU_WU

边沿检测事件可以通过 MCU GPIO 及 AUX 域进行清除。也可以通过对边沿配置执行禁用或启用周期，从 MCU IOC 清除边沿检测事件。MCU GPIO 的每个边沿检测单元都有单独的清除线路，而 AUX 域使用单条线路来清除连接到 AUX 域的引脚上的所有事件。从 AUX 域清除时，将清除与 AUX 域 I/O 相关的所有事件。

EDGE DETECT 块对每个 I/O 使用边沿检测单元。每个检测单元可以标记检测到的边沿并触发中断信号。来自所有单元的中断信号一起进行 OR 运算，形成一条通往 AON 事件结构的中断线路。

当输入 I/O 的任何可编程子集生成事件时，AON IOC 也可以生成中断事件。控制边沿检测电路的寄存器驻留在 MCU IOC 中。

13.4.1 将 DIO 配置为 GPIO 输入，生成 EDGE DETECT 上的中断

通过 IOC:IOCFGn EDGE_IRQ_EN 和 EDGE_DET 位字段配置从 DIO 生成中断和边沿检测事件。为了执行边沿检测，DIO 必须设置输入使能。GPIO 边沿检测事件会发送至 CPU 中断 IRQ0（矢量编号 16）。必须启用此中断才能调用 GPIO 中断处理程序。在该中断处理程序中，必须通过清除相关的 GPIO:EVFLAGS31 事件寄存器 DIOn 位来清除事件源。读取该寄存器会针对触发事件返回 1，而对于非触发事件返回 0。通过切换启用的 EDGE_DET 配置，可以从 MCU IOC 清除该事件。当有效边沿配置被禁用且 IOC:IOCFGn.EDGE_DET 设置为 0 时，会清除该事件。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.5 未使用的 I/O 引脚

默认情况下，I/O 驱动器（输出）和输入缓冲器（输入）在上电或复位时被禁用（三态模式），因此 I/O 引脚可以安全地保持不连接（悬空）。

如果 I/O 引脚处于三态并连接到一个具有不同电压电位的节点，则可能会有少量泄漏电流通过该引脚。同样的情况也适用于配置为输入的 I/O 引脚。此时该引脚连接到一个电压源（例如 VDD/2）。届时输入将是一个未定义的值，要么是
0，要么是 1。

13.6 GPIO

MCU GPIO 是一个通用输入/输出模块，允许软件对 DIO 进行写入和读取。GPIO 支持多达 31 个可编程 I/O 引脚。这些引脚通过 IOC 模块来配置。要修改单个 GPIO 输出值，请使用 GPIO:DOUTn 寄存器（请参阅节 13.10.2）。要将 DIO1 设置为 GPIO 输出并切换该位，请执行以下过程。

1. 通过将 IOC:IOCFG1.PORT_ID 寄存器设置为 0 (GPIO PORDTID)，将 DIO1 映射为 GPIO 输出。
2. 通过清除 IOC:IOCFG1.IE 位来确保 DIO1 设置为输出。更多端口配置也可以在 IOC:IOCFG1 寄存器中设置（有关更多详细信息，请参阅节 13.9.1.2）。
3. 通过发出读取-修改-写入操作来设置 GPIO:DOE31_0.DIO1 中 DIO1 的数据输出使能位。
4. 通过使用 0x100 对 GPIO:DOUT3_0:DIO1 位发出 XOR 操作来切换 DIO1 输出。
5. 调用以下驱动程序库函数：

```
IOCPinTypeGpioOutput(0x1);
GPIOPinToggle(0x1);
```


English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.7 I/O 引脚功能

表 13-2 展示了 DIO 的 I/O 功能。请参阅特定于器件的数据表，以确认封装引脚的 DIO 映射。

表 13-2. CC13x2 和 CC26x2 引脚映射

封装类型		传感器控制器		驱动强度	JTAG
7 × 7 QFN (RGZ)	RSIP / PSIP	支持模拟	AUX 域 I/O		
DIO					
不适用	31(1)		0	不适用	
30	30	是	19	2mA，4mA	
29	29	是	20	2mA，4mA	
28	28	是	21	2mA，4mA	
27	27	是	22	2mA，4mA	
26	26	是	23	2mA，4mA	
25	25	是	24	2mA、4mA	
24	24	是	25	2mA、4mA	
23	23	是	26	2mA、4mA	
22	22		27	2mA、4mA	
21	21		28	2mA，4mA	
20	20		29	2mA，4mA	
19	19		30	2mA，4mA	
18	18		31	2mA，4mA	
17	17		1	2mA、4mA、8mA	TDI
16	16		2	2mA、4mA、8mA	TDO
15	15		3	2mA，4mA	
14	14		4	2mA，4mA	
13	13		5	2mA，4mA	
12	12		6	2mA，4mA	
11	11		7	2mA，4mA	
10	10		8	2mA，4mA	
9	9		9	2mA，4mA	
8	8		10	2mA，4mA	
7	7		11	2mA、4mA、8mA	
6	6		12	2mA、4mA、8mA	
5	5		13	2mA、4mA、8mA	
4	4		14	2mA，4mA	
3(2) (3)	3(4)		15	2mA，4mA	
2(2) (3)	2		16	2mA，4mA	
1(2) (3)	1		17	2mA，4mA	
0(2) (3)	0(4)		18	2mA，4mA	

(1) DIO31 仅支持外设 I/O 功能，而不支持 GPIO
(2) CC1352R 器件不支持 DIO0 至 DIO2。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

(3) CC1352P 器件不支持 DIO0 至 DIO4。
(4) PSIP 模块专门将 DIO0 和 DIO3 位用于外部 RF 开关

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.8 外设 PORTID

表 13-3 列出了不同的 PORTID 信号。

表 13-3. CC13x2 和 CC26x2 PORTID

ID	端口名称	端口描述		ID	端口名称	端口描述
0	GPIO	默认 GPIO 用法		27	PORT_EVENT4	通用 I/O 事件 4
1-6		*保留*		28	PORT_EVENT5	通用 I/O 事件 5
7	AON_CLK32K	AON 32kHz 时钟引脚		29	PORT_EVENT6	通用 I/O 事件 6
8	AUX Domain_IO	AUX 域 I/O 引脚		30	PORT_EVENT7	通用 I/O 事件 7
9	SSI0_RX	SSI 0 RX 引脚		31		*保留*
10	SSI0_TX	SSI 0 TX 引脚		32	CPU_SWV	CPU SWV
11	SSI0_FSS	SSI 0 FSS 引脚		33	SSI1_RX	SSI 1 RX 引脚
12	SSI0_CLK	SSI 0 CLK 引脚		34	SSI1_TX	SSI 1 TX 引脚
13	I2C_MSSDA	I^2C 数据		35	SSI1_FSS	SSI 1 FSS 引脚
14	I2C_MSSCL	I^2C 时钟		36	SSI1_CLK	SSI 1 CLK 引脚
15	UART0_RX	UART 0 RX 引脚		37	I2S_AD0	I^2C 数据 0 引脚
16	UART0_TX	UART 0 TX 引脚		38	I2S_AD1	I^2C 数据 1 引脚
17	UART0_CTS	UART 0 CTS 引脚		39	I2S_WCLK	I^2S WCLK 引脚
18	UART0_RTS	UART 0 RTS 引脚		40	I2S_BCLK	I^2S BCLK 引脚
19	UART1_RX	UART 1 RX 引脚		41	I2S_MCLK	I^2S MCLK 引脚
20	UART1_TX	UART 1 TX 引脚		42-45		*保留*
21	UART1_CTS	UART 1 CTS 引脚		46		射频内核内部信号
22	UART1_RTS	UART 1 RTS 引脚		47	RFC_GPO0	射频内核通用输出 0
23	PORT_EVENT0	通用 I/O 事件 0		48	RFC_GPO1	射频内核通用输出 1
24	PORT_EVENT1	通用 I/O 事件 1		49	RFC_GPO2	射频内核通用输出 2
25	PORT_EVENT2	通用 I/O 事件 2		50	RFC_GPO3	射频内核通用输出 3
26	PORT_EVENT3	通用 I/O 事件 3		51-56		射频内核内部信号

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.9 I/O 引脚

本部分讨论 CC13x2 和 CC26x2 器件平台 I/O 引脚的具体物理细节和配置可能性。

13.9.1 输入/输出模式

每个 I/O 引脚都有独立的输入和输出缓冲器，可以独立配置。输入和输出的主要配置包括：

- 输入模式（断开、迟滞、上拉、下拉）
- 输出模式（三态、推挽、开漏、开源）

输入和输出缓冲器可以同时使能或禁用。通过禁用输出缓冲器，相应的 I/O 引脚将处于三态（高阻态）。如果在输出缓冲器禁用时没有任何器件将 I/O 驱动到有效的逻辑电平，则应禁用输入缓冲器，以避免通过 I/O 输入缓冲器产生过大的电流消耗。节 13.9.1.2 更详细地介绍了 I/O 引脚配置。

13.9.1.1 物理引脚

数字 I/O 驱动器和接收器是一个具有宽电源电压范围的双向缓冲器，结合了输出缓冲器、带可选磁滞的输入缓冲器以及可选的上拉和下拉电路。I/O 具有有限的电源管理特性，包括支持在内核电源门控的情况下从睡眠中唤醒。引脚的灌电流和拉电流能力是对称的，如图 13-2 所示，该图粗略描绘了 I/O 引脚。上拉和下拉电阻见数据表。

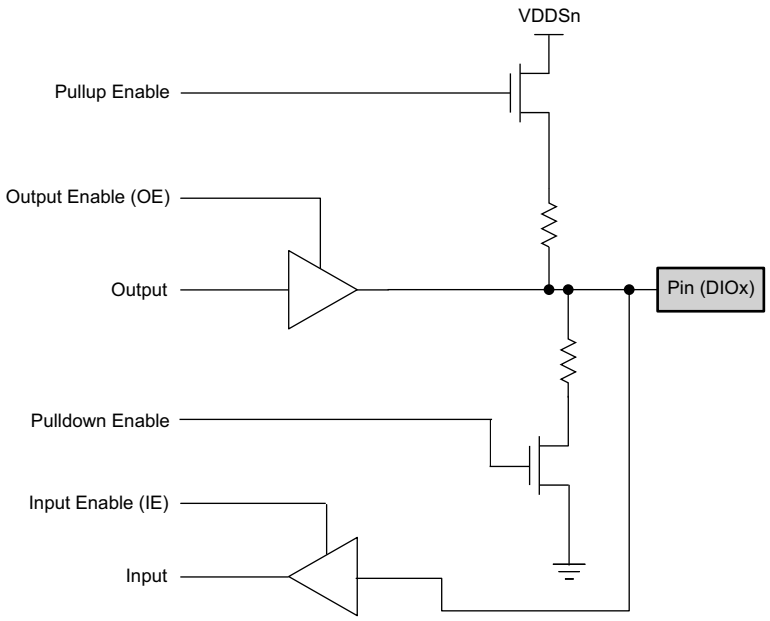


图 13-2. 通用 I/O 引脚（简化版）

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.9.1.2 引脚配置

IOC 允许软件根据应用要求来配置引脚。软件可以为任何或所有 I/O 引脚配置不同特性设置。除输出驱动器（GPIO:DOE31_0 寄存器中设置的输出使能）外，以下所有特性均在 IOC:IOCFGn 寄存器中控制：

- **驱动强度** (IOC:IOCFGn.IOSTR)

  配置 I/O 引脚的驱动强度及最大电流。所有 I/O 引脚均支持 2mA 和 4mA 电流，而五个引脚支持高达 8mA 电流。通过将 IOC:IOCFGn.IOSTR 设置为 0x0，可以根据电池监测器 BATMON 的输入自动更新驱动强度，从而在不同的电池电压下保持所设置的驱动强度水平。
- **拉取** (IOC:IOCFGn.PULL_CTL)

  配置 I/O 引脚上的弱上拉。可以设置以下参数：上拉、下拉或者无上拉。有关特定的上拉和下拉电阻，请参阅数据表。
- **压摆控制** (IOC:IOCFGn.SLEW_RED)

  设置 I/O 引脚上的高压摆率或低压摆率。
- **迟滞** (IOC:IOCFGn.HYST_EN)

  启用或禁用 I/O 引脚上的输入磁滞。
- **开源极或开漏配置** (IOC:IOCFGn.IOMODE)

  将引脚配置为正常、开源或开漏；所有这些可以设置为反相或正常（非反向）。
- **中断和边沿检测**（IOC:IOCFGn.EDGE_IRQ_EN 和 IOC:IOCFGn.EDGE_DET）

  启用由 I/O 引脚上的边沿检测触发的中断。支持不同的边沿检测模式，并且可能的模式为上升沿、下降沿、上升沿和下降沿触发或无边沿检测。
- **输入驱动器** (IOC:IOCFGn.IE)

  启用或禁用 I/O 输入驱动器。
- **输出驱动器**（取决于映射到引脚的特定外设）

  启用或禁用 I/O 输出驱动器。

13.10 IOC 寄存器

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.1 AON_IOC 寄存器

表 13-4 列出了 AON_IOC 寄存器的存储器映射寄存器。表 13-4 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

表 13-4. AON_IOC 寄存器

偏移	首字母缩写词	寄存器名称	部分
0h	IOSTRMIN	内部	IOSTRMIN 寄存器 (偏移 =0h) [复位 = 00000003h]
4h	IOSTRMED	内部	IOSTRMED 寄存器 (偏移 =4h) [复位 = 00000006h]
8h	IOSTRMAX	内部	IOSTRMAX 寄存器 (偏移 = 8h) [复位 = 00000005h]
10h	CLK32KCTL	SCLK_LF 外部输出控制	CLK32KCTL 寄存器 (偏移 = 10h) [复位 = 00000001h]
14h	TCKCTL	TCK IO 引脚控制	TCKCTL 寄存器 (偏移 = 14h) [复位 = 00000001h]

复杂的位访问类型经过编码可适应小型表单元。表 13-5 展示了适用于此部分中访问类型的代码。

表 13-5. AON_IOC 访问类型代码

访问类型	代码	说明
读取类型		
R	R	读取
写入类型		
W	W	写入
复位或默认值		
-n		复位后的值或默认值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.1.1 IOSTRMIN 寄存器（偏移 =0h）[复位 = 00000003h]

图 13-3 展示了 IOSTRMIN，表 13-6 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 13-3. IOSTRMIN 寄存器

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													GRAY_CODE		
R-0h													R/W-3h		

表 13-6. IOSTRMIN 寄存器字段说明

位	字段	类型	复位	说明
31-3	RESERVED	R	0h	保留
2-0	GRAY_CODE	R/W	3h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.1.2 IOSTRMED 寄存器（偏移 =4h）[复位 = 00000006h]

图 13-4 展示了 IOSTRMED，表 13-7 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 13-4. IOSTRMED 寄存器

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													GRAY_CODE		
R-0h													R/W-6h		

表 13-7. IOSTRMED 寄存器字段说明

位	字段	类型	复位	说明
31-3	RESERVED	R	0h	保留
2-0	GRAY_CODE	R/W	6h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.1.3 IOSTRMAX 寄存器（偏移 = 8h）[复位 = 00000005h]

图 13-5 展示了 IOSTRMAX，表 13-8 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 13-5. IOSTRMAX 寄存器

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													GRAY_CODE		
R-0h													R/W-5h		

表 13-8. IOSTRMAX 寄存器字段说明

位	字段	类型	复位	说明
31-3	RESERVED	R	0h	保留
2-0	GRAY_CODE	R/W	5h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.1.4 CLK32KCTL 寄存器（偏移 = 10h）[复位 = 00000001h]

图 13-6 展示了 CLK32KCTL，表 13-9 对其进行了介绍。

返回到汇总表。

SCLK_LF 外部输出控制

图 13-6. CLK32KCTL 寄存器

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

表 13-9. CLK32KCTL 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	OE_N	R/W	1h	0：输出使能有效。将 PORT_ID（例如 IOC:IOCFG0.PORT_ID）设置为 AON_CLK32K 的 IO 引脚上的 SCLK_LF 输出。 1：输出使能无效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.1.5 TCKCTL 寄存器（偏移 = 14h）[复位 = 00000001h]

图 13-7 展示了 TCKCTL，表 13-10 中对此进行了介绍。

返回到汇总表。

TCK IO 引脚控制

图 13-7. TCKCTL 寄存器

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															EN
R-0h															R/W-1h

表 13-10. TCKCTL 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	EN	R/W	1h	0：TCK 的输入驱动器已禁用。 1：TCK 的输入驱动器已使能。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.2 GPIO 寄存器

表 13-11 列出了 GPIO 寄存器的存储器映射寄存器。表 13-11 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

表 13-11. GPIO 寄存器

偏移	首字母缩写词	寄存器名称	部分
0h	DOUT3_0	数据输出 0 至 3	DOUT3_0 寄存器 (偏移 = 0h) [复位 = 00000000h]
4h	DOUT7_4	数据输出 4 至 7	DOUT7_4 寄存器 (偏移 = 4h) [复位 = 00000000h]
8h	DOUT11_8	数据输出 8 至 11	DOUT11_8 寄存器 (偏移 = 8h) [复位 = 00000000h]
Ch	DOUT15_12	数据输出 12 至 15	DOUT15_12 寄存器 (偏移 = Ch) [复位 = 00000000h]
10h	DOUT19_16	数据输出 16 至 19	DOUT19_16 寄存器 (偏移 = 10h) [复位 = 00000000h]
14h	DOUT23_20	数据输出 20 至 23	DOUT23_20 寄存器 (偏移 = 14h) [复位 = 00000000h]
18h	DOUT27_24	数据输出 24 至 27	DOUT27_24 寄存器 (偏移 = 18h) [复位 = 00000000h]
1Ch	DOUT31_28	数据输出 28 至 31	DOUT31_28 寄存器 (偏移 = 1Ch) [复位 = 00000000h]
80h	DOUT31_0	DIO 0 至 31 的数据输出	DOUT31_0 寄存器 (偏移 = 80h) [复位 = 00000000h]
90h	DOUTSET31_0	数据输出置位	DOUTSET31_0 寄存器 (偏移 = 90h) [复位 = 00000000h]
A0h	DOUTCLR31_0	数据输出清除	DOUTCLR31_0 寄存器 (偏移 = A0h) [复位 = 00000000h]
B0h	DOUTTGL31_0	数据输出切换	DOUTTGL31_0 寄存器 (偏移 = B0h) [复位 = 00000000h]
C0h	DIN31_0	DIO 0 至 31 的数据输入	DIN31_0 寄存器 (偏移 = C0h) [复位 = 00000000h]
D0h	DOE31_0	DIO 0 至 31 的数据输出使能	DOE31_0 寄存器 (偏移 = D0h) [复位 = 00000000h]
E0h	EVFLAGS31_0	DIO 0 至 31 的事件寄存器	EVFLAGS31_0 寄存器 (偏移 = E0h) [复位 = 00000000h]

复杂的位访问类型经过编码可适应小型表单元。表 13-12 展示了适用于此部分中访问类型的代码。

表 13-12. GPIO 访问类型代码

访问类型	代码	说明
读取类型		
R	R	读取
写入类型		
W	W	写入
W1C	W 1C	写入 1 以清零
W1S	W 1S	写入 1 以进行设置
复位或默认值		
-n		复位后的值或默认值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.2.1 DOUT3_0 寄存器（偏移 = 0h）[复位 = 00000000h]

图 13-8 展示了 DOUT3_0，表 13-13 对其进行了介绍。

返回到汇总表。

数据输出 0 至 3
用于对 DOUT31_0 中每个位进行字节访问的别名寄存器

图 13-8. DOUT3_0 寄存器

31	30	29	28	27	26	25	24
RESERVED							DIO3
R-0h							W-0h
23	22	21	20	19	18	17	16
RESERVED							DIO2
R-0h							W-0h
15	14	13	12	11	10	9	8
RESERVED							DIO1
R-0h							W-0h
7	6	5	4	3	2	1	0
RESERVED							DIO0
R-0h							W-0h

表 13-13. DOUT3_0 寄存器字段说明

位	字段	类型	复位	说明
31-25	RESERVED	R	0h	保留
24	DIO3	W	0h	如果相应的 DOE31_0 位域已置位，则设置配置为 DIO#3 的引脚的状态。
23-17	RESERVED	R	0h	保留
16	DIO2	W	0h	如果相应的 DOE31_0 位域已置位，则设置配置为 DIO#2 的引脚的状态。
15-9	RESERVED	R	0h	保留
8	DIO1	W	0h	如果相应的 DOE31_0 位域已置位，则设置配置为 DIO#1 的引脚的状态。
7-1	RESERVED	R	0h	保留
0	DIO0	W	0h	如果相应的 DOE31_0 位域已置位，则设置配置为 DIO#0 的引脚的状态。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.2.2 DOUT7_4 寄存器（偏移 = 4h）[复位 = 00000000h]

图 13-9 展示了 DOUT7_4，表 13-14 对其进行了介绍。

返回到汇总表。

数据输出 4 至 7
用于对 DOUT31_0 中每个位进行字节访问的别名寄存器

图 13-9. DOUT7_4 寄存器

31	30	29	28	27	26	25	24
RESERVED							DIO7
R-0h							W-0h
23	22	21	20	19	18	17	16
RESERVED							DIO6
R-0h							W-0h
15	14	13	12	11	10	9	8
RESERVED							DIO5
R-0h							W-0h
7	6	5	4	3	2	1	0
RESERVED							DIO4
R-0h							W-0h

表 13-14. DOUT7_4 寄存器字段说明

位	字段	类型	复位	说明
31-25	RESERVED	R	0h	保留
24	DIO7	W	0h	如果相应的 DOE31_0 位域已置位，则设置配置为 DIO#7 的引脚的状态。
23-17	RESERVED	R	0h	保留
16	DIO6	W	0h	如果相应的 DOE31_0 位域已置位，则设置配置为 DIO#6 的引脚的状态。
15-9	RESERVED	R	0h	保留
8	DIO5	W	0h	如果相应的 DOE31_0 位域已置位，则设置配置为 DIO#5 的引脚的状态。
7-1	RESERVED	R	0h	保留
0	DIO4	W	0h	如果相应的 DOE31_0 位域已置位，则设置配置为 DIO#4 的引脚的状态。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.2.3 DOUT11_8 寄存器（偏移 = 8h）[复位 = 00000000h]

图 13-10 展示了 DOUT11_8，表 13-15 对其进行了介绍。

返回到汇总表。

数据输出 8 至 11
用于对 DOUT31_0 中每个位进行字节访问的别名寄存器

图 13-10. DOUT11_8 寄存器

31	30	29	28	27	26	25	24
RESERVED							DIO11
R-0h							W-0h
23	22	21	20	19	18	17	16
RESERVED							DIO10
R-0h							W-0h
15	14	13	12	11	10	9	8
RESERVED							DIO9
R-0h							W-0h
7	6	5	4	3	2	1	0
RESERVED							DIO8
R-0h							W-0h

表 13-15. DOUT11_8 寄存器字段说明

位	字段	类型	复位	说明
31-25	RESERVED	R	0h	保留
24	DIO11	W	0h	如果相应的 DOE31_0 位域已置位，则设置配置为 DIO#11 的引脚的状态。
23-17	RESERVED	R	0h	保留
16	DIO10	W	0h	如果相应的 DOE31_0 位域已置位，则设置配置为 DIO#10 的引脚的状态。
15-9	RESERVED	R	0h	保留
8	DIO9	W	0h	如果相应的 DOE31_0 位域已置位，则设置配置为 DIO#9 的引脚的状态。
7-1	RESERVED	R	0h	保留
0	DIO8	W	0h	如果相应的 DOE31_0 位域已置位，则设置配置为 DIO#8 的引脚的状态。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.2.4 DOUT15_12 寄存器（偏移 = Ch）[复位 = 00000000h]

图 13-11 展示了 DOUT15_12，表 13-16 对其进行了介绍。

返回到汇总表。

数据输出 12 至 15
用于对 DOUT31_0 中每个位进行字节访问的别名寄存器

图 13-11. DOUT15_12 寄存器

31	30	29	28	27	26	25	24
RESERVED							DIO15
R-0h							W-0h
23	22	21	20	19	18	17	16
RESERVED							DIO14
R-0h							W-0h
15	14	13	12	11	10	9	8
RESERVED							DIO13
R-0h							W-0h
7	6	5	4	3	2	1	0
RESERVED							DIO12
R-0h							W-0h

表 13-16. DOUT15_12 寄存器字段说明

位	字段	类型	复位	说明
31-25	RESERVED	R	0h	保留
24	DIO15	W	0h	如果相应的 DOE31_0 位域已置位，则设置配置为 DIO#15 的引脚的状态。
23-17	RESERVED	R	0h	保留
16	DIO14	W	0h	如果相应的 DOE31_0 位域已置位，则设置配置为 DIO#14 的引脚的状态。
15-9	RESERVED	R	0h	保留
8	DIO13	W	0h	如果相应的 DOE31_0 位域已置位，则设置配置为 DIO#13 的引脚的状态。
7-1	RESERVED	R	0h	保留
0	DIO12	W	0h	如果相应的 DOE31_0 位域已置位，则设置配置为 DIO#12 的引脚的状态。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.2.5 DOUT19_16 寄存器（偏移 = 10h）[复位 = 00000000h]

图 13-12 展示了 DOUT19_16，表 13-17 对其进行了介绍。

返回到汇总表。

数据输出 16 至 19
用于对 DOUT31_0 中每个位进行字节访问的别名寄存器

图 13-12. DOUT19_16 寄存器

31	30	29	28	27	26	25	24
RESERVED							DIO19
R-0h							W-0h
23	22	21	20	19	18	17	16
RESERVED							DIO18
R-0h							W-0h
15	14	13	12	11	10	9	8
RESERVED							DIO17
R-0h							W-0h
7	6	5	4	3	2	1	0
RESERVED							DIO16
R-0h							W-0h

表 13-17. DOUT19_16 寄存器字段说明

位	字段	类型	复位	说明
31-25	RESERVED	R	0h	保留
24	DIO19	W	0h	如果相应的 DOE31_0 位域已置位，则设置配置为 DIO#19 的引脚的状态。
23-17	RESERVED	R	0h	保留
16	DIO18	W	0h	如果相应的 DOE31_0 位域已置位，则设置配置为 DIO#18 的引脚的状态。
15-9	RESERVED	R	0h	保留
8	DIO17	W	0h	如果相应的 DOE31_0 位域已置位，则设置配置为 DIO#17 的引脚的状态。
7-1	RESERVED	R	0h	保留
0	DIO16	W	0h	如果相应的 DOE31_0 位域已置位，则设置配置为 DIO#16 的引脚的状态。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.2.6 DOUT23_20 寄存器（偏移 = 14h）[复位 = 00000000h]

图 13-13 展示了 DOUT23_20，表 13-18 对其进行了介绍。

返回到汇总表。

数据输出 20 至 23
用于对 DOUT31_0 中每个位进行字节访问的别名寄存器

图 13-13. DOUT23_20 寄存器

31	30	29	28	27	26	25	24
RESERVED							DIO23
R-0h							W-0h
23	22	21	20	19	18	17	16
RESERVED							DIO22
R-0h							W-0h
15	14	13	12	11	10	9	8
RESERVED							DIO21
R-0h							W-0h
7	6	5	4	3	2	1	0
RESERVED							DIO20
R-0h							W-0h

表 13-18. DOUT23_20 寄存器字段说明

位	字段	类型	复位	说明
31-25	RESERVED	R	0h	保留
24	DIO23	W	0h	如果相应的 DOE31_0 位域已置位，则设置配置为 DIO#23 的引脚的状态。
23-17	RESERVED	R	0h	保留
16	DIO22	W	0h	如果相应的 DOE31_0 位域已置位，则设置配置为 DIO#22 的引脚的状态。
15-9	RESERVED	R	0h	保留
8	DIO21	W	0h	如果相应的 DOE31_0 位域已置位，则设置配置为 DIO#21 的引脚的状态。
7-1	RESERVED	R	0h	保留
0	DIO20	W	0h	如果相应的 DOE31_0 位域已置位，则设置配置为 DIO#20 的引脚的状态。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.2.7 DOUT27_24 寄存器（偏移 = 18h）[复位 = 00000000h]

图 13-14 展示了 DOUT27_24，表 13-19 对其进行了介绍。

返回到汇总表。

数据输出 24 至 27
用于对 DOUT31_0 中每个位进行字节访问的别名寄存器

图 13-14. DOUT27_24 寄存器

31	30	29	28	27	26	25	24
RESERVED							DIO27
R-0h							W-0h
23	22	21	20	19	18	17	16
RESERVED							DIO26
R-0h							W-0h
15	14	13	12	11	10	9	8
RESERVED							DIO25
R-0h							W-0h
7	6	5	4	3	2	1	0
RESERVED							DIO24
R-0h							W-0h

表 13-19. DOUT27_24 寄存器字段说明

位	字段	类型	复位	说明
31-25	RESERVED	R	0h	保留
24	DIO27	W	0h	如果相应的 DOE31_0 位域已置位，则设置配置为 DIO#27 的引脚的状态。
23-17	RESERVED	R	0h	保留
16	DIO26	W	0h	如果相应的 DOE31_0 位域已置位，则设置配置为 DIO#26 的引脚的状态。
15-9	RESERVED	R	0h	保留
8	DIO25	W	0h	如果相应的 DOE31_0 位域已置位，则设置配置为 DIO#25 的引脚的状态。
7-1	RESERVED	R	0h	保留
0	DIO24	W	0h	如果相应的 DOE31_0 位域已置位，则设置配置为 DIO#24 的引脚的状态。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.2.8 DOUT31_28 寄存器（偏移 = 1Ch）[复位 = 00000000h]

图 13-15 展示了 DOUT31_28，表 13-20 对其进行了介绍。

返回到汇总表。

数据输出 28 至 31
用于对 DOUT31_0 中每个位进行字节访问的别名寄存器

图 13-15. DOUT31_28 寄存器

31	30	29	28	27	26	25	24
RESERVED							DIO31
R-0h							W-0h
23	22	21	20	19	18	17	16
RESERVED							DIO30
R-0h							W-0h
15	14	13	12	11	10	9	8
RESERVED							DIO29
R-0h							W-0h
7	6	5	4	3	2	1	0
RESERVED							DIO28
R-0h							W-0h

表 13-20. DOUT31_28 寄存器字段说明

位	字段	类型	复位	说明
31-25	RESERVED	R	0h	保留
24	DIO31	W	0h	如果相应的 DOE31_0 位域已置位，则设置配置为 DIO#31 的引脚的状态。
23-17	RESERVED	R	0h	保留
16	DIO30	W	0h	如果相应的 DOE31_0 位域已置位，则设置配置为 DIO#30 的引脚的状态。
15-9	RESERVED	R	0h	保留
8	DIO29	W	0h	如果相应的 DOE31_0 位域已置位，则设置配置为 DIO#29 的引脚的状态。
7-1	RESERVED	R	0h	保留
0	DIO28	W	0h	如果相应的 DOE31_0 位域已置位，则设置配置为 DIO#28 的引脚的状态。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.2.9 DOUT31_0 寄存器（偏移 = 80h）[复位 = 00000000h]

图 13-16 展示了 DOUT31_0，表 13-21 对其进行了介绍。

返回到汇总表。

DIO 0 至 31 的数据输出

图 13-16. DOUT31_0 寄存器

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

表 13-21. DOUT31_0 寄存器字段说明

位	字段	类型	复位	说明
31	DIO31	R/W	0h	DIO 31 的数据输出
30	DIO30	R/W	0h	DIO 30 的数据输出
29	DIO29	R/W	0h	DIO 29 的数据输出
28	DIO28	R/W	0h	DIO 28 的数据输出
27	DIO27	R/W	0h	DIO 27 的数据输出
26	DIO26	R/W	0h	DIO 26 的数据输出
25	DIO25	R/W	0h	DIO 25 的数据输出
24	DIO24	R/W	0h	DIO 24 的数据输出
23	DIO23	R/W	0h	DIO 23 的数据输出
22	DIO22	R/W	0h	DIO 22 的数据输出
21	DIO21	R/W	0h	DIO 21 的数据输出
20	DIO20	R/W	0h	DIO 20 的数据输出
19	DIO19	R/W	0h	DIO 19 的数据输出
18	DIO18	R/W	0h	DIO 18 的数据输出
17	DIO17	R/W	0h	DIO 17 的数据输出
16	DIO16	R/W	0h	DIO 16 的数据输出
15	DIO15	R/W	0h	DIO 15 的数据输出
14	DIO14	R/W	0h	DIO 14 的数据输出
13	DIO13	R/W	0h	DIO 13 的数据输出
12	DIO12	R/W	0h	DIO 12 的数据输出
11	DIO11	R/W	0h	DIO 11 的数据输出
10	DIO10	R/W	0h	DIO 10 的数据输出
9	DIO9	R/W	0h	DIO 9 的数据输出

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-21. DOUT31_0 寄存器字段说明（续）

位	字段	类型	复位	说明
8	DIO8	R/W	0h	DIO 8 的数据输出
7	DIO7	R/W	0h	DIO 7 的数据输出
6	DIO6	R/W	0h	DIO 6 的数据输出
5	DIO5	R/W	0h	DIO 5 的数据输出
4	DIO4	R/W	0h	DIO 4 的数据输出
3	DIO3	R/W	0h	DIO 3 的数据输出
2	DIO2	R/W	0h	DIO 2 的数据输出
1	DIO1	R/W	0h	DIO 1 的数据输出
0	DIO0	R/W	0h	DIO 0 的数据输出

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.2.10 DOUTSET31_0 寄存器（偏移 = 90h）[复位 = 00000000h]

图 13-17 展示了 DOUTSET31_0，表 13-22 对其进行了介绍。

返回到汇总表。

数据输出置位
向位元位置写入 1 会置位 DOUT31_0 寄存器中的相应位

图 13-17. DOUTSET31_0 寄存器

31	30	29	28	27	26	25	24
DIO31	DIO30	DIO29	DIO28	DIO27	DIO26	DIO25	DIO24
W1S-0h	W1S-0h	W1S-0h	W1S-0h	W1S-0h	W1S-0h	W1S-0h	W1S-0h
23	22	21	20	19	18	17	16
DIO23	DIO22	DIO21	DIO20	DIO19	DIO18	DIO17	DIO16
W1S-0h	W1S-0h	W1S-0h	W1S-0h	W1S-0h	W1S-0h	W1S-0h	W1S-0h
15	14	13	12	11	10	9	8
DIO15	DIO14	DIO13	DIO12	DIO11	DIO10	DIO9	DIO8
W1S-0h	W1S-0h	W1S-0h	W1S-0h	W1S-0h	W1S-0h	W1S-0h	W1S-0h
7	6	5	4	3	2	1	0
DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1	DIO0
W1S-0h	W1S-0h	W1S-0h	W1S-0h	W1S-0h	W1S-0h	W1S-0h	W1S-0h

表 13-22. DOUTSET31_0 寄存器字段说明

位	字段	类型	复位	说明
31	DIO31	W1S	0h	位 31 置位
30	DIO30	W1S	0h	位 30 置位
29	DIO29	W1S	0h	位 29 置位
28	DIO28	W1S	0h	位 28 置位
27	DIO27	W1S	0h	位 27 置位
26	DIO26	W1S	0h	位 26 置位
25	DIO25	W1S	0h	位 25 置位
24	DIO24	W1S	0h	位 24 置位
23	DIO23	W1S	0h	位 23 置位
22	DIO22	W1S	0h	位 22 置位
21	DIO21	W1S	0h	位 21 置位
20	DIO20	W1S	0h	位 20 置位
19	DIO19	W1S	0h	位 19 置位
18	DIO18	W1S	0h	位 18 置位
17	DIO17	W1S	0h	位 17 置位
16	DIO16	W1S	0h	位 16 置位
15	DIO15	W1S	0h	位 15 置位
14	DIO14	W1S	0h	位 14 置位
13	DIO13	W1S	0h	位 13 置位
12	DIO12	W1S	0h	位 12 置位
11	DIO11	W1S	0h	位 11 置位
10	DIO10	W1S	0h	位 10 置位
9	DIO9	W1S	0h	位 9 置位

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-22. DOUTSET31_0 寄存器字段说明（续）

位	字段	类型	复位	说明
8	DIO8	W1S	0h	位 8 置位
7	DIO7	W1S	0h	位 7 置位
6	DIO6	W1S	0h	位 6 置位
5	DIO5	W1S	0h	位 5 置位
4	DIO4	W1S	0h	位 4 置位
3	DIO3	W1S	0h	位 3 置位
2	DIO2	W1S	0h	位 2 置位
1	DIO1	W1S	0h	位 1 置位
0	DIO0	W1S	0h	位 0 置位

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.2.11 DOUTCLR31_0 寄存器（偏移 = A0h）[复位 = 00000000h]

图 13-18 展示了 DOUTCLR31_0，表 13-23 对其进行了介绍。

返回到汇总表。

数据输出清除
向位元位置写入 1 会清除 DOUT31_0 寄存器中的相应位

图 13-18. DOUTCLR31_0 寄存器

31	30	29	28	27	26	25	24
DIO31	DIO30	DIO29	DIO28	DIO27	DIO26	DIO25	DIO24
W1C-0h	W1C-0h	W1C-0h	W1C-0h	W1C-0h	W1C-0h	W1C-0h	W1C-0h
23	22	21	20	19	18	17	16
DIO23	DIO22	DIO21	DIO20	DIO19	DIO18	DIO17	DIO16
W1C-0h	W1C-0h	W1C-0h	W1C-0h	W1C-0h	W1C-0h	W1C-0h	W1C-0h
15	14	13	12	11	10	9	8
DIO15	DIO14	DIO13	DIO12	DIO11	DIO10	DIO9	DIO8
W1C-0h	W1C-0h	W1C-0h	W1C-0h	W1C-0h	W1C-0h	W1C-0h	W1C-0h
7	6	5	4	3	2	1	0
DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1	DIO0
W1C-0h	W1C-0h	W1C-0h	W1C-0h	W1C-0h	W1C-0h	W1C-0h	W1C-0h

表 13-23. DOUTCLR31_0 寄存器字段说明

位	字段	类型	复位	说明
31	DIO31	W1C	0h	清除位 31
30	DIO30	W1C	0h	清除位 30
29	DIO29	W1C	0h	清除位 29
28	DIO28	W1C	0h	清除位 28
27	DIO27	W1C	0h	清除位 27
26	DIO26	W1C	0h	清除位 26
25	DIO25	W1C	0h	清除位 25
24	DIO24	W1C	0h	清除位 24
23	DIO23	W1C	0h	清除位 23
22	DIO22	W1C	0h	清除位 22
21	DIO21	W1C	0h	清除位 21
20	DIO20	W1C	0h	清除位 20
19	DIO19	W1C	0h	清除位 19
18	DIO18	W1C	0h	清除位 18
17	DIO17	W1C	0h	清除位 17
16	DIO16	W1C	0h	清除位 16
15	DIO15	W1C	0h	清除位 15
14	DIO14	W1C	0h	清除位 14
13	DIO13	W1C	0h	清除位 13
12	DIO12	W1C	0h	清除位 12
11	DIO11	W1C	0h	清除位 11
10	DIO10	W1C	0h	清除位 10
9	DIO9	W1C	0h	清除位 9

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-23. DOUTCLR31_0 寄存器字段说明（续）

位	字段	类型	复位	说明
8	DIO8	W1C	0h	清除位 8
7	DIO7	W1C	0h	清除位 7
6	DIO6	W1C	0h	清除位 6
5	DIO5	W1C	0h	清除位 5
4	DIO4	W1C	0h	清除位 4
3	DIO3	W1C	0h	清除位 3
2	DIO2	W1C	0h	清除位 2
1	DIO1	W1C	0h	清除位 1
0	DIO0	W1C	0h	清除位 0

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.2.12 DOUTTGL31_0 寄存器（偏移 = B0h）[复位 = 00000000h]

图 13-19 展示了 DOUTTGL31_0，表 13-24 对其进行了介绍。

返回到汇总表。

数据输出切换
向位元位置写入 1 将反转相应的 DIO 输出。

图 13-19. DOUTTGL31_0 寄存器

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

表 13-24. DOUTTGL31_0 寄存器字段说明

位	字段	类型	复位	说明
31	DIO31	R/W	0h	切换位 31
30	DIO30	R/W	0h	切换位 30
29	DIO29	R/W	0h	切换位 29
28	DIO28	R/W	0h	切换位 28
27	DIO27	R/W	0h	切换位 27
26	DIO26	R/W	0h	切换位 26
25	DIO25	R/W	0h	切换位 25
24	DIO24	R/W	0h	切换位 24
23	DIO23	R/W	0h	切换位 23
22	DIO22	R/W	0h	切换位 22
21	DIO21	R/W	0h	切换位 21
20	DIO20	R/W	0h	切换位 20
19	DIO19	R/W	0h	切换位 19
18	DIO18	R/W	0h	切换位 18
17	DIO17	R/W	0h	切换位 17
16	DIO16	R/W	0h	切换位 16
15	DIO15	R/W	0h	切换位 15
14	DIO14	R/W	0h	切换位 14
13	DIO13	R/W	0h	切换位 13
12	DIO12	R/W	0h	切换位 12
11	DIO11	R/W	0h	切换位 11
10	DIO10	R/W	0h	切换位 10
9	DIO9	R/W	0h	切换位 9

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-24. DOUTTGL31_0 寄存器字段说明（续）

位	字段	类型	复位	说明
8	DIO8	R/W	0h	切换位 8
7	DIO7	R/W	0h	切换位 7
6	DIO6	R/W	0h	切换位 6
5	DIO5	R/W	0h	切换位 5
4	DIO4	R/W	0h	切换位 4
3	DIO3	R/W	0h	切换位 3
2	DIO2	R/W	0h	切换位 2
1	DIO1	R/W	0h	切换位 1
0	DIO0	R/W	0h	切换位 0

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.2.13 DIN31_0 寄存器（偏移 = C0h）[复位 = 00000000h]

图 13-20 展示了 DIN31_0，表 13-25 对其进行了介绍。

返回到汇总表。

DIO 0 至 31 的数据输入

图 13-20. DIN31_0 寄存器

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

表 13-25. DIN31_0 寄存器字段说明

位	字段	类型	复位	说明
31	DIO31	R	0h	DIO 31 的数据输入
30	DIO30	R	0h	DIO 30 的数据输入
29	DIO29	R	0h	DIO 29 的数据输入
28	DIO28	R	0h	DIO 28 的数据输入
27	DIO27	R	0h	DIO 27 的数据输入
26	DIO26	R	0h	DIO 26 的数据输入
25	DIO25	R	0h	DIO 25 的数据输入
24	DIO24	R	0h	DIO 24 的数据输入
23	DIO23	R	0h	DIO 23 的数据输入
22	DIO22	R	0h	DIO 22 的数据输入
21	DIO21	R	0h	DIO 21 的数据输入
20	DIO20	R	0h	DIO 20 的数据输入
19	DIO19	R	0h	DIO 19 的数据输入
18	DIO18	R	0h	DIO 18 的数据输入
17	DIO17	R	0h	DIO 17 的数据输入
16	DIO16	R	0h	DIO 16 的数据输入
15	DIO15	R	0h	DIO 15 的数据输入
14	DIO14	R	0h	DIO 14 的数据输入
13	DIO13	R	0h	DIO 13 的数据输入
12	DIO12	R	0h	DIO 12 的数据输入
11	DIO11	R	0h	DIO 11 的数据输入
10	DIO10	R	0h	DIO 10 的数据输入
9	DIO9	R	0h	DIO 9 的数据输入

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-25. DIN31_0 寄存器字段说明（续）

位	字段	类型	复位	说明
8	DIO8	R	0h	DIO 8 的数据输入
7	DIO7	R	0h	DIO 7 的数据输入
6	DIO6	R	0h	DIO 6 的数据输入
5	DIO5	R	0h	DIO 5 的数据输入
4	DIO4	R	0h	DIO 4 的数据输入
3	DIO3	R	0h	DIO 3 的数据输入
2	DIO2	R	0h	DIO 2 的数据输入
1	DIO1	R	0h	DIO 1 的数据输入
0	DIO0	R	0h	DIO 0 的数据输入

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.2.14 DOE31_0 寄存器（偏移 = D0h）[复位 = 00000000h]

图 13-21 展示了 DOE31_0，表 13-26 对其进行了介绍。

返回到汇总表。

DIO 0 至 31 的数据输出使能

图 13-21. DOE31_0 寄存器

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

表 13-26. DOE31_0 寄存器字段说明

位	字段	类型	复位	说明
31	DIO31	R/W	0h	DIO 31 的数据输出使能
30	DIO30	R/W	0h	DIO 30 的数据输出使能
29	DIO29	R/W	0h	DIO 29 的数据输出使能
28	DIO28	R/W	0h	DIO 28 的数据输出使能
27	DIO27	R/W	0h	DIO 27 的数据输出使能
26	DIO26	R/W	0h	DIO 26 的数据输出使能
25	DIO25	R/W	0h	DIO 25 的数据输出使能
24	DIO24	R/W	0h	DIO 24 的数据输出使能
23	DIO23	R/W	0h	DIO 23 的数据输出使能
22	DIO22	R/W	0h	DIO 22 的数据输出使能
21	DIO21	R/W	0h	DIO 21 的数据输出使能
20	DIO20	R/W	0h	DIO 20 的数据输出使能
19	DIO19	R/W	0h	DIO 19 的数据输出使能
18	DIO18	R/W	0h	DIO 18 的数据输出使能
17	DIO17	R/W	0h	DIO 17 的数据输出使能
16	DIO16	R/W	0h	DIO 16 的数据输出使能
15	DIO15	R/W	0h	DIO 15 的数据输出使能
14	DIO14	R/W	0h	DIO 14 的数据输出使能
13	DIO13	R/W	0h	DIO 13 的数据输出使能
12	DIO12	R/W	0h	DIO 12 的数据输出使能
11	DIO11	R/W	0h	DIO 11 的数据输出使能
10	DIO10	R/W	0h	DIO 10 的数据输出使能
9	DIO9	R/W	0h	DIO 9 的数据输出使能

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-26. DOE31_0 寄存器字段说明（续）

位	字段	类型	复位	说明
8	DIO8	R/W	0h	DIO 8 的数据输出使能
7	DIO7	R/W	0h	DIO 7 的数据输出使能
6	DIO6	R/W	0h	DIO 6 的数据输出使能
5	DIO5	R/W	0h	DIO 5 的数据输出使能
4	DIO4	R/W	0h	DIO 4 的数据输出使能
3	DIO3	R/W	0h	DIO 3 的数据输出使能
2	DIO2	R/W	0h	DIO 2 的数据输出使能
1	DIO1	R/W	0h	DIO 1 的数据输出使能
0	DIO0	R/W	0h	DIO 0 的数据输出使能

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.2.15 EVFLAGS31_0 寄存器（偏移 = E0h）[复位 = 00000000h]

图 13-22 展示了 EVFLAGS31_0，表 13-27 对其进行了介绍。

返回到汇总表。

DIO 0 至 31 的事件寄存器
读取该寄存器时，已触发事件将返回 1，未触发事件将返回 0。
向位域写入 1 将清除该事件。
事件配置在 MCU IOC 内完成，例如 DIO #0 的事件在 IOC:IOCFG0.EDGE_DET 和 IOC:IOCFG0.EDGE_IRQ_EN 中配置。

图 13-22. EVFLAGS31_0 寄存器

31	30	29	28	27	26	25	24
DIO31	DIO30	DIO29	DIO28	DIO27	DIO26	DIO25	DIO24
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
DIO23	DIO22	DIO21	DIO20	DIO19	DIO18	DIO17	DIO16
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
DIO15	DIO14	DIO13	DIO12	DIO11	DIO10	DIO9	DIO8
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
DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1	DIO0
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h

表 13-27. EVFLAGS31_0 寄存器字段说明

位	字段	类型	复位	说明
31	DIO31	R/W1C	0h	DIO 31 的事件
30	DIO30	R/W1C	0h	DIO 30 的事件
29	DIO29	R/W1C	0h	DIO 29 的事件
28	DIO28	R/W1C	0h	DIO 28 的事件
27	DIO27	R/W1C	0h	DIO 27 的事件
26	DIO26	R/W1C	0h	DIO 26 的事件
25	DIO25	R/W1C	0h	DIO 25 的事件
24	DIO24	R/W1C	0h	DIO 24 的事件
23	DIO23	R/W1C	0h	DIO 23 的事件
22	DIO22	R/W1C	0h	DIO 22 的事件
21	DIO21	R/W1C	0h	DIO 21 的事件
20	DIO20	R/W1C	0h	DIO 20 的事件
19	DIO19	R/W1C	0h	DIO 19 的事件
18	DIO18	R/W1C	0h	DIO 18 的事件
17	DIO17	R/W1C	0h	DIO 17 的事件
16	DIO16	R/W1C	0h	DIO 16 的事件
15	DIO15	R/W1C	0h	DIO 15 的事件
14	DIO14	R/W1C	0h	DIO 14 的事件
13	DIO13	R/W1C	0h	DIO 13 的事件
12	DIO12	R/W1C	0h	DIO 12 的事件

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-27. EVFLAGS31_0 寄存器字段说明（续）

位	字段	类型	复位	说明
11	DIO11	R/W1C	0h	DIO 11 的事件
10	DIO10	R/W1C	0h	DIO 10 的事件
9	DIO9	R/W1C	0h	DIO 9 的事件
8	DIO8	R/W1C	0h	DIO 8 的事件
7	DIO7	R/W1C	0h	DIO 7 的事件
6	DIO6	R/W1C	0h	DIO 6 的事件
5	DIO5	R/W1C	0h	DIO 5 的事件
4	DIO4	R/W1C	0h	DIO 4 的事件
3	DIO3	R/W1C	0h	DIO 3 的事件
2	DIO2	R/W1C	0h	DIO 2 的事件
1	DIO1	R/W1C	0h	DIO 1 的事件
0	DIO0	R/W1C	0h	DIO 0 的事件

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.3 IOC 寄存器

表 13-28 列出了 IOC 寄存器的存储器映射寄存器。表 13-28 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

表 13-28. IOC 寄存器

偏移	首字母缩写词	寄存器名称	部分
0h	IOCFG0	DIO0 的配置	IOCFG0 寄存器（偏移 = 0h）[复位 = 00006000h]
4h	IOCFG1	DIO1 的配置	IOCFG1 寄存器（偏移 = 4h）[复位 = 00006000h]
8h	IOCFG2	DIO2 的配置	IOCFG2 寄存器（偏移 = 8h）[复位 = 00006000h]
Ch	IOCFG3	DIO3 的配置	IOCFG3 寄存器（偏移 = Ch）[复位 = 00006000h]
10h	IOCFG4	DIO4 的配置	IOCFG4 寄存器（偏移 = 10h）[复位 = 00006000h]
14h	IOCFG5	DIO5 的配置	IOCFG5 寄存器（偏移 = 14h）[复位 = 00006000h]
18h	IOCFG6	DIO6 的配置	IOCFG6 寄存器（偏移 = 18h）[复位 = 00006000h]
1Ch	IOCFG7	DIO7 的配置	IOCFG7 寄存器（偏移 = 1Ch）[复位 = 00006000h]
20h	IOCFG8	DIO8 的配置	IOCFG8 寄存器（偏移 = 20h）[复位 = 00006000h]
24h	IOCFG9	DIO9 的配置	IOCFG9 寄存器（偏移 = 24h）[复位 = 00006000h]
28h	IOCFG10	DIO10 的配置	IOCFG10 寄存器（偏移 = 28h）[复位 = 00006000h]
2Ch	IOCFG11	DIO11 的配置	IOCFG11 寄存器（偏移 = 2Ch）[复位 = 00006000h]
30h	IOCFG12	DIO12 的配置	IOCFG12 寄存器（偏移 = 30h）[复位 = 00006000h]
34h	IOCFG13	DIO13 的配置	IOCFG13 寄存器（偏移 = 34h）[复位 = 00006000h]
38h	IOCFG14	DIO14 的配置	IOCFG14 寄存器（偏移 = 38h）[复位 = 00006000h]
3Ch	IOCFG15	DIO15 的配置	IOCFG15 寄存器（偏移 = 3Ch）[复位 = 00006000h]
40h	IOCFG16	DIO16 的配置	IOCFG16 寄存器（偏移 = 40h）[复位 = 00086000h]
44h	IOCFG17	DIO17 的配置	IOCFG17 寄存器（偏移 = 44h）[复位 = 00106000h]
48h	IOCFG18	DIO18 的配置	IOCFG18 寄存器（偏移 = 48h）[复位 = 00006000h]
4Ch	IOCFG19	DIO19 的配置	IOCFG19 寄存器（偏移 = 4Ch）[复位 = 00006000h]
50h	IOCFG20	DIO20 的配置	IOCFG20 寄存器（偏移 = 50h）[复位 = 00006000h]
54h	IOCFG21	DIO21 的配置	IOCFG21 寄存器（偏移 = 54h）[复位 = 00006000h]
58h	IOCFG22	DIO22 的配置	IOCFG22 寄存器（偏移 = 58h）[复位 = 00006000h]

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-28. IOC 寄存器（续）

偏移	首字母缩写词	寄存器名称	部分
5Ch	IOCFG23	DIO23 的配置	IOCFG23 寄存器（偏移 = 5Ch）[复位 = 00006000h]
60h	IOCFG24	DIO24 的配置	IOCFG24 寄存器（偏移 = 60h）[复位 = 00006000h]
64h	IOCFG25	DIO25 的配置	IOCFG25 寄存器（偏移 = 64h）[复位 = 00006000h]
68h	IOCFG26	DIO26 的配置	IOCFG26 寄存器（偏移 = 68h）[复位 = 00006000h]
6Ch	IOCFG27	DIO27 的配置	IOCFG27 寄存器（偏移 = 6Ch）[复位 = 00006000h]
70h	IOCFG28	DIO28 的配置	IOCFG28 寄存器（偏移 = 70h）[复位 = 00006000h]
74h	IOCFG29	DIO29 的配置	IOCFG29 寄存器（偏移 = 74h）[复位 = 00006000h]
78h	IOCFG30	DIO30 的配置	IOCFG30 寄存器（偏移 = 78h）[复位 = 00006000h]
7Ch	IOCFG31	DIO31 的配置	IOCFG31 寄存器（偏移 = 7Ch）[复位 = 00006000h]

复杂的位访问类型经过编码可适应小型表单元。表 13-29 展示了适用于此部分中访问类型的代码。

表 13-29. IOC 访问类型代码

访问类型	代码	说明
读取类型		
R	R	读取
写入类型		
W	W	写入
复位或默认值		
-n		复位后的值或默认值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.3.1 IOCFG0 寄存器（偏移 = 0h）[复位 = 00006000h]

图 13-23 展示了 IOCFG0，表 13-30 对其进行了介绍。

返回到汇总表。

DIO0 的配置

图 13-23. IOCFG0 寄存器

31	30	29	28	27	26	25	24
RESERVED	HYST_EN	IE	WU_CFG		IOMODE		
R-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h		
23	22	21	20	19	18	17	16
IOEV_AON_PROG2_EN	IOEV_AON_PROG1_EN	IOEV_AON_PROG0_EN	RESERVED		EDGE_IRQ_EN	EDGE_DET	
R/W-0h	R/W-0h	R/W-0h	R-0h		R/W-0h	R/W-0h	
15	14	13	12	11	10	9	8
RESERVED	PULL_CTL		SLEW_RED	IOCURR		IOSTR	
R-0h	R/W-3h		R/W-0h	R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
IOEV_RTC_EN	IOEV_MCU_WU_EN	PORT_ID					
R/W-0h	R/W-0h	R/W-0h					

表 13-30. IOCFG0 寄存器字段说明

位	字段	类型	复位	说明
31	RESERVED	R	0h	保留
30	HYST_EN	R/W	0h	0：输入迟滞禁用 1：输入迟滞使能
29	IE	R/W	0h	0：输入禁用 1：输入使能 注意：如果 IO 配置为 AUX PORT_ID = 0x08，则使能将被忽略。
28-27	WU_CFG	R/W	0h	如果 DIO 配置为 GPIO 或非 AON 外设信号（PORT_ID 0x00 或 >0x08）： 00：无唤醒 01：无唤醒 10：如果该焊盘变为低电平，则从关断状态唤醒。 11：如果该焊盘变为高电平，则从关断状态唤醒。 如果 IO 配置为 AON 外设信号或 AUX PORT_ID 0x01-0x08，则该寄存器仅设置是否使能唤醒。 00、01：唤醒禁用 10、11：唤醒使能 极性通过 AON 寄存器进行控制。 注意：当 MSB 置位时，IOC 将停用 DIO 的输出使能。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-30. IOCFG0 寄存器字段说明（续）

位	字段	类型	复位	说明
26-24	IOMODE	R/W	0h	IO 模式 不适用于配置为 AON 外设信号和 AUX PORT_ID 0x01-0x08 的 IO AUX 有自己的 open_source/drain 配置。 0x2: 保留。未定义的行为。 0x3：保留。未定义的行为。 0h = NORMAL：正常输入/输出 1h = INV：反相输入/输出 4h = OPENDR：开漏， 正常输入/输出 5h = OPENDR_INV：开漏 反相输入/输出 6h = OPENSRC：开源 正常输入/输出 7h = OPENSRC_INV：开源 反相输入/输出
23	IOEV_AON_PROG2_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG2 事件置为有效 1：输入边沿检测将 AON_PROG2 事件置为有效
22	IOEV_AON_PROG1_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG1 事件置为有效 1：输入边沿检测将 AON_PROG1 事件置为有效
21	IOEV_AON_PROG0_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG0 事件置为有效 1：输入边沿检测将 AON_PROG0 事件置为有效
20-19	RESERVED	R	0h	保留
18	EDGE_IRQ_EN	R/W	0h	0：无中断生成 1：使能该 IO 的中断生成（仅在 EDGE_DET 已使能时有效）
17-16	EDGE_DET	R/W	0h	使能该 IO 上的边沿检测事件生成 0h = NONE：无边沿检测 1h = 负边沿检测 2h = 正边沿检测 3h = 正负边沿检测
15	RESERVED	R	0h	保留
14-13	PULL_CTL	R/W	3h	上拉/下拉控制 1h = DWN：下拉 2h = UP：上拉 3h = DIS：无上拉/下拉
12	SLEW_RED	R/W	0h	0：标称压摆率 1：在输出驱动器中降低压摆率。
11-10	IOCURR	R/W	0h	选择该 IO 的 IO 电流模式。 0h = 2MA：低电流 (LC) 模式：IOSTR 设置为 AUTO 时最小值为 2mA 1h = 4MA：高电流 (HC) 模式：IOSTR 设置为 AUTO 时最小值为 4mA 2h = 4_8MA：扩展电流 (EC) 模式：当 IOSTR 设为 AUTO 时，双驱动强度 IO 的最小值为 8mA（标准 IO 的最小值为 4mA）

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-30. IOCFG0 寄存器字段说明（续）

位	字段	类型	复位	说明
9-8	IOSTR	R/W	0h	选择该 IO 的驱动强度控制源。 此设置控制低电流 (LC) 模式的驱动强度。可在 IOCURR 中选择更高的驱动强度 0h = 自动驱动强度，由 AON BATMON 根据电池电压控制。（VDDS 时最小 2mA） 1h = 最小驱动强度，由 AON_IOC:IOSTRMIN 控制（默认值为 3.3V 时最小 2mA） 2h = MED：中等驱动强度，由 AON_IOC:IOSTRMED 控制（默认值为 2.5V 时最小 2mA） 3h = 最大驱动强度，由 AON_IOC:IOSTRMAX 控制（默认值为 1.8V 时最小 2mA）
7	IOEV_RTC_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 RTC 事件置为有效 1：输入边沿检测将 RTC 事件置为有效
6	IOEV_MCU_WU_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 MCU_WU 事件置为有效 1：输入边沿检测将 MCU_WU 事件置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-30. IOCFG0 寄存器字段说明（续）

位	字段	类型	复位	说明
5-0	PORT_ID	R/W	0h	选择 DIO0 的用途 注意：除明确要求更改 PORT_ID 值的情况外，不应写入此字段。 0h =通用 IO 7h = AON 32 KHz 时钟 (SCLK_LF) 8h = AUX IO 9h = SSI0_RX：SSI0 RX Ah = SSI0_TX：SSI0 TX Bh = SSI0_FSS：SSI0 FSS Ch = SSI0_CLK：SSI0 CLK Dh = I2C_MSSDA：I2C 数据 Eh = I2C_MSSCL：I2C 时钟 Fh = UART0_RX：UART0 RX 10h = UART0_TX：UART0 TX 11h = UART0_CTS：UART0 CTS 12h = UART0_RTS：UART0 RTS 13h = UART1_RX：UART1 RX 14h = UART1_TX：UART1 TX 15h = UART1_CTS：UART1 CTS 16h = UART1_RTS：UART1 RTS 17h = PORT_EVENT0：端口事件 0 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 18h = PORT_EVENT1： 端口事件 1 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 19h = PORT_EVENT2： 端口事件 2 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ah = PORT_EVENT3： 端口事件 3 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Bh = PORT_EVENT4： 端口事件 4 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ch = PORT_EVENT5： 端口事件 5 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Dh = PORT_EVENT6： 端口事件 6 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Eh = PORT_EVENT7： 端口事件 7 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 20h = CPU_SWV： CPU SWV 21h = SSI1_RX：SSI1 RX 22h = SSI1_TX：SSI1 TX 23h = SSI1_FSS：SSI1 FSS 24h = SSI1_CLK：SSI1 CLK 25h = I2S_AD0：I2S 数据 0 26h = I2S_AD1： I2S 数据 1 27h = I2S_WCLK： I2S WCLK 28h = I2S_BCLK：I2S BCLK 29h = I2S_MCLK：I2S MCLK 2Eh = 射频内核跟踪

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-30. IOCFG0 寄存器字段说明（续）

位	字段	类型	复位	说明
				2Fh = 射频内核数据输出 0 30h = 射频内核数据输出 1 31h = 射频内核数据输出 2 32h = 射频内核数据输出 3 33h = 射频内核数据输入 0 34h = 射频内核数据输入 1 35h = 射频内核 SMI 数据链路输出 36h = 射频内核 SMI 数据链路输入 37h = 射频内核 SMI 命令链路输出 38h = 射频内核 SMI 命令链路输入

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.3.2 IOCFG1 寄存器（偏移 = 4h）[复位 = 00006000h]

图 13-24 展示了 IOCFG1，表 13-31 对其进行了介绍。

返回到汇总表。

DIO1 的配置

图 13-24. IOCFG1 寄存器

31	30	29	28	27	26	25	24
RESERVED	HYST_EN	IE	WU_CFG		IOMODE		
R-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h		
23	22	21	20	19	18	17	16
IOEV_AON_PROG2_EN	IOEV_AON_PROG1_EN	IOEV_AON_PROG0_EN	RESERVED		EDGE_IRQ_EN	EDGE_DET	
R/W-0h	R/W-0h	R/W-0h	R-0h		R/W-0h	R/W-0h	
15	14	13	12	11	10	9	8
RESERVED	PULL_CTL		SLEW_RED	IOCURR		IOSTR	
R-0h	R/W-3h		R/W-0h	R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
IOEV_RTC_EN	IOEV_MCU_WU_EN	PORT_ID					
R/W-0h	R/W-0h	R/W-0h					

表 13-31. IOCFG1 寄存器字段说明

位	字段	类型	复位	说明
31	RESERVED	R	0h	保留
30	HYST_EN	R/W	0h	0：输入迟滞禁用 1：输入迟滞使能
29	IE	R/W	0h	0：输入禁用 1：输入使能 注意：如果 IO 配置为 AUX PORT_ID = 0x08，则使能将被忽略。
28-27	WU_CFG	R/W	0h	如果 DIO 配置为 GPIO 或非 AON 外设信号（PORT_ID 0x00 或 >0x08）： 00：无唤醒 01：无唤醒 10：如果该焊盘变为低电平，则从关断状态唤醒。 11：如果该焊盘变为高电平，则从关断状态唤醒。 如果 IO 配置为 AON 外设信号或 AUX PORT_ID 0x01-0x08，则该寄存器仅设置是否使能唤醒。 00、01：唤醒禁用 10、11：唤醒使能 极性通过 AON 寄存器进行控制。 注意：当 MSB 置位时，IOC 将停用 DIO 的输出使能。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-31. IOCFG1 寄存器字段说明（续）

位	字段	类型	复位	说明
26-24	IOMODE	R/W	0h	IO 模式 不适用于配置为 AON 外设信号和 AUX PORT_ID 0x01-0x08 的 IO AUX 有自己的 open_source/drain 配置。 0x2: 保留。未定义的行为。 0x3：保留。未定义的行为。 0h = NORMAL：正常输入/输出 1h = INV：反相输入/输出 4h = OPENDR： 开漏， 正常输入/输出 5h = OPENDR_INV：开漏 反相输入/输出 6h = OPENSRC：开源 正常输入/输出 7h = OPENSRC_INV：开源 反相输入/输出
23	IOEV_AON_PROG2_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG2 事件置为有效 1：输入边沿检测将 AON_PROG2 事件置为有效
22	IOEV_AON_PROG1_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG1 事件置为有效 1：输入边沿检测将 AON_PROG1 事件置为有效
21	IOEV_AON_PROG0_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG0 事件置为有效 1：输入边沿检测将 AON_PROG0 事件置为有效
20-19	RESERVED	R	0h	保留
18	EDGE_IRQ_EN	R/W	0h	0：无中断生成 1：使能该 IO 的中断生成（仅在 EDGE_DET 已使能时有效）
17-16	EDGE_DET	R/W	0h	使能该 IO 上的边沿检测事件生成 0h = NONE：无边沿检测 1h = 负边沿检测 2h = 正边沿检测 3h = 正负边沿检测
15	RESERVED	R	0h	保留
14-13	PULL_CTL	R/W	3h	上拉/下拉控制 1h = DWN：下拉 2h = UP：上拉 3h = DIS：无上拉/下拉
12	SLEW_RED	R/W	0h	0：标称压摆率 1：在输出驱动器中降低压摆率。
11-10	IOCURR	R/W	0h	选择该 IO 的 IO 电流模式。 0h = 2MA：低电流 (LC) 模式：IOSTR 设置为 AUTO 时最小值为 2mA 1h = 4MA：高电流 (HC) 模式：IOSTR 设置为 AUTO 时最小值为 4mA 2h = 4_8MA：扩展电流 (EC) 模式：当 IOSTR 设为 AUTO 时，双驱动强度 IO 的最小值为 8mA（标准 IO 的最小值为 4mA）

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-31. IOCFG1 寄存器字段说明（续）

位	字段	类型	复位	说明
9-8	IOSTR	R/W	0h	选择该 IO 的驱动强度控制源。 此设置控制低电流 (LC) 模式的驱动强度。可在 IOCURR 中选择更高的驱动强度 0h = 自动驱动强度，由 AON BATMON 根据电池电压控制。（VDDS 时最小 2mA） 1h = 最小驱动强度，由 AON_IOC:IOSTRMIN 控制（默认值为 3.3V 时最小 2mA） 2h = MED：中等驱动强度，由 AON_IOC:IOSTRMED 控制（默认值为 2.5V 时最小 2mA） 3h = 最大驱动强度，由 AON_IOC:IOSTRMAX 控制（默认值为 1.8V 时最小 2mA）
7	IOEV_RTC_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 RTC 事件置为有效 1：输入边沿检测将 RTC 事件置为有效
6	IOEV_MCU_WU_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 MCU_WU 事件置为有效 1：输入边沿检测将 MCU_WU 事件置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-31. IOCFG1 寄存器字段说明（续）

位	字段	类型	复位	说明
5-0	PORT_ID	R/W	0h	选择 DIO1 的用途 注意：除明确要求更改 PORT_ID 值的情况外，不应写入此字段。 0h =通用 IO 7h = AON 32 KHz 时钟 (SCLK_LF) 8h = AUX IO 9h = SSI0_RX：SSI0 RX Ah = SSI0_TX：SSI0 TX Bh = SSI0_FSS：SSI0 FSS Ch = SSI0_CLK：SSI0 CLK Dh = I2C_MSSDA：I2C 数据 Eh = I2C_MSSCL：I2C 时钟 Fh = UART0_RX：UART0 RX 10h = UART0_TX：UART0 TX 11h = UART0_CTS：UART0 CTS 12h = UART0_RTS：UART0 RTS 13h = UART1_RX：UART1 RX 14h = UART1_TX：UART1 TX 15h = UART1_CTS：UART1 CTS 16h = UART1_RTS：UART1 RTS 17h = PORT_EVENT0：端口事件 0 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 18h = PORT_EVENT1： 端口事件 1 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 19h = PORT_EVENT2： 端口事件 2 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ah = PORT_EVENT3： 端口事件 3 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Bh = PORT_EVENT4： 端口事件 4 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ch = PORT_EVENT5： 端口事件 5 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Dh = PORT_EVENT6： 端口事件 6 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Eh = PORT_EVENT7： 端口事件 7 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 20h = CPU_SWV： CPU SWV 21h = SSI1_RX：SSI1 RX 22h = SSI1_TX：SSI1 TX 23h = SSI1_FSS：SSI1 FSS 24h = SSI1_CLK：SSI1 CLK 25h = I2S_AD0：I2S 数据 0 26h = I2S_AD1： I2S 数据 1 27h = I2S_WCLK： I2S WCLK 28h = I2S_BCLK：I2S BCLK 29h = I2S_MCLK：I2S MCLK 2Eh = 射频内核跟踪

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-31. IOCFG1 寄存器字段说明（续）

位	字段	类型	复位	说明
				2Fh = 射频内核数据输出 0 30h = 射频内核数据输出 1 31h = 射频内核数据输出 2 32h = 射频内核数据输出 3 33h = 射频内核数据输入 0 34h = 射频内核数据输入 1 35h = 射频内核 SMI 数据链路输出 36h = 射频内核 SMI 数据链路输入 37h = 射频内核 SMI 命令链路输出 38h = 射频内核 SMI 命令链路输入

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.3.3 IOCFG2 寄存器（偏移 = 8h）[复位 = 00006000h]

图 13-25 展示了 IOCFG2，表 13-32 对其进行了介绍。

返回到汇总表。

DIO2 的配置

图 13-25. IOCFG2 寄存器

31	30	29	28	27	26	25	24
RESERVED	HYST_EN	IE	WU_CFG		IOMODE		
R-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h		
23	22	21	20	19	18	17	16
IOEV_AON_PROG2_EN	IOEV_AON_PROG1_EN	IOEV_AON_PROG0_EN	RESERVED		EDGE_IRQ_EN	EDGE_DET	
R/W-0h	R/W-0h	R/W-0h	R-0h		R/W-0h	R/W-0h	
15	14	13	12	11	10	9	8
RESERVED	PULL_CTL		SLEW_RED	IOCURR		IOSTR	
R-0h	R/W-3h		R/W-0h	R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
IOEV_RTC_EN	IOEV_MCU_WU_EN	PORT_ID					
R/W-0h	R/W-0h	R/W-0h					

表 13-32. IOCFG2 寄存器字段说明

位	字段	类型	复位	说明
31	RESERVED	R	0h	保留
30	HYST_EN	R/W	0h	0：输入迟滞禁用 1：输入迟滞使能
29	IE	R/W	0h	0：输入禁用 1：输入使能 注意：如果 IO 配置为 AUX PORT_ID = 0x08，则使能将被忽略。
28-27	WU_CFG	R/W	0h	如果 DIO 配置为 GPIO 或非 AON 外设信号（PORT_ID 0x00 或 >0x08）： 00：无唤醒 01：无唤醒 10：如果该焊盘变为低电平，则从关断状态唤醒。 11：如果该焊盘变为高电平，则从关断状态唤醒。 如果 IO 配置为 AON 外设信号或 AUX PORT_ID 0x01-0x08，则该寄存器仅设置是否使能唤醒。 00、01：唤醒禁用 10、11：唤醒使能 极性通过 AON 寄存器进行控制。 注意：当 MSB 置位时，IOC 将停用 DIO 的输出使能。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-32. IOCFG2 寄存器字段说明（续）

位	字段	类型	复位	说明
26-24	IOMODE	R/W	0h	IO 模式 不适用于配置为 AON 外设信号和 AUX PORT_ID 0x01-0x08 的 IO AUX 有自己的 open_source/drain 配置。 0x2: 保留。未定义的行为。 0x3：保留。未定义的行为。 0h = NORMAL：正常输入/输出 1h = INV：反相输入/输出 4h = OPENDR： 开漏， 正常输入/输出 5h = OPENDR_INV：开漏 反相输入/输出 6h = OPENSRC：开源 正常输入/输出 7h = OPENSRC_INV：开源 反相输入/输出
23	IOEV_AON_PROG2_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG2 事件置为有效 1：输入边沿检测将 AON_PROG2 事件置为有效
22	IOEV_AON_PROG1_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG1 事件置为有效 1：输入边沿检测将 AON_PROG1 事件置为有效
21	IOEV_AON_PROG0_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG0 事件置为有效 1：输入边沿检测将 AON_PROG0 事件置为有效
20-19	RESERVED	R	0h	保留
18	EDGE_IRQ_EN	R/W	0h	0：无中断生成 1：使能该 IO 的中断生成（仅在 EDGE_DET 已使能时有效）
17-16	EDGE_DET	R/W	0h	使能该 IO 上的边沿检测事件生成 0h = NONE：无边沿检测 1h = 负边沿检测 2h = 正边沿检测 3h = 正负边沿检测
15	RESERVED	R	0h	保留
14-13	PULL_CTL	R/W	3h	上拉/下拉控制 1h = DWN：下拉 2h = UP：上拉 3h = DIS：无上拉/下拉
12	SLEW_RED	R/W	0h	0：标称压摆率 1：在输出驱动器中降低压摆率。
11-10	IOCURR	R/W	0h	选择该 IO 的 IO 电流模式。 0h = 2MA：低电流 (LC) 模式：IOSTR 设置为 AUTO 时最小值为 2mA 1h = 4MA：高电流 (HC) 模式：IOSTR 设置为 AUTO 时最小值为 4mA 2h = 4_8MA：扩展电流 (EC) 模式：当 IOSTR 设为 AUTO 时，双驱动强度 IO 的最小值为 8mA（标准 IO 的最小值为 4mA）

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-32. IOCFG2 寄存器字段说明（续）

位	字段	类型	复位	说明
9-8	IOSTR	R/W	0h	选择该 IO 的驱动强度控制源。 此设置控制低电流 (LC) 模式的驱动强度。可在 IOCURR 中选择更高的驱动强度 0h = 自动驱动强度，由 AON BATMON 根据电池电压控制。（VDDS 时最小 2mA） 1h = 最小驱动强度，由 AON_IOC:IOSTRMIN 控制（默认值为 3.3V 时最小 2mA） 2h = MED：中等驱动强度，由 AON_IOC:IOSTRMED 控制（默认值为 2.5V 时最小 2mA） 3h = 最大驱动强度，由 AON_IOC:IOSTRMAX 控制（默认值为 1.8V 时最小 2mA）
7	IOEV_RTC_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 RTC 事件置为有效 1：输入边沿检测将 RTC 事件置为有效
6	IOEV_MCU_WU_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 MCU_WU 事件置为有效 1：输入边沿检测将 MCU_WU 事件置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-32. IOCFG2 寄存器字段说明（续）

位	字段	类型	复位	说明
5-0	PORT_ID	R/W	0h	选择 DIO2 的用途 注意：除明确要求更改 PORT_ID 值的情况外，不应写入此字段。 0h =通用 IO 7h = AON 32 KHz 时钟 (SCLK_LF) 8h = AUX IO 9h = SSI0_RX：SSI0 RX Ah = SSI0_TX：SSI0 TX Bh = SSI0_FSS：SSI0 FSS Ch = SSI0_CLK：SSI0 CLK Dh = I2C_MSSDA：I2C 数据 Eh = I2C_MSSCL：I2C 时钟 Fh = UART0_RX：UART0 RX 10h = UART0_TX：UART0 TX 11h = UART0_CTS：UART0 CTS 12h = UART0_RTS：UART0 RTS 13h = UART1_RX：UART1 RX 14h = UART1_TX：UART1 TX 15h = UART1_CTS：UART1 CTS 16h = UART1_RTS：UART1 RTS 17h = PORT_EVENT0：端口事件 0 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 18h = PORT_EVENT1： 端口事件 1 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 19h = PORT_EVENT2： 端口事件 2 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ah = PORT_EVENT3： 端口事件 3 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Bh = PORT_EVENT4： 端口事件 4 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ch = PORT_EVENT5： 端口事件 5 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Dh = PORT_EVENT6： 端口事件 6 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Eh = PORT_EVENT7： 端口事件 7 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 20h = CPU_SWV： CPU SWV 21h = SSI1_RX：SSI1 RX 22h = SSI1_TX：SSI1 TX 23h = SSI1_FSS：SSI1 FSS 24h = SSI1_CLK：SSI1 CLK 25h = I2S_AD0：I2S 数据 0 26h = I2S_AD1： I2S 数据 1 27h = I2S_WCLK： I2S WCLK 28h = I2S_BCLK：I2S BCLK 29h = I2S_MCLK：I2S MCLK 2Eh = 射频内核跟踪

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-32. IOCFG2 寄存器字段说明（续）

位	字段	类型	复位	说明
				2Fh = 射频内核数据输出 0 30h = 射频内核数据输出 1 31h = 射频内核数据输出 2 32h = 射频内核数据输出 3 33h = 射频内核数据输入 0 34h = 射频内核数据输入 1 35h = 射频内核 SMI 数据链路输出 36h = 射频内核 SMI 数据链路输入 37h = 射频内核 SMI 命令链路输出 38h = 射频内核 SMI 命令链路输入

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.3.4 IOCFG3 寄存器（偏移 = Ch）[复位 = 00006000h]

图 13-26 展示了 IOCFG3，表 13-33 对其进行了介绍。

返回到汇总表。

DIO3 的配置

图 13-26. IOCFG3 寄存器

31	30	29	28	27	26	25	24
RESERVED	HYST_EN	IE	WU_CFG		IOMODE		
R-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h		
23	22	21	20	19	18	17	16
IOEV_AON_PROG2_EN	IOEV_AON_PROG1_EN	IOEV_AON_PROG0_EN	RESERVED		EDGE_IRQ_EN	EDGE_DET	
R/W-0h	R/W-0h	R/W-0h	R-0h		R/W-0h	R/W-0h	
15	14	13	12	11	10	9	8
RESERVED	PULL_CTL		SLEW_RED	IOCURR		IOSTR	
R-0h	R/W-3h		R/W-0h	R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
IOEV_RTC_EN	IOEV_MCU_WU_EN	PORT_ID					
R/W-0h	R/W-0h	R/W-0h					

表 13-33. IOCFG3 寄存器字段说明

位	字段	类型	复位	说明
31	RESERVED	R	0h	保留
30	HYST_EN	R/W	0h	0：输入迟滞禁用 1：输入迟滞使能
29	IE	R/W	0h	0：输入禁用 1：输入使能 注意：如果 IO 配置为 AUX PORT_ID = 0x08，则使能将被忽略。
28-27	WU_CFG	R/W	0h	如果 DIO 配置为 GPIO 或非 AON 外设信号（PORT_ID 0x00 或 >0x08）： 00：无唤醒 01：无唤醒 10：如果该焊盘变为低电平，则从关断状态唤醒。 11：如果该焊盘变为高电平，则从关断状态唤醒。 如果 IO 配置为 AON 外设信号或 AUX PORT_ID 0x01-0x08，则该寄存器仅设置是否使能唤醒。 00、01：唤醒禁用 10、11：唤醒使能 极性通过 AON 寄存器进行控制。 注意：当 MSB 置位时，IOC 将停用 DIO 的输出使能。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-33. IOCFG3 寄存器字段说明（续）

位	字段	类型	复位	说明
26-24	IOMODE	R/W	0h	IO 模式 不适用于配置为 AON 外设信号和 AUX PORT_ID 0x01-0x08 的 IO AUX 有自己的 open_source/drain 配置。 0x2: 保留。未定义的行为。 0x3：保留。未定义的行为。 0h = NORMAL：正常输入/输出 1h = INV：反相输入/输出 4h = OPENDR： 开漏， 正常输入/输出 5h = OPENDR_INV：开漏 反相输入/输出 6h = OPENSRC：开源 正常输入/输出 7h = OPENSRC_INV：开源 反相输入/输出
23	IOEV_AON_PROG2_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG2 事件置为有效 1：输入边沿检测将 AON_PROG2 事件置为有效
22	IOEV_AON_PROG1_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG1 事件置为有效 1：输入边沿检测将 AON_PROG1 事件置为有效
21	IOEV_AON_PROG0_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG0 事件置为有效 1：输入边沿检测将 AON_PROG0 事件置为有效
20-19	RESERVED	R	0h	保留
18	EDGE_IRQ_EN	R/W	0h	0：无中断生成 1：使能该 IO 的中断生成（仅在 EDGE_DET 已使能时有效）
17-16	EDGE_DET	R/W	0h	使能该 IO 上的边沿检测事件生成 0h = NONE：无边沿检测 1h = 负边沿检测 2h = 正边沿检测 3h = 正负边沿检测
15	RESERVED	R	0h	保留
14-13	PULL_CTL	R/W	3h	上拉/下拉控制 1h = DWN：下拉 2h = UP：上拉 3h = DIS：无上拉/下拉
12	SLEW_RED	R/W	0h	0：标称压摆率 1：在输出驱动器中降低压摆率。
11-10	IOCURR	R/W	0h	选择该 IO 的 IO 电流模式。 0h = 2MA：低电流 (LC) 模式：IOSTR 设置为 AUTO 时最小值为 2mA 1h = 4MA：高电流 (HC) 模式：IOSTR 设置为 AUTO 时最小值为 4mA 2h = 4_8MA：扩展电流 (EC) 模式：当 IOSTR 设为 AUTO 时，双驱动强度 IO 的最小值为 8mA（标准 IO 的最小值为 4mA）

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-33. IOCFG3 寄存器字段说明（续）

位	字段	类型	复位	说明
9-8	IOSTR	R/W	0h	选择该 IO 的驱动强度控制源。 此设置控制低电流 (LC) 模式的驱动强度。可在 IOCURR 中选择更高的驱动强度 0h = 自动驱动强度，由 AON BATMON 根据电池电压控制。（VDDS 时最小 2mA） 1h = 最小驱动强度，由 AON_IOC:IOSTRMIN 控制（默认值为 3.3V 时最小 2mA） 2h = MED：中等驱动强度，由 AON_IOC:IOSTRMED 控制（默认值为 2.5V 时最小 2mA） 3h = 最大驱动强度，由 AON_IOC:IOSTRMAX 控制（默认值为 1.8V 时最小 2mA）
7	IOEV_RTC_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 RTC 事件置为有效 1：输入边沿检测将 RTC 事件置为有效
6	IOEV_MCU_WU_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 MCU_WU 事件置为有效 1：输入边沿检测将 MCU_WU 事件置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-33. IOCFG3 寄存器字段说明（续）

位	字段	类型	复位	说明
5-0	PORT_ID	R/W	0h	选择 DIO3 的用途 注意：除明确要求更改 PORT_ID 值的情况外，不应写入此字段。 0h =通用 IO 7h = AON 32 KHz 时钟 (SCLK_LF) 8h = AUX IO 9h = SSI0_RX：SSI0 RX Ah = SSI0_TX：SSI0 TX Bh = SSI0_FSS：SSI0 FSS Ch = SSI0_CLK：SSI0 CLK Dh = I2C_MSSDA：I2C 数据 Eh = I2C_MSSCL：I2C 时钟 Fh = UART0_RX：UART0 RX 10h = UART0_TX：UART0 TX 11h = UART0_CTS：UART0 CTS 12h = UART0_RTS：UART0 RTS 13h = UART1_RX：UART1 RX 14h = UART1_TX：UART1 TX 15h = UART1_CTS：UART1 CTS 16h = UART1_RTS：UART1 RTS 17h = PORT_EVENT0：端口事件 0 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等）， 可用作通用 IO 事件 18h = PORT_EVENT1： 端口事件 1 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等）， 可用作通用 IO 事件 19h = PORT_EVENT2： 端口事件 2 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等）， 可用作通用 IO 事件 1Ah = PORT_EVENT3： 端口事件 3 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等）， 可用作通用 IO 事件 1Bh = PORT_EVENT4： 端口事件 4 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等）， 可用作通用 IO 事件 1Ch = PORT_EVENT5： 端口事件 5 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等）， 可用作通用 IO 事件 1Dh = PORT_EVENT6： 端口事件 6 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等）， 可用作通用 IO 事件 1Eh = PORT_EVENT7： 端口事件 7 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等）， 可用作通用 IO 事件 20h = CPU_SWV： CPU SWV 21h = SSI1_RX：SSI1 RX 22h = SSI1_TX：SSI1 TX 23h = SSI1_FSS：SSI1 FSS 24h = SSI1_CLK：SSI1 CLK 25h = I2S_AD0：I2S 数据 0 26h = I2S_AD1： I2S 数据 1 27h = I2S_WCLK： I2S WCLK 28h = I2S_BCLK：I2S BCLK 29h = I2S_MCLK：I2S MCLK 2Eh = 射频内核跟踪

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-33. IOCFG3 寄存器字段说明（续）

位	字段	类型	复位	说明
				2Fh = 射频内核数据输出 0 30h = 射频内核数据输出 1 31h = 射频内核数据输出 2 32h = 射频内核数据输出 3 33h = 射频内核数据输入 0 34h = 射频内核数据输入 1 35h = 射频内核 SMI 数据链路输出 36h = 射频内核 SMI 数据链路输入 37h = 射频内核 SMI 命令链路输出 38h = 射频内核 SMI 命令链路输入

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.3.5 IOCFG4 寄存器（偏移 = 10h）[复位 = 00006000h]

图 13-27 展示了 IOCFG4，表 13-34 对其进行了介绍。

返回到汇总表。

DIO4 的配置

图 13-27. IOCFG4 寄存器

31	30	29	28	27	26	25	24
RESERVED	HYST_EN	IE	WU_CFG		IOMODE		
R-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h		
23	22	21	20	19	18	17	16
IOEV_AON_PROG2_EN	IOEV_AON_PROG1_EN	IOEV_AON_PROG0_EN	RESERVED		EDGE_IRQ_EN	EDGE_DET	
R/W-0h	R/W-0h	R/W-0h	R-0h		R/W-0h	R/W-0h	
15	14	13	12	11	10	9	8
RESERVED	PULL_CTL		SLEW_RED	IOCURR		IOSTR	
R-0h	R/W-3h		R/W-0h	R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
IOEV_RTC_EN	IOEV_MCU_WU_EN	PORT_ID					
R/W-0h	R/W-0h	R/W-0h					

表 13-34. IOCFG4 寄存器字段说明

位	字段	类型	复位	说明
31	RESERVED	R	0h	保留
30	HYST_EN	R/W	0h	0：输入迟滞禁用 1：输入迟滞使能
29	IE	R/W	0h	0：输入禁用 1：输入使能 注意：如果 IO 配置为 AUX PORT_ID = 0x08，则使能将被忽略。
28-27	WU_CFG	R/W	0h	如果 DIO 配置为 GPIO 或非 AON 外设信号（PORT_ID 0x00 或 >0x08）： 00：无唤醒 01：无唤醒 10：如果该焊盘变为低电平，则从关断状态唤醒。 11：如果该焊盘变为高电平，则从关断状态唤醒。 如果 IO 配置为 AON 外设信号或 AUX PORT_ID 0x01-0x08，则该寄存器仅设置是否使能唤醒。 00、01：唤醒禁用 10、11：唤醒使能 极性通过 AON 寄存器进行控制。 注意：当 MSB 置位时，IOC 将停用 DIO 的输出使能。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-34. IOCFG4 寄存器字段说明（续）

位	字段	类型	复位	说明
26-24	IOMODE	R/W	0h	IO 模式 不适用于配置为 AON 外设信号和 AUX PORT_ID 0x01-0x08 的 IO AUX 有自己的 open_source/drain 配置。 0x2: 保留。未定义的行为。 0x3：保留。未定义的行为。 0h = NORMAL：正常输入/输出 1h = INV：反相输入/输出 4h = OPENDR： 开漏， 正常输入/输出 5h = OPENDR_INV：开漏 反相输入/输出 6h = OPENSRC：开源 正常输入/输出 7h = OPENSRC_INV：开源 反相输入/输出
23	IOEV_AON_PROG2_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG2 事件置为有效 1：输入边沿检测将 AON_PROG2 事件置为有效
22	IOEV_AON_PROG1_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG1 事件置为有效 1：输入边沿检测将 AON_PROG1 事件置为有效
21	IOEV_AON_PROG0_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG0 事件置为有效 1：输入边沿检测将 AON_PROG0 事件置为有效
20-19	RESERVED	R	0h	保留
18	EDGE_IRQ_EN	R/W	0h	0：无中断生成 1：使能该 IO 的中断生成（仅在 EDGE_DET 已使能时有效）
17-16	EDGE_DET	R/W	0h	使能该 IO 上的边沿检测事件生成 0h = NONE：无边沿检测 1h = 负边沿检测 2h = 正边沿检测 3h = 正负边沿检测
15	RESERVED	R	0h	保留
14-13	PULL_CTL	R/W	3h	上拉/下拉控制 1h = DWN：下拉 2h = UP：上拉 3h = DIS：无上拉/下拉
12	SLEW_RED	R/W	0h	0：标称压摆率 1：在输出驱动器中降低压摆率。
11-10	IOCURR	R/W	0h	选择该 IO 的 IO 电流模式。 0h = 2MA：低电流 (LC) 模式：IOSTR 设置为 AUTO 时最小值为 2mA 1h = 4MA：高电流 (HC) 模式：IOSTR 设置为 AUTO 时最小值为 4mA 2h = 4_8MA：扩展电流 (EC) 模式：当 IOSTR 设为 AUTO 时，双驱动强度 IO 的最小值为 8mA（标准 IO 的最小值为 4mA）

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-34. IOCFG4 寄存器字段说明（续）

位	字段	类型	复位	说明
9-8	IOSTR	R/W	0h	选择该 IO 的驱动强度控制源。 此设置控制低电流 (LC) 模式的驱动强度。可在 IOCURR 中选择更高的驱动强度 0h = 自动驱动强度，由 AON BATMON 根据电池电压控制。（VDDS 时最小 2mA） 1h = 最小驱动强度，由 AON_IOC:IOSTRMIN 控制（默认值为 3.3V 时最小 2mA） 2h = MED：中等驱动强度，由 AON_IOC:IOSTRMED 控制（默认值为 2.5V 时最小 2mA） 3h = 最大驱动强度，由 AON_IOC:IOSTRMAX 控制（默认值为 1.8V 时最小 2mA）
7	IOEV_RTC_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 RTC 事件置为有效 1：输入边沿检测将 RTC 事件置为有效
6	IOEV_MCU_WU_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 MCU_WU 事件置为有效 1：输入边沿检测将 MCU_WU 事件置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-34. IOCFG4 寄存器字段说明（续）

位	字段	类型	复位	说明
5-0	PORT_ID	R/W	0h	选择 DIO4 的用途 注意：除明确要求更改 PORT_ID 值的情况外，不应写入此字段。 0h =通用 IO 7h = AON 32 KHz 时钟 (SCLK_LF) 8h = AUX IO 9h = SSI0_RX：SSI0 RX Ah = SSI0_TX：SSI0 TX Bh = SSI0_FSS：SSI0 FSS Ch = SSI0_CLK：SSI0 CLK Dh = I2C_MSSDA：I2C 数据 Eh = I2C_MSSCL：I2C 时钟 Fh = UART0_RX：UART0 RX 10h = UART0_TX：UART0 TX 11h = UART0_CTS：UART0 CTS 12h = UART0_RTS：UART0 RTS 13h = UART1_RX：UART1 RX 14h = UART1_TX：UART1 TX 15h = UART1_CTS：UART1 CTS 16h = UART1_RTS：UART1 RTS 17h = PORT_EVENT0：端口事件 0 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 18h = PORT_EVENT1： 端口事件 1 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 19h = PORT_EVENT2： 端口事件 2 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ah = PORT_EVENT3： 端口事件 3 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Bh = PORT_EVENT4： 端口事件 4 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ch = PORT_EVENT5： 端口事件 5 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Dh = PORT_EVENT6： 端口事件 6 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Eh = PORT_EVENT7： 端口事件 7 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 20h = CPU_SWV： CPU SWV 21h = SSI1_RX：SSI1 RX 22h = SSI1_TX：SSI1 TX 23h = SSI1_FSS：SSI1 FSS 24h = SSI1_CLK：SSI1 CLK 25h = I2S_AD0：I2S 数据 0 26h = I2S_AD1： I2S 数据 1 27h = I2S_WCLK： I2S WCLK 28h = I2S_BCLK：I2S BCLK 29h = I2S_MCLK：I2S MCLK 2Eh = 射频内核跟踪

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-34. IOCFG4 寄存器字段说明（续）

位	字段	类型	复位	说明
				2Fh = 射频内核数据输出 0 30h = 射频内核数据输出 1 31h = 射频内核数据输出 2 32h = 射频内核数据输出 3 33h = 射频内核数据输入 0 34h = 射频内核数据输入 1 35h = 射频内核 SMI 数据链路输出 36h = 射频内核 SMI 数据链路输入 37h = 射频内核 SMI 命令链路输出 38h = 射频内核 SMI 命令链路输入

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.3.6 IOCFG5 寄存器（偏移 = 14h）[复位 = 00006000h]

图 13-28 展示了 IOCFG5，表 13-35 对其进行了介绍。

返回到汇总表。

DIO5 的配置

图 13-28. IOCFG5 寄存器

31	30	29	28	27	26	25	24
RESERVED	HYST_EN	IE	WU_CFG		IOMODE		
R-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h		
23	22	21	20	19	18	17	16
IOEV_AON_PROG2_EN	IOEV_AON_PROG1_EN	IOEV_AON_PROG0_EN	RESERVED		EDGE_IRQ_EN	EDGE_DET	
R/W-0h	R/W-0h	R/W-0h	R-0h		R/W-0h	R/W-0h	
15	14	13	12	11	10	9	8
RESERVED	PULL_CTL		SLEW_RED	IOCURR		IOSTR	
R-0h	R/W-3h		R/W-0h	R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
IOEV_RTC_EN	IOEV_MCU_WU_EN	PORT_ID					
R/W-0h	R/W-0h	R/W-0h					

表 13-35. IOCFG5 寄存器字段说明

位	字段	类型	复位	说明
31	RESERVED	R	0h	保留
30	HYST_EN	R/W	0h	0：输入迟滞禁用 1：输入迟滞使能
29	IE	R/W	0h	0：输入禁用 1：输入使能 注意：如果 IO 配置为 AUX PORT_ID = 0x08，则使能将被忽略。
28-27	WU_CFG	R/W	0h	如果 DIO 配置为 GPIO 或非 AON 外设信号（PORT_ID 0x00 或 >0x08）： 00：无唤醒 01：无唤醒 10：如果该焊盘变为低电平，则从关断状态唤醒。 11：如果该焊盘变为高电平，则从关断状态唤醒。 如果 IO 配置为 AON 外设信号或 AUX PORT_ID 0x01-0x08，则该寄存器仅设置是否使能唤醒。 00、01：唤醒禁用 10、11：唤醒使能 极性通过 AON 寄存器进行控制。 注意：当 MSB 置位时，IOC 将停用 DIO 的输出使能。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-35. IOCFG5 寄存器字段说明（续）

位	字段	类型	复位	说明
26-24	IOMODE	R/W	0h	IO 模式 不适用于配置为 AON 外设信号和 AUX PORT_ID 0x01-0x08 的 IO AUX 有自己的 open_source/drain 配置。 0x2: 保留。未定义的行为。 0x3：保留。未定义的行为。 0h = NORMAL：正常输入/输出 1h = INV：反相输入/输出 4h = OPENDR： 开漏， 正常输入/输出 5h = OPENDR_INV：开漏 反相输入/输出 6h = OPENSRC：开源 正常输入/输出 7h = OPENSRC_INV：开源 反相输入/输出
23	IOEV_AON_PROG2_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG2 事件置为有效 1：输入边沿检测将 AON_PROG2 事件置为有效
22	IOEV_AON_PROG1_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG1 事件置为有效 1：输入边沿检测将 AON_PROG1 事件置为有效
21	IOEV_AON_PROG0_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG0 事件置为有效 1：输入边沿检测将 AON_PROG0 事件置为有效
20-19	RESERVED	R	0h	保留
18	EDGE_IRQ_EN	R/W	0h	0：无中断生成 1：使能该 IO 的中断生成（仅在 EDGE_DET 已使能时有效）
17-16	EDGE_DET	R/W	0h	使能该 IO 上的边沿检测事件生成 0h = NONE：无边沿检测 1h = 负边沿检测 2h = 正边沿检测 3h = 正负边沿检测
15	RESERVED	R	0h	保留
14-13	PULL_CTL	R/W	3h	上拉/下拉控制 1h = DWN：下拉 2h = UP：上拉 3h = DIS：无上拉/下拉
12	SLEW_RED	R/W	0h	0：标称压摆率 1：在输出驱动器中降低压摆率。
11-10	IOCURR	R/W	0h	选择该 IO 的 IO 电流模式。 0h = 2MA：低电流 (LC) 模式：IOSTR 设置为 AUTO 时最小值为 2mA 1h = 4MA：高电流 (HC) 模式：IOSTR 设置为 AUTO 时最小值为 4mA 2h = 4_8MA：扩展电流 (EC) 模式：当 IOSTR 设为 AUTO 时，双驱动强度 IO 的最小值为 8mA（标准 IO 的最小值为 4mA）

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-35. IOCFG5 寄存器字段说明（续）

位	字段	类型	复位	说明
9-8	IOSTR	R/W	0h	选择该 IO 的驱动强度控制源。 此设置控制低电流 (LC) 模式的驱动强度。可在 IOCURR 中选择更高的驱动强度 0h = 自动驱动强度，由 AON BATMON 根据电池电压控制。（VDDS 时最小 2mA） 1h = 最小驱动强度，由 AON_IOC:IOSTRMIN 控制（默认值为 3.3V 时最小 2mA） 2h = MED：中等驱动强度，由 AON_IOC:IOSTRMED 控制（默认值为 2.5V 时最小 2mA） 3h = 最大驱动强度，由 AON_IOC:IOSTRMAX 控制（默认值为 1.8V 时最小 2mA）
7	IOEV_RTC_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 RTC 事件置为有效 1：输入边沿检测将 RTC 事件置为有效
6	IOEV_MCU_WU_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 MCU_WU 事件置为有效 1：输入边沿检测将 MCU_WU 事件置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-35. IOCFG5 寄存器字段说明（续）

位	字段	类型	复位	说明
5-0	PORT_ID	R/W	0h	选择 DIO5 的用途 注意：除明确要求更改 PORT_ID 值的情况外，不应写入此字段。 0h =通用 IO 7h = AON 32 KHz 时钟 (SCLK_LF) 8h = AUX IO 9h = SSI0_RX：SSI0 RX Ah = SSI0_TX：SSI0 TX Bh = SSI0_FSS：SSI0 FSS Ch = SSI0_CLK：SSI0 CLK Dh = I2C_MSSDA：I2C 数据 Eh = I2C_MSSCL：I2C 时钟 Fh = UART0_RX：UART0 RX 10h = UART0_TX：UART0 TX 11h = UART0_CTS：UART0 CTS 12h = UART0_RTS：UART0 RTS 13h = UART1_RX：UART1 RX 14h = UART1_TX：UART1 TX 15h = UART1_CTS：UART1 CTS 16h = UART1_RTS：UART1 RTS 17h = PORT_EVENT0：端口事件 0 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 18h = PORT_EVENT1： 端口事件 1 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 19h = PORT_EVENT2： 端口事件 2 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ah = PORT_EVENT3： 端口事件 3 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Bh = PORT_EVENT4： 端口事件 4 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ch = PORT_EVENT5： 端口事件 5 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Dh = PORT_EVENT6： 端口事件 6 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Eh = PORT_EVENT7： 端口事件 7 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 20h = CPU_SWV： CPU SWV 21h = SSI1_RX：SSI1 RX 22h = SSI1_TX：SSI1 TX 23h = SSI1_FSS：SSI1 FSS 24h = SSI1_CLK：SSI1 CLK 25h = I2S_AD0：I2S 数据 0 26h = I2S_AD1： I2S 数据 1 27h = I2S_WCLK： I2S WCLK 28h = I2S_BCLK：I2S BCLK 29h = I2S_MCLK：I2S MCLK 2Eh = 射频内核跟踪

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-35. IOCFG5 寄存器字段说明（续）

位	字段	类型	复位	说明
				2Fh = 射频内核数据输出 0 30h = 射频内核数据输出 1 31h = 射频内核数据输出 2 32h = 射频内核数据输出 3 33h = 射频内核数据输入 0 34h = 射频内核数据输入 1 35h = 射频内核 SMI 数据链路输出 36h = 射频内核 SMI 数据链路输入 37h = 射频内核 SMI 命令链路输出 38h = 射频内核 SMI 命令链路输入

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.3.7 IOCFG6 寄存器（偏移 = 18h）[复位 = 00006000h]

图 13-29 展示了 IOCFG6，表 13-36 对其进行了介绍。

返回到汇总表。

DIO6 的配置

图 13-29. IOCFG6 寄存器

31	30	29	28	27	26	25	24
RESERVED	HYST_EN	IE	WU_CFG		IOMODE		
R-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h		
23	22	21	20	19	18	17	16
IOEV_AON_PROG2_EN	IOEV_AON_PROG1_EN	IOEV_AON_PROG0_EN	RESERVED		EDGE_IRQ_EN	EDGE_DET	
R/W-0h	R/W-0h	R/W-0h	R-0h		R/W-0h	R/W-0h	
15	14	13	12	11	10	9	8
RESERVED	PULL_CTL		SLEW_RED	IOCURR		IOSTR	
R-0h	R/W-3h		R/W-0h	R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
IOEV_RTC_EN	IOEV_MCU_WU_EN	PORT_ID					
R/W-0h	R/W-0h	R/W-0h					

表 13-36. IOCFG6 寄存器字段说明

位	字段	类型	复位	说明
31	RESERVED	R	0h	保留
30	HYST_EN	R/W	0h	0：输入迟滞禁用 1：输入迟滞使能
29	IE	R/W	0h	0：输入禁用 1：输入使能 注意：如果 IO 配置为 AUX PORT_ID = 0x08，则使能将被忽略。
28-27	WU_CFG	R/W	0h	如果 DIO 配置为 GPIO 或非 AON 外设信号（PORT_ID 0x00 或 >0x08）： 00：无唤醒 01：无唤醒 10：如果该焊盘变为低电平，则从关断状态唤醒。 11：如果该焊盘变为高电平，则从关断状态唤醒。 如果 IO 配置为 AON 外设信号或 AUX PORT_ID 0x01-0x08，则该寄存器仅设置是否使能唤醒。 00、01：唤醒禁用 10、11：唤醒使能 极性通过 AON 寄存器进行控制。 注意：当 MSB 置位时，IOC 将停用 DIO 的输出使能。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-36. IOCFG6 寄存器字段说明（续）

位	字段	类型	复位	说明
26-24	IOMODE	R/W	0h	IO 模式 不适用于配置为 AON 外设信号和 AUX PORT_ID 0x01-0x08 的 IO AUX 有自己的 open_source/drain 配置。 0x2: 保留。未定义的行为。 0x3：保留。未定义的行为。 0h = NORMAL：正常输入/输出 1h = INV：反相输入/输出 4h = OPENDR： 开漏， 正常输入/输出 5h = OPENDR_INV：开漏 反相输入/输出 6h = OPENSRC：开源 正常输入/输出 7h = OPENSRC_INV：开源 反相输入/输出
23	IOEV_AON_PROG2_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG2 事件置为有效 1：输入边沿检测将 AON_PROG2 事件置为有效
22	IOEV_AON_PROG1_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG1 事件置为有效 1：输入边沿检测将 AON_PROG1 事件置为有效
21	IOEV_AON_PROG0_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG0 事件置为有效 1：输入边沿检测将 AON_PROG0 事件置为有效
20-19	RESERVED	R	0h	保留
18	EDGE_IRQ_EN	R/W	0h	0：无中断生成 1：使能该 IO 的中断生成（仅在 EDGE_DET 已使能时有效）
17-16	EDGE_DET	R/W	0h	使能该 IO 上的边沿检测事件生成 0h = NONE：无边沿检测 1h = 负边沿检测 2h = 正边沿检测 3h = 正负边沿检测
15	RESERVED	R	0h	保留
14-13	PULL_CTL	R/W	3h	上拉/下拉控制 1h = DWN：下拉 2h = UP：上拉 3h = DIS：无上拉/下拉
12	SLEW_RED	R/W	0h	0：标称压摆率 1：在输出驱动器中降低压摆率。
11-10	IOCURR	R/W	0h	选择该 IO 的 IO 电流模式。 0h = 2MA：低电流 (LC) 模式：IOSTR 设置为 AUTO 时最小值为 2mA 1h = 4MA：高电流 (HC) 模式：IOSTR 设置为 AUTO 时最小值为 4mA 2h = 4_8MA：扩展电流 (EC) 模式：当 IOSTR 设为 AUTO 时，双驱动强度 IO 的最小值为 8mA（标准 IO 的最小值为 4mA）

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-36. IOCFG6 寄存器字段说明（续）

位	字段	类型	复位	说明
9-8	IOSTR	R/W	0h	选择该 IO 的驱动强度控制源。 此设置控制低电流 (LC) 模式的驱动强度。可在 IOCURR 中选择更高的驱动强度 0h = 自动驱动强度，由 AON BATMON 根据电池电压控制。（VDDS 时最小 2mA） 1h = 最小驱动强度，由 AON_IOC:IOSTRMIN 控制（默认值为 3.3V 时最小 2mA） 2h = MED：中等驱动强度，由 AON_IOC:IOSTRMED 控制（默认值为 2.5V 时最小 2mA） 3h = 最大驱动强度，由 AON_IOC:IOSTRMAX 控制（默认值为 1.8V 时最小 2mA）
7	IOEV_RTC_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 RTC 事件置为有效 1：输入边沿检测将 RTC 事件置为有效
6	IOEV_MCU_WU_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 MCU_WU 事件置为有效 1：输入边沿检测将 MCU_WU 事件置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-36. IOCFG6 寄存器字段说明（续）

位	字段	类型	复位	说明
5-0	PORT_ID	R/W	0h	选择 DIO6 的用途 注意：除明确要求更改 PORT_ID 值的情况外，不应写入此字段。 0h =通用 IO 7h = AON 32 KHz 时钟 (SCLK_LF) 8h = AUX IO 9h = SSI0_RX：SSI0 RX Ah = SSI0_TX：SSI0 TX Bh = SSI0_FSS：SSI0 FSS Ch = SSI0_CLK：SSI0 CLK Dh = I2C_MSSDA：I2C 数据 Eh = I2C_MSSCL：I2C 时钟 Fh = UART0_RX：UART0 RX 10h = UART0_TX：UART0 TX 11h = UART0_CTS：UART0 CTS 12h = UART0_RTS：UART0 RTS 13h = UART1_RX：UART1 RX 14h = UART1_TX：UART1 TX 15h = UART1_CTS：UART1 CTS 16h = UART1_RTS：UART1 RTS 17h = PORT_EVENT0：端口事件 0 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 18h = PORT_EVENT1： 端口事件 1 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 19h = PORT_EVENT2： 端口事件 2 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ah = PORT_EVENT3： 端口事件 3 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Bh = PORT_EVENT4： 端口事件 4 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ch = PORT_EVENT5： 端口事件 5 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Dh = PORT_EVENT6： 端口事件 6 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Eh = PORT_EVENT7： 端口事件 7 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 20h = CPU_SWV： CPU SWV 21h = SSI1_RX：SSI1 RX 22h = SSI1_TX：SSI1 TX 23h = SSI1_FSS：SSI1 FSS 24h = SSI1_CLK：SSI1 CLK 25h = I2S_AD0：I2S 数据 0 26h = I2S_AD1： I2S 数据 1 27h = I2S_WCLK： I2S WCLK 28h = I2S_BCLK：I2S BCLK 29h = I2S_MCLK：I2S MCLK 2Eh = 射频内核跟踪

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-36. IOCFG6 寄存器字段说明（续）

位	字段	类型	复位	说明
				2Fh = 射频内核数据输出 0 30h = 射频内核数据输出 1 31h = 射频内核数据输出 2 32h = 射频内核数据输出 3 33h = 射频内核数据输入 0 34h = 射频内核数据输入 1 35h = 射频内核 SMI 数据链路输出 36h = 射频内核 SMI 数据链路输入 37h = 射频内核 SMI 命令链路输出 38h = 射频内核 SMI 命令链路输入

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.3.8 IOCFG7 寄存器（偏移 = 1Ch）[复位 = 00006000h]

图 13-30 展示了 IOCFG7，表 13-37 对其进行了介绍。

返回到汇总表。

DIO7 的配置

图 13-30. IOCFG7 寄存器

31	30	29	28	27	26	25	24
RESERVED	HYST_EN	IE	WU_CFG		IOMODE		
R-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h		
23	22	21	20	19	18	17	16
IOEV_AON_PROG2_EN	IOEV_AON_PROG1_EN	IOEV_AON_PROG0_EN	RESERVED		EDGE_IRQ_EN	EDGE_DET	
R/W-0h	R/W-0h	R/W-0h	R-0h		R/W-0h	R/W-0h	
15	14	13	12	11	10	9	8
RESERVED	PULL_CTL		SLEW_RED	IOCURR		IOSTR	
R-0h	R/W-3h		R/W-0h	R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
IOEV_RTC_EN	IOEV_MCU_WU_EN	PORT_ID					
R/W-0h	R/W-0h	R/W-0h					

表 13-37. IOCFG7 寄存器字段说明

位	字段	类型	复位	说明
31	RESERVED	R	0h	保留
30	HYST_EN	R/W	0h	0：输入迟滞禁用 1：输入迟滞使能
29	IE	R/W	0h	0：输入禁用 1：输入使能 注意：如果 IO 配置为 AUX PORT_ID = 0x08，则使能将被忽略。
28-27	WU_CFG	R/W	0h	如果 DIO 配置为 GPIO 或非 AON 外设信号（PORT_ID 0x00 或 >0x08）： 00：无唤醒 01：无唤醒 10：如果该焊盘变为低电平，则从关断状态唤醒。 11：如果该焊盘变为高电平，则从关断状态唤醒。 如果 IO 配置为 AON 外设信号或 AUX PORT_ID 0x01-0x08，则该寄存器仅设置是否使能唤醒。 00、01：唤醒禁用 10、11：唤醒使能 极性通过 AON 寄存器进行控制。 注意：当 MSB 置位时，IOC 将停用 DIO 的输出使能。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-37. IOCFG7 寄存器字段说明（续）

位	字段	类型	复位	说明
26-24	IOMODE	R/W	0h	IO 模式 不适用于配置为 AON 外设信号和 AUX PORT_ID 0x01-0x08 的 IO AUX 有自己的 open_source/drain 配置。 0x2: 保留。未定义的行为。 0x3：保留。未定义的行为。 0h = NORMAL：正常输入/输出 1h = INV：反相输入/输出 4h = OPENDR： 开漏， 正常输入/输出 5h = OPENDR_INV：开漏 反相输入/输出 6h = OPENSRC：开源 正常输入/输出 7h = OPENSRC_INV：开源 反相输入/输出
23	IOEV_AON_PROG2_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG2 事件置为有效 1：输入边沿检测将 AON_PROG2 事件置为有效
22	IOEV_AON_PROG1_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG1 事件置为有效 1：输入边沿检测将 AON_PROG1 事件置为有效
21	IOEV_AON_PROG0_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG0 事件置为有效 1：输入边沿检测将 AON_PROG0 事件置为有效
20-19	RESERVED	R	0h	保留
18	EDGE_IRQ_EN	R/W	0h	0：无中断生成 1：使能该 IO 的中断生成（仅在 EDGE_DET 已使能时有效）
17-16	EDGE_DET	R/W	0h	使能该 IO 上的边沿检测事件生成 0h = NONE：无边沿检测 1h = 负边沿检测 2h = 正边沿检测 3h = 正负边沿检测
15	RESERVED	R	0h	保留
14-13	PULL_CTL	R/W	3h	上拉/下拉控制 1h = DWN：下拉 2h = UP：上拉 3h = DIS：无上拉/下拉
12	SLEW_RED	R/W	0h	0：标称压摆率 1：在输出驱动器中降低压摆率。
11-10	IOCURR	R/W	0h	选择该 IO 的 IO 电流模式。 0h = 2MA：低电流 (LC) 模式：IOSTR 设置为 AUTO 时最小值为 2mA 1h = 4MA：高电流 (HC) 模式：IOSTR 设置为 AUTO 时最小值为 4mA 2h = 4_8MA：扩展电流 (EC) 模式：当 IOSTR 设为 AUTO 时，双驱动强度 IO 的最小值为 8mA（标准 IO 的最小值为 4mA）

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-37. IOCFG7 寄存器字段说明（续）

位	字段	类型	复位	说明
9-8	IOSTR	R/W	0h	选择该 IO 的驱动强度控制源。 此设置控制低电流 (LC) 模式的驱动强度。可在 IOCURR 中选择更高的驱动强度 0h = 自动驱动强度，由 AON BATMON 根据电池电压控制。（VDDS 时最小 2mA） 1h = 最小驱动强度，由 AON_IOC:IOSTRMIN 控制（默认值为 3.3V 时最小 2mA） 2h = MED：中等驱动强度，由 AON_IOC:IOSTRMED 控制（默认值为 2.5V 时最小 2mA） 3h = 最大驱动强度，由 AON_IOC:IOSTRMAX 控制（默认值为 1.8V 时最小 2mA）
7	IOEV_RTC_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 RTC 事件置为有效 1：输入边沿检测将 RTC 事件置为有效
6	IOEV_MCU_WU_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 MCU_WU 事件置为有效 1：输入边沿检测将 MCU_WU 事件置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-37. IOCFG7 寄存器字段说明（续）

位	字段	类型	复位	说明
5-0	PORT_ID	R/W	0h	选择 DIO7 的用途 注意：除明确要求更改 PORT_ID 值的情况外，不应写入此字段。 0h =通用 IO 7h = AON 32 KHz 时钟 (SCLK_LF) 8h = AUX IO 9h = SSI0_RX：SSI0 RX Ah = SSI0_TX：SSI0 TX Bh = SSI0_FSS：SSI0 FSS Ch = SSI0_CLK：SSI0 CLK Dh = I2C_MSSDA：I2C 数据 Eh = I2C_MSSCL：I2C 时钟 Fh = UART0_RX：UART0 RX 10h = UART0_TX：UART0 TX 11h = UART0_CTS：UART0 CTS 12h = UART0_RTS：UART0 RTS 13h = UART1_RX：UART1 RX 14h = UART1_TX：UART1 TX 15h = UART1_CTS：UART1 CTS 16h = UART1_RTS：UART1 RTS 17h = PORT_EVENT0：端口事件 0 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 18h = PORT_EVENT1： 端口事件 1 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 19h = PORT_EVENT2： 端口事件 2 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ah = PORT_EVENT3： 端口事件 3 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Bh = PORT_EVENT4： 端口事件 4 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ch = PORT_EVENT5： 端口事件 5 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Dh = PORT_EVENT6： 端口事件 6 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Eh = PORT_EVENT7： 端口事件 7 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 20h = CPU_SWV： CPU SWV 21h = SSI1_RX：SSI1 RX 22h = SSI1_TX：SSI1 TX 23h = SSI1_FSS：SSI1 FSS 24h = SSI1_CLK：SSI1 CLK 25h = I2S_AD0：I2S 数据 0 26h = I2S_AD1： I2S 数据 1 27h = I2S_WCLK： I2S WCLK 28h = I2S_BCLK：I2S BCLK 29h = I2S_MCLK：I2S MCLK 2Eh = 射频内核跟踪

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-37. IOCFG7 寄存器字段说明（续）

位	字段	类型	复位	说明
				2Fh = 射频内核数据输出 0 30h = 射频内核数据输出 1 31h = 射频内核数据输出 2 32h = 射频内核数据输出 3 33h = 射频内核数据输入 0 34h = 射频内核数据输入 1 35h = 射频内核 SMI 数据链路输出 36h = 射频内核 SMI 数据链路输入 37h = 射频内核 SMI 命令链路输出 38h = 射频内核 SMI 命令链路输入

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.3.9 IOCFG8 寄存器（偏移 = 20h）[复位 = 00006000h]

图 13-31 展示了 IOCFG8，表 13-38 对其进行了介绍。

返回到汇总表。

DIO8 的配置

图 13-31. IOCFG8 寄存器

31	30	29	28	27	26	25	24
RESERVED	HYST_EN	IE	WU_CFG		IOMODE		
R-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h		
23	22	21	20	19	18	17	16
IOEV_AON_PROG2_EN	IOEV_AON_PROG1_EN	IOEV_AON_PROG0_EN	RESERVED		EDGE_IRQ_EN	EDGE_DET	
R/W-0h	R/W-0h	R/W-0h	R-0h		R/W-0h	R/W-0h	
15	14	13	12	11	10	9	8
RESERVED	PULL_CTL		SLEW_RED	IOCURR		IOSTR	
R-0h	R/W-3h		R/W-0h	R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
IOEV_RTC_EN	IOEV_MCU_WU_EN	PORT_ID					
R/W-0h	R/W-0h	R/W-0h					

表 13-38. IOCFG8 寄存器字段说明

位	字段	类型	复位	说明
31	RESERVED	R	0h	保留
30	HYST_EN	R/W	0h	0：输入迟滞禁用 1：输入迟滞使能
29	IE	R/W	0h	0：输入禁用 1：输入使能 注意：如果 IO 配置为 AUX PORT_ID = 0x08，则使能将被忽略。
28-27	WU_CFG	R/W	0h	如果 DIO 配置为 GPIO 或非 AON 外设信号（PORT_ID 0x00 或 >0x08）： 00：无唤醒 01：无唤醒 10：如果该焊盘变为低电平，则从关断状态唤醒。 11：如果该焊盘变为高电平，则从关断状态唤醒。 如果 IO 配置为 AON 外设信号或 AUX PORT_ID 0x01-0x08，则该寄存器仅设置是否使能唤醒。 00、01：唤醒禁用 10、11：唤醒使能 极性通过 AON 寄存器进行控制。 注意：当 MSB 置位时，IOC 将停用 DIO 的输出使能。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-38. IOCFG8 寄存器字段说明（续）

位	字段	类型	复位	说明
26-24	IOMODE	R/W	0h	IO 模式 不适用于配置为 AON 外设信号和 AUX PORT_ID 0x01-0x08 的 IO AUX 有自己的 open_source/drain 配置。 0x2: 保留。未定义的行为。 0x3：保留。未定义的行为。 0h = NORMAL：正常输入/输出 1h = INV：反相输入/输出 4h = OPENDR： 开漏， 正常输入/输出 5h = OPENDR_INV：开漏 反相输入/输出 6h = OPENSRC：开源 正常输入/输出 7h = OPENSRC_INV：开源 反相输入/输出
23	IOEV_AON_PROG2_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG2 事件置为有效 1：输入边沿检测将 AON_PROG2 事件置为有效
22	IOEV_AON_PROG1_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG1 事件置为有效 1：输入边沿检测将 AON_PROG1 事件置为有效
21	IOEV_AON_PROG0_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG0 事件置为有效 1：输入边沿检测将 AON_PROG0 事件置为有效
20-19	RESERVED	R	0h	保留
18	EDGE_IRQ_EN	R/W	0h	0：无中断生成 1：使能该 IO 的中断生成（仅在 EDGE_DET 已使能时有效）
17-16	EDGE_DET	R/W	0h	使能该 IO 上的边沿检测事件生成 0h = NONE：无边沿检测 1h = 负边沿检测 2h = 正边沿检测 3h = 正负边沿检测
15	RESERVED	R	0h	保留
14-13	PULL_CTL	R/W	3h	上拉/下拉控制 1h = DWN：下拉 2h = UP：上拉 3h = DIS：无上拉/下拉
12	SLEW_RED	R/W	0h	0：标称压摆率 1：在输出驱动器中降低压摆率。
11-10	IOCURR	R/W	0h	选择该 IO 的 IO 电流模式。 0h = 2MA：低电流 (LC) 模式：IOSTR 设置为 AUTO 时最小值为 2mA 1h = 4MA：高电流 (HC) 模式：IOSTR 设置为 AUTO 时最小值为 4mA 2h = 4_8MA：扩展电流 (EC) 模式：当 IOSTR 设为 AUTO 时，双驱动强度 IO 的最小值为 8mA（标准 IO 的最小值为 4mA）

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-38. IOCFG8 寄存器字段说明（续）

位	字段	类型	复位	说明
9-8	IOSTR	R/W	0h	选择该 IO 的驱动强度控制源。 此设置控制低电流 (LC) 模式的驱动强度。可在 IOCURR 中选择更高的驱动强度 0h = 自动驱动强度，由 AON BATMON 根据电池电压控制。（VDDS 时最小 2mA） 1h = 最小驱动强度，由 AON_IOC:IOSTRMIN 控制（默认值为 3.3V 时最小 2mA） 2h = MED：中等驱动强度，由 AON_IOC:IOSTRMED 控制（默认值为 2.5V 时最小 2mA） 3h = 最大驱动强度，由 AON_IOC:IOSTRMAX 控制（默认值为 1.8V 时最小 2mA）
7	IOEV_RTC_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 RTC 事件置为有效 1：输入边沿检测将 RTC 事件置为有效
6	IOEV_MCU_WU_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 MCU_WU 事件置为有效 1：输入边沿检测将 MCU_WU 事件置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-38. IOCFG8 寄存器字段说明（续）

位	字段	类型	复位	说明
5-0	PORT_ID	R/W	0h	选择 DIO8 的用途 注意：除明确要求更改 PORT_ID 值的情况外，不应写入此字段。 0h =通用 IO 7h = AON 32 KHz 时钟 (SCLK_LF) 8h = AUX IO 9h = SSI0_RX：SSI0 RX Ah = SSI0_TX：SSI0 TX Bh = SSI0_FSS：SSI0 FSS Ch = SSI0_CLK：SSI0 CLK Dh = I2C_MSSDA：I2C 数据 Eh = I2C_MSSCL：I2C 时钟 Fh = UART0_RX：UART0 RX 10h = UART0_TX：UART0 TX 11h = UART0_CTS：UART0 CTS 12h = UART0_RTS：UART0 RTS 13h = UART1_RX：UART1 RX 14h = UART1_TX：UART1 TX 15h = UART1_CTS：UART1 CTS 16h = UART1_RTS：UART1 RTS 17h = PORT_EVENT0：端口事件 0 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 18h = PORT_EVENT1： 端口事件 1 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 19h = PORT_EVENT2： 端口事件 2 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ah = PORT_EVENT3： 端口事件 3 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Bh = PORT_EVENT4： 端口事件 4 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ch = PORT_EVENT5： 端口事件 5 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Dh = PORT_EVENT6： 端口事件 6 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Eh = PORT_EVENT7： 端口事件 7 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 20h = CPU_SWV： CPU SWV 21h = SSI1_RX：SSI1 RX 22h = SSI1_TX：SSI1 TX 23h = SSI1_FSS：SSI1 FSS 24h = SSI1_CLK：SSI1 CLK 25h = I2S_AD0：I2S 数据 0 26h = I2S_AD1： I2S 数据 1 27h = I2S_WCLK： I2S WCLK 28h = I2S_BCLK：I2S BCLK 29h = I2S_MCLK：I2S MCLK 2Eh = 射频内核跟踪

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-38. IOCFG8 寄存器字段说明（续）

位	字段	类型	复位	说明
				2Fh = 射频内核数据输出 0 30h = 射频内核数据输出 1 31h = 射频内核数据输出 2 32h = 射频内核数据输出 3 33h = 射频内核数据输入 0 34h = 射频内核数据输入 1 35h = 射频内核 SMI 数据链路输出 36h = 射频内核 SMI 数据链路输入 37h = 射频内核 SMI 命令链路输出 38h = 射频内核 SMI 命令链路输入

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.3.10 IOCFG9 寄存器（偏移 = 24h）[复位 = 00006000h]

图 13-32 展示了 IOCFG9，表 13-39 对其进行了介绍。

返回到汇总表。

DIO9 的配置

图 13-32. IOCFG9 寄存器

31	30	29	28	27	26	25	24
RESERVED	HYST_EN	IE	WU_CFG		IOMODE		
R-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h		
23	22	21	20	19	18	17	16
IOEV_AON_PROG2_EN	IOEV_AON_PROG1_EN	IOEV_AON_PROG0_EN	RESERVED		EDGE_IRQ_EN	EDGE_DET	
R/W-0h	R/W-0h	R/W-0h	R-0h		R/W-0h	R/W-0h	
15	14	13	12	11	10	9	8
RESERVED	PULL_CTL		SLEW_RED	IOCURR		IOSTR	
R-0h	R/W-3h		R/W-0h	R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
IOEV_RTC_EN	IOEV_MCU_WU_EN	PORT_ID					
R/W-0h	R/W-0h	R/W-0h					

表 13-39. IOCFG9 寄存器字段说明

位	字段	类型	复位	说明
31	RESERVED	R	0h	保留
30	HYST_EN	R/W	0h	0：输入迟滞禁用 1：输入迟滞使能
29	IE	R/W	0h	0：输入禁用 1：输入使能 注意：如果 IO 配置为 AUX PORT_ID = 0x08，则使能将被忽略。
28-27	WU_CFG	R/W	0h	如果 DIO 配置为 GPIO 或非 AON 外设信号（PORT_ID 0x00 或 >0x08）： 00：无唤醒 01：无唤醒 10：如果该焊盘变为低电平，则从关断状态唤醒。 11：如果该焊盘变为高电平，则从关断状态唤醒。 如果 IO 配置为 AON 外设信号或 AUX PORT_ID 0x01-0x08，则该寄存器仅设置是否使能唤醒。 00、01：唤醒禁用 10、11：唤醒使能 极性通过 AON 寄存器进行控制。 注意：当 MSB 置位时，IOC 将停用 DIO 的输出使能。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-39. IOCFG9 寄存器字段说明（续）

位	字段	类型	复位	说明
26-24	IOMODE	R/W	0h	IO 模式 不适用于配置为 AON 外设信号和 AUX PORT_ID 0x01-0x08 的 IO AUX 有自己的 open_source/drain 配置。 0x2: 保留。未定义的行为。 0x3：保留。未定义的行为。 0h = NORMAL：正常输入/输出 1h = INV：反相输入/输出 4h = OPENDR： 开漏， 正常输入/输出 5h = OPENDR_INV：开漏 反相输入/输出 6h = OPENSRC：开源 正常输入/输出 7h = OPENSRC_INV：开源 反相输入/输出
23	IOEV_AON_PROG2_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG2 事件置为有效 1：输入边沿检测将 AON_PROG2 事件置为有效
22	IOEV_AON_PROG1_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG1 事件置为有效 1：输入边沿检测将 AON_PROG1 事件置为有效
21	IOEV_AON_PROG0_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG0 事件置为有效 1：输入边沿检测将 AON_PROG0 事件置为有效
20-19	RESERVED	R	0h	保留
18	EDGE_IRQ_EN	R/W	0h	0：无中断生成 1：使能该 IO 的中断生成（仅在 EDGE_DET 已使能时有效）
17-16	EDGE_DET	R/W	0h	使能该 IO 上的边沿检测事件生成 0h = NONE：无边沿检测 1h = 负边沿检测 2h = 正边沿检测 3h = 正负边沿检测
15	RESERVED	R	0h	保留
14-13	PULL_CTL	R/W	3h	上拉/下拉控制 1h = DWN：下拉 2h = UP：上拉 3h = DIS：无上拉/下拉
12	SLEW_RED	R/W	0h	0：标称压摆率 1：在输出驱动器中降低压摆率。
11-10	IOCURR	R/W	0h	选择该 IO 的 IO 电流模式。 0h = 2MA：低电流 (LC) 模式：IOSTR 设置为 AUTO 时最小值为 2mA 1h = 4MA：高电流 (HC) 模式：IOSTR 设置为 AUTO 时最小值为 4mA 2h = 4_8MA：扩展电流 (EC) 模式：当 IOSTR 设为 AUTO 时，双驱动强度 IO 的最小值为 8mA（标准 IO 的最小值为 4mA）

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-39. IOCFG9 寄存器字段说明（续）

位	字段	类型	复位	说明
9-8	IOSTR	R/W	0h	选择该 IO 的驱动强度控制源。 此设置控制低电流 (LC) 模式的驱动强度。可在 IOCURR 中选择更高的驱动强度 0h = 自动驱动强度，由 AON BATMON 根据电池电压控制。（VDDS 时最小 2mA） 1h = 最小驱动强度，由 AON_IOC:IOSTRMIN 控制（默认值为 3.3V 时最小 2mA） 2h = MED：中等驱动强度，由 AON_IOC:IOSTRMED 控制（默认值为 2.5V 时最小 2mA） 3h = 最大驱动强度，由 AON_IOC:IOSTRMAX 控制（默认值为 1.8V 时最小 2mA）
7	IOEV_RTC_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 RTC 事件置为有效 1：输入边沿检测将 RTC 事件置为有效
6	IOEV_MCU_WU_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 MCU_WU 事件置为有效 1：输入边沿检测将 MCU_WU 事件置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-39. IOCFG9 寄存器字段说明（续）

位	字段	类型	复位	说明
5-0	PORT_ID	R/W	0h	选择 DIO9 的用途 注意：除明确要求更改 PORT_ID 值的情况外，不应写入此字段。 0h =通用 IO 7h = AON 32 KHz 时钟 (SCLK_LF) 8h = AUX IO 9h = SSI0_RX：SSI0 RX Ah = SSI0_TX：SSI0 TX Bh = SSI0_FSS：SSI0 FSS Ch = SSI0_CLK：SSI0 CLK Dh = I2C_MSSDA：I2C 数据 Eh = I2C_MSSCL：I2C 时钟 Fh = UART0_RX：UART0 RX 10h = UART0_TX：UART0 TX 11h = UART0_CTS：UART0 CTS 12h = UART0_RTS：UART0 RTS 13h = UART1_RX：UART1 RX 14h = UART1_TX：UART1 TX 15h = UART1_CTS：UART1 CTS 16h = UART1_RTS：UART1 RTS 17h = PORT_EVENT0：端口事件 0 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 18h = PORT_EVENT1： 端口事件 1 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 19h = PORT_EVENT2： 端口事件 2 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ah = PORT_EVENT3： 端口事件 3 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Bh = PORT_EVENT4： 端口事件 4 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ch = PORT_EVENT5： 端口事件 5 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Dh = PORT_EVENT6： 端口事件 6 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Eh = PORT_EVENT7： 端口事件 7 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 20h = CPU_SWV： CPU SWV 21h = SSI1_RX：SSI1 RX 22h = SSI1_TX：SSI1 TX 23h = SSI1_FSS：SSI1 FSS 24h = SSI1_CLK：SSI1 CLK 25h = I2S_AD0：I2S 数据 0 26h = I2S_AD1： I2S 数据 1 27h = I2S_WCLK： I2S WCLK 28h = I2S_BCLK：I2S BCLK 29h = I2S_MCLK：I2S MCLK 2Eh = 射频内核跟踪

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-39. IOCFG9 寄存器字段说明（续）

位	字段	类型	复位	说明
				2Fh = 射频内核数据输出 0 30h = 射频内核数据输出 1 31h = 射频内核数据输出 2 32h = 射频内核数据输出 3 33h = 射频内核数据输入 0 34h = 射频内核数据输入 1 35h = 射频内核 SMI 数据链路输出 36h = 射频内核 SMI 数据链路输入 37h = 射频内核 SMI 命令链路输出 38h = 射频内核 SMI 命令链路输入

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.3.11 IOCFG10 寄存器（偏移 = 28h）[复位 = 00006000h]

图 13-33 展示了 IOCFG10，表 13-40 对其进行了介绍。

返回到汇总表。

DIO10 的配置

图 13-33. IOCFG10 寄存器

31	30	29	28	27	26	25	24
RESERVED	HYST_EN	IE	WU_CFG		IOMODE		
R-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h		
23	22	21	20	19	18	17	16
IOEV_AON_PROG2_EN	IOEV_AON_PROG1_EN	IOEV_AON_PROG0_EN	RESERVED		EDGE_IRQ_EN	EDGE_DET	
R/W-0h	R/W-0h	R/W-0h	R-0h		R/W-0h	R/W-0h	
15	14	13	12	11	10	9	8
RESERVED	PULL_CTL		SLEW_RED	IOCURR		IOSTR	
R-0h	R/W-3h		R/W-0h	R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
IOEV_RTC_EN	IOEV_MCU_WU_EN	PORT_ID					
R/W-0h	R/W-0h	R/W-0h					

表 13-40. IOCFG10 寄存器字段说明

位	字段	类型	复位	说明
31	RESERVED	R	0h	保留
30	HYST_EN	R/W	0h	0：输入迟滞禁用 1：输入迟滞使能
29	IE	R/W	0h	0：输入禁用 1：输入使能 注意：如果 IO 配置为 AUX PORT_ID = 0x08，则使能将被忽略。
28-27	WU_CFG	R/W	0h	如果 DIO 配置为 GPIO 或非 AON 外设信号（PORT_ID 0x00 或 >0x08）： 00：无唤醒 01：无唤醒 10：如果该焊盘变为低电平，则从关断状态唤醒。 11：如果该焊盘变为高电平，则从关断状态唤醒。 如果 IO 配置为 AON 外设信号或 AUX PORT_ID 0x01-0x08，则该寄存器仅设置是否使能唤醒。 00、01：唤醒禁用 10、11：唤醒使能 极性通过 AON 寄存器进行控制。 注意：当 MSB 置位时，IOC 将停用 DIO 的输出使能。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-40. IOCFG10 寄存器字段说明（续）

位	字段	类型	复位	说明
26-24	IOMODE	R/W	0h	IO 模式 不适用于配置为 AON 外设信号和 AUX PORT_ID 0x01-0x08 的 IO AUX 有自己的 open_source/drain 配置。 0x2: 保留。未定义的行为。 0x3：保留。未定义的行为。 0h = NORMAL：正常输入/输出 1h = INV：反相输入/输出 4h = OPENDR： 开漏， 正常输入/输出 5h = OPENDR_INV：开漏 反相输入/输出 6h = OPENSRC：开源 正常输入/输出 7h = OPENSRC_INV：开源 反相输入/输出
23	IOEV_AON_PROG2_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG2 事件置为有效 1：输入边沿检测将 AON_PROG2 事件置为有效
22	IOEV_AON_PROG1_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG1 事件置为有效 1：输入边沿检测将 AON_PROG1 事件置为有效
21	IOEV_AON_PROG0_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG0 事件置为有效 1：输入边沿检测将 AON_PROG0 事件置为有效
20-19	RESERVED	R	0h	保留
18	EDGE_IRQ_EN	R/W	0h	0：无中断生成 1：使能该 IO 的中断生成（仅在 EDGE_DET 已使能时有效）
17-16	EDGE_DET	R/W	0h	使能该 IO 上的边沿检测事件生成 0h = NONE：无边沿检测 1h = 负边沿检测 2h = 正边沿检测 3h = 正负边沿检测
15	RESERVED	R	0h	保留
14-13	PULL_CTL	R/W	3h	上拉/下拉控制 1h = DWN：下拉 2h = UP：上拉 3h = DIS：无上拉/下拉
12	SLEW_RED	R/W	0h	0：标称压摆率 1：在输出驱动器中降低压摆率。
11-10	IOCURR	R/W	0h	选择该 IO 的 IO 电流模式。 0h = 2MA：低电流 (LC) 模式：IOSTR 设置为 AUTO 时最小值为 2mA 1h = 4MA：高电流 (HC) 模式：IOSTR 设置为 AUTO 时最小值为 4mA 2h = 4_8MA：扩展电流 (EC) 模式：当 IOSTR 设为 AUTO 时，双驱动强度 IO 的最小值为 8mA（标准 IO 的最小值为 4mA）

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-40. IOCFG10 寄存器字段说明（续）

位	字段	类型	复位	说明
9-8	IOSTR	R/W	0h	选择该 IO 的驱动强度控制源。 此设置控制低电流 (LC) 模式的驱动强度。可在 IOCURR 中选择更高的驱动强度 0h = 自动驱动强度，由 AON BATMON 根据电池电压控制。（VDDS 时最小 2mA） 1h = 最小驱动强度，由 AON_IOC:IOSTRMIN 控制（默认值为 3.3V 时最小 2mA） 2h = MED：中等驱动强度，由 AON_IOC:IOSTRMED 控制（默认值为 2.5V 时最小 2mA） 3h = 最大驱动强度，由 AON_IOC:IOSTRMAX 控制（默认值为 1.8V 时最小 2mA）
7	IOEV_RTC_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 RTC 事件置为有效 1：输入边沿检测将 RTC 事件置为有效
6	IOEV_MCU_WU_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 MCU_WU 事件置为有效 1：输入边沿检测将 MCU_WU 事件置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-40. IOCFG10 寄存器字段说明（续）

位	字段	类型	复位	说明
5-0	PORT_ID	R/W	0h	选择 DIO10 的用途 注意：除明确要求更改 PORT_ID 值的情况外，不应写入此字段。 0h =通用 IO 7h = AON 32 KHz 时钟 (SCLK_LF) 8h = AUX IO 9h = SSI0_RX：SSI0 RX Ah = SSI0_TX：SSI0 TX Bh = SSI0_FSS：SSI0 FSS Ch = SSI0_CLK：SSI0 CLK Dh = I2C_MSSDA：I2C 数据 Eh = I2C_MSSCL：I2C 时钟 Fh = UART0_RX：UART0 RX 10h = UART0_TX：UART0 TX 11h = UART0_CTS：UART0 CTS 12h = UART0_RTS：UART0 RTS 13h = UART1_RX：UART1 RX 14h = UART1_TX：UART1 TX 15h = UART1_CTS：UART1 CTS 16h = UART1_RTS：UART1 RTS 17h = PORT_EVENT0：端口事件 0 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 18h = PORT_EVENT1： 端口事件 1 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 19h = PORT_EVENT2： 端口事件 2 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ah = PORT_EVENT3： 端口事件 3 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Bh = PORT_EVENT4： 端口事件 4 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ch = PORT_EVENT5： 端口事件 5 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Dh = PORT_EVENT6： 端口事件 6 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Eh = PORT_EVENT7： 端口事件 7 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 20h = CPU_SWV： CPU SWV 21h = SSI1_RX：SSI1 RX 22h = SSI1_TX：SSI1 TX 23h = SSI1_FSS：SSI1 FSS 24h = SSI1_CLK：SSI1 CLK 25h = I2S_AD0：I2S 数据 0 26h = I2S_AD1： I2S 数据 1 27h = I2S_WCLK： I2S WCLK 28h = I2S_BCLK：I2S BCLK 29h = I2S_MCLK：I2S MCLK 2Eh = 射频内核跟踪

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-40. IOCFG10 寄存器字段说明（续）

位	字段	类型	复位	说明
				2Fh = 射频内核数据输出 0 30h = 射频内核数据输出 1 31h = 射频内核数据输出 2 32h = 射频内核数据输出 3 33h = 射频内核数据输入 0 34h = 射频内核数据输入 1 35h = 射频内核 SMI 数据链路输出 36h = 射频内核 SMI 数据链路输入 37h = 射频内核 SMI 命令链路输出 38h = 射频内核 SMI 命令链路输入

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.3.12 IOCFG11 寄存器（偏移 = 2Ch）[复位 = 00006000h]

图 13-34 展示了 IOCFG11，表 13-41 对其进行了介绍。

返回到汇总表。

DIO11 的配置

图 13-34. IOCFG11 寄存器

31	30	29	28	27	26	25	24
RESERVED	HYST_EN	IE	WU_CFG		IOMODE		
R-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h		
23	22	21	20	19	18	17	16
IOEV_AON_PROG2_EN	IOEV_AON_PROG1_EN	IOEV_AON_PROG0_EN	RESERVED		EDGE_IRQ_EN	EDGE_DET	
R/W-0h	R/W-0h	R/W-0h	R-0h		R/W-0h	R/W-0h	
15	14	13	12	11	10	9	8
RESERVED	PULL_CTL		SLEW_RED	IOCURR		IOSTR	
R-0h	R/W-3h		R/W-0h	R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
IOEV_RTC_EN	IOEV_MCU_WU_EN	PORT_ID					
R/W-0h	R/W-0h	R/W-0h					

表 13-41. IOCFG11 寄存器字段说明

位	字段	类型	复位	说明
31	RESERVED	R	0h	保留
30	HYST_EN	R/W	0h	0：输入迟滞禁用 1：输入迟滞使能
29	IE	R/W	0h	0：输入禁用 1：输入使能 注意：如果 IO 配置为 AUX PORT_ID = 0x08，则使能将被忽略。
28-27	WU_CFG	R/W	0h	如果 DIO 配置为 GPIO 或非 AON 外设信号（PORT_ID 0x00 或 >0x08）： 00：无唤醒 01：无唤醒 10：如果该焊盘变为低电平，则从关断状态唤醒。 11：如果该焊盘变为高电平，则从关断状态唤醒。 如果 IO 配置为 AON 外设信号或 AUX PORT_ID 0x01-0x08，则该寄存器仅设置是否使能唤醒。 00、01：唤醒禁用 10、11：唤醒使能 极性通过 AON 寄存器进行控制。 注意：当 MSB 置位时，IOC 将停用 DIO 的输出使能。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-41. IOCFG11 寄存器字段说明 （续）

位	字段	类型	复位	说明
26-24	IOMODE	R/W	0h	IO 模式 不适用于配置为 AON 外设信号和 AUX PORT_ID 0x01-0x08 的 IO AUX 有自己的 open_source/drain 配置。 0x2: 保留。未定义的行为。 0x3：保留。未定义的行为。 0h = NORMAL：正常输入/输出 1h = INV：反相输入/输出 4h = OPENDR： 开漏， 正常输入/输出 5h = OPENDR_INV：开漏 反相输入/输出 6h = OPENSRC：开源 正常输入/输出 7h = OPENSRC_INV：开源 反相输入/输出
23	IOEV_AON_PROG2_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG2 事件置为有效 1：输入边沿检测将 AON_PROG2 事件置为有效
22	IOEV_AON_PROG1_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG1 事件置为有效 1：输入边沿检测将 AON_PROG1 事件置为有效
21	IOEV_AON_PROG0_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG0 事件置为有效 1：输入边沿检测将 AON_PROG0 事件置为有效
20-19	RESERVED	R	0h	保留
18	EDGE_IRQ_EN	R/W	0h	0：无中断生成 1：使能该 IO 的中断生成（仅在 EDGE_DET 已使能时有效）
17-16	EDGE_DET	R/W	0h	使能该 IO 上的边沿检测事件生成 0h = NONE：无边沿检测 1h = 负边沿检测 2h = 正边沿检测 3h = 正负边沿检测
15	RESERVED	R	0h	保留
14-13	PULL_CTL	R/W	3h	上拉/下拉控制 1h = DWN：下拉 2h = UP：上拉 3h = DIS：无上拉/下拉
12	SLEW_RED	R/W	0h	0：标称压摆率 1：在输出驱动器中降低压摆率。
11-10	IOCURR	R/W	0h	选择该 IO 的 IO 电流模式。 0h = 2MA：低电流 (LC) 模式：IOSTR 设置为 AUTO 时最小值为 2mA 1h = 4MA：高电流 (HC) 模式：IOSTR 设置为 AUTO 时最小值为 4mA 2h = 4_8MA：扩展电流 (EC) 模式：当 IOSTR 设为 AUTO 时，双驱动强度 IO 的最小值为 8mA（标准 IO 的最小值为 4mA）

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-41. IOCFG11 寄存器字段说明（续）

位	字段	类型	复位	说明
9-8	IOSTR	R/W	0h	选择该 IO 的驱动强度控制源。 此设置控制低电流 (LC) 模式的驱动强度。可在 IOCURR 中选择更高的驱动强度 0h = 自动驱动强度，由 AON BATMON 根据电池电压控制。（VDDS 时最小 2mA） 1h = 最小驱动强度，由 AON_IOC:IOSTRMIN 控制（默认值为 3.3V 时最小 2mA） 2h = MED：中等驱动强度，由 AON_IOC:IOSTRMED 控制（默认值为 2.5V 时最小 2mA） 3h = 最大驱动强度，由 AON_IOC:IOSTRMAX 控制（默认值为 1.8V 时最小 2mA）
7	IOEV_RTC_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 RTC 事件置为有效 1：输入边沿检测将 RTC 事件置为有效
6	IOEV_MCU_WU_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 MCU_WU 事件置为有效 1：输入边沿检测将 MCU_WU 事件置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-41. IOCFG11 寄存器字段说明 （续）

位	字段	类型	复位	说明
5-0	PORT_ID	R/W	0h	选择 DIO11 的用途 注意：除明确要求更改 PORT_ID 值的情况外，不应写入此字段。 0h =通用 IO 7h = AON 32 KHz 时钟 (SCLK_LF) 8h = AUX IO 9h = SSI0_RX：SSI0 RX Ah = SSI0_TX：SSI0 TX Bh = SSI0_FSS：SSI0 FSS Ch = SSI0_CLK：SSI0 CLK Dh = I2C_MSSDA：I2C 数据 Eh = I2C_MSSCL：I2C 时钟 Fh = UART0_RX：UART0 RX 10h = UART0_TX：UART0 TX 11h = UART0_CTS：UART0 CTS 12h = UART0_RTS：UART0 RTS 13h = UART1_RX：UART1 RX 14h = UART1_TX：UART1 TX 15h = UART1_CTS：UART1 CTS 16h = UART1_RTS：UART1 RTS 17h = PORT_EVENT0：端口事件 0 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等）， 可用作通用 IO 事件 18h = PORT_EVENT1： 端口事件 1 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等）， 可用作通用 IO 事件 19h = PORT_EVENT2： 端口事件 2 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等）， 可用作通用 IO 事件 1Ah = PORT_EVENT3： 端口事件 3 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等）， 可用作通用 IO 事件 1Bh = PORT_EVENT4： 端口事件 4 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等）， 可用作通用 IO 事件 1Ch = PORT_EVENT5： 端口事件 5 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等）， 可用作通用 IO 事件 1Dh = PORT_EVENT6： 端口事件 6 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等）， 可用作通用 IO 事件 1Eh = PORT_EVENT7： 端口事件 7 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等）， 可用作通用 IO 事件 20h = CPU_SWV： CPU SWV 21h = SSI1_RX：SSI1 RX 22h = SSI1_TX：SSI1 TX 23h = SSI1_FSS：SSI1 FSS 24h = SSI1_CLK：SSI1 CLK 25h = I2S_AD0：I2S 数据 0 26h = I2S_AD1： I2S 数据 1 27h = I2S_WCLK： I2S WCLK 28h = I2S_BCLK：I2S BCLK 29h = I2S_MCLK：I2S MCLK 2Eh = 射频内核跟踪

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-41. IOCFG11 寄存器字段说明（续）

位	字段	类型	复位	说明
				2Fh = 射频内核数据输出 0 30h = 射频内核数据输出 1 31h = 射频内核数据输出 2 32h = 射频内核数据输出 3 33h = 射频内核数据输入 0 34h = 射频内核数据输入 1 35h = 射频内核 SMI 数据链路输出 36h = 射频内核 SMI 数据链路输入 37h = 射频内核 SMI 命令链路输出 38h = 射频内核 SMI 命令链路输入

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.3.13 IOCFG12 寄存器（偏移 = 30h）[复位 = 00006000h]

图 13-35 展示了 IOCFG12，表 13-42 对其进行了介绍。

返回到汇总表。

DIO12 的配置

图 13-35. IOCFG12 寄存器

31	30	29	28	27	26	25	24
RESERVED	HYST_EN	IE	WU_CFG		IOMODE		
R-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h		
23	22	21	20	19	18	17	16
IOEV_AON_PROG2_EN	IOEV_AON_PROG1_EN	IOEV_AON_PROG0_EN	RESERVED		EDGE_IRQ_EN	EDGE_DET	
R/W-0h	R/W-0h	R/W-0h	R-0h		R/W-0h	R/W-0h	
15	14	13	12	11	10	9	8
RESERVED	PULL_CTL		SLEW_RED	IOCURR		IOSTR	
R-0h	R/W-3h		R/W-0h	R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
IOEV_RTC_EN	IOEV_MCU_WU_EN	PORT_ID					
R/W-0h	R/W-0h	R/W-0h					

表 13-42. IOCFG12 寄存器字段说明

位	字段	类型	复位	说明
31	RESERVED	R	0h	保留
30	HYST_EN	R/W	0h	0：输入迟滞禁用 1：输入迟滞使能
29	IE	R/W	0h	0：输入禁用 1：输入使能 注意：如果 IO 配置为 AUX PORT_ID = 0x08，则使能将被忽略。
28-27	WU_CFG	R/W	0h	如果 DIO 配置为 GPIO 或非 AON 外设信号（PORT_ID 0x00 或 >0x08）： 00：无唤醒 01：无唤醒 10：如果该焊盘变为低电平，则从关断状态唤醒。 11：如果该焊盘变为高电平，则从关断状态唤醒。 如果 IO 配置为 AON 外设信号或 AUX PORT_ID 0x01-0x08，则该寄存器仅设置是否使能唤醒。 00、01：唤醒禁用 10、11：唤醒使能 极性通过 AON 寄存器进行控制。 注意：当 MSB 置位时，IOC 将停用 DIO 的输出使能。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-42. IOCFG12 寄存器字段说明（续）

位	字段	类型	复位	说明
26-24	IOMODE	R/W	0h	IO 模式 不适用于配置为 AON 外设信号和 AUX PORT_ID 0x01-0x08 的 IO AUX 有自己的 open_source/drain 配置。 0x2: 保留。未定义的行为。 0x3：保留。未定义的行为。 0h = NORMAL：正常输入/输出 1h = INV：反相输入/输出 4h = OPENDR： 开漏， 正常输入/输出 5h = OPENDR_INV：开漏 反相输入/输出 6h = OPENSRC：开源 正常输入/输出 7h = OPENSRC_INV：开源 反相输入/输出
23	IOEV_AON_PROG2_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG2 事件置为有效 1：输入边沿检测将 AON_PROG2 事件置为有效
22	IOEV_AON_PROG1_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG1 事件置为有效 1：输入边沿检测将 AON_PROG1 事件置为有效
21	IOEV_AON_PROG0_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG0 事件置为有效 1：输入边沿检测将 AON_PROG0 事件置为有效
20-19	RESERVED	R	0h	保留
18	EDGE_IRQ_EN	R/W	0h	0：无中断生成 1：使能该 IO 的中断生成（仅在 EDGE_DET 已使能时有效）
17-16	EDGE_DET	R/W	0h	使能该 IO 上的边沿检测事件生成 0h = NONE：无边沿检测 1h = 负边沿检测 2h = 正边沿检测 3h = 正负边沿检测
15	RESERVED	R	0h	保留
14-13	PULL_CTL	R/W	3h	上拉/下拉控制 1h = DWN：下拉 2h = UP：上拉 3h = DIS：无上拉/下拉
12	SLEW_RED	R/W	0h	0：标称压摆率 1：在输出驱动器中降低压摆率。
11-10	IOCURR	R/W	0h	选择该 IO 的 IO 电流模式。 0h = 2MA：低电流 (LC) 模式：IOSTR 设置为 AUTO 时最小值为 2mA 1h = 4MA：高电流 (HC) 模式：IOSTR 设置为 AUTO 时最小值为 4mA 2h = 4_8MA：扩展电流 (EC) 模式：当 IOSTR 设为 AUTO 时，双驱动强度 IO 的最小值为 8mA（标准 IO 的最小值为 4mA）

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-42. IOCFG12 寄存器字段说明（续）

位	字段	类型	复位	说明
9-8	IOSTR	R/W	0h	选择该 IO 的驱动强度控制源。 此设置控制低电流 (LC) 模式的驱动强度。可在 IOCURR 中选择更高的驱动强度 0h = 自动驱动强度，由 AON BATMON 根据电池电压控制。（VDDS 时最小 2mA） 1h = 最小驱动强度，由 AON_IOC:IOSTRMIN 控制（默认值为 3.3V 时最小 2mA） 2h = MED：中等驱动强度，由 AON_IOC:IOSTRMED 控制（默认值为 2.5V 时最小 2mA） 3h = 最大驱动强度，由 AON_IOC:IOSTRMAX 控制（默认值为 1.8V 时最小 2mA）
7	IOEV_RTC_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 RTC 事件置为有效 1：输入边沿检测将 RTC 事件置为有效
6	IOEV_MCU_WU_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 MCU_WU 事件置为有效 1：输入边沿检测将 MCU_WU 事件置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-42. IOCFG12 寄存器字段说明（续）

位	字段	类型	复位	说明
5-0	PORT_ID	R/W	0h	选择 DIO12 的用途 注意：除明确要求更改 PORT_ID 值的情况外，不应写入此字段。 0h =通用 IO 7h = AON 32 KHz 时钟 (SCLK_LF) 8h = AUX IO 9h = SSI0_RX：SSI0 RX Ah = SSI0_TX：SSI0 TX Bh = SSI0_FSS：SSI0 FSS Ch = SSI0_CLK：SSI0 CLK Dh = I2C_MSSDA：I2C 数据 Eh = I2C_MSSCL：I2C 时钟 Fh = UART0_RX：UART0 RX 10h = UART0_TX：UART0 TX 11h = UART0_CTS：UART0 CTS 12h = UART0_RTS：UART0 RTS 13h = UART1_RX：UART1 RX 14h = UART1_TX：UART1 TX 15h = UART1_CTS：UART1 CTS 16h = UART1_RTS：UART1 RTS 17h = PORT_EVENT0：端口事件 0 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 18h = PORT_EVENT1： 端口事件 1 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 19h = PORT_EVENT2： 端口事件 2 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ah = PORT_EVENT3： 端口事件 3 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Bh = PORT_EVENT4： 端口事件 4 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ch = PORT_EVENT5： 端口事件 5 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Dh = PORT_EVENT6： 端口事件 6 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Eh = PORT_EVENT7： 端口事件 7 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 20h = CPU_SWV： CPU SWV 21h = SSI1_RX：SSI1 RX 22h = SSI1_TX：SSI1 TX 23h = SSI1_FSS：SSI1 FSS 24h = SSI1_CLK：SSI1 CLK 25h = I2S_AD0：I2S 数据 0 26h = I2S_AD1： I2S 数据 1 27h = I2S_WCLK： I2S WCLK 28h = I2S_BCLK：I2S BCLK 29h = I2S_MCLK：I2S MCLK 2Eh = 射频内核跟踪

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-42. IOCFG12 寄存器字段说明（续）

位	字段	类型	复位	说明
				2Fh = 射频内核数据输出 0 30h = 射频内核数据输出 1 31h = 射频内核数据输出 2 32h = 射频内核数据输出 3 33h = 射频内核数据输入 0 34h = 射频内核数据输入 1 35h = 射频内核 SMI 数据链路输出 36h = 射频内核 SMI 数据链路输入 37h = 射频内核 SMI 命令链路输出 38h = 射频内核 SMI 命令链路输入

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.3.14 IOCFG13 寄存器（偏移 = 34h）[复位 = 00006000h]

图 13-36 展示了 IOCFG13，表 13-43 对其进行了介绍。

返回到汇总表。

DIO13 的配置

图 13-36. IOCFG13 寄存器

31	30	29	28	27	26	25	24
RESERVED	HYST_EN	IE	WU_CFG		IOMODE		
R-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h		
23	22	21	20	19	18	17	16
IOEV_AON_PROG2_EN	IOEV_AON_PROG1_EN	IOEV_AON_PROG0_EN	RESERVED		EDGE_IRQ_EN	EDGE_DET	
R/W-0h	R/W-0h	R/W-0h	R-0h		R/W-0h	R/W-0h	
15	14	13	12	11	10	9	8
RESERVED	PULL_CTL		SLEW_RED	IOCURR		IOSTR	
R-0h	R/W-3h		R/W-0h	R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
IOEV_RTC_EN	IOEV_MCU_WU_EN	PORT_ID					
R/W-0h	R/W-0h	R/W-0h					

表 13-43. IOCFG13 寄存器字段说明

位	字段	类型	复位	说明
31	RESERVED	R	0h	保留
30	HYST_EN	R/W	0h	0：输入迟滞禁用 1：输入迟滞使能
29	IE	R/W	0h	0：输入禁用 1：输入使能 注意：如果 IO 配置为 AUX PORT_ID = 0x08，则使能将被忽略。
28-27	WU_CFG	R/W	0h	如果 DIO 配置为 GPIO 或非 AON 外设信号（PORT_ID 0x00 或 >0x08）： 00：无唤醒 01：无唤醒 10：如果该焊盘变为低电平，则从关断状态唤醒。 11：如果该焊盘变为高电平，则从关断状态唤醒。 如果 IO 配置为 AON 外设信号或 AUX PORT_ID 0x01-0x08，则该寄存器仅设置是否使能唤醒。 00、01：唤醒禁用 10、11：唤醒使能 极性通过 AON 寄存器进行控制。 注意：当 MSB 置位时，IOC 将停用 DIO 的输出使能。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-43. IOCFG13 寄存器字段说明 (续)

位	字段	类型	复位	说明
26-24	IOMODE	R/W	0h	IO 模式 不适用于配置为 AON 外设信号和 AUX PORT_ID 0x01-0x08 的 IO AUX 有自己的 open_source/drain 配置。 0x2: 保留。未定义的行为。 0x3：保留。未定义的行为。 0h = NORMAL：正常输入/输出 1h = INV：反相输入/输出 4h = OPENDR： 开漏， 正常输入/输出 5h = OPENDR_INV：开漏 反相输入/输出 6h = OPENSRC：开源 正常输入/输出 7h = OPENSRC_INV：开源 反相输入/输出
23	IOEV_AON_PROG2_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG2 事件置为有效 1：输入边沿检测将 AON_PROG2 事件置为有效
22	IOEV_AON_PROG1_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG1 事件置为有效 1：输入边沿检测将 AON_PROG1 事件置为有效
21	IOEV_AON_PROG0_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG0 事件置为有效 1：输入边沿检测将 AON_PROG0 事件置为有效
20-19	RESERVED	R	0h	保留
18	EDGE_IRQ_EN	R/W	0h	0：无中断生成 1：使能该 IO 的中断生成 (仅在 EDGE_DET 已使能时有效)
17-16	EDGE_DET	R/W	0h	使能该 IO 上的边沿检测事件生成 0h = NONE：无边沿检测 1h = 负边沿检测 2h = 正边沿检测 3h = 正负边沿检测
15	RESERVED	R	0h	保留
14-13	PULL_CTL	R/W	3h	上拉/下拉控制 1h = DWN：下拉 2h = UP：上拉 3h = DIS：无上拉/下拉
12	SLEW_RED	R/W	0h	0：标称压摆率 1：在输出驱动器中降低压摆率。
11-10	IOCURR	R/W	0h	选择该 IO 的 IO 电流模式。 0h = 2MA：低电流 (LC) 模式：IOSTR 设置为 AUTO 时最小值为 2mA 1h = 4MA：高电流 (HC) 模式：IOSTR 设置为 AUTO 时最小值为 4mA 2h = 4_8MA：扩展电流 (EC) 模式：当 IOSTR 设为 AUTO 时，双驱动强度 IO 的最小值为 8mA (标准 IO 的最小值为 4mA)

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-43. IOCFG13 寄存器字段说明（续）

位	字段	类型	复位	说明
9-8	IOSTR	R/W	0h	选择该 IO 的驱动强度控制源。 此设置控制低电流 (LC) 模式的驱动强度。可在 IOCURR 中选择更高的驱动强度 0h = 自动驱动强度，由 AON BATMON 根据电池电压控制。（VDDS 时最小 2mA） 1h = 最小驱动强度，由 AON_IOC:IOSTRMIN 控制（默认值为 3.3V 时最小 2mA） 2h = MED：中等驱动强度，由 AON_IOC:IOSTRMED 控制（默认值为 2.5V 时最小 2mA） 3h = 最大驱动强度，由 AON_IOC:IOSTRMAX 控制（默认值为 1.8V 时最小 2mA）
7	IOEV_RTC_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 RTC 事件置为有效 1：输入边沿检测将 RTC 事件置为有效
6	IOEV_MCU_WU_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 MCU_WU 事件置为有效 1：输入边沿检测将 MCU_WU 事件置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-43. IOCFG13 寄存器字段说明 （续）

位	字段	类型	复位	说明
5-0	PORT_ID	R/W	0h	选择 DIO13 的用途 注意：除明确要求更改 PORT_ID 值的情况外，不应写入此字段。 0h =通用 IO 7h = AON 32 KHz 时钟 (SCLK_LF) 8h = AUX IO 9h = SSI0_RX：SSI0 RX Ah = SSI0_TX：SSI0 TX Bh = SSI0_FSS：SSI0 FSS Ch = SSI0_CLK：SSI0 CLK Dh = I2C_MSSDA：I2C 数据 Eh = I2C_MSSCL：I2C 时钟 Fh = UART0_RX：UART0 RX 10h = UART0_TX：UART0 TX 11h = UART0_CTS：UART0 CTS 12h = UART0_RTS：UART0 RTS 13h = UART1_RX：UART1 RX 14h = UART1_TX：UART1 TX 15h = UART1_CTS：UART1 CTS 16h = UART1_RTS：UART1 RTS 17h = PORT_EVENT0：端口事件 0 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等）， 可用作通用 IO 事件 18h = PORT_EVENT1： 端口事件 1 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等）， 可用作通用 IO 事件 19h = PORT_EVENT2： 端口事件 2 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等）， 可用作通用 IO 事件 1Ah = PORT_EVENT3： 端口事件 3 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等）， 可用作通用 IO 事件 1Bh = PORT_EVENT4： 端口事件 4 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等）， 可用作通用 IO 事件 1Ch = PORT_EVENT5： 端口事件 5 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等）， 可用作通用 IO 事件 1Dh = PORT_EVENT6： 端口事件 6 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等）， 可用作通用 IO 事件 1Eh = PORT_EVENT7： 端口事件 7 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等）， 可用作通用 IO 事件 20h = CPU_SWV： CPU SWV 21h = SSI1_RX：SSI1 RX 22h = SSI1_TX：SSI1 TX 23h = SSI1_FSS：SSI1 FSS 24h = SSI1_CLK：SSI1 CLK 25h = I2S_AD0：I2S 数据 0 26h = I2S_AD1： I2S 数据 1 27h = I2S_WCLK： I2S WCLK 28h = I2S_BCLK：I2S BCLK 29h = I2S_MCLK：I2S MCLK 2Eh = 射频内核跟踪

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-43. IOCFG13 寄存器字段说明（续）

位	字段	类型	复位	说明
				2Fh = 射频内核数据输出 0 30h = 射频内核数据输出 1 31h = 射频内核数据输出 2 32h = 射频内核数据输出 3 33h = 射频内核数据输入 0 34h = 射频内核数据输入 1 35h = 射频内核 SMI 数据链路输出 36h = 射频内核 SMI 数据链路输入 37h = 射频内核 SMI 命令链路输出 38h = 射频内核 SMI 命令链路输入

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.3.15 IOCFG14 寄存器（偏移 = 38h）[复位 = 00006000h]

图 13-37 展示了 IOCFG14，表 13-44 对其进行了介绍。

返回到汇总表。

DIO14 的配置

图 13-37. IOCFG14 寄存器

31	30	29	28	27	26	25	24
RESERVED	HYST_EN	IE	WU_CFG		IOMODE		
R-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h		
23	22	21	20	19	18	17	16
IOEV_AON_PROG2_EN	IOEV_AON_PROG1_EN	IOEV_AON_PROG0_EN	RESERVED		EDGE_IRQ_EN	EDGE_DET	
R/W-0h	R/W-0h	R/W-0h	R-0h		R/W-0h	R/W-0h	
15	14	13	12	11	10	9	8
RESERVED	PULL_CTL		SLEW_RED	IOCURR		IOSTR	
R-0h	R/W-3h		R/W-0h	R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
IOEV_RTC_EN	IOEV_MCU_WU_EN	PORT_ID					
R/W-0h	R/W-0h	R/W-0h					

表 13-44. IOCFG14 寄存器字段说明

位	字段	类型	复位	说明
31	RESERVED	R	0h	保留
30	HYST_EN	R/W	0h	0：输入迟滞禁用 1：输入迟滞使能
29	IE	R/W	0h	0：输入禁用 1：输入使能 注意：如果 IO 配置为 AUX PORT_ID = 0x08，则使能将被忽略。
28-27	WU_CFG	R/W	0h	如果 DIO 配置为 GPIO 或非 AON 外设信号（PORT_ID 0x00 或 >0x08）： 00：无唤醒 01：无唤醒 10：如果该焊盘变为低电平，则从关断状态唤醒。 11：如果该焊盘变为高电平，则从关断状态唤醒。 如果 IO 配置为 AON 外设信号或 AUX PORT_ID 0x01-0x08，则该寄存器仅设置是否使能唤醒。 00、01：唤醒禁用 10、11：唤醒使能 极性通过 AON 寄存器进行控制。 注意：当 MSB 置位时，IOC 将停用 DIO 的输出使能。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-44. IOCFG14 寄存器字段说明（续）

位	字段	类型	复位	说明
26-24	IOMODE	R/W	0h	IO 模式 不适用于配置为 AON 外设信号和 AUX PORT_ID 0x01-0x08 的 IO AUX 有自己的 open_source/drain 配置。 0x2: 保留。未定义的行为。 0x3：保留。未定义的行为。 0h = NORMAL：正常输入/输出 1h = INV：反相输入/输出 4h = OPENDR： 开漏， 正常输入/输出 5h = OPENDR_INV：开漏 反相输入/输出 6h = OPENSRC：开源 正常输入/输出 7h = OPENSRC_INV：开源 反相输入/输出
23	IOEV_AON_PROG2_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG2 事件置为有效 1：输入边沿检测将 AON_PROG2 事件置为有效
22	IOEV_AON_PROG1_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG1 事件置为有效 1：输入边沿检测将 AON_PROG1 事件置为有效
21	IOEV_AON_PROG0_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG0 事件置为有效 1：输入边沿检测将 AON_PROG0 事件置为有效
20-19	RESERVED	R	0h	保留
18	EDGE_IRQ_EN	R/W	0h	0：无中断生成 1：使能该 IO 的中断生成（仅在 EDGE_DET 已使能时有效）
17-16	EDGE_DET	R/W	0h	使能该 IO 上的边沿检测事件生成 0h = NONE：无边沿检测 1h = 负边沿检测 2h = 正边沿检测 3h = 正负边沿检测
15	RESERVED	R	0h	保留
14-13	PULL_CTL	R/W	3h	上拉/下拉控制 1h = DWN：下拉 2h = UP：上拉 3h = DIS：无上拉/下拉
12	SLEW_RED	R/W	0h	0：标称压摆率 1：在输出驱动器中降低压摆率。
11-10	IOCURR	R/W	0h	选择该 IO 的 IO 电流模式。 0h = 2MA：低电流 (LC) 模式：IOSTR 设置为 AUTO 时最小值为 2mA 1h = 4MA：高电流 (HC) 模式：IOSTR 设置为 AUTO 时最小值为 4mA 2h = 4_8MA：扩展电流 (EC) 模式：当 IOSTR 设为 AUTO 时，双驱动强度 IO 的最小值为 8mA（标准 IO 的最小值为 4mA）

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-44. IOCFG14 寄存器字段说明（续）

位	字段	类型	复位	说明
9-8	IOSTR	R/W	0h	选择该 IO 的驱动强度控制源。 此设置控制低电流 (LC) 模式的驱动强度。可在 IOCURR 中选择更高的驱动强度 0h = 自动驱动强度，由 AON BATMON 根据电池电压控制。（VDDS 时最小 2mA） 1h = 最小驱动强度，由 AON_IOC:IOSTRMIN 控制（默认值为 3.3V 时最小 2mA） 2h = MED：中等驱动强度，由 AON_IOC:IOSTRMED 控制（默认值为 2.5V 时最小 2mA） 3h = 最大驱动强度，由 AON_IOC:IOSTRMAX 控制（默认值为 1.8V 时最小 2mA）
7	IOEV_RTC_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 RTC 事件置为有效 1：输入边沿检测将 RTC 事件置为有效
6	IOEV_MCU_WU_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 MCU_WU 事件置为有效 1：输入边沿检测将 MCU_WU 事件置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-44. IOCFG14 寄存器字段说明（续）

位	字段	类型	复位	说明
5-0	PORT_ID	R/W	0h	选择 DIO14 的用途 注意：除明确要求更改 PORT_ID 值的情况外，不应写入此字段。 0h =通用 IO 7h = AON 32 KHz 时钟 (SCLK_LF) 8h = AUX IO 9h = SSI0_RX：SSI0 RX Ah = SSI0_TX：SSI0 TX Bh = SSI0_FSS：SSI0 FSS Ch = SSI0_CLK：SSI0 CLK Dh = I2C_MSSDA：I2C 数据 Eh = I2C_MSSCL：I2C 时钟 Fh = UART0_RX：UART0 RX 10h = UART0_TX：UART0 TX 11h = UART0_CTS：UART0 CTS 12h = UART0_RTS：UART0 RTS 13h = UART1_RX：UART1 RX 14h = UART1_TX：UART1 TX 15h = UART1_CTS：UART1 CTS 16h = UART1_RTS：UART1 RTS 17h = PORT_EVENT0：端口事件 0 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 18h = PORT_EVENT1： 端口事件 1 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 19h = PORT_EVENT2： 端口事件 2 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ah = PORT_EVENT3： 端口事件 3 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Bh = PORT_EVENT4： 端口事件 4 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ch = PORT_EVENT5： 端口事件 5 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Dh = PORT_EVENT6： 端口事件 6 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Eh = PORT_EVENT7： 端口事件 7 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 20h = CPU_SWV： CPU SWV 21h = SSI1_RX：SSI1 RX 22h = SSI1_TX：SSI1 TX 23h = SSI1_FSS：SSI1 FSS 24h = SSI1_CLK：SSI1 CLK 25h = I2S_AD0：I2S 数据 0 26h = I2S_AD1： I2S 数据 1 27h = I2S_WCLK： I2S WCLK 28h = I2S_BCLK：I2S BCLK 29h = I2S_MCLK：I2S MCLK 2Eh = 射频内核跟踪

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-44. IOCFG14 寄存器字段说明（续）

位	字段	类型	复位	说明
				2Fh = 射频内核数据输出 0 30h = 射频内核数据输出 1 31h = 射频内核数据输出 2 32h = 射频内核数据输出 3 33h = 射频内核数据输入 0 34h = 射频内核数据输入 1 35h = 射频内核 SMI 数据链路输出 36h = 射频内核 SMI 数据链路输入 37h = 射频内核 SMI 命令链路输出 38h = 射频内核 SMI 命令链路输入

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.3.16 IOCFG15 寄存器（偏移 = 3Ch）[复位 = 00006000h]

图 13-38 展示了 IOCFG15，表 13-45 对其进行了介绍。

返回到汇总表。

DIO15 的配置

图 13-38. IOCFG15 寄存器

31	30	29	28	27	26	25	24
RESERVED	HYST_EN	IE	WU_CFG		IOMODE		
R-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h		
23	22	21	20	19	18	17	16
IOEV_AON_PROG2_EN	IOEV_AON_PROG1_EN	IOEV_AON_PROG0_EN	RESERVED		EDGE_IRQ_EN	EDGE_DET	
R/W-0h	R/W-0h	R/W-0h	R-0h		R/W-0h	R/W-0h	
15	14	13	12	11	10	9	8
RESERVED	PULL_CTL		SLEW_RED	IOCURR		IOSTR	
R-0h	R/W-3h		R/W-0h	R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
IOEV_RTC_EN	IOEV_MCU_WU_EN	PORT_ID					
R/W-0h	R/W-0h	R/W-0h					

表 13-45. IOCFG15 寄存器字段说明

位	字段	类型	复位	说明
31	RESERVED	R	0h	保留
30	HYST_EN	R/W	0h	0：输入迟滞禁用 1：输入迟滞使能
29	IE	R/W	0h	0：输入禁用 1：输入使能 注意：如果 IO 配置为 AUX PORT_ID = 0x08，则使能将被忽略。
28-27	WU_CFG	R/W	0h	如果 DIO 配置为 GPIO 或非 AON 外设信号（PORT_ID 0x00 或 >0x08）： 00：无唤醒 01：无唤醒 10：如果该焊盘变为低电平，则从关断状态唤醒。 11：如果该焊盘变为高电平，则从关断状态唤醒。 如果 IO 配置为 AON 外设信号或 AUX PORT_ID 0x01-0x08，则该寄存器仅设置是否使能唤醒。 00、01：唤醒禁用 10、11：唤醒使能 极性通过 AON 寄存器进行控制。 注意：当 MSB 置位时，IOC 将停用 DIO 的输出使能。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-45. IOCFG15 寄存器字段说明（续）

位	字段	类型	复位	说明
26-24	IOMODE	R/W	0h	IO 模式 不适用于配置为 AON 外设信号和 AUX PORT_ID 0x01-0x08 的 IO AUX 有自己的 open_source/drain 配置。 0x2: 保留。未定义的行为。 0x3：保留。未定义的行为。 0h = NORMAL：正常输入/输出 1h = INV：反相输入/输出 4h = OPENDR： 开漏， 正常输入/输出 5h = OPENDR_INV：开漏 反相输入/输出 6h = OPENSRC：开源 正常输入/输出 7h = OPENSRC_INV：开源 反相输入/输出
23	IOEV_AON_PROG2_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG2 事件置为有效 1：输入边沿检测将 AON_PROG2 事件置为有效
22	IOEV_AON_PROG1_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG1 事件置为有效 1：输入边沿检测将 AON_PROG1 事件置为有效
21	IOEV_AON_PROG0_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG0 事件置为有效 1：输入边沿检测将 AON_PROG0 事件置为有效
20-19	RESERVED	R	0h	保留
18	EDGE_IRQ_EN	R/W	0h	0：无中断生成 1：使能该 IO 的中断生成（仅在 EDGE_DET 已使能时有效）
17-16	EDGE_DET	R/W	0h	使能该 IO 上的边沿检测事件生成 0h = NONE：无边沿检测 1h = 负边沿检测 2h = 正边沿检测 3h = 正负边沿检测
15	RESERVED	R	0h	保留
14-13	PULL_CTL	R/W	3h	上拉/下拉控制 1h = DWN：下拉 2h = UP：上拉 3h = DIS：无上拉/下拉
12	SLEW_RED	R/W	0h	0：标称压摆率 1：在输出驱动器中降低压摆率。
11-10	IOCURR	R/W	0h	选择该 IO 的 IO 电流模式。 0h = 2MA：低电流 (LC) 模式：IOSTR 设置为 AUTO 时最小值为 2mA 1h = 4MA：高电流 (HC) 模式：IOSTR 设置为 AUTO 时最小值为 4mA 2h = 4_8MA：扩展电流 (EC) 模式：当 IOSTR 设为 AUTO 时，双驱动强度 IO 的最小值为 8mA（标准 IO 的最小值为 4mA）

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-45. IOCFG15 寄存器字段说明（续）

位	字段	类型	复位	说明
9-8	IOSTR	R/W	0h	选择该 IO 的驱动强度控制源。 此设置控制低电流 (LC) 模式的驱动强度。可在 IOCURR 中选择更高的驱动强度 0h = 自动驱动强度，由 AON BATMON 根据电池电压控制。（VDDS 时最小 2mA） 1h = 最小驱动强度，由 AON_IOC:IOSTRMIN 控制（默认值为 3.3V 时最小 2mA） 2h = MED：中等驱动强度，由 AON_IOC:IOSTRMED 控制（默认值为 2.5V 时最小 2mA） 3h = 最大驱动强度，由 AON_IOC:IOSTRMAX 控制（默认值为 1.8V 时最小 2mA）
7	IOEV_RTC_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 RTC 事件置为有效 1：输入边沿检测将 RTC 事件置为有效
6	IOEV_MCU_WU_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 MCU_WU 事件置为有效 1：输入边沿检测将 MCU_WU 事件置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-45. IOCFG15 寄存器字段说明 （续）

位	字段	类型	复位	说明
5-0	PORT_ID	R/W	0h	选择 DIO15 的用途 注意：除明确要求更改 PORT_ID 值的情况外，不应写入此字段。 0h =通用 IO 7h = AON 32 KHz 时钟 (SCLK_LF) 8h = AUX IO 9h = SSI0_RX：SSI0 RX Ah = SSI0_TX：SSI0 TX Bh = SSI0_FSS：SSI0 FSS Ch = SSI0_CLK：SSI0 CLK Dh = I2C_MSSDA：I2C 数据 Eh = I2C_MSSCL：I2C 时钟 Fh = UART0_RX：UART0 RX 10h = UART0_TX：UART0 TX 11h = UART0_CTS：UART0 CTS 12h = UART0_RTS：UART0 RTS 13h = UART1_RX：UART1 RX 14h = UART1_TX：UART1 TX 15h = UART1_CTS：UART1 CTS 16h = UART1_RTS：UART1 RTS 17h = PORT_EVENT0：端口事件 0 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等）， 可用作通用 IO 事件 18h = PORT_EVENT1： 端口事件 1 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等）， 可用作通用 IO 事件 19h = PORT_EVENT2： 端口事件 2 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等）， 可用作通用 IO 事件 1Ah = PORT_EVENT3： 端口事件 3 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等）， 可用作通用 IO 事件 1Bh = PORT_EVENT4： 端口事件 4 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等）， 可用作通用 IO 事件 1Ch = PORT_EVENT5： 端口事件 5 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等）， 可用作通用 IO 事件 1Dh = PORT_EVENT6： 端口事件 6 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等）， 可用作通用 IO 事件 1Eh = PORT_EVENT7： 端口事件 7 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等）， 可用作通用 IO 事件 20h = CPU_SWV： CPU SWV 21h = SSI1_RX：SSI1 RX 22h = SSI1_TX：SSI1 TX 23h = SSI1_FSS：SSI1 FSS 24h = SSI1_CLK：SSI1 CLK 25h = I2S_AD0：I2S 数据 0 26h = I2S_AD1： I2S 数据 1 27h = I2S_WCLK： I2S WCLK 28h = I2S_BCLK：I2S BCLK 29h = I2S_MCLK：I2S MCLK 2Eh = 射频内核跟踪

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-45. IOCFG15 寄存器字段说明（续）

位	字段	类型	复位	说明
				2Fh = 射频内核数据输出 0 30h = 射频内核数据输出 1 31h = 射频内核数据输出 2 32h = 射频内核数据输出 3 33h = 射频内核数据输入 0 34h = 射频内核数据输入 1 35h = 射频内核 SMI 数据链路输出 36h = 射频内核 SMI 数据链路输入 37h = 射频内核 SMI 命令链路输出 38h = 射频内核 SMI 命令链路输入

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.3.17 IOCFG16 寄存器（偏移 = 40h）[复位 = 00086000h]

图 13-39 展示了 IOCFG16，表 13-46 对其进行了介绍。

返回到汇总表。

DIO16 的配置

图 13-39. IOCFG16 寄存器

31	30	29	28	27	26	25	24
RESERVED	HYST_EN	IE	WU_CFG		IOMODE		
R-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h		
23	22	21	20	19	18	17	16
IOEV_AON_PROG2_EN	IOEV_AON_PROG1_EN	IOEV_AON_PROG0_EN	RESERVED		EDGE_IRQ_EN	EDGE_DET	
R/W-0h	R/W-0h	R/W-0h	R-0h		R/W-0h	R/W-0h	
15	14	13	12	11	10	9	8
RESERVED	PULL_CTL		SLEW_RED	IOCURR		IOSTR	
R-0h	R/W-3h		R/W-0h	R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
IOEV_RTC_EN	IOEV_MCU_WU_EN	PORT_ID					
R/W-0h	R/W-0h	R/W-0h					

表 13-46. IOCFG16 寄存器字段说明

位	字段	类型	复位	说明
31	RESERVED	R	0h	保留
30	HYST_EN	R/W	0h	0：输入迟滞禁用 1：输入迟滞使能
29	IE	R/W	0h	0：输入禁用 1：输入使能 注意：如果 IO 配置为 AUX PORT_ID = 0x08，则使能将被忽略。
28-27	WU_CFG	R/W	0h	如果 DIO 配置为 GPIO 或非 AON 外设信号（PORT_ID 0x00 或 >0x08）： 00：无唤醒 01：无唤醒 10：如果该焊盘变为低电平，则从关断状态唤醒。 11：如果该焊盘变为高电平，则从关断状态唤醒。 如果 IO 配置为 AON 外设信号或 AUX PORT_ID 0x01-0x08，则该寄存器仅设置是否使能唤醒。 00、01：唤醒禁用 10、11：唤醒使能 极性通过 AON 寄存器进行控制。 注意：当 MSB 置位时，IOC 将停用 DIO 的输出使能。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-46. IOCFG16 寄存器字段说明（续）

位	字段	类型	复位	说明
26-24	IOMODE	R/W	0h	IO 模式 不适用于配置为 AON 外设信号和 AUX PORT_ID 0x01-0x08 的 IO AUX 有自己的 open_source/drain 配置。 0x2: 保留。未定义的行为。 0x3：保留。未定义的行为。 0h = NORMAL：正常输入/输出 1h = INV：反相输入/输出 4h = OPENDR： 开漏， 正常输入/输出 5h = OPENDR_INV：开漏 反相输入/输出 6h = OPENSRC：开源 正常输入/输出 7h = OPENSRC_INV：开源 反相输入/输出
23	IOEV_AON_PROG2_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG2 事件置为有效 1：输入边沿检测将 AON_PROG2 事件置为有效
22	IOEV_AON_PROG1_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG1 事件置为有效 1：输入边沿检测将 AON_PROG1 事件置为有效
21	IOEV_AON_PROG0_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG0 事件置为有效 1：输入边沿检测将 AON_PROG0 事件置为有效
20-19	RESERVED	R	0h	保留
18	EDGE_IRQ_EN	R/W	0h	0：无中断生成 1：使能该 IO 的中断生成（仅在 EDGE_DET 已使能时有效）
17-16	EDGE_DET	R/W	0h	使能该 IO 上的边沿检测事件生成 0h = NONE：无边沿检测 1h = 负边沿检测 2h = 正边沿检测 3h = 正负边沿检测
15	RESERVED	R	0h	保留
14-13	PULL_CTL	R/W	3h	上拉/下拉控制 1h = DWN：下拉 2h = UP：上拉 3h = DIS：无上拉/下拉
12	SLEW_RED	R/W	0h	0：标称压摆率 1：在输出驱动器中降低压摆率。
11-10	IOCURR	R/W	0h	选择该 IO 的 IO 电流模式。 0h = 2MA：低电流 (LC) 模式：IOSTR 设置为 AUTO 时最小值为 2mA 1h = 4MA：高电流 (HC) 模式：IOSTR 设置为 AUTO 时最小值为 4mA 2h = 4_8MA：扩展电流 (EC) 模式：当 IOSTR 设为 AUTO 时，双驱动强度 IO 的最小值为 8mA（标准 IO 的最小值为 4mA）

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-46. IOCFG16 寄存器字段说明（续）

位	字段	类型	复位	说明
9-8	IOSTR	R/W	0h	选择该 IO 的驱动强度控制源。 此设置控制低电流 (LC) 模式的驱动强度。可在 IOCURR 中选择更高的驱动强度 0h = 自动驱动强度，由 AON BATMON 根据电池电压控制。（VDDS 时最小 2mA） 1h = 最小驱动强度，由 AON_IOC:IOSTRMIN 控制（默认值为 3.3V 时最小 2mA） 2h = MED：中等驱动强度，由 AON_IOC:IOSTRMED 控制（默认值为 2.5V 时最小 2mA） 3h = 最大驱动强度，由 AON_IOC:IOSTRMAX 控制（默认值为 1.8V 时最小 2mA）
7	IOEV_RTC_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 RTC 事件置为有效 1：输入边沿检测将 RTC 事件置为有效
6	IOEV_MCU_WU_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 MCU_WU 事件置为有效 1：输入边沿检测将 MCU_WU 事件置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-46. IOCFG16 寄存器字段说明（续）

位	字段	类型	复位	说明
5-0	PORT_ID	R/W	0h	选择 DIO16 的用途 注意：除明确要求更改 PORT_ID 值的情况外，不应写入此字段。 0h =通用 IO 7h = AON 32 KHz 时钟 (SCLK_LF) 8h = AUX IO 9h = SSI0_RX：SSI0 RX Ah = SSI0_TX：SSI0 TX Bh = SSI0_FSS：SSI0 FSS Ch = SSI0_CLK：SSI0 CLK Dh = I2C_MSSDA：I2C 数据 Eh = I2C_MSSCL：I2C 时钟 Fh = UART0_RX：UART0 RX 10h = UART0_TX：UART0 TX 11h = UART0_CTS：UART0 CTS 12h = UART0_RTS：UART0 RTS 13h = UART1_RX：UART1 RX 14h = UART1_TX：UART1 TX 15h = UART1_CTS：UART1 CTS 16h = UART1_RTS：UART1 RTS 17h = PORT_EVENT0：端口事件 0 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 18h = PORT_EVENT1： 端口事件 1 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 19h = PORT_EVENT2： 端口事件 2 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ah = PORT_EVENT3： 端口事件 3 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Bh = PORT_EVENT4： 端口事件 4 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ch = PORT_EVENT5： 端口事件 5 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Dh = PORT_EVENT6： 端口事件 6 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Eh = PORT_EVENT7： 端口事件 7 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 20h = CPU_SWV： CPU SWV 21h = SSI1_RX：SSI1 RX 22h = SSI1_TX：SSI1 TX 23h = SSI1_FSS：SSI1 FSS 24h = SSI1_CLK：SSI1 CLK 25h = I2S_AD0：I2S 数据 0 26h = I2S_AD1： I2S 数据 1 27h = I2S_WCLK： I2S WCLK 28h = I2S_BCLK：I2S BCLK 29h = I2S_MCLK：I2S MCLK 2Eh = 射频内核跟踪

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-46. IOCFG16 寄存器字段说明（续）

位	字段	类型	复位	说明
				2Fh = 射频内核数据输出 0 30h = 射频内核数据输出 1 31h = 射频内核数据输出 2 32h = 射频内核数据输出 3 33h = 射频内核数据输入 0 34h = 射频内核数据输入 1 35h = 射频内核 SMI 数据链路输出 36h = 射频内核 SMI 数据链路输入 37h = 射频内核 SMI 命令链路输出 38h = 射频内核 SMI 命令链路输入

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.3.18 IOCFG17 寄存器（偏移 = 44h）[复位 = 00106000h]

图 13-40 展示了 IOCFG17，表 13-47 对其进行了介绍。

返回到汇总表。

DIO17 的配置

图 13-40. IOCFG17 寄存器

31	30	29	28	27	26	25	24
RESERVED	HYST_EN	IE	WU_CFG		IOMODE		
R-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h		
23	22	21	20	19	18	17	16
IOEV_AON_PROG2_EN	IOEV_AON_PROG1_EN	IOEV_AON_PROG0_EN	RESERVED		EDGE_IRQ_EN	EDGE_DET	
R/W-0h	R/W-0h	R/W-0h	R-0h		R/W-0h	R/W-0h	
15	14	13	12	11	10	9	8
RESERVED	PULL_CTL		SLEW_RED	IOCURR		IOSTR	
R-0h	R/W-3h		R/W-0h	R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
IOEV_RTC_EN	IOEV_MCU_WU_EN	PORT_ID					
R/W-0h	R/W-0h	R/W-0h					

表 13-47. IOCFG17 寄存器字段说明

位	字段	类型	复位	说明
31	RESERVED	R	0h	保留
30	HYST_EN	R/W	0h	0：输入迟滞禁用 1：输入迟滞使能
29	IE	R/W	0h	0：输入禁用 1：输入使能 注意：如果 IO 配置为 AUX PORT_ID = 0x08，则使能将被忽略。
28-27	WU_CFG	R/W	0h	如果 DIO 配置为 GPIO 或非 AON 外设信号（PORT_ID 0x00 或 >0x08）： 00：无唤醒 01：无唤醒 10：如果该焊盘变为低电平，则从关断状态唤醒。 11：如果该焊盘变为高电平，则从关断状态唤醒。 如果 IO 配置为 AON 外设信号或 AUX PORT_ID 0x01-0x08，则该寄存器仅设置是否使能唤醒。 00、01：唤醒禁用 10、11：唤醒使能 极性通过 AON 寄存器进行控制。 注意：当 MSB 置位时，IOC 将停用 DIO 的输出使能。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-47. IOCFG17 寄存器字段说明（续）

位	字段	类型	复位	说明
26-24	IOMODE	R/W	0h	IO 模式 不适用于配置为 AON 外设信号和 AUX PORT_ID 0x01-0x08 的 IO AUX 有自己的 open_source/drain 配置。 0x2: 保留。未定义的行为。 0x3：保留。未定义的行为。 0h = NORMAL：正常输入/输出 1h = INV：反相输入/输出 4h = OPENDR： 开漏， 正常输入/输出 5h = OPENDR_INV：开漏 反相输入/输出 6h = OPENSRC：开源 正常输入/输出 7h = OPENSRC_INV：开源 反相输入/输出
23	IOEV_AON_PROG2_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG2 事件置为有效 1：输入边沿检测将 AON_PROG2 事件置为有效
22	IOEV_AON_PROG1_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG1 事件置为有效 1：输入边沿检测将 AON_PROG1 事件置为有效
21	IOEV_AON_PROG0_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG0 事件置为有效 1：输入边沿检测将 AON_PROG0 事件置为有效
20-19	RESERVED	R	0h	保留
18	EDGE_IRQ_EN	R/W	0h	0：无中断生成 1：使能该 IO 的中断生成（仅在 EDGE_DET 已使能时有效）
17-16	EDGE_DET	R/W	0h	使能该 IO 上的边沿检测事件生成 0h = NONE：无边沿检测 1h = 负边沿检测 2h = 正边沿检测 3h = 正负边沿检测
15	RESERVED	R	0h	保留
14-13	PULL_CTL	R/W	3h	上拉/下拉控制 1h = DWN：下拉 2h = UP：上拉 3h = DIS：无上拉/下拉
12	SLEW_RED	R/W	0h	0：标称压摆率 1：在输出驱动器中降低压摆率。
11-10	IOCURR	R/W	0h	选择该 IO 的 IO 电流模式。 0h = 2MA：低电流 (LC) 模式：IOSTR 设置为 AUTO 时最小值为 2mA 1h = 4MA：高电流 (HC) 模式：IOSTR 设置为 AUTO 时最小值为 4mA 2h = 4_8MA：扩展电流 (EC) 模式：当 IOSTR 设为 AUTO 时，双驱动强度 IO 的最小值为 8mA（标准 IO 的最小值为 4mA）

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-47. IOCFG17 寄存器字段说明（续）

位	字段	类型	复位	说明
9-8	IOSTR	R/W	0h	选择该 IO 的驱动强度控制源。 此设置控制低电流 (LC) 模式的驱动强度。可在 IOCURR 中选择更高的驱动强度 0h = 自动驱动强度，由 AON BATMON 根据电池电压控制。（VDDS 时最小 2mA） 1h = 最小驱动强度，由 AON_IOC:IOSTRMIN 控制（默认值为 3.3V 时最小 2mA） 2h = MED：中等驱动强度，由 AON_IOC:IOSTRMED 控制（默认值为 2.5V 时最小 2mA） 3h = 最大驱动强度，由 AON_IOC:IOSTRMAX 控制（默认值为 1.8V 时最小 2mA）
7	IOEV_RTC_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 RTC 事件置为有效 1：输入边沿检测将 RTC 事件置为有效
6	IOEV_MCU_WU_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 MCU_WU 事件置为有效 1：输入边沿检测将 MCU_WU 事件置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-47. IOCFG17 寄存器字段说明（续）

位	字段	类型	复位	说明
5-0	PORT_ID	R/W	0h	选择 DIO17 的用途 注意：除明确要求更改 PORT_ID 值的情况外，不应写入此字段。 0h =通用 IO 7h = AON 32 KHz 时钟 (SCLK_LF) 8h = AUX IO 9h = SSI0_RX：SSI0 RX Ah = SSI0_TX：SSI0 TX Bh = SSI0_FSS：SSI0 FSS Ch = SSI0_CLK：SSI0 CLK Dh = I2C_MSSDA：I2C 数据 Eh = I2C_MSSCL：I2C 时钟 Fh = UART0_RX：UART0 RX 10h = UART0_TX：UART0 TX 11h = UART0_CTS：UART0 CTS 12h = UART0_RTS：UART0 RTS 13h = UART1_RX：UART1 RX 14h = UART1_TX：UART1 TX 15h = UART1_CTS：UART1 CTS 16h = UART1_RTS：UART1 RTS 17h = PORT_EVENT0：端口事件 0 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 18h = PORT_EVENT1： 端口事件 1 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 19h = PORT_EVENT2： 端口事件 2 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ah = PORT_EVENT3： 端口事件 3 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Bh = PORT_EVENT4： 端口事件 4 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ch = PORT_EVENT5： 端口事件 5 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Dh = PORT_EVENT6： 端口事件 6 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Eh = PORT_EVENT7： 端口事件 7 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 20h = CPU_SWV： CPU SWV 21h = SSI1_RX：SSI1 RX 22h = SSI1_TX：SSI1 TX 23h = SSI1_FSS：SSI1 FSS 24h = SSI1_CLK：SSI1 CLK 25h = I2S_AD0：I2S 数据 0 26h = I2S_AD1： I2S 数据 1 27h = I2S_WCLK： I2S WCLK 28h = I2S_BCLK：I2S BCLK 29h = I2S_MCLK：I2S MCLK 2Eh = 射频内核跟踪

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-47. IOCFG17 寄存器字段说明（续）

位	字段	类型	复位	说明
				2Fh = 射频内核数据输出 0 30h = 射频内核数据输出 1 31h = 射频内核数据输出 2 32h = 射频内核数据输出 3 33h = 射频内核数据输入 0 34h = 射频内核数据输入 1 35h = 射频内核 SMI 数据链路输出 36h = 射频内核 SMI 数据链路输入 37h = 射频内核 SMI 命令链路输出 38h = 射频内核 SMI 命令链路输入

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.3.19 IOCFG18 寄存器（偏移 = 48h）[复位 = 00006000h]

图 13-41 展示了 IOCFG18，表 13-48 对其进行了介绍。

返回到汇总表。

DIO18 的配置

图 13-41. IOCFG18 寄存器

31	30	29	28	27	26	25	24
RESERVED	HYST_EN	IE	WU_CFG		IOMODE		
R-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h		
23	22	21	20	19	18	17	16
IOEV_AON_PROG2_EN	IOEV_AON_PROG1_EN	IOEV_AON_PROG0_EN	RESERVED		EDGE_IRQ_EN	EDGE_DET	
R/W-0h	R/W-0h	R/W-0h	R-0h		R/W-0h	R/W-0h	
15	14	13	12	11	10	9	8
RESERVED	PULL_CTL		SLEW_RED	IOCURR		IOSTR	
R-0h	R/W-3h		R/W-0h	R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
IOEV_RTC_EN	IOEV_MCU_WU_EN	PORT_ID					
R/W-0h	R/W-0h	R/W-0h					

表 13-48. IOCFG18 寄存器字段说明

位	字段	类型	复位	说明
31	RESERVED	R	0h	保留
30	HYST_EN	R/W	0h	0：输入迟滞禁用 1：输入迟滞使能
29	IE	R/W	0h	0：输入禁用 1：输入使能 注意：如果 IO 配置为 AUX PORT_ID = 0x08，则使能将被忽略。
28-27	WU_CFG	R/W	0h	如果 DIO 配置为 GPIO 或非 AON 外设信号（PORT_ID 0x00 或 >0x08）： 00：无唤醒 01：无唤醒 10：如果该焊盘变为低电平，则从关断状态唤醒。 11：如果该焊盘变为高电平，则从关断状态唤醒。 如果 IO 配置为 AON 外设信号或 AUX PORT_ID 0x01-0x08，则该寄存器仅设置是否使能唤醒。 00、01：唤醒禁用 10、11：唤醒使能 极性通过 AON 寄存器进行控制。 注意：当 MSB 置位时，IOC 将停用 DIO 的输出使能。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-48. IOCFG18 寄存器字段说明（续）

位	字段	类型	复位	说明
26-24	IOMODE	R/W	0h	IO 模式 不适用于配置为 AON 外设信号和 AUX PORT_ID 0x01-0x08 的 IO AUX 有自己的 open_source/drain 配置。 0x2: 保留。未定义的行为。 0x3：保留。未定义的行为。 0h = NORMAL：正常输入/输出 1h = INV：反相输入/输出 4h = OPENDR： 开漏， 正常输入/输出 5h = OPENDR_INV：开漏 反相输入/输出 6h = OPENSRC：开源 正常输入/输出 7h = OPENSRC_INV：开源 反相输入/输出
23	IOEV_AON_PROG2_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG2 事件置为有效 1：输入边沿检测将 AON_PROG2 事件置为有效
22	IOEV_AON_PROG1_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG1 事件置为有效 1：输入边沿检测将 AON_PROG1 事件置为有效
21	IOEV_AON_PROG0_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG0 事件置为有效 1：输入边沿检测将 AON_PROG0 事件置为有效
20-19	RESERVED	R	0h	保留
18	EDGE_IRQ_EN	R/W	0h	0：无中断生成 1：使能该 IO 的中断生成（仅在 EDGE_DET 已使能时有效）
17-16	EDGE_DET	R/W	0h	使能该 IO 上的边沿检测事件生成 0h = NONE：无边沿检测 1h = 负边沿检测 2h = 正边沿检测 3h = 正负边沿检测
15	RESERVED	R	0h	保留
14-13	PULL_CTL	R/W	3h	上拉/下拉控制 1h = DWN：下拉 2h = UP：上拉 3h = DIS：无上拉/下拉
12	SLEW_RED	R/W	0h	0：标称压摆率 1：在输出驱动器中降低压摆率。
11-10	IOCURR	R/W	0h	选择该 IO 的 IO 电流模式。 0h = 2MA：低电流 (LC) 模式：IOSTR 设置为 AUTO 时最小值为 2mA 1h = 4MA：高电流 (HC) 模式：IOSTR 设置为 AUTO 时最小值为 4mA 2h = 4_8MA：扩展电流 (EC) 模式：当 IOSTR 设为 AUTO 时，双驱动强度 IO 的最小值为 8mA（标准 IO 的最小值为 4mA）

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-48. IOCFG18 寄存器字段说明（续）

位	字段	类型	复位	说明
9-8	IOSTR	R/W	0h	选择该 IO 的驱动强度控制源。 此设置控制低电流 (LC) 模式的驱动强度。可在 IOCURR 中选择更高的驱动强度 0h = 自动驱动强度，由 AON BATMON 根据电池电压控制。（VDDS 时最小 2mA） 1h = 最小驱动强度，由 AON_IOC:IOSTRMIN 控制（默认值为 3.3V 时最小 2mA） 2h = MED：中等驱动强度，由 AON_IOC:IOSTRMED 控制（默认值为 2.5V 时最小 2mA） 3h = 最大驱动强度，由 AON_IOC:IOSTRMAX 控制（默认值为 1.8V 时最小 2mA）
7	IOEV_RTC_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 RTC 事件置为有效 1：输入边沿检测将 RTC 事件置为有效
6	IOEV_MCU_WU_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 MCU_WU 事件置为有效 1：输入边沿检测将 MCU_WU 事件置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-48. IOCFG18 寄存器字段说明（续）

位	字段	类型	复位	说明
5-0	PORT_ID	R/W	0h	选择 DIO18 的用途 注意：除明确要求更改 PORT_ID 值的情况外，不应写入此字段。 0h =通用 IO 7h = AON 32 KHz 时钟 (SCLK_LF) 8h = AUX IO 9h = SSI0_RX：SSI0 RX Ah = SSI0_TX：SSI0 TX Bh = SSI0_FSS：SSI0 FSS Ch = SSI0_CLK：SSI0 CLK Dh = I2C_MSSDA：I2C 数据 Eh = I2C_MSSCL：I2C 时钟 Fh = UART0_RX：UART0 RX 10h = UART0_TX：UART0 TX 11h = UART0_CTS：UART0 CTS 12h = UART0_RTS：UART0 RTS 13h = UART1_RX：UART1 RX 14h = UART1_TX：UART1 TX 15h = UART1_CTS：UART1 CTS 16h = UART1_RTS：UART1 RTS 17h = PORT_EVENT0：端口事件 0 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 18h = PORT_EVENT1： 端口事件 1 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 19h = PORT_EVENT2： 端口事件 2 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ah = PORT_EVENT3： 端口事件 3 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Bh = PORT_EVENT4： 端口事件 4 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ch = PORT_EVENT5： 端口事件 5 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Dh = PORT_EVENT6： 端口事件 6 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Eh = PORT_EVENT7： 端口事件 7 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 20h = CPU_SWV： CPU SWV 21h = SSI1_RX：SSI1 RX 22h = SSI1_TX：SSI1 TX 23h = SSI1_FSS：SSI1 FSS 24h = SSI1_CLK：SSI1 CLK 25h = I2S_AD0：I2S 数据 0 26h = I2S_AD1： I2S 数据 1 27h = I2S_WCLK： I2S WCLK 28h = I2S_BCLK：I2S BCLK 29h = I2S_MCLK：I2S MCLK 2Eh = 射频内核跟踪

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-48. IOCFG18 寄存器字段说明（续）

位	字段	类型	复位	说明
				2Fh = 射频内核数据输出 0 30h = 射频内核数据输出 1 31h = 射频内核数据输出 2 32h = 射频内核数据输出 3 33h = 射频内核数据输入 0 34h = 射频内核数据输入 1 35h = 射频内核 SMI 数据链路输出 36h = 射频内核 SMI 数据链路输入 37h = 射频内核 SMI 命令链路输出 38h = 射频内核 SMI 命令链路输入

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.3.20 IOCFG19 寄存器（偏移 = 4Ch）[复位 = 00006000h]

图 13-42 展示了 IOCFG19，表 13-49 对其进行了介绍。

返回到汇总表。

DIO19 的配置

图 13-42. IOCFG19 寄存器

31	30	29	28	27	26	25	24
RESERVED	HYST_EN	IE	WU_CFG		IOMODE		
R-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h		
23	22	21	20	19	18	17	16
IOEV_AON_PROG2_EN	IOEV_AON_PROG1_EN	IOEV_AON_PROG0_EN	RESERVED		EDGE_IRQ_EN	EDGE_DET	
R/W-0h	R/W-0h	R/W-0h	R-0h		R/W-0h	R/W-0h	
15	14	13	12	11	10	9	8
RESERVED	PULL_CTL		SLEW_RED	IOCURR		IOSTR	
R-0h	R/W-3h		R/W-0h	R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
IOEV_RTC_EN	IOEV_MCU_WU_EN	PORT_ID					
R/W-0h	R/W-0h	R/W-0h					

表 13-49. IOCFG19 寄存器字段说明

位	字段	类型	复位	说明
31	RESERVED	R	0h	保留
30	HYST_EN	R/W	0h	0：输入迟滞禁用 1：输入迟滞使能
29	IE	R/W	0h	0：输入禁用 1：输入使能 注意：如果 IO 配置为 AUX PORT_ID = 0x08，则使能将被忽略。
28-27	WU_CFG	R/W	0h	如果 DIO 配置为 GPIO 或非 AON 外设信号（PORT_ID 0x00 或 >0x08）： 00：无唤醒 01：无唤醒 10：如果该焊盘变为低电平，则从关断状态唤醒。 11：如果该焊盘变为高电平，则从关断状态唤醒。 如果 IO 配置为 AON 外设信号或 AUX PORT_ID 0x01-0x08，则该寄存器仅设置是否使能唤醒。 00、01：唤醒禁用 10、11：唤醒使能 极性通过 AON 寄存器进行控制。 注意：当 MSB 置位时，IOC 将停用 DIO 的输出使能。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-49. IOCFG19 寄存器字段说明（续）

位	字段	类型	复位	说明
26-24	IOMODE	R/W	0h	IO 模式 不适用于配置为 AON 外设信号和 AUX PORT_ID 0x01-0x08 的 IO AUX 有自己的 open_source/drain 配置。 0x2: 保留。未定义的行为。 0x3：保留。未定义的行为。 0h = NORMAL：正常输入/输出 1h = INV：反相输入/输出 4h = OPENDR： 开漏， 正常输入/输出 5h = OPENDR_INV：开漏 反相输入/输出 6h = OPENSRC：开源 正常输入/输出 7h = OPENSRC_INV：开源 反相输入/输出
23	IOEV_AON_PROG2_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG2 事件置为有效 1：输入边沿检测将 AON_PROG2 事件置为有效
22	IOEV_AON_PROG1_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG1 事件置为有效 1：输入边沿检测将 AON_PROG1 事件置为有效
21	IOEV_AON_PROG0_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG0 事件置为有效 1：输入边沿检测将 AON_PROG0 事件置为有效
20-19	RESERVED	R	0h	保留
18	EDGE_IRQ_EN	R/W	0h	0：无中断生成 1：使能该 IO 的中断生成（仅在 EDGE_DET 已使能时有效）
17-16	EDGE_DET	R/W	0h	使能该 IO 上的边沿检测事件生成 0h = NONE：无边沿检测 1h = 负边沿检测 2h = 正边沿检测 3h = 正负边沿检测
15	RESERVED	R	0h	保留
14-13	PULL_CTL	R/W	3h	上拉/下拉控制 1h = DWN：下拉 2h = UP：上拉 3h = DIS：无上拉/下拉
12	SLEW_RED	R/W	0h	0：标称压摆率 1：在输出驱动器中降低压摆率。
11-10	IOCURR	R/W	0h	选择该 IO 的 IO 电流模式。 0h = 2MA：低电流 (LC) 模式：IOSTR 设置为 AUTO 时最小值为 2mA 1h = 4MA：高电流 (HC) 模式：IOSTR 设置为 AUTO 时最小值为 4mA 2h = 4_8MA：扩展电流 (EC) 模式：当 IOSTR 设为 AUTO 时，双驱动强度 IO 的最小值为 8mA（标准 IO 的最小值为 4mA）

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-49. IOCFG19 寄存器字段说明（续）

位	字段	类型	复位	说明
9-8	IOSTR	R/W	0h	选择该 IO 的驱动强度控制源。 此设置控制低电流 (LC) 模式的驱动强度。可在 IOCURR 中选择更高的驱动强度 0h = 自动驱动强度，由 AON BATMON 根据电池电压控制。（VDDS 时最小 2mA） 1h = 最小驱动强度，由 AON_IOC:IOSTRMIN 控制（默认值为 3.3V 时最小 2mA） 2h = MED：中等驱动强度，由 AON_IOC:IOSTRMED 控制（默认值为 2.5V 时最小 2mA） 3h = 最大驱动强度，由 AON_IOC:IOSTRMAX 控制（默认值为 1.8V 时最小 2mA）
7	IOEV_RTC_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 RTC 事件置为有效 1：输入边沿检测将 RTC 事件置为有效
6	IOEV_MCU_WU_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 MCU_WU 事件置为有效 1：输入边沿检测将 MCU_WU 事件置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-49. IOCFG19 寄存器字段说明 （续）

位	字段	类型	复位	说明
5-0	PORT_ID	R/W	0h	选择 DIO19 的用途 注意：除明确要求更改 PORT_ID 值的情况外，不应写入此字段。 0h =通用 IO 7h = AON 32 KHz 时钟 (SCLK_LF) 8h = AUX IO 9h = SSI0_RX：SSI0 RX Ah = SSI0_TX：SSI0 TX Bh = SSI0_FSS：SSI0 FSS Ch = SSI0_CLK：SSI0 CLK Dh = I2C_MSSDA：I2C 数据 Eh = I2C_MSSCL：I2C 时钟 Fh = UART0_RX：UART0 RX 10h = UART0_TX：UART0 TX 11h = UART0_CTS：UART0 CTS 12h = UART0_RTS：UART0 RTS 13h = UART1_RX：UART1 RX 14h = UART1_TX：UART1 TX 15h = UART1_CTS：UART1 CTS 16h = UART1_RTS：UART1 RTS 17h = PORT_EVENT0：端口事件 0 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 18h = PORT_EVENT1： 端口事件 1 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 19h = PORT_EVENT2： 端口事件 2 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ah = PORT_EVENT3： 端口事件 3 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Bh = PORT_EVENT4： 端口事件 4 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ch = PORT_EVENT5： 端口事件 5 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Dh = PORT_EVENT6： 端口事件 6 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Eh = PORT_EVENT7： 端口事件 7 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 20h = CPU_SWV： CPU SWV 21h = SSI1_RX：SSI1 RX 22h = SSI1_TX：SSI1 TX 23h = SSI1_FSS：SSI1 FSS 24h = SSI1_CLK：SSI1 CLK 25h = I2S_AD0：I2S 数据 0 26h = I2S_AD1： I2S 数据 1 27h = I2S_WCLK： I2S WCLK 28h = I2S_BCLK：I2S BCLK 29h = I2S_MCLK：I2S MCLK 2Eh = 射频内核跟踪

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-49. IOCFG19 寄存器字段说明（续）

位	字段	类型	复位	说明
				2Fh = 射频内核数据输出 0 30h = 射频内核数据输出 1 31h = 射频内核数据输出 2 32h = 射频内核数据输出 3 33h = 射频内核数据输入 0 34h = 射频内核数据输入 1 35h = 射频内核 SMI 数据链路输出 36h = 射频内核 SMI 数据链路输入 37h = 射频内核 SMI 命令链路输出 38h = 射频内核 SMI 命令链路输入

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.3.21 IOCFG20 寄存器（偏移 = 50h）[复位 = 00006000h]

图 13-43 展示了 IOCFG20，表 13-50 对其进行了介绍。

返回到汇总表。

DIO20 的配置

图 13-43. IOCFG20 寄存器

31	30	29	28	27	26	25	24
RESERVED	HYST_EN	IE	WU_CFG		IOMODE		
R-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h		
23	22	21	20	19	18	17	16
IOEV_AON_PROG2_EN	IOEV_AON_PROG1_EN	IOEV_AON_PROG0_EN	RESERVED		EDGE_IRQ_EN	EDGE_DET	
R/W-0h	R/W-0h	R/W-0h	R-0h		R/W-0h	R/W-0h	
15	14	13	12	11	10	9	8
RESERVED	PULL_CTL		SLEW_RED	IOCURR		IOSTR	
R-0h	R/W-3h		R/W-0h	R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
IOEV_RTC_EN	IOEV_MCU_WU_EN	PORT_ID					
R/W-0h	R/W-0h	R/W-0h					

表 13-50. IOCFG20 寄存器字段说明

位	字段	类型	复位	说明
31	RESERVED	R	0h	保留
30	HYST_EN	R/W	0h	0：输入迟滞禁用 1：输入迟滞使能
29	IE	R/W	0h	0：输入禁用 1：输入使能 注意：如果 IO 配置为 AUX PORT_ID = 0x08，则使能将被忽略。
28-27	WU_CFG	R/W	0h	如果 DIO 配置为 GPIO 或非 AON 外设信号（PORT_ID 0x00 或 >0x08）： 00：无唤醒 01：无唤醒 10：如果该焊盘变为低电平，则从关断状态唤醒。 11：如果该焊盘变为高电平，则从关断状态唤醒。 如果 IO 配置为 AON 外设信号或 AUX PORT_ID 0x01-0x08，则该寄存器仅设置是否使能唤醒。 00、01：唤醒禁用 10、11：唤醒使能 极性通过 AON 寄存器进行控制。 注意：当 MSB 置位时，IOC 将停用 DIO 的输出使能。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-50. IOCFG20 寄存器字段说明（续）

位	字段	类型	复位	说明
26-24	IOMODE	R/W	0h	IO 模式 不适用于配置为 AON 外设信号和 AUX PORT_ID 0x01-0x08 的 IO AUX 有自己的 open_source/drain 配置。 0x2: 保留。未定义的行为。 0x3：保留。未定义的行为。 0h = NORMAL：正常输入/输出 1h = INV：反相输入/输出 4h = OPENDR： 开漏， 正常输入/输出 5h = OPENDR_INV：开漏 反相输入/输出 6h = OPENSRC：开源 正常输入/输出 7h = OPENSRC_INV：开源 反相输入/输出
23	IOEV_AON_PROG2_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG2 事件置为有效 1：输入边沿检测将 AON_PROG2 事件置为有效
22	IOEV_AON_PROG1_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG1 事件置为有效 1：输入边沿检测将 AON_PROG1 事件置为有效
21	IOEV_AON_PROG0_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG0 事件置为有效 1：输入边沿检测将 AON_PROG0 事件置为有效
20-19	RESERVED	R	0h	保留
18	EDGE_IRQ_EN	R/W	0h	0：无中断生成 1：使能该 IO 的中断生成（仅在 EDGE_DET 已使能时有效）
17-16	EDGE_DET	R/W	0h	使能该 IO 上的边沿检测事件生成 0h = NONE：无边沿检测 1h = 负边沿检测 2h = 正边沿检测 3h = 正负边沿检测
15	RESERVED	R	0h	保留
14-13	PULL_CTL	R/W	3h	上拉/下拉控制 1h = DWN：下拉 2h = UP：上拉 3h = DIS：无上拉/下拉
12	SLEW_RED	R/W	0h	0：标称压摆率 1：在输出驱动器中降低压摆率。
11-10	IOCURR	R/W	0h	选择该 IO 的 IO 电流模式。 0h = 2MA：低电流 (LC) 模式：IOSTR 设置为 AUTO 时最小值为 2mA 1h = 4MA：高电流 (HC) 模式：IOSTR 设置为 AUTO 时最小值为 4mA 2h = 4_8MA：扩展电流 (EC) 模式：当 IOSTR 设为 AUTO 时，双驱动强度 IO 的最小值为 8mA（标准 IO 的最小值为 4mA）

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-50. IOCFG20 寄存器字段说明（续）

位	字段	类型	复位	说明
9-8	IOSTR	R/W	0h	选择该 IO 的驱动强度控制源。 此设置控制低电流 (LC) 模式的驱动强度。可在 IOCURR 中选择更高的驱动强度 0h = 自动驱动强度，由 AON BATMON 根据电池电压控制。（VDDS 时最小 2mA） 1h = 最小驱动强度，由 AON_IOC:IOSTRMIN 控制（默认值为 3.3V 时最小 2mA） 2h = MED：中等驱动强度，由 AON_IOC:IOSTRMED 控制（默认值为 2.5V 时最小 2mA） 3h = 最大驱动强度，由 AON_IOC:IOSTRMAX 控制（默认值为 1.8V 时最小 2mA）
7	IOEV_RTC_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 RTC 事件置为有效 1：输入边沿检测将 RTC 事件置为有效
6	IOEV_MCU_WU_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 MCU_WU 事件置为有效 1：输入边沿检测将 MCU_WU 事件置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-50. IOCFG20 寄存器字段说明（续）

位	字段	类型	复位	说明
5-0	PORT_ID	R/W	0h	选择 DIO20 的用途 注意：除明确要求更改 PORT_ID 值的情况外，不应写入此字段。 0h =通用 IO 7h = AON 32 KHz 时钟 (SCLK_LF) 8h = AUX IO 9h = SSI0_RX：SSI0 RX Ah = SSI0_TX：SSI0 TX Bh = SSI0_FSS：SSI0 FSS Ch = SSI0_CLK：SSI0 CLK Dh = I2C_MSSDA：I2C 数据 Eh = I2C_MSSCL：I2C 时钟 Fh = UART0_RX：UART0 RX 10h = UART0_TX：UART0 TX 11h = UART0_CTS：UART0 CTS 12h = UART0_RTS：UART0 RTS 13h = UART1_RX：UART1 RX 14h = UART1_TX：UART1 TX 15h = UART1_CTS：UART1 CTS 16h = UART1_RTS：UART1 RTS 17h = PORT_EVENT0：端口事件 0 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 18h = PORT_EVENT1： 端口事件 1 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 19h = PORT_EVENT2： 端口事件 2 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ah = PORT_EVENT3： 端口事件 3 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Bh = PORT_EVENT4： 端口事件 4 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ch = PORT_EVENT5： 端口事件 5 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Dh = PORT_EVENT6： 端口事件 6 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Eh = PORT_EVENT7： 端口事件 7 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 20h = CPU_SWV： CPU SWV 21h = SSI1_RX：SSI1 RX 22h = SSI1_TX：SSI1 TX 23h = SSI1_FSS：SSI1 FSS 24h = SSI1_CLK：SSI1 CLK 25h = I2S_AD0：I2S 数据 0 26h = I2S_AD1： I2S 数据 1 27h = I2S_WCLK： I2S WCLK 28h = I2S_BCLK：I2S BCLK 29h = I2S_MCLK：I2S MCLK 2Eh = 射频内核跟踪

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-50. IOCFG20 寄存器字段说明（续）

位	字段	类型	复位	说明
				2Fh = 射频内核数据输出 0 30h = 射频内核数据输出 1 31h = 射频内核数据输出 2 32h = 射频内核数据输出 3 33h = 射频内核数据输入 0 34h = 射频内核数据输入 1 35h = 射频内核 SMI 数据链路输出 36h = 射频内核 SMI 数据链路输入 37h = 射频内核 SMI 命令链路输出 38h = 射频内核 SMI 命令链路输入

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.3.22 IOCFG21 寄存器（偏移 = 54h）[复位 = 00006000h]

图 13-44 展示了 IOCFG21，表 13-51 对其进行了介绍。

返回到汇总表。

DIO21 的配置

图 13-44. IOCFG21 寄存器

31	30	29	28	27	26	25	24
RESERVED	HYST_EN	IE	WU_CFG		IOMODE		
R-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h		
23	22	21	20	19	18	17	16
IOEV_AON_PROG2_EN	IOEV_AON_PROG1_EN	IOEV_AON_PROG0_EN	RESERVED		EDGE_IRQ_EN	EDGE_DET	
R/W-0h	R/W-0h	R/W-0h	R-0h		R/W-0h	R/W-0h	
15	14	13	12	11	10	9	8
RESERVED	PULL_CTL		SLEW_RED	IOCURR		IOSTR	
R-0h	R/W-3h		R/W-0h	R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
IOEV_RTC_EN	IOEV_MCU_WU_EN	PORT_ID					
R/W-0h	R/W-0h	R/W-0h					

表 13-51. IOCFG21 寄存器字段说明

位	字段	类型	复位	说明
31	RESERVED	R	0h	保留
30	HYST_EN	R/W	0h	0：输入迟滞禁用 1：输入迟滞使能
29	IE	R/W	0h	0：输入禁用 1：输入使能 注意：如果 IO 配置为 AUX PORT_ID = 0x08，则使能将被忽略。
28-27	WU_CFG	R/W	0h	如果 DIO 配置为 GPIO 或非 AON 外设信号（PORT_ID 0x00 或 >0x08）： 00：无唤醒 01：无唤醒 10：如果该焊盘变为低电平，则从关断状态唤醒。 11：如果该焊盘变为高电平，则从关断状态唤醒。 如果 IO 配置为 AON 外设信号或 AUX PORT_ID 0x01-0x08，则该寄存器仅设置是否使能唤醒。 00、01：唤醒禁用 10、11：唤醒使能 极性通过 AON 寄存器进行控制。 注意：当 MSB 置位时，IOC 将停用 DIO 的输出使能。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-51. IOCFG21 寄存器字段说明（续）

位	字段	类型	复位	说明
26-24	IOMODE	R/W	0h	IO 模式 不适用于配置为 AON 外设信号和 AUX PORT_ID 0x01-0x08 的 IO AUX 有自己的 open_source/drain 配置。 0x2: 保留。未定义的行为。 0x3：保留。未定义的行为。 0h = NORMAL：正常输入/输出 1h = INV：反相输入/输出 4h = OPENDR： 开漏， 正常输入/输出 5h = OPENDR_INV：开漏 反相输入/输出 6h = OPENSRC：开源 正常输入/输出 7h = OPENSRC_INV：开源 反相输入/输出
23	IOEV_AON_PROG2_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG2 事件置为有效 1：输入边沿检测将 AON_PROG2 事件置为有效
22	IOEV_AON_PROG1_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG1 事件置为有效 1：输入边沿检测将 AON_PROG1 事件置为有效
21	IOEV_AON_PROG0_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG0 事件置为有效 1：输入边沿检测将 AON_PROG0 事件置为有效
20-19	RESERVED	R	0h	保留
18	EDGE_IRQ_EN	R/W	0h	0：无中断生成 1：使能该 IO 的中断生成（仅在 EDGE_DET 已使能时有效）
17-16	EDGE_DET	R/W	0h	使能该 IO 上的边沿检测事件生成 0h = NONE：无边沿检测 1h = 负边沿检测 2h = 正边沿检测 3h = 正负边沿检测
15	RESERVED	R	0h	保留
14-13	PULL_CTL	R/W	3h	上拉/下拉控制 1h = DWN：下拉 2h = UP：上拉 3h = DIS：无上拉/下拉
12	SLEW_RED	R/W	0h	0：标称压摆率 1：在输出驱动器中降低压摆率。
11-10	IOCURR	R/W	0h	选择该 IO 的 IO 电流模式。 0h = 2MA：低电流 (LC) 模式：IOSTR 设置为 AUTO 时最小值为 2mA 1h = 4MA：高电流 (HC) 模式：IOSTR 设置为 AUTO 时最小值为 4mA 2h = 4_8MA：扩展电流 (EC) 模式：当 IOSTR 设为 AUTO 时，双驱动强度 IO 的最小值为 8mA（标准 IO 的最小值为 4mA）

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-51. IOCFG21 寄存器字段说明（续）

位	字段	类型	复位	说明
9-8	IOSTR	R/W	0h	选择该 IO 的驱动强度控制源。 此设置控制低电流 (LC) 模式的驱动强度。可在 IOCURR 中选择更高的驱动强度 0h = 自动驱动强度，由 AON BATMON 根据电池电压控制。（VDDS 时最小 2mA） 1h = 最小驱动强度，由 AON_IOC:IOSTRMIN 控制（默认值为 3.3V 时最小 2mA） 2h = MED：中等驱动强度，由 AON_IOC:IOSTRMED 控制（默认值为 2.5V 时最小 2mA） 3h = 最大驱动强度，由 AON_IOC:IOSTRMAX 控制（默认值为 1.8V 时最小 2mA）
7	IOEV_RTC_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 RTC 事件置为有效 1：输入边沿检测将 RTC 事件置为有效
6	IOEV_MCU_WU_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 MCU_WU 事件置为有效 1：输入边沿检测将 MCU_WU 事件置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-51. IOCFG21 寄存器字段说明（续）

位	字段	类型	复位	说明
5-0	PORT_ID	R/W	0h	选择 DIO21 的用途 注意：除明确要求更改 PORT_ID 值的情况外，不应写入此字段。 0h =通用 IO 7h = AON 32 KHz 时钟 (SCLK_LF) 8h = AUX IO 9h = SSI0_RX：SSI0 RX Ah = SSI0_TX：SSI0 TX Bh = SSI0_FSS：SSI0 FSS Ch = SSI0_CLK：SSI0 CLK Dh = I2C_MSSDA：I2C 数据 Eh = I2C_MSSCL：I2C 时钟 Fh = UART0_RX：UART0 RX 10h = UART0_TX：UART0 TX 11h = UART0_CTS：UART0 CTS 12h = UART0_RTS：UART0 RTS 13h = UART1_RX：UART1 RX 14h = UART1_TX：UART1 TX 15h = UART1_CTS：UART1 CTS 16h = UART1_RTS：UART1 RTS 17h = PORT_EVENT0：端口事件 0 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 18h = PORT_EVENT1： 端口事件 1 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 19h = PORT_EVENT2： 端口事件 2 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ah = PORT_EVENT3： 端口事件 3 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Bh = PORT_EVENT4： 端口事件 4 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ch = PORT_EVENT5： 端口事件 5 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Dh = PORT_EVENT6： 端口事件 6 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Eh = PORT_EVENT7： 端口事件 7 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 20h = CPU_SWV： CPU SWV 21h = SSI1_RX：SSI1 RX 22h = SSI1_TX：SSI1 TX 23h = SSI1_FSS：SSI1 FSS 24h = SSI1_CLK：SSI1 CLK 25h = I2S_AD0：I2S 数据 0 26h = I2S_AD1： I2S 数据 1 27h = I2S_WCLK： I2S WCLK 28h = I2S_BCLK：I2S BCLK 29h = I2S_MCLK：I2S MCLK 2Eh = 射频内核跟踪

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-51. IOCFG21 寄存器字段说明（续）

位	字段	类型	复位	说明
				2Fh = 射频内核数据输出 0 30h = 射频内核数据输出 1 31h = 射频内核数据输出 2 32h = 射频内核数据输出 3 33h = 射频内核数据输入 0 34h = 射频内核数据输入 1 35h = 射频内核 SMI 数据链路输出 36h = 射频内核 SMI 数据链路输入 37h = 射频内核 SMI 命令链路输出 38h = 射频内核 SMI 命令链路输入

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.3.23 IOCFG22 寄存器（偏移 = 58h）[复位 = 00006000h]

图 13-45 展示了 IOCFG22，表 13-52 对其进行了介绍。

返回到汇总表。

DIO22 的配置

图 13-45. IOCFG22 寄存器

31	30	29	28	27	26	25	24
RESERVED	HYST_EN	IE	WU_CFG		IOMODE		
R-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h		
23	22	21	20	19	18	17	16
IOEV_AON_PROG2_EN	IOEV_AON_PROG1_EN	IOEV_AON_PROG0_EN	RESERVED		EDGE_IRQ_EN	EDGE_DET	
R/W-0h	R/W-0h	R/W-0h	R-0h		R/W-0h	R/W-0h	
15	14	13	12	11	10	9	8
RESERVED	PULL_CTL		SLEW_RED	IOCURR		IOSTR	
R-0h	R/W-3h		R/W-0h	R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
IOEV_RTC_EN	IOEV_MCU_WU_EN	PORT_ID					
R/W-0h	R/W-0h	R/W-0h					

表 13-52. IOCFG22 寄存器字段说明

位	字段	类型	复位	说明
31	RESERVED	R	0h	保留
30	HYST_EN	R/W	0h	0：输入迟滞禁用 1：输入迟滞使能
29	IE	R/W	0h	0：输入禁用 1：输入使能 注意：如果 IO 配置为 AUX PORT_ID = 0x08，则使能将被忽略。
28-27	WU_CFG	R/W	0h	如果 DIO 配置为 GPIO 或非 AON 外设信号（PORT_ID 0x00 或 >0x08）： 00：无唤醒 01：无唤醒 10：如果该焊盘变为低电平，则从关断状态唤醒。 11：如果该焊盘变为高电平，则从关断状态唤醒。 如果 IO 配置为 AON 外设信号或 AUX PORT_ID 0x01-0x08，则该寄存器仅设置是否使能唤醒。 00、01：唤醒禁用 10、11：唤醒使能 极性通过 AON 寄存器进行控制。 注意：当 MSB 置位时，IOC 将停用 DIO 的输出使能。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-52. IOCFG22 寄存器字段说明（续）

位	字段	类型	复位	说明
26-24	IOMODE	R/W	0h	IO 模式 不适用于配置为 AON 外设信号和 AUX PORT_ID 0x01-0x08 的 IO AUX 有自己的 open_source/drain 配置。 0x2: 保留。未定义的行为。 0x3：保留。未定义的行为。 0h = NORMAL：正常输入/输出 1h = INV：反相输入/输出 4h = OPENDR： 开漏， 正常输入/输出 5h = OPENDR_INV：开漏 反相输入/输出 6h = OPENSRC：开源 正常输入/输出 7h = OPENSRC_INV：开源 反相输入/输出
23	IOEV_AON_PROG2_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG2 事件置为有效 1：输入边沿检测将 AON_PROG2 事件置为有效
22	IOEV_AON_PROG1_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG1 事件置为有效 1：输入边沿检测将 AON_PROG1 事件置为有效
21	IOEV_AON_PROG0_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG0 事件置为有效 1：输入边沿检测将 AON_PROG0 事件置为有效
20-19	RESERVED	R	0h	保留
18	EDGE_IRQ_EN	R/W	0h	0：无中断生成 1：使能该 IO 的中断生成（仅在 EDGE_DET 已使能时有效）
17-16	EDGE_DET	R/W	0h	使能该 IO 上的边沿检测事件生成 0h = NONE：无边沿检测 1h = 负边沿检测 2h = 正边沿检测 3h = 正负边沿检测
15	RESERVED	R	0h	保留
14-13	PULL_CTL	R/W	3h	上拉/下拉控制 1h = DWN：下拉 2h = UP：上拉 3h = DIS：无上拉/下拉
12	SLEW_RED	R/W	0h	0：标称压摆率 1：在输出驱动器中降低压摆率。
11-10	IOCURR	R/W	0h	选择该 IO 的 IO 电流模式。 0h = 2MA：低电流 (LC) 模式：IOSTR 设置为 AUTO 时最小值为 2mA 1h = 4MA：高电流 (HC) 模式：IOSTR 设置为 AUTO 时最小值为 4mA 2h = 4_8MA：扩展电流 (EC) 模式：当 IOSTR 设为 AUTO 时，双驱动强度 IO 的最小值为 8mA（标准 IO 的最小值为 4mA）

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-52. IOCFG22 寄存器字段说明（续）

位	字段	类型	复位	说明
9-8	IOSTR	R/W	0h	选择该 IO 的驱动强度控制源。 此设置控制低电流 (LC) 模式的驱动强度。可在 IOCURR 中选择更高的驱动强度 0h = 自动驱动强度，由 AON BATMON 根据电池电压控制。（VDDS 时最小 2mA） 1h = 最小驱动强度，由 AON_IOC:IOSTRMIN 控制（默认值为 3.3V 时最小 2mA） 2h = MED：中等驱动强度，由 AON_IOC:IOSTRMED 控制（默认值为 2.5V 时最小 2mA） 3h = 最大驱动强度，由 AON_IOC:IOSTRMAX 控制（默认值为 1.8V 时最小 2mA）
7	IOEV_RTC_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 RTC 事件置为有效 1：输入边沿检测将 RTC 事件置为有效
6	IOEV_MCU_WU_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 MCU_WU 事件置为有效 1：输入边沿检测将 MCU_WU 事件置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-52. IOCFG22 寄存器字段说明（续）

位	字段	类型	复位	说明
5-0	PORT_ID	R/W	0h	选择 DIO22 的用途 注意：除明确要求更改 PORT_ID 值的情况外，不应写入此字段。 0h =通用 IO 7h = AON 32 KHz 时钟 (SCLK_LF) 8h = AUX IO 9h = SSI0_RX：SSI0 RX Ah = SSI0_TX：SSI0 TX Bh = SSI0_FSS：SSI0 FSS Ch = SSI0_CLK：SSI0 CLK Dh = I2C_MSSDA：I2C 数据 Eh = I2C_MSSCL：I2C 时钟 Fh = UART0_RX：UART0 RX 10h = UART0_TX：UART0 TX 11h = UART0_CTS：UART0 CTS 12h = UART0_RTS：UART0 RTS 13h = UART1_RX：UART1 RX 14h = UART1_TX：UART1 TX 15h = UART1_CTS：UART1 CTS 16h = UART1_RTS：UART1 RTS 17h = PORT_EVENT0：端口事件 0 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 18h = PORT_EVENT1： 端口事件 1 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 19h = PORT_EVENT2： 端口事件 2 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ah = PORT_EVENT3： 端口事件 3 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Bh = PORT_EVENT4： 端口事件 4 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ch = PORT_EVENT5： 端口事件 5 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Dh = PORT_EVENT6： 端口事件 6 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Eh = PORT_EVENT7： 端口事件 7 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 20h = CPU_SWV： CPU SWV 21h = SSI1_RX：SSI1 RX 22h = SSI1_TX：SSI1 TX 23h = SSI1_FSS：SSI1 FSS 24h = SSI1_CLK：SSI1 CLK 25h = I2S_AD0：I2S 数据 0 26h = I2S_AD1： I2S 数据 1 27h = I2S_WCLK： I2S WCLK 28h = I2S_BCLK：I2S BCLK 29h = I2S_MCLK：I2S MCLK 2Eh = 射频内核跟踪

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-52. IOCFG22 寄存器字段说明（续）

位	字段	类型	复位	说明
				2Fh = 射频内核数据输出 0 30h = 射频内核数据输出 1 31h = 射频内核数据输出 2 32h = 射频内核数据输出 3 33h = 射频内核数据输入 0 34h = 射频内核数据输入 1 35h = 射频内核 SMI 数据链路输出 36h = 射频内核 SMI 数据链路输入 37h = 射频内核 SMI 命令链路输出 38h = 射频内核 SMI 命令链路输入

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.3.24 IOCFG23 寄存器（偏移 = 5Ch）[复位 = 00006000h]

图 13-46 展示了 IOCFG23，表 13-53 对其进行了介绍。

返回到汇总表。

DIO23 的配置

图 13-46. IOCFG23 寄存器

31	30	29	28	27	26	25	24
RESERVED	HYST_EN	IE	WU_CFG		IOMODE		
R-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h		
23	22	21	20	19	18	17	16
IOEV_AON_PROG2_EN	IOEV_AON_PROG1_EN	IOEV_AON_PROG0_EN	RESERVED		EDGE_IRQ_EN	EDGE_DET	
R/W-0h	R/W-0h	R/W-0h	R-0h		R/W-0h	R/W-0h	
15	14	13	12	11	10	9	8
RESERVED	PULL_CTL		SLEW_RED	IOCURR		IOSTR	
R-0h	R/W-3h		R/W-0h	R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
IOEV_RTC_EN	IOEV_MCU_WU_EN	PORT_ID					
R/W-0h	R/W-0h	R/W-0h					

表 13-53. IOCFG23 寄存器字段说明

位	字段	类型	复位	说明
31	RESERVED	R	0h	保留
30	HYST_EN	R/W	0h	0：输入迟滞禁用 1：输入迟滞使能
29	IE	R/W	0h	0：输入禁用 1：输入使能 注意：如果 IO 配置为 AUX PORT_ID = 0x08，则使能将被忽略。
28-27	WU_CFG	R/W	0h	如果 DIO 配置为 GPIO 或非 AON 外设信号（PORT_ID 0x00 或 >0x08）： 00：无唤醒 01：无唤醒 10：如果该焊盘变为低电平，则从关断状态唤醒。 11：如果该焊盘变为高电平，则从关断状态唤醒。 如果 IO 配置为 AON 外设信号或 AUX PORT_ID 0x01-0x08，则该寄存器仅设置是否使能唤醒。 00、01：唤醒禁用 10、11：唤醒使能 极性通过 AON 寄存器进行控制。 注意：当 MSB 置位时，IOC 将停用 DIO 的输出使能。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-53. IOCFG23 寄存器字段说明（续）

位	字段	类型	复位	说明
26-24	IOMODE	R/W	0h	IO 模式 不适用于配置为 AON 外设信号和 AUX PORT_ID 0x01-0x08 的 IO AUX 有自己的 open_source/drain 配置。 0x2: 保留。未定义的行为。 0x3：保留。未定义的行为。 0h = NORMAL：正常输入/输出 1h = INV：反相输入/输出 4h = OPENDR： 开漏， 正常输入/输出 5h = OPENDR_INV：开漏 反相输入/输出 6h = OPENSRC：开源 正常输入/输出 7h = OPENSRC_INV：开源 反相输入/输出
23	IOEV_AON_PROG2_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG2 事件置为有效 1：输入边沿检测将 AON_PROG2 事件置为有效
22	IOEV_AON_PROG1_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG1 事件置为有效 1：输入边沿检测将 AON_PROG1 事件置为有效
21	IOEV_AON_PROG0_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG0 事件置为有效 1：输入边沿检测将 AON_PROG0 事件置为有效
20-19	RESERVED	R	0h	保留
18	EDGE_IRQ_EN	R/W	0h	0：无中断生成 1：使能该 IO 的中断生成（仅在 EDGE_DET 已使能时有效）
17-16	EDGE_DET	R/W	0h	使能该 IO 上的边沿检测事件生成 0h = NONE：无边沿检测 1h = 负边沿检测 2h = 正边沿检测 3h = 正负边沿检测
15	RESERVED	R	0h	保留
14-13	PULL_CTL	R/W	3h	上拉/下拉控制 1h = DWN：下拉 2h = UP：上拉 3h = DIS：无上拉/下拉
12	SLEW_RED	R/W	0h	0：标称压摆率 1：在输出驱动器中降低压摆率。
11-10	IOCURR	R/W	0h	选择该 IO 的 IO 电流模式。 0h = 2MA：低电流 (LC) 模式：IOSTR 设置为 AUTO 时最小值为 2mA 1h = 4MA：高电流 (HC) 模式：IOSTR 设置为 AUTO 时最小值为 4mA 2h = 4_8MA：扩展电流 (EC) 模式：当 IOSTR 设为 AUTO 时，双驱动强度 IO 的最小值为 8mA（标准 IO 的最小值为 4mA）

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-53. IOCFG23 寄存器字段说明（续）

位	字段	类型	复位	说明
9-8	IOSTR	R/W	0h	选择该 IO 的驱动强度控制源。 此设置控制低电流 (LC) 模式的驱动强度。可在 IOCURR 中选择更高的驱动强度 0h = 自动驱动强度，由 AON BATMON 根据电池电压控制。（VDDS 时最小 2mA） 1h = 最小驱动强度，由 AON_IOC:IOSTRMIN 控制（默认值为 3.3V 时最小 2mA） 2h = MED：中等驱动强度，由 AON_IOC:IOSTRMED 控制（默认值为 2.5V 时最小 2mA） 3h = 最大驱动强度，由 AON_IOC:IOSTRMAX 控制（默认值为 1.8V 时最小 2mA）
7	IOEV_RTC_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 RTC 事件置为有效 1：输入边沿检测将 RTC 事件置为有效
6	IOEV_MCU_WU_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 MCU_WU 事件置为有效 1：输入边沿检测将 MCU_WU 事件置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-53. IOCFG23 寄存器字段说明 （续）

位	字段	类型	复位	说明
5-0	PORT_ID	R/W	0h	选择 DIO23 的用途 注意：除明确要求更改 PORT_ID 值的情况外，不应写入此字段。 0h =通用 IO 7h = AON 32 KHz 时钟 (SCLK_LF) 8h = AUX IO 9h = SSI0_RX：SSI0 RX Ah = SSI0_TX：SSI0 TX Bh = SSI0_FSS：SSI0 FSS Ch = SSI0_CLK：SSI0 CLK Dh = I2C_MSSDA：I2C 数据 Eh = I2C_MSSCL：I2C 时钟 Fh = UART0_RX：UART0 RX 10h = UART0_TX：UART0 TX 11h = UART0_CTS：UART0 CTS 12h = UART0_RTS：UART0 RTS 13h = UART1_RX：UART1 RX 14h = UART1_TX：UART1 TX 15h = UART1_CTS：UART1 CTS 16h = UART1_RTS：UART1 RTS 17h = PORT_EVENT0：端口事件 0 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 18h = PORT_EVENT1： 端口事件 1 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 19h = PORT_EVENT2： 端口事件 2 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ah = PORT_EVENT3： 端口事件 3 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Bh = PORT_EVENT4： 端口事件 4 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ch = PORT_EVENT5： 端口事件 5 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Dh = PORT_EVENT6： 端口事件 6 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Eh = PORT_EVENT7： 端口事件 7 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 20h = CPU_SWV： CPU SWV 21h = SSI1_RX：SSI1 RX 22h = SSI1_TX：SSI1 TX 23h = SSI1_FSS：SSI1 FSS 24h = SSI1_CLK：SSI1 CLK 25h = I2S_AD0：I2S 数据 0 26h = I2S_AD1： I2S 数据 1 27h = I2S_WCLK： I2S WCLK 28h = I2S_BCLK：I2S BCLK 29h = I2S_MCLK：I2S MCLK 2Eh = 射频内核跟踪

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-53. IOCFG23 寄存器字段说明（续）

位	字段	类型	复位	说明
				2Fh = 射频内核数据输出 0 30h = 射频内核数据输出 1 31h = 射频内核数据输出 2 32h = 射频内核数据输出 3 33h = 射频内核数据输入 0 34h = 射频内核数据输入 1 35h = 射频内核 SMI 数据链路输出 36h = 射频内核 SMI 数据链路输入 37h = 射频内核 SMI 命令链路输出 38h = 射频内核 SMI 命令链路输入

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.3.25 IOCFG24 寄存器（偏移 = 60h）[复位 = 00006000h]

图 13-47 展示了 IOCFG24，表 13-54 对其进行了介绍。

返回到汇总表。

DIO24 的配置

图 13-47. IOCFG24 寄存器

31	30	29	28	27	26	25	24
RESERVED	HYST_EN	IE	WU_CFG		IOMODE		
R-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h		
23	22	21	20	19	18	17	16
IOEV_AON_PROG2_EN	IOEV_AON_PROG1_EN	IOEV_AON_PROG0_EN	RESERVED		EDGE_IRQ_EN	EDGE_DET	
R/W-0h	R/W-0h	R/W-0h	R-0h		R/W-0h	R/W-0h	
15	14	13	12	11	10	9	8
RESERVED	PULL_CTL		SLEW_RED	IOCURR		IOSTR	
R-0h	R/W-3h		R/W-0h	R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
IOEV_RTC_EN	IOEV_MCU_WU_EN	PORT_ID					
R/W-0h	R/W-0h	R/W-0h					

表 13-54. IOCFG24 寄存器字段说明

位	字段	类型	复位	说明
31	RESERVED	R	0h	保留
30	HYST_EN	R/W	0h	0：输入迟滞禁用 1：输入迟滞使能
29	IE	R/W	0h	0：输入禁用 1：输入使能 注意：如果 IO 配置为 AUX PORT_ID = 0x08，则使能将被忽略。
28-27	WU_CFG	R/W	0h	如果 DIO 配置为 GPIO 或非 AON 外设信号（PORT_ID 0x00 或 >0x08）： 00：无唤醒 01：无唤醒 10：如果该焊盘变为低电平，则从关断状态唤醒。 11：如果该焊盘变为高电平，则从关断状态唤醒。 如果 IO 配置为 AON 外设信号或 AUX PORT_ID 0x01-0x08，则该寄存器仅设置是否使能唤醒。 00、01：唤醒禁用 10、11：唤醒使能 极性通过 AON 寄存器进行控制。 注意：当 MSB 置位时，IOC 将停用 DIO 的输出使能。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-54. IOCFG24 寄存器字段说明（续）

位	字段	类型	复位	说明
26-24	IOMODE	R/W	0h	IO 模式 不适用于配置为 AON 外设信号和 AUX PORT_ID 0x01-0x08 的 IO AUX 有自己的 open_source/drain 配置。 0x2: 保留。未定义的行为。 0x3：保留。未定义的行为。 0h = NORMAL：正常输入/输出 1h = INV：反相输入/输出 4h = OPENDR： 开漏， 正常输入/输出 5h = OPENDR_INV：开漏 反相输入/输出 6h = OPENSRC：开源 正常输入/输出 7h = OPENSRC_INV：开源 反相输入/输出
23	IOEV_AON_PROG2_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG2 事件置为有效 1：输入边沿检测将 AON_PROG2 事件置为有效
22	IOEV_AON_PROG1_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG1 事件置为有效 1：输入边沿检测将 AON_PROG1 事件置为有效
21	IOEV_AON_PROG0_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG0 事件置为有效 1：输入边沿检测将 AON_PROG0 事件置为有效
20-19	RESERVED	R	0h	保留
18	EDGE_IRQ_EN	R/W	0h	0：无中断生成 1：使能该 IO 的中断生成（仅在 EDGE_DET 已使能时有效）
17-16	EDGE_DET	R/W	0h	使能该 IO 上的边沿检测事件生成 0h = NONE：无边沿检测 1h = 负边沿检测 2h = 正边沿检测 3h = 正负边沿检测
15	RESERVED	R	0h	保留
14-13	PULL_CTL	R/W	3h	上拉/下拉控制 1h = DWN：下拉 2h = UP：上拉 3h = DIS：无上拉/下拉
12	SLEW_RED	R/W	0h	0：标称压摆率 1：在输出驱动器中降低压摆率。
11-10	IOCURR	R/W	0h	选择该 IO 的 IO 电流模式。 0h = 2MA：低电流 (LC) 模式：IOSTR 设置为 AUTO 时最小值为 2mA 1h = 4MA：高电流 (HC) 模式：IOSTR 设置为 AUTO 时最小值为 4mA 2h = 4_8MA：扩展电流 (EC) 模式：当 IOSTR 设为 AUTO 时，双驱动强度 IO 的最小值为 8mA（标准 IO 的最小值为 4mA）

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-54. IOCFG24 寄存器字段说明（续）

位	字段	类型	复位	说明
9-8	IOSTR	R/W	0h	选择该 IO 的驱动强度控制源。 此设置控制低电流 (LC) 模式的驱动强度。可在 IOCURR 中选择更高的驱动强度 0h = 自动驱动强度，由 AON BATMON 根据电池电压控制。（VDDS 时最小 2mA） 1h = 最小驱动强度，由 AON_IOC:IOSTRMIN 控制（默认值为 3.3V 时最小 2mA） 2h = MED：中等驱动强度，由 AON_IOC:IOSTRMED 控制（默认值为 2.5V 时最小 2mA） 3h = 最大驱动强度，由 AON_IOC:IOSTRMAX 控制（默认值为 1.8V 时最小 2mA）
7	IOEV_RTC_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 RTC 事件置为有效 1：输入边沿检测将 RTC 事件置为有效
6	IOEV_MCU_WU_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 MCU_WU 事件置为有效 1：输入边沿检测将 MCU_WU 事件置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-54. IOCFG24 寄存器字段说明（续）

位	字段	类型	复位	说明
5-0	PORT_ID	R/W	0h	选择 DIO24 的用途 注意：除明确要求更改 PORT_ID 值的情况外，不应写入此字段。 0h =通用 IO 7h = AON 32 KHz 时钟 (SCLK_LF) 8h = AUX IO 9h = SSI0_RX：SSI0 RX Ah = SSI0_TX：SSI0 TX Bh = SSI0_FSS：SSI0 FSS Ch = SSI0_CLK：SSI0 CLK Dh = I2C_MSSDA：I2C 数据 Eh = I2C_MSSCL：I2C 时钟 Fh = UART0_RX：UART0 RX 10h = UART0_TX：UART0 TX 11h = UART0_CTS：UART0 CTS 12h = UART0_RTS：UART0 RTS 13h = UART1_RX：UART1 RX 14h = UART1_TX：UART1 TX 15h = UART1_CTS：UART1 CTS 16h = UART1_RTS：UART1 RTS 17h = PORT_EVENT0：端口事件 0 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 18h = PORT_EVENT1： 端口事件 1 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 19h = PORT_EVENT2： 端口事件 2 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ah = PORT_EVENT3： 端口事件 3 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Bh = PORT_EVENT4： 端口事件 4 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ch = PORT_EVENT5： 端口事件 5 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Dh = PORT_EVENT6： 端口事件 6 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Eh = PORT_EVENT7： 端口事件 7 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 20h = CPU_SWV： CPU SWV 21h = SSI1_RX：SSI1 RX 22h = SSI1_TX：SSI1 TX 23h = SSI1_FSS：SSI1 FSS 24h = SSI1_CLK：SSI1 CLK 25h = I2S_AD0：I2S 数据 0 26h = I2S_AD1： I2S 数据 1 27h = I2S_WCLK： I2S WCLK 28h = I2S_BCLK：I2S BCLK 29h = I2S_MCLK：I2S MCLK 2Eh = 射频内核跟踪

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-54. IOCFG24 寄存器字段说明（续）

位	字段	类型	复位	说明
				2Fh = 射频内核数据输出 0 30h = 射频内核数据输出 1 31h = 射频内核数据输出 2 32h = 射频内核数据输出 3 33h = 射频内核数据输入 0 34h = 射频内核数据输入 1 35h = 射频内核 SMI 数据链路输出 36h = 射频内核 SMI 数据链路输入 37h = 射频内核 SMI 命令链路输出 38h = 射频内核 SMI 命令链路输入

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.3.26 IOCFG25 寄存器（偏移 = 64h）[复位 = 00006000h]

图 13-48 展示了 IOCFG25，表 13-55 对其进行了介绍。

返回到汇总表。

DIO25 的配置

图 13-48. IOCFG25 寄存器

31	30	29	28	27	26	25	24
RESERVED	HYST_EN	IE	WU_CFG		IOMODE		
R-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h		
23	22	21	20	19	18	17	16
IOEV_AON_PROG2_EN	IOEV_AON_PROG1_EN	IOEV_AON_PROG0_EN	RESERVED		EDGE_IRQ_EN	EDGE_DET	
R/W-0h	R/W-0h	R/W-0h	R-0h		R/W-0h	R/W-0h	
15	14	13	12	11	10	9	8
RESERVED	PULL_CTL		SLEW_RED	IOCURR		IOSTR	
R-0h	R/W-3h		R/W-0h	R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
IOEV_RTC_EN	IOEV_MCU_WU_EN	PORT_ID					
R/W-0h	R/W-0h	R/W-0h					

表 13-55. IOCFG25 寄存器字段说明

位	字段	类型	复位	说明
31	RESERVED	R	0h	保留
30	HYST_EN	R/W	0h	0：输入迟滞禁用 1：输入迟滞使能
29	IE	R/W	0h	0：输入禁用 1：输入使能 注意：如果 IO 配置为 AUX PORT_ID = 0x08，则使能将被忽略。
28-27	WU_CFG	R/W	0h	如果 DIO 配置为 GPIO 或非 AON 外设信号（PORT_ID 0x00 或 >0x08）： 00：无唤醒 01：无唤醒 10：如果该焊盘变为低电平，则从关断状态唤醒。 11：如果该焊盘变为高电平，则从关断状态唤醒。 如果 IO 配置为 AON 外设信号或 AUX PORT_ID 0x01-0x08，则该寄存器仅设置是否使能唤醒。 00、01：唤醒禁用 10、11：唤醒使能 极性通过 AON 寄存器进行控制。 注意：当 MSB 置位时，IOC 将停用 DIO 的输出使能。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-55. IOCFG25 寄存器字段说明（续）

位	字段	类型	复位	说明
26-24	IOMODE	R/W	0h	IO 模式 不适用于配置为 AON 外设信号和 AUX PORT_ID 0x01-0x08 的 IO AUX 有自己的 open_source/drain 配置。 0x2: 保留。未定义的行为。 0x3：保留。未定义的行为。 0h = NORMAL：正常输入/输出 1h = INV：反相输入/输出 4h = OPENDR： 开漏， 正常输入/输出 5h = OPENDR_INV：开漏 反相输入/输出 6h = OPENSRC：开源 正常输入/输出 7h = OPENSRC_INV：开源 反相输入/输出
23	IOEV_AON_PROG2_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG2 事件置为有效 1：输入边沿检测将 AON_PROG2 事件置为有效
22	IOEV_AON_PROG1_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG1 事件置为有效 1：输入边沿检测将 AON_PROG1 事件置为有效
21	IOEV_AON_PROG0_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG0 事件置为有效 1：输入边沿检测将 AON_PROG0 事件置为有效
20-19	RESERVED	R	0h	保留
18	EDGE_IRQ_EN	R/W	0h	0：无中断生成 1：使能该 IO 的中断生成（仅在 EDGE_DET 已使能时有效）
17-16	EDGE_DET	R/W	0h	使能该 IO 上的边沿检测事件生成 0h = NONE：无边沿检测 1h = 负边沿检测 2h = 正边沿检测 3h = 正负边沿检测
15	RESERVED	R	0h	保留
14-13	PULL_CTL	R/W	3h	上拉/下拉控制 1h = DWN：下拉 2h = UP：上拉 3h = DIS：无上拉/下拉
12	SLEW_RED	R/W	0h	0：标称压摆率 1：在输出驱动器中降低压摆率。
11-10	IOCURR	R/W	0h	选择该 IO 的 IO 电流模式。 0h = 2MA：低电流 (LC) 模式：IOSTR 设置为 AUTO 时最小值为 2mA 1h = 4MA：高电流 (HC) 模式：IOSTR 设置为 AUTO 时最小值为 4mA 2h = 4_8MA：扩展电流 (EC) 模式：当 IOSTR 设为 AUTO 时，双驱动强度 IO 的最小值为 8mA（标准 IO 的最小值为 4mA）

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-55. IOCFG25 寄存器字段说明（续）

位	字段	类型	复位	说明
9-8	IOSTR	R/W	0h	选择该 IO 的驱动强度控制源。 此设置控制低电流 (LC) 模式的驱动强度。可在 IOCURR 中选择更高的驱动强度 0h = 自动驱动强度，由 AON BATMON 根据电池电压控制。（VDDS 时最小 2mA） 1h = 最小驱动强度，由 AON_IOC:IOSTRMIN 控制（默认值为 3.3V 时最小 2mA） 2h = MED：中等驱动强度，由 AON_IOC:IOSTRMED 控制（默认值为 2.5V 时最小 2mA） 3h = 最大驱动强度，由 AON_IOC:IOSTRMAX 控制（默认值为 1.8V 时最小 2mA）
7	IOEV_RTC_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 RTC 事件置为有效 1：输入边沿检测将 RTC 事件置为有效
6	IOEV_MCU_WU_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 MCU_WU 事件置为有效 1：输入边沿检测将 MCU_WU 事件置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-55. IOCFG25 寄存器字段说明 （续）

位	字段	类型	复位	说明
5-0	PORT_ID	R/W	0h	选择 DIO25 的用途 注意：除明确要求更改 PORT_ID 值的情况外，不应写入此字段。 0h =通用 IO 7h = AON 32 KHz 时钟 (SCLK_LF) 8h = AUX IO 9h = SSI0_RX：SSI0 RX Ah = SSI0_TX：SSI0 TX Bh = SSI0_FSS：SSI0 FSS Ch = SSI0_CLK：SSI0 CLK Dh = I2C_MSSDA：I2C 数据 Eh = I2C_MSSCL：I2C 时钟 Fh = UART0_RX：UART0 RX 10h = UART0_TX：UART0 TX 11h = UART0_CTS：UART0 CTS 12h = UART0_RTS：UART0 RTS 13h = UART1_RX：UART1 RX 14h = UART1_TX：UART1 TX 15h = UART1_CTS：UART1 CTS 16h = UART1_RTS：UART1 RTS 17h = PORT_EVENT0：端口事件 0 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 18h = PORT_EVENT1： 端口事件 1 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 19h = PORT_EVENT2： 端口事件 2 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ah = PORT_EVENT3： 端口事件 3 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Bh = PORT_EVENT4： 端口事件 4 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ch = PORT_EVENT5： 端口事件 5 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Dh = PORT_EVENT6： 端口事件 6 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Eh = PORT_EVENT7： 端口事件 7 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 20h = CPU_SWV： CPU SWV 21h = SSI1_RX：SSI1 RX 22h = SSI1_TX：SSI1 TX 23h = SSI1_FSS：SSI1 FSS 24h = SSI1_CLK：SSI1 CLK 25h = I2S_AD0：I2S 数据 0 26h = I2S_AD1： I2S 数据 1 27h = I2S_WCLK： I2S WCLK 28h = I2S_BCLK：I2S BCLK 29h = I2S_MCLK：I2S MCLK 2Eh = 射频内核跟踪

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-55. IOCFG25 寄存器字段说明（续）

位	字段	类型	复位	说明
				2Fh = 射频内核数据输出 0 30h = 射频内核数据输出 1 31h = 射频内核数据输出 2 32h = 射频内核数据输出 3 33h = 射频内核数据输入 0 34h = 射频内核数据输入 1 35h = 射频内核 SMI 数据链路输出 36h = 射频内核 SMI 数据链路输入 37h = 射频内核 SMI 命令链路输出 38h = 射频内核 SMI 命令链路输入

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.3.27 IOCFG26 寄存器（偏移 = 68h）[复位 = 00006000h]

图 13-49 展示了 IOCFG26，表 13-56 对其进行了介绍。

返回到汇总表。

DIO26 的配置

图 13-49. IOCFG26 寄存器

31	30	29	28	27	26	25	24
RESERVED	HYST_EN	IE	WU_CFG		IOMODE		
R-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h		
23	22	21	20	19	18	17	16
IOEV_AON_PROG2_EN	IOEV_AON_PROG1_EN	IOEV_AON_PROG0_EN	RESERVED		EDGE_IRQ_EN	EDGE_DET	
R/W-0h	R/W-0h	R/W-0h	R-0h		R/W-0h	R/W-0h	
15	14	13	12	11	10	9	8
RESERVED	PULL_CTL		SLEW_RED	IOCURR		IOSTR	
R-0h	R/W-3h		R/W-0h	R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
IOEV_RTC_EN	IOEV_MCU_WU_EN	PORT_ID					
R/W-0h	R/W-0h	R/W-0h					

表 13-56. IOCFG26 寄存器字段说明

位	字段	类型	复位	说明
31	RESERVED	R	0h	保留
30	HYST_EN	R/W	0h	0：输入迟滞禁用 1：输入迟滞使能
29	IE	R/W	0h	0：输入禁用 1：输入使能 注意：如果 IO 配置为 AUX PORT_ID = 0x08，则使能将被忽略。
28-27	WU_CFG	R/W	0h	如果 DIO 配置为 GPIO 或非 AON 外设信号（PORT_ID 0x00 或 >0x08）： 00：无唤醒 01：无唤醒 10：如果该焊盘变为低电平，则从关断状态唤醒。 11：如果该焊盘变为高电平，则从关断状态唤醒。 如果 IO 配置为 AON 外设信号或 AUX PORT_ID 0x01-0x08，则该寄存器仅设置是否使能唤醒。 00、01：唤醒禁用 10、11：唤醒使能 极性通过 AON 寄存器进行控制。 注意：当 MSB 置位时，IOC 将停用 DIO 的输出使能。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-56. IOCFG26 寄存器字段说明（续）

位	字段	类型	复位	说明
26-24	IOMODE	R/W	0h	IO 模式 不适用于配置为 AON 外设信号和 AUX PORT_ID 0x01-0x08 的 IO AUX 有自己的 open_source/drain 配置。 0x2: 保留。未定义的行为。 0x3：保留。未定义的行为。 0h = NORMAL：正常输入/输出 1h = INV：反相输入/输出 4h = OPENDR： 开漏， 正常输入/输出 5h = OPENDR_INV：开漏 反相输入/输出 6h = OPENSRC：开源 正常输入/输出 7h = OPENSRC_INV：开源 反相输入/输出
23	IOEV_AON_PROG2_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG2 事件置为有效 1：输入边沿检测将 AON_PROG2 事件置为有效
22	IOEV_AON_PROG1_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG1 事件置为有效 1：输入边沿检测将 AON_PROG1 事件置为有效
21	IOEV_AON_PROG0_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG0 事件置为有效 1：输入边沿检测将 AON_PROG0 事件置为有效
20-19	RESERVED	R	0h	保留
18	EDGE_IRQ_EN	R/W	0h	0：无中断生成 1：使能该 IO 的中断生成（仅在 EDGE_DET 已使能时有效）
17-16	EDGE_DET	R/W	0h	使能该 IO 上的边沿检测事件生成 0h = NONE：无边沿检测 1h = 负边沿检测 2h = 正边沿检测 3h = 正负边沿检测
15	RESERVED	R	0h	保留
14-13	PULL_CTL	R/W	3h	上拉/下拉控制 1h = DWN：下拉 2h = UP：上拉 3h = DIS：无上拉/下拉
12	SLEW_RED	R/W	0h	0：标称压摆率 1：在输出驱动器中降低压摆率。
11-10	IOCURR	R/W	0h	选择该 IO 的 IO 电流模式。 0h = 2MA：低电流 (LC) 模式：IOSTR 设置为 AUTO 时最小值为 2mA 1h = 4MA：高电流 (HC) 模式：IOSTR 设置为 AUTO 时最小值为 4mA 2h = 4_8MA：扩展电流 (EC) 模式：当 IOSTR 设为 AUTO 时，双驱动强度 IO 的最小值为 8mA（标准 IO 的最小值为 4mA）

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-56. IOCFG26 寄存器字段说明（续）

位	字段	类型	复位	说明
9-8	IOSTR	R/W	0h	选择该 IO 的驱动强度控制源。 此设置控制低电流 (LC) 模式的驱动强度。可在 IOCURR 中选择更高的驱动强度 0h = 自动驱动强度，由 AON BATMON 根据电池电压控制。（VDDS 时最小 2mA） 1h = 最小驱动强度，由 AON_IOC:IOSTRMIN 控制（默认值为 3.3V 时最小 2mA） 2h = MED：中等驱动强度，由 AON_IOC:IOSTRMED 控制（默认值为 2.5V 时最小 2mA） 3h = 最大驱动强度，由 AON_IOC:IOSTRMAX 控制（默认值为 1.8V 时最小 2mA）
7	IOEV_RTC_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 RTC 事件置为有效 1：输入边沿检测将 RTC 事件置为有效
6	IOEV_MCU_WU_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 MCU_WU 事件置为有效 1：输入边沿检测将 MCU_WU 事件置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-56. IOCFG26 寄存器字段说明（续）

位	字段	类型	复位	说明
5-0	PORT_ID	R/W	0h	选择 DIO26 的用途 注意：除明确要求更改 PORT_ID 值的情况外，不应写入此字段。 0h =通用 IO 7h = AON 32 KHz 时钟 (SCLK_LF) 8h = AUX IO 9h = SSI0_RX：SSI0 RX Ah = SSI0_TX：SSI0 TX Bh = SSI0_FSS：SSI0 FSS Ch = SSI0_CLK：SSI0 CLK Dh = I2C_MSSDA：I2C 数据 Eh = I2C_MSSCL：I2C 时钟 Fh = UART0_RX：UART0 RX 10h = UART0_TX：UART0 TX 11h = UART0_CTS：UART0 CTS 12h = UART0_RTS：UART0 RTS 13h = UART1_RX：UART1 RX 14h = UART1_TX：UART1 TX 15h = UART1_CTS：UART1 CTS 16h = UART1_RTS：UART1 RTS 17h = PORT_EVENT0：端口事件 0 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 18h = PORT_EVENT1： 端口事件 1 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 19h = PORT_EVENT2： 端口事件 2 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ah = PORT_EVENT3： 端口事件 3 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Bh = PORT_EVENT4： 端口事件 4 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ch = PORT_EVENT5： 端口事件 5 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Dh = PORT_EVENT6： 端口事件 6 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Eh = PORT_EVENT7： 端口事件 7 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 20h = CPU_SWV： CPU SWV 21h = SSI1_RX：SSI1 RX 22h = SSI1_TX：SSI1 TX 23h = SSI1_FSS：SSI1 FSS 24h = SSI1_CLK：SSI1 CLK 25h = I2S_AD0：I2S 数据 0 26h = I2S_AD1： I2S 数据 1 27h = I2S_WCLK： I2S WCLK 28h = I2S_BCLK：I2S BCLK 29h = I2S_MCLK：I2S MCLK 2Eh = 射频内核跟踪

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-56. IOCFG26 寄存器字段说明（续）

位	字段	类型	复位	说明
				2Fh = 射频内核数据输出 0 30h = 射频内核数据输出 1 31h = 射频内核数据输出 2 32h = 射频内核数据输出 3 33h = 射频内核数据输入 0 34h = 射频内核数据输入 1 35h = 射频内核 SMI 数据链路输出 36h = 射频内核 SMI 数据链路输入 37h = 射频内核 SMI 命令链路输出 38h = 射频内核 SMI 命令链路输入

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.3.28 IOCFG27 寄存器（偏移 = 6Ch）[复位 = 00006000h]

图 13-50 展示了 IOCFG27，表 13-57 对其进行了介绍。

返回到汇总表。

DIO27 的配置

图 13-50. IOCFG27 寄存器

31	30	29	28	27	26	25	24
RESERVED	HYST_EN	IE	WU_CFG		IOMODE		
R-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h		
23	22	21	20	19	18	17	16
IOEV_AON_PROG2_EN	IOEV_AON_PROG1_EN	IOEV_AON_PROG0_EN	RESERVED		EDGE_IRQ_EN	EDGE_DET	
R/W-0h	R/W-0h	R/W-0h	R-0h		R/W-0h	R/W-0h	
15	14	13	12	11	10	9	8
RESERVED	PULL_CTL		SLEW_RED	IOCURR		IOSTR	
R-0h	R/W-3h		R/W-0h	R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
IOEV_RTC_EN	IOEV_MCU_WU_EN	PORT_ID					
R/W-0h	R/W-0h	R/W-0h					

表 13-57. IOCFG27 寄存器字段说明

位	字段	类型	复位	说明
31	RESERVED	R	0h	保留
30	HYST_EN	R/W	0h	0：输入迟滞禁用 1：输入迟滞使能
29	IE	R/W	0h	0：输入禁用 1：输入使能 注意：如果 IO 配置为 AUX PORT_ID = 0x08，则使能将被忽略。
28-27	WU_CFG	R/W	0h	如果 DIO 配置为 GPIO 或非 AON 外设信号（PORT_ID 0x00 或 >0x08）： 00：无唤醒 01：无唤醒 10：如果该焊盘变为低电平，则从关断状态唤醒。 11：如果该焊盘变为高电平，则从关断状态唤醒。 如果 IO 配置为 AON 外设信号或 AUX PORT_ID 0x01-0x08，则该寄存器仅设置是否使能唤醒。 00、01：唤醒禁用 10、11：唤醒使能 极性通过 AON 寄存器进行控制。 注意：当 MSB 置位时，IOC 将停用 DIO 的输出使能。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-57. IOCFG27 寄存器字段说明（续）

位	字段	类型	复位	说明
26-24	IOMODE	R/W	0h	IO 模式 不适用于配置为 AON 外设信号和 AUX PORT_ID 0x01-0x08 的 IO AUX 有自己的 open_source/drain 配置。 0x2: 保留。未定义的行为。 0x3：保留。未定义的行为。 0h = NORMAL：正常输入/输出 1h = INV：反相输入/输出 4h = OPENDR： 开漏， 正常输入/输出 5h = OPENDR_INV：开漏 反相输入/输出 6h = OPENSRC：开源 正常输入/输出 7h = OPENSRC_INV：开源 反相输入/输出
23	IOEV_AON_PROG2_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG2 事件置为有效 1：输入边沿检测将 AON_PROG2 事件置为有效
22	IOEV_AON_PROG1_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG1 事件置为有效 1：输入边沿检测将 AON_PROG1 事件置为有效
21	IOEV_AON_PROG0_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG0 事件置为有效 1：输入边沿检测将 AON_PROG0 事件置为有效
20-19	RESERVED	R	0h	保留
18	EDGE_IRQ_EN	R/W	0h	0：无中断生成 1：使能该 IO 的中断生成（仅在 EDGE_DET 已使能时有效）
17-16	EDGE_DET	R/W	0h	使能该 IO 上的边沿检测事件生成 0h = NONE：无边沿检测 1h = 负边沿检测 2h = 正边沿检测 3h = 正负边沿检测
15	RESERVED	R	0h	保留
14-13	PULL_CTL	R/W	3h	上拉/下拉控制 1h = DWN：下拉 2h = UP：上拉 3h = DIS：无上拉/下拉
12	SLEW_RED	R/W	0h	0：标称压摆率 1：在输出驱动器中降低压摆率。
11-10	IOCURR	R/W	0h	选择该 IO 的 IO 电流模式。 0h = 2MA：低电流 (LC) 模式：IOSTR 设置为 AUTO 时最小值为 2mA 1h = 4MA：高电流 (HC) 模式：IOSTR 设置为 AUTO 时最小值为 4mA 2h = 4_8MA：扩展电流 (EC) 模式：当 IOSTR 设为 AUTO 时，双驱动强度 IO 的最小值为 8mA（标准 IO 的最小值为 4mA）

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-57. IOCFG27 寄存器字段说明（续）

位	字段	类型	复位	说明
9-8	IOSTR	R/W	0h	选择该 IO 的驱动强度控制源。 此设置控制低电流 (LC) 模式的驱动强度。可在 IOCURR 中选择更高的驱动强度 0h = 自动驱动强度，由 AON BATMON 根据电池电压控制。（VDDS 时最小 2mA） 1h = 最小驱动强度，由 AON_IOC:IOSTRMIN 控制（默认值为 3.3V 时最小 2mA） 2h = MED：中等驱动强度，由 AON_IOC:IOSTRMED 控制（默认值为 2.5V 时最小 2mA） 3h = 最大驱动强度，由 AON_IOC:IOSTRMAX 控制（默认值为 1.8V 时最小 2mA）
7	IOEV_RTC_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 RTC 事件置为有效 1：输入边沿检测将 RTC 事件置为有效
6	IOEV_MCU_WU_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 MCU_WU 事件置为有效 1：输入边沿检测将 MCU_WU 事件置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-57. IOCFG27 寄存器字段说明（续）

位	字段	类型	复位	说明
5-0	PORT_ID	R/W	0h	选择 DIO27 的用途 注意：除明确要求更改 PORT_ID 值的情况外，不应写入此字段。 0h =通用 IO 7h = AON 32 KHz 时钟 (SCLK_LF) 8h = AUX IO 9h = SSI0_RX：SSI0 RX Ah = SSI0_TX：SSI0 TX Bh = SSI0_FSS：SSI0 FSS Ch = SSI0_CLK：SSI0 CLK Dh = I2C_MSSDA：I2C 数据 Eh = I2C_MSSCL：I2C 时钟 Fh = UART0_RX：UART0 RX 10h = UART0_TX：UART0 TX 11h = UART0_CTS：UART0 CTS 12h = UART0_RTS：UART0 RTS 13h = UART1_RX：UART1 RX 14h = UART1_TX：UART1 TX 15h = UART1_CTS：UART1 CTS 16h = UART1_RTS：UART1 RTS 17h = PORT_EVENT0：端口事件 0 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 18h = PORT_EVENT1： 端口事件 1 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 19h = PORT_EVENT2： 端口事件 2 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ah = PORT_EVENT3： 端口事件 3 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Bh = PORT_EVENT4： 端口事件 4 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ch = PORT_EVENT5： 端口事件 5 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Dh = PORT_EVENT6： 端口事件 6 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Eh = PORT_EVENT7： 端口事件 7 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 20h = CPU_SWV： CPU SWV 21h = SSI1_RX：SSI1 RX 22h = SSI1_TX：SSI1 TX 23h = SSI1_FSS：SSI1 FSS 24h = SSI1_CLK：SSI1 CLK 25h = I2S_AD0：I2S 数据 0 26h = I2S_AD1： I2S 数据 1 27h = I2S_WCLK： I2S WCLK 28h = I2S_BCLK：I2S BCLK 29h = I2S_MCLK：I2S MCLK 2Eh = 射频内核跟踪

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-57. IOCFG27 寄存器字段说明（续）

位	字段	类型	复位	说明
				2Fh = 射频内核数据输出 0 30h = 射频内核数据输出 1 31h = 射频内核数据输出 2 32h = 射频内核数据输出 3 33h = 射频内核数据输入 0 34h = 射频内核数据输入 1 35h = 射频内核 SMI 数据链路输出 36h = 射频内核 SMI 数据链路输入 37h = 射频内核 SMI 命令链路输出 38h = 射频内核 SMI 命令链路输入

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.3.29 IOCFG28 寄存器（偏移 = 70h）[复位 = 00006000h]

图 13-51 展示了 IOCFG28，表 13-58 对其进行了介绍。

返回到汇总表。

DIO28 的配置

图 13-51. IOCFG28 寄存器

31	30	29	28	27	26	25	24
RESERVED	HYST_EN	IE	WU_CFG		IOMODE		
R-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h		
23	22	21	20	19	18	17	16
IOEV_AON_PROG2_EN	IOEV_AON_PROG1_EN	IOEV_AON_PROG0_EN	RESERVED		EDGE_IRQ_EN	EDGE_DET	
R/W-0h	R/W-0h	R/W-0h	R-0h		R/W-0h	R/W-0h	
15	14	13	12	11	10	9	8
RESERVED	PULL_CTL		SLEW_RED	IOCURR		IOSTR	
R-0h	R/W-3h		R/W-0h	R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
IOEV_RTC_EN	IOEV_MCU_WU_EN	PORT_ID					
R/W-0h	R/W-0h	R/W-0h					

表 13-58. IOCFG28 寄存器字段说明

位	字段	类型	复位	说明
31	RESERVED	R	0h	保留
30	HYST_EN	R/W	0h	0：输入迟滞禁用 1：输入迟滞使能
29	IE	R/W	0h	0：输入禁用 1：输入使能 注意：如果 IO 配置为 AUX PORT_ID = 0x08，则使能将被忽略。
28-27	WU_CFG	R/W	0h	如果 DIO 配置为 GPIO 或非 AON 外设信号（PORT_ID 0x00 或 >0x08）： 00：无唤醒 01：无唤醒 10：如果该焊盘变为低电平，则从关断状态唤醒。 11：如果该焊盘变为高电平，则从关断状态唤醒。 如果 IO 配置为 AON 外设信号或 AUX PORT_ID 0x01-0x08，则该寄存器仅设置是否使能唤醒。 00、01：唤醒禁用 10、11：唤醒使能 极性通过 AON 寄存器进行控制。 注意：当 MSB 置位时，IOC 将停用 DIO 的输出使能。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-58. IOCFG28 寄存器字段说明（续）

位	字段	类型	复位	说明
26-24	IOMODE	R/W	0h	IO 模式 不适用于配置为 AON 外设信号和 AUX PORT_ID 0x01-0x08 的 IO AUX 有自己的 open_source/drain 配置。 0x2: 保留。未定义的行为。 0x3：保留。未定义的行为。 0h = NORMAL：正常输入/输出 1h = INV：反相输入/输出 4h = OPENDR： 开漏， 正常输入/输出 5h = OPENDR_INV：开漏 反相输入/输出 6h = OPENSRC：开源 正常输入/输出 7h = OPENSRC_INV：开源 反相输入/输出
23	IOEV_AON_PROG2_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG2 事件置为有效 1：输入边沿检测将 AON_PROG2 事件置为有效
22	IOEV_AON_PROG1_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG1 事件置为有效 1：输入边沿检测将 AON_PROG1 事件置为有效
21	IOEV_AON_PROG0_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG0 事件置为有效 1：输入边沿检测将 AON_PROG0 事件置为有效
20-19	RESERVED	R	0h	保留
18	EDGE_IRQ_EN	R/W	0h	0：无中断生成 1：使能该 IO 的中断生成（仅在 EDGE_DET 已使能时有效）
17-16	EDGE_DET	R/W	0h	使能该 IO 上的边沿检测事件生成 0h = NONE：无边沿检测 1h = 负边沿检测 2h = 正边沿检测 3h = 正负边沿检测
15	RESERVED	R	0h	保留
14-13	PULL_CTL	R/W	3h	上拉/下拉控制 1h = DWN：下拉 2h = UP：上拉 3h = DIS：无上拉/下拉
12	SLEW_RED	R/W	0h	0：标称压摆率 1：在输出驱动器中降低压摆率。
11-10	IOCURR	R/W	0h	选择该 IO 的 IO 电流模式。 0h = 2MA：低电流 (LC) 模式：IOSTR 设置为 AUTO 时最小值为 2mA 1h = 4MA：高电流 (HC) 模式：IOSTR 设置为 AUTO 时最小值为 4mA 2h = 4_8MA：扩展电流 (EC) 模式：当 IOSTR 设为 AUTO 时，双驱动强度 IO 的最小值为 8mA（标准 IO 的最小值为 4mA）

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-58. IOCFG28 寄存器字段说明（续）

位	字段	类型	复位	说明
9-8	IOSTR	R/W	0h	选择该 IO 的驱动强度控制源。 此设置控制低电流 (LC) 模式的驱动强度。可在 IOCURR 中选择更高的驱动强度 0h = 自动驱动强度，由 AON BATMON 根据电池电压控制。（VDDS 时最小 2mA） 1h = 最小驱动强度，由 AON_IOC:IOSTRMIN 控制（默认值为 3.3V 时最小 2mA） 2h = MED：中等驱动强度，由 AON_IOC:IOSTRMED 控制（默认值为 2.5V 时最小 2mA） 3h = 最大驱动强度，由 AON_IOC:IOSTRMAX 控制（默认值为 1.8V 时最小 2mA）
7	IOEV_RTC_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 RTC 事件置为有效 1：输入边沿检测将 RTC 事件置为有效
6	IOEV_MCU_WU_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 MCU_WU 事件置为有效 1：输入边沿检测将 MCU_WU 事件置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-58. IOCFG28 寄存器字段说明（续）

位	字段	类型	复位	说明
5-0	PORT_ID	R/W	0h	选择 DIO28 的用途 注意：除明确要求更改 PORT_ID 值的情况外，不应写入此字段。 0h =通用 IO 7h = AON 32 KHz 时钟 (SCLK_LF) 8h = AUX IO 9h = SSI0_RX：SSI0 RX Ah = SSI0_TX：SSI0 TX Bh = SSI0_FSS：SSI0 FSS Ch = SSI0_CLK：SSI0 CLK Dh = I2C_MSSDA：I2C 数据 Eh = I2C_MSSCL：I2C 时钟 Fh = UART0_RX：UART0 RX 10h = UART0_TX：UART0 TX 11h = UART0_CTS：UART0 CTS 12h = UART0_RTS：UART0 RTS 13h = UART1_RX：UART1 RX 14h = UART1_TX：UART1 TX 15h = UART1_CTS：UART1 CTS 16h = UART1_RTS：UART1 RTS 17h = PORT_EVENT0：端口事件 0 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 18h = PORT_EVENT1： 端口事件 1 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 19h = PORT_EVENT2： 端口事件 2 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ah = PORT_EVENT3： 端口事件 3 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Bh = PORT_EVENT4： 端口事件 4 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ch = PORT_EVENT5： 端口事件 5 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Dh = PORT_EVENT6： 端口事件 6 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Eh = PORT_EVENT7： 端口事件 7 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 20h = CPU_SWV： CPU SWV 21h = SSI1_RX：SSI1 RX 22h = SSI1_TX：SSI1 TX 23h = SSI1_FSS：SSI1 FSS 24h = SSI1_CLK：SSI1 CLK 25h = I2S_AD0：I2S 数据 0 26h = I2S_AD1： I2S 数据 1 27h = I2S_WCLK： I2S WCLK 28h = I2S_BCLK：I2S BCLK 29h = I2S_MCLK：I2S MCLK 2Eh = 射频内核跟踪

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-58. IOCFG28 寄存器字段说明（续）

位	字段	类型	复位	说明
				2Fh = 射频内核数据输出 0 30h = 射频内核数据输出 1 31h = 射频内核数据输出 2 32h = 射频内核数据输出 3 33h = 射频内核数据输入 0 34h = 射频内核数据输入 1 35h = 射频内核 SMI 数据链路输出 36h = 射频内核 SMI 数据链路输入 37h = 射频内核 SMI 命令链路输出 38h = 射频内核 SMI 命令链路输入

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.3.30 IOCFG29 寄存器（偏移 = 74h）[复位 = 00006000h]

图 13-52 展示了 IOCFG29，表 13-59 对其进行了介绍。

返回到汇总表。

DIO29 的配置

图 13-52. IOCFG29 寄存器

31	30	29	28	27	26	25	24
RESERVED	HYST_EN	IE	WU_CFG		IOMODE		
R-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h		
23	22	21	20	19	18	17	16
IOEV_AON_PROG2_EN	IOEV_AON_PROG1_EN	IOEV_AON_PROG0_EN	RESERVED		EDGE_IRQ_EN	EDGE_DET	
R/W-0h	R/W-0h	R/W-0h	R-0h		R/W-0h	R/W-0h	
15	14	13	12	11	10	9	8
RESERVED	PULL_CTL		SLEW_RED	IOCURR		IOSTR	
R-0h	R/W-3h		R/W-0h	R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
IOEV_RTC_EN	IOEV_MCU_WU_EN	PORT_ID					
R/W-0h	R/W-0h	R/W-0h					

表 13-59. IOCFG29 寄存器字段说明

位	字段	类型	复位	说明
31	RESERVED	R	0h	保留
30	HYST_EN	R/W	0h	0：输入迟滞禁用 1：输入迟滞使能
29	IE	R/W	0h	0：输入禁用 1：输入使能 注意：如果 IO 配置为 AUX PORT_ID = 0x08，则使能将被忽略。
28-27	WU_CFG	R/W	0h	如果 DIO 配置为 GPIO 或非 AON 外设信号（PORT_ID 0x00 或 >0x08）： 00：无唤醒 01：无唤醒 10：如果该焊盘变为低电平，则从关断状态唤醒。 11：如果该焊盘变为高电平，则从关断状态唤醒。 如果 IO 配置为 AON 外设信号或 AUX PORT_ID 0x01-0x08，则该寄存器仅设置是否使能唤醒。 00、01：唤醒禁用 10、11：唤醒使能 极性通过 AON 寄存器进行控制。 注意：当 MSB 置位时，IOC 将停用 DIO 的输出使能。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-59. IOCFG29 寄存器字段说明（续）

位	字段	类型	复位	说明
26-24	IOMODE	R/W	0h	IO 模式 不适用于配置为 AON 外设信号和 AUX PORT_ID 0x01-0x08 的 IO AUX 有自己的 open_source/drain 配置。 0x2: 保留。未定义的行为。 0x3：保留。未定义的行为。 0h = NORMAL：正常输入/输出 1h = INV：反相输入/输出 4h = OPENDR： 开漏， 正常输入/输出 5h = OPENDR_INV：开漏 反相输入/输出 6h = OPENSRC：开源 正常输入/输出 7h = OPENSRC_INV：开源 反相输入/输出
23	IOEV_AON_PROG2_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG2 事件置为有效 1：输入边沿检测将 AON_PROG2 事件置为有效
22	IOEV_AON_PROG1_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG1 事件置为有效 1：输入边沿检测将 AON_PROG1 事件置为有效
21	IOEV_AON_PROG0_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG0 事件置为有效 1：输入边沿检测将 AON_PROG0 事件置为有效
20-19	RESERVED	R	0h	保留
18	EDGE_IRQ_EN	R/W	0h	0：无中断生成 1：使能该 IO 的中断生成（仅在 EDGE_DET 已使能时有效）
17-16	EDGE_DET	R/W	0h	使能该 IO 上的边沿检测事件生成 0h = NONE：无边沿检测 1h = 负边沿检测 2h = 正边沿检测 3h = 正负边沿检测
15	RESERVED	R	0h	保留
14-13	PULL_CTL	R/W	3h	上拉/下拉控制 1h = DWN：下拉 2h = UP：上拉 3h = DIS：无上拉/下拉
12	SLEW_RED	R/W	0h	0：标称压摆率 1：在输出驱动器中降低压摆率。
11-10	IOCURR	R/W	0h	选择该 IO 的 IO 电流模式。 0h = 2MA：低电流 (LC) 模式：IOSTR 设置为 AUTO 时最小值为 2mA 1h = 4MA：高电流 (HC) 模式：IOSTR 设置为 AUTO 时最小值为 4mA 2h = 4_8MA：扩展电流 (EC) 模式：当 IOSTR 设为 AUTO 时，双驱动强度 IO 的最小值为 8mA（标准 IO 的最小值为 4mA）

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-59. IOCFG29 寄存器字段说明（续）

位	字段	类型	复位	说明
9-8	IOSTR	R/W	0h	选择该 IO 的驱动强度控制源。 此设置控制低电流 (LC) 模式的驱动强度。可在 IOCURR 中选择更高的驱动强度 0h = 自动驱动强度，由 AON BATMON 根据电池电压控制。（VDDS 时最小 2mA） 1h = 最小驱动强度，由 AON_IOC:IOSTRMIN 控制（默认值为 3.3V 时最小 2mA） 2h = MED：中等驱动强度，由 AON_IOC:IOSTRMED 控制（默认值为 2.5V 时最小 2mA） 3h = 最大驱动强度，由 AON_IOC:IOSTRMAX 控制（默认值为 1.8V 时最小 2mA）
7	IOEV_RTC_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 RTC 事件置为有效 1：输入边沿检测将 RTC 事件置为有效
6	IOEV_MCU_WU_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 MCU_WU 事件置为有效 1：输入边沿检测将 MCU_WU 事件置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-59. IOCFG29 寄存器字段说明 （续）

位	字段	类型	复位	说明
5-0	PORT_ID	R/W	0h	选择 DIO29 的用途 注意：除明确要求更改 PORT_ID 值的情况外，不应写入此字段。 0h =通用 IO 7h = AON 32 KHz 时钟 (SCLK_LF) 8h = AUX IO 9h = SSI0_RX：SSI0 RX Ah = SSI0_TX：SSI0 TX Bh = SSI0_FSS：SSI0 FSS Ch = SSI0_CLK：SSI0 CLK Dh = I2C_MSSDA：I2C 数据 Eh = I2C_MSSCL：I2C 时钟 Fh = UART0_RX：UART0 RX 10h = UART0_TX：UART0 TX 11h = UART0_CTS：UART0 CTS 12h = UART0_RTS：UART0 RTS 13h = UART1_RX：UART1 RX 14h = UART1_TX：UART1 TX 15h = UART1_CTS：UART1 CTS 16h = UART1_RTS：UART1 RTS 17h = PORT_EVENT0：端口事件 0 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等）， 可用作通用 IO 事件 18h = PORT_EVENT1： 端口事件 1 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等）， 可用作通用 IO 事件 19h = PORT_EVENT2： 端口事件 2 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等）， 可用作通用 IO 事件 1Ah = PORT_EVENT3： 端口事件 3 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等）， 可用作通用 IO 事件 1Bh = PORT_EVENT4： 端口事件 4 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等）， 可用作通用 IO 事件 1Ch = PORT_EVENT5： 端口事件 5 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等）， 可用作通用 IO 事件 1Dh = PORT_EVENT6： 端口事件 6 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等）， 可用作通用 IO 事件 1Eh = PORT_EVENT7： 端口事件 7 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等）， 可用作通用 IO 事件 20h = CPU_SWV： CPU SWV 21h = SSI1_RX：SSI1 RX 22h = SSI1_TX：SSI1 TX 23h = SSI1_FSS：SSI1 FSS 24h = SSI1_CLK：SSI1 CLK 25h = I2S_AD0：I2S 数据 0 26h = I2S_AD1： I2S 数据 1 27h = I2S_WCLK： I2S WCLK 28h = I2S_BCLK：I2S BCLK 29h = I2S_MCLK：I2S MCLK 2Eh = 射频内核跟踪

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-59. IOCFG29 寄存器字段说明（续）

位	字段	类型	复位	说明
				2Fh = 射频内核数据输出 0 30h = 射频内核数据输出 1 31h = 射频内核数据输出 2 32h = 射频内核数据输出 3 33h = 射频内核数据输入 0 34h = 射频内核数据输入 1 35h = 射频内核 SMI 数据链路输出 36h = 射频内核 SMI 数据链路输入 37h = 射频内核 SMI 命令链路输出 38h = 射频内核 SMI 命令链路输入

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.3.31 IOCFG30 寄存器（偏移 = 78h）[复位 = 00006000h]

图 13-53 展示了 IOCFG30，表 13-60 对其进行了介绍。

返回到汇总表。

DIO30 的配置

图 13-53. IOCFG30 寄存器

31	30	29	28	27	26	25	24
RESERVED	HYST_EN	IE	WU_CFG		IOMODE		
R-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h		
23	22	21	20	19	18	17	16
IOEV_AON_PROG2_EN	IOEV_AON_PROG1_EN	IOEV_AON_PROG0_EN	RESERVED		EDGE_IRQ_EN	EDGE_DET	
R/W-0h	R/W-0h	R/W-0h	R-0h		R/W-0h	R/W-0h	
15	14	13	12	11	10	9	8
RESERVED	PULL_CTL		SLEW_RED	IOCURR		IOSTR	
R-0h	R/W-3h		R/W-0h	R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
IOEV_RTC_EN	IOEV_MCU_WU_EN	PORT_ID					
R/W-0h	R/W-0h	R/W-0h					

表 13-60. IOCFG30 寄存器字段说明

位	字段	类型	复位	说明
31	RESERVED	R	0h	保留
30	HYST_EN	R/W	0h	0：输入迟滞禁用 1：输入迟滞使能
29	IE	R/W	0h	0：输入禁用 1：输入使能 注意：如果 IO 配置为 AUX PORT_ID = 0x08，则使能将被忽略。
28-27	WU_CFG	R/W	0h	如果 DIO 配置为 GPIO 或非 AON 外设信号（PORT_ID 0x00 或 >0x08）： 00：无唤醒 01：无唤醒 10：如果该焊盘变为低电平，则从关断状态唤醒。 11：如果该焊盘变为高电平，则从关断状态唤醒。 如果 IO 配置为 AON 外设信号或 AUX PORT_ID 0x01-0x08，则该寄存器仅设置是否使能唤醒。 00、01：唤醒禁用 10、11：唤醒使能 极性通过 AON 寄存器进行控制。 注意：当 MSB 置位时，IOC 将停用 DIO 的输出使能。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-60. IOCFG30 寄存器字段说明（续）

位	字段	类型	复位	说明
26-24	IOMODE	R/W	0h	IO 模式 不适用于配置为 AON 外设信号和 AUX PORT_ID 0x01-0x08 的 IO AUX 有自己的 open_source/drain 配置。 0x2: 保留。未定义的行为。 0x3：保留。未定义的行为。 0h = NORMAL：正常输入/输出 1h = INV： 反相输入/输出 4h = OPENDR： 开漏， 正常输入/输出 5h = OPENDR_INV：开漏 反相输入/输出 6h = OPENSRC：开源 正常输入/输出 7h = OPENSRC_INV：开源 反相输入/输出
23	IOEV_AON_PROG2_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG2 事件置为有效 1：输入边沿检测将 AON_PROG2 事件置为有效
22	IOEV_AON_PROG1_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG1 事件置为有效 1：输入边沿检测将 AON_PROG1 事件置为有效
21	IOEV_AON_PROG0_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG0 事件置为有效 1：输入边沿检测将 AON_PROG0 事件置为有效
20-19	RESERVED	R	0h	保留
18	EDGE_IRQ_EN	R/W	0h	0：无中断生成 1：使能该 IO 的中断生成（仅在 EDGE_DET 已使能时有效）
17-16	EDGE_DET	R/W	0h	使能该 IO 上的边沿检测事件生成 0h = NONE：无边沿检测 1h = 负边沿检测 2h = 正边沿检测 3h = 正负边沿检测
15	RESERVED	R	0h	保留
14-13	PULL_CTL	R/W	3h	上拉/下拉控制 1h = DWN：下拉 2h = UP：上拉 3h = DIS：无上拉/下拉
12	SLEW_RED	R/W	0h	0：标称压摆率 1：在输出驱动器中降低压摆率。
11-10	IOCURR	R/W	0h	选择该 IO 的 IO 电流模式。 0h = 2MA：低电流 (LC) 模式：IOSTR 设置为 AUTO 时最小值为 2mA 1h = 4MA：高电流 (HC) 模式：IOSTR 设置为 AUTO 时最小值为 4mA 2h = 4_8MA：扩展电流 (EC) 模式：当 IOSTR 设为 AUTO 时，双驱动强度 IO 的最小值为 8mA（标准 IO 的最小值为 4mA）

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-60. IOCFG30 寄存器字段说明（续）

位	字段	类型	复位	说明
9-8	IOSTR	R/W	0h	选择该 IO 的驱动强度控制源。 此设置控制低电流 (LC) 模式的驱动强度。可在 IOCURR 中选择更高的驱动强度 0h = 自动驱动强度，由 AON BATMON 根据电池电压控制。（VDDS 时最小 2mA） 1h = 最小驱动强度，由 AON_IOC:IOSTRMIN 控制（默认值为 3.3V 时最小 2mA） 2h = MED：中等驱动强度，由 AON_IOC:IOSTRMED 控制（默认值为 2.5V 时最小 2mA） 3h = 最大驱动强度，由 AON_IOC:IOSTRMAX 控制（默认值为 1.8V 时最小 2mA）
7	IOEV_RTC_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 RTC 事件置为有效 1：输入边沿检测将 RTC 事件置为有效
6	IOEV_MCU_WU_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 MCU_WU 事件置为有效 1：输入边沿检测将 MCU_WU 事件置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-60. IOCFG30 寄存器字段说明（续）

位	字段	类型	复位	说明
5-0	PORT_ID	R/W	0h	选择 DIO30 的用途 注意：除明确要求更改 PORT_ID 值的情况外，不应写入此字段。 0h =通用 IO 7h = AON 32 KHz 时钟 (SCLK_LF) 8h = AUX IO 9h = SSI0_RX：SSI0 RX Ah = SSI0_TX：SSI0 TX Bh = SSI0_FSS：SSI0 FSS Ch = SSI0_CLK：SSI0 CLK Dh = I2C_MSSDA：I2C 数据 Eh = I2C_MSSCL：I2C 时钟 Fh = UART0_RX：UART0 RX 10h = UART0_TX：UART0 TX 11h = UART0_CTS：UART0 CTS 12h = UART0_RTS：UART0 RTS 13h = UART1_RX：UART1 RX 14h = UART1_TX：UART1 TX 15h = UART1_CTS：UART1 CTS 16h = UART1_RTS：UART1 RTS 17h = PORT_EVENT0：端口事件 0 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 18h = PORT_EVENT1： 端口事件 1 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 19h = PORT_EVENT2： 端口事件 2 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ah = PORT_EVENT3： 端口事件 3 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Bh = PORT_EVENT4： 端口事件 4 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ch = PORT_EVENT5： 端口事件 5 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Dh = PORT_EVENT6： 端口事件 6 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Eh = PORT_EVENT7： 端口事件 7 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 20h = CPU_SWV： CPU SWV 21h = SSI1_RX：SSI1 RX 22h = SSI1_TX：SSI1 TX 23h = SSI1_FSS：SSI1 FSS 24h = SSI1_CLK：SSI1 CLK 25h = I2S_AD0：I2S 数据 0 26h = I2S_AD1： I2S 数据 1 27h = I2S_WCLK： I2S WCLK 28h = I2S_BCLK：I2S BCLK 29h = I2S_MCLK：I2S MCLK 2Eh = 射频内核跟踪

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-60. IOCFG30 寄存器字段说明（续）

位	字段	类型	复位	说明
				2Fh = 射频内核数据输出 0 30h = 射频内核数据输出 1 31h = 射频内核数据输出 2 32h = 射频内核数据输出 3 33h = 射频内核数据输入 0 34h = 射频内核数据输入 1 35h = 射频内核 SMI 数据链路输出 36h = 射频内核 SMI 数据链路输入 37h = 射频内核 SMI 命令链路输出 38h = 射频内核 SMI 命令链路输入

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

13.10.3.32 IOCFG31 寄存器（偏移 = 7Ch）[复位 = 00006000h]

图 13-54 展示了 IOCFG31，表 13-61 对其进行了介绍。

返回到汇总表。

DIO31 的配置

图 13-54. IOCFG31 寄存器

31	30	29	28	27	26	25	24
RESERVED	HYST_EN	IE	WU_CFG		IOMODE		
R-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h		
23	22	21	20	19	18	17	16
IOEV_AON_PROG2_EN	IOEV_AON_PROG1_EN	IOEV_AON_PROG0_EN	RESERVED		EDGE_IRQ_EN	EDGE_DET	
R/W-0h	R/W-0h	R/W-0h	R-0h		R/W-0h	R/W-0h	
15	14	13	12	11	10	9	8
RESERVED	PULL_CTL		SLEW_RED	IOCURR		IOSTR	
R-0h	R/W-3h		R/W-0h	R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
IOEV_RTC_EN	IOEV_MCU_WU_EN	PORT_ID					
R/W-0h	R/W-0h	R/W-0h					

表 13-61. IOCFG31 寄存器字段说明

位	字段	类型	复位	说明
31	RESERVED	R	0h	保留
30	HYST_EN	R/W	0h	0：输入迟滞禁用 1：输入迟滞使能
29	IE	R/W	0h	0：输入禁用 1：输入使能 注意：如果 IO 配置为 AUX PORT_ID = 0x08，则使能将被忽略。
28-27	WU_CFG	R/W	0h	如果 DIO 配置为 GPIO 或非 AON 外设信号（PORT_ID 0x00 或 >0x08）： 00：无唤醒 01：无唤醒 10：如果该焊盘变为低电平，则从关断状态唤醒。 11：如果该焊盘变为高电平，则从关断状态唤醒。 如果 IO 配置为 AON 外设信号或 AUX PORT_ID 0x01-0x08，则该寄存器仅设置是否使能唤醒。 00、01：唤醒禁用 10、11：唤醒使能 极性通过 AON 寄存器进行控制。 注意：当 MSB 置位时，IOC 将停用 DIO 的输出使能。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-61. IOCFG31 寄存器字段说明（续）

位	字段	类型	复位	说明
26-24	IOMODE	R/W	0h	IO 模式 不适用于配置为 AON 外设信号和 AUX PORT_ID 0x01-0x08 的 IO AUX 有自己的 open_source/drain 配置。 0x2: 保留。未定义的行为。 0x3：保留。未定义的行为。 0h = NORMAL：正常输入/输出 1h = INV：反相输入/输出 4h = OPENDR： 开漏， 正常输入/输出 5h = OPENDR_INV：开漏 反相输入/输出 6h = OPENSRC：开源 正常输入/输出 7h = OPENSRC_INV：开源 反相输入/输出
23	IOEV_AON_PROG2_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG2 事件置为有效 1：输入边沿检测将 AON_PROG2 事件置为有效
22	IOEV_AON_PROG1_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG1 事件置为有效 1：输入边沿检测将 AON_PROG1 事件置为有效
21	IOEV_AON_PROG0_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 AON_PROG0 事件置为有效 1：输入边沿检测将 AON_PROG0 事件置为有效
20-19	RESERVED	R	0h	保留
18	EDGE_IRQ_EN	R/W	0h	0：无中断生成 1：使能该 IO 的中断生成（仅在 EDGE_DET 已使能时有效）
17-16	EDGE_DET	R/W	0h	使能该 IO 上的边沿检测事件生成 0h = NONE：无边沿检测 1h = 负边沿检测 2h = 正边沿检测 3h = 正负边沿检测
15	RESERVED	R	0h	保留
14-13	PULL_CTL	R/W	3h	上拉/下拉控制 1h = DWN：下拉 2h = UP：上拉 3h = DIS：无上拉/下拉
12	SLEW_RED	R/W	0h	0：标称压摆率 1：在输出驱动器中降低压摆率。
11-10	IOCURR	R/W	0h	选择该 IO 的 IO 电流模式。 0h = 2MA：低电流 (LC) 模式：IOSTR 设置为 AUTO 时最小值为 2mA 1h = 4MA：高电流 (HC) 模式：IOSTR 设置为 AUTO 时最小值为 4mA 2h = 4_8MA：扩展电流 (EC) 模式：当 IOSTR 设为 AUTO 时，双驱动强度 IO 的最小值为 8mA（标准 IO 的最小值为 4mA）

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-61. IOCFG31 寄存器字段说明（续）

位	字段	类型	复位	说明
9-8	IOSTR	R/W	0h	选择该 IO 的驱动强度控制源。 此设置控制低电流 (LC) 模式的驱动强度。可在 IOCURR 中选择更高的驱动强度 0h = 自动驱动强度，由 AON BATMON 根据电池电压控制。（VDDS 时最小 2mA） 1h = 最小驱动强度，由 AON_IOC:IOSTRMIN 控制（默认值为 3.3V 时最小 2mA） 2h = MED：中等驱动强度，由 AON_IOC:IOSTRMED 控制（默认值为 2.5V 时最小 2mA） 3h = 最大驱动强度，由 AON_IOC:IOSTRMAX 控制（默认值为 1.8V 时最小 2mA）
7	IOEV_RTC_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 RTC 事件置为有效 1：输入边沿检测将 RTC 事件置为有效
6	IOEV_MCU_WU_EN	R/W	0h	使能边沿检测时，此 IO 置为有效的事件 0：输入边沿检测不会将 MCU_WU 事件置为有效 1：输入边沿检测将 MCU_WU 事件置为有效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-61. IOCFG31 寄存器字段说明（续）

位	字段	类型	复位	说明
5-0	PORT_ID	R/W	0h	选择 DIO31 的用途 注意：除明确要求更改 PORT_ID 值的情况外，不应写入此字段。 0h = 通用 IO 注意：SIP 模块不支持此功能。有关更多详情，请参阅器件特定数据表。 7h = AON 32KHz 时钟 (SCLK_LF) 8h = AUX IO 9h = SSI0_RX： SSI0 RX Ah = SSI0_TX：SSI0 TX Bh = SSI0_FSS：SSI0 FSS Ch = SSI0_CLK：SSI0 CLK Dh = I2C_MSSDA：I2C 数据 Eh = I2C_MSSCL：I2C 时钟 Fh = UART0_RX：UART0 RX 10h = UART0_TX：UART0 TX 11h = UART0_CTS：UART0 CTS 12h = UART0_RTS：UART0 RTS 13h = UART1_RX：UART1 RX 14h = UART1_TX：UART1 TX 15h = UART1_CTS：UART1 CTS 16h = UART1_RTS：UART1 RTS 17h = PORT_EVENT0：端口事件 0 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 18h = PORT_EVENT1： 端口事件 1 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 19h = PORT_EVENT2： 端口事件 2 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ah = PORT_EVENT3： 端口事件 3 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Bh = PORT_EVENT4： 端口事件 4 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Ch = PORT_EVENT5： 端口事件 5 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Dh = PORT_EVENT6： 端口事件 6 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 1Eh = PORT_EVENT7： 端口事件 7 通过 EVENT 模块中的寄存器进行选择（例如 EVENT:GPT0ACAPTSEL.EV、EVENT:UDMACH14BSEL.EV 等），可用作通用 IO 事件 20h = CPU_SWV： CPU SWV 21h = SSI1_RX：SSI1 RX 22h = SSI1_TX：SSI1 TX 23h = SSI1_FSS：SSI1 FSS 24h = SSI1_CLK：SSI1 CLK 25h = I2S_AD0：I2S 数据 0 26h = I2S_AD1： I2S 数据 1 27h = I2S_WCLK： I2S WCLK 28h = I2S_BCLK：I2S BCLK

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 13-61. IOCFG31 寄存器字段说明（续）

位	字段	类型	复位	说明
				29h = I2S_MCLK：I2S MCLK 2Eh = 射频内核跟踪 2Fh = 射频内核数据输出 0 30h = 射频内核数据输出 1 31h = 射频内核数据输出 2 32h = 射频内核数据输出 3 33h = 射频内核数据输入 0 34h = 射频内核数据输入 1 35h = 射频内核 SMI 数据链路输出 36h = 射频内核 SMI 数据链路输入 37h = 射频内核 SMI 命令链路输出 38h = 射频内核 SMI 命令链路输入

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

章节 14

微型直接存储器存取 (µDMA)

本章介绍了称为 µDMA 的直接存储器存取 (DMA) 控制器。

14.1 µDMA 简介.....1287
14.2 方框图.....1288
14.3 功能说明.....1288
14.4 初始化和配置.....1303
14.5 µDMA 寄存器.....1304

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

14.1 µDMA 简介

CC13x2 和 CC26x2 器件平台包含一个直接存储器存取 (DMA) 控制器，称为 µDMA。 µDMA 控制器以某种方式减少 Arm® Cortex®-M4F 处理器参与的数据传输任务，从而更加高效地使用处理器以及可用的总线带宽。 µDMA 控制器能够执行存储器与外设之间的数据传输。该控制器为每个支持的片上模块提供专用的通道，并且经编程可在外设准备好传输更多数据时自动执行外设和存储器之间的数据传输。 µDMA 控制器具有以下特性：

- Arm PrimeCell® 32 通道可配置 µDMA 控制器
- 支持存储器到存储器、存储器到外设以及外设到存储器的多种传输模式：
  - 基本模式，用于简单的传输场景
  - 乒乓模式，用于连续数据流
  - 散聚模式，用于可编程的任意传输列表，由单个请求启动
- 高度灵活且可配置的通道运行：
  - 独立配置和运行的通道
  - 为支持的片上模块提供专用通道
  - 主通道和辅助通道分配
  - 灵活的通道分配
  - 对于双向模块，接收和发送路径各占一个通道
  - 用于软件发起传输的专用通道
  - 每通道可配置的优先级方案
  - 任意通道的可选软件发起请求
- 两个级别的优先级
- 支持 8 位、16 位和 32 位数据宽度
- 传输大小可从 1 到 1024 以二进制步长编程
- 源地址和目标地址可递增，递增单位可以是字节、半字、字或不递增
- 可屏蔽的外设请求
- 传输完成中断，每个通道有独立的中断

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

14.2 方框图

图 14-1 显示了 µDMA 方框图。

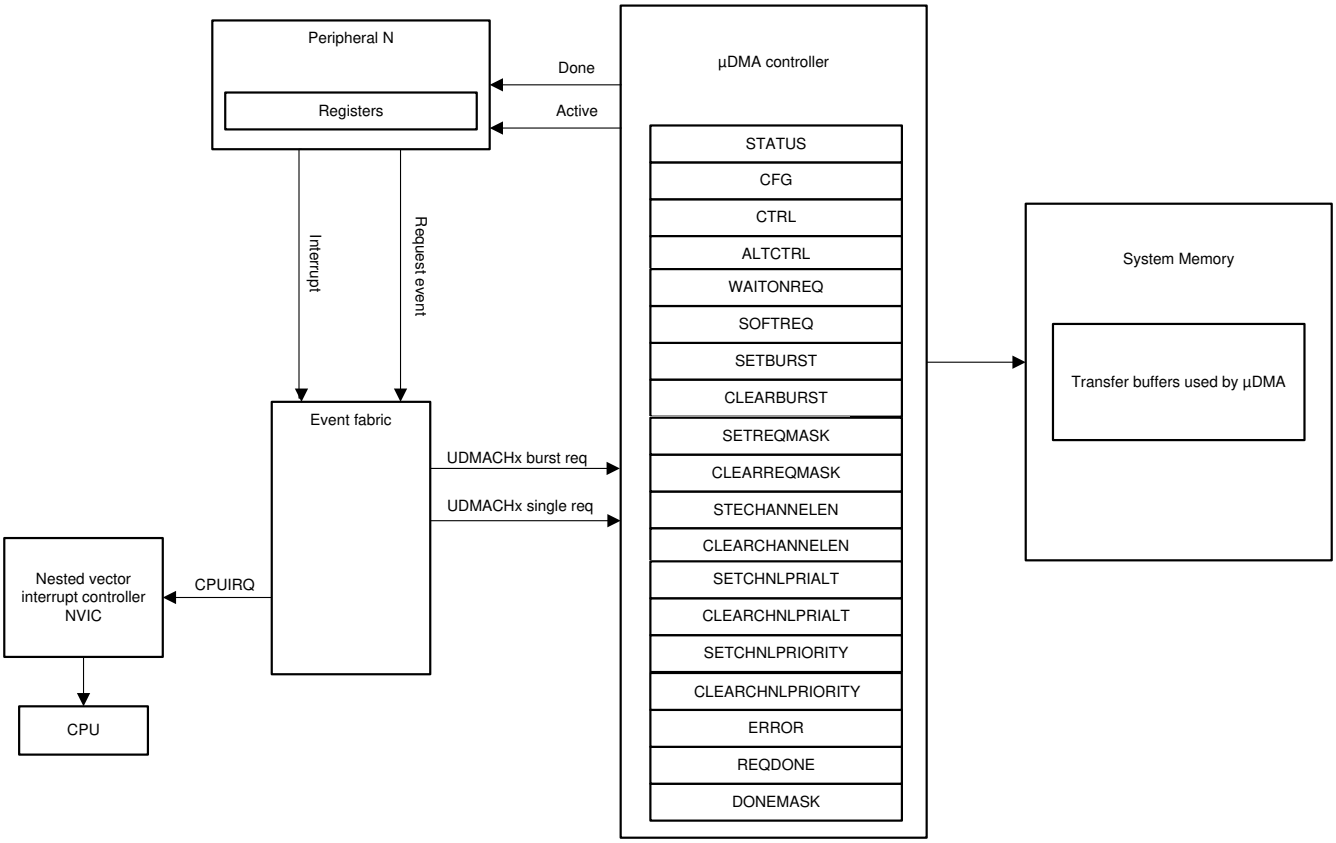


图 14-1. µDMA 方框图

14.3 功能说明

µDMA 控制器是一个灵活且高度可配置的 DMA 控制器，旨在与微控制器的 Arm® Cortex®-M4F 处理器内核高效协同工作。该控制器支持多种数据大小和地址增量方案、DMA 通道间的多级优先级，以及多种传输模式，以实现复杂的编程数据传输。

每个支持的外设功能在 µDMA 控制器上都有一个专用通道，可以独立配置。µDMA 控制器使用由处理器在系统内存中维护的通道控制结构，实现了一种配置方法。除了支持简单的传输模式外，还可以在内存中构建复杂的任务列表，从而允许 µDMA 控制器在单个传输请求中执行任意大小的、往返于任意位置的数据传输。µDMA 控制器还支持使用乒乓缓冲，以适应进出外设的持续数据流。

每个通道还具有可配置的仲裁大小。仲裁大小是在 µDMA 控制器请求通道优先级之前，以突发方式传输的数据项数量。通过使用仲裁大小，可以精确控制在每次发出 µDMA 服务请求时，有多少数据项传入或传出外设。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

14.3.1 通道分配

表 14-1 列出了外设的 µDMA 通道分配。

表 14-1. 通道分配

通道	外设	waitonreq	map _wiatonreq	stall	dma _done	dma _active	DMA_CHANNEL _WITH_2STAGE_SYNC	DMA _ACTIVE_FF	DMA_CHANNEL _ASYNC
21-31	保留	1			是	是	0		0
20(1)	软件 3.	1					0		0
19(1)	软件 2.	1					0		0
18(1)	软件 1.	1			是		0		0
17	SSP1_TX	1			是	是	0		0
16	SSP1_RX	1			是	是	0		0
15	AON_RTC	0					0		1
14	DMA_PROG	0					0		1
13	AON_PROG2	0					0		1
12	GPT1_B	1			是		1		0
11	GPT1_A	1			是		1		0
10	GPT0_B	1			是		1		0
9	GPT0_A	1			是		1		0
8	AUX_SW	0					0		1
7	AUX_ADC	1			是	是	0	1	1
6	UART1_TX	1			是	是	0		0
5	UART1_RX	1			是	是	0		0
4	SSP0_TX	1			是	是	0		0
3	SSP0_RX	1			是	是	0		0
2	UART0_TX	1			是	是	0		0
1	UART0_RX	1			是	是	0		0
0(1)	软件 0.	1		是	是		0		0

(1) DMA 软件触发器

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

14.3.2 优先级

µDMA 控制器根据通道编号和该通道的优先级位来为每个通道分配优先级。通道 0 具有最高优先级，随着通道编号的增加，通道优先级降低。每个通道都有一个优先级位，以提供两个级别的优先级：默认优先级和高优先级。如果设置了优先级位，则该通道的优先级高于所有处于默认优先级的其他通道。如果多个通道被设置为高优先级，则使用通道编号来确定所有高优先级通道之间的相对优先级。

可以使用 UDMA:SETCHNLPRIORITY 寄存器设置通道的优先级位，并使用 UDMA:CLEARCHNLPRIORITY 寄存器清除该位（请参阅节 14.5.1）。

14.3.3 仲裁大小

当 µDMA 通道请求传输时，µDMA 控制器在所有发出请求的通道中进行仲裁，并为优先级最高的 µDMA 通道提供服务。一旦传输开始，它会持续进行可选择次数的传输，然后才在请求通道之间重新进行仲裁。每个通道的仲裁大小可以配置，范围为 1 到 1024 个数据项的传输。当 µDMA 控制器传输了由仲裁大小指定的数据项数量后，控制器会在所有发出请求的通道中进行检查，并为优先级最高的通道提供服务。

如果优先级较低的 µDMA 通道使用较大的仲裁大小，则高优先级通道的延迟会增加，因为 µDMA 控制器在检查高优先级请求之前会先完成低优先级的突发传输。因此，为了在高优先级通道上获得最佳响应，低优先级通道不应使用较大的仲裁大小。

仲裁大小也可以理解为突发大小。仲裁大小是在一次突发传输中任何时间点传输的最大数据项数量。这里的术语*仲裁*指的是确定 µDMA 通道的优先级，而不是总线的仲裁。当 µDMA 控制器为总线进行仲裁时，处理器始终优先。此外，只要处理器必须在同一总线上执行总线事务，µDMA 控制器就会被延迟，即使在突发传输过程中也是如此。

14.3.4 请求类型

µDMA 控制器响应来自外设的两种类型的请求：单个请求或突发请求。每个外设可能支持其中一种或两种请求类型。单次请求意味着外设准备好传输一个数据项，而突发请求意味着外设准备好传输多个数据项。

µDMA 控制器的响应方式取决于外设发出的是单次请求还是突发请求。如果两种类型的请求都被置为有效，并且 µDMA 通道已设置为突发传输，则突发请求优先。表 14-2 列出了每个外设对这两种请求类型的支持情况。

表 14-2. 请求类型支持

外设	单次请求信号	突发请求信号
ADC	无（FIFO 不为空）	序列发生器 IE 位（FIFO 半满）
通用计时器	原始中断脉冲	无
GPIO	原始中断脉冲	无
SSI TX	TX FIFO 未满	TX FIFO 水平（固定为 4）
SSI RX	RX FIFO 不为空	RX FIFO 水平（固定为 4）
UART TX	TX FIFO 未满	TX FIFO 水平（可配置）
UART RX	RX FIFO 不为空	RX FIFO 水平（可配置）

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

14.3.4.1 单个请求

检测到单个请求（不是突发请求）时，µDMA 控制器会在传送一个项目后停止，等待另一个请求。

备注

通道 8、13、14 和 15 不响应单个请求，因为 waitonreq 连接至低电平。

14.3.4.2 突发请求

当检测到突发请求时，µDMA 控制器传输的数据项数量为以下两者中的较小者：仲裁大小或传输中剩余的数据项数量。因此，仲裁大小必须等于外设在发出突发请求时能够容纳的数据项数量。例如，混合使用单次请求或突发请求的 UART 和 SPI 可以根据 FIFO 触发电平生成突发请求。在这种情况下，仲裁大小必须设置为达到触发电平时 FIFO 可以传输的数据量。突发传输一旦启动就会运行到完成，即使有更高优先级的通道也无法中断它。与相同数量的非突发传输相比，突发传输可以在更短的时间内完成。

在某些情况下，可能希望仅使用突发传输而不允许单次传输（例如，当数据的性质决定其作为一个整体单元传输比每次传输一块更有意义时）。可以通过 UDMA:SETBURST 寄存器禁用单次请求。通过为该寄存器中的某个通道设置位，µDMA 控制器将仅响应该通道的突发请求。

14.3.5 通道配置

µDMA 控制器使用系统内存中的一个区域来存储一组通道控制结构表。控制表中的每个 µDMA 通道可能有一个或两个条目。表结构中的每个条目都包含源和目标指针、传输大小和传输模式。控制表可以位于系统内存中的任何位置，但它必须是连续的，并且按 1024 字节边界对齐。

表 14-3 描述了通道控制表的内存布局。每个通道在控制表中可以有一个或两个控制结构：一个主控制结构和一个可选的备用控制结构。该表的组织方式是：所有主条目位于表的前半部分，所有备用结构位于表的后半部分。主条目用于简单传输模式，在该模式下，每次传输完成后都可以重新配置和重新启动传输。在这种情况下，不会使用备用控制结构，因此只需在内存中分配表的前半部分；控制表的后半部分不是必需的，该内存可用于其他用途。如果使用更复杂的传输模式（例如乒乓或散聚模式），则还会使用备用控制结构，并且必须为整个表分配内存空间。

控制表中任何未使用的内存均可由应用程序使用，这包括应用程序未使用的任何通道的控制结构，以及每个通道的未使用控制字。

表 14-3. 控制结构内存映射

偏移	通道
0x0	0，主结构
0x10	1，主结构
...	...
0x1F0	31，主结构
0x200	0，备用结构
0x210	1，备用结构
...	...
0x3F0	31，备用结构

表 14-4 描述了控制表中的单个控制结构条目。每个条目按 16 字节边界对齐。该条目包含四个长字：源结束指针、目标结束指针、控制字和一个未使用条目。包含式结束指针指向传输的结束地址。如果源或目标是非递增的（如外设寄存器），则指针必须指向传输地址。

表 14-4. 通道控制结构

偏移	说明
0x000	源结束指针
0x004	目标结束指针

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 14-4. 通道控制结构 （续）

偏移	说明
0x008	控制字
0x00C	未使用条目

控制字包含以下字段：

- 源和目标数据大小
- 源和目标地址增量大小
- 总线仲裁前的传输次数
- 要传输的总项目数
- Useburst 标志
- 传输模式

通道的控制参数可以使用驱动程序库函数 void uDMAChannelControlSet(); 进行设置。在执行传输时，µDMA 控制器会更新传输大小和传输模式字段。在传输结束时，传输大小指示 0，传输模式指示停止。由于控制字会被 µDMA 控制器修改，因此在每次新传输之前必须重新配置控制字。源和目标结束指针不会被修改，因此如果源或目标地址保持不变，它们可以保持不变。

在开始传输之前，必须通过设置 UDMA:SETCHANNELEN 寄存器中的相应位来使能 µDMA 通道。可通过设置 UDMA:CLEARCHANNELEN 寄存器中的通道位来禁用通道。在完整 µDMA 传输结束时，控制器会自动禁用该通道。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

14.3.6 传输模式

µDMA 控制器支持多种传输模式。其中两种模式支持简单的单次传输。几种复杂模式支持连续的数据流。

14.3.6.1 停止模式

尽管停止模式严格来说并不是一种“传输模式”，但它可以作为控制字中 *mode* 字段的一个有效取值。当 mode 字段取 *stop* 时，µDMA 控制器不会执行任何传输；并且如果该通道已启用，则会禁用该通道。µDMA 控制器会在一次传输结束时更新控制字，将 mode 设置为 stop。该模式在散点-聚合操作中可能会很有用。

14.3.6.2 基本模式

在基本模式下，只要有更多的数据项需要传输，并且存在传输请求，µDMA 控制器就会执行传输。此模式用于只要外设准备好进行数据传输就会将 µDMA 请求信号置为有效的外设。对于瞬时请求，任何情况下都不得使用基本模式，即使整个传输必须完成。

当使用基本模式传输完所有数据项后，µDMA 控制器会将该通道的模式设置为停止。

14.3.6.3 自动模式

自动模式与基本模式类似，不同之处在于：当接收到传输请求后，即使 µDMA 请求被移除，传输仍会完成。此模式适用于软件触发的传输。通常，自动模式不与外设一起使用。

当使用自动模式传输完所有数据项后，µDMA 控制器会将该通道的模式设置为停止。

14.3.6.4 乒乓模式

乒乓模式用于支持与外设之间的连续数据流。使用乒乓模式必须同时实现主数据结构和备用数据结构。两个结构都由处理器设置，用于存储器与外设之间的数据传输。传输使用主控制结构启动。当使用主控制结构的传输完成时，µDMA 控制器会读取该通道的备用控制结构以继续传输。每次发生这种情况时，都会生成一个中断，并且处理器可以重新加载刚完成传输的控制结构。通过这种方式，使用主控制结构和备用控制结构在数据流入或流出外设时在缓冲区之间切换，数据流可以无限期地持续下去。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

图 14-2 显示了乒乓模式下的操作示例。

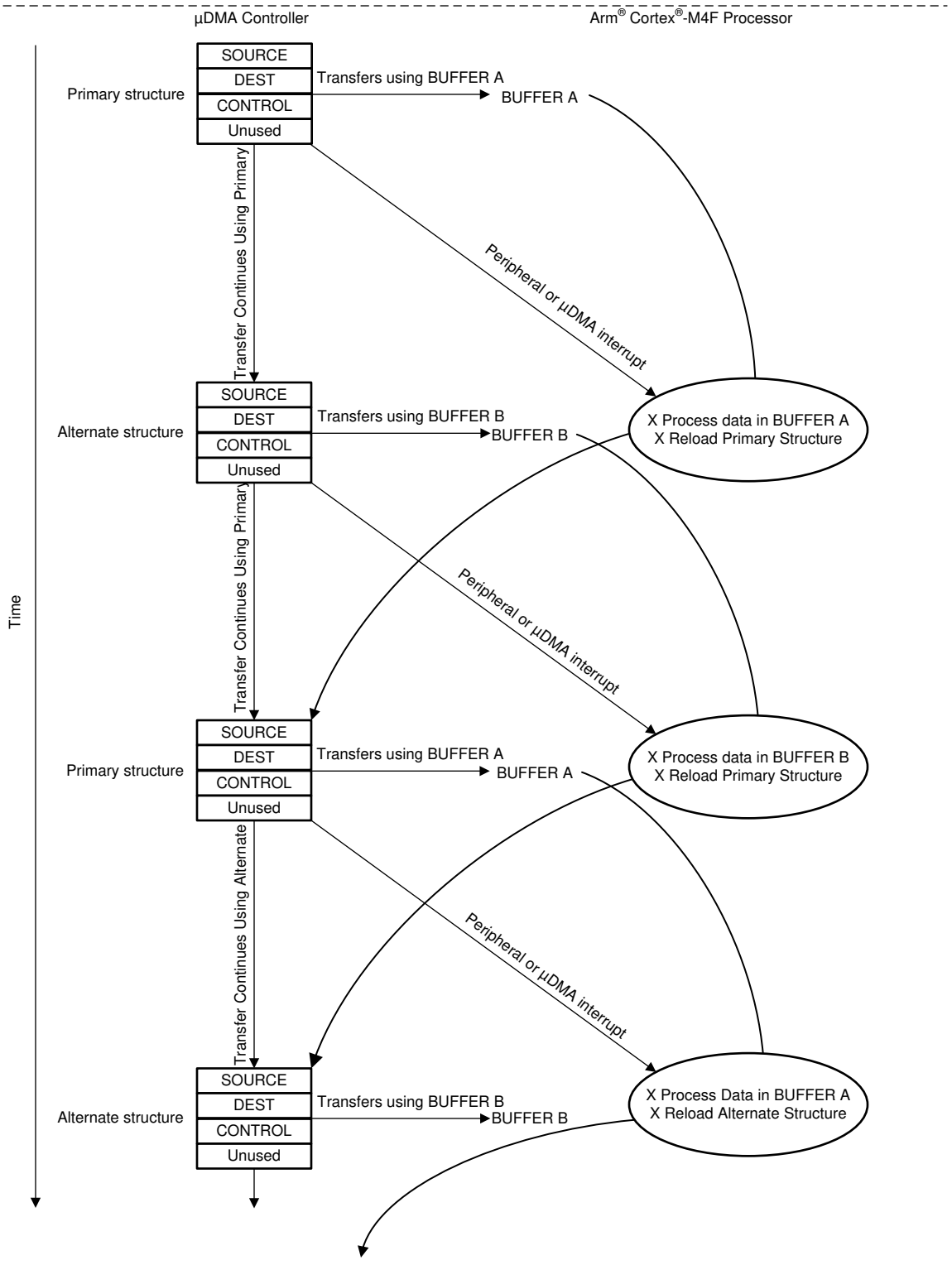


图 14-2. 乒乓 µDMA 事务示例

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

14.3.6.5 内存散聚模式

内存散聚模式是一种复杂的模式，适用于必须在内存中的不同位置（而不是在内存缓冲区中的一组连续位置）之间传输数据的场景。例如，可以使用收集 µDMA 操作选择性地读取通信协议中多个已存储数据包的有效载荷，并将它们按顺序存储在内存缓冲区中。

在内存散聚模式下，主控制结构用于通过内存中的表来编程配置备用控制结构。该表由处理器软件建立，包含一个控制结构列表（每个结构都包含源和目标结束指针）以及针对特定传输的控制字。每个控制字的模式必须设置为内存散聚模式。该表中的每个条目依次复制到随后执行该条目的备用结构中。 µDMA 控制器交替执行以下两个操作：使用主控制结构从列表中复制下一条传输指令，然后执行新的传输指令。通过对最后一个条目的控制字进行编程以使用自动传输模式来标记列表的末尾。使用自动模式执行最后一次传输时， µDMA 控制器停止。只有在完成最后一次传输之后才会生成完成中断。

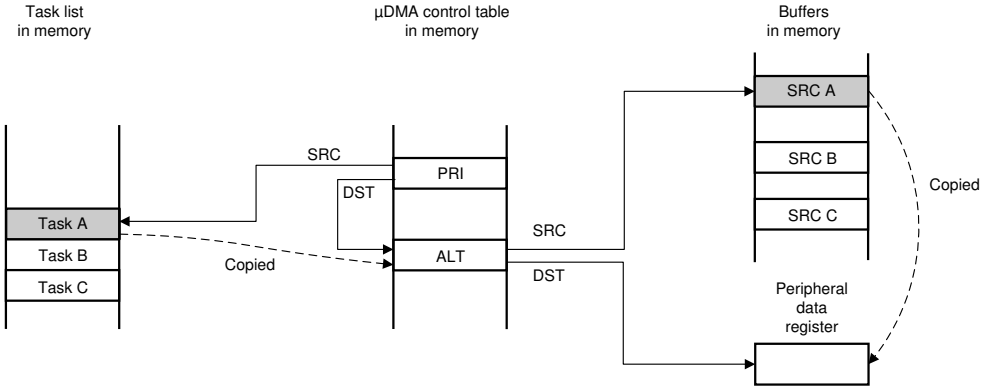
可以通过让最后一个条目复制主控件结构以指向列表的开头（或指向新列表）来循环列表。也可以触发一组其他通道来执行传输，方法是直接对另一个通道的软件触发器写入操作进行编程，或者间接通过引发某个外设动作来产生一个 µDMA 请求。

通过使用此方法对 µDMA 控制器进行编程，可以根据单个 µDMA 请求执行一组任意传输。

图 14-3 所示为内存散聚模式下的操作示例。此示例显示了一个收集操作，其中内存中三个独立缓冲区中的数据被一起复制到一个缓冲区中。图 14-3 显示了该应用如何在内存中建立 µDMA 任务列表，然后控制器使用该列表从内存中的不同位置执行三组复制操作。此操作所用通道的主控制结构配置为从任务列表复制到备用控制结构。

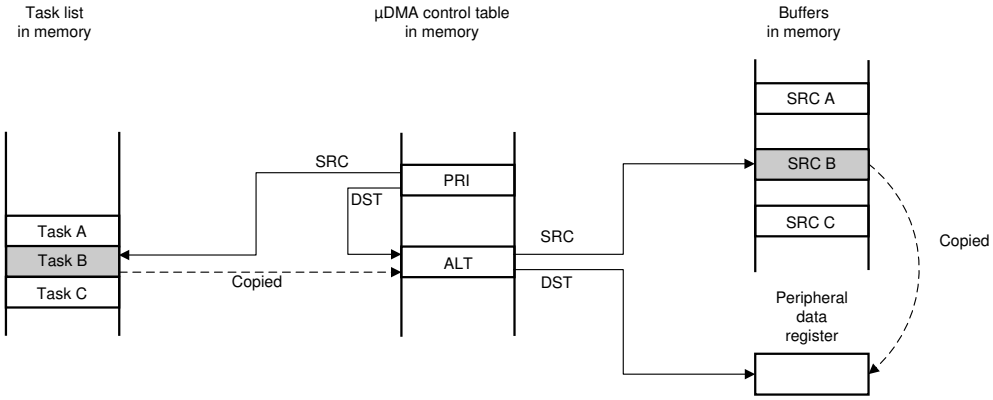
图 14-4 显示 µDMA 控制器执行三组复制操作时的顺序。首先，使用主控制结构 µDMA 控制器加载任务 A 的备用控制结构。然后， µDMA 控制器执行任务 A 指定的复制操作，将数据从源缓冲区 A 复制到目标缓冲区。接下来， µDMA 控制器再次使用主控制结构将任务 B 加载到备用控制结构中，然后使用备用控制结构执行 B 操作。对任务 C 重复该过程。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated



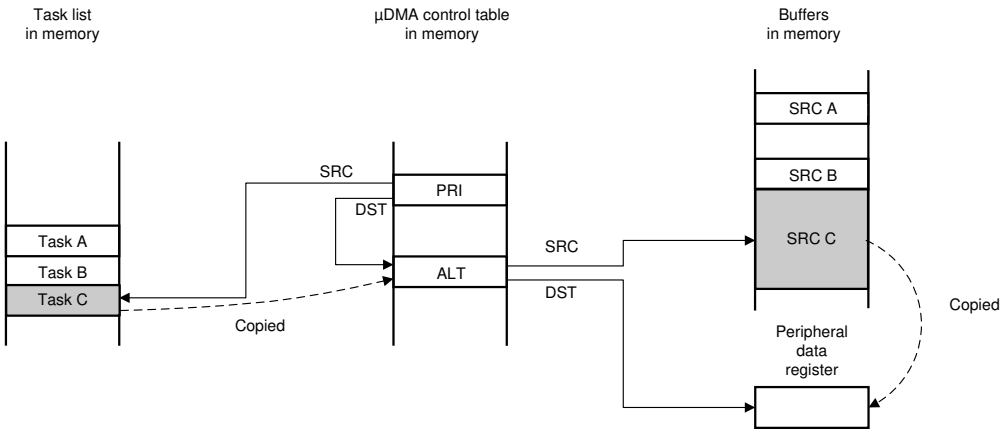
Using the primary control structure of the channel, the µDMA controller copies task A configuration to the alternate control structure of the channel.

Then, using the alternate control structure of the channel, the µDMA controller copies data from source buffer A to the peripheral data register.



Using the primary control structure of the channel, the µDMA controller copies task B configuration to the alternate control structure of the channel.

Then, using the alternate control structure of the channel, the µDMA controller copies data from source buffer B to the peripheral data register.



Using the primary control structure of the channel, the µDMA controller copies task C configuration to the alternate control structure of the channel.

Then, using the alternate control structure of the channel, the µDMA controller copies data from source buffer C to the peripheral data register.

A. 应用程序需要将数据项从内存中的三个不同位置复制到一个组合缓冲区中。

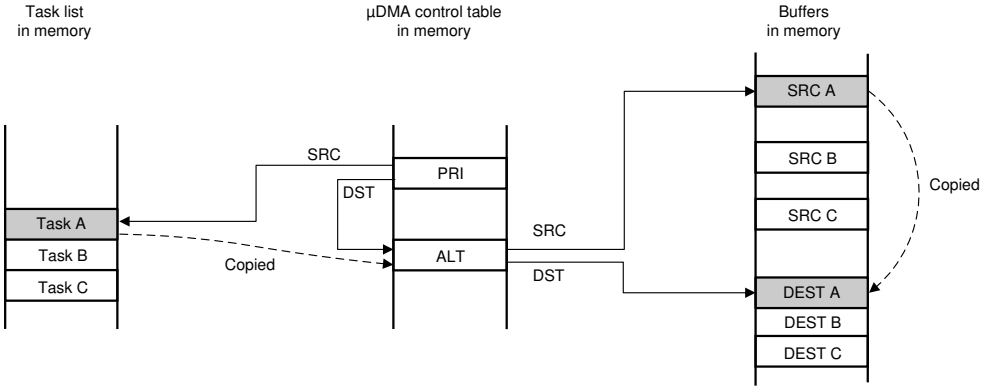
B. 应用程序会在内存中建立 µDMA “任务列表”，其中包含三个 µDMA 复制“任务”的指针和控制配置。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

C. 应用程序建立了通道主控制结构，以将每个任务配置（一次一个）复制到备用控制结构中，在备用控制结构中由 µDMA 控制器执行。

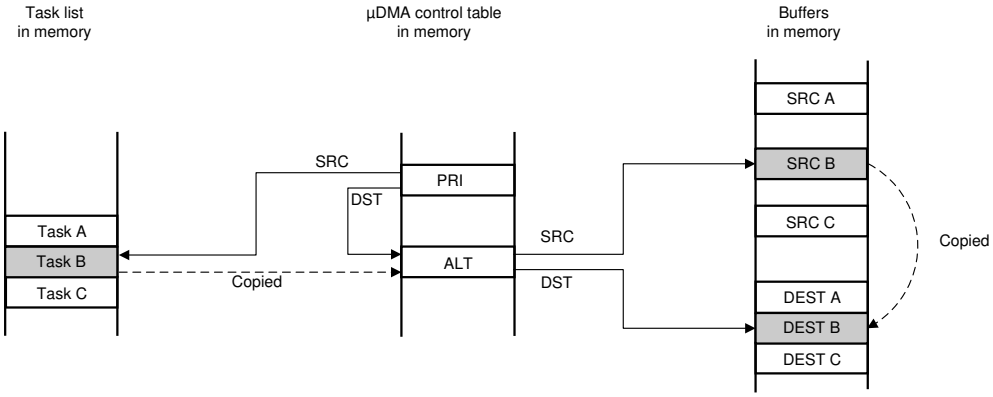
图 14-3. 内存散聚、设置和配置

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated



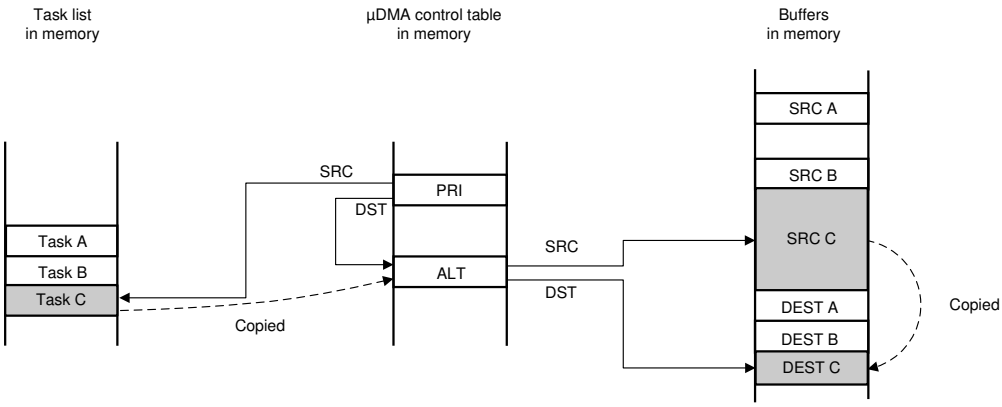
Using the primary control structure of the channel, the µDMA controller copies task A configuration to the alternate control structure of the channel.

Then, using the alternate control structure of the channel, the µDMA controller copies data from source buffer A to the destination buffer.



Using the primary control structure of the channel, the µDMA controller copies task B configuration to the alternate control structure of the channel.

Then, using the alternate control structure of the channel, the µDMA controller copies data from source buffer B to the destination buffer.



Using the primary control structure of the channel, the µDMA controller copies task C configuration to the alternate control structure of the channel.

Then, using the alternate control structure of the channel, the µDMA controller copies data from source buffer C to destination buffer.

图 14-4. 内存散聚、µDMA 复制顺序

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

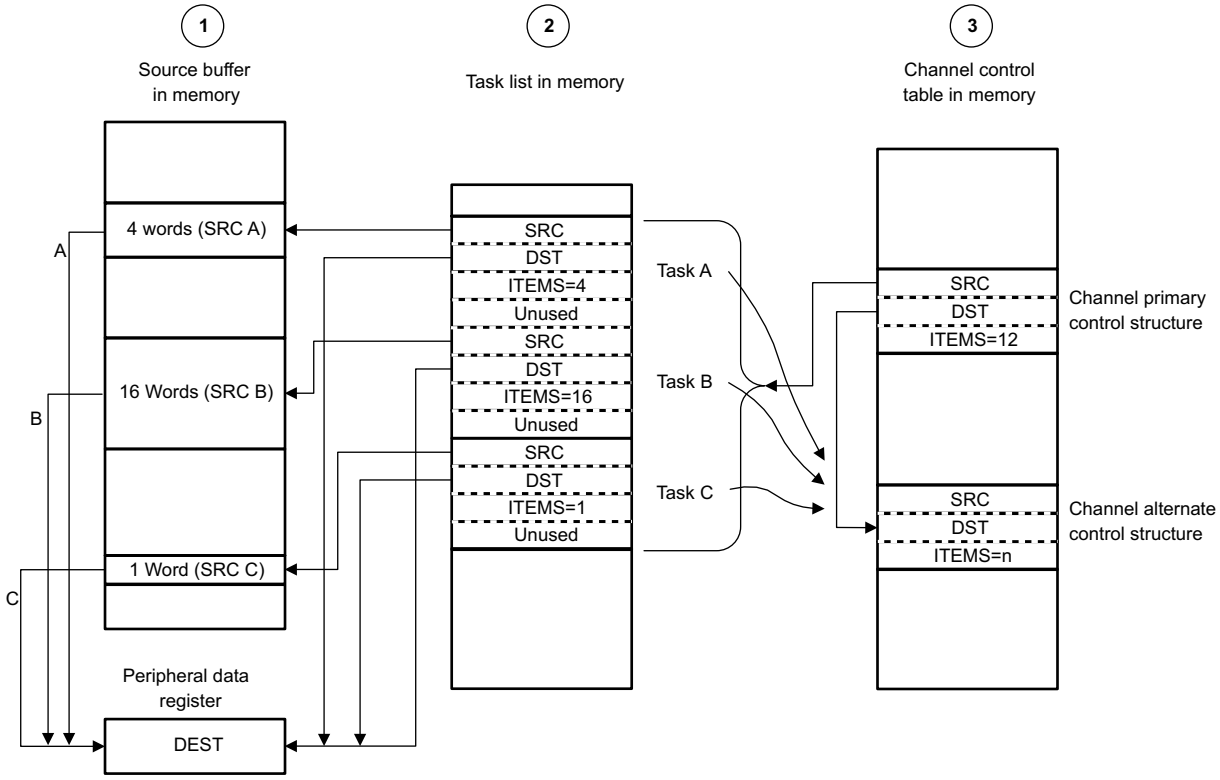
14.3.6.6 外设散聚模式

外设散聚模式与存储器散聚模式类似，不同之处在于传输由发出 µDMA 请求的外设控制。当 µDMA 控制器检测到来自外设的请求时，µDMA 控制器使用主控制结构将列表中的一个条目复制到备用控制结构，然后执行传输。在此次传输结束时，仅当外设再次将 µDMA 请求置为有效时，才会启动下一次传输。µDMA 控制器仅在外设发出请求时才继续执行列表中的传输，直到最后一次传输完成。只有在完成最后一次传输之后才会生成完成中断。

通过使用此方法，µDMA 控制器可以在外设准备好传输数据时，随时将数据从一组任意位置传输到外设或从外设传输到这些位置。

图 14-5 所示为外设散聚模式下的操作示例。此示例显示了一个收集操作，其中存储器中三个独立缓冲区的数据被复制到单个外设数据寄存器中。图 14-5 显示了应用程序如何在存储器中设置 µDMA 任务列表，然后控制器使用该列表从内存中的不同位置执行三组复制操作。此操作所用通道的主控制结构配置为从任务列表复制到备用控制结构。

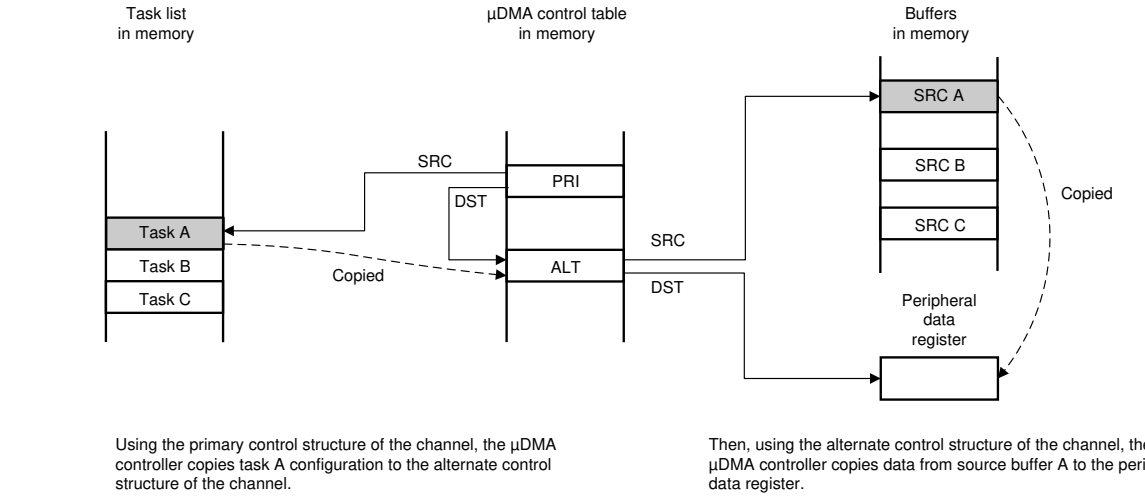
图 14-6 显示了 µDMA 控制器执行三组复制操作时的顺序。首先，µDMA 控制器使用主控制结构，将任务 A 加载到备用控制结构中。然后，µDMA 控制器执行任务 A 指定的复制操作，将数据从源缓冲区 A 复制到外设数据寄存器。接下来，µDMA 控制器再次使用主控制结构将任务 B 加载到备用控制结构中，然后使用备用控制结构执行 B 操作。对任务 C 重复该过程。



A. 应用程序需要将数据项从存储器中的三个独立位置复制到外设数据寄存器中。
B. 应用程序会在存储器中建立 µDMA“任务列表”，其中包含三个 µDMA 复制“任务”的指针和控制配置。
C. 应用程序建立了通道主控制结构，以将每个任务配置（一次一个）复制到备用控制结构中，在备用控制结构中由 µDMA 控制器执行。

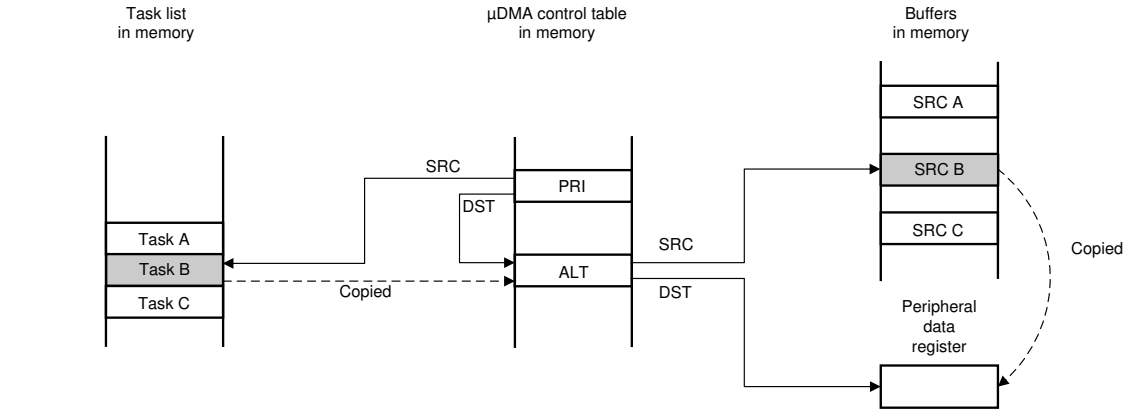
图 14-5. 外设散集、设置和配置

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated



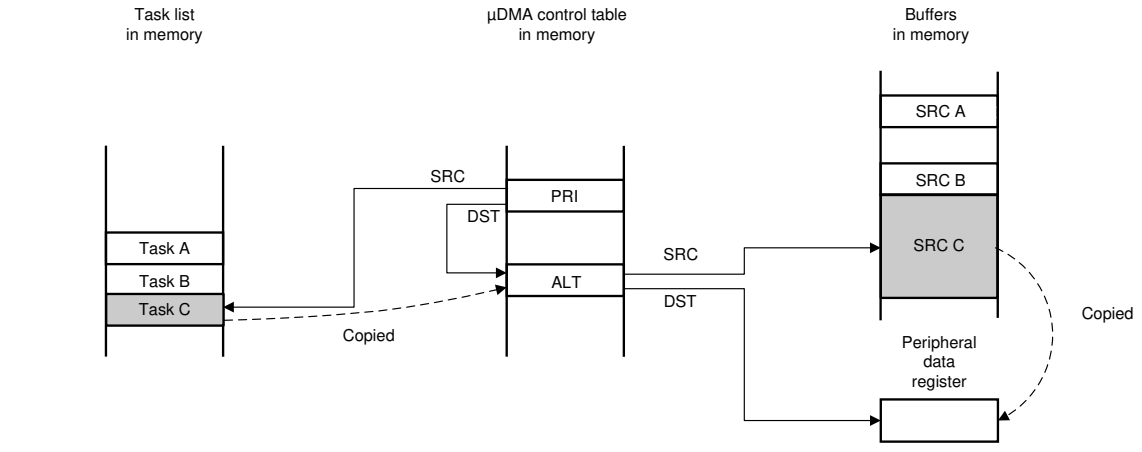
Using the primary control structure of the channel, the µDMA controller copies task A configuration to the alternate control structure of the channel.

Then, using the alternate control structure of the channel, the µDMA controller copies data from source buffer A to the peripheral data register.



Using the primary control structure of the channel, the µDMA controller copies task B configuration to the alternate control structure of the channel.

Then, using the alternate control structure of the channel, the µDMA controller copies data from source buffer B to the peripheral data register.



Using the primary control structure of the channel, the µDMA controller copies task C configuration to the alternate control structure of the channel.

Then, using the alternate control structure of the channel, the µDMA controller copies data from source buffer C to the peripheral data register.

图 14-6. 外设散聚、µDMA 复制顺序

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

14.3.7 传输大小和增量

µDMA 控制器支持 8 位、16 位或 32 位的传输数据大小。对于任何给定的传输，源数据大小和目标数据大小必须相同。源地址和目标地址可以按字节、半字、字自动递增，或设置为不递增。源地址和目标地址的增量值可以独立设置；只要增量相同或大于数据大小，地址增量不必与数据大小匹配。例如，可以使用全字（4 字节）的地址增量来执行 8 位数据大小的传输。要传输的数据必须根据数据大小（8 位、16 位或 32 位）在存储器中对齐。

表 14-5 提供了从提供 8 位数据的外设进行读取的配置。

表 14-5. µDMA 读取示例：8 位外设

字段	配置
源数据大小	8 位
目标数据大小	8 位
源地址增量	无增量
目标地址增量	字节
源结束指针	外设读取 FIFO 寄存器
目标结束指针	存储器中数据缓冲区的结束地址

14.3.8 外设接口

每个支持 µDMA 的外设都有一个单次请求或突发请求信号，当外设准备好传输数据时，该信号会被置为有效（请参阅表 14-2）。可以分别使用 UDMA:SETREQMASK 和 UDMA:CLEARREQMASK 寄存器来禁用或使能该请求信号。当设置通道请求屏蔽位时，将禁用或屏蔽 µDMA 请求信号。当请求未被屏蔽、µDMA 通道配置正确并已使能时，外设会将请求信号置为有效，然后 µDMA 控制器开始传输。

备注

当使用 µDMA 将数据传入或传出外设时，外设必须禁用所有发往事件结构的中断。

当 µDMA 传输完成时，µDMA 控制器会生成一个中断；有关更多信息，请参阅节 14.3.10。

有关特定外设如何与 µDMA 控制器交互的更多信息，请参阅讨论该外设的章节中的“DMA 操作”部分。

14.3.9 软件请求

可以设置通道以通过 UDMA:SOFTREQ 寄存器执行软件传输。如果用于软件的通道也与特定外设绑定，则 dma_done/中断信号将直接提供给 Arm® Cortex®-M4F CPU，而不是发送到外设。所使用的中断是所有软件传输共用的组合中断，编号为 46 — 软件 µDMA 中断。

如果软件使用外设的 µDMA 通道来发起请求，则完成中断将在该外设的中断向量上发生，而不是在软件中断向量上发生。

备注

DMA 软件请求在通道 0、18、19 和 20 上指定。对于通道 0 和通道 18，dma_done 可作为事件 DMA_CH0_DONE 和 DMA_CH18_DONE 在 EVENT:UDMACH14BSEL 或 EVENT:CPUIRQSEL30 寄存器的 EV 字段中使用。

14.3.10 中断和错误

当 µDMA 传输完成时，µDMA 控制器会在该外设的中断向量上生成一个完成中断。因此，如果使用 µDMA 为外设传输数据并且也使用中断，则必须设计该外设的中断处理程序来处理 µDMA 传输完成中断。如果传输使用软件 µDMA 通道，则完成中断发生在专用的软件 µDMA 中断向量上（请参阅表 14-6）。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

当为外设启用 µDMA 时，µDMA 控制器会阻止该外设的正常传输中断到达中断控制器 (INTC)。这些中断仍会在外设的中断寄存器中报告。因此，当使用 µDMA 传输大量数据时，INTC 不会随着数据流接收来自外设的多个中断，而只在传输完成时接收一个中断。未屏蔽的外设错误中断会继续发送到 INTC。

当 µDMA 通道生成完成中断时，将在 DMA 通道请求完成寄存器 UDMA:REQDONE 中设置与外设通道对应的 CHNLS 位。外设的中断处理程序代码可以使用该寄存器来确定中断是由 µDMA 通道引起的，还是由外设中断寄存器报告的错误事件引起的。当中断处理程序被激活时，来自 µDMA 控制器的完成中断请求会自动清除。

如果 µDMA 控制器在尝试执行数据传输时遇到总线或存储器保护错误，控制器会禁用导致错误的 µDMA 通道，并在 µDMA 错误中断向量上生成一个中断。处理器可以读取 DMA 清除总线错误寄存器 UDMA:ERROR 来确定是否有错误挂起。如果发生错误，会设置 STATUS 位。可以通过将 STATUS 位设置为 1 来清除该错误。

备注

错误中断或事件作为 DMA_ERR 传送到事件结构体，并通过 EVENT:CPUIRQSEL25 寄存器作为中断连接到 Arm® Cortex®-M4F 处理器。

表 14-6 列出了 µDMA 控制器的专用中断分配。

表 14-6. µDMA 中断分配

中断	分配
40	µDMA 软件通道传输
41	µDMA 错误

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

14.4 初始化和配置

14.4.1 模块初始化

DMA 控制器位于外设域中，必须为外设域通电才能启用 µDMA 控制器。需要执行以下步骤：

1. 通过设置 PRCM:PDCTL0PERIPH.ON 寄存器位或使用驱动程序库函数 (PRCM_DOMAIN_PERIPH) 启用外设电源域：

```
PRCMPowerDomainOn
```

2. 通过设置 PRCM:SECDMACLKGR.DMA_CLK_EN 寄存器位和 PRCM:SECDMACLKGS.DMA_CLK_EN 寄存器位或使用驱动程序库函数来启用 µDMA 控制器：

```
PRCMPeripheralRunEnable(uint32_t)
```

和

```
PRCMPeripheralSleepEnable(uint32_t)
```

3. 通过设置 PRCM:CLKLOADCTL.LOAD 寄存器位或使用以下函数将设置加载到时钟控制器：

```
PRCMLoadSet()
```

4. 通过设置 DMA 配置寄存器 UDMA:CFG、MASTERENABLE 位来启用 µDMA 控制器。
5. 通过将通道控制表的基地址写入 DMA 通道控制基址指针寄存器 UDMA:CTRL，来编程设定通道控制表的位置。基址必须在
1024 字节边界上保持对齐。

14.4.2 配置存储器到存储器传输

µDMA 通道 0、18、19 和 20 专用于软件发起的传输。此特定示例使用通道 0。对于基于软件的传输，无需设置任何属性。属性在默认情况下会被清除，但在以下各节所示情形中会被显式清除。

14.4.2.1 配置通道属性

按如下方式配置通道属性，或使用以下驱动程序库函数：

```
uDMAChannelAttributeDisable(uint32_t ui32Base, uint32_t ui32ChannelNum, uint32_t ui32Attr)
```

1. 对 DMA 设置通道优先级寄存器 UDMA:SETCHNLPRIORITY 或 DMA 清除通道优先级寄存器 UDMA:CLEARCHNLPRIORITY 的位 0 进行编程，以将通道设置为高优先级或默认优先级。
2. 设置 DMA 清除通道主要备用寄存器 UDMA:CLEARCHNLPRIALT 的位 0，以选择此传输的主要通道控制结构。
3. 设置 DMA 通道清除 Useburst 寄存器 UDMA:CLEARBURST 的位 0，以允许 µDMA 控制器响应单个请求和突发请求。
4. 设置 DMA 清除通道请求屏蔽寄存器 UDMA:CLEARREQMASK 的位 0，以允许 µDMA 控制器识别此通道的请求。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

14.4.2.2 配置通道控制结构

此示例将 256 个字从一个存储器缓冲区传输到另一个存储器缓冲区。通道 0 用于软件传输，通道 0 的控制结构必须配置为传输 8 位数据，源和目标按字节递增，进行字节方式的缓冲区复制。此处可以使用的总线仲裁大小为 8。

现在已配置好传输缓冲区和传输大小。该传输使用自动模式，这意味着传输在第一次请求后将自动运行到完成。

14.4.2.3 启动传输

最后，必须使能该通道。由于这是软件发起的传输，还必须发出一个请求。该请求将启动传输。

1. 使能全局中断 (IntMasterEnable();) 并为 DMA 使能中断 (IntEnable(uint32_t ui32Interrupt))。
2. 通过设置 DMA 设置通道使能寄存器 UDMA:SETCHANNELEN 的位 0 来使能该通道。
3. 通过设置 DMA 通道软件请求寄存器 UDMA:SOFTREQ 的位 0 来发出传输请求。
4. µDMA 传输开始。如果中断已使能，则传输完成时处理器会收到中断通知。

如果需要，可以通过读取 UDMA:SETCHANNELEN 寄存器的位 0 来检查状态。当传输完成时，该位会自动清除。

14.5 µDMA 寄存器

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

14.5.1 UDMA 寄存器

表 14-7 列出了 UDMA 寄存器的存储器映射寄存器。表 14-7 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

表 14-7. UDMA 寄存器

偏移	首字母缩写词	寄存器名称	部分
0h	状态	状态	STATUS 寄存器（偏移 = 0h）[复位 = 001F0000h]
4h	CFG	配置	CFG 寄存器（偏移 = 4h）[复位 = 00000000h]
8h	CTRL	通道控制数据基指针	CTRL 寄存器（偏移 = 8h）[复位 = 00000000h]
Ch	ALTCTRL	通道备用控制数据基指针	ALTCTRL 寄存器（偏移 = Ch）[复位 = 00000200h]
10h	WAITONREQ	通道等待请求状态	WAITONREQ 寄存器（偏移 = 10h）[复位 = FFFF1EFFh]
14h	SOFTREQ	通道软件请求	SOFTREQ 寄存器（偏移 = 14h）[复位 = 00000000h]
18h	SETBURST	通道设置 UseBurst	SETBURST 寄存器（偏移 = 18h）[复位 = 00000000h]
1Ch	CLEARBURST	通道清除 UseBurst	CLEARBURST 寄存器（偏移 = 1Ch）[复位 = 00000000h]
20h	SETREQMASK	通道设置请求屏蔽	SETREQMASK 寄存器（偏移 = 20h）[复位 = 00000000h]
24h	CLEARREQMASK	清除通道请求屏蔽	CLEARREQMASK 寄存器（偏移 = 24h）[复位 = 00000000h]
28h	SETCHANNELEN	设置通道使能	SETCHANNELEN 寄存器（偏移 = 28h）[复位 = 00000000h]
2Ch	CLEARCHANNELEN	清除通道使能	CLEARCHANNELEN 寄存器（偏移 = 2Ch）[复位 = 00000000h]
30h	SETCHNLPRIALT	通道设置主要-备用	SETCHNLPRIALT 寄存器（偏移 = 30h）[复位 = 00000000h]
34h	CLEARCHNLPRIALT	通道清除主要-备用	CLEARCHNLPRIALT 寄存器（偏移 = 34h）[复位 = 00000000h]
38h	SETCHNLPRIORITY	设置通道优先级	SETCHNLPRIORITY 寄存器（偏移 = 38h）[复位 = 00000000h]
3Ch	CLEARCHNLPRIORITY	清除通道优先级	CLEARCHNLPRIORITY 寄存器（偏移 = 3Ch）[复位 = 00000000h]
4Ch	ERROR	错误状态和清除	ERROR 寄存器（偏移 = 4Ch）[复位 = 00000000h]
504h	REQDONE	通道请求完成	REQDONE 寄存器（偏移 = 504h）[复位 = 00000000h]
520h	DONEMASK	通道请求完成屏蔽	DONEMASK 寄存器（偏移 = 520h）[复位 = 00000000h]

复杂的位访问类型经过编码可适应小型表单元。表 14-8 展示了适用于此部分中访问类型的代码。

表 14-8. UDMA 访问类型代码

访问类型	代码	说明
读取类型		
R	R	读取
写入类型		

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 14-8. UDMA 访问类型代码（续）

访问类型	代码	说明
W	W	写入
复位或默认值		
-n		复位后的值或默认值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

14.5.1.1 STATUS 寄存器（偏移 = 0h）[复位 = 001F0000h]

图 14-7 展示了 STATUS，表 14-9 中对此进行了介绍。

返回到汇总表。

状态

图 14-7. STATUS 寄存器

31	30	29	28	27	26	25	24
测试				RESERVED			
R-0h				R-0h			
23	22	21	20	19	18	17	16
RESERVED			TOTALCHANNELS				
R-0h			R-1Fh				
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
STATE				RESERVED			MASTERENABLE
R-0h				R-0h			R-0h

表 14-9. 状态寄存器字段说明

位	字段	类型	复位	说明
31-28	测试	R	0h	0x0: 控制器不包含集成测试逻辑 0x1：控制器包含集成测试逻辑 0x2：未定义 ... 0xF：未定义
27-21	RESERVED	R	0h	保留
20-16	TOTALCHANNELS	R	1Fh	寄存器值返回可用 uDMA 通道数减一。例如以下读取值： 0x00：显示控制器配置为使用 1 个 uDMA 通道 0x01：显示控制器配置为使用 2 个 uDMA 通道 ... 0x1F：显示控制器配置为使用 32 个 uDMA 通道 (32-1=31=0x1F)
15-8	RESERVED	R	0h	保留
7-4	STATE	R	0h	控制状态机的当前状态。状态可以是以下值之一： 0x0：空闲 0x1：读取通道控制器数据 0x2：读取源数据结束指针 0x3：读取目标数据结束指针 0x4：读取源数据 0x5：写入目标数据 0x6：等待 uDMA 请求清除 0x7：写入通道控制器数据 0x8：暂停 0x9：完成 0xA：外设散聚转换 0xB：未定义 ... 0xF：未定义。
3-1	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 14-9. 状态寄存器字段说明（续）

位	字段	类型	复位	说明
0	MASTERENABLE	R	0h	显示由 CFG.MASTERENABLE 配置的控制器的使能状态： 0：控制器禁用 1：控制器使能

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

14.5.1.2 CFG 寄存器（偏移 = 4h）[复位 = 00000000h]

图 14-8 展示了 CFG，表 14-10 中对此进行了介绍。

返回到汇总表。

配置

图 14-8. CFG 寄存器

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

表 14-10. CFG 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-5	PRTOCTRL	W	0h	通过控制 AHB 信号 HProt[3:1] 设置 AHB-Lite 总线协议保护状态，如下所示： 位 [7] 控制 HProt[3] 以指示是否正在发生可缓存访问。 位 [6] 控制 HProt[2] 以指示是否正在发生可缓存访问。 位 [5] 控制 HProt[1] 以指示是否正在发生特权访问。 当位 [n] = 1 时，相应的 HProt 位为高电平。 当位 [n] = 0 时，相应的 HProt 位为低电平。 该字段控制由 uDMA 发起的所有事务的 HProt[3:1] 信号，但以下两个事务除外： - 从源地址指针指示的地址读取 - 向目标地址指针指示的地址写入 这两个例外的 HProt[3:1] 可由通道配置描述符中的专用字段控制。
4-1	RESERVED	R	0h	保留
0	MASTERENABLE	W	0h	使能控制器： 0：禁用控制器 1：使能控制器

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

14.5.1.3 CTRL 寄存器（偏移 = 8h）[复位 = 00000000h]

图 14-9 展示了 CTRL，表 14-11 中对此进行了介绍。

返回到汇总表。

通道控制数据基指针

图 14-9. CTRL 寄存器

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

表 14-11. CTRL 寄存器字段说明

位	字段	类型	复位	说明
31-10	BASEPTR	R/W	0h	此寄存器指向每个 DMA 通道主要数据结构的基地址。该类数据并非存储在模块内部，而是存储在系统内存中，因此在使用 DMA 时，必须为其分配对应的空间
9-0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

14.5.1.4 ALTCTRL 寄存器（偏移 = Ch）[复位 = 00000200h]

图 14-10 展示了 ALTCTRL，表 14-12 中对此进行了介绍。

返回到汇总表。

通道备用控制数据基指针

图 14-10. ALTCTRL 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BASEPTR																															
R-200h																															

表 14-12. ALTCTRL 寄存器字段说明

位	字段	类型	复位	说明
31-0	BASEPTR	R	200h	此寄存器显示备用数据结构的基地址，并由模块计算得出，因此为只读

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

14.5.1.5 WAITONREQ 寄存器（偏移 = 10h）[复位 = FFFF1EFFh]

图 14-11 展示了 WAITONREQ，表 14-13 中对此进行了介绍。

返回到汇总表。

通道等待请求状态

图 14-11. WAITONREQ 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHNLSTATUS																															
R-FFFF1EFFh																															

表 14-13. WAITONREQ 寄存器字段说明

位	字段	类型	复位	说明
31-0	CHNLSTATUS	R	FFFF1EFFh	通道等待请求状态： 位 [Ch] = 0：uDMA 在通道 Ch 上接收到单个或突发请求后，即使请求仍然存在，该通道也可能退出活动状态。 位 [Ch] = 1：uDMA 在通道 Ch 上接收到单个或突发请求后，它会使通道 Ch 保持活动状态，直到请求置为无效。对于请求者处于异步时钟域中或运行时钟速度比 uDMA 慢的通道，此握手机制是必要的

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

14.5.1.6 SOFTREQ 寄存器（偏移 = 14h）[复位 = 00000000h]

图 14-12 展示了 SOFTREQ，表 14-14 中对此进行了介绍。

返回到汇总表。

通道软件请求

图 14-12. SOFTREQ 寄存器

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

表 14-14. SOFTREQ 寄存器字段说明

位	字段	类型	复位	说明
31-0	CHNLS	W	0h	设置相应的位以在相应的 uDMA 通道上生成软件 uDMA 请求 位 [Ch] = 0：不会为通道 Ch 创建 uDMA 请求 位 [Ch] = 1：为通道 Ch 创建 uDMA 请求 写入未实现 uDMA 通道的位不会为该通道创建 uDMA 请求

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

14.5.1.7 SETBURST 寄存器（偏移 = 18h）[复位 = 00000000h]

图 14-13 展示了 SETBURST，表 14-15 中对此进行了介绍。

返回到汇总表。

通道设置 UseBurst

图 14-13. SETBURST 寄存器

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

表 14-15. SETBURST 寄存器字段说明

位	字段	类型	复位	说明
31-0	CHNLS	R/W	0h	返回 useburst 状态，或禁止单个通道生成单次 uDMA 请求。值 R 为仲裁速率，存储在控制器数据结构中。 读取时： 位 [Ch] = 0：uDMA 通道 Ch 响应通道 C 上的突发和单次请求。控制器执行 2^R 或单次总线传输 位 [Ch] = 1：uDMA 通道 Ch 不响应单次传输请求。控制器仅响应突发传输请求并执行 2^R 次传输。 写入时： 位 [Ch] = 0：没有影响。使用 CLEARBURST.CHNLS 将位 [Ch] 设置为 0。 位 [Ch] = 1：禁用通道 Ch 上的单次传输请求。控制器针对突发请求执行 2^R 次传输。 写入到未实现 uDMA 通道的位无效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

14.5.1.8 CLEARBURST 寄存器（偏移 = 1Ch）[复位 = 00000000h]

图 14-14 展示了 CLEARBURST，表 14-16 中对此进行了介绍。

返回到汇总表。

通道清除 UseBurst

图 14-14. CLEARBURST 寄存器

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

表 14-16. CLEARBURST 寄存器字段说明

位	字段	类型	复位	说明
31-0	CHNLS	W	0h	设置相应的位以使能单次传输请求。 写入时： 位 [Ch] = 0：没有影响。使用 SETBURST.CHNLS 禁用单次传输请求。 位 [Ch] = 1：在通道 Ch 上使能单次传输请求。 写入到未实现 DMA 通道的位无效。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

14.5.1.9 SETREQMASK 寄存器（偏移 = 20h）[复位 = 00000000h]

图 14-15 展示了 SETREQMASK，表 14-17 中对此进行了介绍。

返回到汇总表。

通道设置请求屏蔽

图 14-15. SETREQMASK 寄存器

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

表 14-17. SETREQMASK 寄存器字段说明

位	字段	类型	复位	说明
31-0	CHNLS	R/W	0h	返回突发和单次请求屏蔽状态，或禁止相应的通道生成 uDMA 请求。 读取时： 位 [Ch] = 0：为通道 Ch 使能外部请求。 位 [Ch] = 1：为通道 Ch 禁用外部请求。 写入时： 位 [Ch] = 0：没有影响。使用 CLEARREQMASK.CHNLS 使能 uDMA 请求。 位 [Ch] = 1：禁止 uDMA 突发请求通道 [C] 和 uDMA 单次请求通道 [C] 输入生成 uDMA 请求。 写入到未实现 uDMA 通道的位无效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

14.5.1.10 CLEARREQMASK 寄存器（偏移 = 24h）[复位 = 00000000h]

图 14-16 展示了 CLEARREQMASK，表 14-18 中对此进行了介绍。

返回到汇总表。

清除通道请求屏蔽

图 14-16. CLEARREQMASK 寄存器

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

表 14-18. CLEARREQMASK 寄存器字段说明

位	字段	类型	复位	说明
31-0	CHNLS	W	0h	设置相应的位以使能通道的 DMA 请求。 写入时： 位 [Ch] = 0：没有影响。使用 SETREQMASK.CHNLS 禁止通道 C 生成请求。 位 [Ch] = 1：使能通道 [C] 以生成 DMA 请求。 写入到未实现 DMA 通道的位无效。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

14.5.1.11 SETCHANNELEN 寄存器（偏移 = 28h）[复位 = 00000000h]

图 14-17 展示了 SETCHANNELEN，表 14-19 中对此进行了介绍。

返回到汇总表。

设置通道使能

图 14-17. SETCHANNELEN 寄存器

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

表 14-19. SETCHANNELEN 寄存器字段说明

位	字段	类型	复位	说明
31-0	CHNLS	R/W	0h	返回通道的使能状态，或使能相应的通道。 读取时： 位 [Ch] = 0：通道 Ch 禁用。 位 [Ch] = 1：通道 Ch 使能。 写入时： 位 [Ch] = 0：没有影响。使用 CLEARCHANNELEN.CHNLS 禁用通道 位 [Ch] = 1：使能通道 Ch 写入到未实现 DMA 通道的位无效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

14.5.1.12 CLEARCHANNELEN 寄存器（偏移 = 2Ch）[复位 = 00000000h]

图 14-18 展示了 CLEARCHANNELEN，表 14-20 中对此进行了介绍。

返回到汇总表。

清除通道使能

图 14-18. CLEARCHANNELEN 寄存器

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

表 14-20. CLEARCHANNELEN 寄存器字段说明

位	字段	类型	复位	说明
31-0	CHNLS	W	0h	设置相应的位以禁用相应的 uDMA 通道。 写入时： 位 [Ch] = 0：没有影响。使用 SETCHANNELEN.CHNLS 使能 uDMA 通道。 位 [Ch] = 1：禁用通道 Ch 写入到未实现 uDMA 通道的位无效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

14.5.1.13 SETCHNLPRIALT 寄存器（偏移 = 30h）[复位 = 00000000h]

图 14-19 展示了 SETCHNLPRIALT，表 14-21 中对此进行了介绍。

返回到汇总表。

通道设置主要-备用

图 14-19. SETCHNLPRIALT 寄存器

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

表 14-21. SETCHNLPRIALT 寄存器字段说明

位	字段	类型	复位	说明
31-0	CHNLS	R/W	0h	返回通道控制数据结构状态，或为相应的 uDMA 通道选择备用数据结构。 读取时： 位 [Ch] = 0：uDMA 通道 Ch 使用主要数据结构。 位 [Ch] = 1：uDMA 通道 Ch 使用备用数据结构。 写入时： 位 [Ch] = 0：没有影响。使用 CLEARCHNLPRIALT.CHNLS 禁用通道 位 [Ch] = 1：为通道 Ch 选择备用数据结构 写入到未实现 uDMA 通道的位无效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

14.5.1.14 CLEARCHNLPRIALT 寄存器（偏移 = 34h）[复位 = 00000000h]

图 14-20 展示了 CLEARCHNLPRIALT，表 14-22 中对此进行了介绍。

返回到汇总表。

通道清除主要-备用

图 14-20. CLEARCHNLPRIALT 寄存器

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

表 14-22. CLEARCHNLPRIALT 寄存器字段说明

位	字段	类型	复位	说明
31-0	CHNLS	W	0h	清除相应的位以为相应的 uDMA 通道选择主要数据结构。 写入时： 位 [Ch] = 0：没有影响。使用 SETCHNLPRIALT.CHNLS 选择备用数据结构。 位 [Ch] = 1：为通道 Ch 选择主要数据结构。 写入到未实现 uDMA 通道的位无效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

14.5.1.15 SETCHNLPRIORITY 寄存器（偏移 = 38h）[复位 = 00000000h]

图 14-21 展示了 SETCHNLPRIORITY，表 14-23 中对此进行了介绍。

返回到汇总表。

设置通道优先级

图 14-21. SETCHNLPRIORITY 寄存器

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

表 14-23. SETCHNLPRIORITY 寄存器字段说明

位	字段	类型	复位	说明
31-0	CHNLS	R/W	0h	返回通道优先级屏蔽状态，或将通道优先级设置为高。 读取时： 位 [Ch] = 0：uDMA 通道 Ch 使用默认优先级。 位 [Ch] = 1：uDMA 通道 Ch 使用高优先级。 写入时： 位 [Ch] = 0：没有影响。使用 CLEARCHNLPRIORITY.CHNLS 将通道 Ch 设置为默认优先级。 位 [Ch] = 1：通道 Ch 使用高优先级。 写入到未实现 uDMA 通道的位无效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

14.5.1.16 CLEARCHNLPRIORITY 寄存器（偏移 = 3Ch）[复位 = 00000000h]

图 14-22 展示了 CLEARCHNLPRIORITY，表 14-24 中对此进行了介绍。

返回到汇总表。

清除通道优先级

图 14-22. CLEARCHNLPRIORITY 寄存器

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

表 14-24. CLEARCHNLPRIORITY 寄存器字段说明

位	字段	类型	复位	说明
31-0	CHNLS	W	0h	清除相应的位以为指定的 uDMA 通道选择默认优先级。 写入时： 位 [Ch] = 0：没有影响。使用 SETCHNLPRIORITY.CHNLS 将通道 Ch 设置为高优先级。 位 [Ch] = 1：通道 Ch 使用默认优先级。 写入到未实现 uDMA 通道的位无效

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

14.5.1.17 ERROR 寄存器（偏移 = 4Ch）[复位 = 00000000h]

图 14-23 展示了 ERROR，表 14-25 中对此进行了介绍。

返回到汇总表。

错误状态和清除

图 14-23. ERROR 寄存器

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

表 14-25. ERROR 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	状态	R/W	0h	返回 uDMA 中总线错误标志的状态，或清除该位 读取时： 0：未检测到总线错误 1：检测到总线错误 写入时： 0：无效，总线错误标志的状态保持不变。 1：清除总线错误标志。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

14.5.1.18 REQDONE 寄存器（偏移 = 504h）[复位 = 00000000h]

图 14-24 展示了 REQDONE，表 14-26 中对此进行了介绍。

返回到汇总表。

通道请求完成

图 14-24. REQDONE 寄存器

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

表 14-26. REQDONE 寄存器字段说明

位	字段	类型	复位	说明
31-0	CHNLS	R/W	0h	反映给定通道（即通道 [Ch]）的 uDMA 完成状态。这是一个粘滞完成位。除非通过写入 1 来清除，否则它会保存值 1。 读取时： 位 [Ch] = 0：通道 Ch 的请求尚未完成 位 [Ch] = 1：通道 Ch 的请求已完成 向各个位写入 1 将清除相应的位。 写入时： 位 [Ch] = 0：无效。 位 [Ch] = 1：相应的 [Ch] 位被清除并设置为 0

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

14.5.1.19 DONEMASK 寄存器（偏移 = 520h）[复位 = 00000000h]

图 14-25 展示了 DONEMASK，表 14-27 中对此进行了介绍。

返回到汇总表。

通道请求完成屏蔽

图 14-25. DONEMASK 寄存器

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

表 14-27. DONEMASK 寄存器字段说明

位	字段	类型	复位	说明
31-0	CHNLS	R/W	0h	控制 uDMA 完成和活动状态向所分配外设的传播。专门用于软件通道。 读取时： 位 [Ch] = 0：通道 Ch 的 uDMA 完成和活动状态未被阻止到达外设。请注意，通道 [Ch] 的 uDMA 完成状态被阻止参与生成组合 uDMA 完成信号 位 [Ch] = 1：通道 Ch 的 uDMA 完成和活动状态被阻止到达外设。请注意，通道 [Ch] 的 uDMA 完成状态未被阻止参与生成组合 uDMA 完成信号 写入时： 位 [Ch] = 0：允许 uDMA 完成和活动状态传播到外设。请注意，这会禁止通道 [Ch] 的 uDMA 完成状态参与生成组合 uDMA 完成信号 位 [Ch] = 1：阻止 uDMA 完成和活动状态传播到外设。请注意，这会使能通道 [Ch] 的 uDMA 完成状态参与生成组合 uDMA 完成信号。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

章节 15

计时器

本章介绍了通用计时器。

15.1 通用计时器.....1328
15.2 方框图.....1329
15.3 功能说明.....1329
15.4 初始化和配置.....1339
15.5 GPTM 寄存器.....1342

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

15.1 通用计时器

可编程的计时器可对驱动计时器输入引脚的外部事件进行计数或计时。CC13x2 和 CC26x2 器件平台的通用计时器模块 (GPTM) 提供两个 16 位计时器（计时器 A 和计时器 B），这些计时器可以配置为作为计时器独立运行，也可以级联作为一个 32 位计时器运行。

GPTM 是 CC13x2 和 CC26x2 器件平台上提供的一种计时资源。其他计时器资源包括系统计时器 (SysTick) 和看门狗计时器 (WDT)。有关参考，请参阅节 4.2.1 和章节 17。

GPTM 包含四个 GPTM 模块，具有以下功能选项：

- 工作模式：
  - 带有 8 位预分频器的 16 位或 32 位可编程单次计时器
  - 带有 8 位预分频器的 16 位或 32 位可编程周期计时器
  - 每个 32 位计时器有两个捕获比较 PWM 引脚 (CCP)
  - 24 位输入边沿计数或 24 位时间捕获模式
  - 24 位 PWM 模式，支持软件编程的 PWM 信号输出反相
- 向上或向下计数
- 计时器模块的菊花链级联允许单个计时器启动多个计时事件
- 计时器同步允许所选计时器在同一时钟周期开始计数
- 调试期间，微控制器的 CPU 暂停时，用户可启用的停滞
- 可以确定从产生定时器中断到进入中断服务程序所经过的时间

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

15.2 方框图

图 15-1 显示了 GPTM 模块方框图。

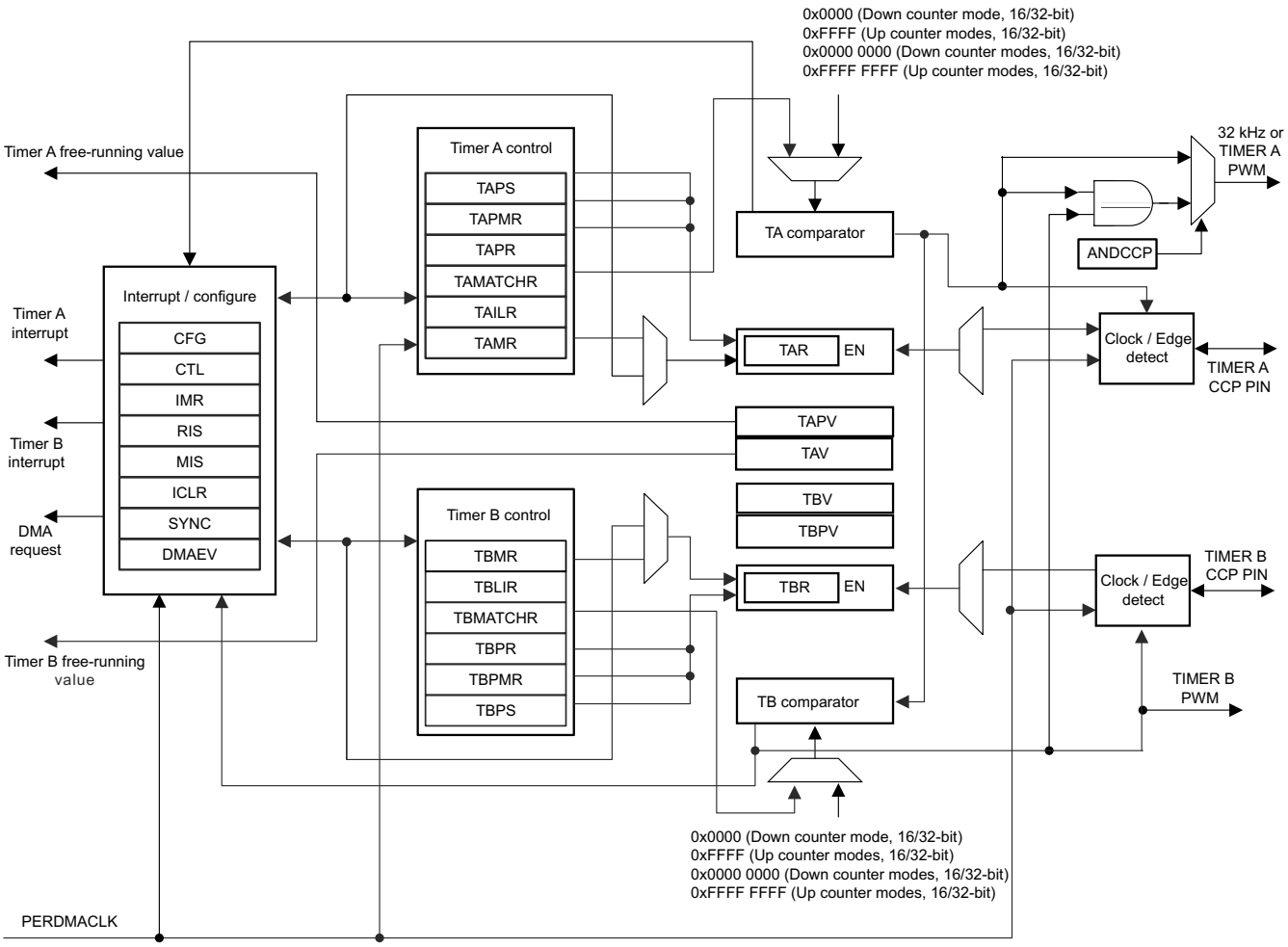


图 15-1. GPTM 模块方框图

15.3 功能说明

每个 GPTM 模块的主要组成部分包括：两个自由运行的向上和向下计数器（计时器 A 和计时器 B）、两个匹配寄存器、两个预分频器匹配寄存器、两个影子寄存器、两个加载和初始化寄存器及其相关的控制功能。

每个 GPTM 的确切功能由软件控制并通过寄存器接口进行配置。计时器 A 和计时器 B 可以单独使用，在这种情况下，它们具有 16 位的计数范围。此外，计时器 A 和计时器 B 可以级联以提供 32 位的计数范围。预分频器仅在计时器单独使用时才可用。

表 15-1 列出了每个 GPTM 模块的可用模式。在单次或周期模式下向下计数时，预分频器充当真正的预分频器，并包含计数的最低有效位 (LSB)。在单次或周期模式下向上计数时，预分频器充当计时器扩展，并保存计数的最高有效位 (MSB)。在输入边沿计数、输入边沿计时和 PWM 模式下，无论计数方向如何，预分频器始终充当计时器扩展。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 15-1. 通用计时器功能

模式	计时器使用	计数方向	计数器大小	预分频器大小(1)	预分频器行为（计数方向）
单次触发	个人	向上或向下	16 位	8 位	计时器扩展（向上），预分频器（向下）
	级联	向上或向下	32 位	-	不适用
周期性	个人	向上或向下	16 位	8 位	计时器扩展（向上），预分频器（向下）
	级联	向上或向下	32 位	-	不适用
边沿计数	个人	向上或向下	16 位	8 位	计时器扩展（两者）
边沿计时	个人	向上或向下	16 位	8 位	计时器扩展（两者）
PWM	个人	向下计数	16 位	8 位	计时器扩展

(1) 预分频器仅在计时器单独使用时可用。

软件使用 GPTM 配置寄存器 (GPT:CFG)、GPTM 计时器 A 模式寄存器 (GPT:TAMR) 和 GPTM 计时器 B 模式寄存器 (GPT:TBMR) 配置 GPTM。当处于某种级联模式时，计时器 A 和计时器 B 只能以一种模式运行。然而，当配置为单独模式时，计时器 A 和计时器 B 能以任何单独模式的组合进行独立配置。

15.3.1 GPTM 复位条件

对 GPTM 施加复位后，模块处于非活动状态，所有控制寄存器都被清除并处于其默认状态。计数器（定时器 A 和定时器 B）及其相应的加载寄存器（GPTM 定时器 A 间隔加载寄存器 GPT:TAILR 和 GPTM 定时器 B 间隔加载寄存器 GPT:TBILR）都被初始化为全 1。预分频计数器被初始化为 0x00：

- GPTM 计时器 A 预分频寄存器 (GPT:TAPR) 和 GPTM 计时器 B 预分频寄存器 (GPT:TBPR)
- GPTM 计时器 A 预分频快照寄存器 (GPT:TAPS) 和 GPTM 计时器 B 预分频快照寄存器 (GPT:TBPS)
- GPTM 计时器 A 预分频值寄存器 (GPT:TAPV) 和 GPTM 计时器 B 预分频值寄存器 (GPT:TBPV)

15.3.2 计时器模式

本节介绍了各种计时器模式的操作。当在级联模式下使用计时器 A 和计时器 B 时，只需使用计时器 A 的控制位和状态位；无需使用计时器 B 的控制位和状态位。通过向 GPTM 配置寄存器 (GPT:CFG) 写入值 0x4，可将 GPTM 置于独立或拆分模式。在以下各节中，变量 n 用于位字段和寄存器名称中，以表示计时器 A 或计时器 B 的功能。在本节中，向下计数模式下的超时事件为 0x0；在向上计数模式下，超时事件为 GPTM 间隔加载寄存器 (GPT:TnILR) 和可选的 GPTM 计时器 n 预分频寄存器 (GPT:TnPR) 中的值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

15.3.2.1 单次触发或周期计时器模式

单次触发模式或周期模式的选择由写入 GPTM 计时器 n 模式寄存器 (GPT:TnMR) TnMR 字段的值决定。计时器配置为使用 GPT:TnMR TnCDIR 位向上或向下计数。

当软件设置 GPTM 控制寄存器 (GPTIMER:CTL) TnEN 位时，计时器从 0x0 开始向上计数，或从其预加载值开始向下计数。或者，如果在设置 TnEN 位时设置了 GPT:TnMR 寄存器 TnWOT 位，计时器会等待触发器才开始计数（请参阅节 15.3.2.5）。

当计时器向下计数并达到超时事件 (0x0) 时，计时器在下一个周期从 GPT:TnILR 和 GPT:TnPR 寄存器重新加载其起始值。当计时器向上计数并达到超时事件（GPT:TnILR 和可选的 GPT:TnPR 寄存器中的值）时，计时器会重新加载 0x0。如果配置为单次触发计时器，则计时器停止计数并清除 GPT:CTL TnEN 寄存器位。如果配置为周期计时器，则计时器会在下一个周期再次开始计数。在周期性快照模式下（TnMR 字段为 0x2，并且设置了 GPT:TnMR TnSNAPS 寄存器位），发生超时事件时计时器的实际自由运行值将加载到 GPT:TnR 寄存器中。通过这种方式，软件可以检查快照值和自由运行计时器的当前值（存储在 GPT:TnV 寄存器中），以确定从中断置为有效到进入 ISR 所经过的时间。当计时器配置为单次触发模式时，快照模式不可用。

除了重新加载计数值之外，GPTM 还会在达到超时事件时生成中断和触发器。GPTM 设置 GPTM 原始中断状态寄存器 (GPT:RIS) TnTORIS 位，并一直保持该位，直到通过写入 GPTM 中断清除寄存器 (GPT:ICR) 将其清除。如果在 GPTM 中断屏蔽寄存器 (GPT:IMR) 中使能了超时中断，则 GPTM 还会设置 GPTM 屏蔽中断状态寄存器 (GPT:MIS) TnTOMIS 位。通过设置 GPT:TnMR TnMIE 寄存器位，当计时器值等于加载到 GPTM 计时器 n 匹配寄存器 (GPT:TnMATCHR) 和 GPTM 计时器 n 预分频匹配寄存器 (GPT:TnPMR) 中的值时，也可以生成中断条件。该中断具有与超时中断相同的状态、屏蔽和清除功能，但使用匹配中断位（例如，通过 GPTM 原始中断状态寄存器 (GPT:RIS) TnMRIS 位来监控原始中断状态）。硬件不会更新中断状态位，除非设置了 GPT:TnMR TnMIE 寄存器位，这与超时中断的行为不同。

如果软件在计数器向下计数时更新 GPT:TnILR 或 GPT:TnPR 寄存器，则在 GPT:TnMR TnILD 寄存器位清零时，计数器会在下一个时钟周期加载新值，并从新值开始继续计数。在设置了 TnILD 位时，计数器将在下一次超时后加载新值。如果软件在计数器向上计数时更新 GPT:TnILR 寄存器或 GPT:TnPR 寄存器，则超时事件将在下一个周期更改为新值。如果软件在计数器向上或向下计数时更新 GPTM 计时器 n 值寄存器 (GPT:TnV)，则计数器会在下一个时钟周期加载新值，并从新值开始继续计数。如果软件在计数器计数时更新 GPT:TnMATCHR 寄存器或 GPT:TnPMR 寄存器，则在 GPT:TnMR TnMRSU 寄存器位清零时，匹配寄存器会在下一个时钟周期反映新值。在设置了 TnMRSU 位时，新值直到下一次超时才会生效。

如果设置了 GPT:CTL TnSTALL 寄存器位，则计时器会在调试器暂停处理器时冻结计数。当处理器恢复执行时，计时器恢复计数。

表 15-2 列出了使用预分频器时 16 位自由运行计时器的各种配置。所有值均假设时钟为 24MHz，Tc = 41.67ns（时钟周期）。预分频器只能在 16 位或 32 位计时器配置为 16 位模式时使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 15-2. 带预分频器的 16 位计时器配置

预分频值 (8 位)	计时器时钟数 (Tc)(1)	最长时间	单位
00000000	1	2.7	ms
00000001	2	5.4	ms
00000010	3	8.1	ms
-	-	-	-
11111101	254	685.8	ms
11111110	255	688.5	ms
11111111	256	691.2	ms

(1) Tc 是时钟周期。

15.3.2.2 输入边沿计数模式

备注

对于上升沿检测，输入信号必须在上升沿之后的至少两个系统时钟周期内保持高电平。同样，对于下降沿检测，输入信号必须在下降沿之后的至少两个系统时钟周期内保持低电平。根据此标准，边沿检测的最大输入频率为系统频率的 ¼。

在边沿计数模式下，计时器配置为 24 位向下计数器，包括可选的预分频器。计数值的高位存储在 GPTM 计时器 n 预分频寄存器 (GPT:TnPR) 中，低位存储在 GPT:TnR 寄存器中。在该模式下，计时器能够捕获三种类型的事件：上升沿、下降沿或两者兼有。要将计时器置于边沿计数模式，必须清除 GPT:TnMR 寄存器 TnCMR 位。计时器计数的边沿类型由 GPT:CTL 寄存器 TnEVENT 字段决定。在向下计数模式下初始化期间，会配置 GPT:TnMATCHR 和 GPT:TnPMR 寄存器，使 GPT:TnILR 和 GPT:TnPR 寄存器中的值与 GPT:TnMATCHR 和 GPT:TnPMR 寄存器中的值之差等于必须计数的边沿事件数。在向上计数模式下，计时器从 0x0 计数到 GPT:TnMATCHR 和 GPT:TnPMR 寄存器中的值。表 15-3 列出了计时器使能时加载到计时器寄存器中的值。

表 15-3. 在输入边沿计数模式下计时器使能时的计数器值

注册	向下计数模式	向上计数模式
GPT:TnR	GPT:TnILR	0x0
GPT:TnV	GPT:TnILR	0x0
GPT:TnPV	GPT:TnPR	0x0

当软件写入 GPTM 控制寄存器 (GPT:CTL) TnEN 位时，计时器被使能以进行事件捕获。CCP 引脚上的每个输入事件都会使计数器递减或递增 1，直到事件计数与 GPT:TnMATCHR 和 GPT:TnPMR 寄存器匹配。当计数匹配时，GPTM 会将 GPTM 原始中断状态寄存器 (GPT:RIS) CnMRIS 位置为有效，并一直保持该位，直到通过写入 GPTM 中断清除寄存器 (GPT:ICR) 将其清除。如果在 GPTM 中断屏蔽寄存器 (GPT:IMR) 中使能了捕获模式匹配中断，则 GPTM 还会设置 GPTM 屏蔽中断状态寄存器 (GPT:MIS) CnMMIS 位。在此模式下，GPT:TnR 寄存器保存输入事件的计数，而 GPT:TnV 和 GPT:TnPV 寄存器保存自由运行计时器值和自由运行预分频器值。

除了生成中断之外，还可以生成 µDMA 触发器。µDMA 触发器通过配置并使能相应的 µDMA 通道来启用。

在向下计数模式下达到匹配值后，计数器会使用 GPT:TnILR 和 GPT:TnPR 寄存器中的值重新加载，并停止计数，因为 GPTM 会自动清除 GPT:CTL 寄存器中的 TnEN 位。一旦达到事件计数，所有后续事件都将被忽略，直到 TnEN 位由软件重新使能。在向上计数模式下，计时器会重新加载 0x0 并继续计数。

图 15-2 显示了输入边沿计数模式的工作原理。在此示例中，计时器起始值设置为 GPT:TnILR = 0x000A，匹配值设置为 GPT:TnMATCHR = 0x0006，因此需要计数 4 个边沿事件。计数器配置为检测输入信号的两个边沿。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

备注

最后两个边沿不会被计数，因为在当前计数与 GPT:TnMATCHR 寄存器中的值匹配后，计时器会自动清除 TnEN 位。

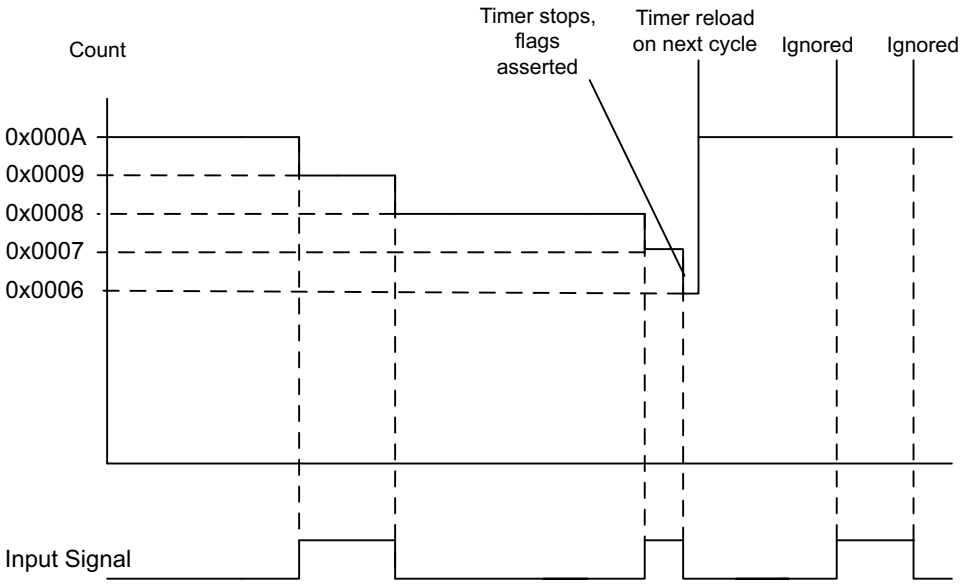


图 15-2. 输入边沿计数模式示例（向下计数）

15.3.2.3 输入边沿时间模式

备注

对于上升沿检测，输入信号必须在上升沿之后的至少两个系统时钟周期内保持高电平。同样，对于下降沿检测，输入信号必须在下降沿之后的至少两个系统时钟周期内保持低电平。根据此标准，边沿检测的最大输入频率为系统频率的 ¼。

在边沿时间模式下，计时器配置为 24 位向下计数器，包括可选的预分频器。计时器值的高位存储在 GPT:TnPR 寄存器中，低位存储在 GPT:TnILR 寄存器中。在此模式下，计时器初始化为：向下计数时加载 GPT:TnILR 和 GPT:TnPR 寄存器中的值，向上计数时加载 0x0。计时器能够捕获三种类型的事件：上升沿、下降沿或两者兼有。通过设置 GPT:TnMR TnCM 寄存器位，可将计时器置于边沿时间模式，计时器捕获的事件类型由 GPT:CTL TnEVENT 寄存器字段决定。表 15-4 列出了计时器使能时加载到计时器寄存器中的值。

表 15-4. 在输入事件计数模式下计时器使能时的计数器值

注册	向下计数模式	向上计数模式
GPT:TnR	GPT:TnILR	0x0
GPT:TnV	GPT:TnILR	0x0
GPT:TnPV	GPT:TnPR	0x0

当软件写入 GPT:CTL TnEN 寄存器位时，计时器被使能以进行事件捕获。检测到所选输入事件时，当前计时器计数器值将被捕获到 GPT:TnR 寄存器中，可供微控制器读取。然后，GPTM 会将 GPTM 原始中断状态寄存器 (GPT:RIS) CnERIS 位置为有效，并一直保持该位，直到通过写入 GPTM 中断清除寄存器 (GPT:ICR) 将其清除。如果在 GPTM 中断屏蔽寄存器 (GPT:IMR) 中使能了捕获模式事件中断，则 GPTM 还会设置 GPTM 屏蔽中断状态

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

寄存器 (GPT:MIS) CnEMIS 位。在此模式下，GPT:TnR 寄存器保存所选输入事件发生的时间，而 GPT:TnV 和 GPT:TnPV 寄存器保存自由运行计时器值和自由运行预分频器值。可以读取这些寄存器来确定从中断置为有效到进入 ISR 之间经过的时间。

除了生成中断之外，还可以生成 µDMA 触发器。该触发器通过配置并使能相应的 µDMA 通道来启用。

捕获到事件后，计时器不会停止计数。计时器继续计数，直到 TnEN 位被清除。当计时器达到超时值时，在向上计数模式下会重新加载 0x0，在向下计数模式下会重新加载 GPT:TnILR 和 GPT:TnPR 寄存器中的值。

图 15-3 显示了输入边沿计时模式的工作原理。图中假设计时器的起始值为默认值 0xFFFF，并将计时器配置为捕获上升沿事件。

每次检测到上升沿事件时，当前计数值都会加载到 GPTIMER:TnR 寄存器中，并保持该值，直到检测到另一个上升沿（此时新的计数值被加载到 GPT:TnR 寄存器中）。

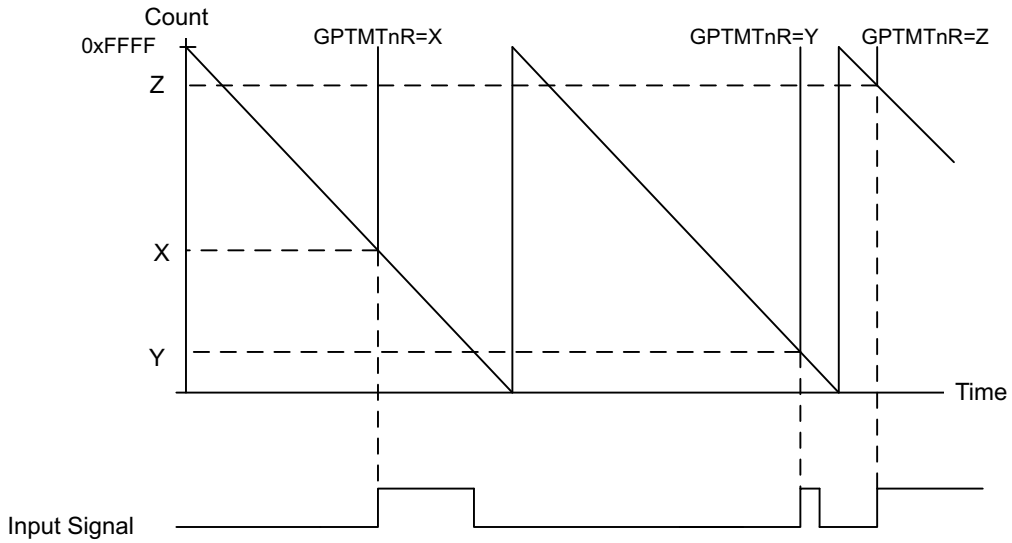


图 15-3. 输入边沿时间模式示例

备注

在边沿时间模式下运行时，如果使能了预分频器，计数器使用模 2^{24} 计数，如果未使能预分频器，则使用模 2^{16} 计数。如果边沿发生的时间可能超过计数周期，则可以使用另一个计时器来确保检测到丢失的边沿。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

15.3.2.4 PWM 模式

GPTM 支持简单的 PWM 生成模式。在 PWM 模式下，计时器配置为 16 位向下计数器，其起始值（以及周期）由 GPT:TnILR 和 GPT:TnPR 寄存器定义。在该模式下，PWM 频率和周期是同步事件；因此，可确保它们无干扰。PWM 模式通过 GPT:TnMR 寄存器使能：将 TnAMS 位设置为 0x1，将 TnCM 位设置为 0x0，并将 TnMR 字段设置为 0x2。表 15-5 列出了计时器使能时加载到计时器寄存器中的值。

备注

在 PWM 模式下不支持等待触发（菊花链）。计时器在使能后立即启动，而不等待前一个计时器的触发器。

表 15-5. PWM 模式下定时器使能时的计数器值

注册	向下计数模式	向上计数模式
GPT:TnR	GPT:TnILR	不适用
GPT:TnV	GPT:TnILR	不适用
GPT:TnPV	GPT:TnPR	不适用

当软件写入 GPT:CTL TnEN 寄存器位时，计数器开始向下计数，直到达到 0x0 状态。或者，如果在设置 TnEN 位时设置了 GPT:TnMR TnWOT 寄存器位，则计时器会等待触发器才开始计数。在周期性模式下的下一个计数器周期，计数器从 GPT:TnILR 和 GPT:TnPR 寄存器重新加载其起始值，并继续计数，直到通过软件清除 GPT:CTL TnEN 寄存器位来禁用。计时器能够基于三种类型的事件生成中断：上升沿、下降沿或两者兼有。该事件由 GPT:CTL TnEVENT 寄存器字段配置，中断通过设置 GPT:TnMR TnPWMIE 寄存器位使能。发生此事件时，会设置 GPTM 原始中断状态寄存器 (GPT:RIS) CnERIS 位，并一直保持该位，直到通过写入 GPTM 中断清除寄存器 (GPT:ICLR) 将其清除。如果在 GPTM 中断屏蔽寄存器 (GPT:IMR) 中使能了捕获模式事件中断，则 GPTM 还会设置 GPTM 屏蔽中断状态寄存器 (GPT:MIS) CnEMIS 位。

备注

除非设置 TnPWMIE 位，否则不会更新中断状态位。

在 PWM 模式下，GPT:TnR 和 GPT:TnV 寄存器始终具有相同的值，GPT:TnPS 和 GPT:TnPV 寄存器也是如此。

当计数器处于 GPT:TnILR 和 GPT:TnPR 寄存器的值（起始状态）时，输出 PWM 信号置为有效；当计数器值等于 GPT:TnMATCHR 和 GPT:TnPMR 寄存器的值时，输出 PWM 信号置为无效。软件可以通过设置 GPT:CTL TnPWML 寄存器位来反转输出 PWM 信号。反转输出 PWM 不影响边沿检测中断。因此，如果设置了正边沿中断触发，则在 PWM 反转产生正边沿时会将事件触发中断置为有效。

备注

如果将 TnILR 更改为小于当前计数器值的值，则必须使能 LD_TO_EN 以避免 PWM 输出上出现瞬态。即使在使能“超时更新”模式时也是如此。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

图 15-4 显示了如何在假设输入时钟为 50MHz 且 TnPWML = 0 的情况下，生成周期为 1ms、占空比为 66% 的输出 PWM（对于 TnPWML = 1 的配置，占空比为 33%）。在此示例中，起始值为 GPT:TnILR = 0xC350，比较值为 GPT:TnMATCHR = 0x411A。

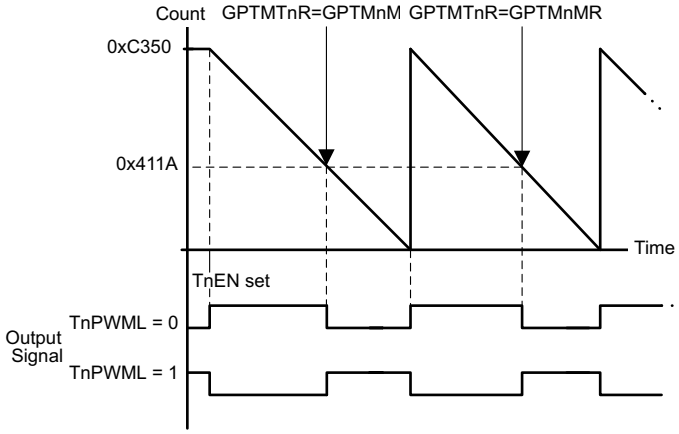


图 15-4. 16 位 PWM 模式示例

使用 GPT:SYNC 寄存器同步计时器时，必须正确配置计时器以避免 CCP 输出上出现干扰。TnPLO 和 TnMRSU 位都必须在 GPT:TnMR 寄存器中设置。图 15-5 显示了当设置 TnPLO 和 TnMRSU 位且 GPT:TnMATCHR 寄存器值大于 GPT:TnILR 寄存器值时，CCP 输出的工作方式。

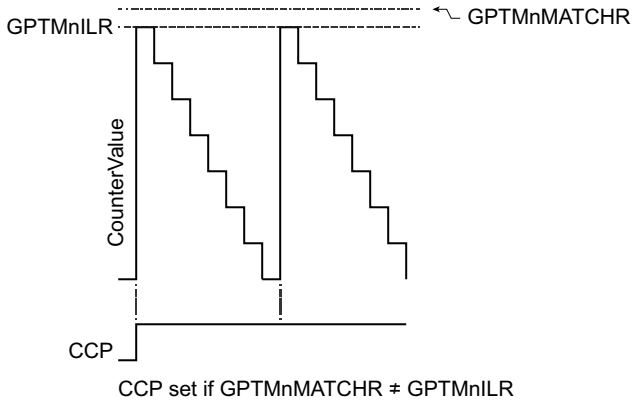


图 15-5. CCP 输出，GPT:TnMATCHR > GPT:TnILR

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

图 15-6 显示了当设置 PLO 和 MRSU 位且 GPT:TnMATCHR 寄存器值与 GPT:TnILR 寄存器值相同时，CCP 输出的工作方式。在这种情况下，如果 PLO 位为 0，则当加载 GPT:TnILR 寄存器值时 CCP 信号变为高电平，匹配结果将基本上被忽略。

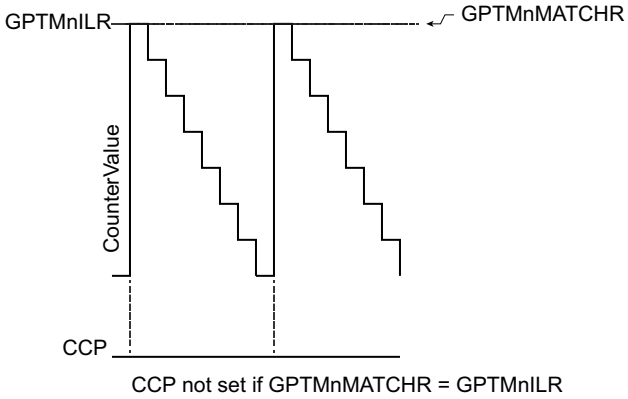


图 15-6. CCP 输出，GPT:TnMATCHR = GPT:TnILR

图 15-7 显示了当设置 PLO 和 MRSU 位且 GPT:TnILR 寄存器值大于 GPT:TnMATCHR 寄存器值时，CCP 输出的工作方式。

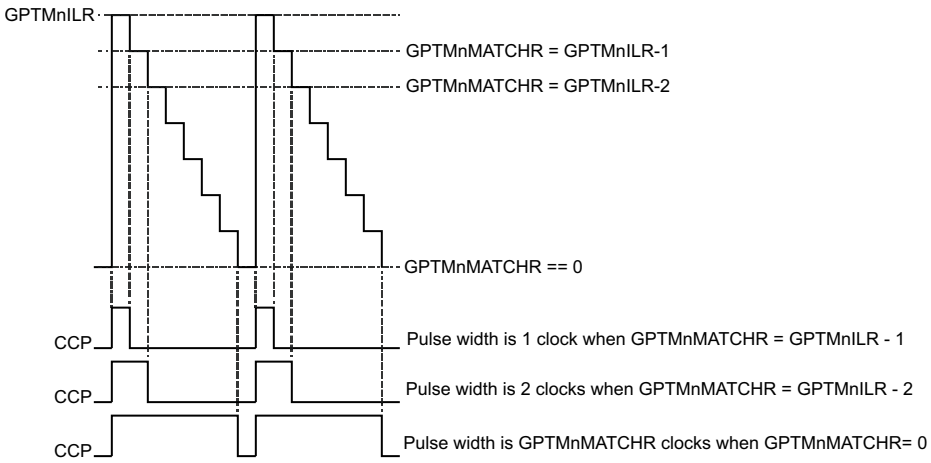


图 15-7. CCP 输出，GPT:TnILR > GPT:TnMATCHR

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

15.3.2.5 等待触发模式

等待触发模式允许计时器模块以菊花链方式级联，这样一旦配置好，单个计时器就可以使用计时器触发来启动多个计时事件。等待触发模式通过设置 GPT:TnMR TnWOT 寄存器位来使能。设置 TnWOT 位时，位于菊花链前一个位置的计时器（计时器 N）达到其超时事件之前，计时器 N+1 不会开始计数。菊花链的配置方式是：GPTM1 总是在 GPTM0 之后，GPTM2 在 GPTM1 之后，依此类推。如果计时器 A 配置为 32 位（16 位或 32 位模式）计时器（由 GPT:CFG 寄存器中的 CFG 字段控制），则它会触发下一个模块中的计时器 A。如果计时器 A 配置为 16 位（16 位或 32 位模式）计时器，则它会触发同一模块中的计时器 B，而计时器 B 会触发下一个模块中的计时器 A。确保在 GPTM0 中永远不要设置 TAWOT 位。图 15-8 显示了 GPT:CFG 寄存器的 CFG 位如何影响菊花链。此功能适用于单次和周期模式。

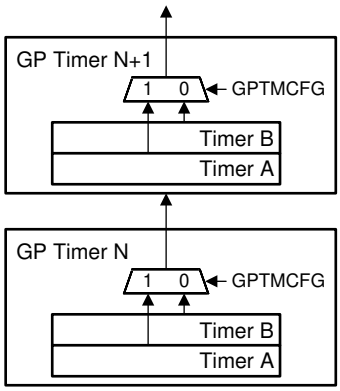


图 15-8. 计时器菊花链

15.3.3 同步 GPT 模块

GPTM0 模块中的 GPTM 同步器控制寄存器 (GPT:SYNC) 可用于同步所选的计时器，使它们同时开始计数。为此，必须首先启动这些计时器。设置 GPT:SYNC 寄存器中的某一位会使关联的计时器执行超时事件的操作。计时器同步时不会产生中断。如果在级联模式下使用计时器，则只需在 GPT:SYNC 寄存器中设置计时器 A 对应的位。寄存器说明显示了可以同步哪些计时器。

表 15-6 列出了在各种计时器模式下，计时器同步时执行的超时事件操作。

表 15-6. GPTM 模式的超时操作

模式	计数方向	超时操作
16 位和 32 位单次（级联计时器）	—	不适用
16 位和 32 位周期（级联计时器）	向下计数	计数值 = ILR
	向上	计数值 = 0
16 位和 32 位单次（独立和拆分计时器）	—	不适用
16 位和 32 位周期（独立和拆分计时器）	向下计数	计数值 = ILR
	向上	计数值 = 0
16 位和 32 位边沿计数（独立和拆分计时器）	向下计数	计数值 = ILR
	向上	计数值 = 0
16 位和 32 位边沿计时（独立和拆分计时器）	向下计数	计数值 = ILR
	向上	计数值 = 0
16 位 PWM	向下计数	计数值 = ILR

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

15.3.4 访问级联的 16 位和 32 位 GPTM 寄存器值

通过向 GPTM 配置寄存器 (GPT:CFG) GPTMCFG 位字段写入 0x0 或 0x1，可将 GPTM 置于级联模式。在这两种配置中，某些 16 位和 32 位的 GPTM 寄存器会被级联以形成伪 32 位寄存器。这些寄存器包括：

- GPTM 计时器 A 间隔加载寄存器 (GPT:TAILR[15:0])
- GPTM 计时器 B 间隔加载寄存器 (GPT:TBILR[15:0])
- GPTM 计时器 A 寄存器 (GPT:TAR[15:0])
- GPTM 计时器 B 寄存器 (GPT:TBR[15:0])
- GPTM 计时器 A 值寄存器 (GPT:TAV[15:0])
- GPTM 计时器 B 值寄存器 (GPT:TBV[15:0])
- GPTM 计时器 A 匹配寄存器 (GPT:TAMATCHR[15:0])
- GPTM 计时器 B 匹配寄存器 (GPT:TBMATCHR[15:0])

在 32 位模式下，GPTM 将对 GPT:TAILR 寄存器的 32 位写访问转换为对 GPT:TAILR 和 GPT:TBILR 寄存器的写访问。这种写操作产生的字顺序为：GPTMTBILR[15:0]:GPTMTAILR[15:0]。同样，对 GPT:TAR 寄存器的 32 位读访问返回的值为：GPTMTBR[15:0]:GPTMTAR[15:0]。对 GPT:TAV 的 32 位读访问返回的值为：GPTMTBV[15:0]:GPTMTAV[15:0]。

15.4 初始化和配置

按照以下顺序初始化和配置计时器：

1. 要使用 GPT 模块，需要通过写入 PRCM:GPTCLKGR、PRCM:GPTCLKGS 和 PRCM:GPTCLKGDS 寄存器，或使用以下驱动库函数，在 PRCM 中使能外设域和相应的 GPT 模块：

```
PRCMPeripheralRunEnable(uint32_t, ui32Peripheral)
```

```
PRCMPeripheralSleepEnable(uint32_t, ui32Peripheral)
```

```
PRCMPeripheralDeepSLeepEnable(uint32_t, ui32Peripheral)
```

2. 然后，通过写入 PRCM:CLKLOADCTL 寄存器，将设置加载到时钟控制器。
3. 配置 IOC 模块，将 GPT 模块的输出路由到 IO。
4. 然后必须配置 IOC 模块，以便在所需的 I/O 引脚上输出定时器信号。为此，必须将 IOCFGn.PORTID 写入正确的 PORTID（有关更多详细信息，请参阅章节 13）。

以下各节介绍了每种受支持定时器模式的模块初始化和配置示例。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

15.4.1 单次触发和周期计时器模式

按照以下顺序将 GPTM 配置为单次和周期模式：

1. 进行任何更改之前，确保禁用计时器（清除 GPT:CTL TnEN 寄存器位）。
2. 向 GPTM 配置寄存器 (GPT:CFG) 写入值 0x0000 0000。
3. 配置 GPTM 计时器 n 模式寄存器 (GPT:TnMR) TnMR 字段：
   a. 写入值 0x1 选择单次模式。
   b. 写入值 0x2 选择周期模式。
4. 选择性地配置 GPT:TnMR TnSNAPS、TnWOT、TnMTE 和 TnCDIR 位，以选择是否在超时时捕获自由运行计时器的值、使用外部触发启动计数、配置附加触发器或中断，以及选择向上或向下计数。
5. 将起始值加载到 GPTM 计时器 n 间隔加载寄存器 (GPT:TnILR) 中。
6. 如果需要中断，在 GPTM 中断屏蔽寄存器 (GPT:IMR) 中设置相应的位。
7. 设置 GPT:CTL TnEN 寄存器位以使能计时器并开始计数。
8. 轮询 GPT:MRIS 寄存器，或等待生成中断（如果已使能）。在这两种情况下，都通过向 GPTM 中断清除寄存器 (GPT:ICR) 的相应位写入 1 来清除状态标志。

在单次模式下，计时器在超时事件发生后停止计数。要重新使能计时器，请重复该序列。配置为周期模式的计时器在超时事件后会重新加载定时器并继续计数。

15.4.2 输入边沿计数模式

按照以下序列将计时器配置为输入边沿计数模式：

1. 进行任何更改之前，确保禁用计时器（清除 TAEN 位）。
2. 向 GPTM 配置寄存器 (GPT:CFG) 写入值 0x0000 0004。
3. 在 GPTM 计时器模式寄存器 (GPT:TnMR) 中，将 TnCMR 字段写入 0x0，并将 TnMR 字段写入 0x3。
4. 通过写入 GPTM 控制寄存器 (GPT:CTL) TnEVENT 字段来配置计时器捕获的事件类型。
5. 如果要使用预分频器，请将预分频值写入 GPTM 计时器 n 预分频寄存器 (GPT:TnPR)。
6. 将计时器起始值加载到 GPTM 计时器 n 间隔加载寄存器 (GPT:TnILR) 中。
7. 将事件计数加载到 GPTM 计时器 n 匹配寄存器 (GPT:TnMATCHR) 中。
8. 如果需要中断，则设置 GPTM 中断屏蔽寄存器 (GPT:IMR) CnMIM 位。
9. 设置 GPT:CTL TnEN 寄存器位以启用计时器并开始等待边沿事件。
10. 轮询 GPT:RIS CnMRIS 寄存器位，或等待生成中断（如果启用）。在这两种情况下，可以通过向 GPTM 中断清除寄存器 (GPT:ICR) CnMCINT 位写入 1 来清除状态标志。

在输入边沿计数模式下向下计数时，计时器会在检测到编程数量的边沿事件后停止。要重新启用计时器，请确保清除 TnEN 位，并重复步骤 4 至步骤 9。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

15.4.3 输入边沿计时模式

按照以下序列将计时器配置为输入边沿计时模式：

1. 进行任何更改之前，确保禁用计时器 (TAEN 位已经清零) 。
2. 向 GPTM 配置寄存器 (GPT:CFG) 写入值 0x0000 0004。
3. 在 GPTM 计时器模式寄存器 (GPT:TnMR) 中，将 TnCM 字段写入 0x1，并将 TnMR 字段写入 0x3。
4. 通过写入 GPTM 控制寄存器 (GPT:CTL) TnEVENT 字段来配置计时器捕获的事件类型。
5. 如果要使用预分频器，请将预分频值写入 GPTM 计时器 n 预分频寄存器 (GPT:TnPR)。
6. 将计时器起始值加载到 GPTM 计时器 n 间隔加载寄存器 (GPT:TnILR) 中。
7. 如果需要中断，则设置 GPTM 中断屏蔽寄存器 (GPT:IMR) CnMIM 位。
8. 设置 GPT:CTL TnEN 寄存器位以启用计时器并开始计数。
9. 轮询 GPT:RIS CnMRIS 寄存器位，或等待生成中断 (如果启用) 。在这两种情况下，可以通过向 GPTM 中断清除寄存器 (GPT:ICR) CnMCINT 位写入 1 来清除状态标志。

在输入边沿计时模式下，计时器在检测到边沿事件后继续运行，但可以通过写入 GPT:TnILR 寄存器来随时更改计时器间隔。更改在写入后的下一个周期生效。

15.4.4 PWM 模式

按照以下序列将计时器配置为 PWM 模式：

1. 进行任何更改之前，确保禁用计时器 (清除 TnEN 位) 。
2. 向 GPTM 配置寄存器 (GPT:CFG) 写入值 0x0000 0004。
3. 在 GPTM 计时器模式寄存器 (GPT:TnMR) 中，将 TnCMR 字段写入 0x1，并将 TnMR 字段写入 0x2。
4. 在 GPTM 控制寄存器 (GPT:CTL) TnPWML 字段中配置 PWM 信号的输出状态 (是否反转) 。
5. 如果要使用预分频器，请将预分频值写入 GPTM 计时器 n 预分频寄存器 (GPT:TnPR)。
6. 如果使用 PWM 中断，在 GPT:CTL TnEVENT 寄存器字段中配置中断条件，并通过设置 GPT:TnMR TnPWMIE 寄存器位来使能中断。
7. 将计时器起始值加载到 GPTM 计时器 n 间隔加载寄存器 (GPT:TnILR) 中。
8. 将匹配值加载到 GPTM 计时器 n 匹配寄存器 (GPT:TnMATCHR) 中。
9. 设置 GPTM 控制寄存器 (GPT:CTL) TnEN 位以使能定时器，并开始生成输出 PWM 信号。

在 PWM 时序模式下，计时器在 PWM 信号生成后继续运行。可以通过写入 GPT:TnILR 寄存器随时调整 PWM 周期，更改将在写入后的下一个周期生效。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

15.4.5 生成 DMA 触发事件

GPT 可以通过事件处理程序生成 DMA 触发事件。通过事件结构为 µDMA 通道选择触发源，可以将单次或突发请求传递给 µDMA 控制器。每个计时器仅为 A 和 B 各产生一个信号，但该信号可以在事件模块中被选择为单次或突发类型。DMA 完成中断被路由回发起触发的计时器模块。

使用以下步骤通过 GPT 事件配置 µDMA 触发：

1. 配置 GPT 的操作。
2. 配置 GPT：DMAEV 寄存器，为 DMA 使能相应的定时器事件。应用程序可以为每个计时器选择匹配、捕获或超时事件。
3. 配置事件结构（请参阅章节 5）以选择相应的计时器。事件结构为 GPT DMA 事件支持 5 个通道，其中 4 个专用于 GPT 模块。这些专用通道为：9、10、11 和 12。对于 GPT DMA 事件，通道 9 至 12 支持单次请求和突发请求。第 5 个支持的通道是 14，该通道可配置用于 GPT 支持，并且仅处理突发请求。配置通过 EVENT:UDMACHcrSEL 寄存器完成，其中 *c* 是通道号，*r* 是 S（单次）或 B（突发）选项。通道 14 可以使用 EVENT:UDMACH14BSEL 寄存器进行配置。每个计时器仅为 A 和 B 各产生一个信号，但该信号可以在事件模块中被选择为单次或突发类型。
4. 重新使能 GPT 模块。
5. DMA 完成中断被路由回发起触发的计时器模块。GPT:RIS 寄存器中的 DMAnRIS 位提供 DMA 传输完成的信息。

15.5 GPTM 寄存器

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

15.5.1 GPT 寄存器

表 15-7 列出了 GPT 寄存器的存储器映射寄存器。表 15-7 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

表 15-7. GPT 寄存器

偏移	首字母缩写词	寄存器名称	部分
0h	CFG	配置	CFG 寄存器（偏移 = 0h）[复位 = 00000000h]
4h	TAMR	计时器 A 模式	TAMR 寄存器（偏移 = 4h）[复位 = 00000000h]
8h	TBMR	计时器 B 模式	TBMR 寄存器（偏移 = 8h）[复位 = 00000000h]
Ch	CTL	控制	CTL 寄存器（偏移 = Ch）[复位 = 00000000h]
10h	SYNC	同步寄存器	SYNC 寄存器（偏移 = 10h）[复位 = 00000000h]
18h	IMR	中断屏蔽	IMR 寄存器（偏移 = 18h）[复位 = 00000000h]
1Ch	RIS	原始中断状态	RIS 寄存器（偏移 = 1Ch）[复位 = 00000000h]
20h	MIS	已屏蔽中断状态	MIS 寄存器（偏移 = 20h）[复位 = 00000000h]
24h	ICLR	中断清除	ICLR 寄存器（偏移 = 24h）[复位 = 00000000h]
28h	TAILR	计时器 A 间隔加载寄存器	TAILR 寄存器（偏移 = 28h）[复位 = FFFFFFFFh]
2Ch	TBILR	计时器 B 间隔加载寄存器	TBILR 寄存器（偏移 = 2Ch）[复位 = 0000FFFFh]
30h	TAMATCHR	计时器 A 匹配寄存器	TAMATCHR 寄存器（偏移 = 30h）[复位 = FFFFFFFFh]
34h	TBMATCHR	计时器 B 匹配寄存器	TBMATCHR 寄存器（偏移 = 34h）[复位 = 0000FFFFh]
38h	TAPR	计时器 A 预分频	TAPR 寄存器（偏移 = 38h）[复位 = 00000000h]
3Ch	TBPR	计时器 B 预分频	TBPR 寄存器（偏移 = 3Ch）[复位 = 00000000h]
40h	TAPMR	计时器 A 预分频匹配	TAPMR 寄存器（偏移 = 40h）[复位 = 00000000h]
44h	TBPMR	计时器 B 预分频匹配	TBPMR 寄存器（偏移 = 44h）[复位 = 00000000h]
48h	TAR	计时器 A 寄存器	TAR 寄存器（偏移 = 48h）[复位 = FFFFFFFFh]
4Ch	TBR	计时器 B 寄存器	TBR 寄存器（偏移 = 4Ch）[复位 = 0000FFFFh]
50h	TAV	计时器 A 值	TAV 寄存器（偏移 = 50h）[复位 = FFFFFFFFh]
54h	TBV	计时器 B 值	TBV 寄存器（偏移 = 54h）[复位 = 0000FFFFh]
5Ch	TAPS	计时器 A 预分频快照	TAPS 寄存器（偏移 = 5Ch）[复位 = 00000000h]
60h	TBPS	计时器 B 预分频快照	TBPS 寄存器（偏移 = 60h）[复位 = 00000000h]

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 15-7. GPT 寄存器（续）

偏移	首字母缩写词	寄存器名称	部分
64h	TAPV	计时器 A 预分频值	TAPV 寄存器（偏移 = 64h）[复位 = 00000000h]
68h	TBPV	计时器 B 预分频值	TBPV 寄存器（偏移 = 68h）[复位 = 00000000h]
6Ch	DMAEV	DMA 事件	DMAEV 寄存器（偏移 = 6Ch）[复位 = 00000000h]
FB0h	VERSION	外设版本	版本寄存器（偏移 = FB0h）[复位 = 00000400h]
FB4h	ANDCCP	组合 CCP 输出	ANDCCP 寄存器（偏移 = FB4h）[复位 = 00000000h]

复杂的位访问类型经过编码可适应小型表单元。表 15-8 展示了适用于此部分中访问类型的代码。

表 15-8. GPT 访问类型代码

访问类型	代码	说明
读取类型		
R	R	读取
写入类型		
W	W	写入
W1C	W 1C	写入 1 以清零
复位或默认值		
-n		复位后的值或默认值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

15.5.1.1 CFG 寄存器（偏移 = 0h）[复位 = 00000000h]

图 15-9 展示了 CFG，表 15-9 中对此进行了介绍。

返回到汇总表。

配置

图 15-9. CFG 寄存器

31	30	29	28	27	26	25	24	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																													CFG		
R-0h																													R/W-0h		

表 15-9. CFG 寄存器字段说明

位	字段	类型	复位	说明
31-3	RESERVED	R	0h	保留
2-0	CFG	R/W	0h	GPT 配置 0x2-0x3 — 保留 0x5-0x7 — 保留 0h = 32BIT_TIMER：32 位计时器配置 4h = 16BIT_TIMER：16 位计时器配置 为两个 16 位计时器配置。 另请参阅 TAMR.TAMR 和 TBMR.TBMR。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

15.5.1.2 TAMR 寄存器（偏移 = 4h）[复位 = 00000000h]

图 15-10 展示了 TAMR，表 15-10 中对此进行了介绍。

返回到汇总表。

计时器 A 模式

图 15-10. TAMR 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
TCACT			TACINTD	TAPLO	TAMRSU	TAPWMIE	TAILD
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
TASNAPS	TAWOT	TAMIE	TACDIR	TAAMS	TACM	TAMR	
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	

表 15-10. TAMR 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-13	TCACT	R/W	0h	计时器比较操作选择 0h = DIS_CMP：禁用比较操作 1h = 超时发生时翻转状态 2h = 超时发生时将 CCP 输出引脚清零 3h = 超时发生时将 CCP 输出引脚置位 4h = 立即将 CCP 输出引脚置位，超时发生时翻转 5h = 立即将 CCP 输出引脚清零，超时发生时翻转 6h = 立即将 CCP 输出引脚置位，超时发生时清零 7h = 立即将 CCP 输出引脚清零，超时发生时置位
12	TACINTD	R/W	0h	单次/周期中断禁用 0h = 超时中断功能正常 1h = 禁用超时中断
11	TAPLO	R/W	0h	GPTM 计时器 A PWM 传统操作 0 传统操作，当计时器达到 0 后重新加载 TAILR 寄存器时，CCP 引脚驱动为低电平。 1 在计时器达到 0 后重新加载 TAILR 寄存器时，CCP 引脚驱动为高电平。 该位仅在 PWM 模式下有效。 0h = 传统操作 1h = 超时发生时，CCP 输出引脚设置为 1
10	TAMRSU	R/W	0h	计时器 A 匹配寄存器更新模式 该位定义 TAMATCHR 和 TAPR 寄存器的更新时间。 如果该位置位时定时器处于禁用状态 (CTL.TAEN = 0)，则在使能计时器时更新 TAMATCHR 和 TAPR。 如果该位置位时计时器处于暂停状态 (CTL.TASTALL = 1)，则根据该位的配置更新 TAMATCHR 和 TAPR。 0h = 在下一个周期更新 TAMATCHR 和 TAPR（如果使用）。 1h = 在下一次超时发生时更新 TAMATCHR 和 TAPR（如果使用）。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 15-10. TAMR 寄存器字段说明（续）

位	字段	类型	复位	说明
9	TAPWMIE	R/W	0h	GPTM 计时器 A PWM 中断使能 该位在 PWM 模式下，根据 CTL.TAEVENT 的定义，使能 CCP 输出上升沿、下降沿或双边沿的中断。 此外，当该位被置位且发生捕获事件时，如果通过分别设置 CTL.TAOTE 位和 DMAEV.CAEDMAEN 位使能触发功能，则定时器 A 会自动向 DMA 产生触发信号。 0 禁用捕获事件中断。 1 使能捕获事件中断。 该位仅在 PWM 模式下有效。 0h = 禁用中断 1h = 使能中断此位仅可在 PWM 模式下有效。
8	TAILD	R/W	0h	GPT 计时器 A PWM 间隔加载写入 0h =在下一个时钟周期使用 TAILR 寄存器中的值更新 TAR 寄存器。如果使用预分频器，则在下一个时钟周期使用 TAPR 寄存器中的值更新 TAPS 寄存器。 1h = 在下一次超时发生时使用 TAILR 寄存器中的值更新 TAR 寄存器。如果使用预分频器，则在下一次超时发生时使用 TAPR 寄存器中的值更新 TAPS 寄存器。
7	TASNAPS	R/W	0h	GPT 计时器 A 快照模式 0h = 禁用快照模式。 1h = 如果计时器 A 配置为周期性模式，则计时器 A 的实际自由运行值会在超时事件发生时加载到 GPT 计时器 A (TAR) 寄存器中。
6	TAWOT	R/W	0h	GPT 计时器 A 等待触发 0h = 计时器 A 在使能后立即开始计数。 1h = 如果已使能计时器 A (CTL.TAEN = 1)，则计时器 A 只有在接收到来自菊花链中前一级计时器的触发信号后才开始计数。对于 GPT 模块 0 的计时器 A，必须将此位清零。此功能对单次模式、周期模式和 PWM 模式均有效
5	TAMIE	R/W	0h	GPT 计时器 A 匹配中断使能 0h = 对匹配事件禁用匹配中断。此外，也会阻止匹配事件上的输出触发。 1h = 在单次模式和周期模式下，达到 TAMATCHR 中的匹配值时生成中断。
4	TACDIR	R/W	0h	GPT 计时器 A 计数方向 0h = 向下：计时器递减计数。 1h = 向上：计时器递增计数。递增计数时，计时器从 0x0 值开始。
3	TAAMS	R/W	0h	GPT 计时器 A 交替模式 注意：要使能 PWM 模式，还必须先将 TACM 清零，然后将 TAMR 字段配置为 0x2。 0h = 使能捕获/比较模式。 1h = 使能 PWM 模式
2	TACM	R/W	0h	GPT 计时器 A 捕获模式 0h = EDGCNT：边沿计数模式 1h = EDGTIME：边沿时间模式
1-0	TAMR	R/W	0h	GPT 计时器 A 模式 0x0 保留 0x1 单次计时器模式 0x2 周期计时器模式 0x3 捕获模式 计时器模式基于由 CFG 寄存器中的位 2:0 定义的计时器配置 1h = 单次计时器模式 2h = 周期计时器模式 3h = 捕获模式

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

15.5.1.3 TBMR 寄存器（偏移 = 8h）[复位 = 00000000h]

图 15-11 展示了 TBMR，表 15-11 中对此进行了介绍。

返回到汇总表。

计时器 B 模式

图 15-11. TBMR 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
TCACT			TBCINTD	TBPLO	TBMRSU	TBPWMIE	TBILD
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
TBSNAPS	TBWOT	TBMIE	TBCDIR	TBAMS	TBCM	TBMR	
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	

表 15-11. TBMR 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-13	TCACT	R/W	0h	计时器比较操作选择 0h = DIS_CMP：禁用比较操作 1h = 超时发生时翻转状态 2h = 超时发生时将 CCP 输出引脚清零 3h = 超时发生时将 CCP 输出引脚置位 4h = 立即将 CCP 输出引脚置位，超时发生时翻转 5h = 立即将 CCP 输出引脚清零，超时发生时翻转 6h = 立即将 CCP 输出引脚置位，超时发生时清零 7h = 立即将 CCP 输出引脚清零，超时发生时置位
12	TBCINTD	R/W	0h	单次/周期中断模式 0h =正常超时中断 1h = 屏蔽超时中断
11	TBPLO	R/W	0h	GPTM 计时器 A PWM 传统操作 0 传统操作，当计时器达到 0 后重新加载 TBILR 寄存器时，CCP 引脚驱动为低电平。 1 在计时器达到 0 后重新加载 TBILR 寄存器时，CCP 引脚驱动为高电平。 该位仅在 PWM 模式下有效。 0h = 传统操作 1h = 超时发生时，CCP 输出引脚设置为 1
10	TBMRSU	R/W	0h	计时器 B 匹配寄存器更新模式 该位定义 TBMATCHR 和 TBPR 寄存器的更新时间。 如果该位置位时定时器处于禁用状态（CTL.TBEN 已清零），则在启用计时器时更新 TBMATCHR 和 TBPR。 如果该位置位时计时器处于暂停状态（CTL.TBSTALL 已置位），则根据该位的配置更新 TBMATCHR 和 TBPR。 0h = 在下一个周期更新 TBMATCHR 和 TBPR（如果使用）。 1h = 在下一次超时发生时更新 TBMATCHR 和 TBPR（如果使用）。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 15-11. TBMR 寄存器字段说明（续）

位	字段	类型	复位	说明
9	TBPWMIE	R/W	0h	GPTM 计时器 B PWM 中断使能 该位在 PWM 模式下，根据 CTL.TBEVENT 的定义，使能 CCP 输出上升沿、下降沿或双边沿的中断。 此外，当该位被置位且发生捕获事件时，如果通过分别设置 CTL.TBOTE 位和 DMAEV.CBEDMAEN 位使能触发功能，则定时器 A 会自动向 DMA 产生触发信号。 0 禁用捕获事件中断。 1 使能捕获事件中断。 该位仅在 PWM 模式下有效。 0h = 禁用中断 1h = 使能中断此位仅可在 PWM 模式下有效。
8	TBILD	R/W	0h	GPT 计时器 B PWM 间隔加载写入 0h = 在下一个时钟周期使用 TBILR 寄存器中的值更新 TBR 寄存器。如果使用预分频器，则在下一个时钟周期使用 TBPR 寄存器中的值更新 TBPS 寄存器。 1h = 在下一次超时发生时使用 TBILR 寄存器中的值更新 TBR 寄存器。如果使用预分频器，则在下一次超时发生时使用 TBPR 寄存器中的值更新 TBPS 寄存器。
7	TBSNAPS	R/W	0h	GPT 计时器 B 快照模式 0h = 禁用快照模式。 1h = 如果计时器 B 配置为周期模式
6	TBWOT	R/W	0h	GPT 计时器 B 等待触发 0h = 计时器 B 在使能后立即开始计数。 1h = 如果已使能计时器 B（CTL.TBEN 已置位），则计时器 B 只有在接收到来自菊花链中前一级计时器的触发信号后才开始计数。此功能对单次模式、周期模式和 PWM 模式有效
5	TBMIE	R/W	0h	GPT 计时器 B 匹配中断使能。 0h = 对匹配事件禁用匹配中断。此外，也会阻止匹配事件上的输出触发。 1h = 在单次模式和周期模式下，达到 TBMATCHR 中的匹配值时生成中断。
4	TBCDIR	R/W	0h	GPT 计时器 B 计数方向 0h = 向下：计时器递减计数。 1h = 向上：计时器递增计数。递增计数时，计时器从 0x0 值开始。
3	TBAMS	R/W	0h	GPT 计时器 B 交替模式 注意：要启用 PWM 模式、您还必须将 TBCM 位清零并将 TBMR 字段配置为 0x2。 0h = 使能捕获/比较模式。 1h = 使能 PWM 模式
2	TBCM	R/W	0h	GPT 计时器 B 捕获模式 0h = EDGCNT：边沿计数模式 1h = EDGTIME：边沿时间模式
1-0	TBMR	R/W	0h	GPT 计时器 B 模式 0x0 保留 0x1 单次计时器模式 0x2 周期计时器模式 0x3 捕获模式 计时器模式基于由 CFG 寄存器中的位 2:0 定义的计时器配置 1h = 单次计时器模式 2h = 周期计时器模式 3h = 捕获模式

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

15.5.1.4 CTL 寄存器（偏移 = Ch）[复位 = 00000000h]

图 15-12 展示了 CTL，表 15-12 中对此进行了介绍。

返回到汇总表。

控制

图 15-12. CTL 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED	TBPWML	RESERVED		TBEVENT		TBSTALL	TBEN
R-0h	R/W-0h	R-0h		R/W-0h		R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED	TAPWML	RESERVED		TAEVENT		TASTALL	TAEN
R-0h	R/W-0h	R-0h		R/W-0h		R/W-0h	R/W-0h

表 15-12. CTL 寄存器字段说明

位	字段	类型	复位	说明
31-15	RESERVED	R	0h	保留
14	TBPWML	R/W	0h	GPT 计时器 B PWM 输出电平 0：输出不受影响。 1：输出反相。 0h = 未反相 1h = 反相
13-12	RESERVED	R	0h	保留
11-10	TBEVENT	R/W	0h	GPT 计时器 B 事件模式 此寄存器中的值定义如下： 值说明 0x0 上升沿 0x1 下降沿 0x2 保留 0x3 双边沿 注意：如果已使能 PWM 输出反相，则边沿检测中断 行为也会反转。因此，如果已设置上升沿中断触发， 而 PWM 反相产生了上升沿， 则不会将事件触发中断置为有效，而是在 PWM 信号的下降沿生成中断 。 0h = 上升沿 1h = 下降沿 3h = 双边沿
9	TBSTALL	R/W	0h	GPT 计时器 B 暂停使能 0h = 计时器 B 在调试器暂停处理器时继续计数。 1h = 计时器 B 在调试器暂停处理器时停止计数。
8	TBEN	R/W	0h	GPT 计时器 B 使能 0h = 禁用计时器 B。 1h = 使能计时器 B 并开始计数，或根据 CFG 寄存器使能捕获逻辑。
7	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 15-12. CTL 寄存器字段说明（续）

位	字段	类型	复位	说明
6	TAPWML	R/W	0h	GPT 计时器 A PWM 输出电平 0h = 未反相 1h = 反相
5-4	RESERVED	R	0h	保留
3-2	TAEVENT	R/W	0h	GPT 计时器 A 事件模式 此寄存器中的值定义如下： 值说明 0x0 上升沿 0x1 下降沿 0x2 保留 0x3 双边沿 注意：如果已使能 PWM 输出反相，则边沿检测中断 行为也会反转。因此，如果已设置上升沿中断触发， 而 PWM 反相产生了上升沿， 则不会将事件触发中断置为有效，而是在 PWM 信号的下降沿生成中断 。 0h = 上升沿 1h = 下降沿 3h = 双边沿
1	TASTALL	R/W	0h	GPT 计时器 A 暂停使能 0h = 计时器 A 在调试器暂停处理器时继续计数。 1h = 计时器 A 在调试器暂停处理器时停止计数。
0	TAEN	R/W	0h	GPT 计时器 A 使能 0h = 禁用计时器 A。 1h = 使能计时器 A 并开始计数，或根据 CFG 寄存器使能捕获逻辑。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

15.5.1.5 SYNC 寄存器（偏移 = 10h）[复位 = 00000000h]

图 15-13 展示了 SYNC，表 15-13 中对此进行了介绍。

返回到汇总表。

同步寄存器

图 15-13. SYNC 寄存器

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								SYNC3		SYNC2		SYNC1		SYNC0	
R-0h								W-0h		W-0h		W-0h		W-0h	

表 15-13. SYNC 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-6	SYNC3	W	0h	同步 GPT 计时器 3。 0h = 不同步 GPT3 不受影响。 1h = 触发 GPT3 的计时器 A 的超时事件 2h = 触发 GPT3 的计时器 B 的超时事件 3h = 触发 GPT3 的计时器 A 和计时器 B 的超时事件
5-4	SYNC2	W	0h	同步 GPT 计时器 2。 0h = 不同步 GPT2 不受影响。 1h = 触发 GPT2 的计时器 A 的超时事件 2h = 触发 GPT2 的计时器 B 的超时事件 3h = 触发 GPT2 的计时器 A 和计时器 B 的超时事件
3-2	SYNC1	W	0h	同步 GPT 计时器 1 0h = 不同步。GPT1 不受影响。 1h = 触发 GPT1 的计时器 A 的超时事件 2h = 触发 GPT1 的计时器 B 的超时事件 3h = 触发 GPT1 的计时器 A 和计时器 B 的超时事件
1-0	SYNC0	W	0h	同步 GPT 计时器 0 0h = 不同步。GPT0 不受影响。 1h = 触发 GPT0 的计时器 A 的超时事件 2h = 触发 GPT0 的计时器 B 的超时事件 3h = 触发 GPT0 的计时器 A 和计时器 B 的超时事件

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

15.5.1.6 IMR 寄存器（偏移 = 18h）[复位 = 00000000h]

图 15-14 展示了 IMR，表 15-14 中对此进行了介绍。

返回到汇总表。

中断屏蔽
该寄存器用于使能中断。
相关寄存器：
RIS、MIS、ICLR

图 15-14. IMR 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED		DMABIM	RESERVED	TBMIM	CBEIM	CBMIM	TBTOIM
R-0h		R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED		DMAAIM	TAMIM	RESERVED	CAEIM	CAMIM	TATOIM
R-0h		R/W-0h	R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h

表 15-14. IMR 寄存器字段说明

位	字段	类型	复位	说明
31-14	RESERVED	R	0h	保留
13	DMABIM	R/W	0h	使能该位将使 RIS.DMABRIS 中断传播到 MIS.DMABMIS 0h = 禁用中断 1h = 使能中断
12	RESERVED	R	0h	保留
11	TBMIM	R/W	0h	使能该位将使 RIS.TBMRIS 中断传播到 MIS.TBMMIS 0h = 禁用中断 1h = 使能中断
10	CBEIM	R/W	0h	使能该位将使 RIS.CBERIS 中断传播到 MIS.CBEMIS 0h = 禁用中断 1h = 使能中断
9	CBMIM	R/W	0h	使能该位将使 RIS.CBMRIS 中断传播到 MIS.CBMMIS 0h = 禁用中断 1h = 使能中断
8	TBTOIM	R/W	0h	使能该位将使 RIS.TBTORIS 中断传播到 MIS.TBTOMIS 0h = 禁用中断 1h = 使能中断
7-6	RESERVED	R	0h	保留
5	DMAAIM	R/W	0h	使能该位将使 RIS.DMAARIS 中断传播到 MIS.DMAAMIS 0h = 禁用中断 1h = 使能中断
4	TAMIM	R/W	0h	使能该位将使 RIS.TAMRIS 中断传播到 MIS.TAMMIS 0h = 禁用中断 1h = 使能中断
3	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 15-14. IMR 寄存器字段说明（续）

位	字段	类型	复位	说明
2	CAEIM	R/W	0h	使能该位将使 RIS.CAERIS 中断传播到 MIS.CAEMIS 0h = 禁用中断 1h = 使能中断
1	CAMIM	R/W	0h	使能该位将使 RIS.CAMRIS 中断传播到 MIS.CAMMIS 0h = 禁用中断 1h = 使能中断
0	TATOIM	R/W	0h	使能该位将使 RIS.TATORIS 中断传播到 MIS.TATOMIS 0h = 禁用中断 1h = 使能中断

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

15.5.1.7 RIS 寄存器（偏移 = 1Ch）[复位 = 00000000h]

图 15-15 展示了 RIS，表 15-15 中对此进行了介绍。

返回到汇总表。

原始中断状态
相关寄存器：
IMR、MIS、ICLR

图 15-15. RIS 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED		DMABRIS	RESERVED	TBMRIS	CBERIS	CBMRIS	TBTORIS
R-0h		R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED		DMAARIS	TAMRIS	RESERVED	CAERIS	CAMRIS	TATORIS
R-0h		R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

表 15-15. RIS 寄存器字段说明

位	字段	类型	复位	说明
31-14	RESERVED	R	0h	保留
13	DMABRIS	R	0h	GPT 计时器 B DMA 完成原始中断状态 0：传输尚未完成 1：传输已完成
12	RESERVED	R	0h	保留
11	TBMRIS	R	0h	GPT 计时器 B 匹配原始中断 0：尚未达到匹配值 1：已达到匹配值。 TBMR.TBMIE 被置位，且在配置为单次模式或周期模式时，已达到 TBMATCHR 和 TBPMR（可选）中的匹配值。
10	CBERIS	R	0h	GPT 计时器 B 捕获模式事件原始中断 0：事件尚未发生。 1：事件已发生。 当子计时器配置为输入边沿计时模式时，此中断置为有效
9	CBMRIS	R	0h	GPT 计时器 B 捕获模式匹配原始中断 0：计时器 B 的捕获模式匹配尚未发生。 1：计时器 B 的捕获模式匹配已发生。当配置为输入边沿计时模式时， 如果 TBR 和 TBPR 中的值 与 TBMATCHR 和 TBPMR 中的值匹配， 则此中断置为有效。 向 ICLR.CBMCINT 位写入 1 可以清除该位。
8	TBTORIS	R	0h	GPT 计时器 B 超时原始中断 0：计时器 B 未超时 1：计时器 B 已超时。 当单次模式或周期模式计时器达到其计数限制时，此中断置为有效。 计数限制为 0 或加载到 TBILR 中的值，具体取决于计数方向。
7-6	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 15-15. RIS 寄存器字段说明（续）

位	字段	类型	复位	说明
5	DMAARIS	R	0h	GPT 计时器 A DMA 完成原始中断状态 0：传输尚未完成 1：传输已完成
4	TAMRIS	R	0h	GPT 计时器 A 匹配原始中断 0：尚未达到匹配值 1：已达到匹配值。 TAMR.TAMIE 被置位，且在配置为单次模式或周期模式时，已达到 TAMATCHR 和 TAPMR（可选）中的匹配值。
3	RESERVED	R	0h	保留
2	CAERIS	R	0h	GPT 计时器 A 捕获模式事件原始中断 0：事件尚未发生。 1：事件已发生。 当子计时器配置为输入边沿计时模式时，此中断置为有效
1	CAMRIS	R	0h	GPT 计时器 A 捕获模式匹配原始中断 0：计时器 A 的捕获模式匹配尚未发生。 1：计时器 A 的捕获模式匹配已发生。当配置为输入边沿计时模式时， 如果 TAR 和 TAPR 中的值 与 TAMATCHR 和 TAPMR 中的值匹配， 则此中断置为有效。 向 ICLR.CAMCINT 位写入 1 可以清除该位。
0	TATORIS	R	0h	GPT 计时器 A 超时原始中断 0：计时器 A 未超时 1：计时器 A 已超时。 当单次模式或周期模式计时器达到其计数限制时，此中断置为有效。 计数限制为 0 或加载到 TAILR 中的值，具体取决于计数方向。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

15.5.1.8 MIS 寄存器（偏移 = 20h）[复位 = 00000000h]

图 15-16 展示了 MIS，表 15-16 中对此进行了介绍。

返回到汇总表。

屏蔽的中断状态
值是 RIS 和 IMR 之间按位与运算的结果
相关的清除寄存器：ICLR

图 15-16. MIS 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED		DMABMIS	RESERVED	TBMMIS	CBEMIS	CBMMIS	TBTOMIS
R-0h		R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED		DMAAMIS	TAMMIS	RESERVED	CAEMIS	CAMMIS	TATOMIS
R-0h		R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

表 15-16. MIS 寄存器字段说明

位	字段	类型	复位	说明
31-14	RESERVED	R	0h	保留
13	DMABMIS	R	0h	0：无中断或中断未使能 1：RIS.DMABRIS = 1 且 IMR.DMABIM = 1
12	RESERVED	R	0h	保留
11	TBMMIS	R	0h	0：无中断或中断未使能 1：RIS.TBMRIS = 1 且 IMR.TBMIM = 1
10	CBEMIS	R	0h	0：无中断或中断未使能 1：RIS.CBERIS = 1 且 IMR.CBEIM = 1
9	CBMMIS	R	0h	0：无中断或中断未使能 1：RIS.CBMRIS = 1 且 IMR.CBMIM = 1
8	TBTOMIS	R	0h	0：无中断或中断未使能 1：RIS.TBTORIS = 1 且 IMR.TBTOIM = 1
7-6	RESERVED	R	0h	保留
5	DMAAMIS	R	0h	0：无中断或中断未使能 1：RIS.DMAARIS = 1 且 IMR.DMAAIM = 1
4	TAMMIS	R	0h	0：无中断或中断未使能 1：RIS.TAMRIS = 1 且 IMR.TAMIM = 1
3	RESERVED	R	0h	保留
2	CAEMIS	R	0h	0：无中断或中断未使能 1：RIS.CAERIS = 1 且 IMR.CAEIM = 1
1	CAMMIS	R	0h	0：无中断或中断未使能 1：RIS.CAMRIS = 1 且 IMR.CAMIM = 1
0	TATOMIS	R	0h	0：无中断或中断未使能 1：RIS.TATORIS = 1 且 IMR.TATOIM = 1

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

15.5.1.9 ICLR 寄存器（偏移 = 24h）[复位 = 00000000h]

图 15-17 展示了 ICLR，表 15-17 中对此进行了介绍。

返回到汇总表。

中断清除
该寄存器用于清除 RIS 和 MIS 寄存器中的状态位

图 15-17. ICLR 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED		DMABINT	RESERVED	TBMCINT	CBECINT	CBMCINT	TBTOCINT
R-0h		R/W1C-0h	R-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h
7	6	5	4	3	2	1	0
RESERVED		DMAAINT	TAMCINT	RESERVED	CAECINT	CAMCINT	TATOCINT
R-0h		R/W1C-0h	R/W1C-0h	R-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h

表 15-17. ICLR 寄存器字段说明

位	字段	类型	复位	说明
31-14	RESERVED	R	0h	保留
13	DMABINT	R/W1C	0h	0：无操作。 1：清除 RIS.DMABRIS 和 MIS.DMABMIS
12	RESERVED	R	0h	保留
11	TBMCINT	R/W1C	0h	0：无操作。 1：清除 RIS.TBMRIS 和 MIS.TBMMIS
10	CBECINT	R/W1C	0h	0：无操作。 1：清除 RIS.CBERIS 和 MIS.CBEMIS
9	CBMCINT	R/W1C	0h	0：无操作。 1：清除 RIS.CBMRIS 和 MIS.CBMMIS
8	TBTOCINT	R/W1C	0h	0：无操作。 1：清除 RIS.TBTORIS 和 MIS.TBTOMIS
7-6	RESERVED	R	0h	保留
5	DMAAINT	R/W1C	0h	0：无操作。 1：清除 RIS.DMAARIS 和 MIS.DMAAMIS
4	TAMCINT	R/W1C	0h	0：无操作。 1：清除 RIS.TAMRIS 和 MIS.TAMMIS
3	RESERVED	R	0h	保留
2	CAECINT	R/W1C	0h	0：无操作。 1：清除 RIS.CAERIS 和 MIS.CAEMIS
1	CAMCINT	R/W1C	0h	0：无操作。 1：清除 RIS.CAMRIS 和 MIS.CAMMIS
0	TATOCINT	R/W1C	0h	0：无操作。 1：清除 RIS.TATORIS 和 MIS.TATOMIS

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

15.5.1.10 TAILR 寄存器（偏移 = 28h）[复位 = FFFFFFFFh]

图 15-18 展示了 TAILR，表 15-18 中对此进行了介绍。

返回到汇总表。

计时器 A 间隔加载寄存器

图 15-18. TAILR 寄存器

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

表 15-18. TAILR 寄存器字段说明

位	字段	类型	复位	说明
31-0	TAILR	R/W	FFFFFFFFh	GPT 计时器 A 间隔加载寄存器 写入该字段会加载计时器 A 的计数器。读取操作会返回 TAILR 的当前值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

15.5.1.11 TBILR 寄存器（偏移 = 2Ch）[复位 = 0000FFFFh]

图 15-19 展示了 TBILR，表 15-19 中对此进行了介绍。

返回到汇总表。

计时器 B 间隔加载寄存器

图 15-19. TBILR 寄存器

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

表 15-19. TBILR 寄存器字段说明

位	字段	类型	复位	说明
31-0	TBILR	R/W	FFFFh	GPT 计时器 B 间隔加载寄存器 写入该字段会加载计时器 B 的计数器。读取操作会返回 TBILR 的当前值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

15.5.1.12 TAMATCHR 寄存器（偏移 = 30h）[复位 = FFFFFFFFh]

图 15-20 展示了 TAMATCHR，表 15-20 中对此进行了介绍。

返回到汇总表。

定时器 A 匹配寄存器
在单次模式或周期模式下，当定时器值等于该寄存器中的值时，可以产生中断。
在边沿计数模式下，该寄存器与 TAILR 共同决定计数的边沿事件数量。
计数的边沿事件总数为 TAILR 中的值减去该值。
注意，在边沿计数模式下执行向上计数时，TAPR 和 TAILR 的值必须大于 TAPMR 和该寄存器的值。
在 PWM 模式下，该值与 TAILR 共同决定输出 PWM 信号的占空比。
当 16/32 位 GPT 配置为 32 位模式之一时，TAMATCHR 表现为一个 32 位寄存器。（高 16 位对应于 TBMATCHR 的内容）。
在 16 位模式下，该寄存器的高 16 位读数为 0，对 TBMATCHR 的状态没有影响。
注意：该寄存器根据 TAMR.TAMRSU 进行内部更新（生效）

图 15-20. TAMATCHR 寄存器

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

表 15-20. TAMATCHR 寄存器字段说明

位	字段	类型	复位	说明
31-0	TAMATCHR	R/W	FFFFFFFFh	GPT 计时器 A 匹配寄存器

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

15.5.1.13 TBMATCHR 寄存器（偏移 = 34h）[复位 = 0000FFFFh]

图 15-21 展示了 TBMATCHR，表 15-21 中对此进行了介绍。

返回到汇总表。

计时器 B 匹配寄存器
当 GPT 配置为 32 位模式之一时，该寄存器中位 15:0 的内容会被加载到 TAMATCHR 的高 16 位。
读取该寄存器将返回计时器 B 的当前匹配值，写操作将被忽略。
在 16 位模式下，位 15:0 用作匹配值。在这两种情况下，位 31:16 都是保留位。
注意：该寄存器根据 TBMR.TBMRSU 进行内部更新（生效）

图 15-21. TBMATCHR 寄存器

31	30	29	28	27	26	25	24	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																TBMATCHR															
R-0h																R/W-FFFFh															

表 15-21. TBMATCHR 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	TBMATCHR	R/W	FFFFh	GPT 计时器 B 匹配寄存器

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

15.5.1.14 TAPR 寄存器（偏移 = 38h）[复位 = 00000000h]

图 15-22 展示了 TAPR，表 15-22 中对此进行了介绍。

返回到汇总表。

计时器 A 预分频
该寄存器允许软件在计时器单独使用时扩展计时器的范围。
在单次模式或周期性递减计数模式下，该寄存器充当计时器计数器的真正预分频器。
当用作真正的预分频器时，预分频器会在 TAR 和 TAV 寄存器中的值递增之前向下计数到 0。
在所有其他独立/拆分模式下，该寄存器是计时器计数器上限范围的线性扩展，在 16/32 位 GPT 的 16 位模式下，用于存放位 23:16。

图 15-22. TAPR 寄存器

31	30	29	28	27	26	25	24	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																								TAPSR							
R-0h																								R/W-0h							

表 15-22. TAPR 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-0	TAPSR	R/W	0h	计时器 A 预分频。 在单次和周期计数模式下，预分频器比率为 TAPSR + 1，即： 0：预分频器比率 = 1 1：预分频器比率 = 2 2：预分频器比率= 3 ... 255：预分频器比率 = 256

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

15.5.1.15 TBPR 寄存器（偏移 = 3Ch）[复位 = 00000000h]

图 15-23 展示了 TBPR，表 15-23 中对此进行了介绍。

返回到汇总表。

计时器 B 预分频
该寄存器允许软件在计时器单独使用时扩展计时器的范围。
在单次模式或周期性递减计数模式下，该寄存器充当计时器计数器的真正预分频器。
当用作真正的预分频器时，预分频器会在 TBR 和 TBV 寄存器中的值递增之前向下计数到 0。
在所有其他独立/拆分模式下，该寄存器是计时器计数器上限范围的线性扩展，在 16/32 位 GPT 的 16 位模式下，用于存放位 23:16。

图 15-23. TBPR 寄存器

31	30	29	28	27	26	25	24	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																								TBPSR							
R-0h																								R/W-0h							

表 15-23. TBPR 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-0	TBPSR	R/W	0h	计时器 B 预分频。 在单次和周期计数模式下，预分频器比率为 TBPSR + 1，即： 0：预分频器比率 = 1 1：预分频器比率 = 2 2：预分频器比率= 3 ... 255：预分频器比率 = 256

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

15.5.1.16 TAPMR 寄存器（偏移 = 40h）[复位 = 00000000h]

图 15-24 展示了 TAPMR，表 15-24 中对此进行了介绍。

返回到汇总表。

计时器 A 预分频匹配
该寄存器允许软件在 TAMATCHR 单独使用时扩展 TAMATCHR 的范围。

图 15-24. TAPMR 寄存器

31	30	29	28	27	26	25	24	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																								TAPSMR							
R-0h																								R/W-0h							

表 15-24. TAPMR 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-0	TAPSMR	R/W	0h	GPT 计时器 A 预分频匹配。在 16 位模式下，该字段用于存放位 23 至 16。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

15.5.1.17 TBPMR 寄存器（偏移 = 44h）[复位 = 00000000h]

图 15-25 展示了 TBPMR，表 15-25 中对此进行了介绍。

返回到汇总表。

计时器 B 预分频匹配
该寄存器允许软件在 TBMATCHR 单独使用时扩展 TBMATCHR 的范围。

图 15-25. TBPMR 寄存器

31	30	29	28	27	26	25	24	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																								TBPSMR							
R-0h																								R/W-0h							

表 15-25. TBPMR 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-0	TBPSMR	R/W	0h	GPT 计时器 B 预分频匹配寄存器。在 16 位模式下，该字段用于存放位 23 至 16。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

15.5.1.18 TAR 寄存器（偏移 = 48h）[复位 = FFFFFFFFh]

图 15-26 展示了 TAR，表 15-26 中对此进行了介绍。

返回到汇总表。

计时器 A 寄存器
除输入边沿计数模式和输入边沿计时模式外，该寄存器在所有情况下均显示计时器 A 计数器的当前值。在输入边沿计数模式下，该寄存器包含
已发生的边沿数。在输入边沿计时模式下，该寄存器包含最后一个边沿事件发生的时间。
当 GPT 配置为 32 位模式之一时，该寄存器显示为 32 位寄存器（高 16 位对应于计时器 B (TBR) 寄存器的内容）。在
16 位输入边沿计数模式、输入边沿计时模式和 PWM 模式下，位 15:0 包含计数器的值，位 23:16 包含预分频器的值，即计数的高 8 位。位
31:24 始终读为 0。要在 16 位单次模式和周期模式下读取预分频器的值，请读取 TAV 寄存器中的位 [23:16]。要在周期性快照模式下读取预分频器的值，
请读取计时器 A 预分频快照 (TAPS) 寄存器。

图 15-26. TAR 寄存器

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

表 15-26. TAR 寄存器字段说明

位	字段	类型	复位	说明
31-0	TAR	R	FFFFFFFFh	GPT 计时器 A 寄存器 根据寄存器字段 TAMR.TAILD 中的值，该寄存器将在下一个周期或下一次超时发生时使用 TAILR 寄存器中的值进行更新。 除输入边沿计数模式和计时模式之外，读取操作在所有情况下都会返回计时器 A 计数寄存器的当前值。 在输入边沿计数模式下，该寄存器包含已发生的边沿数。在输入边沿计时模式下，该寄存器包含最后一个边沿事件发生的时间。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

15.5.1.19 TBR 寄存器（偏移 = 4Ch）[复位 = 0000FFFFh]

图 15-27 展示了 TBR，表 15-27 中对此进行了介绍。

返回到汇总表。

计时器 B 寄存器
除输入边沿计数模式和输入边沿计时模式外，该寄存器在所有情况下均显示计时器 B 计数器的当前值。在输入边沿计数模式下，该寄存器包含
已发生的边沿数。在输入边沿计时模式下，该寄存器包含最后一个边沿事件发生的时间。
当 GPTM 配置为 32 位模式之一时，该寄存器中位 15:0 的内容将被加载到 TAR 寄存器的高 16 位中。读取该寄存器会返回计时器 B 的当前
值。在 16 位模式下，位 15:0 包含计数器的值，位 23:16 包含输入边沿计数、输入边沿计时和 PWM 模式下预分频器的值，即
计数值的高 8 位。位 31:24 始终读为 0。要在 16 位单次模式和周期模式下读取预分频器的值，请读取 TBV 寄存器中的位 [23:16]。要在周期性快照模式下读取预分频器的值，
请读取计时器 B 预分频快照 (TBPS) 寄存器。

图 15-27. TBR 寄存器

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

表 15-27. TBR 寄存器字段说明

位	字段	类型	复位	说明
31-0	TBR	R	FFFFh	GPT 计时器 B 寄存器 根据寄存器字段 TBMR.TBILD 中的值，该寄存器将在下一个周期或下一次超时发生时使用 TBILR 寄存器中的值进行更新。 除输入边沿计数模式和计时模式之外，读取操作在所有情况下都会返回计时器 B 计数寄存器的当前值。 在输入边沿计数模式下，该寄存器包含已发生的边沿数。在输入边沿计时模式下，该寄存器包含最后一个边沿事件发生的时间。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

15.5.1.20 TAV 寄存器（偏移 = 50h）[复位 = FFFFFFFFh]

图 15-28 展示了 TAV，表 15-28 中对此进行了介绍。

返回到汇总表。

计时器 A 值
读取时，该寄存器在所有模式下均显示计时器 A 的当前自由运行值。在周期性运行模式下使用快照功能时，软件可以使用该值来确定从中断发生到 ISR 入口经过的时间。执行写操作时，写入该寄存器的值将在下一个时钟周期被加载到 TAR 寄存器中。
当 16/32 位 GPTM 配置为 32 位模式之一时，该寄存器表现为 32 位寄存器（高 16 位对应于 GPTM 计时器 B 值 (TBV) 寄存器的内容）。在 16 位模式下，位 15:0 包含计数器的值,位 23:16 包含预分频器的当前自由运行值，即输入边沿计数模式、输入边沿计时模式、PWM 模式以及单次或周期性递增计数模式中计数的高 8 位。在单次或周期性
递减计数模式下，存储在 23:16 中的预分频器是真正预分频器，这意味着位 23:16 会在位 15:0 中的值递减之前递减。位 31:24 中的预分频器读数始终为 0。

图 15-28. TAV 寄存器

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

表 15-28. TAV 寄存器字段说明

位	字段	类型	复位	说明
31-0	TAV	R/W	FFFFFFFFh	GPT 计时器 A 寄存器 读取操作时，在所有模式下均返回计时器 A 的当前自由运行值。 写入操作时，写入该寄存器的值将在下一个时钟周期加载到 TAR 寄存器中。 注意：在 16 位模式下，只能向该寄存器的低 16 位写入一个新值。写入预分频器位无影响

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

15.5.1.21 TBV 寄存器（偏移 = 54h）[复位 = 0000FFFFh]

图 15-29 展示了 TBV，表 15-29 中对此进行了介绍。

返回到汇总表。

计时器 B 值
读取时，该寄存器在所有模式下均显示计时器 B 的当前自由运行值。软件可以使用该值来确定从中断发生到 ISR 入口经过的时间。执行
写操作时，写入该寄存器的值将在下一个时钟周期被加载到 TBR 寄存器中。
当 16/32 位 GPTM 配置为 32 位模式之一时，该寄存器中位 15:0 的内容将加载到 TAV 寄存器的高 16 位中。读取该寄存器会返回
计时器 B 的当前自由运行值。在 16 位模式下，位 15:0 包含计数器的值，位 23:16 包含预分频器的当前自由运行值，即输入边沿计数模式、输入边沿计时模式、PWM 模式以及单次或周期性递增计数模式中
计数的高 8 位。
在单次或周期性递减计数模式下，存储在 23:16 中的预分频器是真正预分频器，这意味着位 23:16 会在位 15:0 中的值递减之前递减。位 31:24 中的预分频器读数始终为 0。

图 15-29. TBV 寄存器

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

表 15-29. TBV 寄存器字段说明

位	字段	类型	复位	说明
31-0	TBV	R/W	FFFFh	GPT 计时器 B 寄存器 读取操作时，在所有模式下均返回计时器 B 的当前自由运行值。 写入操作时，写入该寄存器的值将在下一个时钟周期加载到 TBR 寄存器中。 注意：在 16 位模式下，只能向该寄存器的低 16 位 写入一个新值。写入预分频器位无影响

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

15.5.1.22 TAPS 寄存器（偏移 = 5Ch）[复位 = 00000000h]

图 15-30 展示了 TAPS，表 15-30 中对此进行了介绍。

返回到汇总表。

计时器 B 预分频快照
根据寄存器字段 TAMR.TAILD 中的值，该寄存器将在下一个周期或下一个超时使用 TAPR 寄存器中的值进行更新。
该寄存器显示 16 位模式下计时器 A 预分频器的当前值。

图 15-30. TAPS 寄存器

31	30	29	28	27	26	25	24	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																								PSS							
R-0h																								R-0h							

表 15-30. TAPS 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-0	PSS	R	0h	GPT 计时器 A 预分频器

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

15.5.1.23 TBPS 寄存器（偏移 = 60h）[复位 = 00000000h]

图 15-31 展示了 TBPS，表 15-31 中对此进行了介绍。

返回到汇总表。

计时器 B 预分频快照
根据寄存器字段 TBMR.TBILD 中的值，该寄存器将在下一个周期或下一个超时使用 TBPR 寄存器中的值进行更新。
该寄存器显示 16 位模式下计时器 B 预分频器的当前值。

图 15-31. TBPS 寄存器

31	30	29	28	27	26	25	24	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																								PSS							
R-0h																								R-0h							

表 15-31. TBPS 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-0	PSS	R	0h	GPT 计时器 B 预分频器

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

15.5.1.24 TAPV 寄存器（偏移 = 64h）[复位 = 00000000h]

图 15-32 展示了 TAPV，表 15-32 中对此进行了介绍。

返回到汇总表。

计时器 A 预分频值
该寄存器显示在 16 位模式下计时器 A 自由运行预分频器的当前值。

图 15-32. TAPV 寄存器

31	30	29	28	27	26	25	24	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																								PSV							
R-0h																								R-0h							

表 15-32. TAPV 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-0	PSV	R	0h	GPT 计时器 A 预分频器值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

15.5.1.25 TBPV 寄存器（偏移 = 68h）[复位 = 00000000h]

图 15-33 展示了 TBPV，表 15-33 中对此进行了介绍。

返回到汇总表。

计时器 B 预分频值
该寄存器显示在 16 位模式下计时器 B 自由运行预分频器的当前值。

图 15-33. TBPV 寄存器

31	30	29	28	27	26	25	24	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																								PSV							
R-0h																								R-0h							

表 15-33. TBPV 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-0	PSV	R	0h	GPT 计时器 B 预分频器值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

15.5.1.26 DMAEV 寄存器（偏移 = 6Ch）[复位 = 00000000h]

图 15-34 展示了 DMAEV，表 15-34 中对此进行了介绍。

返回到汇总表。

DMA 事件
该寄存器允许软件启用/禁用 GPT DMA 触发事件。

图 15-34. DMAEV 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED				TBMDMAEN	CBEDMAEN	CBMDMAEN	TBTODMAEN
R-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED			TAMDMAEN	RESERVED	CAEDMAEN	CAMDMAEN	TATODMAEN
R/W-0h			R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h

表 15-34. DMAEV 寄存器字段说明

位	字段	类型	复位	说明
31-12	RESERVED	R	0h	软件不应依赖保留字段的值。写入任何其他值都可能导致未定义的行为。
11	TBMDMAEN	R/W	0h	GPT 计时器 B 匹配 DMA 触发使能
10	CBEDMAEN	R/W	0h	GPT 计时器 B 捕获事件 DMA 触发使能
9	CBMDMAEN	R/W	0h	GPT 计时器 B 捕获匹配 DMA 触发使能
8	TBTODMAEN	R/W	0h	GPT 计时器 B 超时 DMA 触发使能
7-5	RESERVED	R/W	0h	软件不应依赖保留字段的值。写入任何其他值都可能导致未定义的行为。
4	TAMDMAEN	R/W	0h	GPT 计时器 A 匹配 DMA 触发使能
3	RESERVED	R	0h	保留
2	CAEDMAEN	R/W	0h	GPT 计时器 A 捕获事件 DMA 触发使能
1	CAMDMAEN	R/W	0h	GPT 计时器 A 捕获匹配 DMA 触发使能
0	TATODMAEN	R/W	0h	GPT 计时器 A 超时 DMA 触发使能

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

15.5.1.27 版本寄存器（偏移 = FB0h）[复位 = 00000400h]

图 15-35 展示了 VERSION，表 15-35 中对此进行了介绍。

返回到汇总表。

外设版本
此寄存器提供有关 GPT 版本的信息

图 15-35. 版本寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VERSION																															
R-400h																															

表 15-35. 版本寄存器字段说明

位	字段	类型	复位	说明
31-0	VERSION	R	400h	计时器版本。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

15.5.1.28 ANDCCP 寄存器（偏移 = FB4h）[复位 = 00000000h]

图 15-36 展示了 ANDCCP，表 15-36 中对此进行了介绍。

返回到汇总表。

组合 CCP 输出
此寄存器用于对每个计时器的 CCP 输出对进行逻辑“与”运算

图 15-36. ANDCCP 寄存器

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

表 15-36. ANDCCP 寄存器字段说明

位	字段	类型	复位	说明
31-2	RESERVED	R	0h	保留
1	LD_TO_EN	R/W	0h	超时发生时，PWM 被置为有效 0：当计数器与加载值匹配时，PWM 被置为有效 1：当计数器超时时，PWM 被置为有效
0	CCP_AND_EN	R/W	0h	使能计时器 A 和 B 的 CCP 输出的“与”运算。 0：计时器 A 和计时器 B 的 PWM 输出是相应计时器内部生成的 PWM 信号。 1：计时器 A 的 PWM 输出是计时器 A 和计时器 B 的 PWM 信号相与后的结果，计时器 B 的 PWM 输出仅为计时器 B 的 PWM 信号。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

章节 16

实时时钟 (RTC)

本章介绍了 CC13x2 和 CC26x2 器件平台的常开实时时钟 (AON_RTC) 的功能和设计。

16.1 简介 1379
16.2 功能规格 1379
16.3 RTC 寄存器信息 1381
16.4 RTC 寄存器 1382

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

16.1 简介

本节介绍了 CC13x2 和 CC26x2 器件平台的常开实时时钟 (AON_RTC) 的功能和设计。AON_RTC 实现了一个秒和亚秒计数器，支持对 ppm 偏移进行软件补偿，并具有 3 个匹配寄存器和 1 个比较寄存器。

AON_RTC 具有一种特殊机制，支持在其继续运行时关闭 MCU 域电源。除了最深度的断电模式，即关断模式，AON_RTC 在所有电源模式下均保持供电。

16.2 功能规格

本节提供了 AON_RTC 的功能说明。

16.2.1 功能概述

AON_RTC 的功能说明如下：

- 在 32kHz 时钟的正边沿递增
- 一个 70 位的递增计数器，支持可编程增量以支持 ppm 调整
- 三个通用通道（0、1 和 2），带有支持事件生成的比较器
- 软件和硬件复位事件
- 所有事件都可以延迟可编程的时间量，以生成相应的延迟事件。
- 一组可编程的延迟事件可以组合起来生成延迟的组合事件。

16.2.2 自由运行计数器

AON_RTC 实现了一个 70 位自由运行的计数器，每个 32kHz 时钟递增一个可编程的值。该可编程值允许补偿 32kHz 时钟的 ppm 偏移，从而使计数器能够以非常高的精度运行。

在 AON_RTC 上电后启用时，计数器从 0 开始计数，但也可以由软件复位为 0 或任何其他新值。计数器测量秒（32 位）和亚秒（32 位）。

默认情况下，AON_RTC 以每个 32kHz 时钟周期 1/32768 秒递增其计数器。亚秒增量值 0x20 000 对应于 1/32768 秒。增加或减少亚秒增量值会同比增加或减少 AON_RTC 的速度。

通过更新 AUX_SYSIF:RTCSUBSECINC0 和 AUX_SYSIF:RTCSUBSECINC1 寄存器来更改增量，然后通过写入 AUX_SYSIF:RTCSUBSECINCCTL.UPD_REQ 寄存器将新设置加载到 AON_RTC。在接收到来自 AON_RTC 的确认之前，AUX 不得更改新的亚秒增量值。可以从 AUX_SYSIF:RTCSUBSECINCCTL.UPD_ACK 寄存器读取确认信息。在收到确认后，可以将 AUX_SYSIF:RTCSUBSECINCCTL.UPD_REQ 寄存器写回 0，如果需要，可以上传新的亚秒增量。

为了对自由运行计数器执行原子读取，必须首先读取秒部分 AON_RTC:SEC。这将锁存亚秒部分 AON_RTC:SUBSEC，直到被读取。此外，AON_RTC:TIME 寄存器返回 AON_RTC:SEC 寄存器的下半字和 AON_RTC:SUBSEC 寄存器的上半字。这与匹配寄存器和捕获寄存器的格式相同。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

16.2.3 通道

AON_RTC 包含三个独立的通道（0、1 和 2），它们的行为略有不同。

所有通道都可以在比较模式下运行；当达到或超过可编程的时间限制时，每个通道都会生成一个比较事件。这是通道 0 的唯一工作模式。

通道 1 可以在捕获模式下运行，在该模式下，外部事件会导致自由运行计数器的当前值被锁存，以记住事件发生的时间。随后会生成一个捕获事件。由于通道 1 计时器在比较模式或捕获模式下运行，每种模式下都会生成相同的物理事件。

通道 2 可以在连续比较模式下运行，在比较事件发生后自动递增其比较值。这样能够生成完全等距的事件。

图 16-1 显示了三个 AON_RTC 通道。

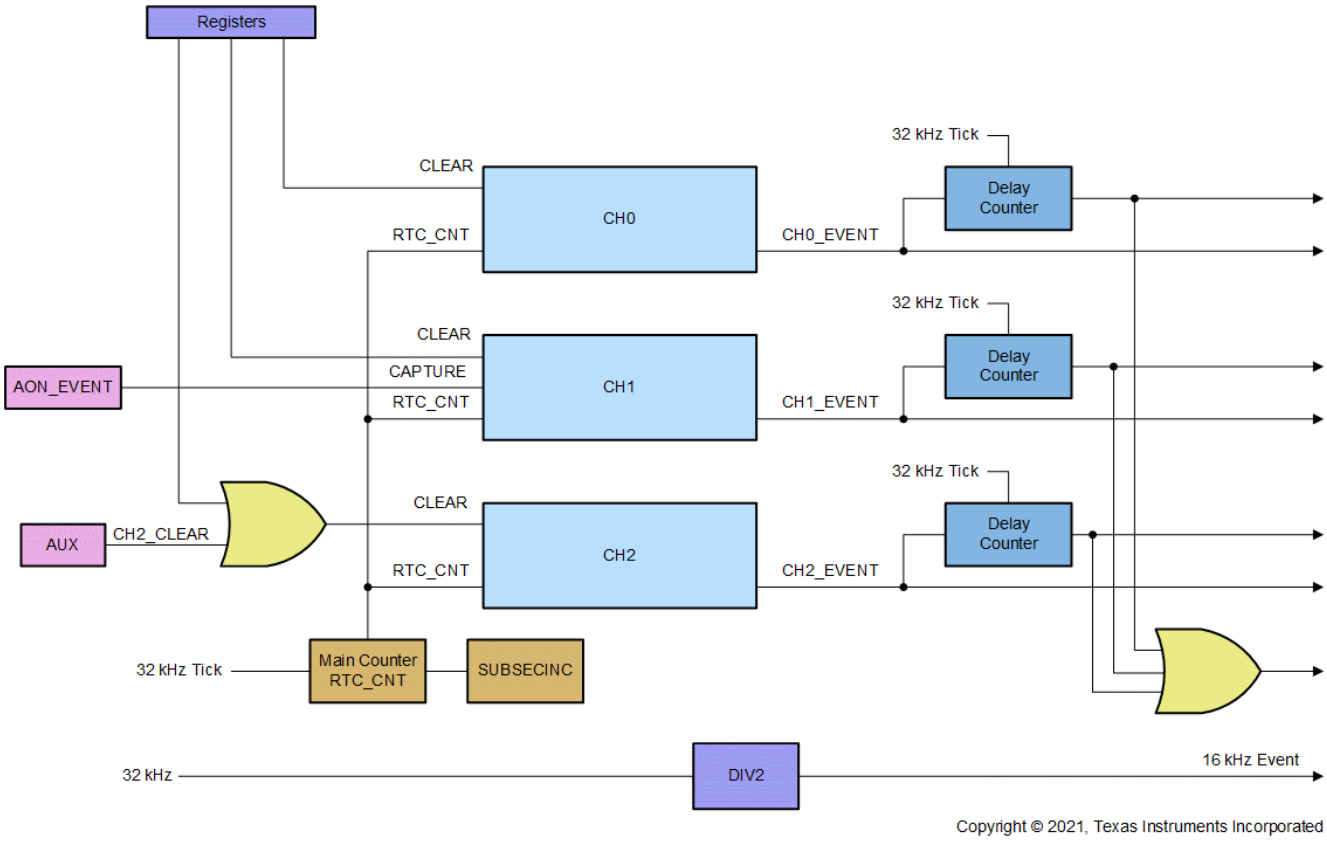


图 16-1. AON_RTC 通道

16.2.3.1 捕获和比较

通过在 AON_EVENT:RTCSEL 中配置捕获源，并在 AON_RTC:CHCTL 中将通道 1 设置为捕获模式，可在通道 1 上设置 RTC 捕获事件。捕获事件发生后，捕获的 RTC 值在寄存器 AON_RTC:CH1CAPT 中可用

比较事件通过写入相应的通道比较寄存器 AON_RTC:CHnCMP 进行配置。

备注

如果设置的比较值使得比较值减当前值大于秒回绕时间减 1 秒 ($2^{32} \times SCLK_LF_{period} - 1$)，则会立即设置一个比较事件，以避免丢失该事件。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

16.2.4 事件

三个延迟事件的可编程组合可以组合成第七个延迟事件。所有事件都被馈送到 AON 事件结构中，作为 MCU 或 AUX 域的唤醒事件等使用。

三个通道可以单独生成一个事件。这三个事件均可使用延迟计数器，将通道事件延迟可编程数量的时钟来生成相应的事件。延迟计数器使用未补偿的 32kHz 时钟；因此，延迟时间会随该时钟变化。此过程可以在未来生成精确的事件，即使在事件发生后必须唤醒 MCU 等情况下也是如此 — 此过程需要一段不确定但有界的时间。

备注

禁用通道不会清除该通道中的任何待处理事件。清除事件的唯一方法是将外部清除信号置为有效，或向 AON_RTC:EVFLAGS 寄存器中的相应 CHx 位写入 1。

16.3 RTC 寄存器信息

RTC 寄存器位于 AON 域中，并使用 2MHz MF 时钟进行时钟控制。除 SHUTDOWN 模式外，所有配置和状态寄存器在所有电源模式下均得以保持。MCU 域包含一个连接到 AON_RTC 寄存器的接口，以确保在系统总线上以最小延迟进行快速访问。由于 MCU 接口与 AON 域之间存在同步，系统中会存在一定的延迟，软件必须将其考虑在内。

16.3.1 寄存器访问

写入访问会延迟一个或两个 2MHz MF 时钟周期。系统总线不受此延迟影响，因此 MCU 在实际写入 AON_RTC 寄存器之前就完成了总线事务。此过程允许应用程序连续写入多个寄存器，而不会因同步而产生任何额外延迟。

由于同步的原因，读访问读取的值总是延迟两到三个系统时钟 (48MHz)。在这种情况下，系统总线不会暂停。

AON_RTC:EVFLAGS 寄存器具有快速清除特性。当写入 1 时，MCU 会立即清除 EVFLAGS 位字段。如果该状态被用作中断或事件，此过程可使 MCU 快速清除其源。由于同步的原因，RTC 中的实际标志要等到 1 或 2 个时钟周期后才会被清除。因此，新事件会在最多两个 2MHz MF 周期内被屏蔽。

16.3.2 进入睡眠和从睡眠中唤醒

在进入睡眠之前，必须确保所有对 AON 寄存器的写入请求都已完成。这是由 MCU 域中的硬件完成的。

从睡眠中唤醒后，应用程序必须等待一个 2MHz MF 周期。此等待可确保 MCU 域寄存器接口正确同步。如果在同步完成之前读取寄存器，读取的值可能尚未更新。例如，读取 AON_RTC:SEC 寄存器可能显示的是进入睡眠之前的值，而不是当前值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

16.3.3 AON_RTC:SYNC 寄存器

AON_RTC:SYNC 寄存器用于在 MCU 域和 AON 域之间同步。

如果存在对 AON 寄存器未完成的写入请求，则来自 AON_RTC:SYNC 寄存器的读取请求不会返回；换句话说，总线会暂停，直到所有未完成的请求完成。

写入请求会触发对 AON 域的虚拟写入。该写入操作可以确保与 MF 时钟同步。该虚拟写入操作需要一到两个 2MHz MF 时钟周期。

1. 写入 AON_RTC:SYNC 寄存器或 AON 域中的任何其他寄存器。该写入操作会触发未完成的写入请求在 AON 域上注册。
2. 从 AON_RTC:SYNC 寄存器读取。在所有未完成的请求完成之前，该读取操作不会返回。

当必须确保特定顺序时，通常使用 AON_RTC:SYNC 寄存器操作。例如，禁用通道时，可以轮询 AON_RTC:SYNC 寄存器以确保该通道已被禁用并且不会发生进一步事件：

1. 设置 AON_RTC：CHCTL.CH2_EN = 0。
2. 读取 AON_RTC：SYNC 寄存器。
3. 通道现已禁用。不会发生进一步事件。

另一个典型用途是确保唤醒时在 MCU 域中更新正确的值。该 MCU 域仅在 2MHz MF 时钟的正边沿更新。

1. 写入 AON_RTC:SYNC 寄存器。
2. 读取 AON_RTC：SYNC 寄存器。
3. 现在可以安全地读取其他 AON_RTC 寄存器，因为其信息已正确更新。

16.4 RTC 寄存器

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

16.4.1 AON_RTC 寄存器

表 16-1 列出了 AON_RTC 寄存器的存储器映射寄存器。表 16-1 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

表 16-1. AON_RTC 寄存器

偏移	首字母缩写词	寄存器名称	部分
0h	CTL	控制	CTL 寄存器（偏移 = 0h）[复位 = 00000000h]
4h	EVFLAGS	事件标志，RTC 状态	EVFLAGS 寄存器（偏移 = 4h）[复位 = 00000000h]
8h	SEC	第二个计数器值，整数部分	SEC 寄存器（偏移 = 8h）[复位 = 00000000h]
Ch	SUBSEC	第二个计数器值，分数部分	SUBSEC 寄存器（偏移 = Ch）[复位 = 00000000h]
10h	SUBSECINC	亚秒增量	SUBSECINC 寄存器（偏移 = 10h）[复位 = 00800000h]
14h	CHCTL	通道配置	CHCTL 寄存器（偏移 = 14h）[复位 = 00000000h]
18h	CH0CMP	通道 0 比较值	CH0CMP 寄存器（偏移 = 18h）[复位 = 00000000h]
1Ch	CH1CMP	通道 1 比较值	CH1CMP 寄存器（偏移 = 1Ch）[复位 = 00000000h]
20h	CH2CMP	通道 2 比较值	CH2CMP 寄存器（偏移 = 20h）[复位 = 00000000h]
24h	CH2CMPINC	通道 2 比较值自动递增	CH2CMPINC 寄存器（偏移 = 24h）[复位 = 00000000h]
28h	CH1CAPT	通道 1 捕获值	CH1CAPT 寄存器（偏移 = 28h）[复位 = 00000000h]
2Ch	SYNC	AON 同步	SYNC 寄存器（偏移 = 2Ch）[复位 = 00000000h]
30h	时间	当前计数器值	TIME 寄存器（偏移 = 30h）[复位 = 00000000h]
34h	SYNCLF	与 SCLK_LF 同步	SYNCLF 寄存器（偏移 = 34h）[复位 = 00000000h]

复杂的位访问类型经过编码可适应小型表单元。表 16-2 展示了适用于此部分中访问类型的代码。

表 16-2. AON_RTC 访问类型代码

访问类型	代码	说明
读取类型		
R	R	读取
写入类型		
W	W	写入
W1C	W 1C	写入 1 以清零
复位或默认值		
-n		复位后的值或默认值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

16.4.1.1 CTL 寄存器（偏移 = 0h）[复位 = 00000000h]

图 16-2 展示了 CTL，表 16-3 中对此进行了介绍。

返回到汇总表。

控制
该寄存器包含用于配置 RTC 的各种位字段
RTL 名称 = CONFIG

图 16-2. CTL 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED					COMB_EV_MASK		
R-0h					R/W-0h		
15	14	13	12	11	10	9	8
RESERVED				EV_DELAY			
R-0h				R/W-0h			
7	6	5	4	3	2	1	0
复位	RESERVED				RTC_4KHZ_EN	RTC_UPD_EN	EN
W1C-0h	R-0h				R/W-0h	R/W-0h	R/W-0h

表 16-3. CTL 寄存器字段说明

位	字段	类型	复位	说明
31-19	RESERVED	R	0h	保留
18-16	COMB_EV_MASK	R/W	0h	事件屏蔽，用于选择构成组合事件的延迟事件。 0h = 未为组合事件选择事件。 1h = 在组合事件中使用通道 0 延迟事件 2h = 在组合事件中使用通道 1 延迟事件 4h = 在组合事件中使用通道 2 延迟事件
15-12	RESERVED	R	0h	保留
11-8	EV_DELAY	R/W	0h	生成延迟事件之前等待的 SCLK_LF 时钟周期数。（所有 RTC 通道的通用设置）。延迟事件被推迟 0h = 延迟事件无延迟 1h = 延迟 1 个时钟周期 2h = 延迟 2 个时钟周期 3h = 延迟 4 个时钟周期 4h = 延迟 8 个时钟周期 5h = 延迟 16 个时钟周期 6h = 延迟 32 个时钟周期 7h = 延迟 48 个时钟周期 8h = 延迟 64 个时钟周期 9h = 延迟 80 个时钟周期 Ah = 延迟 96 个时钟周期 Bh = 延迟 112 个时钟周期 Ch = 延迟 128 个时钟周期 Dh = 延迟 144 个时钟周期
7	复位	W1C	0h	RTC 计数器复位。 向该位写入 1 将复位 RTC 计数器。 当复位生效时，该位被清除
6-3	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 16-3. CTL 寄存器字段说明（续）

位	字段	类型	复位	说明
2	RTC_4KHZ_EN	R/W	0h	RTC_4KHZ 是一个 4KHz 参考输出，从 SUBSEC.VALUE 位 19 抽头，供 AUX 计时器使用。 0：RTC_4KHZ 信号被强制为 0 1：RTC_4KHZ 使能（前提是 RTC 已通过 EN 位使能）
1	RTC_UPD_EN	R/W	0h	RTC_UPD 是用于同步无线电计时器的 16KHz 信号。16kHz 等于 SCLK_LF 除以 2 0：RTC_UPD 信号强制为 0 1：RTC_UPD 信号以 16kHz 频率切换
0	EN	R/W	0h	使能 RTC 计数器 0：暂停（冻结） 1：运行

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

16.4.1.2 EVFLAGS 寄存器（偏移 = 4h）[复位 = 00000000h]

图 16-3 展示了 EVFLAGS，表 16-4 中对此进行了介绍。

返回到汇总表。

事件标志，RTC 状态
该寄存器包含来自 3 个 RTC 通道的事件标志。向相应的位字段写入“1”时，每个标志都将被清除。

图 16-3. EVFLAGS 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															CH2
R-0h															R/W1C-0h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							CH1	RESERVED							CH0
R-0h							R/W1C-0h	R-0h							R/W1C-0h

表 16-4. EVFLAGS 寄存器字段说明

位	字段	类型	复位	说明
31-17	RESERVED	R	0h	保留
16	CH2	R/W1C	0h	通道 2 事件标志，当 CHCTL.CH2_EN = 1 且 RTC 值匹配或传递 CH2CMP 值时设置。 在写入 CH2CMP 时，只要通道已使能，且新值与下一个 RTC 值到过去 1 秒之间的任何时间点匹配，事件将被调度为尽快发生 写入 1 可清除该标志。 AUX_SCE 可以通过 AUX_EVCTL:EVSTAT2.AON_RTC_CH2 读取该标志，并使用 AUX_SYSIF:RTCEVCLR.RTC_CH2_EV_CLR 清除该标志。
15-9	RESERVED	R	0h	保留
8	CH1	R/W1C	0h	通道 1 事件标志，当 CHCTL.CH1_EN = 1 且满足以下任一条件时设置： - CHCTL.CH1_CAPT_EN = 0 且 RTC 值匹配或传递 CH1CMP 值。 - CHCTL.CH1_CAPT_EN = 1 且发生捕获。 在写入 CH1CMP 时，只要通道已使能、处于比较模式，且新值与下一个 RTC 值到过去 1 秒之间的任何时间点匹配，事件将被调度为尽快发生。 写入 1 可清除此标志。
7-1	RESERVED	R	0h	保留
0	CH0	R/W1C	0h	通道 0 事件标志，当 CHCTL.CH0_EN = 1 且 RTC 值匹配或传递 CH0CMP 值时设置。 在写入 CH0CMP 时，只要通道已使能，且新值与下一个 RTC 值到过去 1 秒之间的任何时间点匹配，事件将被调度为尽快发生。 写入 1 可清除此标志。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

16.4.1.3 SEC 寄存器（偏移 = 8h）[复位 = 00000000h]

图 16-4 展示了 SEC，表 16-5 中对此进行了介绍。

返回到汇总表。

第二个计数器值，整数部分

图 16-4. SEC 寄存器

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

表 16-5. SEC 寄存器字段说明

位	字段	类型	复位	说明
31-0	值	R/W	0h	无符号整数，以秒为单位表示实时时钟。 读取该寄存器时，SUBSEC.VALUE 的内容同时被锁存。通过首先读取该寄存器，然后读取 SUBSEC 寄存器，可以获得组合实时时钟的一致读数。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

16.4.1.4 SUBSEC 寄存器（偏移 = Ch）[复位 = 00000000h]

图 16-5 展示了 SUBSEC，表 16-6 中对此进行了介绍。

返回到汇总表。

第二个计数器值，分数部分

图 16-5. SUBSEC 寄存器

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

表 16-6. SUBSEC 寄存器字段说明

位	字段	类型	复位	说明
31-0	值	R/W	0h	无符号整数，表示读取 SEC 寄存器时刻的实时时钟，以几毫秒（VALUE/2^{32} 秒）为单位。 示例： - 0x0000_0000 = 0.0 秒 - 0x4000_0000 = 0.25 秒 - 0x8000_0000 = 0.5 秒 - 0xC000_0000 = 0.75 秒

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

16.4.1.5 SUBSECINC 寄存器（偏移 = 10h）[复位 = 00800000h]

图 16-6 展示了 SUBSECINC，表 16-7 中对此进行了介绍。

返回到汇总表。

亚秒递增
在每个 SCLK_LFclock 周期添加到 SUBSEC.VALUE 的值。

图 16-6. SUBSECINC 寄存器

31	30	29	28	27	26	25	24	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								VALUEINC																							
R-0h								R-00800000h																							

表 16-7. SUBSECINC 寄存器字段说明

位	字段	类型	复位	说明
31-24	RESERVED	R	0h	保留
23-0	VALUEINC	R	00800000h	该值可补偿从 32768Hz 偏移的 SCLK_LF 时钟。 补偿值的计算公式为 2^{38}/freq，其中 freq 为 SCLK_LF 时钟频率（以赫兹为单位） 该值在每个周期添加到 SUBSEC.VALUE 中，其进位被添加到 SEC.VALUE 上。为了执行加法，位 [23:6] 与 SUBSEC.VALUE 位 [17:0] 对齐。其余位 [5:0] 累积在一个隐藏的 6 位寄存器中，该寄存器会在溢出时产生对上述加法的进位。 默认值对应于精确递增 1/32768 秒。 注意：该寄存器是只读的。必须使用寄存器 AUX_SYSIF:RTCSUBSECINC0、AUX_SYSIF:RTCSUBSECINC1 和 AUX_SYSIF:RTCSUBSECINCCTL 来修改寄存器值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

16.4.1.6 CHCTL 寄存器（偏移 = 14h）[复位 = 00000000h]

图 16-7 展示了 CHCTL，表 16-8 中对此进行了介绍。

返回到汇总表。

通道配置

图 16-7. CHCTL 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED					CH2_CONT_EN	RESERVED	CH2_EN
R-0h					R/W-0h	R-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED						CH1_CAPT_EN	CH1_EN
R-0h						R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED							CH0_EN
R-0h							R/W-0h

表 16-8. CHCTL 寄存器字段说明

位	字段	类型	复位	说明
31-19	RESERVED	R	0h	保留
18	CH2_CONT_EN	R/W	0h	置位以使能通道 2 的连续操作。
17	RESERVED	R	0h	保留
16	CH2_EN	R/W	0h	RTC 通道 2 使能 0：禁用 RTC 通道 2 1：使能 RTC 通道 2
15-10	RESERVED	R	0h	保留
9	CH1_CAPT_EN	R/W	0h	设置通道 1 模式 0：比较模式（默认） 1：捕获模式
8	CH1_EN	R/W	0h	RTC 通道 1 使能 0：禁用 RTC 通道 1 1：使能 RTC 通道 1
7-1	RESERVED	R	0h	保留
0	CH0_EN	R/W	0h	RTC 通道 0 使能 0：禁用 RTC 通道 0 1：使能 RTC 通道 0

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

16.4.1.7 CH0CMP 寄存器（偏移 = 18h）[复位 = 00000000h]

图 16-8 展示了 CH0CMP，表 16-9 对其进行了介绍。

返回到汇总表。

通道 0 比较值

图 16-8. CH0CMP 寄存器

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

表 16-9. CH0CMP 寄存器字段说明

位	字段	类型	复位	说明
31-0	值	R/W	0h	RTC 通道 0 比较值。 位 31 至 16 表示秒，位 15 至 0 表示比较值的亚秒。 比较值与实时时钟寄存器的 SEC.VALUE (15:0) 和 SUBSEC.VALUE (31:16) 值进行比较。当 {SEC.VALUE(15:0),SUBSEC.VALUE (31:16)} 达到或超过比较值时，将生成通道 0 事件。 如果新比较值与从过去 1 秒到当前实时时钟值的实时时钟值匹配、则写入该寄存器将触发立即*) 事件。 示例： 要生成 RTC 启动后 5.5 秒的比较事件，将该值设置为 = 0x0005_8000 *) 由于同步原因，事件发生前可能需要最多一个 SCLK_LF 时钟周期。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

16.4.1.8 CH1CMP 寄存器（偏移 = 1Ch）[复位 = 00000000h]

图 16-9 展示了 CH1CMP，表 16-10 对其进行了介绍。

返回到汇总表。

通道 1 比较值

图 16-9. CH1CMP 寄存器

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

表 16-10. CH1CMP 寄存器字段说明

位	字段	类型	复位	说明
31-0	值	R/W	0h	RTC 通道 1 比较值。 位 31 至 16 表示秒，位 15 至 0 表示比较值的亚秒。 比较值与实时时钟寄存器的 SEC.VALUE (15:0) 和 SUBSEC.VALUE (31:16) 值进行比较。当 {SEC.VALUE(15:0),SUBSEC.VALUE (31:16)} 达到或超过比较值时，将生成通道 0 事件。 如果新比较值与从过去 1 秒到当前实时时钟值的实时时钟值匹配、则写入该寄存器将触发立即*) 事件。 示例： 要生成 RTC 启动后 5.5 秒的比较事件，将该值设置为 = 0x0005_8000 *) 由于同步原因，事件发生前可能需要最多一个 SCLK_LF 时钟周期。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

16.4.1.9 CH2CMP 寄存器（偏移 = 20h）[复位 = 00000000h]

图 16-10 展示了 CH2CMP，表 16-11 对其进行了介绍。

返回到汇总表。

通道 2 比较值

图 16-10. CH2CMP 寄存器

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

表 16-11. CH2CMP 寄存器字段说明

位	字段	类型	复位	说明
31-0	值	R/W	0h	RTC 通道 2 比较值。 位 31 至 16 表示秒，位 15 至 0 表示比较值的亚秒。 比较值与实时时钟寄存器的 SEC.VALUE (15:0) 和 SUBSEC.VALUE (31:16) 值进行比较。当 {SEC.VALUE(15:0),SUBSEC.VALUE (31:16)} 达到或超过比较值时，将生成通道 0 事件。 如果新比较值与从过去 1 秒到当前实时时钟值的实时时钟值匹配、则写入该寄存器将触发立即*) 事件。 示例： 要生成 RTC 启动后 5.5 秒的比较事件，将该值设置为 = 0x0005_8000 *) 由于同步原因，事件发生前可能需要最多一个 SCLK_LF 时钟周期。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

16.4.1.10 CH2CMPINC 寄存器（偏移 = 24h）[复位 = 00000000h]

图 16-11 展示了 CH2CMPINC，表 16-12 对其进行了介绍。

返回到汇总表。

通道 2 比较值自动递增
该寄存器主要用于通过 [AUX_EVCTL.EVSTAT0.AON_RTC] 事件为 AUX_SCE 模块生成周期性唤醒。

图 16-11. CH2CMPINC 寄存器

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

表 16-12. CH2CMPINC 寄存器字段说明

位	字段	类型	复位	说明
31-0	值	R/W	0h	如果设置了 CHCTL.CH2_CONT_EN，则该值在每次通道 2 比较事件发生时添加到 CH2CMP.VALUE。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

16.4.1.11 CH1CAPT 寄存器（偏移 = 28h）[复位 = 00000000h]

图 16-12 展示了 CH1CAPT，表 16-13 对其进行了介绍。

返回到汇总表。

通道 1 捕获值
如果 CHCTL.CH1_EN = 1 且 CHCTL.CH1_CAPT_EN = 1，则在 AON_EVENT:RTCSEL 中选择的事件的每个上升沿发生捕获。

图 16-12. CH1CAPT 寄存器

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

表 16-13. CH1CAPT 寄存器字段说明

位	字段	类型	复位	说明
31-16	SEC	R	0h	捕获时 SEC.VALUE 位 15:0 的值。
15-0	SUBSEC	R	0h	捕获时 SUBSEC.VALUE 位 31:16 的值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

16.4.1.12 SYNC 寄存器（偏移 = 2Ch）[复位 = 00000000h]

图 16-13 展示了 SYNC，表 16-14 中对此进行了介绍。

返回到汇总表。

AON 同步
该寄存器用于在 MCU 和整个 AON 域之间同步。

图 16-13. SYNC 寄存器

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

表 16-14. SYNC 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	WBUSY	R/W	0h	该寄存器将始终返回 0，但只有在 MCU 和 AON 之间没有未完成的写入请求时，它才会返回值 注意：在读取之前写入该寄存器将强制等待，直到下一个 SCLK_MF 边沿。当从睡眠状态唤醒后需要同步读取 AON 的寄存器时，建议执行此操作 否则可能会导致在进入睡眠状态之前读取 AON 值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

16.4.1.13 TIME 寄存器（偏移 = 30h）[复位 = 00000000h]

图 16-14 展示了 TIME，表 16-15 中对此进行了介绍。

返回到汇总表。

当前计数器值

图 16-14. TIME 寄存器

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

表 16-15. TIME 寄存器字段说明

位	字段	类型	复位	说明
31-16	SEC_L	R	0h	返回 SEC 寄存器的下半字。
15-0	SUBSEC_H	R	0h	返回 SUBSEC 寄存器的上半字。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

16.4.1.14 SYNCLF 寄存器（偏移 = 34h）[复位 = 00000000h]

图 16-15 展示了 SYNCLF，表 16-16 中对此进行了介绍。

返回到汇总表。

同步到 SCLK_LF
该寄存器用于将 MCU 同步到 SCLK_LF 的正边沿或负边沿。

图 16-15. SYNCLF 寄存器

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

表 16-16. SYNCLF 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	PHASE	R	0h	该位将始终返回 SCLK_LF 相位。返回值将延迟，直到检测到 SCLK_LF 的正边沿或负边沿。 0：SCLK_LF 的下降沿 1：SCLK_LF 的上升沿

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

章节 17

看门狗计时器 (WDT)

当系统由于软件错误或外部器件未能以预期方式响应而发生故障时，看门狗计时器 (WDT) 用于重新获得控制权。当达到超时值时，WDT 会生成不可屏蔽中断 (NMI)、常规中断或复位。此外，WDT 可以配置为在第一次超时时向微控制器 (MCU) 生成中断，并在第二次超时时生成复位信号。

17.1 简介 1400
17.2 功能说明 1400
17.3 初始化和配置 1401
17.4 WDT 寄存器 1401

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

17.1 简介

WDT 具有以下特性：

- 32 位向下计数器，带有可编程的加载寄存器
- 可编程的中断生成逻辑，支持中断屏蔽和可选的 NMI 功能
- 锁定寄存器保护，防止软件失控
- 复位生成逻辑，可启用或禁用
- 调试期间，微控制器将 CPU 暂停标志置为有效时，用户可启用停滞

WDT 可以配置为在第一次超时时向控制器生成中断，并在第二次超时时生成复位信号。一旦 WDT 配置完成，可以写入锁定寄存器以防止计时器配置被无意更改。

WDT 可以驱动两种可能的中断。中断选择通过 WDT:CTL.INTTYPE 寄存器进行控制。

17.2 功能说明

WDT 模块在启用后，当 32 位计数器到达零状态时，会生成第一个超时信号；在启用计数器的同时也会启用 WDT 中断。图 17-1 展示了 WDT 方框图。

可通过 WDT:CTL.INTTYPE 寄存器将看门狗中断配置为不可屏蔽中断 (NMI)。在首次发生超时事件后，32 位计数器会重新加载 WDT 加载寄存器 (WDT:LOAD) 中的值，并从该值开始继续倒计时。为防止 WDT 配置被软件意外修改，可通过向 WDT:LOCK 寄存器写入任意值来锁定对看门狗寄存器的写入访问。要解锁 WDT，请向 WDT:LOCK 寄存器写入值 0x1ACC E551。

如果计时器在第一次超时中断被清除之前再次向下计数至零状态，并且通过将 WDT:CTL.RESEN 寄存器设为 1 启用了复位信号，则 WDT 会将其复位信号置为有效。如果在 32 位计数器到达第二次超时之前将中断清除，则 32 位计数器会被加载为 WDT:LOAD 寄存器中的值，并从该值继续计数。

如果在 WDT 计数过程中对 WDT:LOAD 寄存器写入了新的值，则计数器会被加载为新写入的值，并继续计数。

向 WDT:LOAD 寄存器执行写入操作不会清除处于活动状态的中断。必须通过向“看门狗中断清除寄存器 (WDT:ICR)”执行写入操作来清除中断。可以根据需要启用或禁用看门狗模块中断和复位生成功能。再次启用中断后，32 位计数器会用装载寄存器的值进行预加载（而不是使用其上一次的状态）。

备注

看门狗会触发系统的热复位。该热复位可由 ICEPick 阻断，这对于调试非常有用。当 ICEPick 被激活时，热复位信号会被阻断，无法传导至系统的其余部分；不过，看门狗本身仍会被复位。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

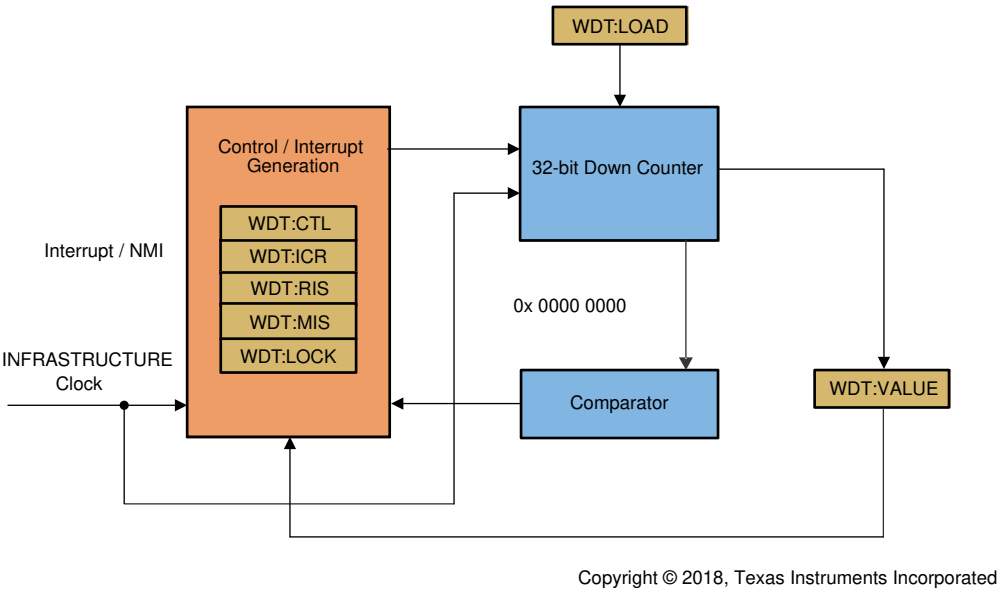


图 17-1. WDT 方框图

17.3 初始化和配置

要使用 WDT，必须使能其外设时钟。WDT 使用由 MCU PRCM 模块提供时钟源的 INFRASTRUCTURE 时钟运行。然后按照以下顺序配置 WDT：

1. 使用所需的计时器加载值加载到 WDT:LOAD 寄存器中。
2. 如果看门狗配置为触发系统复位，则设置 WDT:CTL.RESEN 位。
3. 设置 WDT:CTL.INTEN 寄存器位以使能 WDT。
4. 使用 WDT:LOCK 寄存器锁定 WDT 模块。

17.4 WDT 寄存器

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

17.4.1 WDT 寄存器

表 17-1 列出了 WDT 寄存器的存储器映射寄存器。表 17-1 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

表 17-1. WDT 寄存器

偏移	首字母缩写词	寄存器名称	部分
0h	LOAD	配置	LOAD 寄存器 (偏移 = 0h) [复位 = FFFFFFFFh]
4h	值	当前计数值	VALUE 寄存器 (偏移 = 4h) [复位 = FFFFFFFFh]
8h	CTL	控制	CTL 寄存器 (偏移 = 8h) [复位 = 00000000h]
Ch	ICR	中断清除	ICR 寄存器 (偏移 = Ch) [复位 = 00000000h]
10h	RIS	原始中断状态	RIS 寄存器 (偏移 = 10h) [复位 = 00000000h]
14h	MIS	已屏蔽中断状态	MIS 寄存器 (偏移 = 14h) [复位 = 00000000h]
418h	测试	测试模式	TEST 寄存器 (偏移 = 418h) [复位 = 00000000h]
41Ch	INT_CAUS	中断原因测试模式	INT_CAUS 寄存器 (偏移 = 41Ch) [复位 = 00000000h]
C00h	LOCK	Lock	LOCK 寄存器 (偏移 = C00h) [复位 = 00000000h]

复杂的位访问类型经过编码可适应小型表单元。表 17-2 展示了适用于此部分中访问类型的代码。

表 17-2. WDT 访问类型代码

访问类型	代码	说明
读取类型		
R	R	读取
写入类型		
W	W	写入
复位或默认值		
-n		复位后的值或默认值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

17.4.1.1 LOAD 寄存器（偏移 = 0h）[复位 = FFFFFFFFh]

图 17-2 展示了 LOAD，表 17-3 中对此进行了介绍。

返回到汇总表。

配置

图 17-2. LOAD 寄存器

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

表 17-3. LOAD 寄存器字段说明

位	字段	类型	复位	说明
31-0	WDTLOAD	R/W	FFFFFFFFh	此寄存器用于存储 32 位计数器使用的 32 位间隔值。对该寄存器执行写入操作时，该值会被立即加载，计数器重新启动并从新值开始向下计数。如果此寄存器中加载了 0x0000.0000，则会立即产生中断。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

17.4.1.2 VALUE 寄存器（偏移 = 4h）[复位 = FFFFFFFFh]

图 17-3 展示了 VALUE，表 17-4 中对此进行了介绍。

返回到汇总表。

当前计数值

图 17-3. VALUE 寄存器

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

表 17-4. VALUE 寄存器字段说明

位	字段	类型	复位	说明
31-0	WDTVALUE	R	FFFFFFFFh	该寄存器包含计时器的当前计数值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

17.4.1.3 CTL 寄存器（偏移 = 8h）[复位 = 00000000h]

图 17-4 展示了 CTL，表 17-5 中对此进行了介绍。

返回到汇总表。

控制

图 17-4. CTL 寄存器

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

表 17-5. CTL 寄存器字段说明

位	字段	类型	复位	说明
31-3	RESERVED	R	0h	保留
2	INTTYPE	R/W	0h	WDT 中断类型 0：WDT 中断是标准中断。 1：WDT 中断是不可屏蔽中断。 0h = 可屏蔽中断 1h = 不可屏蔽中断
1	RESEN	R/W	0h	WDT 复位启用。定义 WDT 复位源的功能（如已经启用，请参阅 PRCM:WARMRESET.WDT_STAT） 0：禁用。 1：启用看门狗复位输出。 0h = 复位输出已禁用 1h = 复位输出已启用
0	INTEN	R/W	0h	WDT 中断使能 0：中断事件已禁用。 1：中断事件已启用。一旦设置，该位只能通过硬件复位来清除。 0h = 禁用中断 1h = 启用中断

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

17.4.1.4 ICR 寄存器（偏移 = Ch）[复位 = 00000000h]

图 17-5 展示了 ICR，表 17-6 中对此进行了介绍。

返回到汇总表。

中断清除

图 17-5. ICR 寄存器

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

表 17-6. ICR 寄存器字段说明

位	字段	类型	复位	说明
31-0	WDTICR	W	0h	该寄存器是中断清除寄存器。向该寄存器写入任何值都会清除 WDT 中断，并从 LOAD 寄存器重新加载 32 位计数器。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

17.4.1.5 RIS 寄存器（偏移 = 10h）[复位 = 00000000h]

图 17-6 展示了 RIS，表 17-7 中对此进行了介绍。

返回到汇总表。

原始中断状态

图 17-6. RIS 寄存器

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

表 17-7. RIS 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	WDTRIS	R	0h	该寄存器是中断状态寄存器。如果控制器中断被屏蔽，则可以通过该寄存器监控 WDT 中断事件。 值说明 0：WDT 未超时 1：已发生 WDT 超时事件

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

17.4.1.6 MIS 寄存器（偏移 = 14h）[复位 = 00000000h]

图 17-7 展示了 MIS，表 17-8 中对此进行了介绍。

返回到汇总表。

已屏蔽中断状态

图 17-7. MIS 寄存器

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

表 17-8. MIS 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	WDTMIS	R	0h	该寄存器是已屏蔽中断状态寄存器。该寄存器的值是原始中断位和 WDT 中断使能位 CTL.INTEN 的逻辑与。 值说明 0：WDT 未超时或被屏蔽。 1：发生了未屏蔽的 WDT 超时事件。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

17.4.1.7 TEST 寄存器（偏移 = 418h）[复位 = 00000000h]

图 17-8 展示了 TEST，表 17-9 中对此进行了介绍。

返回到汇总表。

测试模式

图 17-8. TEST 寄存器

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

表 17-9. TEST 寄存器字段说明

位	字段	类型	复位	说明
31-9	RESERVED	R	0h	保留
8	STALL	R/W	0h	WDT 停顿使能 0：如果 CPU 因调试程序而停止，则 WDT 计时器继续计数。 1：如果 CPU 因调试程序而停止，则 WDT 停止计数。一旦 CPU 重新启动，WDT 将恢复计数。 0h = 禁用 STALL 1h = 启用 STALL
7-1	RESERVED	R	0h	保留
0	TEST_EN	R/W	0h	测试使能位 0：启用外部复位 1：禁止生成外部复位。而是设置 INT_CAUS 寄存器的位 1 并生成中断 0h = 禁用测试模式 1h = 启用测试模式

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

17.4.1.8 INT_CAUS 寄存器（偏移 = 41Ch）[复位 = 00000000h]

图 17-9 展示了 INT_CAUS，表 17-10 中对此进行了介绍。

返回到汇总表。

中断原因测试模式

图 17-9. INT_CAUS 寄存器

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

表 17-10. INT_CAUS 寄存器字段说明

位	字段	类型	复位	说明
31-2	RESERVED	R	0h	保留
1	CAUSE_RESET	R	0h	指示中断原因是复位已生成，但由于 TEST.TEST_EN 而被阻止（仅在设置 TEST.TEST_EN 时才可能出现）。
0	CAUSE_INTR	R	0h	RIS.WDTRIS 的副本

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

17.4.1.9 LOCK 寄存器（偏移 = C00h）[复位 = 00000000h]

图 17-10 展示了 LOCK，表 17-11 中对此进行了介绍。

返回到汇总表。

Lock

图 17-10. LOCK 寄存器

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

表 17-11. LOCK 寄存器字段说明

位	字段	类型	复位	说明
31-0	WDTLOCK	R/W	0h	WDT 锁定：写入值 0x1ACC.E551 会解锁看门狗寄存器以进行写入访问。写入任何其他值都会重新应用锁定，以防止任何寄存器更新（注意：TEST.TEST_EN 位不可锁定）。 读取该寄存器会返回以下值： 0x0000.0000: 已解锁 0x0000.0001：已锁定

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

章节 18

真随机数发生器 (TRNG)

真随机数发生器 (TRNG) 模块提供真实、非确定性的噪声源，用于生成密钥、初始化向量 (IV) 和其他所需的随机数。TRNG 基于 24 个环形振荡器构建而成，这些振荡器产生不可预测的输出以馈入复杂的非线性组合电路。为了获得加密安全的随机数据，需要对输出数据进行后处理。典型的应用可能（但不限于）如下：

- 生成加密密钥材料
- 生成初始化向量
- 生成 Cookie 和随机数
- 统计采样
- 通信协议中的重试计时器
- 噪声生成

18.1 简介 1413
18.2 方框图 1413
18.3 TRNG 软件复位 1414
18.4 中断请求 1414
18.5 TRNG 操作说明 1415
18.6 TRNG 底层编程指南 1417
18.7 TRNG 寄存器 1419

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

18.1 简介

该 TRNG 具有以下特性：

- TRNG 基于 24 个环形振荡器（脉冲噪声）来产生熵。为了生成此熵，系统至少需要 2^8 个系统时钟周期（作为参考）来产生第一个随机输出。然后，TRNG 需要至少 2^6 个系统时钟周期来生成每个后续的 64 位随机数。
- 启动时间和熵再生时间可以在 2^8 到 2^{24} 个采样时钟周期之间进行控制，熵再生时间可以在 2^6 到 2^{24} 个采样时钟周期之间进行控制，以使熵累积时间适应基本熵生成率。可以在随机数生成速度与每个随机数中的熵量之间进行权衡，以此来调整熵再生时间。
- TRNG 架构基于线性反馈移位寄存器 (LFSR) 以及与之关联的非线性熵哈希器。
- 应用程序可通过一个 64 位只读寄存器访问随机数。当读取该寄存器时，TRNG 会立即生成一个新值，并在就绪后将其移入输出寄存器。
- 如果在最大超时窗口内未读取就绪值，TRNG 将设置为空闲模式。
- TRNG 提供内置自检功能，用于检查采样到的连续比特数，以提供 FIPS 140 所需的统计稳健性。系统警报基于该自检的反馈生成。
- 内部省电模式旨在精细管理先前生成的熵。
- 中断通道允许传输 32 位数据块。

18.2 方框图

TRNG 内核使用双脉冲噪声发生器，当由提供给 TRNG 的系统时钟进行异步采样时，这些发生器会产生不可预测的抖动输出。来自脉冲噪声发生器的输出馈送一个复杂的非线性组合电路（混频器），该电路产生最终的 TRNG 输出（请参阅图 18-1）。

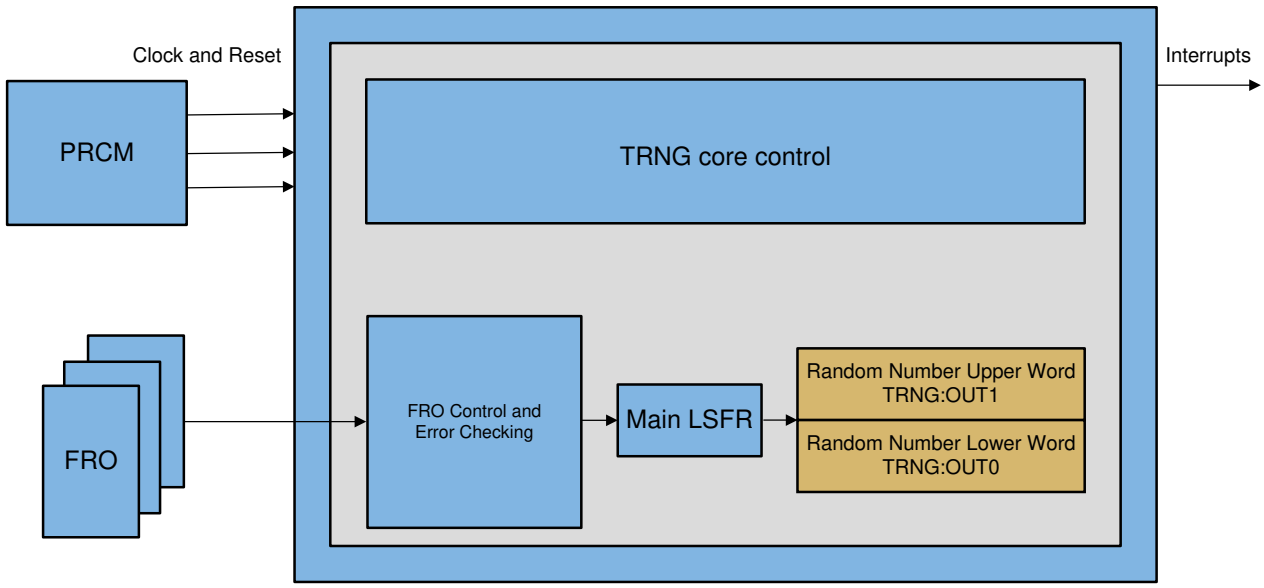


图 18-1. 随机数发生器方框图

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

TRNG 内核由两部分组成：

- 第一部分包含 FRO，其输出信号以固定间隔进行采样。FRO 彼此异步，并且与采样时钟异步，以使其行为真正具有不确定性。每个 FRO 都有一个错误检测电路，用于检查 FRO 是否出现重复模式。如果检测到重复模式，则该 FRO 被怀疑锁定到了采样时钟上，这会显著降低该 FRO 生成的熵量（这会作为 FRO 错误事件发出信号）。
- 第二部分是熵累积电路，该电路使用异或树来组合采样的 FRO 时钟输出，并使用一个 81 位 LFSR 来累积熵（TRNG:LFSR0、TRNG:LFSR1 和 TRNG:LFSR2 寄存器提供 81 位的主熵累积 LFSR）。

真正的熵源基于预定数量的自由运行振荡器 (FRO)。（主要）由脉冲噪声引起的时序抖动的累积为每个 FRO 的输出转换创造了不确定性区间。在不确定性区间内采样会产生少量的熵并累积在 LFSR 中。使用多个 FRO 并行生成熵，使得熵累积的速度比使用单个 FRO 快得多。

18.3 TRNG 软件复位

可以通过向 TRNG:SWRESET.RESET 寄存器写入 1 来执行模块的软件复位。当软件复位完成时，TRNG:SWRESET.RESET 寄存器会自动复位为 0。通过轮询 TRNG:SWRESET.RESET 寄存器直到其值为 0，软件可以确保复位已完成。在执行任何 TRNG 操作之前，必须完成软件复位。

18.4 中断请求

当数据准备好传输（或警报被触发）时，会生成中断请求 TRNG_IRQ。表 18-1 列出了可能导致模块中断的事件标志及其掩码。

表 18-1. 事件

事件标志	事件屏蔽	说明
TRNG:IRQSTAT.STAT	TRNG:IRQFLAGMASK.RDY 和 TRNG:IRQFLAGMASK. SHUTDOWN_OVF	未使用，但可以读取以获取两个可用中断的组合状态
TRNG:IRQFLAGSTAT.RDY	TRNG:IRQFLAGMASK.RDY	当为 1 时，表示 TRNG:OUT1 和 TRNG:OUT0 寄存器中有可用数据。使用 TRNG:IRQFLAGCLR.RDY 将其清除。
TRNG:IRQFLAGSTAT. SHUTDOWN_OVF	TRNG:IRQFLAGMASK. SHUTDOWN_OFV	当为 1 时，表示在第二次错误事件后关闭的 FRO 数量（TRNG:ALARMSTOP 寄存器中为 1 的位数量）已超过 TRNG:ALARMCNT.SHUTDOWN_THR 寄存器设置的阈值。使用 TRNG:IRQFLAGCLR.SHUTDOWN_OVF 寄存器将其清除。

主机可以使用 TRNG:ALARMCNT 寄存器以及 TRNG:ALARMMASK 和 TRNG:ALARMSTOP 寄存器来确定 FRO 或采样周期锁定是否存在问题。

在功能模式下，使用各个 FRO 的采样输出执行锁定检测。当连续检测到重复模式（最多四个样本长度）达到 TRNG:ALARMCNT:ALARM_THR 寄存器定义的样本数量时，将声明 FRO 警报事件。通过在 TRNG:ALARMMASK 寄存器中设置一个位来精确定位 FRO，从而记录该警报事件。如果已设置 TRNG:ALARMMASK 寄存器中的该位，则会设置 TRNG:ALARMSTOP 寄存器中的相应位，并关闭 FRO，以防止来自该 FRO 的进一步警报事件。如果尚未设置 TRNG:ALARMMASK 寄存器位，则 FRO 会自动重新启动，以尝试打破锁定。如果一个 FRO 在解调和重新启用后再次被锁定，软件必须使该 FRO 保持禁用状态。

TRNG:ALARMCNT.SHUTDOWN_CNT 寄存器位字段记录已关闭的 FRO 的数量（实际上，它是对 TRNG:ALARMSTOP 寄存器中为 1 的位数的计数）。TRNG:ALARMCNT.SHUTDOWN_THR 寄存器位字段允许设置一个可配置的阈值，以生成 SHUTDOWN_OVF 中断。当 TRNG:ALARMCNT.SHUTDOWN_CNT 寄存器超过 TRNG:ALARMCNT.SHUTDOWN_THR 寄存器时，TRNG:IRQFLAGSTAT.SHUTDOWN_OVF 寄存器位会设置为 1，这可用于生成中断。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

18.5 TRNG 操作说明

在首次生成随机数之前，必须写入 TRNG:CTL 和 TRNG:CFG0 寄存器以开始累积熵。熵是与随机值相关联的不确定性的度量。应用程序可通过 64 位只读寄存器 TRNG:OUT0 和 TRNG:OUT1 访问随机数。当读取该寄存器时，TRNG 会生成一个新值，该值在 TRNG:CFG0.MIN_REFILL_CYCLES 寄存器所设定的系统时钟周期后可用，然后被移入输出寄存器。软件可以使用两种策略来操作 TRNG：

- **监视模式**：软件定期（大约每隔几秒）检查 TRNG:ALARMMASK 寄存器。如果设置了其中某个位，则还必须检查 TRNG:ALARMSTOP 寄存器，以查看是否有 FRO 因多次警报事件而关闭。如果没有 FRO 关闭，则可以清除 TRNG:ALARMMASK 寄存器以消除虚假的警报事件。如果一个或多个 FRO 关闭，软件可以修改 TRNG:FRODETUNE 寄存器中这些 FRO 的延迟选择，以试图防止进一步的锁定。对于此类操作，TRNG:ALARMCNT.SHUTDOWN_THR 寄存器通常设置为一个较低的值（例如 2），然后可以使用 SHUTDOWN_OVF 中断来指示异常操作条件和/或 FRO 故障。
- **非监视模式**：软件将 TRNG:ALARMCNT.SHUTDOWN_THR 寄存器设置为在必须采取纠正措施之前允许关闭的 FRO 数量，然后使用 SHUTDOWN_OVF 中断来启动这些纠正措施（清除 TRNG:ALARMMASK 和 TRNG:ALARMSTOP 寄存器，切换 TRNG:FRODETUNE 寄存器中的位）。软件必须跟踪这些中断之间的时间间隔 — 如果它们发生得过于频繁，则表明存在异常操作条件和/或 FRO 故障。

18.5.1 TRNG 关断

TRNG 可以通过多种方式关闭，但并非所有方式都能保存熵。下文讨论不同的模式。

最好的方法是不读取最后生成的随机数。在 TRNG:CFG0 寄存器中定义的 MAX_REFILL 时间（最多 2^{24} 个周期）之后，TRNG 进入空闲模式，此时所有 FRO 都将关闭。读取生成的值时，TRNG 会再次启动并生成一个新的随机种子，该种子在 TRNG:CFG0.MIN_REFILL_CYCLES 寄存器设定的时间后准备就绪。当 TRNG 处于空闲模式时，可以关闭模块时钟。熵在两次随机数创建之间得以保留，因此无需对模块进行复位 (SW)。

关闭 TRNG 的另一种方法是仅停止模块时钟。通过停止模块时钟来关闭 TRNG，熵也会被保留（即不影响随机性），但 FRO 可能仍在运行。时钟可以随时启用，随机种子的生成会继续。无需对模块进行软复位。

如果时钟停止，则无法访问 TRNG，并且会（在互连结构内）生成总线故障。

如果一个不再需要 TRNG 的应用程序必须进入深度睡眠模式而无法等待，则应用程序可以在 TRNG:CTL.TRNG_EN 寄存器位中写入 0，并且可以关闭输入的系统时钟。以该方式关闭后，在生成新的随机数之前，应执行 TRNG 模块的软复位（请参阅节 18.7.1 中的寄存器说明），因为无法保证随机性。这种关闭方法的缺点是在下一个随机数准备就绪之前需要熵累积时间。

18.5.2 TRNG 警报

TRNG 警报发生时，最可能的原因是 FRO 时钟与采样时钟频率锁定。采样时钟与 TRNG 的系统时钟相同，当 FRO 振荡频率过于接近该时钟的倍数时，频率锁定可能导致在同一相位上对 FRO 时钟采样次数过多，从而检测到重复模式。当检测到这种重复模式时，会计数一次，当该计数超过 TRNG:ALARMCNT.ALARM_THR 寄存器设置的限制时，便会触发警报事件。将此值设置得较高可以限制警报数量，默认值为在启用警报事件之前累积 255 次警报指示。

当警报事件被触发时，关联的 FRO 会自动关闭，不再允许其参与熵累积。然后，用户必须决定如何处理此事件。有两种选择：

- 更改 FRO 振荡频率
- 保持 FRO 关闭

对于第一种选择，将 TRNG:FRODETUNE.FRO_MASK 寄存器中的一个位设置为 1，可使相关的 FRO 运行速度大约提高 5%。其中某个位的值只能在相应的 FRO 关闭时更改（通过在 TRNG:FROEN 寄存器的相应位中临时写入 0 — 在发生警报的情况下，该位已设置为 0）。当值更新后，必须再次启用相应的 FRO。

对于第二种选择，解调可能没有效果，或者 FRO 没有振荡。必须存储此状态，以便将 TRNG:FROEN 寄存器中的相应位保持为关闭状态，从而消除由该特定 FRO 引起的新警报触发。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

18.5.3 TRNG 熵

熵定义为以下因素的结果：

- 启用了多少个 FRO — 越多，熵达成越快
- 累积了多少个样本 — 运行时间越长，熵越高

启用的 FRO 越多，运行时间越长（即已存储在 LSFR 中的样本越多），熵就越高。

TRNG 模块在创建随机值时必须以最大频率运行。

随机值的创建时间由 TRNG:CTL.STARTUP_CYCLES 寄存器、TRNG:CFG0.MIN_REFILL_CYCLES 或 TRNG:CFG0.MAX_REFILL_CYCLES 寄存器以及 TRNG:CFG0.SMPL_DIV 寄存器中设置的值共同决定；所有这些寄存器的修改只能在 TRNG:CTL.TRNG_EN 寄存器为 0 时进行。

TRNG:CFG0.SMPL_DIV 寄存器定义了从 FRO 收集样本的频率，默认值 0 表示每个时钟周期采样一次，最大值 0xF 表示每 16 个周期采样一次。SAMPLE_DIV 的所有值都可在此器件上使用，并且必须设置得尽可能小。

为了在每个创建的种子中具有相同数量的熵，启动时间和最小重新填充时间必须相同。使用最小的启动时间和最小的重新填充时间时，每比特的熵非常低。当所有 FRO 都启用时，5 ms 的启动时间可以生成一个具有 64 位熵的字。

TRNG:CTL.STARTUP_CYCLES 寄存器和 TRNG:CFG0.MIN_REFILL_CYCLES 或 TRNG:CFG0.MAX_REFILL_CYCLES 寄存器中的低值只能用于为非安全用途（如同步字、CRC 初始化等）生成随机值。对于更安全的用途，必须定义最低 64 位熵及更高的熵要求。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

18.6 TRNG 底层编程指南

本节介绍了用于配置和使用该模块的底层硬件编程序列。

18.6.1 初始化

18.6.1.1 接口模块

本节介绍了在器件复位后首次使用 TRNG 时初始化接口模块的要求。表 18-2 列出了接口模块的初始化。

表 18-2. 周边模块的初始化

接口模块	注释
PRCM	必须使能 TRNG 模块接口时钟。请参见节 7.8.2 中的 PRCM 寄存器 PRCM:PDCTL0.PERIPH_ON 和 PRCM:SECDMACLKGR.TRNG_CLK_EN。
Arm® Cortex®-M4F	必须进行 NVIC 配置以使能来自 TRNG 的中断。仅当使用基于中断的通信时才需要。
互联	必须使能互联以与 TRNG 通信，这作为许多设置（如 CPU 运行、睡眠或深度睡眠模式、DMA、I^2S、RFCORE 和加密引擎的使用）的结果在 PRCM 中处理。

18.6.1.2 TRNG 主序列

此过程在上电复位 (POR) 后初始化 TRNG。表 18-3 列出了 TRNG 主初始化序列。

表 18-3. TRNG 初始化序列

步骤	寄存器或位字段
执行软件复位	TRNG:SWRESET.RESET
通过轮询等待软件复位完成	TRNG:SWRESET.RESET
选择两个样本之间的 FRO 时钟输入周期数	TRNG:CFG0.SMPL_DIV
选择用于在模块的 FRO 中收集足够熵并生成第一个随机值所需的样本数	TRNG:CTL.STARTUP_CYCLES
选择用于在模块的 FRO 中重新生成熵并生成后续随机值所需的最小样本数	TRNG:CFG0.MIN_REFILL_CYCLES
选择用于在模块的 FRO 中重新生成熵并生成后续随机值的最样本数 同时定义不活动后关闭 FRO 的超时周期	TRNG:CFG0.MAX_REFILL_CYCLES
配置所需的 FRO 以加快 5% 的速度运行	TRNG:FRODETUNE.FRO_MASK
使能所有 FRO	TRNG:FROEN.FRO_MASK
设定一个最大样本数，当检测到的重复模式达到该样本数时，将生成警报事件	TRNG:ALARMCNT.ALARM_THR
将关闭阈值设置为允许关闭的 FRO 数量(1)	TRNG:ALARMCNT.SHUTDOWN_THR
使能并启动	TRNG:CTL.TRNG_EN

(1) 仅在使用非监视模式时，才需要执行此步骤。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

18.6.1.3 TRNG 工作模式

本节介绍了 TRNG 模块不同工作模式的流程。

18.6.1.3.1 轮询模式

在轮询模式下，涵盖了监视模式和非监视模式。图 18-2 显示了 TRNG 轮询模式。

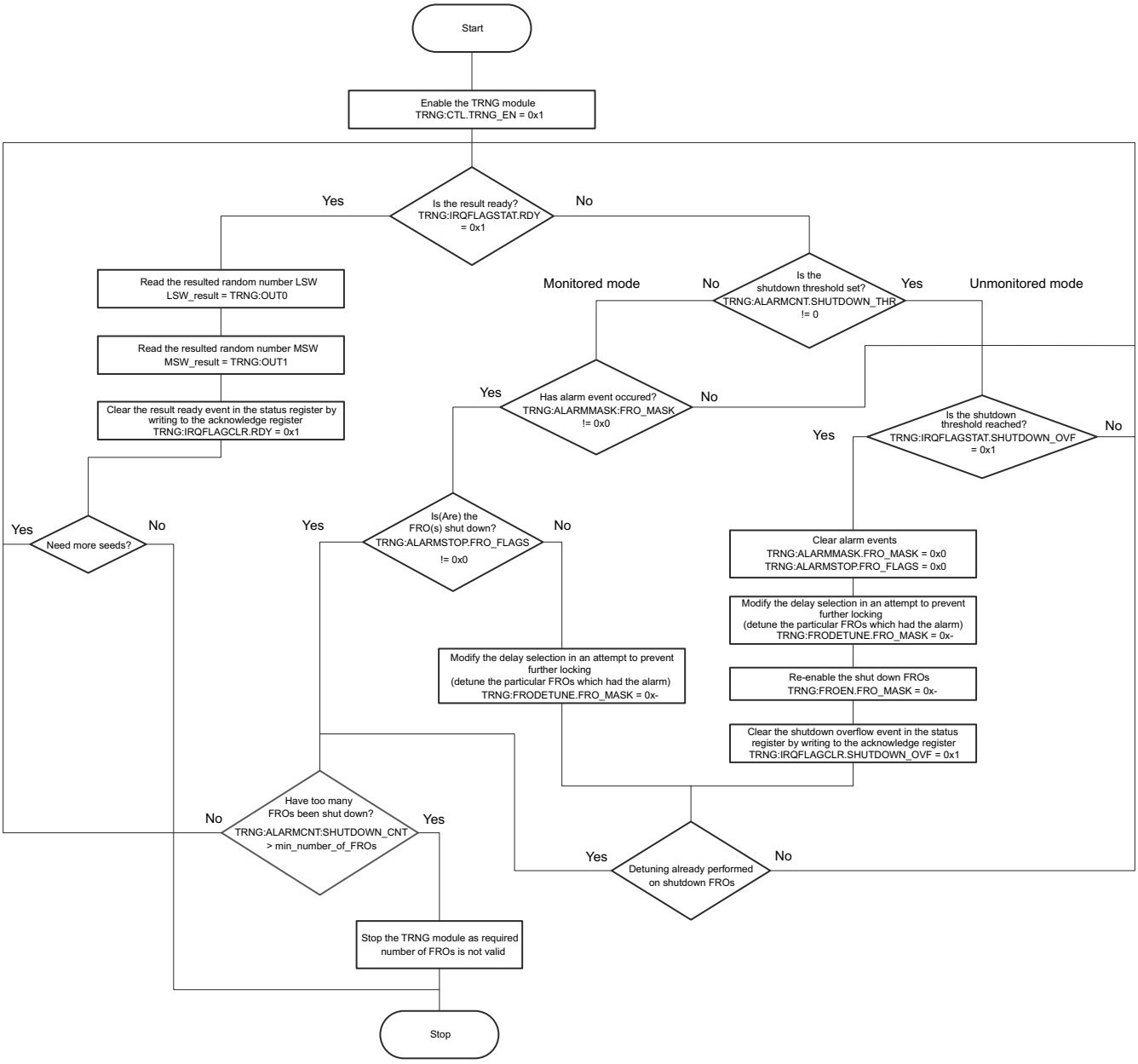


图 18-2. TRNG 轮询模式

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

18.6.1.3.2 中断模式

本节介绍模块的事件服务。仅涵盖非监视模式。表 18-4 列出了 TRNG 中断模式的步骤，图 18-3 显示了中断服务例程流程。

表 18-4. TRNG 中断模式

步骤	寄存器或位字段	值
当数据在输出寄存器中就绪（可用）时，使能中断生成。	TRNG:IRQFLAGMASK.RDY	0x1
当最大可能的 FRO 关闭数达到所关断选阈值时，使能关断溢出中断生成	TRNG:IRQFLAGMASK.SHUTDOWN_OVF	0x1
启用 TRNG	TRNG:CTL.TRNG_EN	0x1

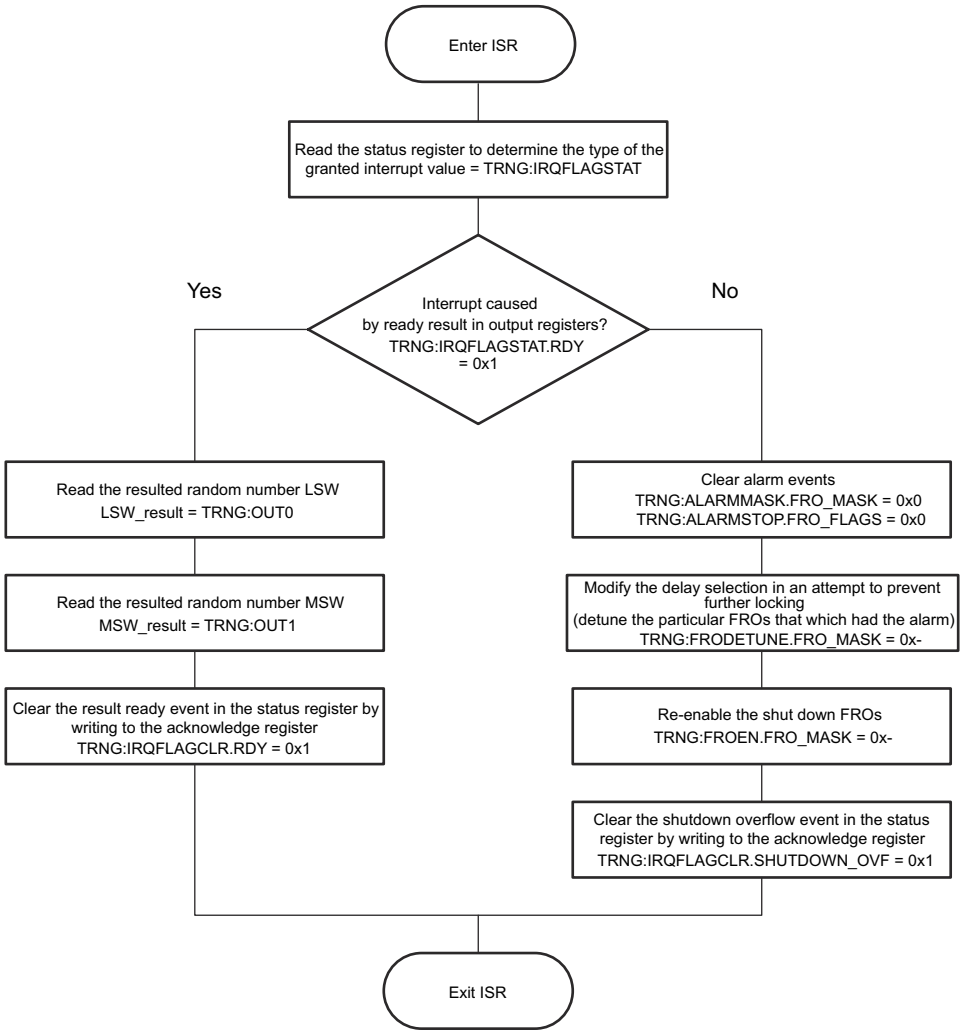


图 18-3. 中断服务例程

18.7 TRNG 寄存器

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

18.7.1 TRNG 寄存器

表 18-5 列出了 TRNG 寄存器的存储器映射寄存器。表 18-5 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

表 18-5. TRNG 寄存器

偏移	首字母缩写词	寄存器名称	部分
0h	OUT0	随机数低位字读取值	OUT0 寄存器 (偏移 = 0h) [复位 = 00000000h]
4h	OUT1	随机数高位字读取值	OUT1 寄存器 (偏移 = 4h) [复位 = 00000000h]
8h	IRQFLAGSTAT	中断状态	IRQFLAGSTAT 寄存器 (偏移 = 8h) [复位 = 00000000h]
Ch	IRQFLAGMASK	中断屏蔽	IRQFLAGMASK 寄存器 (偏移 = Ch) [复位 = 00000000h]
10h	IRQFLAGCLR	中断标志清除	IRQFLAGCLR 寄存器 (偏移 = 10h) [复位 = 00000000h]
14h	CTL	控制	CTL 寄存器 (偏移 = 14h) [复位 = 00000000h]
18h	CFG0	配置 0	CFG0 寄存器 (偏移 = 18h) [复位 = 00000000h]
1Ch	ALARMCNT	警报控制	ALARMCNT 寄存器 (偏移 = 1Ch) [复位 = 000000FFh]
20h	FROEN	FRO 使能	FROEN 寄存器 (偏移 = 20h) [复位 = 00FFFFFFh]
24h	FRODETUNE	FRO 失谐位	FRODETUNE 寄存器 (偏移 =24h) [复位 = 00000000h]
28h	ALARMMASK	警报事件	ALARMMASK 寄存器 (偏移 = 28h) [复位 = 00000000h]
2Ch	ALARMSTOP	警报关断	ALARMSTOP 寄存器 (偏移 = 2Ch) [复位 = 00000000h]
30h	LFSR0	LFSR 读取值	LFSR0 寄存器 (偏移 = 30h) [复位 = 00000000h]
34h	LFSR1	LFSR 读取值	LFSR1 寄存器 (偏移 = 34h) [复位 = 00000000h]
38h	LFSR2	LFSR 读取值	LFSR2 寄存器 (偏移 = 38h) [复位 = 00000000h]
78h	HWOPT	TRNG 引擎选项信息	HWOPT 寄存器 (偏移 = 78h) [复位 = 00000600h]
7Ch	HWVER0	HW 版本 0	HWVER0 寄存器 (偏移 = 7Ch) [复位 = 0200B44Bh]
1FD8h	IRQSTATMASK	屏蔽后的中断状态	IRQSTATMASK 寄存器 (偏移 = 1FD8h) [复位 = 00000000h]
1FE0h	HWVER1	HW 版本 1	HWVER1 寄存器 (偏移 = 1FE0h) [复位 = 00000020h]
1FECh	IRQSET	中断设置	IRQSET 寄存器 (偏移 = 1FECh) [复位 = 00000000h]
1FF0h	SWRESET	SW 复位控制	SWRESET 寄存器 (偏移 = 1FF0h) [复位 = 00000000h]
1FF8h	IRQSTAT	中断状态	IRQSTAT 寄存器 (偏移 = 1FF8h) [复位 = 00000000h]

复杂的位访问类型经过编码可适应小型表单元。表 18-6 展示了适用于此部分中访问类型的代码。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 18-6. TRNG 访问类型代码

访问类型	代码	说明
读取类型		
R	R	读取
写入类型		
W	W	写入
复位或默认值		
-n		复位后的值或默认值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

18.7.1.1 OUT0 寄存器（偏移 = 0h）[复位 = 00000000h]

图 18-4 展示了 OUT0，表 18-7 对其进行了介绍。

返回到汇总表。

随机数低位字读取值

图 18-4. OUT0 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE_31_0																															
R-0h																															

表 18-7. OUT0 寄存器字段说明

位	字段	类型	复位	说明
31-0	VALUE_31_0	R	0h	64 位随机值的 LSW。当 IRQFLAGSTAT.RDY = 1 时，新值就绪。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

18.7.1.2 OUT1 寄存器（偏移 = 4h）[复位 = 00000000h]

图 18-5 展示了 OUT1，表 18-8 对其进行了介绍。

返回到汇总表。

随机数高位字读取值

图 18-5. OUT1 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE_63_32																															
R-0h																															

表 18-8. OUT1 寄存器字段说明

位	字段	类型	复位	说明
31-0	VALUE_63_32	R	0h	64 位随机值的 MSW。当 IRQFLAGSTAT.RDY = 1 时，新值就绪。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

18.7.1.3 IRQFLAGSTAT 寄存器（偏移 = 8h）[复位 = 00000000h]

图 18-6 展示了 IRQFLAGSTAT，表 18-9 中对此进行了介绍。

返回到汇总表。

中断状态

图 18-6. IRQFLAGSTAT 寄存器

31	30	29	28	27	26	25	24
NEED_CLOCK	RESERVED						
R-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						SHUTDOWN_OVF	RDY
R-0h						R-0h	R-0h

表 18-9. IRQFLAGSTAT 寄存器字段说明

位	字段	类型	复位	说明
31	NEED_CLOCK	R	0h	1：表示 TRNG 正在忙于生成熵或处于其测试模式之一 — 此时不能关闭时钟，且必须保持电源电压稳定。 0：TRNG 处于空闲状态，可以关断
30-2	RESERVED	R	0h	保留
1	SHUTDOWN_OVF	R	0h	1：已关断的 FRO 数量（即 ALARMSTOP 寄存器中“1”位的数量）已超过 ALARMCNT.SHUTDOWN_THR 设置的阈值 将“1”写入 IRQFLAGCLR.SHUTDOWN_OVF 会再次将该位清除为“0”。
0	RDY	R	0h	1：OUT0 和 OUT1 中有数据可用。 通过向 IRQFLAGCLR.RDY 写入“1”来确认此状态会将该位清除为“0”。 如果 TRNG 的内部寄存器中已经有新数字，该数字将直接在结果寄存器中计时。在这种情况下，该状态位将在一个时钟周期后再次置为有效。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

18.7.1.4 IRQFLAGMASK 寄存器（偏移 = Ch）[复位 = 00000000h]

图 18-7 展示了 IRQFLAGMASK，表 18-10 中对此进行了介绍。

返回到汇总表。

中断屏蔽

图 18-7. IRQFLAGMASK 寄存器

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

表 18-10. IRQFLAGMASK 寄存器字段说明

位	字段	类型	复位	说明
31-2	RESERVED	R	0h	保留
1	SHUTDOWN_OVF	R/W	0h	1：允许 IRQFLAGSTAT.SHUTDOWN_OVF 激活来自该模块的中断。
0	RDY	R/W	0h	1：允许 IRQFLAGSTAT.RDY 激活来自该模块的中断。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

18.7.1.5 IRQFLAGCLR 寄存器（偏移 = 10h）[复位 = 00000000h]

图 18-8 展示了 IRQFLAGCLR，表 18-11 中对此进行了介绍。

返回到汇总表。

中断标志清除

图 18-8. IRQFLAGCLR 寄存器

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

表 18-11. IRQFLAGCLR 寄存器字段说明

位	字段	类型	复位	说明
31-2	RESERVED	R	0h	保留
1	SHUTDOWN_OVF	W	0h	1：清除 IRQFLAGSTAT.SHUTDOWN_OVF。
0	RDY	W	0h	1：清除 IRQFLAGSTAT.RDY。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

18.7.1.6 CTL 寄存器（偏移 = 14h）[复位 = 00000000h]

图 18-9 展示了 CTL，表 18-12 中对此进行了介绍。

返回到汇总表。

控制

图 18-9. CTL 寄存器

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

表 18-12. CTL 寄存器字段说明

位	字段	类型	复位	说明
31-16	STARTUP_CYCLES	R/W	0h	此字段确定启动期间从 FRO 收集熵所需的样本数（介于 2^8 和 2^{24} 之间）。如果此字段的写入值为零，则样本数为 2^{24}，否则样本数等于写入值乘以 2^8。 0x0000：2^{24} 个样本 0x0001：$1*2^8$ 个样本 0x0002：$2*2^8$ 个样本 0x0003：$3*2^8$ 个样本 ... 0x8000：$32768*2^8$ 个样本 0xC000：$49152*2^8$ 个样本 ... 0xFFFF：$65535*2^8$ 个样本 仅当 TRNG_EN 为 0 时，才能修改该字段。如果为 1，则会忽略更新。
15-11	RESERVED	R	0h	保留
10	TRNG_EN	R/W	0h	0：立即强制所有 TRNG 逻辑回到空闲状态。 1：启动 TRNG，根据 STARTUP_CYCLES 确定的样本数从 FRO 收集熵。
9-3	RESERVED	R	0h	保留
2	NO_LFSR_FB	R/W	0h	1：从主 LFSR 删除 XNOR 反馈，将其转换为用于 FRO 异或输出的正常移位寄存器（在 LSB 侧移入数据）。“1”还会强制 LFSR 连续采样。 仅当 TEST_MODE 也设置为“1”时，该位才能设置为“1”，不应该用于测试以外的用途
1	TEST_MODE	R/W	0h	1：使能对 TESTCNT 和 LFSR0/LFSR1/LFSR2 寄存器的访问（后几个寄存器在使能访问之前自动清除）并将 IRQFLAGSTAT.NEED_CLOCK 保持为“1”。 除非在创建新的随机值之前需要更改 LFSR 种子，否则不应使用该位。所有其他测试均在寄存器控制外部进行。
0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

18.7.1.7 CFG0 寄存器（偏移 = 18h）[复位 = 00000000h]

图 18-10 展示了 CFG0，表 18-13 对其进行了介绍。

返回到汇总表。

配置 0

图 18-10. CFG0 寄存器

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

表 18-13. CFG0 寄存器字段说明

位	字段	类型	复位	说明
31-16	MAX_REFILL_CYCLES	R/W	0h	该字段确定读取 64 位随机数后从 FRO 重新生成熵所需的最大样本数（介于 2^8 和 2^{24} 之间）。如果此字段的写入值为零，则样本数为 2^{24}，否则样本数等于写入值乘以 2^8。 0x0000：2^{24} 个样本 0x0001：$1*2^8$ 个样本 0x0002：$2*2^8$ 个样本 0x0003：$3*2^8$ 个样本 ... 0x8000：$32768*2^8$ 个样本 0xC000：$49152*2^8$ 个样本 ... 0xFFFF：$65535*2^8$ 个样本 仅当 CTL.TRNG_EN 为 0 时，才能修改该字段。
15-12	RESERVED	R	0h	保留
11-8	SMPL_DIV	R/W	0h	该字段直接控制从 FRO 采集的样本之间的时钟周期数。默认值 0 表示每个时钟周期进行采样， 最大值 0xF 每 16 个时钟周期进行一次采样。 该字段必须设置为适当的值，使最慢 FRO（即使在最坏的情况下）的周期时间小于采样周期的两倍。 仅当 CTL.TRNG_EN 为“0”时，才能修改此字段。
7-0	MIN_REFILL_CYCLES	R/W	0h	该字段确定读取 64 位随机数后从 FRO 重新生成熵所需的最小样本数（介于 2^6 和 2^{14} 之间）。如果此字段的值为零，则样本数固定为 MAX_REFILL_CYCLES 字段确定的值，否则最小样本数等于写入的值乘以 64（最多可达 2^{14}）。为了确保所有生成的随机数中具有相同的熵，应使用值 0。然后，MAX_REFILL_CYCLES 控制最小重新填充间隔。此处定义的样本数不能高于“max_refill_cycles”字段定义的数字（即该字段优先）。如果最小重新填充次数 > 最大重新填充次数，则不会创建随机值。 仅当 CTL.TRNG_EN = 0 时，才能修改该字段。 0x00：最小样本数 = MAX_REFILL_CYCLES（所有数字都具有相同的熵） 0x01：$1*2^6$ 个样本 0x02：$2*2^6$ 个样本 ... 0xFF：$255*2^6$ 个样本

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

18.7.1.8 ALARMCNT 寄存器（偏移 = 1Ch）[复位 = 000000FFh]

图 18-11 展示了 ALARMCNT，表 18-14 中对此进行了介绍。

返回到汇总表。

警报控制

图 18-11. ALARMCNT 寄存器

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

表 18-14. ALARMCNT 寄存器字段说明

位	字段	类型	复位	说明
31-30	RESERVED	R	0h	保留
29-24	SHUTDOWN_CNT	R/W	0h	只读，指示 ALARMSTOP 寄存器中“1”位的数量。 最大值等于 FRO 的数量。
23-21	RESERVED	R	0h	保留
20-16	SHUTDOWN_THR	R/W	0h	用于生成 IRQFLAGSTAT.SHUTDOWN_OVF 中断的阈值设置。当 SHUTDOWN_CNT 值超过该位字段时，会触发中断。
15-8	RESERVED	R	0h	保留
7-0	ALARM_THR	R/W	FFh	每个 FRO 上的重复模式检测器的警报检测阈值。当针对此字段的值定义的样本数连续检测到重复模式（最多四个样本长度）时，将声明 FRO“警报事件”。复位值 0xFF 应将“警报事件”的数量保持在可管理的水平。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

18.7.1.9 FROEN 寄存器（偏移 = 20h）[复位 = 00FFFFFFh]

图 18-12 展示了 FROEN，表 18-15 中对此进行了介绍。

返回到汇总表。

FRO 使能

图 18-12. FROEN 寄存器

31	30	29	28	27	26	25	24	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								FRO_MASK																							
R-0h								R/W-00FFFFFFh																							

表 18-15. FROEN 寄存器字段说明

位	字段	类型	复位	说明
31-24	RESERVED	R	0h	保留
23-0	FRO_MASK	R/W	00FFFFFFh	单个 FRO 的使能位。位 [n] 中的“1”使能 FRO“n”。默认状态为全“1”，用于在上电后使能所有 FRO。请注意，在 CTL.TRNG_EN 位设置为“1”之前，它们实际上并不会启动。 此处的位会自动强制为“0”（不能写入“1”），而 ALARMSTOP.FRO_FLAGS 中相应位的值为“1”。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

18.7.1.10 FRODETUNE 寄存器（偏移 =24h）[复位 = 00000000h]

图 18-13 展示了 FRODETUNE，表 18-16 中对此进行了介绍。

返回到汇总表。

FRO 失谐位

图 18-13. FRODETUNE 寄存器

31	30	29	28	27	26	25	24	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								FRO_MASK																							
R-0h								R/W-0h																							

表 18-16. FRODETUNE 寄存器字段说明

位	字段	类型	复位	说明
31-24	RESERVED	R	0h	保留
23-0	FRO_MASK	R/W	0h	单个 FRO 失谐位。位 [n] 中的“1”使 FRO“n”的运行速度大约加快 5%。仅当相应的 FRO 关闭时，才能更改这些位之一的值（通过临时在 FROEN.FRO_MASK 寄存器的 相应位中写入“0”。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

18.7.1.11 ALARMMASK 寄存器（偏移 = 28h）[复位 = 00000000h]

图 18-14 展示了 ALARMMASK，表 18-17 中对此进行了介绍。

返回到汇总表。

警报事件

图 18-14. ALARMMASK 寄存器

31	30	29	28	27	26	25	24	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								FRO_MASK																							
R-0h								R/W-0h																							

表 18-17. ALARMMASK 寄存器字段说明

位	字段	类型	复位	说明
31-24	RESERVED	R	0h	保留
23-0	FRO_MASK	R/W	0h	记录单个 FRO 的“警报事件”位。位 [n] 中的“1”指示 FRO“n”发生了“警报事件”。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

18.7.1.12 ALARMSTOP 寄存器（偏移 = 2Ch）[复位 = 00000000h]

图 18-15 展示了 ALARMSTOP，表 18-18 中对此进行了介绍。

返回到汇总表。

警报关断

图 18-15. ALARMSTOP 寄存器

31	30	29	28	27	26	25	24	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								FRO_FLAGS																							
R-0h								R/W-0h																							

表 18-18. ALARMSTOP 寄存器字段说明

位	字段	类型	复位	说明
31-24	RESERVED	R	0h	保留
23-0	FRO_FLAGS	R/W	0h	记录单个 FRO 的“警报事件”位。位 [n] 中的“1”指示 FRO“n”快速连续发生了多个“警报事件”，已关闭。该字段中的“1”会强制 FROEN.FRO_MASK 中的相应位为“0”。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

18.7.1.13 LFSR0 寄存器（偏移 = 30h）[复位 = 00000000h]

图 18-16 展示了 LFSR0，表 18-19 对其进行了介绍。

返回到汇总表。

LFSR 读取值

图 18-16. LFSR0 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LFSR_31_0																															
R/W-0h																															

表 18-19. LFSR0 寄存器字段说明

位	字段	类型	复位	说明
31-0	LFSR_31_0	R/W	0h	主熵累积 LFSR 的位 [31:0]。仅当 CTL.TEST_MODE = 1 时才能访问寄存器。 在使能访问之前，寄存器内容将清零。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

18.7.1.14 LFSR1 寄存器（偏移 = 34h）[复位 = 00000000h]

图 18-17 展示了 LFSR1，表 18-20 对其进行了介绍。

返回到汇总表。

LFSR 读取值

图 18-17. LFSR1 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LFSR_63_32																															
R/W-0h																															

表 18-20. LFSR1 寄存器字段说明

位	字段	类型	复位	说明
31-0	LFSR_63_32	R/W	0h	主熵累积 LFSR 的位 [63:32]。仅当 CTL.TEST_MODE = 1 时才能访问寄存器。 在使能访问之前，寄存器内容将清零。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

18.7.1.15 LFSR2 寄存器（偏移 = 38h）[复位 = 00000000h]

图 18-18 展示了 LFSR2，表 18-21 对其进行了介绍。

返回到汇总表。

LFSR 读取值

图 18-18. LFSR2 寄存器

31	30	29	28	27	26	25	24	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																LFSR_80_64															
R-0h																R/W-0h															

表 18-21. LFSR2 寄存器字段说明

位	字段	类型	复位	说明
31-17	RESERVED	R	0h	保留
16-0	LFSR_80_64	R/W	0h	主熵累积 LFSR 的位 [80:64]。仅当 CTL.TEST_MODE = 1 时才能访问寄存器。 在使能访问之前，寄存器内容将清零。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

18.7.1.16 HWOPT 寄存器（偏移 = 78h）[复位 = 00000600h]

图 18-19 展示了 HWOPT，表 18-22 中对此进行了介绍。

返回到汇总表。

TRNG 引擎选项信息

图 18-19. HWOPT 寄存器

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				NR_OF_FROS						RESERVED					
R-0h				R-18h						R-0h					

表 18-22. HWOPT 寄存器字段说明

位	字段	类型	复位	说明
31-12	RESERVED	R	0h	保留
11-6	NR_OF_FROS	R	18h	此 TRNG 中实现的 FRO 数量，值 24（十进制）。
5-0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

18.7.1.17 HWVER0 寄存器（偏移 = 7Ch）[复位 = 0200B44Bh]

图 18-20 展示了 HWVER0，表 18-23 对其进行了介绍。

返回到汇总表。

硬件版本 0
EIP 编号和核心版本

图 18-20. HWVER0 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED				HW_MAJOR_VER				HW_MINOR_VER				HW_PATCH_LVL			
R-0h				R-2h				R-0h				R-0h			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EIP_NUM_COMPL								EIP_NUM							
R-B4h								R-4Bh							

表 18-23. HWVER0 寄存器字段说明

位	字段	类型	复位	说明
31-28	RESERVED	R	0h	保留
27-24	HW_MAJOR_VER	R	2h	主要硬件版本号的 4 位二进制编码。
23-20	HW_MINOR_VER	R	0h	次要硬件版本号的 4 位二进制编码。
19-16	HW_PATCH_LVL	R	0h	硬件补丁级别的 4 位二进制编码，初始版本值为零。
15-8	EIP_NUM_COMPL	R	B4h	位 [7:0] 的逐位逻辑补码。此 TRNG 的值为 0xB4。
7-0	EIP_NUM	R	4Bh	模块编号的 8 位二进制编码。此 TRNG 的值为 0x4B。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

18.7.1.18 IRQSTATMASK 寄存器（偏移 = 1FD8h）[复位 = 00000000h]

图 18-21 展示了 IRQSTATMASK，表 18-24 中对此进行了介绍。

返回到汇总表。

屏蔽后的中断状态

图 18-21. IRQSTATMASK 寄存器

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

表 18-24. IRQSTATMASK 寄存器字段说明

位	字段	类型	复位	说明
31-2	RESERVED	R	0h	保留
1	SHUTDOWN_OVF	R	0h	关断溢出（IRQFLAGSTAT.SHUTDOWN_OVF 与 IRQFLAGMASK.SHUTDOWN_OVF 进行与运算的结果）
0	RDY	R	0h	可用的新随机值（IRQFLAGSTAT.RDY 与 IRQFLAGMASK.RDY 进行与运算的结果）

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

18.7.1.19 HWVER1 寄存器（偏移 = 1FE0h）[复位 = 00000020h]

图 18-22 展示了 HWVER1，表 18-25 对其进行了介绍。

返回到汇总表。

硬件版本 1
TRNG 版本号

图 18-22. HWVER1 寄存器

31	30	29	28	27	26	25	24	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																								版本							
R-0h																								R-20h							

表 18-25. HWVER1 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-0	版本	R	20h	此模块的版本号为 2.0 版。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

18.7.1.20 IRQSET 寄存器（偏移 = 1FECh）[复位 = 00000000h]

图 18-23 展示了 IRQSET，表 18-26 中对此进行了介绍。

返回到汇总表。

中断设置

图 18-23. IRQSET 寄存器

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

表 18-26. IRQSET 寄存器字段说明

位	字段	类型	复位	说明
31-0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

18.7.1.21 SWRESET 寄存器（偏移 = 1FF0h）[复位 = 00000000h]

图 18-24 展示了 SWRESET，表 18-27 中对此进行了介绍。

返回到汇总表。

SW 复位控制

图 18-24. SWRESET 寄存器

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

表 18-27. SWRESET 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	复位	R/W	0h	写入“1”执行软复位，复位信号将保持低电平 4-5 个时钟周期。轮询该位直到为 0，以完成复位。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

18.7.1.22 IRQSTAT 寄存器（偏移 = 1FF8h）[复位 = 00000000h]

图 18-25 展示了 IRQSTAT，表 18-28 中对此进行了介绍。

返回到汇总表。

中断状态

图 18-25. IRQSTAT 寄存器

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

表 18-28. IRQSTAT 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	STAT	R	0h	TRNG 中断状态。IRQFLAGSTAT.SHUTDOWN_OVF 与 IRQFLAGSTAT.RDY 的或运算版本

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

章节 19

AUX 域传感器控制器和外设

本章介绍了 CC13x2 和 CC26x2 器件平台上 AUX 子系统的功能。

19.1 简介 1445
19.2 电源和时钟管理 1447
19.3 传感器控制器 1450
19.4 数字外设模块 1467
19.5 模拟外设模块 1493
19.6 事件路由和使用 1509
19.7 传感器控制器别名寄存器空间 1515
19.8 AUX 域传感器控制器和外设寄存器 1520

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.1 简介

辅助 (AUX) 域是一个位于 AON 域内部的超低功耗独立协处理器子系统。AUX 域包含一个高能效、可编程的 16 位传感器控制器引擎 (SCE)，该引擎可以访问以下资源：

- AUX SRAM：传感器控制器的指令和数据存储器
  - 4kB = 2048 个传感器控制器 16 位指令/数据字
- 模拟外设：
  - ADC：12 位模数转换器，最大采样率为 200kHz
  - COMPA：连续时间比较器
  - COMPB：低功耗时钟比较器
  - ISRC：0 至 20µA 电流源
  - 参考 DAC：数模转换器
- 数字外设：
  - TDC：高精度时间数字转换器
  - AUX Timer0 和 Timer1：简单的 16 位同步计时器
  - AUX Timer2：具有 4 个捕获/比较通道的异步通用计时器
  - 硬件信标：用于在传感器控制器和系统 CPU 之间共享外设
  - AUX I/O 控制器
  - AUX SPI 主器件：用于高能效的传感器轮询
  - 数学加速器：包括 40 位累加器 (有符号/无符号加法、乘法)
- I/O 引脚：
  - 多达 32 个仅用作数字功能的通用 I/O 引脚
  - 多达 8 个支持模拟功能的通用 I/O 引脚
- 系统振荡器控制
- 多种时钟和电源选项

传感器控制器无法访问 MCU 域的外设、RAM、闪存或寄存器。这种分离允许 MCU 域独立于传感器控制器进入和退出待机模式。然而，系统 CPU 和 µDMA 始终可以访问所有 AUX 域的外设，并且可以使用传感器控制器未使用的外设。

来自 MCU 域的访问速度与所选的 AUX 时钟无关，并且传感器控制器的运行与 MCU 域无关。传感器控制器对某些 AON 域外设 (如*实时时钟*、*电池监测器和温度传感器*) 具有有限的访问权限。传感器控制器还可以请求对 VDDR 进行充电并监测此类事件。

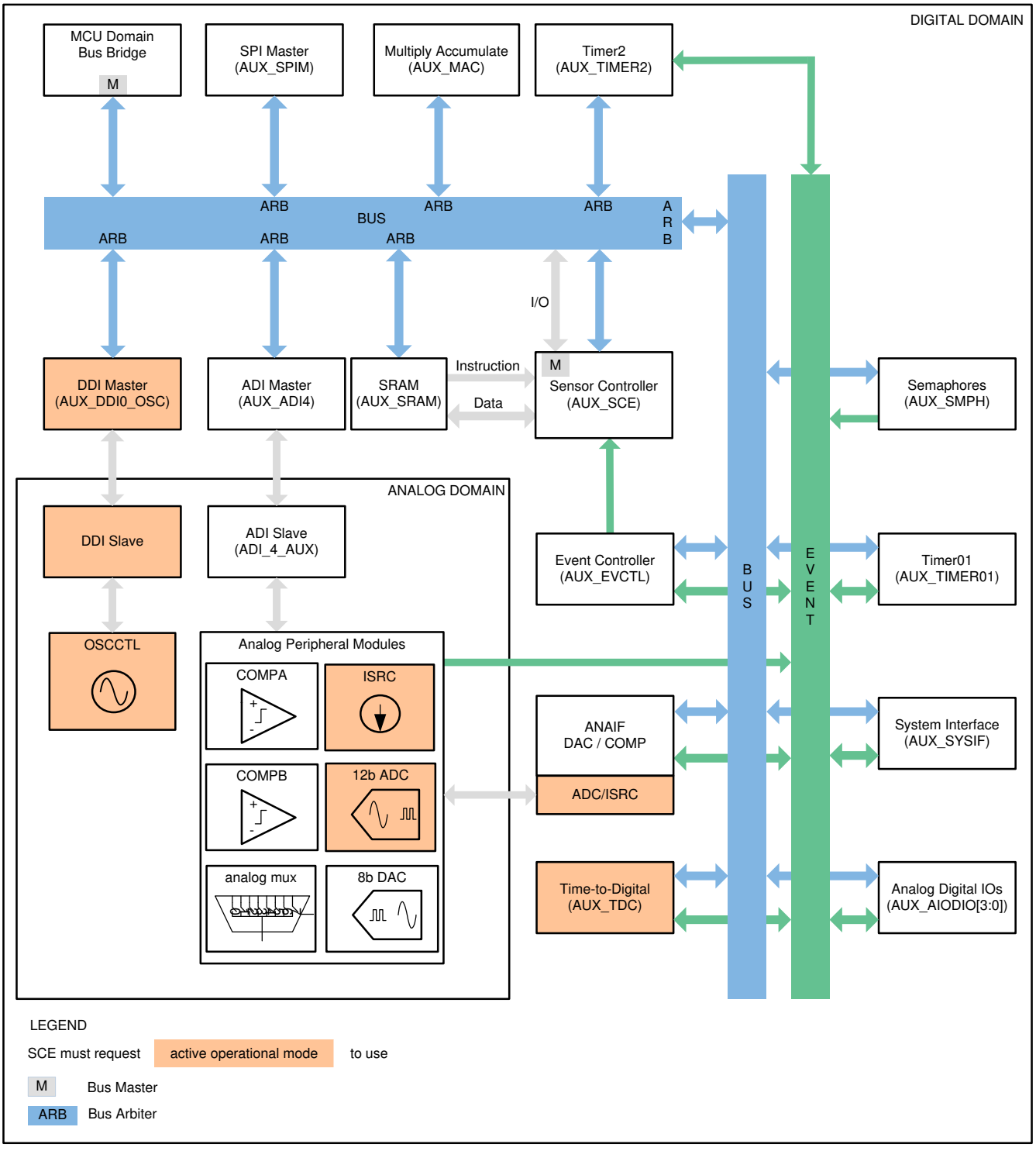
系统 CPU 可以直接使用 AUX 域中的外设。然而，通过利用传感器控制器，可以独立于系统 CPU 运行简单的后台任务，从而实现最小的系统功耗。传感器控制器任务可以设置为周期性运行，或基于异步事件 (如 GPIO 事件) 运行。以下是一些适合传感器控制器执行的任务示例：

- 使用 ADC 或比较器进行模拟传感器轮询
- 使用 SPI、位拆裂 I^2C 或其他标准进行数字传感器轮询
- 使用集成电流源、比较器和时间数字转换器进行电容式感应
- I/O 波形生成
  - 使用传感器控制器进行通用 I/O 控制
  - 使用 AUX Timer2 生成 PWM
- 使用 TDC 或 AUX Timer2 进行信号周期和脉冲宽度测量
- 根据任务需要唤醒系统 CPU：数据已收集、选择性地进行了滤波，并准备好供系统 CPU 处理

19.1.1 AUX 方框图

图 19-1 显示 AUX 域中模块的连接。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated



Copyright © 2017, Texas Instruments Incorporated

图 19-1. AUX 域方框图

图 19-1 显示 AUX 域中有两个总线主控器，分别传感器控制器和 MCU 域总线桥接器。系统 CPU 和 µDMA 通过总线桥接器访问 AUX 域。当传感器控制器与 MCU 域总线桥接器访问连接到同一总线仲裁器的模块时，两者之间会发生总线仲裁。总线仲裁器始终优先于传感器控制器。因此，系统 CPU 可以访问振荡器接口，同时传感器控制器可访问 SPI 主控器，而不会相互干扰。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.2 电源和时钟管理

AUX 电源和时钟管理在很大程度上独立于 MCU 域。MCU 域处于活动状态时可能会：

- 提高 AUX 总线时钟速度以尽可能缩短访问时间。
- 在由 AON_PMCTL:AUXSCECLK 选择的、为传感器控制器和选定外设提供时钟的时钟上产生小幅时钟抖动。

这些主题在节 19.2.1 至节 19.2.4 中进行了说明。当 MCU 域接收到 SCLK_HF 时，它处于活动状态，这发生在 TI-RTOS 电源模式（主动和空闲）下。

19.2.1 工作模式

AUX 域可以请求在活动、低功耗或断电模式下运行（请参阅图 19-2）。

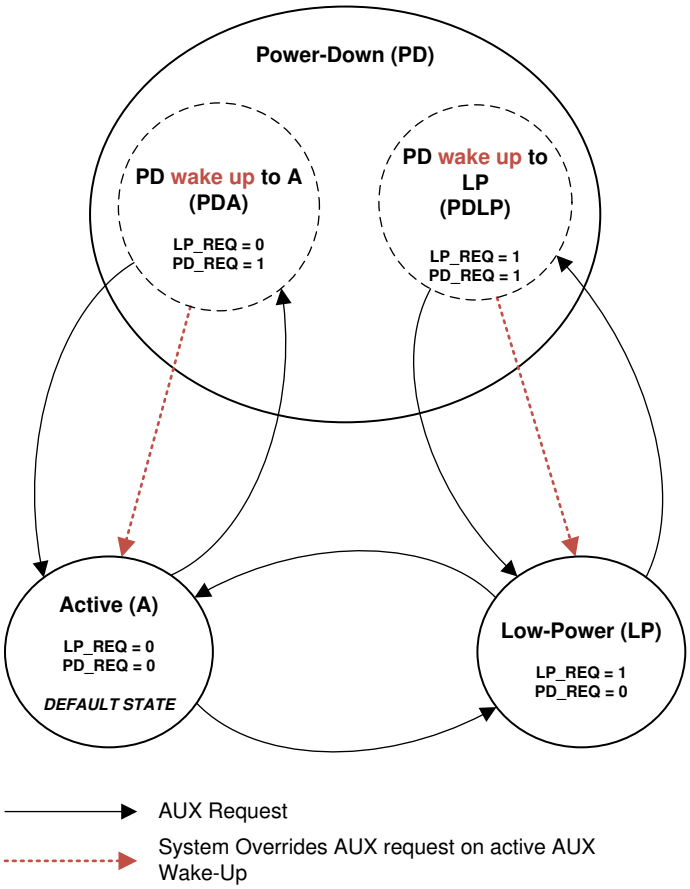


图 19-2. AUX 状态图

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

每种工作模式的特征由以下方面定义：

- 系统电源请求
- 传感器控制器操作时钟速率（SCE 时钟速率）
- 传感器控制器模块可用性
- 系统对活动 AUX 唤醒事件的响应

系统不会更改 AUX 工作模式请求。如图 19-2 所示，只要 AUX 唤醒事件处于活动状态，系统就会覆盖该请求。AUX 唤醒仅适用于传感器控制器（有关说明，请参阅节 19.1）。

表 19-1 总结了每种工作模式的属性。

表 19-1. AUX 工作模式

运行模式		有效		低功耗	断电	
AUX SCE 时钟源		AON_PMCTL:AUXSCECLK.SRC		SCLK_MF	AON_PMCTL:AUXSCECLK.PDSRC	
SCE 时钟速率		24MHz	2MHz	2MHz	32.768kHz	NO_CLOCK
AUX 总线时钟速率	MCU 不活动	24MHz	24MHz	2MHz	32.768kHz	NO_CLOCK
	MCU 有效	24MHz	24MHz	24MHz	24MHz	24MHz
唤醒系统响应		无		无	活动或低功耗	
电源请求		DCDC 或 GLDO		µLDO	µLDO	
SCE 模块可用性		全部		请勿使用 ADC、ISRC、TDC 或 OSCCTL。		

按照 AUX_SYSIF:OPMODEREQ 中所述的步骤来请求工作模式（请参阅节 19.8.9）。有关 AUX 唤醒，请参阅节 19.1。活动工作模式请求会阻止系统进入待机状态。

备注

活动工作模式请求会阻止系统进入待机或关机状态，因为电源控制器会请求一个高频时钟源。

19.2.1.1 双速率 AUX 时钟

AUX 模块在内部以传感器控制器引擎 (SCE) 时钟速率或总线时钟速率运行。传感器控制器以 SCE 时钟速率访问 AUX 模块，而系统 CPU 和 µDMA 的访问始终以总线时钟速率进行。如表 19-1 所列，在两种条件下，AUX 总线时钟速率可以高于 SCE 时钟速率：

- SCE 时钟速率在活动工作模式下为 2MHz
- MCU 域处于活动状态

在这两种情况下，总线时钟速率都等于 24MHz。表 19-2 列出了各种 AUX 模块的操作时钟速率。

为了确保正确的时序，传感器控制器使用的模块和时钟资源必须尽可能以 SCE 时钟速率运行，因为总线时钟速率会在 MCU 电源模式改变时发生变化。这对于 SPIM 和 Timer01 外设尤为重要。

为了确保正确的时序，传感器控制器使用的模块和时钟资源必须尽可能以 SCE 时钟速率运行，因为总线时钟速率会在 MCU 电源模式改变时发生变化。这对于 SPIM 和 Timer01 外设尤为重要。

当总线时钟速率高于 SCE 时钟速率时，有两个副作用：

- 当总线时钟速率变得高于 SCE 时钟速率时，传感器控制器对 DDI 从器件、ADI 从器件和 Timer2 的访问会更快完成。
- 某些传感器控制器指令（如 *iobset* 和 *iobclr*）变得非原子化。传感器控制器与系统 CPU 或 µDMA 必须实现资源或模块所有权概念以避免不一致。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-2. AUX 操作时钟速率

模块	操作时钟速率(1)		
	AUX 总线速率	AUX SCE 速率	AUX Timer2 速率
别名空间		x	
AUX_SPIM	(X)	X	
AUX_MAC	(X)	X	
AUX_TIMER2			X
AUX_TDC	X		
AUX_EVCTL	X	(XX)	
AUX_SYSIF	X		
AUX_TIMER01	(X)	X	
AUX_SMPH	X		
AUX_ANAIF:ADC	X		
AUX_ANAIF:DAC	(X)	X	
AUX_DDI0_OSC	X		
AUX_ADI4	X		
AUX_AIODIO0		X	
AUX_AIODIO1		X	
AUX_AIODIO2		X	
AUX_AIODIO3		X	
AUX_RAM	X		
AUX_SCE		X	

(1) X = 默认速率
(X) = 可覆盖速率
(XX) = 事件同步的可覆盖速率

19.2.2 使用场景

19.2.2.1 MCU

当 MCU 域处于活动状态时，所有 AUX 模块都可以由 MCU 主器件以最小延迟访问并正常工作，且独立于 AUX 时钟和电源管理。因此，系统 CPU 应用程序无需配置 AUX 工作模式即可使用外设。系统 CPU 应用程序无法在待机 TI-RTOS 电源模式下使用 AUX 外设（唤醒除外）。然而，传感器控制器引擎可以在系统其余部分处于待机状态时运行，因此可用于将电流消耗降至最低。

TI-RTOS 默认请求断电工作模式。

19.2.2.2 传感器控制器

Sensor Controller Studio 框架确保传感器控制器在不同的工作模式之间主动切换，以尽可能降低任务电流消耗。

仅当 AON_PMCTL:RECHARGECFG.MODE = COMPARATOR 时，才允许传感器控制器在低功耗和断电工作模式下执行代码。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.2.3 SCE 时钟仿真

当 SCE 时钟源等于 SCLK_MF 或 SCLK_LF 且 MCU 域处于活动状态时，SCE 时钟源会被仿真。时钟仿真会导致 SCE 时钟周期抖动：

- SCLK_MF 时钟仿真：

  瞬时 SCE 时钟周期抖动等于 ±2 个 SCLK_HF 周期。
- SCLK_LF 时钟仿真：

  瞬时 SCE 时钟周期抖动为 2 个 SCLK_HF 周期。当仿真开始或结束时，单个 SCE 时钟周期会增加或减少 6 到 8 个 SCLK_HF 周期。

时钟仿真具有以下影响：

- 时钟仿真引入的微小时间不确定性仅影响以 SCE 时钟速率运行的模块。
- SPI SCLK 的频率配置可能需要防范较短的基时钟周期，以满足 I/O 时序要求。受影响的用例中 SCLK 频率为 1MHz 或更低，这种情况很少见。
- SCE 时钟周期抖动不具有累积性。

19.2.4 AUX RAM 保持

AUX RAM 可以在有或无阵列保持的情况下关闭。在 AUX RAM 电源时序控制期间、AUX 时钟处于非活动状态；因此，在 RAM 电源时序控制完成之前，所有对 AUX 的访问都会被暂停。

AON_PMCTL:RAMCFG.AUX_SRAM_RET_EN 控制 RAM 保持，而
AON_PMCTL:RAMCFG.AUX_SRAM_PWR_OFF 控制 RAM 电源。

19.3 传感器控制器

19.3.1 Sensor Controller Studio

TI 为传感器控制器提供了一款名为 Sensor Controller Studio 的开发工具。本节概述了传感器控制器的编程模型、传感器控制器任务开发，以及任务测试、调试和运行时日志记录。有关如何使用传感器控制器的完整文档，请参见 Sensor Controller Studio 本身，如节 19.3.1.4 所述。

编程模型和固件框架经过优化，实现了与系统 CPU 应用程序通信时的低开销、较小的 AUX RAM 内存占用，以及传感器控制器指令集的高效使用。为将功耗降至最低，传感器控制器在空闲时会自动进入待机模式。

19.3.1.1 编程模型

Sensor Controller Studio 基于工程。每个传感器控制器工程可能包含多达 8 个传感器控制器任务。工程的输出是一组被称为传感器控制器接口 (SCIF) 驱动程序的 C 源文件，这些文件可以轻松集成到系统 CPU 应用中。

传感器控制器是用户可编程的。传感器控制器任务代码编程语言使用与 C 语言类似的语法，但与 C 语言相比功能有限，因为它专门针对传感器控制器引擎的指令集和固件框架。例如，它仅支持 16 位变量，并且每个任务代码块都有预定义的角色（初始化、调度执行、事件处理和终止）。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

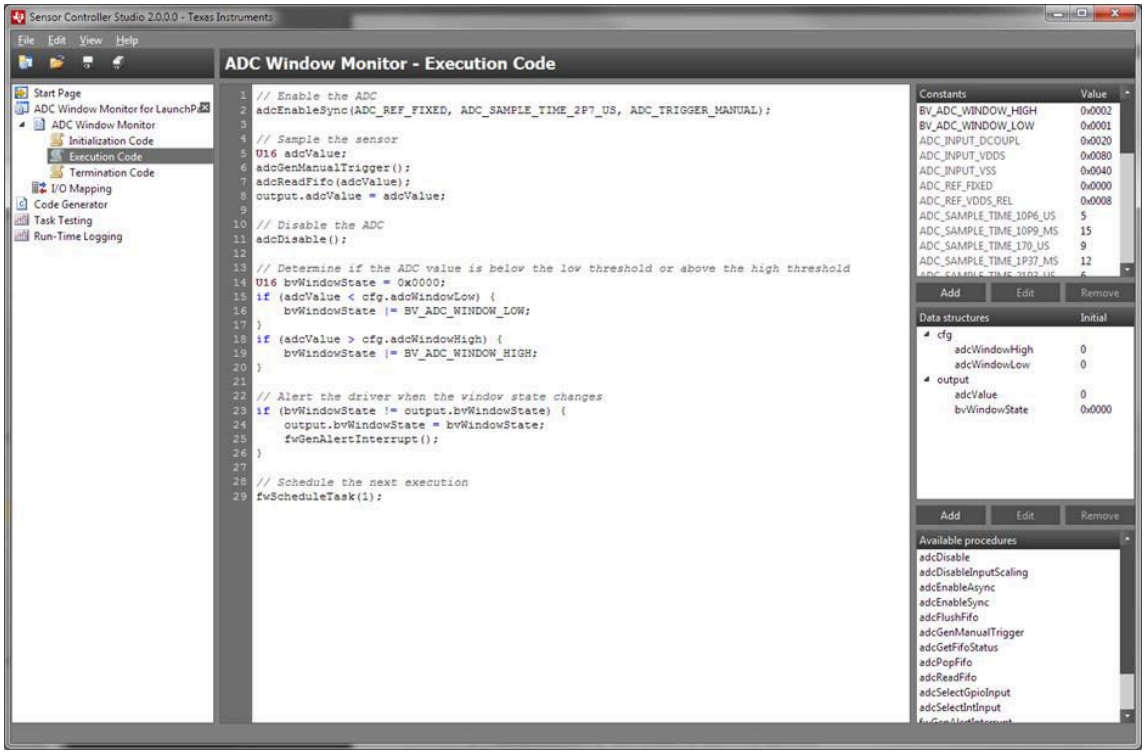


图 19-3. ADC 窗口监控器 — 执行代码

硬件外设和软件算法作为资源提供。对于每个任务，必须选择和配置一组资源以启用不同类型的功能，例如：

- 模拟和数字外设模块（例如 ADC）
- 模拟和数字通用 I/O 引脚
- 简单和复杂的软件算法
- 位操作串行接口
- 任务调度和事件处理
- 与系统 CPU 的通信

每个任务资源都会启用一组过程（相当于 C 语言中的函数）以及关联的常量和全局变量。传感器控制器从任务代码调用这些过程，从而可以组合使用多个资源，也可以在多个任务之间图 19-4 重复使用资源（例如 ADC）。

传感器控制器任务和系统 CPU 应用通过数据结构变量（位于 AUX RAM 中）交换数据。Sensor Controller Studio 会生成类型定义和其他所需定义，以便轻松访问这些变量。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

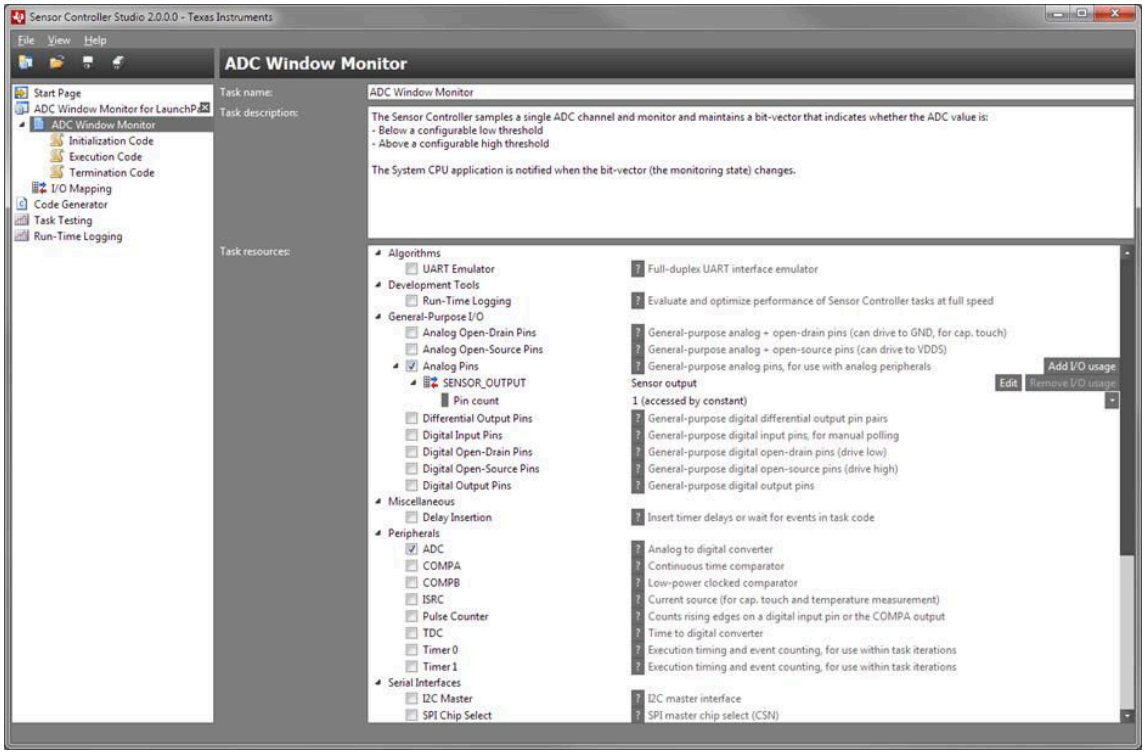


图 19-4. ADC 窗口监控器

19.3.1.2 任务开发

一个传感器控制器工程主要包含以下内容：

- 名称和说明以及其他工程属性
- 目标芯片选择及传感器控制器的其他静态配置
- 对于每个任务：
  - 名称和说明
  - 资源选择和配置
  - 资源定义的和用户定义的常量
  - 资源定义的和用户定义的数据结构变量（位于 AUX RAM 中）
  - 传感器控制器任务代码：
    - 初始化代码：当系统 CPU 应用程序启动该任务时运行
    - 执行代码：可以由 AON_RTC 定期触发，或由软件触发
    - 事件处理程序代码：可以由各种外设事件触发
    - 终止代码：当系统 CPU 应用程序停止该任务时运行
- I/O 引脚映射

生成的 SCIF 驱动程序由三部分组成：

- 一个通用的应用程序编程接口
- 驱动程序设置，包含所有项目特定的定义和数据，包括 AUX RAM 映像、I/O 引脚配置代码、供系统 CPU 应用程序使用的资源特定函数等
- 一个操作系统抽象层，允许驱动程序无缝地用于 TI-RTOS 等环境

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.3.1.3 任务测试、任务调试和运行时日志记录

Sensor Controller Studio 提供了用于测试、调试、评估和调优传感器控制器任务的工具。这些工具的共同点是，它们在目标芯片的实际传感器控制器上运行任务代码，并为实时显示数据结构变量提供通用支持。数据结构值也可以保存到文件中以供外部分析。

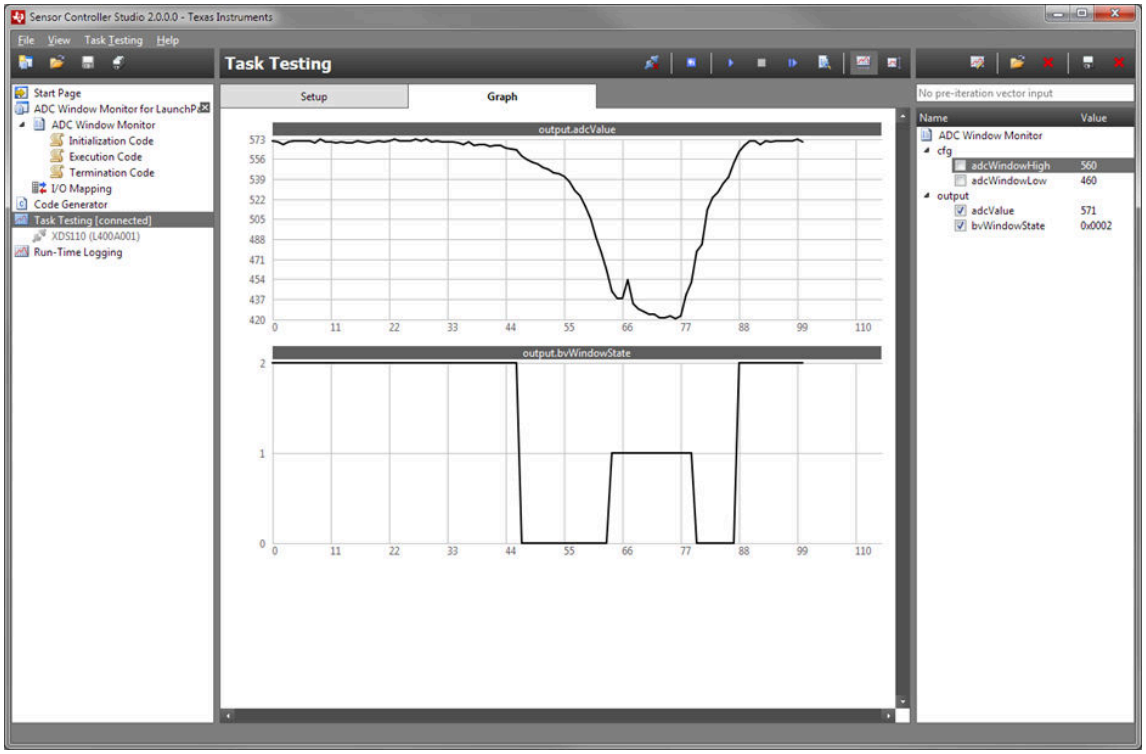


图 19-5. 任务测试

表 19-3 总结并比较了可用工具的最重要属性。

表 19-3. 任务测试和任务调试

	任务测试和任务调试	运行时日志记录
用途	评估、系统性测试和调试。 可以对选定的任务迭代进行指令级调试。	性能评估和优化。
执行速度	任务代码块内的所有代码均全速运行。 任务代码块之间的时序无法精确控制，并且取决于与 PC 的通信。	时序与实际应用程序匹配。
准备工作	为每个任务迭代指定一系列操作。	启用运行时日志记录资源，并使用任务代码中的相关过程来控制日志记录。
通信接口	JTAG	UART + JTAG、仅 UART 或 TCP/IP
支持的任务数量	一次一个任务。	一个项目中的一个或多个任务。
数据结构日志记录	记录所有数据结构。 无溢出。	记录选定的数据结构。 可能发生溢出，以静默方式丢弃数据。
数据结构编辑	可以修改所有数据结构值，但只能在目标停止时进行。 可以从文件加载新的数据结构成员值，并在任务迭代之间应用。	未记录的数据结构可以随时修改。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.3.1.4 文档

Sensor Controller Studio 附带帮助查看器 (图 19-6)，其中提供了与 Sensor Controller 开发相关的所有文档。本文档包括：

- 传感器控制器概念简介
- 详细的 Sensor Controller Studio 工具文档
- 资源和过程文档
- 任务代码编程语言参考和其他参考文档
- 详细修订历史记录

使用 Doxygen 语法记录了生成的 SCIF 驱动程序。此文档包括特定于项目和任务的信息。

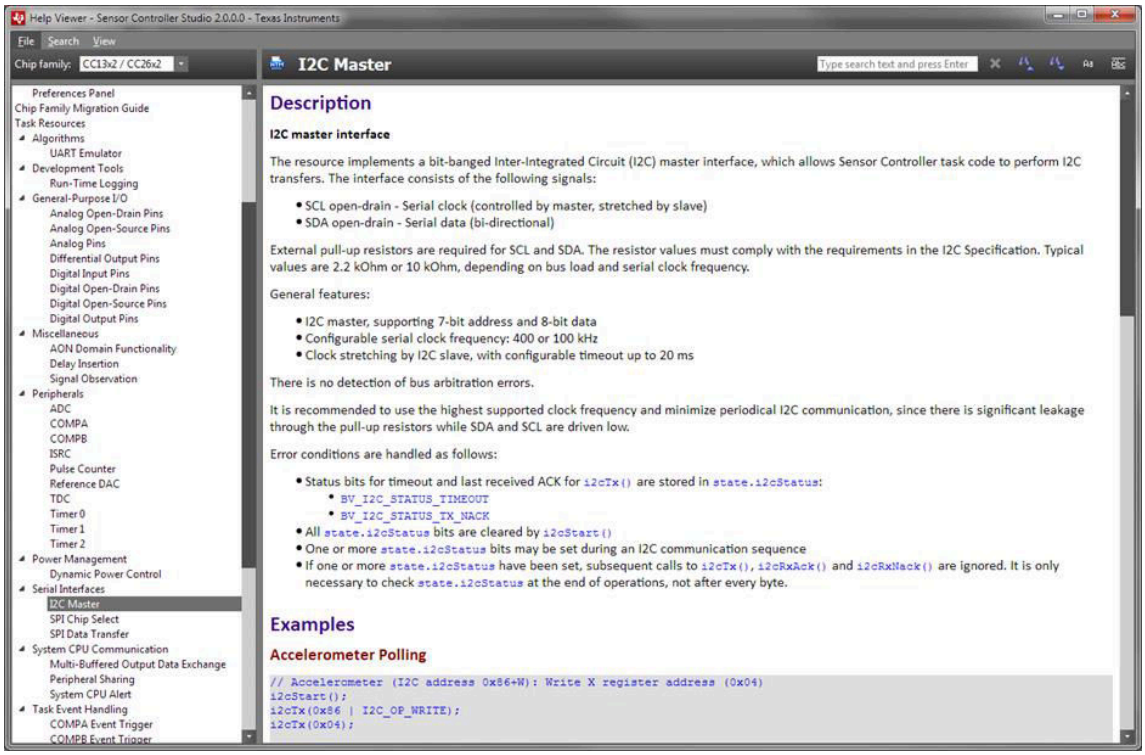


图 19-6. 帮助查看器

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.3.2 传感器控制器引擎 (SCE)

本节概述了传感器控制器引擎 CPU 内核。对于大多数用户而言，这些信息可以帮助理解传感器控制器引擎的功能和限制，同时也可作为调试传感器控制器任务时的参考。

通常不需要为传感器控制器编写任何汇编代码。如有必要，可以在 Sensor Controller Studio 中创建自己的任务资源和过程，例如在以下情况下：

- 需要访问所提供过程不支持的硬件寄存器或寄存器字段。
- 需要极致优化的代码，使用 Sensor Controller Studio 编译器无法生成的代码结构。

19.3.2.1 寄存器

共有八个 16 位通用寄存器：R0 至 R7。少数指令需要专用 R0 和 R1。

共有四个状态标志：零标志 (Z)、负标志 (N)、进位标志 (C) 和溢出标志 (V)。

有一个 16 位程序计数器 PC。还有一个专用的堆栈用于存储程序计数器，允许三级子例程调用。传感器控制器固件框架使用了其中一级，这意味着过程可以使用两级。

流水线风险

由于内部流水线以及对最小化寄存器旁路逻辑的考虑，专用 R0 的指令需要特别注意。在任何将 R0 用作专用寄存器的指令之前紧接的指令不得从存储器或 I/O 加载 R0。受影响的指令包括：

```
ld Rd, [Rs+R0]
```

```
st Rd, [Rs+R0]
```

```
jmp R0
```

```
jsr R0
```

违反此规则将导致使用 R0 的先前值，而不是新加载的值。

19.3.2.2 存储器架构

指令和数据的存储器访问

数据和指令存储器寻址以 16 位为导向（地址递增 1 对应于 2 个字节）。数据字和指令映射到同一存储器地址空间。在 CC13x2/CC26x2 器件上，AUX RAM 为 4kB，地址范围为 0x0000 至 0x07FF。

数据存储器通过加载和存储指令进行访问，支持直接寻址、寄存器间接寻址、带后增量的寄存器间接寻址和寄存器索引寻址模式。唯一支持的访问大小为 16 位。

除 AUX RAM 外，无法访问其他存储器。

模块寄存器的 I/O 访问

传感器控制器可以通过 I/O 地址空间访问 AUX 域模块寄存器。I/O 寻址以 8 位为导向（地址递增 1 对应于 1 个字节），但寄存器别名空间（即地址 0 至 255）除外，其中每个条目映射到一个 16 位寄存器。

I/O 寄存器通过输入和输出指令进行访问，支持：

- 对 16 位寄存器的 8 个 LSB 进行直接 1 位访问
- 直接或寄存器间接 16 位访问

不支持非对齐的 16 位访问。

无法访问 AUX 域之外的寄存器。同样，也无法通过 I/O 地址空间访问 AUX RAM 和 AUX_SCE 模块。

19.3.2.3 程序流程

常规程序流程通过显式使用跳转和分支、条件分支、子例程调用和返回指令来控制。还可以通过使用 wev0 或 wev1 指令暂停执行，在恢复执行之前等待指定事件。任何情况下都不支持抢占。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

Sensor Controller 通过执行 sleep 指令进入待机模式。然后，它可以通过八个事件向量之一从待机模式唤醒并开始执行，这些事件向量通过 AUX_SYSIF 配置。

零开销循环

支持通过 loop 指令实现一级零开销循环。该机制存储循环的第一条指令和最后一条指令的地址，并启动循环计数器。

19.3.2.4 指令集

19.3.2.4.1 指令时序

传感器控制器在工作模式下以 24MHz 运行，在低功耗模式下以 2MHz 运行。所有传感器控制器指令都使用 2 个时钟周期，但以下情况除外：

- I/O 访问 (in out 、iobset 、iobclr 和 iobtst) ADI 和 DDI 寄存器，它们通过多周期接口访问。这些寄存器与模拟外设和振荡器相关。
- 电源管理指令 (wev0 、wev1 和 sleep)，用于等待事件信号。

在访问内存和模块寄存器时，传感器控制器的优先级高于系统 CPU 和 µDMA。ADI 和 DDI 的访问可能会因正在进行的系统 CPU 访问而被延迟。

19.3.2.4.2 指令前缀

某些带有 8 位或 10 位立即数的指令可以通过在其紧邻前方执行一条前缀指令 pfix 来扩展为 16 位立即数。该前缀仅对下一条指令有效。

下表中的最后一列指示了可以扩展的立即数 (如果有)。

Sensor Controller Studio 汇编器会根据需要自动添加前缀指令，并检测无效或超出范围的立即数。汇编器还会修复目标地址超出范围的分支指令。

19.3.2.4.3 指令

表 19-4. 存储器字访问

语法	说明	运行	状态标志				编码 [15:0]	前缀
			Z	N	C	V		
ld Rd, [#addr]	直接加载	Rd = mem[addr]	—	—	—	—	0ddd.10aa.aaaa.aaaa	addr
ld Rd, [Rs]	间接加载	Rd = mem[Rs]	—	—	—	—	1ddd.1111.0000.1sss	
ld Rd, [Rs++]	间接加载，后增量	Rd = mem[Rs], Rs++	—	—	—	—	1ddd.1111.0001.0sss	
ld Rd, [Rs+R0]	索引加载	Rd = mem[Rs+R0]	—	—	—	—	1ddd.1111.0001.1sss	
st Rd, [#addr]	直接存储	mem[addr] = Rd	—	—	—	—	0ddd.11aa.aaaa.aaaa	addr
st Rd, [Rs]	间接存储	mem[Rs] = Rd	—	—	—	—	1ddd.1111.0010.1sss	
st Rd, [Rs++]	间接存储后增量	mem[Rs] = Rd, Rs++	—	—	—	—	1ddd.1111.0011.0sss	
st Rd, [Rs+R0]	索引存储	mem[Rs+R0] = Rd	—	—	—	—	1ddd.1111.0011.1sss	

表 19-5. I/O 字访问

语法	说明	运行	状态标志				编码 [15:0]	前缀
			Z	N	C	V		
in Rd, [#addr]	直接输入	Rd = reg[addr]	—	—	—	—	1ddd.1001.aaaa.aaaa	addr
in Rd, [Rs]	间接输入	Rd = reg[Rs]	—	—	—	—	1ddd.1111.0000.0sss	
out Rd, [#addr]	直接输出	reg[addr] = Rd	—	—	—	—	1ddd.1011.aaaa.aaaa	addr

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-5. I/O 字访问（续）

语法	说明	运行	状态标志				编码 [15:0]	前缀
			Z	N	C	V		
out Rd, [Rs]	间接输出	reg[Rs] = Rd	—	—	—	—	1ddd.1111.0010.0sss	

表 19-6. I/O 位访问

语法	说明	运行	状态标志				编码 [15:0]	前缀
			Z	N	C	V		
iobclr #imm, [#addr]	直接 I/O 位清零	reg[addr] &= ~2^imm	—	—	—	—	010i.01ii.aaaa.aaaa	addr
iobset #imm, [#addr]	直接 I/O 位设置	reg[addr] |= 2^imm	—	—	—	—	011i.01ii.aaaa.aaaa	addr
iobtst #imm, [#addr]	直接 I/O 位测试	reg[addr] & 2^imm	—	—	√	—	001i.01ii.aaaa.aaaa	addr

表 19-7. 寄存器访问

语法	说明	运行	状态标志				编码 [15:0]	前缀
			Z	N	C	V		
ld Rd, #simm	立即数加载	Rd = simm	—	—	—	—	0ddd.00ii.iiii.iiii	simm
ld Rd, Rs	Load register（加载寄存器）	Rd = Rs	—	—	—	—	1ddd.1101.0100.0sss	

表 19-8. 逻辑运算

语法	说明	运行	状态标志				编码 [15:0]	前缀
			Z	N	C	V		
and Rd, #imm	立即数与	Rd &= imm	x	x	0	0	1ddd.0000.iiii.iiii	imm
or Rd, #imm	立即数或	Rd |= imm	x	x	0	0	1ddd.0010.iiii.iiii	imm
xor Rd, #imm	立即数异或	Rd ^= imm	x	x	0	0	1ddd.0100.iiii.iiii	imm
tst Rd, #imm	立即数测试	Rd & imm	x	x	0	0	1ddd.1100.iiii.iiii	imm
and Rd, Rs	寄存器与	Rd &= Rs	x	x	0	0	1ddd.1101.0000.0sss	
or Rd, Rs	寄存器或	Rd |= Rs	x	x	0	0	1ddd.1101.0000.1sss	
xor Rd, Rs	寄存器异或	Rd ^= Rs	x	x	0	0	1ddd.1101.0001.0sss	
tst Rd, Rs	寄存器测试	Rd & Rs	x	x	0	0	1ddd.1101.0011.0sss	
inv Rd	寄存器取反	Rd = ~Rd	x	x	0	0	1ddd.1101.1001.0010	

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-9. 算术运算

语法	说明	运行	状态标志				编码 [15:0]	前缀
			Z	N	C	V		
add Rd, #simm	立即数加	Rd += simm	x	x	x	x	1ddd.1000.iiii.iiii	simm
cmp Rd, #simm	立即数比较	Rd – simm	x	x	x	x	1ddd.1010.iiii.iiii	simm
sub Rd, Rs	寄存器减	Rd -= Rs	x	x	x	x	1ddd.1101.0001.1sss	
add Rd, Rs	寄存器加	Rd += Rs	x	x	x	x	1ddd.1101.0010.0sss	
cmp Rd, Rs	寄存器比较	Rd – Rs	x	x	x	x	1ddd.1101.0010.1sss	
subr Rd, Rs	寄存器反向减	Rd = Rs – Rd	x	x	x	x	1ddd.1101.0011.1sss	
abs Rd	寄存器绝对值	Rd = (Rd >= 0) ? Rd : -Rd	x	x	x	x	1ddd.1101.1001.0000	
neg Rd	寄存器取反	Rd = -Rd	x	x	x	x	1ddd.1101.1001.0001	

表 19-10. 移位操作

语法	说明	运行	状态标志				编码 [15:0]	前缀
			Z	N	C	V		
lsl Rd, Rs	寄存器逻辑左移	Rd <<= Rs	x	x	x	0	1ddd.1101.1000.0sss	
lsr Rd, Rs	寄存器逻辑右移	Rd >>= Rs	x	x	x	0	1ddd.1101.1000.1sss	
asr Rd, Rs	寄存器算术右移	Rd >>= Rs, 保留符号	x	x	x	0	1ddd.1101.1001.1sss	
lsl Rd, #imm	立即数逻辑左移(1)	Rd <<= imm	x	x	x	0	1ddd.1101.1010.0iii	
lsr Rd, #imm	立即数逻辑右移(1)	Rd >>= imm	x	x	x	0	1ddd.1101.1010.1iii	
asr Rd, #imm	立即数算术右移(1)	Rd >>= imm, preserve sign	x	x	x	0	1ddd.1101.1011.1iii	

(1) 立即数移位可为 1 到 8 位，其中 imm = 0 对应 8 位。

表 19-11. 程序流控制

语法	说明	运行	状态标志				编码 [15:0]	前缀
			Z	N	C	V		
jmp addr	直接跳转	pc = addr	—	—	—	—	0000.01aa.aaaa.aaaa	addr
jsr addr	直接子例程跳转	push(stack, pc + 1), pc = addr	—	—	—	—	0001.01aa.aaaa.aaaa	addr
jmp R0	间接跳转	pc = R0	—	—	—	—	1000.1101.1011.0111	
jsr R0	间接子例程跳转	push(stack, pc + 1), pc = R0	—	—	—	—	1001.1101.1011.0111	
rts	子例程返回	pc = pop(stack)	—	—	—	—	1010.1101.1011.0111	
bra rel	相对分支	pc = pc + 1 + rel	—	—	—	—	1000.1110.rrrr.rrrr	
b<cc> rel	条件满足时相对分支	if (cc) pc = pc + 1 + rel	—	—	—	—	1ccc.c110.rrrr.rrrr	
bev0 #ev, rel	事件 0 时分支	if (!events[ev]) pc = pc + 1 + rel	—	—	—	—	1eee.0001.rrrr.rrrr	
bev01#ev, rel	事件 1 时分支	if (events[ev]) pc = pc + 1 + rel	—	—	—	—	1eee.0011.rrrr.rrrr	

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

对于所有其他指令以及当不执行条件分支时，程序计数器递增 1: pc = pc + 1。有关条件操作，请参阅表 19-12。

表 19-12. 程序流程条件

语法 <cc>	说明	条件	编码 [3:0]
gtu	大于，无符号	!C & !Z	0010
geu / iob0	大于或等于，无符号/ 测试 I/O 位 = 0	!C	0100
eq / z	相等/为零	Z	0110
novf	无溢出	!V	1000
pos	正	!N	1010
ges	大于或等于，有符号	(N & V) | (!N & !V)	1100
gts	大于，有符号	((N & V) | (!N & !V)) & !Z	1110
leu	小于或等于，无符号	C | Z	0011
ltu / iob1	小于，无符号/测试 I/O 位 = 1	C	0101
neq / nz	不等于/非零	!Z	0111
ovf	溢出	V	1001
neg	负	N	1011
lts	小于，有符号	(N & !V) | (!N & V)	1101
les	小于或等于，有符号	(N & !V) | (!N & V) | Z	1111

表 19-13. 循环流控制

语法	说明	运行	状态标志				编码 [15:0]	前缀
			Z	N	C	V		
loop R1, rel	循环寄存器， n = 1..255(1)	lc = LSB(R1), ls = pc + 1, le = pc + rel	x	x	x	x	1000.0101.rrrr.rrrr	
loop #n, rel	循环立即数， n = 2^x (x = 1..7)(1)	lc = n, ls = pc + 1, le = pc + rel	x	x	x	x	1nnn.0101.rrrr.rrrr	

(1) 汇编器会调整循环结束标签的偏移量，以便将其放置在循环的最后一条指令之后。

表 19-14. 电源管理

语法	说明	运行	状态标志				编码 [15:0]	前缀
			Z	N	C	V		
wev0 #ev	等待事件 0	停止直到 events[ev] == 0	x	x	x	x	1eee.1101.1011.0000	
wev1 #ev	等待事件 1	停止直到 events[ev] == 1	x	x	x	x	1eee.1101.1011.0001	
休眠	睡眠	停止直到唤醒，然后 pc = 2 * 向量	x	x	x	x	1011.1101.1011.0111	

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-15. 其他

语法	说明	运行	状态标志				编码 [15:0]	前缀
			Z	N	C	V		
nop	无操作	R7 = R7（无影响）	x	x	x	x	1111.1101.0100.0111	
pfix #imm	前缀(1)	在下一条指令上使用前缀	x	x	x	x	1000.0110.iiii.iiii	
dw #imm	数据字	插入 16 位立即数据值	x	x	x	x	iiii.iiii.iiii.iiii	

(1) 前缀指令 (pfix) 由汇编器在需要时自动插入。请勿手动插入此指令。

19.3.2.5 SCE 事件接口

表 19-16 总结了传感器控制器事件接口。

表 19-16. SCE 事件接口

事件名称	编号	说明
AUX_PROG_DLY_IDLE	0	可编程延迟事件。请参阅节 19.3.2.7 或 AUX_EVCTL:PROGDLY（节 19.8.3）。
AUX_COMPA	1	比较器 A 事件
AUX_COMPB	2	比较器 B 事件
AUX_TDC_DONE	3	TDC 转换完成或超时事件
AUX_TIMER0_EV_OR_IDLE	4	Timer0 已达到其目标值或处于空闲状态
AUX_TIMER1_EV_OR_IDLE	5	Timer1 已达到其目标值或处于空闲状态
AUX_ADC_FIFO_NOT_EMPTY	6	ADC FIFO 包含样本
SCEWEV_PROG	7	可编程事件。请参阅节 19.8.3 中的 AUX_EVCTL:SCEWEVCFG0 和 AUX_EVCTL:SCEWEVCFG1。

以下指令使用表 19-16 列出的事件进行控制器内核电源管理或程序流控制：

- *wev0* 和 *wev1* 指令：

  传感器控制器暂停指令执行，冻结内部状态，并进入时钟门控状态，直到所选事件变为 0 或 1。
- *bev0* 和 *bev1* 指令：

  传感器控制器根据事件电平进行条件分支。

SCEWEV_PROG 事件可以编程为最多两个具有可配置极性的事件的函数。例如，当传感器控制器必须等待带超时的 I/O 事件时，此编程功能非常有用，可以在消耗最小功率的同时实现这一等待。

19.3.2.6 数学加速器 (MAC)

传感器控制器可以利用 MAC 模块进行数学运算，例如：

- 16 位有符号或无符号乘法，并可选择对结果进行累加
- 可编程项和累加器的 16 位或 32 位有符号或无符号加法

这些运算的结果始终存储在 40 位有符号累加器中。传感器控制器可以直接对累加器执行以下操作：

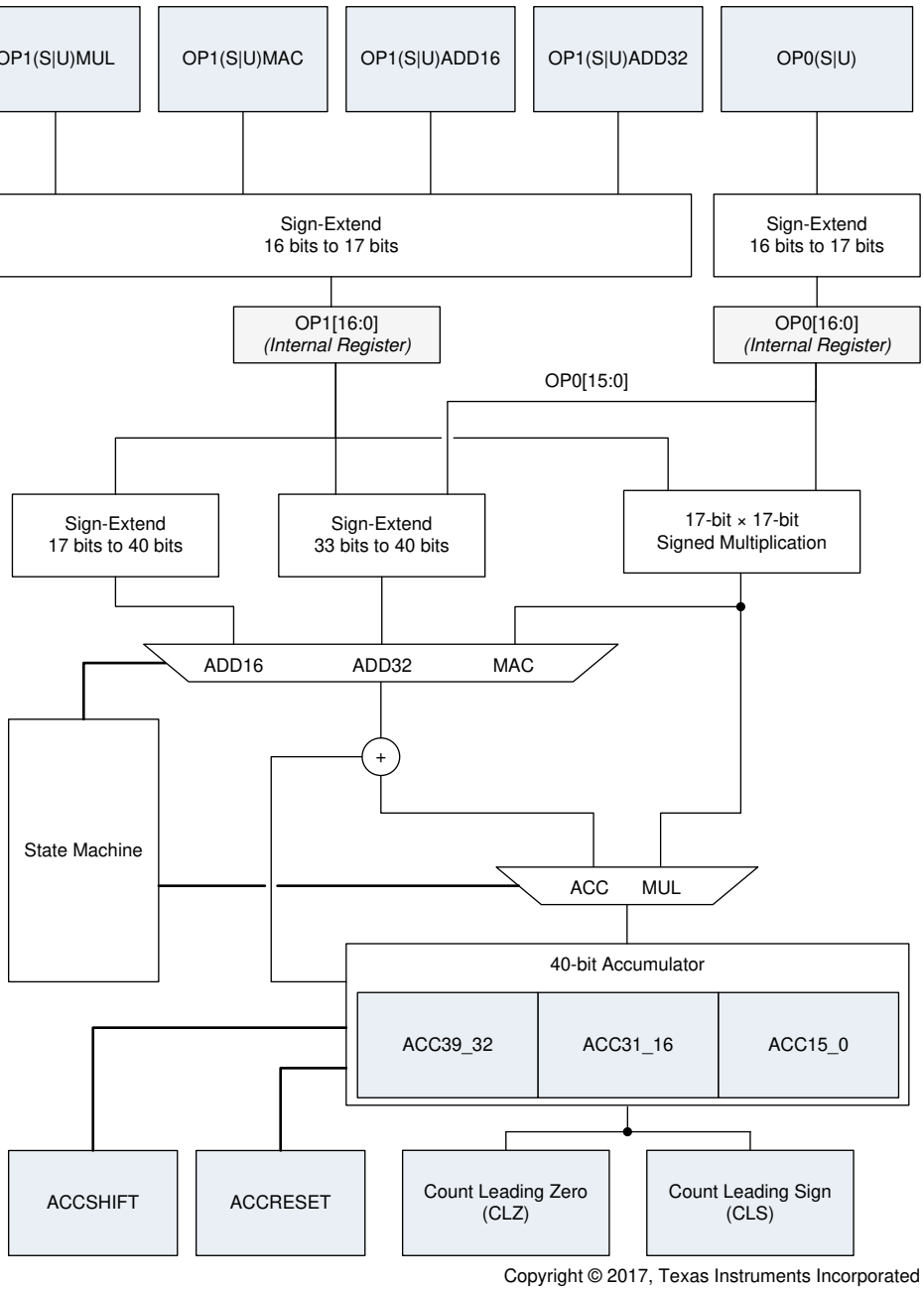
- 初始化累加器
- 复位累加器
- 读取累加器的任意 16 位切片
- 读取累加器中前导符号位或前导零位的数量
- 对累加器执行逻辑移位和算术右移

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

图 19-7 显示了 MAC 的方框图。

为了保持时序一致，传感器控制器和 MAC 的操作速率必须相等。因此，传感器控制器应用程序必须将 AUX_SYSIF:PEROPRATE 寄存器的 MAC_OP_RATE 位设置为 SCE_RATE（请参阅节 19.8.9）。

虽然不是主要用例，但系统 CPU 也可以使用 MAC。为了在这种情况下实现一致的时序，应用程序必须将 AUX_SYSIF:PEROPRATE 寄存器的 MAC_OP_RATE 位设置为 BUS_RATE（请参阅节 19.8.9）。



Copyright © 2017, Texas Instruments Incorporated

图 19-7. MAC 方框图

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

如果操作速率不相等，加法和乘法运算的结果仍然有效，但完成运算所需的时间可能会变化。

加法和乘法运算需要多个外设时钟周期才能完成。请考虑以下从 Sensor Controller Studio 生成的汇编示例。

```
# Sample assembly (Example code produced by the Sensor Controller Studio)
# The contents of r4 are assumed to be signed in this example
0: out     R4,[#IOP_MAC_OP1SADD16]    ; add r4 to the accumulator
1: in      R1,[#IOP_MAC_15_0]         ; Store result bits 15:0 in r1
2: out     R4,[#IOP_MAC_OP1SMUL]      ; Multiply r4 with operand 0(assumed pre-programmed)
3: in      R2,[#IOP_MAC_31_16]        ; Store result bits 31:16 in r2
4: in      R3,[#IOP_MAC_15_0]         ; Store result bits 15:0  in r3
# The Sensor Controller uses the alias I/O memory space
IOP_MAC_OP1SMUL                 equ 11
IOP_MAC_OP1SADD16               equ 15
IOP_MAC_31_16                   equ 24
IOP_MAC_15_0                    equ 23
```

如图 19-8 所示，加法运算需要一个外设时钟周期才能完成。由于传感器控制器在触发运算后会立即尝试读取结果，传感器控制器会暂停一个时钟周期。这种情况会门控传感器控制器的时钟，直到结果就绪。

图 19-8 还显示乘法运算需要五个外设时钟周期才能完成。由于传感器控制器在触发运算后会立即尝试读取结果，传感器控制器会暂停五个时钟周期。传感器控制器的时钟会被门控，直到结果就绪。乘加运算需要相同数量的外设时钟周期来完成。

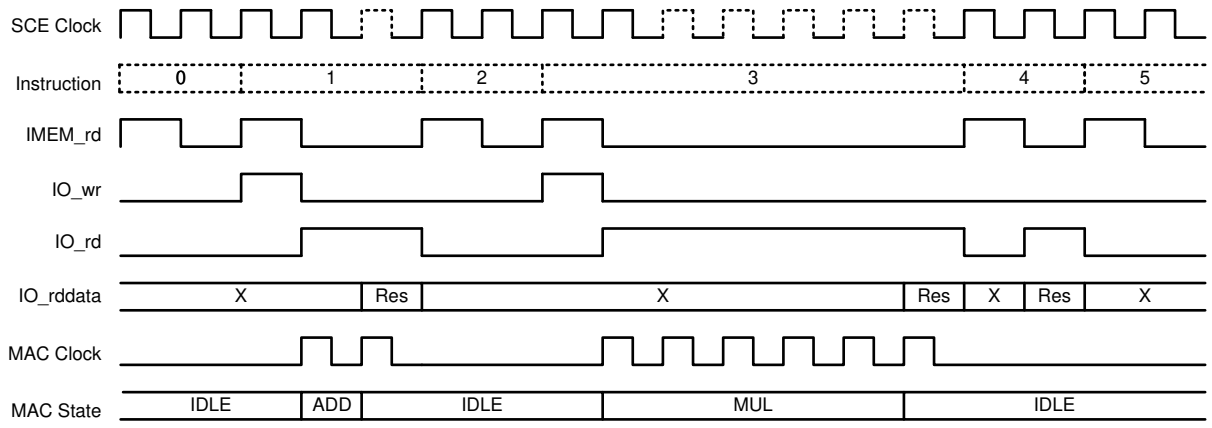


图 19-8. MAC 时序图

19.3.2.7 可编程微秒延迟

表 19-16 介绍了 AUX_PROG_DLY_IDLE 事件。该事件的电平取决于 16 位 AUX_EVCTL:PROGDLY 计数器的值。当计数器值为零时，电平为高；否则，电平为低。

当计数器被加载一个非零值时，它会随着占空比为 50% 的 500kHz 时钟的每个边沿递减 1 个 LSB，直到计数器值为零。500kHz 时钟信号以 SCE 速率同步。因此，由于边沿检测逻辑易受同步抖动的影响，平均而言，1 个 LSB 对应 1µs 的延迟。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

下面的示例展示了如何将传感器控制器的后续执行延迟至少 15µs。

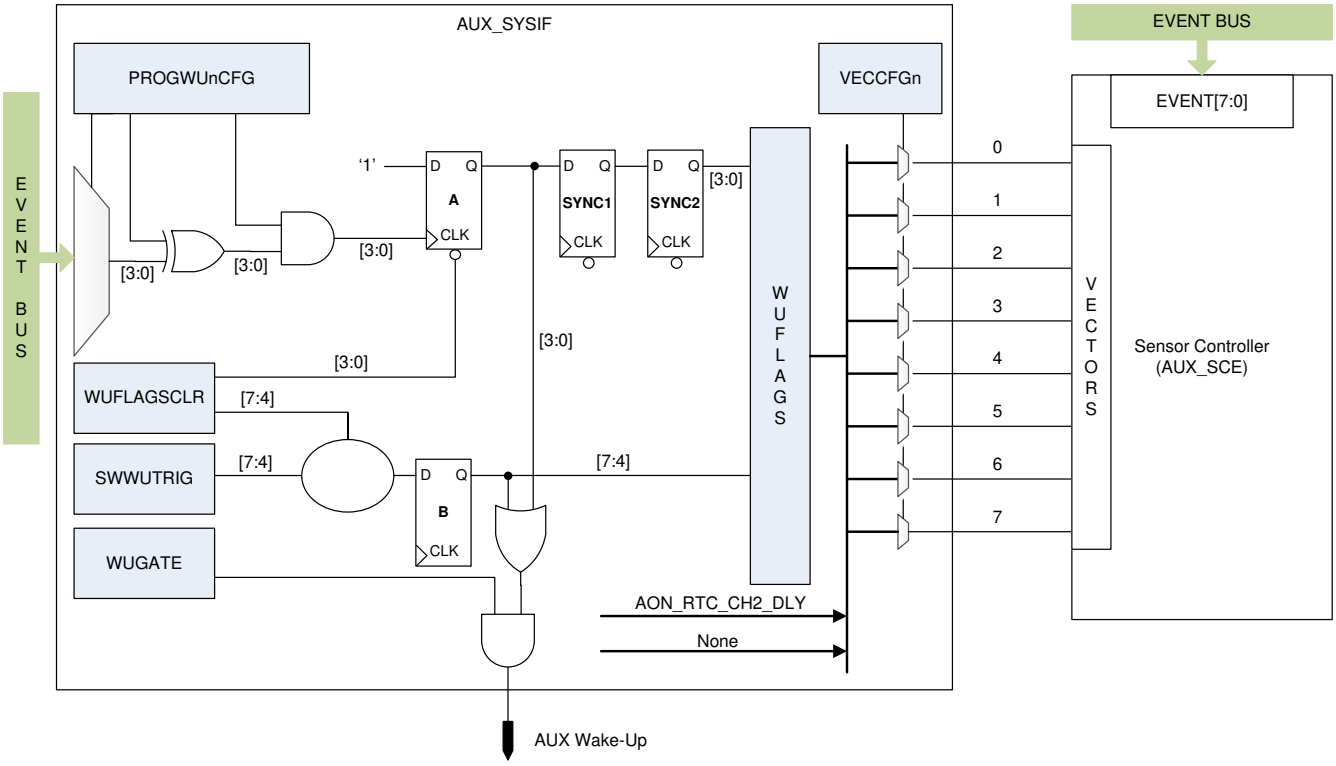
```
# Sample assembly (Example code produced by the Sensor Controller Studio)
0: ld      R4, #(15+1)              ; Load value for AUX_EVCTL:PROGDLY counter
1: out     R4, [#IOP_EVCTL_PROGDLY] ; Load the AUX_EVCTL:PROGDLY counter
2: wev1    #0                       ; Wait until AUX_EVCTL:PROGDLY counter is 0.
3: ...
# The Sensor Controller uses the alias I/O memory space
IOP_EVCTL_PROGDLY                 equ 73
```

AUX_EVCTL:PROGDLY 的写入时刻与 500kHz 时钟边沿之间存在随机相位。因此，需要在 15 µ s 的基础上增加 1 µ s，以确保至少 15 µ s 的延迟。

AUX_EVCTL:PROGDLY 计数器仅在活动和低功耗工作模式下有效。在断电工作模式下不得使用 AUX_PROG_DLY_IDLE 事件。

19.3.2.8 唤醒事件处理

图 19-9 显示传感器控制器实现了一个 8 位事件接口和一个 8 位矢量化唤醒事件接口。



Copyright © 2017, Texas Instruments Incorporated

图 19-9. SCE 唤醒和事件接口

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

AUX_SYSIF:VECCFGn 寄存器将每个矢量输入映射到一个可选的唤醒事件（请参阅 节 19.8.9）。表 19-17 列出了可能的唤醒事件。

表 19-17. 唤醒事件接口

事件映射	值	说明
无	0	禁用矢量输入
PROG_WU0	1	AUX_SYSIF:WUFLAGS.PROG_WU0
PROG_WU1	2	AUX_SYSIF:WUFLAGS.PROG_WU1
PROG_WU2	3	AUX_SYSIF:WUFLAGS.PROG_WU2
PROG_WU3	4	AUX_SYSIF:WUFLAGS.PROG_WU3
SW_WU0	5	AUX_SYSIF:WUFLAGS.SW_WU0
SW_WU1	6	AUX_SYSIF:WUFLAGS.SW_WU1
SW_WU2	7	AUX_SYSIF:WUFLAGS.SW_WU2
SW_WU3	8	AUX_SYSIF:WUFLAGS.SW_WU3
AON_RTC_CH2_DLY	9	AUX_EVCTL:EVSTAT2.AON_RTC_CH2_DLY

唤醒事件接口配置正确后，传感器控制器会执行*睡眠*指令。*睡眠*指令会停止指令执行、冻结内部状态并对传感器控制器进行时钟选通，直到设置一个或多个唤醒输入。然后，指令执行将从与最高优先级的唤醒输入相对应的矢量地址恢复。最高优先级分配给矢量 0，优先级线性下降到矢量输入 7，如 表 19-18 中所示。

表 19-18. 唤醒矢量优先级

唤醒输入	处理程序地址	优先级
0	0x0	最高
1	0x2	
2	0x4	
3	0x6	
4	0x8	
5	0xA	
6	0xC	
7	0xE	最低

如果已经设置一个或多个唤醒输入，则睡眠指令会立即使传感器控制器从最高优先级的矢量地址执行。

将 表 19-17 中列出的唤醒事件与 图 19-9 进行比较，可以看出除 AON_RTC_CH2_DLY 之外的所有可能的唤醒事件也可以驱动 AUX 域唤醒事件。AUX 域唤醒事件由以下事件驱动：

- 四个可编程 AUX 唤醒事件可以配置如下：
  - 事件源，请参阅 节 19.8.9 中的 AUX_SYSIF:PROGWUnCFG.WU_SRC 寄存器。
  - 极性，请参阅 节 19.8.9 中的 AUX_SYSIF:PROGWUnCFG.POL 寄存器。
  - 级别或边缘选择性，请参阅 节 19.8.9 中的 AUX_SYSIF:WUFLAGSCLR 寄存器。
- 四个软件事件，请参阅 节 19.8.9 中的 AUX_SYSIF:SWWUTRIG 寄存器。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

系统 CPU 利用这些事件与传感器控制器进行握手。

因此，传感器控制器通常会：

- 配置一个或多个可编程的 AUX 唤醒事件
- 配置传感器控制器唤醒矢量接口
- 请求断电工作模式
- 执行*睡眠*指令
- 唤醒：
  - 请求“低功耗”或“有效”工作模式
  - 清除唤醒标志
  - 执行一个任务并在可能的情况下通知系统 CPU
- 启用 AUX 唤醒事件
- 请求断电工作模式
- 执行*睡眠*指令

*睡眠*指令期间请求的系统资源极少，这可以降低整体系统功耗。AUX 唤醒后，系统电源控制器会从外部强制进入低功耗模式或有效 AUX 工作模式。在清除唤醒标志之前，传感器控制器必须请求相同的工作模式。有关 AUX 工作模式切换的详细信息，请参阅 节 19.2。

19.3.2.9 访问 AON 域寄存器

传感器控制器通常无法访问 AON 域。不过，它可以访问某些系统功能。这些访问包括：

对以下电池监测器和温度传感器寄存器的读访问：

- AON_BATMON:BAT
- AON_BATMON:TEMP

有关更多说明，请参阅节 19.8.9 中的 AUX_SYSIF:BATMONBAT 和 AUX_SYSIF:BATMONTEMP 寄存器。

- 对以下实时时钟寄存器的读访问：
  - AON_RTC:SEC.VALUE[15:0]
  - AON_RTC:SUBSEC.VALUE[31:16]

  有关更多说明，请参阅节 19.8.9 中的 AUX_SYSIF:RTCSEC 和 AUX_SYSIF:RTCSUBSEC 寄存器。
- 更新实时时钟亚秒增量：

  有关更多说明，请参阅节 19.8.9 中的 AUX_SYSIF:RTCSUBSECINC0、AUX_SYSIF:RTCSUBSECINC1 和 AUX_SYSIF:RTCSUBSECINCCTL 寄存器。
- 清除实时时钟通道 2 事件：

  有关更多说明，请参阅节 19.8.9 中的 AUX_SYSIF:RTCEVCLR 寄存器。
- 触发 VDDR 电源轨的充电，并检测此类事件是否发生在某个时间间隔内：

  有关更说多明，请参阅节 19.3.2.10。
- 检测某个时间间隔内 MCU 域电源状态的变化：

  此类变化可能导致 SCE 时钟速率发生非累积性变化。有关更多说明，请参阅节 19.8.9 中的 AUX_SYSIF:CLKSHIFTDET 寄存器。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.3.2.10 VDDR 充电

当传感器控制器请求低功耗或断电工作模式时，可能会发生 VDDR 充电。充电会引起 VDDS 电流尖峰，这可能会影响 AUX 测量。传感器控制器必须遵循图 19-10 中所示的例程，以避免在敏感测量期间因 VDDR 充电而引起电源噪声。

该例程仅在 AON_PMCTL:RECHARGECFG.MODE 设置为 COMPARATOR 时有效，因为此设置允许传感器控制器在低功耗或断电工作模式下执行任务。

传感器控制器还可以检测系统电源控制器是否在某个时间间隔内对 VDDR 进行了充电。AUX_SYSIF:RECHARGEDET 中介绍了该过程（请参阅节 19.8.9）。

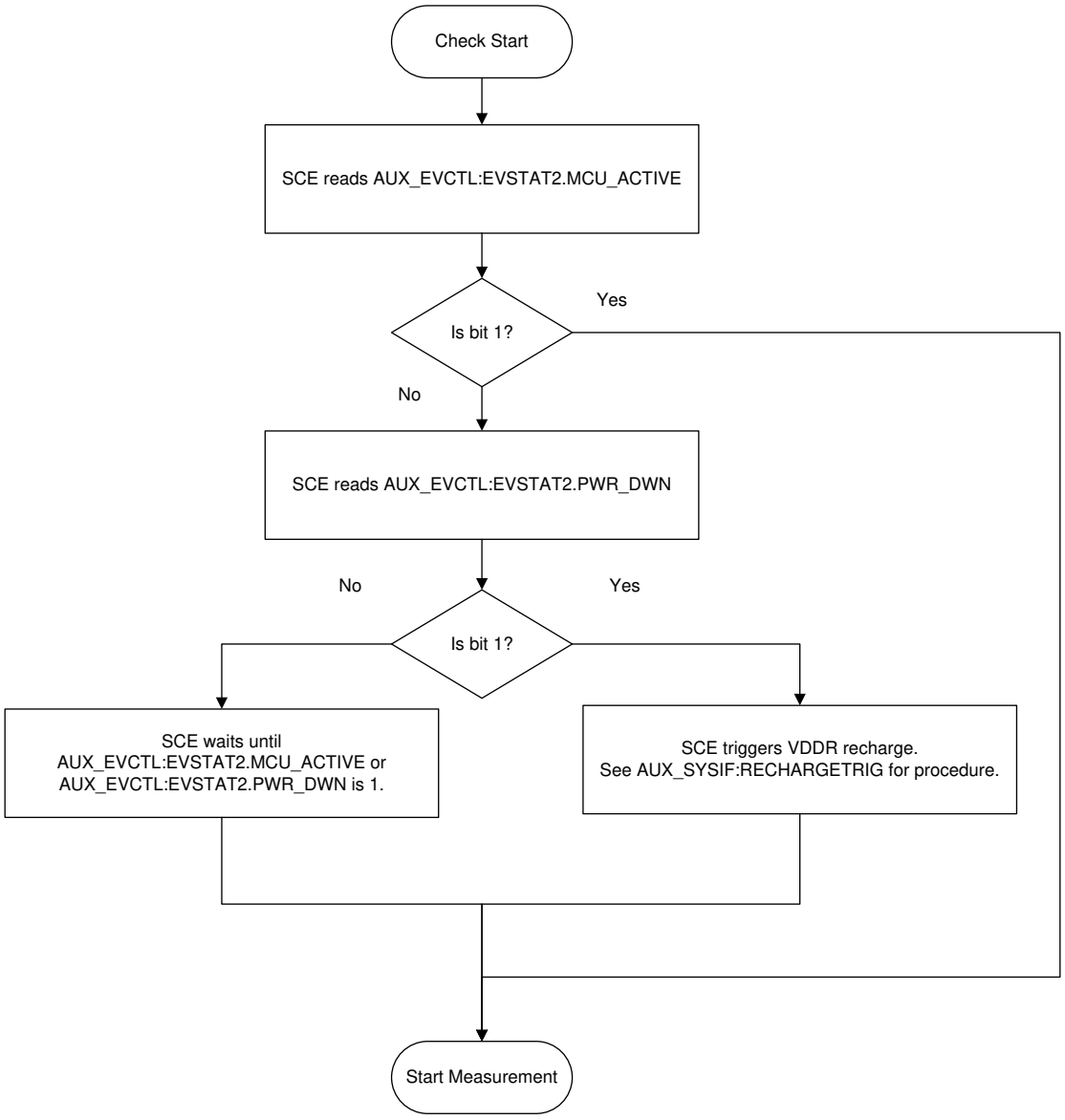


图 19-10. 避免 VDDR 充电电源噪声

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.4 数字外设模块

19.4.1 概述

表 19-19 列出了 AUX 域中的数字外设及其各自的操作速率和地址空间。有关操作速率的说明，请参阅节 19.2。

表 19-19. 数字外设

数字外设	操作时钟速率(1)		
	AUX 总线速率	AUX SCE 速率	AUX Timer2 速率
AUX_SPIM	(X)	X	
AUX_TIMER2			X
AUX_TDC	X		
AUX_TIMER01	(X)	X	
AUX_SMPH	X		
AUX_AIODIO0		X	
AUX_AIODIO1		X	
AUX_AIODIO2		X	
AUX_AIODIO3		X	
AUX_DDI0_OSC	X		

(1) X = 默认速率
(X) = 可覆盖速率
(XX) = 事件同步的可覆盖速率

传感器控制器和系统 CPU 可以共享 AUX 域中的数字外设。如果需要共享，建议使用 SMPH 中的硬件信标来仲裁对外设的访问。出于同样的原因，建议系统 CPU 和传感器控制器在使用后将外设保持在复位状态。表 19-21 列出了 TI 软件如何将资源分配给信标。

19.4.1.1 DDI 控制配置

AUX_DDI_OSC（在表 19-19 中列出）实际上并不是一个数字外设。如图 19-1 所示，系统 CPU 和传感器控制器都通过 AUX_DDI0_OSC 模块访问 DDI_0_OSC，以控制和配置系统时钟。AUX_DDI_OSC 模块支持四种不同类型的访问以提高效率，因为它通过多周期总线接口连接到模拟域。访问类型包括：

- 直接
- 设置
- 清除
- 已屏蔽

当总线时钟速率高于 SCE 时钟速率时，传感器控制器检测到的访问延迟更短。

如果系统 CPU 应用程序需要访问此模块，则必须使用 TI 提供的 DriverLib API 函数。Sensor Controller Studio API 函数在必要时包含访问，用户在开发代码时无需考虑此模块。

有关系统时钟说明，请参阅节 19.2。

19.4.2 AIODIO

19.4.2.1 简介

AUX 模拟 I/O 数字 I/O (AUX_AIODIOx) 外设控制着映射到独立封装 DIO 的 32 个 AUX I/O。传感器控制器是 AUX I/O 的主要拥有者，但系统 CPU 也可以配置和使用它们。系统 CPU 唯一相关的使用场景是在将控制权移交给传感器控制器之前配置 AUX I/O。

四个 AIODIO 实例配置和控制 32 个 AUX I/O。每个 AIODIO 外设管理 8 个 AUX I/O，如表 19-20 所示。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-20. AUX I/O 控制

外设	AUX I/O
AUX_AIODIO0	0 至 7
AUX_AIODIO1	8 至 15
AUX_AIODIO2	16 至 23
AUX_AIODIO3	24 至 31

外设特性包括：

- 32 个通用 I/O，具有可配置的 I/O 模式：
  - 数字输入
  - 数字输出
  - 开漏数字输出
  - 开源数字输出
- 多达八个 I/O 支持模拟传输。
- 可配置的输出数据源：
  - 外设
  - 事件
  - 寄存器

图 19-11 显示了单个 AUX I/O(k) 的方框图及其与 DIO(n) 的连接方式。

19.4.2.2 功能说明

19.4.2.2.1 映射到 DIO 引脚

表 13-2 中列出了特定封装型号的 AUXIO－DIO 映射。要将 DIO(n) 连接到 AUX IO(k)，用户必须将 IOC:IOCFGn.PORT_ID 设置为 AUX_IO。在 Sensor Controller Studio 中开发代码时，此操作不是必需的，因为该工具会处理引脚映射和所需的 IOC 配置。

19.4.2.2.2 配置

节 19.4.2.2.3 介绍了 AUX I/O 配置。在 Sensor Controller Studio 中开发代码时，不需要用户直接进行配置。这些用户直接与 Sensor Controller Studio API 进行交互。

19.4.2.2.3 GPIO 模式

AUX_AIODIOn:IOMODE（请参阅 节 19.8.2）将每个 AUX I/O 的 I/O 模式配置为：

- 输入
- 数字输出
- 开漏数字输出
- 开源数字输出

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

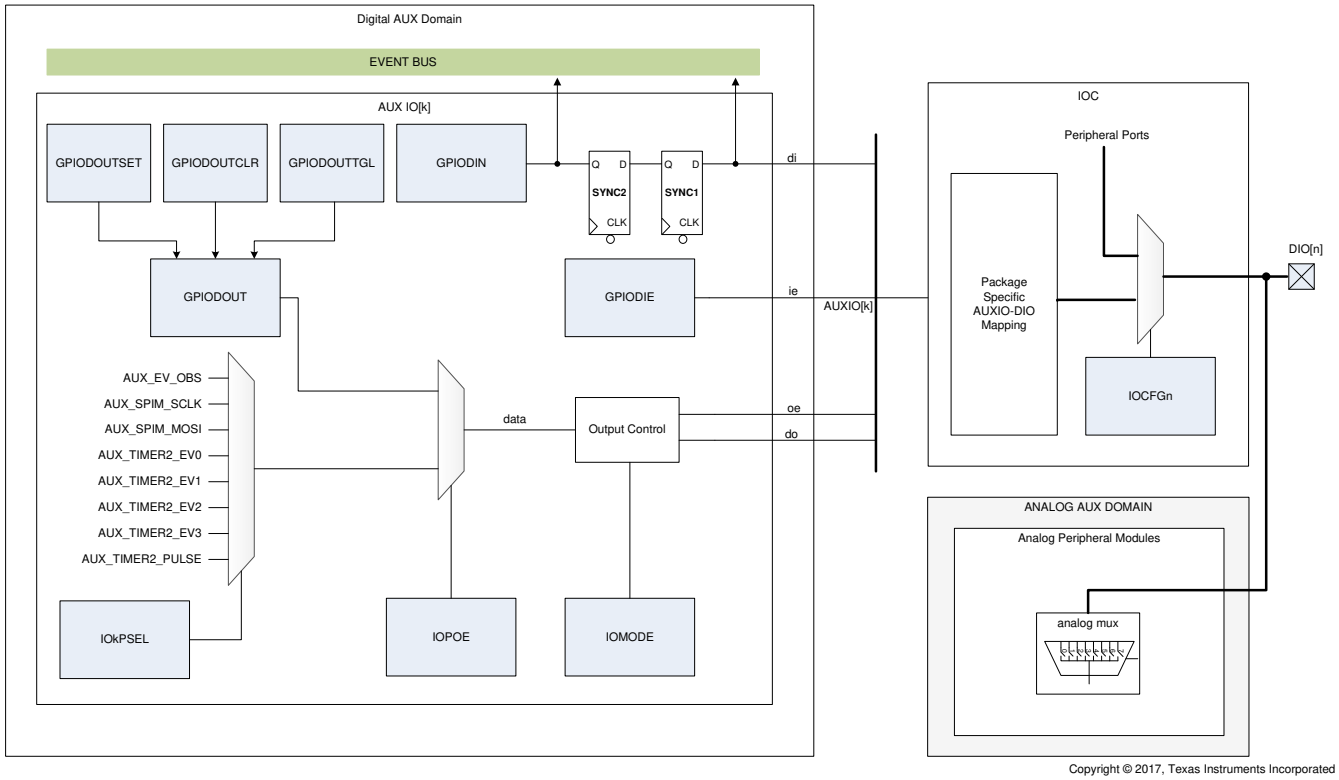


图 19-11. AUX I/O 方框图

19.4.2.2.4 输入缓冲器

要将 AUX I/O 用作数字输入，必须使能数字输入缓冲器。当用户在 AUX_AIODIOn:GPIODIE 中设置相应的位时，该缓冲器被使能。

默认情况下，AUX I/O 以 SCE 时钟速率同步，并且当系统 CPU 写入 AIODIO 实例时也会进行同步。后者必须限制在 AUX 外设或传感器控制器功能之前进行初始化，以获得恒定的同步时钟速率。

最多有 8 个 AUX I/O 支持 DIO 与模拟 AUX 外设之间的模拟传输。要启用模拟传输，请将 I/O 模式设置为*输入*并禁用数字输入缓冲器。要确定哪些 AUX I/O 支持模拟传输，请参见表 13-2。

19.4.2.2.5 数据输出源

每个 AUX I/O 根据 I/O 模式的输出配置传播数据。数据源由 AUX_AIODIOn:IOPOE 控制，具体如下：

- **直接模式：**数据源是内部 AUX_AIODIOn:GPIODOUT 寄存器。
- **外设模式：**数据源由 AUX_AIODIOn:IOnPSEL.SRC 选择。

有关寄存器说明，请参阅节 19.8.2。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.4.3 SMPH

19.4.3.1 简介

AUX 信标 (AUX_SMPH) 外设包含八个硬件信标。系统 CPU 和传感器控制器使用这些信标来实现资源所有权管理。

19.4.3.2 功能说明

硬件信标功能很简单。要保留一个资源（例如访问和使用模块的权限），系统 CPU 应用程序通过读取操作来请求一个特定的信标。读取的值决定了所有权：

- 0 = 传感器控制器拥有该信标及其关联的资源。
- 1 = 系统 CPU 拥有该信标及其关联的资源。

要保留一个资源（例如访问和使用模块的权限），传感器控制器任务通过读取操作来请求一个特定的信标。读取的值决定了所有权：

- 0 = 系统 CPU 拥有该信标及其关联的资源。
- 1 = 传感器控制器拥有该信标及其关联的资源。

只有信标的所有者才能通过向信标写入值 1 来释放它。没有强制执行此规则的硬件机制。

为了避免轮询并节省功耗，系统 CPU 可以通过将 SMPH_ID 写入 AUX_SMPH:AUTOTAKE.SMPH_ID 来请求一个信标。当该信标属于系统 CPU 时，AUX_EVCTL:EVTOMCUFLAGS 中的 AUX_SMPH_AUTOTAKE_DONE 事件将变为 1。传感器控制器不得使用此功能。没有强制执行此规则的硬件机制。

19.4.3.3 TI 软件中的信标分配

表 19-21 总结了 TI 软件如何将资源分配给信标。

根据表 19-21 中列出的资源到信标映射关系，当应用程序需要多个信标时，系统 CPU 和传感器控制器必须按相同的顺序获取信标。因此，如果系统 CPU 和传感器控制器应用程序都需要 COMPA 和参考 DAC 外设，则两个应用程序都必须先请求 SMPH_ID 4，然后再请求 SMPH_ID 3，或者按相反的顺序。

表 19-21. 信标资源分配

SMPH_ID	资源分配
0	TI 未分配
1	TDC
2	ADC + COMPB
3	COMPA + ISRC
4	参考 DAC
5	Timer2
6	TI 未分配
7	TI 未分配

19.4.4 SPIM

19.4.4.1 简介

AUX SPI 主器件 (AUX_SPIM) 外设为传感器控制器提供高能效的 SPI 通信能力。传感器控制器是此外设的主要拥有者，但它也可以与系统 CPU 共享或由系统 CPU 拥有。

SPI 主器件负责 SCLK 信号、MOSI 信号的生成以及 MISO 的采样。SPI 片选 (CS) 信号不受该外设控制；因此，传感器控制器必须单独控制该引脚。

外设特性包括：

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

- 可配置的 SCLK 频率
- 可配置的相位和极性
- 支持 8 位和 16 位传输大小
- SCLK、MOSI 和 MISO 可以连接到所有 AUX I/O
- 无缓冲的 RX 和 TX 移位寄存器

图 19-12 显示了 SPI 主器件的方框图及其与 DIO 的连接方式。

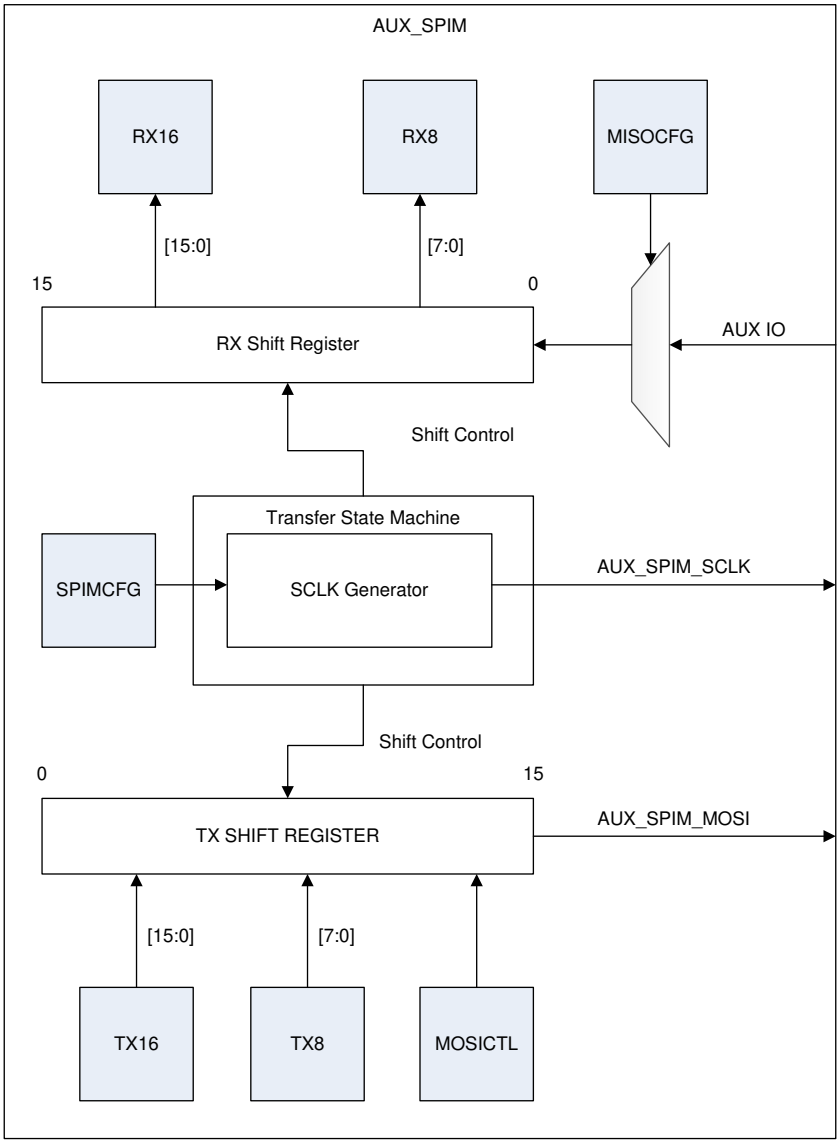


图 19-12. AUX_SPIM 方框图

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.4.4.2 功能说明

19.4.4.2.1 TX 和 RX 操作

写入 AUX_SPIM:TX8 或 AUX_SPIM:TX16 以在 SPI 主器件和从器件之间传输 8 位或 16 位数据。数据首先传播到 I/O 的 MSB。当传输完成时，MOSI 保持在 LSB 的值。

正在进行的 SPI 传输会阻塞寄存器访问：

- 对 AUX_SPIM:SCLKIDLE 的读取访问在 SCLK 空闲时完成。
- 对 AUX_SPIM:DATAIDLE 的读取访问在 LSB 位周期完成时完成。
- 对 AUX_SPIM:RX8 和 AUX_SPIM:RX16 的读取访问在 LSB 被捕获时完成。
- 对任何寄存器的写入访问在 LSB 位周期完成时完成。

当传感器控制器的访问被阻塞时，它会暂停执行，这一特性可用于减少代码大小和节省功耗。

19.4.4.2.2 配置

设置 SPI 主器件和所需的 I/O 映射只需最小配置。

AUX_SPIM:SPIMCFG 寄存器配置以下内容：

- SCLK 预分频器：
  - SCLK 由外设时钟分频得到，由 DIV 字段的值决定。
- MOSI 和 MISO 数据信号的相位：
  - PHA 字段选择何时对 MOSI 进行移位以及何时对 MISO 进行采样。MISO 始终在位周期中间的 SCLK 边沿进行采样。
- SCLK 极性：
  - POL 字段选择 SCLK 的空闲状态。

通过以下方式将外设 SPI 信号连接到 AUX I/O：

- SCLK：
  - 确定连接到 SCLK 的 AUX I/O。
  - 将该 AUX I/O 外设的输出设置为 AUX_SPIM_SCLK。
  - 使能该 AUX I/O 的外设输出。
  - 将 I/O 模式设置为数字输出。
- MOSI：
  - 确定连接到 MOSI 的 AUX I/O。
  - 将该 AUX I/O 外设的输出设置为 AUX_SPIM_MOSI。
  - 使能该 AUX I/O 的外设输出。
  - 将 I/O 模式设置为数字输出。
- MISO：
  - 确定作为 MISO 连接的 AUX I/O。
  - 将 I/O 模式设置为输入。
  - 使能数字输入缓冲器。
  - 将 AUX_SPIM:MISOFCG 配置为所选的 AUX I/O。

片选 (CS) 由主机单独控制。将用作 CS 的 AUX I/O 配置如下：

- 确定用作 CS 的 AUX I/O。
- 禁用该 AUX I/O 的外设输出。
- 将 I/O 模式设置为数字输出。

Sensor Controller Studio 会为用户处理必要的配置。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.4.4.2.3 时序图

图 19-13 和图 19-14 显示了 AUX_SPIM:SPIMCFG 字段的配置如何影响 8 位数据传输以及寄存器访问阻塞的持续时间。为了便于说明，AUX_SPIM:SPIMCFG.DIV 设置为 0x4，这使得 SCLK 周期是外设时钟周期的 10 倍。

在图 19-13 和图 19-14 中，当用户写入 AUX_SPIM:TX8（表示传输开始）时，外设时钟开始切换，阻塞开始。POL 字段的值不影响波形的时序；它只设置 SCLK 的空闲状态。PHA 字段的值决定了 MOSI 和 MISO 信号相对于 SCLK 的相位。如图 19-13 和图 19-14 所示，这也影响了对 AUX_SPIM:SCLKIDLE 寄存器的阻塞持续时间。当 PHA = 1 时，访问完成得更快，因为 SCLK 更早进入空闲状态。

备注

使用 AUX_SYSIF:PEROPRATE.SPIM_OP_RATE 设置外设时钟频率，具体如下：

- 当传感器控制器拥有该外设时，必须将其设置为 SCE_RATE。
- 当系统 CPU 拥有该外设时，必须将其设置为 BUS_RATE。

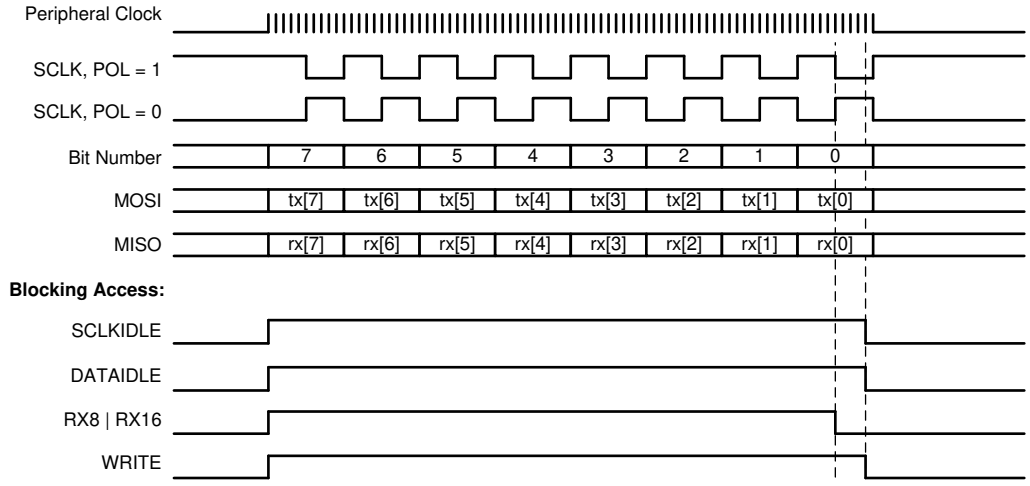


图 19-13. SPI 计时示意图：PHA = 0

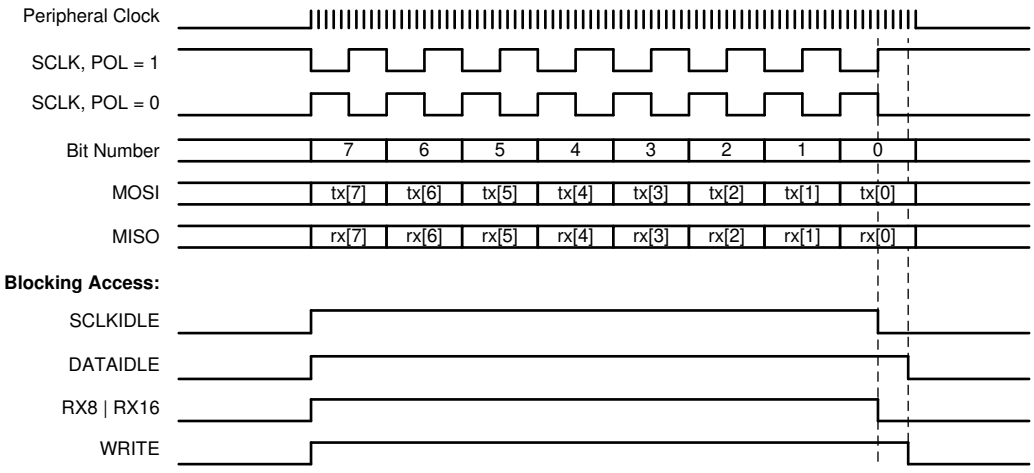


图 19-14. SPI 计时示意图：PHA = 1

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.4.5 时间数字转换器 (TDC)

19.4.5.1 简介

AUX TDC (AUX_TDC) 外设是一个高精度 TDC，用于测量可配置的启动事件和另一个可配置的停止事件之间的时间间隔。TDC 使用所选计数器时钟源的双边沿，以 96MHz 或 48MHz 的频率进行计数。

TDC 外设的特性包括：

- 可配置的计数器时钟源
- 用于片上时钟校准的可配置基准时钟源
- 可配置的启动事件源和极性
- 可配置的停止事件源和极性
- 可选的启动和停止事件预分频：
  - 预分频器也可用于独立的异步脉冲计数。
- 可选的停止事件忽略计数器：
  - 在单次 TDC 转换期间可以测量多个周期。
- 带有可配置饱和限制的 25 位 TDC 转换计数器
- TDC 测量完成事件

TDC 通常用于以下用途：

- 频率测量
- 结合模拟 AUX 外设 *ISRC* 和 *COMPA* 进行电容式感应
- 任意事件测量之间的精确时间间隔
- 使用预分频器进行独立的异步脉冲计数

传感器控制器和系统 CPU 共享 TDC 外设。TI-RTOS 使用 TDC 外设进行 RCOSC 校准。两个主器件都必须实现并遵守资源共享才能使用此外设。相关详细信息，请参阅节 19.4.3.3。

图 19-15 展示了 TDC 的方框图。它还显示了与振荡器控制 (OSCCTL) 模块的连接，该模块需要进行配置才能使用 TDC。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

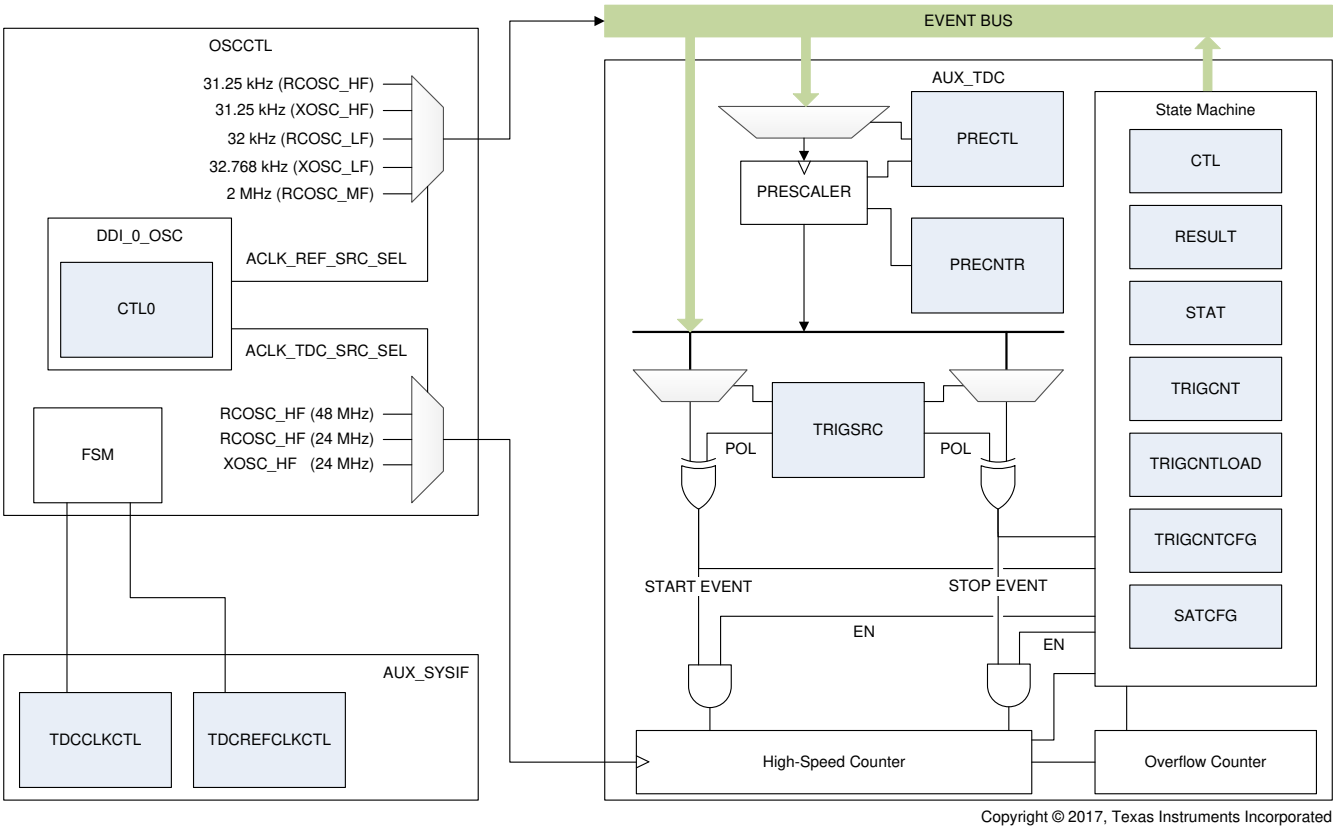


图 19-15. AUX_TDC 方框图

19.4.5.2 功能说明

图 19-15 中所示的 TDC 状态机控制和监视 TDC 转换过程。TDC 状态机具有以下用户接口：

- 命令
- 转换时间配置
- 状态和结果

TDC 状态机基于以下配置执行转换：

- 时钟源选择
- 启动和停止事件选择
- 预分频器配置

节 19.4.5.2.1 至节 19.4.5.2.6 介绍了这些接口和配置。

备注

- 传感器控制器必须请求活动 AUX 工作模式才能使用 TDC。
- 系统 CPU 和传感器控制器必须在 AUX_TDC:STAT.STATE = IDLE 时配置 TDC。允许的例外情况如下：
  - 更新 AUX_TDC:TRIGCNT
  - 中止命令

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.4.5.2.1 命令

TDC 状态机支持以下四种用户命令：

- 同步计数器启动：

  TDC 转换在启动事件边沿发生时开始，这适用于频率测量。
- 异步计数器启动：

  TDC 转换在启动事件电平发生时立即开始。除非用户控制事件源，否则无法保证精确的边沿到边沿测量。用户必须确保在 AUX_TDC:STAT.STATE 小于 0x5 之前（大约需要 420ns）启动事件不会到达。
- 中止：

  强制状态机中止正在进行的 TDC 测量。如果 AUX_TDC:STAT.STATE 等于以下任何值，请勿写入此命令：
  - CLR_CNT
  - WAIT_CLR_CNT_DONE

  不遵守此规则可能会使 TDC 在这些状态下停滞。
- Clear：

  清除结果和状态寄存器。

有关更多说明，请参阅节 19.8.5 中的 AUX_TDC:CTL。

19.4.5.2.2 转换时间配置

用户可将 TDC 状态机配置为：

- 将转换结果饱和至选定值：

  这相当于指定一个超时。有关详细信息，请参阅节 19.8.5 中的 AUX_TDC:SATCFG。
- 在转换期间忽略数量为 AUX_TDC:TRIGCNTLOAD 的停止事件数。

配置 AUX_TDC:TRIGCNTLOAD 并设置 AUX_TDC:TRIGCNTCFG.EN，以便在 TDC 状态机处于 IDLE 状态时使能内部停止计数器。

要在 TDC 转换期间覆盖内部停止计数器的当前值，请写入 AUX_TDC:TRIGCNT。对此寄存器的任何读回操作都必须谨慎处理，因为用户在读取时停止计数器可能会递减。在这种情况下，用户可能读取到的值为 1，而实际上已经没有需要忽略的停止事件了。此时，TDC 测量将忽略对停止计数器的任何更新。为了在 TDC 转换期间安全地更新停止计数器，更新必须在 AUX_TDC:TRIGCNT > 1 时进行。

19.4.5.2.3 状态和结果

AUX_TDC:STAT 寄存器中提供以下状态和转换状态标志：

- 饱和标志
- 完成标志
- TDC 状态机的状态

当转换自行完成或因中止命令结束时，TDC 状态机会更新 AUX_TDC:RESULT。同时，AUX_TDC:STAT.DONE 从低电平转换为高电平。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.4.5.2.4 时钟源选择

19.4.5.2.4.1 计数器时钟

TDC 高速计数器时钟源可通过 DDI_0_OSC: CTL0.ACLK_TDC_SRC_SEL 进行配置，如表 19-22 所列。

表 19-22. TDC 计数器时钟源

ACLK_TDC_SRC_SEL	振荡器	频率 (MHz)
0x0	RCOSC_HF	48
0x1	RCOSC_HF	24
0x2	XOSC_HF	24
0x3	未使用	不适用

DDI_0_OSC:CTL0.ACLK_TDC_SRC_SEL 的配置通过 DDI 控制配置接口完成。有关详细信息，请参阅节 19.4.1.1。

在开始 TDC 测量之前，用户必须先激活所选的计数器时钟源：

- 设置 AUX_SYSIF:TDCCLKCTL.REQ 以请求时钟源。
- 设置 AUX_SYSIF:TDCCLKCTL.ACK 时，时钟源处于活动状态。

备注

TDC 测量完成后，用户必须停用计数器时钟源。否则，由于振荡器控制模块会向供电系统请求资源，系统将无法进入待机模式。要移除时钟源请求，请清除 AUX_SYSIF:TDCCLKCTL.REQ。

19.4.5.2.4.2 基准时钟

配置基准时钟，以测量片上振荡器的频率。基准时钟源可通过 DDI_0_OSC:CTL0.ACLK_REF_SRC_SEL 进行配置（具体如 表 19-23 所示）。

表 19-23. TDC 基准时钟源

ACLK_REF_SRC_SEL	振荡器	频率 (kHz)
0x0	RCOSC_HF	31.25
0x1	XOSC_HF	31.25
0x2	RCOSC_LF	32
0x3	XOSC_LF	32.786
0x4	RCOSC_MF	2000
0x5 – 0x7	未使用	不适用

DDI_0_OSC:CTL0.ACLK_REF_SRC_SEL 的配置通过 DDI 控制-配置接口完成。相关详细信息，请参阅节 19.4.1.1。

在开始 TDC 测量之前，用户必须先启用所选的基准时钟源：

- 将 AUX_SYSIF:TDCREFCLKCTL.REQ 置位以请求基准时钟源。
- 当 AUX_SYSIF:TDCREFCLKCTL.ACK 被置位时，表示该时钟源已处于激活状态。

备注

当 TDC 测量完成后，用户必须取消来自高频振荡器的任何请求。如果不取消，将导致系统无法进入待机模式，因为振荡器控制模块会向供电系统请求资源。清除 AUX_SYSIF:TDCREFCLKCTL.REQ 以删除时钟源请求。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.4.5.2.5 启动和停止事件

TDC 转换的启动事件和停止事件来源于 AUX 事件总线上的事件。如图 19-15 所示，AUX_TDC:TRIGSRC 寄存器为此分别选择两个事件及其对应的极性。根据用户命令和配置，这两个事件都由 TDC 状态机选择性地进行门控。

19.4.5.2.6 预分频器

图 19-15 中所示的预分频器用于对高频信号进行分频，以满足节 19.4.5.3 中详述的时序要求。预分频器对来自事件总线的事件进行分频，以生成一个名为 *AUX_TDC_PRE* 的低频事件。分频比可以是 16 或 64，生成的事件波形具有 50% 的占空比。用户必须选择 AUX_TDC_PRE 作为 TDC 转换的启动和停止事件。

按照以下顺序初始化预分频器：

1. 通过将 AUX_TDC:PRECTL.RESET_N 设置为 0 来复位预分频器。这将复位内部计数器并清除 AUX_TDC_PRE 事件。
2. 配置预分频器事件源 AUX_TDC:PRECTL.SRC。
3. 配置分频比 AUX_TDC:PRECTL.RATIO。

然后，通过将 AUX_TDC:PRECTL.RESET_N 设置为 1 来使能预分频器事件生成。

当预分频器用于事件预分频时，存在以下限制：

- 唯一支持的 TDC 命令是同步计数器启动。
- 预分频器输入频率必须小于 24MHz。
- TDC 转换结果不考虑事件预分频，软件必须适当地缩放 TDC 结果。

19.4.5.3 支持的测量类型

TDC 允许用户测量两个任意选定事件之间的时间。本节 (节 19.4.5.3.1 至节 19.4.5.3.4) 指出了最重要的测量类型，并列出了相关的事件时序要求。必须确保满足这些要求，以保证测量的正确性和可重复性。

19.4.5.3.1 测量脉冲宽度

测量事件源波形的高电平相位或低电平相位的时间需要进行如下配置。请使用以下顺序：

1. 将 AUX_TDC:TRIGSRC.START_SRC 设置为等于 AUX_TDC:TRIGSRC.STOP_SRC。
2. 决定要测量的事件源相位：
   - 高电平相位：
     - 设置 AUX_TDC:TRIGSRC.START_POL = 0。
     - 设置 AUX_TDC:TRIGSRC.STOP_POL = 1。
   - 低电平相位：
     - 设置 AUX_TDC:TRIGSRC.START_POL = 1。
     - 设置 AUX_TDC:TRIGSRC.STOP_POL = 0。
3. 设置 AUX_TDC:TRIGCNTLOAD = 0 以将停止计数器配置为忽略零个停止事件。
4. 设置 AUX_TDC:TRIGCNTCFG.EN = 1 以使能停止计数器。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

图 19-16 展示了一个通用的事件源波形。

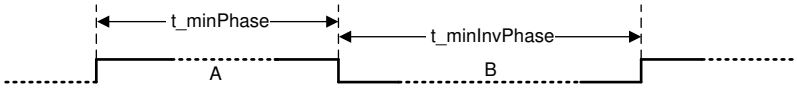


图 19-16. 相位宽度时序要求

在图 19-16 中，要测量的相位标记为 A，并任意设置为事件源的高电平相位。不测量的相反相位标记为 B。表 19-24 列出了这两个相位的时序要求。

表 19-24. 相位宽度时序要求

说明	标签	要求	条件
最小相位	t_minPhase	292ns	DDI_0_OSC:CTL0.ACLK_TDC_SRC_SEL = 0x0
最小相位	t_minPhase	230ns	DDI_0_OSC:CTL0.ACLK_TDC_SRC_SEL = (0x1 | 0x2)
最小反向相位	t_minInvPhase	126ns	AUX_TDC:CTL.CMD = 0x1
最小反向相位	t_minInvPhase	42ns	AUX_TDC:CTL.CMD = 0x2

如果 AUX_TDC:CTL.CMD = 0x2，则在 AUX_TDC:STAT.STATE < 5 之前，事件源电平必须等于反向相位的电平。如果不满足此要求，TDC 转换将启动得过晚。

19.4.5.3.2 测量频率

测量事件源波形频率需要进行如下配置。请使用以下顺序：

- 将 AUX_TDC:TRIGSRC.START_SRC 设置为等于 AUX_TDC:TRIGSRC.STOP_SRC。
- 将 AUX_TDC:TRIGSRC.START_POL 设置为等于 AUX_TDC:TRIGSRC.STOP_POL。
- 将 AUX_TDC:TRIGCNTLOAD 设置为 N 以允许 TDC 转换跨越 N 个信号周期。
- 设置 AUX_TDC:TRIGCNTCFG.EN = 1 以使能停止计数器。
- 设置 AUX_TDC:CTL.CMD = 1 以启动 TDC 转换。

图 19-17 显示了一个通用的事件源波形。

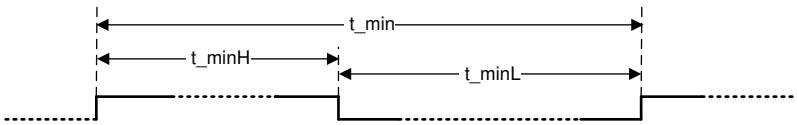


图 19-17. 频率测量波形

如果所选事件源波形满足表 19-25 中列出的要求，TDC 将测量正确的信号频率。

表 19-25. 频率测量的时序要求

说明	标签	要求
最短低电平时间	t_minL	208ns
最短高电平时间	t_minH	208ns
最小周期	t_min	416ns

如果事件波形不能直接满足这些要求，用户必须完成以下顺序：

1. 将 AUX_TDC:TRIGSRC.START_SRC 设置为 AUX_TDC_PRE。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2. 初始化并使能预分频器。

有关更多详细信息，请参阅节 19.4.5.2.6。

19.4.5.3.3 测量不同事件源之间的边沿时间间隔

要测量由不同事件源产生的事件之间的时间间隔，请按以下顺序完成以下操作：

1. 将 AUX_TDC：TRIGSRC.START_SRC 设置为不等于 AUX_TDC：TRIGSRC.STOP_SRC。
2. 指定 AUX_TDC:TRIGSRC.START_POL 和 AUX_TDC:TRIGSRC.STOP_POL。

以下小节介绍了执行此类测量的四种方法。

19.4.5.3.3.1 异步计数器启动 — 忽略 0 个停止事件

在此场景中，用户：

- 希望测量启动事件与第一个停止事件之间的时间
- 设置 AUX_TDC:TRIGCNTCFG.EN = 0 以禁用停止计数器
- 通过写入 AUX_TDC:CTL.CMD = 2 开始 TDC 测量
- 确保启动事件为低电平，直到 AUX_TDC.STAT.STATE = WAIT_START
- 确保停止事件变为高电平的时间晚于启动事件

图 19-18 显示了异步计数器启动 — 忽略 0 个停止事件的场景。

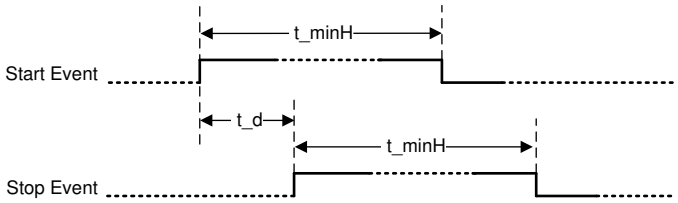


图 19-18. 任意时间测量 1

表 19-26 列出了此场景的相关时序要求。

表 19-26. 任意时间测量 1

说明	标签	要求
最短高电平时间	t_minH	42ns
最短延迟时间	t_d	21ns

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.4.5.3.3.2 同步计数器启动 — 忽略 0 个停止事件

在此场景中，用户：

- 希望测量跟随下降沿之后的启动事件与第一个停止事件之间的时间
- 设置 AUX_TDC:TRIGCNTCFG.EN = 0 以禁用停止计数器
- 通过写入 AUX_TDC:CTL.CMD = 1 开始 TDC 测量
- 确保停止事件变为高电平的时间晚于启动事件

图 19-19 显示了同步计数器启动 — 忽略 0 个停止事件的场景。

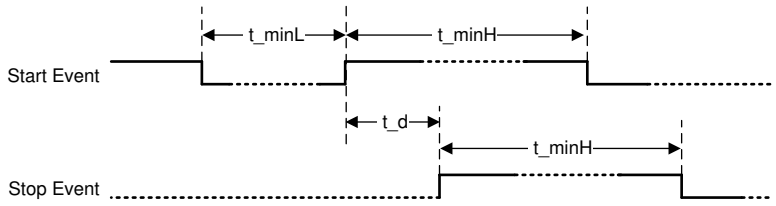


图 19-19. 任意时间测量 2

表 19-27 列出了此场景的相关时序要求。

表 19-27. 任意时间测量 2

说明	标签	要求
最短高电平时间	t_minH	42ns
最短低电平时间	t_minL	126ns
最短延迟时间	t_d	21ns

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.4.5.3.3.3 异步计数器启动 — 忽略停止事件

在此场景中，用户：

- 希望测量启动事件与第 N 个停止事件之间的时间，其中 N > 0：
  - 将 AUX_TDC:TRIGCNTLOAD 设置为 N 减 1。
  - 设置 AUX_TDC:TRIGCNTCFG.EN = 1 以使能停止计数器。
- 通过写入 AUX_TDC:CTL.CMD = 2 开始 TDC 测量。
- 确保启动事件为低电平，直到 AUX_TDC.STAT.STATE = WAIT_START。
- 确保停止事件变为高电平的时间晚于启动事件。

图 19-20 显示了 N = 2 时的场景，此时忽略了 1 个停止事件。

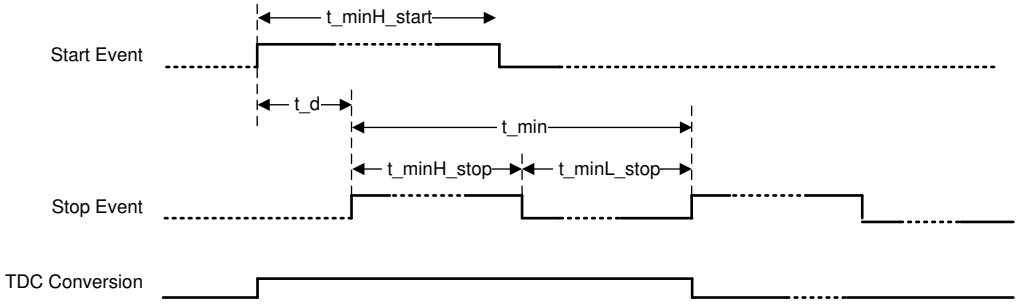


图 19-20. 任意时间测量 3

表 19-28 列出了此场景的相关时序要求。

表 19-28. 任意时间测量 3

说明	标签	要求
最小高电平时间，启动	t_minH_start	42ns
停止事件最小高电平时间	t_minH_stop	42ns
最小低电平时间，停止	t_minL_Stop	168ns
最小停止事件周期	t_min	210ns
最小延迟 | TRIGCNTLOAD > 0	t_d	168ns
最小延迟 | TRIGCNTLOAD = 0	t_d	292ns

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.4.5.3.3.4 同步计数器启动 — 忽略停止事件

在此场景中，用户：

- 希望测量跟随下降沿之后的启动事件与第 N 个停止事件之间的时间，其中 N > 0。
  - 将 AUX_TDC:TRIGCNTLOAD 设置为 N 减 1。
  - 设置 AUX_TDC:TRIGCNTCFG.EN = 1 以使能停止计数器。
- 通过写入 AUX_TDC:CTL.CMD = 1 开始 TDC 测量。
- 确保停止事件变为高电平的时间晚于启动事件。

图 19-21 显示了 N = 2 时的场景，此时忽略了 1 个停止事件。

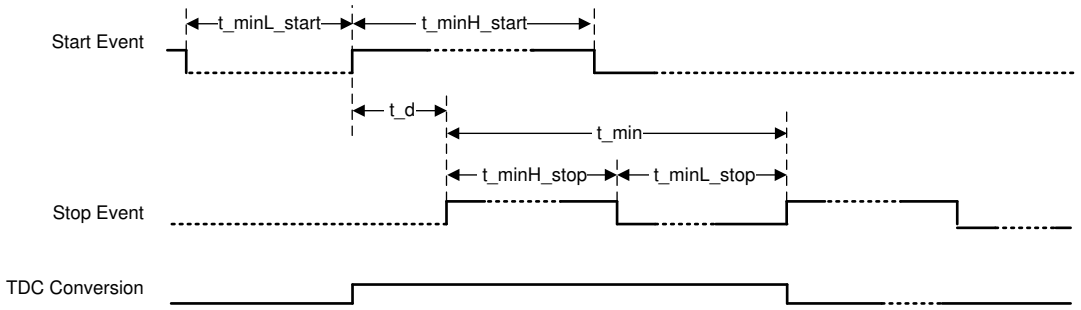


图 19-21. 任意时间测量 4

表 19-29 列出了此场景的相关时序要求。

表 19-29. 任意时间测量 4

说明	标签	要求
最小高电平时间，启动	t_minH_start	42ns
最小低电平时间，启动	t_minL_start	126ns
最小高电平时间，停止	t_minH_stop	42ns
最小低电平时间，停止	t_minL_Stop	168ns
最小停止事件周期	t_min	210ns
最小延迟 | TRIGCNTLOAD > 0	t_d	168ns
最小延迟 | TRIGCNTLOAD = 0	t_d	292ns

19.4.5.3.4 脉冲计数

当预分频器不用作 TDC 启动事件和停止事件的事件预分频器时，它可以用作脉冲计数器。

若要初始化预分频器以进行脉冲计数，请使用以下顺序：

1. 将 AUX_TDC:PRECTL.RESET_N 设置为 0 以复位预分频器。这将复位内部计数器并清除 AUX_TDC_PRE 事件。但是，在计数脉冲时不使用 AUX_TDC_PRE 事件。
2. 配置 AUX_TDC:PRECTL.SRC 以设置预分频器事件源。

将 AUX_TDC：PRECTL.RESET_N 设置为 1 以使能预分频器进行脉冲计数。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

预分频器现在需要三个或四个事件脉冲才能开始计数。所需脉冲的数量取决于复位释放与事件脉冲到达之间的时序，如图 19-22 所示。

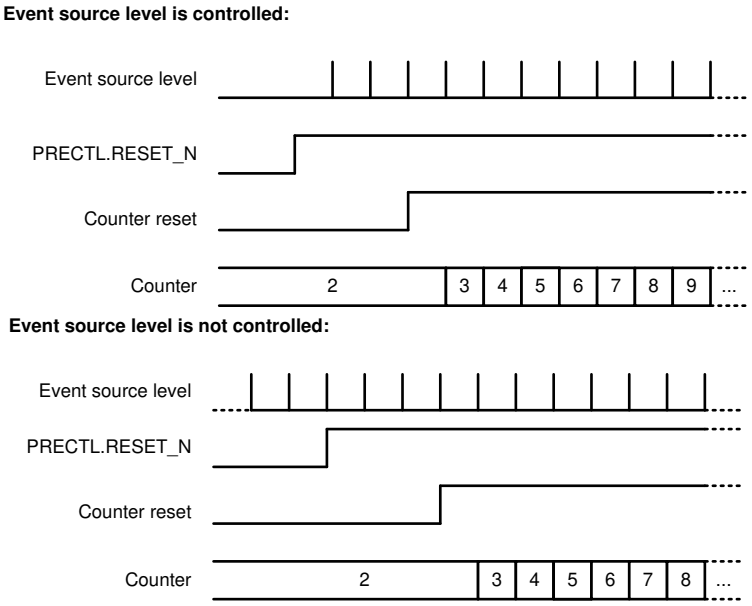


图 19-22. 脉冲计数

如果事件脉冲在复位释放时处于非活动状态，则计数器在第四个事件脉冲时开始递增。因此，计数器值如下：

- 如果事件脉冲数小于 3，则计数器等于 2。
- 否则，计数器等于事件脉冲数减 1。

如果用户在复位释放时无法控制事件的电平，情况则有所不同。在这种情况下，计数器在开始递增之前可能需要再多一个事件脉冲。因此，计数器值可能表现为：

- 如果事件脉冲数小于 4，则计数器等于 2。
- 否则，计数器等于事件脉冲数减 2。

由于这种不确定性，建议在复位释放时控制所选事件的电平。

有关如何读取预分频计数器值的信息，请参阅节 19.8.5 中的 AUX_TDC:PRECNTR。

19.4.6 Timer01

19.4.6.1 简介

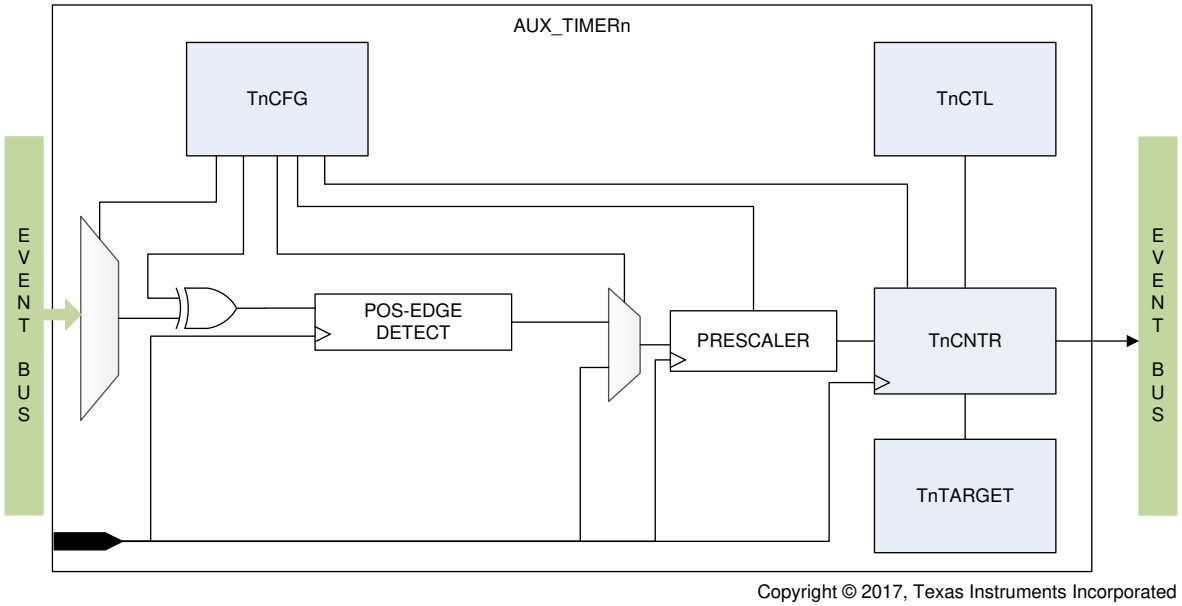
AUX Timer0 和 Timer1 (AUX_TIMER01) 外设是两个相同的 16 位比较计时器。传感器控制器是这些外设的主要拥有者。但是，Timer01 的所有权可以与系统 CPU 共享或由系统 CPU 拥有。

Timer01 外设的特性包括：

- 可配置的计数器目标值
- 可选的时钟源预分频
- 可配置的预分频器时钟源：
  - 外设时钟
  - 来自 AUX 域事件总线的可配置事件
- 单次触发和连续计数器模式：
  - 单次或周期性事件生成

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

这些计时器的主要用途是为 AUX 域事件总线订阅者生成事件。计时器不直接用于波形生成。图 19-23 显示了单个计时器的方框图。



Copyright © 2017, Texas Instruments Incorporated

图 19-23. AUX_TIMER01 方框图

19.4.6.2 功能说明

计时器在每个计数器周期结束时生成一个事件 (AUX_TIMERn_EV)。计数器周期由以下因素决定：

- 计数器时钟源选择：
  - AUX_TIMER01:TnCFG.MODE 选择外设时钟或可配置事件作为计数器时钟源。因此，它设定了计数器时钟源的基频。
- 计数器时钟源预分频：
  - AUX_TIMER01:TnCFG.PRE 对计数器时钟源基频进行分频，得到计数器频率。默认为除以 1。
- 计数器目标值：
  - 只要 AUX_TIMER01:TnCNTR 小于 AUX_TIMER01:TnTARGET，计数器就会在每个计数器周期递增。

当 AUX_TIMER01:TnCNTR 等于或大于 AUX_TIMER01:TnTARGET 时，计数器周期结束。AUX_TIMER01:TnCFG.RELOAD 的值决定计数器是重新启动还是进入空闲状态。

备注

外设时钟频率由 AUX_SYSIF:PEROPRATE.TIMER01_OP_RATE 设置。
- 当传感器控制器拥有该外设时，必须将其设置为 SCE_RATE。
- 当系统 CPU 拥有该外设时，必须将其设置为 BUS_RATE。

19.4.7 Timer2

19.4.7.1 简介

AUX Timer2 (AUX_TIMER2) 外设是单个异步 16 位捕获/比较计时器。传感器控制器是此外设的主要拥有者，但它也可以与系统 CPU 共享或由系统 CPU 拥有。

外设特性包括：

- 欧计时器时钟独立于 AUX 工作模式
- 计时器时钟源预分频器

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

  - 计数器时钟频率等于分频后的计时器时钟源频率
- 可配置的计数器目标值：
  - 静态值
  - 直接更新
  - 影子目标更新
- 单次触发和连续计数器模式
- 四个事件输出：
  - 手动控制
  - 通道控制
- 四个捕获比较通道：
  - 15 种不同的通道功能提供灵活的事件生成：
    - 简单的捕获时事件
    - 简单的比较时事件
    - PWM
    - 周期脉冲宽度测量
  - 通道功能可以是单次触发型或连续型
  - 通道事件可以路由到任意事件输出
  - 流水线比较寄存器
- 时钟脉冲传播
  - 将单个时钟脉冲传播到 AUX I/O

图 19-24 显示了具有单个捕获/比较通道的计时器方框图。

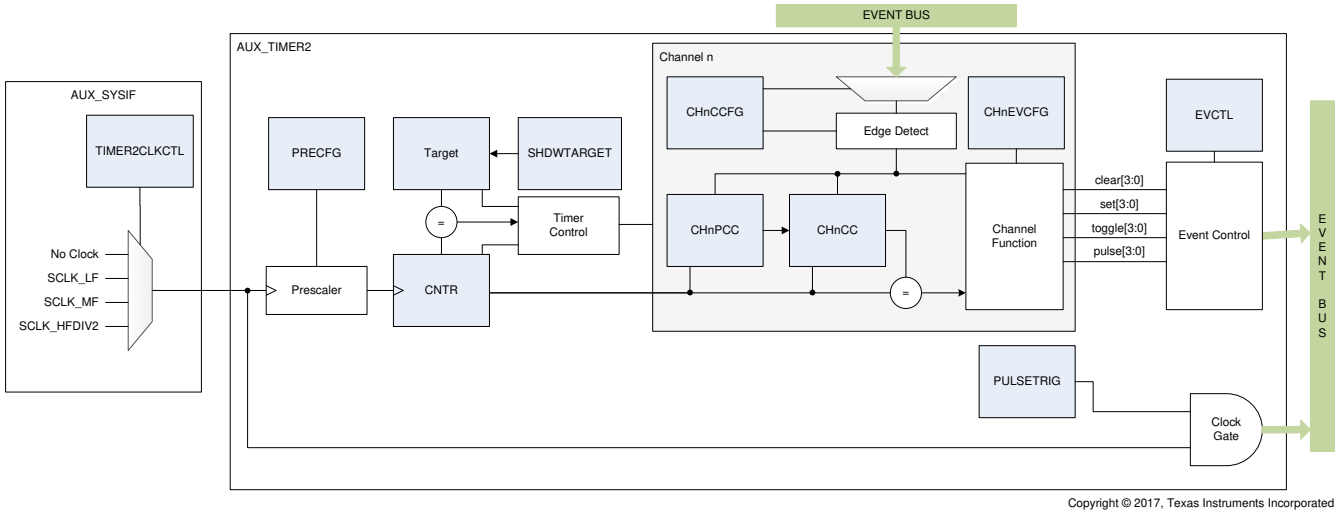


图 19-24. AUX_TIMER2 方框图

19.4.7.2 功能说明

19.4.7.2.1 时钟源

Timer2 时钟独立于 AUX 工作模式运行。Timer2 时钟有四种可能的来源：

- NO_CLOCK (默认)
- SCLK_LF
- SCLK_MF
- SCLK_HFDIV2 – SCLK_HF/2

图 19-24 显示时钟源配置位于 AUX_SYSIF 内，该模块处理 AUX 域时钟和电源管理。仅当选择了时钟时，才能访问 Timer2。要选择时钟，请使用以下顺序：

1. 等待直到 AUX_SYSIF:TIMER2CLKSWITCH.RDY = 1。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

2. 将 AUX_SYSIF:TIMER2CLKCTL.SRC 设置为适当的值。
3. 等待直到 AUX_SYSIF:TIMER2CLKSWITCH.RDY = 1。

   时钟切换完成，AUX Timer2 接收到所选时钟。

要选择 SCLK_HFDIV2，必须将 AUX 工作模式设置为活动。仅当 AUX_SYSIF:TIMER2CLKSWITCH.RDY = 1 时才更改工作模式。切换到其他时钟源时没有类似的要求。

19.4.7.2.2 时钟预分频器

预分频器选择性地对 Timer2 时钟进行分频。分频后的时钟决定：

- 计数器时钟
- 通道事件同步时钟
- 事件控制时钟
  - 通道在此时钟上更新事件输出。

AUX_TIMER2:PRECFG.CLKDIV 设置分频值。分频范围为 1 至 256。寄存器在 Timer2 时钟下进行访问。

19.4.7.2.3 计数器

写入 AUX_TIMER2:CTL.MODE 的值决定计数器模式，如下所示：

- UP_ONCE：计数器从 0 计数到选定的目标值。然后计时器变被禁用。
- UP_PER：计数器从 0 计数到选定的目标值，并重复进行。
- UPDWN_PER：计数器从 0 计数到选定的目标值，然后递减回 0，并重复进行。

AUX_TIMER2:CTL.TARGET_EN 选择以下之一：

- 静态计数器目标值 65535
- AUX_TIMER2:TARGET 给出的用户可配置目标值

目标值必须是用户可配置的，以允许通过 AUX_TIMER2:SHDWTARGET 寄存器进行影子更新。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.4.7.2.4 事件输出

Timer2 有四个事件输出：

- AUX_TIMER2_EV0 连接到 AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV0
- AUX_TIMER2_EV1 连接到 AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV1
- AUX_TIMER2_EV2 连接到 AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV2
- AUX_TIMER2_EV3 连接到 AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV3

用户可以通过写入 AUX_TIMER2:EVCTL 来手动设置和清除事件。手动更新事件输出优先于对同一事件的自动通道更新。按优先级降序排列，每个事件输出可以：

1. 清除
2. 设置
3. 切换
4. Pulse

事件输出在高电平持续两个计数器时钟周期后变为低电平。

一个事件输出可能同时接收到来自多个通道的更新请求。在这种情况下，事件输出根据前面所述的优先级列表进行更新。

19.4.7.2.5 通道操作

每个通道可实现 15 种不同的通道操作。它们分为单次型和连续型：

- 单次型通道操作仅执行一次功能，然后计时器禁用该通道。
- 连续型通道操作会一直执行功能，直到用户禁用该通道。

表 19-30 列出了这 15 种通道操作。

表 19-30. 通道操作

索引	通道操作	类别
0	禁用通道	单次触发
1	捕获时设置，然后禁用通道	单次触发
2	归零时清除，比较时翻转，然后禁用通道	单次触发
3	归零时设置，比较时翻转，然后禁用通道	单次触发
4	比较时清除，然后禁用通道	单次触发
5	比较时设置，然后禁用通道	单次触发
6	比较时翻转，然后禁用通道	单次触发
7	比较时产生脉冲，然后禁用通道	单次触发
8	周期和脉冲宽度测量	持续
9	捕获时重复设置	持续
10	归零时清除，比较时重复翻转	持续
11	归零时设置，比较时重复翻转	持续
12	比较时重复清除	持续
13	比较时重复设置	持续
14	比较时重复翻转	持续
15	比较时重复产生脉冲	持续

配置完成后，通道会根据表 19-30 中的通道操作说明，请求更新已使能的事件输出。有三种通道操作需要进一步说明。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.4.7.2.5.1 周期和脉冲宽度测量

此通道操作连续捕获由 CHnCCFG.CAPT_SRC 选择的信号相对于 CHnCCFG.EDGE 给出的信号边沿的周期和脉冲宽度。当 CHnCC.VALUE 包含信号周期且 CHnPCC.VALUE 包含信号脉冲宽度时，该通道请求置位已使能的事件。

该通道功能将定时器计数器同步到输入信号的所选信号边沿。因此：

- 计数器会定期重新启动，因此在选择其他通道操作时必须考虑到这一点。
- 配置为此通道操作的通道无法同时进行测量。测量是以时间交错的方式进行的。

有关更多说明，请参见节 19.8.7 中的 AUX_TIMER2:CHnEVCFG.CCACT。

示例 19-1. 计时器周期和脉冲宽度捕获

计时器测量两个不同信号 *A* 和 *B* 的信号周期和脉冲宽度。在此示例中，假设两个信号的周期都小于计数器范围。因此，不需要寄存器文档中所述的超时检测。配置如下：

- 通道 0:
  1. AUX_TIMER2:CH0EVCFG. CCACT = PER_PULSE_WIDTH_MEAS
  2. AUX_TIMER2:CH0EVCFG.EV0_GEN = 1
  3. AUX_TIMER2:CH0CCFG.CAPT_SRC = Signal A
  4. AUX_TIMER2:CH0CCFG.EDGE = RISING
- 通道 1:
  1. AUX_TIMER2:CH1EVCFG.CCACT = PER_PULSE_WIDTH_MEAS
  2. AUX_TIMER2:CH1EVCFG.EV1_GEN = 1
  3. AUX_TIMER2:CH1CCFG.CAPT_SRC = Signal B
  4. AUX_TIMER2:CH1CCFG.EDGE = FALLING
- 计时器：
  1. CTL.MODE = UP_PER

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

图 19-25 显示了计时器计数器首先如何同步到信号 A。然后通道 0 在时间 t_0 将信号 A 的高电平相位捕获到 CH0PCC 中。最后，在时间 t_1 将信号 A 的周期捕获到 CH0CC 中。同时，通道 0 将事件输出 0 设置为高电平，计时器计数器开始同步到信号 B。然后通道 1 在时间 t_2 将信号 B 的低电平相位捕获到 CH1PCC 中。最后，在时间 t_3 将信号 B 的周期捕获到 CH1CC 中。同时，通道 1 将事件输出 1 设置为高电平，计时器计数器开始同步到信号 A。然后该序列重复执行，直到被用户停止。

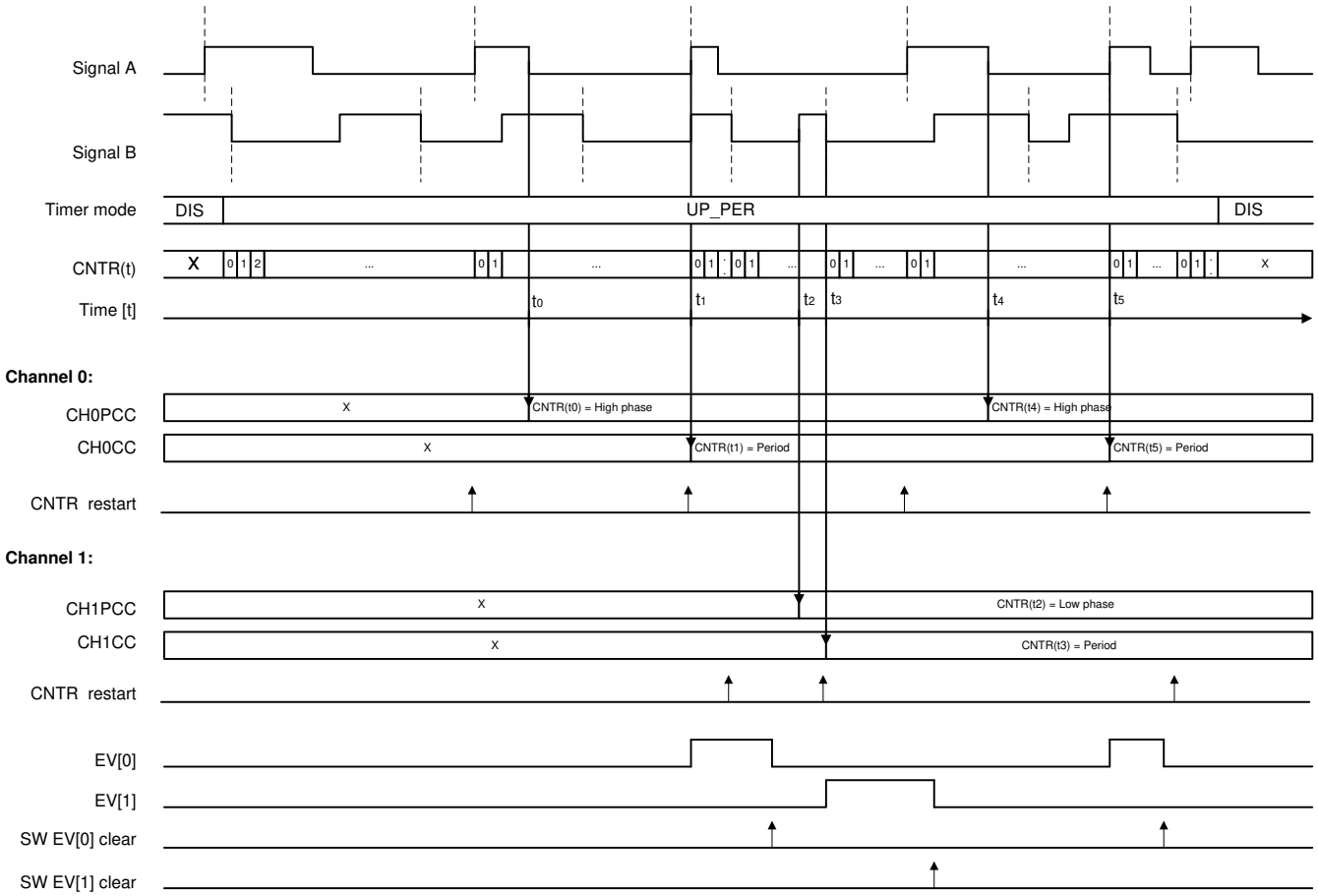


图 19-25. 周期脉冲宽度测量

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.4.7.2.5.2 归零时清除，比较时重复翻转

此通道操作会持续执行以下动作：

- 当 AUX_TIMER2:CNTR = 0 时，清除已使能的事件输出
- 当 AUX_TIMER2:CNTR = CHnCC 时，翻转已使能的事件输出

当 AUX_TIMER2:CTL.MODE = UPDWN_PER 时，该通道生成中心对齐的 PWM 波形。

当 AUX_TIMER2:CNTR 变为 0 时，该通道会将写入 AUX_TIMER2:CHnPCC 的新值复制到 AUX_TIMER2:CHnCC。此操作可防止生成的 PWM 信号边沿出现抖动。类似地，当 AUX_TIMER2:CNTR 变为 0 时，计时器会将写入 AUX_TIMER2:SHDWTARGET 的新值复制到 AUX_TIMER2:TARGET。此操作可避免周期随时间变化的 PWM 应用中出现周期抖动。

有关更多说明，请参见节 19.8.7 中的 AUX_TIMER2:CHnEVCFG.CCACT。

示例 19-2. 通道 0 生成中心对齐 PWM

此示例演示了通道 0 生成中心对齐的 PWM（请参见图 19-26）。该波形在事件输出 0 上合成。计时器周期保持静态，目标值设置为周期的一半。配置如下：

- 通道 0:
  1. AUX_TIMER2:CH0EVCFG.CCACT = CLR_ON_0_TGL_ON_CMP
  2. AUX_TIMER2:CH0EVCFG.EV0_GEN = 1
  3. AUX_TIMER2:CH0CC = C0
- 计时器：
  1. TARGET = PERIOD / 2
  2. CTL.MODE = UPDWN_PER

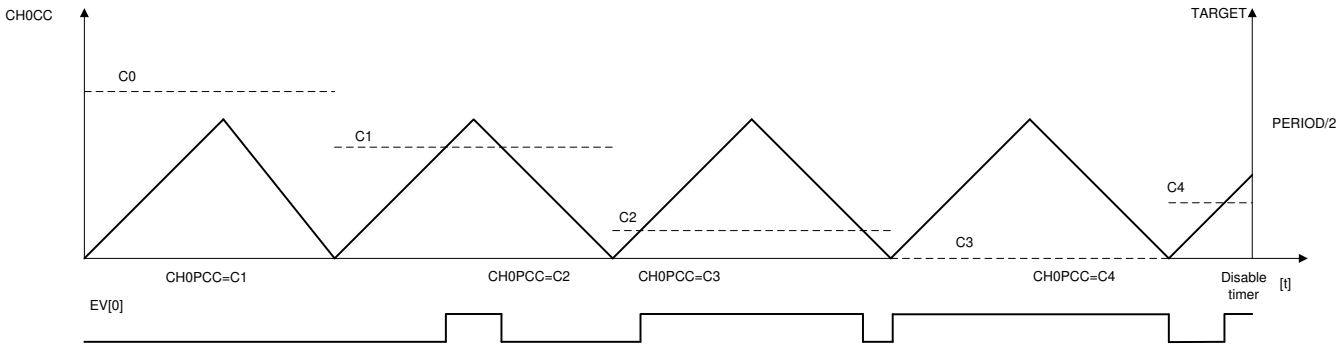


图 19-26. 中心对齐 PWM

生成的 PWM 波形的占空比通过更新 AUX_TIMER2:CH0PCC 来控制。事件输出 0 上的波形生成会持续进行，直到被用户中止。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.4.7.2.5.3 归零时设置，比较时重复翻转

此通道操作会持续执行以下动作：

- 当 AUX_TIMER2:CNTR = 0 时，设置已使能的事件输出
- 当 AUX_TIMER2:CNTR = CHnCC 时，翻转已使能的事件输出

当 AUX_TIMER2:CTL.MODE = UP_PER 时，该通道生成边沿对齐的 PWM 波形。

当 AUX_TIMER2:CNTR 变为 0 时，该通道会将写入 AUX_TIMER2:CHnPCC 的新值复制到 AUX_TIMER2:CHnCC。这可防止生成的 PWM 信号边沿出现抖动。类似地，当 AUX_TIMER2:CNTR 变为 0 时，计时器会将写入 AUX_TIMER2:SHDWTARGET 的新值复制到 AUX_TIMER2:TARGET。这可避免周期随时间变化的 PWM 应用中出现周期抖动。

有关更多说明，请参见节 19.8.7 中的 AUX_TIMER2:CHnEVCFG.CCACT。

示例 19-3. 通道 0 生成边沿对齐 PWM

此示例演示了通道 0 生成边沿对齐的 PWM（请参见图 19-27）。该波形在事件输出 0 上合成。计时器周期保持静态，目标值设置为周期减 1。配置如下：

- 通道 0:
  1. AUX_TIMER2:CH0EVCFG.CCACT = SET_ON_0_TGL_ON_CMP
  2. AUX_TIMER2:CH0EVCFG.EV0_GEN = 1
  3. AUX_TIMER2:CH0CC = C0
- 计时器：
  1. TARGET = PERIOD - 1
  2. CTL.MODE = UP_PER

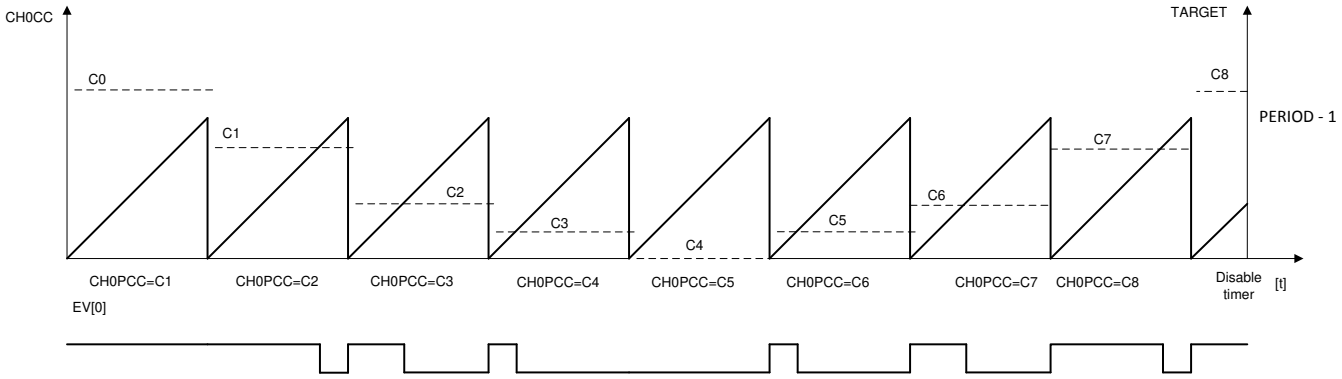


图 19-27. 边沿对齐的 PWM

生成的 PWM 波形的占空比通过更新 AUX_TIMER2:CH0PCC 来控制。事件输出 0 上的波形生成会持续进行，直到被用户中止。

19.4.7.2.6 异步总线桥

AUX Timer2 与 AUX 时钟异步。它通过一个异步总线桥进行访问。主端由 AUX 总线时钟提供时钟，从端由 Timer2 时钟提供时钟。该桥具有单个写入缓冲区。正在进行的总线传输会暂停新的访问，直到前一次传输完成。

为尽量减少访问延迟，TI 建议在配置期间将 Timer2 时钟频率设置为等于或高于 AUX 时钟频率。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.5 模拟外设模块

19.5.1 概述

表 19-31 列出了 AUX 域中的模拟外设。AUX 模拟接口 (AUX_ANAIF) 保存与 ADC、ISRC 和 DAC 外设的数字控制和数据接口。表 19-31 中列出了这些接口的操作时钟速率。有关操作速率的说明，请参阅节 19.2。COMPA 和 COMPB 模块完全通过 AUX_ADI4 进行控制和配置。

AUX 域中的模拟外设可在传感器控制器和系统 CPU 之间共享。如果需要共享，建议使用 AUX 中的硬件信标来仲裁对外设的访问。出于同样的原因，建议系统 CPU 和传感器控制器在使用后将外设保持在复位状态。有关 TI 软件如何将资源分配给信标的信息，请参见表 19-21。

表 19-31. 模拟外设

模拟外设	直接数字接口	操作时钟速率	
		总线	SCE
ADC	AUX_ANAIF:ADCx	X	
ISRC	AUX_ANAIF:ISRCCTL	X	
DAC	AUX_ANAIF:DACx	(X)	X
COMPA	-	-	
COMPB	-	-	
AUX_ADI4	-	X	

19.5.1.1 ADI 控制配置

AUX_ADI4（在表 19-31 中列出）实际上并不是一个数字外设。如图 19-1 所示，系统 CPU 和传感器控制器都通过 AUX_ADI4 模块访问 ADI_4_AUX，以控制和配置模拟 AUX 外设。AUX_ADI4 模块支持四种不同类型的访问以提高效率，因为它通过多周期总线接口连接到模拟域。访问类型包括：

- 直接
- 设置
- 清除
- 已屏蔽

当总线时钟速率高于 SCE 时钟速率时，传感器控制器检测到的访问延迟更短。

如果系统 CPU 应用程序需要访问此模块，则必须使用 TI 提供的 DriverLib API 函数。Sensor Controller Studio API 函数在必要时包含访问，用户在开发代码时无需考虑此模块。

19.5.1.2 方框图

图 19-28 显示了 AUX 模拟方框图。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

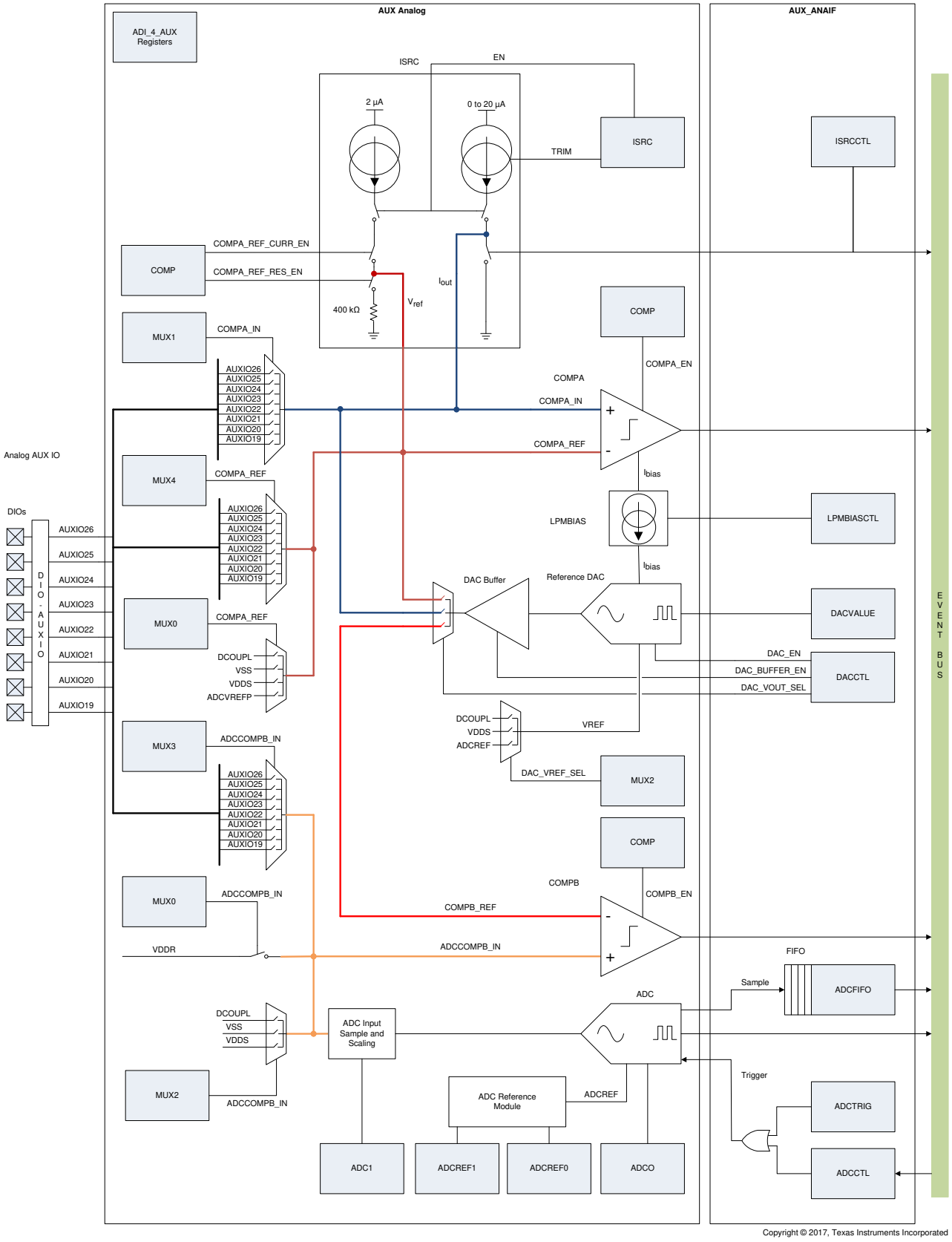


图 19-28. AUX 模拟方框图

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.5.2 模数转换器 (ADC)

19.5.2.1 简介

AUX ADC 外设具有以下特性：

- 12 位通用逐次逼近型 (SAR) ADC
- 对具有模拟功能的 AUX I/O 引脚或内部芯片电压进行采样
- 采样率高达 200ksps
- 手动或事件驱动的 ADC 触发器可根据 ADC 工作模式启动 ADC 采样或转换：
  - ADC 触发器先启动采样，然后进行转换
  - 连续采样时，ADC 触发器启动转换
- 可配置的采样时间
- 固定或相对 ADC 基准
- 具有事件接口的四元件 ADC 采样 FIFO

图 19-29 显示了 ADC 如何连接到模拟节点以及 AUX_ANAIF 和 ADI_4_AUX 寄存器提供的数字控制。

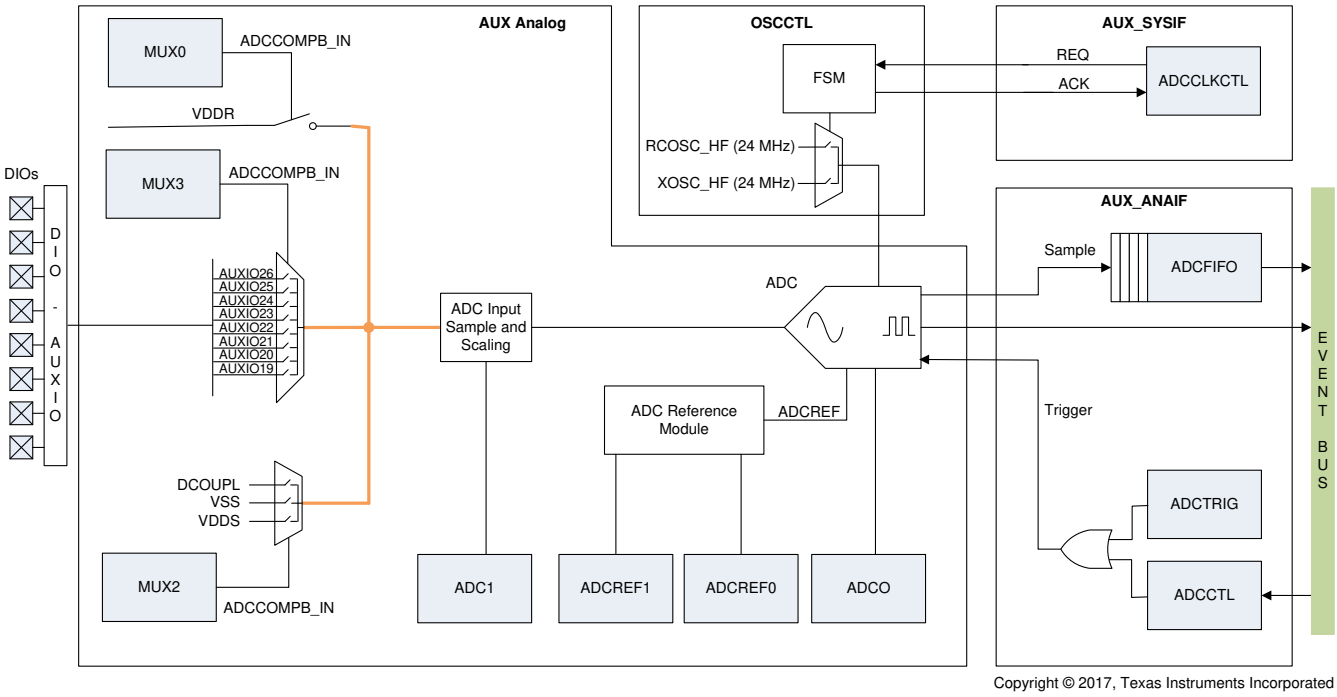


图 19-29. ADC 方框图

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.5.2.2 功能说明

19.5.2.2.1 输入选择和缩放

支持模拟功能的 AUX I/O（AUXIO19 至 AUXIO26）以及一些内部芯片电压可以连接到 ADC 输入端。

默认情况下，输入在用于 ADC 之前会在内部按 1408/4095 的比例进行下调。可以禁用 ADC 输入的下调缩放，以提高分辨率，但代价是输入范围缩小。

> **小心**
>
> 禁用下调缩放会改变 ADC 输入的最大额定值。更高的电压可能会损坏 ADC。参阅特定器件的数据表。

用户必须确保 ADC 输入始终只有一个驱动源。为了避免在内部将信号短接在一起，TI 提供了基于 ROM 的 DriverLib 函数，以确保内部多路复用器的先断后合切换。

19.5.2.2.2 基准选择

ADC 基准模块的配置只能在 ADI_4_AUX.ADC0.RESET_N = 0 时进行。必须使能 ADC 基准模块才能执行 ADC 采样和转换。该模块提供由 ADI_4_AUX:ADCREF0.SRC 选择的 ADC 参考电压，该电压可以是固定电压或相对于 VDDS 的电压。

如节 19.5.2.2.3 中所述，在同步 ADC 采样模式下，可以关闭 ADC 基准模块以节省功耗，这要求可配置的采样周期长于 ADC 基准模块的上电时间。有关说明，请参见节 19.8.1 中的 ADI_4_AUX.ADCREF0.REF_ON_IDLE。

设置 ADI_4_AUX.ADCREF0.EN 以使能 ADC 基准模块。

19.5.2.2.3 ADC 采样模式

采样模式和采样持续时间的配置只能在 ADI_4_AUX.ADC0.RESET_N = 0 时进行。ADC 输入是一个开关电容级，输入电压在转换完成之前先被采样并保持。ADC 在同步模式或异步模式下运行，如下所述：

- 在同步模式下，ADC 触发使能对输入进行可编程时间的采样，然后进行转换。此模式允许对高阻抗源进行正确采样。ADI_4_AUX:ADC0.SMPL_CYCLE_EXP 设置采样持续时间。
- 在异步模式下，ADC 持续采样，直到 ADC 触发停止采样并开始转换。此模式可以实现无抖动的采样，但代价是电流消耗增加。

ADI_4_AUX.ADC0.SMPL_MODE 选择采样模式。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.5.2.2.4 ADC 时钟源

ADC 内核使用内部 24MHz 时钟进行采样和转换。要使用 ADC，必须使能此内部时钟源以启用 ADC 内核时钟。有关说明，请参见节 19.8.9 中的 AUX_SYSIF:ADCCLKCTL。

一旦请求 ADC 时钟，时钟源遵循表 19-32。

表 19-32. ADC 时钟源

RCOSC_HF	XOSC_HF	ADC 时钟源
关闭	关闭	RCOSC_HF
关闭	开启	XOSC_HF
开启	关闭	RCOSC_HF
开启	开启	XOSC_HF

为了在异步模式下实现精确、低抖动的采样，软件必须确保 SCLK_HF 源自 24MHz 晶体。

备注

当 ADC 时钟使能时，系统无法进入待机或关断模式，因为电源控制器会请求一个高频时钟源。

最后，用户必须设置 ADI_4_AUX:ADC0.EN 以使能模拟 ADC 内核并启用对 ADC 触发的响应。

19.5.2.2.5 ADC 触发器

按照以下顺序配置 ADC 启动触发：

1. 事件驱动：AUX_ANAIF:ADCCTL 选择一个事件源和极性来触发 ADC
2. 手动：将 AUX_ANAIF:ADCCTL.START_SRC 设置为 NO_EVENT 以手动触发 ADC

节 19.5.2.2.3 中描述的 ADC 采样模式控制着 ADC 触发发生时的操作。

19.5.2.2.6 采样 FIFO

ADC 连接到一个可容纳最多四个 ADC 样本的 FIFO。读取 AUX_ANAIF:ADCFIFO 可获取最旧的样本。出于调试目的，也可以向 FIFO 写入虚拟样本。FIFO 中的样本未针对 ADC 的偏移和增益误差进行补偿。用户必须使用芯片生产期间存储的出厂配置数据手动执行补偿，请参阅节 11.2.1。系统 CPU 可以使用 AUX_ADC TI-DriverLib 模块完成此任务。

AUX_ANAIF:ADCFIFOSTAT 提供 FIFO 状态标志，如溢出、下溢和使用情况。所有 FIFO 状态标志以及 ADC 转换完成事件都连接到 AUX 事件总线。

要从溢出或下溢状况中恢复，用户必须刷新 FIFO 并重新使能数字 ADC 接口，如 AUX_ANAIF:ADCCTL.CMD 中所述（请参阅节 19.8.8）。

备注

调试软件时，显示 AUX_ANAIF_ADCFIFO 寄存器会导致 JTAG 读取 FIFO，这会将样本从 FIFO 中弹出，因此软件将无法读取它。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.5.2.2.7 µDMA 接口

µDMA 通道 7 专用于从 ADC FIFO 传输 ADC 样本。按照以下顺序设置 µDMA 传输：

1. 在 AUX_EVCTL:DMACTL 中配置并使能 µDMA 接口
   - 突发请求或单次请求
2. µDMA 传输触发
   - AUX_ADC_FIFO_NOT_EMPTY
   - AUX_ADC_FIFO_ALMOST_FULL
3. 在 µDMA 中配置块大小

当 µDMA 完成数据块传输时，会设置 AUX_ADC_IRQ 事件。AUX_ADC_IRQ 映射到系统 CPU 中断线 32（有关更多说明，请参见节 5.7.2 中的 EVENT:CPUIRQSEL32）。

备注

当 ADC FIFO 发生溢出或下溢时，也会设置此事件，如 AUX_ANAIF:ADCFIFOSTAT 所示。

19.5.2.2.8 资源所有权和使用情况

使用 ADC 需要拥有应用程序所有权。有关 TI 软件中的信标分配，请参阅节 19.4.3.3。

系统 CPU 可以在 TI-RTOS 功耗模式（工作和空闲）下使用外设。传感器控制器必须请求活动 AUX 工作模式才能使用外设。

19.5.3 COMPA

19.5.3.1 简介

AUX COMPA 外设是一个高性能连续时间比较器，具有以下特性：

- 输入来自支持模拟功能的 AUX I/O 引脚
- 基准电压来自支持模拟功能的 AUX I/O 引脚或内部芯片电压

对于电容式触控感应，COMPA 与 ISRC 和 TDC 外设配合使用。然后，ISRC 将为 COMPA 驱动一个内部基准电压（标称值为 0.8V），同时对连接到 COMPA 输入端的电容进行充电。从开始充电到 COMPA 输出变为高电平的时间使用 TDC 进行测量。

脉冲计数是另一个涉及 COMPA 和 TDC 的应用。在此应用中，TDC 预分频器对 COMPA 事件进行计数。

图 19-30 显示了 COMPA 如何连接到模拟节点、模拟外设以及数字接口。节 19.1 中描述了正确的参考 DAC 控制和连接。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

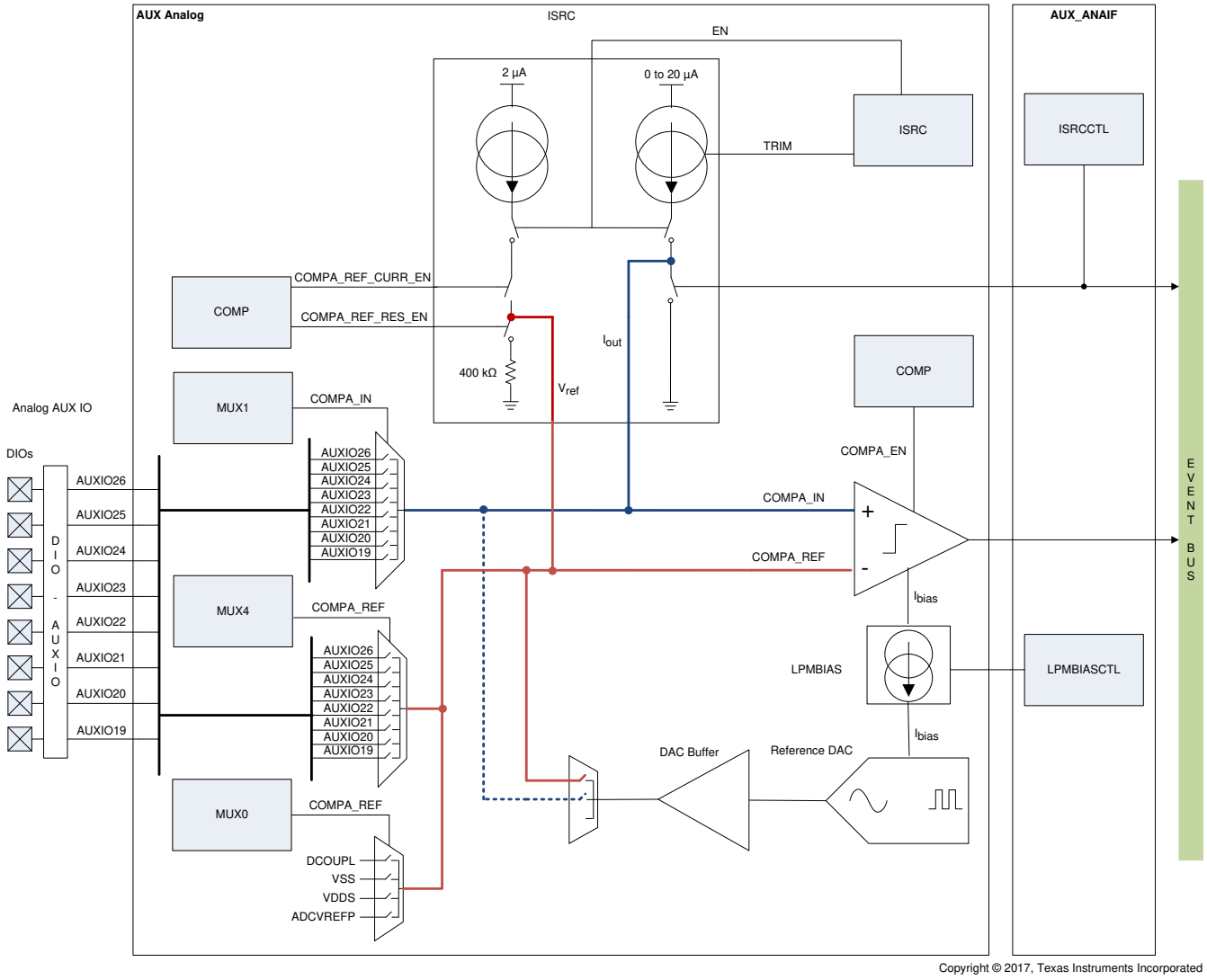


图 19-30. COMPA 方框图

19.5.3.2 功能说明

19.5.3.2.1 输入选择

COMPA 正输入可连接到：

- AUXIO19 至 AUXIO26，这些引脚支持模拟功能
- ISRC 外设的可编程电流输出
- 参考 DAC 输出

参考 DAC 输出连接并非为 COMPA 使用而设计。因此，该连接在图 19-30 中以虚线显示。

用户必须确保 COMPA 外设的正输入始终只有一个驱动源。为了避免在内部将信号短接在一起，TI 提供了基于 ROM 的 TI-DriverLib 函数，以确保内部多路复用器的先断后合切换。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.5.3.2.2 基准选择

COMPA 的基准输入可连接到：

- AUXIO19 至 AUXIO26，这些引脚支持模拟功能
- ISRC 外设的基准电压输出
- 参考 DAC 输出

用户必须确保 COMPA 的基准输入始终只有一个驱动源。为了避免在内部将信号短接在一起，TI 提供了基于 ROM 的 TI-DriverLib 函数，以确保内部多路复用器的先断后合切换。

19.5.3.2.3 LPM 偏置和 COMPA 使能

在以下情况下，COMPA 外设需要来自低功耗模式偏置 (LPMBIAS) 模块的偏置电流：

- 传感器控制器拥有该外设，并在低功耗和断电工作模式下使用它
- 系统 CPU 拥有该外设，并在待机 TI-RTOS 电源模式下使用它

设置 AUX_ANAIF:LPMBIASCTL.EN 以使能 LPM 偏置模块，然后设置 ADI_4_AUX:COMP.COMPA_EN 以使能 COMPA。

19.5.3.2.4 资源所有权和使用情况

使用 COMPA 需要应用程序拥有 COMPA 的所有权，并且如果参考 DAC 用于生成基准电压，也需要拥有参考 DAC 的所有权。有关 TI 软件中的信标分配，请参阅节 19.4.3.3。

系统 CPU 可以在 TI-RTOS 电源模式（工作和空闲）下使用外设。在这种情况下，将 AUX_SYSIF:EVSYNCRATE.AUX_COMPA_SYNC_RATE 设置为 BUS_RATE。要在待机 TI-RTOS 电源模式下使用比较器，请参阅节 19.5.5.2.6。

Sensor Controller Studio 提供了使用 COMPA 所需的 API，并且传感器控制器可以在所有 AUX 工作模式下使用此外设。

19.5.4 COMPB

19.5.4.1 简介

AUX COMPB 外设是一款低功耗时钟比较器，其更新频率为 32kHz，具有以下特性

- 输入来自支持模拟功能的 AUX I/O 引脚
- 可编程基准电压

COMPB 的主要应用是持续监测慢速信号并唤醒系统 CPU 或传感器控制器。

图 19-31 显示了 COMPB 如何连接到模拟节点、模拟外设以及数字接口。节 19.1 中描述了正确的参考 DAC 控制和连接。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

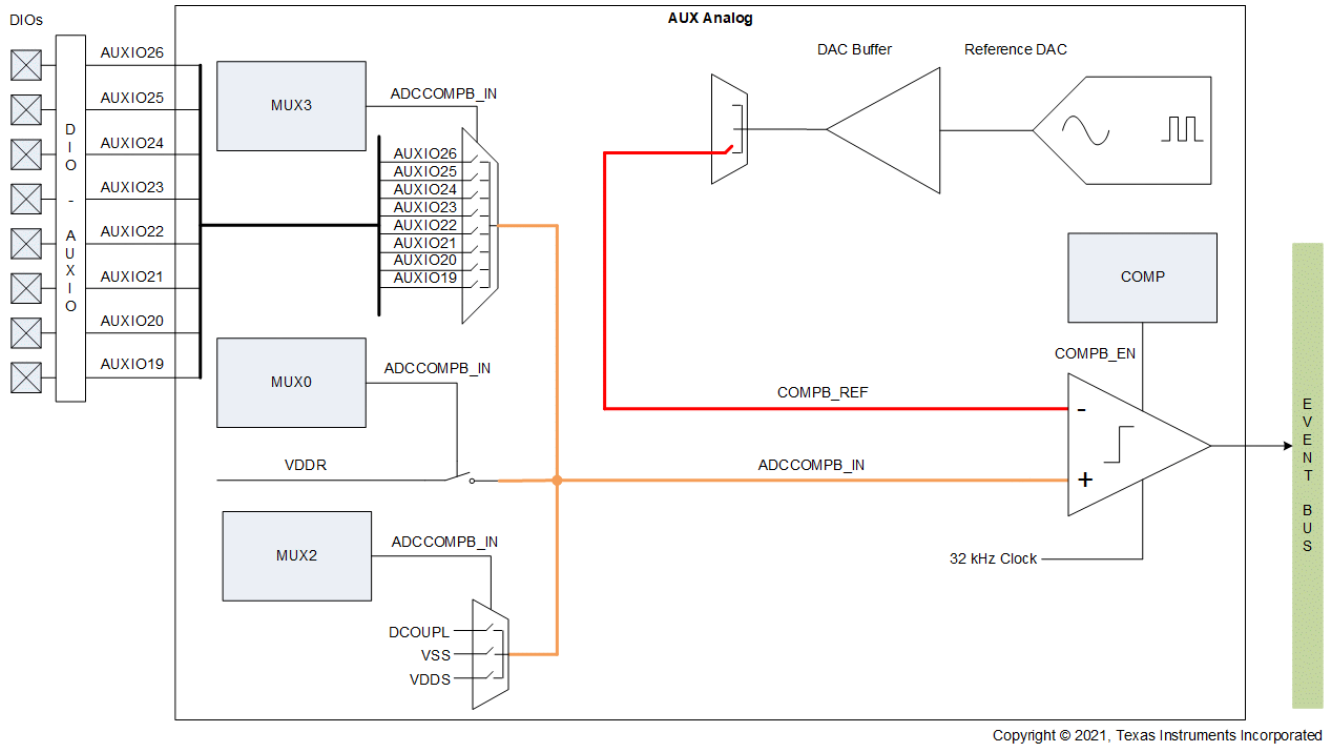


图 19-31. COMPB 方框图

19.5.4.2 功能说明

19.5.4.2.1 输入选择

COMPB 正输入可连接到：

- AUXIO19 至 AUXIO26，这些引脚支持模拟功能
- 内部电源电压

用户必须确保 COMPB 正输入始终只有一个驱动源。为了避免在内部将信号短接在一起，TI 提供了基于 ROM 的 TI-DriverLib 函数，以确保内部多路复用器的先断后合切换。

19.5.4.2.2 基准选择

参考 DAC 提供可编程的 COMPB 电压基准。将 AUX_ANAIF:DACCTL.DAC_VOUT_SEL 设置为 COMPB_REF 以连接模拟节点。

19.5.4.2.3 资源所有权和使用情况

使用 COMPB 需要应用程序拥有 COMPB 以及用于生成基准电压的参考 DAC 的所有权。有关 TI 软件中的信标分配，请参阅节 19.4.3.3。

COMPB 的主要应用是唤醒系统 CPU 或传感器控制器。

19.5.4.2.3.1 传感器控制器唤醒

Sensor Controller Studio 提供了将 COMPB 事件用作传感器控制器唤醒所需的 API。传感器控制器可以在所有 AUX 工作模式下使用此外设。

19.5.4.2.3.2 系统 CPU 唤醒

当应用程序未使用传感器控制器时，COMPB 事件可以通过两种方式触发系统 CPU 唤醒：

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

- 直接唤醒：选择 AUX_COMPB_ASYNC 或 AUX_COMPB_ASYNC_N 作为 AON_EVENT:MCUWUSEL 事件源。
- 间接唤醒：选择 AUX_COMPB 作为 AON_EVENT:MCUWUSEL 事件源，并配置 COMPB 事件极性以设置 AUX_EVCTL:EVTOAONFLAGS.AUX_COMPB 标志。

直接唤醒具有最小的唤醒延迟。但是，由于事件直接来自比较器，因此无法在 AON_EVENT 中清除 COMPB 事件。

间接唤醒会产生与 SCE 时钟速率成比例的延迟，而 SCE 时钟速率取决于 AUX 工作模式（如节 19.2.1 所述）。该延迟由两级同步器和事件标志捕获决定，并以三个 SCE 时钟周期为上界。系统 CPU 可以在唤醒时清除事件标志，因此可以在触发唤醒的 COMPB 事件电平仍然有效时返回睡眠状态。有关 AUX 到 AON 域的事件接口，请参阅节 19.6.6。

由于在此场景下未使用传感器控制器，系统 CPU 可以将 COMPB 事件配置为 AUX 的唤醒事件源。这可以确保与 2MHz 时钟成比例的低唤醒延迟。有关说明，请参阅节 19.1 和节 19.8.9 中的 AUX_SYSIF:PROGWUnCFG 寄存器。

对于间接唤醒，系统 CPU 应用程序必须使用以下顺序：

1. 将 AUX_SYSIF:EVSYNCRATE.AUX_COMPB_SYNC_RATE 设置为 BUS_RATE。
2. 将 AUX 工作模式设置为低功耗。
3. 使用有效的断电配置来配置参考 DAC（请参阅节 19.5.5.2.6）。
4. 等待基准电压稳定。
5. 配置 COMB 事件边沿以触发 AUX 唤醒。
6. 将 AUX 工作模式设置为断电，AUX_SYSIF:OPMODE.REQ = PDLP。
7. 唤醒后，按照 AUX_SYSIF:WUFLAGSCLR（请参阅节 19.8.9）中的说明清除唤醒标志。

当应用程序使用传感器控制器时，必须由传感器控制器提供系统 CPU 唤醒。相关设置在 Sensor Controller Studio 中完成。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.5.5 参考 DAC

19.5.5.1 简介

AUX 基准数模转换器 (DAC) 外设具有以下特性：

- 一个 8 位数字控制字
- 可配置的输出范围
- 为 COMPA 和 COMPB 生成基准电压
- 为支持模拟功能的 AUX I/O 引脚预充电

参考 DAC 的主要应用是为各种电源模式下的比较器提供可配置的基准电压。因此，参考 DAC 通常与其他模拟外设一起使用。

图 19-32 显示了参考 DAC 如何连接到模拟节点和数字接口。

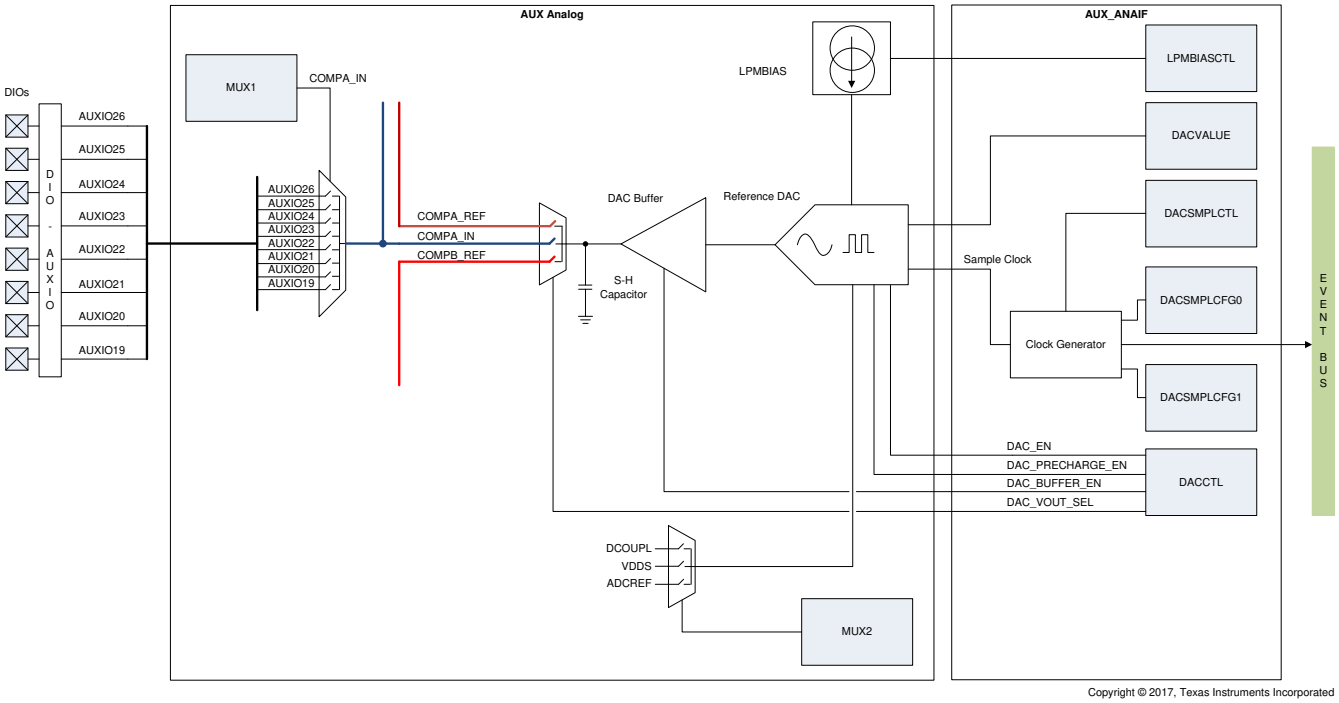


图 19-32. 参考设计方框图

19.5.5.2 功能说明

参考 DAC 使用可配置的采样时钟，将 8 位数字代码转换为存储在采样保持 (S-H) 电容器中的模拟电压。S-H 电容器直接连接到所选负载，或在驱动较高负载时通过缓冲器连接。

19.5.5.2.1 基准选择

ADI_4_AUX:MUX2.DAC_VREF_SEL 将 DAC 基准设置为：

- DCOUPL
- ADCREF
- VDDS

ADI_4_AUX:MUX2.DAC_VREF_SEL 寄存器一次仅允许选择一个基准电压。该基准电压决定了输出电压范围。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.5.5.2.2 输出电压控制和范围

S-H 电容器的目标电压由 AUX_ANAIF:DACVALUE 设置（请参阅节 19.8.8）。

参考 DAC 的传递函数表现出非线性。

参考 DAC 的校准数据在芯片生产期间存储在工厂配置中，请参见节 11.4.1 中的 DAC_CAL0 和 DAC_CAL1。

19.5.5.2.3 采样时钟

DAC 采样时钟对存储在 S-H 电容器中的 DAC 电压进行充电和保持。DAC 采样时钟波形由建立阶段和紧随其后的保持阶段组成。与保持阶段相比，建立阶段的采样时钟频率通常更高。

在建立阶段，采样保持电容器充电至目标电压，而保持阶段则维持 S-H 电压。通过降低保持阶段的采样时钟频率可以节省功耗。图 19-33 阐述了这一概念。

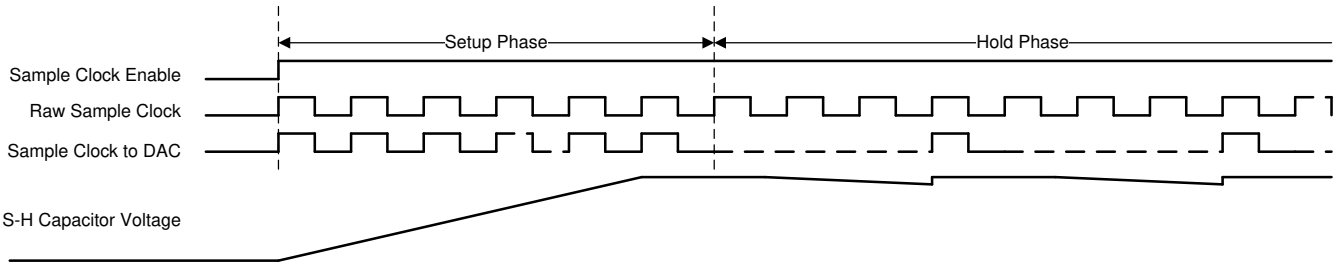


图 19-33. DAC 采样时钟相位

采样时钟周期由 AUX_ANAIF:DACSMPLCFG0 和 AUX_ANAIF:DACSMPLCFG1 寄存器配置，并通过 AUX_ANAIF:DACSMPLCTL 寄存器使能。采样时钟周期随 AUX_SYSIF:PEROPRATE.ANAIF_DAC_OP_RATE 设置的外设时钟频率成比例缩放。图 19-34 显示了采样时钟周期如何从外设时钟派生。

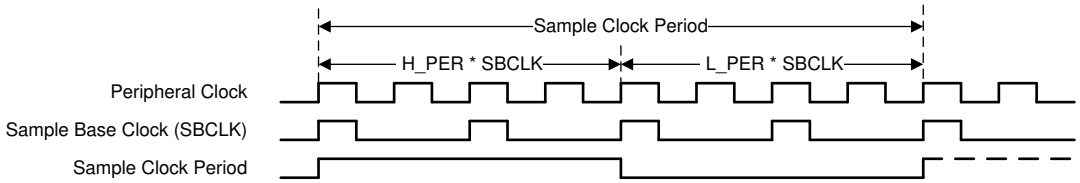


图 19-34. 采样时钟周期配置

AUX_ANAIF:DACSMPLCFG0 控制从外设时钟到采样基时钟的时钟分频。AUX_ANAIF:DACSMPLCFG0 寄存器在图 19-34 中设置为 2。

AUX_ANAIF:DACSMPLCFG.H_PER 和 AUX_ANAIF:DACSMPLCFG.L_PER 分别以基时钟周期数为单位，指定单个采样时钟周期的高电平和低电平阶段的持续时间。它们都在图 19-34 中设置为 2 个周期。

一个状态机可以控制从建立阶段到保持阶段的转换，或者用户也可以手动控制此转换。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.5.5.2.3.1 自动相位控制

自动相位转换要求建立阶段需要少于 17 个采样时钟周期才能将 S-H 电容器充电至目标电压。当参考 DAC 驱动外部负载时，通常不是这种情况。

按以下顺序配置自动相位控制：

1. AUX_ANAIF:DACSMPLCFG1.SETUP_CNT 指定建立阶段的采样时钟周期数。
2. AUX_ANAIF:DACSMPLCFG1.HOLD_INTERVAL 指定每次 S-H 刷新时钟之间的非活动采样时钟周期数。

该转换会设置 AUX_EVCTL:EVSTAT3.AUX_DAC_HOLD_ACTIVE 事件。

19.5.5.2.3.2 手动相位控制

如果建立阶段需要超过 16 个采样时钟周期才能将 S-H 电容器充电至目标值，则用户必须手动控制相位转换。按如下方式配置手动相位控制：

1. 将 AUX_ANAIF:DACSMPLCFG1.SETUP_CNT 设置为 0。
2. 将 AUX_ANAIF:DACSMPLCFG1.HOLD_INTERVAL 初始设置为 0。

当 S-H 电容器达到目标电压时，用户必须更新 AUX_ANAIF:DACSMPLCFG1.HOLD_INTERVAL。

或者，用户可以利用 AUX_EVCTL:EVSTAT3.AUX_DAC_HOLD_ACTIVE 并在循环中重新启动采样时钟生成。此方法不需要任何其他硬件资源。

19.5.5.2.3.3 工作模式依赖

在 AUX 断电工作模式下，当 AUX_SYSIF:OPMODEREQ.REQ 等于 LPPD 时，只允许一种采样时钟配置。

- 将所有 AUX_ANAIF:DACSMPLCFG1 字段设置为 0。

在此场景中，用户必须在 AUX_SYSIF:OPMODEREQ.REQ = LP 时使能采样时钟生成，然后再请求 LPPD。

当 AUX_SYSIF:OPMODEREQ.REQ = PDA 时，不允许生成采样时钟。

19.5.5.2.4 输出选择

AUX_ANAIF:DACCTL.DAC_VOUT_SEL 选择参考 DAC 输出连接：

- 无连接
- 外部负载，COMPA_IN 模拟节点
- COMPA 基准输入，COMPA_REF 模拟节点
- COMPB 基准输入，COMPB_REF 模拟节点

每次只允许选择一个输出连接。用户必须确保所连接的模拟节点始终只有一个驱动源。为了避免在内部将信号短接在一起，TI 提供了基于 ROM 的 TI-DriverLib 函数，以确保内部多路复用器的先断后合切换。

以下各小节描述了不同的连接。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.5.5.2.4.1 缓存器

基准 DAC 可通过缓冲器连接至选定的负载。当基准 DAC 连接外部负载时，必须启用该缓冲器。不建议将内部负载与缓冲器连接起来。

若要启用缓冲器，请将 AUX_ANAIF:DACCTL.DAC_BUFFER_EN 设置为 1。若要在 LPM（低功耗模式）下启用缓冲器，还需将 AUX_ANAIF:LPMBIASCTL.EN 设置为 1 以启用偏置模块。当 AUX 请求断电操作模式时，无法使用该缓冲器。

19.5.5.2.4.2 外部负载

当外设时钟频率为 2MHz 或 24MHz 时，参考 DAC 可以驱动连接到 COMPA_IN 模拟节点的外部负载。这适用于以下情况：

- 活动和低功耗 AUX 工作模式
- 活动和空闲 TI-RTOS 电源模式

当参考 DAC 连接外部负载时，用户必须使能缓冲器。

负载支持：

- 电阻负载 > 10MΩ
- 根据 AUX_DAC_Drvier 规范：“如果 DAC 在 2.0 V 下工作，则要求负载的电阻 >>10 MΩ 或为纯容性。如果不满足此条件，则必须使用 CHP 来减小 VOUT 误差。”
- 容性负载 < 200pF（始终稳定）
- 容性负载
- R+C

19.5.5.2.4.3 COMPA_REF

参考 DAC 可在所有电源模式下驱动 COMPA_REF 模拟节点。在这种情况下，图 19-28 中所示的 COMPA_REF 多路复用器与 COMPA_REF 节点断开连接。通常不建议为此连接使用缓冲器。

19.5.5.2.4.4 COMPB_REF

基准 DAC 可在所有电源模式下驱动 COMPB_REF 模拟节点。通常不建议为此连接使用缓冲器。

19.5.5.2.5 LPM 偏置

在以下情况下，参考 DAC 需要来自低功耗模式偏置 (LPMBIAS) 模块的偏置电流：

- 传感器控制器拥有该外设，并在低功耗和断电工作模式下使用它。
- 系统 CPU 拥有该外设，并在待机 TI-RTOS 电源模式下使用它。

设置 AUX_ANAIF:LPMBIASCTL.EN 以使能 LPM 偏置模块，然后设置 AUX_ANAIF:DACCTL.DAC_EN 以使能参考 DAC。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.5.5.2.6 资源所有权和使用情况

使用参考 DAC 需要拥有应用程序所有权。有关 TI 软件中的信标分配，请参阅节 19.4.3.3。

Sensor Controller Studio 提供了使用参考 DAC 所需的 API，并且传感器控制器可以在所有 AUX 工作模式下使用此外设。

系统 CPU 可以在 TI-RTOS 电源模式（工作和空闲）下使用外设。在这种情况下，将 AUX_SYSIF:PEROPRATE.ANAIF_DAC_OP_RATE 设置为 BUS_RATE。

如果应用程序未使用传感器控制器，系统 CPU 可以在待机 TI-RTOS 电源模式下使用此外设。在这种情况下，系统 CPU 必须将 AON_PMCTL.AUXSCECLK.PD_SRC 设置为 SCLK_LF，并在低功耗和断电工作模式之间切换。需要遵循以下顺序：

1. 将 AUX_SYSIF:PEROPRATE.ANAIF_DAC_OP_RATE 设置为 SCE_RATE。
2. 将 AUX_SYSIF:OPMODEREQ 设置为 LP。
3. 设置所需的 AUX_ANAIF:DACVALUE。
4. 配置 AUX_ANAIF:DACCTL：
   a. 将 DAC_PRECHARGE_EN 设置为所需值。
   b. 将 DAC_VOUT_SEL 设置为所需值。
   c. 将 AUX_ANAIF:DACCTL.DAC_BUFFER_EN 设置为 0。
5. 将 AUX_ANAIF:LPMBIASCTL 设置为 1。
6. 等待 15 µs。
7. 将 AUX_ANAIF:DACSMPLCFG0.CLKDIV 设置为 0。
8. 配置 AUX_ANAIF:DACSMPLCFG1：
   a. 将 H_PER 设置为 1。
   b. 将 L_PER 设置为 0。
   c. 将 SETUP_CNT 设置为 0。
   d. 将 HOLD_INTERVAL 设置为 0。
9. 将 AUX_ANAIF:DACSMPLCTL 设置为 1 以使能采样时钟生成。
10. 将 AUX_ANAIF:DACSMPLCFG1.H_PER 设置为 0。
11. 等待所需时间，将 S-H 电容器充电至请求的电压。
12. 将 AUX_SYSIF:OPMODEREQ 设置为 PDLP。
13. 进入睡眠并等待中断。

19.5.6 ISRC

19.5.6.1 简介

AUX Current Source (ISRC) 模块具有以下特性：

- 可编程电流输出：0A 至 20µA
- 电压基准输出：标称值为 0.8V

ISRC 外设的主要应用是电容式触控感应，在此应用中，它与 COMPA 和 TDC 外设配合使用。在此应用中，ISRC 为 COMPA 驱动一个内部基准电压（标称值为 0.8 V），同时对连接到 COMPA 输入端的电容进行充电。从开始充电到 COMPA 输出变为高电平的时间使用 TDC 进行测量。

图 19-35 显示了 ISRC 的模拟连接和数字控制。节 19.1 中描述了正确的参考 DAC 控制和连接。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

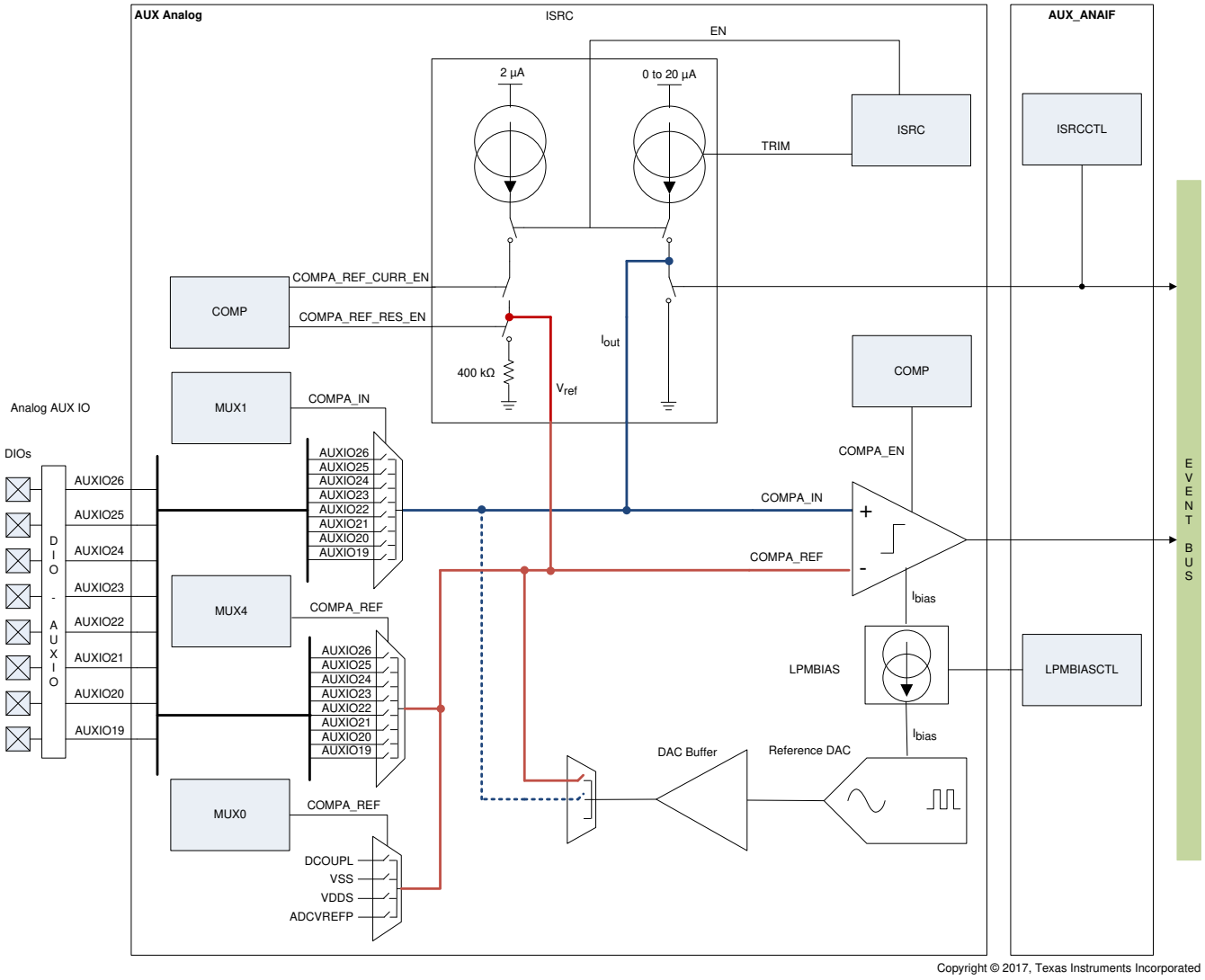


图 19-35. ISRC 方框图

19.5.6.2 功能说明

19.5.6.2.1 可编程电流

可编程电流输出连接到 COMPA 的正输入端，从而可以对连接到支持模拟功能的 AUX I/O 的外部电容进行充电。物理上，该输出也连接到参考 DAC 输出。此连接并非为 ISRC 使用而设计。因此，该连接在图 19-35 中以虚线显示。

当 ISRC 使能时，它驱动由 ADI_4_AUX:ISRC.TRIM 配置的电流进入 COMPA_IN 模拟节点或流入地。该连接由 AUX_ANAIF:ISRCCTL.RESET_N 控制。

用户必须确保 COMPA 正输入始终只有一个驱动源。为了避免在内部将信号短接在一起，TI 提供了基于 ROM 的 TI-DriverLib 函数，以确保内部多路复用器的先断后合切换。

19.5.6.2.2 电压基准

电压基准输出连接到 COMPA 的基准输入。使用内部 400kΩ 电阻器时，生成的电压基准的标称值为 0.8V。要生成基准电压并将其连接至 COMPA 基准输入：

- 设置 ADI_4_AUX:COMP.COMPA_REF_RES_EN 以使能内部 400kΩ 电阻器。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

- 设置 ADI_4_AUX:COMP.COMPA_REF_CURR_EN 以使能 2µA 电流。

还可以使用外部电阻器来生成其他电压。在这种情况下，要生成基准电压并将其连接到 COMPA 基准输入：

- 将一个电阻器连接到具有模拟功能的 AUX I/O。
- 将该 AUX I/O 连接到 COMPA 基准输入。
- 设置 ADI_4_AUX:COMP.COMPA_REF_CURR_EN 以使能 2µA 电流。

用户必须确保 COMPA 基准输入始终只有一个驱动源。为了避免在内部将信号短接在一起，TI 提供了基于 ROM 的 TI-DriverLib 函数，以确保内部多路复用器的先断后合切换。

19.5.6.2.3 ISRC 使能

设置 ADI_4_AUX:ISRC.EN 以使能该模块并生成电流。

19.5.6.2.4 不受温度影响

ISRC 输出与温度相关。当 ISRC 为 COMPA 正输入节点充电并将内部生成的 0.8 基准电压连接到 COMPA 基准输入时，该效应会消除。

19.5.6.2.5 资源所有权和使用情况

使用 ISRC 需要应用程序。有关 TI 软件中的信标分配，请参阅节 19.4.3.3。

系统 CPU 可以在 TI-RTOS 功耗模式（工作和空闲）下使用外设。传感器控制器必须请求活动 AUX 工作模式才能使用外设。

19.6 事件路由和使用

19.6.1 AUX 事件总线

AUX 域事件总线汇集来自以下源的事件：

- AUX 外设
- AUX I/O 引脚
- 实时时钟
- 电池监测器和温度传感器
- 系统电源和时钟管理

事件汇集成一个 64 位的事件总线，该总线存在两种版本，一种是部分异步的，另一种是与 AUX 时钟完全同步的。用户可以在 AUX_EVCTL:EVSTAT0 至 AUX_EVCTL:EVSTAT3 寄存器中读取同步事件的电平。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.6.1.1 事件信号

表 19-33 总结了事件信号。有关详细的事件说明，请参阅节 19.8.3 中的 AUX_EVCTL:EVSTAT0 至 AUX_EVCTL:EVSTAT3 寄存器。

表 19-33. AUX 事件总线

索引	异步事件总线(1) (2)		同步事件总线(3) (4) (5)	
	起点	信号	SCE 速率	总线速率
[0:31]	DIO	AUXIO[0:31]	X	
32	AUX	MANUAL_EV	X	
33	系统	AON_RTC_CH2	X	
34		AON_RTC_CH2_DLY	X	
35		AON_RTC_4KHZ	X	
36		AON_BATMON_BAT_UPD	X	
37		AON_BATMON_TEMP_UPD	X	
38		SCLK_LF	X	
39		PWR_DWN	X	
40		MCU_ACTIVE	X	
41		VDDR_RECHARGE	X	
42		ACLK_REF	X	
43		MCU_EV	X	
44		MCU_OBSMUX0	X	
45		MCU_OBSMUX1	X	
46	AUX	AUX_COMPA	X	(X)
47		AUX_COMPB	X	(X)
48		AUX_TIMER2_EV0	X	(X)
49		AUX_TIMER2_EV1	X	(X)
50		AUX_TIMER2_EV2	X	(X)
51		AUX_TIMER2_EV3	X	(X)
52		AUX_TIMER2_PULSE	X	(X)
53	AUX	AUX_TIMER1_EV	X	(X)
54		AUX_TIMER0_EV	X	(X)
55		AUX_TDC_DONE		X
56		AUX_ISRC_RESET_N		X
57		AUX_ADC_DONE		X
58		AUX_ADC_IRQ		X
59		AUX_ADC_FIFO_ALMOST_FULL		X
60		AUX_ADC_FIFO_NOT_EMPTY		X
61		AUX_SMPH_AUTOTAKE_DONE		X
62		DAC_HOLD_ACTIVE	X	(X)
63		TIMER2_CLKSWITCH_RDY		X

(1) 无阴影部分的事件具有以 AUX 时钟为基准的异步源。
(2) 蓝色阴影部分的事件具有以 AUX 时钟为基准的同步源。
(3) X = 默认速率
(4) (X) = 可覆盖速率
(5) 这些列显示了同步事件总线上的事件更新速率。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

更新速率由事件同步或生成该事件的同步外设的运行速率决定。

表 19-33 显示了智能超声波探头的原理框图：

- 某些异步源事件具有可配置的同步速率。该速率通过 AUX_SYSIF:EVSYNCRATE 配置。
- 某些同步源站事件具有可配置的更新速率。该速率通过 AUX_SYSIF:PEROPRATE 配置。

这种可配置性用于在系统 CPU 和传感器控制器之间共享资源。

由于同步，异步源事件的异步版本与同步版本之间始终存在延时时间。系统 CPU 可能需要最小的同步延迟，在这种情况下，同步必须以总线时钟速率发生。较慢的 AUX SCE 时钟速率不会影响延迟。另一方面，当 SCE 时钟速率低于总线时钟速率时，任何以总线时钟速率同步的事件将在 MCU 域活动时经历较低的延迟。这对于传感器控制器任务或 Timer01 来说可能不是期望的行为。在这种情况下，事件同步速率必须等于 SCE 时钟速率。

19.6.1.2 事件订阅者

表 19-34 列出 AUX 事件总线订阅者及其订阅的事件总线版本。

表 19-34. AUX 事件订阅者

订阅者	异步 事件总线	同步 事件总线
传感器控制器		X
Timer0 和 Timer1		X
TDC 状态机	X	
TDC 预分频器	X	
TDC 高速计数器	X	
Timer2	X	
ADC 接口	X	
AUX 可编程唤醒	X	

19.6.1.2.1 事件检测

订阅者以不同的方式检测事件。一般而言，无法保证事件持续足够长的时间以便被检测到。

19.6.1.2.1.1 异步事件检测

异步事件由同步器检测，或该事件被用作时钟来设置一个检测标志，然后该标志实现同步。表 19-35 列出了用于订阅者的检测方案。

表 19-35. 异步事件检测

订阅者	检测方案
TDC 预分频器	事件作为时钟
TDC 高速计数器	同步器
Timer 2	同步器
ADC 接口	事件作为时钟
AUX 可编程唤醒	事件作为时钟

任何异步源的事件（例如 AUX I/O 或 COMPA 事件）可能持续时间过短，无法在给定的时钟频率下被同步器正确检测。同样，任何 Timer2 事件可能持续时间过短，无法正确同步到较慢的 AUX SCE 时钟。通常，异步事件的脉冲宽度必须长于采样时钟周期，才能被同步器正确检测。

为了确保事件订阅者能够捕获同步后的事件，必须满足以下条件之一：

- 事件订阅者的操作时钟频率必须等于或高于同步器的时钟频率。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

- 同步后的事件的持续时间必须至少等于订阅者模块的有效时钟周期。

当异步事件被用作时钟来设置检测标志时，所允许的事件脉冲宽度远小于同步器所允许的宽度。

19.6.1.2.1.2 同步事件检测

当传感器控制器和 Timer01 使用具有同步来源的事件时，也存在类似的要求。如果事件订阅者的运行速率低于事件发布者的运行速率，订阅者可能会丢失事件。当事件持续时间短于订阅者模块的有效时钟周期时，就会发生这种情况。由此产生的后果如下：

- 由于 ADC 接口以 SCLK_HFDIV2 时钟速率运行，传感器控制器和 Timer01 运行速率必须等于 SCLK_HFDIV2 才能订阅 AUX_ADC_DONE 事件。
- 如果传感器控制器订阅 AUX_TIMER0_EV 或 AUX_TIMER1_EV，则计时器运行速率必须等于或低于传感器控制器运行速率。

19.6.2 在外部引脚上观察事件

可以通过将 AUX_EVCTL.EVOBSCFG 选中的异步事件输出到外部 AUX I/O 引脚上来进行观察。为此，用户必须将相应的 AUX I/O 配置为外设模式输出，并将 AUX_EV_OBS 事件设置为数据源。

此功能适用于以下场景：

- 调试
- 设置外部引脚电平，同时触发内部外设操作，例如 TDC 转换

有关 AUX I/O 配置，请参见节 19.4.2 和图 19-11。

19.6.3 来自 MCU 域的事件

MCU 事件结构在表 19-33 中定义了索引 43 处的 MCU_EV 事件。EVENT:AUXSEL0 配置此事件的源。

AUX_EVCTL:EVTOMCUFLAGS 保存了 16 个连接到 MCU 事件结构和系统 CPU 的事件标志。事件标志名称与其所源自的同步事件总线上的事件相同。表 19-36 列出了事件标志及其灵敏度类型。

19.6.4 发送到 MCU 域的事件

AUX_EVCTL:EVTOMCUFLAGS 保存了 16 个连接到 MCU 事件结构和系统 CPU 的事件标志。事件标志名称与其所源自的同步事件总线上的事件相同。表 19-36 列出了事件标志及其灵敏度类型。

当 AUX_EVCTL:EVTOMCUPOL 中为事件配置的电平或边沿发生时，会设置相应的事件标志。为了清除电平敏感标志，必须在用户清除事件标志之前清除相应的事件。向 AUX_EVCTL:EVTOMCUFLAGSCLR 中的事件索引写入 1 以清除电平或边沿敏感标志。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-36. 发送到 MCU 的事件

索引	事件标志名称	灵敏度	说明
0	AUX_WU_EV	电平	AUX 唤醒事件(1)
1	AUX_COMPA	边缘	比较器 A 事件
2	AUX_COMPB	边缘	比较器 B 事件
3	AUX_TDC_DONE	电平	TDC 转换完成或超时
4	AUX_TIMER0_EV	电平	AUX Timer0 已达到其目标值
5	AUX_TIMER1_EV	电平	AUX Timer1 已达到其目标值
6	AUX_SMPH_AUTOTAKE_DONE	电平	已释放用户指定的信标
7	AUX_ADC_DONE	电平	ADC 转换完成
8	AUX_ADC_FIFO_ALMOST_FULL	电平	ADC FIFO 几乎已满
9	MCU_OBSMUX0	电平	MCU 可观察事件。
10	AUX_ADC_IRQ	电平	ADC IRQ(2)
11	AUX_TIMER2_EV0	电平	AUX 计时器 2 事件 0
12	AUX_TIMER2_EV1	电平	AUX 计时器 2 事件 1
13	AUX_TIMER2_EV2	电平	AUX 计时器 2 事件 2
14	AUX_TIMER2_EV3	电平	AUX 计时器 2 事件 3
15	AUX_TIMER2_PULSE	电平	AUX 计时器 2 单个时钟脉冲

(1) AUX_SYSIF:WUFLAGS 寄存器中 AUX 唤醒事件的逻辑或。
(2) ADC 新样本可用，FIFO 下溢或上溢。如果使用 DMA：ADC DMA 完成，FIFO 下溢或上溢。

用户可以选择将 AUX_COMB 事件作为 AUX_EVCTL:EVTOMCUFLAGS 中 16 个事件标志的函数使能。AUX_EVCTL:COMBEVTOMCUMASK 配置确定是否通过设置事件标志来设置 AUX_COMB 事件。只要设置了 AUX_EVCTL:COMBEVTOMCUMASK 选择的一个或多个事件标志，AUX_COMB 事件就为高电平。

MCU 事件结构将 16 个事件标志和 AUX_COMB 事件相连接，作为：

- DMA 触发条件
- 无线电计时器捕获
- 系统 CPU 中断
- 通用计时器捕获

请参阅章节 5 中的 EVENT:* 了解所有可能的连接。

19.6.5 来自 AON 域的事件

表 19-33 中索引 32 至 41 的事件来自 AON 域。这些事件主要用于两个目的：

- WAKEUP（唤醒）：这些事件用作 AUX 域和传感器控制器的唤醒源，既可以直接使用，也可以通过 Timer01 进行分频后使用。
- Wait-Event（等待事件）：当事件达到指定的事件电平时，传感器控制器会暂停执行，直到满足该事件条件。

唤醒用法示例：

- Timer0 将 AON_RTC_4KHZ 事件分频，用于生成 AUX 域唤醒以及传感器控制器任务的执行，使其每 10 ms 运行一次。
- AON_RTC_CH2 事件唤醒 AUX 域。AON_RTC_CH2_DLY 事件触发传感器控制器的任务执行。

等待事件用法示例：

- 当 AON_BATMON_BAT_UPD 为高电平时，表示 AUX_SYSIF:BATMONBAT 的新值已就绪。
- 当 AON_BATMON_TEMP_UPD 为高电平时，表示 AUX_SYSIF:BATMONTEMP 的新值已就绪。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.6.6 发送到 AON 域的事件

AUX_EVCTL:EVTOAONFLAGS 保存了九个连接到 AON 事件结构的事件标志。事件标志名称与其所源自的同步事件总线上的事件相同。表 19-37 列出了事件标志及其灵敏度类型。

表 19-37. 发送到 AON 的事件

索引	事件标志名称	灵敏度	说明
0	SWEV0	不适用	直接映射到 AUX_EVCTL:SWEVSET.SWEV0
1	SWEV1	不适用	直接映射到 AUX_EVCTL:SWEVSET.SWEV1
2	SWEV2	不适用	直接映射到 AUX_EVCTL:SWEVSET.SWEV2
3	AUX_COMPA	边缘	比较器 A 事件
4	AUX_COMPB	边缘	比较器 B 事件
5	AUX_ADC_DONE	电平	ADC 转换完成
6	AUX_TDC_DONE	电平	TDC 转换完成或超时
7	AUX_TIMER0_EV	电平	AUX Timer0 已达到其目标值
8	AUX_TIMER1_EV	电平	AUX Timer1 已达到其目标值

当 AUX_EVCTL:EVTOAONPOL 中为事件配置的电平或边沿发生时，会设置相应的事件标志。SWEV0 到 SWEV2 事件标志的电平由寄存器控制。要清除电平敏感标志，必须在用户清除事件标志之前清除相应的事件。向 AUX_EVCTL:EVTOAONFLAGSCLR 中的事件索引写入 1 以清除电平或边沿敏感标志。

事件标志 SWEV0 和 SWEV1 也直接连接到系统 CPU。

这些事件标志可用于：

- MCU 域唤醒（请参阅节 5.7.1 中的 AON_EVENT:MCUWUSEL）
- 可编程事件到 MCU（请参阅节 5.7.1 中的 AON_EVENT:EVTOMCUSEL）
- RTC 捕获事件（请参阅节 5.7.1 中的 AON_EVENT:RTCSEL）
- 系统 CPU 中断：
  - SWEV0 映射到 EVENT:CPUIRQSEL6
  - SWEV1 映射到 EVENT:CPUIRQSEL13
- 软件定义的握手协议

AUX_TIMER2_EV* 事件也会直接路由到 AON_EVENT，无需经过 AUX 时钟的中间同步。

19.6.7 µDMA 接口

µDMA 通道 7 专用于从 ADC FIFO 传输 ADC 样本。有关更说多明，请参阅节 19.5.2.2.7。

系统 CPU 和传感器控制器可以通过 AUX_EVCTL:DMASWREQ 寄存器在通道 8 上请求 µDMA 传输。这需要正确设置 µDMA 通道 8。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.7 传感器控制器别名寄存器空间

大多数外设地址在 AUX 存储器空间中被混叠到较短的 256 个字（512 字节）。只有传感器控制器才能访问别名寄存器空间。传感器控制器会尽可能使用别名地址（请参阅 表 19-38）。与使用非别名化的 16 位外设地址相比，使用别名地址可使指令执行速度提高一个 SCE 时钟周期。

表 19-38. 别名寄存器映射

寄存器组	寄存器名称	原始地址	别名地址
AUX_SPIM	SPIMCFG	0x400C 1000 + 0x00	0
AUX_SPIM	MISOCFG	0x400C 1000 + 0x04	1
AUX_SPIM	MOSICTL	0x400C 1000 + 0x08	2
AUX_SPIM	TX8	0x400C 1000 + 0x0C	3
AUX_SPIM	TX16	0x400C 1000 + 0x10	4
AUX_SPIM	RX8	0x400C 1000 + 0x14	5
AUX_SPIM	RX16	0x400C 1000 + 0x18	6
AUX_SPIM	SCLKIDLE	0x400C 1000 + 0x1C	7
AUX_SPIM	DATAIDLE	0x400C 1000 + 0x20	8
AUX_MAC	OP0S	0x400C 2000 + 0x00	9
AUX_MAC	OP0U	0x400C 2000 + 0x04	10
AUX_MAC	OP1SMUL	0x400C 2000 + 0x08	11
AUX_MAC	OP1UMUL	0x400C 2000 + 0x0C	12
AUX_MAC	OP1SMAC	0x400C 2000 + 0x10	13
AUX_MAC	OP1UMAC	0x400C 2000 + 0x14	14
AUX_MAC	OP1SADD16	0x400C 2000 + 0x18	15
AUX_MAC	OP1UADD16	0x400C 2000 + 0x1C	16
AUX_MAC	OP1SADD32	0x400C 2000 + 0x20	17
AUX_MAC	OP1UADD32	0x400C 2000 + 0x24	18
AUX_MAC	CLZ	0x400C 2000 + 0x28	19
AUX_MAC	CLS	0x400C 2000 + 0x2C	20
AUX_MAC	ACCSHIFT	0x400C 2000 + 0x30	21
AUX_MAC	ACCRESET	0x400C 2000 + 0x34	22
AUX_MAC	ACC15_0	0x400C 2000 + 0x38	23
AUX_MAC	ACC31_16	0x400C 2000 + 0x78	24
AUX_MAC	ACC39_32	0x400C 2000 + 0x9C	25
AUX_TIMER2	CTL	0x400C 3000 + 0x00	26
AUX_TIMER2	目标	0x400C 3000 + 0x04	27
AUX_TIMER2	SHDWTARGET	0x400C 3000 + 0x08	28
AUX_TIMER2	CNTR	0x400C 3000 + 0x0C	29
AUX_TIMER2	PRECFG	0x400C 3000 + 0x10	30
AUX_TIMER2	EVCTL	0x400C 3000 + 0x14	31
AUX_TIMER2	PULSETRIG	0x400C 3000 + 0x18	32
AUX_TIMER2	CH0EVCFG	0x400C 3000 + 0x80	33
AUX_TIMER2	CH1EVCFG	0x400C 3000 + 0x90	34
AUX_TIMER2	CH2EVCFG	0x400C 3000 + 0xA0	35
AUX_TIMER2	CH3EVCFG	0x400C 3000 + 0xB0	36
AUX_TIMER2	CH0CCFG	0x400C 3000 + 0x84	37
AUX_TIMER2	CH1CCFG	0x400C 3000 + 0x94	38
AUX_TIMER2	CH2CCFG	0x400C 3000 + 0xA4	39

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-38. 别名寄存器映射（续）

寄存器组	寄存器名称	原始地址	别名地址
AUX_TIMER2	CH3CCFG	0x400C 3000 + 0xB4	40
AUX_TIMER2	CH0PCC	0x400C 3000 + 0x88	41
AUX_TIMER2	CH1PCC	0x400C 3000 + 0x98	42
AUX_TIMER2	CH2PCC	0x400C 3000 + 0xA8	43
AUX_TIMER2	CH3PCC	0x400C 3000 + 0xB8	44
AUX_TIMER2	CH0CC	0x400C 3000 + 0x8C	45
AUX_TIMER2	CH1CC	0x400C 3000 + 0x9C	46
AUX_TIMER2	CH2CC	0x400C 3000 + 0xAC	47
AUX_TIMER2	CH3CC	0x400C 3000 + 0xBC	48
AUX_TDC	CTL	0x400C 4000 + 0x00	49
AUX_TDC	STAT	0x400C 4000 + 0x04	50
AUX_TDC	RESULT_LO	0x400C 4000 + 0x08	51
AUX_TDC	RESULT_HI	0x400C 4000 + 0x0A	52
AUX_TDC	SATCFG	0x400C 4000 + 0x0C	53
AUX_TDC	TRIGSRC	0x400C 4000 + 0x10	54
AUX_TDC	TRIGCNT	0x400C 4000 + 0x14	55
AUX_TDC	TRIGCNTLOAD	0x400C 4000 + 0x18	56
AUX_TDC	TRIGCNTCFG	0x400C 4000 + 0x1C	57
AUX_TDC	PRECTL	0x400C 4000 + 0x20	58
AUX_TDC	PRECNTR	0x400C 4000 + 0x24	59
AUX_EVCTL	SCEWEVCFG0	0x400C 5000 + 0x10	60
AUX_EVCTL	SCEWEVCFG1	0x400C 5000 + 0x14	61
AUX_EVCTL	DMACTL	0x400C 5000 + 0x18	62
AUX_EVCTL	DMASWREQ	0x400C 5000 + 0x1C	63
AUX_EVCTL	SWEVSET	0x400C 5000 + 0x20	64
AUX_EVCTL	EVTOAONFLAGS	0x400C 5000 + 0x24	65
AUX_EVCTL	EVTOAONPOL	0x400C 5000 + 0x28	66
AUX_EVCTL	EVTOAONFLAGSCLR	0x400C 5000 + 0x2C	67
AUX_EVCTL	EVTOMCUFLAGS	0x400C 5000 + 0x30	68
AUX_EVCTL	EVTOMCUPOL	0x400C 5000 + 0x34	69
AUX_EVCTL	EVTOMCUFLAGSCLR	0x400C 5000 + 0x38	70
AUX_EVCTL	COMBEVTOMCUMASK	0x400C 5000 + 0x3C	71
AUX_EVCTL	EVOBSCFG	0x400C 5000 + 0x40	72
AUX_EVCTL	PROGDLY	0x400C 5000 + 0x44	73
AUX_EVCTL	手动	0x400C 5000 + 0x48	74
AUX_EVCTL	EVSTAT0L	0x400C 5000 + 0x4C	75
AUX_EVCTL	EVSTAT0H	0x400C 5000 + 0x50	76
AUX_EVCTL	EVSTAT1L	0x400C 5000 + 0x54	77
AUX_EVCTL	EVSTAT1H	0x400C 5000 + 0x58	78
AUX_EVCTL	EVSTAT2L	0x400C 5000 + 0x5C	79
AUX_EVCTL	EVSTAT2H	0x400C 5000 + 0x60	80
AUX_EVCTL	EVSTAT3L	0x400C 5000 + 0x64	81
AUX_EVCTL	EVSTAT3H	0x400C 5000 + 0x68	82
AUX_SYSIF	OPMODEREQ	0x400C 6000 + 0x00	83
AUX_SYSIF	OPMODEACK	0x400C 6000 + 0x04	84

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-38. 别名寄存器映射（续）

寄存器组	寄存器名称	原始地址	别名地址
AUX_SYSIF	PROGWU0CFG	0x400C 6000 + 0x08	85
AUX_SYSIF	PROGWU1CFG	0x400C 6000 + 0x0C	86
AUX_SYSIF	PROGWU2CFG	0x400C 6000 + 0x10	87
AUX_SYSIF	PROGWU3CFG	0x400C 6000 + 0x14	88
AUX_SYSIF	SWWUTRIG	0x400C 6000 + 0x18	89
AUX_SYSIF	WUFLAGS	0x400C 6000 + 0x1C	90
AUX_SYSIF	WUFLAGSCLR	0x400C 6000 + 0x20	91
AUX_SYSIF	WUGATE	0x400C 6000 + 0x24	92
AUX_SYSIF	VECCFG0	0x400C 6000 + 0x28	93
AUX_SYSIF	VECCFG1	0x400C 6000 + 0x2C	94
AUX_SYSIF	VECCFG2	0x400C 6000 + 0x30	95
AUX_SYSIF	VECCFG3	0x400C 6000 + 0x34	96
AUX_SYSIF	VECCFG4	0x400C 6000 + 0x38	97
AUX_SYSIF	VECCFG5	0x400C 6000 + 0x3C	98
AUX_SYSIF	VECCFG6	0x400C 6000 + 0x40	99
AUX_SYSIF	VECCFG7	0x400C 6000 + 0x44	100
AUX_SYSIF	EVSYNCRATE	0x400C 6000 + 0x48	101
AUX_SYSIF	PEROPRATE	0x400C 6000 + 0x4C	102
AUX_SYSIF	ADCCLKCTL	0x400C 6000 + 0x50	103
AUX_SYSIF	TDCCLKCTL	0x400C 6000 + 0x54	104
AUX_SYSIF	TDCREFCLKCTL	0x400C 6000 + 0x58	105
AUX_SYSIF	TIMER2CLKCTL	0x400C 6000 + 0x5C	106
AUX_SYSIF	TIMER2CLKSTAT	0x400C 6000 + 0x60	107
AUX_SYSIF	TIMER2CLKSWITCH	0x400C 6000 + 0x64	108
AUX_SYSIF	TIMER2DBGCTL	0x400C 6000 + 0x68	109
AUX_SYSIF	BGAPREQ	0x400C 6000 + 0x6C	110
AUX_SYSIF	CLKSHIFTDET	0x400C 6000 + 0x70	111
AUX_SYSIF	RECHARGETRIG	0x400C 6000 + 0x74	112
AUX_SYSIF	RECHARGEDET	0x400C 6000 + 0x78	113
AUX_SYSIF	RTCSUBSECINC0	0x400C 6000 + 0x7C	114
AUX_SYSIF	RTCSUBSECINC1	0x400C 6000 + 0x80	115
AUX_SYSIF	RTCSUBSECINCCTL	0x400C 6000 + 0x84	116
AUX_SYSIF	RTCSEC	0x400C 6000 + 0x88	117
AUX_SYSIF	RTCSUBSEC	0x400C 6000 + 0x8C	118
AUX_SYSIF	RTCEVCLR	0x400C 6000 + 0x90	119
AUX_SYSIF	BATMONBAT	0x400C 6000 + 0x94	120
AUX_SYSIF	BATMONTEMP	0x400C 6000 + 0x9C	121
AUX_SYSIF	TIMERHALT	0x400C 6000 + 0xA0	122
AUX_SYSIF	OPMODESTAT	0x400C 6000 + 0xAC	123
AUX_SYSIF	TIMER2BRIDGE	0x400C 6000 + 0xB0	124
AUX_TIMER01	T0CFG	0x400C 7000 + 0x00	125
AUX_TIMER01	T1CFG	0x400C 7000 + 0x10	126
AUX_TIMER01	T0CTL	0x400C 7000 + 0x04	127
AUX_TIMER01	T1CTL	0x400C 7000 + 0x14	128
AUX_TIMER01	T0TARGET	0x400C 7000 + 0x08	129

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-38. 别名寄存器映射（续）

寄存器组	寄存器名称	原始地址	别名地址
AUX_TIMER01	T1TARGET	0x400C 7000 + 0x18	130
AUX_TIMER01	T0CNTR	0x400C 7000 + 0x0C	131
AUX_TIMER01	T1CNTR	0x400C 7000 + 0x1C	132
AUX_SMPH	SMPH0	0x400C 8000 + 0x00	133
AUX_SMPH	SMPH1	0x400C 8000 + 0x04	134
AUX_SMPH	SMPH2	0x400C 8000 + 0x08	135
AUX_SMPH	SMPH3	0x400C 8000 + 0x0C	136
AUX_SMPH	SMPH4	0x400C 8000 + 0x10	137
AUX_SMPH	SMPH5	0x400C 8000 + 0x14	138
AUX_SMPH	SMPH6	0x400C 8000 + 0x18	139
AUX_SMPH	SMPH7	0x400C 8000 + 0x1C	140
AUX_SMPH	AUTOTAKE	0x400C 8000 + 0x20	141
AUX_ANAIF	ADCCTL	0x400C 9000 + 0x10	142
AUX_ANAIF	ADCFIFOSTAT	0x400C 9000 + 0x14	143
AUX_ANAIF	ADCFIFO	0x400C 9000 + 0x18	144
AUX_ANAIF	ADCTRIG	0x400C 9000 + 0x1C	145
AUX_ANAIF	ISRCCTL	0x400C 9000 + 0x20	146
AUX_ANAIF	DACCTL	0x400C 9000 + 0x30	147
AUX_ANAIF	LPMBIASCTL	0x400C 9000 + 0x34	148
AUX_ANAIF	DACSMPLCTL	0x400C 9000 + 0x38	149
AUX_ANAIF	DACSMPLCFG0	0x400C 9000 + 0x3C	150
AUX_ANAIF	DACSMPLCFG1	0x400C 9000 + 0x40	151
AUX_ANAIF	DACVALUE	0x400C 9000 + 0x44	152
AUX_ANAIF	DACSTAT	0x400C 9000 + 0x48	153
AUX_ADI4	SET01	0x400C B000 + 0x10	154
AUX_ADI4	SET23	0x400C B000 + 0x12	155
AUX_ADI4	SET45	0x400C B000 + 0x14	156
AUX_ADI4	SET67	0x400C B000 + 0x16	157
AUX_ADI4	SET89	0x400C B000 + 0x18	158
AUX_ADI4	SET1011	0x400C B000 + 0x1A	159
AUX_ADI4	SET1213	0x400C B000 + 0x1C	160
AUX_ADI4	SET1415	0x400C B000 + 0x1E	161
AUX_ADI4	CLR01	0x400C B000 + 0x20	162
AUX_ADI4	CLR23	0x400C B000 + 0x22	163
AUX_ADI4	CLR45	0x400C B000 + 0x24	164
AUX_ADI4	CLR67	0x400C B000 + 0x26	165
AUX_ADI4	CLR89	0x400C B000 + 0x28	166
AUX_ADI4	CLR1011	0x400C B000 + 0x2A	167
AUX_ADI4	CLR1213	0x400C B000 + 0x2C	168
AUX_ADI4	CLR1415	0x400C B000 + 0x2E	169
AUX_AIODIO0	IOMODEL	0x400C C000 + 0x40	170
AUX_AIODIO0	IOMODEH	0x400C C000 + 0x44	171
AUX_AIODIO1	IOMODEL	0x400C D000 + 0x40	172
AUX_AIODIO1	IOMODEH	0x400C D000 + 0x44	173
AUX_AIODIO2	IOMODEL	0x400C E000 + 0x40	174

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-38. 别名寄存器映射（续）

寄存器组	寄存器名称	原始地址	别名地址
AUX_AIODIO2	IOMODEH	0x400C E000 + 0x44	175
AUX_AIODIO3	IOMODEL	0x400C F000 + 0x40	176
AUX_AIODIO3	IOMODEH	0x400C F000 + 0x44	177
AUX_AIODIO0	GPIODIE	0x400C C000 + 0x04	178
AUX_AIODIO1	GPIODIE	0x400C D000 + 0x04	179
AUX_AIODIO2	GPIODIE	0x400C E000 + 0x04	180
AUX_AIODIO3	GPIODIE	0x400C F000 + 0x04	181
AUX_AIODIO0	IOPOE	0x400C C000 + 0x08	182
AUX_AIODIO1	IOPOE	0x400C D000 + 0x08	183
AUX_AIODIO2	IOPOE	0x400C E000 + 0x08	184
AUX_AIODIO3	IOPOE	0x400C F000 + 0x08	185
AUX_AIODIO0	GPIODOUT	0x400C C000 + 0x0C	186
AUX_AIODIO1	GPIODOUT	0x400C D000 + 0x0C	187
AUX_AIODIO2	GPIODOUT	0x400C E000 + 0x0C	188
AUX_AIODIO3	GPIODOUT	0x400C F000 + 0x0C	189
AUX_AIODIO0	GPIODIN	0x400C C000 + 0x10	190
AUX_AIODIO1	GPIODIN	0x400C D000 + 0x10	191
AUX_AIODIO2	GPIODIN	0x400C E000 + 0x10	192
AUX_AIODIO3	GPIODIN	0x400C F000 + 0x10	193
AUX_AIODIO0	GPIODOUTSET	0x400C C000 + 0x14	194
AUX_AIODIO1	GPIODOUTSET	0x400C D000 + 0x14	195
AUX_AIODIO2	GPIODOUTSET	0x400C E000 + 0x14	196
AUX_AIODIO3	GPIODOUTSET	0x400C F000 + 0x14	197
AUX_AIODIO0	GPIODOUTCLR	0x400C C000 + 0x18	198
AUX_AIODIO1	GPIODOUTCLR	0x400C D000 + 0x18	199
AUX_AIODIO2	GPIODOUTCLR	0x400C E000 + 0x18	200
AUX_AIODIO3	GPIODOUTCLR	0x400C F000 + 0x18	201
AUX_AIODIO0	GPIODOUTTGL	0x400C C000 + 0x1C	202
AUX_AIODIO1	GPIODOUTTGL	0x400C D000 + 0x1C	203
AUX_AIODIO2	GPIODOUTTGL	0x400C E000 + 0x1C	204
AUX_AIODIO3	GPIODOUTTGL	0x400C F000 + 0x1C	205
AUX_AIODIO0	IO0PSEL	0x400C C000 + 0x20	206
AUX_AIODIO0	IO1PSEL	0x400C C000 + 0x24	207
AUX_AIODIO0	IO2PSEL	0x400C C000 + 0x28	208
AUX_AIODIO0	IO3PSEL	0x400C C000 + 0x2C	209
AUX_AIODIO0	IO4PSEL	0x400C C000 + 0x30	210
AUX_AIODIO0	IO5PSEL	0x400C C000 + 0x34	211
AUX_AIODIO0	IO6PSEL	0x400C C000 + 0x38	212
AUX_AIODIO0	IO7PSEL	0x400C C000 + 0x3C	213
AUX_AIODIO1	IO0PSEL	0x400C D000 + 0x20	214
AUX_AIODIO1	IO1PSEL	0x400C D000 + 0x24	215
AUX_AIODIO1	IO2PSEL	0x400C D000 + 0x28	216
AUX_AIODIO1	IO3PSEL	0x400C D000 + 0x2C	217
AUX_AIODIO1	IO4PSEL	0x400C D000 + 0x30	218
AUX_AIODIO1	IO5PSEL	0x400C D000 + 0x34	219

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-38. 别名寄存器映射（续）

寄存器组	寄存器名称	原始地址	别名地址
AUX_AIODIO1	IO6PSEL	0x400C D000 + 0x38	220
AUX_AIODIO1	IO7PSEL	0x400C D000 + 0x3C	221
AUX_AIODIO2	IO0PSEL	0x400C E000 + 0x20	222
AUX_AIODIO2	IO1PSEL	0x400C E000 + 0x24	223
AUX_AIODIO2	IO2PSEL	0x400C E000 + 0x28	224
AUX_AIODIO2	IO3PSEL	0x400C E000 + 0x2C	225
AUX_AIODIO2	IO4PSEL	0x400C E000 + 0x30	226
AUX_AIODIO2	IO5PSEL	0x400C E000 + 0x34	227
AUX_AIODIO2	IO6PSEL	0x400C E000 + 0x38	228
AUX_AIODIO2	IO7PSEL	0x400C E000 + 0x3C	229
AUX_AIODIO3	IO0PSEL	0x400C F000 + 0x20	230
AUX_AIODIO3	IO1PSEL	0x400C F000 + 0x24	231
AUX_AIODIO3	IO2PSEL	0x400C F000 + 0x28	232
AUX_AIODIO3	IO3PSEL	0x400C F000 + 0x2C	233
AUX_AIODIO3	IO4PSEL	0x400C F000 + 0x30	234
AUX_AIODIO3	IO5PSEL	0x400C F000 + 0x34	235
AUX_AIODIO3	IO6PSEL	0x400C F000 + 0x38	236
AUX_AIODIO3	IO7PSEL	0x400C F000 + 0x3C	237

19.8 AUX 域传感器控制器和外设寄存器

表 19-39 列出了 AUX 子系统中的寄存器。有关存储器映射，请参阅章节 3。

表 19-39. AUX 域传感器控制器和外设寄存器

模块	名称	部分
AUX_ADI4	ADI 主模式	节 19.8.1
AUX_AIODIO	模拟数字 IO	节 19.8.2
AUX_EVCTL	事件控制器	节 19.8.3
AUX_SMPH	信标	节 19.8.4
AUX_TDC	时间数字转换器	节 19.8.5
AUX_TIMER01	Timer01	节 19.8.6
AUX_TIMER2	Timer2	节 19.8.7
AUX_ANAIF	模拟接口	节 19.8.8
AUX_SYSIF	系统接口	节 19.8.9
AUX_SPIM	SPI 主模式	节 19.8.10
AUX_MAC	数学加速器	节 19.8.11

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.1 ADI_4_AUX 寄存器

表 19-40 列出了 ADI_4_AUX 寄存器的存储器映射寄存器。表 19-40 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

表 19-40. ADI_4_AUX 寄存器

偏移	首字母缩写词	寄存器名称	部分
0h	MUX0	内部	MUX0 寄存器（偏移 = 0h）[复位 = 00h]
1h	MUX1	内部	MUX1 寄存器（偏移 = 1h）[复位 = 00h]
2h	MUX2	内部	MUX2 寄存器（偏移 = 2h）[复位 = 00h]
3h	MUX3	内部	MUX3 寄存器（偏移 = 3h）[复位 = 00h]
4h	ISRC	电流源	ISRC 寄存器（偏移 = 4h）[复位 = 00h]
5h	COMP	比较器	COMP 寄存器（偏移 = 5h）[复位 = 00h]
7h	MUX4	内部	MUX4 寄存器（偏移 = 7h）[复位 = 00h]
8h	ADC0	ADC 控制 0	ADC0 寄存器（偏移 = 8h）[复位 = 00h]
9h	ADC1	ADC 控制 1	ADC1 寄存器（偏移 = 9h）[复位 = 00h]
Ah	ADCREF0	ADC 基准 0	ADCREF0 寄存器（偏移 = Ah）[复位 = 00h]
Bh	ADCREF1	ADC 基准 1	ADCREF1 寄存器（偏移 = Bh）[复位 = 00h]
Eh	LPMBIAS	内部	LPMBIAS 寄存器（偏移 = Eh）[复位 = 00h]

复杂的位访问类型经过编码可适应小型表单元。表 19-41 展示了适用于此部分中访问类型的代码。

表 19-41. ADI_4_AUX 访问类型代码

访问类型	代码	说明
读取类型		
R	R	读取
写入类型		
W	W	写入
复位或默认值		
-n		复位后的值或默认值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.1.1 MUX0 寄存器（偏移 = 0h）[复位 = 00h]

图 19-36 展示了 MUX0，表 19-42 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 19-36. MUX0 寄存器

7	6	5	4	3	2	1	0
RESERVED	ADCCOMPB_IN	RESERVED		COMPA_REF			
R-0h	R/W-0h	R-0h		R/W-0h			

表 19-42. MUX0 寄存器字段说明

位	字段	类型	复位	说明
7	RESERVED	R	0h	保留
6	ADCCOMPB_IN	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
5-4	RESERVED	R	0h	保留
3-0	COMPA_REF	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.1.2 MUX1 寄存器（偏移 = 1h）[复位 = 00h]

图 19-37 展示了 MUX1，表 19-43 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 19-37. MUX1 寄存器

7	6	5	4	3	2	1	0
COMPA_IN							
R/W-0h							

表 19-43. MUX1 寄存器字段说明

位	字段	类型	复位	说明
7-0	COMPA_IN	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.1.3 MUX2 寄存器（偏移 = 2h）[复位 = 00h]

图 19-38 展示了 MUX2，表 19-44 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 19-38. MUX2 寄存器

7	6	5	4	3	2	1	0
ADCCOMPB_IN					DAC_VREF_SEL		
R/W-0h					R/W-0h		

表 19-44. MUX2 寄存器字段说明

位	字段	类型	复位	说明
7-3	ADCCOMPB_IN	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
2-0	DAC_VREF_SEL	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.1.4 MUX3 寄存器（偏移 = 3h）[复位 = 00h]

图 19-39 展示了 MUX3，表 19-45 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 19-39. MUX3 寄存器

7	6	5	4	3	2	1	0
ADCCOMPB_IN							
R/W-0h							

表 19-45. MUX3 寄存器字段说明

位	字段	类型	复位	说明
7-0	ADCCOMPB_IN	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.1.5 ISRC 寄存器（偏移 = 4h）[复位 = 00h]

图 19-40 展示了 ISRC，表 19-46 中对此进行了介绍。

返回到汇总表。

电流源
电流源的强度与微调控制。仅可通过 TI 提供的 API 使用。

图 19-40. ISRC 寄存器

7	6	5	4	3	2	1	0
TRIM						RESERVED	EN
R/W-0h						R-0h	R/W-0h

表 19-46. ISRC 寄存器字段说明

位	字段	类型	复位	说明
7-2	TRIM	R/W	0h	调节电流源的电流。 可将输出电流进行组合，以获得所需的总电流。 0h = 未连接电流 1h = 0P25U：0.25uA 2h = 0P5U： 0.5uA 4h = 1P0U： 1.0 uA 8h = 2P0U： 2.0uA 10h = 4P5U： 4.5 uA 20h = 11P75U： 11.75uA
1	RESERVED	R	0h	保留
0	EN	R/W	0h	电流源使能

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.1.6 COMP 寄存器（偏移 = 5h）[复位 = 00h]

图 19-41 展示了 COMP，表 19-47 中对此进行了介绍。

返回到汇总表。

比较器
控制 COMPA 和 COMPB 比较器。仅可通过 TI 提供的 API 使用。

图 19-41. COMP 寄存器

7	6	5	4	3	2	1	0
COMPA_REF_RES_EN	COMPA_REF_CURR_EN	LPM_BIAS_WIDTH_TRIM			COMPB_EN	RESERVED	COMPA_EN
R/W-0h	R/W-0h	R/W-0h			R/W-0h	R-0h	R/W-0h

表 19-47. COMP 寄存器字段说明

位	字段	类型	复位	说明
7	COMPA_REF_RES_EN	R/W	0h	在 COMPA 基准节点与接地之间建立 400kΩ 电阻连接。与 COMPA_REF_CURR_EN 配合使用，用于为电容感应生成电压基准。
6	COMPA_REF_CURR_EN	R/W	0h	启用从 ISRC 到 COMPA 基准节点的 2uA IPTAT 电流。要求 ISRC.EN = 1。与 COMPA_REF_RES_EN 配合使用，用于为电容感应生成电压基准。
5-3	LPM_BIAS_WIDTH_TRIM	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
2	COMPB_EN	R/W	0h	COMPB 使能
1	RESERVED	R	0h	保留
0	COMPA_EN	R/W	0h	COMPA 使能

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.1.7 MUX4 寄存器（偏移 = 7h）[复位 = 00h]

图 19-42 展示了 MUX4，表 19-48 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 19-42. MUX4 寄存器

7	6	5	4	3	2	1	0
COMPA_REF							
R/W-0h							

表 19-48. MUX4 寄存器字段说明

位	字段	类型	复位	说明
7-0	COMPA_REF	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.1.8 ADC0 寄存器（偏移 = 8h）[复位 = 00h]

图 19-43 展示了 ADC0，表 19-49 对其进行了介绍。

返回到汇总表。

ADC 控制 0
ADC 采样控制。仅可通过 TI 提供的 API 使用。

图 19-43. ADC0 寄存器

7	6	5	4	3	2	1	0
SMPL_MODE	SMPL_CYCLE_EXP				RESERVED	RESET_N	EN
R/W-0h	R/W-0h				R-0h	R/W-0h	R/W-0h

表 19-49. ADC0 寄存器字段说明

位	字段	类型	复位	说明
7	SMPL_MODE	R/W	0h	ADC 采样模式： 0：同步模式 1：异步模式 ADC 在转换前执行采样保持。在同步模式下，当 ADC 时钟检测到触发信号的上升沿时，采样即开始。如果触发信号源自与 ADC 时钟异步的时钟域，则检测过程中会引入抖动或不确定性。 SMPL_CYCLE_EXP 参数决定了采样持续时间。 采样结束后，转换立即开始。 在异步模式下，一旦启用，采样便会持续进行。当触发信号出现上升沿时，采样立即结束并随即开始转换。转换完成后，采样将重新开始。 当需要避免外部驱动信号的采样时刻出现抖动时，异步模式非常有用
6-3	SMPL_CYCLE_EXP	R/W	0h	当 ADC 在同步模式 (SMPL_MODE = 0) 下运行时，该控制转换前的采样持续时间。该设置在异步模式下无效。采样持续时间计算公式为：$2^{SMPL_CYCLE_EXP + 1}$ / 6us。 3h = 2P7_US：16 x 6MHz 时钟周期 = 2.7us 4h = 5P3_US： 32 x 6MHz 时钟周期 = 5.3us 5h = 10P6_US： 64 x 6MHz 时钟周期 = 10.6us 6h = 21P3_US： 128 x 6MHz 时钟周期 = 21.3us 7h = 42P6_US： 256 x 6MHz 时钟周期 = 42.6us 8h = 85P3_US： 512 x 6MHz 时钟周期 = 85.3us 9h = 170_US： 1024 x 6MHz 时钟周期 = 170us Ah = 341_US： 2048 x 6MHz 时钟周期 = 341us Bh = 682_US： 4096 x 6MHz 时钟周期 = 682us Ch = 1P37_MS： 8192 x 6MHz 时钟周期 = 1.37ms Dh = 2P73_MS： 16384 x 6MHz 时钟周期 = 2.73ms Eh = 5P46_MS： 32768 x 6MHz 时钟周期 = 5.46ms Fh = 10P9_MS： 65536 x 6MHz 时钟周期 = 10.9ms
2	RESERVED	R	0h	保留
1	RESET_N	R/W	0h	复位 ADC 数字子芯片，低电平有效。每次重新配置 ADC 时都必须将其复位。 0：复位 1：正常运行
0	EN	R/W	0h	ADC 启用 0：禁用 1：启用

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.1.9 ADC1 寄存器（偏移 = 9h）[复位 = 00h]

图 19-44 展示了 ADC1，表 19-50 对其进行了介绍。

返回到汇总表。

ADC 控制 1
ADC 比较器控制。仅可通过 TI 提供的 API 使用。

图 19-44. ADC1 寄存器

7	6	5	4	3	2	1	0
RESERVED							SCALE_DIS
R-0h							R/W-0h

表 19-50. ADC1 寄存器字段说明

位	字段	类型	复位	说明
7-1	RESERVED	R	0h	保留
0	SCALE_DIS	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.1.10 ADCREF0 寄存器（偏移 = Ah）[复位 = 00h]

图 19-45 展示了 ADCREF0，表 19-51 中对此进行了介绍。

返回到汇总表。

ADC 基准 0
ADC 使用的控制基准。仅可通过 TI 提供的 API 使用。

图 19-45. ADCREF0 寄存器

7	6	5	4	3	2	1	0
RESERVED	REF_ON_IDLE	IOMUX	EXT	SRC	RESERVED		EN
R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h		R/W-0h

表 19-51. ADCREF0 寄存器字段说明

位	字段	类型	复位	说明
7	RESERVED	R	0h	保留
6	REF_ON_IDLE	R/W	0h	在 IDLE 状态下启用 ADCREF。 0：在 IDLE 状态下禁用 1：在 IDLE 状态下启用 当 ADC0.SMPL_MODE = 0 时，令 ADCREF 保持启用状态。 建议：当 ADC0.SMPL_CYCLE_EXP 小于 0x6（采样时间为 21.3µs）时，务必启用 ADCREF。
5	IOMUX	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
4	EXT	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
3	SRC	R/W	0h	ADC 基准源： 0：固定基准 = 4.3V 1：相对基准 = VDDS
2-1	RESERVED	R	0h	保留
0	EN	R/W	0h	ADC 基准模块使能： 0：ADC 基准模块已断电： 1：ADC 基准模块已使能

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.1.11 ADCREF1 寄存器（偏移 = Bh）[复位 = 00h]

图 19-46 展示了 ADCREF1，表 19-52 中对此进行了介绍。

返回到汇总表。

ADC 基准 1
ADC 使用的控制基准。仅可通过 TI 提供的 API 使用。

图 19-46. ADCREF1 寄存器

7	6	5	4	3	2	1	0
RESERVED		VTRIM					
R-0h		R/W-0h					

表 19-52. ADCREF1 寄存器字段说明

位	字段	类型	复位	说明
7-6	RESERVED	R	0h	保留
5-0	VTRIM	R/W	0h	微调 ADC 固定基准的输出电压（64 个步进，二进制补码）。仅适用于 ADCREF0.SRC = 0。 示例： 0x00 - 标称电压 1.43V 0x01 - 标称值 + 0.4% 1.435V 0x3F - 标称值 - 0.4% 1.425V 0x1F - 最大电压 1.6V 0x20 - 最小电压 1.3V

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.1.12 LPMBIAS 寄存器（偏移 = Eh）[复位 = 00h]

图 19-47 展示了 LPMBIAS，表 19-53 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 19-47. LPMBIAS 寄存器

7	6	5	4	3	2	1	0
RESERVED		LPM_TRIM_IOUT					
R-0h		R/W-0h					

表 19-53. LPMBIAS 寄存器字段说明

位	字段	类型	复位	说明
7-6	RESERVED	R	0h	保留
5-0	LPM_TRIM_IOUT	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.2 AUX_AIODIO 寄存器

表 19-54 列出了 AUX_AIODIO 寄存器的存储器映射寄存器。表 19-54 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

表 19-54. AUX_AIODIO 寄存器

偏移	首字母缩写词	寄存器名称	部分
0h	IOMODE	输入输出模式	IOMODE 寄存器（偏移 = 0h）[复位 = 00000000h]
4h	GPIODIE	通用输入输出数字输入使能	GPIODIE 寄存器（偏移 = 4h）[复位 = 00000000h]
8h	IOPOE	输入输出外设输出使能	IOPOE 寄存器（偏移 = 8h）[复位 = 00000000h]
Ch	GPIODOUT	通用输入输出数据输出	GPIODOUT 寄存器（偏移 = Ch）[复位 = 00000000h]
10h	GPIODIN	通用输入输出数据输入	GPIODIN 寄存器（偏移 = 10h）[复位 = 00000000h]
14h	GPIODOUTSET	通用输入输出数据输出设置	GPIODOUTSET 寄存器（偏移 = 14h）[复位 = 00000000h]
18h	GPIODOUTCLR	通用输入输出数据输出清除	GPIODOUTCLR 寄存器（偏移 = 18h）[复位 = 00000000h]
1Ch	GPIODOUTTGL	通用输入输出数据输出切换	GPIODOUTTGL 寄存器（偏移 = 1Ch）[复位 = 00000000h]
20h	IO0PSEL	输入输出 0 外设选择	IO0PSEL 寄存器（偏移 = 20h）[复位 = 00000000h]
24h	IO1PSEL	输入输出 1 外设选择	IO1PSEL 寄存器（偏移 = 24h）[复位 = 00000000h]
28h	IO2PSEL	输入输出 2 外设选择	IO2PSEL 寄存器（偏移 = 28h）[复位 = 00000000h]
2Ch	IO3PSEL	输入输出 3 外设选择	IO3PSEL 寄存器（偏移 = 2Ch）[复位 = 00000000h]
30h	IO4PSEL	输入输出 4 外设选择	IO4PSEL 寄存器（偏移 = 30h）[复位 = 00000000h]
34h	IO5PSEL	输入输出 5 外设选择	IO5PSEL 寄存器（偏移 = 34h）[复位 = 00000000h]
38h	IO6PSEL	输入输出 6 外设选择	IO6PSEL 寄存器（偏移 = 38h）[复位 = 00000000h]
3Ch	IO7PSEL	输入输出 7 外设选择	IO7PSEL 寄存器（偏移 = 3Ch）[复位 = 00000000h]
40h	IOMODEL	输入输出模式低电平	IOMODEL 寄存器（偏移 = 40h）[复位 = 00000000h]
44h	IOMODEH	输入输出模式高电平	IOMODEH 寄存器（偏移 = 44h）[复位 = 00000000h]

复杂的位访问类型经过编码可适应小型表单元。表 19-55 展示了适用于此部分中访问类型的代码。

表 19-55. AUX_AIODIO 访问类型代码

访问类型	代码	说明
读取类型		
R	R	读取
写入类型		
W	W	写入
复位或默认值		

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-55. AUX_AIODIO 访问类型代码（续）

访问类型	代码	说明
-n		复位后的值或默认值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.2.1 IOMODE 寄存器（偏移 = 0h）[复位 = 00000000h]

图 19-48 展示了 IOMODE，表 19-56 中对此进行了介绍。

返回到汇总表。

输入输出模式
此寄存器控制由 AUX_AIODIO 实例 i 控制的 AUXIO 的上拉、下拉和输出模式。因此，在下面的公式中，i = 0 表示 AUX_AIODIO0，i = 1 表示 AUX_AIODIO1，以此类推。

图 19-48. IOMODE 寄存器

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
IO7		IO6		IO5		IO4		IO3		IO2		IO1		IO0	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

表 19-56. IOMODE 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-14	IO7	R/W	0h	选择 AUXIO[8i+7] 的模式。 0h = 输出模式： 当 IOPOE 的位 7 为 0 时：GPIODOUT 的位 7 驱动 AUXIO[8i+7]。 当 IOPOE 的位 7 为 1 时：IO7PSEL.SRC 选择的信号驱动 AUXIO[8i+7]。 1h = 输入模式： 当 GPIODIE 的位 7 为 0 时：使能 AUXIO[8i+7] 以进行模拟信号传输。 当 GPIODIE 的位 7 为 1 时：使能 AUXIO[8i+7]，以进行数字输入。 2h = 漏极开路模式： 当 IOPOE 的位 7 为 0 时： - 如果 GPIODOUT 的位 7 为 0：AUXIO[8i+7] 被驱动为低电平。 - 如果 GPIODOUT 的位 7 为 1：AUXIO[8i+7] 处于三态或被拉取。这取决于 IOC:IOCFGn.PULL_CTL。 当 IOPOE 的位 7 为 1 时： - 如果 IO7PSEL.SRC 选择的信号为 0：AUXIO[8i+7] 被驱动为低电平。 - 如果 IO7PSEL.SRC 选择的信号为 1：AUXIO[8i+7] 处于三态或被拉取。这取决于 IOC:IOCFGn.PULL_CTL。 3h = 开源模式： 当 IOPOE 的位 7 为 0 时： - 如果 GPIODOUT 的位 7 为 0：AUXIO[8i+7] 处于三态或被拉取。这取决于 IOC:IOCFGn.PULL_CTL。 - 如果 GPIODOUT 的位 7 为 1：AUXIO[8i+7] 被驱动为高电平。 当 IOPOE 的位 7 为 1 时： - 如果 IO7PSEL.SRC 选择的信号为 0：AUXIO[8i+7] 处于三态或被拉取。这取决于 IOC:IOCFGn.PULL_CTL。 - 如果 IO7PSEL.SRC 选择的信号为 1：AUXIO[8i+7] 被驱动为高电平。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-56. IOMODE 寄存器字段说明（续）

位	字段	类型	复位	说明
13-12	IO6	R/W	0h	选择 AUXIO[8i+6] 的模式。 0h = 输出模式： 当 IOPOE 的位 6 为 0 时：GPIODOUT 的位 6 驱动 AUXIO[8i+6]。 当 IOPOE 的位 6 为 1 时：IO6PSEL.SRC 选择的信号驱动 AUXIO[8i+6]。 1h = 输入模式： 当 GPIODIE 的位 6 为 0 时：使能 AUXIO[8i+6] 以进行模拟信号传输。 当 GPIODIE 的位 6 为 1 时：使能 AUXIO[8i+6]，以进行数字输入。 2h = 漏极开路模式： 当 IOPOE 的位 6 为 0 时： - 如果 GPIODOUT 的位 6 为 0：AUXIO[8i+6] 被驱动为低电平。 - 如果 GPIODOUT 的位 6 为 1：AUXIO[8i+6] 处于三态或被拉取。这取决于 IOC:IOCFGn.PULL_CTL。 当 IOPOE 的位 6 为 1 时： - 如果 IO6PSEL.SRC 选择的信号为 0：AUXIO[8i+6] 被驱动为低电平。 - 如果 IO6PSEL.SRC 选择的信号为 1：AUXIO[8i+6] 处于三态或被拉取。这取决于 IOC:IOCFGn.PULL_CTL。 3h = 开源模式： 当 IOPOE 的位 6 为 0 时： - 如果 GPIODOUT 的位 6 为 0：AUXIO[8i+6] 处于三态或被拉取。这取决于 IOC:IOCFGn.PULL_CTL。 - 如果 GPIODOUT 的位 6 为 1：AUXIO[8i+6] 被驱动为高电平。 当 IOPOE 的位 6 为 1 时： - 如果 IO6PSEL.SRC 选择的信号为 0：AUXIO[8i+6] 处于三态或被拉取。这取决于 IOC:IOCFGn.PULL_CTL。 - 如果 IO6PSEL.SRC 选择的信号为 1：AUXIO[8i+6] 被驱动为高电平。
11-10	IO5	R/W	0h	选择 AUXIO[8i+5] 的模式。 0h = 输出模式： 当 IOPOE 的位 5 为 0 时：GPIODOUT 的位 5 驱动 AUXIO[8i+5]。 当 IOPOE 的位 5 为 1 时：IO5PSEL.SRC 选择的信号驱动 AUXIO[8i+5]。 1h = 输入模式： 当 GPIODIE 的位 5 为 0 时：使能 AUXIO[8i+5] 以进行模拟信号传输。 当 GPIODIE 的位 5 为 1 时：使能 AUXIO[8i+5]，以进行数字输入。 2h = 漏极开路模式： 当 IOPOE 的位 5 为 0 时： - 如果 GPIODOUT 的位 5 为 0：AUXIO[8i+5] 被驱动为低电平。 - 如果 GPIODOUT 的位 5 为 1：AUXIO[8i+5] 处于三态或被拉取。这取决于 IOC:IOCFGn.PULL_CTL。 当 IOPOE 的位 5 为 1 时： - 如果 IO5PSEL.SRC 选择的信号为 0：AUXIO[8i+5] 被驱动为低电平。 - 如果 IO5PSEL.SRC 选择的信号为 1：AUXIO[8i+5] 处于三态或被拉取。这取决于 IOC:IOCFGn.PULL_CTL。 3h = 开源模式： 当 IOPOE 的位 5 为 0 时： - 如果 GPIODOUT 的位 5 为 0：AUXIO[8i+5] 处于三态或被拉取。这取决于 IOC:IOCFGn.PULL_CTL。 - 如果 GPIODOUT 的位 5 为 1：AUXIO[8i+5] 被驱动为高电平。 当 IOPOE 的位 5 为 1 时： - 如果 IO5PSEL.SRC 选择的信号为 0：AUXIO[8i+5] 处于三态或被拉取。这取决于 IOC:IOCFGn.PULL_CTL。 - 如果 IO5PSEL.SRC 选择的信号为 1：AUXIO[8i+5] 被驱动为高电平。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-56. IOMODE 寄存器字段说明（续）

位	字段	类型	复位	说明
9-8	IO4	R/W	0h	选择 AUXIO[8i+4] 的模式。 0h = 输出模式： 当 IOPOE 的位 4 为 0 时：GPIODOUT 的位 4 驱动 AUXIO[8i+4]。 当 IOPOE 的位 4 为 1 时：IO4PSEL.SRC 选择的信号驱动 AUXIO[8i+4]。 1h = 输入模式： 当 GPIODIE 的位 4 为 0 时：使能 AUXIO[8i+4] 以进行模拟信号传输。 当 GPIODIE 的位 4 为 1 时：使能 AUXIO[8i+4]，以进行数字输入。 2h = 漏极开路模式： 当 IOPOE 的位 4 为 0 时： - 如果 GPIODOUT 的位 4 为 0：AUXIO[8i+4] 被驱动为低电平。 - 如果 GPIODOUT 的位 4 为 1：AUXIO[8i+4] 处于三态或被拉取。这取决于 IOC:IOCFGn.PULL_CTL。 当 IOPOE 的位 4 为 1 时： - 如果 IO4PSEL.SRC 选择的信号为 0：AUXIO[8i+4] 被驱动为低电平。 - 如果 IO4PSEL.SRC 选择的信号为 1：AUXIO[8i+4] 处于三态或被拉取。这取决于 IOC:IOCFGn.PULL_CTL。 3h = 开源模式： 当 IOPOE 的位 4 为 0 时： - 如果 GPIODOUT 的位 4 为 0：AUXIO[8i+4] 处于三态或被拉取。这取决于 IOC:IOCFGn.PULL_CTL。 - 如果 GPIODOUT 的位 4 为 1：AUXIO[8i+4] 被驱动为高电平。 当 IOPOE 的位 4 为 1 时： - 如果 IO4PSEL.SRC 选择的信号为 0：AUXIO[8i+4] 处于三态或被拉取。这取决于 IOC:IOCFGn.PULL_CTL。 - 如果 IO4PSEL.SRC 选择的信号为 1：AUXIO[8i+4] 被驱动为高电平。
7-6	IO3	R/W	0h	选择 AUXIO[8i+3] 的模式。 0h = 输出模式： 当 IOPOE 的位 3 为 0 时：GPIODOUT 的位 3 驱动 AUXIO[8i+3]。 当 IOPOE 的位 3 为 1 时：IO3PSEL.SRC 选择的信号驱动 AUXIO[8i+3]。 1h = 输入模式： 当 GPIODIE 的位 3 为 0 时：使能 AUXIO[8i+3] 以进行模拟信号传输。 当 GPIODIE 的位 3 为 1 时：使能 AUXIO[8i+3]，以进行数字输入。 2h = 漏极开路模式： 当 IOPOE 的位 3 为 0 时： - 如果 GPIODOUT 的位 3 为 0：AUXIO[8i+3] 被驱动为低电平。 - 如果 GPIODOUT 的位 3 为 1：AUXIO[8i+3] 处于三态或被拉取。这取决于 IOC:IOCFGn.PULL_CTL。 当 IOPOE 的位 3 为 1 时： - 如果 IO3PSEL.SRC 选择的信号为 0：AUXIO[8i+3] 被驱动为低电平。 - 如果 IO3PSEL.SRC 选择的信号为 1：AUXIO[8i+3] 处于三态或被拉取。这取决于 IOC:IOCFGn.PULL_CTL。 3h = 开源模式： 当 IOPOE 的位 3 为 0 时： - 如果 GPIODOUT 的位 3 为 0：AUXIO[8i+3] 处于三态或被拉取。这取决于 IOC:IOCFGn.PULL_CTL。 - 如果 GPIODOUT 的位 3 为 1：AUXIO[8i+3] 被驱动为高电平。 当 IOPOE 的位 3 为 1 时： - 如果 IO3PSEL.SRC 选择的信号为 0：AUXIO[8i+3] 处于三态或被拉取。这取决于 IOC:IOCFGn.PULL_CTL。 - 如果 IO3PSEL.SRC 选择的信号为 1：AUXIO[8i+3] 被驱动为高电平。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-56. IOMODE 寄存器字段说明（续）

位	字段	类型	复位	说明
5-4	IO2	R/W	0h	选择 AUXIO[8i+2] 的模式。 0h = 输出模式： 当 IOPOE 的位 2 为 0 时：GPIODOUT 的位 2 驱动 AUXIO[8i+2]。 当 IOPOE 的位 2 为 1 时：IO2PSEL.SRC 选择的信号驱动 AUXIO[8i+2]。 1h = 输入模式： 当 GPIODIE 的位 2 为 0 时：使能 AUXIO[8i+2] 以进行模拟信号传输。 当 GPIODIE 的位 2 为 1 时：使能 AUXIO[8i+2]，以进行数字输入。 2h = 漏极开路模式： 当 IOPOE 的位 2 为 0 时： - 如果 GPIODOUT 的位 2 为 0：AUXIO[8i+2] 被驱动为低电平。 - 如果 GPIODOUT 的位 2 为 1：AUXIO[8i+2] 处于三态或被拉取。这取决于 IOC:IOCFGn.PULL_CTL。 当 IOPOE 的位 2 为 1 时： - 如果 IO2PSEL.SRC 选择的信号为 0：AUXIO[8i+2] 被驱动为低电平。 - 如果 IO2PSEL.SRC 选择的信号为 1：AUXIO[8i+2] 处于三态或被拉取。这取决于 IOC:IOCFGn.PULL_CTL。 3h = 开源模式： 当 IOPOE 的位 2 为 0 时： - 如果 GPIODOUT 的位 2 为 0：AUXIO[8i+2] 处于三态或被拉取。这取决于 IOC:IOCFGn.PULL_CTL。 - 如果 GPIODOUT 的位 2 为 1：AUXIO[8i+2] 被驱动为高电平。 当 IOPOE 的位 2 为 1 时： - 如果 IO2PSEL.SRC 选择的信号为 0：AUXIO[8i+2] 处于三态或被拉取。这取决于 IOC:IOCFGn.PULL_CTL。 - 如果 IO2PSEL.SRC 选择的信号为 1：AUXIO[8i+2] 被驱动为高电平。
3-2	IO1	R/W	0h	选择 AUXIO[8i+1] 的模式。 0h = 输出模式： 当 IOPOE 的位 1 为 0 时：GPIODOUT 的位 1 驱动 AUXIO[8i+1]。 当 IOPOE 的位 1 为 1 时：IO1PSEL.SRC 选择的信号驱动 AUXIO[8i+1]。 1h = 输入模式： 当 GPIODIE 的位 1 为 0 时：使能 AUXIO[8i+1] 以进行模拟信号传输。 当 GPIODIE 的位 1 为 1 时：使能 AUXIO[8i+1]，以进行数字输入。 2h = 漏极开路模式： 当 IOPOE 的位 1 为 0 时： - 如果 GPIODOUT 的位 1 为 0：AUXIO[8i+1] 被驱动为低电平。 - 如果 GPIODOUT 的位 1 为 1：AUXIO[8i+1] 处于三态或被拉取。这取决于 IOC:IOCFGn.PULL_CTL。 当 IOPOE 的位 1 为 1 时： - 如果 IO1PSEL.SRC 选择的信号为 0：AUXIO[8i+1] 被驱动为低电平。 - 如果 IO1PSEL.SRC 选择的信号为 1：AUXIO[8i+1] 处于三态或被拉取。这取决于 IOC:IOCFGn.PULL_CTL。 3h = 开源模式： 当 IOPOE 的位 1 为 0 时： - 如果 GPIODOUT 的位 1 为 0：AUXIO[8i+1] 处于三态或被拉取。这取决于 IOC:IOCFGn.PULL_CTL。 - 如果 GPIODOUT 的位 1 为 1：AUXIO[8i+1] 被驱动为高电平。 当 IOPOE 的位 1 为 1 时： - 如果 IO1PSEL.SRC 选择的信号为 0：AUXIO[8i+1] 处于三态或被拉取。这取决于 IOC:IOCFGn.PULL_CTL。 - 如果 IO1PSEL.SRC 选择的信号为 1：AUXIO[8i+1] 被驱动为高电平。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-56. IOMODE 寄存器字段说明（续）

位	字段	类型	复位	说明
1-0	IO0	R/W	0h	选择 AUXIO[8i+0] 的模式。 0h = 输出模式： 当 IOPOE 的位 0 为 0 时：GPIODOUT 的位 0 驱动 AUXIO[8i+0]。 当 IOPOE 的位 0 为 1 时：IO0PSEL.SRC 选择的信号驱动 AUXIO[8i+0]。 1h = 输入模式： 当 GPIODIE 的位 0 为 0 时：使能 AUXIO[8i+0] 以进行模拟信号传输。 当 GPIODIE 的位 0 为 1 时：使能 AUXIO[8i+0]，以进行数字输入。 2h = 漏极开路模式： 当 IOPOE 的位 0 为 0 时： - 如果 GPIODOUT 的位 0 为 0：AUXIO[8i+0] 被驱动为低电平。 - 如果 GPIODOUT 的位 0 为 1：AUXIO[8i+0] 处于三态或被拉取。这取决于 IOC:IOCFGn.PULL_CTL。 当 IOPOE 的位 0 为 1 时： - 如果 IO0PSEL.SRC 选择的信号为 0：AUXIO[8i+0] 被驱动为低电平。 - 如果 IO0PSEL.SRC 选择的信号为 1：AUXIO[8i+0] 处于三态或被拉取。这取决于 IOC:IOCFGn.PULL_CTL。 3h = 开源模式： 当 IOPOE 的位 0 为 0 时： - 如果 GPIODOUT 的位 0 为 0：AUXIO[8i+0] 处于三态或被拉取。这取决于 IOC:IOCFGn.PULL_CTL。 - 如果 GPIODOUT 的位 0 为 1：AUXIO[8i+0] 被驱动为高电平。 当 IOPOE 的位 0 为 1 时： - 如果 IO0PSEL.SRC 选择的信号为 0：AUXIO[8i+0] 处于三态或被拉取。这取决于 IOC:IOCFGn.PULL_CTL。 - 如果 IO0PSEL.SRC 选择的信号为 1：AUXIO[8i+0] 被驱动为高电平。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.2.2 GPIODIE 寄存器（偏移 = 4h）[复位 = 00000000h]

图 19-49 展示了 GPIODIE，表 19-57 中对此进行了介绍。

返回到汇总表。

通用输入输出数字输入使能
该寄存器控制由 AUX_AIODIO 的实例 i 控制的 AUXIO 的输入缓冲器。因此，在下面的公式中，i = 0 表示 AUX_AIODIO0，i = 1 表示 AUX_AIODIO1，以此类推。

图 19-49. GPIODIE 寄存器

31	30	29	28	27	26	25	24	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																								IO7_0							
R-0h																								R/W-0h							

表 19-57. GPIODIE 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-0	IO7_0	R/W	0h	向此位向量中的位索引 n 写入 1，以使能 AUXIO[8i+n] 的数字输入缓冲器。 向此位向量中的位索引 n 写入 0 以禁用 AUXIO[8i+n] 的数字输入缓冲器。 必须使能 AUXIO[8i+n] 的数字输入缓冲器才能读取 GPIODIN 中的引脚值。 您必须禁用模拟输入或悬空引脚的数字输入缓冲器以避免漏电流。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.2.3 IOPOE 寄存器（偏移 = 8h）[复位 = 00000000h]

图 19-50 展示了 IOPOE，表 19-58 中对此进行了介绍。

返回到汇总表。

输入输出外设输出使能
该寄存器选择由 AUX_AIODIO 的实例 i 控制的 AUXIO 的输出源。因此，在下面的公式中，i = 0 表示 AUX_AIODIO0，i = 1 表示 AUX_AIODIO1，以此类推。

图 19-50. IOPOE 寄存器

31	30	29	28	27	26	25	24	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																								IO7_0							
R-0h																								R/W-0h							

表 19-58. IOPOE 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-0	IO7_0	R/W	0h	向此位向量中的位索引 n 写入 1，以将 AUXIO[8i+n] 配置为从 [IOnPSEL.*] 中给定的源驱动。 向此位向量中的位索引 n 写入 0，以将 AUXIO[8i+n] 配置为从 GPIODOUT 中的位 n 驱动。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.2.4 GPIODOUT 寄存器（偏移 = Ch）[复位 = 00000000h]

图 19-51 展示了 GPIODOUT，表 19-59 中对此进行了介绍。

返回到汇总表。

通用输入输出数据输出
此输出数据寄存器用于设置由 AUX_AIODIO 的实例 i 控制的 AUXIO 上的数据。因此，在下面的公式中，i = 0 表示 AUX_AIODIO0，i = 1 表示 AUX_AIODIO1，以此类推。

图 19-51. GPIODOUT 寄存器

31	30	29	28	27	26	25	24	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																								IO7_0							
R-0h																								R/W-0h							

表 19-59. GPIODOUT 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-0	IO7_0	R/W	0h	向此位向量中的位索引 n 写入 1 以设置 AUXIO[8i+n]。 向此位向量中的位索引 n 写入 0 以清除 AUXIO[8i+n]。 必须清除 IOPOE 中的位 n 才能将此位向量中的位 n 连接到 AUXIO[8i+n]。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.2.5 GPIODIN 寄存器（偏移 = 10h）[复位 = 00000000h]

图 19-52 展示了 GPIODIN，表 19-60 中对此进行了介绍。

返回到汇总表。

通用输入输出数据输入
此寄存器为由 AUX_AIODIO 的实例 i 控制的 AUXIO 提供同步输入数据。因此，在下面的公式中，i = 0 表示 AUX_AIODIO0，i = 1 表示 AUX_AIODIO1，依此类推

图 19-52. GPIODIN 寄存器

31	30	29	28	27	26	25	24	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																								IO7_0							
R-0h																								R-0h							

表 19-60. GPIODIN 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-0	IO7_0	R	0h	设置 GPIODIE 的位 n 时，此位向量中的位 n 包含 AUXIO[8i+n] 的值。否则，位 n 读取为 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.2.6 GPIODOUTSET 寄存器（偏移 = 14h）[复位 = 00000000h]

图 19-53 展示了 GPIODOUTSET，表 19-61 中对此进行了介绍。

返回到汇总表。

通用输入输出数据输出
设置 AUX_AIODIO 实例 i 中 GPIODOUT 的位。因此，在下面的公式中，i = 0 表示 AUX_AIODIO0，i = 1 表示 AUX_AIODIO1，以此类推。

图 19-53. GPIODOUTSET 寄存器

31	30	29	28	27	26	25	24	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																								IO7_0							
R-0h																								R/W-0h							

表 19-61. GPIODOUTSET 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-0	IO7_0	R/W	0h	向此位向量中的位索引 n 写入 1 以设置 GPIODOUT 的位 n。读取值为 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.2.7 GPIODOUTCLR 寄存器（偏移 = 18h）[复位 = 00000000h]

图 19-54 展示了 GPIODOUTCLR，表 19-62 中对此进行了介绍。

返回到汇总表。

通用输入输出数据输出清除
清除 AUX_AIODIO 实例 I 中 GPIODOUT 的位。因此，在下面的公式中，i = 0 表示 AUX_AIODIO0，i = 1 表示 AUX_AIODIO1，以此类推。

图 19-54. GPIODOUTCLR 寄存器

31	30	29	28	27	26	25	24	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																								IO7_0							
R-0h																								R/W-0h							

表 19-62. GPIODOUTCLR 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-0	IO7_0	R/W	0h	向此位向量中的位索引 n 写入 1 以清除 GPIODOUT 的位 n。 读取值为 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.2.8 GPIODOUTTGL 寄存器（偏移 = 1Ch）[复位 = 00000000h]

图 19-55 展示了 GPIODOUTTGL，表 19-63 中对此进行了介绍。

返回到汇总表。

通用输入输出数据输出切换
切换 AUX_AIODIO 实例 i 中 GPIODOUT 的位。因此，在下面的公式中，i = 0 表示 AUX_AIODIO0，i = 1 表示 AUX_AIODIO1，以此类推。

图 19-55. GPIODOUTTGL 寄存器

31	30	29	28	27	26	25	24	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																								IO7_0							
R-0h																								R/W-0h							

表 19-63. GPIODOUTTGL 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-0	IO7_0	R/W	0h	向此位向量中的位索引 n 写入 1 以切换 GPIODOUT 的位 n。 读取值为 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.2.9 IO0PSEL 寄存器（偏移 = 20h）[复位 = 00000000h]

图 19-56 展示了 IO0PSEL，表 19-64 对其进行了介绍。

返回到汇总表。

输入输出 0 外设选择
当 IOPOE 的位 0 为 1 时，该寄存器选择连接到 AUXIO[8i+0] 的外设信号。
为避免 AUXIO[8i+0] 上出现干扰，必须在 IOPOE 的位 0 为 0 时配置此寄存器。
在下列公式中，i = 0 表示 AUX_AIODIO0，i = 1 表示 AUX_AIODIO1，依此类推。

图 19-56. IO0PSEL 寄存器

31	30	29	28	27	26	25	24	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																													SRC		
R-0h																													R/W-0h		

表 19-64. IO0PSEL 寄存器字段说明

位	字段	类型	复位	说明
31-3	RESERVED	R	0h	保留
2-0	SRC	R/W	0h	设置 IOPOE 的位 0 时，选择连接到 AUXIO[8i+0] 的外设信号。 0h = 外设输出多路复用器选择 AUX_EVCTL:EVOBSCFG 选择的事件 1h = 外设输出多路复用器选择 AUX_SPIM SCLK。 2h = 外设输出多路复用器选择 AUX_SPIM MOSI。 3h = 外设输出多路复用器选择 AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV0 的异步版本。 4h = 外设输出多路复用器选择 AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV1 的异步版本。 5h = 外设输出多路复用器选择 AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV2 的异步版本。 6h = 外设输出多路复用器选择 AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV3 的异步版本。 7h = 外设输出多路复用器选择 AUX_EVCTL:EVSTAT3.AUX_TIMER2_PULSE 的异步版本。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.2.10 IO1PSEL 寄存器（偏移 = 24h）[复位 = 00000000h]

图 19-57 展示了 IO1PSEL，表 19-65 对其进行了介绍。

返回到汇总表。

输入输出 1 外设选择
当 IOPOE 的位 1 为 1 时，该寄存器选择连接到 AUXIO[8i+1] 的外设信号。
为避免 AUXIO[8i+1] 上出现干扰，必须在 IOPOE 的位 1 为 0 时配置此寄存器。
在下列公式中，i = 0 表示 AUX_AIODIO0，i = 1 表示 AUX_AIODIO1，依此类推。

图 19-57. IO1PSEL 寄存器

31	30	29	28	27	26	25	24	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																													SRC		
R-0h																													R/W-0h		

表 19-65. IO1PSEL 寄存器字段说明

位	字段	类型	复位	说明
31-3	RESERVED	R	0h	保留
2-0	SRC	R/W	0h	设置 IOPOE 的位 1 时，选择连接到 AUXIO[8i+1] 的外设信号。 0h = 外设输出多路复用器选择 AUX_EVCTL:EVOBSCFG 选择的事件 1h = 外设输出多路复用器选择 AUX_SPIM SCLK。 2h = 外设输出多路复用器选择 AUX_SPIM MOSI。 3h = 外设输出多路复用器选择 AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV0 的异步版本。 4h = 外设输出多路复用器选择 AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV1 的异步版本。 5h = 外设输出多路复用器选择 AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV2 的异步版本。 6h = 外设输出多路复用器选择 AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV3 的异步版本。 7h = 外设输出多路复用器选择 AUX_EVCTL:EVSTAT3.AUX_TIMER2_PULSE 的异步版本。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.2.11 IO2PSEL 寄存器（偏移 = 28h）[复位 = 00000000h]

图 19-58 展示了 IO2PSEL，表 19-66 对其进行了介绍。

返回到汇总表。

输入输出 2 外设选择
当 IOPOE 的位 2 为 1 时，该寄存器选择连接到 AUXIO[8i+2] 的外设信号。
为避免 AUXIO[8i+2] 上出现干扰，必须在 IOPOE 的位 2 为 0 时配置此寄存器。
在下列公式中，i = 0 表示 AUX_AIODIO0，i = 1 表示 AUX_AIODIO1，依此类推。

图 19-58. IO2PSEL 寄存器

31	30	29	28	27	26	25	24	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																													SRC		
R-0h																													R/W-0h		

表 19-66. IO2PSEL 寄存器字段说明

位	字段	类型	复位	说明
31-3	RESERVED	R	0h	保留
2-0	SRC	R/W	0h	设置 IOPOE 的位 2 时，选择连接到 AUXIO[8i+2] 的外设信号。 0h = 外设输出多路复用器选择 AUX_EVCTL:EVOBSCFG 选择的事件 1h = 外设输出多路复用器选择 AUX_SPIM SCLK。 2h = 外设输出多路复用器选择 AUX_SPIM MOSI。 3h = 外设输出多路复用器选择 AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV0 的异步版本。 4h = 外设输出多路复用器选择 AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV1 的异步版本。 5h = 外设输出多路复用器选择 AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV2 的异步版本。 6h = 外设输出多路复用器选择 AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV3 的异步版本。 7h = 外设输出多路复用器选择 AUX_EVCTL:EVSTAT3.AUX_TIMER2_PULSE 的异步版本。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.2.12 IO3PSEL 寄存器（偏移 = 2Ch）[复位 = 00000000h]

图 19-59 展示了 IO3PSEL，表 19-67 对其进行了介绍。

返回到汇总表。

输入输出 3 外设选择
当 IOPOE 的位 3 为 1 时，该寄存器选择连接到 AUXIO[8i+3] 的外设信号。
为避免 AUXIO[8i+3] 上出现干扰，必须在 IOPOE 的位 3 为 0 时配置此寄存器。
在下列公式中，i = 0 表示 AUX_AIODIO0，i = 1 表示 AUX_AIODIO1，依此类推。

图 19-59. IO3PSEL 寄存器

31	30	29	28	27	26	25	24	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																													SRC		
R-0h																													R/W-0h		

表 19-67. IO3PSEL 寄存器字段说明

位	字段	类型	复位	说明
31-3	RESERVED	R	0h	保留
2-0	SRC	R/W	0h	设置 IOPOE 的位 3 时，选择连接到 AUXIO[8i+3] 的外设信号。 0h = 外设输出多路复用器选择 AUX_EVCTL:EVOBSCFG 选择的事件 1h = 外设输出多路复用器选择 AUX_SPIM SCLK。 2h = 外设输出多路复用器选择 AUX_SPIM MOSI。 3h = 外设输出多路复用器选择 AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV0 的异步版本。 4h = 外设输出多路复用器选择 AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV1 的异步版本。 5h = 外设输出多路复用器选择 AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV2 的异步版本。 6h = 外设输出多路复用器选择 AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV3 的异步版本。 7h = 外设输出多路复用器选择 AUX_EVCTL:EVSTAT3.AUX_TIMER2_PULSE 的异步版本。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.2.13 IO4PSEL 寄存器（偏移 = 30h）[复位 = 00000000h]

图 19-60 展示了 IO4PSEL，表 19-68 对其进行了介绍。

返回到汇总表。

输入输出 4 外设选择
当 IOPOE 的位 4 为 1 时，该寄存器选择连接到 AUXIO[8i+4] 的外设信号。
为避免 AUXIO[8i+4] 上出现干扰，必须在 IOPOE 的位 4 为 0 时配置此寄存器。
在下列公式中，i = 0 表示 AUX_AIODIO0，i = 1 表示 AUX_AIODIO1，依此类推。

图 19-60. IO4PSEL 寄存器

31	30	29	28	27	26	25	24	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																													SRC		
R-0h																													R/W-0h		

表 19-68. IO4PSEL 寄存器字段说明

位	字段	类型	复位	说明
31-3	RESERVED	R	0h	保留
2-0	SRC	R/W	0h	设置 IOPOE 的位 4 时，选择连接到 AUXIO[8i+4] 的外设信号。 0h = 外设输出多路复用器选择 AUX_EVCTL:EVOBSCFG 选择的事件 1h = 外设输出多路复用器选择 AUX_SPIM SCLK。 2h = 外设输出多路复用器选择 AUX_SPIM MOSI。 3h = 外设输出多路复用器选择 AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV0 的异步版本。 4h = 外设输出多路复用器选择 AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV1 的异步版本。 5h = 外设输出多路复用器选择 AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV2 的异步版本。 6h = 外设输出多路复用器选择 AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV3 的异步版本。 7h = 外设输出多路复用器选择 AUX_EVCTL:EVSTAT3.AUX_TIMER2_PULSE 的异步版本。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.2.14 IO5PSEL 寄存器（偏移 = 34h）[复位 = 00000000h]

图 19-61 展示了 IO5PSEL，表 19-69 对其进行了介绍。

返回到汇总表。

输入输出 5 外设选择
当 IOPOE 的位 5 为 1 时，该寄存器选择连接到 AUXIO[8i+5] 的外设信号。
为避免 AUXIO[8i+5] 上出现干扰，必须在 IOPOE 的位 5 为 0 时配置此寄存器。
在下列公式中，i = 0 表示 AUX_AIODIO0，i = 1 表示 AUX_AIODIO1，依此类推。

图 19-61. IO5PSEL 寄存器

31	30	29	28	27	26	25	24	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																													SRC		
R-0h																													R/W-0h		

表 19-69. IO5PSEL 寄存器字段说明

位	字段	类型	复位	说明
31-3	RESERVED	R	0h	保留
2-0	SRC	R/W	0h	设置 IOPOE 的位 5 时，选择连接到 AUXIO[8i+5] 的外设信号。 0h = 外设输出多路复用器选择 AUX_EVCTL:EVOBSCFG 选择的事件 1h = 外设输出多路复用器选择 AUX_SPIM SCLK。 2h = 外设输出多路复用器选择 AUX_SPIM MOSI。 3h = 外设输出多路复用器选择 AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV0 的异步版本。 4h = 外设输出多路复用器选择 AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV1 的异步版本。 5h = 外设输出多路复用器选择 AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV2 的异步版本。 6h = 外设输出多路复用器选择 AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV3 的异步版本。 7h = 外设输出多路复用器选择 AUX_EVCTL:EVSTAT3.AUX_TIMER2_PULSE 的异步版本。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.2.15 IO6PSEL 寄存器（偏移 = 38h）[复位 = 00000000h]

图 19-62 展示了 IO6PSEL，表 19-70 对其进行了介绍。

返回到汇总表。

输入输出 6 外设选择
当 IOPOE 的位 6 为 1 时，该寄存器选择连接到 AUXIO[8i+6] 的外设信号。
为避免 AUXIO[8i+6] 上出现干扰，必须在 IOPOE 的位 6 为 0 时配置此寄存器。
在下列公式中，i = 0 表示 AUX_AIODIO0，i = 1 表示 AUX_AIODIO1，依此类推。

图 19-62. IO6PSEL 寄存器

31	30	29	28	27	26	25	24	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																													SRC		
R-0h																													R/W-0h		

表 19-70. IO6PSEL 寄存器字段说明

位	字段	类型	复位	说明
31-3	RESERVED	R	0h	保留
2-0	SRC	R/W	0h	设置 IOPOE 的位 6 时，选择连接到 AUXIO[8i+6] 的外设信号。 0h = 外设输出多路复用器选择 AUX_EVCTL:EVOBSCFG 选择的事件 1h = 外设输出多路复用器选择 AUX_SPIM SCLK。 2h = 外设输出多路复用器选择 AUX_SPIM MOSI。 3h = 外设输出多路复用器选择 AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV0 的异步版本。 4h = 外设输出多路复用器选择 AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV1 的异步版本。 5h = 外设输出多路复用器选择 AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV2 的异步版本。 6h = 外设输出多路复用器选择 AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV3 的异步版本。 7h = 外设输出多路复用器选择 AUX_EVCTL:EVSTAT3.AUX_TIMER2_PULSE 的异步版本。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.2.16 IO7PSEL 寄存器（偏移 = 3Ch）[复位 = 00000000h]

图 19-63 展示了 IO7PSEL，表 19-71 对其进行了介绍。

返回到汇总表。

输入输出 7 外设选择
当 IOPOE 的位 7 为 1 时，该寄存器选择连接到 AUXIO[8i+7] 的外设信号。
为避免 AUXIO[8i+7] 上出现干扰，必须在 IOPOE 的位 7 为 0 时配置此寄存器。
在下列公式中，i = 0 表示 AUX_AIODIO0，i = 1 表示 AUX_AIODIO1，依此类推。

图 19-63. IO7PSEL 寄存器

31	30	29	28	27	26	25	24	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																													SRC		
R-0h																													R/W-0h		

表 19-71. IO7PSEL 寄存器字段说明

位	字段	类型	复位	说明
31-3	RESERVED	R	0h	保留
2-0	SRC	R/W	0h	设置 IOPOE 的位 7 时，选择连接到 AUXIO[8i+7] 的外设信号。 0h = 外设输出多路复用器选择 AUX_EVCTL:EVOBSCFG 选择的事件 1h = 外设输出多路复用器选择 AUX_SPIM SCLK。 2h = 外设输出多路复用器选择 AUX_SPIM MOSI。 3h = 外设输出多路复用器选择 AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV0 的异步版本。 4h = 外设输出多路复用器选择 AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV1 的异步版本。 5h = 外设输出多路复用器选择 AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV2 的异步版本。 6h = 外设输出多路复用器选择 AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV3 的异步版本。 7h = 外设输出多路复用器选择 AUX_EVCTL:EVSTAT3.AUX_TIMER2_PULSE 的异步版本。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.2.17 IOMODEL 寄存器（偏移 = 40h）[复位 = 00000000h]

图 19-64 展示了 IOMODEL，表 19-72 中对此进行了介绍。

返回到汇总表。

输入输出模式低电平
这是 IOMODE.IO0 至 IOMODE.IO3 的别名寄存器。

图 19-64. IOMODEL 寄存器

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								IO3		IO2		IO1		IO0	
R-0h								R/W-0h		R/W-0h		R/W-0h		R/W-0h	

表 19-72. IOMODEL 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-6	IO3	R/W	0h	参阅 IOMODE.IO3。
5-4	IO2	R/W	0h	参阅 IOMODE.IO2。
3-2	IO1	R/W	0h	参阅 IOMODE.IO1。
1-0	IO0	R/W	0h	参阅 IOMODE.IO0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.2.18 IOMODEH 寄存器（偏移 = 44h）[复位 = 00000000h]

图 19-65 展示了 IOMODEH，表 19-73 中对此进行了介绍。

返回到汇总表。

输入输出模式高电平
这是 IOMODE.IO4 到 IOMODE.IO7 的别名寄存器。

图 19-65. IOMODEH 寄存器

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								IO7		IO6		IO5		IO4	
R-0h								R/W-0h		R/W-0h		R/W-0h		R/W-0h	

表 19-73. IOMODEH 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-6	IO7	R/W	0h	参阅 IOMODE.IO7。
5-4	IO6	R/W	0h	参阅 IOMODE.IO6。
3-2	IO5	R/W	0h	参阅 IOMODE.IO5。
1-0	IO4	R/W	0h	参阅 IOMODE.IO4。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.3 AUX_EVCTL 寄存器

表 19-74 列出了 AUX_EVCTL 寄存器的存储器映射寄存器。表 19-74 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

表 19-74. AUX_EVCTL 寄存器

偏移	首字母缩写词	寄存器名称	部分
0h	EVSTAT0	事件状态 0	EVSTAT0 寄存器 (偏移 = 0h) [复位 = 00000000h]
4h	EVSTAT1	事件状态 1	EVSTAT1 寄存器 (偏移 = 4h) [复位 = 00000000h]
8h	EVSTAT2	事件状态 2	EVSTAT2 寄存器 (偏移 = 8h) [复位 = 00000000h]
Ch	EVSTAT3	事件状态 3	EVSTAT3 寄存器 (偏移 = Ch) [复位 = 00000000h]
10h	SCEWEVCFG0	传感器控制器引擎等待事件配置 0	SCEWEVCFG0 寄存器 (偏移 = 10h) [复位 = 00000000h]
14h	SCEWEVCFG1	传感器控制器引擎等待事件配置 1	SCEWEVCFG1 寄存器 (偏移 = 14h) [复位 = 00000000h]
18h	DMACTL	直接存储器存取控制	DMACTL 寄存器 (偏移 = 18h) [复位 = 00000000h]
20h	SWEVSET	软件事件集	SWEVSET 寄存器 (偏移 = 20h) [复位 = 00000000h]
24h	EVTOAONFLAGS	事件至 AON 标志	EVTOAONFLAGS 寄存器 (偏移 = 24h) [复位 = 00000000h]
28h	EVTOAONPOL	事件至 AON 极性	EVTOAONPOL 寄存器 (偏移 = 28h) [复位 = 00000000h]
2Ch	EVTOAONFLAGSCLR	事件至 AON 清除	EVTOAONFLAGSCLR 寄存器 (偏移 = 2Ch) [复位 = 00000000h]
30h	EVTOMCUFLAGS	事件至 MCU 标志	EVTOMCUFLAGS 寄存器 (偏移 = 30h) [复位 = 00000000h]
34h	EVTOMCUPOL	事件至 MCU 极性	EVTOMCUPOL 寄存器 (偏移 = 34h) [复位 = 00000000h]
38h	EVTOMCUFLAGSCLR	事件至 MCU 标志清除	EVTOMCUFLAGSCLR 寄存器 (偏移 = 38h) [复位 = 00000000h]
3Ch	COMBEVTOMCUMASK	组合事件至 MCU 屏蔽	COMBEVTOMCUMASK 寄存器 (偏移 = 3Ch) [复位 = 00000000h]
40h	EVOBSCFG	事件观测配置	EVOBSCFG 寄存器 (偏移 = 40h) [复位 = 00000000h]
44h	PROGDLY	可编程延迟	PROGDLY 寄存器 (偏移 = 44h) [复位 = 00000000h]
48h	手动	手动	MANUAL 寄存器 (偏移 = 48h) [复位 = 00000000h]
4Ch	EVSTAT0L	事件状态 0 低字节	EVSTAT0L 寄存器 (偏移 = 4Ch) [复位 = 00000000h]
50h	EVSTAT0H	事件状态 0 高字节	EVSTAT0H 寄存器 (偏移 = 50h) [复位 = 00000000h]
54h	EVSTAT1L	事件状态 1 低字节	EVSTAT1L 寄存器 (偏移 = 54h) [复位 = 00000000h]
58h	EVSTAT1H	事件状态 1 高字节	EVSTAT1H 寄存器 (偏移 = 58h) [复位 = 00000000h]
5Ch	EVSTAT2L	事件状态 2 低字节	EVSTAT2L 寄存器 (偏移 = 5Ch) [复位 = 00000000h]

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-74. AUX_EVCTL 寄存器（续）

偏移	首字母缩写词	寄存器名称	部分
60h	EVSTAT2H	事件状态 2 高字节	EVSTAT2H 寄存器（偏移 = 60h）[复位 = 00000000h]
64h	EVSTAT3L	事件状态 3 低字节	EVSTAT3L 寄存器（偏移 = 64h）[复位 = 00000000h]
68h	EVSTAT3H	事件状态 3 高字节	EVSTAT3H 寄存器（偏移 = 68h）[复位 = 00000000h]

复杂的位访问类型经过编码可适应小型表单元。表 19-75 展示了适用于此部分中访问类型的代码。

表 19-75. AUX_EVCTL 访问类型代码

访问类型	代码	说明
读取类型		
R	R	读取
写入类型		
W	W	写入
W0C	W 0C	写入 0 以清零
复位或默认值		
-n		复位后的值或默认值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.3.1 EVSTAT0 寄存器（偏移 = 0h）[复位 = 00000000h]

图 19-66 展示了 EVSTAT0，表 19-76 对其进行了介绍。

返回到汇总表。

事件状态 0
寄存器保存与 AUX 时钟同步的 64 位事件总线的事件 0 至 15。除非另有说明，否则通过此寄存器读取的所有事件都以 SCE 时钟速率同步。以下订阅者使用此寄存器中的异步版本事件。
- AUX_TIMER2。
- AUX_ANAIF。
- AUX_TDC。
- AUX_SYSIF。
- AUX_AIODIOn。
- EVOBSCFG。

图 19-66. EVSTAT0 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
AUXIO15	AUXIO14	AUXIO13	AUXIO12	AUXIO11	AUXIO10	AUXIO9	AUXIO8
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
AUXIO7	AUXIO6	AUXIO5	AUXIO4	AUXIO3	AUXIO2	AUXIO1	AUXIO0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

表 19-76. EVSTAT0 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15	AUXIO15	R	0h	AUXIO15 引脚电平，读取值对应于 AUX_AIODIO1:GPIODIN 位 7。
14	AUXIO14	R	0h	AUXIO14 引脚电平，读取值对应于 AUX_AIODIO1:GPIODIN 位 6。
13	AUXIO13	R	0h	AUXIO13 引脚电平，读取值对应于 AUX_AIODIO1:GPIODIN 位 5。
12	AUXIO12	R	0h	AUXIO12 引脚电平，读取值对应于 AUX_AIODIO1:GPIODIN 位 4。
11	AUXIO11	R	0h	AUXIO11 引脚电平，读取值对应于 AUX_AIODIO1:GPIODIN 位 3。
10	AUXIO10	R	0h	AUXIO10 引脚电平，读取值对应于 AUX_AIODIO1:GPIODIN 位 2。
9	AUXIO9	R	0h	AUXIO9 引脚电平，读取值对应于 AUX_AIODIO1:GPIODIN 位 1。
8	AUXIO8	R	0h	AUXIO8 引脚电平，读取值对应于 AUX_AIODIO1:GPIODIN 位 0。
7	AUXIO7	R	0h	AUXIO7 引脚电平，读取值对应于 AUX_AIODIO0:GPIODIN 位 7。
6	AUXIO6	R	0h	AUXIO6 引脚电平，读取值对应于 AUX_AIODIO0:GPIODIN 位 6。
5	AUXIO5	R	0h	AUXIO5 引脚电平，读取值对应于 AUX_AIODIO0:GPIODIN 位 5。
4	AUXIO4	R	0h	AUXIO4 引脚电平，读取值对应于 AUX_AIODIO0:GPIODIN 位 4。
3	AUXIO3	R	0h	AUXIO3 引脚电平，读取值对应于 AUX_AIODIO0:GPIODIN 位 3。
2	AUXIO2	R	0h	AUXIO2 引脚电平，读取值对应于 AUX_AIODIO0:GPIODIN 位 2。
1	AUXIO1	R	0h	AUXIO1 引脚电平，读取值对应于 AUX_AIODIO0:GPIODIN 位 1。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-76. EVSTAT0 寄存器字段说明（续）

位	字段	类型	复位	说明
0	AUXIO0	R	0h	AUXIO0 引脚电平，读取值对应于 AUX_AIODIO0:GPIODIN 位 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.3.2 EVSTAT1 寄存器（偏移 = 4h）[复位 = 00000000h]

图 19-67 展示了 EVSTAT1，表 19-77 对其进行了介绍。

返回到汇总表。

事件状态 1
寄存器保存与 AUX 时钟同步的 64 位事件总线的事件 16 至 31。除非另有说明，否则通过此寄存器读取的所有事件都以 SCE 时钟速率同步。以下订阅者使用此寄存器中的异步版本事件。
- AUX_TIMER2。
- AUX_ANAIF。
- AUX_TDC。
- AUX_SYSIF。
- AUX_AIODIOn。
- EVOBSCFG。

图 19-67. EVSTAT1 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
AUXIO31	AUXIO30	AUXIO29	AUXIO28	AUXIO27	AUXIO26	AUXIO25	AUXIO24
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
AUXIO23	AUXIO22	AUXIO21	AUXIO20	AUXIO19	AUXIO18	AUXIO17	AUXIO16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

表 19-77. EVSTAT1 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15	AUXIO31	R	0h	AUXIO31 引脚电平，读取值对应于 AUX_AIODIO3:GPIODIN 位 7。
14	AUXIO30	R	0h	AUXIO30 引脚电平，读取值对应于 AUX_AIODIO3:GPIODIN 位 6。
13	AUXIO29	R	0h	AUXIO29 引脚电平，读取值对应于 AUX_AIODIO3:GPIODIN 位 5。
12	AUXIO28	R	0h	AUXIO28 引脚电平，读取值对应于 AUX_AIODIO3:GPIODIN 位 4。
11	AUXIO27	R	0h	AUXIO27 引脚电平，读取值对应于 AUX_AIODIO3:GPIODIN 位 3。
10	AUXIO26	R	0h	AUXIO26 引脚电平，读取值对应于 AUX_AIODIO3:GPIODIN 位 2。
9	AUXIO25	R	0h	AUXIO25 引脚电平，读取值对应于 AUX_AIODIO3:GPIODIN 位 1。
8	AUXIO24	R	0h	AUXIO24 引脚电平，读取值对应于 AUX_AIODIO3:GPIODIN 位 0。
7	AUXIO23	R	0h	AUXIO23 引脚电平，读取值对应于 AUX_AIODIO2:GPIODIN 位 7。
6	AUXIO22	R	0h	AUXIO22 引脚电平，读取值对应于 AUX_AIODIO2:GPIODIN 位 6。
5	AUXIO21	R	0h	AUXIO21 引脚电平，读取值对应于 AUX_AIODIO2:GPIODIN 位 5。
4	AUXIO20	R	0h	AUXIO20 引脚电平，读取值对应于 AUX_AIODIO2:GPIODIN 位 4。
3	AUXIO19	R	0h	AUXIO19 引脚电平，读取值对应于 AUX_AIODIO2:GPIODIN 位 3。
2	AUXIO18	R	0h	AUXIO18 引脚电平，读取值对应于 AUX_AIODIO2:GPIODIN 位 2。
1	AUXIO17	R	0h	AUXIO17 引脚电平，读取值对应于 AUX_AIODIO2:GPIODIN 位 1。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-77. EVSTAT1 寄存器字段说明（续）

位	字段	类型	复位	说明
0	AUXIO16	R	0h	AUXIO16 引脚电平，读取值对应于 AUX_AIODIO2:GPIODIN 位 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.3.3 EVSTAT2 寄存器（偏移 = 8h）[复位 = 00000000h]

图 19-68 展示了 EVSTAT2，表 19-78 对其进行了介绍。

返回到汇总表。

事件状态 2
寄存器保存与 AUX 时钟同步的 64 位事件总线的事件 32 至 47。除非另有说明，否则通过此寄存器读取的所有事件都以 SCE 时钟速率同步。以下订阅者使用此寄存器中的异步版本事件。
- AUX_TIMER2。
- AUX_ANAIF。
- AUX_TDC。
- AUX_SYSIF。
- AUX_AIODIOn。
- EVOBSCFG。

图 19-68. EVSTAT2 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
AUX_COMPB	AUX_COMPA	MCU_OBSMUX1	MCU_OBSMUX0	MCU_EV	ACLK_REF	VDDR_RECHARGE	MCU_ACTIVE
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
PWR_DWN	SCLK_LF	AON_BATMON_TEMP_UPD	AON_BATMON_BAT_UPD	AON_RTC_4KHZ	AON_RTC_CH2_DLY	AON_RTC_CH2	MANUAL_EV
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

表 19-78. EVSTAT2 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15	AUX_COMPB	R	0h	比较器 B 的输出。 AUX_SYSIF:EVSYNCRATE.AUX_COMPB_SYNC_RATE 的配置设置该事件的同步速率。
14	AUX_COMPA	R	0h	比较器 A 的输出。 AUX_SYSIF:EVSYNCRATE.AUX_COMPA_SYNC_RATE 的配置设置该事件的同步速率。
13	MCU_OBSMUX1	R	0h	来自 IOC 的观测输入 1。 此事件由 IOC:OBSAUXOUTPUT.SEL1 配置。
12	MCU_OBSMUX0	R	0h	来自 IOC 的观测输入 0。 此事件由 IOC:OBSAUXOUTPUT.SEL0 配置，并可由 IOC:OBSAUXOUTPUT.SEL_MISC 覆盖。
11	MCU_EV	R	0h	由 EVENT:AUXSEL0 配置的事件中的事件。
10	ACLK_REF	R	0h	TDC 基准时钟。 它由 DDI_0_OSC:CTL0.ACLK_REF_SRC_SEL 配置，并由 AUX_SYSIF:TDCREFCLKCTL.REQ 使能。
9	VDDR_RECHARGE	R	0h	在 VDDR 充电期间，事件为高电平。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-78. EVSTAT2 寄存器字段说明（续）

位	字段	类型	复位	说明
8	MCU_ACTIVE	R	0h	系统（MCU、AUX 或 JTAG 域）处于运行状态或转换至运行状态（GLDO 或直流/直流电源状态）时，事件为高电平。在 VDDR 充电期间，事件不处于高电平。
7	PWR_DWN	R	0h	系统（MCU、AUX 或 JTAG 域）处于断电状态（uLDO 电源）时，事件为高电平。
6	SCLK_LF	R	0h	SCLK_LF 时钟
5	AON_BATMON_TEMP_UPD	R	0h	当更新 AON_BATMON:TEMP 时，事件在两个 SCLK_MF 时钟周期内为高电平。
4	AON_BATMON_BAT_UPD	R	0h	当更新 AON_BATMON:BAT 时，事件在两个 SCLK_MF 时钟周期内为高电平。
3	AON_RTC_4KHZ	R	0h	AON_RTC:SUBSEC.VALUE 位 19。 AON_RTC:CTL.RTC_4KHZ_EN 使能此事件。
2	AON_RTC_CH2_DLY	R	0h	AON_RTC:EVFLAGS.CH2 被 AON_RTC:CTL.EV_DELAY 配置延迟。
1	AON_RTC_CH2	R	0h	AON_RTC:EVFLAGS.CH2.
0	MANUAL_EV	R	0h	可编程事件。相关说明请参阅 MANUAL。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.3.4 EVSTAT3 寄存器（偏移 = Ch）[复位 = 00000000h]

图 19-69 展示了 EVSTAT3，表 19-79 对此进行了介绍。

返回到汇总表。

事件状态 3
寄存器保存与 AUX 时钟同步的 64 位事件总线的事件 48 至 63。除非另有说明，否则通过此寄存器读取的所有事件都以 SCE 时钟速率同步。以下订阅者使用此寄存器中的异步版本事件。
- AUX_TIMER2。
- AUX_ANAIF。
- AUX_TDC。
- AUX_SYSIF。
- AUX_AIODIOn。
- EVOBSCFG。

图 19-69. EVSTAT3 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
AUX_TIMER2_CLKSWITCH_RDY	AUX_DAC_HOLD_ACTIVE	AUX_SMPH_AUTOTAKE_DONE	AUX_ADC_FIFO_NOT_EMPTY	AUX_ADC_FIFO_ALMOST_FULL	AUX_ADC_IRQ	AUX_ADC_DONE	AUX_ISRC_RESET_N
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
AUX_TDC_DONE	AUX_TIMER0_EV	AUX_TIMER1_EV	AUX_TIMER2_PULSE	AUX_TIMER2_EV3	AUX_TIMER2_EV2	AUX_TIMER2_EV1	AUX_TIMER2_EV0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

表 19-79. EVSTAT3 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15	AUX_TIMER2_CLKSWITCH_RDY	R	0h	AUX_SYSIF:TIMER2CLKSWITCH.RDY
14	AUX_DAC_HOLD_ACTIVE	R	0h	AUX_ANAIF:DACSTAT.HOLD_ACTIVE
13	AUX_SMPH_AUTOTAKE_DONE	R	0h	相关说明，请参阅 AUX_SMPH:AUTOTAKE.SMPH_ID。
12	AUX_ADC_FIFO_NOT_EMPTY	R	0h	AUX_ANAIF:ADCFIFOSTAT.EMPTY 取反
11	AUX_ADC_FIFO_ALMOST_FULL	R	0h	AUX_ANAIF:ADCFIFOSTAT.ALMOST_FULL

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-79. EVSTAT3 寄存器字段说明（续）

位	字段	类型	复位	说明
10	AUX_ADC_IRQ	R	0h	该事件的逻辑函数是可配置的。 当 DMACTL.EN = 1 时： 事件 = UDMA0 通道 7 完成事件或 AUX_ANAIF:ADCFIFOSTAT.OVERFLOW 或 AUX_ANAIF:ADCFIFOSTAT.UNDERFLOW 当 DMACTL.EN = 0 时： 事件 =（非 AUX_ANAIF:ADCFIFOSTAT.EMPTY）或 AUX_ANAIF:ADCFIFOSTAT.OVERFLOW 或 AUX_ANAIF:ADCFIFOSTAT.UNDERFLOW UDMA0:DONEMASK 中的位 7 必须为 0。
9	AUX_ADC_DONE	R	0h	AUX_ANAIF ADC 转换完成事件。 事件以 AUX 总线速率同步。
8	AUX_ISRC_RESET_N	R	0h	AUX_ANAIF:ISRCCTL.RESET_N
7	AUX_TDC_DONE	R	0h	AUX_TDC:STAT.DONE
6	AUX_TIMER0_EV	R	0h	AUX_TIMER0_EV 事件，相关说明请参阅 AUX_TIMER01:T0TARGET。
5	AUX_TIMER1_EV	R	0h	AUX_TIMER1_EV 事件，相关说明请参阅 AUX_TIMER01:T1TARGET。
4	AUX_TIMER2_PULSE	R	0h	AUX_TIMER2 脉冲事件。 AUX_SYSIF:EVSYNCRATE.AUX_TIMER2_SYNC_RATE 的配置设置该事件的同步速率。
3	AUX_TIMER2_EV3	R	0h	AUX_TIMER2 事件输出 3。 AUX_SYSIF:EVSYNCRATE.AUX_TIMER2_SYNC_RATE 的配置设置该事件的同步速率。
2	AUX_TIMER2_EV2	R	0h	AUX_TIMER2 事件输出 2。 AUX_SYSIF:EVSYNCRATE.AUX_TIMER2_SYNC_RATE 的配置设置该事件的同步速率。
1	AUX_TIMER2_EV1	R	0h	AUX_TIMER2 事件输出 1。 AUX_SYSIF:EVSYNCRATE.AUX_TIMER2_SYNC_RATE 的配置设置该事件的同步速率。
0	AUX_TIMER2_EV0	R	0h	AUX_TIMER2 事件输出 0。 AUX_SYSIF:EVSYNCRATE.AUX_TIMER2_SYNC_RATE 的配置设置该事件的同步速率。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.3.5 SCEWEVCFG0 寄存器（偏移 = 10h）[复位 = 00000000h]

图 19-70 展示了 SCEWEVCFG0，表 19-80 对其进行了介绍。

返回到汇总表。

传感器控制器引擎等待事件配置 0
此寄存器的配置和 SCEWEVCFG1 控制 AUX_SCE:WUSTAT.EV_SIGNALS 中的位索引 7。此位可由 AUX_SCE WEV0、WEV1、BEV0 和 BEV1 指令使用。
当 COMB_EV_EN = 0 时：
AUX_SCE:WUSTAT.EV_SIGNALS (7) = EV0_SEL 事件
当 COMB_EV_EN = 1 时：
AUX_SCE:WUSTAT.EV_SIGNALS (7) = （EV0_SEL 事件）或（SCEWEVCFG1.EV1_SEL 事件）
位字段 SCEWEVCFG1.EV0_POL 和 SCEWEVCFG1.EV1_POL 控制所选事件的极性。
当需要带超时等待某个特定条件时，事件组合非常有用。

图 19-70. SCEWEVCFG0 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED	COMB_EV_EN	EV0_SEL					
R-0h	R/W-0h	R/W-0h					

表 19-80. SCEWEVCFG0 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6	COMB_EV_EN	R/W	0h	事件组合控制： 0：禁用事件组合。 1：启用事件组合。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-80. SCEWEVCFG0 寄存器字段说明（续）

位	字段	类型	复位	说明
5-0	EV0_SEL	R/W	0h	从同步事件总线中选择要在事件公式中使用的事件源。 0h = EVSTAT0.AUXIO0 1h = EVSTAT0.AUXIO1 2h = EVSTAT0.AUXIO2 3h = EVSTAT0.AUXIO3 4h = EVSTAT0.AUXIO4 5h = EVSTAT0.AUXIO5 6h = EVSTAT0.AUXIO6 7h = EVSTAT0.AUXIO7 8h = EVSTAT0.AUXIO8 9h = EVSTAT0.AUXIO9 Ah = EVSTAT0.AUXIO10 Bh = EVSTAT0.AUXIO11 Ch = EVSTAT0.AUXIO12 Dh = EVSTAT0.AUXIO13 Eh = EVSTAT0.AUXIO14 Fh = EVSTAT0.AUXIO15 10h = EVSTAT1.AUXIO16 11h = EVSTAT1.AUXIO17 12h = EVSTAT1.AUXIO18 13h = EVSTAT1.AUXIO19 14h = EVSTAT1.AUXIO20 15h = EVSTAT1.AUXIO21 16h = EVSTAT1.AUXIO22 17h = EVSTAT1.AUXIO23 18h = EVSTAT1.AUXIO24 19h = EVSTAT1.AUXIO25 1Ah = EVSTAT1.AUXIO26 1Bh = EVSTAT1.AUXIO27 1Ch = EVSTAT1.AUXIO28 1Dh = EVSTAT1.AUXIO29 1Eh = EVSTAT1.AUXIO30 1Fh = EVSTAT1.AUXIO31 20h = PROGDLY 中说明的可编程事件 21h = EVSTAT2.AON_RTC_CH2 22h = EVSTAT2.AON_RTC_CH2_DLY 23h = EVSTAT2.AON_RTC_4KHZ 24h = EVSTAT2.AON_BATMON_BAT_UPD 25h = EVSTAT2.AON_BATMON_TEMP_UPD 26h = EVSTAT2.SCLK_LF 27h = EVSTAT2.PWR_DWN 28h = EVSTAT2.MCU_ACTIVE 29h = EVSTAT2.VDDR_RECHARGE 2Ah = EVSTAT2.ACLK_REF 2Bh = EVSTAT2.MCU_EV 2Ch = EVSTAT2.MCU_OBSMUX0 2Dh = EVSTAT2.MCU_OBSMUX1 2Eh = EVSTAT2.AUX_COMPA 2Fh = EVSTAT2.AUX_COMPB 30h = EVSTAT3.AUX_TIMER2_EV0 31h = EVSTAT3.AUX_TIMER2_EV1 32h = EVSTAT3.AUX_TIMER2_EV2 33h = EVSTAT3.AUX_TIMER2_EV3 34h = EVSTAT3.AUX_TIMER2_PULSE 35h = EVSTAT3.AUX_TIMER1_EV 36h = EVSTAT3.AUX_TIMER0_EV 37h = EVSTAT3.AUX_TDC_DONE 38h = EVSTAT3.AUX_ISRC_RESET_N 39h = EVSTAT3.AUX_ADC_DONE 3Ah = EVSTAT3.AUX_ADC_IRQ 3Bh = EVSTAT3.AUX_ADC_FIFO_ALMOST_FULL 3Ch = EVSTAT3.AUX_ADC_FIFO_NOT_EMPTY 3Dh = EVSTAT3.AUX_SMPH_AUTOTAKE_DONE 3Eh = EVSTAT3.AUX_DAC_HOLD_ACTIVE

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-80. SCEWEVCFG0 寄存器字段说明（续）

位	字段	类型	复位	说明
				3Fh = EVSTAT3.AUX_TIMER2_CLKSWITCH_RDY

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.3.6 SCEWEVCFG1 寄存器（偏移 = 14h）[复位 = 00000000h]

图 19-71 展示了 SCEWEVCFG1，表 19-81 对其进行了介绍。

返回到汇总表。

传感器控制器引擎等待事件配置 1
相关说明请参阅 SCEWEVCFG0。

图 19-71. SCEWEVCFG1 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
EV0_POL	EV1_POL	EV1_SEL					
R/W-0h	R/W-0h	R/W-0h					

表 19-81. SCEWEVCFG1 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7	EV0_POL	R/W	0h	SCEWEVCFG0.EV0_SEL 事件的极性。 当 SCEWEVCFG0.COMB_EV_EN 为 0 时： 0：非反相。 1：非反相。 当 SCEWEVCFG0.COMB_EV_EN 为 1 时。 0：非反相。 1：反相。
6	EV1_POL	R/W	0h	EV1_SEL 事件的极性。 当 SCEWEVCFG0.COMB_EV_EN 为 0 时： 0：非反相。 1：非反相。 当 SCEWEVCFG0.COMB_EV_EN 为 1 时。 0：非反相。 1：反相。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-81. SCEWEVCFG1 寄存器字段说明（续）

位	字段	类型	复位	说明
5-0	EV1_SEL	R/W	0h	从同步事件总线中选择要在事件公式中使用的事件源。 0h = EVSTAT0.AUXIO0 1h = EVSTAT0.AUXIO1 2h = EVSTAT0.AUXIO2 3h = EVSTAT0.AUXIO3 4h = EVSTAT0.AUXIO4 5h = EVSTAT0.AUXIO5 6h = EVSTAT0.AUXIO6 7h = EVSTAT0.AUXIO7 8h = EVSTAT0.AUXIO8 9h = EVSTAT0.AUXIO9 Ah = EVSTAT0.AUXIO10 Bh = EVSTAT0.AUXIO11 Ch = EVSTAT0.AUXIO12 Dh = EVSTAT0.AUXIO13 Eh = EVSTAT0.AUXIO14 Fh = EVSTAT0.AUXIO15 10h = EVSTAT1.AUXIO16 11h = EVSTAT1.AUXIO17 12h = EVSTAT1.AUXIO18 13h = EVSTAT1.AUXIO19 14h = EVSTAT1.AUXIO20 15h = EVSTAT1.AUXIO21 16h = EVSTAT1.AUXIO22 17h = EVSTAT1.AUXIO23 18h = EVSTAT1.AUXIO24 19h = EVSTAT1.AUXIO25 1Ah = EVSTAT1.AUXIO26 1Bh = EVSTAT1.AUXIO27 1Ch = EVSTAT1.AUXIO28 1Dh = EVSTAT1.AUXIO29 1Eh = EVSTAT1.AUXIO30 1Fh = EVSTAT1.AUXIO31 20h = PROGDLY 中说明的可编程事件 21h = EVSTAT2.AON_RTC_CH2 22h = EVSTAT2.AON_RTC_CH2_DLY 23h = EVSTAT2.AON_RTC_4KHZ 24h = EVSTAT2.AON_BATMON_BAT_UPD 25h = EVSTAT2.AON_BATMON_TEMP_UPD 26h = EVSTAT2.SCLK_LF 27h = EVSTAT2.PWR_DWN 28h = EVSTAT2.MCU_ACTIVE 29h = EVSTAT2.VDDR_RECHARGE 2Ah = EVSTAT2.ACLK_REF 2Bh = EVSTAT2.MCU_EV 2Ch = EVSTAT2.MCU_OBSMUX0 2Dh = EVSTAT2.MCU_OBSMUX1 2Eh = EVSTAT2.AUX_COMPA 2Fh = EVSTAT2.AUX_COMPB 30h = EVSTAT3.AUX_TIMER2_EV0 31h = EVSTAT3.AUX_TIMER2_EV1 32h = EVSTAT3.AUX_TIMER2_EV2 33h = EVSTAT3.AUX_TIMER2_EV3 34h = EVSTAT3.AUX_TIMER2_PULSE 35h = EVSTAT3.AUX_TIMER1_EV 36h = EVSTAT3.AUX_TIMER0_EV 37h = EVSTAT3.AUX_TDC_DONE 38h = EVSTAT3.AUX_ISRC_RESET_N 39h = EVSTAT3.AUX_ADC_DONE 3Ah = EVSTAT3.AUX_ADC_IRQ 3Bh = EVSTAT3.AUX_ADC_FIFO_ALMOST_FULL 3Ch = EVSTAT3.AUX_ADC_FIFO_NOT_EMPTY 3Dh = EVSTAT3.AUX_SMPH_AUTOTAKE_DONE 3Eh = EVSTAT3.AUX_DAC_HOLD_ACTIVE

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-81. SCEWEVCFG1 寄存器字段说明（续）

位	字段	类型	复位	说明
				3Fh = EVSTAT3.AUX_TIMER2_CLKSWITCH_RDY

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.3.7 DMACTL 寄存器（偏移 = 18h）[复位 = 00000000h]

图 19-72 展示了 DMACTL，表 19-82 中对此进行了介绍。

返回到汇总表。

直接存储器存取控制

图 19-72. DMACTL 寄存器

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

表 19-82. DMACTL 寄存器字段说明

位	字段	类型	复位	说明
31-3	RESERVED	R	0h	保留
2	REQ_MODE	R/W	0h	UDMA0 请求模式 0h = 当满足 SEL 中配置的条件时，在 UDMA0 通道 7 上生成突发请求。 1h = 当满足 SEL 中配置的条件时，在 UDMA0 通道 7 上生成单个请求。
1	EN	R/W	0h	UDMA ADC 接口使能。 0：禁用 UDMA0 至 ADC 的接口。 1：使能 UDMA0 到 ADC 的接口。
0	SEL	R/W	0h	选择触发 ADC FIFO 数据的 UDMA0 传输所需的 FIFO 水线位。 0h = 当 ADC FIFO 中有样本时，将生成 UDMA0 触发事件。 1h = 当 ADC FIFO 几乎已满（3/4 满）时，将生成 UDMA0 触发事件。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.3.8 SWEVSET 寄存器（偏移 = 20h）[复位 = 00000000h]

图 19-73 展示了 SWEVSET，表 19-83 中对此进行了介绍。

返回到汇总表。

软件事件设置
设置从 AUX 域到 AON 和 MCU 域的软件事件标志。MCU 域中的 CPU 可以从 EVTOAONFLAGS 中读取事件标志，并在 EVTOAONFLAGSCLR 中清除事件标志。
这些事件标志的使用由软件定义。

图 19-73. SWEVSET 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED					SWEV2	SWEV1	SWEV0
R-0h					W-0h	W-0h	W-0h

表 19-83. SWEVSET 寄存器字段说明

位	字段	类型	复位	说明
31-3	RESERVED	R	0h	保留
2	SWEV2	W	0h	软件事件标志 2。 0：无效。 1：设置软件事件标志 2。
1	SWEV1	W	0h	软件事件标志 1。 0：无效。 1：设置软件事件标志 1。
0	SWEV0	W	0h	软件事件标志 0。 0：无效。 1：设置软件事件标志 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.3.9 EVTOAONFLAGS 寄存器（偏移 = 24h）[复位 = 00000000h]

图 19-74 展示了 EVTOAONFLAGS，表 19-84 中对此进行了介绍。

返回到汇总表。

AON 标志的事件
该寄存器包含一组路由到 AON_EVENT 的事件标志。
要清除事件标志，请写入 EVTOAONFLAGSCLR 或向该寄存器中的事件标志写入 0。

图 19-74. EVTOAONFLAGS 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							AUX_TIMER1_EV
R-0h							R/W0C-0h
7	6	5	4	3	2	1	0
AUX_TIMER0_EV	AUX_TDC_DONE	AUX_ADC_DONE	AUX_COMPB	AUX_COMPA	SWEV2	SWEV1	SWEV0
R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h

表 19-84. EVTOAONFLAGS 寄存器字段说明

位	字段	类型	复位	说明
31-9	RESERVED	R	0h	保留
8	AUX_TIMER1_EV	R/W0C	0h	当 EVSTAT3.AUX_TIMER1_EV 上出现由 EVTOAONPOL.AUX_TIMER1_EV 所选的电平时，该事件标志被置位。
7	AUX_TIMER0_EV	R/W0C	0h	当 EVSTAT3.AUX_TIMER0_EV 上出现由 EVTOAONPOL.AUX_TIMER0_EV 所选的电平时，该事件标志被置位。
6	AUX_TDC_DONE	R/W0C	0h	当 EVSTAT3.AUX_TDC_DONE 上出现由 EVTOAONPOL.AUX_TDC_DONE 所选的电平时，该事件标志被置位。
5	AUX_ADC_DONE	R/W0C	0h	当 EVSTAT3.AUX_ADC_DONE 上出现由 EVTOAONPOL.AUX_ADC_DONE 所选的电平时，该事件标志被置位。
4	AUX_COMPB	R/W0C	0h	当 EVSTAT2.AUX_COMPB 上出现由 EVTOAONPOL.AUX_COMPB 所选的电平时，该事件标志被置位。
3	AUX_COMPA	R/W0C	0h	当 EVSTAT2.AUX_COMPA 上出现由 EVTOAONPOL.AUX_COMPA 所选的电平时，该事件标志被置位。
2	SWEV2	R/W0C	0h	当软件将 1 写入 SWEVSET.SWEV2 时，该事件标志被置位。
1	SWEV1	R/W0C	0h	当软件将 1 写入 SWEVSET.SWEV1 时，该事件标志被置位。
0	SWEV0	R/W0C	0h	当软件将 1 写入 SWEVSET.SWEV0 时，该事件标志被置位。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.3.10 EVTOAONPOL 寄存器（偏移 = 28h）[复位 = 00000000h]

图 19-75 展示了 EVTOAONPOL，表 19-85 中对此进行了介绍。

返回到汇总表。

事件至 AON 极性
EVTOAONFLAGS 的事件源极性配置。

图 19-75. EVTOAONPOL 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							AUX_TIMER1_EV
R-0h							R/W-0h
7	6	5	4	3	2	1	0
AUX_TIMER0_EV	AUX_TDC_DONE	AUX_ADC_DONE	AUX_COMPB	AUX_COMPA	RESERVED		
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h		

表 19-85. EVTOAONPOL 寄存器字段说明

位	字段	类型	复位	说明
31-9	RESERVED	R	0h	保留
8	AUX_TIMER1_EV	R/W	0h	选择用于设置 EVTOAONFLAGS.AUX_TIMER1_EV 的 EVSTAT3.AUX_TIMER1_EV 电平。 0h = 高电平 1h = 低电平
7	AUX_TIMER0_EV	R/W	0h	选择用于设置 EVTOAONFLAGS.AUX_TIMER0_EV 的 EVSTAT3.AUX_TIMER0_EV 电平。 0h = 高电平 1h = 低电平
6	AUX_TDC_DONE	R/W	0h	选择用于设置 EVTOAONFLAGS.AUX_TDC_DONE 的 EVSTAT3.AUX_TDC_DONE 电平。 0h = 高电平 1h = 低电平
5	AUX_ADC_DONE	R/W	0h	选择用于设置 EVTOAONFLAGS.AUX_ADC_DONE 的 EVSTAT3.AUX_ADC_DONE 电平。 0h = 高电平 1h = 低电平
4	AUX_COMPB	R/W	0h	选择用于设置 EVTOAONFLAGS.AUX_COMPB 的 EVSTAT2.AUX_COMPB 边沿。 0h = 上升沿 1h = 下降沿
3	AUX_COMPA	R/W	0h	选择用于设置 EVTOAONFLAGS.AUX_COMPA 的 EVSTAT2.AUX_COMPA 边沿。 0h = 上升沿 1h = 下降沿
2-0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.3.11 EVTOAONFLAGSCLR 寄存器（偏移 = 2Ch）[复位 = 00000000h]

图 19-76 展示了 EVTOAONFLAGSCLR，表 19-86 中对此进行了介绍。

返回到汇总表。

事件至 AON 清除
清除 EVTOAONFLAGS 中的事件标志。
为了清除电平敏感事件标志，必须将此事件置为无效。

图 19-76. EVTOAONFLAGSCLR 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							AUX_TIMER1_EV
R-0h							W-0h
7	6	5	4	3	2	1	0
AUX_TIMER0_EV	AUX_TDC_DONE	AUX_ADC_DONE	AUX_COMPB	AUX_COMPA	SWEV2	SWEV1	SWEV0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

表 19-86. EVTOAONFLAGSCLR 寄存器字段说明

位	字段	类型	复位	说明
31-9	RESERVED	R	0h	保留
8	AUX_TIMER1_EV	W	0h	写入 1 以清除 EVTOAONFLAGS.AUX_TIMER1_EV。 读取值为 0。
7	AUX_TIMER0_EV	W	0h	写入 1 以清除 EVTOAONFLAGS.AUX_TIMER0_EV。 读取值为 0。
6	AUX_TDC_DONE	W	0h	写入 1 以清除 EVTOAONFLAGS.AUX_TDC_DONE。 读取值为 0。
5	AUX_ADC_DONE	W	0h	写入 1 以清除 EVTOAONFLAGS.AUX_ADC_DONE。 读取值为 0。
4	AUX_COMPB	W	0h	写入 1 以清除 EVTOAONFLAGS.AUX_COMPB。 读取值为 0。
3	AUX_COMPA	W	0h	写入 1 以清除 EVTOAONFLAGS.AUX_COMPA。 读取值为 0。
2	SWEV2	W	0h	写入 1 以清除 EVTOAONFLAGS.SWEV2。 读取值为 0。
1	SWEV1	W	0h	写入 1 以清除 EVTOAONFLAGS.SWEV1。 读取值为 0。
0	SWEV0	W	0h	写入 1 以清除 EVTOAONFLAGS.SWEV0。 读取值为 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.3.12 EVTOMCUFLAGS 寄存器（偏移 = 30h）[复位 = 00000000h]

图 19-77 展示了 EVTOMCUFLAGS，表 19-87 中对此进行了介绍。

返回到汇总表。

事件至 MCU 标志
该寄存器包含一组路由到 MCU 域的事件标志。
要清除事件标志，请写入 EVTOMCUFLAGSCLR 或向该寄存器中的事件标志写入 0。按照 AUX_SYSIF:WUCLR 中介绍的程序清除 AUX_WU_EV 事件标志。

图 19-77. EVTOMCUFLAGS 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
AUX_TIMER2_PULSE	AUX_TIMER2_EV3	AUX_TIMER2_EV2	AUX_TIMER2_EV1	AUX_TIMER2_EV0	AUX_ADC_IRQ	MCU_OBSMUX0	AUX_ADC_FIFO_ALMOST_FULL
R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h
7	6	5	4	3	2	1	0
AUX_ADC_DONE	AUX_SMPH_AUTOTAKE_DONE	AUX_TIMER1_EV	AUX_TIMER0_EV	AUX_TDC_DONE	AUX_COMPB	AUX_COMPA	AUX_WU_EV
R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h	R/W0C-0h

表 19-87. EVTOMCUFLAGS 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15	AUX_TIMER2_PULSE	R/W0C	0h	当 EVSTAT3.AUX_TIMER2_PULSE 上出现由 EVTOMCUPOL.AUX_TIMER2_PULSE 所选的电平时，该事件标志被置位。
14	AUX_TIMER2_EV3	R/W0C	0h	当 EVSTAT3.AUX_TIMER2_EV3 上出现由 EVTOMCUPOL.AUX_TIMER2_EV3 所选的电平时，该事件标志被置位。
13	AUX_TIMER2_EV2	R/W0C	0h	当 EVSTAT3.AUX_TIMER2_EV2 上出现由 EVTOMCUPOL.AUX_TIMER2_EV2 所选的电平时，该事件标志被置位。
12	AUX_TIMER2_EV1	R/W0C	0h	当 EVSTAT3.AUX_TIMER2_EV1 上出现由 EVTOMCUPOL.AUX_TIMER2_EV1 所选的电平时，该事件标志被置位。
11	AUX_TIMER2_EV0	R/W0C	0h	当 EVSTAT3.AUX_TIMER2_EV0 上出现由 EVTOMCUPOL.AUX_TIMER2_EV0 所选的电平时，该事件标志被置位。
10	AUX_ADC_IRQ	R/W0C	0h	当 EVSTAT3.AUX_ADC_IRQ 上出现由 EVTOMCUPOL.AUX_ADC_IRQ 所选的电平时，该事件标志被置位。
9	MCU_OBSMUX0	R/W0C	0h	当 EVSTAT2.MCU_OBSMUX0 上出现由 EVTOMCUPOL.MCU_OBSMUX0 所选的电平时，该事件标志被置位。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-87. EVTOMCUFLAGS 寄存器字段说明（续）

位	字段	类型	复位	说明
8	AUX_ADC_FIFO_ALMOST_FULL	R/W0C	0h	当 EVSTAT3.AUX_ADC_FIFO_ALMOST_FULL 上出现由 EVTOMCUPOL.AUX_ADC_FIFO_ALMOST_FULL 所选的电平时，该事件标志被置位。
7	AUX_ADC_DONE	R/W0C	0h	当 EVSTAT3.AUX_ADC_DONE 上出现由 EVTOMCUPOL.AUX_ADC_DONE 所选的电平时，该事件标志被置位。
6	AUX_SMPH_AUTOTAKE_DONE	R/W0C	0h	当 EVSTAT3.AUX_SMPH_AUTOTAKE_DONE 上出现由 EVTOMCUPOL.AUX_SMPH_AUTOTAKE_DONE 所选的电平时，该事件标志被置位。
5	AUX_TIMER1_EV	R/W0C	0h	当 EVSTAT3.AUX_TIMER1_EV 上出现由 EVTOMCUPOL.AUX_TIMER1_EV 所选的电平时，该事件标志被置位。
4	AUX_TIMER0_EV	R/W0C	0h	当 EVSTAT3.AUX_TIMER0_EV 上出现由 EVTOMCUPOL.AUX_TIMER0_EV 所选的电平时，该事件标志被置位。
3	AUX_TDC_DONE	R/W0C	0h	当 EVSTAT3.AUX_TDC_DONE 上出现由 EVTOMCUPOL.AUX_TDC_DONE 所选的电平时，该事件标志被置位。
2	AUX_COMPB	R/W0C	0h	当 EVSTAT2.AUX_COMPB 上出现由 EVTOMCUPOL.AUX_COMPB 所选的边沿时，该事件标志被置位。
1	AUX_COMPA	R/W0C	0h	当 EVSTAT2.AUX_COMPA 上出现由 EVTOMCUPOL.AUX_COMPA 所选的边沿时，该事件标志被置位。
0	AUX_WU_EV	R/W0C	0h	当 AUX_SYSIF:WUFLAGS 寄存器的 reduction-OR 结果上出现由 EVTOMCUPOL.AUX_WU_EV 所选的电平时，该事件标志被置位。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.3.13 EVTOMCUPOL 寄存器（偏移 = 34h）[复位 = 00000000h]

图 19-78 展示了 EVTOMCUPOL，表 19-88 中对此进行了介绍。

返回到汇总表。

事件至 MCU 极性
EVTOMCUFLAGS 的事件源极性配置。

图 19-78. EVTOMCUPOL 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
AUX_TIMER2_PULSE	AUX_TIMER2_EV3	AUX_TIMER2_EV2	AUX_TIMER2_EV1	AUX_TIMER2_EV0	AUX_ADC_IRQ	MCU_OBSMUX0	AUX_ADC_FIFO_ALMOST_FULL
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
AUX_ADC_DONE	AUX_SMPH_AUTOTAKE_DONE	AUX_TIMER1_EV	AUX_TIMER0_EV	AUX_TDC_DONE	AUX_COMPB	AUX_COMPA	AUX_WU_EV
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 19-88. EVTOMCUPOL 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15	AUX_TIMER2_PULSE	R/W	0h	选择设置 EVTOMCUFLAGS.AUX_TIMER2_PULSE 的事件源电平。 0h = 高电平 1h = 低电平
14	AUX_TIMER2_EV3	R/W	0h	选择设置 EVTOMCUFLAGS.AUX_TIMER2_EV3 的事件源电平。 0h = 高电平 1h = 低电平
13	AUX_TIMER2_EV2	R/W	0h	选择设置 EVTOMCUFLAGS.AUX_TIMER2_EV2 的事件源电平。 0h = 高电平 1h = 低电平
12	AUX_TIMER2_EV1	R/W	0h	选择设置 EVTOMCUFLAGS.AUX_TIMER2_EV1 的事件源电平。 0h = 高电平 1h = 低电平
11	AUX_TIMER2_EV0	R/W	0h	选择设置 EVTOMCUFLAGS.AUX_TIMER2_EV0 的事件源电平。 0h = 高电平 1h = 低电平
10	AUX_ADC_IRQ	R/W	0h	选择设置 EVTOMCUFLAGS.AUX_ADC_IRQ 的事件电平。 0h = 高电平 1h = 低电平
9	MCU_OBSMUX0	R/W	0h	选择设置 EVTOMCUFLAGS.MCU_OBSMUX0 的事件源电平。 0h = 高电平 1h = 低电平

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-88. EVTOMCUPOL 寄存器字段说明（续）

位	字段	类型	复位	说明
8	AUX_ADC_FIFO_ALMOST_FULL	R/W	0h	选择设置 EVTOMCUFLAGS.AUX_ADC_FIFO_ALMOST_FULL 的事件源电平。 0h = 高电平 1h = 低电平
7	AUX_ADC_DONE	R/W	0h	选择设置 EVTOMCUFLAGS.AUX_ADC_DONE 的事件源电平。 0h = 高电平 1h = 低电平
6	AUX_SMPH_AUTOTAKE_DONE	R/W	0h	选择设置 EVTOMCUFLAGS.AUX_SMPH_AUTOTAKE_DONE 的事件源电平。 0h = 高电平 1h = 低电平
5	AUX_TIMER1_EV	R/W	0h	选择设置 EVTOMCUFLAGS.AUX_TIMER1_EV 的事件源电平。 0h = 高电平 1h = 低电平
4	AUX_TIMER0_EV	R/W	0h	选择设置 EVTOMCUFLAGS.AUX_TIMER0_EV 的事件源电平。 0h = 高电平 1h = 低电平
3	AUX_TDC_DONE	R/W	0h	选择设置 EVTOMCUFLAGS.AUX_TDC_DONE 的事件源电平。 0h = 高电平 1h = 低电平
2	AUX_COMPB	R/W	0h	选择设置 EVTOMCUFLAGS.AUX_COMPB 的事件源边沿。 0h = 上升沿 1h = 下降沿
1	AUX_COMPA	R/W	0h	选择设置 EVTOMCUFLAGS.AUX_COMPA 的事件源边沿。 0h = 上升沿 1h = 下降沿
0	AUX_WU_EV	R/W	0h	选择设置 EVTOMCUFLAGS.AUX_WU_EV 的事件源电平。 0h = 高电平 1h = 低电平

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.3.14 EVTOMCUFLAGSCLR 寄存器（偏移 = 38h）[复位 = 00000000h]

图 19-79 展示了 EVTOMCUFLAGSCLR，表 19-89 中对此进行了介绍。

返回到汇总表。

事件至 MCU 标志清除
清除 EVTOMCUFLAGS 中的事件标志。
为了清除电平敏感事件标志，必须将此事件置为无效。

图 19-79. EVTOMCUFLAGSCLR 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
AUX_TIMER2_PULSE	AUX_TIMER2_EV3	AUX_TIMER2_EV2	AUX_TIMER2_EV1	AUX_TIMER2_EV0	AUX_ADC_IRQ	MCU_OBSMUX0	AUX_ADC_FIFO_ALMOST_FULL
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
7	6	5	4	3	2	1	0
AUX_ADC_DONE	AUX_SMPH_AUTOTAKE_DONE	AUX_TIMER1_EV	AUX_TIMER0_EV	AUX_TDC_DONE	AUX_COMPB	AUX_COMPA	AUX_WU_EV
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

表 19-89. EVTOMCUFLAGSCLR 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15	AUX_TIMER2_PULSE	W	0h	写入 1 以清除 EVTOMCUFLAGS.AUX_TIMER2_PULSE。 读取值为 0。
14	AUX_TIMER2_EV3	W	0h	写入 1 以清除 EVTOMCUFLAGS.AUX_TIMER2_EV3。 读取值为 0。
13	AUX_TIMER2_EV2	W	0h	写入 1 以清除 EVTOMCUFLAGS.AUX_TIMER2_EV2。 读取值为 0。
12	AUX_TIMER2_EV1	W	0h	写入 1 以清除 EVTOMCUFLAGS.AUX_TIMER2_EV1。 读取值为 0。
11	AUX_TIMER2_EV0	W	0h	写入 1 以清除 EVTOMCUFLAGS.AUX_TIMER2_EV0。 读取值为 0。
10	AUX_ADC_IRQ	W	0h	写入 1 以清除 EVTOMCUFLAGS.AUX_ADC_IRQ。 读取值为 0。
9	MCU_OBSMUX0	W	0h	写入 1 以清除 EVTOMCUFLAGS.MCU_OBSMUX0。 读取值为 0。
8	AUX_ADC_FIFO_ALMOST_FULL	W	0h	写入 1 以清除 EVTOMCUFLAGS.AUX_ADC_FIFO_ALMOST_FULL。 读取值为 0。
7	AUX_ADC_DONE	W	0h	写入 1 以清除 EVTOMCUFLAGS.AUX_ADC_DONE。 读取值为 0。
6	AUX_SMPH_AUTOTAKE_DONE	W	0h	写入 1 以清除 EVTOMCUFLAGS.AUX_SMPH_AUTOTAKE_DONE。 读取值为 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-89. EVTOMCUFLAGSCLR 寄存器字段说明（续）

位	字段	类型	复位	说明
5	AUX_TIMER1_EV	W	0h	写入 1 以清除 EVTOMCUFLAGS.AUX_TIMER1_EV。 读取值为 0。
4	AUX_TIMER0_EV	W	0h	写入 1 以清除 EVTOMCUFLAGS.AUX_TIMER0_EV。 读取值为 0。
3	AUX_TDC_DONE	W	0h	写入 1 以清除 EVTOMCUFLAGS.AUX_TDC_DONE。 读取值为 0。
2	AUX_COMPB	W	0h	写入 1 以清除 EVTOMCUFLAGS.AUX_COMPB。 读取值为 0。
1	AUX_COMPA	W	0h	写入 1 以清除 EVTOMCUFLAGS.AUX_COMPA。 读取值为 0。
0	AUX_WU_EV	W	0h	写入 1 以清除 EVTOMCUFLAGS.AUX_WU_EV。 读取值为 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.3.15 COMBEVTOMCUMASK 寄存器（偏移 = 3Ch）[复位 = 00000000h]

图 19-80 展示了 COMBEVTOMCUMASK，表 19-90 中对此进行了介绍。

返回到汇总表。

组合事件至 MCU 屏蔽
选择 EVTOMCUFLAGS 中参与 AUX_COMB 事件的事件标志，并将其贡献给 EVENT 和系统 CPU。
只要有一个或多个包含的事件标志被置位，AUX_COMB 事件就保持高电平。

图 19-80. COMBEVTOMCUMASK 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
AUX_TIMER2_PULSE	AUX_TIMER2_EV3	AUX_TIMER2_EV2	AUX_TIMER2_EV1	AUX_TIMER2_EV0	AUX_ADC_IRQ	MCU_OBSMUX0	AUX_ADC_FIFO_ALMOST_FULL
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
AUX_ADC_DONE	AUX_SMPH_AUTOTAKE_DONE	AUX_TIMER1_EV	AUX_TIMER0_EV	AUX_TDC_DONE	AUX_COMPB	AUX_COMPA	AUX_WU_EV
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 19-90. COMBEVTOMCUMASK 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15	AUX_TIMER2_PULSE	R/W	0h	EVTOMCUFLAGS.AUX_TIMER2_PULSE 对 AUX_COMB 事件的贡献。 0：排除。 1：包含。
14	AUX_TIMER2_EV3	R/W	0h	EVTOMCUFLAGS.AUX_TIMER2_EV3 对 AUX_COMB 事件的贡献。 0：排除。 1：包含。
13	AUX_TIMER2_EV2	R/W	0h	EVTOMCUFLAGS.AUX_TIMER2_EV2 对 AUX_COMB 事件的贡献。 0：排除。 1：包含。
12	AUX_TIMER2_EV1	R/W	0h	EVTOMCUFLAGS.AUX_TIMER2_EV1 对 AUX_COMB 事件的贡献。 0：排除。 1：包含。
11	AUX_TIMER2_EV0	R/W	0h	EVTOMCUFLAGS.AUX_TIMER2_EV0 对 AUX_COMB 事件的贡献。 0：排除。 1：包含。
10	AUX_ADC_IRQ	R/W	0h	EVTOMCUFLAGS.AUX_ADC_IRQ 对 AUX_COMB 事件的贡献。 0：排除。 1：包含。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-90. COMBEVTOMCUMASK 寄存器字段说明（续）

位	字段	类型	复位	说明
9	MCU_OBSMUX0	R/W	0h	EVTOMCUFLAGS.MCU_OBSMUX0 对 AUX_COMB 事件的贡献。 0：排除。 1：包含。
8	AUX_ADC_FIFO_ALMOST_FULL	R/W	0h	EVTOMCUFLAGS.AUX_ADC_FIFO_ALMOST_FULL 对 AUX_COMB 事件的贡献。 0：排除。 1：包含。
7	AUX_ADC_DONE	R/W	0h	EVTOMCUFLAGS.AUX_ADC_DONE 对 AUX_COMB 事件做出贡献。 0：排除。 1：包含。
6	AUX_SMPH_AUTOTAKE_DONE	R/W	0h	EVTOMCUFLAGS.AUX_SMPH_AUTOTAKE_DONE 对 AUX_COMB 事件的贡献。 0：排除。 1：包含。
5	AUX_TIMER1_EV	R/W	0h	EVTOMCUFLAGS.AUX_TIMER1_EV 对 AUX_COMB 事件的贡献。 0：排除。 1：包含。
4	AUX_TIMER0_EV	R/W	0h	EVTOMCUFLAGS.AUX_TIMER0_EV 对 AUX_COMB 事件的贡献。 0：排除。 1：包含。
3	AUX_TDC_DONE	R/W	0h	EVTOMCUFLAGS.AUX_TDC_DONE 对 AUX_COMB 事件做出贡献。 0：排除。 1：包含。
2	AUX_COMPB	R/W	0h	EVTOMCUFLAGS.AUX_COMPB 对 AUX_COMB 事件的贡献。 0：排除 1：包含。
1	AUX_COMPA	R/W	0h	EVTOMCUFLAGS.AUX_COMPA 对 AUX_COMB 事件的贡献。 0：排除。 1：包含。
0	AUX_WU_EV	R/W	0h	EVTOMCUFLAGS.AUX_WU_EV 对 AUX_COMB 事件的贡献。 0：排除。 1：包含。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.3.16 EVOBSCFG 寄存器（偏移 = 40h）[复位 = 00000000h]

图 19-81 展示了 EVOBSCFG，表 19-91 中对此进行了介绍。

返回到汇总表。

事件观测配置

图 19-81. EVOBSCFG 寄存器

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										EVOBS_SEL					
R-0h										R/W-0h					

表 19-91. EVOBSCFG 寄存器字段说明

位	字段	类型	复位	说明
31-6	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-91. EVOBSCFG 寄存器字段说明（续）

位	字段	类型	复位	说明
5-0	EVOBS_SEL	R/W	0h	从异步事件总线中选择代表 AUX_AIODIOn 中 AUX_EV_OBS 的事件。 0h = EVSTAT0.AUXIO0 1h = EVSTAT0.AUXIO1 2h = EVSTAT0.AUXIO2 3h = EVSTAT0.AUXIO3 4h = EVSTAT0.AUXIO4 5h = EVSTAT0.AUXIO5 6h = EVSTAT0.AUXIO6 7h = EVSTAT0.AUXIO7 8h = EVSTAT0.AUXIO8 9h = EVSTAT0.AUXIO9 Ah = EVSTAT0.AUXIO10 Bh = EVSTAT0.AUXIO11 Ch = EVSTAT0.AUXIO12 Dh = EVSTAT0.AUXIO13 Eh = EVSTAT0.AUXIO14 Fh = EVSTAT0.AUXIO15 10h = EVSTAT1.AUXIO16 11h = EVSTAT1.AUXIO17 12h = EVSTAT1.AUXIO18 13h = EVSTAT1.AUXIO19 14h = EVSTAT1.AUXIO20 15h = EVSTAT1.AUXIO21 16h = EVSTAT1.AUXIO22 17h = EVSTAT1.AUXIO23 18h = EVSTAT1.AUXIO24 19h = EVSTAT1.AUXIO25 1Ah = EVSTAT1.AUXIO26 1Bh = EVSTAT1.AUXIO27 1Ch = EVSTAT1.AUXIO28 1Dh = EVSTAT1.AUXIO29 1Eh = EVSTAT1.AUXIO30 1Fh = EVSTAT1.AUXIO31 20h = EVSTAT2.MANUAL_EV 21h = EVSTAT2.AON_RTC_CH2 22h = EVSTAT2.AON_RTC_CH2_DLY 23h = EVSTAT2.AON_RTC_4KHZ 24h = EVSTAT2.AON_BATMON_BAT_UPD 25h = EVSTAT2.AON_BATMON_TEMP_UPD 26h = EVSTAT2.SCLK_LF 27h = EVSTAT2.PWR_DWN 28h = EVSTAT2.MCU_ACTIVE 29h = EVSTAT2.VDDR_RECHARGE 2Ah = EVSTAT2.ACLK_REF 2Bh = EVSTAT2.MCU_EV 2Ch = EVSTAT2.MCU_OBSMUX0 2Dh = EVSTAT2.MCU_OBSMUX1 2Eh = EVSTAT2.AUX_COMPA 2Fh = EVSTAT2.AUX_COMPB 30h = EVSTAT3.AUX_TIMER2_EV0 31h = EVSTAT3.AUX_TIMER2_EV1 32h = EVSTAT3.AUX_TIMER2_EV2 33h = EVSTAT3.AUX_TIMER2_EV3 34h = EVSTAT3.AUX_TIMER2_PULSE 35h = EVSTAT3.AUX_TIMER1_EV 36h = EVSTAT3.AUX_TIMER0_EV 37h = EVSTAT3.AUX_TDC_DONE 38h = EVSTAT3.AUX_ISRC_RESET_N 39h = EVSTAT3.AUX_ADC_DONE 3Ah = EVSTAT3.AUX_ADC_IRQ 3Bh = EVSTAT3.AUX_ADC_FIFO_ALMOST_FULL 3Ch = EVSTAT3.AUX_ADC_FIFO_NOT_EMPTY 3Dh = EVSTAT3.AUX_SMPH_AUTOTAKE_DONE 3Eh = EVSTAT3.AUX_DAC_HOLD_ACTIVE

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-91. EVOBSCFG 寄存器字段说明（续）

位	字段	类型	复位	说明
				3Fh = EVSTAT3.AUX_TIMER2_CLKSWITCH_RDY

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.3.17 PROGDLY 寄存器（偏移 = 44h）[复位 = 00000000h]

图 19-82 展示了 PROGDLY，表 19-92 中对此进行了介绍。

返回到汇总表。

可编程延迟

图 19-82. PROGDLY 寄存器

31	30	29	28	27	26	25	24	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																值															
R-0h																R/W-0h															

表 19-92. PROGDLY 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	值	R/W	0h	值在 1MHz 速率下递减至 0。 当值为 0 时，事件 AUX_PROG_DLY_IDLE 为高电平，否则为低电平。 仅当 AUX_SYSIF:OPMODEACK.ACK 等于 A 或 LP 时，才使用可编程延迟计数器和 AUX_PROG_DLY_IDLE 事件。 当以下任一条件为真时，VALUE 的递减会暂停： 已设置 AUX_SCE:CTL.DBG_FREEZE_EN 且系统 CPU 处于调试模式下的暂停状态。 - 已设置 AUX_SYSIF:TIMERHALT.PROGDLY。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.3.18 MANUAL 寄存器（偏移 = 48h）[复位 = 00000000h]

图 19-83 展示了 MANUAL，表 19-93 中对此进行了介绍。

返回到汇总表。

手动
可编程事件。

图 19-83. MANUAL 寄存器

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															EV
R-0h															R/W-0h

表 19-93. MANUAL 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	EV	R/W	0h	该位字段设置 EVSTAT2.MANUAL_EV 的值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.3.19 EVSTAT0L 寄存器（偏移 = 4Ch）[复位 = 00000000h]

图 19-84 展示了 EVSTAT0L，表 19-94 对其进行了介绍。

返回到汇总表。

事件状态 0 低字节

图 19-84. EVSTAT0L 寄存器

31	30	29	28	27	26	25	24	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																								ALIAS_EV							
R-0h																								R-0h							

表 19-94. EVSTAT0L 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-0	ALIAS_EV	R	0h	EVSTAT0 事件 7 的别名降至 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.3.20 EVSTAT0H 寄存器（偏移 = 50h）[复位 = 00000000h]

图 19-85 展示了 EVSTAT0H，表 19-95 对其进行了介绍。

返回到汇总表。

事件状态 0 高字节

图 19-85. EVSTAT0H 寄存器

31	30	29	28	27	26	25	24	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																								ALIAS_EV							
R-0h																								R-0h							

表 19-95. EVSTAT0H 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-0	ALIAS_EV	R	0h	EVSTAT0 事件 15 的别名降至 8。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.3.21 EVSTAT1L 寄存器（偏移 = 54h）[复位 = 00000000h]

图 19-86 展示了 EVSTAT1L，表 19-96 对其进行了介绍。

返回到汇总表。

事件状态 1 低字节

图 19-86. EVSTAT1L 寄存器

31	30	29	28	27	26	25	24	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																								ALIAS_EV							
R-0h																								R-0h							

表 19-96. EVSTAT1L 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-0	ALIAS_EV	R	0h	EVSTAT1 事件 7 的别名降至 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.3.22 EVSTAT1H 寄存器（偏移 = 58h）[复位 = 00000000h]

图 19-87 展示了 EVSTAT1H，表 19-97 对其进行了介绍。

返回到汇总表。

事件状态 1 高字节

图 19-87. EVSTAT1H 寄存器

31	30	29	28	27	26	25	24	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																								ALIAS_EV							
R-0h																								R-0h							

表 19-97. EVSTAT1H 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-0	ALIAS_EV	R	0h	EVSTAT1 事件 15 的别名降至 8。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.3.23 EVSTAT2L 寄存器（偏移 = 5Ch）[复位 = 00000000h]

图 19-88 展示了 EVSTAT2L，表 19-98 对其进行了介绍。

返回到汇总表。

事件状态 2 低字节

图 19-88. EVSTAT2L 寄存器

31	30	29	28	27	26	25	24	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																								ALIAS_EV							
R-0h																								R-0h							

表 19-98. EVSTAT2L 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-0	ALIAS_EV	R	0h	EVSTAT2 事件 7 的别名降至 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.3.24 EVSTAT2H 寄存器（偏移 = 60h）[复位 = 00000000h]

图 19-89 展示了 EVSTAT2H，表 19-99 对其进行了介绍。

返回到汇总表。

事件状态 2 高字节

图 19-89. EVSTAT2H 寄存器

31	30	29	28	27	26	25	24	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																								ALIAS_EV							
R-0h																								R-0h							

表 19-99. EVSTAT2H 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-0	ALIAS_EV	R	0h	EVSTAT2 事件 15 的别名降至 8。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.3.25 EVSTAT3L 寄存器（偏移 = 64h）[复位 = 00000000h]

图 19-90 展示了 EVSTAT3L，表 19-100 对其进行了介绍。

返回到汇总表。

事件状态 3 低字节

图 19-90. EVSTAT3L 寄存器

31	30	29	28	27	26	25	24	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																								ALIAS_EV							
R-0h																								R-0h							

表 19-100. EVSTAT3L 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-0	ALIAS_EV	R	0h	EVSTAT3 事件 7 的别名降至 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.3.26 EVSTAT3H 寄存器（偏移 = 68h）[复位 = 00000000h]

图 19-91 展示了 EVSTAT3H，表 19-101 对其进行了介绍。

返回到汇总表。

事件状态 3 高字节

图 19-91. EVSTAT3H 寄存器

31	30	29	28	27	26	25	24	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																								ALIAS_EV							
R-0h																								R-0h							

表 19-101. EVSTAT3H 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-0	ALIAS_EV	R	0h	EVSTAT3 事件 15 的别名降至 8。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.4 AUX_SMPH 寄存器

表 19-102 列出了 AUX_SMPH 寄存器的存储器映射寄存器。表 19-102 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

表 19-102. AUX_SMPH 寄存器

偏移	首字母缩写词	寄存器名称	部分
0h	SMPH0	信标 0	SMPH0 寄存器（偏移 = 0h）[复位 = 00000001h]
4h	SMPH1	信标 1	SMPH1 寄存器（偏移 = 4h）[复位 = 00000001h]
8h	SMPH2	信标 2	SMPH2 寄存器（偏移 = 8h）[复位 = 00000001h]
Ch	SMPH3	信标 3	SMPH3 寄存器（偏移 = Ch）[复位 = 00000001h]
10h	SMPH4	信标 4	SMPH4 寄存器（偏移 = 10h）[复位 = 00000001h]
14h	SMPH5	信标 5	SMPH5 寄存器（偏移 = 14h）[复位 = 00000001h]
18h	SMPH6	信标 6	SMPH6 寄存器（偏移 = 18h）[复位 = 00000001h]
1Ch	SMPH7	信标 7	SMPH7 寄存器（偏移 = 1Ch）[复位 = 00000001h]
20h	AUTOTAKE	自动获取	AUTOTAKE 寄存器（偏移 = 20h）[复位 = 00000000h]

复杂的位访问类型经过编码可适应小型表单元。表 19-103 展示了适用于此部分中访问类型的代码。

表 19-103. AUX_SMPH 访问类型代码

访问类型	代码	说明
读取类型		
R	R	读取
写入类型		
W	W	写入
复位或默认值		
-n		复位后的值或默认值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.4.1 SMPH0 寄存器（偏移 = 0h）[复位 = 00000001h]

图 19-92 展示了 SMPH0，表 19-104 对其进行了介绍。

返回到汇总表。

信标 0

图 19-92. SMPH0 寄存器

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

表 19-104. SMPH0 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	STAT	R/W	1h	请求或释放信标。 通过读取请求： 0：信标不可用。 1：已授予信标。 通过写入释放： 0：请勿使用。 1：释放信标。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.4.2 SMPH1 寄存器（偏移 = 4h）[复位 = 00000001h]

图 19-93 展示了 SMPH1，表 19-105 对其进行了介绍。

返回到汇总表。

信标 1

图 19-93. SMPH1 寄存器

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

表 19-105. SMPH1 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	STAT	R/W	1h	请求或释放信标。 通过读取请求： 0：信标不可用。 1：已授予信标。 通过写入释放： 0：请勿使用。 1：释放信标。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.4.3 SMPH2 寄存器（偏移 = 8h）[复位 = 00000001h]

图 19-94 展示了 SMPH2，表 19-106 对其进行了介绍。

返回到汇总表。

信标 2

图 19-94. SMPH2 寄存器

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

表 19-106. SMPH2 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	STAT	R/W	1h	请求或释放信标。 通过读取请求： 0：信标不可用。 1：已授予信标。 通过写入释放： 0：请勿使用。 1：释放信标。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.4.4 SMPH3 寄存器（偏移 = Ch）[复位 = 00000001h]

图 19-95 展示了 SMPH3，表 19-107 对其进行了介绍。

返回到汇总表。

信标 3

图 19-95. SMPH3 寄存器

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

表 19-107. SMPH3 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	STAT	R/W	1h	请求或释放信标。 通过读取请求： 0：信标不可用。 1：已授予信标。 通过写入释放： 0：请勿使用。 1：释放信标。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.4.5 SMPH4 寄存器（偏移 = 10h）[复位 = 00000001h]

图 19-96 展示了 SMPH4，表 19-108 对其进行了介绍。

返回到汇总表。

信标 4

图 19-96. SMPH4 寄存器

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

表 19-108. SMPH4 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	STAT	R/W	1h	请求或释放信标。 通过读取请求： 0：信标不可用。 1：已授予信标。 通过写入释放： 0：请勿使用。 1：释放信标。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.4.6 SMPH5 寄存器（偏移 = 14h）[复位 = 00000001h]

图 19-97 展示了 SMPH5，表 19-109 对其进行了介绍。

返回到汇总表。

信标 5

图 19-97. SMPH5 寄存器

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

表 19-109. SMPH5 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	STAT	R/W	1h	请求或释放信标。 通过读取请求： 0：信标不可用。 1：已授予信标。 通过写入释放： 0：请勿使用。 1：释放信标。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.4.7 SMPH6 寄存器（偏移 = 18h）[复位 = 00000001h]

图 19-98 展示了 SMPH6，表 19-110 对其进行了介绍。

返回到汇总表。

信标 6

图 19-98. SMPH6 寄存器

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

表 19-110. SMPH6 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	STAT	R/W	1h	请求或释放信标。 通过读取请求： 0：信标不可用。 1：已授予信标。 通过写入释放： 0：请勿使用。 1：释放信标。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.4.8 SMPH7 寄存器（偏移 = 1Ch）[复位 = 00000001h]

图 19-99 展示了 SMPH7，表 19-111 对其进行了介绍。

返回到汇总表。

信标 7

图 19-99. SMPH7 寄存器

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

表 19-111. SMPH7 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	STAT	R/W	1h	请求或释放信标。 通过读取请求： 0：信标不可用。 1：已授予信标。 通过写入释放： 0：请勿使用。 1：释放信标。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.4.9 AUTOTAKE 寄存器（偏移 = 20h）[复位 = 00000000h]

图 19-100 展示了 AUTOTAKE，表 19-112 中对此进行了介绍。

返回到汇总表。

自动获取
单个信标的粘滞请求。

图 19-100. AUTOTAKE 寄存器

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													SMPH_ID		
R-0h													R/W-0h		

表 19-112. AUTOTAKE 寄存器字段说明

位	字段	类型	复位	说明
31-3	RESERVED	R	0h	保留
2-0	SMPH_ID	R/W	0h	将信标 ID 0x0-0x7 写入 SMPH_ID 以请求该信标，直到其被授予。 授予信标 SMPH_ID 时，事件 AUX_EVCTL:EVSTAT3.AUX_SMPH_AUTOTAKE_DONE 变为 1。 当软件释放信标或向 SMPH_ID 写入新值时，事件变为 0。 为避免信标损坏： - 此功能的使用必须限制在一个 CPU 内核上。 - 软件必须等到 AUX_EVCTL:EVSTAT3.AUX_SMPH_AUTOTAKE_DONE 为 1 后才能将新值写入 SMPH_ID。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.5 AUX_TDC 寄存器

表 19-113 列出了 AUX_TDC 寄存器的存储器映射寄存器。表 19-113 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

表 19-113. AUX_TDC 寄存器

偏移	首字母缩写词	寄存器名称	部分
0h	CTL	控件	CTL 寄存器 (偏移 = 0h) [复位 = 00000000h]
4h	STAT	状态	STAT 寄存器 (偏移 = 4h) [复位 = 00000006h]
8h	结果	结果	RESULT 寄存器 (偏移 = 8h) [复位 = 00000002h]
Ch	SATCFG	饱和配置	SATCFG 寄存器 (偏移 = Ch) [复位 = 0000000Fh]
10h	TRIGSRC	触发源	TRIGSRC 寄存器 (偏移 = 10h) [复位 = 00000000h]
14h	TRIGCNT	触发计数器	TRIGCNT 寄存器 (偏移 = 14h) [复位 = 00000000h]
18h	TRIGCNTLOAD	触发计数器加载	TRIGCNTLOAD 寄存器 (偏移 = 18h) [复位 = 00000000h]
1Ch	TRIGCNTCFG	触发计数器配置	TRIGCNTCFG 寄存器 (偏移 = 1Ch) [复位 = 00000000h]
20h	PRECTL	预分频器控制	PRECTL 寄存器 (偏移 = 20h) [复位 = 0000003Fh]
24h	PRECNTR	预分频器计数器	PRECNTR 寄存器 (偏移 = 24h) [复位 = 00000000h]

复杂的位访问类型经过编码可适应小型表单元。表 19-114 展示了适用于此部分中访问类型的代码。

表 19-114. AUX_TDC 访问类型代码

访问类型	代码	说明
读取类型		
R	R	读取
写入类型		
W	W	写入
复位或默认值		
-n		复位后的值或默认值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.5.1 CTL 寄存器（偏移 = 0h）[复位 = 00000000h]

图 19-101 展示了 CTL，表 19-115 中对此进行了介绍。

返回到汇总表。

控件

图 19-101. CTL 寄存器

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														CMD	
R-0h														W-0h	

表 19-115. CTL 寄存器字段说明

位	字段	类型	复位	说明
31-2	RESERVED	R	0h	保留
1-0	CMD	W	0h	TDC 命令。 0h = 清除 STAT.SAT、STAT.DONE 和 RESULT.VALUE。 这不是测量的先决条件。仅在 IDLE 状态下，才能保证可靠的清除操作。 1h = 同步计数器启动。 该计数器在开始计数之前，会在所选的起始事件到来时，先查找该事件的相反边沿；当所选边沿发生时才开始计数。这可确保实现边沿触发的启动，并且推荐用于频率测量。 2h = 异步计数器启动。 当起始事件为高电平时，计数器开始计数。为了实现精确的边沿到边沿测量，必须确保在你写下该命令之后，起始事件在至少 420ns 内保持为低电平。 3h = 强制 TDC 状态机返回 IDLE 状态。 当 AUX_TDC:STAT.STATE 等于 CLR_CNT 或 WAIT_CLR_CNT_DONE 时，切勿执行此命令。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.5.2 STAT 寄存器（偏移 = 4h）[复位 = 00000006h]

图 19-102 展示了 STAT，表 19-116 中对此进行了介绍。

返回到汇总表。

状态

图 19-102. STAT 寄存器

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

表 19-116. STAT 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7	SAT	R	0h	TDC 测量饱和标志。 0：转换尚未达到饱和。 1：由于达到饱和，转换停止。 当开始新的测量，或向 CTL.CMD 写入 CLR_RESULT 时，该字段将被清除。
6	DONE	R	0h	TDC 测量完成标志。 0：TDC 测量尚未完成。 1：TDC 测量已完成。 当开始新的 TDC 测量，或向 CTL.CMD 写入 CLR_RESULT 时，该字段将被清除。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-116. STAT 寄存器字段说明（续）

位	字段	类型	复位	说明
5-0	STATE	R	6h	TDC 状态机状态。 0h = 当前状态为 TDC_STATE_WAIT_START。 快速计数电路检测起始条件。状态机等待快速计数器递增。 4h = 当前状态为 TDC_STATE_WAIT_STARTSTOPCNTEN。 快速计数电路检测起始条件。状态机等待快速计数器递增。 6h = 当前状态为 TDC_STATE_IDLE。 这是复位和中止后的默认状态。当您将 CTL.CMD 写入 RUN_SYNC_START 或 RUN 时，状态将会发生变化。 7h = 当前状态为 TDC_STATE_CLRCNT。快速计数器电路复位。 8h =当前状态为 TDC_STATE_WAIT_STOP。 状态机等待快速计数器电路停止。 Ch = 当前状态为 TDC_STATE_WAIT_STOPCNTDOWN。 快速计数器电路检测停止条件。它将忽略 TRIGCNTLOAD_CNT 中配置的许多停止事件。 Eh = 当前状态为 TDC_STATE_GETRESULTS。 状态机从快速计数器电路复制计数器值。 Fh = 当前状态为 TDC_STATE_POR。 这是复位状态。 16h = 当前状态为 TDC_STATE_WAIT_CLRCNT_DONE。 状态机等待快速计数器电路完成复位。 1Eh = 当前状态为 TDC_WAIT_STARTFALL。 快速计数器电路等待起始事件的下降沿。 2Eh = 当前状态为 TDC_FORCESTOP。 您向 CTL.CMD 写入了 ABORT 以中止 TDC 测量。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.5.3 RESULT 寄存器（偏移 = 8h）[复位 = 00000002h]

图 19-103 展示了 RESULT，表 19-117 中对此进行了介绍。

返回到汇总表。

结果
上次 TDC 转换的结果。

图 19-103. RESULT 寄存器

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

表 19-117. RESULT 寄存器字段说明

位	字段	类型	复位	说明
31-25	RESERVED	R	0h	保留
24-0	值	R	2h	TDC 转换结果。 TDC 转换结果以 DDI_0_OSC:CTL0.ACLK_TDC_SRC_SEL 中选择的时钟源的时钟边沿个数来表示。计数同时统计上升沿和下降沿。 如果 TDC 计数器饱和、则该值略高于 SATCFG.LIMIT、因为需要非零时间来停止测量。因此，如果您将 SATCFG.LIMIT 配置为 R24，该字段的最大值将略高于 2^{24}。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.5.4 SATCFG 寄存器（偏移 = Ch）[复位 = 0000000Fh]

图 19-104 展示了 SATCFG，表 19-118 中对此进行了介绍。

返回到汇总表。

饱和配置

图 19-104. SATCFG 寄存器

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												限制			
R-0h												R/W-Fh			

表 19-118. SATCFG 寄存器字段说明

位	字段	类型	复位	说明
31-4	RESERVED	R	0h	保留
3-0	限制	R/W	Fh	饱和限值。 当 TDC 计数器饱和时，会设置标志 STAT.SAT。 不支持未在枚举中列出的值 3h = 结果位 12：设置 Result.Value[12] 后，TDC 转换将饱和并停止。 4h = 结果位 13：设置 Result.Value[13] 后，TDC 转换将饱和并停止。 5h = 结果位 14：设置 Result.Value[14] 后，TDC 转换将饱和并停止。 6h = 结果位 15：设置 Result.Value[15] 后，TDC 转换将饱和并停止。 7h = 结果位 16：设置 Result.Value[16] 后，TDC 转换将饱和并停止。 8h = 结果位 17：设置 Result.Value[17] 后，TDC 转换将饱和并停止。 9h = 结果位 18：设置 Result.Value[18] 后，TDC 转换将饱和并停止。 Ah = 结果位 19：设置 Result.Value[19] 后，TDC 转换将饱和并停止。 Bh = 结果位 20：设置 Result.Value[20] 后，TDC 转换将饱和并停止。 Ch = 结果位 21：设置 Result.Value[21] 后，TDC 转换将饱和并停止。 Dh = 结果位 22：设置 Result.Value[22] 后，TDC 转换将饱和并停止。 Eh = 结果位 23：设置 Result.Value[23] 后，TDC 转换将饱和并停止。 Fh =结果位 24：设置 Result.Value[24] 后，TDC 转换将饱和并停止。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.5.5 TRIGSRC 寄存器（偏移 = 10h）[复位 = 00000000h]

图 19-105 展示了 TRIGSRC，表 19-119 中对此进行了介绍。

返回到汇总表。

触发源
选择 TDC 启动和停止事件的源和极性。如要了解事件时序要求，请参阅技术参考手册。

图 19-105. TRIGSRC 寄存器

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

表 19-119. TRIGSRC 寄存器字段说明

位	字段	类型	复位	说明
31-15	RESERVED	R	0h	保留
14	STOP_POL	R/W	0h	停止源的极性。 仅当 STAT.STATE 为 IDLE 时进行更改。 0h = 检测到高电平时，TDC 转换停止。 1h = 检测到低电平时，TDC 转换停止。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-119. TRIGSRC 寄存器字段说明（续）

位	字段	类型	复位	说明
13-8	STOP_SRC	R/W	0h	从异步 AUX 事件总线选择停止源。 仅当 STAT.STATE 为 IDLE 时进行更改。 0h = AUX_EVCTL:EVSTAT0.AUXIO0 1h = AUX_EVCTL:EVSTAT0.AUXIO1 2h = AUX_EVCTL:EVSTAT0.AUXIO2 3h = AUX_EVCTL:EVSTAT0.AUXIO3 4h = AUX_EVCTL:EVSTAT0.AUXIO4 5h = AUX_EVCTL:EVSTAT0.AUXIO5 6h = AUX_EVCTL:EVSTAT0.AUXIO6 7h = AUX_EVCTL:EVSTAT0.AUXIO7 8h = AUX_EVCTL:EVSTAT0.AUXIO8 9h = AUX_EVCTL:EVSTAT0.AUXIO9 Ah = AUX_EVCTL:EVSTAT0.AUXIO10 Bh = AUX_EVCTL:EVSTAT0.AUXIO11 Ch = AUX_EVCTL:EVSTAT0.AUXIO12 Dh = AUX_EVCTL:EVSTAT0.AUXIO13 Eh = AUX_EVCTL:EVSTAT0.AUXIO14 Fh = AUX_EVCTL:EVSTAT0.AUXIO15 10h = AUX_EVCTL:EVSTAT1.AUXIO16 11h = AUX_EVCTL:EVSTAT1.AUXIO17 12h = AUX_EVCTL:EVSTAT1.AUXIO18 13h = AUX_EVCTL:EVSTAT1.AUXIO19 14h = AUX_EVCTL:EVSTAT1.AUXIO20 15h = AUX_EVCTL:EVSTAT1.AUXIO21 16h = AUX_EVCTL:EVSTAT1.AUXIO22 17h = AUX_EVCTL:EVSTAT1.AUXIO23 18h = AUX_EVCTL:EVSTAT1.AUXIO24 19h = AUX_EVCTL:EVSTAT1.AUXIO25 1Ah = AUX_EVCTL:EVSTAT1.AUXIO26 1Bh = AUX_EVCTL:EVSTAT1.AUXIO27 1Ch = AUX_EVCTL:EVSTAT1.AUXIO28 1Dh = AUX_EVCTL:EVSTAT1.AUXIO29 1Eh = AUX_EVCTL:EVSTAT1.AUXIO30 1Fh = AUX_EVCTL:EVSTAT1.AUXIO31 20h = AUX_EVCTL:EVSTAT2.MANUAL_EV 21h = AUX_EVCTL:EVSTAT2.AON_RTC_CH2 22h = AUX_EVCTL:EVSTAT2.AON_RTC_CH2_DLY 23h = AUX_EVCTL:EVSTAT2.AON_RTC_4KHZ 24h = AUX_EVCTL:EVSTAT2.AON_BATMON_BAT_UPD 25h = AUX_EVCTL:EVSTAT2.AON_BATMON_TEMP_UPD 26h = AUX_EVCTL:EVSTAT2.SCLK_LF 27h = AUX_EVCTL:EVSTAT2.PWR_DWN 28h = AUX_EVCTL:EVSTAT2.MCU_ACTIVE 29h = AUX_EVCTL:EVSTAT2.VDDR_RECHARGE 2Ah = AUX_EVCTL:EVSTAT2.ACLK_REF 2Bh = AUX_EVCTL:EVSTAT2.MCU_EV 2Ch = AUX_EVCTL:EVSTAT2.MCU_OBSMUX0 2Dh = AUX_EVCTL:EVSTAT2.MCU_OBSMUX1 2Eh = AUX_EVCTL:EVSTAT2.AUX_COMPA 2Fh = AUX_EVCTL:EVSTAT2.AUX_COMPB 30h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV0 31h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV1 32h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV2 33h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV3 34h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_PULSE 35h = AUX_EVCTL:EVSTAT3.AUX_TIMER1_EV 36h = AUX_EVCTL:EVSTAT3.AUX_TIMER0_EV 37h = AUX_EVCTL:EVSTAT3.AUX_TDC_DONE 38h = AUX_EVCTL:EVSTAT3.AUX_ISRC_RESET_N 39h = AUX_EVCTL:EVSTAT3.AUX_ADC_DONE 3Ah = AUX_EVCTL:EVSTAT3.AUX_ADC_IRQ 3Bh = AUX_EVCTL:EVSTAT3.AUX_ADC_FIFO_ALMOST_FULL 3Ch = AUX_EVCTL:EVSTAT3.AUX_ADC_FIFO_NOT_EMPTY 3Dh = AUX_EVCTL:EVSTAT3.AUX_SMPH_AUTOTAKE_DONE 3Eh = 选择通过 PRECTL 配置生成的 TDC 预分频器事件。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-119. TRIGSRC 寄存器字段说明（续）

位	字段	类型	复位	说明
				3Fh = 无事件。
7	RESERVED	R	0h	保留
6	START_POL	R/W	0h	启动源的极性。 仅当 STAT.STATE 为 IDLE 时进行更改。 0h = 检测到高电平时，TDC 转换开始。 1h = 检测到低电平时，TDC 转换开始。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-119. TRIGSRC 寄存器字段说明（续）

位	字段	类型	复位	说明
5-0	START_SRC	R/W	0h	从异步 AUX 事件总线选择开始源。 仅当 STAT.STATE 为 IDLE 时进行更改。 0h = AUX_EVCTL:EVSTAT0.AUXIO0 1h = AUX_EVCTL:EVSTAT0.AUXIO1 2h = AUX_EVCTL:EVSTAT0.AUXIO2 3h = AUX_EVCTL:EVSTAT0.AUXIO3 4h = AUX_EVCTL:EVSTAT0.AUXIO4 5h = AUX_EVCTL:EVSTAT0.AUXIO5 6h = AUX_EVCTL:EVSTAT0.AUXIO6 7h = AUX_EVCTL:EVSTAT0.AUXIO7 8h = AUX_EVCTL:EVSTAT0.AUXIO8 9h = AUX_EVCTL:EVSTAT0.AUXIO9 Ah = AUX_EVCTL:EVSTAT0.AUXIO10 Bh = AUX_EVCTL:EVSTAT0.AUXIO11 Ch = AUX_EVCTL:EVSTAT0.AUXIO12 Dh = AUX_EVCTL:EVSTAT0.AUXIO13 Eh = AUX_EVCTL:EVSTAT0.AUXIO14 Fh = AUX_EVCTL:EVSTAT0.AUXIO15 10h = AUX_EVCTL:EVSTAT1.AUXIO16 11h = AUX_EVCTL:EVSTAT1.AUXIO17 12h = AUX_EVCTL:EVSTAT1.AUXIO18 13h = AUX_EVCTL:EVSTAT1.AUXIO19 14h = AUX_EVCTL:EVSTAT1.AUXIO20 15h = AUX_EVCTL:EVSTAT1.AUXIO21 16h = AUX_EVCTL:EVSTAT1.AUXIO22 17h = AUX_EVCTL:EVSTAT1.AUXIO23 18h = AUX_EVCTL:EVSTAT1.AUXIO24 19h = AUX_EVCTL:EVSTAT1.AUXIO25 1Ah = AUX_EVCTL:EVSTAT1.AUXIO26 1Bh = AUX_EVCTL:EVSTAT1.AUXIO27 1Ch = AUX_EVCTL:EVSTAT1.AUXIO28 1Dh = AUX_EVCTL:EVSTAT1.AUXIO29 1Eh = AUX_EVCTL:EVSTAT1.AUXIO30 1Fh = AUX_EVCTL:EVSTAT1.AUXIO31 20h = AUX_EVCTL:EVSTAT2.MANUAL_EV 21h = AUX_EVCTL:EVSTAT2.AON_RTC_CH2 22h = AUX_EVCTL:EVSTAT2.AON_RTC_CH2_DLY 23h = AUX_EVCTL:EVSTAT2.AON_RTC_4KHZ 24h = AUX_EVCTL:EVSTAT2.AON_BATMON_BAT_UPD 25h = AUX_EVCTL:EVSTAT2.AON_BATMON_TEMP_UPD 26h = AUX_EVCTL:EVSTAT2.SCLK_LF 27h = AUX_EVCTL:EVSTAT2.PWR_DWN 28h = AUX_EVCTL:EVSTAT2.MCU_ACTIVE 29h = AUX_EVCTL:EVSTAT2.VDDR_RECHARGE 2Ah = AUX_EVCTL:EVSTAT2.ACLK_REF 2Bh = AUX_EVCTL:EVSTAT2.MCU_EV 2Ch = AUX_EVCTL:EVSTAT2.MCU_OBSMUX0 2Dh = AUX_EVCTL:EVSTAT2.MCU_OBSMUX1 2Eh = AUX_EVCTL:EVSTAT2.AUX_COMPA 2Fh = AUX_EVCTL:EVSTAT2.AUX_COMPB 30h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV0 31h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV1 32h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV2 33h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV3 34h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_PULSE 35h = AUX_EVCTL:EVSTAT3.AUX_TIMER1_EV 36h = AUX_EVCTL:EVSTAT3.AUX_TIMER0_EV 37h = AUX_EVCTL:EVSTAT3.AUX_TDC_DONE 38h = AUX_EVCTL:EVSTAT3.AUX_ISRC_RESET_N 39h = AUX_EVCTL:EVSTAT3.AUX_ADC_DONE 3Ah = AUX_EVCTL:EVSTAT3.AUX_ADC_IRQ 3Bh = AUX_EVCTL:EVSTAT3.AUX_ADC_FIFO_ALMOST_FULL 3Ch = AUX_EVCTL:EVSTAT3.AUX_ADC_FIFO_NOT_EMPTY 3Dh = AUX_EVCTL:EVSTAT3.AUX_SMPH_AUTOTAKE_DONE 3Eh = 选择通过 PRECTL 配置生成的 TDC 预分频器事件。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-119. TRIGSRC 寄存器字段说明（续）

位	字段	类型	复位	说明
				3Fh = 无事件。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.5.6 TRIGCNT 寄存器（偏移 = 14h）[复位 = 00000000h]

图 19-106 展示了 TRIGCNT，表 19-120 中对此进行了介绍。

返回到汇总表。

触发计数器
停止计数器控制和状态。

图 19-106. TRIGCNT 寄存器

31	30	29	28	27	26	25	24	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																CNT															
R-0h																R/W-0h															

表 19-120. TRIGCNT 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	CNT	R/W	0h	当 AUX_TDC:TRIGCNTCFG.EN 为 1 时，需忽略的停止事件数量。 读取 CNT，以获取在 TDC 测量期间需忽略的剩余停止事件数量。 写入 CNT，以更新 TDC 测量期间需忽略的剩余停止事件数量。如果已无剩余的停止事件可供忽略，TDC 测量将忽略对 CNT 的更新。 当 AUX_TDC:TRIGCNTCFG.EN 为 1 时，TRIGCNTLOAD.CNT 会在测量开始时加载到 CNT 中。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.5.7 TRIGCNTLOAD 寄存器（偏移 = 18h）[复位 = 00000000h]

图 19-107 展示了 TRIGCNTLOAD，表 19-121 中对此进行了介绍。

返回到汇总表。

触发计数器加载
停止计数器加载。

图 19-107. TRIGCNTLOAD 寄存器

31	30	29	28	27	26	25	24	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																CNT															
R-0h																R/W-0h															

表 19-121. TRIGCNTLOAD 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	CNT	R/W	0h	当 AUX_TDC:TRIGCNTCFG.EN 为 1 时，需忽略的停止事件数量。 如要测量事件源的频率，请执行以下操作： - 将起始事件设置为与停止事件相同。 - 将 CNT 设置为待测周期的数量。取值 0 或 1 均表示测量单个事件源周期。 如要测量事件源的脉冲宽度，请执行以下操作： - 将起始事件源设置为与停止事件源相同。 - 为起始与停止事件选择不同极性。 - 将 CNT 设为 0。 如要测量从起始事件到第 N 个停止事件（其中 N > 1）的时间，请执行以下操作： - 选择不同的起始事件源和停止事件源。 - 将 CNT 设为 (N-1)。 如要了解事件时序要求，请参阅技术参考手册。 当 AUX_TDC:TRIGCNTCFG.EN 为 1 时，系统会在测量开始时将 CNT 加载到 TRIGCNT.CNT 中。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.5.8 TRIGCNTCFG 寄存器（偏移 = 1Ch）[复位 = 00000000h]

图 19-108 展示了 TRIGCNTCFG，表 19-122 中对此进行了介绍。

返回到汇总表。

触发计数器配置
停止计数器配置。

图 19-108. TRIGCNTCFG 寄存器

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															EN
R-0h															R/W-0h

表 19-122. TRIGCNTCFG 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	EN	R/W	0h	启用停止计数器。 0：禁用停止计数器。 1：启用停止计数器。 仅当 STAT.STATE 为 IDLE 时进行更改。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.5.9 PRECTL 寄存器（偏移 = 20h）[复位 = 0000003Fh]

图 19-109 展示了 PRECTL，表 19-123 中对此进行了介绍。

返回到汇总表。

预分频器控制
预分频器可用于统计速率高于 AUX 总线速率的事件。
用途包括：
- 在异步事件总线上对指定事件进行脉冲计数。
- 对异步事件总线上的指定事件进行预分频。
若要在 TDC 测量中将预分频器输出用作事件源，必须将 TRIGSRC.START_SRC 和 TRIGSRC.STOP_SRC 均设置为 AUX_TDC_PRE。
建议在待测信号频率超过 AUX 总线速率的十分之一时使用预分频器。

图 19-109. PRECTL 寄存器

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

表 19-123. PRECTL 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7	RESET_N	R/W	0h	预分频器复位。 0：复位预分频器。 1：释放预分频器的复位状态。 复位预分频器后，AUX_TDC_PRE 事件变为 0。
6	RATIO	R/W	0h	预分频比。 这用于控制预分频器生成 AUX_TDC_PRE 事件的频率。 0h = 预分频器将输入除以 16。 AUX_TDC_PRE 事件每接收到 16 个输入信号的上升沿，便产生一个上升沿。输入信号每出现第 8 个上升沿，AUX_TDC_PRE 事件便翻转一次。 1h = 预分频器将输入除以 64。 AUX_TDC_PRE 事件每接收到 64 个输入信号的上升沿，便产生一个上升沿。输入信号每出现第 32 个上升沿，AUX_TDC_PRE 事件便翻转一次。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-123. PRECTL 寄存器字段说明（续）

位	字段	类型	复位	说明
5-0	SRC	R/W	3Fh	预分频器事件源。 从异步 AUX 事件总线中选择一个事件以连接到预分频器输入。 仅在 RESET_N 为 0 时配置。 0h = AUX_EVCTL:EVSTAT0.AUXIO0 1h = AUX_EVCTL:EVSTAT0.AUXIO1 2h = AUX_EVCTL:EVSTAT0.AUXIO2 3h = AUX_EVCTL:EVSTAT0.AUXIO3 4h = AUX_EVCTL:EVSTAT0.AUXIO4 5h = AUX_EVCTL:EVSTAT0.AUXIO5 6h = AUX_EVCTL:EVSTAT0.AUXIO6 7h = AUX_EVCTL:EVSTAT0.AUXIO7 8h = AUX_EVCTL:EVSTAT0.AUXIO8 9h = AUX_EVCTL:EVSTAT0.AUXIO9 Ah = AUX_EVCTL:EVSTAT0.AUXIO10 Bh = AUX_EVCTL:EVSTAT0.AUXIO11 Ch = AUX_EVCTL:EVSTAT0.AUXIO12 Dh = AUX_EVCTL:EVSTAT0.AUXIO13 Eh = AUX_EVCTL:EVSTAT0.AUXIO14 Fh = AUX_EVCTL:EVSTAT0.AUXIO15 10h = AUX_EVCTL:EVSTAT1.AUXIO16 11h = AUX_EVCTL:EVSTAT1.AUXIO17 12h = AUX_EVCTL:EVSTAT1.AUXIO18 13h = AUX_EVCTL:EVSTAT1.AUXIO19 14h = AUX_EVCTL:EVSTAT1.AUXIO20 15h = AUX_EVCTL:EVSTAT1.AUXIO21 16h = AUX_EVCTL:EVSTAT1.AUXIO22 17h = AUX_EVCTL:EVSTAT1.AUXIO23 18h = AUX_EVCTL:EVSTAT1.AUXIO24 19h = AUX_EVCTL:EVSTAT1.AUXIO25 1Ah = AUX_EVCTL:EVSTAT1.AUXIO26 1Bh = AUX_EVCTL:EVSTAT1.AUXIO27 1Ch = AUX_EVCTL:EVSTAT1.AUXIO28 1Dh = AUX_EVCTL:EVSTAT1.AUXIO29 1Eh = AUX_EVCTL:EVSTAT1.AUXIO30 1Fh = AUX_EVCTL:EVSTAT1.AUXIO31 20h = AUX_EVCTL:EVSTAT2.MANUAL_EV 21h = AUX_EVCTL:EVSTAT2.AON_RTC_CH2 22h = AUX_EVCTL:EVSTAT2.AON_RTC_CH2_DLY 23h = AUX_EVCTL:EVSTAT2.AON_RTC_4KHZ 24h = AUX_EVCTL:EVSTAT2.AON_BATMON_BAT_UPD 25h = AUX_EVCTL:EVSTAT2.AON_BATMON_TEMP_UPD 26h = AUX_EVCTL:EVSTAT2.SCLK_LF 27h = AUX_EVCTL:EVSTAT2.PWR_DWN 28h = AUX_EVCTL:EVSTAT2.MCU_ACTIVE 29h = AUX_EVCTL:EVSTAT2.VDDR_RECHARGE 2Ah = AUX_EVCTL:EVSTAT2.ACLK_REF 2Bh = AUX_EVCTL:EVSTAT2.MCU_EV 2Ch = AUX_EVCTL:EVSTAT2.MCU_OBSMUX0 2Dh = AUX_EVCTL:EVSTAT2.MCU_OBSMUX1 2Eh = AUX_EVCTL:EVSTAT2.AUX_COMPA 2Fh = AUX_EVCTL:EVSTAT2.AUX_COMPB 30h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV0 31h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV1 32h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV2 33h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV3 34h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_PULSE 35h = AUX_EVCTL:EVSTAT3.AUX_TIMER1_EV 36h = AUX_EVCTL:EVSTAT3.AUX_TIMER0_EV 37h = AUX_EVCTL:EVSTAT3.AUX_TDC_DONE 38h = AUX_EVCTL:EVSTAT3.AUX_ISRC_RESET_N 39h = AUX_EVCTL:EVSTAT3.AUX_ADC_DONE 3Ah = AUX_EVCTL:EVSTAT3.AUX_ADC_IRQ 3Bh = AUX_EVCTL:EVSTAT3.AUX_ADC_FIFO_ALMOST_FULL 3Ch = AUX_EVCTL:EVSTAT3.AUX_ADC_FIFO_NOT_EMPTY 3Dh = AUX_EVCTL:EVSTAT3.AUX_SMPH_AUTOTAKE_DONE

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-123. PRECTL 寄存器字段说明（续）

位	字段	类型	复位	说明
				3Fh = 无事件。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.5.10 PRECNTR 寄存器（偏移 = 24h）[复位 = 00000000h]

图 19-110 展示了 PRECNTR，表 19-124 中对此进行了介绍。

返回到汇总表。

预分频器计数器

图 19-110. PRECNTR 寄存器

31	30	29	28	27	26	25	24	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																CNT															
R-0h																R/W-0h															

表 19-124. PRECNTR 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	CNT	R/W	0h	预分频器计数器值。 向 CNT 写入一个值，以将 16 位预分频计数器的值记录到 CNT 中。 读取 CNT 以获取记录的值。 当您释放复位时，如果事件源电平发生上升，则读取值会有 1 LSB 的不确定性。 当您记录分频器计数器时，如果事件源电平发生上升，则读取值会有 1 LSB 的不确定性。 请注意以下几点： - 预分频器计数器由 PRECTL.RESET_N 复位为 2。 - 当预分频器输入端的上升沿数量少于 3 个时，记录值为 2；否则，记录值等于事件脉冲数减 1。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.6 AUX_TIMER01 寄存器

表 19-125 列出了 AUX_TIMER01 寄存器的存储器映射寄存器。表 19-125 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

表 19-125. AUX_TIMER01 寄存器

偏移	首字母缩写词	寄存器名称	部分
0h	T0CFG	计时器 0 的配置	T0CFG 寄存器 (偏移 = 0h) [复位 = 00000000h]
4h	T0CTL	计时器 0 的控制	T0CTL 寄存器 (偏移 = 4h) [复位 = 00000000h]
8h	T0TARGET	计时器 0 的目标	T0TARGET 寄存器 (偏移 = 8h) [复位 = 00000000h]
Ch	T0CNTR	计时器 0 的计数器	T0CNTR 寄存器 (偏移 = Ch) [复位 = 00000000h]
10h	T1CFG	计时器 1 的配置	T1CFG 寄存器 (偏移 = 10h) [复位 = 00000000h]
14h	T1CTL	计时器 1 的控制	T1CTL 寄存器 (偏移 = 14h) [复位 = 00000000h]
18h	T1TARGET	计时器 1 的目标	T1TARGET 寄存器 (偏移 = 18h) [复位 = 00000000h]
1Ch	T1CNTR	计时器 1 的计数器	T1CNTR 寄存器 (偏移 = 1Ch) [复位 = 00000000h]

复杂的位访问类型经过编码可适应小型表单元。表 19-126 展示了适用于此部分中访问类型的代码。

表 19-126. AUX_TIMER01 访问类型代码

访问类型	代码	说明
读取类型		
R	R	读取
写入类型		
W	W	写入
复位或默认值		
-n		复位后的值或默认值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.6.1 T0CFG 寄存器（偏移 = 0h）[复位 = 00000000h]

图 19-111 展示了 T0CFG，表 19-127 对其进行了介绍。

返回到汇总表。

计时器 0 的配置

图 19-111. T0CFG 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED	TICK_SRC_POL	TICK_SRC					
R-0h	R/W-0h	R/W-0h					
7	6	5	4	3	2	1	0
PRE				RESERVED		模式	重新加载
R/W-0h				R-0h		R/W-0h	R/W-0h

表 19-127. T0CFG 寄存器字段说明

位	字段	类型	复位	说明
31-15	RESERVED	R	0h	保留
14	TICK_SRC_POL	R/W	0h	勾选计时器 0 的源极性。 0h = TICK_SRC 上升沿的计数。 1h = TICK_SRC 下降沿的计数。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-127. T0CFG 寄存器字段说明（续）

位	字段	类型	复位	说明
13-8	TICK_SRC	R/W	0h	从同步事件总线中选择计时器 0 的节拍源。 0h = AUX_EVCTL:EVSTAT0.AUXIO0 1h = AUX_EVCTL:EVSTAT0.AUXIO1 2h = AUX_EVCTL:EVSTAT0.AUXIO2 3h = AUX_EVCTL:EVSTAT0.AUXIO3 4h = AUX_EVCTL:EVSTAT0.AUXIO4 5h = AUX_EVCTL:EVSTAT0.AUXIO5 6h = AUX_EVCTL:EVSTAT0.AUXIO6 7h = AUX_EVCTL:EVSTAT0.AUXIO7 8h = AUX_EVCTL:EVSTAT0.AUXIO8 9h = AUX_EVCTL:EVSTAT0.AUXIO9 Ah = AUX_EVCTL:EVSTAT0.AUXIO10 Bh = AUX_EVCTL:EVSTAT0.AUXIO11 Ch = AUX_EVCTL:EVSTAT0.AUXIO12 Dh = AUX_EVCTL:EVSTAT0.AUXIO13 Eh = AUX_EVCTL:EVSTAT0.AUXIO14 Fh = AUX_EVCTL:EVSTAT0.AUXIO15 10h = AUX_EVCTL:EVSTAT1.AUXIO16 11h = AUX_EVCTL:EVSTAT1.AUXIO17 12h = AUX_EVCTL:EVSTAT1.AUXIO18 13h = AUX_EVCTL:EVSTAT1.AUXIO19 14h = AUX_EVCTL:EVSTAT1.AUXIO20 15h = AUX_EVCTL:EVSTAT1.AUXIO21 16h = AUX_EVCTL:EVSTAT1.AUXIO22 17h = AUX_EVCTL:EVSTAT1.AUXIO23 18h = AUX_EVCTL:EVSTAT1.AUXIO24 19h = AUX_EVCTL:EVSTAT1.AUXIO25 1Ah = AUX_EVCTL:EVSTAT1.AUXIO26 1Bh = AUX_EVCTL:EVSTAT1.AUXIO27 1Ch = AUX_EVCTL:EVSTAT1.AUXIO28 1Dh = AUX_EVCTL:EVSTAT1.AUXIO29 1Eh = AUX_EVCTL:EVSTAT1.AUXIO30 1Fh = AUX_EVCTL:EVSTAT1.AUXIO31 20h = AUX_EVCTL:EVSTAT2.MANUAL_EV 21h = AUX_EVCTL:EVSTAT2.AON_RTC_CH2 22h = AUX_EVCTL:EVSTAT2.AON_RTC_CH2_DLY 23h = AUX_EVCTL:EVSTAT2.AON_RTC_4KHZ 24h = AUX_EVCTL:EVSTAT2.AON_BATMON_BAT_UPD 25h = AUX_EVCTL:EVSTAT2.AON_BATMON_TEMP_UPD 26h = AUX_EVCTL:EVSTAT2.SCLK_LF 27h = AUX_EVCTL:EVSTAT2.PWR_DWN 28h = AUX_EVCTL:EVSTAT2.MCU_ACTIVE 29h = AUX_EVCTL:EVSTAT2.VDDR_RECHARGE 2Ah = AUX_EVCTL:EVSTAT2.ACLK_REF 2Bh = AUX_EVCTL:EVSTAT2.MCU_EV 2Ch = AUX_EVCTL:EVSTAT2.MCU_OBSMUX0 2Dh = AUX_EVCTL:EVSTAT2.MCU_OBSMUX1 2Eh = AUX_EVCTL:EVSTAT2.AUX_COMPA 2Fh = AUX_EVCTL:EVSTAT2.AUX_COMPB 30h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV0 31h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV1 32h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV2 33h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV3 34h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_PULSE 35h = AUX_EVCTL:EVSTAT3.AUX_TIMER1_EV 36h = No event. 37h = AUX_EVCTL:EVSTAT3.AUX_TDC_DONE 38h = AUX_EVCTL:EVSTAT3.AUX_ISRC_RESET_N 39h = AUX_EVCTL:EVSTAT3.AUX_ADC_DONE 3Ah = AUX_EVCTL:EVSTAT3.AUX_ADC_IRQ 3Bh = AUX_EVCTL:EVSTAT3.AUX_ADC_FIFO_ALMOST_FULL 3Ch = AUX_EVCTL:EVSTAT3.AUX_ADC_FIFO_NOT_EMPTY 3Dh = AUX_EVCTL:EVSTAT3.AUX_SMPH_AUTOTAKE_DONE 3Eh = AUX_EVCTL:EVSTAT3.AUX_DAC_HOLD_ACTIVE 3Fh = AUX_EVCTL:EVSTAT3.AUX_TIMER2_CLKSWITCH_RDY

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-127. T0CFG 寄存器字段说明（续）

位	字段	类型	复位	说明
7-4	PRE	R/W	0h	预分频比为 2^{PRE}: 0x0：除以 1。 0x1：除以 2。 0x2：除以 4。 ... 0xF: 除以 32,768。
3-2	RESERVED	R	0h	保留
1	模式	R/W	0h	计时器 0 模式。 配置计时器 0 预分频器的源。 0h = 使用时钟作为预分频器的源。请注意，AUX_SYSIF:PEROPRATE.TIMER01_OP_RATE 设置时钟频率。 1h = 使用 TICK_SRC 设置的事件作为预分频器的源。
0	重新加载	R/W	0h	计时器 0 重载模式。 0h = 手动模式。 当计数器值等于或大于 T0TARGET.VALUE 时，计时器 0 停止且 T0CTL.EN 变为 0。 1h = 连续模式。 当计数器值等于或大于 (T0TARGET.VALUE - 1) 时，计时器 0 重新启动。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.6.2 T0CTL 寄存器（偏移 = 4h）[复位 = 00000000h]

图 19-112 展示了 T0CTL，表 19-128 对其进行了介绍。

返回到汇总表。

计时器 0 的控制

图 19-112. T0CTL 寄存器

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															EN
R-0h															R/W-0h

表 19-128. T0CTL 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	EN	R/W	0h	计时器 0 启用。 0：禁用计时器 0。 1：启用计时器 0。 当您启用计时器 0 时，计数器将从 0 重新启动。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.6.3 T0TARGET 寄存器（偏移 = 8h）[复位 = 00000000h]

图 19-113 展示了 T0TARGET，表 19-129 对其进行了介绍。

返回到汇总表。

计时器 0 的目标

图 19-113. T0TARGET 寄存器

31	30	29	28	27	26	25	24	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																值															
R-0h																R/W-0h															

表 19-129. T0TARGET 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	值	R/W	0h	计时器 0 的目标值。 手动重新加载模式： - 计时器 0 递增，直至计数器值等于或大于 VALUE。 - 当计数器值等于或大于 VALUE 时，AUX_TIMER0_EV 在 1 个外设时钟周期内发出高脉冲。 注意：当 VALUE 为 0 时，计时器 0 计数为 1。AUX_TIMER0_EV 在 1 个外设时钟周期内产生高脉冲。 连续重新加载模式： - 计时器 0 递增，直到计数器值等于或大于 (VALUE - 1)，然后从 0 重新启动。 - 当计数器值为 0 时，AUX_TIMER0_EV 在 1 个外设时钟周期内发出高电平脉冲，但启用计时器时除外。 注意：当 VALUE 小于 2 时，计时器 0 的计数器值保持为 0。启用计时器后，AUX_TIMER0_EV 变为高电平并在 1 个外设时钟周期内保持高电平。 允许在计时器运行时更新该 VALUE。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.6.4 T0CNTR 寄存器（偏移 = Ch）[复位 = 00000000h]

图 19-114 中显示 T0CNTR，表 19-130 中提供相关说明。

返回到汇总表。

计时器 0 的计数器

图 19-114. T0CNTR 寄存器

31	30	29	28	27	26	25	24	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																值															
R-0h																R-0h															

表 19-130. T0CNTR 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	值	R	0h	计时器 0 的计数器值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.6.5 T1CFG 寄存器（偏移 = 10h）[复位 = 00000000h]

图 19-115 展示了 T1CFG，表 19-131 对其进行了介绍。

返回到汇总表。

计时器 1 的配置

图 19-115. T1CFG 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED	TICK_SRC_POL	TICK_SRC					
R-0h	R/W-0h	R/W-0h					
7	6	5	4	3	2	1	0
PRE				RESERVED		模式	重新加载
R/W-0h				R-0h		R/W-0h	R/W-0h

表 19-131. T1CFG 寄存器字段说明

位	字段	类型	复位	说明
31-15	RESERVED	R	0h	保留
14	TICK_SRC_POL	R/W	0h	勾选计时器 1 的源极性。 0h = TICK_SRC 上升沿的计数。 1h = TICK_SRC 下降沿的计数。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-131. T1CFG 寄存器字段说明（续）

位	字段	类型	复位	说明
13-8	TICK_SRC	R/W	0h	从同步事件总线中选择计时器 1 节拍源。 0h = AUX_EVCTL:EVSTAT0.AUXIO0 1h = AUX_EVCTL:EVSTAT0.AUXIO1 2h = AUX_EVCTL:EVSTAT0.AUXIO2 3h = AUX_EVCTL:EVSTAT0.AUXIO3 4h = AUX_EVCTL:EVSTAT0.AUXIO4 5h = AUX_EVCTL:EVSTAT0.AUXIO5 6h = AUX_EVCTL:EVSTAT0.AUXIO6 7h = AUX_EVCTL:EVSTAT0.AUXIO7 8h = AUX_EVCTL:EVSTAT0.AUXIO8 9h = AUX_EVCTL:EVSTAT0.AUXIO9 Ah = AUX_EVCTL:EVSTAT0.AUXIO10 Bh = AUX_EVCTL:EVSTAT0.AUXIO11 Ch = AUX_EVCTL:EVSTAT0.AUXIO12 Dh = AUX_EVCTL:EVSTAT0.AUXIO13 Eh = AUX_EVCTL:EVSTAT0.AUXIO14 Fh = AUX_EVCTL:EVSTAT0.AUXIO15 10h = AUX_EVCTL:EVSTAT1.AUXIO16 11h = AUX_EVCTL:EVSTAT1.AUXIO17 12h = AUX_EVCTL:EVSTAT1.AUXIO18 13h = AUX_EVCTL:EVSTAT1.AUXIO19 14h = AUX_EVCTL:EVSTAT1.AUXIO20 15h = AUX_EVCTL:EVSTAT1.AUXIO21 16h = AUX_EVCTL:EVSTAT1.AUXIO22 17h = AUX_EVCTL:EVSTAT1.AUXIO23 18h = AUX_EVCTL:EVSTAT1.AUXIO24 19h = AUX_EVCTL:EVSTAT1.AUXIO25 1Ah = AUX_EVCTL:EVSTAT1.AUXIO26 1Bh = AUX_EVCTL:EVSTAT1.AUXIO27 1Ch = AUX_EVCTL:EVSTAT1.AUXIO28 1Dh = AUX_EVCTL:EVSTAT1.AUXIO29 1Eh = AUX_EVCTL:EVSTAT1.AUXIO30 1Fh = AUX_EVCTL:EVSTAT1.AUXIO31 20h = AUX_EVCTL:EVSTAT2.MANUAL_EV 21h = AUX_EVCTL:EVSTAT2.AON_RTC_CH2 22h = AUX_EVCTL:EVSTAT2.AON_RTC_CH2_DLY 23h = AUX_EVCTL:EVSTAT2.AON_RTC_4KHZ 24h = AUX_EVCTL:EVSTAT2.AON_BATMON_BAT_UPD 25h = AUX_EVCTL:EVSTAT2.AON_BATMON_TEMP_UPD 26h = AUX_EVCTL:EVSTAT2.SCLK_LF 27h = AUX_EVCTL:EVSTAT2.PWR_DWN 28h = AUX_EVCTL:EVSTAT2.MCU_ACTIVE 29h = AUX_EVCTL:EVSTAT2.VDDR_RECHARGE 2Ah = AUX_EVCTL:EVSTAT2.ACLK_REF 2Bh = AUX_EVCTL:EVSTAT2.MCU_EV 2Ch = AUX_EVCTL:EVSTAT2.MCU_OBSMUX0 2Dh = AUX_EVCTL:EVSTAT2.MCU_OBSMUX1 2Eh = AUX_EVCTL:EVSTAT2.AUX_COMPA 2Fh = AUX_EVCTL:EVSTAT2.AUX_COMPB 30h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV0 31h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV1 32h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV2 33h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV3 34h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_PULSE 35h = No event. 36h = AUX_EVCTL:EVSTAT3.AUX_TIMER0_EV 37h = AUX_EVCTL:EVSTAT3.AUX_TDC_DONE 38h = AUX_EVCTL:EVSTAT3.AUX_ISRC_RESET_N 39h = AUX_EVCTL:EVSTAT3.AUX_ADC_DONE 3Ah = AUX_EVCTL:EVSTAT3.AUX_ADC_IRQ 3Bh = AUX_EVCTL:EVSTAT3.AUX_ADC_FIFO_ALMOST_FULL 3Ch = AUX_EVCTL:EVSTAT3.AUX_ADC_FIFO_NOT_EMPTY 3Dh = AUX_EVCTL:EVSTAT3.AUX_SMPH_AUTOTAKE_DONE 3Eh = AUX_EVCTL:EVSTAT3.AUX_DAC_HOLD_ACTIVE 3Fh = AUX_EVCTL:EVSTAT3.AUX_TIMER2_CLKSWITCH_RDY

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-131. T1CFG 寄存器字段说明（续）

位	字段	类型	复位	说明
7-4	PRE	R/W	0h	预分频比为 2^{PRE}: 0x0：除以 1。 0x1：除以 2。 0x2：除以 4。 ... 0xF: 除以 32,768。
3-2	RESERVED	R	0h	保留
1	模式	R/W	0h	计时器 1 模式。 配置计时器 1 预分频器的源。 0h = 使用时钟作为预分频器的源。请注意，AUX_SYSIF:PEROPRATE.TIMER01_OP_RATE 设置时钟频率。 1h = 使用 TICK_SRC 设置的事件作为预分频器的源。
0	重新加载	R/W	0h	计时器 1 重载模式。 0h = 手动模式。 当计数器值等于或大于 T1TARGET.VALUE 时，计时器 1 停止且 T1CTL.EN 变为 0。 1h = 连续模式。 当计数器值等于或大于 (T1TARGET.VALUE - 1) 时，计时器 1 重新启动。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.6.6 T1CTL 寄存器（偏移 = 14h）[复位 = 00000000h]

图 19-116 展示了 T1CTL，表 19-132 对其进行了介绍。

返回到汇总表。

计时器 1 的控制

图 19-116. T1CTL 寄存器

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															EN
R-0h															R/W-0h

表 19-132. T1CTL 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	EN	R/W	0h	计时器 1 启用。 0：禁用计时器 1。 1：启用计时器 1。 当您启用计时器 1 时，计数器将从 0 重新启动。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.6.7 T1TARGET 寄存器（偏移 = 18h）[复位 = 00000000h]

图 19-117 展示了 T1TARGET，表 19-133 对其进行了介绍。

返回到汇总表。

计时器 1 的目标
计时器 1 的计数器目标值

图 19-117. T1TARGET 寄存器

31	30	29	28	27	26	25	24	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																值															
R-0h																R/W-0h															

表 19-133. T1TARGET 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	值	R/W	0h	计时器 1 的目标值。 手动重新加载模式： — 计时器 1 递增，直至计数器值等于或大于该值。 - 当计数器值等于或大于该值时，AUX_TIMER1_EV 在 1 个外设时钟周期内发出高脉冲。 注意：当 VALUE 为 0 时，计时器 1 计数为 1。AUX_TIMER1_EV 在 1 个外设时钟周期内产生高脉冲。 连续重新加载模式： - 计时器 1 递增，直到计数器值等于或大于 (VALUE - 1)，然后从 0 重新启动。 - 当计数器值为 0 时，AUX_TIMER1_EV 在 1 个外设时钟周期内发出高电平脉冲，但启用计时器时除外。 注意：当 VALUE 小于 2 时，计时器 1 的计数器值保持为 0。启用计时器后，AUX_TIMER1_EV 变为高电平并在 1 个外设时钟周期内保持高电平。 允许在计时器运行时更新该 VALUE。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.6.8 T1CNTR 寄存器（偏移 = 1Ch）[复位 = 00000000h]

图 19-118 展示了 T1CNTR，表 19-134 对其进行了介绍。

返回到汇总表。

计时器 1 的计数器

图 19-118. T1CNTR 寄存器

31	30	29	28	27	26	25	24	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																值															
R-0h																R-0h															

表 19-134. T1CNTR 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	值	R	0h	计时器 1 的计数器值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.7 AUX_TIMER2 寄存器

表 19-135 列出了 AUX_TIMER2 寄存器的存储器映射寄存器。表 19-135 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

表 19-135. AUX_TIMER2 寄存器

偏移	首字母缩写词	寄存器名称	部分
0h	CTL	计时器控制	CTL 寄存器（偏移 = 0h）[复位 = 00000000h]
4h	目标	目标	TARGET 寄存器（偏移 = 4h）[复位 = 00000000h]
8h	SHDWTARGET	影子目标	SHDWTARGET 寄存器（偏移 = 8h）[复位 = 00000000h]
Ch	CNTR	计数器	CNTR 寄存器（偏移 = Ch）[复位 = 00000000h]
10h	PRECFG	时钟预分频器配置	PRECFG 寄存器（偏移 = 10h）[复位 = 00000000h]
14h	EVCTL	事件控制	EVCTL 寄存器（偏移 = 14h）[复位 = 00000000h]
18h	PULSETRIG	脉冲触发器	PULSETRIG 寄存器（偏移 = 18h）[复位 = 00000000h]
80h	CH0EVCFG	通道 0 事件配置	CH0EVCFG 寄存器（偏移 = 80h）[复位 = 00000000h]
84h	CH0CCFG	通道 0 捕获配置	CH0CCFG 寄存器（偏移 = 84h）[复位 = 00000000h]
88h	CH0PCC	通道 0 流水线捕获对比情况	CH0PCC 寄存器（偏移 = 88h）[复位 = 00000000h]
8Ch	CH0CC	通道 0 捕获对标情况	CH0CC 寄存器（偏移 = 8Ch）[复位 = 00000000h]
90h	CH1EVCFG	通道 1 事件配置	CH1EVCFG 寄存器（偏移 = 90h）[复位 = 00000000h]
94h	CH1CCFG	通道 1 捕获配置	CH1CCFG 寄存器（偏移 = 94h）[复位 = 00000000h]
98h	CH1PCC	通道 1 流水线捕获对比情况	CH1PCC 寄存器（偏移 = 98h）[复位 = 00000000h]
9Ch	CH1CC	通道 1 捕获对标情况	CH1CC 寄存器（偏移 = 9Ch）[复位 = 00000000h]
A0h	CH2EVCFG	通道 2 事件配置	CH2EVCFG 寄存器（偏移 = A0h）[复位 = 00000000h]
A4h	CH2CCFG	通道 2 捕获配置	CH2CCFG 寄存器（偏移 = A4h）[复位 = 00000000h]
A8h	CH2PCC	通道 2 流水线捕获对比情况	CH2PCC 寄存器（偏移 = A8h）[复位 = 00000000h]
ACh	CH2CC	通道 2 捕获对标情况	CH2CC 寄存器（偏移 = ACh）[复位 = 00000000h]
B0h	CH3EVCFG	通道 3 事件配置	CH3EVCFG 寄存器（偏移 = B0h）[复位 = 00000000h]
B4h	CH3CCFG	通道 3 捕获配置	CH3CCFG 寄存器（偏移 = B4h）[复位 = 00000000h]
B8h	CH3PCC	通道 3 流水线捕获对比情况	CH3PCC 寄存器（偏移 = B8h）[复位 = 00000000h]
BCh	CH3CC	通道 3 捕获对标情况	CH3CC 寄存器（偏移 = BCh）[复位 = 00000000h]

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

复杂的位访问类型经过编码可适应小型表单元。表 19-136 展示了适用于此部分中访问类型的代码。

表 19-136. AUX_TIMER2 访问类型代码

访问类型	代码	说明
读取类型		
R	R	读取
写入类型		
W	W	写入
复位或默认值		
-n		复位后的值或默认值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.7.1 CTL 寄存器（偏移 = 0h）[复位 = 00000000h]

图 19-119 展示了 CTL，表 19-137 中对此进行了介绍。

返回到汇总表。

计时器控制

图 19-119. CTL 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED	CH3_RESET	CH2_RESET	CH1_RESET	CH0_RESET	TARGET_EN	模式	
R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	

表 19-137. CTL 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6	CH3_RESET	R/W	0h	通道 3 复位。 0：没有影响。 1：复位 CH3CC、CH3PCC、CH3EVCFG 和 CH3CCFG。 读取返回 0。
5	CH2_RESET	R/W	0h	通道 2 复位。 0：没有影响。 1：复位 CH2CC、CH2PCC、CH2EVCFG 和 CH2CCFG。 读取返回 0。
4	CH1_RESET	R/W	0h	通道 1 复位。 0：没有影响。 1：复位 CH1CC、CH1PCC、CH1EVCFG 和 CH1CCFG。 读取返回 0。
3	CH0_RESET	R/W	0h	通道 0 复位。 0：没有影响。 1：复位 CH0CC、CH0PCC、CH0EVCFG 和 CH0CCFG。 读取返回 0。
2	TARGET_EN	R/W	0h	选择计数器目标值。 必须选择目标才能使用影子目标功能。 0h = 65535 1h = TARGET.VALUE

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-137. CTL 寄存器字段说明（续）

位	字段	类型	复位	说明
1-0	模式	R/W	0h	计时器模式控制。 当您将 MODE 设置为 UP_ONCE、UP_PER 或 UPDWN_PER 时，计时器会从 0 重新启动。 当您写入 MODE 时，所有在内部队列中的更新都会被写入到 [CHnCC.*]，同时清除 TARGET。 0h = 禁用计时器。对计数器、通道和事件的更新停止。 1h = 递增一次。计时器从 0 递增到目标值，然后停止并将 MODE 设置为 DIS。 2h = 周期性递增。计时器会反复从 0 递增到目标值。 周期= (目标值 + 1) *计时器时钟周期 3h = 周期性递增与递减。计时器从 0 计数到目标值，再返回 0，如此循环往复。 周期= (目标值 * 2) *计时器时钟周期

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.7.2 TARGET 寄存器（偏移 = 4h）[复位 = 00000000h]

图 19-120 展示了 TARGET，表 19-138 中对此进行了介绍。

返回到汇总表。

目标
用户定义的计数器目标。

图 19-120. TARGET 寄存器

31	30	29	28	27	26	25	24	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																值															
R-0h																R/W-0h															

表 19-138. TARGET 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	值	R/W	0h	16 位用户定义计数器目标值，当由 CTL.TARGET_EN 选中时使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.7.3 SHDWTARGET 寄存器（偏移 = 8h）[复位 = 00000000h]

图 19-121 展示了 SHDWTARGET，表 19-139 中对此进行了介绍。

返回到汇总表。

影子目标

图 19-121. SHDWTARGET 寄存器

31	30	29	28	27	26	25	24	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																值															
R-0h																R/W-0h															

表 19-139. SHDWTARGET 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	值	R/W	0h	下一个计数器周期的目标值。 当 CNTR.VALUE 变为 0 时，计时器会将 VALUE 复制到 TARGET.VALUE。重新启动计时器时，不会进行复制操作。 这有助于避免周期随时间变化的 PWM 应用（有时称为相位校正 PWM）中出现周期抖动。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.7.4 CNTR 寄存器（偏移 = Ch）[复位 = 00000000h]

图 19-122 展示了 CNTR，表 19-140 中对此进行了介绍。

返回到汇总表。

计数器

图 19-122. CNTR 寄存器

31	30	29	28	27	26	25	24	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																值															
R-0h																R-0h															

表 19-140. CNTR 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	值	R	0h	16 位当前计数器值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.7.5 PRECFG 寄存器（偏移 = 10h）[复位 = 00000000h]

图 19-123 展示了 PRECFG，表 19-141 中对此进行了介绍。

返回到汇总表。

时钟预分频器配置

图 19-123. PRECFG 寄存器

31	30	29	28	27	26	25	24	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																								CLKDIV							
R-0h																								R/W-0h							

表 19-141. PRECFG 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-0	CLKDIV	R/W	0h	时钟分频。 CLKDIV 决定了用于计数器、同步及定时器事件更新的定时器时钟频率。该定时器时钟频率等于由 AUX_SYSIF:TIMER2CLKCTL.SRC 选定的时钟频率除以 (CLKDIV + 1) 的值。其倒数即为定时器时钟周期。 0x00：除以 1。 0x01：除以 2。 ... 0xFF：除以 256。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.7.6 EVCTL 寄存器（偏移 = 14h）[复位 = 00000000h]

图 19-124 展示了 EVCTL，表 19-142 中对此进行了介绍。

返回到汇总表。

事件控制
手动设置和清除单个事件。手动更新事件优先于对同一事件的自动通道更新。无法同时对同一事件执行“置位”和“清除”操作；此类请求将被忽略。
各通道均可对事件执行自动清除、置位、翻转或脉冲操作，这些操作按优先级从高到低排列。当多个通道试图同时更新同一事件时，优先级最高的操作将生效。
以下四个事件连接到异步 AUX 事件总线：
- 事件 0 连接到 AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV0。
- 事件 1 连接到 AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV1。
- 事件 2 连接到 AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV2。
- 事件 3 连接到 AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV3。

图 19-124. EVCTL 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
EV3_SET	EV3_CLR	EV2_SET	EV2_CLR	EV1_SET	EV1_CLR	EV0_SET	EV0_CLR
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

表 19-142. EVCTL 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7	EV3_SET	W	0h	设置事件 3。 写入 1 以设置事件 3。
6	EV3_CLR	W	0h	清除事件 3。 写入 1 以清除事件 3。
5	EV2_SET	W	0h	设置事件 2。 写入 1 以设置事件 2。
4	EV2_CLR	W	0h	清除事件 2。 写入 1 以清除事件 2。
3	EV1_SET	W	0h	设置事件 1。 写入 1 以设置事件 1。
2	EV1_CLR	W	0h	清除事件 1。 写入 1 以清除事件 1。
1	EV0_SET	W	0h	设置事件 0。 写入 1 以设置事件 0。
0	EV0_CLR	W	0h	清除事件 0。 写入 1 以清除事件 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.7.7 PULSETRIG 寄存器（偏移 = 18h）[复位 = 00000000h]

图 19-125 展示了 PULSETRIG，表 19-143 中对此进行了介绍。

返回到汇总表。

脉冲触发器

图 19-125. PULSETRIG 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	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

表 19-143. PULSETRIG 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	TRIG	W	0h	脉冲触发器。 写入 1 以向 AUX_EVCTL:EVSTAT3.AUX_TIMER2_PULSE 发送脉冲。脉冲宽度等于 AUX_SYSIF:TIMER2CLKCTL.SRC 的占空比。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.7.8 CH0EVCFG 寄存器（偏移 = 80h）[复位 = 00000000h]

图 19-126 展示了 CH0EVCFG，表 19-144 对其进行了介绍。

返回到汇总表。

通道 0 事件配置
该寄存器用于配置通道功能并启用事件输出。
每个通道都包含一个带存储功能的边沿检测电路。该电路的运行机制如下：
- 当 CCACT 选择捕获功能且 CTL.MODE 不为 DIS 时，电路处于启用状态。
- 当 CCACT 选择捕获功能且把 CTL.MODE 从 DIS 切换为其他模式时：电路会被刷新。
刷新操作使用两个 AUX_SYSIF:TIMER2CLKCTL.SRC 时钟周期。它能防止因存储在边沿检测电路中的信号值过期而引发的捕获事件。

图 19-126. CH0EVCFG 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
EV3_GEN	EV2_GEN	EV1_GEN	EV0_GEN	CCACT			
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h			

表 19-144. CH0EVCFG 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7	EV3_GEN	R/W	0h	事件 3 启用。 0：通道 0 不控制事件 3。 1：通道 0 控制事件 3。 当 0 < CCACT < 8 时，EV3_GEN 在捕获或比较事件发生后变为零。
6	EV2_GEN	R/W	0h	事件 2 启用。 0：通道 0 不控制事件 2。 1：通道 0 控制事件 2。 当 0 < CCACT < 8 时，EV2_GEN 在捕获或比较事件发生后变为零。
5	EV1_GEN	R/W	0h	事件 1 启用。 0：通道 0 不控制事件 1。 1：通道 0 控制事件 1。 当 0 < CCACT < 8 时，EV1_GEN 在捕获或比较事件发生后变为零。
4	EV0_GEN	R/W	0h	事件 0 启用。 0：通道 0 不控制事件 0。 1：通道 0 控制事件 0。 当 0 < CCACT < 8 时，EV0_GEN 在捕获或比较事件发生后变为零。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-144. CH0EVCFG 寄存器字段说明（续）

位	字段	类型	复位	说明
3-0	CCACT	R/W	0h	捕获-比较操作。 捕获-比较操作定义了 15 种不同的通道功能，这些功能会利用捕获、比较和归零事件。 0h = 禁用通道。 1h = 在捕获时复位，然后禁用通道。 通道功能序列： - 在捕获事件发生时启用相关事件，并将 CNTR.VALUE 复制到 CH0CC.VALUE。 - 禁用通道。 主要使用场景是在启动定时器之前选择此功能。 如果需要在 CTL.MODE 不为 DIS 的情况下选择此功能，请按照以下步骤操作： - 将 CCACT 设置为 SET_ON_CAPT，且不启用事件。 - 配置 CH0CCFG（可选）。 - 等待 PRECFG 定义的三个定时器时钟周期，然后将 CCACT 设置为 SET_ON_CAPT_DIS。事件启用为可选项。 这些步骤可防止因边沿检测电路中残留的旧信号值而触发捕获事件。 2h = 在零时清除，在比较时翻转，随后禁用通道。 通道功能序列： - 当 CNTR.VALUE = 0 时，清除已启用的事件。 - 当 CH0CC.VALUE = CNTR.VALUE 时，翻转已启用的事件。 - 禁用通道。 当 CH0CC.VALUE = 0 且 CNTR.VALUE = 0 时，设置已启用的事件。 3h = 在零时复位，在比较匹配时翻转，随后禁用通道。 通道功能序列： - 当 CNTR.VALUE = 0 时，置位已启用的事件。 - 当 CH0CC.VALUE = CNTR.VALUE 时，翻转已启用的事件。 - 禁用通道。 当 CH0CC.VALUE = 0 且 CNTR.VALUE = 0 时，已启用的事件会被清除。 4h = 在比较时清除，然后禁用通道。 通道功能序列： - 当 CH0CC.VALUE = CNTR.VALUE 时，清除已启用的事件。 - 禁用通道。 5h = 在比较时复位，然后禁用通道。 通道功能序列： - 当 CH0CC.VALUE = CNTR.VALUE 时，复位已启用的事件。 - 禁用通道。 6h = 在比较时翻转，然后禁用通道。 通道功能序列： - 当 CH0CC.VALUE = CNTR.VALUE 时，翻转已启用的事件。 - 禁用通道。 7h = 在比较时产生脉冲，然后禁用通道。 通道功能序列： - 当 CH0CC.VALUE = CNTR.VALUE 时，为已启用的事件产生脉冲。 - 禁用通道。 该事件在两个计时器时钟周期内保持高电平。 8h = 周期和脉冲宽度测量。 根据 CH0CCFG.EDGE 指定的信号边沿，持续捕获由 CH0CCFG.CAPT_SRC 选定信号的周期和脉冲宽度。 当 CH0CC.VALUE 包含信号周期且 CH0PCC.VALUE 包含信号脉冲宽度时，触发已启用的事件。 注意： - 请确保在 CTL.MODE 为 DIS 时配置 CH0CCFG.CAPT_SRC 和 CCACT，然后再将 CTL.MODE 设置为 UP_ONCE 或 UP_PER。 - 当 CH0CC.VALUE 包含信号周期时，计数器将在选定的定时器模式下重新开始计数。 - 如果多个通道使用此功能，这些通道将依次执行该功能。编号最小的通道具有最高优先级并首先执行该功能。下一次测量将在当前测量成功完成或超时后开始。当计数器值达到目标值时，即发生超时。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-144. CH0EVCFG 寄存器字段说明（续）

位	字段	类型	复位	说明
				- 若要监测超时事件，请将另一个通道配置为 SET_ON_CAPT 模式。 信号特性要求： - 信号周期 >= 2 * (1 + PRECFG.CLKDIV) * 定时器时钟周期。 - 信号周期 <= 65535 * (1 + PRECFG.CLKDIV) * 定时器时钟周期。 - 信号低电平和高电平持续时间 >= (1 + PRECFG.CLKDIV) * 定时器时钟周期。 9h = 在捕获时重复置位。 通道功能序列： - 在捕获事件发生时启用相关事件，并将 CNTR.VALUE 复制到 CH0CC.VALUE。 主要使用场景是在启动定时器之前选择此功能。 如果需要在 CTL.MODE 不为 DIS 的情况下选择此功能，请按照以下步骤操作： - 选择此功能，但不启用事件。 - 配置 CH0CCFG（可选）。 - 在启用事件之前，等待 PRECFG 所定义的三个定时器时钟周期。 这些步骤可防止因边沿检测电路中残留的旧信号值而触发捕获事件。 Ah = 在零时清除，在比较时反复翻转。 通道功能序列： - 当 CNTR.VALUE = 0 时，清除已启用的事件。 - 当 CH0CC.VALUE = CNTR.VALUE 时，翻转已启用的事件。 将 CTL.MODE 设置为 UPDWN_PER 以生成中心对齐 PWM。占空比计算方式如下： 当 CH0CC.VALUE <= TARGET.VALUE 时： 占空比 = 1 - (CH0CC.VALUE / TARGET.VALUE)。 当 CH0CC.VALUE > TARGET.VALUE 时： 占空比 = 0。 当 CH0CC.VALUE = 0 且 CNTR.VALUE = 0 时，设置已启用的事件。 Bh = 在零时置位，在比较时反复翻转。 通道功能序列： - 当 CNTR.VALUE = 0 时，置位已启用的事件。 - 当 CH0CC.VALUE = CNTR.VALUE 时，翻转已启用的事件。 将 CTL.MODE 设置为 UP_PER 以生成边沿对齐 PWM。占空比计算方式如下： 当 CH0CC.VALUE <= TARGET.VALUE 时： 占空比 = CH0CC.VALUE / (TARGET.VALUE + 1)。 当 CH0CC.VALUE > TARGET.VALUE 时： 占空比 = 1。 当 CH0CC.VALUE = 0 且 CNTR.VALUE = 0 时，已启用的事件会被清除。 Ch = 在比较时反复清除。 通道功能序列： - 当 CH0CC.VALUE = CNTR.VALUE 时，清除已启用的事件。 Dh = 在比较时反复置位。 通道功能序列： - 当 CH0CC.VALUE = CNTR.VALUE 时，复位已启用的事件。 Eh = 在比较时反复翻转。 通道功能序列： - 当 CH0CC.VALUE = CNTR.VALUE 时，翻转已启用的事件。 Fh = 在比较时反复产生脉冲。 通道功能序列： - 当 CH0CC.VALUE = CNTR.VALUE 时，为已启用的事件产生脉冲。该事件在两个计时器时钟周期内保持高电平。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.7.9 CH0CCFG 寄存器（偏移 = 84h）[复位 = 00000000h]

图 19-127 展示了 CH0CCFG，表 19-145 对其进行了介绍。

返回到汇总表。

通道 0 捕获配置

图 19-127. CH0CCFG 寄存器

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

表 19-145. CH0CCFG 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-145. CH0CCFG 寄存器字段说明（续）

位	字段	类型	复位	说明
6-1	CAPT_SRC	R/W	0h	从异步 AUX 事件总线中选择捕获信号源。 所选信号进入边沿检测电路。在以下情况下，可能会发生错误的捕获事件： - 边沿检测电路中包含过期的信号采样，且在未执行 CH0EVCFG 所述的刷新操作的情况下启用了该电路 - 在 CTL.MODE 不为 DIS 的状态下重新配置了该寄存器。 您可以避免误捕获事件。当所需的通道功能为以下情况时： - SET_ON_CAPT_DIS：请参阅 CH0EVCFG.CCACT 中关于 SET_ON_CAPT_DIS 的说明。 - SET_ON_CAPT：请参阅 CH0EVCFG.CCACT 中关于 SET_ON_CAPT 的说明。 - PER_PULSE_WIDTH_MEAS：请参阅 CH0EVCFG.CCACT 中关于 PER_PULSE_WIDTH_MEAS 的说明。 如果写入非枚举值，其行为与 NO_EVENT 相同；读取时将返回写入的值。 0h = AUX_EVCTL:EVSTAT0.AUXIO0 1h = AUX_EVCTL:EVSTAT0.AUXIO1 2h = AUX_EVCTL:EVSTAT0.AUXIO2 3h = AUX_EVCTL:EVSTAT0.AUXIO3 4h = AUX_EVCTL:EVSTAT0.AUXIO4 5h = AUX_EVCTL:EVSTAT0.AUXIO5 6h = AUX_EVCTL:EVSTAT0.AUXIO6 7h = AUX_EVCTL:EVSTAT0.AUXIO7 8h = AUX_EVCTL:EVSTAT0.AUXIO8 9h = AUX_EVCTL:EVSTAT0.AUXIO9 Ah = AUX_EVCTL:EVSTAT0.AUXIO10 Bh = AUX_EVCTL:EVSTAT0.AUXIO11 Ch = AUX_EVCTL:EVSTAT0.AUXIO12 Dh = AUX_EVCTL:EVSTAT0.AUXIO13 Eh = AUX_EVCTL:EVSTAT0.AUXIO14 Fh = AUX_EVCTL:EVSTAT0.AUXIO15 10h = AUX_EVCTL:EVSTAT1.AUXIO16 11h = AUX_EVCTL:EVSTAT1.AUXIO17 12h = AUX_EVCTL:EVSTAT1.AUXIO18 13h = AUX_EVCTL:EVSTAT1.AUXIO19 14h = AUX_EVCTL:EVSTAT1.AUXIO20 15h = AUX_EVCTL:EVSTAT1.AUXIO21 16h = AUX_EVCTL:EVSTAT1.AUXIO22 17h = AUX_EVCTL:EVSTAT1.AUXIO23 18h = AUX_EVCTL:EVSTAT1.AUXIO24 19h = AUX_EVCTL:EVSTAT1.AUXIO25 1Ah = AUX_EVCTL:EVSTAT1.AUXIO26 1Bh = AUX_EVCTL:EVSTAT1.AUXIO27 1Ch = AUX_EVCTL:EVSTAT1.AUXIO28 1Dh = AUX_EVCTL:EVSTAT1.AUXIO29 1Eh = AUX_EVCTL:EVSTAT1.AUXIO30 1Fh = AUX_EVCTL:EVSTAT1.AUXIO31 20h = AUX_EVCTL:EVSTAT2.MANUAL_EV 21h = AUX_EVCTL:EVSTAT2.AON_RTC_CH2 22h = AUX_EVCTL:EVSTAT2.AON_RTC_CH2_DLY 23h = AUX_EVCTL:EVSTAT2.AON_RTC_4KHZ 24h = AUX_EVCTL:EVSTAT2.AON_BATMON_BAT_UPD 25h = AUX_EVCTL:EVSTAT2.AON_BATMON_TEMP_UPD 26h = AUX_EVCTL:EVSTAT2.SCLK_LF 27h = AUX_EVCTL:EVSTAT2.PWR_DWN 28h = AUX_EVCTL:EVSTAT2.MCU_ACTIVE 29h = AUX_EVCTL:EVSTAT2.VDDR_RECHARGE 2Ah = AUX_EVCTL:EVSTAT2.ACLK_REF 2Bh = AUX_EVCTL:EVSTAT2.MCU_EV 2Ch = AUX_EVCTL:EVSTAT2.MCU_OBSMUX0 2Dh = AUX_EVCTL:EVSTAT2.MCU_OBSMUX1 2Eh = AUX_EVCTL:EVSTAT2.AUX_COMPA 2Fh = AUX_EVCTL:EVSTAT2.AUX_COMPB 30h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV0

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-145. CH0CCFG 寄存器字段说明（续）

位	字段	类型	复位	说明
				31h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV1 32h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV2 33h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV3 35h = AUX_EVCTL:EVSTAT3.AUX_TIMER1_EV 36h = AUX_EVCTL:EVSTAT3.AUX_TIMER0_EV 37h = AUX_EVCTL:EVSTAT3.AUX_TDC_DONE 38h = AUX_EVCTL:EVSTAT3.AUX_ISRC_RESET_N 39h = AUX_EVCTL:EVSTAT3.AUX_ADC_DONE 3Ah = AUX_EVCTL:EVSTAT3.AUX_ADC_IRQ 3Bh = AUX_EVCTL:EVSTAT3.AUX_ADC_FIFO_ALMOST_FULL 3Ch = AUX_EVCTL:EVSTAT3.AUX_ADC_FIFO_NOT_EMPTY 3Dh = AUX_EVCTL:EVSTAT3.AUX_SMPH_AUTOTAKE_DONE 3Fh = 无事件。
0	EDGE	R/W	0h	边沿配置。 通道在由 CAPT_SRC 选定的信号源上，于选定边沿处捕获计数值。 参阅 CH0EVCFG.CCACT。 0h = 在 CAPT_SRC 的下降沿捕获 CNTR.VALUE。 1h = 在 CAPT_SRC 的上升沿捕获 CNTR.VALUE。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.7.10 CH0PCC 寄存器（偏移 = 88h）[复位 = 00000000h]

图 19-128 展示了 CH0PCC，表 19-146 对其进行了介绍。

返回到汇总表。

通道 0 流水线捕获对比情况

图 19-128. CH0PCC 寄存器

31	30	29	28	27	26	25	24	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																值															
R-0h																R/W-0h															

表 19-146. CH0PCC 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	值	R/W	0h	流水线捕获比较值。 16 位用户自定义的流水线比较值，或通道更新后的捕获值。 比较模式： 当下一个 CNTR.VALUE 为零且 CTL.MODE 不为 DIS 时，VALUE 的更新值将传输至 CH0CC.VALUE。此机制适用于 PWM 生成，并能防止生成信号的边沿出现抖动。 捕捉模式： 当 CH0EVCFG.CCACT 设置为 PER_PULSE_WIDTH_MEAS 时，VALUE 存储所选信号的低电平或高电平脉冲宽度。具体测量对象由 CH0CCFG.EDGE 和 CH0CCFG.CAPT_SRC 决定。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.7.11 CH0CC 寄存器（偏移 = 8Ch）[复位 = 00000000h]

图 19-129 展示了 CH0CC，表 19-147 对其进行了介绍。

返回到汇总表。

通道 0 捕获对标情况

图 19-129. CH0CC 寄存器

31	30	29	28	27	26	25	24	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																值															
R-0h																R/W-0h															

表 19-147. CH0CC 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	值	R/W	0h	捕获比较值。 16 位用户定义的比较值，或通道更新后的捕获值。 比较模式： 将 VALUE 与 CNTR.VALUE 进行比较；当两者相等时，根据 CH0EVCFG.CCACT 的设定生成事件。 捕捉模式： 当发生捕获事件时，当前的计数器值被存储到 VALUE 中。 CH0EVCFG.CCACT 确定值是信号周期还是常规捕获值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.7.12 CH1EVCFG 寄存器（偏移 = 90h）[复位 = 00000000h]

图 19-130 展示了 CH1EVCFG，表 19-148 对其进行了介绍。

返回到汇总表。

通道 1 事件配置
该寄存器用于配置通道功能并启用事件输出。
每个通道都包含一个带存储功能的边沿检测电路。该电路的运行机制如下：
- 当 CCACT 选择捕获功能且 CTL.MODE 不为 DIS 时，电路处于启用状态。
- 当 CCACT 选择捕获功能且把 CTL.MODE 从 DIS 切换为其他模式时：电路会被刷新。
刷新操作使用两个 AUX_SYSIF:TIMER2CLKCTL.SRC 时钟周期。它能防止因存储在边沿检测电路中的信号值过期而引发的捕获事件。

图 19-130. CH1EVCFG 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
EV3_GEN	EV2_GEN	EV1_GEN	EV0_GEN	CCACT			
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h			

表 19-148. CH1EVCFG 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7	EV3_GEN	R/W	0h	事件 3 启用。 0：通道 1 不控制事件 3。 1：通道 1 控制事件 3。 当 0 < CCACT < 8 时，EV3_GEN 在捕获或比较事件发生后变为零。
6	EV2_GEN	R/W	0h	事件 2 启用。 0：通道 1 不控制事件 2。 1：通道 1 控制事件 2。 当 0 < CCACT < 8 时，EV2_GEN 在捕获或比较事件发生后变为零。
5	EV1_GEN	R/W	0h	事件 1 启用。 0：通道 1 不控制事件 1。 1：通道 1 控制事件 1。 当 0 < CCACT < 8 时，EV1_GEN 在捕获或比较事件发生后变为零。
4	EV0_GEN	R/W	0h	事件 0 启用。 0：通道 1 不控制事件 0。 1：通道 1 控制事件 0。 当 0 < CCACT < 8 时，EV0_GEN 在捕获或比较事件发生后变为零。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-148. CH1EVCFG 寄存器字段说明（续）

位	字段	类型	复位	说明
3-0	CCACT	R/W	0h	捕获-比较操作。 捕获-比较操作定义了 15 种不同的通道功能，这些功能会利用捕获、比较和归零事件。 0h = 禁用通道。 1h = 在捕获时复位，然后禁用通道。 通道功能序列： - 在捕获事件发生时启用相关事件，并将 CNTR.VALUE 复制到 CH1CC.VALUE。 - 禁用通道。 主要使用场景是在启动定时器之前选择此功能。 如果需要在 CTL.MODE 不为 DIS 的情况下选择此功能，请按照以下步骤操作： - 将 CCACT 设置为 SET_ON_CAPT，且不启用事件。 - 配置 CH1CCFG（可选）。 - 等待 PRECFG 定义的三个定时器时钟周期，然后将 CCACT 设置为 SET_ON_CAPT_DIS。事件启用为可选项。 这些步骤可防止因边沿检测电路中残留的旧信号值而触发捕获事件。 2h = 在零时清除，在比较时翻转，随后禁用通道。 通道功能序列： - 当 CNTR.VALUE = 0 时，清除已启用的事件。 - 当 CH1CC.VALUE = CNTR.VALUE 时，翻转已启用的事件。 - 禁用通道。 当 CH1CC.VALUE = 0 且 CNTR.VALUE = 0 时，设置已启用的事件。 3h = 在零时复位，在比较匹配时翻转，随后禁用通道。 通道功能序列： - 当 CNTR.VALUE = 0 时，置位已启用的事件。 - 当 CH1CC.VALUE = CNTR.VALUE 时，翻转已启用的事件。 - 禁用通道。 当 CH1CC.VALUE = 0 且 CNTR.VALUE = 0 时，已启用的事件会被清除。 4h = 在比较时清除，然后禁用通道。 通道功能序列： - 当 CH1CC.VALUE = CNTR.VALUE 时，清除已启用的事件。 - 禁用通道。 5h = 在比较时复位，然后禁用通道。 通道功能序列： - 当 CH1CC.VALUE = CNTR.VALUE 时，复位已启用的事件。 - 禁用通道。 6h = 在比较时翻转，然后禁用通道。 通道功能序列： - 当 CH1CC.VALUE = CNTR.VALUE 时，翻转已启用的事件。 - 禁用通道。 7h = 在比较时产生脉冲，然后禁用通道。 通道功能序列： - 当 CH1CC.VALUE = CNTR.VALUE 时，为已启用的事件产生脉冲。 - 禁用通道。 该事件在两个计时器时钟周期内保持高电平。 8h = 周期和脉冲宽度测量。 根据 CH1CCFG.EDGE 指定的信号边沿，持续捕获由 CH1CCFG.CAPT_SRC 选定信号的周期和脉冲宽度。 当 CH1CC.VALUE 包含信号周期且 CH1PCC.VALUE 包含信号脉冲宽度时，触发已启用的事件。 注意： - 请确保在 CTL.MODE 为 DIS 时配置 CH1CCFG.CAPT_SRC 和 CCACT，然后再将 CTL.MODE 设置为 UP_ONCE 或 UP_PER。 - 当 CH1CC.VALUE 包含信号周期时，计数器将在选定的定时器模式下重新开始计数。 - 如果多个通道使用此功能，这些通道将依次执行该功能。编号最小的通道具有最高优先级并首先执行该功能。下一次测量将在当前测量成功完成或超时后开始。当计数器值达到目标值时，即发生超时。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-148. CH1EVCFG 寄存器字段说明（续）

位	字段	类型	复位	说明
				- 若要监测超时事件，请将另一个通道配置为 SET_ON_CAPT 模式。 信号特性要求： - 信号周期 >= 2 * (1 + PRECFG.CLKDIV) * 定时器时钟周期。 - 信号周期 <= 65535 * (1 + PRECFG.CLKDIV) * 定时器时钟周期。 - 信号低电平和高电平持续时间 >= (1 + PRECFG.CLKDIV) * 定时器时钟周期。 9h = 在捕获时重复置位。 通道功能序列： - 在捕获事件发生时启用相关事件，并将 CNTR.VALUE 复制到 CH1CC.VALUE。 主要使用场景是在启动定时器之前选择此功能。 如果需要在 CTL.MODE 不为 DIS 的情况下选择此功能，请按照以下步骤操作： - 选择此功能，但不启用事件。 - 配置 CH1CCFG（可选）。 - 在启用事件之前，等待 PRECFG 所定义的三个定时器时钟周期。 这些步骤可防止因边沿检测电路中残留的旧信号值而触发捕获事件。 Ah = 在零时清除，在比较时反复翻转。 通道功能序列： - 当 CNTR.VALUE = 0 时，清除已启用的事件。 - 当 CH1CC.VALUE = CNTR.VALUE 时，翻转已启用的事件。 将 CTL.MODE 设置为 UPDWN_PER 以生成中心对齐 PWM。占空比计算方式如下： 当 CH1CC.VALUE <= TARGET.VALUE 时： 占空比 = 1 - (CH1CC.VALUE / TARGET.VALUE)。 当 CH1CC.VALUE > TARGET.VALUE 时： 占空比 = 0。 当 CH1CC.VALUE = 0 且 CNTR.VALUE = 0 时，设置已启用的事件。 Bh = 在零时置位，在比较时反复翻转。 通道功能序列： - 当 CNTR.VALUE = 0 时，置位已启用的事件。 - 当 CH1CC.VALUE = CNTR.VALUE 时，翻转已启用的事件。 将 CTL.MODE 设置为 UP_PER 以生成边沿对齐 PWM。占空比计算方式如下： 当 CH1CC.VALUE <= TARGET.VALUE 时： 占空比 = CH1CC.VALUE / (TARGET.VALUE + 1)。 当 CH1CC.VALUE > TARGET.VALUE 时： 占空比 = 1。 当 CH1CC.VALUE = 0 且 CNTR.VALUE = 0 时，已启用的事件会被清除。 Ch = 在比较时反复清除。 通道功能序列： - 当 CH1CC.VALUE = CNTR.VALUE 时，清除已启用的事件。 Dh = 在比较时反复置位。 通道功能序列： - 当 CH1CC.VALUE = CNTR.VALUE 时，复位已启用的事件。 Eh = 在比较时反复翻转。 通道功能序列： - 当 CH1CC.VALUE = CNTR.VALUE 时，翻转已启用的事件。 Fh = 在比较时反复产生脉冲。 通道功能序列： - 当 CH1CC.VALUE = CNTR.VALUE 时，为已启用的事件产生脉冲。该事件在两个计时器时钟周期内保持高电平。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.7.13 CH1CCFG 寄存器（偏移 = 94h）[复位 = 00000000h]

图 19-131 展示了 CH1CCFG，表 19-149 对其进行了介绍。

返回到汇总表。

通道 1 捕获配置

图 19-131. CH1CCFG 寄存器

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

表 19-149. CH1CCFG 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-149. CH1CCFG 寄存器字段说明（续）

位	字段	类型	复位	说明
6-1	CAPT_SRC	R/W	0h	从异步 AUX 事件总线中选择捕获信号源。 所选信号进入边沿检测电路。在以下情况下，可能会发生错误的捕获事件： - 边沿检测电路中包含过期的信号采样，且在未执行 CH1EVCFG 所述的刷新操作的情况下启用了该电路 - 在 CTL.MODE 不为 DIS 的状态下重新配置了该寄存器。 您可以避免误捕获事件。当所需的通道功能为以下情况时： - SET_ON_CAPT_DIS：请参阅 CH1EVCFG.CCACT 中关于 SET_ON_CAPT_DIS 的说明。 - SET_ON_CAPT：请参阅 CH1EVCFG.CCACT 中关于 SET_ON_CAPT 的说明。 - PER_PULSE_WIDTH_MEAS：请参阅 CH1EVCFG.CCACT 中关于 PER_PULSE_WIDTH_MEAS 的说明。 如果写入非枚举值，其行为与 NO_EVENT 相同；读取时将返回写入的值。 0h = AUX_EVCTL:EVSTAT0.AUXIO0 1h = AUX_EVCTL:EVSTAT0.AUXIO1 2h = AUX_EVCTL:EVSTAT0.AUXIO2 3h = AUX_EVCTL:EVSTAT0.AUXIO3 4h = AUX_EVCTL:EVSTAT0.AUXIO4 5h = AUX_EVCTL:EVSTAT0.AUXIO5 6h = AUX_EVCTL:EVSTAT0.AUXIO6 7h = AUX_EVCTL:EVSTAT0.AUXIO7 8h = AUX_EVCTL:EVSTAT0.AUXIO8 9h = AUX_EVCTL:EVSTAT0.AUXIO9 Ah = AUX_EVCTL:EVSTAT0.AUXIO10 Bh = AUX_EVCTL:EVSTAT0.AUXIO11 Ch = AUX_EVCTL:EVSTAT0.AUXIO12 Dh = AUX_EVCTL:EVSTAT0.AUXIO13 Eh = AUX_EVCTL:EVSTAT0.AUXIO14 Fh = AUX_EVCTL:EVSTAT0.AUXIO15 10h = AUX_EVCTL:EVSTAT1.AUXIO16 11h = AUX_EVCTL:EVSTAT1.AUXIO17 12h = AUX_EVCTL:EVSTAT1.AUXIO18 13h = AUX_EVCTL:EVSTAT1.AUXIO19 14h = AUX_EVCTL:EVSTAT1.AUXIO20 15h = AUX_EVCTL:EVSTAT1.AUXIO21 16h = AUX_EVCTL:EVSTAT1.AUXIO22 17h = AUX_EVCTL:EVSTAT1.AUXIO23 18h = AUX_EVCTL:EVSTAT1.AUXIO24 19h = AUX_EVCTL:EVSTAT1.AUXIO25 1Ah = AUX_EVCTL:EVSTAT1.AUXIO26 1Bh = AUX_EVCTL:EVSTAT1.AUXIO27 1Ch = AUX_EVCTL:EVSTAT1.AUXIO28 1Dh = AUX_EVCTL:EVSTAT1.AUXIO29 1Eh = AUX_EVCTL:EVSTAT1.AUXIO30 1Fh = AUX_EVCTL:EVSTAT1.AUXIO31 20h = AUX_EVCTL:EVSTAT2.MANUAL_EV 21h = AUX_EVCTL:EVSTAT2.AON_RTC_CH2 22h = AUX_EVCTL:EVSTAT2.AON_RTC_CH2_DLY 23h = AUX_EVCTL:EVSTAT2.AON_RTC_4KHZ 24h = AUX_EVCTL:EVSTAT2.AON_BATMON_BAT_UPD 25h = AUX_EVCTL:EVSTAT2.AON_BATMON_TEMP_UPD 26h = AUX_EVCTL:EVSTAT2.SCLK_LF 27h = AUX_EVCTL:EVSTAT2.PWR_DWN 28h = AUX_EVCTL:EVSTAT2.MCU_ACTIVE 29h = AUX_EVCTL:EVSTAT2.VDDR_RECHARGE 2Ah = AUX_EVCTL:EVSTAT2.ACLK_REF 2Bh = AUX_EVCTL:EVSTAT2.MCU_EV 2Ch = AUX_EVCTL:EVSTAT2.MCU_OBSMUX0 2Dh = AUX_EVCTL:EVSTAT2.MCU_OBSMUX1 2Eh = AUX_EVCTL:EVSTAT2.AUX_COMPA 2Fh = AUX_EVCTL:EVSTAT2.AUX_COMPB 30h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV0

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-149. CH1CCFG 寄存器字段说明（续）

位	字段	类型	复位	说明
				31h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV1 32h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV2 33h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV3 35h = AUX_EVCTL:EVSTAT3.AUX_TIMER1_EV 36h = AUX_EVCTL:EVSTAT3.AUX_TIMER0_EV 37h = AUX_EVCTL:EVSTAT3.AUX_TDC_DONE 38h = AUX_EVCTL:EVSTAT3.AUX_ISRC_RESET_N 39h = AUX_EVCTL:EVSTAT3.AUX_ADC_DONE 3Ah = AUX_EVCTL:EVSTAT3.AUX_ADC_IRQ 3Bh = AUX_EVCTL:EVSTAT3.AUX_ADC_FIFO_ALMOST_FULL 3Ch = AUX_EVCTL:EVSTAT3.AUX_ADC_FIFO_NOT_EMPTY 3Dh = AUX_EVCTL:EVSTAT3.AUX_SMPH_AUTOTAKE_DONE 3Fh = 无事件。
0	EDGE	R/W	0h	边沿配置。 通道在由 CAPT_SRC 选定的信号源上，于选定边沿处捕获计数值。 参阅 CH1EVCFG.CCACT。 0h = 在 CAPT_SRC 的下降沿捕获 CNTR.VALUE。 1h = 在 CAPT_SRC 的上升沿捕获 CNTR.VALUE。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.7.14 CH1PCC 寄存器（偏移 = 98h）[复位 = 00000000h]

图 19-132 展示了 CH1PCC，表 19-150 对其进行了介绍。

返回到汇总表。

通道 1 流水线捕获对比情况

图 19-132. CH1PCC 寄存器

31	30	29	28	27	26	25	24	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																值															
R-0h																R/W-0h															

表 19-150. CH1PCC 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	值	R/W	0h	流水线捕获比较值。 16 位用户自定义的流水线比较值，或通道更新后的捕获值。 比较模式： 当下一个 CNTR.VALUE 为零且 CTL.MODE 不为 DIS 时，VALUE 的更新值将传输至 CH1CC.VALUE。此机制适用于 PWM 生成，并能防止生成信号的边沿出现抖动。 捕捉模式： 当 CH1EVCFG.CCACT 设置为 PER_PULSE_WIDTH_MEAS 时，VALUE 存储所选信号的低电平或高电平脉冲宽度。具体测量对象由 CH1CCFG.EDGE 和 CH1CCFG.CAPT_SRC 决定。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.7.15 CH1CC 寄存器（偏移 = 9Ch）[复位 = 00000000h]

图 19-133 展示了 CH1CC，表 19-151 对其进行了介绍。

返回到汇总表。

通道 1 捕获对标情况

图 19-133. CH1CC 寄存器

31	30	29	28	27	26	25	24	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																值															
R-0h																R/W-0h															

表 19-151. CH1CC 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	值	R/W	0h	捕获比较值。 16 位用户定义的比较值，或通道更新后的捕获值。 比较模式： 将 VALUE 与 CNTR.VALUE 进行比较；当两者相等时，根据 CH1EVCFG.CCACT 的设定生成事件。 捕捉模式： 当发生捕获事件时，当前的计数器值被存储到 VALUE 中。 CH1EVCFG.CCACT 确定值是信号周期还是常规捕获值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.7.16 CH2EVCFG 寄存器（偏移 = A0h）[复位 = 00000000h]

图 19-134 展示了 CH2EVCFG，表 19-152 对其进行了介绍。

返回到汇总表。

通道 2 事件配置
该寄存器用于配置通道功能并启用事件输出。
每个通道都包含一个带存储功能的边沿检测电路。该电路的运行机制如下：
- 当 CCACT 选择捕获功能且 CTL.MODE 不为 DIS 时，电路处于启用状态。
- 当 CCACT 选择捕获功能且把 CTL.MODE 从 DIS 切换为其他模式时：电路会被刷新。
刷新操作使用两个 AUX_SYSIF:TIMER2CLKCTL.SRC 时钟周期。它能防止因存储在边沿检测电路中的信号值过期而引发的捕获事件。

图 19-134. CH2EVCFG 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
EV3_GEN	EV2_GEN	EV1_GEN	EV0_GEN	CCACT			
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h			

表 19-152. CH2EVCFG 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7	EV3_GEN	R/W	0h	事件 3 启用。 0：通道 2 不控制事件 3。 1：通道 2 控制事件 3。 当 0 < CCACT < 8 时，EV3_GEN 在捕获或比较事件发生后变为零。
6	EV2_GEN	R/W	0h	事件 2 启用。 0：通道 2 不控制事件 2。 1：通道 2 控制事件 2。 当 0 < CCACT < 8 时，EV2_GEN 在捕获或比较事件发生后变为零。
5	EV1_GEN	R/W	0h	事件 1 启用。 0：通道 2 不控制事件 1。 1：通道 2 控制事件 1。 当 0 < CCACT < 8 时，EV1_GEN 在捕获或比较事件发生后变为零。
4	EV0_GEN	R/W	0h	事件 0 启用。 0：通道 2 不控制事件 0。 1：通道 2 控制事件 0。 当 0 < CCACT < 8 时，EV0_GEN 在捕获或比较事件发生后变为零。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-152. CH2EVCFG 寄存器字段说明（续）

位	字段	类型	复位	说明
3-0	CCACT	R/W	0h	捕获-比较操作。 捕获-比较操作定义了 15 种不同的通道功能，这些功能会利用捕获、比较和归零事件。 0h = 禁用通道。 1h = 在捕获时复位，然后禁用通道。 通道功能序列： - 在捕获事件发生时启用相关事件，并将 CNTR.VALUE 复制到 CH2CC.VALUE。 - 禁用通道。 主要使用场景是在启动定时器之前选择此功能。 如果需要在 CTL.MODE 不为 DIS 的情况下选择此功能，请按照以下步骤操作： - 设为 SET_ON_CAPT，且不启用事件。 - 配置 CH2CCFG（可选）。 - 等待 PRECFG 定义的三个定时器时钟周期，然后设为 SET_ON_CAPT_DIS。事件启用为可选项。 这些步骤可防止因边沿检测电路中残留的旧信号值而触发捕获事件。 2h = 在零时清除，在比较时翻转，随后禁用通道。 通道功能序列： - 当 CNTR.VALUE = 0 时，清除已启用的事件。 - 当 CH2CC.VALUE = CNTR.VALUE 时，翻转已启用的事件。 - 禁用通道。 当 CH2CC.VALUE = 0 且 CNTR.VALUE = 0 时，设置已启用的事件。 3h = 在零时复位，在比较匹配时翻转，随后禁用通道。 通道功能序列： - 当 CNTR.VALUE = 0 时，置位已启用的事件。 - 当 CH2CC.VALUE = CNTR.VALUE 时，翻转已启用的事件。 - 禁用通道。 当 CH2CC.VALUE = 0 且 CNTR.VALUE = 0 时，已启用的事件会被清除。 4h = 在比较时清除，然后禁用通道。 通道功能序列： - 当 CH2CC.VALUE = CNTR.VALUE 时，清除已启用的事件。 - 禁用通道。 5h = 在比较时复位，然后禁用通道。 通道功能序列： - 当 CH2CC.VALUE = CNTR.VALUE 时，复位已启用的事件。 - 禁用通道。 6h = 在比较时翻转，然后禁用通道。 通道功能序列： - 当 CH2CC.VALUE = CNTR.VALUE 时，翻转已启用的事件。 - 禁用通道。 7h = 在比较时产生脉冲，然后禁用通道。 通道功能序列： - 当 CH2CC.VALUE = CNTR.VALUE 时，为已启用的事件产生脉冲。 - 禁用通道。 该事件在两个计时器时钟周期内保持高电平。 8h = 周期和脉冲宽度测量。 根据 CH2CCFG.EDGE 指定的信号边沿，持续捕获由 CH2CCFG.CAPT_SRC 选定信号的周期和脉冲宽度。 当 CH2CC.VALUE 包含信号周期且 CH2PCC.VALUE 包含信号脉冲宽度时，触发已启用的事件。 注意： - 请确保在 CTL.MODE 为 DIS 时配置 CH2CCFG.CAPT_SRC 和 CCACT，然后再将 CTL.MODE 设置为 UP_ONCE 或 UP_PER。 - 当 CH2CC.VALUE 包含信号周期时，计数器将在选定的定时器模式下重新开始计数。 - 如果多个通道使用此功能，这些通道将依次执行该功能。编号最小的通道具有最高优先级并首先执行该功能。下一次测量将在当前测量成功完成或超时后开始。当计数器值达到目标值时，即发生超时。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-152. CH2EVCFG 寄存器字段说明（续）

位	字段	类型	复位	说明
				- 若要监测超时事件，请将另一个通道配置为 SET_ON_CAPT 模式。 信号特性要求： - 信号周期 >= 2 * (1 + PRECFG.CLKDIV) * 定时器时钟周期。 - 信号周期 <= 65535 * (1 + PRECFG.CLKDIV) * 定时器时钟周期。 - 信号低电平和高电平持续时间 >= (1 + PRECFG.CLKDIV) * 定时器时钟周期。 9h = 在捕获时重复置位。 通道功能序列： - 在捕获事件发生时启用相关事件，并将 CNTR.VALUE 复制到 CH2CC.VALUE。 主要使用场景是在启动定时器之前选择此功能。 如果需要在 CTL.MODE 不为 DIS 的情况下选择此功能，请按照以下步骤操作： - 选择此功能，但不启用事件。 - 配置 CH2CCFG（可选）。 - 在启用事件之前，等待 PRECFG 所定义的三个定时器时钟周期。 这些步骤可防止因边沿检测电路中残留的旧信号值而触发捕获事件。 Ah = 在零时清除，在比较时反复翻转。 通道功能序列： - 当 CNTR.VALUE = 0 时，清除已启用的事件。 - 当 CH2CC.VALUE = CNTR.VALUE 时，翻转已启用的事件。 将 CTL.MODE 设置为 UPDWN_PER 以生成中心对齐 PWM。占空比计算方式如下： 当 CH2CC.VALUE <= TARGET.VALUE 时： 占空比 = 1 - (CH2CC.VALUE / TARGET.VALUE)。 当 CH2CC.VALUE > TARGET.VALUE 时： 占空比 = 0。 当 CH2CC.VALUE = 0 且 CNTR.VALUE = 0 时，设置已启用的事件。 Bh = 在零时置位，在比较时反复翻转。 通道功能序列： - 当 CNTR.VALUE = 0 时，置位已启用的事件。 - 当 CH2CC.VALUE = CNTR.VALUE 时，翻转已启用的事件。 将 CTL.MODE 设置为 UP_PER 以生成边沿对齐 PWM。占空比计算方式如下： 当 CH2CC.VALUE <= TARGET.VALUE 时： 占空比 = CH2CC.VALUE / (TARGET.VALUE + 1)。 当 CH2CC.VALUE > TARGET.VALUE 时： 占空比 = 1。 当 CH2CC.VALUE = 0 且 CNTR.VALUE = 0 时，已启用的事件会被清除。 Ch = 在比较时反复清除。 通道功能序列： - 当 CH2CC.VALUE = CNTR.VALUE 时，清除已启用的事件。 Dh = 在比较时反复置位。 通道功能序列： - 当 CH2CC.VALUE = CNTR.VALUE 时，复位已启用的事件。 Eh = 在比较时反复翻转。 通道功能序列： - 当 CH2CC.VALUE = CNTR.VALUE 时，翻转已启用的事件。 Fh = 在比较时反复产生脉冲。 通道功能序列： - 当 CH2CC.VALUE = CNTR.VALUE 时，为已启用的事件产生脉冲。该事件在两个计时器时钟周期内保持高电平。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.7.17 CH2CCFG 寄存器（偏移 = A4h）[复位 = 00000000h]

图 19-135 展示了 CH2CCFG，表 19-153 对其进行了介绍。

返回到汇总表。

通道 2 捕获配置

图 19-135. CH2CCFG 寄存器

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

表 19-153. CH2CCFG 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-153. CH2CCFG 寄存器字段说明（续）

位	字段	类型	复位	说明
6-1	CAPT_SRC	R/W	0h	从异步 AUX 事件总线中选择捕获信号源。 所选信号进入边沿检测电路。在以下情况下，可能会发生错误的捕获事件： - 边沿检测电路中包含过期的信号采样，且在未执行 CH2EVCFG 所述的刷新操作的情况下启用了该电路 - 在 CTL.MODE 不为 DIS 的状态下重新配置了该寄存器。 您可以避免误捕获事件。当所需的通道功能为以下情况时： - SET_ON_CAPT_DIS：请参阅 CH2EVCFG.CCACT 中关于 SET_ON_CAPT_DIS 的说明。 - SET_ON_CAPT：请参阅 CH2EVCFG.CCACT 中关于 SET_ON_CAPT 的说明。 - PER_PULSE_WIDTH_MEAS：请参阅 CH2EVCFG.CCACT 中关于 PER_PULSE_WIDTH_MEAS 的说明。 如果写入非枚举值，其行为与 NO_EVENT 相同；读取时将返回写入的值。 0h = AUX_EVCTL:EVSTAT0.AUXIO0 1h = AUX_EVCTL:EVSTAT0.AUXIO1 2h = AUX_EVCTL:EVSTAT0.AUXIO2 3h = AUX_EVCTL:EVSTAT0.AUXIO3 4h = AUX_EVCTL:EVSTAT0.AUXIO4 5h = AUX_EVCTL:EVSTAT0.AUXIO5 6h = AUX_EVCTL:EVSTAT0.AUXIO6 7h = AUX_EVCTL:EVSTAT0.AUXIO7 8h = AUX_EVCTL:EVSTAT0.AUXIO8 9h = AUX_EVCTL:EVSTAT0.AUXIO9 Ah = AUX_EVCTL:EVSTAT0.AUXIO10 Bh = AUX_EVCTL:EVSTAT0.AUXIO11 Ch = AUX_EVCTL:EVSTAT0.AUXIO12 Dh = AUX_EVCTL:EVSTAT0.AUXIO13 Eh = AUX_EVCTL:EVSTAT0.AUXIO14 Fh = AUX_EVCTL:EVSTAT0.AUXIO15 10h = AUX_EVCTL:EVSTAT1.AUXIO16 11h = AUX_EVCTL:EVSTAT1.AUXIO17 12h = AUX_EVCTL:EVSTAT1.AUXIO18 13h = AUX_EVCTL:EVSTAT1.AUXIO19 14h = AUX_EVCTL:EVSTAT1.AUXIO20 15h = AUX_EVCTL:EVSTAT1.AUXIO21 16h = AUX_EVCTL:EVSTAT1.AUXIO22 17h = AUX_EVCTL:EVSTAT1.AUXIO23 18h = AUX_EVCTL:EVSTAT1.AUXIO24 19h = AUX_EVCTL:EVSTAT1.AUXIO25 1Ah = AUX_EVCTL:EVSTAT1.AUXIO26 1Bh = AUX_EVCTL:EVSTAT1.AUXIO27 1Ch = AUX_EVCTL:EVSTAT1.AUXIO28 1Dh = AUX_EVCTL:EVSTAT1.AUXIO29 1Eh = AUX_EVCTL:EVSTAT1.AUXIO30 1Fh = AUX_EVCTL:EVSTAT1.AUXIO31 20h = AUX_EVCTL:EVSTAT2.MANUAL_EV 21h = AUX_EVCTL:EVSTAT2.AON_RTC_CH2 22h = AUX_EVCTL:EVSTAT2.AON_RTC_CH2_DLY 23h = AUX_EVCTL:EVSTAT2.AON_RTC_4KHZ 24h = AUX_EVCTL:EVSTAT2.AON_BATMON_BAT_UPD 25h = AUX_EVCTL:EVSTAT2.AON_BATMON_TEMP_UPD 26h = AUX_EVCTL:EVSTAT2.SCLK_LF 27h = AUX_EVCTL:EVSTAT2.PWR_DWN 28h = AUX_EVCTL:EVSTAT2.MCU_ACTIVE 29h = AUX_EVCTL:EVSTAT2.VDDR_RECHARGE 2Ah = AUX_EVCTL:EVSTAT2.ACLK_REF 2Bh = AUX_EVCTL:EVSTAT2.MCU_EV 2Ch = AUX_EVCTL:EVSTAT2.MCU_OBSMUX0 2Dh = AUX_EVCTL:EVSTAT2.MCU_OBSMUX1 2Eh = AUX_EVCTL:EVSTAT2.AUX_COMPA 2Fh = AUX_EVCTL:EVSTAT2.AUX_COMPB 30h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV0

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-153. CH2CCFG 寄存器字段说明（续）

位	字段	类型	复位	说明
				31h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV1 32h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV2 33h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV3 35h = AUX_EVCTL:EVSTAT3.AUX_TIMER1_EV 36h = AUX_EVCTL:EVSTAT3.AUX_TIMER0_EV 37h = AUX_EVCTL:EVSTAT3.AUX_TDC_DONE 38h = AUX_EVCTL:EVSTAT3.AUX_ISRC_RESET_N 39h = AUX_EVCTL:EVSTAT3.AUX_ADC_DONE 3Ah = AUX_EVCTL:EVSTAT3.AUX_ADC_IRQ 3Bh = AUX_EVCTL:EVSTAT3.AUX_ADC_FIFO_ALMOST_FULL 3Ch = AUX_EVCTL:EVSTAT3.AUX_ADC_FIFO_NOT_EMPTY 3Dh = AUX_EVCTL:EVSTAT3.AUX_SMPH_AUTOTAKE_DONE 3Fh = 无事件。
0	EDGE	R/W	0h	边沿配置。 通道在由 CAPT_SRC 选定的信号源上，于选定边沿处捕获计数值。 参阅 CH2EVCFG.CCACT。 0h = 在 CAPT_SRC 的下降沿捕获 CNTR.VALUE。 1h = 在 CAPT_SRC 的上升沿捕获 CNTR.VALUE。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.7.18 CH2PCC 寄存器（偏移 = A8h）[复位 = 00000000h]

图 19-136 展示了 CH2PCC，表 19-154 对其进行了介绍。

返回到汇总表。

通道 2 流水线捕获对比情况

图 19-136. CH2PCC 寄存器

31	30	29	28	27	26	25	24	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																值															
R-0h																R/W-0h															

表 19-154. CH2PCC 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	值	R/W	0h	流水线捕获比较值。 16 位用户自定义的流水线比较值，或通道更新后的捕获值。 比较模式： 当下一个 CNTR.VALUE 为零且 CTL.MODE 不为 DIS 时，VALUE 的更新值将传输至 CH2CC.VALUE。此机制适用于 PWM 生成，并能防止生成信号的边沿出现抖动。 捕捉模式： 当 CH2EVCFG.CCACT 设置为 PER_PULSE_WIDTH_MEAS 时，VALUE 存储所选信号的低电平或高电平脉冲宽度。这由 CH2CCFG.EDGE 和 CH2CCFG.CAPT_SRC 指定。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.7.19 CH2CC 寄存器（偏移 = ACh）[复位 = 00000000h]

图 19-137 展示了 CH2CC，表 19-155 中对此进行了介绍。

返回到汇总表。

通道 2 捕获对标情况

图 19-137. CH2CC 寄存器

31	30	29	28	27	26	25	24	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																值															
R-0h																R/W-0h															

表 19-155. CH2CC 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	值	R/W	0h	捕获比较值。 16 位用户定义的比较值，或通道更新后的捕获值。 比较模式： 将 VALUE 与 CNTR.VALUE 进行比较；当两者相等时，根据 CH2EVCFG.CCACT 的设定生成事件。 捕捉模式： 当发生捕获事件时，当前的计数器值被存储到 VALUE 中。 CH2EVCFG.CCACT 确定值是信号周期还是常规捕获值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.7.20 CH3EVCFG 寄存器（偏移 = B0h）[复位 = 00000000h]

图 19-138 展示了 CH3EVCFG，表 19-156 对其进行了介绍。

返回到汇总表。

通道 3 事件配置
该寄存器用于配置通道功能并启用事件输出。
每个通道都包含一个带存储功能的边沿检测电路。该电路的运行机制如下：
- 当 CCACT 选择捕获功能且 CTL.MODE 不为 DIS 时，电路处于启用状态。
- 当 CCACT 选择捕获功能且把 CTL.MODE 从 DIS 切换为其他模式时：电路会被刷新。
刷新操作使用两个 AUX_SYSIF:TIMER2CLKCTL.SRC 时钟周期。它能防止因存储在边沿检测电路中的信号值过期而引发的捕获事件。

图 19-138. CH3EVCFG 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
EV3_GEN	EV2_GEN	EV1_GEN	EV0_GEN	CCACT			
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h			

表 19-156. CH3EVCFG 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7	EV3_GEN	R/W	0h	事件 3 启用。 0：通道 3 不控制事件 3。 1：通道 3 控制事件 3。 当 0 < CCACT < 8 时，EV3_GEN 在捕获或比较事件发生后变为零。
6	EV2_GEN	R/W	0h	事件 2 启用。 0：通道 3 不控制事件 2。 1：通道 3 控制事件 2。 当 0 < CCACT < 8 时，EV2_GEN 在捕获或比较事件发生后变为零。
5	EV1_GEN	R/W	0h	事件 1 启用。 0：通道 3 不控制事件 1。 1：通道 3 控制事件 1。 当 0 < CCACT < 8 时，EV1_GEN 在捕获或比较事件发生后变为零。
4	EV0_GEN	R/W	0h	事件 0 启用。 0：通道 3 不控制事件 0。 1：通道 3 控制事件 0。 当 0 < CCACT < 8 时，EV0_GEN 在捕获或比较事件发生后变为零。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-156. CH3EVCFG 寄存器字段说明（续）

位	字段	类型	复位	说明
3-0	CCACT	R/W	0h	捕获-比较操作。 捕获-比较操作定义了 15 种不同的通道功能，这些功能会利用捕获、比较和归零事件。 0h = 禁用通道。 1h = 在捕获时复位，然后禁用通道。 通道功能序列： - 在捕获事件发生时启用相关事件，并将 CNTR.VALUE 复制到 CH3CC.VALUE。 - 禁用通道。 主要使用场景是在启动定时器之前选择此功能。 如果需要在 CTL.MODE 不为 DIS 的情况下选择此功能，请按照以下步骤操作： - 将 CCACT 设置为 SET_ON_CAPT，且不启用事件。 - 配置 CH3CCFG（可选）。 - 等待 PRECFG 定义的三个定时器时钟周期，然后将 CCACT 设置为 SET_ON_CAPT_DIS。事件启用为可选项。 这些步骤可防止因边沿检测电路中残留的旧信号值而触发捕获事件。 2h = 在零时清除，在比较时翻转，随后禁用通道。 通道功能序列： - 当 CNTR.VALUE = 0 时，清除已启用的事件。 - 当 CH3CC.VALUE = CNTR.VALUE 时，翻转已启用的事件。 - 禁用通道。 当 CH3CC.VALUE = 0 且 CNTR.VALUE = 0 时，设置已启用的事件。 3h = 在零时复位，在比较匹配时翻转，随后禁用通道。 通道功能序列： - 当 CNTR.VALUE = 0 时，置位已启用的事件。 - 当 CH3CC.VALUE = CNTR.VALUE 时，翻转已启用的事件。 - 禁用通道。 当 CH3CC.VALUE = 0 且 CNTR.VALUE = 0 时，已启用的事件会被清除。 4h = 在比较时清除，然后禁用通道。 通道功能序列： - 当 CH3CC.VALUE = CNTR.VALUE 时，清除已启用的事件。 - 禁用通道。 5h = 在比较时复位，然后禁用通道。 通道功能序列： - 当 CH3CC.VALUE = CNTR.VALUE 时，复位已启用的事件。 - 禁用通道。 6h = 在比较时翻转，然后禁用通道。 通道功能序列： - 当 CH3CC.VALUE = CNTR.VALUE 时，翻转已启用的事件。 - 禁用通道。 7h = 在比较时产生脉冲，然后禁用通道。 通道功能序列： - 当 CH3CC.VALUE = CNTR.VALUE 时，为已启用的事件产生脉冲。 - 禁用通道。 该事件在两个计时器时钟周期内保持高电平。 8h = 周期和脉冲宽度测量。 根据 CH3CCFG.EDGE 指定的信号边沿，持续捕获由 CH3CCFG.CAPT_SRC 选定信号的周期和脉冲宽度。 当 CH3CC.VALUE 包含信号周期且 CH3PCC.VALUE 包含信号脉冲宽度时，触发已启用的事件。 注意： - 请确保在 CTL.MODE 为 DIS 时配置 CH3CCFG.CAPT_SRC 和 CCACT，然后再将 CTL.MODE 设置为 UP_ONCE 或 UP_PER。 - 当 CH3CC.VALUE 包含信号周期时，计数器将在选定的定时器模式下重新开始计数。 - 如果多个通道使用此功能，这些通道将依次执行该功能。编号最小的通道具有最高优先级并首先执行该功能。下一次测量将在当前测量成功完成或超时后开始。当计数器值达到目标值时，即发生超时。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-156. CH3EVCFG 寄存器字段说明（续）

位	字段	类型	复位	说明
				- 若要监测超时事件，请将另一个通道配置为 SET_ON_CAPT 模式。 信号特性要求： - 信号周期 >= 2 * (1 + PRECFG.CLKDIV) * 定时器时钟周期。 - 信号周期 <= 65535 * (1 + PRECFG.CLKDIV) * 定时器时钟周期。 - 信号低电平和高电平持续时间 >= (1 + PRECFG.CLKDIV) * 定时器时钟周期。 9h = 在捕获时重复置位。 通道功能序列： - 在捕获事件发生时启用相关事件，并将 CNTR.VALUE 复制到 CH3CC.VALUE。 主要使用场景是在启动定时器之前选择此功能。 如果需要在 CTL.MODE 不为 DIS 的情况下选择此功能，请按照以下步骤操作： - 选择此功能，但不启用事件。 - 配置 CH3CCFG（可选）。 - 在启用事件之前，等待 PRECFG 所定义的三个定时器时钟周期。 这些步骤可防止因边沿检测电路中残留的旧信号值而触发捕获事件。 Ah = 在零时清除，在比较时反复翻转。 通道功能序列： - 当 CNTR.VALUE = 0 时，清除已启用的事件。 - 当 CH3CC.VALUE = CNTR.VALUE 时，翻转已启用的事件。 将 CTL.MODE 设置为 UPDWN_PER 以生成中心对齐 PWM。占空比计算方式如下： 当 CH3CC.VALUE <= TARGET.VALUE 时： 占空比 = 1 - (CH3CC.VALUE / TARGET.VALUE)。 当 CH3CC.VALUE > TARGET.VALUE 时： 占空比 = 0。 当 CH3CC.VALUE = 0 且 CNTR.VALUE = 0 时，设置已启用的事件。 Bh = 在零时置位，在比较时反复翻转。 通道功能序列： - 当 CNTR.VALUE = 0 时，置位已启用的事件。 - 当 CH3CC.VALUE = CNTR.VALUE 时，翻转已启用的事件。 将 CTL.MODE 设置为 UP_PER 以生成边沿对齐 PWM。占空比计算方式如下： 当 CH3CC.VALUE <= TARGET.VALUE 时： 占空比 = CH3CC.VALUE / (TARGET.VALUE + 1)。 当 CH3CC.VALUE > TARGET.VALUE 时： 占空比 = 1。 当 CH3CC.VALUE = 0 且 CNTR.VALUE = 0 时，已启用的事件会被清除。 Ch = 在比较时反复清除。 通道功能序列： - 当 CH3CC.VALUE = CNTR.VALUE 时，清除已启用的事件。 Dh = 在比较时反复置位。 通道功能序列： - 当 CH3CC.VALUE = CNTR.VALUE 时，复位已启用的事件。 Eh = 在比较时反复翻转。 通道功能序列： - 当 CH3CC.VALUE = CNTR.VALUE 时，翻转已启用的事件。 Fh = 在比较时反复产生脉冲。 通道功能序列： - 当 CH3CC.VALUE = CNTR.VALUE 时，为已启用的事件产生脉冲。该事件在两个计时器时钟周期内保持高电平。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.7.21 CH3CCFG 寄存器（偏移 = B4h）[复位 = 00000000h]

图 19-139 展示了 CH3CCFG，表 19-157 对其进行了介绍。

返回到汇总表。

通道 3 捕获配置

图 19-139. CH3CCFG 寄存器

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

表 19-157. CH3CCFG 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-157. CH3CCFG 寄存器字段说明（续）

位	字段	类型	复位	说明
6-1	CAPT_SRC	R/W	0h	从异步 AUX 事件总线中选择捕获信号源。 所选信号进入边沿检测电路。在以下情况下，可能会发生错误的捕获事件： - 边沿检测电路中包含过期的信号采样，且在未执行 CH3EVCFG 所述的刷新操作的情况下启用了该电路 - 在 CTL.MODE 不为 DIS 的状态下重新配置了该寄存器。 您可以避免误捕获事件。当所需的通道功能为以下情况时： - SET_ON_CAPT_DIS：请参阅 CH3EVCFG.CCACT 中关于 SET_ON_CAPT_DIS 的说明。 - SET_ON_CAPT：请参阅 CH3EVCFG.CCACT 中关于 SET_ON_CAPT 的说明。 - PER_PULSE_WIDTH_MEAS：请参阅 CH3EVCFG.CCACT 中关于 PER_PULSE_WIDTH_MEAS 的说明。 如果写入非枚举值，其行为与 NO_EVENT 相同；读取时将返回写入的值。 0h = AUX_EVCTL:EVSTAT0.AUXIO0 1h = AUX_EVCTL:EVSTAT0.AUXIO1 2h = AUX_EVCTL:EVSTAT0.AUXIO2 3h = AUX_EVCTL:EVSTAT0.AUXIO3 4h = AUX_EVCTL:EVSTAT0.AUXIO4 5h = AUX_EVCTL:EVSTAT0.AUXIO5 6h = AUX_EVCTL:EVSTAT0.AUXIO6 7h = AUX_EVCTL:EVSTAT0.AUXIO7 8h = AUX_EVCTL:EVSTAT0.AUXIO8 9h = AUX_EVCTL:EVSTAT0.AUXIO9 Ah = AUX_EVCTL:EVSTAT0.AUXIO10 Bh = AUX_EVCTL:EVSTAT0.AUXIO11 Ch = AUX_EVCTL:EVSTAT0.AUXIO12 Dh = AUX_EVCTL:EVSTAT0.AUXIO13 Eh = AUX_EVCTL:EVSTAT0.AUXIO14 Fh = AUX_EVCTL:EVSTAT0.AUXIO15 10h = AUX_EVCTL:EVSTAT1.AUXIO16 11h = AUX_EVCTL:EVSTAT1.AUXIO17 12h = AUX_EVCTL:EVSTAT1.AUXIO18 13h = AUX_EVCTL:EVSTAT1.AUXIO19 14h = AUX_EVCTL:EVSTAT1.AUXIO20 15h = AUX_EVCTL:EVSTAT1.AUXIO21 16h = AUX_EVCTL:EVSTAT1.AUXIO22 17h = AUX_EVCTL:EVSTAT1.AUXIO23 18h = AUX_EVCTL:EVSTAT1.AUXIO24 19h = AUX_EVCTL:EVSTAT1.AUXIO25 1Ah = AUX_EVCTL:EVSTAT1.AUXIO26 1Bh = AUX_EVCTL:EVSTAT1.AUXIO27 1Ch = AUX_EVCTL:EVSTAT1.AUXIO28 1Dh = AUX_EVCTL:EVSTAT1.AUXIO29 1Eh = AUX_EVCTL:EVSTAT1.AUXIO30 1Fh = AUX_EVCTL:EVSTAT1.AUXIO31 20h = AUX_EVCTL:EVSTAT2.MANUAL_EV 21h = AUX_EVCTL:EVSTAT2.AON_RTC_CH2 22h = AUX_EVCTL:EVSTAT2.AON_RTC_CH2_DLY 23h = AUX_EVCTL:EVSTAT2.AON_RTC_4KHZ 24h = AUX_EVCTL:EVSTAT2.AON_BATMON_BAT_UPD 25h = AUX_EVCTL:EVSTAT2.AON_BATMON_TEMP_UPD 26h = AUX_EVCTL:EVSTAT2.SCLK_LF 27h = AUX_EVCTL:EVSTAT2.PWR_DWN 28h = AUX_EVCTL:EVSTAT2.MCU_ACTIVE 29h = AUX_EVCTL:EVSTAT2.VDDR_RECHARGE 2Ah = AUX_EVCTL:EVSTAT2.ACLK_REF 2Bh = AUX_EVCTL:EVSTAT2.MCU_EV 2Ch = AUX_EVCTL:EVSTAT2.MCU_OBSMUX0 2Dh = AUX_EVCTL:EVSTAT2.MCU_OBSMUX1 2Eh = AUX_EVCTL:EVSTAT2.AUX_COMPA 2Fh = AUX_EVCTL:EVSTAT2.AUX_COMPB 30h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV0

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-157. CH3CCFG 寄存器字段说明（续）

位	字段	类型	复位	说明
				31h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV1 32h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV2 33h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV3 35h = AUX_EVCTL:EVSTAT3.AUX_TIMER1_EV 36h = AUX_EVCTL:EVSTAT3.AUX_TIMER0_EV 37h = AUX_EVCTL:EVSTAT3.AUX_TDC_DONE 38h = AUX_EVCTL:EVSTAT3.AUX_ISRC_RESET_N 39h = AUX_EVCTL:EVSTAT3.AUX_ADC_DONE 3Ah = AUX_EVCTL:EVSTAT3.AUX_ADC_IRQ 3Bh = AUX_EVCTL:EVSTAT3.AUX_ADC_FIFO_ALMOST_FULL 3Ch = AUX_EVCTL:EVSTAT3.AUX_ADC_FIFO_NOT_EMPTY 3Dh = AUX_EVCTL:EVSTAT3.AUX_SMPH_AUTOTAKE_DONE 3Fh = 无事件。
0	EDGE	R/W	0h	边沿配置。 通道在由 CAPT_SRC 选定的信号源上，于选定边沿处捕获计数值。 参阅 CH3EVCFG.CCACT。 0h = 在 CAPT_SRC 的下降沿捕获 CNTR.VALUE。 1h = 在 CAPT_SRC 的上升沿捕获 CNTR.VALUE。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.7.22 CH3PCC 寄存器（偏移 = B8h）[复位 = 00000000h]

图 19-140 展示了 CH3PCC，表 19-158 对其进行了介绍。

返回到汇总表。

通道 3 流水线捕获对比情况

图 19-140. CH3PCC 寄存器

31	30	29	28	27	26	25	24	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																值															
R-0h																R/W-0h															

表 19-158. CH3PCC 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	值	R/W	0h	流水线捕获比较值。 16 位用户自定义的流水线比较值，或通道更新后的捕获值。 比较模式： 当下一个 CNTR.VALUE 为零且 CTL.MODE 不为 DIS 时，VALUE 的更新值将传输至 CH3CC.VALUE。此机制适用于 PWM 生成，并能防止生成信号的边沿出现抖动。 捕捉模式： 当 CH3EVCFG.CCACT 设置为 PER_PULSE_WIDTH_MEAS 时，VALUE 存储所选信号的低电平或高电平脉冲宽度。这由 CH3CCFG.EDGE 和 CH3CCFG.CAPT_SRC 指定。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.7.23 CH3CC 寄存器（偏移 = BCh）[复位 = 00000000h]

图 19-141 展示了 CH3CC，表 19-159 中对此进行了介绍。

返回到汇总表。

通道 3 捕获对标情况

图 19-141. CH3CC 寄存器

31	30	29	28	27	26	25	24	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																值															
R-0h																R/W-0h															

表 19-159. CH3CC 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	值	R/W	0h	捕获比较值。 16 位用户定义的比较值，或通道更新后的捕获值。 比较模式： 将 VALUE 与 CNTR.VALUE 进行比较；当两者相等时，根据 CH3EVCFG.CCACT 的设定生成事件。 捕捉模式： 当发生捕获事件时，当前的计数器值被存储到 VALUE 中。 CH3EVCFG.CCACT 确定值是信号周期还是常规捕获值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.8 AUX_ANAIF 寄存器

表 19-160 列出了 AUX_ANAIF 寄存器的存储器映射寄存器。表 19-160 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

表 19-160. AUX_ANAIF 寄存器

偏移	首字母缩写词	寄存器名称	部分
10h	ADCCTL	ADC 控制寄存器	ADCCTL 寄存器（偏移 = 10h）[复位 = 00003F00h]
14h	ADCFIFOSTAT	ADC FIFO 状态	ADCFIFOSTAT 寄存器（偏移 = 14h）[复位 = 00000001h]
18h	ADCFIFO	ADC FIFO	ADCFIFO 寄存器（偏移 = 18h）[复位 = 00000000h]
1Ch	ADCTRIG	ADC 触发器	ADCTRIG 寄存器（偏移 =1Ch）[复位 = 00000000h]
20h	ISRCCTL	电流源控制	ISRCCTL 寄存器（偏移 = 20h）[复位 = 00000001h]
30h	DACCTL	DAC 控制	DACCTL 寄存器（偏移 = 30h）[复位 = 00000000h]
34h	LPMBIASCTL	低功耗模式偏置控制	LPMBIASCTL 寄存器（偏移 = 34h）[复位 = 00000000h]
38h	DACSMPLCTL	DAC 采样控制	DACSMPLCTL 寄存器（偏移 = 38h）[复位 = 00000000h]
3Ch	DACSMPLCFG0	DAC 采样配置 0	DACSMPLCFG0 寄存器（偏移 = 3Ch）[复位 = 00000000h]
40h	DACSMPLCFG1	DAC 采样配置 1	DACSMPLCFG1 寄存器（偏移 = 40h）[复位 = 00000000h]
44h	DACVALUE	DAC 值	DACVALUE 寄存器（偏移 = 44h）[复位 = 00000000h]
48h	DACSTAT	DAC 状态	DACSTAT 寄存器（偏移 = 48h）[复位 = 00000000h]

复杂的位访问类型经过编码可适应小型表单元。表 19-161 展示了适用于此部分中访问类型的代码。

表 19-161. AUX_ANAIF 访问类型代码

访问类型	代码	说明
读取类型		
R	R	读取
写入类型		
W	W	写入
复位或默认值		
-n		复位后的值或默认值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.8.1 ADCCTL 寄存器（偏移 = 10h）[复位 = 00003F00h]

图 19-142 展示了 ADCCTL，表 19-162 中对此进行了介绍。

返回到汇总表。

ADI_4_AUX:ADC0.SMPL_MODE 的 ADC 控制
配置决定 ADC 触发器是开始采样还是转换。

图 19-142. ADCCTL 寄存器

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

表 19-162. ADCCTL 寄存器字段说明

位	字段	类型	复位	说明
31-15	RESERVED	R	0h	保留
14	START_POL	R/W	0h	为 START_SRC 事件选择有效极性。 0h = 设置在事件源的上升沿触发 ADC。 1h = 设置在事件源的下降沿触发 ADC。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-162. ADCCTL 寄存器字段说明（续）

位	字段	类型	复位	说明
13-8	START_SRC	R/W	3Fh	从异步 AUX 事件总线中选择 ADC 触发事件源。 如果您要通过 ADCTRIG.START 手动触发 ADC，请将 START_SRC 设置为 NO_EVENT。 如果写入非枚举值，则行为与 NO_EVENT 相同。写入的值在读取时返回。 0h = AUX_EVCTL:EVSTAT0.AUXIO0 1h = AUX_EVCTL:EVSTAT0.AUXIO1 2h = AUX_EVCTL:EVSTAT0.AUXIO2 3h = AUX_EVCTL:EVSTAT0.AUXIO3 4h = AUX_EVCTL:EVSTAT0.AUXIO4 5h = AUX_EVCTL:EVSTAT0.AUXIO5 6h = AUX_EVCTL:EVSTAT0.AUXIO6 7h = AUX_EVCTL:EVSTAT0.AUXIO7 8h = AUX_EVCTL:EVSTAT0.AUXIO8 9h = AUX_EVCTL:EVSTAT0.AUXIO9 Ah = AUX_EVCTL:EVSTAT0.AUXIO10 Bh = AUX_EVCTL:EVSTAT0.AUXIO11 Ch = AUX_EVCTL:EVSTAT0.AUXIO12 Dh = AUX_EVCTL:EVSTAT0.AUXIO13 Eh = AUX_EVCTL:EVSTAT0.AUXIO14 Fh = AUX_EVCTL:EVSTAT0.AUXIO15 10h = AUX_EVCTL:EVSTAT1.AUXIO16 11h = AUX_EVCTL:EVSTAT1.AUXIO17 12h = AUX_EVCTL:EVSTAT1.AUXIO18 13h = AUX_EVCTL:EVSTAT1.AUXIO19 14h = AUX_EVCTL:EVSTAT1.AUXIO20 15h = AUX_EVCTL:EVSTAT1.AUXIO21 16h = AUX_EVCTL:EVSTAT1.AUXIO22 17h = AUX_EVCTL:EVSTAT1.AUXIO23 18h = AUX_EVCTL:EVSTAT1.AUXIO24 19h = AUX_EVCTL:EVSTAT1.AUXIO25 1Ah = AUX_EVCTL:EVSTAT1.AUXIO26 1Bh = AUX_EVCTL:EVSTAT1.AUXIO27 1Ch = AUX_EVCTL:EVSTAT1.AUXIO28 1Dh = AUX_EVCTL:EVSTAT1.AUXIO29 1Eh = AUX_EVCTL:EVSTAT1.AUXIO30 1Fh = AUX_EVCTL:EVSTAT1.AUXIO31 20h = AUX_EVCTL:EVSTAT2.MANUAL_EV 21h = AUX_EVCTL:EVSTAT2.AON_RTC_CH2 22h = AUX_EVCTL:EVSTAT2.AON_RTC_CH2_DLY 23h = AUX_EVCTL:EVSTAT2.AON_RTC_4KHZ 24h = AUX_EVCTL:EVSTAT2.AON_BATMON_BAT_UPD 25h = AUX_EVCTL:EVSTAT2.AON_BATMON_TEMP_UPD 26h = AUX_EVCTL:EVSTAT2.SCLK_LF 27h = AUX_EVCTL:EVSTAT2.PWR_DWN 28h = AUX_EVCTL:EVSTAT2.MCU_ACTIVE 29h = AUX_EVCTL:EVSTAT2.VDDR_RECHARGE 2Ah = AUX_EVCTL:EVSTAT2.ACLK_REF 2Bh = AUX_EVCTL:EVSTAT2.MCU_EV 2Eh = AUX_EVCTL:EVSTAT2.AUX_COMPA 2Fh = AUX_EVCTL:EVSTAT2.AUX_COMPB 30h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV0 31h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV1 32h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV2 33h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV3 34h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_PULSE 35h = AUX_EVCTL:EVSTAT3.AUX_TIMER1_EV 36h = AUX_EVCTL:EVSTAT3.AUX_TIMER0_EV 37h = AUX_EVCTL:EVSTAT3.AUX_TDC_DONE 38h = AUX_EVCTL:EVSTAT3.AUX_ISRC_RESET_N 3Dh = AUX_EVCTL:EVSTAT3.AUX_SMPH_AUTOTAKE_DONE 3Fh = 无事件。
7-2	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-162. ADCCTL 寄存器字段说明（续）

位	字段	类型	复位	说明
1-0	CMD	R/W	0h	ADC 接口命令。 不支持非枚举值。写入的值在读取时返回。 0h = 禁用 ADC 接口。 1h = 启用 ADC 接口。 3h = 刷新 ADC FIFO。 刷新后，必须将 CMD 设置为 EN 或 DIS。 系统 CPU 必须等待两个时钟周期，然后才能将 CMD 设置为 EN 或 DIS。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.8.2 ADCFIFOSTAT 寄存器（偏移 = 14h）[复位 = 00000001h]

图 19-143 展示了 ADCFIFOSTAT，表 19-163 中对此进行了介绍。

返回到汇总表。

ADC FIFO 状态
FIFO 最多可保存四个 ADC 样本。

图 19-143. ADCFIFOSTAT 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED			OVERFLOW	UNDERFLOW	FULL	ALMOST_FULL	EMPTY
R-0h			R-0h	R-0h	R-0h	R-0h	R-1h

表 19-163. ADCFIFOSTAT 寄存器字段说明

位	字段	类型	复位	说明
31-5	RESERVED	R	0h	保留
4	OVERFLOW	R	0h	FIFO 溢出标志 0：FIFO 尚未溢出。 1：FIFO 已溢出。该标志是粘滞位，直到您刷新 FIFO 才会清除。当设置为该标志时，ADC FIFO 的写入指针为静态。无法将更多样本添加到 ADC FIFO 中。刷新 FIFO 以清除标志。
3	UNDERFLOW	R	0h	FIFO 下溢标志 0：FIFO 未下溢。 1：FIFO 已下溢。此标志是粘滞位，直到您刷新 FIFO 才会清除。当设置该标志时，ADC FIFO 的读取指针为静态。读取将返回所读取样本的前一个样本。刷新 FIFO 以清除标志。
2	FULL	R	0h	FIFO 已满标志。 0：FIFO 未满，FIFO 中的样本不足 4 个。 1：FIFO 已满，FIFO 中有 4 个样本。 设置该标志后，无法向 ADC FIFO 添加更多样本。尝试添加样本会设置“溢出”标志。
1	ALMOST_FULL	R	0h	FIFO 几乎已满标志。 0：FIFO 中的样本少于 3 个，或者 FIFO 已满。在后一种情况下，“已满”标志也会被置为有效。 1：FIFO 中有 3 个样本，还可以再存储一个样本。
0	EMPTY	R	1h	FIFO 空标志。 0：FIFO 包含一个或多个样本。 1：FIFO 为空。 设置该标志后，读取将返回所读取样本的前一个样本并设置“下溢”标志。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.8.3 ADCFIFO 寄存器（偏移 = 18h）[复位 = 00000000h]

图 19-144 展示了 ADCFIFO，表 19-164 中对此进行了介绍。

返回到汇总表。

ADC FIFO

图 19-144. ADCFIFO 寄存器

31	30	29	28	27	26	25	24	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											

表 19-164. ADCFIFO 寄存器字段说明

位	字段	类型	复位	说明
31-12	RESERVED	R	0h	保留
11-0	DATA	R/W	0h	FIFO 数据。 读取： 从 FIFO 获取最早的 ADC 样本。 写入： 将虚拟样本写入 FIFO。当您没有真实 ADC 样本时，这对于代码开发非常有用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.8.4 ADCTRIG 寄存器（偏移 =1Ch）[复位 = 00000000h]

图 19-145 中所示为 ADCTRIG，相关产明请参阅 表 19-165。

返回到汇总表。

ADC 触发器

图 19-145. ADCTRIG 寄存器

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

表 19-165. ADCTRIG 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	启动	W	0h	手动 ADC 触发器。 0：无效。 1：单个 ADC 触发器。 要手动触发 ADC，必须将 ADCCTL.START_SRC 设置为 NO_EVENT，以避免与事件驱动的 ADC 触发器发生冲突。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.8.5 ISRCCTL 寄存器（偏移 = 20h）[复位 = 00000001h]

图 19-146 展示了 ISRCCTL，表 19-166 中对此进行了介绍。

返回到汇总表。

电流源控制

图 19-146. ISRCCTL 寄存器

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

表 19-166. ISRCCTL 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	RESET_N	R/W	1h	ISRC 复位控制。 0：ISRC 驱动 0uA。 1：ISRC 将电流 ADI_4_AUX:ISRC.TRIM 驱动至 COMPA_IN。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.8.6 DACCTL 寄存器（偏移 = 30h）[复位 = 00000000h]

图 19-147 展示了 DACCTL，表 19-167 中对此进行了介绍。

返回到汇总表。

DAC 控制
该寄存器控制 DAC 的模拟部分。

图 19-147. DACCTL 寄存器

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

表 19-167. DACCTL 寄存器字段说明

位	字段	类型	复位	说明
31-6	RESERVED	R	0h	保留
5	DAC_EN	R/W	0h	DAC 模块使能。 0：禁用 DAC。 1：启用 DAC。 当 AUX_SYSIF:OPMODEREQ.REQ 等于 PDA 时，传感器控制器不得使用 DAC。 在待机 TI-RTOS 电源模式下，当 AUX_SYSIF:OPMODEREQ.REQ 等于 PDA 时，系统 CPU 不得使用 DAC。系统 CPU 必须将 AUX_SYSIF:PEROPRATE.ANAIF_DAC_OP_RATE 设置为 BUS_RATE，才能在工作和空闲 TI-RTOS 电源模式下使用 DAC。
4	DAC_BUFFER_EN	R/W	0h	DAC 缓冲器使能。 DAC 缓冲器能够缩短生成编程电压所需的时间，但会增加电流消耗。 0：禁用 DAC 缓冲器。 1：启用 DAC 缓冲器。 当 DAC_VOUT_SEL 等于 COMPA_IN 时，启用缓冲器。 当 AUX_SYSIF:OPMODEREQ.REQ 等于 PDA 或 PDLP 时，不要启用缓冲器。
3	DAC_PRECHARGE_EN	R/W	0h	DAC 预充电使能。 仅当 ADI_4_AUX:MUX2.DAC_VREF_SEL 等于 DCOUPL 且 VDDS 高于 2.65V 时，启用预充电。 DAC 输出电压范围： 0：0V 至 1.28V。 1：1.28V 至 2.56V。 否则，请参阅 ADI_4_AUX:MUX2.DAC_VREF_SEL，了解 DAC 输出电压范围。 在启用 DAC 和缓冲器之前 1 µs 启用预充电。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-167. DACCTL 寄存器字段说明（续）

位	字段	类型	复位	说明
2-0	DAC_VOUT_SEL	R/W	0h	DAC 输出连接。 一个模拟节点只能有一个驱动源。以下模拟节点的其他驱动源在 [ANATOP_MMAP::ADI_4_AUX:*] 中配置。 0h = 不连接任何节点 当更改 DAC_VOUT_SEL 时，建议使用 NC 作为中间步骤。 1h = 连接到 COMPB_REF 模拟节点。 使用比较器 B 所需的设置。 2h = 连接到 COMPA_REF 模拟节点。 使用此设置无法驱动连接到 COMPA_REF I/O 多路复用器的外部负载。 4h = 连接到 COMPA_IN 模拟节点。 驱动 ADI_4_AUX:MUX1.COMPA_IN 中所选外部负载所需的设置。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.8.7 LPMBIASCTL 寄存器（偏移 = 34h）[复位 = 00000000h]

图 19-148 展示了 LPMBIASCTL，表 19-168 中对此进行了介绍。

返回到汇总表。

低功耗模式偏置控制
当 AUX_SYSIF:OPMODEREQ.REQ 与 A 不同时，低功耗模式偏置模块为 DAC 和比较器 A 提供偏置电流。

图 19-148. LPMBIASCTL 寄存器

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															EN
R-0h															R/W-0h

表 19-168. LPMBIASCTL 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	EN	R/W	0h	模块使能。 0：禁用低功耗模式偏置模块。 1：启用低功耗模式偏置模块。 在启用 DAC 或比较器 A 之前 15 µs，将 EN 设置为 1。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.8.8 DACSMPLCTL 寄存器（偏移 = 38h）[复位 = 00000000h]

图 19-149 展示了 DACSMPLCTL，表 19-169 中对此进行了介绍。

返回到汇总表。

DAC 采样控制
DAC 采样时钟保持存储在采样保持电容器中的 DAC 电压。DAC 采样时钟波形由建立阶段和紧随其后的保持阶段组成。在建立阶段，采样保持电容器会充电至编程电压。保持阶段则以最低功耗维持该电压。
DACSMPLCFG0 和 DACSMPLCFG1 配置 DAC 采样时钟波形。

图 19-149. DACSMPLCTL 寄存器

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															EN
R-0h															R/W-0h

表 19-169. DACSMPLCTL 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	EN	R/W	0h	DAC 采样时钟使能。 0：禁用采样时钟。当前采样时钟周期完成时，采样时钟停止并保持低电平，DACSTAT 变为 0。 1：启用 DAC 采样时钟。启用采样时钟之前，DACSTAT 必须为 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.8.9 DACSMPLCFG0 寄存器（偏移 = 3Ch）[复位 = 00000000h]

图 19-150 展示了 DACSMPLCFG0，表 19-170 对其进行了介绍。

返回到汇总表。

DAC 采样配置 0

图 19-150. DACSMPLCFG0 寄存器

31	30	29	28	27	26	25	24	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																										CLKDIV					
R-0h																										R/W-0h					

表 19-170. DACSMPLCFG0 寄存器字段说明

位	字段	类型	复位	说明
31-6	RESERVED	R	0h	保留
5-0	CLKDIV	R/W	0h	时钟分频。 AUX_SYSIF:PEROPRATE.ANAIF_DAC_OP_RATE 除以 (CLKDIV + 1) 决定采样时钟基频。 0：除以 1。 1：除以 2。 ... 63：除以 64。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.8.10 DACSMPLCFG1 寄存器（偏移 = 40h）[复位 = 00000000h]

图 19-151 展示了 DACSMPLCFG1，表 19-171 对其进行了介绍。

返回到汇总表。

DAC 采样配置 1
采样时钟周期等于（高电平时间 + 低电平时间）* 基准周期。DACSMPLCFG0.CLKDIV 决定基准周期。
时序要求（DAC 缓冲器开启/DAC 缓冲器关闭）：
- （高电平时间 + 低电平时间）* 基准周期 > (4 µ s/1 µ s)
- （高电平时间 * 基准周期）> (2 µ s/0.5 µ s)
- （低电平时间 * 基准周期）> (2 µ s/0.5 µ s)
- （低电平时间 * 基准周期 + HOLD_INTERVAL * 采样时钟周期）< 32 µ s
如果 AUX_SYSIF:OPMODEREQ.REQ 等于 PDLP，则必须设置：
- H_PER = L_PER = HOLD_INTERVAL = 0。

图 19-151. DACSMPLCFG1 寄存器

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

表 19-171. DACSMPLCFG1 寄存器字段说明

位	字段	类型	复位	说明
31-15	RESERVED	R	0h	保留
14	H_PER	R/W	0h	高电平时间。 采样时钟周期在该数量的基准周期内保持高电平。 0：2 个周期 1：4 个周期
13-12	L_PER	R/W	0h	低电平时间。 采样时钟周期在该数量的基准周期内保持低电平。 0：1 个周期 1：2 个周期 2：3 个周期 3：4 个周期
11-8	SETUP_CNT	R/W	0h	建立计数。 建立阶段的有效采样时钟周期数。 0：1 个采样时钟周期 1：2 个采样时钟周期 ... 15：16 个采样时钟周期
7-0	HOLD_INTERVAL	R/W	0h	保持间隔。 保持阶段每个有效采样时钟周期之间的非活动采样时钟周期数。当处于非活动状态时，采样时钟为低电平。 范围为 0 至 255。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.8.11 DACVALUE 寄存器（偏移 = 44h）[复位 = 00000000h]

图 19-152 展示了 DACVALUE，表 19-172 中对此进行了介绍。

返回到汇总表。

DAC 值

图 19-152. DACVALUE 寄存器

31	30	29	28	27	26	25	24	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																								值							
R-0h																								R/W-0h							

表 19-172. DACVALUE 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-0	值	R/W	0h	DAC 值。 DAC 的数字数据字。 仅在 DACCTL.DAC_EN 为 0 时更改值。然后，等待 1 µs 后再启用 DAC。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.8.12 DACSTAT 寄存器（偏移 = 48h）[复位 = 00000000h]

图 19-153 展示了 DACSTAT，表 19-173 中对此进行了介绍。

返回到汇总表。

DAC 状态

图 19-153. DACSTAT 寄存器

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

表 19-173. DACSTAT 寄存器字段说明

位	字段	类型	复位	说明
31-2	RESERVED	R	0h	保留
1	SETUP_ACTIVE	R	0h	DAC 建立阶段状态。 0：采样时钟被禁用或设置阶段已经完成。 1：建立阶段正在进行。
0	HOLD_ACTIVE	R	0h	DAC 保持阶段状态。 0：采样时钟被禁用或 DAC 未处于保持阶段持。 1：保持阶段正在进行。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.9 AUX_SYSIF 寄存器

表 19-174 列出了 AUX_SYSIF 寄存器的存储器映射寄存器。表 19-174 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

表 19-174. AUX_SYSIF 寄存器

偏移	首字母缩写词	寄存器名称	部分
0h	OPMODEREQ	工作模式请求	OPMODEREQ 寄存器（偏移 = 0h）[复位 = 00000000h]
4h	OPMODEACK	工作模式确认	OPMODEACK 寄存器（偏移 = 4h）[复位 = 00000000h]
8h	PROGWU0CFG	可编程唤醒 0 配置	PROGWU0CFG 寄存器（偏移 = 8h）[复位 = 00000000h]
Ch	PROGWU1CFG	可编程唤醒 1 配置	PROGWU1CFG 寄存器（偏移 = Ch）[复位 = 00000000h]
10h	PROGWU2CFG	可编程唤醒 2 配置	PROGWU2CFG 寄存器（偏移 = 10h）[复位 = 00000000h]
14h	PROGWU3CFG	可编程唤醒 3 配置	PROGWU3CFG 寄存器（偏移 = 14h）[复位 = 00000000h]
18h	SWWUTRIG	软件唤醒触发器	SWWUTRIG 寄存器（偏移 = 18h）[复位 = 00000000h]
1Ch	WUFLAGS	唤醒标志	WUFLAGS 寄存器（偏移 = 1Ch）[复位 = 00000000h]
20h	WUFLAGSCLR	唤醒标志清除	WUFLAGSCLR 寄存器（偏移 = 20h）[复位 = 0000000Fh]
24h	WUGATE	唤醒门控	WUGATE 寄存器（偏移 = 24h）[复位 = 00000000h]
28h	VECCFG0	矢量配置 0	VECCFG0 寄存器（偏移 = 28h）[复位 = 00000000h]
2Ch	VECCFG1	矢量配置 1	VECCFG1 寄存器（偏移 = 2Ch）[复位 = 00000000h]
30h	VECCFG2	矢量配置 2	VECCFG2 寄存器（偏移 = 30h）[复位 = 00000000h]
34h	VECCFG3	矢量配置 3	VECCFG3 寄存器（偏移 = 34h）[复位 = 00000000h]
38h	VECCFG4	矢量配置 4	VECCFG4 寄存器（偏移 = 38h）[复位 = 00000000h]
3Ch	VECCFG5	矢量配置 5	VECCFG5 寄存器（偏移 = 3Ch）[复位 = 00000000h]
40h	VECCFG6	矢量配置 6	VECCFG6 寄存器（偏移 = 40h）[复位 = 00000000h]
44h	VECCFG7	矢量配置 7	VECCFG7 寄存器（偏移 = 44h）[复位 = 00000000h]
48h	EVSYNCRATE	事件同步速率	EVSYNCRATE 寄存器（偏移 = 48h）[复位 = 00000000h]
4Ch	PEROPRATE	外设运行速率	PEROPRATE 寄存器（偏移 = 4Ch）[复位 = 00000000h]
50h	ADCCLKCTL	ADC 时钟控制	ADCCLKCTL 寄存器（偏移 = 50h）[复位 = 00000000h]
54h	TDCCLKCTL	TDC 计数器时钟控制	TDCCLKCTL 寄存器（偏移 = 54h）[复位 = 00000000h]
58h	TDCREFCLKCTL	TDC 基准时钟控制	TDCREFCLKCTL 寄存器（偏移 = 58h）[复位 = 00000000h]

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-174. AUX_SYSIF 寄存器（续）

偏移	首字母缩写词	寄存器名称	部分
5Ch	TIMER2CLKCTL	AUX_TIMER2 时钟控制	TIMER2CLKCTL 寄存器（偏移 = 5Ch）[复位 = 00000000h]
60h	TIMER2CLKSTAT	AUX_TIMER2 时钟状态	TIMER2CLKSTAT 寄存器（偏移 = 60h）[复位 = 00000000h]
64h	TIMER2CLKSWITCH	AUX_TIMER2 时钟切换	TIMER2CLKSWITCH 寄存器（偏移 = 64h）[复位 = 00000001h]
68h	TIMER2DBGCTL	AUX_TIMER2 调试控制	TIMER2DBGCTL 寄存器（偏移 = 68h）[复位 = 00000000h]
70h	CLKSHIFTDET	时钟移位检测	CLKSHIFTDET 寄存器（偏移 = 70h）[复位 = 00000001h]
74h	RECHARGETRIG	VDDR 充电触发器	RECHARGETRIG 寄存器（偏移 = 74h）[复位 = 00000000h]
78h	RECHARGEDET	VDDR 充电检测	RECHARGEDET 寄存器（偏移 = 78h）[复位 = 00000000h]
7Ch	RTCSUBSECINC0	实时计数器亚秒增量 0	RTCSUBSECINC0 寄存器（偏移 = 7Ch）[复位 = 00000000h]
80h	RTCSUBSECINC1	实时计数器亚秒增量 1	RTCSUBSECINC1 寄存器（偏移 = 80h）[复位 = 00000000h]
84h	RTCSUBSECINCCTL	实时计数器亚秒增量控制	RTCSUBSECINCCTL 寄存器（偏移 = 84h）[复位 = 00000000h]
88h	RTCSEC	实时计数器秒	RTCSEC 寄存器（偏移 = 88h）[复位 = 00000000h]
8Ch	RTCSUBSEC	实时计数器亚秒	RTCSUBSEC 寄存器（偏移 = 8Ch）[复位 = 00000000h]
90h	RTCEVCLR	AON_RTC 事件清除	RTCEVCLR 寄存器（偏移 = 90h）[复位 = 00000000h]
94h	BATMONBAT	AON_BATMON 电池电压值	BATMONBAT 寄存器（偏移 = 94h）[复位 = 00000000h]
9Ch	BATMONTEMP	AON_BATMON 温度值	BATMONTEMP 寄存器（偏移 = 9Ch）[复位 = 00000000h]
A0h	TIMERHALT	计时器暂停	TIMERHALT 寄存器（偏移 = A0h）[复位 = 00000000h]
B0h	TIMER2BRIDGE	AUX_TIMER2 桥接器	TIMER2BRIDGE 寄存器（偏移 = B0h）[复位 = 00000000h]
B4h	SWPWRPROF	软件功耗分析器	SWPWRPROF 寄存器（偏移 = B4h）[复位 = 00000000h]

复杂的位访问类型经过编码可适应小型表单元。表 19-175 展示了适用于此部分中访问类型的代码。

表 19-175. AUX_SYSIF 访问类型代码

访问类型	代码	说明
读取类型		
R	R	读取
RH	R H	读取 由硬件置位或清零
写入类型		
W	W	写入
复位或默认值		
-n		复位后的值或默认值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.9.1 OPMODEREQ 寄存器（偏移 = 0h）[复位 = 00000000h]

图 19-154 展示了 OPMODEREQ，表 19-176 中对此进行了介绍。

返回到汇总表。

工作模式请求
AUX 可以在三种工作模式下运行。每种模式都与以下因素相关联：
- SCE 时钟源或速率，由 AON_PMCTL:AUXSCECLK 给出。此速率称为 SCE_RATE。
- 系统电源状态请求。AUX 可以请求掉电 (uLDO) 或活动（GLDO 或 DCDC）系统电源状态。
- 对活动 AUX 唤醒标志的特定系统响应。响应取决于请求的工作模式。
uLDO 电源状态提供的电流能力有限。在此电源状态下，AUX_SCE 无法使用某些外设和功能，例如 AUX_DDI0_OSC、AUX_TDC 和 AUX_ANAIF ADC 接口。
请遵循以下规则：
- 在通过 OPMODEACK 确认请求之前，不允许更改请求。
- 模式请求的更改必须按照以下顺序逐步进行（方向无关）：
PDA - A - LP - PDLP。
不遵守这些规则可能会导致意外行为，必须加以避免。

图 19-154. OPMODEREQ 寄存器

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														REQ	
R-0h														R/W-0h	

表 19-176. OPMODEREQ 寄存器字段说明

位	字段	类型	复位	说明
31-2	RESERVED	R	0h	保留
1-0	REQ	R/W	0h	AUX 工作模式请求。 0h = 活动工作模式，特征如下： - 活动系统电源状态（GLDO 或 DCDC）请求。 - AON_PMCTL:AUXSCECLK.SRC 设置 SCE 时钟频率 (SCE_RATE)。 - 活动唤醒标志不会改变工作模式。 1h = 低功耗工作模式，特征如下： - 掉电系统电源状态 (uLDO) 请求。 - SCE 时钟频率 (SCE_RATE) 等于 SCLK_MF。 - 活动唤醒标志不会改变工作模式。 2h = 唤醒至活动模式的掉电工作模式，特征如下： - 掉电系统电源状态 (uLDO) 请求。 - AON_PMCTL:AUXSCECLK.PD_SRC 设置 SCE 时钟频率 (SCE_RATE)。 - 只要设置了活动唤醒标志，该标志就会在外部将工作模式覆盖为活动模式 (A)。 3h = 唤醒至低功耗模式的掉电工作模式，特征如下： - 掉电系统电源状态 (uLDO) 请求。 - AON_PMCTL:AUXSCECLK.PD_SRC 设置 SCE 时钟频率 (SCE_RATE)。 - 只要设置了活动唤醒标志，该标志就会在外部将工作模式覆盖为低功耗模式 (LP)。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.9.2 OPMODEACK 寄存器（偏移 = 4h）[复位 = 00000000h]

图 19-155 展示了 OPMODEACK，表 19-177 中对此进行了介绍。

返回到汇总表。

工作模式确认
AUX_SCE 程序必须假定当前工作模式是已确认的模式。

图 19-155. OPMODEACK 寄存器

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														ACK	
R-0h														R-0h	

表 19-177. OPMODEACK 寄存器字段说明

位	字段	类型	复位	说明
31-2	RESERVED	R	0h	保留
1-0	ACK	R	0h	AUX 工作模式确认。 0h = 确认活动工作模式。 1h = 确认低功耗工作模式。 2h = 确认唤醒至活动模式的掉电工作模式。 3h = 确认唤醒至低功耗模式的掉电工作模式。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.9.3 PROGWU0CFG 寄存器（偏移 = 8h）[复位 = 00000000h]

图 19-156 展示了 PROGWU0CFG，表 19-178 对其进行了介绍。

返回到汇总表。

可编程唤醒 0 配置
配置该寄存器以使能自定义 AUX 唤醒标志。唤醒标志将由 AON_PMCTL 捕获，并根据当前工作模式做出响应。您可以选择 WUFLAGS.PROG_WU0 以通过 VECCFGn 配置来触发可编程 AUX_SCE 矢量的执行。您需要按照 WUFLAGSCLR 中所述的步骤清除此标志。您需要按照 WUGATE 中所述的步骤完成相关配置。

图 19-156. PROGWU0CFG 寄存器

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								POL	EN	WU_SRC					
R-0h								R/W-0h	R/W-0h	R/W-0h					

表 19-178. PROGWU0CFG 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7	POL	R/W	0h	WU_SRC 的极性。 用于清除唤醒标志的过程决定了电平或边沿灵敏度，请参阅 WUFLAGSCLR.PROG_WU0。 0h = 当 WU_SRC 为高电平或变为高电平时，设置唤醒标志。 1h = 当 WU_SRC 为低电平或变为低电平时，设置唤醒标志。
6	EN	R/W	0h	可编程唤醒标志使能。 0：禁用唤醒标志。 1：使能唤醒标志。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-178. PROGWU0CFG 寄存器字段说明（续）

位	字段	类型	复位	说明
5-0	WU_SRC	R/W	0h	来自异步 AUX 事件总线的唤醒源。 仅当 EN 为 0 或 WUFLAGSCLR.PROG_WU0 为 1 时才更改 WU_SRC。 如果写入非枚举值，则行为与 NO_EVENT 相同。写入的值在读取时返回。 0h = AUX_EVCTL:EVSTAT0.AUXIO0 1h = AUX_EVCTL:EVSTAT0.AUXIO1 2h = AUX_EVCTL:EVSTAT0.AUXIO2 3h = AUX_EVCTL:EVSTAT0.AUXIO3 4h = AUX_EVCTL:EVSTAT0.AUXIO4 5h = AUX_EVCTL:EVSTAT0.AUXIO5 6h = AUX_EVCTL:EVSTAT0.AUXIO6 7h = AUX_EVCTL:EVSTAT0.AUXIO7 8h = AUX_EVCTL:EVSTAT0.AUXIO8 9h = AUX_EVCTL:EVSTAT0.AUXIO9 Ah = AUX_EVCTL:EVSTAT0.AUXIO10 Bh = AUX_EVCTL:EVSTAT0.AUXIO11 Ch = AUX_EVCTL:EVSTAT0.AUXIO12 Dh = AUX_EVCTL:EVSTAT0.AUXIO13 Eh = AUX_EVCTL:EVSTAT0.AUXIO14 Fh = AUX_EVCTL:EVSTAT0.AUXIO15 10h = AUX_EVCTL:EVSTAT1.AUXIO16 11h = AUX_EVCTL:EVSTAT1.AUXIO17 12h = AUX_EVCTL:EVSTAT1.AUXIO18 13h = AUX_EVCTL:EVSTAT1.AUXIO19 14h = AUX_EVCTL:EVSTAT1.AUXIO20 15h = AUX_EVCTL:EVSTAT1.AUXIO21 16h = AUX_EVCTL:EVSTAT1.AUXIO22 17h = AUX_EVCTL:EVSTAT1.AUXIO23 18h = AUX_EVCTL:EVSTAT1.AUXIO24 19h = AUX_EVCTL:EVSTAT1.AUXIO25 1Ah = AUX_EVCTL:EVSTAT1.AUXIO26 1Bh = AUX_EVCTL:EVSTAT1.AUXIO27 1Ch = AUX_EVCTL:EVSTAT1.AUXIO28 1Dh = AUX_EVCTL:EVSTAT1.AUXIO29 1Eh = AUX_EVCTL:EVSTAT1.AUXIO30 1Fh = AUX_EVCTL:EVSTAT1.AUXIO31 20h = AUX_EVCTL:EVSTAT2.MANUAL_EV 21h = AUX_EVCTL:EVSTAT2.AON_RTC_CH2 22h = AUX_EVCTL:EVSTAT2.AON_RTC_CH2_DLY 23h = AUX_EVCTL:EVSTAT2.AON_RTC_4KHZ 24h = AUX_EVCTL:EVSTAT2.AON_BATMON_BAT_UPD 25h = AUX_EVCTL:EVSTAT2.AON_BATMON_TEMP_UPD 26h = AUX_EVCTL:EVSTAT2.SCLK_LF 27h = AUX_EVCTL:EVSTAT2.PWR_DWN 28h = AUX_EVCTL:EVSTAT2.MCU_ACTIVE 29h = AUX_EVCTL:EVSTAT2.VDDR_RECHARGE 2Ah = AUX_EVCTL:EVSTAT2.ACLK_REF 2Bh = AUX_EVCTL:EVSTAT2.MCU_EV 2Ch = AUX_EVCTL:EVSTAT2.MCU_OBSMUX0 2Dh = AUX_EVCTL:EVSTAT2.MCU_OBSMUX1 2Eh = AUX_EVCTL:EVSTAT2.AUX_COMPA 2Fh = AUX_EVCTL:EVSTAT2.AUX_COMPB 30h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV0 31h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV1 32h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV2 33h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV3 34h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_PULSE 35h = AUX_EVCTL:EVSTAT3.AUX_TIMER1_EV 36h = AUX_EVCTL:EVSTAT3.AUX_TIMER0_EV 37h = AUX_EVCTL:EVSTAT3.AUX_TDC_DONE 38h = AUX_EVCTL:EVSTAT3.AUX_ISRC_RESET_N 39h = AUX_EVCTL:EVSTAT3.AUX_ADC_DONE 3Ah = AUX_EVCTL:EVSTAT3.AUX_ADC_IRQ 3Bh = AUX_EVCTL:EVSTAT3.AUX_ADC_FIFO_ALMOST_FULL

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-178. PROGWU0CFG 寄存器字段说明（续）

位	字段	类型	复位	说明
				3Ch = AUX_EVCTL:EVSTAT3.AUX_ADC_FIFO_NOT_EMPTY 3Dh = AUX_EVCTL:EVSTAT3.AUX_SMPH_AUTOTAKE_DONE 3Fh = 无事件。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.9.4 PROGWU1CFG 寄存器（偏移 = Ch）[复位 = 00000000h]

图 19-157 展示了 PROGWU1CFG，表 19-179 对其进行了介绍。

返回到汇总表。

可编程唤醒 1 配置
配置该寄存器以使能自定义 AUX 唤醒标志。唤醒标志将由 AON_PMCTL 捕获，并根据当前工作模式做出响应。您可以选择 WUFLAGS.PROG_WU1 以通过 VECCFGn 配置来触发可编程 AUX_SCE 矢量的执行。您需要按照 WUFLAGSCLR 中所述的步骤清除此标志。您需要按照 WUGATE 中所述的步骤完成相关配置。

图 19-157. PROGWU1CFG 寄存器

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								POL	EN	WU_SRC					
R-0h								R/W-0h	R/W-0h	R/W-0h					

表 19-179. PROGWU1CFG 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7	POL	R/W	0h	WU_SRC 的极性。 用于清除唤醒标志的过程决定了电平或边沿灵敏度，请参阅 WUFLAGSCLR.PROG_WU1。 0h = 当 WU_SRC 为高电平或变为高电平时，设置唤醒标志。 1h = 当 WU_SRC 为低电平或变为低电平时，设置唤醒标志。
6	EN	R/W	0h	可编程唤醒标志使能。 0：禁用唤醒标志。 1：使能唤醒标志。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-179. PROGWU1CFG 寄存器字段说明（续）

位	字段	类型	复位	说明
5-0	WU_SRC	R/W	0h	来自异步 AUX 事件总线的唤醒源。 仅当 EN 为 0 或 WUFLAGSCLR.PROG_WU1 为 1 时才更改 WU_SRC。 如果写入非枚举值，则行为与 NO_EVENT 相同。写入的值在读取时返回。 0h = AUX_EVCTL:EVSTAT0.AUXIO0 1h = AUX_EVCTL:EVSTAT0.AUXIO1 2h = AUX_EVCTL:EVSTAT0.AUXIO2 3h = AUX_EVCTL:EVSTAT0.AUXIO3 4h = AUX_EVCTL:EVSTAT0.AUXIO4 5h = AUX_EVCTL:EVSTAT0.AUXIO5 6h = AUX_EVCTL:EVSTAT0.AUXIO6 7h = AUX_EVCTL:EVSTAT0.AUXIO7 8h = AUX_EVCTL:EVSTAT0.AUXIO8 9h = AUX_EVCTL:EVSTAT0.AUXIO9 Ah = AUX_EVCTL:EVSTAT0.AUXIO10 Bh = AUX_EVCTL:EVSTAT0.AUXIO11 Ch = AUX_EVCTL:EVSTAT0.AUXIO12 Dh = AUX_EVCTL:EVSTAT0.AUXIO13 Eh = AUX_EVCTL:EVSTAT0.AUXIO14 Fh = AUX_EVCTL:EVSTAT0.AUXIO15 10h = AUX_EVCTL:EVSTAT1.AUXIO16 11h = AUX_EVCTL:EVSTAT1.AUXIO17 12h = AUX_EVCTL:EVSTAT1.AUXIO18 13h = AUX_EVCTL:EVSTAT1.AUXIO19 14h = AUX_EVCTL:EVSTAT1.AUXIO20 15h = AUX_EVCTL:EVSTAT1.AUXIO21 16h = AUX_EVCTL:EVSTAT1.AUXIO22 17h = AUX_EVCTL:EVSTAT1.AUXIO23 18h = AUX_EVCTL:EVSTAT1.AUXIO24 19h = AUX_EVCTL:EVSTAT1.AUXIO25 1Ah = AUX_EVCTL:EVSTAT1.AUXIO26 1Bh = AUX_EVCTL:EVSTAT1.AUXIO27 1Ch = AUX_EVCTL:EVSTAT1.AUXIO28 1Dh = AUX_EVCTL:EVSTAT1.AUXIO29 1Eh = AUX_EVCTL:EVSTAT1.AUXIO30 1Fh = AUX_EVCTL:EVSTAT1.AUXIO31 20h = AUX_EVCTL:EVSTAT2.MANUAL_EV 21h = AUX_EVCTL:EVSTAT2.AON_RTC_CH2 22h = AUX_EVCTL:EVSTAT2.AON_RTC_CH2_DLY 23h = AUX_EVCTL:EVSTAT2.AON_RTC_4KHZ 24h = AUX_EVCTL:EVSTAT2.AON_BATMON_BAT_UPD 25h = AUX_EVCTL:EVSTAT2.AON_BATMON_TEMP_UPD 26h = AUX_EVCTL:EVSTAT2.SCLK_LF 27h = AUX_EVCTL:EVSTAT2.PWR_DWN 28h = AUX_EVCTL:EVSTAT2.MCU_ACTIVE 29h = AUX_EVCTL:EVSTAT2.VDDR_RECHARGE 2Ah = AUX_EVCTL:EVSTAT2.ACLK_REF 2Bh = AUX_EVCTL:EVSTAT2.MCU_EV 2Ch = AUX_EVCTL:EVSTAT2.MCU_OBSMUX0 2Dh = AUX_EVCTL:EVSTAT2.MCU_OBSMUX1 2Eh = AUX_EVCTL:EVSTAT2.AUX_COMPA 2Fh = AUX_EVCTL:EVSTAT2.AUX_COMPB 30h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV0 31h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV1 32h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV2 33h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV3 34h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_PULSE 35h = AUX_EVCTL:EVSTAT3.AUX_TIMER1_EV 36h = AUX_EVCTL:EVSTAT3.AUX_TIMER0_EV 37h = AUX_EVCTL:EVSTAT3.AUX_TDC_DONE 38h = AUX_EVCTL:EVSTAT3.AUX_ISRC_RESET_N 39h = AUX_EVCTL:EVSTAT3.AUX_ADC_DONE 3Ah = AUX_EVCTL:EVSTAT3.AUX_ADC_IRQ 3Bh = AUX_EVCTL:EVSTAT3.AUX_ADC_FIFO_ALMOST_FULL

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-179. PROGWU1CFG 寄存器字段说明（续）

位	字段	类型	复位	说明
				3Ch = AUX_EVCTL:EVSTAT3.AUX_ADC_FIFO_NOT_EMPTY 3Dh = AUX_EVCTL:EVSTAT3.AUX_SMPH_AUTOTAKE_DONE 3Fh = 无事件。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.9.5 PROGWU2CFG 寄存器（偏移 = 10h）[复位 = 00000000h]

图 19-158 展示了 PROGWU2CFG，表 19-180 对其进行了介绍。

返回到汇总表。

可编程唤醒 2 配置
配置该寄存器以使能自定义 AUX 唤醒标志。唤醒标志将由 AON_PMCTL 捕获，并根据当前工作模式做出响应。您可以选择 WUFLAGS.PROG_WU2 以通过 VECCFGn 配置来触发可编程 AUX_SCE 矢量的执行。您需要按照 WUFLAGSCLR 中所述的步骤清除此标志。您需要按照 WUGATE 中所述的步骤完成相关配置。

图 19-158. PROGWU2CFG 寄存器

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								POL	EN	WU_SRC					
R-0h								R/W-0h	R/W-0h	R/W-0h					

表 19-180. PROGWU2CFG 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7	POL	R/W	0h	WU_SRC 的极性。 用于清除唤醒标志的过程决定了电平或边沿灵敏度，请参阅 WUFLAGSCLR.PROG_WU2。 0h = 当 WU_SRC 为高电平或变为高电平时，设置唤醒标志。 1h = 当 WU_SRC 为低电平或变为低电平时，设置唤醒标志。
6	EN	R/W	0h	可编程唤醒标志使能。 0：禁用唤醒标志。 1：使能唤醒标志。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-180. PROGWU2CFG 寄存器字段说明（续）

位	字段	类型	复位	说明
5-0	WU_SRC	R/W	0h	来自异步 AUX 事件总线的唤醒源。 仅当 EN 为 0 或 WUFLAGSCLR.PROG_WU2 为 1 时才更改 WU_SRC。 如果写入非枚举值，则行为与 NO_EVENT 相同。写入的值在读取时返回。 0h = AUX_EVCTL:EVSTAT0.AUXIO0 1h = AUX_EVCTL:EVSTAT0.AUXIO1 2h = AUX_EVCTL:EVSTAT0.AUXIO2 3h = AUX_EVCTL:EVSTAT0.AUXIO3 4h = AUX_EVCTL:EVSTAT0.AUXIO4 5h = AUX_EVCTL:EVSTAT0.AUXIO5 6h = AUX_EVCTL:EVSTAT0.AUXIO6 7h = AUX_EVCTL:EVSTAT0.AUXIO7 8h = AUX_EVCTL:EVSTAT0.AUXIO8 9h = AUX_EVCTL:EVSTAT0.AUXIO9 Ah = AUX_EVCTL:EVSTAT0.AUXIO10 Bh = AUX_EVCTL:EVSTAT0.AUXIO11 Ch = AUX_EVCTL:EVSTAT0.AUXIO12 Dh = AUX_EVCTL:EVSTAT0.AUXIO13 Eh = AUX_EVCTL:EVSTAT0.AUXIO14 Fh = AUX_EVCTL:EVSTAT0.AUXIO15 10h = AUX_EVCTL:EVSTAT1.AUXIO16 11h = AUX_EVCTL:EVSTAT1.AUXIO17 12h = AUX_EVCTL:EVSTAT1.AUXIO18 13h = AUX_EVCTL:EVSTAT1.AUXIO19 14h = AUX_EVCTL:EVSTAT1.AUXIO20 15h = AUX_EVCTL:EVSTAT1.AUXIO21 16h = AUX_EVCTL:EVSTAT1.AUXIO22 17h = AUX_EVCTL:EVSTAT1.AUXIO23 18h = AUX_EVCTL:EVSTAT1.AUXIO24 19h = AUX_EVCTL:EVSTAT1.AUXIO25 1Ah = AUX_EVCTL:EVSTAT1.AUXIO26 1Bh = AUX_EVCTL:EVSTAT1.AUXIO27 1Ch = AUX_EVCTL:EVSTAT1.AUXIO28 1Dh = AUX_EVCTL:EVSTAT1.AUXIO29 1Eh = AUX_EVCTL:EVSTAT1.AUXIO30 1Fh = AUX_EVCTL:EVSTAT1.AUXIO31 20h = AUX_EVCTL:EVSTAT2.MANUAL_EV 21h = AUX_EVCTL:EVSTAT2.AON_RTC_CH2 22h = AUX_EVCTL:EVSTAT2.AON_RTC_CH2_DLY 23h = AUX_EVCTL:EVSTAT2.AON_RTC_4KHZ 24h = AUX_EVCTL:EVSTAT2.AON_BATMON_BAT_UPD 25h = AUX_EVCTL:EVSTAT2.AON_BATMON_TEMP_UPD 26h = AUX_EVCTL:EVSTAT2.SCLK_LF 27h = AUX_EVCTL:EVSTAT2.PWR_DWN 28h = AUX_EVCTL:EVSTAT2.MCU_ACTIVE 29h = AUX_EVCTL:EVSTAT2.VDDR_RECHARGE 2Ah = AUX_EVCTL:EVSTAT2.ACLK_REF 2Bh = AUX_EVCTL:EVSTAT2.MCU_EV 2Ch = AUX_EVCTL:EVSTAT2.MCU_OBSMUX0 2Dh = AUX_EVCTL:EVSTAT2.MCU_OBSMUX1 2Eh = AUX_EVCTL:EVSTAT2.AUX_COMPA 2Fh = AUX_EVCTL:EVSTAT2.AUX_COMPB 30h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV0 31h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV1 32h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV2 33h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV3 34h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_PULSE 35h = AUX_EVCTL:EVSTAT3.AUX_TIMER1_EV 36h = AUX_EVCTL:EVSTAT3.AUX_TIMER0_EV 37h = AUX_EVCTL:EVSTAT3.AUX_TDC_DONE 38h = AUX_EVCTL:EVSTAT3.AUX_ISRC_RESET_N 39h = AUX_EVCTL:EVSTAT3.AUX_ADC_DONE 3Ah = AUX_EVCTL:EVSTAT3.AUX_ADC_IRQ 3Bh = AUX_EVCTL:EVSTAT3.AUX_ADC_FIFO_ALMOST_FULL

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-180. PROGWU2CFG 寄存器字段说明（续）

位	字段	类型	复位	说明
				3Ch = AUX_EVCTL:EVSTAT3.AUX_ADC_FIFO_NOT_EMPTY 3Dh = AUX_EVCTL:EVSTAT3.AUX_SMPH_AUTOTAKE_DONE 3Fh = 无事件。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.9.6 PROGWU3CFG 寄存器（偏移 = 14h）[复位 = 00000000h]

图 19-159 展示了 PROGWU3CFG，表 19-181 对其进行了介绍。

返回到汇总表。

可编程唤醒 3 配置
配置该寄存器以使能自定义 AUX 唤醒标志。唤醒标志将由 AON_PMCTL 捕获，并根据当前工作模式做出响应。您可以选择 WUFLAGS.PROG_WU3 以通过 VECCFGn 配置来触发可编程 AUX_SCE 矢量的执行。您需要按照 WUFLAGSCLR 中所述的步骤清除此标志。您需要按照 WUGATE 中所述的步骤完成相关配置。

图 19-159. PROGWU3CFG 寄存器

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								POL	EN	WU_SRC					
R-0h								R/W-0h	R/W-0h	R/W-0h					

表 19-181. PROGWU3CFG 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7	POL	R/W	0h	WU_SRC 的极性。 用于清除唤醒标志的过程决定了电平或边沿灵敏度，请参阅 WUFLAGSCLR.PROG_WU3。 0h = 当 WU_SRC 为高电平或变为高电平时，设置唤醒标志。 1h = 当 WU_SRC 为低电平或变为低电平时，设置唤醒标志。
6	EN	R/W	0h	可编程唤醒标志使能。 0：禁用唤醒标志。 1：使能唤醒标志。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-181. PROGWU3CFG 寄存器字段说明（续）

位	字段	类型	复位	说明
5-0	WU_SRC	R/W	0h	来自异步 AUX 事件总线的唤醒源。 仅当 EN 为 0 或 WUFLAGSCLR.PROG_WU3 为 1 时才更改 WU_SRC。 如果写入非枚举值，则行为与 NO_EVENT 相同。写入的值在读取时返回。 0h = AUX_EVCTL:EVSTAT0.AUXIO0 1h = AUX_EVCTL:EVSTAT0.AUXIO1 2h = AUX_EVCTL:EVSTAT0.AUXIO2 3h = AUX_EVCTL:EVSTAT0.AUXIO3 4h = AUX_EVCTL:EVSTAT0.AUXIO4 5h = AUX_EVCTL:EVSTAT0.AUXIO5 6h = AUX_EVCTL:EVSTAT0.AUXIO6 7h = AUX_EVCTL:EVSTAT0.AUXIO7 8h = AUX_EVCTL:EVSTAT0.AUXIO8 9h = AUX_EVCTL:EVSTAT0.AUXIO9 Ah = AUX_EVCTL:EVSTAT0.AUXIO10 Bh = AUX_EVCTL:EVSTAT0.AUXIO11 Ch = AUX_EVCTL:EVSTAT0.AUXIO12 Dh = AUX_EVCTL:EVSTAT0.AUXIO13 Eh = AUX_EVCTL:EVSTAT0.AUXIO14 Fh = AUX_EVCTL:EVSTAT0.AUXIO15 10h = AUX_EVCTL:EVSTAT1.AUXIO16 11h = AUX_EVCTL:EVSTAT1.AUXIO17 12h = AUX_EVCTL:EVSTAT1.AUXIO18 13h = AUX_EVCTL:EVSTAT1.AUXIO19 14h = AUX_EVCTL:EVSTAT1.AUXIO20 15h = AUX_EVCTL:EVSTAT1.AUXIO21 16h = AUX_EVCTL:EVSTAT1.AUXIO22 17h = AUX_EVCTL:EVSTAT1.AUXIO23 18h = AUX_EVCTL:EVSTAT1.AUXIO24 19h = AUX_EVCTL:EVSTAT1.AUXIO25 1Ah = AUX_EVCTL:EVSTAT1.AUXIO26 1Bh = AUX_EVCTL:EVSTAT1.AUXIO27 1Ch = AUX_EVCTL:EVSTAT1.AUXIO28 1Dh = AUX_EVCTL:EVSTAT1.AUXIO29 1Eh = AUX_EVCTL:EVSTAT1.AUXIO30 1Fh = AUX_EVCTL:EVSTAT1.AUXIO31 20h = AUX_EVCTL:EVSTAT2.MANUAL_EV 21h = AUX_EVCTL:EVSTAT2.AON_RTC_CH2 22h = AUX_EVCTL:EVSTAT2.AON_RTC_CH2_DLY 23h = AUX_EVCTL:EVSTAT2.AON_RTC_4KHZ 24h = AUX_EVCTL:EVSTAT2.AON_BATMON_BAT_UPD 25h = AUX_EVCTL:EVSTAT2.AON_BATMON_TEMP_UPD 26h = AUX_EVCTL:EVSTAT2.SCLK_LF 27h = AUX_EVCTL:EVSTAT2.PWR_DWN 28h = AUX_EVCTL:EVSTAT2.MCU_ACTIVE 29h = AUX_EVCTL:EVSTAT2.VDDR_RECHARGE 2Ah = AUX_EVCTL:EVSTAT2.ACLK_REF 2Bh = AUX_EVCTL:EVSTAT2.MCU_EV 2Ch = AUX_EVCTL:EVSTAT2.MCU_OBSMUX0 2Dh = AUX_EVCTL:EVSTAT2.MCU_OBSMUX1 2Eh = AUX_EVCTL:EVSTAT2.AUX_COMPA 2Fh = AUX_EVCTL:EVSTAT2.AUX_COMPB 30h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV0 31h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV1 32h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV2 33h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV3 34h = AUX_EVCTL:EVSTAT3.AUX_TIMER2_PULSE 35h = AUX_EVCTL:EVSTAT3.AUX_TIMER1_EV 36h = AUX_EVCTL:EVSTAT3.AUX_TIMER0_EV 37h = AUX_EVCTL:EVSTAT3.AUX_TDC_DONE 38h = AUX_EVCTL:EVSTAT3.AUX_ISRC_RESET_N 39h = AUX_EVCTL:EVSTAT3.AUX_ADC_DONE 3Ah = AUX_EVCTL:EVSTAT3.AUX_ADC_IRQ 3Bh = AUX_EVCTL:EVSTAT3.AUX_ADC_FIFO_ALMOST_FULL

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-181. PROGWU3CFG 寄存器字段说明（续）

位	字段	类型	复位	说明
				3Ch = AUX_EVCTL:EVSTAT3.AUX_ADC_FIFO_NOT_EMPTY 3Dh = AUX_EVCTL:EVSTAT3.AUX_SMPH_AUTOTAKE_DONE 3Fh = 无事件。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.9.7 SWWUTRIG 寄存器（偏移 = 18h）[复位 = 00000000h]

图 19-160 展示了 SWWUTRIG，表 19-182 中对此进行了介绍。

返回到汇总表。

软件唤醒触发器
系统 CPU 使用这些唤醒标志来执行与 AUX_SCE 的握手。唤醒标志可以更改 AUX 的工作模式并保证非零 SCE 时钟速率。AUX_SCE 唤醒矢量在 VECCFGn 中配置。

图 19-160. SWWUTRIG 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				SW_WU3	SW_WU2	SW_WU1	SW_WU0
R-0h				W-0h	W-0h	W-0h	W-0h

表 19-182. SWWUTRIG 寄存器字段说明

位	字段	类型	复位	说明
31-4	RESERVED	R	0h	保留
3	SW_WU3	W	0h	软件唤醒 3 触发器。 0：没有影响。 1：设置 WUFLAGS.SW_WU3 并触发 AUX 唤醒。
2	SW_WU2	W	0h	软件唤醒 2 触发器。 0：没有影响。 1：设置 WUFLAGS.SW_WU2 并触发 AUX 唤醒。
1	SW_WU1	W	0h	软件唤醒 1 触发器。 0：没有影响。 1：设置 WUFLAGS.SW_WU1 并触发 AUX 唤醒。
0	SW_WU0	W	0h	软件唤醒 0 触发器。 0：没有影响。 1：设置 WUFLAGS.SW_WU0 并触发 AUX 唤醒。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.9.8 WUFLAGS 寄存器（偏移 = 1Ch）[复位 = 00000000h]

图 19-161 展示了 WUFLAGS，表 19-183 中对此进行了介绍。

返回到汇总表。

唤醒标志
该寄存器保存八个 AUX 唤醒标志。每个标志都会按照 OPMODEREQ 中的配置导致 AUX 工作模式发生变化。要清除标志 n，必须在 WUFLAGSCLR 中设置位 n，直到标志 n 读取为 0。必须先在 WUFLAGSCLR 中清除位 n，然后才能再次设置标志 n。

图 19-161. WUFLAGS 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
SW_WU3	SW_WU2	SW_WU1	SW_WU0	PROG_WU3	PROG_WU2	PROG_WU1	PROG_WU0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

表 19-183. WUFLAGS 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7	SW_WU3	R	0h	软件唤醒 3 标志。 0：软件唤醒 3 未触发。 1：软件唤醒 3 已触发。
6	SW_WU2	R	0h	软件唤醒 2 标志。 0：软件唤醒 2 未触发。 1：软件唤醒 2 已触发。
5	SW_WU1	R	0h	软件唤醒 1 标志。 0：软件唤醒 1 未触发。 1：软件唤醒 1 已触发。
4	SW_WU0	R	0h	软件唤醒 0 标志。 0：软件唤醒 0 未触发。 1：软件唤醒 0 已触发。
3	PROG_WU3	R	0h	可编程唤醒 3。 0：可编程唤醒 3 未触发。 1：可编程唤醒 3 已触发。
2	PROG_WU2	R	0h	可编程唤醒 2。 0：可编程唤醒 2 未触发。 1：可编程唤醒 2 已触发。
1	PROG_WU1	R	0h	可编程唤醒 1。 0：可编程唤醒 1 未触发。 1：可编程唤醒 1 已触发。
0	PROG_WU0	R	0h	可编程唤醒 0。 0：可编程唤醒 0 未触发。 1：可编程唤醒 0 已触发。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.9.9 WUFLAGSCLR 寄存器（偏移 = 20h）[复位 = 0000000Fh]

图 19-162 展示了 WUFLAGSCLR，表 19-184 中对此进行了介绍。

返回到汇总表。

唤醒标志清除
该寄存器清除 AUX 唤醒标志 WUFLAGS。
要清除可编程唤醒标志，必须先禁用 AUX 唤醒输出。清除可编程唤醒标志后，必须重新使能 AUX 唤醒输出。写入 WUGATE 以禁用或使能 AUX 唤醒输出。想要清除软件触发的唤醒时，不需要执行此步骤。

图 19-162. WUFLAGSCLR 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
SW_WU3	SW_WU2	SW_WU1	SW_WU0	PROG_WU3	PROG_WU2	PROG_WU1	PROG_WU0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-1h	R/W-1h	R/W-1h	R/W-1h

表 19-184. WUFLAGSCLR 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7	SW_WU3	R/W	0h	清除软件唤醒标志 3。 0：没有影响。 1：清除 WUFLAGS.SW_WU3。保持高电平，直到 WUFLAGS.SW_WU3 为 0。
6	SW_WU2	R/W	0h	清除软件唤醒标志 2。 0：没有影响。 1：清除 WUFLAGS.SW_WU2。保持高电平，直到 WUFLAGS.SW_WU2 为 0。
5	SW_WU1	R/W	0h	清除软件唤醒标志 1。 0：没有影响。 1：清除 WUFLAGS.SW_WU1。保持高电平，直到 WUFLAGS.SW_WU1 为 0。
4	SW_WU0	R/W	0h	清除软件唤醒标志 0。 0：没有影响。 1：清除 WUFLAGS.SW_WU0。保持高电平，直到 WUFLAGS.SW_WU0 为 0。
3	PROG_WU3	R/W	1h	可编程唤醒标志 3。 0：没有影响。 1：清除 WUFLAGS.PROG_WU3。保持高电平，直到 WUFLAGS.PROG_WU3 为 0。 如果在 PROGWU3CFG.EN 为 1 时将 PROG_WU3 写入 0，则唤醒标志会变为边沿敏感。 如果在 PROGWU3CFG.EN 为 0 时将 PROG_WU3 写入 0，则唤醒标志会变为电平敏感，然后设置 PROGWU3CFG.EN。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-184. WUFLAGSCLR 寄存器字段说明（续）

位	字段	类型	复位	说明
2	PROG_WU2	R/W	1h	可编程唤醒标志 2。 0：没有影响。 1：清除 WUFLAGS.PROG_WU2。保持高电平，直到 WUFLAGS.PROG_WU2 为 0。 如果在 PROGWU2CFG.EN 为 1 时将 PROG_WU2 写入 0，则唤醒标志会变为边沿敏感。 如果在 PROGWU2CFG.EN 为 0 时将 PROG_WU2 写入 0，则唤醒标志会变为电平敏感，然后设置 PROGWU2CFG.EN。
1	PROG_WU1	R/W	1h	可编程唤醒标志 1。 0：没有影响。 1：清除 WUFLAGS.PROG_WU1。保持高电平，直到 WUFLAGS.PROG_WU1 为 0。 如果在 PROGWU1CFG.EN 为 1 时将 PROG_WU1 写入 0，则唤醒标志会变为边沿敏感。 如果在 PROGWU1CFG.EN 为 0 时将 PROG_WU1 写入 0，则唤醒标志会变为电平敏感，然后设置 PROGWU1CFG.EN。
0	PROG_WU0	R/W	1h	可编程唤醒标志 0。 0：没有影响。 1：清除 WUFLAGS.PROG_WU0。保持高电平，直到 WUFLAGS.PROG_WU0 为 0。 如果在 PROGWU0CFG.EN 为 1 时将 PROG_WU0 写入 0，则唤醒标志会变为边沿敏感。 如果在 PROGWU0CFG.EN 为 0 时将 PROG_WU0 写入 0，则唤醒标志会变为电平敏感，然后设置 PROGWU0CFG.EN。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.9.10 WUGATE 寄存器（偏移 = 24h）[复位 = 00000000h]

图 19-163 展示了 WUGATE，表 19-185 中对此进行了介绍。

返回到汇总表。

唤醒门控
在以下操作之前，必须禁用 AUX 唤醒输出：
- 清除可编程唤醒标志之前。
- 更改 [PROGWUnCFG.EN] 或 [PROGWUnCFG.WU_SRC] 的值之前。
清除操作或可编程唤醒配置后，必须重新使能 AUX 唤醒输出。

图 19-163. WUGATE 寄存器

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															EN
R-0h															R/W-0h

表 19-185. WUGATE 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	EN	R/W	0h	唤醒输出使能。 0：禁用 AUX 唤醒输出。 1：使能 AUX 唤醒输出。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.9.11 VECCFG0 寄存器（偏移 = 28h）[复位 = 00000000h]

图 19-164 展示了 VECCFG0，表 19-186 对其进行了介绍。

返回到汇总表。

矢量配置 0
AUX_SCE 唤醒矢量 0 配置

图 19-164. VECCFG0 寄存器

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												VEC_EV			
R-0h												R/W-0h			

表 19-186. VECCFG0 寄存器字段说明

位	字段	类型	复位	说明
31-4	RESERVED	R	0h	保留
3-0	VEC_EV	R/W	0h	选择向量 0 的触发事件。 非枚举值被视为 NONE。 0h = 向量被禁用。 1h = WUFLAGS.PROG_WU0 2h = WUFLAGS.PROG_WU1 3h = WUFLAGS.PROG_WU2 4h = WUFLAGS.PROG_WU3 5h = WUFLAGS.SW_WU0 6h = WUFLAGS.SW_WU1 7h = WUFLAGS.SW_WU2 8h = WUFLAGS.SW_WU3 9h = AUX_EVCTL:EVSTAT2.AON_RTC_CH2_DLY

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.9.12 VECCFG1 寄存器（偏移 = 2Ch）[复位 = 00000000h]

图 19-165 展示了 VECCFG1，表 19-187 对其进行了介绍。

返回到汇总表。

矢量配置 1
AUX_SCE 唤醒矢量 1 配置

图 19-165. VECCFG1 寄存器

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												VEC_EV			
R-0h												R/W-0h			

表 19-187. VECCFG1 寄存器字段说明

位	字段	类型	复位	说明
31-4	RESERVED	R	0h	保留
3-0	VEC_EV	R/W	0h	选择向量 1 的触发事件。 非枚举值被视为 NONE。 0h = 向量被禁用。 1h = WUFLAGS.PROG_WU0 2h = WUFLAGS.PROG_WU1 3h = WUFLAGS.PROG_WU2 4h = WUFLAGS.PROG_WU3 5h = WUFLAGS.SW_WU0 6h = WUFLAGS.SW_WU1 7h = WUFLAGS.SW_WU2 8h = WUFLAGS.SW_WU3 9h = AUX_EVCTL:EVSTAT2.AON_RTC_CH2_DLY

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.9.13 VECCFG2 寄存器（偏移 = 30h）[复位 = 00000000h]

图 19-166 展示了 VECCFG2，表 19-188 对其进行了介绍。

返回到汇总表。

矢量配置 2
AUX_SCE 唤醒矢量 2 配置

图 19-166. VECCFG2 寄存器

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												VEC_EV			
R-0h												R/W-0h			

表 19-188. VECCFG2 寄存器字段说明

位	字段	类型	复位	说明
31-4	RESERVED	R	0h	保留
3-0	VEC_EV	R/W	0h	选择向量 2 的触发事件。 非枚举值被视为 NONE。 0h = 向量被禁用。 1h = WUFLAGS.PROG_WU0 2h = WUFLAGS.PROG_WU1 3h = WUFLAGS.PROG_WU2 4h = WUFLAGS.PROG_WU3 5h = WUFLAGS.SW_WU0 6h = WUFLAGS.SW_WU1 7h = WUFLAGS.SW_WU2 8h = WUFLAGS.SW_WU3 9h = AUX_EVCTL:EVSTAT2.AON_RTC_CH2_DLY

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.9.14 VECCFG3 寄存器（偏移 = 34h）[复位 = 00000000h]

图 19-167 展示了 VECCFG3，表 19-189 对其进行了介绍。

返回到汇总表。

矢量配置 3
AUX_SCE 唤醒矢量 3 配置

图 19-167. VECCFG3 寄存器

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												VEC_EV			
R-0h												R/W-0h			

表 19-189. VECCFG3 寄存器字段说明

位	字段	类型	复位	说明
31-4	RESERVED	R	0h	保留
3-0	VEC_EV	R/W	0h	选择向量 3 的触发事件。 非枚举值被视为 NONE。 0h = 向量被禁用。 1h = WUFLAGS.PROG_WU0 2h = WUFLAGS.PROG_WU1 3h = WUFLAGS.PROG_WU2 4h = WUFLAGS.PROG_WU3 5h = WUFLAGS.SW_WU0 6h = WUFLAGS.SW_WU1 7h = WUFLAGS.SW_WU2 8h = WUFLAGS.SW_WU3 9h = AUX_EVCTL:EVSTAT2.AON_RTC_CH2_DLY

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.9.15 VECCFG4 寄存器（偏移 = 38h）[复位 = 00000000h]

图 19-168 展示了 VECCFG4，表 19-190 对其进行了介绍。

返回到汇总表。

矢量配置 4
AUX_SCE 唤醒矢量 4 配置

图 19-168. VECCFG4 寄存器

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												VEC_EV			
R-0h												R/W-0h			

表 19-190. VECCFG4 寄存器字段说明

位	字段	类型	复位	说明
31-4	RESERVED	R	0h	保留
3-0	VEC_EV	R/W	0h	选择向量 4 的触发事件。 非枚举值被视为 NONE。 0h = 向量被禁用。 1h = WUFLAGS.PROG_WU0 2h = WUFLAGS.PROG_WU1 3h = WUFLAGS.PROG_WU2 4h = WUFLAGS.PROG_WU3 5h = WUFLAGS.SW_WU0 6h = WUFLAGS.SW_WU1 7h = WUFLAGS.SW_WU2 8h = WUFLAGS.SW_WU3 9h = AUX_EVCTL:EVSTAT2.AON_RTC_CH2_DLY

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.9.16 VECCFG5 寄存器（偏移 = 3Ch）[复位 = 00000000h]

图 19-169 展示了 VECCFG5，表 19-191 对其进行了介绍。

返回到汇总表。

矢量配置 5
AUX_SCE 唤醒矢量 5 配置

图 19-169. VECCFG5 寄存器

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												VEC_EV			
R-0h												R/W-0h			

表 19-191. VECCFG5 寄存器字段说明

位	字段	类型	复位	说明
31-4	RESERVED	R	0h	保留
3-0	VEC_EV	R/W	0h	选择向量 5 的触发事件。 非枚举值被视为 NONE。 0h = 向量被禁用。 1h = WUFLAGS.PROG_WU0 2h = WUFLAGS.PROG_WU1 3h = WUFLAGS.PROG_WU2 4h = WUFLAGS.PROG_WU3 5h = WUFLAGS.SW_WU0 6h = WUFLAGS.SW_WU1 7h = WUFLAGS.SW_WU2 8h = WUFLAGS.SW_WU3 9h = AUX_EVCTL:EVSTAT2.AON_RTC_CH2_DLY

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.9.17 VECCFG6 寄存器（偏移 = 40h）[复位 = 00000000h]

图 19-170 展示了 VECCFG6，表 19-192 对其进行了介绍。

返回到汇总表。

矢量配置 6
AUX_SCE 唤醒矢量 6 配置

图 19-170. VECCFG6 寄存器

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												VEC_EV			
R-0h												R/W-0h			

表 19-192. VECCFG6 寄存器字段说明

位	字段	类型	复位	说明
31-4	RESERVED	R	0h	保留
3-0	VEC_EV	R/W	0h	选择向量 6 的触发事件。 非枚举值被视为 NONE。 0h = 向量被禁用。 1h = WUFLAGS.PROG_WU0 2h = WUFLAGS.PROG_WU1 3h = WUFLAGS.PROG_WU2 4h = WUFLAGS.PROG_WU3 5h = WUFLAGS.SW_WU0 6h = WUFLAGS.SW_WU1 7h = WUFLAGS.SW_WU2 8h = WUFLAGS.SW_WU3 9h = AUX_EVCTL:EVSTAT2.AON_RTC_CH2_DLY

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.9.18 VECCFG7 寄存器（偏移 = 44h）[复位 = 00000000h]

图 19-171 展示了 VECCFG7，表 19-193 对其进行了介绍。

返回到汇总表。

矢量配置 7
AUX_SCE 唤醒矢量 7 配置

图 19-171. VECCFG7 寄存器

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												VEC_EV			
R-0h												R/W-0h			

表 19-193. VECCFG7 寄存器字段说明

位	字段	类型	复位	说明
31-4	RESERVED	R	0h	保留
3-0	VEC_EV	R/W	0h	选择向量 7 的触发事件。 非枚举值被视为 NONE。 0h = 向量被禁用。 1h = WUFLAGS.PROG_WU0 2h = WUFLAGS.PROG_WU1 3h = WUFLAGS.PROG_WU2 4h = WUFLAGS.PROG_WU3 5h = WUFLAGS.SW_WU0 6h = WUFLAGS.SW_WU1 7h = WUFLAGS.SW_WU2 8h = WUFLAGS.SW_WU3 9h = AUX_EVCTL:EVSTAT2.AON_RTC_CH2_DLY

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.9.19 EVSYNCRATE 寄存器（偏移 = 48h）[复位 = 00000000h]

图 19-172 展示了 EVSYNCRATE，表 19-194 中对此进行了介绍。

返回到汇总表。

事件同步速率
为同步 AUX 事件总线上的某些事件配置同步速率。
当 AUX_SCE 使用该事件时，必须选择 SCE 速率。当系统 CPU 使用该事件时，必须选择 AUX 总线速率。SCE 速率等于在 AON_PMCTL:AUXSCECLK 中配置的速率。AUX 总线速率等于 SCE 速率，当 MCU 域处于活动状态时，等于 SCLK_HF 除以 2。

图 19-172. EVSYNCRATE 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED					AUX_COMPA_SYNC_RATE	AUX_COMPB_SYNC_RATE	AUX_TIMER2_SYNC_RATE
R-0h					R/W-0h	R/W-0h	R/W-0h

表 19-194. EVSYNCRATE 寄存器字段说明

位	字段	类型	复位	说明
31-3	RESERVED	R	0h	保留
2	AUX_COMPA_SYNC_RATE	R/W	0h	为 AUX_EVCTL:EVSTAT2.AUX_COMPA 事件选择同步速率。 0h = SCE 速率 1h = AUX 总线速率
1	AUX_COMPB_SYNC_RATE	R/W	0h	为 AUX_EVCTL:EVSTAT2.AUX_COMPB 事件选择同步速率。 0h = SCE 速率 1h = AUX 总线速率
0	AUX_TIMER2_SYNC_RATE	R/W	0h	为以下事件选择同步速率： - AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV0 - AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV1 - AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV2 - AUX_EVCTL:EVSTAT3.AUX_TIMER2_EV3 - AUX_EVCTL:EVSTAT3.AUX_TIMER2_PULSE 0h = SCE 速率 1h = AUX 总线速率

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.9.20 PEROPRATE 寄存器（偏移 = 4Ch）[复位 = 00000000h]

图 19-173 展示了 PEROPRATE，表 19-195 中对此进行了介绍。

返回到汇总表。

外设运行速率
一些 AUX 外设以 SCE 或 AUX 总线速率运行。
当 AUX_SCE 使用此类外设或由其生成的事件时，必须选择 SCE 速率。当系统 CPU 使用此类外设时，必须选择 AUX 总线速率。
SCE 速率等于在 AON_PMCTL:AUXSCECLK 中配置的速率。AUX 总线速率等于 SCE 速率，当 MCU 域处于活动状态时，等于 SCLK_HF 除以 2。

图 19-173. PEROPRATE 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				ANAIF_DAC_OP_RATE	TIMER01_OP_RATE	SPIM_OP_RATE	MAC_OP_RATE
R-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 19-195. PEROPRATE 寄存器字段说明

位	字段	类型	复位	说明
31-4	RESERVED	R	0h	保留
3	ANAIF_DAC_OP_RATE	R/W	0h	为 AUX_ANAIF DAC 采样时钟状态机选择运行速率。 0h = SCE 速率 1h = AUX 总线速率
2	TIMER01_OP_RATE	R/W	0h	为 AUX_TIMER01 选择运行速率。 0h = SCE 速率 1h = AUX 总线速率
1	SPIM_OP_RATE	R/W	0h	为 AUX_SPIM 选择运行速率。 0h = SCE 速率 1h = AUX 总线速率
0	MAC_OP_RATE	R/W	0h	为 AUX_MAC 选择运行速率。 0h = SCE 速率 1h = AUX 总线速率

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.9.21 ADCCLKCTL 寄存器（偏移 = 50h）[复位 = 00000000h]

图 19-174 展示了 ADCCLKCTL，表 19-196 中对此进行了介绍。

返回到汇总表。

ADC 时钟控制

图 19-174. ADCCLKCTL 寄存器

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														ACK	REQ
R-0h														R-0h	R/W-0h

表 19-196. ADCCLKCTL 寄存器字段说明

位	字段	类型	复位	说明
31-2	RESERVED	R	0h	保留
1	ACK	R	0h	时钟确认。 0：ADC 时钟禁用。 1：ADC 时钟使能。
0	REQ	R/W	0h	ADC 时钟请求。 0：禁用 ADC 时钟。 1：使能 ADC 时钟。 仅在 REQ 等于 ACK 时修改 REQ。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.9.22 TDCCLKCTL 寄存器（偏移 = 54h）[复位 = 00000000h]

图 19-175 展示了 TDCCLKCTL，表 19-197 中对此进行了介绍。

返回到汇总表。

TDC 计数器时钟控制
控制 AUX_TDC 计数器时钟源是否使能。TDC 计数器时钟源在 DDI_0_OSC:CTL0.ACLK_TDC_SRC_SEL 中配置。

图 19-175. TDCCLKCTL 寄存器

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														ACK	REQ
R-0h														R-0h	R/W-0h

表 19-197. TDCCLKCTL 寄存器字段说明

位	字段	类型	复位	说明
31-2	RESERVED	R	0h	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。
1	ACK	R	0h	TDC 计数器时钟确认。 0：TDC 计数器时钟禁用。 1：TDC 计数器时钟使能。
0	REQ	R/W	0h	TDC 计数器时钟请求。 0：禁用 TDC 计数器时钟。 1：使能 TDC 计数器时钟。 仅在 REQ 等于 ACK 时修改 REQ。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.9.23 TDCREFCLKCTL 寄存器（偏移 = 58h）[复位 = 00000000h]

图 19-176 展示了 TDCREFCLKCTL，表 19-198 中对此进行了介绍。

返回到汇总表。

TDC 基准时钟控制
控制 AUX_TDC 基准时钟源是否使能。该时钟将与 AUX_TDC 计数器时钟进行比较。TDC 基准时钟源在 DDI_0_OSC:CTL0.ACLK_REF_SRC_SEL 中配置。

图 19-176. TDCREFCLKCTL 寄存器

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														ACK	REQ
R-0h														R-0h	R/W-0h

表 19-198. TDCREFCLKCTL 寄存器字段说明

位	字段	类型	复位	说明
31-2	RESERVED	R	0h	保留
1	ACK	R	0h	TDC 基准时钟确认。 0：TDC 基准时钟禁用。 1：TDC 基准时钟使能。
0	REQ	R/W	0h	TDC 基准时钟请求。 0：禁用 TDC 基准时钟。 1：使能 TDC 基准时钟。 仅在 REQ 等于 ACK 时修改 REQ。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.9.24 TIMER2CLKCTL 寄存器（偏移 = 5Ch）[复位 = 00000000h]

图 19-177 展示了 TIMER2CLKCTL，表 19-199 对其进行了介绍。

返回到汇总表。

AUX_TIMER2 时钟控制
仅当 TIMER2CLKSTAT.STAT 不等于 NONE 时，才能访问 AUX_TIMER2。

图 19-177. TIMER2CLKCTL 寄存器

31	30	29	28	27	26	25	24	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																													SRC		
R-0h																													R/W-0h		

表 19-199. TIMER2CLKCTL 寄存器字段说明

位	字段	类型	复位	说明
31-3	RESERVED	R	0h	保留
2-0	SRC	R/W	0h	为 AUX_TIMER2 选择时钟源。 仅当 SRC 等于 TIMER2CLKSTAT.STAT 或 TIMER2CLKSWITCH.RDY 为 1 时才接受更新。 建议仅在 TIMER2BRIDGE.BUSY 为 0 时选择 NONE。 非枚举值将被忽略。 0h = 无时钟 1h = SCLK_LF 2h = SCLK_MF 4h = SCLK_HF/2

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.9.25 TIMER2CLKSTAT 寄存器（偏移 = 60h）[复位 = 00000000h]

图 19-178 展示了 TIMER2CLKSTAT，表 19-200 对其进行了介绍。

返回到汇总表。

AUX_TIMER2 时钟状态

图 19-178. TIMER2CLKSTAT 寄存器

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

表 19-200. TIMER2CLKSTAT 寄存器字段说明

位	字段	类型	复位	说明
31-3	RESERVED	R	0h	保留
2-0	STAT	R	0h	AUX_TIMER2 时钟源状态。 0h = 无时钟 1h = SCLK_LF 2h = SCLK_MF 4h = SCLK_HF/2

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.9.26 TIMER2CLKSWITCH 寄存器（偏移 = 64h）[复位 = 00000001h]

图 19-179 展示了 TIMER2CLKSWITCH，表 19-201 对其进行了介绍。

返回到汇总表。

AUX_TIMER2 时钟切换

图 19-179. TIMER2CLKSWITCH 寄存器

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

表 19-201. TIMER2CLKSWITCH 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	RDY	R	1h	时钟切换器状态。 0：TIMER2CLKCTL.SRC 与 TIMER2CLKSTAT.STAT 不同。 1：TIMER2CLKCTL.SRC 等于 TIMER2CLKSTAT.STAT。 RDY 连接到 AUX_EVCTL:EVSTAT3.AUX_TIMER2_CLKSWITCH_RDY。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.9.27 TIMER2DBGCTL 寄存器（偏移 = 68h）[复位 = 00000000h]

图 19-180 展示了 TIMER2DBGCTL，表 19-202 对其进行了介绍。

返回到汇总表。

AUX_TIMER2 调试控制

图 19-180. TIMER2DBGCTL 寄存器

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

表 19-202. TIMER2DBGCTL 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	DBG_FREEZE_EN	R/W	0h	调试冻结使能。 0：当系统 CPU 在调试模式下暂停时，AUX_TIMER2 不会暂停。 1：当系统 CPU 在调试模式下暂停时，暂停 AUX_TIMER2。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.9.28 CLKSHIFTDET 寄存器（偏移 = 70h）[复位 = 00000001h]

图 19-181 展示了 CLKSHIFTDET，表 19-203 中对此进行了介绍。

返回到汇总表。

时钟移位检测
当 AUX 时钟速率源自 SCLK_MF 或 SCLK_LF 时，MCU 域状态的转换会导致 SCE 时钟周期发生非累加式变化：
- 当 MCU 域进入活动状态时，单个 SCE 时钟周期会变长 6 到 8 个 SCLK_HF 周期。
- 当 MCU 域退出活动状态时，单个 SCE 时钟周期会缩短 6 到 8 个 SCLK_HF 周期。
AUX_SCE 会检测在清除 STAT 到读取 STAT 之间的时间段内，SCE 时钟是否发生了此类事件。

图 19-181. CLKSHIFTDET 寄存器

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

表 19-203. CLKSHIFTDET 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	STAT	R/W	1h	时钟移位检测。 写入: 0：重启时钟移位检测。 1：请勿使用。 读取： 0：自上次向 STAT 写入 0 以来，MCU 域未曾进入或退出活动状态。 1：自上次向 STAT 写入 0 以来，MCU 域曾经进入或退出活动状态。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.9.29 RECHARGETRIG 寄存器（偏移 = 74h）[复位 = 00000000h]

图 19-182 展示了 RECHARGETRIG，表 19-204 中对此进行了介绍。

返回到汇总表。

VDDR 充电触发器

图 19-182. RECHARGETRIG 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	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							R/W-0h

表 19-204. RECHARGETRIG 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	TRIG	R/W	0h	充电触发器。 0：没有影响。 1：请求 VDDR 充电。 仅当 AUX_EVCTL:EVSTAT2.PWR_DWN 为 1 时请求 VDDR 充电。 当 OPMODEREQ.REQ 为 LP 时，请遵循以下顺序： - 设置 TRIG。 - 等待 AUX_EVCTL:EVSTAT2.VDDR_RECHARGE 变为 1。 - 清除 TRIG。 - 等待 AUX_EVCTL:EVSTAT2.VDDR_RECHARGE 变为 0。 当 OPMODEREQ.REQ 为 PDA 或 PDLP 时，请遵循以下顺序： - 设置 TRIG。 - 清除 TRIG。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.9.30 RECHARGEDET 寄存器（偏移 = 78h）[复位 = 00000000h]

图 19-183 展示了 RECHARGEDET，表 19-205 中对此进行了介绍。

返回到汇总表。

VDDR 充电检测
某些应用可能对 VDDR 充电引起的电源噪声很敏感。您可以检测是否发生 VDDR 充电。

图 19-183. RECHARGEDET 寄存器

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

表 19-205. RECHARGEDET 寄存器字段说明

位	字段	类型	复位	说明
31-2	RESERVED	R	0h	保留
1	STAT	R	0h	VDDR 充电检测器状态。 0：自 EN 置位以来，未发生过 VDDR 充电。 1：自 EN 置位以来，发生过 VDDR 充电。
0	EN	R/W	0h	VDDR 充电检测器使能。 0：禁用充电检测。STAT 变为零。 1：使能充电检测。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.9.31 RTCSUBSECINC0 寄存器（偏移 = 7Ch）[复位 = 00000000h]

图 19-184 展示了 RTCSUBSECINC0，表 19-206 对其进行了介绍。

返回到汇总表。

实时计数器亚秒增量 0
当 RTCSUBSECINCCTL.UPD_REQ 置位时，INC15_0 将替换 AON_RTC:SUBSECINC 中的位 15:0。

图 19-184. RTCSUBSECINC0 寄存器

31	30	29	28	27	26	25	24	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																INC15_0															
R-0h																R/W-0h															

表 19-206. RTCSUBSECINC0 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	INC15_0	R/W	0h	AON_RTC:SUBSECINC 中的位 15:0 的新值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.9.32 RTCSUBSECINC1 寄存器（偏移 = 80h）[复位 = 00000000h]

图 19-185 展示了 RTCSUBSECINC1，表 19-207 对其进行了介绍。

返回到汇总表。

实时计数器亚秒增量 1
当 RTCSUBSECINCCTL.UPD_REQ 置位时，INC23_16 将替换 AON_RTC:SUBSECINC 中的位 23:16。

图 19-185. RTCSUBSECINC1 寄存器

31	30	29	28	27	26	25	24	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																								INC23_16							
R-0h																								R/W-0h							

表 19-207. RTCSUBSECINC1 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-0	INC23_16	R/W	0h	AON_RTC:SUBSECINC 中的位 23:16 的新值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.9.33 RTCSUBSECINCCTL 寄存器（偏移 = 84h）[复位 = 00000000h]

图 19-186 展示了 RTCSUBSECINCCTL，表 19-208 中对此进行了介绍。

返回到汇总表。

实时计数器亚秒增量控制

图 19-186. RTCSUBSECINCCTL 寄存器

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

表 19-208. RTCSUBSECINCCTL 寄存器字段说明

位	字段	类型	复位	说明
31-2	RESERVED	R	0h	保留
1	UPD_ACK	R	0h	更新确认。 0：AON_RTC 未确认 UPD_REQ。 1：AON_RTC 已确认 UPD_REQ。
0	UPD_REQ	R/W	0h	请求 AON_RTC 更新 AON_RTC:SUBSECINC。 0：清除更新请求。 1：设置更新请求。 仅在 UPD_REQ 等于 UPD_ACK 时更改 UPD_REQ。UPD_ACK 变为 1 后清除 UPD_REQ。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.9.34 RTCSEC 寄存器（偏移 = 88h）[复位 = 00000000h]

图 19-187 展示了 RTCSEC，表 19-209 中对此进行了介绍。

返回到汇总表。

实时计数器秒
系统 CPU 不得访问此寄存器。相反，系统 CPU 必须直接访问 AON_RTC:SEC.VALUE。

图 19-187. RTCSEC 寄存器

31	30	29	28	27	26	25	24	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																SEC															
R-0h																R-0h															

表 19-209. RTCSEC 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	SEC	R	0h	AON_RTC:SEC.VALUE 中的位 15:0。 请按照以下过程获取正确的值： - 执行两次 SEC 虚拟读取。 - 然后读取 SEC，直到连续两次读取的值相等。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.9.35 RTCSUBSEC 寄存器（偏移 = 8Ch）[复位 = 00000000h]

图 19-188 展示了 RTCSUBSEC，表 19-210 中对此进行了介绍。

返回到汇总表。

实时计数器亚秒
系统 CPU 不得访问此寄存器。相反，系统 CPU 必须直接访问 AON_RTC:SUBSEC.VALUE。

图 19-188. RTCSUBSEC 寄存器

31	30	29	28	27	26	25	24	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																SUBSEC															
R-0h																R-0h															

表 19-210. RTCSUBSEC 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	SUBSEC	R	0h	AON_RTC:SUBSEC.VALUE 中的位 31:16。 请按照以下过程获取正确的值： - 执行两次 SUBSEC 虚拟读取。 - 然后读取 SUBSEC，直到连续两次读取的值相等。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.9.36 RTCEVCLR 寄存器（偏移 = 90h）[复位 = 00000000h]

图 19-189 展示了 RTCEVCLR，表 19-211 中对此进行了介绍。

返回到汇总表。

AON_RTC 事件清除
请求清除以下事件：
- AON_RTC:EVFLAGS.CH2。
- AON_RTC:EVFLAGS.CH2 延迟版本。
- AUX_EVCTL:EVSTAT2.AON_RTC_CH2。
- AUX_EVCTL:EVSTAT2.AON_RTC_CH2_DLY。

图 19-189. RTCEVCLR 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							RTC_CH2_EV_CLR
R-0h							R/W-0h

表 19-211. RTCEVCLR 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	RTC_CH2_EV_CLR	R/W	0h	清除来自 AON_RTC 通道 2 的事件。 0：没有影响。 1：清除来自 AON_RTC 通道 2 的事件。 将 RTC_CH2_EV_CLR 保持为高电平，直到 AUX_EVCTL:EVSTAT2.AON_RTC_CH2 和 AUX_EVCTL:EVSTAT2.AON_RTC_CH2_DLY 为 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.9.37 BATMONBAT 寄存器（偏移 = 94h）[复位 = 00000000h]

图 19-190 展示了 BATMONBAT，表 19-212 中对此进行了介绍。

返回到汇总表。

AON_BATMON 电池电压值
对 AON_BATMON:BAT 的读取访问。系统 CPU 不得访问此寄存器。相反，系统 CPU 必须直接访问 AON_BATMON:BAT。AON_BATMON:BAT 在 VDDR 充电或活动工作模式期间更新。

图 19-190. BATMONBAT 寄存器

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

表 19-212. BATMONBAT 寄存器字段说明

位	字段	类型	复位	说明
31-11	RESERVED	R	0h	保留
10-8	INT	RH	0h	请参阅 AON_BATMON:BAT.INT。 请按照以下过程获取正确的值： - 执行两次 INT 虚拟读取。 - 然后读取 INT，直到连续两次读取的值相等。
7-0	FRAC	R	0h	请参阅 AON_BATMON:BAT.FRAC。 请按照以下过程获取正确的值： - 执行两次 FRAC 虚拟读取。 - 然后读取 FRAC，直到连续两次读取的值相等。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.9.38 BATMONTEMP 寄存器（偏移 = 9Ch）[复位 = 00000000h]

图 19-191 展示了 BATMONTEMP，表 19-213 中对此进行了介绍。

返回到汇总表。

AON_BATMON 温度值
对 AON_BATMON:TEMP 的读取访问。系统 CPU 不得访问此寄存器。相反，系统 CPU 必须直接访问 AON_BATMON:TEMP。AON_BATMON:TEMP 在 VDDR 充电或活动工作模式期间更新。

图 19-191. BATMONTEMP 寄存器

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
SIGN					INT									FRAC	
R-0h					RH-0h									R-0h	

表 19-213. BATMONTEMP 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-11	SIGN	R	0h	INT 的符号扩展。 请按照以下过程获取正确的值： - 执行两次 SIGN 虚拟读取。 - 然后读取 SIGN，直到连续两次读取的值相等。
10-2	INT	RH	0h	请参阅 AON_BATMON:TEMP.INT。 请按照以下过程获取正确的值： - 执行两次 INT 虚拟读取。 - 然后读取 INT，直到连续两次读取的值相等。
1-0	FRAC	R	0h	请参阅 AON_BATMON:TEMP.FRAC。 请按照以下过程获取正确的值： - 执行两次 FRAC 虚拟读取。 - 然后读取 FRAC，直到连续两次读取的值相等。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.9.39 TIMERHALT 寄存器（偏移 = A0h）[复位 = 00000000h]

图 19-192 展示了 TIMERHALT，表 19-214 中对此进行了介绍。

返回到汇总表。

计时器暂停
调试寄存器

图 19-192. TIMERHALT 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				PROGDLY	AUX_TIMER2	AUX_TIMER1	AUX_TIMER0
R-0h				RH/W-0h	RH/W-0h	RH/W-0h	RH/W-0h

表 19-214. TIMERHALT 寄存器字段说明

位	字段	类型	复位	说明
31-4	RESERVED	R	0h	保留
3	PROGDLY	RH/W	0h	暂停可编程延迟。 0：AUX_EVCTL:PROGDLY.VALUE 正常递减。 1：暂停 AUX_EVCTL:PROGDLY.VALUE 递减。
2	AUX_TIMER2	RH/W	0h	暂停 AUX_TIMER2。 0：AUX_TIMER2 正常运行。 1：暂停 AUX_TIMER2 运行。
1	AUX_TIMER1	RH/W	0h	暂停 AUX_TIMER01 计时器 1。 0：AUX_TIMER01 计时器 1 正常运行。 1：暂停 AUX_TIMER01 计时器 1 运行。
0	AUX_TIMER0	RH/W	0h	暂停 AUX_TIMER01 计时器 0。 0：AUX_TIMER01 计时器 0 正常运行。 1：暂停 AUX_TIMER01 计时器 0 运行。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.9.40 TIMER2BRIDGE 寄存器（偏移 = B0h）[复位 = 00000000h]

图 19-193 展示了 TIMER2BRIDGE，表 19-215 对其进行了介绍。

返回到汇总表。

AUX_TIMER2 桥接器

图 19-193. TIMER2BRIDGE 寄存器

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

表 19-215. TIMER2BRIDGE 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	BUSY	R	0h	AUX_TIMER2 的总线事务状态。 0：没有未完成的总线事务。 1：总线事务正在进行中。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.9.41 SWPWRPROF 寄存器（偏移 = B4h）[复位 = 00000000h]

图 19-194 展示了 SWPWRPROF，表 19-216 中对此进行了介绍。

返回到汇总表。

软件功耗分析器

图 19-194. SWPWRPROF 寄存器

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

表 19-216. SWPWRPROF 寄存器字段说明

位	字段	类型	复位	说明
31-3	RESERVED	R	0h	保留
2-0	STAT	R/W	0h	功耗分析器可以读取的软件状态位。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.10 AUX_SPIM 寄存器

表 19-217 列出了 AUX_SPIM 寄存器的存储器映射寄存器。表 19-217 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

表 19-217. AUX_SPIM 寄存器

偏移	首字母缩写词	寄存器名称	部分
0h	SPIMCFG	SPI 主模式配置	SPIMCFG 寄存器 (偏移 = 0h) [复位 = 00000000h]
4h	MISOCFG	MISO 配置	MISOCFG 寄存器 (偏移 = 4h) [复位 = 00000000h]
8h	MOSICTL	MOSI 控制	MOSICTL 寄存器 (偏移 = 8h) [复位 = 00000000h]
Ch	TX8	发送 8 位	TX8 寄存器 (偏移 = Ch) [复位 = 00000000h]
10h	TX16	发送 16 位	TX16 寄存器 (偏移 = 10h) [复位 = 00000000h]
14h	RX8	接收 8 位	RX8 寄存器 (偏移 = 14h) [复位 = 00000000h]
18h	RX16	接收 16 位	RX16 寄存器 (偏移 = 18h) [复位 = 00000000h]
1Ch	SCLKIDLE	SCLK 空闲	SCLKIDLE 寄存器 (偏移 = 1Ch) [复位 = 00000001h]
20h	DATAIDLE	数据空闲	DATAIDLE 寄存器 (偏移 = 20h) [复位 = 00000001h]

复杂的位访问类型经过编码可适应小型表单元。表 19-218 展示了适用于此部分中访问类型的代码。

表 19-218. AUX_SPIM 访问类型代码

访问类型	代码	说明
读取类型		
R	R	读取
写入类型		
W	W	写入
复位或默认值		
-n		复位后的值或默认值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.10.1 SPIMCFG 寄存器（偏移 = 0h）[复位 = 00000000h]

图 19-195 展示了 SPIMCFG，表 19-219 中对此进行了介绍。

返回到汇总表。

SPI 主器件配置
写入操作暂停，直到当前传输完成。

图 19-195. SPIMCFG 寄存器

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								DIV						PHA	POL
R-0h								R/W-0h						R/W-0h	R/W-0h

表 19-219. SPIMCFG 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-2	DIV	R/W	0h	SCLK 分频器。 外设时钟频率分频提供 SCLK 时钟频率。分频因子等于 (2 * (DIV+1)) : 0x00：除以 2。 0x01：除以 4。 0x02：除以 6。 ... 0x3F: 除以 128。
1	PHA	R/W	0h	MOSI 和 MISO 数据信号的相位。 0：在 SCLK 的前（奇数）边沿对 MISO 进行采样，在后（偶数）边沿对 MOSI 进行移位。 1：在 SCLK 的后（偶数）边沿对 MISO 进行采样，在前（奇数）边沿对 MOSI 进行移位。
0	POL	R/W	0h	SCLK 信号的极性。 0：SCLK 在空闲时为低电平，第一个时钟边沿上升。 1：SCLK 在空闲时为高电平，第一个时钟边沿下降。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.10.2 MISOCFG 寄存器（偏移 = 4h）[复位 = 00000000h]

图 19-196 展示了 MISOCFG，表 19-220 中对此进行了介绍。

返回到汇总表。

MISO 配置
写入操作暂停，直到当前传输完成。

图 19-196. MISOCFG 寄存器

31	30	29	28	27	26	25	24	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																											AUXIO			
R-0h																											R/W-0h			

表 19-220. MISOCFG 寄存器字段说明

位	字段	类型	复位	说明
31-5	RESERVED	R	0h	保留
4-0	AUXIO	R/W	0h	AUXIO 至 MISO 多路复用器。 选择连接到 MISO 的 AUXIO 引脚。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.10.3 MOSICTL 寄存器（偏移 = 8h）[复位 = 00000000h]

图 19-197 展示了 MOSICTL，表 19-221 中对此进行了介绍。

返回到汇总表。

MOSI 控制
写入操作暂停，直到当前传输完成。

图 19-197. MOSICTL 寄存器

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

表 19-221. MOSICTL 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	值	W	0h	MOSI 电平控制。 0：将 MOSI 设置为低电平。 1：将 MOSI 设置为高电平。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.10.4 TX8 寄存器（偏移 = Ch）[复位 = 00000000h]

图 19-198 展示了 TX8，表 19-222 对其进行了介绍。

返回到汇总表。

发送 8 位
写入操作暂停，直到当前传输完成。

图 19-198. TX8 寄存器

31	30	29	28	27	26	25	24	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							

表 19-222. TX8 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-0	DATA	W	0h	8 位数据传输。 写入 DATA 以开始传输，MSB 在前。传输完成时，MOSI 保持在 LSB 的值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.10.5 TX16 寄存器（偏移 = 10h）[复位 = 00000000h]

图 19-199 展示了 TX16，表 19-223 对其进行了介绍。

返回到汇总表。

发送 16 位
写入操作暂停，直到当前传输完成。

图 19-199. TX16 寄存器

31	30	29	28	27	26	25	24	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															

表 19-223. TX16 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	DATA	W	0h	16 位数据传输。 写入 DATA 以开始传输，MSB 在前。传输完成时，MOSI 保持在 LSB 的值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.10.6 RX8 寄存器（偏移 = 14h）[复位 = 00000000h]

图 19-200 展示了 RX8，表 19-224 对其进行了介绍。

返回到汇总表。

接收 8 位
读取操作暂停，直到当前传输完成。

图 19-200. RX8 寄存器

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

表 19-224. RX8 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-0	DATA	R	0h	MISO 上接收到的最新 8 位。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.10.7 RX16 寄存器（偏移 = 18h）[复位 = 00000000h]

图 19-201 展示了 RX16，表 19-225 对其进行了介绍。

返回到汇总表。

接收 16 位
读取操作暂停，直到当前传输完成。

图 19-201. RX16 寄存器

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

表 19-225. RX16 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	DATA	R	0h	MISO 上接收到的最新 16 位。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.10.8 SCLKIDLE 寄存器（偏移 = 1Ch）[复位 = 00000001h]

图 19-202 展示了 SCLKIDLE，表 19-226 中对此进行了介绍。

返回到汇总表。

SCLK 空闲
读取操作暂停，直到 SCLK 空闲且没有剩余的时钟边沿。

图 19-202. SCLKIDLE 寄存器

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

表 19-226. SCLKIDLE 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	STAT	R	1h	等待 SCLK 空闲。 读取操作暂停，直到 SCLK 空闲且没有剩余的时钟边沿。然后读取返回 1。 AUX_SCE 可以使用此功能来控制 CS 置为无效。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.10.9 DATAIDLE 寄存器（偏移 = 20h）[复位 = 00000001h]

图 19-203 展示了 DATAIDLE，表 19-227 中对此进行了介绍。

返回到汇总表。

数据空闲
读取操作暂停，直到当前传输完成。

图 19-203. DATAIDLE 寄存器

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

表 19-227. DATAIDLE 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	STAT	R	1h	等待数据空闲。 读取操作暂停，直到与 LSB 传输相关的 SCLK 周期完成。然后读取返回 1。 AUX_SCE 可以使用此功能来控制 CS 置为无效。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.11 AUX_MAC 寄存器

表 19-228 列出了 AUX_MAC 寄存器的存储器映射寄存器。表 19-228 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

表 19-228. AUX_MAC 寄存器

偏移	首字母缩写词	寄存器名称	部分
0h	OP0S	有符号操作数 0	OP0S 寄存器（偏移 = 0h）[复位 = 00000000h]
4h	OP0U	无符号操作数 0	OP0U 寄存器（偏移 = 4h）[复位 = 00000000h]
8h	OP1SMUL	有符号操作数 1 和乘法	OP1SMUL 寄存器（偏移 = 8h）[复位 = 00000000h]
Ch	OP1UMUL	无符号操作数 1 和乘法	OP1UMUL 寄存器（偏移 = Ch）[复位 = 00000000h]
10h	OP1SMAC	有符号操作数 1 和乘法累加	OP1SMAC 寄存器（偏移 = 10h）[复位 = 00000000h]
14h	OP1UMAC	无符号操作数 1 和乘法累加	OP1UMAC 寄存器（偏移 = 14h）[复位 = 00000000h]
18h	OP1SADD16	有符号操作数 1 和 16 位加法	OP1SADD16 寄存器（偏移 = 18h）[复位 = 00000000h]
1Ch	OP1UADD16	无符号操作数 1 和 16 位加法	OP1UADD16 寄存器（偏移 = 1Ch）[复位 = 00000000h]
20h	OP1SADD32	有符号操作数 1 和 32 位加法	OP1SADD32 寄存器（偏移 = 20h）[复位 = 00000000h]
24h	OP1UADD32	无符号操作数 1 和 32 位加法	OP1UADD32 寄存器（偏移 = 24h）[复位 = 00000000h]
28h	CLZ	前导零计数	CLZ 寄存器（偏移 = 28h）[复位 = 00000028h]
2Ch	CLS	前导符号位计数	CLS 寄存器（偏移 = 2Ch）[复位 = 00000028h]
30h	ACCSHIFT	累加器移位	ACCSHIFT 寄存器（偏移 = 30h）[复位 = 00000000h]
34h	ACCRESET	累加器复位	ACCRESET 寄存器（偏移 = 34h）[复位 = 00000000h]
38h	ACC15_0	累加器位 15:0	ACC15_0 寄存器（偏移 = 38h）[复位 = 00000000h]
3Ch	ACC16_1	累加器位 16:1	ACC16_1 寄存器（偏移 = 3Ch）[复位 = 00000000h]
40h	ACC17_2	累加器位 17:2	ACC17_2 寄存器（偏移 = 40h）[复位 = 00000000h]
44h	ACC18_3	累加器位 18:3	ACC18_3 寄存器（偏移 = 44h）[复位 = 00000000h]
48h	ACC19_4	累加器位 19:4	ACC19_4 寄存器（偏移 = 48h）[复位 = 00000000h]
4Ch	ACC20_5	累加器位 20:5	ACC20_5 寄存器（偏移 = 4Ch）[复位 = 00000000h]
50h	ACC21_6	累加器位 21:6	ACC21_6 寄存器（偏移 = 50h）[复位 = 00000000h]
54h	ACC22_7	累加器位 22:7	ACC22_7 寄存器（偏移 = 54h）[复位 = 00000000h]
58h	ACC23_8	累加器位 23:8	ACC23_8 寄存器（偏移 = 58h）[复位 = 00000000h]

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 19-228. AUX_MAC 寄存器（续）

偏移	首字母缩写词	寄存器名称	部分
5Ch	ACC24_9	累加器位 24:9	ACC24_9 寄存器（偏移 = 5Ch）[复位 = 00000000h]
60h	ACC25_10	累加器位 25:10	ACC25_10 寄存器（偏移 = 60h）[复位 = 00000000h]
64h	ACC26_11	累加器位 26:11	ACC26_11 寄存器（偏移 = 64h）[复位 = 00000000h]
68h	ACC27_12	累加器位 27:12	ACC27_12 寄存器（偏移 = 68h）[复位 = 00000000h]
6Ch	ACC28_13	累加器位 28:13	ACC28_13 寄存器（偏移 = 6Ch）[复位 = 00000000h]
70h	ACC29_14	累加器位 29:14	ACC29_14 寄存器（偏移 = 70h）[复位 = 00000000h]
74h	ACC30_15	累加器位 30:15	ACC30_15 寄存器（偏移 = 74h）[复位 = 00000000h]
78h	ACC31_16	累加器位 31:16	ACC31_16 寄存器（偏移 = 78h）[复位 = 00000000h]
7Ch	ACC32_17	累加器位 32:17	ACC32_17 寄存器（偏移 = 7Ch）[复位 = 00000000h]
80h	ACC33_18	累加器位 33:18	ACC33_18 寄存器（偏移 = 80h）[复位 = 00000000h]
84h	ACC34_19	累加器位 34:19	ACC34_19 寄存器（偏移 = 84h）[复位 = 00000000h]
88h	ACC35_20	累加器位 35:20	ACC35_20 寄存器（偏移 = 88h）[复位 = 00000000h]
8Ch	ACC36_21	累加器位 36:21	ACC36_21 寄存器（偏移 = 8Ch）[复位 = 00000000h]
90h	ACC37_22	累加器位 37:22	ACC37_22 寄存器（偏移 = 90h）[复位 = 00000000h]
94h	ACC38_23	累加器位 38:23	ACC38_23 寄存器（偏移 = 94h）[复位 = 00000000h]
98h	ACC39_24	累加器位 39:24	ACC39_24 寄存器（偏移 = 98h）[复位 = 00000000h]
9Ch	ACC39_32	累加器位 39:32	ACC39_32 寄存器（偏移 = 9Ch）[复位 = 00000000h]

复杂的位访问类型经过编码可适应小型表单元。表 19-229 展示了适用于此部分中访问类型的代码。

表 19-229. AUX_MAC 访问类型代码

访问类型	代码	说明
读取类型		
R	R	读取
写入类型		
W	W	写入
复位或默认值		
-n		复位后的值或默认值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.11.1 OP0S 寄存器（偏移 = 0h）[复位 = 00000000h]

图 19-204 展示了 OP0S，表 19-230 对其进行了介绍。

返回到汇总表。

有符号操作数 0

图 19-204. OP0S 寄存器

31	30	29	28	27	26	25	24	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																OP0_VALUE															
R-0h																W-0h															

表 19-230. OP0S 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	OP0_VALUE	W	0h	有符号操作数 0。 乘法、乘法和累加或 32 位加法运算的操作数。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.11.2 OP0U 寄存器（偏移 = 4h）[复位 = 00000000h]

图 19-205 展示了 OP0U，表 19-231 对其进行了介绍。

返回到汇总表。

无符号操作数 0

图 19-205. OP0U 寄存器

31	30	29	28	27	26	25	24	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																OP0_VALUE															
R-0h																W-0h															

表 19-231. OP0U 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	OP0_VALUE	W	0h	无符号操作数 0。 乘法、乘法和累加或 32 位加法运算的操作数。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.11.3 OP1SMUL 寄存器（偏移 = 8h）[复位 = 00000000h]

图 19-206 展示了 OP1SMUL，表 19-232 对其进行了介绍。

返回到汇总表。

有符号操作数 1 和乘法

图 19-206. OP1SMUL 寄存器

31	30	29	28	27	26	25	24	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																OP1_VALUE															
R-0h																W-0h															

表 19-232. OP1SMUL 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	OP1_VALUE	W	0h	有符号操作数 1 和乘法触发器。 写入 OP1_VALUE 以设置有符号操作数 1 并触发以下操作： 将操作数 0 写入 OP0S.OP0_VALUE 时：ACC = OP1_VALUE * OP0S.OP0_VALUE。 将操作数 0 写入 OP0U.OP0_VALUE 时：ACC = OP1_VALUE * OP0U.OP0_VALUE。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.11.4 OP1UMUL 寄存器（偏移 = Ch）[复位 = 00000000h]

图 19-207 展示了 OP1UMUL，表 19-233 对其进行了介绍。

返回到汇总表。

无符号操作数 1 和乘法

图 19-207. OP1UMUL 寄存器

31	30	29	28	27	26	25	24	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																OP1_VALUE															
R-0h																W-0h															

表 19-233. OP1UMUL 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	OP1_VALUE	W	0h	无符号操作数 1 和乘法触发器。 写入 OP1_VALUE 以设置无符号操作数 1 并触发以下操作： 将操作数 0 写入 OP0S.OP0_VALUE 时：ACC = OP1_VALUE * OP0S.OP0_VALUE。 将操作数 0 写入 OP0U.OP0_VALUE 时：ACC = OP1_VALUE * OP0U.OP0_VALUE。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.11.5 OP1SMAC 寄存器（偏移 = 10h）[复位 = 00000000h]

图 19-208 展示了 OP1SMAC，表 19-234 对其进行了介绍。

返回到汇总表。

有符号操作数 1 和乘法累加

图 19-208. OP1SMAC 寄存器

31	30	29	28	27	26	25	24	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																OP1_VALUE															
R-0h																W-0h															

表 19-234. OP1SMAC 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	OP1_VALUE	W	0h	有符号操作数 1 和乘法累加触发器。 写入 OP1_VALUE 以设置有符号操作数 1 并触发以下操作： 将操作数 0 写入 OP0S.OP0_VALUE 时：ACC = ACC + (OP1_VALUE * OP0S.OP0_VALUE)。 将操作数 0 写入 OP0U.OP0_VALUE 时：ACC = ACC + (OP1_VALUE * OP0U.OP0_VALUE)。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.11.6 OP1UMAC 寄存器（偏移 = 14h）[复位 = 00000000h]

图 19-209 展示了 OP1UMAC，表 19-235 对其进行了介绍。

返回到汇总表。

无符号操作数 1 和乘法累加

图 19-209. OP1UMAC 寄存器

31	30	29	28	27	26	25	24	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																OP1_VALUE															
R-0h																W-0h															

表 19-235. OP1UMAC 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	OP1_VALUE	W	0h	无符号操作数 1 和乘法累加触发器。 写入 OP1_VALUE 以设置无符号操作数 1 并触发以下操作： 将操作数 0 写入 OP0S.OP0_VALUE 时：ACC = ACC + (OP1_VALUE * OP0S.OP0_VALUE)。 将操作数 0 写入 OP0U.OP0_VALUE 时：ACC = ACC + (OP1_VALUE * OP0U.OP0_VALUE)。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.11.7 OP1SADD16 寄存器（偏移 = 18h）[复位 = 00000000h]

图 19-210 展示了 OP1SADD16，表 19-236 对其进行了介绍。

返回到汇总表。

有符号操作数 1 和 16 位加法

图 19-210. OP1SADD16 寄存器

31	30	29	28	27	26	25	24	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																OP1_VALUE															
R-0h																W-0h															

表 19-236. OP1SADD16 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	OP1_VALUE	W	0h	有符号操作数 1 和 16 位加法触发器。 写入 OP1_VALUE 以设置有符号操作数 1 并触发以下操作： ACC = ACC + OP1_VALUE。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.11.8 OP1UADD16 寄存器（偏移 = 1Ch）[复位 = 00000000h]

图 19-211 展示了 OP1UADD16，表 19-237 对其进行了介绍。

返回到汇总表。

无符号操作数 1 和 16 位加法

图 19-211. OP1UADD16 寄存器

31	30	29	28	27	26	25	24	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																OP1_VALUE															
R-0h																W-0h															

表 19-237. OP1UADD16 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	OP1_VALUE	W	0h	无符号操作数 1 和 16 位加法触发器。 写入 OP1_VALUE 以设置无符号操作数 1 并触发以下操作： ACC = ACC + OP1_VALUE。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.11.9 OP1SADD32 寄存器（偏移 = 20h）[复位 = 00000000h]

图 19-212 展示了 OP1SADD32，表 19-238 对其进行了介绍。

返回到汇总表。

有符号操作数 1 和 32 位加法

图 19-212. OP1SADD32 寄存器

31	30	29	28	27	26	25	24	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																OP1_VALUE															
R-0h																W-0h															

表 19-238. OP1SADD32 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	OP1_VALUE	W	0h	有符号 32 位操作数的高半部分和加法触发器。 写入 OP1_VALUE 以设置有符号 32 位操作数的高半部分并触发以下操作： 将 32 位操作数的低半部分写入 OP0S.OP0_VALUE 时：ACC = ACC + ((OP1_VALUE << 16) | OP0S.OP0_VALUE)。 将 32 位操作数的低半部分写入 OP0U.OP0_VALUE 时：ACC = ACC + ((OP1_VALUE << 16) | OP0U.OP0_VALUE)。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.11.10 OP1UADD32 寄存器（偏移 = 24h）[复位 = 00000000h]

图 19-213 展示了 OP1UADD32，表 19-239 对其进行了介绍。

返回到汇总表。

无符号操作数 1 和 32 位加法

图 19-213. OP1UADD32 寄存器

31	30	29	28	27	26	25	24	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																OP1_VALUE															
R-0h																W-0h															

表 19-239. OP1UADD32 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	OP1_VALUE	W	0h	无符号 32 位操作数的高半部分和加法触发器。 写入 OP1_VALUE 以设置无符号 32 位操作数的高半部分并触发以下操作： 将 32 位操作数的低半部分写入 OP0S.OP0_VALUE 时：ACC = ACC + ((OP1_VALUE << 16) | OP0S.OP0_VALUE)。 将 32 位操作数的低半部分写入 OP0U.OP0_VALUE 时：ACC = ACC + ((OP1_VALUE << 16) | OP0U.OP0_VALUE)。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.11.11 CLZ 寄存器（偏移 = 28h）[复位 = 00000028h]

图 19-214 展示了 CLZ，表 19-240 中对此进行了介绍。

返回到汇总表。

前导零计数

图 19-214. CLZ 寄存器

31	30	29	28	27	26	25	24	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																										值					
R-0h																										R-28h					

表 19-240. CLZ 寄存器字段说明

位	字段	类型	复位	说明
31-6	RESERVED	R	0h	保留
5-0	值	R	28h	累加器中的前导零位数： 0x00：0 个前导零。 0x01：1 个前导零。 ... 0x28: 40 个前导零（累加器值为 0）。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.11.12 CLS 寄存器（偏移 = 2Ch）[复位 = 00000028h]

图 19-215 展示了 CLS，表 19-241 中对此进行了介绍。

返回到汇总表。

前导符号位计数

图 19-215. CLS 寄存器

31	30	29	28	27	26	25	24	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																										值					
R-0h																										R-28h					

表 19-241. CLS 寄存器字段说明

位	字段	类型	复位	说明
31-6	RESERVED	R	0h	保留
5-0	值	R	28h	累加器中的前导符号位数。 当累加器的 MSB 为 0 时，VALUE 是前导零的数量，包括 MSB。 当累加器的 MSB 为 1 时，VALUE 是前导 1 的数量，包括 MSB。 VALUE 范围为 1 到 40。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.11.13 ACCSHIFT 寄存器（偏移 = 30h）[复位 = 00000000h]

图 19-216 展示了 ACCSHIFT，表 19-242 中对此进行了介绍。

返回到汇总表。

累加器移位
每次寄存器写入只能触发一次移位操作。

图 19-216. ACCSHIFT 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED					LSL1	LSR1	ASR1
R-0h					W-0h	W-0h	W-0h

表 19-242. ACCSHIFT 寄存器字段说明

位	字段	类型	复位	说明
31-3	RESERVED	R	0h	保留
2	LSL1	W	0h	逻辑左移 1 位。 写入 1 将累加器向左移动一位，在位 0 处插入 0。
1	LSR1	W	0h	逻辑右移 1 位。 写入 1 将累加器向右移动一位，在位 39 处插入 0。
0	ASR1	W	0h	算术右移 1 位。 写入 1 将累加器向右移动一位，在位 39 处插入前一个符号位。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.11.14 ACCRESET 寄存器（偏移 = 34h）[复位 = 00000000h]

图 19-217 展示了 ACCRESET，表 19-243 中对此进行了介绍。

返回到汇总表。

累加器复位

图 19-217. ACCRESET 寄存器

31	30	29	28	27	26	25	24	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																TRG															
R-0h																W-0h															

表 19-243. ACCRESET 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	TRG	W	0h	向该寄存器写入任何值可触发累加器中所有位的复位。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.11.15 ACC15_0 寄存器（偏移 = 38h）[复位 = 00000000h]

图 19-218 展示了 ACC15_0，表 19-244 对其进行了介绍。

返回到汇总表。

累加器位 15:0

图 19-218. ACC15_0 寄存器

31	30	29	28	27	26	25	24	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																值															
R-0h																R/W-0h															

表 19-244. ACC15_0 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	值	R/W	0h	累加器的值，位 15:0。 写入 VALUE 以初始化累加器的位 15:0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.11.16 ACC16_1 寄存器（偏移 = 3Ch）[复位 = 00000000h]

图 19-219 展示了 ACC16_1，表 19-245 对其进行了介绍。

返回到汇总表。

累加器位 16:1

图 19-219. ACC16_1 寄存器

31	30	29	28	27	26	25	24	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																值															
R-0h																R-0h															

表 19-245. ACC16_1 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	值	R	0h	累加器的值，位 16:1。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.11.17 ACC17_2 寄存器（偏移 = 40h）[复位 = 00000000h]

图 19-220 展示了 ACC17_2，表 19-246 对其进行了介绍。

返回到汇总表。

累加器位 17:2

图 19-220. ACC17_2 寄存器

31	30	29	28	27	26	25	24	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																值															
R-0h																R-0h															

表 19-246. ACC17_2 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	值	R	0h	累加器的值，位 17:2。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.11.18 ACC18_3 寄存器（偏移 = 44h）[复位 = 00000000h]

图 19-221 展示了 ACC18_3，表 19-247 对其进行了介绍。

返回到汇总表。

累加器位 18:3

图 19-221. ACC18_3 寄存器

31	30	29	28	27	26	25	24	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																值															
R-0h																R-0h															

表 19-247. ACC18_3 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	值	R	0h	累加器的值，位 18:3。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.11.19 ACC19_4 寄存器（偏移 = 48h）[复位 = 00000000h]

图 19-222 展示了 ACC19_4，表 19-248 对其进行了介绍。

返回到汇总表。

累加器位 19:4

图 19-222. ACC19_4 寄存器

31	30	29	28	27	26	25	24	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																值															
R-0h																R-0h															

表 19-248. ACC19_4 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	值	R	0h	累加器的值，位 19:4。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.11.20 ACC20_5 寄存器（偏移 = 4Ch）[复位 = 00000000h]

图 19-223 展示了 ACC20_5，表 19-249 对其进行了介绍。

返回到汇总表。

累加器位 20:5

图 19-223. ACC20_5 寄存器

31	30	29	28	27	26	25	24	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																值															
R-0h																R-0h															

表 19-249. ACC20_5 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	值	R	0h	累加器的值，位 20:5。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.11.21 ACC21_6 寄存器（偏移 = 50h）[复位 = 00000000h]

图 19-224 展示了 ACC21_6，表 19-250 对其进行了介绍。

返回到汇总表。

累加器位 21:6

图 19-224. ACC21_6 寄存器

31	30	29	28	27	26	25	24	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																值															
R-0h																R-0h															

表 19-250. ACC21_6 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	值	R	0h	累加器的值，位 21:6。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.11.22 ACC22_7 寄存器（偏移 = 54h）[复位 = 00000000h]

图 19-225 展示了 ACC22_7，表 19-251 对其进行了介绍。

返回到汇总表。

累加器位 22:7

图 19-225. ACC22_7 寄存器

31	30	29	28	27	26	25	24	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																值															
R-0h																R-0h															

表 19-251. ACC22_7 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	值	R	0h	累加器的值，位 22:7。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.11.23 ACC23_8 寄存器（偏移 = 58h）[复位 = 00000000h]

图 19-226 展示了 ACC23_8，表 19-252 对其进行了介绍。

返回到汇总表。

累加器位 23:8

图 19-226. ACC23_8 寄存器

31	30	29	28	27	26	25	24	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																值															
R-0h																R-0h															

表 19-252. ACC23_8 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	值	R	0h	累加器的值，位 23:8。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.11.24 ACC24_9 寄存器（偏移 = 5Ch）[复位 = 00000000h]

图 19-227 展示了 ACC24_9，表 19-253 对其进行了介绍。

返回到汇总表。

累加器位 24:9

图 19-227. ACC24_9 寄存器

31	30	29	28	27	26	25	24	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																值															
R-0h																R-0h															

表 19-253. ACC24_9 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	值	R	0h	累加器的值，位 24:9。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.11.25 ACC25_10 寄存器（偏移 = 60h）[复位 = 00000000h]

图 19-228 展示了 ACC25_10，表 19-254 对其进行了介绍。

返回到汇总表。

累加器位 25:10

图 19-228. ACC25_10 寄存器

31	30	29	28	27	26	25	24	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																值															
R-0h																R-0h															

表 19-254. ACC25_10 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	值	R	0h	累加器的值，位 25:10。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.11.26 ACC26_11 寄存器（偏移 = 64h）[复位 = 00000000h]

图 19-229 展示了 ACC26_11，表 19-255 对其进行了介绍。

返回到汇总表。

累加器位 26:11

图 19-229. ACC26_11 寄存器

31	30	29	28	27	26	25	24	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																值															
R-0h																R-0h															

表 19-255. ACC26_11 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	值	R	0h	累加器的值，位 26:11。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.11.27 ACC27_12 寄存器（偏移 = 68h）[复位 = 00000000h]

图 19-230 展示了 ACC27_12，表 19-256 对其进行了介绍。

返回到汇总表。

累加器位 27:12

图 19-230. ACC27_12 寄存器

31	30	29	28	27	26	25	24	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																值															
R-0h																R-0h															

表 19-256. ACC27_12 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	值	R	0h	累加器的值，位 27:12。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.11.28 ACC28_13 寄存器（偏移 = 6Ch）[复位 = 00000000h]

图 19-231 展示了 ACC28_13，表 19-257 对其进行了介绍。

返回到汇总表。

累加器位 28:13

图 19-231. ACC28_13 寄存器

31	30	29	28	27	26	25	24	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																值															
R-0h																R-0h															

表 19-257. ACC28_13 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	值	R	0h	累加器的值，位 28:13。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.11.29 ACC29_14 寄存器（偏移 = 70h）[复位 = 00000000h]

图 19-232 展示了 ACC29_14，表 19-258 对其进行了介绍。

返回到汇总表。

累加器位 29:14

图 19-232. ACC29_14 寄存器

31	30	29	28	27	26	25	24	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																值															
R-0h																R-0h															

表 19-258. ACC29_14 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	值	R	0h	累加器的值，位 29:14。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.11.30 ACC30_15 寄存器（偏移 = 74h）[复位 = 00000000h]

图 19-233 展示了 ACC30_15，表 19-259 对其进行了介绍。

返回到汇总表。

累加器位 30:15

图 19-233. ACC30_15 寄存器

31	30	29	28	27	26	25	24	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																值															
R-0h																R-0h															

表 19-259. ACC30_15 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	值	R	0h	累加器的值，位 30:15。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.11.31 ACC31_16 寄存器（偏移 = 78h）[复位 = 00000000h]

图 19-234 展示了 ACC31_16，表 19-260 对其进行了介绍。

返回到汇总表。

累加器位 31:16

图 19-234. ACC31_16 寄存器

31	30	29	28	27	26	25	24	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																值															
R-0h																R/W-0h															

表 19-260. ACC31_16 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	值	R/W	0h	累加器的值，位 31:16。 写入 VALUE 以初始化累加器的位 31:16。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.11.32 ACC32_17 寄存器（偏移 = 7Ch）[复位 = 00000000h]

图 19-235 展示了 ACC32_17，表 19-261 对其进行了介绍。

返回到汇总表。

累加器位 32:17

图 19-235. ACC32_17 寄存器

31	30	29	28	27	26	25	24	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																值															
R-0h																R-0h															

表 19-261. ACC32_17 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	值	R	0h	累加器的值，位 32:17。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.11.33 ACC33_18 寄存器（偏移 = 80h）[复位 = 00000000h]

图 19-236 展示了 ACC33_18，表 19-262 对其进行了介绍。

返回到汇总表。

累加器位 33:18

图 19-236. ACC33_18 寄存器

31	30	29	28	27	26	25	24	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																值															
R-0h																R-0h															

表 19-262. ACC33_18 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	值	R	0h	累加器的值，位 33:18。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.11.34 ACC34_19 寄存器（偏移 = 84h）[复位 = 00000000h]

图 19-237 展示了 ACC34_19，表 19-263 对其进行了介绍。

返回到汇总表。

累加器位 34:19

图 19-237. ACC34_19 寄存器

31	30	29	28	27	26	25	24	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																值															
R-0h																R-0h															

表 19-263. ACC34_19 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	值	R	0h	累加器的值，位 34:19。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.11.35 ACC35_20 寄存器（偏移 = 88h）[复位 = 00000000h]

图 19-238 展示了 ACC35_20，表 19-264 对其进行了介绍。

返回到汇总表。

累加器位 35:20

图 19-238. ACC35_20 寄存器

31	30	29	28	27	26	25	24	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																值															
R-0h																R-0h															

表 19-264. ACC35_20 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	值	R	0h	累加器的值，位 35:20。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.11.36 ACC36_21 寄存器（偏移 = 8Ch）[复位 = 00000000h]

图 19-239 展示了 ACC36_21，表 19-265 对其进行了介绍。

返回到汇总表。

累加器位 36:21

图 19-239. ACC36_21 寄存器

31	30	29	28	27	26	25	24	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																值															
R-0h																R-0h															

表 19-265. ACC36_21 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	值	R	0h	累加器的值，位 36:21。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.11.37 ACC37_22 寄存器（偏移 = 90h）[复位 = 00000000h]

图 19-240 展示了 ACC37_22，表 19-266 对其进行了介绍。

返回到汇总表。

累加器位 37:22

图 19-240. ACC37_22 寄存器

31	30	29	28	27	26	25	24	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																值															
R-0h																R-0h															

表 19-266. ACC37_22 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	值	R	0h	累加器的值，位 37:22。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.11.38 ACC38_23 寄存器（偏移 = 94h）[复位 = 00000000h]

图 19-241 展示了 ACC38_23，表 19-267 对其进行了介绍。

返回到汇总表。

累加器位 38:23

图 19-241. ACC38_23 寄存器

31	30	29	28	27	26	25	24	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																值															
R-0h																R-0h															

表 19-267. ACC38_23 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	值	R	0h	累加器的值，位 38:23。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.11.39 ACC39_24 寄存器（偏移 = 98h）[复位 = 00000000h]

图 19-242 展示了 ACC39_24，表 19-268 对其进行了介绍。

返回到汇总表。

累加器位 39:24

图 19-242. ACC39_24 寄存器

31	30	29	28	27	26	25	24	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																值															
R-0h																R-0h															

表 19-268. ACC39_24 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	值	R	0h	累加器的值，位 39:24。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.11.40 ACC39_32 寄存器（偏移 = 9Ch）[复位 = 00000000h]

图 19-243 展示了 ACC39_32，表 19-269 对其进行了介绍。

返回到汇总表。

累加器位 39:32

图 19-243. ACC39_32 寄存器

31	30	29	28	27	26	25	24	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																								值							
R-0h																								R/W-0h							

表 19-269. ACC39_32 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-0	值	R/W	0h	累加器的值，位 39:32。 写入 VALUE 以初始化累加器的位 39:32。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.12 AUX_SCE 寄存器

表 19-270 列出了 AUX_SCE 寄存器的存储器映射寄存器。表 19-270 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

表 19-270. AUX_SCE 寄存器

偏移	首字母缩写词	寄存器名称	部分
0h	CTL	内部	CTL 寄存器（偏移 = 0h）[复位 = 00000000h]
4h	FETCHSTAT	内部	FETCHSTAT 寄存器（偏移 = 4h）[复位 = 00000000h]
8h	CPUSTAT	内部	CPUSTAT 寄存器（偏移 = 8h）[复位 = 00000000h]
Ch	WUSTAT	内部	WUSTAT 寄存器（偏移 = Ch）[复位 = 00000000h]
10h	REG1_0	内部	REG1_0 寄存器（偏移 = 10h）[复位 = 00000000h]
14h	REG3_2	内部	REG3_2 寄存器（偏移 = 14h）[复位 = 00000000h]
18h	REG5_4	内部	REG5_4 寄存器（偏移 = 18h）[复位 = 00000000h]
1Ch	REG7_6	内部	REG7_6 寄存器（偏移 = 1Ch）[复位 = 00000000h]
20h	LOOPADDR	内部	LOOPADDR 寄存器（偏移 = 20h）[复位 = 00000000h]
24h	LOOPCNT	内部	LOOPCNT 寄存器（偏移 = 24h）[复位 = 00000000h]

复杂的位访问类型经过编码可适应小型表单元。表 19-271 展示了适用于此部分中访问类型的代码。

表 19-271. AUX_SCE 访问类型代码

访问类型	代码	说明
读取类型		
R	R	读取
写入类型		
W	W	写入
复位或默认值		
-n		复位后的值或默认值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.12.1 CTL 寄存器（偏移 = 0h）[复位 = 00000000h]

图 19-244 展示了 CTL，表 19-272 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 19-244. CTL 寄存器

31	30	29	28	27	26	25	24
FORCE_EV_LOW							
R/W-0h							
23	22	21	20	19	18	17	16
FORCE_EV_HIGH							
R/W-0h							
15	14	13	12	11	10	9	8
RESET_VECTOR							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED	DBG_FREEZE_EN	FORCE_WU_LOW	FORCE_WU_HIGH	RESTART	SINGLE_STEP	挂起	CLK_EN
R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 19-272. CTL 寄存器字段说明

位	字段	类型	复位	说明
31-24	FORCE_EV_LOW	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
23-16	FORCE_EV_HIGH	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
15-8	RESET_VECTOR	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
7	RESERVED	R	0h	保留
6	DBG_FREEZE_EN	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
5	FORCE_WU_LOW	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
4	FORCE_WU_HIGH	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
3	RESTART	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
2	SINGLE_STEP	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
1	挂起	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
0	CLK_EN	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.12.2 FETCHSTAT 寄存器（偏移 = 4h）[复位 = 00000000h]

图 19-245 展示了 FETCHSTAT，表 19-273 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 19-245. FETCHSTAT 寄存器

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

表 19-273. FETCHSTAT 寄存器字段说明

位	字段	类型	复位	说明
31-16	操作码	R	0h	内部。仅可通过 TI 提供的 API 使用。
15-0	PC	R	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.12.3 CPUSTAT 寄存器（偏移 = 8h）[复位 = 00000000h]

图 19-246 展示了 CPUSTAT，表 19-274 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 19-246. CPUSTAT 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED				BUS_ERROR	SLEEP	WEV	HALTED
R-0h				R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED				V_FLAG	C_FLAG	N_FLAG	Z_FLAG
R-0h				R-0h	R-0h	R-0h	R-0h

表 19-274. CPUSTAT 寄存器字段说明

位	字段	类型	复位	说明
31-12	RESERVED	R	0h	保留
11	BUS_ERROR	R	0h	内部。仅可通过 TI 提供的 API 使用。
10	SLEEP	R	0h	内部。仅可通过 TI 提供的 API 使用。
9	WEV	R	0h	内部。仅可通过 TI 提供的 API 使用。
8	HALTED	R	0h	内部。仅可通过 TI 提供的 API 使用。
7-4	RESERVED	R	0h	保留
3	V_FLAG	R	0h	内部。仅可通过 TI 提供的 API 使用。
2	C_FLAG	R	0h	内部。仅可通过 TI 提供的 API 使用。
1	N_FLAG	R	0h	内部。仅可通过 TI 提供的 API 使用。
0	Z_FLAG	R	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.12.4 WUSTAT 寄存器（偏移 = Ch）[复位 = 00000000h]

图 19-247 展示了 WUSTAT，表 19-275 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 19-247. WUSTAT 寄存器

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

表 19-275. WUSTAT 寄存器字段说明

位	字段	类型	复位	说明
31-19	RESERVED	R	0h	保留
18-16	EXC_VECTOR	R	0h	内部。仅可通过 TI 提供的 API 使用。
15-9	RESERVED	R	0h	保留
8	WU_SIGNAL	R	0h	内部。仅可通过 TI 提供的 API 使用。
7-0	EV_SIGNALS	R	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.12.5 REG1_0 寄存器（偏移 = 10h）[复位 = 00000000h]

图 19-248 展示了 REG1_0，表 19-276 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 19-248. REG1_0 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG1																REG0															
R-0h																R-0h															

表 19-276. REG1_0 寄存器字段说明

位	字段	类型	复位	说明
31-16	REG1	R	0h	内部。仅可通过 TI 提供的 API 使用。
15-0	REG0	R	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.12.6 REG3_2 寄存器（偏移 = 14h）[复位 = 00000000h]

图 19-249 展示了 REG3_2，表 19-277 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 19-249. REG3_2 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG3																REG2															
R-0h																R-0h															

表 19-277. REG3_2 寄存器字段说明

位	字段	类型	复位	说明
31-16	REG3	R	0h	内部。仅可通过 TI 提供的 API 使用。
15-0	REG2	R	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.12.7 REG5_4 寄存器（偏移 = 18h）[复位 = 00000000h]

图 19-250 展示了 REG5_4，表 19-278 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 19-250. REG5_4 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG5																REG4															
R-0h																R-0h															

表 19-278. REG5_4 寄存器字段说明

位	字段	类型	复位	说明
31-16	REG5	R	0h	内部。仅可通过 TI 提供的 API 使用。
15-0	REG4	R	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.12.8 REG7_6 寄存器（偏移 = 1Ch）[复位 = 00000000h]

图 19-251 展示了 REG7_6，表 19-279 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 19-251. REG7_6 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG7																REG6															
R-0h																R-0h															

表 19-279. REG7_6 寄存器字段说明

位	字段	类型	复位	说明
31-16	REG7	R	0h	内部。仅可通过 TI 提供的 API 使用。
15-0	REG6	R	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.12.9 LOOPADDR 寄存器（偏移 = 20h）[复位 = 00000000h]

图 19-252 展示了 LOOPADDR，表 19-280 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 19-252. LOOPADDR 寄存器

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

表 19-280. LOOPADDR 寄存器字段说明

位	字段	类型	复位	说明
31-16	STOP	R	0h	内部。仅可通过 TI 提供的 API 使用。
15-0	启动	R	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

19.8.12.10 LOOPCNT 寄存器（偏移 = 24h）[复位 = 00000000h]

图 19-253 展示了 LOOPCNT，表 19-281 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 19-253. LOOPCNT 寄存器

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								ITER_LEFT							
R-0h								R-0h							

表 19-281. LOOPCNT 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-0	ITER_LEFT	R	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

章节 20

电池监测器和温度传感器 (BATMON)

本章介绍了 CC13x2 和 CC26x2 器件平台中的电池监测器和温度传感器。

20.1 简介 1815
20.2 功能说明 1815
20.3 BATMON 寄存器 1815

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

20.1 简介

电池监测器是一个在启动时自动使能的小型模块，它通过片上温度传感器监测 VDDS 电源电压和温度。使能后，电池和温度监测器模块在除最深功耗模式（STANDBY 和 SHUTDOWN）之外的所有工作模式下均可运行。当器件处于 STANDBY 模式时，电池监测器模块的测量将仅限于充电周期。在每个充电周期中至少会执行两次测量。

电池监测器向多个模块（包括闪存和无线电）提供电压和温度信息，以确保正确运行和最低功耗。因此，不建议修改电池监测器中的任何设置或将其关闭。

20.2 功能说明

电池监测器是一款运行频率为 125kHz 的 7 位类 SAR ADC，可对电源电压和温度传感器执行交替测量。当电池监测器完成第一次测量并趋稳后，它会停止在 SAR 模式下工作，并开始对电压和温度进行线性跟踪。一个小型数字内核将这些测量值转换为以 °C 为单位的电压和温度值，可直接从 AON_BATMON:BAT 和 AON_BATMON:TEMP 寄存器中读取。

当检测到电源电压或温度发生变化时，电池监测器将仅跟踪电压，直到其稳定在新的恒定水平。ADC 的分辨率和 125kHz 时钟速度将限制电池监测器测量电压尖峰的能力。由于电池监测器不仅会在温度和电池电压之间交替，还会检查自上次读取以来是发生了正向变化还是负向变化，因此从电压骤降发生到 ADC 注意到温度或电压发生变化之间可能存在 6 个时钟周期的延迟。由于检测到电压变化后会优先进行电压跟踪，如果电压同时也在变化，温度变化的检测可能会有更多延迟。务必记住这一点，因为电池监测器设计的目的是测量电池电压，而不是测量因短时间内较高电流消耗而产生的电压杂散。

该模块还包含一个事件寄存器 AON_BATMON:EVENT，其中包括六个事件：

- TEMP_UPDATE：表示温度已发生变化
- BATT_UPDATE：表示电压已更改
- TEMP_BELOW_LL：指示温度低于在 AON_BATMON:TEMPLL 寄存器中设置的下限值
- TEMP_OVER_UL：表示温度超过在 AON_BATMON:TEMPUL 寄存器中设置的上限值
- BATT_BELOW_LL：表示电压低于在 AON_BATMON:BATTLL 寄存器中设置的下限值
- BATT_OVER_UL：表示电压高于在 AON_BATMON:BATTUL 寄存器中设置的上限值

必须通过写入 AON_BATMON:EVENT 寄存器来清除这些事件。如果满足事件的条件，事件将再次置为有效（新事件的置位优先于事件的清除）。除了前面列出的各个事件外，电池监测器模块还有一个组合事件作为中断线连接到 CPU。屏蔽寄存器 AON_BATMON:EVENTMASK 可用于选择 AON_BATMON:EVENT 中的哪些事件参与生成组合事件。这些事件连接到 AON 事件结构。相关详细信息，请参阅章节 5。

20.3 BATMON 寄存器

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

20.3.1 AON_BATMON 寄存器

表 20-1 列出了 AON_BATMON 寄存器的存储器映射寄存器。表 20-1 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

表 20-1. AON_BATMON 寄存器

偏移	首字母缩写词	寄存器名称	部分
0h	CTL	内部	CTL 寄存器 (偏移 = 0h) [复位 = 00000000h]
4h	MEASCFG	内部	MEASCFG 寄存器 (偏移 = 4h) [复位 = 00000000h]
Ch	TEMPP0	内部	TEMPP0 寄存器 (偏移 = Ch) [复位 = 00000000h]
10h	TEMPP1	内部	TEMPP1 寄存器 (偏移 = 10h) [复位 = 00000000h]
14h	TEMPP2	内部	TEMPP2 寄存器 (偏移 = 14h) [复位 = 00000000h]
18h	BATMONP0	内部	BATMONP0 寄存器 (偏移 = 18h) [复位 = 00000000h]
1Ch	BATMONP1	内部	BATMONP1 寄存器 (偏移 = 1Ch) [复位 = 00000000h]
20h	IOSTRP0	内部	IOSTRP0 寄存器 (偏移 = 20h) [复位 = 00000028h]
24h	FLASHPUMPP0	内部	FLASHPUMPP0 寄存器 (偏移 = 24h) [复位 = 00000000h]
28h	BAT	最后测量的电池电压	BAT 寄存器 (偏移 = 28h) [复位 = 00000000h]
2Ch	BATUPD	电池更新	BATUPD 寄存器 (偏移 = 2Ch) [复位 = 00000000h]
30h	TEMP	温度	TEMP 寄存器 (偏移 = 30h) [复位 = 00000000h]
34h	TEMPUPD	温度更新	TEMPUPD 寄存器 (偏移 = 34h) [复位 = 00000000h]
48h	EVENTMASK	事件屏蔽	EVENTMASK 寄存器 (偏移 = 48h) [复位 = 00000000h]
4Ch	事件	事件	EVENT 寄存器 (偏移 = 4Ch) [复位 = 00000000h]
50h	BATTUL	电池上限	BATTUL 寄存器 (偏移 = 50h) [复位 = 000007FFh]
54h	BATTLL	电池下限	BATTLL 寄存器 (偏移 = 54h) [复位 = 00000000h]
58h	TEMPUL	温度上限	TEMPUL 寄存器 (偏移 = 58h) [复位 = 0000FFC0h]
5Ch	TEMPLL	温度下限	TEMPLL 寄存器 (偏移 = 5Ch) [复位 = 00010000h]

复杂的位访问类型经过编码可适应小型表单元。表 20-2 展示了适用于此部分中访问类型的代码。

表 20-2. AON_BATMON 访问类型代码

访问类型	代码	说明
读取类型		
R	R	读取
写入类型		
W	W	写入

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 20-2. AON_BATMON 访问类型代码（续）

访问类型	代码	说明
W1C	W 1C	写入 1 以清零
复位或默认值		
-n		复位后的值或默认值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

20.3.1.1 CTL 寄存器（偏移 = 0h）[复位 = 00000000h]

图 20-1 展示了 CTL，表 20-3 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 20-1. CTL 寄存器

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

表 20-3. CTL 寄存器字段说明

位	字段	类型	复位	说明
31-2	RESERVED	R	0h	保留
1	CALC_EN	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
0	MEAS_EN	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

20.3.1.2 MEASCFG 寄存器（偏移 = 4h）[复位 = 00000000h]

图 20-2 展示了 MEASCFG，表 20-4 中对此进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 20-2. MEASCFG 寄存器

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														PER	
R-0h														R/W-0h	

表 20-4. MEASCFG 寄存器字段说明

位	字段	类型	复位	说明
31-2	RESERVED	R	0h	保留
1-0	PER	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

20.3.1.3 TEMPP0 寄存器（偏移 = Ch）[复位 = 00000000h]

图 20-3 展示了 TEMPP0，表 20-5 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 20-3. TEMPP0 寄存器

31	30	29	28	27	26	25	24	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																								CFG							
R-0h																								R/W-0h							

表 20-5. TEMPP0 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-0	CFG	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

20.3.1.4 TEMPP1 寄存器（偏移 = 10h）[复位 = 00000000h]

图 20-4 展示了 TEMPP1，表 20-6 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 20-4. TEMPP1 寄存器

31	30	29	28	27	26	25	24	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																										CFG					
R-0h																										R/W-0h					

表 20-6. TEMPP1 寄存器字段说明

位	字段	类型	复位	说明
31-6	RESERVED	R	0h	保留
5-0	CFG	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

20.3.1.5 TEMPP2 寄存器（偏移 = 14h）[复位 = 00000000h]

图 20-5 展示了 TEMPP2，表 20-7 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 20-5. TEMPP2 寄存器

31	30	29	28	27	26	25	24	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																											CFG				
R-0h																											R/W-0h				

表 20-7. TEMPP2 寄存器字段说明

位	字段	类型	复位	说明
31-5	RESERVED	R	0h	保留
4-0	CFG	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

20.3.1.6 BATMONP0 寄存器（偏移 = 18h）[复位 = 00000000h]

图 20-6 展示了 BATMONP0，表 20-8 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 20-6. BATMONP0 寄存器

31	30	29	28	27	26	25	24	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																									CFG						
R-0h																									R/W-0h						

表 20-8. BATMONP0 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	CFG	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

20.3.1.7 BATMONP1 寄存器（偏移 = 1Ch）[复位 = 00000000h]

图 20-7 展示了 BATMONP1，表 20-9 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 20-7. BATMONP1 寄存器

31	30	29	28	27	26	25	24	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																										CFG					
R-0h																										R/W-0h					

表 20-9. BATMONP1 寄存器字段说明

位	字段	类型	复位	说明
31-6	RESERVED	R	0h	保留
5-0	CFG	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

20.3.1.8 IOSTRP0 寄存器（偏移 = 20h）[复位 = 00000028h]

图 20-8 展示了 IOSTRP0，表 20-10 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 20-8. IOSTRP0 寄存器

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										CFG2		CFG1			
R-0h										R/W-2h		R/W-8h			

表 20-10. IOSTRP0 寄存器字段说明

位	字段	类型	复位	说明
31-6	RESERVED	R	0h	保留
5-4	CFG2	R/W	2h	内部。仅可通过 TI 提供的 API 使用。
3-0	CFG1	R/W	8h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

20.3.1.9 FLASHPUMPP0 寄存器（偏移 = 24h）[复位 = 00000000h]

图 20-9 展示了 FLASHPUMPP0，表 20-11 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 20-9. FLASHPUMPP0 寄存器

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

表 20-11. FLASHPUMPP0 寄存器字段说明

位	字段	类型	复位	说明
31-10	RESERVED	R	0h	保留
9	DIS_NOISE_FILTER	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
8	FALLB	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
7-6	HIGHLIM	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
5	LOWLIM	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
4	OVR	R/W	0h	内部。仅可通过 TI 提供的 API 使用。
3-0	CFG	R/W	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

20.3.1.10 BAT 寄存器（偏移 = 28h）[复位 = 00000000h]

图 20-10 展示了 BAT，表 20-12 中对此进行了介绍。

返回到汇总表。

最后测量的电池电压
当 BATUPD.STAT = 1 时，可以读取该寄存器

图 20-10. BAT 寄存器

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

表 20-12. BAT 寄存器字段说明

位	字段	类型	复位	说明
31-11	RESERVED	R	0h	保留
10-8	INT	R	0h	整数部分： 0x0：0V + 小数部分 ... 0x3：3V + 小数部分 0x4：4V + 小数部分
7-0	FRAC	R	0h	小数部分，标准二进制分数编码。 0x00：.0V ... 0x20：1/8 = .125V 0x40：1/4 = .25V 0x80：1/2 = .5V ... 0xA0：1/2 + 1/8 = .625V ... 0xFF：最大值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

20.3.1.11 BATUPD 寄存器（偏移 = 2Ch）[复位 = 00000000h]

图 20-11 展示了 BATUPD，表 20-13 中对此进行了介绍。

返回到汇总表。

电池更新
指示 BAT 更新

图 20-11. BATUPD 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							STAT
R-0h							R/W1C-0h

表 20-13. BATUPD 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	STAT	R/W1C	0h	0：自上次清除后无更新 1：存在新电池电压。 写入 1 以清除状态。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

20.3.1.12 TEMP 寄存器（偏移 = 30h）[复位 = 00000000h]

图 20-12 展示了 TEMP，表 20-14 中对此进行了介绍。

返回到汇总表。

温度
最后测量的温度（以摄氏度为单位）
当 TEMPUPD.STAT = 1 时，可以读取该寄存器。

图 20-12. TEMP 寄存器

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

表 20-14. TEMP 寄存器字段说明

位	字段	类型	复位	说明
31-17	RESERVED	R	0h	保留
16-8	INT	R	0h	温度值的整数部分（有符号）。 总值 = 整数部分 + 小数部分 二进制补码编码 0x100：最小值 0x1D8：-40C 0x1FF：-1C 0x00：0C 0x1B：27C 0x55：85C 0xFF：最大值
7-0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

20.3.1.13 TEMPUPD 寄存器（偏移 = 34h）[复位 = 00000000h]

图 20-13 展示了 TEMPUPD，表 20-15 中对此进行了介绍。

返回到汇总表。

温度更新
指示 TEMP 更新

图 20-13. TEMPUPD 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							STAT
R-0h							R/W1C-0h

表 20-15. TEMPUPD 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	STAT	R/W1C	0h	0：自上次清除后无更新 1：存在新温度。 写入 1 以清除状态。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

20.3.1.14 EVENTMASK 寄存器（偏移 = 48h）[复位 = 00000000h]

图 20-14 展示了 EVENTMASK，表 20-16 中对此进行了介绍。

返回到汇总表。

事件屏蔽

图 20-14. EVENTMASK 寄存器

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

表 20-16. EVENTMASK 寄存器字段说明

位	字段	类型	复位	说明
31-6	RESERVED	R	0h	保留
5	TEMP_UPDATE_MASK	R/W	0h	1：EVENT.TEMP_UPDATE 参与生成来自 BATMON 的组合事件 0：EVENT.TEMP_UPDATE 不参与生成来自 BATMON 的组合事件
4	BATT_UPDATE_MASK	R/W	0h	1：EVENT.BATT_UPDATE 参与生成来自 BATMON 的组合事件 0：EVENT.BATT_UPDATE 不参与生成来自 BATMON 的组合事件
3	TEMP_BELOW_LL_MASK	R/W	0h	1：EVENT.TEMP_BELOW_LL 参与生成来自 BATMON 的组合事件 0：EVENT.TEMP_BELOW_LL 不参与生成来自 BATMON 的组合事件
2	TEMP_OVER_UL_MASK	R/W	0h	1：EVENT.TEMP_OVER_UL 参与生成来自 BATMON 的组合事件 0：EVENT.TEMP_OVER_UL 不参与生成来自 BATMON 的组合事件
1	BATT_BELOW_LL_MASK	R/W	0h	1：EVENT.BATT_BELOW_LL 参与生成来自 BATMON 的组合事件 0：EVENT.BATT_BELOW_LL 不参与生成来自 BATMON 的组合事件
0	BATT_OVER_UL_MASK	R/W	0h	1：EVENT.BATT_OVER_UL 参与生成来自 BATMON 的组合事件 0：EVENT.BATT_OVER_UL 不参与生成来自 BATMON 的组合事件

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

20.3.1.15 EVENT 寄存器（偏移 = 4Ch）[复位 = 00000000h]

图 20-15 展示了 EVENT，表 20-17 中对此进行了介绍。

返回到汇总表。

事件

图 20-15. EVENT 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED		TEMP_UPDATE	BATT_UPDATE	TEMP_BELOW_LL	TEMP_OVER_UL	BATT_BELOW_LL	BATT_OVER_UL
R-0h		R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h

表 20-17. EVENT 寄存器字段说明

位	字段	类型	复位	说明
31-6	RESERVED	R	0h	保留
5	TEMP_UPDATE	R/W1C	0h	TEMPUPD.STAT 的别名
4	BATT_UPDATE	R/W1C	0h	BATUPD.STAT 的别名
3	TEMP_BELOW_LL	R/W1C	0h	读取： 1：温度水平低于 TEMPLL 设置的下限。 0：温度水平不低于 TEMPLL 设置的下限。 写入： 1：清除标志 0：标志无变化
2	TEMP_OVER_UL	R/W1C	0h	读取： 1：温度水平高于 TEMPUL 设置的上限。 0：温度水平不高于 TEMPUL 设置的上限。 写入： 1：清除标志 0：标志无变化
1	BATT_BELOW_LL	R/W1C	0h	读取： 1：电池电量低于 BATTLL 设置的下限。 0：电池电量不低于 BATTLL 设置的下限。 写入： 1：清除标志 0：标志无变化
0	BATT_OVER_UL	R/W1C	0h	读取： 1：电池电量高于 BATTUL 设置的上限。 0：电池电量不高于 BATTUL 设置的上限。 写入： 1：清除标志 0：标志无变化

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

20.3.1.16 BATTUL 寄存器（偏移 = 50h）[复位 = 000007FFh]

图 20-16 展示了 BATTUL，表 20-18 中对此进行了介绍。

返回到汇总表。

电池上限

图 20-16. BATTUL 寄存器

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

表 20-18. BATTUL 寄存器字段说明

位	字段	类型	复位	说明
31-11	RESERVED	R	0h	保留
10-8	INT	R/W	7h	整数部分： 0x0：0V + 小数部分 ... 0x3：3V + 小数部分 0x4：4V + 小数部分
7-0	FRAC	R/W	FFh	小数部分，标准二进制分数编码。 0x00：.0V ... 0x20：1/8 = .125V 0x40：1/4 = .25V 0x80：1/2 = .5V ... 0xA0：1/2 + 1/8 = .625V ... 0xFF：最大值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

20.3.1.17 BATTLL 寄存器（偏移 = 54h）[复位 = 00000000h]

图 20-17 展示了 BATTLL，表 20-19 中对此进行了介绍。

返回到汇总表。

电池下限

图 20-17. BATTLL 寄存器

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

表 20-19. BATTLL 寄存器字段说明

位	字段	类型	复位	说明
31-11	RESERVED	R	0h	保留
10-8	INT	R/W	0h	整数部分： 0x0：0V + 小数部分 ... 0x3：3V + 小数部分 0x4：4V + 小数部分
7-0	FRAC	R/W	0h	小数部分，标准二进制分数编码。 0x00：.0V ... 0x20：1/8 = .125V 0x40：1/4 = .25V 0x80：1/2 = .5V ... 0xA0：1/2 + 1/8 = .625V ... 0xFF：最大值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

20.3.1.18 TEMPUL 寄存器（偏移 = 58h）[复位 = 0000FFC0h]

图 20-18 展示了 TEMPUL，表 20-20 中对此进行了介绍。

返回到汇总表。

温度上限

图 20-18. TEMPUL 寄存器

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

表 20-20. TEMPUL 寄存器字段说明

位	字段	类型	复位	说明
31-17	RESERVED	R	0h	保留
16-8	INT	R/W	FFh	温度上限的整数部分（有符号）。 总值 = 整数部分 + 小数部分 二进制补码编码 0x100：最小值 0x1D8：-40C 0x1FF：-1C 0x00：0C 0x1B：27C 0x55：85C 0xFF：最大值
7-6	FRAC	R/W	3h	温度上限的小数部分。 总值 = 整数部分 + 小数部分 编码为二进制补码编码的扩展。 00：0.0C 01：0.25C 10：0.5C 11：0.75C 例如： 000000001,00 = (1+0,00) = 1,00 000000000,11 = (0+0,75) = 0,75 000000000,10 = (0+0,50) = 0,50 000000000,01 = (0+0,25) = 0,25 000000000,00 = (0+0,00) = 0,00 111111111,11 = (-1+0,75) = -0,25 111111111,10 = (-1+0,50) = -0,50 111111111,01 = (-1+0,25) = -0,75 111111111,00 = (-1+0,00) = -1,00 111111110,11 = (-2+0,75) = -1,25
5-0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

20.3.1.19 TEMPLL 寄存器（偏移 = 5Ch）[复位 = 00010000h]

图 20-19 展示了 TEMPLL，表 20-21 中对此进行了介绍。

返回到汇总表。

温度下限

图 20-19. TEMPLL 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															INT
R-0h															R/W-100h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT								FRAC		RESERVED					
R/W-100h								R/W-0h		R-0h					

表 20-21. TEMPLL 寄存器字段说明

位	字段	类型	复位	说明
31-17	RESERVED	R	0h	保留
16-8	INT	R/W	100h	温度下限的整数部分（有符号）。 总值 = 整数部分 + 小数部分 二进制补码编码 0x100：最小值 0x1D8：-40C 0x1FF：-1C 0x00：0C 0x1B：27C 0x55：85C 0xFF：最大值
7-6	FRAC	R/W	0h	温度下限的小数部分。 总值 = 整数部分 + 小数部分 编码为二进制补码编码的扩展。 00：0.0C 01：0.25C 10：0.5C 11：0.75C 例如： 000000001,00 = (1+0,00) = 1,00 000000000,11 = (0+0,75) = 0,75 000000000,10 = (0+0,50) = 0,50 000000000,01 = (0+0,25) = 0,25 000000000,00 = (0+0,00) = 0,00 111111111,11 = (-1+0,75) = -0,25 111111111,10 = (-1+0,50) = -0,50 111111111,01 = (-1+0,25) = -0,75 111111111,00 = (-1+0,00) = -1,00 111111110,11 = (-2+0,75) = -1,25
5-0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

章节 21

通用异步收发器 (UART)

本章介绍通用异步收发器 (UART)。

21.1 简介 1838
21.2 方框图 1839
21.3 信号描述 1839
21.4 功能说明 1839
21.5 用于连接 DMA 的接口 1844
21.6 初始化和配置 1845
21.7 UART 寄存器 1846

Copyright © 2026 Texas Instruments Incorporated

21.1 简介

CC13x2 和 CC26x2 器件平台的控制器包含一个具有以下特性的 UART：

- 可编程的波特率发生器，支持高达 3Mbps 的速度
- 独立的 32 × 8 发送 (TX) 和 32 × 12 接收 (RX) 先入先出 (FIFO) 缓冲器，可减少 CPU 中断服务负载
- FIFO 长度可编程，包括提供传统双缓冲接口的 1 字节深的操作
- FIFO 触发级别有 ⅛、¼、½、¾ 和 ⅞
- 标准的异步通讯位：起始位、停止位、奇偶校验位
- 线中止的产生与检测；
- 完全可编程的串行接口特性：
  - 可包含 5、6、7 或 8 个数据位
  - 偶校验、奇校验、固定校验或无奇偶校验位生成与检测
  - 可产生 1 或 2 个停止位
- 支持调制解调器控制功能 CTS 和 RTS
- 独立屏蔽 TX FIFO、RX FIFO RX 超时、调制解调器状态和错误条件
- 提供标准的基于 FIFO 深度的中断以及发送结束中断
- 用微型直接内存访问控制器 (µDMA) 进行的高效传输：
  - 相互独立的发送通道和接收通道
  - 当接收 FIFO 中有数据时产生单次请求；当接收 FIFO 到达预设的触发深度时产生猝发请求
  - 当发送 FIFO 中有空闲单元时产生单次请求；当发送 FIFO 到达预设的触发深度时产生猝发请求
- 可编程硬件流控制

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

21.2 方框图

图 21-1 显示了 UART 模块方框图。

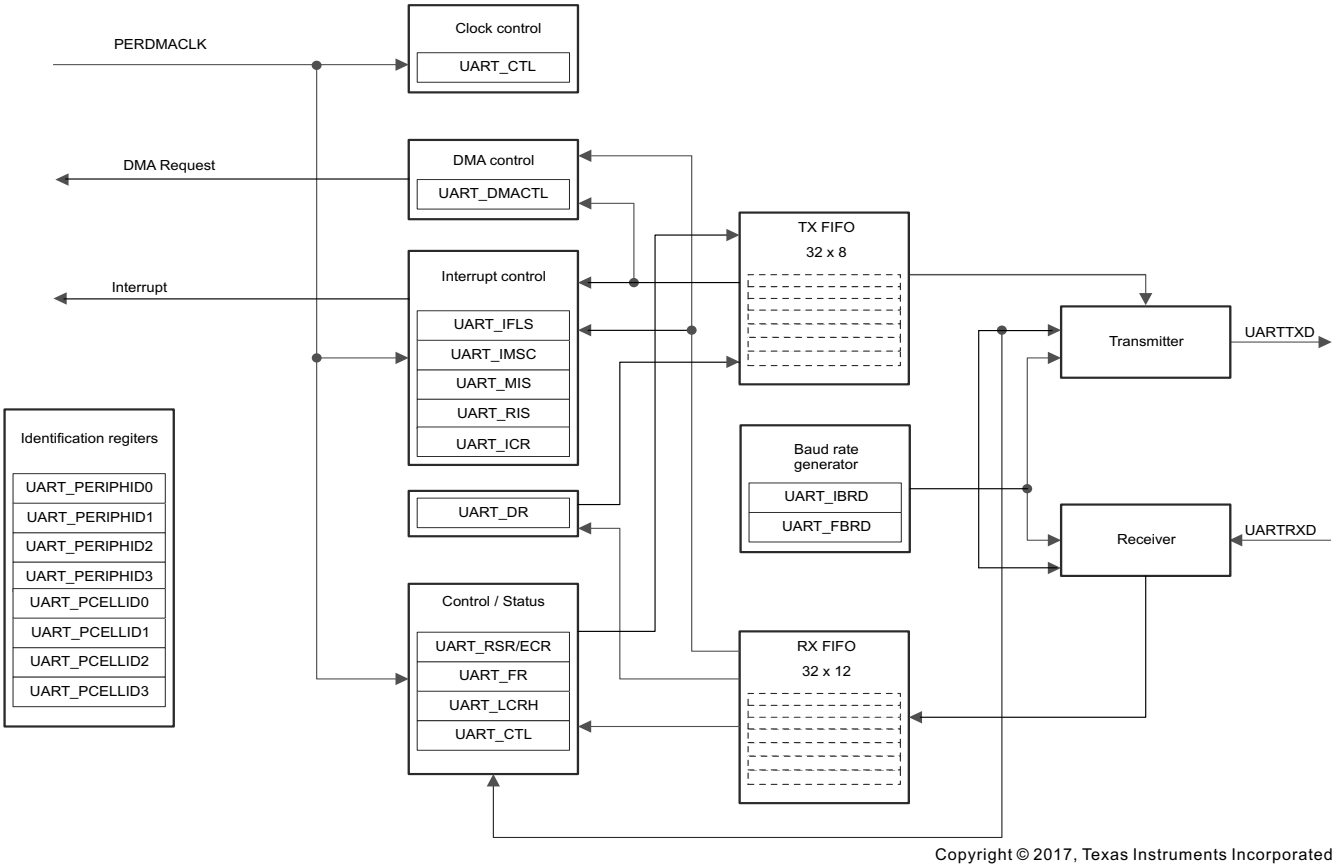


图 21-1. UART 模块方框图

21.3 信号描述

表 21-1 列出了 UART 模块的外部信号，并说明了每个模块的功能。UART 信号在 IOCFGn 寄存器中设置。有关配置 GPIO 的更多信息，请参阅章节 13。

表 21-1. UART 信号

引脚名称	引脚编号	引脚类型(1)	说明
UARTRxD	通过 GPIO 配置分配	I	UART 模块 0 接收
UARTTxD		O	UART 模块 0 发送

(1) I = 输入；O = 输出；I/O = 双向

21.4 功能说明

每个 CC13x2 和 CC26x2 UART 执行并行至串行和串行至并行转换的功能。CC13x2 和 CC26x2 UART 的功能与 16C550 UART 相似，但不兼容寄存器。

UART 通过 UART 控制寄存器 (UART:CTL) TXE 和 RXE 位配置为发送和接收。发送和接收在复位后均被使能。在对任何控制寄存器进行编程之前，必须通过清除 UART:CTL 寄存器的 UARTEN 位来禁用 UART。果在发送或接收操作期间禁用 UART，则当前事务会在 UART 停止之前完成。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

21.4.1 发送和接收逻辑

发送逻辑对从 TX FIFO 读取的数据执行并行至串行转换。控制逻辑输出串行位流时，最先输出起始位，然后按照控制寄存器中编程的配置依次输出若干数据位（LSB 在前）、奇偶校验位和停止位。相关详细信息，请参阅图 21-2。

接收逻辑在检测到有效的起始脉冲后，对接收到的位流执行串行至并行转换。此外还会对溢出错误、奇偶校验错误、帧错误和线路中断错误进行检测，并将检测到的状态附加到被写入 RX FIFO 的数据中。

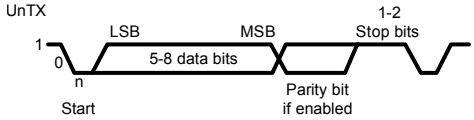


图 21-2. UART 字符帧

21.4.2 波特率生成

波特率除数 (BRD) 是一个 22 位数字，由 16 位整数和 6 位小数部分组成。波特率发生器使用这两个值形成的数字来确定位周期。分数波特率分频器可让 UART 生成所有标准波特率。

16 位整数通过 UART 整数波特率分频器寄存器 (UART:IBRD) 进行加载，而 6 位小数则通过 UART 分数波特率分频器寄存器 (UART:FBRD) 进行加载。方程式 4 显示了 BRD 与系统时钟的关系。

$$BRD = BRDI + BRDF = PERDMACLK / (ClkDiv \times Baud\ Rate) \quad (4)$$

其中：

- BRDI 是 BRD 的整数部分
- BRDF 是小数部分，用小数点分隔
- PERDMACLK 是连接到 UART 的系统时钟
- ClkDiv 是 16

加载到 UART:FBRD DIVFRAC 位字段的 6 位小数可以通过以下方法计算得出：取波特率除数的小数部分，乘以 64，然后加上 0.5 以解决舍入误差，如方程式 5 所示。

$$UART:FBRD.DIVFRAC = integer\ (BRDF \times 64 + 0.5) \quad (5)$$

UART:IBRD 和 UART:FBRD 寄存器与 UART 线路控制高字节寄存器 (UART:LCRH) 一起构成一个内部的 30 位寄存器。这个内部寄存器仅在执行对 UART:LCRH 寄存器的写操作时才会更新，因此，任何对 BRD 的更改都必须随后写入 UART:LCRH 寄存器，更改才能生效。

更新波特率寄存器的四种可能顺序如下：

- 写入 UART:IBRD，写入 UART:FBRD，然后写入 UART:LCRH
- 写入 UART:FBRD，写入 UART:IBRD，然后写入 UART:LCRH
- 写入 UART:IBRD，然后写入 UART:LCRH
- 写入 UART:FBRD，然后写入 UART:LCRH

21.4.3 数据传输

接收或发送的数据存储在两个 FIFO 中，但是 RX FIFO 的每个字符都有额外的 4 位用于指示状态。当需要进行发送时，先将数据写入 TX FIFO。如果启用 UART，则数据帧会根据 UART:LCRH 寄存器中指示的参数开始发送。数据持续发送，直到 TX FIFO 中没有剩余数据为止。数据一经写入 TX FIFO（即，如果 FIFO 不为空），UART 标志寄存器 (UART:FR) 的 BUSY 位即会生效，并在数据发送期间一直保持有效。仅当 TX FIFO 为空，且最后一个字符（包括停止位）已从移位寄存器发送时，BUSY 位才为负。即使 UART 可能已不再使能，UART 也可以指示它正忙。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

当接收器空闲时（UARTRXD 信号持续为 1），并且数据输入变为低电平（接收到起始位）时，接收计数器开始运行并对数据进行采样。

如果在波特率时钟的第八个周期时 UARTRXD 信号仍为低电平，则起始位有效并被识别，否则起始位被忽略。检测到有效的起始位后，后续数据位在波特率时钟的每第十六个周期进行采样。之后将捕获并校验奇偶校验位（如果使能了奇偶校验）。数据长度和奇偶校验都在 UART:LCRH 寄存器中定义。

最后，如果 UARTRXD 信号为高电平，则确认有效停止位，否则发生成帧错误。若成功接收到一帧数据，则数据连同与之相关的错误标志都将保存到接收 FIFO 中。

21.4.4 调制解调器握手支持

本节介绍 UART0 在作为数据终端设备 (DTE) 或数据通信设备 (DCE) 连接时，如何配置和使用调制解调器流控制信号。调制解调器是 DCE，而连接到调制解调器的计算设备是 DTE。

21.4.4.1 信号

UART0 提供的状态信号根据 UART 是用作 DTE 还是 DCE 而有所不同。当用作 DTE 时，调制解调器流控制信号定义如下：

- **UART0CTS** 是允许发送
- **UART0RTS** 是请求发送

当用作 DCE 时，调制解调器流控制信号定义如下：

- **UART0CTS** 是请求发送
- **UART0RTS** 是允许发送

21.4.4.2 流控

硬件或软件都可以实现流控制。以下部分介绍了几种不同的方法。

21.4.4.2.1 硬件流控制（RTS 和 CTS）

通过将 UART0RTS 输出连接到接收器件上的“允许发送”输入，并将接收器件上的“请求发送”输出连接到 UART0CTS 输入来完成两个器件之间的硬件流量控制。

UART0CTS 输入控制发送器。仅当 UART0CTS 输入被置为有效时，发送器才能发送数据。UART0RTS 输出信号指示接收 FIFO 的状态。在达到预编程的水线位之前，UART0CTS 保持有效，表明 RX FIFO 没有空间来存储额外的字符。

UART:CTL 寄存器的 CTSEN 和 RTSEN 位指定流量控制模式，如表 21-2 所示。

表 21-2. 流控制模式

CTSEN	RTSEN	说明
1	1	启用 RTS 与 CTS 流量控制
1	0	仅启用 CTS 流量控制
0	1	仅启用 RTS 流量控制
0	0	禁用 RTS 和 CTS 流量控制

21.4.4.2.2 软件流控制（调制解调器状态中断）

两个器件之间的软件流控制通过使用中断指示 UART 的状态来实现。可以使用 UART:IMSC 寄存器的第 3 位为 U1CTS 信号生成中断。可以使用 UART:RIS 和 UART:MIS 寄存器检查原始中断状态和屏蔽后的中断状态。这些中断可以使用 UART:ICR 寄存器进行清除。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

21.4.5 FIFO 操作

UART 有两个 32 个条目的 FIFO，一个用于发送，一个用于接收。两个 FIFO 都通过 UART 数据寄存器 (UART:DR) 进行访问。UART:DR 寄存器的读取操作返回一个 12 位值，该值由 8 个数据位和 4 个错误标志组成，而写入操作将 8 位数据放置在 TX FIFO 中。

复位后，两个 FIFO 都被禁用，并充当 1 字节深的保持寄存器。可通过设置 UART:LCRH FEN 寄存器位来使能 FIFO。

可以通过 UART 标志寄存器 (UART:FR) 和 UART 接收状态寄存器 (UART:RSR) 监测 FIFO 状态。硬件监测空、满和超限情况。UART:FR 寄存器包含空和满标志（TXFE、TXFF、RXFE 和 RXFF 位），而 UART:RSR 寄存器通过 OE 位显示超限状态。如果禁用 FIFO，则根据 1 字节深保持寄存器的状态来设置空和满标志。

FIFO 生成中断的触发点通过 UART 中断 FIFO 电平选择寄存器 (UART:IFLS) 进行控制。两个 FIFO 可以分别配置为在不同电平触发中断。可用的配置包括 ⅛、¼、½、¾, 和 ⅞。例如，如果为接收 FIFO 选择了 ¼ 选项，则 UART 在接收了 4 个数据字节之后会生成接收中断。复位结束后，两个 FIFO 均配置为在 ½ 标记处触发中断。

21.4.6 中断

在观察到以下条件时，UART 可以生成中断：

- 超限错误
- 中断错误
- 奇偶校验错误
- 组帧错误
- 接收超时
- 发送（当满足 UART:IFLS TXSEL 寄存器位中定义的条件时）
- 接收（当满足 UART:IFLS RXSEL 寄存器位中定义的条件时）
- 传输结束（当 TX 线路上没有数据且 TX FIFO 下溢时）

所有中断事件在发送到中断控制器之前都一起进行“或”运算，因此 UART 在任何给定的时间只能向控制器生成一个中断请求。软件可以通过读取 UART 屏蔽中断状态寄存器 (UART:MIS) 在单个中断服务例程 (ISR) 中处理多个中断事件。

可以触发控制器级中断的中断事件通过在 UART 中断屏蔽寄存器 (UART:IMSC) 中设置相应的位来定义。如果不使用中断，原始中断状态始终可以通过 UART 原始中断状态寄存器 (UART:RIS) 看到。

通过设置 UART 中断清除寄存器 (UART:ICR) 中的相应位，可以清除中断（对于 UART:MIS 和 UART:RIS 寄存器）。

当 RX FIFO 不为空且在 32 位周期内没有接收到更多数据时，此接收超时中断有效。当 FIFO 通过读取所有数据（或通过读取保持寄存器）变为空，或者设置 UART:ICR 寄存器中的相应位时，接收超时中断将被清除。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

UART 模块支持使用 UART 中断屏蔽设置/清除寄存器 (UART:IMSC) 为每个单独的中断源设置和清除屏蔽。可能导致组合中断发送到 CPU 的五个事件为：

- RX：发生以下事件之一时，接收中断将更改状态：
  - 如果启用 FIFO 且接收 FIFO 到达编程的触发电平。当发生这种情况时，接收中断会置为高电平。通过从接收 FIFO 读取数据直至其低于触发电平，或通过清除中断来清除接收中断。
  - 如果 FIFO 禁用（深度为一个位置）并且接收到数据从而填充该位置，接收中断会置为高电平。通过对接收 FIFO 执行单次读取，或通过清除中断来清除接收中断。
- TX：发生以下事件之一时，发送中断将更改状态：
  - 如果 FIFO 使能且发送 FIFO 等于或低于编程的触发电平，则发送中断会置为高电平。通过向发送 FIFO 写入数据直至其高于触发电平，或通过清除中断来清除发送中断。
  - 如果 FIFO 禁用（深度为一个位置）并且发送器单个位置中没有数据，则发送中断会置为高电平。通过对发送 FIFO 执行单次写入，或通过清除中断来清除中断。
- RX 超时：当接收 FIFO 不为空且在 32 位周期内没有接收到更多数据时，此接收超时中断有效。当 FIFO 通过读取所有数据（或通过读取保持寄存器）变为空，或者中断清除寄存器 (UART:ICR) 中的相应位写入 1 时，接收超时中断将被清除。
- 调制解调器状态：如果调制解调器状态信号 *uart_cts* 发生变化，则调制解调器状态中断置为有效。可以使用 UART:ICR 寄存器中的相应清除位来清除它。
- 错误：当 UART 在接收数据过程中出现错误时，错误中断置为有效。中断可能由多个不同的错误条件引起：
  - 帧
  - 奇偶校验
  - 中断
  - 超限

  可以通过读取 UART:RIS 寄存器或 UART:MIS 寄存器来确定中断原因。可以通过写入 UART:ICR 寄存器的相关位来清除中断。

除了 UART 模块产生的五个事件外，还有两个额外的事件会被或运算到中断线路中：

- RX DMA 完成：表示接收器 DMA 已完成其任务。这是由 DMA 模块提供的电平中断，使用 DMA 模块中的 dma_done 清除寄存器 (UDMA:REQDONE) 清除。
- TX DMA 完成：表示发送 DMA 已完成其任务。这是由 DMA 模块提供的电平中断，使用 DMA 模块中的 dma_done 清除寄存器 (UDMA:REQDONE) 清除。

21.4.7 环回操作

通过设置 UART:CTL LBE 寄存器位，可以将 UART 置于内部环回模式，以开展诊断或调试工作。在环回模式下，从 UARTTXD 输出端发送的数据会在 UARTRXD 输入端收到。LBE 位必须在 UART 使能之前设置。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

21.5 用于连接 DMA 的接口

CC13x2 和 CC26x2 器件提供了一个连接到 DMA 控制器的接口。图 21-3 显示了 DMA 和 UART 之间的接口。该接口包含四个作为输出的 DMA 请求（RX 单次、RX 突发、TX 单次和 TX 突发）。DMA 接口还有两个作为输入的 DMA 请求清除信号（用于清除 TX 和 RX DMA 请求）。每个 DMA 请求信号保持有效状态，直到相关的 DMA 清除信号置为有效。在 DMA 清除信号置为无效后，如果条件设置正确，请求信号可以再次变为有效。DMA 清除信号必须连接到来自 DMA 模块的 DMA 有效信号。当 DMA 被授予访问权限并处于活动状态时，该信号被置为有效。当 DMA 传输完成时，DMA 有效信号被置为无效。将来自 DMA 的 DMA 有效信号连接到 UART 模块的 DMA 请求清除输入，可确保在 DMA 处于活动状态时，UART 模块不会生成任何请求。

突发传输和单次传输请求信号并不互斥，两者可以同时被置为有效。例如，当接收 FIFO 中的数据多于水线位时，突发传输请求和单次传输请求都会被置为有效。

UART 模块不能单独屏蔽单次和突发请求，如果启用了相应的 DMA（RX 或 TX），这两个请求都会被发送到 DMA。DMA 配置选择单次或突发请求作为触发器。如果 UART 被禁用，或者 DMA 控制寄存器 (UART:DMACTL) 中的相关 DMA 使能位（TXDMAE 或 RXDMAE）被清除，则所有请求信号都将被置为无效。

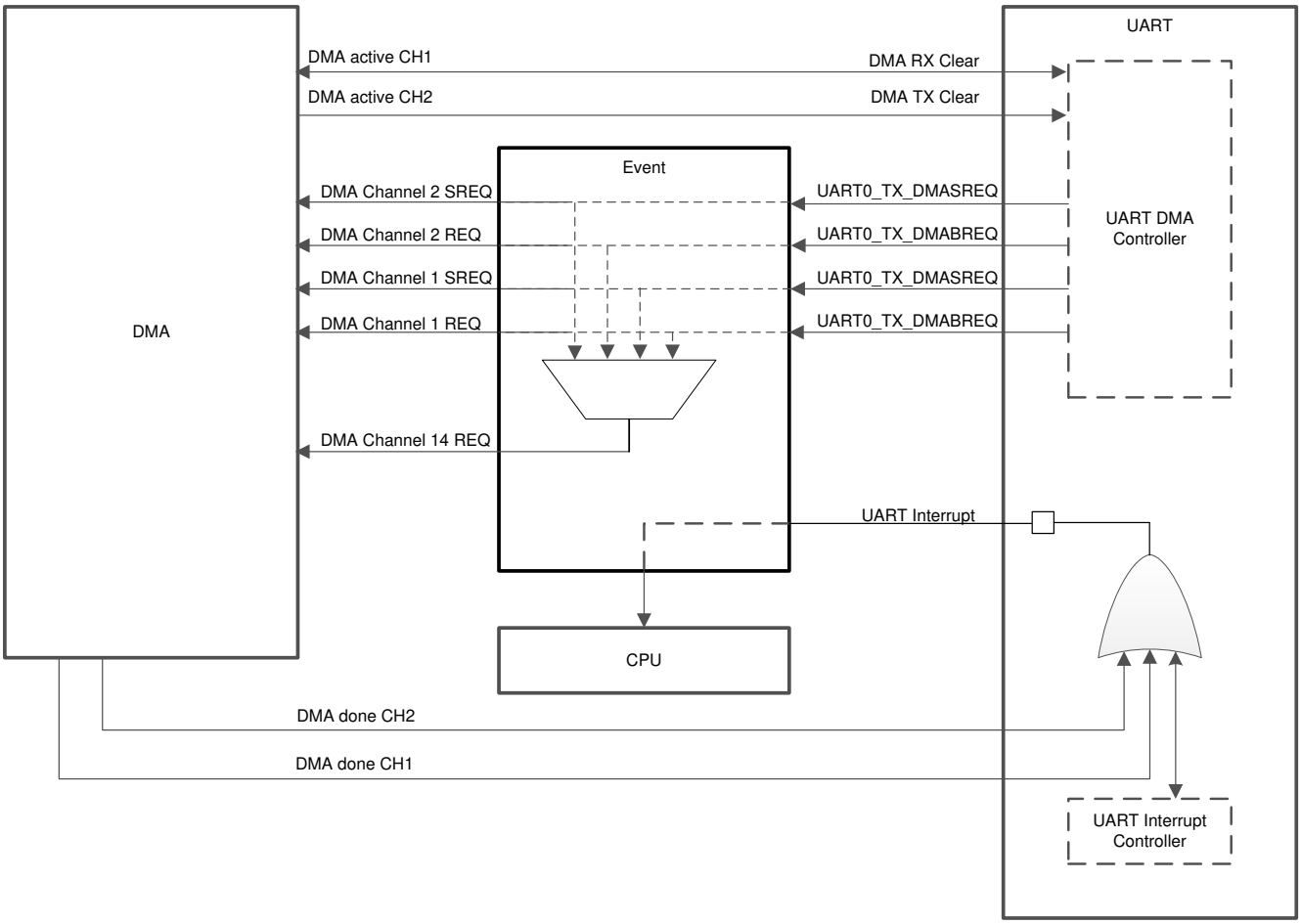


图 21-3. µDMA 示例

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

21.6 初始化和配置

UART 模块提供四个 I/O 信号，用于路由到 DIO。通过 IOC 模块中的 IOCFGn 寄存器选择以下信号。

- 输入：RXD、CTS
- 输出：TXD、RTS

CTS 和 RTS 线路为低电平有效。

备注

在启用 UART 之前必须配置 IOC，否则输入信号上的不必要转换可能会干扰 UART 对传入事务的处理。当 IOC 配置为 UART 专用 I/O（RXD、CTS、TXD 或 RTS）时，IOC 会向 DIO 设置静态输出驱动器使能（对于输出 TXD 和 RTS，输出驱动器使能 = 1；对于输入 RXD 和 CTS，输出驱动器使能 = 0）。

要启用和初始化 UART，请执行以下步骤：

1. 通过以下方式来启用串行电源域并在 PRCM 模块中启用 UART 模块：写入 PRCM:UARTCLKGR 寄存器、PRCM:UARTCLKGS 寄存器和 PRCM:UARTCLKGDS 寄存器，或使用驱动库函数：

   ```
   PRCMPeripheralRunEnable(uint32_t), PRCMPeripheralSleepEnable(uint32_t),
   PRCMPeripheralDeepSleepEnable(unit32_t)
   ```

   ，并通过写入 PRCM:CLKLOADCTL 寄存器或使用以下函数将设置加载到时钟控制器

   ```
   PRCMLoadSet().
   ```

2. 配置 IOC 模块，将 UART 信号映射到正确的 GPIO 引脚。有关引脚连接的更多信息，请参阅章节 13。

本节讨论使用 UART 模块所需的步骤。在本例中，假设 UART 时钟为 24MHz，并且期望的 UART 配置如下：

- 波特率：115 200
- 数据长度：8 位
- 1 个停止位
- 无奇偶校验位
- 禁用 FIFO
- No interrupts

对 UART 进行编程时首先要考虑的是 BRD，因为必须在写入 UART:LCRH 寄存器之前写入 UART:IBRD 和 UART:FBRD 寄存器。可以使用节 21.4.2 中所述的公式计算 BRD。

$$\text{BRD} = 24\,000\,000 / (16 \times 115\,200) = 13.0208 \quad (6)$$

方程式 6 的结果表明 UART:IBRD DIVINT 字段必须设置为十进制 13 或 0xD。方程式 7 计算要加载到 UART:FBRD 寄存器中的值。

$$\text{UART:FBRD.DIVFRAC} = \text{integer}\,(0.0208 \times 64 + 0.5) = 1 \quad (7)$$

获得 BRD 值后，按以下顺序将 UART 配置写入模块：

1. 通过清除 UART:CTL UARTEN 位来禁用 UART。
2. 将 BRD 的整数部分写入 UART:IBRD 寄存器。
3. 将 BRD 的小数部分写入 UART:FBRD 寄存器。
4. 将所需的串行参数写入 UART:LCRH 寄存器（在本例中，值为 0x0000 0060）。
5. 通过设置 UART:CTL UARTEN 位来启用 UART。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

21.7 UART 寄存器

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

21.7.1 UART 寄存器

表 21-3 列出了 UART 寄存器的存储器映射寄存器。表 21-3 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

表 21-3. UART 寄存器

偏移	首字母缩写词	寄存器名称	部分
0h	DR	数据	DR 寄存器 (偏移 = 0h) [复位 = 00000000h]
4h	RSR	状态	RSR 寄存器 (偏移 = 4h) [复位 = 00000000h]
4h	ECR	错误清除	ECR 寄存器 (偏移 = 4h) [复位 = 00000000h]
18h	FR	标志	FR 寄存器 (偏移 = 18h) [复位 = 00000090h]
24h	IBRD	整数波特率分频	IBRD 寄存器 (偏移 = 24h) [复位 = 00000000h]
28h	FBRD	分数波特率分频	FBRD 寄存器 (偏移 = 28h) [复位 = 00000000h]
2Ch	LCRH	线路控制	LCRH 寄存器 (偏移 = 2Ch) [复位 = 00000000h]
30h	CTL	控制	CTL 寄存器 (偏移 = 30h) [复位 = 00000300h]
34h	IFLS	中断 FIFO 级别选择	IFLS 寄存器 (偏移 = 34h) [复位 = 00000012h]
38h	IMSC	中断掩码设置/清除	IMSC 寄存器 (偏移 = 38h) [复位 = 00000000h]
3Ch	RIS	原始中断状态	RIS 寄存器 (偏移 = 3Ch) [复位 = 0000000Dh]
40h	MIS	已屏蔽中断状态	MIS 寄存器 (偏移 = 40h) [复位 = 00000000h]
44h	ICR	中断清除	ICR 寄存器 (偏移 = 44h) [复位 = 00000000h]
48h	DMACTL	DMA 控制	DMACTL 寄存器 (偏移 = 48h) [复位 = 00000000h]

复杂的位访问类型经过编码可适应小型表单元。表 21-4 展示了适用于此部分中访问类型的代码。

表 21-4. UART 访问类型代码

访问类型	代码	说明
读取类型		
R	R	读取
写入类型		
W	W	写入
复位或默认值		
-n		复位后的值或默认值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

21.7.1.1 DR 寄存器（偏移 = 0h）[复位 = 00000000h]

图 21-4 展示了 DR，表 21-5 中对此进行了介绍。

返回到汇总表。

数据
针对要传输的字：
- 如果 FIFO 已使能 (LCRH.FEN = 1)，则写入该位置的数据将被推入传输 FIFO
- 如果 FIFO 未使能 (LCRH.FEN=0)，数据将存储在传输器保持寄存器中（传输 FIFO 的底部字）。
该写入操作会启动来自 UART 的传输。数据的前缀为一个开始位，附加适当的奇偶校验位（如果奇偶校验已使能）和一个停止位。
然后，传输结果字。
对于接收的字：
- 如果 FIFO 已使能 (LCRH.FEN = 1)，数据字节和 4 位状态（中断、帧、奇偶校验和超限）会被推入 12 位宽的接收 FIFO
- 如果 FIFO 未使能 (LCRH.FEN = 0)，数据字节和状态将存储在接收保持寄存器中（接收 FIFO 的底部字）。
通过对该寄存器执行读取操作，并结合相应的状态信息来读取接收到的数据字节。也可以通过读取 RSR 寄存器来读取状态信息。

图 21-4. DR 寄存器

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				OE	BE	PE	FE	DATA							
R-0h				R-X	R-X	R-X	R-X	R/W-X							

表 21-5. DR 寄存器字段说明

位	字段	类型	复位	说明
31-12	RESERVED	R	0h	保留
11	OE	R	X	UART 超限错误： 如果接收到数据且接收 FIFO 已满，则该位设置为 1。FIFO 内容保持有效，因为当 FIFO 已满时，不会再写入任何数据，只有移位寄存器的内容会被覆盖。 一旦 FIFO 中有空闲空间并且可以写入新字符，该位就会清除为 0。
10	BE	R	X	UART 中断错误： 如果检测到中断条件，则该位设置为 1，表示接收到的数据输入（UARTRXD 输入引脚）保持低电平的时间超过一个完整字传输时间（定义为开始位，数据位、奇偶校验位和停止位）。 在 FIFO 模式下，该错误与 FIFO 顶部的字符（即，自上次读取以来最早接收的数据字符）有关。发生中断时，一个 0 字符会被加载到 FIFO 中。仅在接收数据输入变为 1（标记状态）且接收到下一个有效的起始位后，才启用下一字符。
9	PE	R	X	UART 奇偶校验错误： 设置为 1 时，表示接收到的数据字符的奇偶校验与 LCRH.EPS 和 LCRH.SPS 选择的奇偶校验不匹配。 在 FIFO 模式下，该错误与 FIFO 顶部的字符（即自上次读取以来最早接收的数据字符）有关。
8	FE	R	X	UART 组帧错误： 设置为 1 时，表示接收到的字符没有有效的停止位（有效的停止位为 1）。 在 FIFO 模式下，该错误与 FIFO 顶部的字符（即，自上次读取以来最早接收到的数据字符）有关。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 21-5. DR 寄存器字段说明（续）

位	字段	类型	复位	说明
7-0	DATA	R/W	X	已传输或已接收数据： 写入时，传输数据字符被推入 FIFO 中。 读取时，会返回自上次读取以来最早接收到的数据字符。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

21.7.1.2 RSR 寄存器（偏移 = 4h）[复位 = 00000000h]

图 21-5 展示了 RSR，表 21-6 中对此进行了介绍。

返回到汇总表。

状态
此寄存器映射到与 ECR 寄存器相同的地址。从该地址读取时关联的是 RSR 寄存器，并返回接收状态。对该地址的写入与 ECR 寄存器相关联并会清除接收状态标志（组帧、奇偶校验、间断和超限错误）。
如果从该寄存器读取状态，则间断、组帧和奇偶校验的状态信息与读取 RSR 之前从数据寄存器 DR 中读取的数据字符相对应。发生超限条件时，会立即设置超限状态信息。

图 21-5. RSR 寄存器

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												OE	BE	PE	FE
R-0h												R-0h	R-0h	R-0h	R-0h

表 21-6. RSR 寄存器字段说明

位	字段	类型	复位	说明
31-4	RESERVED	R	0h	保留
3	OE	R	0h	UART 超限错误： 如果接收到数据且接收 FIFO 已满，则该位设置为 1。FIFO 内容保持有效，因为当 FIFO 已满时，不会再写入任何数据，只有移位寄存器的内容会被覆盖。 一旦 FIFO 中有空闲空间并且可以写入新字符，该位就会清除为 0。
2	BE	R	0h	UART 中断错误： 如果检测到中断条件，则该位设置为 1，表示接收到的数据输入（UARTRXD 输入引脚）保持低电平的时间超过一个完整字传输时间（定义为开始位，数据位，奇偶校验位和停止位）。 发生中断时，一个 0 字符会被加载到 FIFO 中。仅在接收数据输入变为 1（标记状态）且接收到下一个有效的起始位后，才启用下一字符。
1	PE	R	0h	UART 奇偶校验错误： 设置为 1 时，表示所接收数据字符的奇偶校验与 LCRH.EPS 和 LCRH.SPS 选择的奇偶校验不匹配。
0	FE	R	0h	UART 组帧错误： 设置为 1 时，表示接收到的字符没有有效的停止位（有效的停止位为 1）。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

21.7.1.3 ECR 寄存器（偏移 = 4h）[复位 = 00000000h]

图 21-6 展示了 ECR，表 21-7 中对此进行了介绍。

返回到汇总表。

错误清除
该寄存器映射到与 RSR 寄存器相同的地址。从该地址读取时关联的是 RSR 寄存器，并返回接收状态。对该地址的写入将与 ECR 寄存器关联并清除接收状态标志（组帧、奇偶校验、中断和超限错误）。

图 21-6. ECR 寄存器

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												OE	BE	PE	FE
R-0h												W-0h	W-0h	W-0h	W-0h

表 21-7. ECR 寄存器字段说明

位	字段	类型	复位	说明
31-4	RESERVED	R	0h	保留
3	OE	W	0h	对该寄存器进行任何写入都会将组帧 (FE)、奇偶校验 (PE)、中断 (BE) 和超限 (OE) 错误清除为 0。
2	BE	W	0h	对该寄存器进行任何写入都会将组帧 (FE)、奇偶校验 (PE)、中断 (BE) 和超限 (OE) 错误清除为 0。
1	PE	W	0h	对该寄存器进行任何写入都会将组帧 (FE)、奇偶校验 (PE)、中断 (BE) 和超限 (OE) 错误清除为 0。
0	FE	W	0h	对该寄存器进行任何写入都会将组帧 (FE)、奇偶校验 (PE)、中断 (BE) 和超限 (OE) 错误清除为 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

21.7.1.4 FR 寄存器（偏移 = 18h）[复位 = 00000090h]

图 21-7 展示了 FR，表 21-8 中对此进行了介绍。

返回到汇总表。

标志
对该寄存器执行读取操作会返回 UART 标志。

图 21-7. FR 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
TXFE	RXFF	TXFF	RXFE	BUSY	RESERVED		CTS
R-1h	R-0h	R-0h	R-1h	R-0h	R-0h		R-X

表 21-8. FR 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7	TXFE	R	1h	UART 传输 FIFO 为空： 该位的含义取决于 LCRH.FEN 寄存器的状态。 - 如果 FIFO 已禁用，则当传输保持寄存器为空时，该位被置位。 - 如果 FIFO 已使能，则当传输 FIFO 为空时，该位被置位。 该位并不指示传输移位寄存器中是否有数据。
6	RXFF	R	0h	UART 接收 FIFO 已满： 此位的含义取决于 LCRH.FEN 的状态。 - 如果 FIFO 已禁用，则当接收保持寄存器为满时，该位被置位。 - 如果 FIFO 已使能，则当接收 FIFO 为满时，该位被置位。
5	TXFF	R	0h	UART 传输 FIFO 已满： 传输 FIFO 已满。此位的含义取决于 LCRH.FEN 的状态。 - 如果 FIFO 已禁用，则当传输保持寄存器为满时，该位被置位。 - 如果 FIFO 已使能，则当传输 FIFO 为满时，该位被置位。
4	RXFE	R	1h	UART 接收 FIFO 为空： 接收 FIFO 为空。此位的含义取决于 LCRH.FEN 的状态。 - 如果 FIFO 已禁用，则当接收保持寄存器为空时，该位被置位。 - 如果 FIFO 已使能，则当接收 FIFO 为空时，该位被置位。
3	BUSY	R	0h	UART 忙： 如果该位设置为 1，则 UART 正忙于传输数据。该位保持置位状态，直到包含所有停止位在内的完整字节已从移位寄存器发送出去。 只要传输 FIFO 变为非空，该位就会被置位，而不论 UART 是否已使能。
2-1	RESERVED	R	0h	保留
0	CTS	R	X	允许发送： 该位是低电平有效 UART CTS 输入引脚的补码。 也就是说，当 CTS 输入引脚为低电平时，该位为 1。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

21.7.1.5 IBRD 寄存器（偏移 = 24h）[复位 = 00000000h]

图 21-8 展示了 IBRD，表 21-9 中对此进行了介绍。

返回到汇总表。

整数波特率分频
如果在传输或接收过程中修改该寄存器，则波特率不会立即更新，而是会等到当前字符传输或接收完成后才生效。

图 21-8. IBRD 寄存器

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

表 21-9. IBRD 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	DIVINT	R/W	0h	整数波特率分频： 此波特率分频系数的计算公式如下： 波特率分频系数 = (UART 参考时钟频率) / (16 * 波特率) 波特率分频系数必须最小值为 1，最大值为 65535。 也就是说，DIVINT=0 不提供有效的波特率。 同样，如果 DIVINT=0xFFFF，则 FBRD.DIVFRAC 中的任何非零值都是非法的。 必须向该字段写入一个有效值，才能将 UART 用于 RX 或 TX 操作。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

21.7.1.6 FBRD 寄存器（偏移 = 28h）[复位 = 00000000h]

图 21-9 展示了 FBRD，表 21-10 中对此进行了介绍。

返回到汇总表。

分数波特率分频
如果在传输或接收过程中修改该寄存器，则波特率不会立即更新，而是会等到当前字符传输或接收完成后才生效。

图 21-9. FBRD 寄存器

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

表 21-10. FBRD 寄存器字段说明

位	字段	类型	复位	说明
31-6	RESERVED	R	0h	保留
5-0	DIVFRAC	R/W	0h	分数波特率分频： 此波特率分频系数的计算公式如下： 波特率分频系数 = (UART 参考时钟频率) / (16 * 波特率) 波特率分频系数必须最小值为 1，最大值为 65535。 也就是说，IBRD.DIVINT=0 不提供有效的波特率。 同样，如果 IBRD.DIVINT=0xFFFF，则 DIVFRAC 中的任何非零值都是非法的。 必须向该字段写入一个有效值，才能将 UART 用于 RX 或 TX 操作。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

21.7.1.7 LCRH 寄存器（偏移 = 2Ch）[复位 = 00000000h]

图 21-10 展示了 LCRH，表 21-11 中对此进行了介绍。

返回到汇总表。

线路控制

图 21-10. LCRH 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
SPS	WLEN		FEN	STP2	EPS	PEN	BRK
R/W-0h	R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 21-11. LCRH 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7	SPS	R/W	0h	UART 固定奇偶校验位选择： 0：禁用固定奇偶校验位 1：奇偶校验位的传输和检查采用 EPS 字段的取反值（即当 EPS = 0 时，奇偶校验位作为 1 传输和检查）。 当 PEN 禁用奇偶校验检查和生成时，该位无效。
6-5	WLEN	R/W	0h	UART 字长： 这些位指示帧中传输或接收的数据位数。 0h = 5：字长为 5 位 1h = 6：字长为 6 位 2h = 7：字长为 7 位 3h = 8：字长为 8 位
4	FEN	R/W	0h	UART 使能 FIFO 0h = 禁用 FIFO（字符模式），即 FIFO 变为 1 字节深保持寄存器。 1h = 启用传输和接收 FIFO 缓冲器（FIFO 模式）
3	STP2	R/W	0h	UART 两个停止位选择： 如果该位设置为 1，则在帧末尾传输两个停止位。接收逻辑不会检查是否在接收两个停止位。
2	EPS	R/W	0h	UART 偶校验选择 0h = 奇校验：UART 生成或检查数据和校验位中 1 的个数是否为奇数。 1h = 偶校验：UART 生成或检查数据和校验位中 1 的个数是否为偶数。
1	PEN	R/W	0h	UART 奇偶校验使能 该位控制奇偶校验位的生成和检查。 0h = 禁用奇偶校验且不向数据帧添加奇偶校验位 1h = 启用奇偶校验检查和生成。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 21-11. LCRH 寄存器字段说明（续）

位	字段	类型	复位	说明
0	BRK	R/W	0h	UART 发送中断 如果该位设置为 1，在完成当前字符的传输后，UARTTXD 输出引脚上会持续输出低电平。为了正确执行中断命令， 软件必须令该位处于置位状态，并且持续至少 2 个完整帧。为确保正常使用，该位必须清零。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

21.7.1.8 CTL 寄存器（偏移 = 30h）[复位 = 00000300h]

图 21-11 展示了 CTL，表 21-12 中对此进行了介绍。

返回到汇总表。

控制

图 21-11. CTL 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
CTSEN	RTSEN	RESERVED		RTS	RESERVED	RXE	TXE
R/W-0h	R/W-0h	R-0h		R/W-0h	R-0h	R/W-1h	R/W-1h
7	6	5	4	3	2	1	0
LBE	RESERVED						UARTEN
R/W-0h	R-0h						R/W-0h

表 21-12. CTL 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15	CTSEN	R/W	0h	CTS 硬件流控制使能 0h = 禁用 CTS 硬件流控制 1h = 启用 CTS 硬件流控制
14	RTSEN	R/W	0h	RTS 硬件流控制使能 0h = 禁用 RTS 硬件流控制 1h = 启用 RTS 硬件流控制
13-12	RESERVED	R	0h	保留
11	RTS	R/W	0h	请求发送 该位是低电平有效 UART RTS 输出的补码。也就是说，当该位被编程为 1 时，引脚上的 RTS 输出为低电平。
10	RESERVED	R	0h	保留
9	RXE	R/W	1h	UART 接收使能 如果在接收过程中禁用 UART，则会在接收完当前字符后再停止。 0h = 禁用 UART 接收 1h = 启用 UART 接收
8	TXE	R/W	1h	UART 传输使能 如果传输在过程中禁用 UART，则会在发送完当前字符后再停止。 0h = 禁用 UART 传输 1h = 启用 UART 传输
7	LBE	R/W	0h	UART 回环使能： 启用回环模式后，UART 的发送输出引脚 UARTTXD 会被连接到 UART 的接收输入引脚 UARTRXD。 0h = 禁用回环 1h = 启用回环
6-1	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 21-12. CTL 寄存器字段说明（续）

位	字段	类型	复位	说明
0	UARTEN	R/W	0h	DAC 使能 0h = 禁用 UART 1h = 启用 UART

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

21.7.1.9 IFLS 寄存器（偏移 = 34h）[复位 = 00000012h]

图 21-12 展示了 IFLS，表 21-13 中对此进行了介绍。

返回到汇总表。

中断 FIFO 级别选择

图 21-12. IFLS 寄存器

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										RXSEL			TXSEL		
R-0h										R/W-2h			R/W-2h		

表 21-13. IFLS 寄存器字段说明

位	字段	类型	复位	说明
31-6	RESERVED	R	0h	保留
5-3	RXSEL	R/W	2h	接收中断 FIFO 级别选择： 该字段设置接收中断的触发点。值 0b101-0b111 被保留。 0h = 1_8：接收 FIFO 变为 >= 1/8 满 1h = 2_8： 接收 FIFO 变为 >= 1/4 满 2h = 4_8： 接收 FIFO 变为 >= 1/2 满 3h = 6_8： 接收 FIFO 变为 >= 3/4 满 4h = 7_8： 接收 FIFO 变为 >= 7/8 满
2-0	TXSEL	R/W	2h	传输中断 FIFO 级别选择： 该字段设置传输中断的触发点。值 0b101-0b111 被保留。 0h = 1_8：传输 FIFO 变为 <= 1/8 满 1h = 2_8： 传输 FIFO 变为 <= 1/4 满 2h = 4_8： 传输 FIFO 变为 <= 1/2 满 3h = 6_8： 传输 FIFO 变为 <= 3/4 满 4h = 7_8： 传输 FIFO 变为 <= 7/8 满

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

21.7.1.10 IMSC 寄存器（偏移 = 38h）[复位 = 00000000h]

图 21-13 展示了 IMSC，表 21-14 中对此进行了介绍。

返回到汇总表。

中断掩码设置/清除

图 21-13. IMSC 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED				EOTIM	OEIM	BEIM	PEIM
R-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
FEIM	RTIM	TXIM	RXIM	RESERVED		CTSMIM	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h		R/W-0h	R-0h

表 21-14. IMSC 寄存器字段说明

位	字段	类型	复位	说明
31-12	RESERVED	R	0h	保留
11	EOTIM	R/W	0h	传输结束中断掩码。读取操作返回 UART EoT 中断的当前掩码状态。写入 1 时，EoT 中断的掩码被置位，这意味着中断状态会反映在 MIS.EOTMIS 中。写入 0 时，该掩码被清除，这意味着 MIS.EOTMIS 不会反映该中断的状态。
10	OEIM	R/W	0h	超限错误中断掩码。读取操作返回 UART 超限错误中断的当前掩码。写入 1 时，此超限错误中断的掩码被置位，这意味着中断状态会反映在 MIS.OEMIS 中。写入 0 时，该掩码被清除，这意味着 MIS.OEMIS 不会反映该中断的状态。
9	BEIM	R/W	0h	间断错误中断掩码。读取操作返回 UART 间断错误中断的当前掩码。写入 1 时，此超限错误中断的掩码被置位，这意味着中断状态会反映在 MIS.BEMIS 中。写入 0 时，该掩码被清除，这意味着 MIS.BEMIS 不会反映该中断的状态。
8	PEIM	R/W	0h	奇偶校验错误中断掩码。读取操作返回 UART 奇偶校验错误中断的当前掩码。写入 1 时，此超限错误中断的掩码被置位，这意味着中断状态会反映在 MIS.PEMIS 中。写入 0 时，该掩码被清除，这意味着 MIS.PEMIS 不会反映该中断的状态。
7	FEIM	R/W	0h	组帧错误中断掩码。读取操作返回 UART 组帧错误中断的当前掩码。写入 1 时，此超限错误中断的掩码被置位，这意味着中断状态会反映在 MIS.FEMIS 中。写入 0 时，该掩码被清除，这意味着 MIS.FEMIS 不会反映该中断的状态。
6	RTIM	R/W	0h	接收超时中断掩码。读取操作返回 UART 接收超时中断的当前掩码。写入 1 时，此超限错误中断的掩码被置位，这意味着中断状态会反映在 MIS.RTMIS 中。写入 0 时，该掩码被清除，这意味着此位字段不会反映该中断的状态。 除非掩码已置位 (RTIM = 1)，否则无法设置接收超时 RIS.RTRIS 的原始中断。这是因为该掩码可启用省电节能功能。也就是说，从 MIS.RTMIS 和 RIS.RTRIS 中可以读取到相同的状态。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 21-14. IMSC 寄存器字段说明（续）

位	字段	类型	复位	说明
5	TXIM	R/W	0h	传输中断掩码。读取操作返回 UART 传输中断的当前掩码。写入 1 时，此超限错误中断的掩码被置位，这意味着中断状态会反映在 MIS.TXMIS 中。写入 0 会清除屏蔽，这意味着 MIS.TXMIS 不会反映中断。
4	RXIM	R/W	0h	接收中断掩码。读取操作返回 UART 接收中断的当前掩码。写入 1 时，此超限错误中断的掩码被置位，这意味着中断状态会反映在 MIS.RXMIS 中。写入 0 时，该掩码被清除，这意味着 MIS.RXMIS 不会反映该中断的状态。
3-2	RESERVED	R	0h	保留
1	CTSMIM	R/W	0h	允许发送 (CTS) 调制解调器中断掩码。读取操作返回 UART 允许发送中断的当前掩码。写入 1 时，此超限错误中断的掩码被置位，这意味着中断状态会反映在 MIS.CTSMMIS 中。写入 0 时，该掩码被清除，这意味着 MIS.CTSMMIS 不会反映该中断的状态。
0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

21.7.1.11 RIS 寄存器（偏移 = 3Ch）[复位 = 0000000Dh]

图 21-14 展示了 RIS，表 21-15 中对此进行了介绍。

返回到汇总表。

原始中断状态

图 21-14. RIS 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED				EOTRIS	OERIS	BERIS	PERIS
R-0h				R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
FERIS	RTRIS	TXRIS	RXRIS	RESERVED		CTSRMIS	RESERVED
R-0h	R-0h	R-0h	R-0h	R-0h		R-X	R-0h

表 21-15. RIS 寄存器字段说明

位	字段	类型	复位	说明
31-12	RESERVED	R	0h	保留
11	EOTRIS	R	0h	传输结束中断状态： 该字段返回 UART 传输结束中断的原始中断状态。当 FIFO 中以及 TX 线路上的所有传输数据均已传输完毕时，传输结束标志被置位。
10	OERIS	R	0h	超限错误中断状态： 该字段返回 UART 超限错误中断的原始中断状态。如果接收到数据且接收 FIFO 已满，则会发超限错误。
9	BERIS	R	0h	间断错误中断状态： 该字段返回 UART 间断错误中断的原始中断状态。当检测到间断条件时，间断错误被置位，这表示接收数据输入（UARTRXD 输入引脚）保持低电平的时间过长，超过一个完整字传输时间（包含起始位、数据位、奇偶校验位和停止位）。
8	PERIS	R	0h	奇偶校验错误中断状态： 该字段返回 UART 奇偶校验错误中断的原始中断状态。如果所接收数据字符的奇偶校验与 LCRH.EPS 和 LCRH.SPS 选择的奇偶校验不匹配，则奇偶校验错误被置位。
7	FERIS	R	0h	组帧错误中断状态： 该字段返回 UART 组帧错误中断的原始中断状态。如果接收到的字符没有有效的停止位（有效的停止位为 1），则组帧错误被置位。
6	RTRIS	R	0h	接收超时中断状态： 该字段返回 UART 接收超时中断的原始中断状态。当接收 FIFO 不为空且在 32 位周期内没有接收到更多数据时，此接收超时中断被置为有效。接收超时中断可通过以下两种方式清除：一是通过读取所有数据使 FIFO 变空，二是向 ICR.RTIC 写入 1。 除非掩码被置位 (IMSC.RTIM = 1)，否则无法设置接收超时的原始中断。这是因为该掩码可启用省电节能功能。也就是说，从 MIS.RTMIS 和 RTRIS 中可以读取到相同的状态。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 21-15. RIS 寄存器字段说明（续）

位	字段	类型	复位	说明
5	TXRIS	R	0h	传输中断状态： 该字段返回 UART 传输中断的原始中断状态。 当 FIFO 被使能时 (LCRH.FEN = 1)，如果传输 FIFO 中的字节数小于或等于编程设定的触发阈值 (IFLS.TXSEL)，则传输中断被置为有效。传输中断可通过以下两种方式清除：向传输 FIFO 写入数据，直至其中的数据量大于触发阈值，或者通过 ICR.TXIC 清除中断。 当 FIFO 被禁用时 (LCRH.FEN = 0)，即 FIFO 深度为 1 个单元时，如果发送器的单个存储单元中没有数据，则传输中断被置为有效。可以通过对传输 FIFO 执行单次写入操作来清除中断，或通过 ICR.TXIC 清除中断。
4	RXRIS	R	0h	接收中断状态： 该字段返回 UART 接收中断的原始中断状态。 当 FIFO 被使能时 (LCRH.FEN = 1)，如果接收 FIFO 达到编程设定的触发阈值 (IFLS.RXSEL)， 则接收中断被置为有效。接收中断可通过以下两种方式清除：从接收 FIFO 读取数据，直至其中的数据量小于触发阈值，或者通过 ICR.RXIC 清除中断。 当 FIFO 被禁用时 (LCRH.FEN = 0)，即 FIFO 深度为 1 个单元时，如果接收到的数据充满该存储单元， 则接收中断被置为有效。通过对接收 FIFO 执行单次读取，或通过 ICR.RXIC 清除中断来清除接收中断。
3-2	RESERVED	R	0h	保留
1	CTSRMIS	R	X	允许发送 (CTS) 调制解调器中断状态： 该字段返回 UART 允许发送中断的原始中断状态。
0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

21.7.1.12 MIS 寄存器（偏移 = 40h）[复位 = 00000000h]

图 21-15 展示了 MIS，表 21-16 中对此进行了介绍。

返回到汇总表。

已屏蔽中断状态

图 21-15. MIS 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED				EOTMIS	OEMIS	BEMIS	PEMIS
R-0h				R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
FEMIS	RTMIS	TXMIS	RXMIS	RESERVED		CTSMMIS	RESERVED
R-0h	R-0h	R-0h	R-0h	R-0h		R-0h	R-0h

表 21-16. MIS 寄存器字段说明

位	字段	类型	复位	说明
31-12	RESERVED	R	0h	保留
11	EOTMIS	R	0h	传输结束中断状态： 该字段返回超限中断的屏蔽后中断状态，即原始中断状态 RIS.EOTRIS 与掩码设置 IMSC.EOTIM 的“与”运算结果。
10	OEMIS	R	0h	超限错误的屏蔽后中断状态： 该字段返回超限中断的屏蔽后中断状态，即原始中断状态 RIS.OERIS 与掩码设置 IMSC.OEIM 的“与”运算结果。
9	BEMIS	R	0h	间断错误的屏蔽后中断状态： 该字段返回间断错误中断的屏蔽后中断状态，即原始中断状态 RIS.BERIS 与掩码设置 IMSC.BEIM 的“与”运算结果。
8	PEMIS	R	0h	奇偶校验错误的屏蔽后中断状态： 该字段返回奇偶校验错误中断的屏蔽后中断状态，即原始中断状态 RIS.PERIS 与掩码设置 IMSC.PEIM 的“与”运算结果。
7	FEMIS	R	0h	组帧错误的屏蔽后中断状态：返回组帧错误中断的屏蔽后中断状态，即原始中断状态 RIS.FERIS 与掩码设置 IMSC.FEIM 的“与”运算结果。
6	RTMIS	R	0h	接收超时错误的屏蔽后中断状态： 返回接收超时中断的屏蔽后中断状态。 除非掩码已置位 (IMSC.RTIM = 1)、否则无法设置接收超时的原始中断。这是因为该掩码可启用省电节能功能。也就是说，从 RTMIS 和 RIS.RTRIS 中可以读取到相同的状态。
5	TXMIS	R	0h	传输屏蔽中断状态： 该字段返回传输中断的屏蔽后中断状态，即原始中断状态 RIS.TXRIS 与掩码设置 IMSC.TXIM 的“与”运算结果。
4	RXMIS	R	0h	接收屏蔽后中断状态： 该字段返回接收中断的屏蔽后中断状态，即原始中断状态 RIS.RXRIS 与掩码设置 IMSC.RXIM 的“与”运算结果。
3-2	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 21-16. MIS 寄存器字段说明（续）

位	字段	类型	复位	说明
1	CTSMMIS	R	0h	允许发送 (CTS) 调制解调器屏蔽后中断状态： 该字段返回允许发送中断的屏蔽后中断状态，即原始中断状态 RIS.CTSRMIS 与掩码设置 IMSC.CTSMIM 的“与”运算结果。
0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

21.7.1.13 ICR 寄存器（偏移 = 44h）[复位 = 00000000h]

图 21-16 展示了 ICR，表 21-17 中对此进行了介绍。

返回到汇总表。

中断清除
写入 1 即可清除相应的中断。写入 0 不起作用。

图 21-16. ICR 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED				EOTIC	OEIC	BEIC	PEIC
R-0h				W-X	W-X	W-X	W-X
7	6	5	4	3	2	1	0
FEIC	RTIC	TXIC	RXIC	RESERVED		CTSMIC	RESERVED
W-X	W-X	W-X	W-X	W-X		W-X	W-X

表 21-17. ICR 寄存器字段说明

位	字段	类型	复位	说明
31-12	RESERVED	R	0h	保留
11	EOTIC	W	X	传输结束中断清除： 向该字段写入 1 会清除超限错误中断 (RIS.EOTRIS)。写入 0 无效。
10	OEIC	W	X	超限错误中断清除： 向该字段写入 1 会清除超限错误中断 (RIS.OERIS)。写入 0 无效。
9	BEIC	W	X	中断错误中断清除： 向该字段写入 1 会清除中断错误中断 (RIS.BERIS)。写入 0 无效。
8	PEIC	W	X	奇偶校验错误中断清除： 向该字段写入 1 会清除奇偶校验错误中断 (RIS.PERIS)。写入 0 无效。
7	FEIC	W	X	组帧错误中断清除： 向该字段写入 1 会清除组帧错误中断 (RIS.FERIS)。写入 0 无效。
6	RTIC	W	X	接收超时中断清除： 向该字段写入 1 会清除接收超时中断 (RIS.RTRIS)。写入 0 无效。
5	TXIC	W	X	传输中断清除： 向该字段写入 1 会清除传输中断 (RIS.TXRIS)。写入 0 无效。
4	RXIC	W	X	接收中断清除： 向该字段写入 1 会清除接收中断 (RIS.RXRIS)。写入 0 无效。
3-2	RESERVED	W	X	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。写入 0
1	CTSMIC	W	X	允许发送 (CTS) 调制解调器中断清除： 向该字段写入 1 会清除允许发送中断 (RIS.CTSRMIS)。写入 0 无效。
0	RESERVED	W	X	软件不应依赖保留位的值。写入除复位值之外的任何其他值都可能导致未定义的行为。写入 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

21.7.1.14 DMACTL 寄存器（偏移 = 48h）[复位 = 00000000h]

图 21-17 展示了 DMACTL，表 21-18 中对此进行了介绍。

返回到汇总表。

DMA 控制

图 21-17. DMACTL 寄存器

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

表 21-18. DMACTL 寄存器字段说明

位	字段	类型	复位	说明
31-3	RESERVED	R	0h	保留
2	DMAONERR	R/W	0h	DMA 开启错误。如果该位设置为 1，则在 UART 错误中断被置为有效（更具体地说，如果任何错误中断 RIS.PERIS、RIS.BERIS、RIS.FERIS 或 RIS.OERIS 被置为有效）时，禁用 DMA 接收请求输出（用于单个和突发请求）。
1	TXDMAE	R/W	0h	传输 DMA 使能。如果该位设置为 1，则启用传输 FIFO 的 DMA。
0	RXDMAE	R/W	0h	接收 DMA 使能。如果该位设置为 1，则启用接收 FIFO 的 DMA。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

章节 22

同步串行接口 (SSI)

本章介绍了同步串行接口 (SSI)。

22.1 简介 1869
22.2 方框图 1870
22.3 信号描述 1871
22.4 功能说明 1871
22.5 DMA 操作 1880
22.6 初始化和配置 1880
22.7 SSI 寄存器 1881

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

22.1 简介

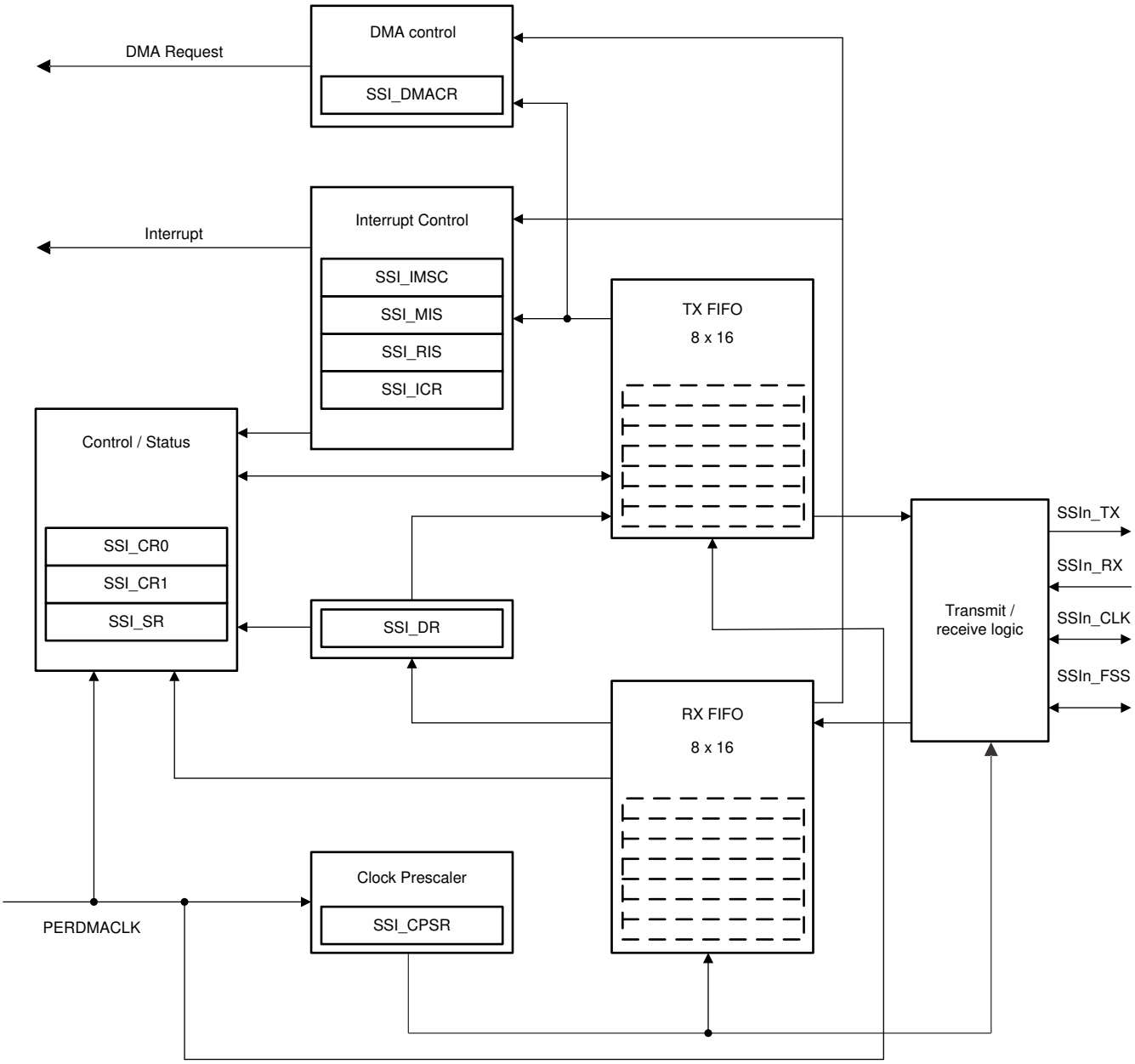
CC13x2 和 CC26x2 器件平台的两个 SSI 模块具有以下特性：

- 可编程接口操作，支持 Motorola SPI、MICROWIRE 或 TI SSI
- 可配置为接口上的主器件或从器件
- 可编程的时钟位速率以及预分频器；
- 独立的发送 (TX) 和接收 (RX) 先入先出缓冲区 (FIFO)，每个 16 位宽，深度为 8
- 可编程数据帧大小为 4 位到 16 位
- 内部环回测试模式，用于诊断和调试测试
- 发送和接收 FIFO 中断、超限和超时中断以及 DMA 完成中断
- 用微型直接内存访问控制器 (µDMA) 进行的高效传输：
  - 相互独立的发送通道和接收通道
  - 当数据位于 FIFO 中时，接收单个请求被置为有效；当 FIFO 包含四个或更多条目时，突发请求被置为有效
  - 当 FIFO 中有空间时，发出单次发送请求；当 FIFO 包含四个或更少条目时，发出突发请求

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

22.2 方框图

图 22-1 展示了 SSI 方框图。



Copyright © 2017, Texas Instruments Incorporated

图 22-1. SSI 模块方框图

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

22.3 信号描述

表 22-1 列出了 SSI 模块的外部信号，并说明了每个模块的功能。SSI 信号通过 IOC 模块中的 IOCFGn 寄存器进行选择。有关 GPIO 配置的更多信息，请参阅章节 5。

表 22-1. SSI 信号

信号名称	引脚编号	引脚类型(1)	说明
SSI0_CLK	在 I/O 控制器中分配	I/O	SSI 模块 0 时钟引脚
SSI0_FSS		I/O	SSI 模块 0 帧引脚
SSI0_RX		I	SSI 模块 0 RX 引脚
SSI0_TX		O	SSI 模块 0 TX 引脚
SSI1_CLK		I/O	SSI 模块 1 时钟引脚
SSI1_FSS		I/O	SSI 模块 1 帧引脚
SSI1_RX		I	SSI 模块 1 RX 引脚
SSI1_TX		O	SSI 模块 1 TX 引脚

(1) I = 输入；O = 输出；I/O = 双向

22.4 功能说明

SSI 对从外设器件接收到的数据执行串行至并行转换。CPU 访问数据、控制和状态信息。内部 FIFO 存储器缓冲发送和接收路径，允许在发送和接收模式下独立存储多达 8 个 16 位值。SSI 还支持 µDMA 接口。TX 和 RX FIFO 可以在 µDMA 模块中编程为目标地址或源地址。通过设置 SSI:DMACR 寄存器中的相应位来使能 µDMA 操作。

22.4.1 比特率生成

SSI 包含一个可编程位速率时钟分频器和预分频器来生成串行输出时钟。支持的比特率可达 2MHz 及以上，最大比特率由外设器件决定。

串行位速率是通过对输入时钟 (SysClk) 分频得到的。首先，时钟被一个范围为 2 到 254 的偶数预分频值 CPSDVSR 分频，该值在 SSI 时钟预分频寄存器 (SSI:CPSR) 中编程（请参阅节 22.7.1）。然后，时钟再被一个 1 到 256 之间的值分频，该值为 1 + SCR，其中 SCR 是 SSI 控制 0 寄存器 (SSI:CR0) 中编程的值（请参阅节 22.7.1）。

方程式 8 定义了输出时钟 SSIn_CLK 的频率。

$$SSIn_CLK = PERDMACLK / [CPSDVSR \times (1 + SCR)] \quad (8)$$

备注

对于从模式，内核时钟 (PERDMACLK) 必须至少比 SSIn_CLK 快 12 倍。

对于主模式，内核时钟 (PERDMACLK) 必须至少比 SSIn_CLK 快 2 倍。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

22.4.2 FIFO 操作

22.4.2.1 发送 FIFO

公共 TX FIFO 是一个 16 位宽、8 位置深的先入先出存储器缓冲区。CPU 通过写入 SSI 数据寄存器 SSI:DR（请参见方程式 8）数据写入 FIFO，数据存储在 FIFO 中，直到被传输逻辑读出。

当配置为主器件或从器件时，并行数据在串行转换之前写入 TX FIFO，并分别通过 SSIn_TX 引脚传输到连接的从器件或主器件。

在从器件模式下，每次主器件发起事务时，SSI 都会发送数据。如果 TX FIFO 为空且主器件发起事务，则从器件发送零。用户或软件负责根据需要在 FIFO 中提供有效数据。SSI 可以配置为当 FIFO 半满（< 4 个字）时生成中断，或者当 FIFO 未满时生成 µDMA 请求。

22.4.2.2 接收 FIFO

公共 RX FIFO 是一个 16 位宽、8 位置深的先入先出存储器缓冲区。从串行接口接收到的数据在由 CPU 读出之前一直存储在缓冲区中，CPU 通过读取 SSI:DR 寄存器来访问读取 FIFO。

当配置为主器件或从器件时，通过 SSIn_RX 引脚接收到的串行数据在被分别并行加载到所连接的从器件或主器件的 RX FIFO 之前，会先进行寄存器同步。

22.4.3 中断

在观察到以下条件时，SSI 可以生成中断：

- TX FIFO 服务（当 TX FIFO 半满或更少时）
- RX FIFO 服务（当 RX FIFO 半满或更多时）
- RX FIFO 超时
- RX FIFO 超限

所有中断事件在发送到事件结构之前都一起进行“或”运算，因此无论活动中断的数量是多少，SSI 都只向控制器生成一个中断请求。通过清除 SSI:IMSC 寄存器中的相应位，可以屏蔽 TX FIFO、RX FIFO、RX 超时和 RX 超限中断。设置 SSI:IMSC 寄存器中的相应屏蔽位可使能该中断。通过设置 UDMA 通道请求完成屏蔽寄存器 (UDMA:DONEMASK) 中的相应位，可以屏蔽 RX DMA 完成和 TX DMA 完成中断。清除 UDMA:DONEMASK 寄存器中的相应位可使能 RX 或 TX DMA 完成中断。

各个中断源的状态可从 SSI 原始中断状态寄存器 (SSI:RIS) 和 SSI 屏蔽中断状态寄存器 (SSI:MIS) 读取（请参阅节 22.7.1）。

当接收 FIFO 中有四个或更多的有效条目时，接收 FIFO 服务中断请求 SSI:RIS.RXRIS 将置为有效。

当发送 FIFO 中有四个或更少的有效条目时，发送 FIFO 服务中断请求 SSI:RIS.TXRIS 将置为有效。发送器中断不受 SSP 使能信号的限制，这允许在启用 SSP 和中断之前将数据写入发送 FIFO，并允许使能 SSP 和中断后由中断服务例程 (ISR) 将数据写入发送 FIFO。

当 FIFO 已满且接收到额外的数据帧时，接收超限中断 SSI:RIS.RORRIS 请求置为有效，从而导致 FIFO 超限。数据在接收移位寄存器中被覆盖，但在 FIFO 中不会被覆盖。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

RX FIFO 的超时周期为 32 个周期，速率为 SSIn_CLK (无论 SSIn_CLK 当前是否处于活动状态) ，并在 RX FIFO 从空变为非空时启动。如果在 32 个时钟过去之前清空 RX FIFO，则超时周期会复位。因此，ISR 会在读取 RX FIFO 后立即通过将 SSI 中断清除 SSI:ICR 寄存器中的 RTIC 位设置为 1 来清除 RX FIFO 超时中断。

备注

清除中断的操作不能过晚，以免在中断实际被清除之前 ISR 就已返回，否则可能会导致 ISR 被不必要地重新激活。

22.4.4 帧格式

每个数据帧的长度介于 4 位至 16 位之间，具体取决于编程的数据大小，并且数据帧从最高有效位 (MSB) 开始传输。可以选择以下三种基本帧类型：

- TI 同步串行
- ™Motorola SPI
- National MICROWIRE

对于这三种格式，串行时钟 (SSIn_CLK) 在 SSI 空闲时保持非活动状态，并且 SSIn_CLK 仅在正在传输或接收数据期间以编程频率进行转换。如果 RX FIFO 在超时期间之后仍包含数据，则会发生接收超时，而 SSIn_CLK 的 IDLE 状态会提供相应的超时指示。

对于 Motorola SPI 和 MICROWIRE 帧格式，串行帧 (SSIn_FSS) 引脚为低电平有效并在整个帧传输期间置为有生效 (下拉) 。

对于 TI 同步串行帧格式，SSIn_FSS 引脚以脉冲形式持续一个串行时钟周期，该周期从每个帧传输之前的上升沿开始。对于这种帧格式，SSI 和片外从器件在 SSIn_CLK 的上升沿驱动其输出数据，并在下降沿从另一个器件锁存数据。

与另外两种帧格式的全双工传输不同，MICROWIRE 格式使用特殊的半双工主从消息传递技术。当一帧开始时，会先向片外从器件发送一条 8 位控制消息。在此传输期间，SSI 不接收任何传入数据。消息发出后，片外从器件对其进行解码，并在 8 位控制消息的最后一位发送后等待一个串行时钟后以请求的数据进行响应。返回的数据长度可以是 4 到 16 位，因此总帧长度为 13 到 25 位。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

22.4.4.1 德州仪器 (TI) 同步串行帧格式

图 22-2 所示为单个传输帧的 TI 同步串行帧格式。

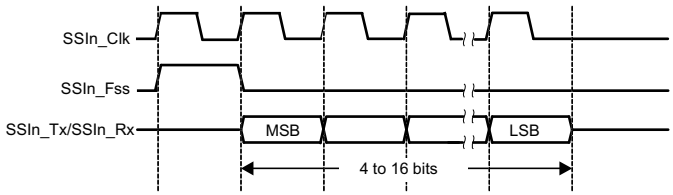


图 22-2. TI 同步串行帧格式（单个传送）

当 SSI 处于空闲状态时，SSIn_CLK 和 SSIn_FSS 被强制拉低，传输数据线 SSIn_TX 则被置于高阻态。当 TX FIFO 的底部条目包含数据时，SSIn_FSS 会输出高脉冲并持续一个 SSIn_CLK 周期。发送的值也从 TX FIFO 传输到发送逻辑的串行移位寄存器。在 SSIn_CLK 的下一个上升沿，4 至 16 位数据帧的 MSB 在 SSIn_TX 引脚上移出。同样，接收到的数据的 MSB 也通过片外串从器件移到 SSIn_RX 引脚上。

然后，SSI 和片外串行从器件均在 SSIn_CLK 的每个下降沿将数据位逐个记录到相应的串行移位器中。在最低有效位 (LSB) 被锁存之后 SSIn_CLK 的第一个上升沿，接收到的数据从串行移位器传输到 RX FIFO。

图 22-3 所示为传输背对背帧时的 TI 同步串行帧格式。

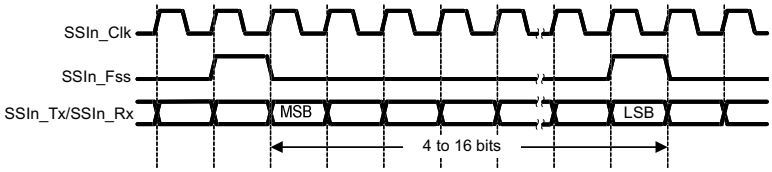


图 22-3. TI 同步串行帧格式（连续传送）

22.4.4.2 Motorola SPI 帧格式

Motorola SPI 接口是一个 4 线接口，其中 SSIn_FSS 信号充当从器件选择。Motorola SPI 格式的主要特点是：可以通过 SSI:CR0 控制寄存器中的 SPO 和 SPH 位，对 SSIn_CLK 信号的非活动状态和相位进行编程。

22.4.4.2.1 SPO 时钟极性位

当 SPO 时钟极性控制位清零时，该位会在 SSIn_CLK 引脚上产生一个稳态低电平值。如果设置了 SPO 位，则当不传输数据时，该位会在 SSIn_CLK 引脚上放置一个稳态高电平值。

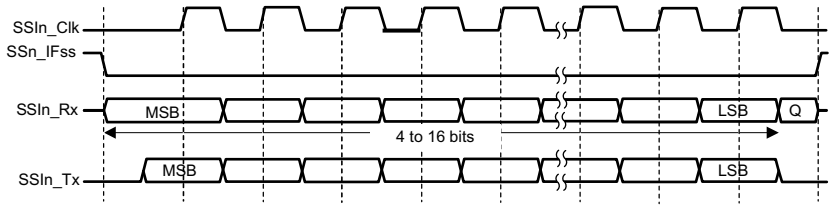
22.4.4.2.2 SPH 相位控制位

SPH 相位控制位用来选择捕捉数据的时钟边沿，并允许边沿改变状态。该位的状态对发送的第一个位影响最大，即允许或不允许在第一个数据捕捉边沿之前进行时钟转换。当 SPH 相位控制位清零时，将在第一个时钟边沿转换时捕获数据。如果 SPH 位为高，则在第二个时钟边沿转换时捕获数据。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

22.4.4.3 SPO = 0 和 SPH = 0 时的 Motorola SPI 帧格式

图 22-4 和图 22-5 分别显示了 SPO = 0 和 SPH = 0 时 Motorola SPI 格式的单次和连续发送信号序列。



Q 未定义。

图 22-4. SPO = 0 和 SPH = 0 时的 Motorola SPI 格式（单次发送）

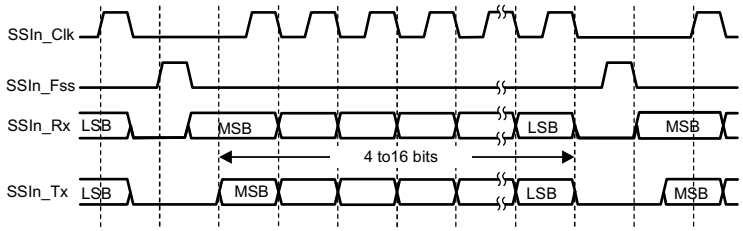


图 22-5. SPO = 0 和 SPH = 0 时的 Motorola SPI 格式（连续发送）

在此配置中，空闲期间会发生以下情况：

- SSIn_CLK 被强制置为低电平
- SSIn_FSS 被强制置为高电平
- 发送数据线 SSIn_TX 被任意强制置为高电平
- 当 SSI 配置为主器件时，SSI 启用 SSIn_CLK DIO
- 当 SSI 配置为从器件时，SSI 禁用 SSIn_CLK DIO

如果启用了 SSI 并且 TX FIFO 中存在有效数据，则 SSIn_FSS 主器件信号在发送开始时被驱动为低电平，这会导致使从器件数据能够进入到主器件的 SSIn_RX 输入线路上。启用主器件 SSIn_TX 输出 DIO。

半个 SSIn_CLK 周期后，有效的主器件数据会传输到 SSIn_TX 引脚上。一旦同时设置了主器件和从器件数据，SSIn_CLK 主器件时钟引脚就会在额外的半个 SSIn_CLK 周期后变为高电平。

现在，将在 SSIn_CLK 信号的上升沿捕捉数据，并在下降沿传播数据。

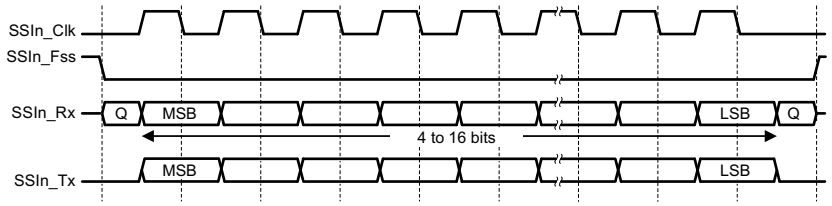
对于单字发送，在传输数据字的所有位之后，SSIn_FSS 线会在最后一个位被捕获后的一个 SSIn_CLK 周期内返回其 IDLE 高电平状态。

对于连续的背靠背传输，SSIn_FSS 信号必须在每个数据字传输之间产生高脉冲，因为当 SPH 位清零时，从器件选择引脚会冻结其串行外设寄存器中的数据，并且不允许更改数据。主器件必须在每次数据传输之间拉高从器件的 SSIn_FSS 引脚，以启用串行外设数据写入。当连续传输完成时，SSIn_FSS 引脚将在最后一个位被捕捉后的一个 SSIn_CLK 周期内返回到其 IDLE 状态。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

22.4.4.4 SPO = 0 和 SPH = 1 时的 Motorola SPI 帧格式

图 22-6 显示了 SPO = 0 和 SPH = 1 时 Motorola SPI 格式的传输信号序列，涵盖了单次传输和连续传输。



Q 未定义。

图 22-6. SPO = 0 和 SPH = 1 时的 Motorola SPI 帧格式

在此配置中，空闲期间会发生以下情况：

- SSIn_CLK 被强制置为低电平
- SSIn_FSS 被强制置为高电平
- 发送数据线 SSIn_TX 被任意强制置为高电平
- 当 SSI 配置为主器件时，SSI 启用 SSIn_CLK DIO
- 当 SSI 配置为从器件时，SSI 禁用 SSIn_CLK DIO

如果启用了 SSI 并且 TX FIFO 中存在有效数据，则 SSIn_FSS 主器件信号在开始发送时变为低电平。启用主器件 SSIn_TX 输出。在额外的半个 SSIn_CLK 周期之后，主器件和从器件有效数据都将能够进入到各自的发送线路上。同时，利用上升沿转换启用 SSIn_CLK。然后，在 SSIn_CLK 信号的下降沿捕捉数据，并在上升沿传播数据。

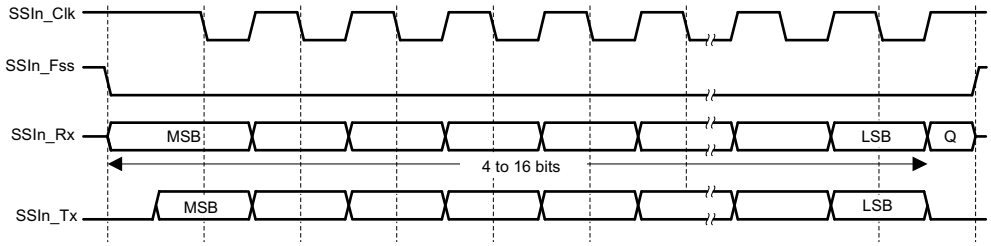
对于单字传输，在传输所有位之后，SSIn_FSS 线路会在最后一个位被捕捉后的一个 SSIn_CLK 周期内返回其 IDLE 高电平状态。

对于连续的背靠背传输，SSIn_FSS 引脚在连续数据字之间保持低电平，并像单字传输一样终止。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

22.4.4.5 SPO = 1 和 SPH = 0 时的 Motorola SPI 帧格式

图 22-7 和图 22-8 分别显示了 SPO = 1 和 SPH = 0 时 Motorola SPI 格式的单次和连续发送信号序列。



Q 未定义。

图 22-7. SPO = 1 和 SPH = 0 时的 Motorola SPI 帧格式（单次发送）

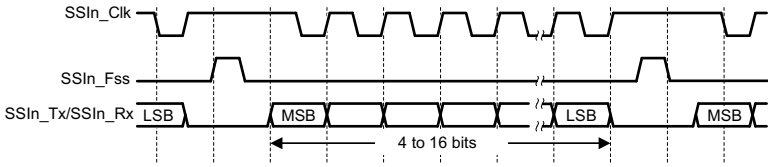


图 22-8. SPO = 1 和 SPH = 0 时的 Motorola SPI 帧格式（连续发送）

在此配置中，空闲期间会发生以下情况：

- SSIn_CLK 被强制置为高电平
- SSIn_FSS 被强制置为高电平
- 传输数据线 SSIn_TX 被任意强制置为低电平
- 当 SSI 配置为主器件时，SSI 启用 SSIn_CLK DIO
- 当 SSI 配置为从器件时，SSI 禁用 SSIn_CLK DIO

如果启用了 SSI 并且 TX FIFO 中存在有效数据，则 SSIFss 主器件信号在开始发送时变为低电平，并立即将从器件数据传输到主器件的 SSIn_RX 线上。启用主器件 SSIn_TX 输出 DIO。

半个 SSIn_CLK 周期后，有效的主器件数据会传输到 SSIn_TX 线上。当主器件和从器件数据都已设置时，SSIn_CLK 主器件时钟引脚在额外的半个 SSIn_CLK 周期后变为低电平。然后，在 SSIn_CLK 信号的下降沿捕获数据并在上升沿传播。

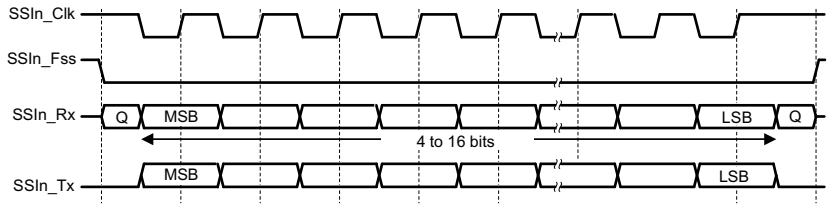
对于单字发送，在传输数据字的所有位之后，SSIn_FSS 线会在最后一个位被捕获后的一个 SSIn_CLK 周期内返回其 IDLE 高电平状态。

对于连续的背靠背发送，SSIn_FSS 信号必须在每个数据字传输之间产生高脉冲，因为当 SPH 位清零时，从器件选择引脚会冻结其串行外设寄存器中的数据，并防止更改数据。主器件必须在每次数据传输之间拉高从器件的 SSIn_FSS 引脚，以启用串行外设数据写入。当连续传输完成时，SSIn_FSS 引脚将在最后一个位被捕获后的一个 SSIn_CLK 周期内返回到其 IDLE 状态。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

22.4.4.6 SPO = 1 和 SPH = 1 时的 Motorola SPI 帧格式

图 22-9 显示了 SPO = 1 和 SPH = 1 时 Motorola SPI 格式的传输信号序列，其中涵盖单次和连续传输。



Q 未定义。

图 22-9. SPO = 1 和 SPH = 1 时的 Motorola SPI 帧格式

在此配置中，空闲期间会发生以下情况：

- SSIClk 被强制为高电平
- SSIn_FSS 被强制为高电平
- 发送数据线路 SSIn_TX 被任意强制低电平
- 当 SSI 配置为主器件时，SSI 启用 SSIn_CLK DIO
- 当 SSI 配置为从器件时，SSI 禁用 SSIn_CLK DIO

如果启用了 SSI 并且 TX FIFO 中存在有效数据，则通过 SSIn_FSS 主信号变为低电平来指示开始发送。启用主 SSIn_TX 输出 DIO。在额外的半个 SSIn_CLK 周期之后，主器件和从器件数据都将能够进入到其各自的发送线路上。同时，利用下降沿转换启用 SSIn_CLK。然后，在 SSIn_CLK 信号的上升沿捕捉数据，并在下降沿传播数据。

对于单字传输，在传输所有位之后，SSIn_FSS 线路会在最后一个位被捕捉后的一个 SSIn_CLK 周期内返回其 IDLE 高电平状态。

对于连续的背靠背传输，SSIn_FSS 引脚保持在其低电平有效状态，直到最后一个字的最后一个位被捕捉，然后返回其 IDLE 状态。

对于连续的背靠背传输，SSIn_FSS 引脚在连续数据字之间保持低电平，并像单字传输一样终止。

22.4.4.7 MICROWIRE 帧格式

图 22-10 显示了单个帧的 MICROWIRE 帧格式。图 22-11 显示了传输背对背帧时的相同格式。

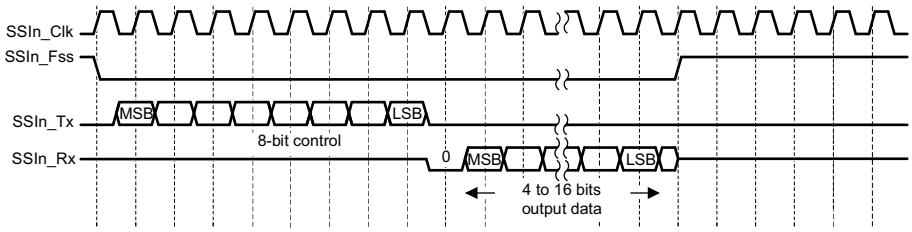


图 22-10. MICROWIRE 帧格式（单帧）

MICROWIRE 格式类似于 SPI 格式，不同之处在于传输为半双工模式，并使用主从器件消息传递技术。每次串行传输都以一个 8 位控制字开始，该控制字从 SSI 传输到片外从器件。在此传输期间，SSI 不接收传入的数据。发送消息后，片外从器件对其进行解码，并在 8 位控制消息的最后一位发送后等待一个串行时钟。然后，片外从器件以所需的数据进行响应。返回的数据长度为 4 到 16 位，使得总帧长度为 13 到 25 位。

在此配置中，空闲期间会发生以下情况：

- SSIn_CLK 被强制为低电平

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

- SSIn_FSS 被强制为高电平
- 发送数据线路 SSIn_TX 被任意强制低电平

向 TX FIFO 写入控制字节会触发传输。SSIn_FSS 的下降沿将 TX FIFO 底部条目中的值传输到发送逻辑的串行移位寄存器，并将 8 位控制帧的 MSB 移出到 SSIn_TX 引脚上。SSIn_FSS 在帧传输的持续时间内保持低电平。在此传输期间，SSIn_RX 引脚保持三态。

片外串行从器件在 SSIn_CLK 的每个上升沿将每个控制位锁存至其串行移位器中。在从器件锁存最后一位后，控制字节在一个时钟等待状态期间被解码，从器件通过将数据发送回 SSI 进行响应。每个位在 SSIn_CLK 的下降沿上驱动到 SSIn_RX 线路上。SSI 锁存 SSIn_CLK 上升沿上的每个位。在单次传输的帧结束时，SSIFss 信号在接收串行移位器中的最后一位锁存后的一个时钟周期内被拉高，从而将数据传输到 RX FIFO。

备注

片外从器件可以将接收线路置于三态，即在 SSIn_CLK 的下降沿（在 LSB 被接收移位器锁存之后）或 SSIn_FSS 引脚变为高电平时。

对于连续传输，数据传输像单次传输一样开始和结束，但 SSIn_FSS 线路保持低电平，数据背靠背传输。下一帧的控制字节遵循从当前帧接收到的数据的 LSB。将帧的 LSB 锁存到 SSI 后，在 SSIn_CLK 的下降沿上从接收移位器传输每个接收到的值。

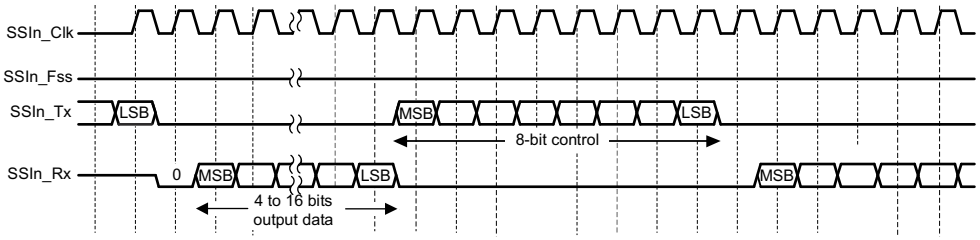


图 22-11. MICROWIRE 帧格式（连续传输）

在 MICROWIRE 模式下，SSI 从器件在 SSIFss 变为低电平后，在 SSIn_CLK 的上升沿上对接收数据的第一个位进行采样。与 SSIn_CLK 的上升沿相比，驱动自由运行 SSIn_CLK 的主器件必须确保 SSIFss 信号具有足够的建立和保持裕度。

图 22-12 显示了这些建立时间和保持时间要求。对于 SSIn_CLK 上升沿（SSI 从器件将对接收数据的第一位进行采样），SSIn_FSS 的设置必须至少是 SSI 在其上运行的 SSIn_CLK 周期的两倍。相对于该边沿之前的 SSIn_CLK 上升沿，SSIn_FSS 必须具有至少一个 SSIn_CLK 周期的保持时间。

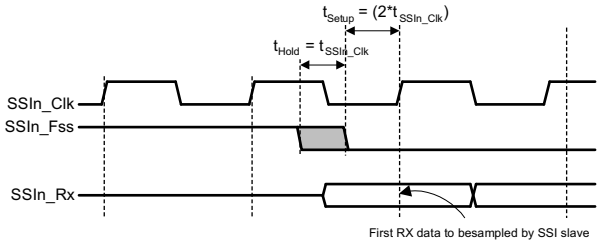


图 22-12. MICROWIRE 帧格式、SSIFss 输入设置及保持要求

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

22.5 DMA 操作

SSI 外设为 µDMA 控制器提供了一个接口，该接口具有独立的传输和接收通道。SSI DMA 控制寄存器 (SSI:DMACR) 允许 µDMA 操作 SSI。启用 µDMA 操作后，当相关的 FIFO 可以传输数据时，SSI 会在接收通道或传输通道上发出 µDMA 请求。对于接收通道，每当 RX FIFO 中有任何数据时，就会发出单次传输请求。每当 RX FIFO 中的数据达到四项或更多时，就会发出突发传输请求。对于传输通道，每当 TX FIFO 中至少有一个空位置时，就会发出单次传输请求。每当 TX FIFO 有四个或更多空槽时，就会发出突发请求。 µDMA 控制器会根据 µDMA 通道的配置自动处理单次和突发 µDMA 传输请求。

要启用接收通道的 µDMA 操作，请置位 SSI:DMACR RXDMAE 寄存器位。要启用传输通道的 µDMA 操作，请置位 SSI:MAC RTXDMAE 寄存器位。如果启用了 µDMA，并且在 DMA 完成屏蔽寄存器 (UDMA:DONEMASK) 中清除了相应位，则 µDMA 控制器会在传输完成时触发中断。该中断发生在 SSI 中断向量上。如果为 SSI 操作使用中断且启用了 µDMA，则必须设计 SSI 中断处理程序来处理 µDMA 完成中断。TX 和 RX DMA 完成中断的状态可从通道请求完成寄存器 (UDMA:REQDONE) 中读取。要清除 TX 和 RX DMA 完成中断，UDMA:REQDONE 寄存器中的相应位必须为 1。

有关编程 µDMA 控制器的更多详细信息，请参阅章节 14。

22.6 初始化和配置

要启用和初始化 SSI，请执行以下步骤：

1. 确保相应的电源域已正确加电。相关详细信息，请参阅章节 7。
2. 通过以下操作启用 PRCM 中的 SSI 模块：写入 PRCM:SSICLKGR 寄存器、PRCM:SSICLKGS 寄存器、PRCM:SSICLKGDS 寄存器，或使用以下 DriverLib 函数：

```
PRCMPeripheralRunEnable(uint32_t)
```

```
PRCMPeripheralSleepEnable(uint32_t)
```

```
PRCMPeripheralDeepSleepEnable(uint32_t)
```

```
and then loading the setting to clock controller by writing to PRCM:CLKLOADCTL
```

   或者使用 DriverLib 函数。

```
PRCMLoadSet().
```

3. 配置 IOC 模块，以将 SSIn_RX、SSIn_TX、SSIn_FSS 和 SSIn_CLK 功能从 I/O 路由到 SSI 模块。IOCFGn.PORTID 必须写入正确的 PORTID。

对于每种帧格式，应按照如下步骤配置 SSI：

1. 在进行任何配置更改之前，确保 SSI:CR1 寄存器中的 SSE 位清零。
2. 选择 SSI 用作主器件还是从器件：
   a. 对于主器件操作，将 SSI:CR1 寄存器设置为 0x0000 0000。
   b. 对于从器件模式（启用输出），将 SSI:CR1 寄存器设置为 0x0000 0004。
   c. 对于从器件模式（禁用输出），将 SSI:CR1 寄存器设置为 0x0000 000C。
3. 通过写入 SSI:CPSR 寄存器来配置时钟预分频除数。
4. 使用以下配置写入 SSI:CR0 寄存器：
   - 串行时钟速率 (SCR)
   - 如果使用 Motorola SPI 模式，需设置所需的时钟相位和极性（SPH 和 SPO）
   - 协议模式：Motorola SPI、TI SSF、MICROWIRE (FRF)
   - 数据大小 (DSS)

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

5. （可选）配置 µDMA 通道（请参阅章节 14），并在 SSI:DMACR 寄存器中启用 DMA 选项。
6. 通过置位 SSI:CR1 寄存器中的 SSE 位来启用 SSI。

例如，假设要求 SSI 配置按以下参数运行：

- 主器件运行
- 德州仪器 (TI) SSI 模式
- 1Mbps 比特率
- 8 个数据位

假设系统时钟为 48MHz，比特率计算如方程式 9 所示。

$$\text{SSIn_CLK} = \text{PERDMACLK} / [\text{CPSDVSR} \times (1 + \text{SCR})] \; 1 \times 106 = 20 \times 106 / [\text{CPSDVSR} \times (1 + \text{SCR})] \; 1000000 \text{ bps} = 48000000 \text{ Hz} / [2 \times (1 + 23)] \quad (9)$$

在这种情况下，如果 CPSDVSR = 0x2，则 SCR 必须为 0x18。

配置顺序为：

1. 确保 SSI:CR1 寄存器中的 SSE 位清零。
2. 向 SSI:CR1 寄存器写入值 0x0000 0000。
3. 向 SSI:CPSR 寄存器写入值 0x0000 0002。
4. 向 SSI:CR0 寄存器写入值 0x0000 1817。
5. 然后，通过置位 SSI:CR1 寄存器中的 SSE 位来启用 SSI。

22.7 SSI 寄存器

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

22.7.1 SSI 寄存器

表 22-2 列出了 SSI 寄存器的存储器映射寄存器。表 22-2 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

表 22-2. SSI 寄存器

偏移	首字母缩写词	寄存器名称	部分
0h	CR0	控制 0	CR0 寄存器（偏移 = 0h）[复位 = 00000000h]
4h	CR1	控制 1	CR1 寄存器（偏移 = 4h）[复位 = 00000000h]
8h	DR	数据	DR 寄存器（偏移 = 8h）[复位 = 00000000h]
Ch	SR	状态	SR 寄存器（偏移 = Ch）[复位 = 00000003h]
10h	CPSR	时钟预分频	CPSR 寄存器（偏移 = 10h）[复位 = 00000000h]
14h	IMSC	中断屏蔽设置和清除	IMSC 寄存器（偏移 = 14h）[复位 = 00000000h]
18h	RIS	原始中断状态	RIS 寄存器（偏移 = 18h）[复位 = 00000008h]
1Ch	MIS	已屏蔽中断状态	MIS 寄存器（偏移 = 1Ch）[复位 = 00000000h]
20h	ICR	中断清除	ICR 寄存器（偏移 = 20h）[复位 = 00000000h]
24h	DMACR	DMA 控制	DMACR 寄存器（偏移 = 24h）[复位 = 00000000h]

复杂的位访问类型经过编码可适应小型表单元。表 22-3 展示了适用于此部分中访问类型的代码。

表 22-3. SSI 访问类型代码

访问类型	代码	说明
读取类型		
R	R	读取
写入类型		
W	W	写入
复位或默认值		
-n		复位后的值或默认值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

22.7.1.1 CR0 寄存器（偏移 = 0h）[复位 = 00000000h]

图 22-13 展示了 CR0，表 22-4 对其进行了介绍。

返回到汇总表。

控制 0

图 22-13. CR0 寄存器

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
SCR								SPH	SPO	FRF		DSS			
R/W-0h								R/W-0h	R/W-0h	R/W-0h		R/W-0h			

表 22-4. CR0 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-8	SCR	R/W	0h	串行时钟速率： 这用于生成 SSI 的发送和接收比特率。比特率为 (SSI 的时钟频率)/((SCR+1)*CPSR.CPSDVSR)。 SCR 是 0-255 之间的值。
7	SPH	R/W	0h	CLKOUT 相位（仅适用于 Motorola SPI 帧格式） 该位选择捕获数据的时钟沿并使其能够更改状态。 由于可以允许或不允许在第一个数据捕获沿之前进行时钟转换，所以对第一个发送的位的影响最大。 0h = 1ST_CLK_EDGE：在第一次时钟沿转换时捕获数据。 1h = 2ND_CLK_EDGE：在第二次时钟沿转换时捕获数据。
6	SPO	R/W	0h	CLKOUT 极性（仅适用于 Motorola SPI 帧格式） 0h = 不传输数据时，SSI 在 CLKOUT 引脚上 生成稳态低电平值。 1h = 不传输数据时，SSI 在 CLKOUT 引脚上生成稳态高电平值。
5-4	FRF	R/W	0h	帧格式。 支持的帧格式为 Motorola SPI、TI 同步串行和 National Microwire。 值 0'b11 是保留值，不得使用。 0h = Motorola SPI 帧格式 1h = TI 同步串行帧格式 2h = National Microwire 帧格式
3-0	DSS	R/W	0h	数据大小选择。 值 0b0000、0b0001、0b0010 是保留值，不得使用。 3h = 4_BIT：4 位数据 4h = 5_BIT： 5 位数据 5h = 6_BIT： 6 位数据 6h = 7_BIT： 7 位数据 7h = 8_BIT： 8 位数据 8h = 9_BIT： 9 位数据 9h = 10_BIT： 10 位数据 Ah = 11_BIT： 11 位数据 Bh = 12_BIT： 12 位数据 Ch = 13_BIT： 13 位数据 Dh = 14_BIT： 14 位数据 Eh = 15_BIT： 15 位数据 Fh = 16_BIT： 16 位数据

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

22.7.1.2 CR1 寄存器（偏移 = 4h）[复位 = 00000000h]

图 22-14 展示了 CR1，表 22-5 对其进行了介绍。

返回到汇总表。

控制 1

图 22-14. CR1 寄存器

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												SOD	MS	SSE	LBM
R-0h												R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 22-5. CR1 寄存器字段说明

位	字段	类型	复位	说明
31-4	RESERVED	R	0h	保留
3	SOD	R/W	0h	禁用从模式输出 该位仅在从模式 (MS=1) 中适用。在多从系统中，SSI 主器件可以向系统中的所有从器件广播一条消息，同时确保只有一个从器件将数据驱动到其串行输出线路上。在这样的系统中，来自多个从器件的 RXD 线路可以连接在一起。要在此类系统中运行，如果 SSI 从器件不应驱动 TXD 线路，则可以设置该位字段： 0：SSI 可在从模式下驱动 TXD 输出。 1：SSI 无法在从模式下驱动 TXD 输出。
2	MS	R/W	0h	主或从模式选择。仅当禁用 SSI (SSE=0) 时才能修改该位。 0h = 器件配置为主器件 1h = 器件配置为从器件
1	SSE	R/W	0h	同步串行接口使能。 0h = SSI_DISABLED：操作已禁用 1h = SSI_ENABLED： 操作已启用
0	LBM	R/W	0h	环回模式： 0：正常串行端口操作已启用。 1：发送串行移位器的输出在内部连接到接收串行移位器的输入。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

22.7.1.3 DR 寄存器（偏移 = 8h）[复位 = 00000000h]

图 22-15 展示了 DR，表 22-6 中对此进行了介绍。

返回到汇总表。

数据
16 位宽数据寄存器：
读取时，会访问接收 FIFO 内部由当前 FIFO 读取指针指向的条目。当接收逻辑从传入数据帧中删除数据值时，会将这些数据值放入接收 FIFO 内部由当前 FIFO 写入指针指向的条目中。
写入时，会写入发送 FIFO 内部由写入指针指向的条目。发送逻辑从发送 FIFO 中一次移出一个数据值。这个数据值加载到发送串行移位器中，然后以编程的比特率串行移出到 TXD 输出引脚。
当所选的数据大小小于 16 位时，用户必须将写入发送 FIFO 的数据右对齐。发送逻辑忽略未使用的位。接收到的小于 16 位的数据在接收缓冲器中自动右对齐。

图 22-15. DR 寄存器

31	30	29	28	27	26	25	24	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-X															

表 22-6. DR 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	DATA	R/W	X	发送/接收数据 当 SSI 编程为小于 16 位的数据大小 (CR0.DSS != 0b1111) 时，从该字段读取或写入该字段的值必须右对齐。发送逻辑会忽略顶部未使用的位。接收逻辑会自动右对齐。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

22.7.1.4 SR 寄存器（偏移 = Ch）[复位 = 00000003h]

图 22-16 展示了 SR，表 22-7 中对此进行了介绍。

返回到汇总表。

状态

图 22-16. SR 寄存器

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											BSY	RFF	RNE	TNF	TFE
R-0h											R-0h	R-0h	R-0h	R-1h	R-1h

表 22-7. SR 寄存器字段说明

位	字段	类型	复位	说明
31-5	RESERVED	R	0h	保留
4	BSY	R	0h	串行接口忙： 0：SSI 空闲 1：SSI 当前正在发送和/或接收帧，或者发送 FIFO 不为空。
3	RFF	R	0h	接收 FIFO 已满： 0：接收 FIFO 未满。 1：接收 FIFO 已满。
2	RNE	R	0h	接收 FIFO 不为空 0：接收 FIFO 为空。 1：接收 FIFO 不为空。
1	TNF	R	1h	发送 FIFO 未满： 0：发送 FIFO 已满。 1：发送 FIFO 未满。
0	TFE	R	1h	发送 FIFO 为空： 0：发送 FIFO 不为空。 1：发送 FIFO 为空。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

22.7.1.5 CPSR 寄存器（偏移 = 10h）[复位 = 00000000h]

图 22-17 展示了 CPSR，表 22-8 中对此进行了介绍。

返回到汇总表。

时钟预分频

图 22-17. CPSR 寄存器

31	30	29	28	27	26	25	24	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																								CPSDVSR							
R-0h																								R/W-0h							

表 22-8. CPSR 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-0	CPSDVSR	R/W	0h	时钟预分频因子： 该字段指定 SSI 输入系统时钟在进一步使用之前必须进行内部分频的分频因子。 编程到此字段中的值必须为偶数非零值 (2-254)。编程数字的最低有效位被硬编码为零。如果向该寄存器写入奇数，则从该寄存器读回的数据 的最低有效位为零。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

22.7.1.6 IMSC 寄存器（偏移 = 14h）[复位 = 00000000h]

图 22-18 展示了 IMSC，表 22-9 中对此进行了介绍。

返回到汇总表。

中断屏蔽设置和清除

图 22-18. IMSC 寄存器

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

表 22-9. IMSC 寄存器字段说明

位	字段	类型	复位	说明
31-4	RESERVED	R	0h	保留
3	TXIM	R/W	0h	发送 FIFO 中断屏蔽： 读取操作返回发送 FIFO 中断的当前屏蔽。写入 1 时会设置发送 FIFO 中断的屏蔽，这意味着中断状态将反映在 MIS.TXMIS 中。写入 0 会清除屏蔽，这意味着 MIS.TXMIS 不会反映中断。
2	RXIM	R/W	0h	接收 FIFO 中断屏蔽： 读取操作返回接收 FIFO 中断的当前屏蔽。写入 1 时会设置接收 FIFO 中断的屏蔽，这意味着中断状态将反映在 MIS.RXMIS 中。写入 0 会清除屏蔽，这意味着 MIS.RXMIS 不会反映中断。
1	RTIM	R/W	0h	接收超时中断屏蔽： 读取操作返回接收超时中断的当前屏蔽。写入 1 时会设置接收超时中断的屏蔽，这意味着中断状态将反映在 MIS.RTMIS 中。写入 0 会清除屏蔽，这意味着 MIS.RTMIS 不会反映中断。
0	RORIM	R/W	0h	接收超限中断屏蔽： 读取操作返回接收溢出中断的当前屏蔽。写入 1 时会设置接收超限中断的屏蔽，这意味着中断状态将反映在 MIS.RORMIS 中。写入 0 会清除屏蔽，这意味着 MIS.RORMIS 不会反映中断。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

22.7.1.7 RIS 寄存器（偏移 = 18h）[复位 = 00000008h]

图 22-19 展示了 RIS，表 22-10 中对此进行了介绍。

返回到汇总表。

原始中断状态

图 22-19. RIS 寄存器

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

表 22-10. RIS 寄存器字段说明

位	字段	类型	复位	说明
31-4	RESERVED	R	0h	保留
3	TXRIS	R	1h	原始发送 FIFO 中断状态： 当发送 FIFO 中有四个或更少的有效条目时，发送中断将置为有效。 发送中断不受 SSI 使能信号的限定。因此，可以使用以下方式之一： - 在使能 SSI 和中断之前，可以将数据写入 发送 FIFO。 - 可以使能 SSI 和中断，以便通过中断服务例程将数据写入发送 FIFO。
2	RXRIS	R	0h	接收 FIFO 中断的原始中断状态： 当接收 FIFO 中有四个或更多的有效条目时，接收中断将置为有效。
1	RTRIS	R	0h	接收超时中断的原始中断状态： 当接收 FIFO 不为空且 SSI 在固定的 32 位周期内保持空闲时，此接收超时中断有效。该机制可用于通知用户，接收 FIFO 中仍有数据需要处理。如果后续读取使接收 FIFO 变为空或在 RXD 上接收到新数据，则该中断置为无效。 也可以通过写入 ICR.RTIC 来清除该中断。
0	RORRIS	R	0h	接收超限中断的原始中断状态： 当 FIFO 已满且接收到额外的数据帧时，接收超限中断将置为有效，从而导致 FIFO 超限。数据在接收移位寄存器中 被覆盖，但 FIFO 中的数据不受影响，因此 FIFO 内容保持有效。 也可以通过写入 ICR.RORIC 来清除该中断。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

22.7.1.8 MIS 寄存器（偏移 = 1Ch）[复位 = 00000000h]

图 22-20 展示了 MIS，表 22-11 中对此进行了介绍。

返回到汇总表。

已屏蔽中断状态

图 22-20. MIS 寄存器

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

表 22-11. MIS 寄存器字段说明

位	字段	类型	复位	说明
31-4	RESERVED	R	0h	保留
3	TXMIS	R	0h	发送 FIFO 中断的屏蔽中断状态： 该字段返回发送 FIFO 中断的屏蔽中断状态，该状态是原始中断状态 RIS.TXRIS 与屏蔽设置 IMSC.TXIM 的“与”运算结果。
2	RXMIS	R	0h	接收 FIFO 中断的屏蔽中断状态： 该字段返回接收 FIFO 中断的屏蔽中断状态，该状态是原始中断状态 RIS.RXRIS 与屏蔽设置 IMSC.RXIM 的“与”运算结果。
1	RTMIS	R	0h	接收超时中断的屏蔽中断状态： 该字段返回接收超时中断的屏蔽中断状态，该状态是原始中断状态 RIS.RTRIS 与屏蔽设置 IMSC.RTIM 的“与”运算结果。
0	RORMIS	R	0h	接收超限中断的屏蔽中断状态： 该字段返回接收溢出中断的屏蔽中断状态，该状态是原始中断状态 RIS.RORRIS 与屏蔽设置 IMSC.RORIM 的“与”运算结果。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

22.7.1.9 ICR 寄存器（偏移 = 20h）[复位 = 00000000h]

图 22-21 展示了 ICR，表 22-12 中对此进行了介绍。

返回到汇总表。

中断清除
写入 1 即可清除相应的中断。写入 0 不起作用。

图 22-21. ICR 寄存器

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

表 22-12. ICR 寄存器字段说明

位	字段	类型	复位	说明
31-2	RESERVED	R	0h	保留
1	RTIC	W	0h	清除接收超时中断： 向该字段写入 1 会清除超时中断 (RIS.RTRIS)。写入 0 无效。
0	RORIC	W	0h	清除接收超限中断： 向该字段写入 1 会清除超限错误中断 (RIS.RORRIS)。写入 0 无效。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

22.7.1.10 DMACR 寄存器（偏移 = 24h）[复位 = 00000000h]

图 22-22 展示了 DMACR，表 22-13 中对此进行了介绍。

返回到汇总表。

DMA 控制

图 22-22. DMACR 寄存器

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

表 22-13. DMACR 寄存器字段说明

位	字段	类型	复位	说明
31-2	RESERVED	R	0h	保留
1	TXDMAE	R/W	0h	发送 DMA 使能。如果该位设置为 1，则发送 FIFO 的 DMA 被使能。
0	RXDMAE	R/W	0h	接收 DMA 使能。如果该位设置为 1，则接收 FIFO 的 DMA 被使能。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

章节 23

内部集成电路 (I2C)

本章介绍内部集成电路接口。

23.1 简介 1894
23.2 方框图 1894
23.3 功能说明 1895
23.4 初始化和配置 1906
23.5 I^2C 寄存器 1906

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

23.1 简介

I²C 总线通过双线设计（串行数据线 (SDA) 和串行时钟线 (SCL)）提供双向数据传输，并可与诸如串行存储器（RAM 和 ROM）、网络器件、LCD、音调发生器等外部 I²C 器件相连。在产品开发和制造过程中，I²C 总线还可用于系统测试和诊断用途。CC13x2 和 CC26x2 器件平台包含一个 I²C 模块，能够与总线上其他 I²C 器件进行交互（包括发送和接收）。

CC13x2 和 CC26x2 器件平台包含一个 I²C 模块，其具有以下特性：

- I²C 总线上的器件可指定为主器件或从器件：
  - 支持作为主器件或从器件发送和接收数据
  - 同时支持主和从工作模式
- 四种 I²C 模式：
  - 主器件发送
  - 主器件接收
  - 从器件发送
  - 从器件接收
- 两种传输速度：标准 (100Kbps) 和快速 (400Kbps)
- 主器件和从器件中断生成：
  - 主器件会在发送或接收操作完成（或因为错误而中止）时生成中断。
  - 当主器件已传输或请求数据时，或者在检测到启动或停止条件时，从器件就会生成中断。
- 主器件具有仲裁和时钟同步、多主器件支持以及 7 位寻址模式

23.2 方框图

图 23-1 展示了 I²C 模块方框图。

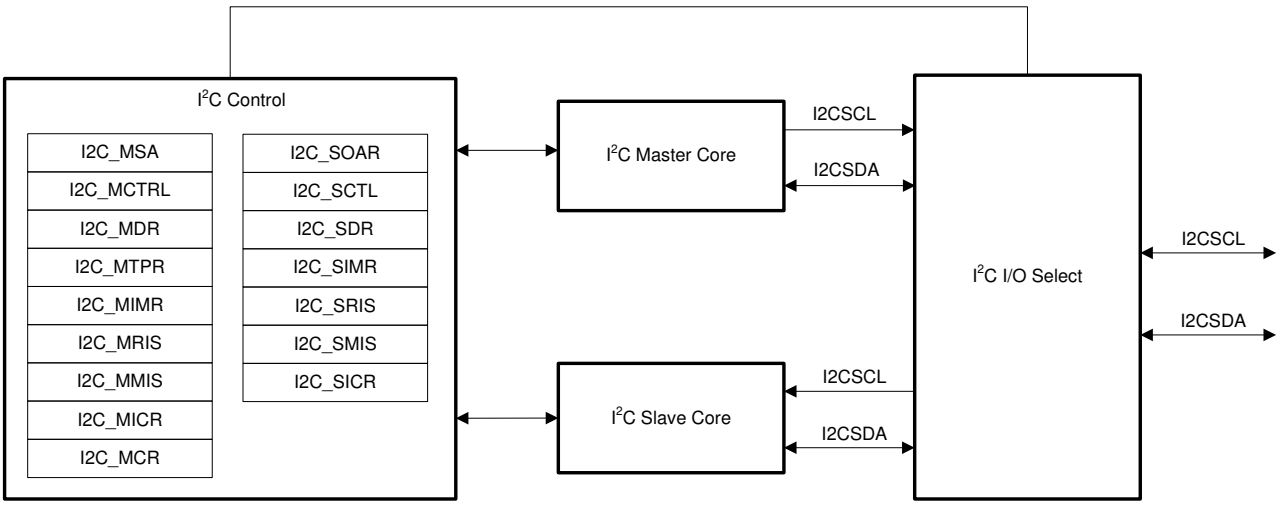


图 23-1. I²C 模块方框图

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

23.3 功能说明

I²C 模块由主功能和从功能组成。为确保正常运行，SDA 引脚必须配置为开漏信号。图 23-2 展示了典型 I²C 总线配置。

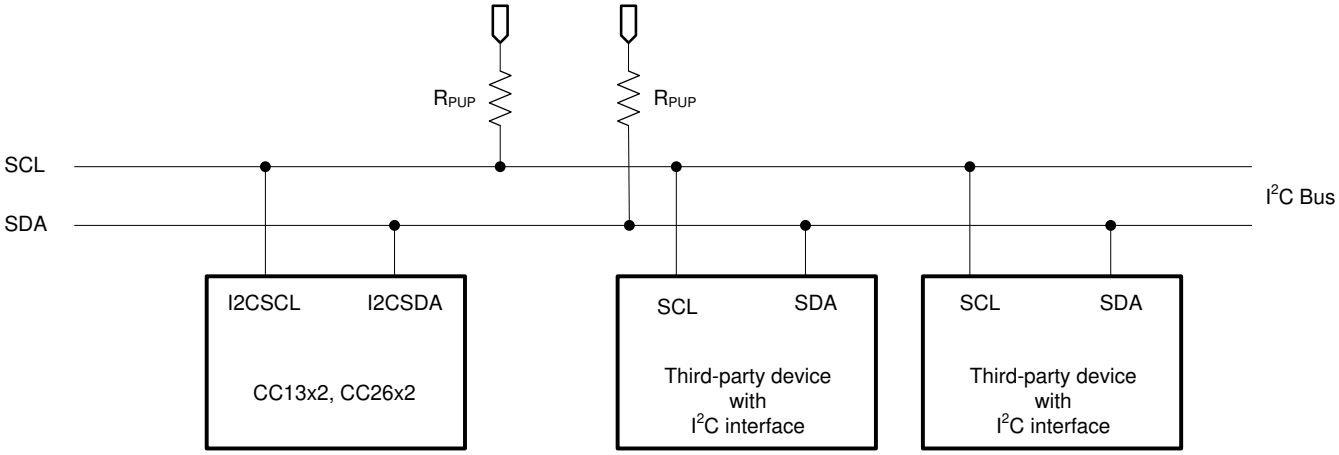


图 23-2. I²C 总线配置

23.3.1 I²C 总线功能概述

I²C 总线仅使用两个信号：SDA 和 SCL，在 CC13x2 和 CC26x2 控制器上名为 I2CSDA 和 I2CSCL。SDA 是双向串行数据线，SCL 是双向串行时钟线。当两条线都为高电平时，总线视为处于空闲状态。

I²C 总线上每个事务的长度为 9 位，其中包括 8 个数据位和 1 个确认位。每次传输的字节数（定义为有效开始和停止条件的间隔时间，如节 23.3.1.1 中所述）不受限制，每个字节后跟一个确认位，并且数据必须以 MSB 优先方式传输。当接收器无法完整接收另一个字节时，它可以将时钟线 SCL 保持为低电平，并强制传输器进入等待状态。当接收器释放时钟 SCL 时，数据传输将继续。

23.3.1.1 启动和停止条件

I²C 总线的协议定义了两种开始及结束事务的状态：启动和停止。当 SCL 为高电平时 SDA 线路上从高电平到低电平的跳变被定义为开始条件，而当 SCL 线路为高电平时 SDA 线路上从低电平到高电平的跳变被定义为停止条件。总线在启动条件之后被视为忙状态，在停止条件之后被视为空闲状态（请参阅图 23-3）。

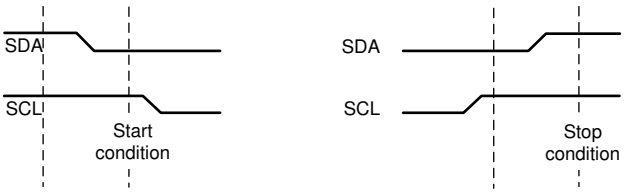


图 23-3. 启动和停止条件

STOP 位决定循环是在数据周期结束时停止还是继续直至重复的开始条件。为生成单个发送周期，I²C 主从地址 I2C:MSA 寄存器中写入所需的地址，R/S 位清零，控制寄存器 I2C:MCTRL 中写入 ACK = X（0 或 1）、STOP = 1、START = 1、RUN = 1，从而执行操作并停止。当操作完成（或由于错误而中止）时，中断引脚将变为有效状态，并且可以从 I²C 主器件数据 I2C:MDR 寄存器中读取数据。I²C 模块以主接收器模式运行时，通常会设置 ACK 位，从而使 I²C 总线控制器在每个字节接收完之后自动发送一个确认。当 I²C 总线控制器无需从发送器再进行数据传输时，必须将 ACK 位清零。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

在从模式下运行时，I²C 从器件原始中断状态 I2C:SRIS 寄存器中的 2 位指示检测到总线上的启动和停止条件，而 I²C 从器件屏蔽中断状态 I2C:SMIS 寄存器中的 2 位允许将启动和停止条件提升为控制器中断（启用了中断时）。

23.3.1.2 带有 7 位地址的数据格式

数据传输遵循图 23-4 所示的格式。在开始条件之后，发送一个从器件地址。该地址长 7 位，后面是第八位，这是一个数据方向位（I2C:MSA 寄存器中的 R/S 位）。如果 RS 位清除，则表示发送操作（发送）；如果设置了它，则表示请求数据（接收）。数据传输总是由主器件生成的停止条件终止；然而，主器件可以通过生成重复启动条件并寻址另一个从器件来发起与总线上另一个器件的通信，而无需首先生成停止条件。然后，在一次传输中可以实现各种接收和发送格式组合。

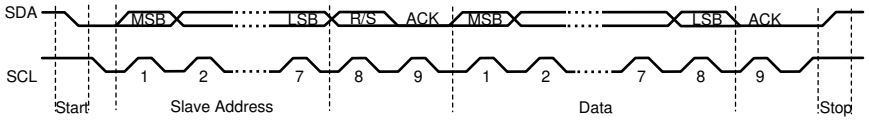


图 23-4. 使用 7 位地址完成数据传输

第一个字节的前 7 位包含从器件地址（请参阅图 23-5）。第八位决定消息的方向。第一个字节的 R/S 位置为 0 时表示主器件向选定的从器件发送（传送）数据，该位置为 1 时表示主器件接收从器件的数据。

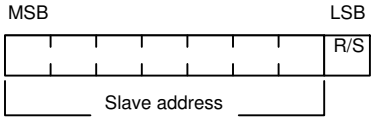


图 23-5. 第一个字节中的 R/S 位

23.3.1.3 数据有效性

SDA 线在时钟的高电平期间必须包含稳定的数据，并且数据线只能在 SCL 为低电平时改变（请参阅图 23-6）。

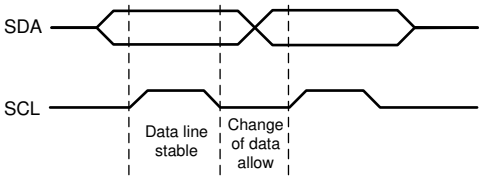


图 23-6. 在 I²C 总线上传输位期间的数据有效性

23.3.1.4 响应

所有总线事务都带有所需的确认时钟周期，该时钟周期由主器件产生。在确认周期内，发射机（主或从）释放 SDA 线路。为了响应传输，接收器必须在应答时钟周期过程中拉低 SDA。接收器在确认周期中传输的数据必须符合节 23.3.1.3 中所述的数据有效性要求。

当从接收器没有确认从器件地址时，从接收器必须将 SDA 保持为高电平，以便主器件可以产生停止条件并中止当前传输。如果主器件在传输期间充当接收器，则它负责确认从器件进行的每次传输。因为主器件控制传输中的字节数，所以它会通过不对最后一个数据字节产生确认来向从发送器发出数据结束的信号。然后，从发送机必须释放 SDA，让主发送机产生停止或重复启动条件。

23.3.1.5 仲裁

只有在总线空闲时，主器件才可以启动传输。在启动条件的最短保持时间内，两个或两个以上的主器件可以生成一个启动条件。在这些情况下，当 SCL 为高电平时，SDA 线上产生仲裁机制。在仲裁期间，竞争的主器件中，第

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

一个在 SDA 上置 1（高电平）而另一个主器件传输 0（低电平）以关闭其数据输出级并退出，直到总线再次空闲。

仲裁可以跨越多个位进行。仲裁的第一个阶段是比较地址位，如果两个控制器都试图寻址同一个器件，仲裁将继续比较数据位。

23.3.2 可用速度模式

I²C 总线可在标准模式 (100kbps) 或快速模式 (400kbps) 下运行。所选模式必须与总线上其他 I²C 器件的速度相匹配。

23.3.2.1 标准和快速模式

可以使用 I²C 主器件计时器周期 I2C:MTPR 寄存器中的值来选择标准和快速模式，使得 SCL 频率在标准模式下为 100kbps，在快速模式下为 400kbps。

I²C 时钟速率由参数 CLK_PRD、TIMER_PRD、SCL_LP 和 SCL_HP 决定，其中：

- CLK_PRD 是系统时钟周期。
- TIMER_PRD 是 I2C:MTPR 寄存器中的编程值。
- SCL_LP 为 SCL 的低相位（固定为 6）。
- SCL_HP 为 SCL 的高相位（固定为 4）。

I²C 时钟周期计算方法如下：

$$SCL_PERIOD = 2 \times (1 + TIMER_PRD) \times (SCL_LP + SCL_HP) \times CLK_PRD \quad (10)$$

例如：
CLK_PRD = 50ns
TIMER_PRD = 2
SCL_LP = 6
SCL_HP = 4

可得出 SCL 频率：
1 / SCL_PERIOD = 333 kHz

表 23-1 列出了基于各种系统时钟频率生成标准和快速模式 SCL 频率的计时器周期示例。

表 23-1. I²C 主计时器周期和速度模式的示例

系统时钟 (MHz)	计时器周期	标准模式 (kpbs)	计时器周期	快速模式 (kbps)
4	0x01	100	-	-
8	0x03	100	0x01	-
16	0x07	100	0x01	400

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

23.3.3 中断

当观察到以下情况时，I²C 可以产生中断：

- 主模块事务已完成
- 主模块仲裁丢失
- 主模块事务错误
- 主模块总线超时
- 接收到从模块事务
- 已请求从模块事务
- 检测到总线的停止状况
- 检测到总线上的启动状况

I²C 主模块和 I²C 从模块具有单独的中断信号。虽然这两个模块都可以针对多种情况生成中断，但只向中断控制器 (INTC) 发送单个中断信号。

23.3.3.1 I²C 主器件中断

当一次事务完成（传输或接收）、仲裁丢失或事务过程中发生错误时，I²C 主器件模块就会产生中断。要启用 I²C 主器件中断，软件必须置位 I²C 主器件中断屏蔽寄存器 (I2C:MIMR) 中的 IM 位。满足中断条件时，软件必须检查 I²C 主器件控制与状态寄存器 (I2C:MSTAT) 的 ERR 和 ARBLST 位，以确认上一次事务未发生错误，并确保未丢失仲裁。如果上一次事务未收到从器件的确认，则会触发错误状态。如果未检测到错误且主器件未丢失仲裁，应用即可继续传输。将 I²C 主器件中断清除寄存器 (I2C:MICR) 的 IC 位设置为 1，即可清除该中断。

如果应用不需要使用中断，原始中断状态始终可通过 I²C 主器件原始中断状态寄存器 (I2C:MRIS) 呈现。

23.3.3.2 I²C 从器件中断

从模块可以在接收或请求数据时生成中断。通过在 I²C 从中断掩码寄存器 (I2C:SIMR) 中设置来启用此中断。软件通过检查 I²C 从属控制和状态寄存器 (I2C:SSTAT) 的 RREQ 和 TREQ 位，确定模块是否必须从 I²C 从数据寄存器 (I2C:SDR) DATAIM 位写入（发送）或读取（接收）数据。如果从模块处于接收模式并且接收到传输的第一个字节，则设置 FBR 和 RREQ 位。可以通过设置 I²C 从器件中断清除寄存器 (I2C:SICR) DATAIC 位来清除中断。

此外，从模块还会在检测到启动和停止条件时生成中断。可以通过设置 I2C:SIMR 寄存器 STARTIM 和 STOPIM 位来启用这些中断；可以通过将 I2C:SICR 寄存器 STOPIC 和 STARTIC 位设置为 1 来清除这些中断。

如果应用不需要使用中断，原始中断状态始终可以通过 I²C 从器件原始中断状态寄存器 (I2C：SRIS) 看到。

23.3.4 环回操作

通过设置 I²C 主器件配置寄存器 (I2C:MCR) 的 LPBK 位，可将 I²C 模块置于内部环回模式，用于诊断或调试。在环回模式下，主器件和从器件模块的 SDA 与 SCL 信号连接在一起。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

23.3.5 命令序列流程图

本节详细介绍了在主模式和从模式下执行各种 I²C 传输类型需要执行的步骤。为此，必须在 IOC:IOCFG 寄存器中完成 SDA 和 SCL 信号配置。

23.3.5.1 I²C 主器件命令序列

图 23-7 至图 23-12 展示了可用于 I²C 主器件的命令序列。

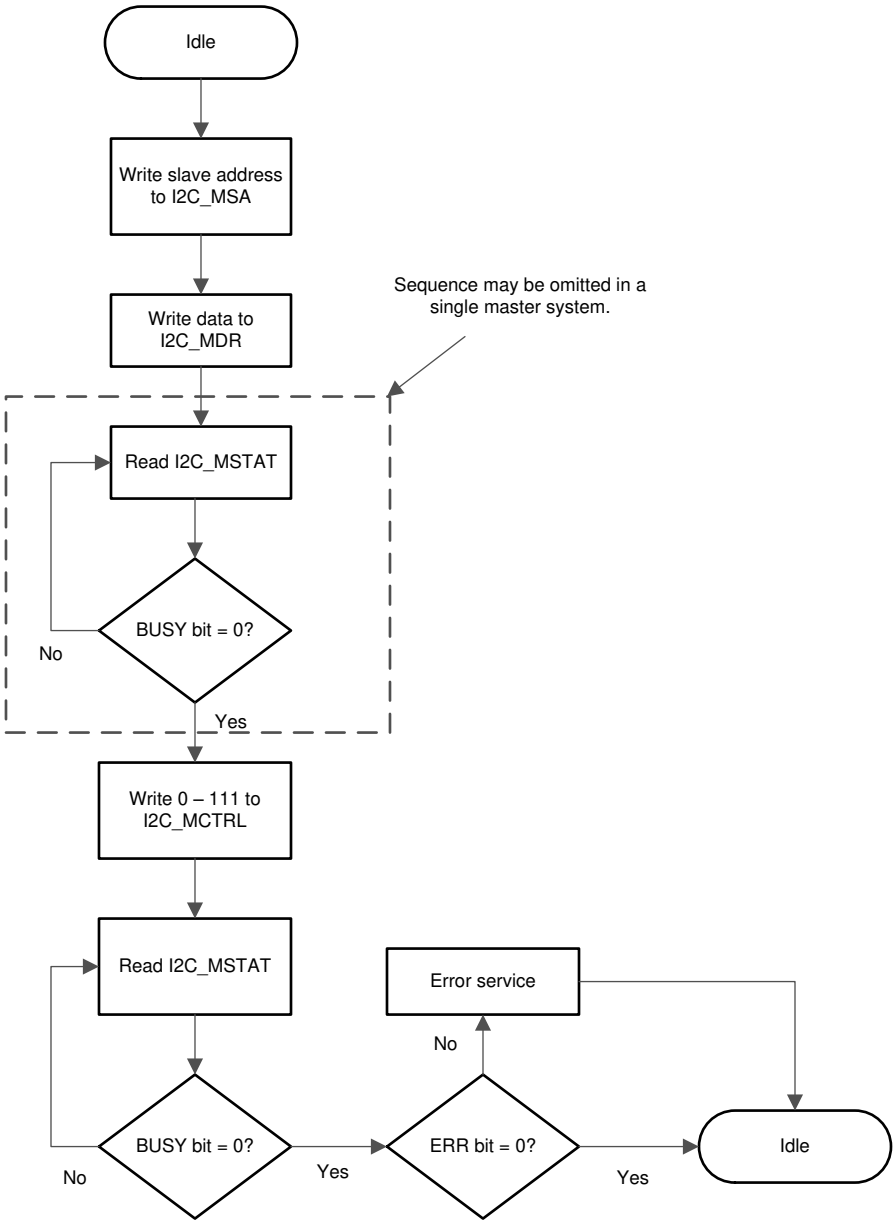


图 23-7. 主器件单次传输

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

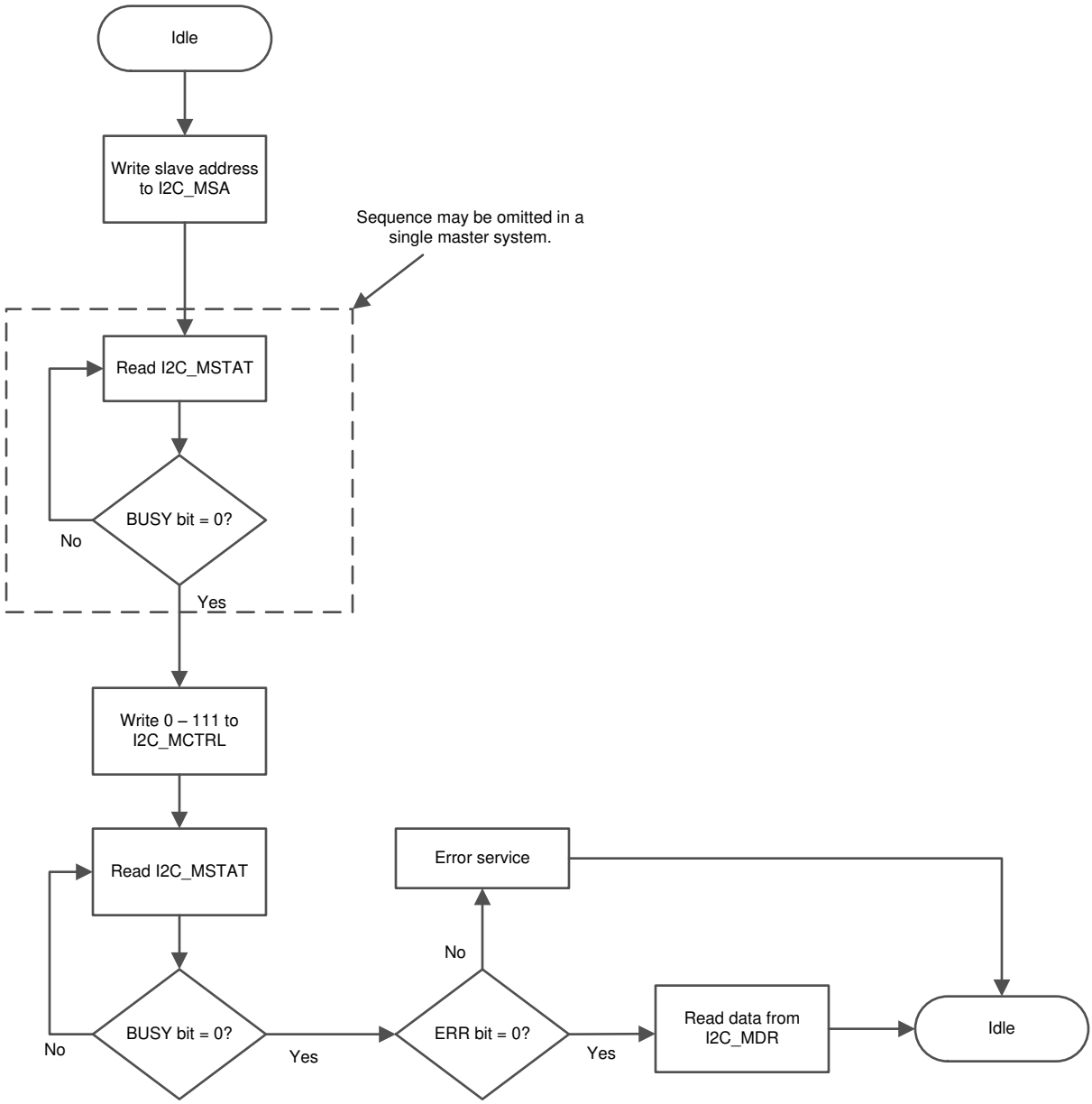


图 23-8. 主器件单次接收

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

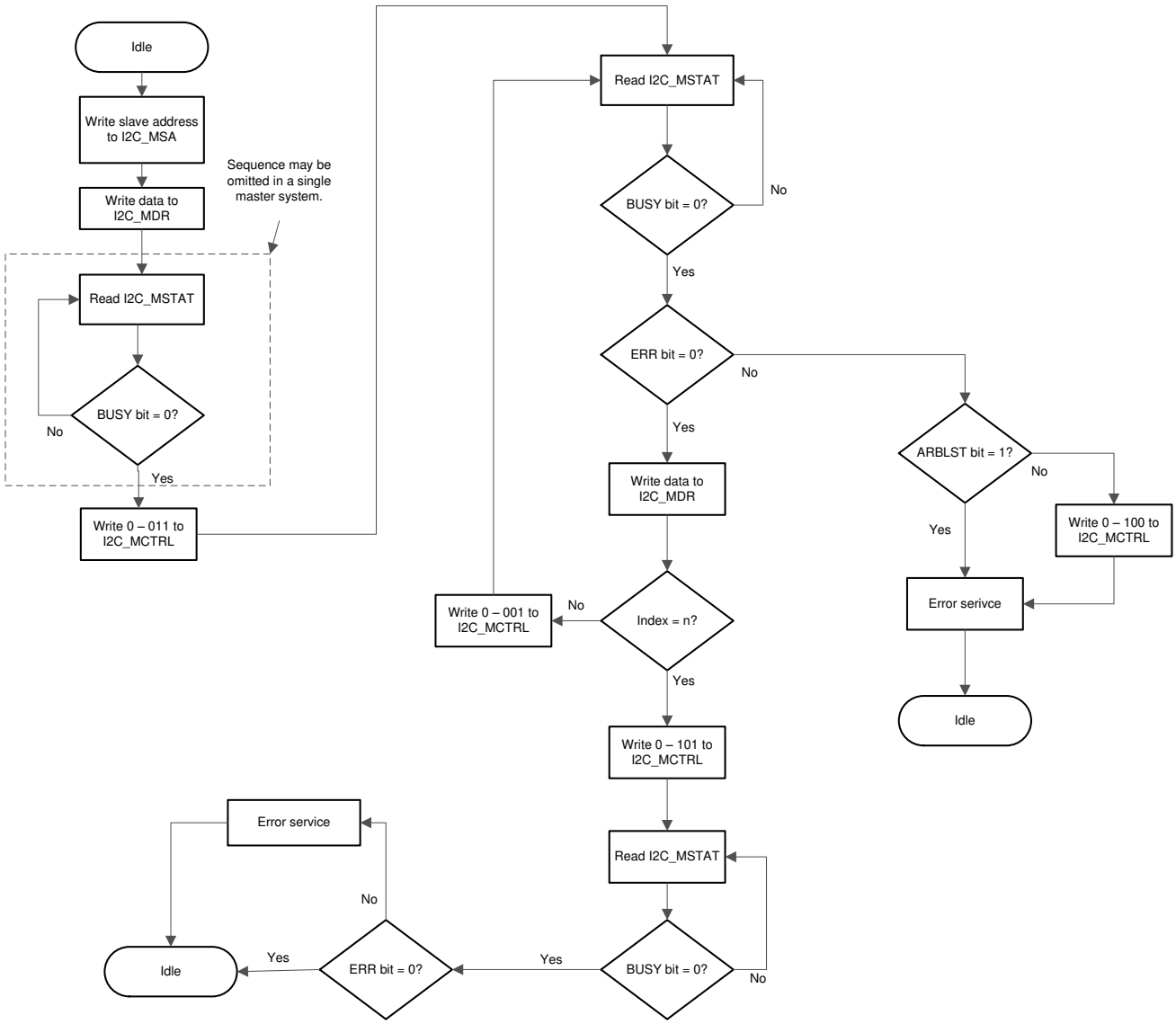


图 23-9. 带重复启动条件的主器件传输

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

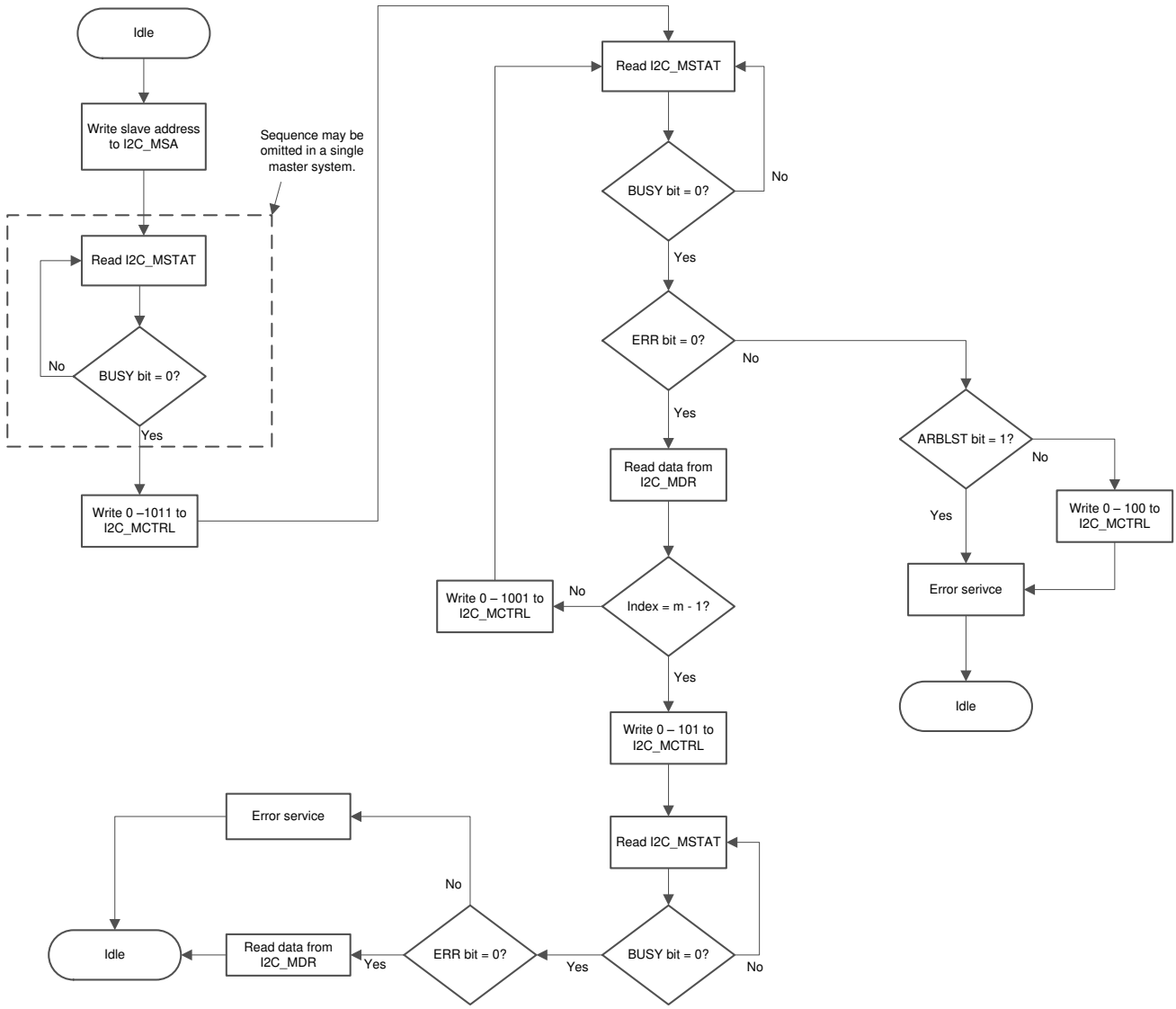


图 23-10. 带重复启动条件的主器件接收

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

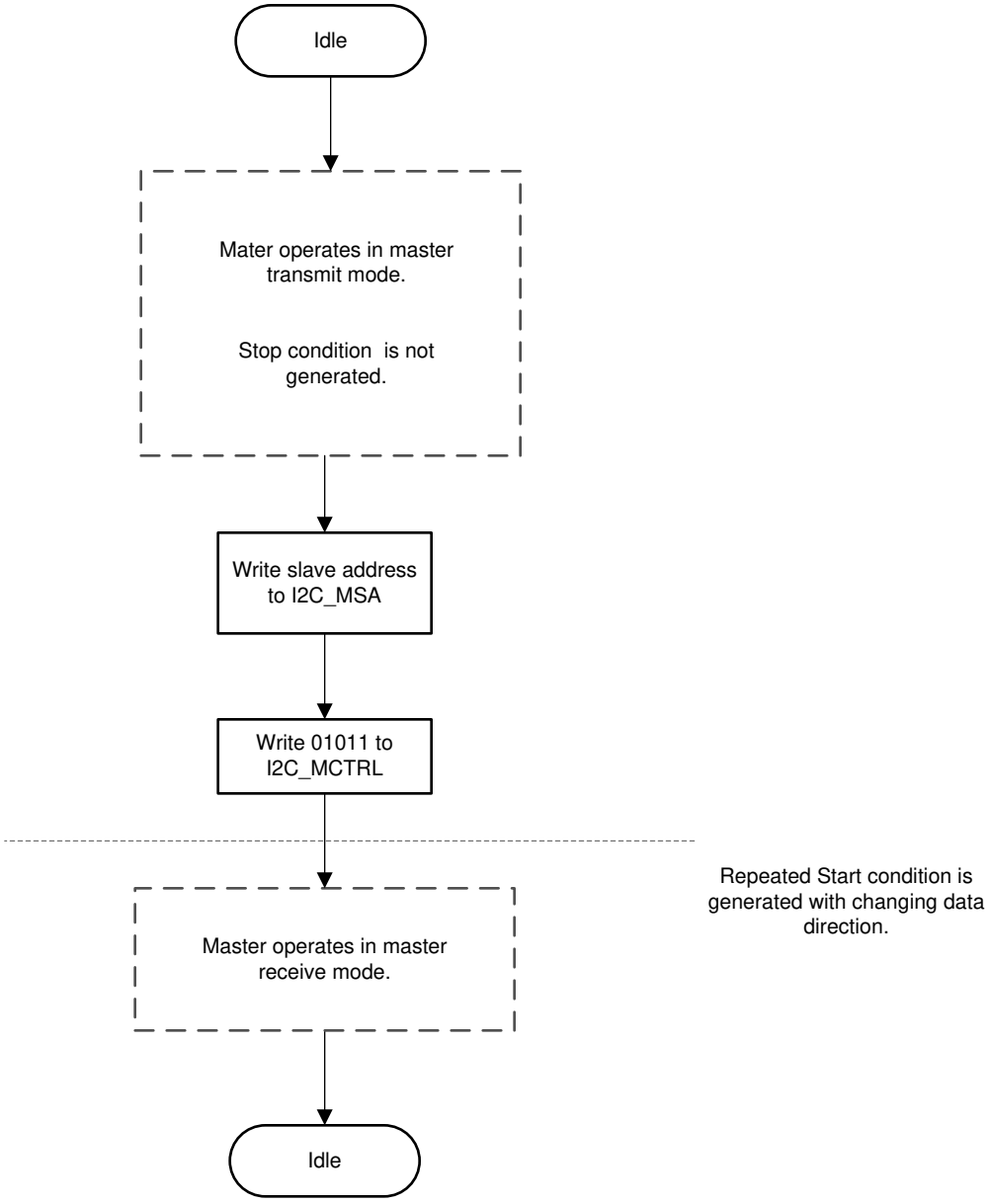


图 23-11. 主器件在带重复启动条件的传输之后进行带重复启动条件的接收

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

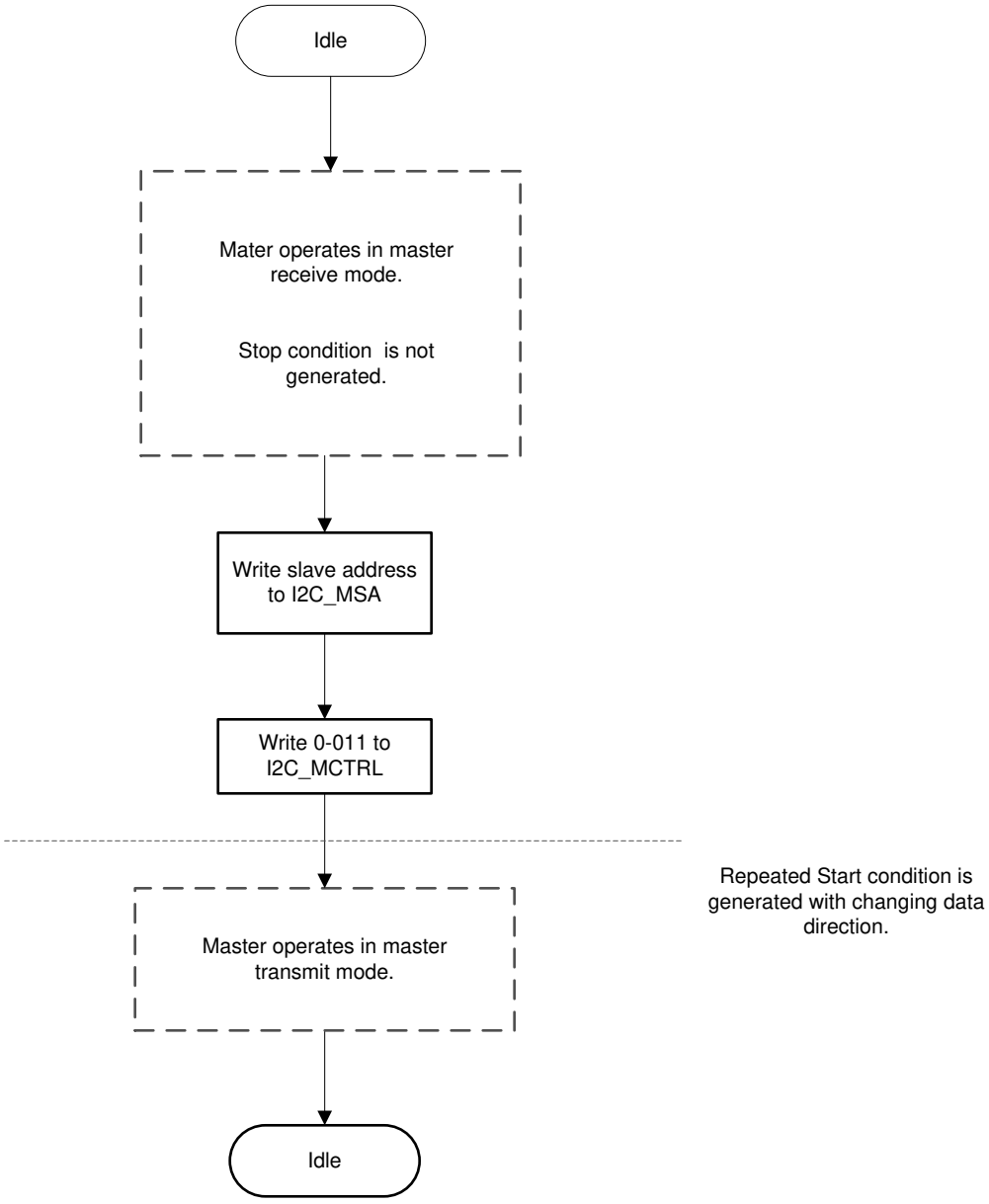


图 23-12. 主器件在带重复启动条件的接收之后进行带重复启动条件的传输

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

23.3.5.2 I²C 从器件命令序列

图 23-13 展示了可用于 I²C 从器件的命令序列。

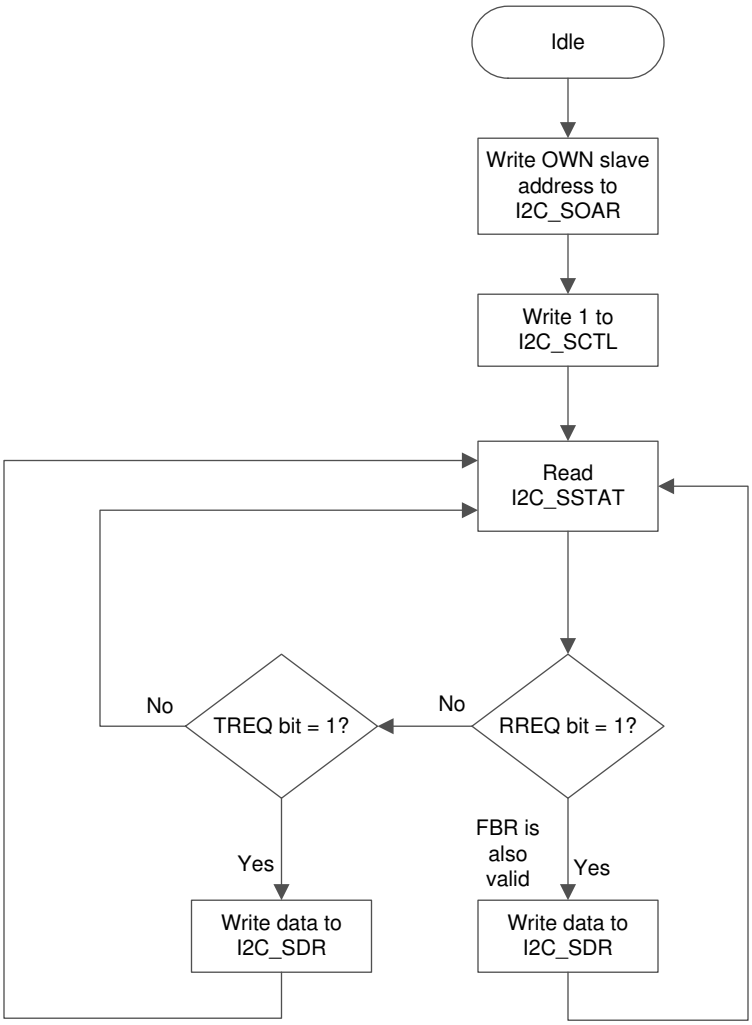


图 23-13. 从器件命令序列

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

23.4 初始化和配置

下面的示例展示了如何在系统时钟为 24MHz 的情况下，配置 I²C 模块以作为主器件发送单个字节。

1. 通过以下操作启用串行电源域并启用 PRCM 中的 I²C 模块：写入 PRCM:I2CCLKGR 寄存器、PRCM:I2CCLKGS 寄存器、PRCM:I2CCLKGDS 寄存器，或使用以下驱动程序库函数：

```
PRCMPeripheralRunEnabe(uint32_t)
```

```
PRCMPeripheralSleepEnable(uint32_t)
```

```
PRCMPeripheralDeepSLeepEnable(uint32_t)
```

   并通过写入 PRCM:CLKLOADCTL 寄存器将设置加载到时钟控制器

   或者使用 driverlib 函数 PRCMLoadSet()。
2. 配置 IOC 模块，以将 SDA 和 SCL 信号从 I/O 路由到 I²C 模块。
3. 通过向 I2C:MCR 寄存器写入值 0x0000 0010 来初始化 I²C 主器件。
4. 通过向 I2C:MTPR 寄存器写入正确的值，将所需的 SCL 时钟速度设置为 100kbps。写入 I2C:MTPR 寄存器的值表示一个 SCL 时钟周期内的系统时钟周期数。TPR 值由方程式 11、方程式 12 和方程式 13 确定。

$$TPR = [PERDMACLK / (2 \times (SCL_LP + SCL_HP) \times SCL_CLK)] - 1 \quad (11)$$

$$TPR = [24\ MHz / (2 \times (6 + 4) \times 100000)] - 1 \quad (12)$$

$$TPR = 11 \quad (13)$$

   向 I2C:MTPR 寄存器写入值 0x0000 000B。
5. 通过向 I2C:MSA 寄存器写入值 0x0000 0076，以指定主器件的从器件地址并指定下一次操作为发送，这会将从器件地址设置为 0x3B。
6. 通过向 I2C:MDR 寄存器写入所需数据，以将要发送的数据（字节）放入数据寄存器。
7. 通过向 I2C:MCTRL 寄存器写入值 0x0000 0007（停止、开始、运行），以启动从主器件到从器件的单字节数据发送。
8. 等待发送完成，即轮询 I2C:MSTAT BUSBSY 寄存器位直到其清零。
9. 检查 I2C:MSTAT ERR 寄存器位，确认发送已得到确认。

23.5 I²C 寄存器

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

23.5.1 I2C 寄存器

表 23-2 列出了 I2C 寄存器的存储器映射寄存器。表 23-2 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

表 23-2. I2C 寄存器

偏移	首字母缩写词	寄存器名称	部分
0h	SOAR	从机自身地址	SOAR 寄存器 (偏移 = 0h) [复位 = 00000000h]
4h	SSTAT	从机状态	SSTAT 寄存器 (偏移 = 4h) [复位 = 00000000h]
4h	SCTL	从机控制	SCTL 寄存器 (偏移 = 4h) [复位 = 00000000h]
8h	SDR	从机数据	SDR 寄存器 (偏移 = 8h) [复位 = 00000000h]
Ch	SIMR	从机中断屏蔽	SIMR 寄存器 (偏移 = Ch) [复位 = 00000000h]
10h	SRIS	从机原始中断状态	SRIS 寄存器 (偏移 = 10h) [复位 = 00000000h]
14h	SMIS	从机屏蔽中断状态	SMIS 寄存器 (偏移 = 14h) [复位 = 00000000h]
18h	SICR	从机中断清除	SICR 寄存器 (偏移 = 18h) [复位 = 00000000h]
800h	MSA	主控从机地址	MSA 寄存器 (偏移 = 800h) [复位 = 00000000h]
804h	MSTAT	主控状态	MSTAT 寄存器 (偏移 = 804h) [复位 = 00000020h]
804h	MCTRL	主控制	MCTRL 寄存器 (偏移 = 804h) [复位 = 00000000h]
808h	MDR	主控数据	MDR 寄存器 (偏移 = 808h) [复位 = 00000000h]
80Ch	MTPR	I2C 主控计时器周期	MTPR 寄存器 (偏移 = 80Ch) [复位 = 00000001h]
810h	MIMR	主控中断屏蔽	MIMR 寄存器 (偏移 = 810h) [复位 = 00000000h]
814h	MRIS	主控原始中断状态	MRIS 寄存器 (偏移 = 814h) [复位 = 00000000h]
818h	MMIS	主控屏蔽中断状态	MMIS 寄存器 (偏移 = 818h) [复位 = 00000000h]
81Ch	MICR	主控中断清除	MICR 寄存器 (偏移 = 81Ch) [复位 = 00000000h]
820h	MCR	主控配置	MCR 寄存器 (偏移 = 820h) [复位 = 00000000h]

复杂的位访问类型经过编码可适应小型表单元。表 23-3 展示了适用于此部分中访问类型的代码。

表 23-3. I2C 访问类型代码

访问类型	代码	说明
读取类型		
R	R	读取
写入类型		
W	W	写入
复位或默认值		

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 23-3. I2C 访问类型代码（续）

访问类型	代码	说明
-n		复位后的值或默认值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

23.5.1.1 SOAR 寄存器（偏移 = 0h）[复位 = 00000000h]

图 23-14 展示了 SOAR，表 23-4 中对此进行了介绍。

返回到汇总表。

从机自身地址
该寄存器包含七个地址位，用于在 I2C 总线上识别该 I2C 器件。

图 23-14. SOAR 寄存器

31	30	29	28	27	26	25	24	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																									OAR						
R-0h																									R/W-0h						

表 23-4. SOAR 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6-0	OAR	R/W	0h	I2C 从机自身地址 该字段指定从机地址的 a6 到 a0 位。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

23.5.1.2 SSTAT 寄存器（偏移 = 4h）[复位 = 00000000h]

图 23-15 展示了 SSTAT，表 23-5 中对此进行了介绍。

返回到汇总表。

从机状态
注意：该寄存器与 SCTL 共享地址，这意味着该寄存器在写入时用作控制寄存器，在读取时用作状态寄存器。

图 23-15. SSTAT 寄存器

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

表 23-5. SSTAT 寄存器字段说明

位	字段	类型	复位	说明
31-3	RESERVED	R	0h	保留
2	FBR	R	0h	已接收第一个字节 0：尚未接收第一个字节。 1：已接收从机自身地址后的第一个字节。 该位仅在 RREQ 位置位时有效，并在从 SDR 寄存器读取数据后自动清除。 注意：该位不用于从机发送操作。
1	TREQ	R	0h	发送请求 0：没有待处理的发送请求。 1：I2C 控制器已被寻址为从机发送器，并且正在使用时钟延展技术来延迟主控，直到数据被写入 SDR 寄存器。
0	RREQ	R	0h	接收请求 0：没有待处理的接收数据 1：I2C 控制器有来自 I2C 主控的待处理接收数据，并且正在使用时钟延展技术来延迟主控，直到已从 SDR 寄存器读取数据。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

23.5.1.3 SCTL 寄存器（偏移 = 4h）[复位 = 00000000h]

图 23-16 展示了 SCTL，表 23-6 中对此进行了介绍。

返回到汇总表。

从机控制
注意：该寄存器与 SSTAT 共享地址，这意味着该寄存器在写入时用作控制寄存器，在读取时用作状态寄存器。

图 23-16. SCTL 寄存器

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

表 23-6. SCTL 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	W	0h	软件不应依赖保留字段的值。写入任何其他值都可能导致未定义的行为。
0	DA	W	0h	器件激活 0：禁用 I2C 从机操作 1：使能 I2C 从机操作

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

23.5.1.4 SDR 寄存器（偏移 = 8h）[复位 = 00000000h]

图 23-17 展示了 SDR，表 23-7 中对此进行了介绍。

返回到汇总表。

从机数据
该寄存器包含处于从机发送状态时待发送的数据，以及处于从机接收状态时接收的数据。

图 23-17. SDR 寄存器

31	30	29	28	27	26	25	24	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							

表 23-7. SDR 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-0	DATA	R/W	0h	待传输数据 该字段包含在从机接收或发送操作期间待传输的数据。写入时，寄存器数据用作发送数据。读取时，该寄存器返回最近接收的数据。 数据将存储直到下一次更新：即通过系统写入进行发送，或通过外部主控进行接收。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

23.5.1.5 SIMR 寄存器（偏移 = Ch）[复位 = 00000000h]

图 23-18 展示了 SIMR，表 23-8 中对此进行了介绍。

返回到汇总表。

从机中断屏蔽
该寄存器控制原始中断是否提升为控制器中断。

图 23-18. SIMR 寄存器

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

表 23-8. SIMR 寄存器字段说明

位	字段	类型	复位	说明
31-3	RESERVED	R	0h	保留
2	STOPIM	R/W	0h	停止条件中断屏蔽 0：SRIS.STOPRIS 中断被抑制，不会发送到中断控制器。 1：SRIS.STOPRIS 中断被使能并发送到中断控制器。 0h = 禁用中断 1h = 使能中断
1	STARTIM	R/W	0h	启动条件中断屏蔽 0：SRIS.STARTRIS 中断被抑制，不会发送到中断控制器。 1：SRIS.STARTRIS 中断被使能并发送到中断控制器。 0h = 禁用中断 1h = 使能中断
0	DATAIM	R/W	0h	数据中断屏蔽 0：SRIS.DATARIS 中断被抑制，不会发送到中断控制器。 1：SRIS.DATARIS 中断被使能并发送到中断控制器。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

23.5.1.6 SRIS 寄存器（偏移 = 10h）[复位 = 00000000h]

图 23-19 展示了 SRIS，表 23-9 中对此进行了介绍。

返回到汇总表。

从机原始中断状态
该寄存器显示未屏蔽的中断状态。

图 23-19. SRIS 寄存器

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

表 23-9. SRIS 寄存器字段说明

位	字段	类型	复位	说明
31-3	RESERVED	R	0h	保留
2	STOPRIS	R	0h	停止条件原始中断状态 0：无中断 1：停止条件中断待处理。 可通过向 SICR.STOPIC 写入 1 清除该位。
1	STARTRIS	R	0h	启动条件原始中断状态 0：无中断 1：启动条件中断待处理。 可通过向 SICR.STARTIC 写入 1 清除该位。
0	DATARIS	R	0h	数据原始中断状态 0：无中断 1：已接收数据或已请求数据的中断待处理。 可通过向 SICR.DATAIC 写入 1 清除该位。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

23.5.1.7 SMIS 寄存器（偏移 = 14h）[复位 = 00000000h]

图 23-20 展示了 SMIS，表 23-10 中对此进行了介绍。

返回到汇总表。

从机屏蔽中断状态
该寄存器显示哪个中断处于激活状态（基于 SRIS 和 SIMR 的结果）。

图 23-20. SMIS 寄存器

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

表 23-10. SMIS 寄存器字段说明

位	字段	类型	复位	说明
31-3	RESERVED	R	0h	保留
2	STOPMIS	R	0h	停止条件屏蔽中断状态 0：未发生中断或中断被屏蔽/禁用。 1：未屏蔽的停止条件中断待处理。 可通过向 SICR.STOPIC 写入 1 清除该位。
1	STARTMIS	R	0h	启动条件屏蔽中断状态 0：未发生中断或中断被屏蔽/禁用。 1：未屏蔽的启动条件中断待处理。 可通过向 SICR.STARTIC 写入 1 清除该位。
0	DATAMIS	R	0h	数据屏蔽中断状态 0：未发生中断或中断被屏蔽/禁用。 1：未屏蔽的已接收数据或已请求数据的中断待处理。 可通过向 SICR.DATAIC 写入 1 清除该位。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

23.5.1.8 SICR 寄存器（偏移 = 18h）[复位 = 00000000h]

图 23-21 展示了 SICR，表 23-11 中对此进行了介绍。

返回到汇总表。

从机中断清除
该寄存器清除原始中断 SRIS。

图 23-21. SICR 寄存器

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

表 23-11. SICR 寄存器字段说明

位	字段	类型	复位	说明
31-3	RESERVED	R	0h	保留
2	STOPIC	W	0h	停止条件中断清除 向该位写入 1 会清除 SRIS.STOPRIS 和 SMIS.STOPMIS。
1	STARTIC	W	0h	启动条件中断清除 向该位写入 1 会清除 SRIS.STARTRIS 和 SMIS.STARTMIS。
0	DATAIC	W	0h	数据中断清除 向该位写入 1 会清除 SRIS.DATARIS 和 SMIS.DATAMIS。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

23.5.1.9 MSA 寄存器（偏移 = 800h）[复位 = 00000000h]

图 23-22 展示了 MSA，表 23-12 中对此进行了介绍。

返回到汇总表。

主控从机地址
该寄存器包含主控要访问的从机的七个地址位 (a6-a0)，以及确定下一个操作是接收还是发送的一个 RS 位。

图 23-22. MSA 寄存器

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								SA							RS
R-0h								R/W-0h							R/W-0h

表 23-12. MSA 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-1	SA	R/W	0h	I2C 主控从机地址 定义在主控模式下，进行事务处理寻址哪个从机
0	RS	R/W	0h	接收或发送 该位域指定下一个操作是来自寻址从机 SA 的接收（高电平）还是发送（低电平）。 0h = 向从机发送数据 1h = 接收来自从机的数据

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

23.5.1.10 MSTAT 寄存器（偏移 = 804h）[复位 = 00000020h]

图 23-23 展示了 MSTAT，表 23-13 中对此进行了介绍。

返回到汇总表。

主控状态

图 23-23. MSTAT 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED	BUSBSY	空闲	ARBLST	DATACK_N	ADRACK_N	ERR	BUSY
R-0h	R-0h	R-1h	R-0h	R-0h	R-0h	R-0h	R-0h

表 23-13. MSTAT 寄存器字段说明

位	字段	类型	复位	说明
31-7	RESERVED	R	0h	保留
6	BUSBSY	R	0h	总线忙 0：I2C 总线空闲。 1：I2C 总线忙。 该位根据 MCTRL.START 和 MCTRL.STOP 条件而变化。
5	空闲	R	1h	I2C 空闲 0：I2C 控制器未空闲。 1：I2C 控制器空闲。
4	ARBLST	R	0h	仲裁失败 0：I2C 控制器仲裁成功。 1：I2C 控制器仲裁失败。
3	DATACK_N	R	0h	数据未应答 0：发送的数据已应答。 1：发送的数据未应答。
2	ADRACK_N	R	0h	地址未应答 0：发送的地址已应答。 1：发送的地址未应答。
1	ERR	R	0h	错误 0：在上一次操作中未检测到错误。 1：在上一次操作中发生错误。
0	BUSY	R	0h	I2C 忙 0：控制器空闲。 1：控制器正忙。 当该位域置位时，其他状态位无效。 注意：在通过 MCTRL 寄存器启动 I2C 主控操作后，I2C 控制器需要四个 SYSBUS 时钟周期才能将 BUSY 状态置为有效。 因此，对 MCTRL 寄存器进行编程后，应用需要等待四个 SYSBUS 时钟周期，然后才能通过 MSTAT 寄存器发出控制器状态查询。 任何提前查询都将导致报告错误的状态。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

23.5.1.11 MCTRL 寄存器（偏移 = 804h）[复位 = 00000000h]

图 23-24 展示了 MCTRL，表 23-14 中对此进行了介绍。

返回到汇总表。

主控制
该寄存器在读取时访问状态位，在写入时访问控制位。读取时，状态寄存器指示 I2C 总线控制器的状态，如 MSTAT 中所述。写入时，控制寄存器配置 I2C 控制器操作。
为生成单个发送周期，向 I2C 主控从机地址 (MSA) 寄存器写入所需的地址，清除 MSA.RS 位，并向该寄存器写入：
* ACK=X（0 或 1）、
* STOP=1、
* START=1、
* RUN=1
，从而执行操作并停止。
当操作完成（或由于错误而中止）时，中断将激活，并可从 MDR 寄存器中读取数据。

图 23-24. MCTRL 寄存器

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

表 23-14. MCTRL 寄存器字段说明

位	字段	类型	复位	说明
31-4	RESERVED	R	0h	保留
3	ACK	W	0h	数据应答使能 0：主控不自动应答接收的数据字节。 1：主控自动应答接收的数据字节。 当 I2C 总线控制器无需接收通过从机发送器发送的更多数据时，必须清除该位域。 0h = 禁用应答 1h = 使能应答
2	STOP	W	0h	该位决定周期是在数据周期结束时停止还是继续直至重复的 START 条件。 0：控制器不生成停止条件。 1：控制器生成停止条件。 0h = 禁用 STOP 1h = 启用 STOP
1	启动	W	0h	此位域生成启动或重复启动条件。 0：控制器不生成启动条件。 1：控制器生成启动条件。 0h = 禁用 START 1h = 启用 START

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 23-14. MCTRL 寄存器字段说明（续）

位	字段	类型	复位	说明
0	RUN	W	0h	I2C 主控使能 0：主控被禁用。 1：主控使能以发送或接收数据。 0h = 禁用主控 1h = 使能主控

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

23.5.1.12 MDR 寄存器（偏移 = 808h）[复位 = 00000000h]

图 23-25 展示了 MDR，表 23-15 中对此进行了介绍。

返回到汇总表。

主控数据
该寄存器包含处于主控发送状态时待发送的数据，以及处于主控接收状态时接收的数据。

图 23-25. MDR 寄存器

31	30	29	28	27	26	25	24	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							

表 23-15. MDR 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-0	DATA	R/W	0h	读取时：返回上一个 RX 数据 写入时：数据在 TX 事务期间传输

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

23.5.1.13 MTPR 寄存器（偏移 = 80Ch）[复位 = 00000001h]

图 23-26 展示了 MTPR，表 23-16 中对此进行了介绍。

返回到汇总表。

I2C 主控计时器周期
该寄存器指定 SCL 时钟的周期。

图 23-26. MTPR 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
TPR_7	TPR						
R/W-0h	R/W-1h						

表 23-16. MTPR 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7	TPR_7	R/W	0h	必须置 0 才能设置 TPR。如果置 1，则对 TPR 的写入操作将被忽略。
6-0	TPR	R/W	1h	SCL 时钟周期 该字段指定 SCL 时钟的周期。 SCL_PRD = 2*(1+TPR)*(SCL_LP + SCL_HP)*CLK_PRD 其中： SCL_PRD 是 SCL 线路周期（I2C 时钟）。 TPR 是计时器周期寄存器值（范围为 1 至 127） SCL_LP 是 SCL 低电平周期（固定为 6）。 SCL_HP 是 SCL 高电平周期（固定为 4）。 CLK_PRD 是以 ns 为单位的系统时钟周期。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

23.5.1.14 MIMR 寄存器（偏移 = 810h）[复位 = 00000000h]

图 23-27 展示了 MIMR，表 23-17 中对此进行了介绍。

返回到汇总表。

主控中断屏蔽
该寄存器控制原始中断是否提升为控制器中断。

图 23-27. MIMR 寄存器

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															IM
R-0h															R/W-0h

表 23-17. MIMR 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	IM	R/W	0h	中断屏蔽 0：MRIS.RIS 中断被抑制，不会发送到中断控制器。 1：当 MRIS.RIS 置位时，会向中断控制器发送主控中断。 0h = 禁用中断 1h = 使能中断

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

23.5.1.15 MRIS 寄存器（偏移 = 814h）[复位 = 00000000h]

图 23-28 展示了 MRIS，表 23-18 中对此进行了介绍。

返回到汇总表。

主控原始中断状态
该寄存器显示未屏蔽的中断状态。

图 23-28. MRIS 寄存器

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															RIS
R-0h															R-0h

表 23-18. MRIS 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	RIS	R	0h	原始中断状态 0：无中断 1：主控中断待处理。 可通过向 MICR.IC 位写入 1 清除该位。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

23.5.1.16 MMIS 寄存器（偏移 = 818h）[复位 = 00000000h]

图 23-29 展示了 MMIS，表 23-19 中对此进行了介绍。

返回到汇总表。

主控屏蔽中断状态
该寄存器显示哪个中断处于激活状态（基于 MRIS 和 MIMR 的结果）。

图 23-29. MMIS 寄存器

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															MIS
R-0h															R-0h

表 23-19. MMIS 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	MIS	R	0h	已屏蔽中断状态 0：未发生或屏蔽中断。 1：主控中断待处理。 可通过向 MICR.IC 位写入 1 清除该位。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

23.5.1.17 MICR 寄存器（偏移 = 81Ch）[复位 = 00000000h]

图 23-30 展示了 MICR，表 23-20 中对此进行了介绍。

返回到汇总表。

主控中断清除
该寄存器清除原始中断和屏蔽中断。

图 23-30. MICR 寄存器

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															IC
R-0h															W-0h

表 23-20. MICR 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	IC	W	0h	中断清除 向该位写入 1 会清除 MRIS.RIS 和 MMIS.MIS。 读取该寄存器不会返回任何有意义的数据。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

23.5.1.18 MCR 寄存器（偏移 = 820h）[复位 = 00000000h]

图 23-31 展示了 MCR，表 23-21 中对此进行了介绍。

返回到汇总表。

主控配置
该寄存器配置模式（主控或从机）并设置测试模式环回接口。

图 23-31. MCR 寄存器

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

表 23-21. MCR 寄存器字段说明

位	字段	类型	复位	说明
31-6	RESERVED	R	0h	保留
5	SFE	R/W	0h	I2C 从机功能使能 0h = 禁用从机模式。 1h = 使能从机模式。
4	MFE	R/W	0h	I2C 主控功能使能 0h = 禁用主控模式。 1h = 使能主控模式。
3-1	RESERVED	R	0h	保留
0	LPBK	R/W	0h	I2C 环回 0：正常运行 1：环回操作（测试模式） 0h = 禁用测试模式 1h = 使能测试模式

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

章节 24

IC 间音频 (I^2S)

本章介绍了 IC 间音频 (I^2S) 模块。

24.1 简介 1929
24.2 方框图 1930
24.3 信号描述 1931
24.4 功能说明 1931
24.5 存储器接口 1937
24.6 采样时间戳发生器 1940
24.7 错误检测 1943
24.8 用途 1943
24.9 I^2S 寄存器 1943

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

24.1 简介

I²S 模块提供一个标准化的串行接口，用于在 CC13x2 和 CC26x2 器件平台与外部音频器件（如编解码器、DAC 或 ADC）之间传输音频采样。

I²S 模块还可以从器件（如数字麦克风）接收脉冲密度调制 (PDM) 信号，然后在软件中将其转换为脉冲编码调制 (PCM)。

I²S 模块具有以下特性：

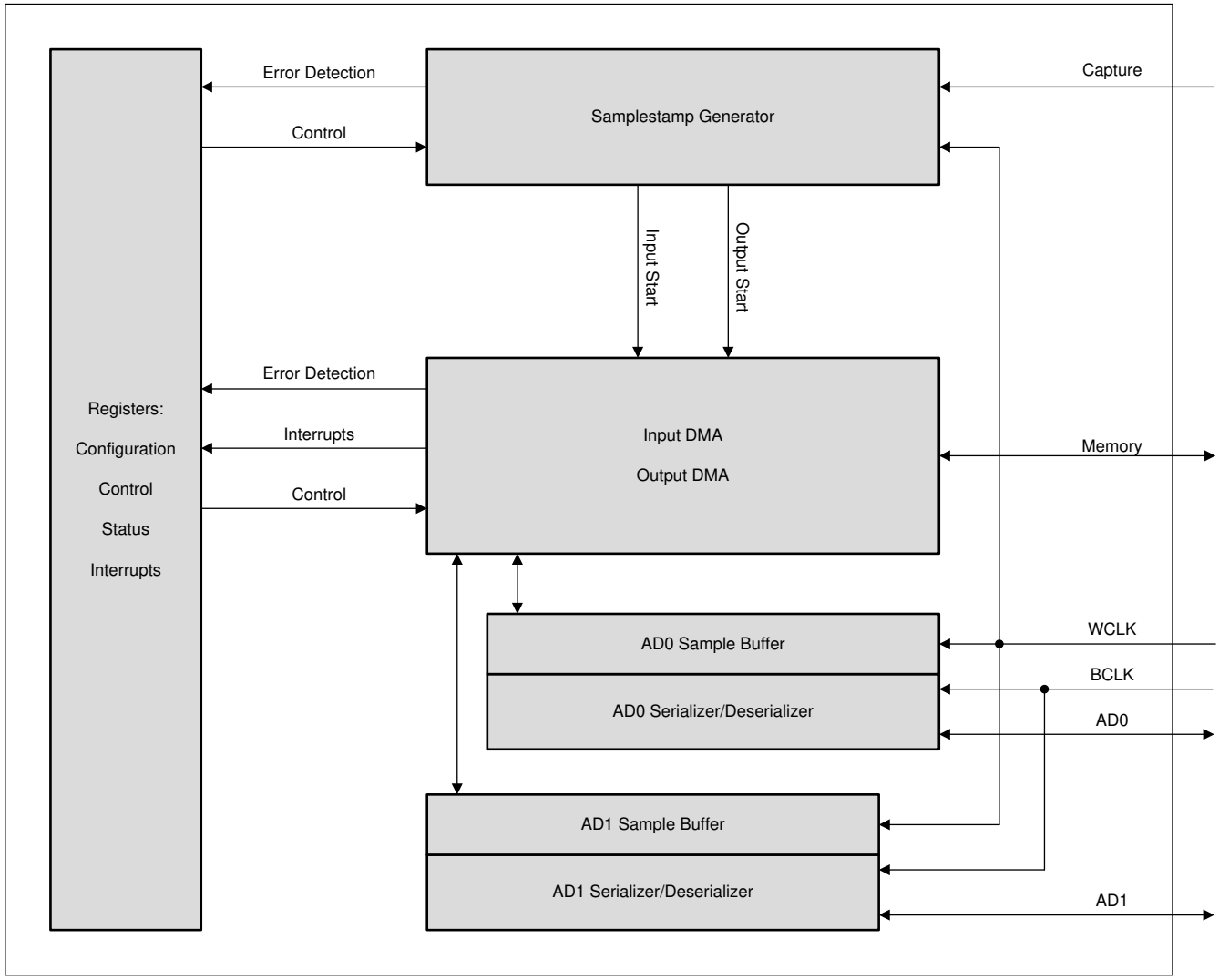
- 音频时钟信号由 PRCM 模块在内部生成，或由另一个器件在外部提供。
- 一个或两个数据引脚，可独立配置为输入或输出
- I²S、左对齐 (LJF) 和右对齐 (RJF) 串行接口格式，每个数据引脚最多支持两个音频通道
- DSP 串行接口格式，每个数据引脚最多支持八个音频通道
- 最多 24 位采样字长，如果不匹配，则进行截断或补零：
  - 串行接口：每个样本字 8 至 24 位，首先传输最高有效位 (MSB)
  - 内存：每个样本字 16 或 24 位，以小端字节序存储
- 具有双缓冲指针的 DMA
- 针对 DMA 和音频时钟信号完整性的错误检测
- 采样时间戳发生器，当音频采样由一个 CC13x2 或 CC26x2 器件上的 I²S 模块生成，并由另一个 CC13x2 或 CC26x2 器件上的 I²S 模块消耗时，可保持恒定的音频延迟。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

24.2 方框图

图 24-1 所示为 I²S 模块的方框图。

图 24-1. I²S 模块简化方框图



Copyright © 2017, Texas Instruments Incorporated

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

24.3 信号描述

串行音频接口由两个或三个时钟信号以及一个或两个数据信号组成，具体取决于 I²S 模块的使用方式。时钟信号可以由内部（由 CC13x2 和 CC26x2 器件平台的 PRCM 模块）或外部（由音频器件或其他时钟源）生成。

表 24-1 列出了串行音频引脚接口的详细信息。

表 24-1. 串行音频引脚接口

CC13x2 和 CC26x2 引脚	功能和方向	说明
MCLK（可选）	主时钟输出	用于外部音频器件中的采样转换的主时钟。CC13x2 或 CC26x2 器件平台内部不使用该信号，因此该信号永远不会用作 CC13x2 或 CC26x2 器件平台的输入。某些外部音频器件不需要该信号。
BCLK	位时钟输入/输出	WCLK 及 AD0、AD1 信号的位时钟。使用外部时钟源时，该信号为输入；使用内部时钟源时，该信号为输出。
WCLK	字时钟输入/输出	采样成帧信号，定义音频采样频率以及 AD0 和 AD1 串行数据流中的采样字边界。 WCLK 频率与音频采样频率相同。使用外部时钟源时，该信号为输入；使用内部时钟源时，该信号为输出。
AD0 AD1	串行数据输入/输出	负责传输音频采样字的串行数据信号。每个引脚可独立配置为输入、输出或不使用。所有引脚使用相同的接口格式（例如 LJF）。 AD0 和 AD1 引脚在后文统称为 *ADx 引脚*。

CC13x2 和 CC26x2 器件平台无法动态地将 ADx 引脚置于三态。因此，TDM 模式支持仅由外部音频器件驱动这些信号的 ADx 输入引脚，但不支持 ADx 输出引脚。

24.4 功能说明

24.4.1 依赖项

在访问任何 I²S 模块寄存器之前，启用 I²S 模块的系统时钟：

- 对于系统 CPU 运行模式：PRCM:I2SCLKGR.CLK_EN = 1（加载了 PRCM:CLKLOADCTL）
- 对于系统 CPU 睡眠模式：PRCM:I2SCLKGS.CLK_EN = 1（加载了 PRCM:CLKLOADCTL）
- 对于系统 CPU 深度睡眠模式：PRCM:I2SCLKGDS.CLK_EN = 1（加载了 PRCM:CLKLOADCTL）

24.4.1.1 系统 CPU 深度睡眠模式

如果系统总线关闭，则 I²S DMA 将失去对 RAM 和闪存的访问。

如果满足以下所有条件，系统总线将关闭：

- PRCM:PDCTL1.CPU_ON = 0
- PRCM:PDCTL1.VIMS_MODE = 0
- PRCM:SECDMACLKGDS.DMA_CLK_EN = 0
- PRCM:SECDMACLKGDS.CRYPTO_CLK_EN = 0
- RFCORE 不请求访问 BUS。
- 系统 CPU 处于深度睡眠模式。

为了实现最低电流消耗，请设置 PRCM:SECDMACLKGDS.DMA_CLK_EN = 1（加载 PRCM:CLKLOADCTL）以使系统总线为 I²S DMA 保持启用状态。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

24.4.2 引脚配置

I2S:AIFDIRCFG 寄存器配置每个 ADx 信号是输入、输出还是未使用。

必须通过配置相应的 IOC:IOCFGn 寄存器，将每个使用的 I²S 信号映射到物理 I/O 引脚 DIOn。

24.4.3 串行格式配置

WCLK 和 ADx 信号在 BCLK 的一个边沿上更新，在相反边沿上采样。

根据配置的串行接口格式，在 ADx 引脚上传输的采样字与 WCLK 信号对齐。采样字的第一个 WCLK 边沿可以是上升沿或下降沿，具体取决于配置的串行接口格式。

从一次音频采样（一个或多个通道）的第一个 WCLK 边沿到下一次音频采样的第一个 WCLK 边沿之间的时间段称为*一帧*。一帧包含一个或两个阶段。一个阶段分为以下区间：

- 数据延迟（可选）：该阶段的第一个 WCLK 边沿与（第一个）音频通道数据传输的 MSB 之间的 BCLK 周期数
- 字：在 ADx 引脚上传输采样字的 BCLK 周期
  - 对于单阶段，背靠背传输 1 到 8 个采样字。
  - 对于双阶段，传输一个采样字。采样字的 LSB 可以延伸到下一个阶段的数据延迟区间。
- 空闲（可选）：字区间与下一个阶段之间的 BCLK 周期

串行接口上的一个采样字可以包含 8 到 24 位。

24.4.4 I²S

I²S 是一种双相格式，其 WCLK 占空比为 50%，每个采样字 MSB 的起始位置与 WCLK 的每个边沿延迟一个 BCLK 周期后对齐。对于任何给定的帧，当 WCLK 为低电平时，首先传输左通道；然后当 WCLK 为高电平时，再传输右通道。图 24-2 展示了 I²S 串行格式。

数据会在 BCLK 的上升沿采样，在 BCLK 的下降沿更新。

I²S 格式的独特之处在于，CC13x2 和 CC26x2 器件平台能够自动检测每个 WCLK 周期内的 BCLK 周期数。因此，I²S 支持来自外部音频时钟源的任何 BCLK 速率，同时支持可变采样字长：

- 如果配置的采样字长大于每个 WCLK 周期的位数，则采样字将被截断。
- 如果配置的采样字长小于每个 WCLK 周期的位数，则采样字将被补零。

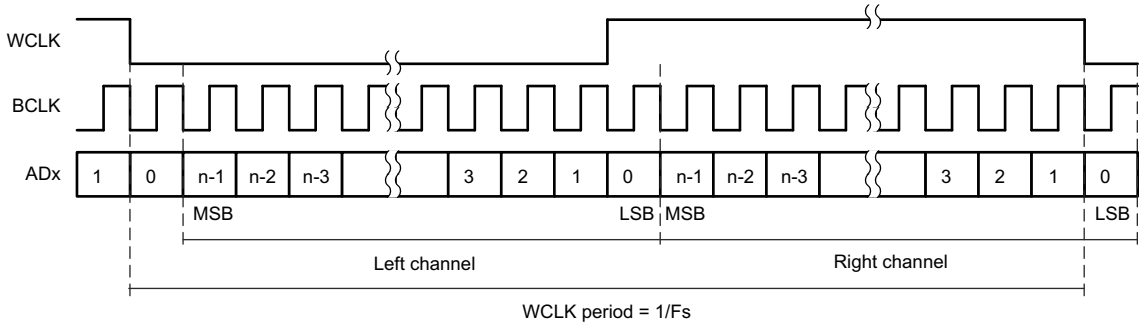


图 24-2. I²S 串行格式

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

24.4.4.1 寄存器配置

寄存器配置如下：

- I2S:AIFWCLKSRC.WCLK_INV = 1
- I2S:AIFFMTCFG.DUAL_PHASE = 1
- I2S:AIFFMTCFG.SMPL_EDGE = 1
- I2S:AIFFMTCFG.WORD_LEN = 每个样本字的最大位数
- I2S:AIFFMTCFG.DATA_DELAY = 1

24.4.5 左对齐 (LJF)

LJF 是一种双相格式，具有 50% 的 WCLK 占空比，且每个采样字的 MSB 开头与 WCLK 的每个边沿对齐。对于任意给定帧，左通道首先在 WCLK 为高电平时传输，右通道随后在 WCLK 为低电平时传输。

数据在 BCLK 的上升沿采样，在 BCLK 的下降沿更新。图 24-3 展示了 LJF 串行格式。

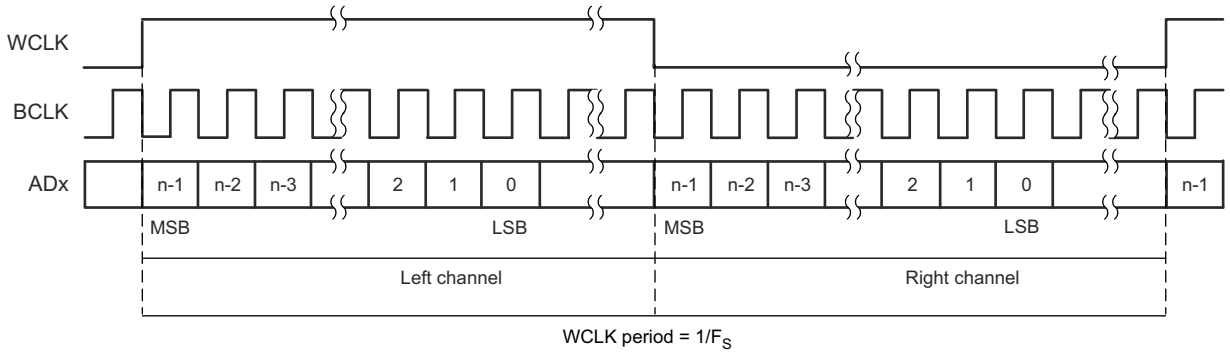


图 24-3. LJF 串行格式

24.4.5.1 寄存器配置

寄存器配置如下：

- I2S:AIFWCLKSRC.WCLK_INV = 0
- I2S:AIFFMTCFG.DUAL_PHASE = 1
- I2S:AIFFMTCFG.SMPL_EDGE = 1
- I2S：AIFFMTCFG.WORD_LEN = 每个样本字的最大位数
- I2S:AIFFMTCFG.DATA_DELAY = 0

WORD_LEN 必须等于或小于每相位的 BCLK 周期数。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

24.4.6 右对齐 (RJF)

RJF 是一种双相格式，具有 50% 的 WCLK 占空比，且每个采样字的 LSB 末尾与 WCLK 的每个边沿对齐。对于任意给定帧，左通道首先在 WCLK 为高电平时传输，右通道随后在 WCLK 为低电平时传输。

数据在 BCLK 的上升沿采样，在 BCLK 的下降沿更新。图 24-4 展示了 RJF 串行格式。

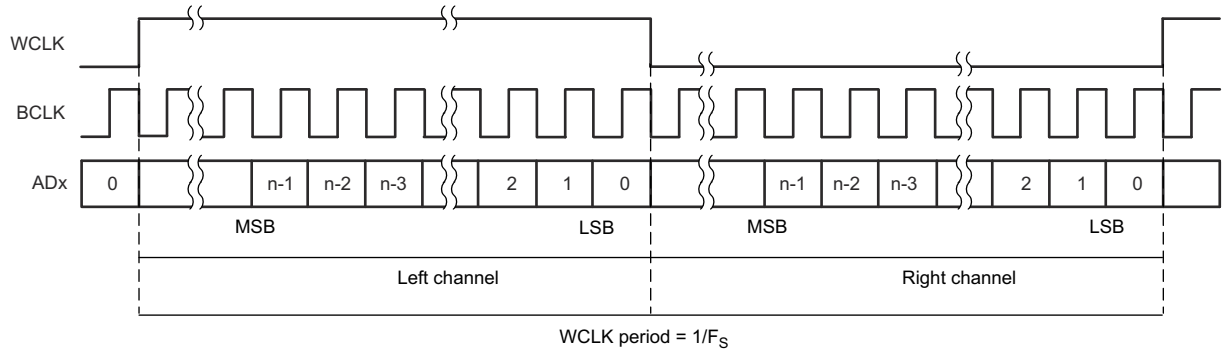


图 24-4. RJF 串行格式

24.4.6.1 寄存器配置

寄存器配置如下：

- I2S:AIFWCLKSRC.WCLK_INV = 0
- I2S:AIFFMTCFG.DUAL_PHASE = 1
- I2S:AIFFMTCFG.SMPL_EDGE = 1
- I2S:AIFFMTCFG.WORD_LEN = 每个样本字的确切位数
- I2S:AIFFMTCFG.DATA_DELAY = 每相位的 BCLK 周期数减去 I2S:AIFFMTCFG.WORD_LEN 的值

DATA_DELAY + WORD_LEN 必须等于或小于每相位的 BCLK 周期数。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

24.4.7 DSP

DSP 是一种单相位格式，其中 WCLK 在一个 BCLK 周期内为高电平，并且第一个采样字的 MSB 通常与此 WCLK 脉冲对齐，或者在下一个 BCLK 周期中跟随。随后各个音频通道的采样字连续传输，然后进入空闲周期，直到下一个相位或帧开始。

数据在 BCLK 的下降沿采样，在 BCLK 的上升沿更新。图 24-5 显示了具有零数据延迟的 DSP 串行格式。

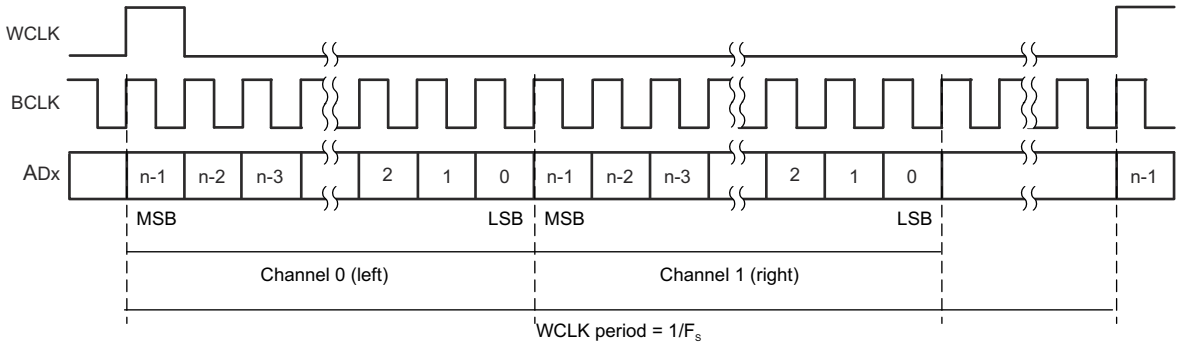


图 24-5. DSP 串行格式（零数据延迟）

24.4.7.1 寄存器配置

寄存器配置如下：

- I2S:AIFWCLKSRC.WCLK_INV = 0
- I2S:AIFFMTCFG.DUAL_PHASE = 0
- I2S:AIFFMTCFG.SMPL_EDGE = 0
- I2S:AIFFMTCFG.WORD_LEN = 每个样本字的确切位数
- I2S:AIFFMTCFG.DATA_DELAY = 0 或 1

DATA_DELAY + (WORD_LEN × *channel count*) 必须等于或小于每个相位的 BCLK 周期数。

*通道数*由 I2S:AIFWMASK0 和 I2S:AIFWMASK1 寄存器中设置的 MSB 确定。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

24.4.8 时钟配置

音频时钟信号 MCLK、BCLK 和 WCLK 可由 PRCM 模块在内部生成，也可由外部时钟源生成。

内部生成的音频时钟信号可能并不适用于所有应用，原因如下：

- 抖动性能
- 时钟配置仅支持可以从 48MHz 分频的频率。
- 无法通过调谐来保持恒定音频延迟的频率

当使用内部生成的音频时钟时，48MHz 高频系统时钟 SCLK_HF 必须始终源自 XOSC_HF。完成此推导是为了减少抖动，并避免将 SCLK_HF 源从 RCOSC_HF 切换到 XOSC_HF（缺少时钟周期和频率漂移）产生的副作用。

24.4.8.1 内部音频时钟源

必须为 I²S 模块及 PRCM 模块选择内部音频时钟源：

- I2S:AIFWCLKSRC = 2
- PRCM:I2SBCLKSEL.SRC = 1

PRCM:I2SCLKCTL.SMPL_ON_POSEDGE 寄存器的设置指定了应在哪个 BCLK 边沿对 WCLK 信号进行采样。此设置必须等于 I2S:AIFFMTCFG.SMPL_EDGE 寄存器的设置。

MCLK、BCLK、WCLK 频率及 WCLK 占空比配置如下：

- MCLK 频率 = 48 MHz / PRCM:I2SMCLKDIV.MDIV
- BCLK 频率 = 48 MHz / PRCM:I2SBCLKDIV.BDIV
- 对于 WCLK，配置取决于占空比（PRCM:I2SWCLKDIV.WDIV 称为 WDIV）：
  - 单相（DSP 格式）：PRCM:I2SCLKCTL.WCLK_PHASE = 0
    - WCLK 在 1 个 BCLK 周期内为高电平，在 WDIV[9:0] (1 to 1023) BCLK 周期内为低电平。
    - WCLK 频率 = BCLK 频率 / (1 + PRCM:I2SWCLKDIV.WDIV[9:0])
- 双相（I²S，LJF 和 RJF 格式）：PRCM:I2SCLKCTL.WCLK_PHASE = 1
  - WCLK 在 WDIV[9:0]（1 至 1023）BCLK 周期内为高电平，在 WDIV[9:0]（1 至 1023）BCLK 周期内为低电平。
  - WCLK 频率= BCLK 频率 / (2 × WDIV[9:0])
- 用户自定义：PRCM:I2SCLKCTL.WCLK_PHASE = 2
  - WCLK 在 WDIV[7:0]（1 至 255）BCLK 周期内为高电平，在 WDIV[15:8]（1 至 255）BCLK 周期内为低电平。
  - WCLK 频率 = BCLK 频率 / (WDIV[7:0] + WDIV[15:8])

必须通过设置 PRCM:I2SCLKCTL.EN = 1 来启用时钟信号 MCLK、BCLK 和 WCLK 的信号生成。当 PRCM:I2SCLKCTL.EN = 0 时，MCLK、BCLK 和 WCLK 信号为静态低电平。

24.4.8.2 外部音频时钟源

必须为 I²S 模块及 PRCM 模块选择外部音频时钟源：

- I2S:AIFWCLKSRC = 1
- PRCM:I2SBCLKSEL.SRC = 0

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

24.5 存储器接口

集成式直接存储器存取控制器 (DMA) 可独立处理输入样本（从一个或两个 ADx 引脚到 RAM）和输出样本（从 RAM 或闪存到一个或两个 ADx 引脚）。

每个 ADx 引脚都有一个移位寄存器及一个采样字缓冲器。DMA 在接收下一个样本字时将输入样本字存储到存储器中，并在传输最后一个加载的样本字时从存储器加载输出样本字。

DMA 在存储器块上运行。当 DMA 处理一个存储器块时，软件必须针对输入样本将下一个存储器块的起始地址写入 I2S:AIFINPTRNEXT，针对输出样本则写入 I2S:AIFOUTPTRNEXT。

24.5.1 采样字长

存储器中的采样字长（16 位或 24 位）独立于串行接口上的采样字长（8 至 24 位）进行配置。当目标位宽小于源位宽时，采样字会被截断；当目标位宽大于源位宽时，会进行补零。

I2S:AIFFMTCFG.MEM_LEN_24 字段用于配置存储器中的采样字是 16 位还是 24 位：

- 0：每个 16 位采样字通过一次 16 位传输移入或移出存储器。写入 I2S:AIFINPTRNEXT 和 I2S:AIFOUTPTRNEXT 寄存器的 DMA 指针必须为半字对齐。
- 1：每个 24 位采样字通过一次锁定的总线传输（包含一次 16 位传输和一次 8 位传输）移入或移出存储器。写入 I2S:AIFINPTRNEXT 和 I2S:AIFOUTPTRNEXT 寄存器的 DMA 指针无需对齐到任何存储器大小。

24.5.2 通道映射

对于每个 ADx 引脚，相应的 I2S:AIFWMASKx 寄存器决定了存储器中存在哪些采样字：

- 当 I2S:AIFFMTCFG.DUAL_PHASE = 0 时（DSP 格式），对于每一帧：
  - 输入：I2S:AIFWMASKx.MASK 寄存器决定是否将通道存储到存储器。
  - 输出：I2S:AIFWMASKx.MASK 寄存器决定是否从存储器获取通道。对于排除的通道，ADx 输出为低电平。
- 当 I2S:AIFFMTCFG.DUAL_PHASE = 1 时（I²、LJF 和 RJF 格式），对于每一帧：
  - 单声道： I2S:AIFWMASKx.MASK = 0x01
    - 输入：左 (0) 通道存储到存储器。
    - 输出：从存储器获取左 (0) 通道，并在右通道重复播放。
  - 立体声： I2S:AIFWMASKx.MASK = 0x03
    - 输入：左 (0) 和右 (1) 通道均存储到存储器。
    - 输出：(0) 和右 (1) 通道均从存储器获取。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

24.5.3 存储器中的采样存储

采样字以小端字节序存储在存储器中，即最低有效字节 (LSByte) 存储在较低的字节地址，最高有效字节 (MSByte) 存储在较高的字节地址。

如果两个 ADx 引脚均配置为输入或均配置为输出，则每个音频通道的采样字首先存储 AD0，最后存储 AD1。

图 24-6、图 24-7、图 24-8、图 24-9 和图 24-10 展示了典型用例下的采样存储示例（F = 帧索引，C = 通道索引，P = ADx 引脚索引）。

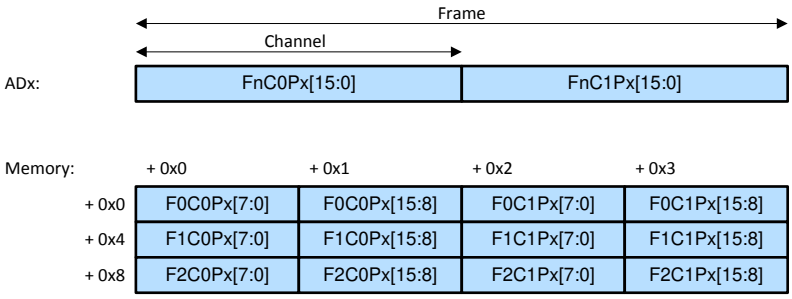


图 24-6. 一个 ADx 引脚上的 16 位单声道 I²S、LJF 和 RJF 格式，存储器中显示六个帧

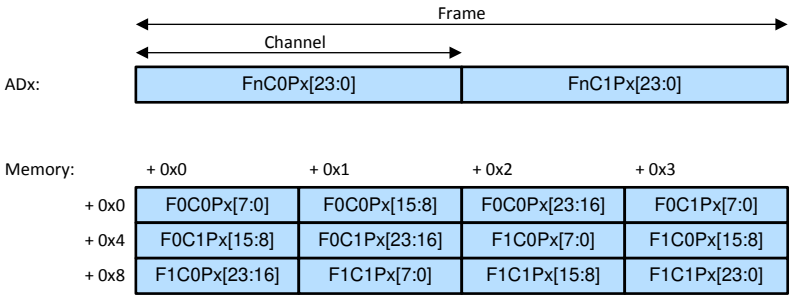


图 24-7. 一个 ADx 引脚上的 16 位立体声 I²S、LJF 和 RJF 格式，存储器中显示三个帧

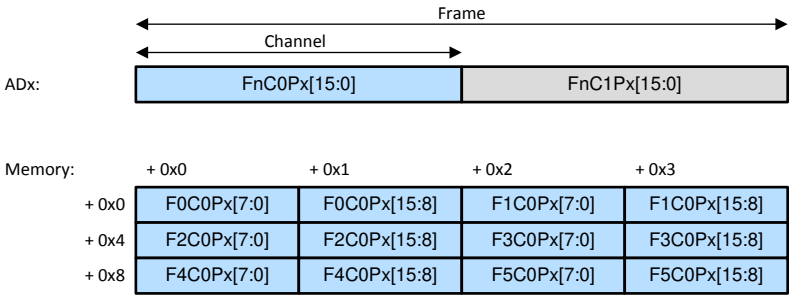


图 24-8. 一个 ADx 引脚上的 24 位立体声 I²S、LJF 和 RJF 格式，存储器中显示两个帧

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

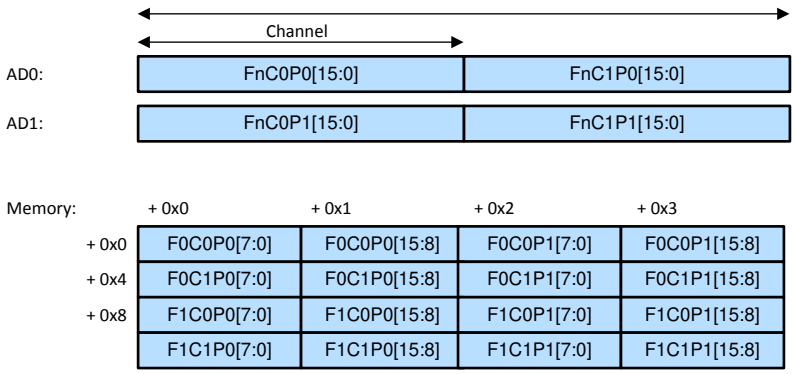


图 24-9. AD0 和 AD1 引脚上的 16 位 I²S 格式，存储器中显示两个帧

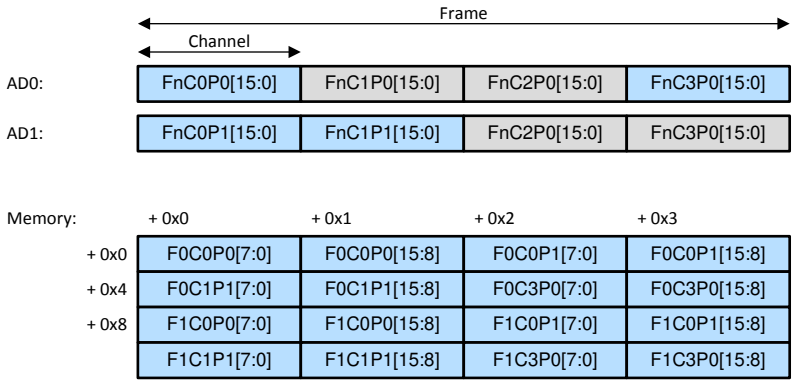


图 24-10. AD0 和 AD1 引脚上的 16 位 DSP 格式，存储器中显示两个帧

24.5.4 DMA 运行

DMA 在存储器块上运行。每个输入和输出 DMA 存储器块分别包含 I2S:AIFDMACFG.END_FRAME_IDX + 1 帧的输入和输出采样字。

向 I2S:AIFDMACFG.END_FRAME_IDX 写入非零值会初始化 DMA 并准备好启动。向 I2S:AIFDMACFG.END_FRAME_IDX 写入零会禁用 DMA 并重置串行音频接口。

如果使用输入 ADx 引脚，软件必须将输入 DMA 的存储器块起始地址写入 I2S:AIFINPTRNEXT。可以在 I2S:AIFINPTR 中观察当前输入 DMA 存储器位置。

如果使用输出 ADx 引脚，软件必须将输出 DMA 的存储器块起始地址写入 I2S:AIFOUTPTRNEXT。可以在 I2S:AIFOUTPTR 中观察当前输出 DMA 存储器位置。

此写入操作或 DMA 运行允许软件在内存中实现采样块环形缓冲区，其中包含任意数量的块用于输入和输出采样。

24.5.4.1 启动

在向 I2S:AIFDMACFG.END_FRAME_IDX 寄存器写入非零值之前，必须完成所有其他音频接口相关寄存器的配置（引脚、串行格式、时钟、采样字长和通道映射）。

为了针对启动而准备输入和输出 DMA，软件必须预加载要使用的第一个和第二个 DMA 指针，并激活 DMA：

- 将要使用的第一个存储器块起始地址写入 I2S:AIFINPTRNEXT 和（或）I2S:AIFOUTPTRNEXT

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

- 设置 I2S:AIFDMACFG.END_FRAME_IDX = 每块的帧数减 1。
  - 这会将 I2S:AIFINPTRNEXT 加载到 I2S:AIFINPTR，将 I2S:AIFOUTPTRNEXT 加载到 I2S:AIFOUTPTR，随后输出 DMA 将立即预取前两个音频通道的采样字。
- 将要使用的第二个存储器块起始地址写入 I2S:AIFINPTRNEXT 和（或）I2S:AIFOUTPTRNEXT。

然后，输入和输出 DMA 由采样时间戳发生器独立启动，如节 24.6.2 所述。

24.5.4.2 运行

为了维持 DMA 运行，每次存储块完成时，软件都必须提供新的存储块起始地址。

一个块完成后，会发生以下情况：

- 对于输入存储器接口块：
  - I2S:AIFINPTR = I2S:AIFINPTRNEXT
  - I2S:AIFINPTRNEXT = 0x0000 0000
  - I2S:IRQFLAGS.AIF_DMA_IN 设置为生成 I2S_IRQ 中断。
- 对于输出存储器接口块：
- - I2S:AIFOUTPTR = I2S:AIFOUTPTRNEXT
  - I2S:AIFOUTPTRNEXT = 0x0000 0000
  - I2S:IRQFLAGS.AIF_DMA_OUT 设置为生成 I2S_IRQ 中断。

为了处理此操作，软件必须轮询 I2S:AIFINPTRNEXT 和（或）I2S:AIFOUTPTRNEXT 寄存器是否为零，或使用 I2S:IRQFLAGS.AIF_DMA_IN 和（或）I2S:IRQFLAGS.AIF_DMA_OUT 中断请求。

在运行的块完成之前，软件必须将新存储器块起始地址写入 I2S:AIFINPTRNEXT 和（或）I2S:AIFOUTPTRNEXT。如果运行块在 I2S:AIFINPTRNEXT 和（或）I2S:AIFOUTPTRNEXT 为零时完成，则受影响的 DMA 通道将停止并设置 I2S:IRQFLAGS.PTR_ERR。

24.5.4.3 关断

在 DMA 关闭之前，所有外部输出音频器件（例如 DAC）都应静音，或者输出 ADx 引脚上应传输静默信号。

在可能仍在进行 DMA 存储器传输时，不得停止 DMA。

当使用内部音频时钟源或不会意外停止的外部音频时钟源时，软件应按以下过程停止 DMA：

- 停止写入 I2S:AIFINPTRNEXT 和/或 I2S:AIFOUTPTRNEXT 寄存器。
- 可选：等待 I2S:IRQFLAGS.PTR_ERR 发生。
- 等待 I2S:AIFINPTRNEXT 和 I2S:AIFOUTPTRNEXT 变为零。
- 写入 I2S:AIFDMACFG.END_FRAME_IDX = 0。

当使用可能意外停止的外部音频时钟源时，软件应按以下过程停止 DMA：

- 停止写入 I2S:AIFINPTRNEXT 和（或）I2S:AIFOUTPTRNEXT 寄存器。
- 停止外部音频时钟源。
- 等待 I2S:IRQFLAGS.WCLK_TIMEOUT 发生，或等待 I2S:AIFINPTRNEXT 和 I2S:AIFOUTPTRNEXT 变为零，以先发生者为准。
- 写入 I2S:AIFDMACFG.END_FRAME_IDX = 0。

24.6 采样时间戳发生器

采样时间戳发生器用于启动输入和输出 DMA 操作。采样时间戳发生器还用于在无线网络上同步 I²S 模块，从而实现正确且固定的音频延迟。通过无线网络进行同步是一项可选特性，可以旁路掉。

当 I2S:STMPCTL.STMP_EN = 1 时，采样时间戳发生器启用并运行。当软件写入 I2S:STMPCTL.STMP_EN = 0 时，会复位计数器和捕获寄存器。

图 24-11 展示了采样时间戳发生器的简化方框图：

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

- 图的下半部分与输入和输出 DMA 的启动有关。
- 图的上半部分与用于 I²S 模块同步的 RFCORE 事件捕获有关。

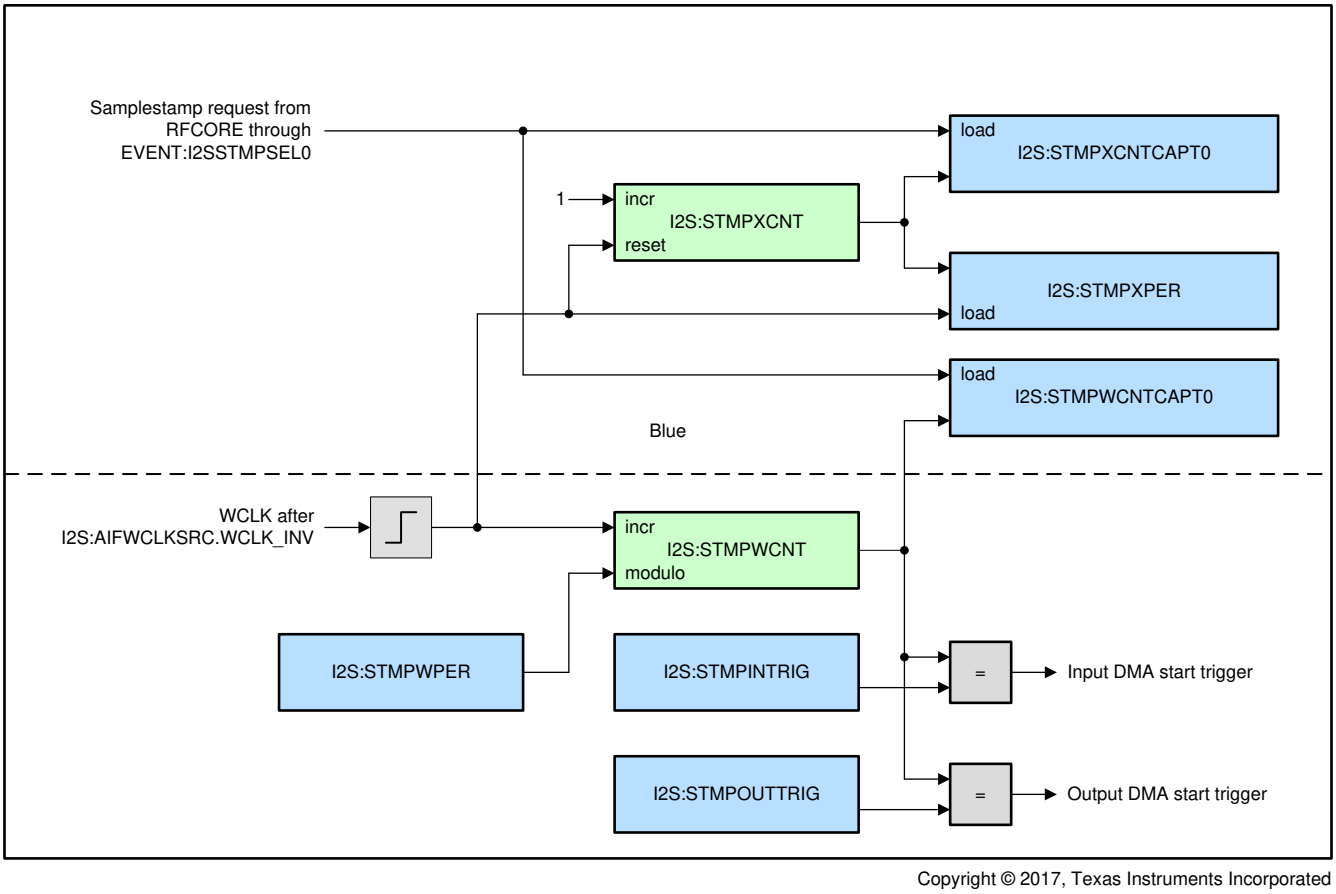


图 24-11. 采样时间戳发生器

24.6.1 样片戳计数器

采样时间戳发生器在每个帧内对帧（WCLK 周期）及 48MHz 时钟周期（晶体振荡器周期）进行计数：

- I2S:STMPWCNT 在每帧的第一个 WCLK 边沿递增，模块周期值 I2S:STMPWPER。
- I2S:STMPXCNT 在每帧的第一个 WCLK 边沿复位为 0，然后针对每个 48MHz 时钟周期递增 1。读取 I2S:STMPWCNT 会锁存 I2S:STMPXCNT 的读取值。

软件可以通过向 I2S:STMPWSET 写入绝对值或向 I2S:STMPWADD 写入相对值来修改 I2S:STMPWCNT 的值。

24.6.2 启动触发

I2S:STMPINTRIG 和 I2S:STMPOUTTRIG 寄存器包含 I2S:STMPWCNT 比较值，分别用于启动输入和输出 DMA：

- 当 I2S:STMPWCNT 等于 I2S:STMPINTRIG 时，输入 DMA 在下一帧开始将采样字存储到存储器中：(I2S:STMPINTRIG + 1) % I2S:STMPWPER
- 当 I2S:STMPWCNT 等于 I2S:STMPOUTTRIG 时，输出 DMA 在下一帧开始从存储器输出采样字：(I2S:STMPOUTTRIG + 1) % I2S:STMPWPER

为避免虚假启动触发，I2S:STMPINTRIG 和 I2S:STMPOUTTRIG 最初必须等于或高于 I2S:STMPWPER。

I2S:STMPCTL.IN_RDY 和 I2S:STMPCTL.OUT_RDY 状态位会在输入和输出 DMA 准备好启动时设置，并在 DMA 启动触发发生时清除。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

24.6.3 采样时间戳捕获

可以从 RFCORE 路由捕获请求信号以触发采样时间戳捕获。EVENT:I2SSTMPSEL0 寄存器选择要使用的 RFCORE 信号。每当此捕获请求信号为高电平时：

- I2S:STMPXCNT 的当前值被复制到 I2S:STMPXCNTCAPT0。
- I2S:STMPWCNT 的当前值被复制到 I2S:STMPWCNTCAPT0。

此外，在每帧的第一个 WCLK 边沿，在 I2S:STMPXPER 中捕获 I2S:STMPXCNT 的当前值，然后 I2S:STMPXCNT 从 0 重新开始计数。

使用这些值，可以计算定点采样时间戳值：

I2S:STMPWCNTCAPT0 + (I2S:STMPXCNTCAPT0 / I2S:STMPXPER)

请注意，捕获 I2S:STMPXPER 的值时，通常不会同时捕获其他值。因此，I2S:STMPXPER 可以小于 I2S:STMPXCNTCAPT0。

24.6.4 实现恒定音频延迟

可以采取以下操作来在无线网络上实现在两个方向都相同的恒定音频延迟（从一个 CC13x2 和 CC26x2 器件平台上的 I²S 引脚到另一个 CC13x2 和 CC26x2 器件平台上的 I²S 引脚）：

- 必须将一个节点定义为音频时钟主器件，将另一个节点定义为音频时钟从器件。从器件必须使用速率可调的外部音频时钟源。
- 对于这两个节点，设置 I2S:STMPWPER = N × (I2S:AIFDMACFG.END_FRAME_IDX + 1)，其中 N 为整数。
  - I2S:STMPWPER 的值等于以帧数表示的音频延迟。
  - I2S:STMPWPER 的值也等于以样本数表示的存储器缓冲区大小。
- 当主器件发送射频数据包同步字时，对其执行采样时间戳捕获，并将定点采样时间戳的值包含在发送的数据包中。
- 当从器件收到射频数据包同步字时，对其执行采样时间戳捕获，并将主器件的采样时间戳存储到射频数据包中。计算主器件和从器件的采样时间戳值之差，用于：
  - 初步偏移从器件的 I2S:STMPWCNT 计数器，使其与器件的采样时间戳值匹配。
  - 运行时，调整外部音频时钟源速率，使从器件和主器件的采样时间戳值之差趋近于 0。
- 对于这两个节点，设置 DMA 指针和 DMA 启动触发条件，以便 STMPWCNT 的值表示当前帧在 ADx 引脚上的输入和输出缓冲区位置。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

24.7 错误检测

I²S 模块可以检测与以下各项相关的错误：

- DMA 运行
- 音频时钟信号完整性

检测到以下错误：

- WCLK 频率误差 (STMPXPERMIN:VALUE)
- WCLK 信号上的噪声 (IRQFLAGS:WCLK_ERR)
- 音频时钟丢失 (IRQFLAGS:WCLK_TIMEOUT)
- DMA 指针未及时加载 (IRQFLAGS:PTR_ERR)
- DMA 传输未及时完成 (IRQFLAGS:BUS_ERR)

24.8 用途

24.8.1 启动序列

按指示的顺序执行以下步骤，以开始 I²S 模块操作：

1. 设置依赖关系（请参阅节 24.4.1）。
2. 配置引脚（请参阅章节 13）。
3. 配置串行格式（请参阅节 24.4.3）。
4. 配置时钟（请参阅节 24.4.8）。
5. 配置采样字长（请参阅节 24.5.1）。
6. 配置通道映射（请参阅节 24.5.2）。
7. 执行 DMA 启动序列（请参阅节 24.5.4.1）。
8. 设置采样时间戳发生器：
   - 设置 I2S:STMPWPER 寄存器。
   - 设置 I2S:STMPINTRIG 和 I2S:STMPOUTTRIG > I2S:STMPWPER，以避免错误的 DMA 启动触发条件。
   - 设置 I2S:STMPCTL.EN = 1。
   - 如果需要，遵循实现恒定音频延迟的指南（请参阅节 24.6.4）。
   - 否则，只需设置 I2S:STMPINTRIG 和 I2S:STMPOUTTRIG 以匹配当前的 (I2S:STMPWCNT + 2) % I2S:STMPWPER。

备注

DMA 中断将在 DMA 完成第一个或多个采样块后开始。

24.8.2 关断序列

按指示的顺序执行以下步骤，以结束 I²S 模块操作：

1. DMA 关断序列（请参阅节 24.5.4.3）。
2. 设置 I2S:STMPCTL.EN = 0。
3. 禁用内部或外部音频时钟源。
4. 禁用依赖项（请参阅节 24.4.1）。

24.9 I²S 寄存器

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

24.9.1 I2S 寄存器

表 24-2 列出了 I2S 寄存器的存储器映射寄存器。表 24-2 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

表 24-2. I2S 寄存器

偏移	首字母缩写词	寄存器名称	部分
0h	AIFWCLKSRC	WCLK 源选择	AIFWCLKSRC 寄存器（偏移 = 0h）[复位 = 00000000h]
4h	AIFDMACFG	DMA 缓冲区大小配置	AIFDMACFG 寄存器（偏移 = 4h）[复位 = 00000000h]
8h	AIFDIRCFG	引脚方向	AIFDIRCFG 寄存器（偏移 = 8h）[复位 = 00000000h]
Ch	AIFFMTCFG	串行接口格式配置	AIFFMTCFG 寄存器（偏移 = Ch）[复位 = 00000170h]
10h	AIFWMASK0	引脚 0 的字选择位掩码	AIFWMASK0 寄存器（偏移 = 10h）[复位 = 00000003h]
14h	AIFWMASK1	引脚 1 的字选择位掩码	AIFWMASK1 寄存器（偏移 = 14h）[复位 = 00000003h]
1Ch	AIFPWMVALUE	音频接口 PWM 调试值	AIFPWMVALUE 寄存器（偏移 = 1Ch）[复位 = 00000000h]
20h	AIFINPTRNEXT	DMA 输入缓冲区下一个指针	AIFINPTRNEXT 寄存器（偏移 = 20h）[复位 = 00000000h]
24h	AIFINPTR	DMA 输入缓冲区当前指针	AIFINPTR 寄存器（偏移 = 24h）[复位 = 00000000h]
28h	AIFOUTPTRNEXT	DMA 输出缓冲区下一个指针	AIFOUTPTRNEXT 寄存器（偏移 = 28h）[复位 = 00000000h]
2Ch	AIFOUTPTR	DMA 输出缓冲区当前指针	AIFOUTPTR 寄存器（偏移 = 2Ch）[复位 = 00000000h]
34h	STMPCTL	采样戳发生器控制寄存器	STMPCTL 寄存器（偏移 =34h）[复位 = 00000000h]
38h	STMPXCNTCAPT0	捕获的 XOSC 计数器值捕获通道 0	STMPXCNTCAPT0 寄存器（偏移 = 38h）[复位 = 00000000h]
3Ch	STMPXPER	XOSC 周期值	STMPXPER 寄存器（偏移 = 3Ch）[复位 = 00000000h]
40h	STMPWCNTCAPT0	捕获的 WCLK 计数器值捕获通道 0	STMPWCNTCAPT0 寄存器（偏移 = 40h）[复位 = 00000000h]
44h	STMPWPER	WCLK 计数器周期值	STMPWPER 寄存器（偏移 = 44h）[复位 = 00000000h]
48h	STMPINTRIG	输入引脚的 WCLK 计数器触发值	STMPINTRIG 寄存器（偏移 = 48h）[复位 = 00000000h]
4Ch	STMPOUTTRIG	输出引脚的 WCLK 计数器触发值	STMPOUTTRIG 寄存器（偏移 = 4Ch）[复位 = 00000000h]
50h	STMPWSET	WCLK 计数器设置操作	STMPWSET 寄存器（偏移 = 50h）[复位 = 00000000h]
54h	STMPWADD	WCLK 计数器加法操作	STMPWADD 寄存器（偏移 = 54h）[复位 = 00000000h]
58h	STMPXPERMIN	XOSC 最小周期值	STMPXPERMIN 寄存器（偏移 = 58h）[复位 = 0000FFFFh]
5Ch	STMPWCNT	WCNT 的当前值	STMPWCNT 寄存器（偏移 = 5Ch）[复位 = 00000000h]
60h	STMPXCNT	XCNT 的当前值	STMPXCNT 寄存器（偏移 = 60h）[复位 = 00000000h]

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 24-2. I2S 寄存器（续）

偏移	首字母缩写词	寄存器名称	部分
64h	STMPXCNTCAPT1	内部	STMPXCNTCAPT1 寄存器（偏移 = 64h）[复位 = 00000000h]
68h	STMPWCNTCAPT1	内部	STMPWCNTCAPT1 寄存器（偏移 = 68h）[复位 = 00000000h]
70h	IRQMASK	中断屏蔽寄存器	IRQMASK 寄存器（偏移 = 70h）[复位 = 00000000h]
74h	IRQFLAGS	原始中断状态寄存器	IRQFLAGS 寄存器（偏移 = 74h）[复位 = 00000000h]
78h	IRQSET	中断集寄存器	IRQSET 寄存器（偏移 = 78h）[复位 = 00000000h]
7Ch	IRQCLR	中断清除寄存器	IRQCLR 寄存器（偏移 = 7Ch）[复位 = 00000000h]

复杂的位访问类型经过编码可适应小型表单元。表 24-3 展示了适用于此部分中访问类型的代码。

表 24-3. I2S 访问类型代码

访问类型	代码	说明
读取类型		
R	R	读取
写入类型		
W	W	写入
复位或默认值		
-n		复位后的值或默认值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

24.9.1.1 AIFWCLKSRC 寄存器（偏移 = 0h）[复位 = 00000000h]

图 24-12 展示了 AIFWCLKSRC，表 24-4 中对此进行了介绍。

返回到汇总表。

WCLK 源选择

图 24-12. AIFWCLKSRC 寄存器

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

表 24-4. AIFWCLKSRC 寄存器字段说明

位	字段	类型	复位	说明
31-3	RESERVED	R	0h	保留
2	WCLK_INV	R/W	0h	设置时反转 WCLK 源（焊盘或内部）。 0：不反转 1：反转
1-0	WCLK_SRC	R/W	0h	为 AIF 选择 WCLK 源（应与 BCLK 源相同）。BCLK 源在 PRCM:I2SBCLKSEL.SRC 中定义 0h = 无 ('0') 1h = 外部 WCLK 发生器，来自焊盘 2h = 内部 WCLK 发生器，来自 PRCM 模块 3h = 不支持。将产生与 'NONE' ('00') 相同的 WCLK

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

24.9.1.2 AIFDMACFG 寄存器（偏移 = 4h）[复位 = 00000000h]

图 24-13 展示了 AIFDMACFG，表 24-5 中对此进行了介绍。

返回到汇总表。

DMA 缓冲区大小配置

图 24-13. AIFDMACFG 寄存器

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								END_FRAME_IDX							
R-0h								R/W-0h							

表 24-5. AIFDMACFG 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-0	END_FRAME_IDX	R/W	0h	定义 DMA 缓冲区的长度。向该寄存器字段写入非零值将使能和初始化 AIF。请注意，执行此操作之前，必须已完成所有其他配置，并且必须已加载 AIFINPTRNEXT/AIFOUTPTRNEXT。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

24.9.1.3 AIFDIRCFG 寄存器（偏移 = 8h）[复位 = 00000000h]

图 24-14 展示了 AIFDIRCFG，表 24-6 中对此进行了介绍。

返回到汇总表。

引脚方向

图 24-14. AIFDIRCFG 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	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	

表 24-6. AIFDIRCFG 寄存器字段说明

位	字段	类型	复位	说明
31-6	RESERVED	R	0h	保留
5-4	AD1	R/W	0h	配置 AD1 音频数据引脚用途： 0x3：保留 0h = 未使用（禁用） 1h = 输入模式 2h = 输出模式
3-2	RESERVED	R	0h	保留
1-0	AD0	R/W	0h	配置 AD0 音频数据引脚用途： 0x3：保留 0h = 未使用（禁用） 1h = 输入模式 2h = 输出模式

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

24.9.1.4 AIFFMTCFG 寄存器（偏移 = Ch）[复位 = 00000170h]

图 24-15 展示了 AIFFMTCFG，表 24-7 中对此进行了介绍。

返回到汇总表。

串行接口格式配置

图 24-15. AIFFMTCFG 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
DATA_DELAY							
R/W-1h							
7	6	5	4	3	2	1	0
MEM_LEN_24	SMPL_EDGE	DUAL_PHASE	WORD_LEN				
R/W-0h	R/W-1h	R/W-1h	R/W-10h				

表 24-7. AIFFMTCFG 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-8	DATA_DELAY	R/W	1h	WCLK 边沿与一个相位中第一个字的 MSB 之间的 BCLK 周期数： 0x00：LJF 和 DSP 格式 0x01：I2S 和 DSP 格式 0x02：RJF 格式 ... 0xFF：RJF 格式 注意：当设置为 0 时，下一个字的 MSB 将在前一个字的 LSB 和下一个字开始之间的空闲周期内输出。否则将输出逻辑 0，直到数据延迟到期。
7	MEM_LEN_24	R/W	0h	存储到存储器或从存储器加载的每个字的大小： 0h = 16BIT：16 位（每个样本一次 16 位访问） 1h = 24BIT：24 位（每个样本一次 8 位和一次 16 位锁定访问）
6	SMPL_EDGE	R/W	1h	在串行音频接口上，数据（和 wclk）在 BCLK 的相反边沿上进行采样和时钟输出。 0h = 数据在负边沿采样，在正边沿时钟输出。 1h = 数据在正边沿采样，在负边沿时钟输出。
5	DUAL_PHASE	R/W	1h	选择双相或单相格式。 0：单相：DSP 格式 1：双相：I2S、LJF 和 RJF 格式
4-0	WORD_LEN	R/W	10h	每个字的位数 (8-24)： 在单相格式中，这是每个字的确切位数。 在双相格式中，这是每个字的最大位数。 低于 8 和高于 24 的值会产生未定义的行为。写入存储器的数据始终与 MEM_LEN_24 定义的 16 或 24 位对齐。与此对齐方式不同的位宽将被截断或补零。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

24.9.1.5 AIFWMASK0 寄存器（偏移 = 10h）[复位 = 00000003h]

图 24-16 展示了 AIFWMASK0，表 24-8 对其进行了介绍。

返回到汇总表。

引脚 0 的字选择位掩码

图 24-16. AIFWMASK0 寄存器

31	30	29	28	27	26	25	24	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-3h							

表 24-8. AIFWMASK0 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-0	MASK	R/W	3h	位掩码，表示 AD0 上的一个帧中的有效通道。 在单相模式下，每个位代表一个通道，从 LSB 开始对应帧中的第一个字。一个帧最多可以包含 8 个通道。未包含在掩码中的通道将不会被采样或存储到存储器中，并且时钟输出时为“0”。 在双相模式下，仅考虑两个 LSB。对于立体声配置，请设置两个位。对于单声道配置，请仅设置位 0。在单声道模式下，仅通道 0 会被采样并存储到存储器中，并且在时钟输出时，通道 0 的内容会重复。 仅通道 0 会被采样并存储到存储器中，并且在时钟输出时，通道 0 的内容会在第二相中重复。 如果所有位都为零，则不会将任何输入字存储到存储器中，并且输出数据线将保持为常量“0”。当需要在没有任何实际使用的输出引脚的情况下输出 PWM 调试信号时，可以利用此功能。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

24.9.1.6 AIFWMASK1 寄存器（偏移 = 14h）[复位 = 00000003h]

图 24-17 展示了 AIFWMASK1，表 24-9 对其进行了介绍。

返回到汇总表。

引脚 1 的字选择位掩码

图 24-17. AIFWMASK1 寄存器

31	30	29	28	27	26	25	24	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-3h							

表 24-9. AIFWMASK1 寄存器字段说明

位	字段	类型	复位	说明
31-8	RESERVED	R	0h	保留
7-0	MASK	R/W	3h	位掩码，表示 AD1 上的一个帧中的有效通道。 在单相模式下，每个位代表一个通道，从 LSB 开始对应帧中的第一个字。一个帧最多可以包含 8 个通道。未包含在掩码中的通道将不会被采样或存储到存储器中，并且时钟输出时为“0”。 在双相模式下，仅考虑两个 LSB。对于立体声配置，请设置两个位。对于单声道配置，请仅设置位 0。在单声道模式下，仅通道 0 会被采样并存储到存储器中，并且在时钟输出时，通道 0 的内容会重复。仅通道 0 会被采样并存储到存储器中，并且在时钟输出时，通道 0 的内容会在第二相中重复。 如果所有位都为零，则不会将任何输入字存储到存储器中，并且输出数据线将保持为常量“0”。当需要在没有任何实际使用的输出引脚的情况下输出 PWM 调试信号时，可以利用此功能。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

24.9.1.7 AIFPWMVALUE 寄存器（偏移 = 1Ch）[复位 = 00000000h]

图 24-18 展示了 AIFPWMVALUE，表 24-10 中对此进行了介绍。

返回到汇总表。

音频接口 PWM 调试值

图 24-18. AIFPWMVALUE 寄存器

31	30	29	28	27	26	25	24	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																PULSE_WIDTH															
R-0h																R/W-0h															

表 24-10. AIFPWMVALUE 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	PULSE_WIDTH	R/W	0h	写入此寄存器的值决定了高电平有效 PWM 脉冲 (pwm_debug) 的宽度，从 DMA 缓冲器区中第一个输出字的 MSB 同时开始： 0x0000：恒定低电平 0x0001：脉冲宽度（BCLK 周期数，此处为 1）。 ... 0xFFFE：脉冲宽度（BCLK 周期数，处为 65534）。 0xFFFF：恒定高电平

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

24.9.1.8 AIFINPTRNEXT 寄存器（偏移 = 20h）[复位 = 00000000h]

图 24-19 展示了 AIFINPTRNEXT，表 24-11 中对此进行了介绍。

返回到汇总表。

DMA 输入缓冲区下一个指针

图 24-19. AIFINPTRNEXT 寄存器

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

表 24-11. AIFINPTRNEXT 寄存器字段说明

位	字段	类型	复位	说明
31-0	PTR	R/W	0h	指向下一个 DMA 输入缓冲区中第一个字节的指针。 读取的值等于在当前使用的 DMA 输入缓冲区完成之前最后写入的值，然后在最后写入的值传输到 DMA 控制器以在下一个缓冲区上开始时变为空。此事件由 IRQFLAGS.AIF_DMA_IN 标志指示。 启动时，必须在 AIFDMACFG 中配置 DMA 缓冲区大小之前和之后分别写入一次该值。 在 DMA 功能使用先前写入的指针时，必须将下一个指针写入该寄存器。如果未及时写入，则将产生 IRQFLAGS.PTR_ERR，并将禁用所有输入引脚。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

24.9.1.9 AIFINPTR 寄存器（偏移 = 24h）[复位 = 00000000h]

图 24-20 展示了 AIFINPTR，表 24-12 中对此进行了介绍。

返回到汇总表。

DMA 输入缓冲区当前指针

图 24-20. AIFINPTR 寄存器

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

表 24-12. AIFINPTR 寄存器字段说明

位	字段	类型	复位	说明
31-0	PTR	R	0h	DMA 控制器当前使用的 DMA 输入缓冲区指针的值。对于每次 AHB 访问，递增 1（字节）或 2（字）。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

24.9.1.10 AIFOUTPTRNEXT 寄存器（偏移 = 28h）[复位 = 00000000h]

图 24-21 展示了 AIFOUTPTRNEXT，表 24-13 中对此进行了介绍。

返回到汇总表。

DMA 输出缓冲区下一个指针

图 24-21. AIFOUTPTRNEXT 寄存器

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

表 24-13. AIFOUTPTRNEXT 寄存器字段说明

位	字段	类型	复位	说明
31-0	PTR	R/W	0h	指向下一个 DMA 输出缓冲区中第一个字节的指针。 读取的值等于在当前使用的 DMA 输出缓冲区完成之前最后写入的值，然后在最后写入的值传输到 DMA 控制器以在下一个缓冲区上开始时变为空。此事件由 IRQFLAGS.AIF_DMA_OUT 标志指示。 启动时，必须在 AIFDMACFG 中配置 DMA 缓冲区大小之前和之后分别写入一次该值。此时将从存储器中获取前两个样本。 在 DMA 功能使用先前写入的指针时，必须将下一个指针写入该寄存器。如果未及时写入，则将产生 IRQFLAGS.PTR_ERR，并将禁用所有输出引脚。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

24.9.1.11 AIFOUTPTR 寄存器（偏移 = 2Ch）[复位 = 00000000h]

图 24-22 展示了 AIFOUTPTR，表 24-14 中对此进行了介绍。

返回到汇总表。

DMA 输出缓冲区当前指针

图 24-22. AIFOUTPTR 寄存器

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

表 24-14. AIFOUTPTR 寄存器字段说明

位	字段	类型	复位	说明
31-0	PTR	R	0h	对于每次 AHB 访问，DMA 控制器当前使用的 DMA 输出缓冲区指针的值递增 1（字节）或 2（字）。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

24.9.1.12 STMPCTL 寄存器（偏移 =34h）[复位 = 00000000h]

图 24-23 展示了 STMPCTL，表 24-15 中对此进行了介绍。

返回到汇总表。

采样戳发生器控制寄存器

图 24-23. STMPCTL 寄存器

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

表 24-15. STMPCTL 寄存器字段说明

位	字段	类型	复位	说明
31-3	RESERVED	R	0h	保留
2	OUT_RDY	R	0h	保持低电平，直到输出引脚准备好由采样戳发生器启动。当启动时（即 STMPOUTTRIG 等于 WCLK 计数器），该位恢复为低电平。
1	IN_RDY	R	0h	保持低电平，直到输入引脚准备好由采样戳发生器启动。当启动时（即 STMPINTRIG 等于 WCLK 计数器），该位恢复为低电平。
0	STMP_EN	R/W	0h	使能采样戳发生器。采样戳发生器必须在正确配置后才能使能。清除后，所有采样戳发生器计数器和捕获值都会被清除。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

24.9.1.13 STMPXCNTCAPT0 寄存器（偏移 = 38h）[复位 = 00000000h]

图 24-24 展示了 STMPXCNTCAPT0，表 24-16 对其进行了介绍。

返回到汇总表。

捕获的 XOSC 计数器值捕获通道 0

图 24-24. STMPXCNTCAPT0 寄存器

31	30	29	28	27	26	25	24	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																CAPT_VALUE															
R-0h																R-0h															

表 24-16. STMPXCNTCAPT0 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	CAPT_VALUE	R	0h	上次事件脉冲发生时采样戳 XOSC 计数器 (STMPXCNT.CURR_VALUE) 的值（事件源在通道 0 的 [EVENT.I2SSTMPSEL0.EV] 中选择）。该数字对应于自所选 WCLK 最后一个正边沿以来的 24MHz 时钟周期数。 当 STMPCTL.STMP_EN = 0 时，该值被清除。 注意：由于缓冲和同步，WCLK 会延迟少量 BCLK 周期和 clk 周期。 注意：在计算采样时间戳的小数部分时，STMPXPER 可能小于该位字段。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

24.9.1.14 STMPXPER 寄存器（偏移 = 3Ch）[复位 = 00000000h]

图 24-25 展示了 STMPXPER，表 24-17 中对此进行了介绍。

返回到汇总表。

XOSC 周期值

图 24-25. STMPXPER 寄存器

31	30	29	28	27	26	25	24	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																值															
R-0h																R-0h															

表 24-17. STMPXPER 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	值	R	0h	在上一个 WCLK 周期内的 24MHz 时钟周期数（即：XOSC 计数器在 WCLK 正边沿的下一个值 — 假设它未复位为 0）。 当 STMPCTL.STMP_EN = 0 时，该值被清除。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

24.9.1.15 STMPWCNTCAPT0 寄存器（偏移 = 40h）[复位 = 00000000h]

图 24-26 展示了 STMPWCNTCAPT0，表 24-18 对其进行了介绍。

返回到汇总表。

捕获的 WCLK 计数器值捕获通道 0

图 24-26. STMPWCNTCAPT0 寄存器

31	30	29	28	27	26	25	24	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																CAPT_VALUE															
R-0h																R-0h															

表 24-18. STMPWCNTCAPT0 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	CAPT_VALUE	R	0h	上次事件脉冲发生时采样戳 WCLK 计数器 (STMPWCNT.CURR_VALUE) 的值（事件源在通道 0 的 EVENT:I2SSTMPSEL0.EV 中选择）。该数字对应于自采样戳发生器使能以来的正 WCLK 边沿数（不考虑通过 STMPWADD/STMPWSET 进行的修改）。 当 STMPCTL.STMP_EN = 0 时，该值被清除。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

24.9.1.16 STMPWPER 寄存器（偏移 = 44h）[复位 = 00000000h]

图 24-27 展示了 STMPWPER，表 24-19 中对此进行了介绍。

返回到汇总表。

WCLK 计数器周期值

图 24-27. STMPWPER 寄存器

31	30	29	28	27	26	25	24	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																值															
R-0h																R/W-0h															

表 24-19. STMPWPER 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	值	R/W	0h	用于定义何时复位 STMPWCNT，以便找到适合采样缓冲区大小的 WCLK 边沿数。因此，这是 WCLK 计数器的模数值。该数字必须与系统使用的样本缓冲区的大小相对应（即最后一个样本的索引加 1）。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

24.9.1.17 STMPINTRIG 寄存器（偏移 = 48h）[复位 = 00000000h]

图 24-28 展示了 STMPINTRIG，表 24-20 中对此进行了介绍。

返回到汇总表。

输入引脚的 WCLK 计数器触发值

图 24-28. STMPINTRIG 寄存器

31	30	29	28	27	26	25	24	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																IN_START_WCNT															
R-0h																R/W-0h															

表 24-20. STMPINTRIG 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	IN_START_WCNT	R/W	0h	用于启动传入音频流的比较值。 该位字段应等于在第一个输入字被采样并存储到存储器（即第一个 DMA 输入缓冲区起始处的采样）所在 WCLK 周期内的 WCLK 计数器值。 当满足以下条件时，该寄存器的值将生效： - 一个或多个引脚配置为 AIFDIRCFG 中的输入。 - AIFDMACFG 已配置为正确的缓冲区大小，并且至少发生了 32 个 BCLK 周期节拍。 注意：为了避免误触发，该位字段应设置为高于 STMPWPER.VALUE。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

24.9.1.18 STMPOUTTRIG 寄存器（偏移 = 4Ch）[复位 = 00000000h]

图 24-29 展示了 STMPOUTTRIG，表 24-21 中对此进行了介绍。

返回到汇总表。

输出引脚的 WCLK 计数器触发值

图 24-29. STMPOUTTRIG 寄存器

31	30	29	28	27	26	25	24	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																OUT_START_WCNT															
R-0h																R/W-0h															

表 24-21. STMPOUTTRIG 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	OUT_START_WCNT	R/W	0h	用于启动传出音频流的比较值。 该位字段必须等于从存储器读取的第一个输出字被时钟输出（即第一个 DMA 输出缓冲区起始处的采样）所在 WCLK 周期内的 WCLK 计数器值。 当满足以下条件时，该寄存器的值将生效： - 一个或多个引脚配置为 AIFDIRCFG 中的输出。 - AIFDMACFG 已配置为正确的缓冲区大小，并且发生了 32 个 BCLK 周期节拍。 - 已从存储器中预加载 2 个样本（如有必要，请检查 AIFOUTPTR 寄存器）。 注意：仅在需要时才会执行存储器读取访问，即必须在 AIFWMASK0/AIFWMASK1 中选择通道 0/1。 注意：为了避免误触发，该位字段应设置为高于 STMPWPER.VALUE。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

24.9.1.19 STMPWSET 寄存器（偏移 = 50h）[复位 = 00000000h]

图 24-30 展示了 STMPWSET，表 24-22 中对此进行了介绍。

返回到汇总表。

WCLK 计数器设置操作

图 24-30. STMPWSET 寄存器

31	30	29	28	27	26	25	24	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																值															
R-0h																R/W-0h															

表 24-22. STMPWSET 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	值	R/W	0h	WCLK 计数器修改：将正在运行的 WCLK 计数器设置为等于写入的值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

24.9.1.20 STMPWADD 寄存器（偏移 = 54h）[复位 = 00000000h]

图 24-31 展示了 STMPWADD，表 24-23 中对此进行了介绍。

返回到汇总表。

WCLK 计数器加法操作

图 24-31. STMPWADD 寄存器

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

表 24-23. STMPWADD 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	VALUE_INC	R/W	0h	WCLK 计数器修改：将写入的值加到正在运行的 WCLK 计数器上。如果在执行此操作的同时发生 WCLK 正边沿，则此边沿会被考虑在内。如需加上一个负值，请写入“STMPWPER.VALUE - 所需减去的值”。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

24.9.1.21 STMPXPERMIN 寄存器（偏移 = 58h）[复位 = 0000FFFFh]

图 24-32 展示了 STMPXPERMIN，表 24-24 中对此进行了介绍。

返回到汇总表。

XOSC 最小周期值
STMPXPER 的最小值

图 24-32. STMPXPERMIN 寄存器

31	30	29	28	27	26	25	24	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																值															
R-0h																R/W-FFFFh															

表 24-24. STMPXPERMIN 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	值	R/W	FFFFh	每次更新 STMPXPER 时，只要其值小于该寄存器中的当前值，也将加载到该寄存器中。 写入时，无论写入的值如何，该寄存器都会复位为 0xFFFF (65535)。 最小值可用于检测额外的 WCLK 脉冲（该寄存器值将显著小于 STMPXPER.VALUE）。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

24.9.1.22 STMPWCNT 寄存器（偏移 = 5Ch）[复位 = 00000000h]

图 24-33 展示了 STMPWCNT，表 24-25 中对此进行了介绍。

返回到汇总表。

WCNT 的当前值

图 24-33. STMPWCNT 寄存器

31	30	29	28	27	26	25	24	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																CURR_VALUE															
R-0h																R-0h															

表 24-25. STMPWCNT 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	CURR_VALUE	R	0h	WCLK 计数器的当前值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

24.9.1.23 STMPXCNT 寄存器（偏移 = 60h）[复位 = 00000000h]

图 24-34 展示了 STMPXCNT，表 24-26 中对此进行了介绍。

返回到汇总表。

XCNT 的当前值

图 24-34. STMPXCNT 寄存器

31	30	29	28	27	26	25	24	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																CURR_VALUE															
R-0h																R-0h															

表 24-26. STMPXCNT 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	CURR_VALUE	R	0h	XOSC 计数器的当前值，读取 STMPWCNT 时锁存。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

24.9.1.24 STMPXCNTCAPT1 寄存器（偏移 = 64h）[复位 = 00000000h]

图 24-35 展示了 STMPXCNTCAPT1，表 24-27 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 24-35. STMPXCNTCAPT1 寄存器

31	30	29	28	27	26	25	24	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																CAPT_VALUE															
R-0h																R-0h															

表 24-27. STMPXCNTCAPT1 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	CAPT_VALUE	R	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

24.9.1.25 STMPWCNTCAPT1 寄存器（偏移 = 68h）[复位 = 00000000h]

图 24-36 展示了 STMPWCNTCAPT1，表 24-28 对其进行了介绍。

返回到汇总表。

内部。仅可通过 TI 提供的 API 使用。

图 24-36. STMPWCNTCAPT1 寄存器

31	30	29	28	27	26	25	24	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																CAPT_VALUE															
R-0h																R-0h															

表 24-28. STMPWCNTCAPT1 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-0	CAPT_VALUE	R	0h	内部。仅可通过 TI 提供的 API 使用。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

24.9.1.26 IRQMASK 寄存器（偏移 = 70h）[复位 = 00000000h]

图 24-37 展示了 IRQMASK，表 24-29 中对此进行了介绍。

返回到汇总表。

中断屏蔽寄存器
选择 IRQFLAGS 中参与生成 I2S_IRQ 事件的标志的屏蔽状态。

图 24-37. IRQMASK 寄存器

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

表 24-29. IRQMASK 寄存器字段说明

位	字段	类型	复位	说明
31-6	RESERVED	R	0h	保留
5	AIF_DMA_IN	R/W	0h	IRQFLAGS.AIF_DMA_IN 中断屏蔽 0：禁用 1：启用
4	AIF_DMA_OUT	R/W	0h	IRQFLAGS.AIF_DMA_OUT 中断屏蔽 0：禁用 1：启用
3	WCLK_TIMEOUT	R/W	0h	IRQFLAGS.WCLK_TIMEOUT 中断屏蔽 0：禁用 1：启用
2	BUS_ERR	R/W	0h	IRQFLAGS.BUS_ERR 中断屏蔽 0：禁用 1：启用
1	WCLK_ERR	R/W	0h	IRQFLAGS.WCLK_ERR 中断屏蔽 0：禁用 1：启用
0	PTR_ERR	R/W	0h	IRQFLAGS.PTR_ERR 中断屏蔽。 0：禁用 1：启用

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

24.9.1.27 IRQFLAGS 寄存器（偏移 = 74h）[复位 = 00000000h]

图 24-38 展示了 IRQFLAGS，表 24-30 中对此进行了介绍。

返回到汇总表。

原始中断状态寄存器

图 24-38. IRQFLAGS 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED		AIF_DMA_IN	AIF_DMA_OUT	WCLK_TIMEOUT	BUS_ERR	WCLK_ERR	PTR_ERR
R-0h		R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

表 24-30. IRQFLAGS 寄存器字段说明

位	字段	类型	复位	说明
31-6	RESERVED	R	0h	保留
5	AIF_DMA_IN	R	0h	当该位字段事件的条件发生时设置（更新输入指针时自动清除 — AIFINPTRNEXT），有关详细信息，请参阅 AIFINPTRNEXT 寄存器的说明。
4	AIF_DMA_OUT	R	0h	当该位字段事件的条件发生时设置（更新输出指针时自动清除 — AIFOUTPTRNEXT），有关详细信息，请参阅 AIFOUTPTRNEXT 寄存器的说明
3	WCLK_TIMEOUT	R	0h	当采样时间戳发生器在 65535 个 clk 周期内未检测到正 WCLK 边沿时设置。这表示内部或外部 BCLK 和 WCLK 发生器源已禁用。 该位为粘滞位，只能通过软件清除（通过向 IRQCLR.WCLK_TIMEOUT 写入“1”）。
2	BUS_ERR	R	0h	当 DMA 操作未及时完成（即音频输出缓冲区下溢或音频输入缓冲区上溢）时设置。 由于字同步已丢失，此错误需要完全重新启动。该位为粘滞位，只能通过软件清除（通过向 IRQCLR.BUS_ERR 写入“1”）。 请注意，DMA 发起的对非法地址的事务不会触发中断。对此类事务的响应未定义。
1	WCLK_ERR	R	0h	在以下情况下置 1： - 在相位的数据延迟期间发生意外的 WCLK 边沿。请注意，在相位的字和空闲期间，不会检测到意外的 WCLK 边沿。 - 在双相模式下，当两个 WCLK 边沿相距小于 4 个 BCLK 周期时。 - 在单相模式下，当一个 WCLK 脉冲发生在最后一个通道之前。 由于字同步已丢失，此错误需要完全重新启动。该位为粘滞位，只能通过软件清除（通过向 IRQCLR.WCLK_ERR 写入“1”）。
0	PTR_ERR	R	0h	当 AIFINPTRNEXT 或 AIFOUTPTRNEXT 未及时加载下一个块地址时设置。 由于字同步已丢失，此错误需要完全重新启动。该位为粘滞位，只能通过软件清除（通过向 IRQCLR.PTR_ERR 写入“1”）。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

24.9.1.28 IRQSET 寄存器（偏移 = 78h）[复位 = 00000000h]

图 24-39 展示了 IRQSET，表 24-31 中对此进行了介绍。

返回到汇总表。

中断集寄存器

图 24-39. IRQSET 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED		AIF_DMA_IN	AIF_DMA_OUT	WCLK_TIMEOUT	BUS_ERR	WCLK_ERR	PTR_ERR
R-0h		W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

表 24-31. IRQSET 寄存器字段说明

位	字段	类型	复位	说明
31-6	RESERVED	R	0h	保留
5	AIF_DMA_IN	W	0h	1：设置 IRQFLAGS.AIF_DMA_IN 的中断（除非同时满足了自动清除条件，在这种情况下该设置将被忽略）
4	AIF_DMA_OUT	W	0h	1：设置 IRQFLAGS.AIF_DMA_OUT 的中断（除非同时满足了自动清除条件，在这种情况下该设置将被忽略）
3	WCLK_TIMEOUT	W	0h	1：设置 IRQFLAGS.WCLK_TIMEOUT 的中断
2	BUS_ERR	W	0h	1：设置 IRQFLAGS.BUS_ERR 的中断
1	WCLK_ERR	W	0h	1：设置 IRQFLAGS.WCLK_ERR 的中断
0	PTR_ERR	W	0h	1：设置 IRQFLAGS.PTR_ERR 的中断

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

24.9.1.29 IRQCLR 寄存器（偏移 = 7Ch）[复位 = 00000000h]

图 24-40 展示了 IRQCLR，表 24-32 中对此进行了介绍。

返回到汇总表。

中断清除寄存器

图 24-40. IRQCLR 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED		AIF_DMA_IN	AIF_DMA_OUT	WCLK_TIMEOUT	BUS_ERR	WCLK_ERR	PTR_ERR
R-0h		W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

表 24-32. IRQCLR 寄存器字段说明

位	字段	类型	复位	说明
31-6	RESERVED	R	0h	保留
5	AIF_DMA_IN	W	0h	1：清除 IRQFLAGS.AIF_DMA_IN 的中断（除非同时满足了设置条件，在这种情况下该清除将被忽略）
4	AIF_DMA_OUT	W	0h	1：清除 IRQFLAGS.AIF_DMA_OUT 的中断（除非同时满足了设置条件，在这种情况下该清除将被忽略）
3	WCLK_TIMEOUT	W	0h	1：清除 IRQFLAGS.WCLK_TIMEOUT 的中断（除非同时满足了设置条件，在这种情况下该清除将被忽略）
2	BUS_ERR	W	0h	1：清除 IRQFLAGS.BUS_ERR 的中断（除非同时满足了设置条件，在这种情况下该清除将被忽略）
1	WCLK_ERR	W	0h	1：清除 IRQFLAGS.WCLK_ERR 的中断（除非同时满足了设置条件，在这种情况下该清除将被忽略）
0	PTR_ERR	W	0h	1：清除 IRQFLAGS.PTR_ERR 的中断（除非同时满足了设置条件，在这种情况下该清除将被忽略）

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

章节 25

无线电

CC13x2 和 CC26x2 器件平台中的无线电提供了多种不同的工作模式，涵盖许多不同的数据包格式。无线电固件在 CC13x2 和 CC26x2 平台无线电域中的 Arm® Cortex®-M0 处理器上执行，可提供广泛的基带自动化能力。应用软件使用共享内存接口（系统 RAM 或无线电 RAM）和特定的握手硬件（无线电门铃）与无线电固件进行接口连接和互操作。

25.1 RF 内核 1976
25.2 无线电门铃 1978
25.3 RF 内核 HAL 1981
25.4 数据队列使用情况 2020
25.5 IEEE 802.15.4 2024
25.6 低功耗 Bluetooth® 2048
25.7 数据处理 2068
25.8 无线电操作命令说明 2069
25.9 即时命令 2106
25.10 专有无线电 2106
25.11 无线电寄存器 2126

Copyright © 2026 Texas Instruments Incorporated

25.1 RF 内核

射频内核包含一个 Arm® Cortex®-M0 处理器，该处理器与模拟射频和基带电路建立接口连接，处理与系统侧之间的数据传输，并将信息位组装到给定的数据包结构中。射频内核为系统 CPU（Arm Cortex-M4F 处理器）提供了一个基于命令的高级应用程序接口 (API)。射频内核可以自主处理无线电协议（802.15.4 RF4CE 和 Zigbee®、低功耗 Bluetooth® 等）中的时间关键型部分，从而减轻系统 CPU 的负担，并为用户应用留出更多资源。

射频内核具有专用的 8kB 保持 SRAM 块和 4kB 非保持 SRAM 块，并且几乎完全从独立的 ROM 运行。每次无线电断电时，非保持 SRAM 块的内容都会丢失。

25.1.1 概要说明和概述

RF 内核接收来自系统 CPU 的高级请求，并执行所有必要的事务来完成这些请求。这些请求基本上面向通过无线电通道传输和接收信息，但也可以包括额外的维护任务，例如校准、测试或调试功能。

作为一般框架，系统 CPU 和 RF 内核之间的事务如下运行：

- RF 内核可以从系统 RAM 访问数据和配置参数。这降低了 RF 内核的内存要求，避免了系统不同器件之间不必要的流量，并降低了总能耗。
- 类似地，RF 内核可以解码接收到的无线电数据包的内容以及状态信息，并将其写回系统 RAM。
- 出于协议机密性和身份验证支持目的，RF 内核还可以访问安全子系统。
- 通常，RF 内核会识别来自系统 CPU 的复杂命令（CCA 传输、具有自动确认功能的 RX 等）并将其拆分为子命令，而无需系统 CPU 的进一步干预。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

图 25-1 展示了 RF 内核的外部接口和依赖项。

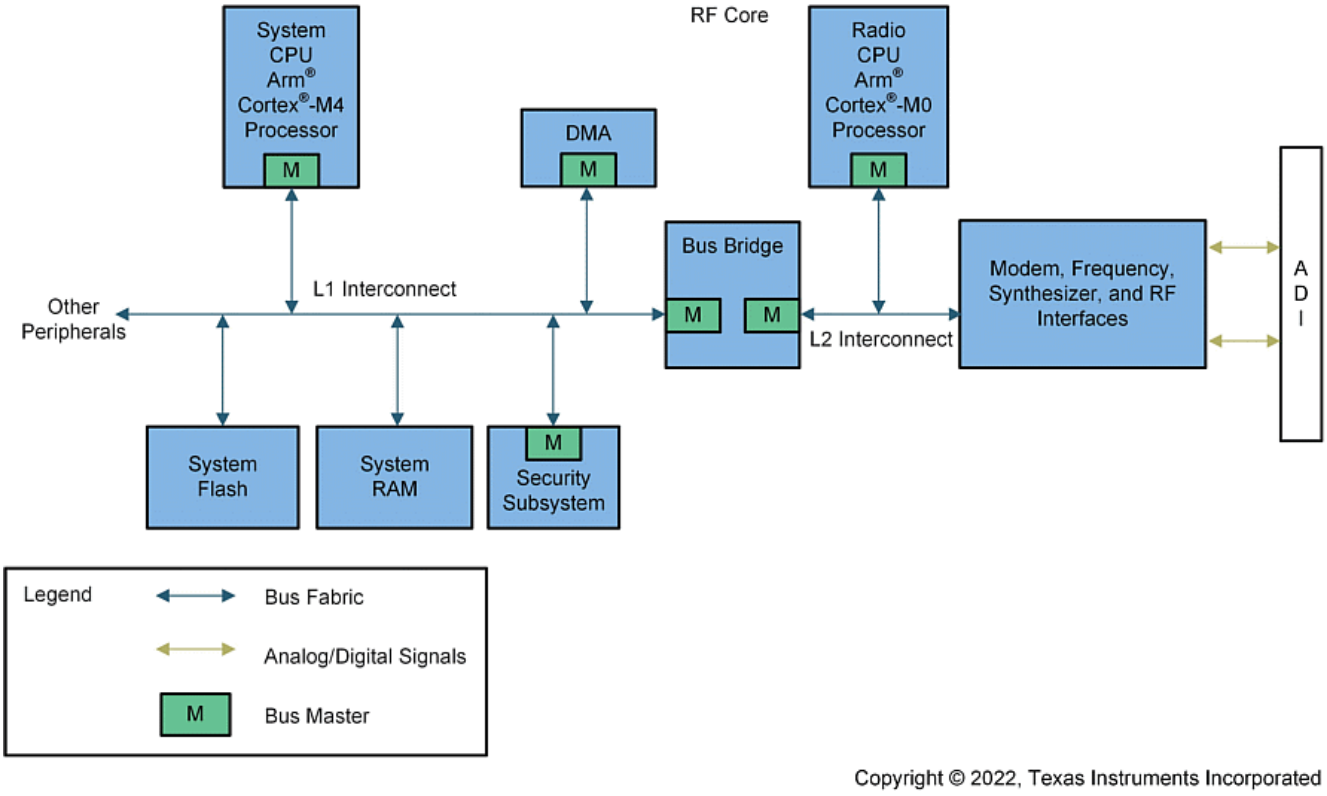


图 25-1. 具有外部依赖性的有限 RF 内核概述

图 25-1 中显示的每个块执行以下功能：

系统侧

- 系统 CPU：运行用户应用的主系统处理器，与高级协议栈（适用于许多受支持的配置）一起，最终为某些协议提供一些更高级别的 MAC 功能。系统 CPU 运行来自引导 ROM 及系统闪存的代码。
- 系统 RAM：包含数据包信息（TX 和 RX 有效载荷）以及给定事务的不同参数或者配置选项。
- 安全子系统：包含用于提供协议机密性及身份验证的不同元素。
- DMA：如果不使用直接 CPU 访问，则可选地用于将信息从无线电 RAM 移动到系统 RAM，反之亦然。

无线电侧

- 无线电 CPU：主 RF 内核处理器。接收来自系统 CPU 的高级命令，并将其调度到 RF 内核的不同部分。
- 调制解调器、频率合成器、RF 接口：这是无线电的核心，将位转换为调制信号，反之亦可。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.2 无线电门铃

无线电门铃模块 (RFC_DBELL) 是系统 CPU 和无线电 CPU（也称为命令和数据包引擎 CPE）之间的主要通信方式。无线电门铃包含一组专用寄存器、器件平台任何 RAM 中的参数，以及一组发给无线电 CPU 和系统 CPU 的中断。

此外，参数和有效载荷通过系统 RAM 或无线电 RAM 进行传输。如果任何参数或有效载荷位于系统 RAM 中，系统 CPU 必须保持上电。但是，如果所有内容都在无线电 RAM 中，系统 CPU 可以进入掉电模式以节省电流。

在操作期间，无线电 CPU 会更新 RAM 中的参数和有效载荷，并产生中断。系统 CPU 可以屏蔽中断，使其保持在空闲或掉电模式，直到整个无线电操作完成。

由于系统 CPU 和无线电 CPU 共享一个公共的 RAM 区域，因此必须确保不会发生争用或竞态条件。这是通过在无线电硬件抽象层 (HAL) 中设置的软件规则来实现的。

图 25-2 显示了用于 CPU 之间信息交换的相关模块。

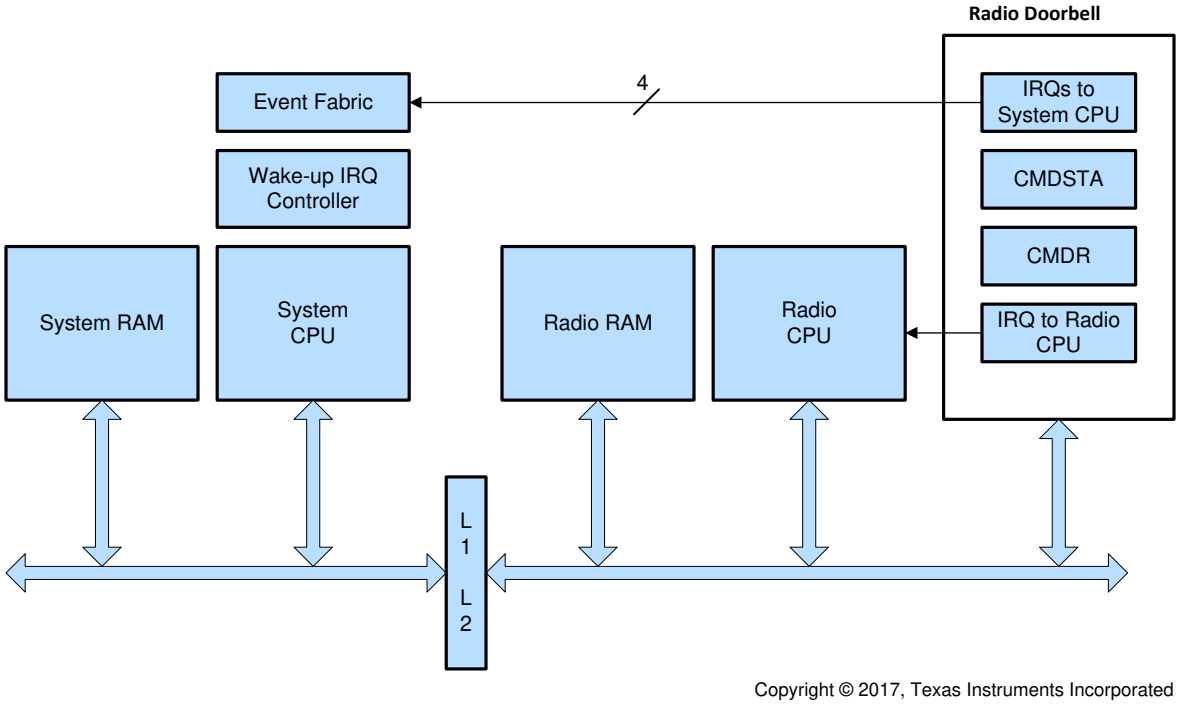


图 25-2. HAL 的硬件支持

25.2.1 特殊引导过程

PRCM:RFCBITS 寄存器聚合了供 CPE 在引导时评估的控制信息。可以通过设置相应位来单独启用/禁用这些功能，这有助于自动化最常用的功能，请参阅表 25-1。

表 25-1. 用于在引导过程中启用选定特性的 RFCBITS 格式

位索引	值	说明
0	1	特殊引导序列
1..2	0	引导选项
3	x	如果为 1，则强制开启在引导时启用的所有时钟
4	x	如果为 1，则立即请求系统总线
5	x	如果为 1，则立即启动无线电计时器
6	x	如果为 1，则立即禁用指针检查
7	x	如果为 1，则假定射频内核保持 RAM 已初始化

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-1. 用于在引导过程中启用选定特性的 RFCBITS 格式（续）

位索引	值	说明
8..11	x	保留
12	x	如果为 1，则启用要生成的 Command_Started 和 FG_Command_Started 中断（仅限 PG2.0）
13..28	0	保留
29..31	7	特殊引导选项

25.2.2 命令及状态寄存器和事件

向无线电发送命令通过 CMDR 寄存器完成，而 CMDSTA 只读寄存器提供从无线电返回的状态。CMDR 寄存器只能在读数为 0 时被写入；否则，写入将被忽略。当 CMDR 寄存器为 0 并且向其写入非零值时，将通知无线电 CPU，并且 CMDSTA 寄存器将变为 0。执行此操作后，可以从 CMDR 寄存器读取写入的值，直到无线电 CPU 处理该命令，此时它返回到 0。

当无线电 CPU 处理该命令时，CMDSTA 寄存器包含非零状态，当 CMDR 寄存器恢复到 0 时，会在同一时刻提供该状态。此时，会发生 RFCMDACK 中断。该中断也被映射到 RFACKIFG 寄存器，该寄存器在中断被处理时应该被清除。

有关命令和状态寄存器的格式，请参阅节 25.3.2。

25.2.3 RF 内核中断

RF 内核有四条连接到 Arm Cortex-M4F 处理器的中断线路（请参阅图 25-2）。以下中断通过无线电门铃模块来控制：

- RF_CPE0（中断编号 9）
- RF_CPE1（中断编号 2）
- RF_HW（中断编号 10）
- RF_CMD_ACK（中断编号 11）

25.2.3.1 RF 命令和数据包引擎中断

两个系统级中断 RF_CPE0 和 RF_CPE1 可由 CPE 产生的许多低级中断生成。这些低级中断中的每一个都可使用 RFCPEISL 寄存器映射到 RF_CPE0 或 RF_CPE1。此外，可以使用 RFCPEIEN 寄存器打开和关闭系统级中断生成。

如果一个事件触发低电平中断，RFCPEIFG 寄存器中的相应位将设置为 1。每当 RFCPEIFG 中的某个位和 RFCPEIEN 中的相应位都为 1 时，RFCPEISL 中选择的系统级中断就会升高。这意味着中断服务例程 (ISR) 必须清除 RFCPEIFG 中与已经处理的低级中断相对应的位。

节 25.11.2 中 RFCPEIFG 的寄存器描述提供了可用中断的列表。

通过向这些位写入 0 来清除 RFCPEIFG 中的位，而写入 1 的任何位都保持不变。

备注

当清除 RFCPEIFG 寄存器中的位时，如果执行读-修改-写操作，中断可能会丢失，因为在读和写操作之间变为活动的中断标志可能会丢失。因此，应按以下方式清除中断标志：

```
HWREG(RFC_DBELL_BASE + RFC_DBELL_O_RFCPEIFG) = ~(1 << irq_no);
```

而不是以下方式：

```
HWREG(RFC_DBELL_BASE + RFC_DBELL_O_RFCPEIFG) &= ~(1 << irq_no); // wrong
```


English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.2.3.2 射频内核硬件中断

系统级中断 RF_HW 可以由射频内核硬件产生的多个低级中断产生。通过使用 RFHWEN 寄存器，可以针对每个源开启或关闭系统级中断的产生。

当发生触发低级中断的事件时，RFHWIFG 寄存器中的相应位会设置为 1。每当 RFHWIFG 中的某位和 RFHWIEN 中的对应位同时为 1 时，就会引发 RF_HW 中断。这意味着 ISR 应清除 RFHWIFG 中与已处理低级中断相对应的位。

节 25.11.2 中的 RFHWIFG 寄存器说明提供了可用中断的列表。通常，TI 不建议在系统 CPU 中处理这些中断，但可用的无线电计时器通道中断可以通过此方式处理。

清除 RFHWIFG 中的位通过向这些位写入 0 来完成，而任何写入 1 的位保持不变。

备注

清除 RFHWIFG 中的位时，如果完成了读取-修改-写入操作，中断可能会丢失。因此，RFHWIFG 寄存器适用与 RFCPEIFG 相同的规则（请参阅节 25.2.3.1）。

25.2.3.3 RF 内核命令确认中断

当 RF 内核命令被确认时，即状态在 CMDSTA 中变为可用时，会生成系统级中断 RF_CMD_ACK（请参阅节 25.11.2）。当状态变为可用时，RFACKIFG.ACKFLAG 寄存器位被设置为 1。只要该位为 1，就会产生 RF_CMD_ACK 中断，这意味着 ISR 在处理 RF_CMD_ACK 中断时必须清除 RFACKIFG.ACKFLAG。

25.2.4 无线电计时器

无线电拥有自己的专用计时器，即无线电计时器 (RAT) 模块。RAT 是一个运行频率为 4MHz 的自由运行 32 位计时器。RAT 具有八个通道，具备比较和捕获功能。其中五个通道保留给无线电 CPU，其余三个通道可供系统 CPU 使用。可用通道的编号为 5、6 和 7。

RAT 仅在射频内核加电时才能运行。RAT 必须通过 CMD_START_RAT 或 CMD_SYNC_START_RAT 命令启动。要运行带延迟启动的无线电操作命令，或任何运行接收器或传输器的无线电操作命令，无线电计时器必须正在运行。

当 RAT 运行时，可以从 RATCNT 寄存器读取计时器的当前值（请参阅节 25.11.1）。

25.2.4.1 比较和捕获事件

可用的通道可设置成比较模式或捕获模式。

可以使用 CMD_SET_RAT_CMP 命令设置比较模式（请参阅节 25.3.3.2.10）。在这种情况下，当计数器达到 compareTime 给出的值时，计时器会生成中断。中断被映射到 RFHWIFG（请参阅节 25.2.3.2 和节 25.11.2）。对于可用的 RAT 通道，使用的中断标志为 RATCH5、RATCH6 和 RATCH7。或者，当计数器达到 compareTime 给出的值时，也可以控制 I/O 引脚（请参阅节 25.2.4.2）。发送 CMD_SET_RAT_CMP 命令时，compareTime 的值存储在与所选通道相对应的无线电通道值寄存器 (RATCHnVAL) 中（请参阅节 25.11.2）。

捕获模式可用于捕获输入引脚上的转换并且记录转换发生时的 RAT 计数器值。可以使用 CMD_SET_RAT_CPT 命令设置比较模式（请参阅节 25.3.3.2.11）。转换发生时，RAT 的当前值存储在与所选通道相对应的 RATCHnVAL 寄存器中（请参阅节 25.11.1），计时器会产生中断。至于比较模式，中断会映射到 RFHWIFG。对于可用的 RAT 通道，使用的中断标志为 RATCH5、RATCH6 和 RATCH7。如果在 CMD_SET_CPT 中配置了单次捕获模式，则仅捕获第一次转换，除非通道再次处于准备状态，如下一段所述。如果配置了重复模式，则捕获每次转换。

备注

在这种情况下，如果发生新的转换，RATCHnVAL 寄存器中捕获的值可能会随时被覆盖。

在比较模式或单次捕获模式下设置的通道可以处于启用或禁用状态。发送 CMD_SET_RAT_CMP 或 CMD_SET_RAT_CPT 时，通道会自动启用，而当捕获或比较事件发生时，通道会自动禁用。禁用的通道不会产

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

生任何中断或导致捕获任何计时器值。此外，可以使用 CMD_ARM_RAT_CH 或 CMD_DISARM_RAT_CH 启用或禁用通道（请参阅节 25.3.3.2.14 至节 25.3.3.2.15）。禁用时，通道保持其配置。要禁用不会使用相同配置重新启用的通道，可以使用 CMD_DISABLE_RAT_CH 命令（请参阅节 25.3.3.2.12）。

25.2.4.2 无线电计时器输出

RAT 模块有四个可控输出，即 RAT_GPO0 至 RAT_GPO3。这些信号可以由其中一个 RAT 通道控制，并使用 SYSGPOCTL 寄存器映射到 IOC 可用的信号（请参阅节 25.11.2）。信号 RAT_GPO0 保留用于启动发射器，并由无线电 CPU 在内部控制（请参阅节 25.3.2.8）。其他三个信号可以使用 CMD_SET_RAT_OUTPUT 命令进行配置（请参阅节 25.3.3.2.11）。不同的输出模式指示当所选 RAT 通道上发生中断时输出的转换，始终为 0 和始终为 1 配置除外，它们会立即生效并可用于初始化。

25.2.4.3 与实时时钟同步

当无线电断电时，RAT 模块不计数。为了使同步协议的时基保持一致，可以将无线电计时器与实时时钟 (RTC) 同步（请参阅章节 16）。

为了在上电后允许同步，必须在 RF 内核断电之前发送 CMD_SYNC_STOP_RAT 命令（请参阅节 25.3.3.1.10）。此命令（直到下一个 RTC 节拍）停止无线电计时器并返回参数 rat0，在无线电计时器重新启动时，应存储并提供该参数。

下次 RF 内核上电并启动 RAT 时，必须使用 CMD_SYNC_START_RAT（请参阅节 25.3.3.1.11）完成此同步，其中必须提供从 CMD_SYNC_STOP_RAT 获取的 rat0 参数。此命令用于启动 RAT，等待 RTC 节拍，并调整 RAT。根据应用的不同，每次无线电断电时可能没有必要运行 CMD_SYNC_STOP_RAT 命令；rat0 的先前值可能会被重复使用。不过，在某些情况下，如果无线电长时间通电并且低频和高频晶体振荡器相对于彼此具有显著的误差，这可能会导致问题。

为了获得精确同步，系统必须在运行 CMD_SYNC_START_RAT 命令之前开始在高频晶体振荡器上运行，并扩展到 CMD_SYNC_STOP_RAT 命令完成后运行。

备注

对于 CMD_SYNC_START_RAT 和 CMD_SYNC_STOP_RAT 命令，AON_RTC:CTL RTC_UPD_EN 寄存器位必须设置为 1（请参阅节 16.4.1）。无需将该位复位为 0；在 RTC 启动时，该位可永久设置为 1。

25.3 RF 内核 HAL

射频内核通过向系统 CPU 提供统一的 HAL，隐藏了无线电操作的复杂性。

备注

为确保最佳的无线电性能，请始终使用 TI 提供的最新无线电补丁。有关最新补丁，请参阅 www.ti.com 上的产品页面。

25.3.1 硬件支持

无线电 HAL 由硬件提供支持，具体通过射频内核区域的无线电门铃模块和系统 RAM 中的命令描述符实现。

25.3.2 固件支持

射频内核接受一组高级原语。节 25.3.2.1 至节 25.3.2.8 简要介绍了所需的功能。

25.3.2.1 命令

无线电 CPU 允许用户从系统 CPU 运行一组高级原语或命令。通过 CMDR 寄存器发出命令后，无线电 CPU 会对其进行检查并决定操作过程。

可发出三类命令：

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

- 无线电操作命令
- 即时命令
- 直接命令

对于前两类命令，CMDR 包含指向命令结构的指针。该指针必须是 32 位字对齐的有效指针，因此 2 个最低有效位 (LSB) 必须为 0 0，如图 25-3 所示。通过将 2 个 LSB 设置为 01 或 10 并将命令 ID 编号放在 CMDR 的位 16 至 31 中，可发出直接命令。位 8 至位 15 或位 2 至 15 可用作可选的字节参数。图 25-4 展示了直接命令的格式。

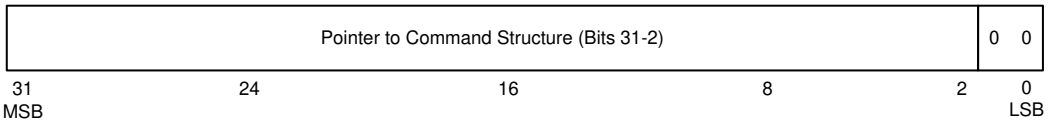


图 25-3. 用于无线操作命令和即时命令的 CMDR 寄存器

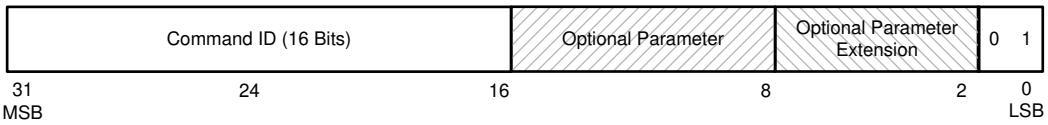


图 25-4. 用于直接命令的 CMDR 寄存器

无线电操作命令和即时命令的 CMDR 寄存器所指向的数据结构可能位于系统 RAM 或无线电 RAM 中。系统 CPU 必须确保正在使用的存储器区域可供自由访问，特别是在使用无线电 RAM 时，因为其中部分存储器空间是保留给无线电 CPU 使用的。此信息可通过 CMD_GET_FW_INFO 命令获取（请参阅节 25.3.3.2.6）。命令格式遵循节 25.3.2.6 和节 25.3.2.6.1 中给出的结构，并且每条命令有更详细的定义。

在决定将数据放入哪个存储器区域时，请考虑哪些模块可能会被断电：

- 无线电 RAM 随时可供无线电点 CPU 访问，但在无线电断电时不具备保持能力。因此，必须保持的数据必须在无线电加电或断电时分别复制进无线电 RAM 或从中复制出来。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

- 系统 RAM 在大多数低功耗模式下具有保持能力。如果系统侧断电，无线电 CPU 会请求重新为其加电以访问 RAM。无线电 CPU 访问系统 RAM 时的有功电流消耗高于访问无线电 RAM 时的电流消耗，尤其是在系统侧本可以处于断电状态的情况下。
- 将所有数据结构都放入无线电 RAM 中，并在无线电 CPU 运行时断开系统侧电源，可获得最低的峰值功耗。在某些情况下，将数据结构放入系统 RAM 中可能会获得更低的平均功耗，因为此时所需的复制操作更少，且系统侧仍可长时间处于断电状态（例如，当接收器正在进行同步字搜索时）。

无线电操作命令会导致访问无线电硬件。无线电操作命令可以执行一些操作，例如传输或接收数据包、设置无线电硬件寄存器，或进行更复杂、依赖于协议的操作。无线电操作命令通常只能在无线电空闲时发出。

即时命令用于更改或请求无线电状态，或操作 TX 或 RX 数据队列。中间命令可监控收到的信号强度等状态。即时命令可以在任何时间发出，但在许多情况下，仅当无线电操作正在进行时，其响应才有意义。

直接命令是无参数的即时命令，或在某些情况下带有 1 字节或 2 字节参数的即时命令。直接命令通过向 CMDR 寄存器发送具有图 25-4 所示格式的值来发出。16 个最高有效位 (MSB) 包含要运行的即时命令的命令 ID。如果为命令指定了可选参数，可放在位 8 至 15 或位 2 至 15 中。

25.3.2.2 状态命令

发出命令后，无线电 CPU 会更新 CMDSTA 寄存器，从而将 RFCMDACK 中断发送回系统 CPU。此更新在命令完成立即命令和直接命令之后以及在命令针对无线电操作命令进行调度之后发生。在接收到该中断之前，不能发出新命令。CMDSTA 寄存器由 32 位组成；8 个 LSB 提供结果，而高 24 位可用于每条命令中的特定信令。图 25-5 显示了此格式。

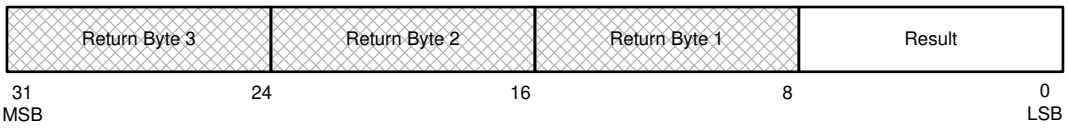


图 25-5. CMDSTA 寄存器的格式

在结果字节中，第 7 位指示是否发生了错误。0x00 的结果字节（表示*挂起*）在发出命令时由无线电门铃硬件自动生成，CMDSTA 寄存器中的其他位也变为 0，这是 RF_CMD_ACK 中断发出之前 CMDSTA 的值。

表 25-2 列出了 CMDSTA 寄存器中的结果字节的值。

表 25-2. CMDSTA 寄存器中结果字节的值

值	名称	说明
无错误		
0x00	待定	命令尚未解析。
0x01	完成	立即命令：命令成功完成。 无线电操作命令：已成功提交命令以供执行。
误差		
0x81	IllegalPointer	CMDR 中发出信号的指针无效。
0x82	UnknownCommand	命令结构中的命令 ID 号未知。
0x83	UnknownDirCommand	直接命令的命令编号未知，或者该命令不是直接命令。
0x85	ContextError	立即或直接命令在不受支持的上下文中发出。
0x86	SchedulingError	在 RF 内核中已运行另一项操作时，尝试调度无线电操作命令。新命令会被拒绝，而已运行的命令不受影响。
0x87	ParError	提交时解析的命令参数中存在错误。对于无线电操作命令，命令启动后解析的参数中的错误通过命令结束来指示，并在该命令结构的状态字段中指示错误。
0x88	QueueError	已尝试对数据条目队列执行操作，但在当前状态下队列不支持该操作。
0x89	QueueBusy	在数据项繁忙时尝试了对该项的操作。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

除命令状态寄存器外，每个无线电操作命令还包含一个状态字段（请参阅表 25-9）。此字段可能具有以下类别中的值。

- 空闲：命令尚未启动。
- 待定：已解析命令，但尚未发生启动触发。
- 激活：命令正在运行。
- 已暂停：该命令处于活动状态，可能会再次变为活动状态。该命令仅受某些 IEEE 802.15.4 命令支持。
- 结束：命令完成后，系统 CPU 可自由修改命令结构或释放内存。
- 已跳过：该命令被跳过而从未执行。

表 25-3 列出了常用命令以及在开始之前解析任何命令时的状态代码。

表 25-3. 常见无线电操作状态代码

编号	名称	说明
操作未完成		
0x0000	IDLE	操作尚未开始。
0x0001	PENDING	等待启动触发。
0x0002	ACTIVE	运行操作。
0x0003	SKIPPED	由于另一条命令中的条件而跳过操作。
操作正常完成		
0x0400	DONE_OK	操作正常结束。
0x0401	DONE_COUNTDOWN	计数器达到零。
0x0402	DONE_RXERR	操作因 CRC 错误而结束。
0x0403	DONE_TIMEOUT	操作以超时结束。
0x0404	DONE_STOPPED	在收到 CMD_STOP 命令后停止操作。
0x0405	DONE_ABORT	由 CMD_ABORT 命令中止操作。
操作完成。存在错误		
0x0800	ERROR_PAST_START	过去发生了启动触发。
0x0801	ERROR_START_TRIG	非法的启动触发参数。
0x0802	ERROR_CONDITION	下一个操作的条件非法。
0x0803	ERROR_PAR	命令特定参数中出现错误。
0x0804	ERROR_POINTER	指向下一个操作的指针无效。
0x0805	ERROR_CMDID	下一个操作的命令 ID 未定义或者不是无线电操作命令。
0x0807	ERROR_NO_SETUP	在没有 CMD_RADIO_SETUP 的情况下尝试使用 RX、TX 或者合成器的操作。
0x0808	ERROR_NO_FS	在没有对合成器进行编程或通电的情况下尝试使用 RX 或者 TX 的运行。
0x0809	ERROR_SYNTH_PROG	合成器编程失败。
0x080A	ERROR_TXUNF	观察到调制解调器 TX 下溢。
0x080B	ERROR_RXOVF	观察到调制解调器 RX 上溢。
0x080C	ERROR_NO_RX	不存在此类数据时从最后一个 RX 请求的数据。

当系统 CPU 准备命令结构时，CPU 应将状态字段初始化为 IDLE。可以循环设置命令。如果是这样，在无线电 CPU 空闲之前，系统 CPU 不得修改命令结构，即使状态已完成或跳过，系统 CPU 也会收到 LAST_COMMAND_DONE 中断（请参阅节 25.11.2）。

25.3.2.3 中断

无线电 CPU 具有 32 个软件中断源，可在系统 CPU 中生成 RFCPE0 和 RFCPE1 中断。中断标志寄存器可判断引发了哪个软件中断，并单独启用中断。此外，当 CMDSTA 更新时，RFCMDACK 中断会自动发出。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

一些软件定义的中断在所有命令中都具有共同的含义；为使用特定中断的每个协议定义了其他每个中断的详细信息。某些中断仅在一个协议中使用，而其他中断则在多个协议中使用。RFCPEIFG 寄存器的说明中列出了这些中断（请参阅节 25.11.2）。

25.3.2.4 传递数据

通过无线方式传输或接收的数据有两种基本方式：直接传输或者通过队列传输。

传递数据的最直接方法是将数据作为命令参数的一部分附加（直接或者通过指针）。确切的格式取决于正在运行的命令；通常、TX 有一个长度字段和一个数据缓冲器，以及 RX 的最大长度、接收长度（如果是可变长度）和接收缓冲器。

队列用于支持无法提前知晓接收或发送的数据包数量的操作。一个操作可以使用一个或多个队列，例如将一个 RX 和一个 TX 队列用于组合 RX/TX 操作。使用队列的任何操作都使用通用系统来维护它们，如节 25.3.2.7 中所述。对于每个无线电命令，都会指明它们是使用队列还是单个缓冲器。

无线电操作命令声明所使用的数据方法。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.3.2.5 命令调度

系统 CPU 负责根据需要调度命令。使用低功耗模式时，系统 CPU 必须在开始下一次操作之前的短时间唤醒（使用 RTC）。

可以通过延迟启动来安排无线电操作命令（请参阅节 25.3.2.5.1）。如果命令有延迟启动，则无线电 CPU 将进入空闲模式，直至命令启动。无线电操作命令在此延迟期间被视为正在运行，除非挂起的命令先中止或停止，否则不能安排其他无线电操作命令。

系统 CPU 可通过在任何无线电操作命令中使用下一个操作指针来调度背对背无线电操作命令。该指针可以指向要在链中执行的下一个命令，通过该方法可以进行复杂的运算。在某些情况下（例如计时器出现错误或到期），下一条命令不会启动。相反，操作将结束，或者可以跳过许多命令（请参阅节 25.3.2.5.2）。如果在运行另一条命令时调度新命令，则系统 CPU 必须等待前一条命令或命令链完成。IEEE 802.15.4 命令对此规则有例外。

当无线电操作命令完成时，无线电 CPU 会向系统 CPU 发出 COMMAND_DONE 中断。如果如前所述链接了许多命令，则在每个命令之后发出 COMMAND_DONE 中断，而 LAST_COMMAND_DONE 中断在链中的最后一条命令之后发出。对于一条非链接命令，LAST_COMMAND_DONE 中断也会在命令之后发出。发出 LAST_COMMAND_DONE 时，同时始终发出 COMMAND_DONE。在发出 COMMAND_DONE 中断之前，无线电 CPU 会将命令结构的状态字段更新为指示命令已完成的状态。发出 COMMAND_DONE 中断后，无线电 CPU 不会访问命令结构。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.3.2.5.1 触发条件

触发条件可用于设置启动时间，或用于特定无线电操作命令中的其他特定目的。有一个通用的触发条件字节，定义见表 25-4。

表 25-4. 触发条件定义字节的格式

位索引	字段	说明
0-3	triggerType	触发条件类型
4	bEnaCmd	0：无替代触发条件命令。 1：CMD_TRIGGER 可用作替代触发条件。
5-6	triggerNo	触发此操作的 CMD_TRIGGER 命令的触发条件编号。
7	pastTrig	0：过去的触发条件绝不会触发，否则在启动命令时会产生错误。 1：过去的触发条件会尽快触发。

triggerType 可取表 25-5 中列出的值。其他值保留。

表 25-5. 支持的触发条件类型

编号	名称	说明
0	TRIG_NOW	现在（不适用于结束触发条件）
1	TRIG_NEVER	从不（但 bEnaCmd = 1 时可能由 CMD_TRIGGER 触发）
2	TRIG_ABSTIME	由计时器参数给出的绝对时间
3	TRIG_REL_SUBMIT	相对于命令提交时间的时间
4	TRIG_REL_START	相对于此命令开始时间的时间（不允许用于启动触发条件）
5	TRIG_REL_PREVSTART	相对于上一条命令开始时间的时间
6	TRIG_REL_FIRSTSTART	相对于链中第一条命令开始时间的时间
7	TRIG_REL_PREVEND	相对于上一条命令结束时间的时间
8	TRIG_REL_EVT1	相对于上一条命令的事件 1 的时间
9	TRIG_REL_EVT2	相对于上一条命令的事件 2 的时间
10	TRIG_EXTERNAL	在 RAT 的外部触发输入上

除 TRIG_NOW 和 TRIG_NEVER 之外，所有触发条件都会使用一个 32 位时间参数。绝对时序使用 32 位 RAT 的值。相对时序使用 RAT 节拍数。外部触发条件使用源和边沿的标识符，定义见表 25-6。

表 25-6. 外部事件触发条件的时间参数字段

位索引	字段	说明
0-1		保留
2-3	inputMode	输入模式 00：上升沿 01：下降沿 10：两个边沿 11：保留
4-7		保留
8-12	source	22：RFC_GPI0 23：RFC_GPI1 其他：保留
13-31		保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

相对时序可以相对于提交命令链的时间、命令的启动时间、前一个或第一个命令的启动时间，或命令内某些观察到的事件，需要针对每个命令进行定义。以下规则适用：

- 对于链中的第一条命令，如果启动触发条件是类型 5 到 9 中的任何一种，则立即启动。如果链中第一条命令引用的其他触发条件是类型 5 到 9 中的任何一种，则触发时间为相对于命令提交的时间。
- 如果命令的启动触发条件是 TRIG_REL_START，则会产生错误。
- 如果命令的启动触发条件是 TRIG_NEVER 且 bEnaCmd 为 0，则会产生错误。
- 一些无线电操作命令定义了事件 1 和事件 2。这些事件依赖于上下文，可被无线电 CPU 观察到。有关该上下文中的定义，请参阅每条命令的说明。如果未定义，这些事件即为命令启动的时间。

如果 bEnaCmd 为 1，该操作也可以由命令触发（CMD_TRIGGER 命令，请参阅节 25.3.3.2.5）。triggerNo 参数标识该命令的触发条件编号。

如果在评估时触发条件发生在过去，则其行为取决于 pastTrig 位。如果该位为 0，则触发条件不会发生，否则对于启动触发条件，将会产生错误。如果该位为 1，则触发条件会尽快发生。如果启动触发条件的 pastTrig 位为 1，则相对于命令启动时间的时序是相对于编程的启动时间，而不是实际的启动时间。

对于外部触发条件，无线电 CPU 会设置 RAT 以使用选定的输入事件作为单次捕获触发条件；然后 CPU 使用此捕获中断来触发操作。如果事件发生在设置完成之前，则不会捕获该事件，并且会忽略 pastTrig 位。

25.3.2.5.2 条件执行

命令的执行可能取决于上一条命令的结果。对于每个命令，可能有三个结果：

- TRUE
- FALSE
- ABORT

为每个命令定义条件。如果未定义，则结果为 TRUE，除非命令以错误结束、在这种情况下，结果为 ABORT。

每个命令结构都包含运行下一个命令的条件。条件字节如表 25-7 中所示。如果规则为 COND_SKIP_ON_FALSE 或 COND_SKIP_ON_TRUE，则在 nSkip 字段中指示要跳过的命令数量。如果跳过数为零，则重新运行相同的命令。如果跳过数为 1，请在链中运行下一个命令。如果跳过次数是两次，请在下一次跳过后运行命令，依此类推。如果规则为 COND_NEVER 并且以前没有任何命令使用跳过，则下一个命令指针将被忽略，并且可能为 NULL。

表 25-7. 条件字节的格式

位索引	字段名称	说明
0-3	规则	如何继续的规则，如表 25-8 中所定义
4-7	nSkip	如果规则涉及跳过，则跳过次数 + 1 0：相同、1：下一个、2：跳过下一个...

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-8. 条件规则

编号	名称	说明
0	COND_ALWAYS	始终运行下一个命令（ABORT 时除外）。
1	COND_NEVER	从不运行下一个命令（下一个命令指针仍可用于跳过）。
2	COND_STOP_ON_FALSE	如果此命令返回 TRUE，则运行下一个命令，如果返回 FALSE，则停止。
3	COND_STOP_ON_TRUE	如果此命令返回 TRUE，则停止，如果返回 FALSE，则运行下一个命令。
4	COND_SKIP_ON_FALSE	如果此命令返回 TRUE 则运行下一个命令，如果返回 FALSE 则跳过许多命令。
5	COND_SKIP_ON_TRUE	如果此命令返回 TRUE，则跳过许多命令；如果返回 FALSE，则运行下一个命令。

如果执行停止，无线电 CPU 将恢复空闲状态，在通过 CMDR 寄存器输入新命令之前，不会运行任何进一步的命令。发出 LAST_COMMAND_DONE 中断。

如果命令以 ABORT 结果结束，则无论条件如何，执行都将结束。发出 LAST_COMMAND_DONE 中断。ABORT 结果的条件的一个示例是发出了 CMD_ABORT 命令。

25.3.2.5.3 命令启动前的处理

对于所有无线电操作命令，在解析命令的其余部分之前，需要先检查启动触发条件和条件代码。如果启动触发条件的触发类型非法（包括不允许用于启动触发条件的 TRIG_REL_START，以及 TRIG_NEVER 与无命令触发条件结合使用），无线电 CPU 会将状态字段设置为 ERROR_START_TRIG。如果条件字段的值非法，无线电 CPU 会将状态字段设置为 ERROR_CONDITION。如果启动触发条件发生在过去且 startTrigger.pastTrig 为 0，无线电 CPU 会将状态字段设置为 ERROR_PAST_START。

25.3.2.6 命令数据结构

本章各表格中列出了数据结构。字节索引是相对于指向该结构的指针的偏移量。多字节字段采用小端字节序，且根据字段大小要求进行 16 位半字或 32 位字对齐。对于位编号，0 表示 LSB。R/W 列的含义如下：

R：系统 CPU 可以读回结果；无线电 CPU 不读取该字段。

W: 系统 CPU 写入一个值，无线电 CPU 读取它且不会修改。

R/W：系统 CPU 写入初始值，无线电 CPU 可能会修改它。

对于从另一数据结构特化而来的数据结构，来自父结构的字段不会重复列出，但“字节索引”列会体现它们的存在。

所有命令的唯一强制字段是命令 ID 编号，该编号为 16 位数，作为命令结构的前 2 个字节发送。

某些即时命令具有额外的字段，这些字段针对每条命令单独定义。无线电操作命令具有额外的强制字段，定义见表 25-9。

所有标记为*保留*的命令字段都应写入 0。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.3.2.6.1 无线电操作命令结构

表 25-9 列出了无线电操作命令的命令结构。某些命令在此后附加了其他字段。

表 25-9. 无线电操作命令格式

字节索引	字段名称	位	位字段名称	类型	说明
0-1	commandNo			W	命令 ID 编号
2-3	状态			R/W	指示命令状态的整数。此值在运行期间通过无线电 CPU 更新，可随时由系统 CPU 读取。
4-7	pNextOp			W	指向此操作完成后要运行的下一个操作的指针
8-11	startTime			W	绝对或者相对开始时间（取决于 startTrigger 字段的值）
12	startTrigger				启动操作的触发器的标识
13	条件			W	运行下一个操作的条件

25.3.2.7 数据输入结构

数据条目必须属于队列。队列作为无线电操作命令的命令结构的一部分设置。

有关作为命令可用的队列的操作，请参阅节 25.3.4。

25.3.2.7.1 数据条目队列

使用队列的任何命令都包含一个指向数据条目队列结构的指针，如
表 25-10 所示。系统 CPU 分配并初始化此队列结构。

表 25-10. 数据条目队列结构

字节索引	字段名称	位	位字段名称	类型	说明
0-3	pCurrEntry			R/W	指向射频 CPU 当前正在使用的数据条目的指针（如果射频未使用队列，则指向下一个待使用的条目）。NULL 表示队列中当前没有缓冲区。
4-7	pLastEntry			R/W	指向在此队列中最后一个入队条目的指针。如果 pCurrEntry 不为 NULL 且 pLastEntry 为 NULL，则可能无法追加后续条目。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.3.2.7.2 数据条目

数据条目队列包含表 25-11 中列出的类型的数据条目。这些条目组织在链接列表中。队列结构的 pCurrEntry 字段指向队列的第一个条目（请参阅表 25-10）。每个 pNextEntry 字段都指向下一个条目。队列结构的 pLastEntry 字段也指向队列中的最后一个条目。

表 25-11. 一般数据条目结构

字节索引	字段名称	位	位字段名称	类型	说明
0-3	pNextEntry			R/W	指向队列中的下一个条目的指针，如果这是最后一个条目，则为 NULL
4	状态			R/W	指示条目的状态，包括是否可以自由接收来自系统 CPU 的写入
5	config	0-1	类型	W	数据条目结构类型 0：常规数据条目 1：保留 2：指针条目 3：部分读取 RX 条目
		2-3	lenSz	W	每个 RX 条目元素开头的长度字的大小 0：无长度指示器 1：1 字节长度指示器 2：2 字节长度指示器 3：保留
		4-7	irqIntv	W	仅对于部分读取 RX 条目：无线电 CPU 生成的中断之间的字节数（0000：16 字节）
6-7	length			W	数据缓冲器的数据字段长度或指针条目长度。对于 TX 条目，这对应于一个条目元素（数据包）。对于 RX 条目，这将提供总可用存储空间。
8 – (7+n)	data			R/W	要接收或发送的数据数组 （n = 长度）

状态字段可以采用以下值：

- 0：待定：无线电 CPU 尚未使用该条目。这是在提交条目之前由系统 CPU 写入的状态。
- 1：激活：该条目是当前队列中打开的条目，可供无线电 CPU 进行写入 (RX) 或读取 (TX)。
- 2：忙：正在进行的无线电操作正在写入或读取未完成的数据包。当条目处于此状态时，不允许执行某些操作（请参阅节 25.3.4）。
- 3：结束：无线电 CPU 已完成将数据写入此条目的操作，并且可供系统 CPU 重复使用或释放存储器（如果动态分配）。

对于数据条目，系统 CPU 在系统 RAM 或无线电 RAM 的可用部分中设置所需的数据结构。如果数据结构是动态分配的，则系统 CPU 会在使用后释放内存。

在用于接收数据的条目中，无线电 CPU 可以用长度指示符启动条目元素。如果 config.lenSz 为 00，则不会写入此类指示器。仅当接收到的数据包的长度可以通过其他方式确定时，才必须使用此选项。如果 config.lenSz 为 01，则 1 个字节表示长度字节之后的字节数。只有在没有将超过 255 字节的元素写入条目时，才可以使用这种方法。如果 config.lenSz 为 10，则 16 位字将指示长度字之后的字节数。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.3.2.7.3 指针条目

指针条目是一个条目，其中数据不包含在条目本身中，但该条目保存指向缓冲器的指针。这种条目通过将 config.type 设置成 2 来指示。指针将替换数据字段，如表 25-12 中所示。

表 25-12. 指针入口结构中的指针字段

字节索引	字段名称	位	位字段名称	类型	说明
8-11	pData			W	指向大小长度字节的数据缓冲器的指针

数据从 pData 提供的缓冲器中读取或存储在其中。该缓冲器的大小由长度给出，和通用数据条目中数据字段的大小一样。

25.3.2.7.4 部分读取 RX 条目

专有模式支持一种 RX 条目，可以在整个数据包通过无线方式接收完成之前读取数据。该条目可用于以下用途：

- 必须在接收到整个数据包之前读取数据时
- 当数据包长度在数据包开头未知时
- 当数据包的长度过长，无法将整个有效载荷同时保存在内存中时

为了支持此功能，使用了表 25-11 中结构的一种特殊变体。多个条目元素可以包含在同一条目中。每个条目元素对应于通过无线方式接收的一个数据包或其中的一部分。该元素还可能包含附加字段。可以通过将 config.type 设置为 3 来选择此类型。在条目中，数据字段的组成如表 25-13 所示（索引相对于整个条目结构）。

表 25-13. 部分读取 RX 条目中的字段

字节索引	字节字段名称	位	位字段名称	类型	说明
8-9	pktStatus	0-12	numElements	R	条目中提交的条目元素的数量
		13	bEntryOpen	R	该条目包含一个仍可附加数据的元素。
		14	bFirstCont	R	第一个元素是上一条目中最后一个数据包的延续。
		15	bLastCont	R	最后一个元素中的数据包在下一个条目中继续。
10-11	nextIndex			R	指向无线电 CPU 提交的最后一个条目元素最后一个字节之后字节的索引
12 – (7+n)	rxData			R	接收的数据。确切的格式取决于正在运行的操作。每个条目元素可能以一个长度字节或字开头。

条目的更新方式如下：

- nextIndex 字段会随着新字节写入缓冲区而更新。
- 接收数据包时，无线电 CPU 将 pktStatus.bEntryOpen 设置为 1。

当条目元素完成时（由于数据包结束或由于该元素到达条目的末尾），无线电 CPU 会将 pktStatus.bEntryOpen 设置为 0，并使 pktStatus.numElements 递增。如果数据包在下一个条目中继续，则无线电 CPU 会将 pktStatus.bLastCont 设置为 1。在这种情况下，无线电 CPU 也会将下一个条目的 pktStatus.bFirstCont 位设置为 1。如果没有可用的下一个条目，则状态设置为“未完成”，否则设置为“已完成”。

在条目元素开头指定的长度字段（取决于 config.lenSz）提供条目内该条目元素的长度，而不是整个数据包的长度。如果条目打开时长度未知，则长度字段将写入条目的剩余长度，并在条目完成之前由无线电 CPU 进行更新。

对于部分读取 RX 条目，每当向该条目写入一个或多个字节时，无线电 CPU 会向系统 CPU 生成一个 Rx_Data_Written 中断。此外，当自上次中断或条目元素开始以来已写入 k 个字节时，它还会生成 Rx_N_Data_Written 中断，其中 k 由 config.irqIntv 给出。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.3.2.8 外部信令

无线电 CPU 可以控制信号 CPEGPO0 和 CPEGPO1。为了控制外部前端，当必须启用 LNA 时，CPEGPO0 为高电平，否则为低电平。当 PA 必须启用时，CPEGPO1 为高电平，否则为低电平。无线电 CPU 还控制信号 CPEGPO2 在合成器校准开始时变为高电平，在校准完成时变为低电平。该控制可用于调试。

无线电计时器的两个输出信号具有可以用于观察的自动配置。发起数据包传输时，信号 RATGPO0 变为高电平，发送完成时，信号变为低电平。为了实现数据包传输的准确时序，可以观察 RATGPO0 信号，因为内部使用相同的信号。信号与 CPEGPO1 非常相似，但它会提前几微秒变为高电平，与调制解调器发出的第一个传输符号相比，时序更准确。不过，建议使用 CPEGPO1 控制外部 PA，以避免过早将其导通。当在接收器中找到同步时，信号 RATGPO1 可配置为高电平；当接收到数据包或接收中止时，该信号可配置为低电平（这对 IEEE 802.15.4 接收器命令不起作用）。要将 RATGPO1 映射到 RFC_GPO 信号，需要额外的覆盖功能。可以使用 CMD_SET_RAT_OUTPUT 将其他无线电计时器输出配置成根据需要生成事件。

默认情况下，无线电 CPU 在引导时将 CPEGPO0 映射到信号 RFC_GPO0，将 CPEGPO1 映射到信号 RFC_GPO1，将 CPEGPO2 映射到信号 RFC_GPO2，将 RATGPO0 映射到信号 RFC_GPO3。此映射可以通过写入 RFC_DBELL:SYSGPOCTL 寄存器来修改。

可以使用系统 I/O 控制器将 RFC_GPO 信号映射至输出引脚。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.3.3 命令定义

有一组独立于当前射频协议的命令。这些命令与无线电的低级操作相关。

25.3.3.1 与协议无关的无线电操作命令

对于后面各小节（至）中介绍的无线电操作命令（节 25.3.3.1.1 至 节 25.3.3.1.14 ），操作将由于表 25-14 中列出的原因之一或为每个命令列出的其他状态而结束。操作结束后，命令结构的状态字段会指示操作结束的原因。在每种情况下，都指示结果是 TRUE、FALSE 还是 ABORT（请参阅节 25.3.2.5.2 ）。此结果指示是启动 pNextOp 中指示的下一个命令（如果有），还是返回到 IDLE 状态。

表 25-14. 无线电操作结束命令

条件	状态代码	结果
已完成操作	DONE_OK	TRUE
在等待启动触发期间接收到 CMD_STOP	DONE_STOPPED	FALSE
收到了 CMD_ABORT	DONE_ABORT	ABORT
启动触发发生在过去，其中 startTrigger.pastTrig = 0	ERROR_PAST_START	ABORT
非法的启动触发参数	ERROR_START_TRIG	ABORT
下一个操作的条件非法	ERROR_CONDITION	ABORT
观察到的非法参数	ERROR_PAR	ABORT
指向下一个操作的指针无效	ERROR_POINTER	ABORT
下一个操作的命令 ID 未定义或者不是无线电操作命令	ERROR_CMDID	ABORT
在没有 CMD_RADIO_SETUP 的情况下尝试使用 RX、TX 或者合成器的操作	ERROR_NO_SETUP	ABORT
在没有对合成器进行编程的情况下尝试使用 RX 或者 TX 的运行	ERROR_NO_FS	ABORT

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.3.3.1.1 CMD_NOP：无操作命令

命令 ID 编号：0x0801

CMD_NOP 是一条无线电操作命令，仅接受表 25-9 中列出的强制参数。该命令仅等待启动触发条件，然后结束。该命令可用于测试系统 CPU 与无线电 CPU 之间的通信，或者插入一段等待时间。

25.3.3.1.2 CMD_RADIO_SETUP：设置无线电设置命令

命令 ID 编号：0x0802

CMD_RADIO_SETUP 是无线电操作命令。除了表 25-9 中列出的参数外，命令结构还包含表 25-15 中列出的字段。

表 25-15. CMD_RADIO_SETUP 命令格式

字节索引	字段	位索引	位字段名称	类型	说明
14	mode			W	这是要使用的主模式。 0x00：低功耗蓝牙 0x01：IEEE 802.15.4 0x02：2-Mbps GFSK 0x05：5Mbps 编码 8-FSK 0xFF：保留现有模式；仅更新覆盖设置。
15	loDivider			W	要使用的分频器设置。有关每个器件和频段的建议设置，请参阅 Smart RF Studio。 某些芯片版本的设置范围有限。
16-17	config	0-2	frontEndMode		0x00：差分模式 0x01：单端模式 RFP 0x02：单端模式 RFN 其他：保留
		3	biasMode	W	0：内部偏置 1：外部偏置
		4-9	analogCfgMode	W	0x00：写入模拟配置。引导后首次配置以及更改频段或前端配置时要求使用。 0x2D：保留模拟配置。可在待机后或在相同频段和前端配置下更改模式时使用。 其他：保留
		10	bNoFsPowerup	W	0：给频率合成器上电。 1：不给频率合成器上电。
		11	lnaIbBoost		使用 SmartRF Studio 中的设置
		12	bSynthNarrowBand		使用 SmartRF Studio 中的设置
		13-15			保留
18-19	txPower			W	输出功率设置；使用 SmartRF Studio 的值。如需了解更多详情，请参阅节 25.3.3.2.16。 0xFFFF：使用 20dBm PA
20-23	pRegOverride			W	指向要覆盖的硬件和配置寄存器列表的指针。如果为 NULL，则不进行任何覆盖。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

在启动时，无线电 CPU 会根据设置 mode.radioMode 给定工作模式设置参数，并应用 pRegOverride 中的修改。pRegOverride 是一个指针，其指向的结构包含某些硬件寄存器的覆盖值、由无线电 CPU 控制的无线电配置以及协议相关变量。如果 pRegOverride 为 NULL，则不会覆盖任何寄存器。覆盖值结构是由 TI 提供或由 SmartRF Studio 生成的一个包含 32 位条目的字符串。

在使用任何涉及接收器、发送器或频率合成器的命令之前，必须运行 CMD_RADIO_SETUP 或其他无线电设置命令。如果射频内核复位，则必须重新运行 CMD_RADIO_SETUP。

当 CMD_RADIO_SETUP 执行时，修整值将从 FCFG1 中读取，除非已在其他地方提供。如果这些值是从 FCFG1 读取的，则在 CMD_RADIO_SETUP 执行期间存在以下限制：

VIMS 模块必须通电，以允许闪存读取。

如果违反了之前定义的限制，内部系统总线可能最终处于无响应状态。要退出此状态，需要系统复位。为了确保在运行 CMD_RADIO_SETUP 时可以读取 FCFG1，TI 提供的 RF 驱动程序会在运行无线电设置命令时自动使能 VIMS。

表 25-16 列出了硬件寄存器覆盖条目的格式。表 25-17 列出了数组启动器的格式。表 25-18 列出了 ADI 寄存器覆盖条目的格式。表 25-19 列出了固件定义的参数覆盖条目的格式。表 25-20 列出了 MCE/RFE 覆盖模式条目的格式。表 25-21 列出了中心频率条目的格式。

每次数据包传输开始时都会存储并应用 txPower 参数，以设置具有温度补偿的输出功率。稍后可以使用 CMD_SET_TX_POWER 命令更改此设置（请参阅节 25.3.3.2.16）。如果 txPower 为 0xFFFF，则选择 20dBm PA 而不是常规 PA。此设置仅允许在带有“P”后缀的器件上使用。如果使用此设置，则必须按表 25-22 的定义进行 20dBm PA 功率覆盖；如果缺少该覆盖，设置命令将以错误结束。

表 25-16. 硬件寄存器覆盖条目的格式

位索引	位字段名称	说明
0-1	entryType	00：硬件寄存器 01：数组启动器，请参阅表 25-17 10：ADI 寄存器，请参阅表 25-18，或 MCE/RFE 覆盖 11：固件定义参数，请参阅表 25-19
2-15	hwAddr	硬件寄存器地址的位 2 – 15。地址的位 0 – 1 为 0。
16-31	value	写入寄存器的值。

表 25-17. 数组启动器的格式

位索引	位字段名称	说明
0-1	entryType	01：数组启动器
2-15	startAddr	要写入的第一个地址或索引： 硬件寄存器：地址的位 2 – 15（位 0 – 1 为 0） ADI 寄存器：ADI 总线地址，位 6 中的半字节指示器，位 7 中的 ADI 选择器 固件定义参数：字节索引
16-29	length	条目数
30-31	arrayType	数组类型： 00：具有 16 位值的硬件寄存器 01：具有 32 位值的硬件寄存器 10：ADI 寄存器 11：固件定义的参数

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-18. ADI 寄存器覆盖条目的格式

位索引	位字段名称	说明
0-1	entryType	10：ADI 寄存器
2-9	adiValue2	可选的第二个写入值
10-15	adiAddr2	可选的第二个 ADI 总线地址
16-23	adiValue	要写入寄存器的值
24-29	adiAddr	ADI 总线地址
30	bHalfSize	0：使用全大小写入 1：使用半大小写入，从而实现读取-修改-写入功能
31	adiNo	0：写入 ADI 0 (RF) 1：写入 ADI 1（合成器）

表 25-19. 固件定义的参数覆盖条目的格式

位索引	位字段名称	说明
0-1	entryType	11：固件定义的参数
2-3	entrySubType	00：固件定义的参数 01：MCE/RFE 覆盖模式（必须在第一个条目中），请参阅表 25-20 10：保留 11：覆盖列表结束
4-14	fwAddr	参数结构中的字节索引
15	bByte	0：16 位值 1：8 位值
16-31	value	写入参数的值

表 25-20. MCE/RFE 覆盖模式条目的格式

位索引	位字段名称	说明
0-1	entryType	11：固件定义的参数
2-3	entrySubType	01：MCE/RFE 覆盖模式
4	bMceCopyRam	如果为 1，则在 MCE 完成设置后，将 mceRomBank 提供的 MDM ROM 组的内容复制到 RAM。
5	bRfeCopyRam	如果为 1，则在 MCE 完成设置后，将 rfeRomBank 提供的 RFE ROM 组的内容复制到 RAM。
6	bMceUseRam	0：从 ROM 运行 MCE 1：从 RAM 运行 MCE
7-10	mceRomBank	要从中运行的 MCE ROM 组
11	bRfeUseRam	0：从 ROM 运行 RFE 1：从 RAM 运行 RFE
12-15	rfeRomBank	要从中运行的 RFE ROM 组
16-23	mceMode	要发送到 MCE 的模式
24-31	rfeMode	要发送到 RFE 的模式

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-21. 中心频率条目的格式

位索引	位字段名称	说明
0-1	entryType	11：固件定义的参数
2-3	entrySubType	10：特殊配置
4-7	specialType	0001：中心频率条目
8		保留
9	bAutoTxIf	如果为 1，则将 TX IF 设置为 RX IF。
10	bApplyRx	如果为 1，则使用 invRfFreq 重新计算 RX IF。
11	bApplyTx	如果为 1，则使用 invRfFreq 重新计算 TX 形状。
12-31	invRfFreq	fRFMHz 为中心频率（以 MHz 为单位）时的值： (12 × 24 × 220)/(fRFMHz × loDivider)

表 25-22. 20dBm PA TX 功率进入

位索引	位字段名称	说明
0-1	entryType	11：固件定义的参数
2-3	entrySubType	10：特殊配置
4-7	specialType	0010：20dBm PA TX 功率进入
8-9		保留
10-15	IB	新 TX 功率设置。TI 建议使用 SmartRF Studio 中的值。在 25°C 下写入 PA 功率控制字段的值。有关详细信息，请参阅方程式 14。
16-17	ibBoost	要写入 PA 的偏置控制字段的值
18	boost	输入 PA 的驱动强度 0：低驱动器强度 1：高驱动强度
19-25	tempCoeff	IB 的温度系数 0：无温度补偿
26-31	paLdoTrim	要写入 20dBm PA LDO 输出电压控制的值

表 25-23. 列表结束条目的格式

位索引	位字段名称	说明
0-1	entryType	11：固件定义的参数
2-3	entrySubType	11：列表段结束
4-7	nextEntryRegion	0x0：列表结束 0x1：SRAM。基址 = 0x2000 0000 0x2：RF 内核 RAM。基址 = 0x2100 0000 0x3：闪存。基址 = 0xA000 0000 0xF：列表结束 其他：保留
8-31	addrOffset	下一列表部分的地址偏移量。下一地址为： 基址 + (addrOffset × 4)

使用 SmartRF Studio 中的 Code Export 功能可以为不同的 PHY 和设置生成覆盖列表。

如果 pRegOverride 中的指针无效，则任何覆盖条目都无效。如果数组的长度过大或为零，则操作以 ERROR_PAR 状态结束。如果 config.bNoFsPowerup = 0 并且合成器上电失败，该命令将以 ERROR_SYNTH_PROG 状态结束。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

如果在等待启动触发器时接收到 CMD_ABORT 或 CMD_STOP，则操作结束，不进行任何设置。如果在启动触发器生效之后接收到 CMD_STOP，设置会继续进行，直到完成。如果在启动触发器生效之后接收到 CMD_ABORT，则设置过程会中止。这样会使寄存器处于不完整状态，因此在使用无线电之前必须发出另一个 CMD_RADIO_SETUP 命令。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.3.3.1.3 CMD_FS_POWERUP：上电频率合成器

命令 ID 编号：0x080C

CMD_FS_POWERUP 是无线电操作命令。除了表 25-9 中列出的参数外，命令结构还包含表 25-24 中列出的字段。

启动时，无线电 CPU 为频率合成器上电并应用 pRegOverride 中给出的寄存器修改。如果 pRegOverride 为 NULL，则不会覆盖任何寄存器。覆盖结构的格式与 CMD_RADIO_SETUP 的格式相同（请参阅节 25.3.3.1.2）。仅应用适用于合成器硬件的覆盖。

在使用任何使用频率合成器（以及发送器或接收器）的命令之前，除非合成器作为无线电设置的一部分上电，否则必须运行 CMD_FS_POWERUP。必须在 CMD_FS_POWERUP 之前使用 CMD_RADIO_SETUP 或另一个设置命令来设置无线电。

如果 pRegOverride 中的指针无效，地址或索引无效，数组的长度为零或过大。如果条目中的另一个参数不被允许，操作将以状态 ERROR_PAR 结束。如果合成器上电失败，命令以 ERROR_SYNTH_PROG 作为状态结束。否则完成后，命令以 DONE_OK 作为状态结束。

表 25-24. CMD_FS_POWERUP 命令格式

字节索引	字段名称	位索引	位字段名称	类型	说明
14-15					保留
16-19	pRegOverride			W	指向要覆盖的硬件和配置寄存器列表的指针。如果为 NULL，不使用覆盖。

25.3.3.1.4 CMD_FS_POWERDOWN：关闭频率合成器电源

命令 ID 编号：0x080D

CMD_FS_POWERDOWN 是一条无线电操作命令，仅接受表 25-9 中列出的强制参数。该命令等待启动触发条件，然后关闭合成器电源。此断电操作不仅如 CMD_FS_OFF（请参阅节 25.3.3.1.6）或在某些其他无线电操作命令结束时那样停止合成器，还会关闭模拟模块。

运行 CMD_FS_POWERDOWN 后，必须使用 CMD_FS_POWERUP 或另一个能给合成器加电的命令重新加电，才能使用合成器。

在无线电断电前（例如器件进入低功耗模式时），必须始终先运行 CMD_FS_POWERDOWN。

完成后，CMD_FS_POWERDOWN 命令以 DONE_OK 状态结束。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.3.3.1.5 CMD_FS：频率合成器控制命令

命令 ID 编号：0x0803

CMD_FS 是一条无线电操作命令。除了表 25-9 中列出的参数外，命令结构还包含表 25-25 中列出的字段，并可将合成器编程为特定频率。

要使用的频率由 frequency 和 fractFreq 给出，必须尽可能接近
(frequency + fractFreq / 65536)MHz。

合成器根据 synthConf.bTxMode 设置为 RX 模式或 TX 模式。此模式可由无线电操作命令在设置 RX 或 TX 时更改。如果 synthConf.refFreq 非零，则使用 24MHz / synthConf.refFreq 的基准频率代替默认值。

如果合成器已编程并在锁定之后报告失锁，无线电 CPU 将引发 Synth_No_Lock 中断。合成器保持运行，但系统 CPU 可以使用此信息来停止并重启无线电。每次对合成器编程时，Synth_No_Lock 中断最多引发一次。该中断也可能因具有隐式频率编程的命令而发生。

如果使用非法的频率或分频器设置调用命令，命令将以 ERROR_PAR 状态结束。如果在无线电未配置时调用命令，则以 ERROR_NO_SETUP 状态结束。如果在合成器未加电时调用命令，则以 ERROR_NO_FS 状态结束。

表 25-25. CMD_FS 命令格式

字节索引	字段名称	位索引	位字段名称	类型	说明
14-15	frequency			W	合成器应调谐到的频率（以 MHz 为单位）
16-17	fractFreq			W	合成器应调谐到的频率的分数部分
18	synthConf	0	bTxMode	W	0：在 RX 模式下启动合成器。 1：在 TX 模式下启动合成器。
		1-6	refFreq	W	0：使用默认基准频率。 其他：使用基准频率 24MHz/refFreq。
		7	保留	W	保留
19-23			保留	W	保留

25.3.3.1.6 CMD_FS_OFF：关闭频率合成器

命令 ID 编号：0x0804

CMD_FS_OFF 是一个无线电操作命令，仅接受
表 25-9 中列出的必需参数，并且如果频率合成器由 CMD_FS 启动或由不关闭合成器的无线电操作命令保持打开状态，则关闭该合成器。

该命令启动时，合成器输出被禁用，状态机被复位。模拟器件仍处于通电状态；CMD_FS_POWERDOWN（请参阅节 25.3.3.1.4）可以关闭合成器以进一步降低电流消耗。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.3.3.1.7 CMD_RX_TEST：接收器测试命令

命令 ID 编号：0x0807

CMD_RX_TEST 是一个无线电操作命令，用于将接收器设置为无限 RX 模式以进行测试。

syncWord 的 LSB 中提供了接收器中编程的同步字。如果 config.bNoSync 为 1，则同步搜索的相关阈值将设置为最大值以避免同步。命令结束后将恢复阈值。

如果 pktConfig.bFsOff 为 1，则在操作完成后合成器将关闭（对应于 CMD_FS_OFF；请参阅节 25.3.3.1.6）；否则合成器将保持打开状态。

结束操作的触发器由 endTrigger 和 endTime 设置（请参阅节 25.3.2.5.1）。如果发生此参数定义的触发，无线电操作将结束。

操作以表 25-14 中列出的原因之一结束。

CMD_RX_TEST 的命令结构包含表 25-26 中列出的字段。

表 25-26. CMD_RX_TEST 命令格式

字节索引	字节字段名称	位	位字段名称	类型	说明
14	config	0	保留	W	设置为 0
		1	bFsOff	W	0：在执行命令后使频率合成器保持打开状态。 1：发出命令后关闭频率合成器。
		2	bNoSync	W	0：对配置的模式如常运行同步搜索。 1：将相关阈值写入最大值以避免同步。
15	endTrigger			W	用于结束操作的触发器分类器
16-19	syncWord			W	要用于接收器的同步字
20-23	endTime			W	结束操作的时间

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.3.3.1.8 CMD_TX_TEST：发送器测试命令

命令 ID 编号：0x0808

CMD_TX_TEST 是一个无线电操作命令，用于将接收器设置为无限 TX 模式，并发送 CW 或调制数据，以进行测试。

启动时，对讲机 CPU 会启动发送器。在发出 CMD_TX_TEST 之前，必须使用 CMD_RADIO_SETUP 配置无线电，并且必须通过 CMD_FS 无线电命令将无线电合成器启动至 TX 模式。无线电发送前导码和同步字，其大小由当前无线电配置（使用 CMD_RADIO_SETUP 指定）给出。同步字在 syncWord 的 LSB 中给出。同步字之后的有效载荷由 16 位字 txWord 组成，并无限重复。此字可以根据 config.whitenMode 中的选项进行白化处理，该字段可以取以下值：

- 0：不使用白化。此值对于使用重复模式进行测试很有用，但如果用于频谱测量则会产生杂散。
- 1：默认白化。此值表示使用为当前模式配置的白化（可能带有覆盖值）。如果该模式不使用白化，则不应用白化。
- 2：PRBS-15：使用多项式 $x^{15} + x^{14} + 1$。此值产生长度为 32767 的伪噪声序列。
- 3：PRBS-31：使用多项式 $x^{32} + x^{22} + x^2 + x + 1$。此值产生长度为 4294967295 的伪噪声序列。

对于白化模式 2 和 3，初始化通过无线电 CPU 在传输开始前将 0xAAAA 0000 写入 PRBS 值寄存器来完成。对于白化模式 1，使用默认初始化。

发送器持续运行，直到出现由 endTrigger 和 endTime 设置的触发条件（请参阅节 25.3.2.5.1），或者直到发出中止命令为止。

如果 pktConfig.bFsOn 为 1，则在操作完成后关闭合成器（对应于 CMD_FS_OFF；请参阅节 25.3.3.1.6）；否则保持开启状态。

操作以表 25-14 中列出的原因之一结束。

CMD_TX_TEST 的命令结构包含表 25-27 中列出的字段。

表 25-27. CMD_TX_TEST 命令格式

字节索引	字段名称	位	位字段名称	类型	说明
14	config	0	bUseCw	W	0：发送调制信号。 1：发送连续波。
		1	bFsOff	W	0：命令完成后保持频率合成器开启。 1：命令完成后关闭频率合成器。
		2-3	whitenMode	W	0：无白化 1：默认白化 2：PRBS-15 3：PRBS-32
15					保留
16-17	txWord			W	在白化之前发送到调制解调器的值
18					保留
19	endTrigger			W	用于结束操作的触发分类器
20-23	syncWord			W	发送器使用的同步字
24-27	endTime			W	结束操作的时间

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.3.3.1.9 CMD_SYNC_STOP_RAT：同步和停止无线电计时器命令

命令 ID 编号：0x0809

CMD_SYNC_STOP_RAT 是一条无线电操作命令。除了表 25-9 中列出的参数外，命令结构还包含表 25-28 中列出的字段。

如需了解更多详情，请参阅章节 16。

AON_RTC:CTL.RTC_UPD_EN 必须为 1。

表 25-28. CMD_SYNC_STOP_RAT 命令格式

字节索引	字段名称	位	位字段名称	类型	说明
14-15					未使用
16-19	rat0			R	返回的 RAT 值，对应于 RTC 为零时 RAT 将具有的值。

启动时，无线电 CPU 会设置捕获一个 RTC 节拍并等待该节拍到来，然后停止 RAT 并计算 rat0 值。该值必须存储起来，以便在 RAT 重新启动时使用。

25.3.3.1.10 CMD_SYNC_START_RAT：同步启动无线电计时器命令

命令 ID 编号：0x080A

CMD_SYNC_START_RAT 是无线电操作命令。除了表 25-9 中列出的参数外，命令结构还包含表 25-29 中列出的字段。

如需了解更多详情，请参阅章节 16。

TOP:AON_RTC:CTL.RTC_UPD_EN 必须为 1。

表 25-29. CMD_SYNC_START_RAT 命令格式

字节索引	字段名称	位	位字段名称	类型	说明
14-15					未使用
16-19	rat0			W	RTC 为零时，与 RAT 应具有的值相对应的所需 RAT 值。该参数由 CMD_SYNC_STOP_RAT 返回。

启动时，无线电 CPU 启动 RAT 并设置 RTC 节拍的捕获并等待此节拍，然后根据输入参数 rat0 计算必要的计时器调整并执行此调整。输入参数 rat0 是 CMD_SYNC_STOP_RAT 之前返回的值。

由于发出此命令时无线电计时器通常不运行，因此启动触发必须为 TRIG_NOW（请参阅节 25.3.2.5.1）。

系统引导后首次启动 RAT 时，必须使用命令 CMD_START_RAT（请参阅节 25.3.3.2.7）。或者，可以使用 0 等固定参数发出 CMD_SYNC_START_RAT；然而，这为 RAT 提供了任意的起始值。在关闭无线电电源之前，系统 CPU 必须运行 CMD_SYNC_STOP_RAT 命令。再次为无线电通电后，系统 CPU 必须使用与发出 CMD_SYNC_STOP_RAT 时接收到的参数相同的参数运行 CMD_SYNC_START_RAT 命令。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.3.3.1.11 CMD_COUNT：计数器命令

命令 ID 编号：0x080B

CMD_COUNT 是一条无线电操作命令。除了表 25-9 中列出的参数外，命令结构还包含表 25-30 中列出的字段。

表 25-30. CMD_COUNT 命令格式

字节索引	字段名称	位	位字段名称	类型	说明
14-15	计数器			R/W	计数器。启动时，无线电 CPU 会将该值递减，若结果为零，操作的结束状态会有所不同。

启动时，无线电 CPU 会将计数器字段递减 1，并将结果写回该字段。如果递减结果为零，操作以状态 DONE_COUNTDOWN 和结果 FALSE 结束。否则，操作以状态 DONE_OK 和结果 TRUE 结束，该结果可在条件执行中用于创建循环。

如果启动操作时计数器等于零，则此参数非法，因此操作以状态 ERROR_PAR 结束。

操作会因表 25-26 或表 25-31 中列出的某一原因而结束。

表 25-31. CMD_COUNT 的其他结束原因

条件	状态代码	结果
完成操作且计数器 > 0	DONE_OK	真
完成操作且计数器 = 0	DONE_COUNTDOWN	否

25.3.3.1.12 CMD_SCH_IMM：以无线电操作形式运行即时命令

命令 ID 编号：0x0810

CMD_SCH_IMM 是一条无线电操作命令。除了表 25-9 中列出的参数外，命令结构还包含表 25-32 中列出的字段。

表 25-32. CMD_SCH_IMM 命令格式

字节索引	字段名称	位	位字段名称	类型	说明
14-15					保留
16-19	cmdrVal			W	将写入 CMDR 的值
20-23	cmdstaVal			R	将返回到 CMDSTA 中的值

启动时，无线电 CPU 获取 cmdrVal 中的值，并如同该值已写入 CMDR 寄存器一样对其进行处理。该命令可以是指向即时命令的指针，也可以构成如图 25-4 所示的直接命令。指向无线电操作命令的指针会导致调度错误。通常返回到 CMDSTA 的值会被写入 cmdstaVal，这意味着不会引发命令 RF_CMD_ACK 中断。相反，会像任何其他无线电操作命令一样引发 COMMAND_DONE 中断。

根据即时命令或直接命令的结果，无线电操作命令的状态和结果如表 25-33 所示。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

CMD_SCH_IMM 可以在无线电操作命令链中运行即时命令，或者在未来调度这些命令。如果 CMDR 寄存器中收到的即时命令或直接命令正在处理，而同时已调度的 CMD_SCH_IMM 命令启动，则已调度的命令会等到另一条命令完成之后才开始处理。

表 25-33. CMD_SCH_IMM 的结束状态

条件	状态代码	结果
即时命令以结果 DONE 结束	DONE_OK	真
执行命令时出错，导致 CMDSTA 结果与下表行中列出的结果不符。	DONE_FAILED	否
命令中存在错误，导致以下结果之一： • SchedulingError • UnknownCommand • UnknownDirCommand • IllegalPointer • ParError.	ERROR_PAR	ABORT

25.3.3.1.13 CMD_COUNT_BRANCH：带有命令链分支的计数器命令

命令 ID 编号：0x0812

CMD_COUNT_BRANCH 是无线电操作命令。除了表 25-9 中列出的参数外，命令结构还包含表 25-34 中列出的字段。

表 25-34. CMD_COUNT_BRANCH 命令格式

字节索引	字段名称	位	位字段名称	类型	说明
14-15	计数器			R/W	计数器 启动时，无线电 CPU 将递减该值，如果结果为零，则操作的结束状态将有所不同。
16-19	pNextOpIfOk			W	如果计数器未到期，则指向下一个操作的指针。

开始时，无线电 CPU 将计数器字段递减 1（除非已为 0），然后将结果写回该字段。如果递减的结果为零，则操作以状态 DONE_COUNTDOWN 和结果 FALSE 结束。否则，操作以状态 DONE_OK 和结果 TRUE 结束。在这种情况下，要运行的下一个无线电操作命令由 pNextOpIfOk 提供，而不是 pNextOp（请参阅表 25-9），后者可在条件执行中用于创建循环。

如果操作在计数器等于零的情况下开始，则操作以状态 DONE_OK 结束，下一个操作从 pNextOpIfOk 进行。如果上一条命令设置为可选择跳过，则可以使用此操作，因为从链中的上一条命令跳过遵循 pNextOp。

操作以表 25-14 或表 25-35 中列出的原因之一结束。

表 25-35. CMD_COUNT_BRANCH 的其他结束原因

条件	状态代码	结果
启动时，以计数器 = 0 完成操作	DONE_OK	TRUE
递减后，计数器 > 0 时完成操作	DONE_OK	TRUE
递减后，计数器 = 0 时完成操作	DONE_COUNTDOWN	FALSE

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.3.3.1.14 CMD_PATTERN_CHECK：将存储器中的值与模式进行比较

命令 ID 编号：0x0813

CMD_PATTERN_CHECK 是一条无线电操作命令。除了表 25-9 中列出的参数外，命令结构还包含表 25-36 中列出的字段。

表 25-36. CMD_PATTERN_CHECK 命令格式

字节索引	字段名称	位	位字段名称	类型	说明
14-15	patternOpt	0-1	操作	W	要执行的操作： 0：如果 value == compareVal，则为 TRUE 1：如果 value < compareVal，则为 TRUE 2：如果 value > compareVal，则为 TRUE 3：保留
		2	bByteRev	W	如果为 1，交换值的 4 个字节，以便首先读取 MSB。
		3	bBitRev	W	如果为 1，则对值执行位反转。
		4-8	signExtend	W	0：将 value 和 compareVal 视为无符号数。 1 – 31：将 value 和 compareVal 视为有符号数，其中 value 给出有符号数中 MSB 的位号。
		9	bRxVal	W	0：使用 pValue 作为指针。 1：使用 pValue 作为距离最后提交的 RX 条目元素起始地址的有符号偏移量。
16-19	pNextOpIfOk			W	如果比较结果为 TRUE，则为下一个操作的指针
20-23	pValue			W	要读取值的指针，或者如果 patternOpt.bRxVal == 1，则为相对于最后的 RX 条目的偏移量
24-27	mask			W	比较前要应用的位掩码
28-31	compareVal			W	要与之比较的值

启动时，如果 patternOpt.bRxVal == 0，则无线电 CPU 从 pValue 指向的位置读取一个 4 字节值。
如果 patternOpt.bRxVal == 1，则通过获取最后提交的 RX 条目元素的起始地址指针，并加上在 pValue 中找到的有符号数作为字节偏移量，从而确定要读取的位置。无论属于哪种情况，此指针都不需要 4 字节对齐。如果指针未按字节对齐，则逐字节读取值。

然后，按以下顺序对值执行以下操作：

1. 如果 patternOpt.bByteRev == 1，交换字节 3 与字节 0，以及字节 1 与字节 2，如同首先读取字节的 MSB 一样。
2. 如果 patternOpt.bBitRev == 1，则反转整个 32 位字的位序。
3. 在值与掩码之间执行按位“与”操作。
4. 如果 patternOpt.signExtend > 0，将位号 patternOpt.signExtend（其中位 0 为 LSB）的值复制到所有更高有效位中。
5. 根据 patternOpt.operation，在结果值与 compareVal 之间执行比较操作（请参阅表 25-36）。如果 patternOpt.signExtend == 0，
   则比较为无符号形式；否则为有符号形式。

如果 patternOpt.operation 或 pValue 具有非法值，则操作以状态 ERROR_PAR 结束。否则，根据前面列表中第 5 步的比较结果，操作以表 25-14 或表 25-37 中列出的原因之一结束。如果比较结果为 TRUE，则要运行的下一个无线电操作命令由 pNextOpIfOk 给出，而非 pNextOp。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-37. CMD_PATTERN_CHECK 的其他结束原因

条件	状态代码	结果
比较结果为 TRUE。	DONE_OK	真
比较结果为 FALSE。	DONE_FAILED	否
在没有完整接收任何 RX 数据的情况下，使用 patternOpt.bRxVal 命令	ERROR_NO_RX	ABORT

25.3.3.2 与协议无关的直接和即时命令

本节介绍可跨协议使用的即时命令。节 25.3.4 介绍了用于操作数据队列的命令。

25.3.3.2.1 CMD_ABORT：ABORT 命令

命令 ID 编号：0x0401

CMD_ABORT 是一条直接命令。

收到该命令后，无线电 CPU 将尽快结束正在运行的无线电操作命令。用于 RX 和 TX 的模拟电路会被安全关闭，并且数据结构将会更新，因此它们不会停留在未完成状态。

如果发出 CMD_ABORT 时有无线电操作命令正在运行，当该无线电操作命令结束时，无线电 CPU 会产生 COMMAND_DONE 和 LAST_COMMAND_DONE 中断。该无线电操作命令的命令结构状态会反映命令已被中止。

如果没有运行无线电操作命令，则不执行任何操作。在所有情况下，CMDSTA 寄存器中报告的结果均为 DONE。如果正在运行无线电操作命令，则 CMDSTA 可能会在无线电操作结束之前更新。

25.3.3.2.2 CMD_STOP：停止命令

命令 ID 编号：0x0402

CMD_STOP 是一条直接命令。

接收到该命令后，无线电 CPU 会通知当前正在运行的无线电操作命令，请求其停止。CMD_STOP 命令比 CMD_ABORT 命令更为*平稳*，但可能需要更多时间才能完成。通常，正在接收或发送的数据包会被处理完成。每个无线电操作命令在接收到 CMD_STOP 时的具体行为会相应说明。有些命令总是在已知时间内结束，并且不会响应 CMD_STOP。

如果没有运行无线电操作命令，则不执行任何操作。在所有情况下，CMDSTA 中指示的结果均为 DONE。如果正在运行无线电操作命令，则 CMDSTA 可能会在无线电操作结束之前更新。

25.3.3.2.3 CMD_GET_RSSI：读取 RSSI 命令

命令 ID 编号：0x0403

CMD_GET_RSSI 是一个不接受参数的立即命令，因此可用作直接命令。

接收时，无线电 CPU 从底层接收器读取 RSSI。RSSI 在 CMDSTA 的结果字节 2（位 23 – 16）中返回（请参阅图 25-5）。RSSI 在有符号表单上以 dBm 为单位给出。如果没有 RSSI 可用，则会使用 RSSI 的特殊值（−128 或 0x80）来指示这点。

如果未运行无线电操作命令，则无线电 CPU 在 CMDSTA 中返回结果 ContextError。否则，无线电 CPU 将返回 DONE 及 RSSI 值。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.3.3.2.4 CMD_UPDATE_RADIO_SETUP：更新无线电设置命令

命令 ID 编号：0x0001

CMD_UPDATE_RADIO_SETUP 是一个直接的命令，用于获取表 25-38 中列出的参数。

表 25-38. CMD_UPDATE_RADIO_SETUP 命令格式

字节索引	字段名称	位	位字段名称	类型	说明
0-1	commandNo			W	命令 ID 编号
2-3					保留
4-7	pRegOverride			W	指向要覆盖的硬件及配置寄存器列表的指针

接收到这些寄存器后，无线电 CPU 会更新 pRegOverride 中给出的寄存器。这是一个指向包含某些硬件寄存器的覆盖值、由无线电 CPU 控制的无线电配置和协议相关变量的结构的指针。格式与 CMD_RADIO_SETUP 相同（请参阅节 25.3.3.1）。如果在无线电运行时完成更新，则更新必须主要在无线电和协议配置上完成，因为对硬件寄存器的修改可能会导致不良行为。

25.3.3.2.5 CMD_TRIGGER：生成命令触发器

命令 ID 编号：0x0404

CMD_TRIGGER 是一个立即执行的命令，用于获取表 25-39 中列出的参数。

表 25-39. CMD_TRIGGER 命令格式

字节索引	字段名称	位	位字段名称	类型	说明
0-1	commandNo			W	命令 ID 编号
2	triggerNo			W	命令触发器编号

接收时，无线电 CPU 生成由 triggerNo 指定的命令触发器，因此运行无线电操作命令会相应地做出响应（请参阅节 25.3.2.5.1）。

如果触发器编号超出有效范围 0 – 3，无线电 CPU 将在 CMDSTA 中返回结果 ParError。如果在发送的触发号上没有运行的无线电操作命令挂起，则无线电 CPU 在 CMDSTA 中返回结果 ContextError。否则，无线电 CPU 返回 DONE，这可能会在运行无线电操作命令响应触发器之前返回。

CMD_TRIGGER 可以作为直接命令发送。如果是，触发编号由 CMDR 第 8 – 15 位中的参数给出。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.3.3.2.6 CMD_GET_FW_INFO：请求有关正在运行的固件的信息

命令 ID 编号：0x0002

CMD_GET_FW_INFO 是一个立即命令，它采用表 25-40 中列出的参数。

表 25-40. CMD_GET_FW_INFO 命令格式

字节索引	字段名称	位	位字段名称	类型	说明
0-1	commandNo			W	命令 ID 编号
2-3	versionNo			R	固件版本号
4-5	startOffset			R	空闲 RAM 的开始
6-7	freeRamSz			R	可用 RAM 的大小
8-9	availRatCh			R	可用 RAT 通道的位图

接收时，无线电 CPU 报告有关正在运行的无线电固件的信息。版本号在 versionNo 中返回。startOffset 和 freeRamSz 字段包含有关无线电 RAM 非保留部分中的区域的信息，无线电 CPU 不用于数据（包括栈和堆）。此区域可供系统 CPU 自由使用，用于数据交换、无线电 CPU 补丁或其他用途。空闲 RAM 的起始地址及结束地址以相对于无线电 RAM 起始的偏移量的形式给出。除前 36 个字节外，RF 内核保留 RAM 可供自由使用。

备注

其中一些可用的 RAM 用于 TI 提供的补丁。

availRatCh 字段是位图，其中位的位置 n 指示系统 CPU 是否可以使用 RAT 通道 n。位值为 1 表示相应的通道可由系统 CPU 使用，而位值为 0 表示该通道已为无线电 CPU 保留或不存在。

25.3.3.2.7 CMD_START_RAT：异步启动无线电计时器命令

命令 ID 编号：0x0405

CMD_ START_RAT 是直接命令。

接收时，如果尚未启动，无线电 CPU 将启动 RAT。

如果 RAT 已在运行、则无线电 CPU 在 CMDSTA 中返回结果 ContextError。否则，无线电 CPU 返回 DONE。

25.3.3.2.8 CMD_PING：以中断进行响应

命令 ID 编号：0x0406

CMD_PING 是直接命令。

接收时，无线电 CPU 在 CMDSTA 中返回 DONE。此命令可以测试两个 CPU 之间的通信，或检查无线电 CPU 在引导后何时准备就绪。

Copyright © 2026 Texas Instruments Incorporated

25.3.3.2.9 CMD_READ_RFREG：读取射频内核寄存器

命令 ID 编号：0x0601

CMD_READ_RFREG 是一条即时命令，采用表 25-41 中列出的参数。

表 25-41. CMD_READ_RFREG 命令格式

字节索引	字段名称	位	位字段名称	类型	说明
0-1	commandNo			W	命令 ID 编号
2-3	address			W	相对于射频内核硬件寄存器组 (0x4004 0000) 开头的偏移量
4-7	value			R	寄存器的返回值

收到该命令后，无线电 CPU 读取地址为 0x4004 0000 + address 的射频内核寄存器。结果写入 value。如果该地址不能被 4 整除，无线电 CPU 在 CMDSTA 中返回 ParError。

CMD_READ_RFREG 可作为直接命令发送。如果这么做，地址由 CMDR 的位 2 – 15 给出，地址的 2 个 LSB 设置为 00。

执行读取时，结果通过 value 返回。结果的 24 个 LSB 通过 CMDSTA 的位 8 – 31 返回。CMDSTA 中返回的结果为 DONE。

25.3.3.2.10 CMD_SET_RAT_CMP：将 RAT 通道设置为比较模式

命令 ID 编号：0x000A

CMD_SET_RAT_CMP 是一个直接命令，用于获取表 25-42 中列出的参数。

表 25-42. CMD_SET_RAT_CMP 命令格式

字节索引	字段名称	位	位字段名称	类型	说明
0-1	commandNo			W	命令 ID 编号
2	ratCh			W	无线电计时器通道编号
3					保留
4-7	compareTime			W	比较发生的时间

接收时，无线电 CPU 将 RAT 通道设置为比较模式下 ratCh 提供的 RAT 通道、并将通道比较时间设置为 compareTime，这也会激活通道。在给定的时间发生一个通道事件，可以通过 RFC_DBELL 模块将其作为系统 CPU 的 RF 硬件中断启用。

通道编号必须指示无线电 CPU 未保留以供使用的通道。否则，无线电 CPU 在 CMDSTA 中返回 ParError。如果比较时间是过去评估命令的时间、则无线电 CPU 在 CMDSTA 中返回 ContextError 并禁用 RAT 通道。如果比较事件已成功设置，则无线电 CPU 在 CMDSTA 中返回 DONE。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.3.3.2.11 CMD_SET_RAT_CPT：将 RAT 通道设置为捕获模式

命令 ID 编号：0x0603

CMD_SET_RAT_CPT 是一个直接命令，用于获取表 25-43 中列出的参数。

表 25-43. CMD_SET_RAT_CPT 命令格式

字节索引	字段名称	位	位字段名称	类型	说明
0-1	commandNo			W	命令 ID 编号
2-3	config	0-2			保留
		3-7	inputSrc	W	输入源指示器： 22：RFC_GPI0 23：RFC_GPI1 其他值：保留
		8-11	ratCh	W	无线电计时器通道编号
		12	bRepeated	W	0：单次捕获模式 1：重复捕获模式
		13-14	inputMode	W	输入模式： 00：上升沿 01：下降沿 10：两个边沿 11：保留

接收时，无线电 CPU 会将 config.ratCh 提供的 RAT 通道设置为捕获模式。如果 config.bRepeated 为 0，则通道设置为单次捕获模式；否则，通道设置为重复捕获模式。无线电 CPU 将输入源设置为 config.inputSrc，将输入模式设置为 config.inputMode。如果通道设置为单次捕获模式，则该通道也会被激活。在发生捕获时，这会导致通道事件发生，并可通过 RFC_DBELL 模块作为系统 CPU 的 RF 硬件中断启用。

CMD_SET_RAT_CMP 可作为直接命令发送。如果是这样，配置字的位 2 – 15 由 CMDR 的位 2 – 15 给出（不使用配置的位 0 – 1）。

通道编号必须指示无线电 CPU 未保留以供使用的通道。否则，无线电 CPU 在 CMDSTA 中返回 ParError。如果通道已成功设置，则无线电 CPU 在 CMDSTA 中返回 DONE。

25.3.3.2.12 CMD_DISABLE_RAT_CH：禁用 RAT 通道

命令 ID 编号：0x0408

CMD_DISABLE_RAT_CH 是一个直接命令，用于获取表 25-44 中列出的参数。

表 25-44. CMD_DISABLE_RAT_CH 命令格式

字节索引	字段名称	位	位字段名称	类型	说明
0-1	commandNo			W	命令 ID 编号
2	ratCh			W	RAT 通道编号

接收时，无线电 CPU 会禁用 ratCh 提供的 RAT 通道。这会禁用该通道先前由 CMD_SET_RAT_CMP 或 CMD_SET_RAT_CPT 命令完成的配置。

CMD_DISABLE_RAT_CH 可作为直接命令发送。如果是这样，ratCh 由 CMDR 第 8 – 15 位中的参数给出。

通道编号必须指示无线电 CPU 未保留以供使用的通道。否则，无线电 CPU 在 CMDSTA 中返回 ParError。如果通道编号有效，则 CPU 会在通道被禁用后在 CMDSTA 中返回 DONE。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.3.3.2.13 CMD_SET_RAT_OUTPUT：将 RAT 输出设置为指定模式

命令 ID 编号：0x0604

CMD_SET_RAT_OUTPUT 是一个直接命令，用于获取表 25-45 中列出的参数。

表 25-45. CMD_SET_RAT_OUTPUT 命令格式

字节索引	字段名称	位	位字段名称	类型	说明
0-1	commandNo			W	命令 ID 编号
2-3	config	0-1			保留
		2-4	outputSel	W	输出事件指示器： 1：RAT_GPO1 2：RAT_GPO2 3：RAT_GPO3 其他值：保留
		5-7	outputMode	W	输出模式： 000：脉冲 001：设置 010：清除 011：切换 100：始终为 0 101：始终为 1 其他：保留
		8-11	ratCh	W	RAT 通道编号

接收时，无线电 CPU 在 config.outputMode 给定的模式下设置 config.outputSel 给出的 RAT 输出事件，并由 config.ratCh 给出的 RAT 通道控制。此命令必须与使用 CMD_SET_RAT_CMP 命令将此通道设置为比较模式相结合。

CMD_SET_RAT_OUTPUT 可作为直接命令发送。如果是这样，配置字的位 2 – 15 由 CMDR 的位 2 – 15 给出（不使用配置的位 0 – 1）。

通道编号 config.ratCh 必须指示未保留以供无线电 CPU 使用的通道，并且输出编号 config.outputSel 不能是无线电 CPU 使用的输出。否则，无线电 CPU 在 CMDSTA 中返回 ParError。如果输出事件已成功设置，则无线电 CPU 在 CMDSTA 中返回 DONE。

25.3.3.2.14 CMD_ARM_RAT_CH：配置 RAT 通道

命令 ID 编号：0x0409

CMD_ARM_RAT_CH 是一条即时命令，采用表 25-46 中列出的参数。

表 25-46. CMD_ARM_RAT_CH 命令格式

字节索引	字段名称	位	位字段名称	类型	说明
0-1	commandNo			W	命令 ID 编号
2	ratCh			W	RAT 通道编号

收到该命令后，无线电 CPU 配置 ratCh 所指定的 RAT 通道。

CMD_DISABLE_RAT_CH 命令可作为直接命令发送。如果是这样，ratCh 由 CMDR 位 8 – 15 中的参数给出。

通道编号必须指示未保留给无线电 CPU 使用的通道。否则，无线电 CPU 在 CMDSTA 中返回 ParError。如果通道编号有效，则 CPU 在通道配置完成后在 CMDSTA 中返回 DONE。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.3.3.2.15 CMD_DISARM_RAT_CH：解除 RAT 通道配置

命令 ID 编号：0x040A

CMD_DISARM_RAT_CH 是一条即时命令，采用表 25-47 中列出的参数。

表 25-47. CMD_DISARM_RAT_CH 命令格式

字节索引	字段名称	位	位字段名称	类型	说明
0-1	commandNo			W	命令 ID 编号
2	ratCh			W	RAT 通道编号

收到该命令后，无线电 CPU 解除 ratCh 所指定 RAT 通道的配置。

CMD_DISABLE_RAT_CH 可作为直接命令发送。如果是这样，ratCh 由 CMDR 位 8 – 15 中的参数给出。

通道编号必须指示未保留给无线电 CPU 使用的通道。否则，无线电 CPU 在 CMDSTA 中返回 ParError。如果通道编号有效，则 CPU 在通道配置完成后在 CMDSTA 中返回 DONE。

25.3.3.2.16 CMD_SET_TX_POWER：设置发射功率

命令 ID 编号：0x0010

CMD_SET_TX_POWER 是一个直接命令，用于获取表 25-48 中列出的参数。

表 25-48. CMD_SET_TX_POWER 命令格式

字节索引	字段名称	位	位字段名称	类型	说明
0-1	commandNo			W	命令 ID 编号
2-3	txPower			W	新的 TX 功率设置。TI 建议使用 SmartRF Studio 中的值。 位 0-6：IB 25°C 下要写入 PA 功率控制字段的值。 有关详细信息，请参阅方程式 14。 位 6-7：GC 要写入 PA 第一级的增益控制的值。 位 8：升压 PA 中的驱动器强度。 0：低驱动器强度 1：高驱动器强度 位 9-15：tempCoeff IB 的温度系数。 0：无温度补偿。

$$IB = IB_{25°C} + \frac{(Temperature[°C] - 25) \times tempCoeff}{256} \quad (14)$$

接收时，无线电 CPU 设置发射功率，以便在下次开始传输时使用。如果正在发送一个数据包，则在下一个数据包开始发送之前，发送功率不会更新。

每次开始数据包发送时，就会完成发射功率的温度补偿。

如果在无线电设置命令中配置了 20 dBm PA，则不允许使用 CMD_SET_TX_POWER；在这种情况下，应使用 CMD_SET_TX20_POWER（请参阅节 25.3.3.2.17）。

完成后，无线电 CPU 在 CMDSTA 中返回 DONE。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.3.3.2.17 CMD_SET_TX20_POWER：设置 20dBm PA 的传输功率

命令 ID 编号：0x0014

CMD_SET_TX20_POWER 是一条即时命令，采用表 25-49 中列出的参数。

表 25-49. CMD_SET_TX20_POWER 命令格式

字节索引	字段名称	位	位字段名称	类型	说明
0-1	commandNo			W	命令 ID 编号
2-3					保留
4-7	txPower	0-6	IB	W	新 TX 功率设置。TI 建议使用来自 SmartRF Studio 的值。 在 25°C 时写入 PA 功率控制字段的值。 有关详细信息，请参阅方程式 14。
		6-7	ibBoost	W	写入 PA 的偏置控制字段的值
		8	boost	W	PA 的驱动器强度 0：低驱动器强度 1：高驱动器强度
		9-15	tempCoeff	W	IB 的温度系数 0：无温度补偿
		16-21	paLdoTrim	W	写入 20dBm PA LDO 输出电压控制的值
		22-31			保留

收到该命令后，无线电 CPU 将设置 20dBm PA 的传输功率和温度系数，以供下次启动发送时使用。如果正在传输数据包，则在下一个数据包开始发送之前，不会更新传输功率。

每次开始发送数据包时，都会对传输功率进行温度补偿。

如果在无线电设置命令中未配置 20dBm PA，则不允许使用 CMD_SET_TX20_POWER；此时应使用 CMD_SET_TX_POWER（请参阅节 25.3.3.2.16）。

完成后，无线电 CPU 会在 CMDSTA 中返回 DONE。

25.3.3.2.18 CMD_UPDATE_FS：设置新的合成器频率而不进行重新校准（已弃用）

命令 ID 编号：0x0011

CMD_UPDATE_FS 是一个直接命令，用于获取表 25-50 中列出的参数。提供该命令是为了向后兼容其他设备，但不建议使用。而应使用 CMD_MODIFY_FS（请参阅节 25.3.3.2.19）。

表 25-50. CMD_UPDATE_FS 命令格式

字节索引	字段名称	位	位字段名称	类型	说明
0-1	commandNo			W	命令 ID 编号
2-13					保留
14-15	frequency			W	要调谐到的频率（以 MHz 为单位）
16-17	fractFreq			W	要调谐到的频率小数部分

该命令的作用与 CMD_MODIFY_FS 相同，但命令格式有很多未使用的空间，如果 LO 分频器不是 2，则无法正确解释频率。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.3.3.2.19 CMD_MODIFY_FS：设置新的合成器频率而不进行重新校准

命令 ID 编号：0x0013

CMD_MODIFY_FS 是一个直接命令，用于获取表 25-51 中列出的参数。

表 25-51. CMD_MODIFY_FS 命令格式

字节索引	字段名称	位	位字段名称	类型	说明
0-1	commandNo			W	命令 ID 编号
2-3	frequency			W	要调谐到的频率（以 MHz 为单位）
4-5	fractFreq			W	要调谐到的频率小数部分

接收时，无线电 CPU 将在合成器中编程一个新频率，而无需重新启动校准。与校准下使用的频率相比，这一变化很小，否则合成器很可能无法重新锁定。如果命令在 RX 或 TX 期间完成，则可能会发生额外的失真。

使用的频率由 frequency 和 fractFreq 给出，频率将尽可能接近 (frequency + fractFreq / 65536) MHz。

如果合成器未运行且校准已完成，则无线电 CPU 将在 CMDSTA 中返回 ContextError。如果频率无效，无线电 CPU 将在 CMDSTA 中返回 ParError。否则，更新完成后，无线电 CPU 将在 CMDSTA 中返回 DONE。

25.3.3.2.20 CMD_BUS_REQUEST：请求可用于 RF 内核的系统总线

命令 ID 编号：0x040E

CMD_BUS_REQUEST 是一个直接命令，用于获取表 25-52 中列出的参数。

表 25-52. CMD_BUS_REQUEST 命令格式

字节索引	字段名称	位	位字段名称	类型	说明
0-1	commandNo			W	命令 ID 编号
2	bSysBusNeeded			W	0：系统总线可能睡眠 1：需要系统总线访问权限

接收时，如果 bSysBusNeeded 为非零，则无线电 CPU 将朝向 PRCM 的总线请求位设置为 1，如果 bSysBusNeeded 为零，则设置为 0。如果 bSysBusNeeded 为非零值，这表示即使系统进入深度睡眠状态，系统总线也保持唤醒状态，由于以下原因之一，RF 内核必须处于该条件下才能运行并访问系统端：

- 由发送到 RF 内核的指针指向的任何命令结构、数据结构等都将置于系统 RAM 或闪存中。
- 因为 TX 功率具有非零温度系数，所以 RF 内核必须读取温度。
- RF 内核必须在 CMD_SYNC_STOP_RAT 或 CMD_SYNC_START_RAT 期间读取 RTC 以与 RAT 同步。

CMD_BUS_REQUEST 可作为直接命令发送。如果是这样，bSysBusNeeded 由 CMDR 第 8 – 15 位中的参数给出。

当更新完成时，无线电 CPU 会在 CMDSTA 中返回 DONE。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.3.4 数据队列操作的立即命令

以下命令是用于所有使用数据队列的无线电操作的数据队列操作的立即命令。

25.3.4.1 CMD_ADD_DATA_ENTRY：将数据条目添加到队列

命令 ID 编号：0x0005

CMD_ADD_DATA_ENTRY 是一条即时命令，采用表 25-53 中列出的参数。

表 25-53. CMD_ADD_DATA_ENTRY 命令格式

字节索引	字段名称	位	位字段名称	类型	说明
0-1	commandNo			W	命令 ID 编号
2-3					保留
4-7	pQueue			W	指向将条目添加到的队列结构的指针
8-11	pEntry			W	指向条目的指针

收到该命令后，无线电 CPU 将提供的数据条目追加到所指示的队列中。无线电 CPU 执行以下操作：

```
Set pQueue-> pLastEntry-> pNextEntry = pEntry
Set pQueue-> pLastEntry = pEntry
```

如果指针 pQueue 或 pEntry 中任何一个无效（即位于非存储器地址范围或未按 32 位字对齐），则命令失败，无线电 CPU 将 CMDSTA 的结果字节设置为 ParError。如果 pQueue 中指定的队列设置为不允许追加条目（请参阅节 25.3.2.7.1），则命令失败，无线电 CPU 将 CMDSTA 的结果字节设置为 QueueError。

25.3.4.2 CMD_REMOVE_DATA_ENTRY：从队列中删除第一个数据条目

命令 ID 编号：0x0006

CMD_REMOVE_DATA_ENTRY 是一个直接的命令，用于获取
表 25-54 中列出的参数。

表 25-54. CMD_REMOVE_DATA_ENTRY 命令格式

字节索引	字段名称	位	位字段名称	类型	说明
0-1	commandNo			W	命令 ID 编号
2-3					保留
4-7	pQueue			W	指向从中删除条目的队列结构的指针
8-11	pEntry			R	指向已删除条目的指针

接收时，无线电 CPU 会从指示的队列中删除第一个数据条目。该命令返回指向已删除条目的指针。无线电 CPU 执行以下操作：

```
Set pEntry = pQueue->pCurrEntry
Set pQueue->pCurrEntry = pEntry->pNextEntry
Set pEntry->status = Finished
```

如果指针 pQueue 无效，此命令将失败，并且无线电 CPU 将 CMDSTA 的结果字节设置为 ParError。如果 pQueue 中指定的队列为空，则命令失败，无线电 CPU 将 CMDSTA 的结果字节设置为 QueueError。如果要删除的条目处于 BUSY 状态，则该命令将失败，并且无线电 CPU 将 CMDSTA 的结果字节设置为 QueueBusy。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.3.4.3 CMD_FLUSH_QUEUE：刷新队列

命令 ID 编号：0x0007

CMD_FLUSH_QUEUE 是一条即时命令，采用表 25-55 中列出的参数。

表 25-55. CMD_FLUSH_QUEUE 命令格式

字节索引	字段名称	位	位字段名称	类型	说明
0-1	commandNo			W	命令 ID 编号
2-3					保留
4-7	pQueue			W	指向要刷新的队列结构的指针
8-11	pFirstEntry			R	指向被移除的第一个条目的指针

接收到该命令后，无线电 CPU 刷新所指示的队列，并返回一个指向被移除的第一个条目的指针。无线电 CPU 执行以下操作：

```
Set pFirstEntry = pQueue->pCurrEntry
Set pQueue->pCurrEntry = NULL
Set pQueue->pLastEntry = NULL
```

如果指针 pQueue 无效，该命令将失败，并且无线电 CPU 将 CMDSTA 的结果字节设置为 ParError。如果要移除的第一个条目处于 BUSY 状态，该命令将失败，并且无线电 CPU 将 CMDSTA 的结果字节设置为 QueueBusy。如果 pQueue 中指定的队列为空，则无线电 CPU 无需执行任何操作，但仍视为成功。在这种情况下，返回的 pFirstEntry 为 NULL。

25.3.4.4 CMD_CLEAR_RX：清除所有 RX 队列条目

命令 ID 编号：0x0008

CMD_CLEAR_RX 是一条即时命令，采用表 25-56 中列出的参数。

表 25-56. CMD_CLEAR_RX 命令格式

字节索引	字段名称	位	位字段名称	类型	说明
0-1	commandNo			W	命令 ID 编号
2 – 3					保留
4-7	pQueue			W	指向要清除的队列结构的指针

接收到该命令后，无线电 CPU 会使所有 RX 条目指示为空。无线电 CPU 执行以下操作：

```
Set pTemp = pQueue->pCurrEntry
Loop: Set pTemp->status = Pending
If pTemp->type == 1 then:
Set pTemp->nextIndex = 0
Set pTemp->numElements = 0
Set pTemp = pTemp->nextIndex
If pTemp != NULL and pTemp != pQueue->pCurrEntry, repeat from Loop
```

如果指针 pQueue 无效，该命令将失败，并且无线电 CPU 将 CMDSTA 的结果字节设置为 ParError。如果 pQueue 中指定的队列为空，该命令将失败，并且无线电 CPU 将 CMDSTA 的结果字节设置为 QueueError。如果要移除的第一个条目处于 BUSY 状态，该命令将失败，并且无线电 CPU 将 CMDSTA 的结果字节设置为 QueueBusy。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.3.4.5 CMD_REMOVE_PENDING_ENTRIES：从队列中删除待处理条目

命令 ID 编号：0x0009

CMD_REMOVE_PENDING_ENTRIES 是一个直接命令，用于获取表 25-57 中列出的参数。

表 25-57. CMD_REMOVE_PENDING_ENTRIES 命令格式

字节索引	字段名称	位	位字段名称	类型	说明
0-1	commandNo			W	命令 ID 编号
2-3					保留
4-7	pQueue			W	指向要刷新的队列结构的指针
8-11	pFirstEntry			R	指向已删除的第一个条目的指针

接收时，无线电 CPU 从指示的队列中删除所有处于待处理状态的条目，并返回指向已删除的第一个条目的指针。无线电 CPU 执行以下操作：

```
If pQueue->pCurrEntry->status = Pending, then
Set pFirstEntry = pQueue->pCurrEntry
Set pQueue->pCurrEntry = NULL
Set pQueue->pLastEntry = NULL
else
Set pFirstEntry = pQueue->pCurrEntry->pNextEntry
Set pQueue->pCurrEntry->pNextEntry = NULL
Set pQueue->pLastEntry = pQueue->pCurrEntry
```

如果指针 pQueue 无效，此命令将失败，并且无线电 CPU 将 CMDSTA 的结果字节设置为 ParError。如果 pQueue 中指定的队列为空，则无线电 CPU 不需要执行任何操作，但这仍被视为成功。在这种情况下，返回的 pFirstEntry 为 NULL。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.4 数据队列使用情况

本节介绍无线电 CPU 如何使用数据队列。

25.4.1 数据队列上的操作仅可用于内部无线电 CPU 操作

节 25.3.4 列出用于数据队列操作的命令。对于所述的内部无线电 CPU 操作，可以使用额外的操作。节 25.4.1.1 至节 25.4.1.5 说明了这些操作。

25.4.1.1 PROC_ALLOCATE_TX：分配 TX 条目以进行读取

该过程采用以下输入参数：

- 队列指针 pQueue

该过程返回以下内容：

- 已分配数据条目的指针 pEntry

如果指定的队列为空，或队列的第一个条目已为繁忙状态，则该过程返回错误。否则，执行以下操作：

```
Set pQueue->pCurrEntry->status = Busy
Set pEntry = pQueue->pCurrEntry
```

25.4.1.2 PROC_FREE_DATA_ENTRY：自由分配的数据条目

该过程采用以下输入参数：

- 指向队列的指针，pQueue

该过程返回以下内容：

- 指向已分配数据条目的指针，pEntry

如果指定的队列为空，该过程返回错误。否则，完成以下操作：

```
Set pQueue->pCurrEntry->status = Active
```

25.4.1.3 PROC_FINISH_DATA_ENTRY：完成队列中第一个数据条目的使用

该过程采用以下输入参数：

- 指向队列的指针 pQueue

该过程返回以下内容：

- 指向新条目的指针 pEntry

如果指定的队列为空，该过程将返回错误。否则完成以下操作：

```
Set pTemp = pQueue->pCurrEntry
Set pQueue->pCurrEntry = pTemp->pNextEntry
Set pTemp->status = Finished
Set pEntry = pQueue->pCurrEntry
```


English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.4.1.4 PROC_ALLOCATE_RX：分配 RX 缓冲区来存储数据

该过程采用以下输入参数：

- 指向队列的指针 pQueue
- 要存储的条目元素大小 size

该过程返回以下内容：

- 指向存储数据的数据条目的指针 pEntry
- 指向已完成数据条目的指针 pFinishedEntry，或者如果未完成，则为 NULL

如果队列中的第一个条目已处于忙碌状态，该过程将返回错误。如果没有空间容纳指定大小的条目元素（包括队列为空），则返回*无空间*错误。以下过程描述了该操作：

```
Set pFinishedEntry == NULL
If pQueue->pCurrEntry == NULL then
Return with no space error
end if
If pQueue->pCurrEntry->type != 1 then
if pQueue->pCurrEntry->length < size then
Return with no space error
else
Set pQueue->pCurrEntry->status = Busy
Set pEntry = pQueue->pCurrEntry
end if
else
Set pTemp = pQueue->pCurrEntry
If pTemp->nextIndex + 2 + size > pTemp->length then
Set pQueue->pCurrEntry = pTemp->pNextEntry
Set pTemp->status = Finished
Set pFinishedEntry = pTemp
Set pTemp = pTemp->pNextEntry
If pTemp == NULL or pTemp->length < size + 2 then
Return with no space error
end if
end if
Set pTemp->status = Busy
Set pEntry = pTemp
end if
```


English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.4.1.5 PROC_FINISH_RX：将收到的数据提交到 RX 数据条目

该过程采用以下输入参数：

- 指向队列的指针 pQueue
- 存储的条目元素大小 size

该过程返回以下内容：

- 指向存储数据的数据条目的指针 pEntry
- 指向已完成数据条目的指针 pFinishedEntry，或者如果未完成，则为 NULL

如果队列为空，或者没有足够空间存放指定大小的条目元素，该过程返回错误。否则完成以下操作：

```
If pQueue->pCurrEntry->type != 1 then
Set pTemp = pQueue->pCurrEntry
Set pQueue->pCurrEntry = pTemp->pNextEntry
Set pTemp->status = Finished
else
Increase pQueue->pCurrEntry->nextIndex by size
Increment pQueue->pCurrEntry->numElements by 1
If pQueue->pCurrEntry->nextIndex + 2 == pQueue->pCurrEntry->length then
Set pTemp = pQueue->pCurrEntry
Set pQueue->pCurrEntry = pTemp->pNextEntry
Set pTemp->status = Finished
Set pFinishedEntry == pTemp
else
Set pQueue->pCurrEntry->status = Active
Set pFinishedEntry == NULL
end if
end if
```

此操作在完成 PROC_ALLOCATE_RX 并写入缓冲区中的正确位置后执行；大小必须与 PROC_ALLOCATE_RX 中的值相同。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.4.2 无线电 CPU 使用模型

25.4.2.1 接收队列

当无线电 CPU 接收到数据包时，它通过调用 PROC_ALLOCATE_RX 准备一个缓冲器以供读取。如果成功，则分配的缓冲器将用于存储为每个协议定义的传入数据包。如果未发生空间错误，则无法存储接收到的数据，并为每个协议定义处理方式。

收到数据包后，可能会根据为每个协议定义的规则保留或丢弃该数据包。为了保留数据包，无线电 CPU 会调用 PROC_FINISH_RX。这使得接收到的数据可供系统 CPU 使用。为了丢弃数据包，无线电 CPU 调用 PROC_FREE_DATA_ENTRY，表示下一个数据包可能会覆盖最后一个数据包中接收到的数据。

25.4.2.2 发送队列

当无线电 CPU 即将从 TX 队列发送数据包时，它会调用 PROC_ALLOCATE_TX 以获取指向要发送的数据的指针。数据包传输时，无线电 CPU 调用 PROC_FINISH_DATA_ENTRY 或 PROC_FREE_DATA_ENTRY。如果调用 PROC_FINISH_DATA_ENTRY，则会通知系统 CPU 该条目已完成，并且可以重复使用。如果无法重新传输数据包，则必须使用此调用过程。如果调用了 PROC_FREE_DATA_ENTRY，则发送的条目会保留在队列中的第一个条目，以便可以发送，在需要确认时使用。

如果在发送的数据包上收到确认，然后是无线电 CPU 调用 PROC_FREE_DATA_ENTRY，则无线电 CPU 将调用 PROC_ALLOCATE_TX，后跟 PROC_FINISH_DATA_ENTRY（这相当于 CMD_REMOVE_DATA_ENTRY，请参阅节 25.3.3.2）。此调用过程将发送队列中的下一个条目。如果未收到确认，则重新传输最后发送的数据包。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.5 IEEE 802.15.4

本节介绍了特定于 IEEE 802.15.4 的命令结构、中断、数据处理、无线电操作命令及立即命令。

25.5.1 IEEE 802.15.4 命令

表 25-58 和表 25-59 定义 IEEE 802.15.4 专用的无线电操作命令。

表 25-58. 后台级 IEEE 802.15.4 无线电操作命令

ID	命令名称	说明
0x2801	CMD_IEEE_RX	运行接收器
0x2802	CMD_IEEE_ED_SCAN	运行能量检测扫描

表 25-59. 前台级 IEEE 802.15.4 无线电操作命令

ID	命令名称	说明
0x2C01	CMD_IEEE_TX	传输数据包
0x2C02	CMD_IEEE_CSMA	执行 CSMA-CA
0x2C03	CMD_IEEE_RX_ACK	接收确认
0x2C04	CMD_IEEE_ABORT_BG	中止后台级别操作

表 25-60. 前台级别的常见无线电操作命令

ID	命令名称	说明
0x0C10	CMD_FG_SCH_IMM	行为类似于 CMD_SCH_IMM

表 25-61 定义立即命令。

表 25-61. IEEE 802.15.4 立即命令

ID	命令名称	说明
0x2001	CMD_IEEE_MOD_CCA	修改用于运行接收器的 CCA 参数
0x2002	CMD_IEEE_MOD_FILT	修改运行接收器的帧过滤参数
0x2003	CMD_IEEE_MOD_SRC_MATCH	修改运行接收器的源匹配参数
0x2401	CMD_IEEE_ABORT_FG	中止前台级别操作
0x2402	CMD_IEEE_STOP_FG	停止前台级别操作
0x2403	CMD_IEEE_CCA_REQ	请求 CCA 和 RSSI 信息

Copyright © 2026 Texas Instruments Incorporated

25.5.1.1 IEEE 802.15.4 无线电操作命令结构

表 25-9 定义所有无线电操作命令的前 14 个字节。CMD_IEEE-ABORT_BG 命令对这 14 个字节没有任何附加字段。

表 25-62. IEEE 802.15.4 RX 命令结构

字节索引	字段名称	类型	说明
14	通道	W	运行开始时调优到的通道： 0：使用现有通道 11 – 26：用作 IEEE 802.15.4 通道，即频率为 [2405 + 5 × (通道 − 11)] MHz 60 – 207：频率为（2300 + 通道）MHz 其他：保留
15	rxConfig	W	接收队列条目的配置位（有关详细信息，请参阅表 25-72）
16-19	pRxQ	W	接收队列
20-23	pOutput	W	指向结果结构的指针（请参阅表 25-71） （NULL：不存储结果）
24-25	frameFiltOpt	R/W	帧筛选选项（有关详细信息，请参阅表 25-74）
26	frameTypes	R/W	在帧筛选中接收的帧类型（有关详细信息，请参阅表 25-75）
27	ccaOpt	R/W	CCA 选项（有关详细信息，请参阅表 25-73）
28	ccaRssiThr	R/W	CCA 的 RSSI 阈值
29			保留
30	numExtEntries	W	扩展地址条目数
31	numShortEntries	W	短地址条目数
32-35	pExtEntryList	W	指向扩展地址条目的列表的指针
36-39	pShortEntryList	W	指向短地址条目的列表的指针
40-47	localExtAddr	W	本地器件的扩展地址
48-49	localShortAddr	W	本地器件的短地址
50-51	localPanID	W	本地器件的 PAN ID
52-54			保留
55	endTrigger	W	导致器件结束 RX 操作的触发器
56-59	endTime	W	endTrigger 的时间参数

表 25-63. IEEE 802.15.4 能量检测扫描命令结构

字节索引	字段名称	类型	说明
14	通道	W	运行开始时调优到的通道： 0：使用现有通道 11 – 26：用作 IEEE 802.15.4 通道，即频率为 [2405 + 5 × (通道 − 11)] MHz 60 – 207：频率为（2300 + 通道）MHz 其他：保留
15	ccaOpt	R/W	CCA 选项（有关详细信息，请参阅表 25-73）
16	ccaRssiThr	R/W	CCA 的 RSSI 阈值
17			保留
18	maxRssi	R	ED 扫描期间记录的最大 RSSI
19	endTrigger	W	导致器件结束 RX 操作的触发器
20-23	endTime	W	endTrigger 的时间参数

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-64. IEEE 802.15.4 CSMA-CA 命令结构

字节索引	字段名称	位	位字段名称	类型	说明
14-15	randomState			R/W	假随机发生器的状态
16	macMaxBE			W	IEEE 802.15.4 MAC 参数 macMaxBE
17	macMaxCSMABackoffs			W	IEEE 802.15.4 MAC 参数 macMaxCSMABackoffs
18	csmaConfig	0-4	initCW	W	CW 参数的初始化值
		5	bSlotted	W	0 用于非开槽 CSMA，1 用于开槽 CSMA
		6-7	rxOffMode	W	0：在 CSMA 回退期间，RX 保持开启 1：如果没有接收到帧，CSMA-CA 算法会暂停接收器 2：如果没有接收到帧，或者在完成帧（包括自动 ACK）后，CSMA-CA 算法会暂停接收器，否则 3：CSMA-CA 算法在退避期间立即暂停接收器
19	NB			R/W	IEEE 802.15.4 CSMA-CA 算法中的 NB 参数
20	BE			R/W	IEEE 802.15.4 CSMA-CA 算法中的 BE 参数
21	remainingPeriods			R/W	暂停的退避倒计时后的剩余周期数
22	lastRssi			R	在最后一次 CCA 操作中测量的 RSSI
23	endTrigger			W	导致器件结束 CSMA-CA 操作的触发器
24-27	lastTimeStamp			R	最后一次 CCA 操作的时间
28-31	endTime			W	endTrigger 的时间参数

表 25-65. IEEE 802.15.4 TX 命令结构

字节索引	字段名称	位	位字段名称	类型	说明
14	txOpt	0	bIncludePhyHdr	W	0：自动查找 PHY 标头。 1：从缓冲器插入 PHY 标头。
		1	bIncludeCrc	W	0：附加自动计算的 CRC。 1：从缓冲器中插入 FCS (CRC)。
		2			保留
		3-7	payloadLenMsb	W	有效载荷长度的最高有效位。只能为非零值才能创建用于测试目的的长非标准数据包
15	payloadLen			W	有效载荷中的字节数
16-19	pPayload			W	指向大小为 payloadLen 的有效载荷缓冲器的指针
20-23	timeStamp			R	发送帧的时间戳

表 25-66. IEEE 802.15.4 接收 ACK 命令结构

字节索引	字段名称	类型	说明
14	seqNo	W	预期的序列号
15	endTrigger	W	触发，导致器件放弃确认接收
16-19	endTime	W	endTrigger 的时间参数

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.5.1.2 IEEE 802.15.4 即时命令结构

表 25-67. IEEE 802.15.4 修改 CCA 即时命令结构

字节索引	字段名称	类型	说明
0-1	commandNo	W	命令编号
2	newCcaOpt	W	用于运行后台级操作的新 ccaOpt 值（有关详细信息，请参阅表 25-73）
3	newCcaRssiThr	W	用于运行后台级操作的新 ccaRssiThr 值

表 25-68. IEEE 802.15.4 修改帧过滤即时命令结构

字节索引	字段名称	类型	说明
0-1	commandNo	W	命令编号
2-3	newFrameFiltOpt	W	用于运行后台级操作的新 frameFiltOpt 值
4	newFrameTypes	W	用于运行后台级操作的新 frameTypes 值

表 25-69. IEEE 802.15.4 启用或禁用源匹配条目即时命令结构

字节索引	字段名称	位	位字段名称	类型	说明
0-1	commandNo			W	命令编号
2	选项	0	bEnable	W	0：禁用条目 1：启用条目
		1	srcPend	W	条目的挂起位的新值
		2	entryType	W	0：扩展地址 1：短地址
		3-7			保留
3	entryNo			W	要启用或禁用的条目索引

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-70. IEEE 802.15.4 请求 CCA 状态即时命令结构

字节索引	字段名称	位	位字段名称	类型	说明
0-1	commandNo			W	命令编号
2	currentRssi			R	当前在通道上观察到的 RSSI
3	maxRssi			R	由于 RX 已启动，在通道上观察到的最大 RSSI
4	ccaInfo	0-1	ccaState	R	当前 CCA 状态的值： 00：空闲 01：忙碌 10：无效
		2-3	ccaEnergy	R	当前能量检测 CCA 状态的值： 00：空闲 01：忙碌 10：无效
		4-5	ccaCorr	R	基于当前相关器的载波侦听 CCA 状态值： 00：空闲 01：忙碌 10：无效
		6	ccaSync	R	基于所发现当前同步字的载波侦听 CCA 状态值： 0：空闲 1：忙碌
		7			保留

25.5.1.3 输出结构

表 25-71. RX 命令

字节索引	字段名称	类型	说明
0	nTxAck	R/W	传输的 ACK 帧总数
1	nRxBeacon	R/W	接收到的信标帧数
2	nRxData	R/W	接收到的数据帧数
3	nRxAck	R/W	接收到的确认帧数
4	nRxMacCmd	R/W	接收到的 MAC 命令帧数
5	nRxReserved	R/W	接收到的具有保留帧类型的帧数
6	nRxOk	R/W	具有 CRC 错误的接收到的帧数
7	nRxIgnored	R/W	接收到的要忽略的帧数
8	nRxBufFull	R/W	由于 RX 缓冲器已满，接收到的帧数被丢弃
9	lastRssi	R	最后接收帧的 RSSI
10	maxRssi	R	在运行中观察到的最高 RSSI
11			保留
12-15	beaconTimeStamp	R	最后接收到的信标帧的时间戳

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.5.1.4 其他结构和位字段

表 25-72. 接收队列条目配置位字段

位	位字段名称	说明
0	bAutoFlushCrc	如果为 1，则自动从 RX 队列中移除 CRC 错误的数据包。
1	bAutoFlushIgn	如果为 1，则自动从 RX 队列中移除根据帧过滤可被忽略的数据包。
2	bIncludePhyHdr	如果为 1，则将接收到的 PHY 标头字段包含在存储的数据包中；否则将其丢弃。
3	bIncludeCrc	如果为 1，则将接收到的 CRC 字段包含在存储的数据包中；否则将其丢弃。这要求 pktConf.bUseCrc 为 1。
4	bAppendRssi	如果为 1，则将 RSSI 字节附加到 RX 队列中的数据包。
5	bAppendCorrCrc	如果为 1，则将相关值和 CRC 结果字节附加到 RX 队列中的数据包。
6	bAppendSrcInd	如果为 1，则附加来自源匹配算法的索引。
7	bAppendTimestamp	如果为 1，则将时间戳附加到 RX 队列中的数据包。

表 25-73. CCA 配置位字段

位	位字段名称	说明
0	ccaEnEnergy	启用能量扫描作为 CCA 源。
1	ccaEnCorr	启用基于相关器的载波感应作为 CCA 源。
2	ccaEnSync	启用基于发现的同步的载波感应作为 CCA 源。
3	ccaCorrOp	基于能量的 CCA 与基于相关器的 CCA 之间使用的运算 0：如果 ccaEnergy 或 ccaCorr 任一为忙，则报告信道忙。 1：如果 ccaEnergy 或 ccaCorr 都为忙，则报告信道忙。
4	ccaSyncOp	基于发现的同步的 CCA 与其他 CCA 之间使用的运算 0：如果 ccaSync 为忙，则始终报告信道忙。 1：如果 ccaSync 空闲，则始终报告信道空闲。
5-6	ccaCorrThr	基于相关器的载波感应中相关峰数量的阈值
7		保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-74. 帧过滤配置位字段

位	位字段名称	说明
0	frameFiltEn	0：禁用帧过滤。 1：启用帧过滤。
1	frameFiltStop	0：接收所有数据包直到结束。 1：一旦帧过滤导致该帧被拒绝，则停止接收该帧。
2	autoAckEn	0：禁用自动 ACK。 1：启用自动 ACK。
3	slottedAckEn	0：非时隙 ACK。 1：时隙 ACK。
4	autoPendEn	0：禁用自动挂起。 1：启用自动挂起。
5	defaultPend	不受自动挂起影响的自动 ACK 数据包中待处理数据位的值。
6	bPendDataReqOnly	0：对任何数据包使用自动挂起。 1：仅对数据请求数据包使用自动挂起。
7	bPanCoord	0：器件不是 PAN 协调器。 1：器件是 PAN 协调器。
8-9	maxFrameVersion	如果 FCF 中的帧版本字段大于此值，则拒绝该帧。
10-12	fcfReservedMask	与 FCF 保留部分进行与运算的值；如果结果非零，则拒绝该帧。
13-14	modifyFtFilter	帧类型过滤前帧类型字段的 MSB 处理： 0：无修改。 1：反转 MSB。 2：将 MSB 设置为 0。 3：将 MSB 设置为 1。
15	bStrictLenFilter	0：接受任何长度 ≥5 的确认帧。 1：仅接受长度为 5 的确认帧。

表 25-75. 帧类型过滤位字段

位	位字段名称	说明
0	bAcceptFt0Beacon	帧类型为 000（信标）的帧处理： 0：拒绝 1：接受
1	bAcceptFt1Data	帧类型为 001（数据）的帧处理： 0：拒绝 1：接受
2	bAcceptFt2Ack	帧类型为 010 (ACK) 的帧处理： 0：拒绝，除非运行 ACK 接收命令 1：始终接受
3	bAcceptFt3MacCmd	帧类型为 011（MAC 命令）的帧处理： 0：拒绝 1：接受
4	bAcceptFt4Reserved	帧类型为 100（保留）的帧处理： 0：拒绝 1：接受
5	bAcceptFt5Reserved	帧类型为 101（保留）的帧处理： 0：拒绝 1：接受
6	bAcceptFt6Reserved	帧类型为 110（保留）的帧处理： 0：拒绝 1：接受

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-75. 帧类型过滤位字段（续）

位	位字段名称	说明
7	bAcceptFt7Reserved	帧类型为 111（保留）的帧处理： 0：拒绝 1：接受

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-76. 短地址条目结构

字节索引	字段名称	说明
0-1	shortAddr	条目的短地址
2-3	panID	条目的 PAN ID

表 25-77. 扩展地址列表结构

字节索引	字段名称	类型	说明
0 – 4K−1	srcMatchEn	R/W	每个 extAddrEntry 的使能位组成的字；第一个字的 LSB 对应于条目 0。数组大小 K = ceil (N/32)，其中 N 为条目数（由 numExtEntries 给出，请参阅 表 25-62），ceil 表示向上取整。
4K – 8K−1	srcPendEn	R/W	每个 extAddrEntry 的待处理数据位组成的字；第一个字的 LSB 对应于条目 0。
8K – 8K+7	extAddrEntry[0]	W	扩展地址编号 0
...			
8K+8n – 8K+8n+7	extAddrEntry[n]	W	扩展地址编号 n
...			
8K+8(N−1) – 8K+8N+7	extAddrEntry[N-1]	W	扩展地址编号 N−1（最后一个条目）

表 25-78. 短地址列表结构

字节索引	字段名称	类型	说明
0：4K−1	srcMatchEn	R/W	每个 shortAddrEntry 的使能位组成的字；第一个字的 LSB 对应于条目 0。数组大小 K = ceil (N/32)，其中 N 为条目数（由 numShortEntries 给出，请参阅表 25-62），ceil 表示向上取整。
4K:8K−1	srcPendEn	R/W	每个 shortAddrEntry 的待处理数据位组成的字；第一个字的 LSB 对应于条目 0。
8K – 8K+3	shortAddrEntry[0]	W	短地址编号 0；该条目是表 25-76 中定义的地址/PAN ID 对。
...			
8K+4n – 8K+4n+3	shortAddrEntry[n]	W	短地址编号 n；该条目是表 25-76 中定义的地址/PAN ID 对。
...			
8K+4(N−1) – 8K+4N+3	shortAddrEntry[N-1]	W	短地址编号 N−1（最后一个条目）；该条目是表 25-76 中定义的地址/PAN ID 对。

表 25-79. 接收相关值/CRC 结果位字段

位	位字段名称	说明
0-5	corr	相关值
6	bIgnore	如果数据包必须被帧过滤拒绝，则为 1；否则为 0。
7	bCrcErr	如果接收到的数据包存在 CRC 错误，则为 1；否则为 0。

Copyright © 2026 Texas Instruments Incorporated

25.5.2 中断

表 25-80 中列出了 IEEE 802.15.4 命令要使用的中断。每个中断都可在系统 CPU 中单独启用。有关何时生成中断的详细信息，请参见节 25.5.4。

表 25-80. 适用于 IEEE 802.15.4 的中断定义

中断号	中断名称	说明
0	COMMAND_DONE	后台级无线电操作命令已完成。
1	LAST_COMMAND_DONE	命令链中最后一个后台级无线电操作命令已完成。
2	FG_COMMAND_DONE	前台无线电操作命令已完成。
3	LAST_FG_COMMAND_DONE	命令链中最后一个前台无线电操作命令已完成。
4	TX_DONE	传输的帧
5	TX_ACK	发送的自动 ACK 帧
16	RX_OK	CRC OK 的已接收帧
17	RX_NOK	具有 CRC 错误的已接收帧
18	RX_IGNORED	设置了忽略标志的已接收帧
22	RX_BUF_FULL	不适合 TX 队列的已接收帧
23	RX_ENTRY_DONE	TX 队列数据条目将状态更改为“已完成”
29	MODULES_UNLOCKED	在启动过程中，Arm Cortex-M0 处理器已打开对 RF 内核模块和存储器的访问权限。
30	BOOT_DONE	RF 内核 CPU 引导完成。
31	INTERNAL_ERROR	无线电 CPU 观察到意外错误。

25.5.3 数据处理

对于所有 IEEE 802.15.4 命令，通过无线方式接收的数据存储在接收队列中。

要传输的数据从发送命令中指定的缓冲区获取。

25.5.3.1 接收缓冲器

正在接收的帧存储在接收缓冲器中。首先，如果在 RX 条目中配置了长度字节或字，由 config.lenSz 存储，并根据无线接收的长度和附加状态信息的配置进行计算。

pRxQ 指向的接收队列中条目元素的格式由节 25.6.1.4 中定义的配置 rxConfig 提供。

在长度字段之后，如果 rxConfig.bIncludePhyHdr 为 1，则会存储接收到的 PHY 标头字节。如果存在长度字段，则该字节是冗余的，保留位除外。接收到的 MAC 标头和 MAC 有效载荷按无线方式存储。如果 rxConfig.bIncludeCrc 为 1，则会存储包含 16 位帧校验序列的 MAC 脚注。

如果 rxConfig.bAppendRssi 为 1，则附加一个指示接收到的 RSSI 值的字节。如果 rxConfig.bAppendCorrCrc 为 1，则附加在表 25-79 中定义的类型的状态字节。如果 rxConfig.bAppendSrcInd 为 1，则会附加一个字节，该字节提供与接收到的数据包的标头匹配的第一个源匹配条目的索引；如果不匹配，则附加一个 0xFF。如果 rxConfig.bAppendTimeStamp 为 1，则附加一个指示帧开始的时间戳。此时间戳是来自无线电计时器的四字节编号。虽然时间戳是多字节的，但不会进行字地址对齐，因此必须按字节写入和读取时间戳。当找到 SFD 时捕获时间戳，但调整时间戳以反映帧的开始（假设标准中有 8 个前导码字节），其定义使其对应于传输端使用的启动触发时间。调整在 syncTimeAdjust 固件定义参数中定义，并可能被覆盖。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

图 25-6 显示了 RX 队列中条目元素的格式。

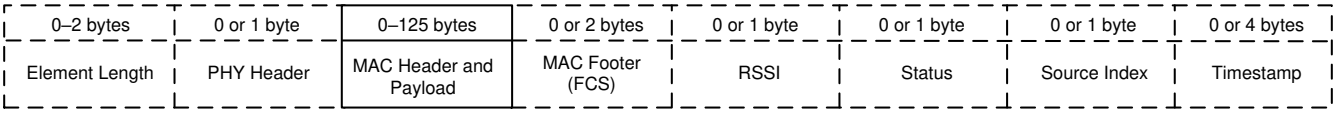


图 25-6. RX 队列条目元素（装订字段可选）

25.5.3.2 发送缓冲区

在发送操作中，指向包含有效载荷的缓冲区的指针由 pPayload 给出。该缓冲区的长度由 payloadLen 单独给出。发送缓冲区的内容由 txOpt 参数给出。发送缓冲区始终包含 MAC 标头和 MAC 有效载荷。如果 txOpt.bIncludePhyHdr 为 1，则缓冲区还包含要作为 PHY 标头发送的字节，作为缓冲区中的第一个字节。如果 txOpt.bIncludeCrc 为 1，则缓冲区的最后两个字节将作为 CRC 发送，而不是自动计算 CRC。

25.5.4 无线电操作命令

在运行本文档中描述的任何无线电操作命令之前，必须使用命令 CMD_RADIO_SETUP 在 IEEE 802.15.4 模式下设置无线电。否则，操作将以错误结束。

在 IEEE 802.15.4 模式下，无线电 CPU 接受两个级别的无线电操作命令。操作可以在后台级别或前台级别运行。每个操作只能以这两个级别之一运行。前台级别的操作通常需要同时运行一个后台级别的操作。

后台级别的操作包括接收操作和能量检测扫描操作。同一时间只能运行其中之一。前台级别的操作包括 CSMA-CA 操作、接收 ACK 操作、发送操作、中止后台级别操作以及修改无线电设置操作。即使有后台级别操作正在运行，这些操作也可以作为一个命令或命令链进入。CSMA-CA 和接收 ACK 操作与后台级别操作同时运行。发送操作会导致后台级别操作被挂起，直到发送完成。表 25-81 显示了后台和前台级别操作的允许组合。违反这些组合会导致前台级别命令即将启动时产生错误，并通过前台级别命令结构的状态字段中的 ERROR_WRONG_BG 状态指示。

表 25-81. 前台和后台级别操作的允许组合

前台级别操作	后台级别操作		
	无	CMD_IEEE_RX	CMD_IEEE_ED_SCAN
无	允许	允许	允许
CMD_IEEE_TX	Allowed1	允许	允许
CMD_IEEE_CSMA	禁止	允许	允许
CMD_IEEE_RX_ACK	禁止	允许	禁止
CMD_IEEE_ABORT_BG	Allowed2	允许	允许

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

非 15.4 无线电操作不能与 15.4 无线电操作同时运行；如果在任一级别上有 15.4 操作正在运行时进入非 15.4 无线电操作，则会发生调度错误。但是，可以创建 15.4 和非 15.4 操作链。

当前台级别操作完成时，会产生 FG_COMMAND_DONE 中断。如果该命令是链中的最后一个命令，则还会产生 LAST_FG_COMMAND_DONE 中断（请参阅表 25-80）。后台级别操作使用公共中断 COMMAND_DONE 和 LAST_COMMAND_DONE（请参阅表 25-80）。

命令结构的状态字段在操作期间会更新。提交命令时，系统 CPU 会以 IDLE 状态写入该字段。在操作期间，无线电 CPU 会更新该字段以指示工作模式。当操作完成时，无线电 CPU 会写入一个状态，指示命令已完成。表 25-82 列出了 IEEE 802.15.4 无线电操作的状态代码。

表 25-82. IEEE 802.15.4 无线电操作状态代码

编号	名称	说明
操作未完成		
0x0000	空闲	操作未启动
0x0001	PENDING	等待启动触发器
0x0002	运行	正在运行操作
0x2001	IEEE_SUSPENDED	操作已挂起
正常操作结束		
0x2400	IEEE_DONE_OK	操作正常结束
0x2401	IEEE_DONE_BUSY	CSMA-CA 操作以失败结束
0x2402	IEEE_DONE_STOPPED	操作在停止命令后停止
0x2403	IEEE_DONE_ACK	接收到 ACK 数据包，待处理数据位已清除
0x2404	IEEE_DONE_ACKPEND	接收到 ACK 数据包，待处理数据位已设置
0x2405	IEEE_DONE_TIMEOUT	操作因超时而结束
0x2406	IEEE_DONE_BGEND	前台操作因必要的后台级别操作结束而结束
0x2407	IEEE_DONE_ABORT	操作被命令中止
操作以错误结束		
0x0806	ERROR_WRONG_BG	前台级别操作与正在运行的后台级别操作不兼容
0x2800	IEEE_ERROR_PAR	非法参数
0x2801	IEEE_ERROR_NO_SETUP	无线电未设置为 IEEE 802.15.4 模式
0x2802	IEEE_ERROR_NO_FS	运行 RX 或 TX 时合成器未编程
0x2803	IEEE_ERROR_SYNTH_PROG	合成器编程失败
0x2804	IEEE_ERROR_RXOVF	操作期间发生 RX 溢出
0x2805	IEEE_ERROR_TXUNF	操作期间发生 TX 下溢

每个操作都会列出给出每种状态的条件。表 25-82 中列出的一些错误原因不会在这些列表中重复出现。在某些情况下，可能发生节 25.3 中描述的一般错误原因。在所有这些情况下，节 25.3 中定义的操作结果都是 ABORT。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.5.4.1 RX 操作

接收无线电操作是一个后台级操作，使用 CMD_IEEE-RX 命令和表 25-62 中给出的命令结构启动。

在 RX 操作开始时，无线电 CPU 会等待开始触发器，然后根据通道参数对频率进行编程。如果通道为 0xFF，则操作继续在已配置的通道上运行。这要求在另一个接收操作或者合成器编程操作之后进行操作。如果频率合成器未运行，则运行以错误结束。对频率进行编程后，无线电 CPU 将接收器配置为接收 IEEE 802.15.4 数据包。

当解调器在帧上获得同步时，首先读取 PHY 标头。该字节的 7LSB 提供帧长度。进一步的治疗取决于 frameFiltOpt 的设置。如果 frameFiltOpt.frameFiltEn 为 1，则会按照如下所述执行进一步的帧滤波。如果其为 0，则不执行帧滤波。

由接收到的 PHY 标头给出的字节数会接收并存储在 pRxQ 提供的接收队列中。如节 25.7.1 中所述，格式取决于 rxConfig。PHY 有效载荷的最后两个字节是数据包的 FCS 或者 CRC。根据 FCS 规范检查这些字节，进一步的处理取决于 CRC 结果。

如果存在 CRC 错误且 rxConfig.bAutoFlushCrc 为 1，则将从 RX 缓冲器中丢弃数据包。如果没有可用的 RX 缓冲器和足够的可用空间来保存接收到的数据包，则接收到的数据将被丢弃。如果 frameFiltOpt.frameFiltStop i 为 1，则接收停止，否则接收数据包以便检查 CRC。

25.5.4.1.1 帧过滤和源匹配

如果 frameFiltOpt.frameFiltEn 为 1，则按照本节所述执行帧过滤和源匹配。帧过滤有以下几种用途：

- 区分不同的数据包类型
- 拒绝目标地址不匹配的数据包
- 拒绝版本未知或字段非法的数据包
- 自动识别源地址
- 自动发送确认帧
- 基于源地址自动插入待处理数据位

25.5.4.1.1.1 帧过滤

当启用帧过滤时，无线电 CPU 会检查数据包的 MAC 标头。首先检查帧控制字段 (FCF)。帧类型子字段是 FCF 中要检查的第一个子字段，它决定了后续处理过程。在进行检查之前，会根据 frameFiltOpt.modifyFtFilter 来处理帧类型的 MSB。此修改的结果仅在检查时使用，而不会在将 FCF 存储到 RX 队列条目时使用。对于帧类型字段的八个可能值（包括四个保留值）中的每一个，都可以设置为接受或拒绝该帧。这由 frameTypes 的各个位控制。如果帧类型为确认帧 (010b) 且前台正在运行 CMD_RX_ACK 操作，则即使 frameTypes.bAcceptFt2Ack 为 0，也会进一步处理该数据包。节 25.5.4.5 详细介绍了这种情况下的处理过程。

对 Frame Version 和 Reserved 子字段执行过滤。如果帧版本大于 frameFiltOpt.maxFrameVersion，则拒绝该帧。

如果 Reserved 子字段与 frameFiltOpt.fcfReservedMask 进行逻辑与运算的结果非零，则拒绝该帧。检查地址字段以确定是否必须接受该帧。此过滤遵循第三级过滤的规则（请参阅 IEEE 802.15.4 标准）。检查本地地址时，使用 localExtAddr 或 localShortAddr 字段；检查本地 PAN ID 时，使用 localPanID 字段。

如果 frameFiltOpt.bStrictLenFilter 为 1 且帧类型指示该帧为确认帧，而且 PHY 有效载荷的长度不是 5（这是格式正确的 ACK 帧的长度），则拒绝该帧。

如果 frameFiltOpt.frameFiltStop 为 1 且帧过滤得出的结论是该帧应被拒绝，则停止接收，无线电返回同步搜索状态。否则，将接收该帧直至结束。

无线电 CPU 检查标头以确定是否需要传输确认帧。这会得出一个初步结果；ACK 实际发送与否取决于帧结束时的状态。发送传输帧的条件在节 25.5.4.1.3 中给出。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.5.4.1.1.2 源匹配

源匹配在源地址存在的帧过滤所接受的帧上执行。如果源地址是扩展地址，则将接收到的地址与列表 pExtEntryList 中的条目进行比较。如果源地址是短地址，则会将接收到的地址和源 pan ID 与列表 pShortEntryList 中的条目进行比较。

列表可以保留的条目数由 numExtEntries 和 numShortEntries 给出。如果这些值中的任何一个为 0，则不会对相应类型的地址执行源匹配，相应的指针为 NULL。列表以源映射使能位 srcMatchEn 开头，并继续以待处理的使能位 srcPendEn 开头，后接列表条目，请参阅表 25-76 和表 25-77。使能位包含保存列表中每个条目的使能位所需的 32 位字的数量。对于相应 srcMatchEn 位为 1 的每个条目，将该条目与接收到的扩展地址源地址进行比较，或者与接收到的源地址和短地址的 PAN ID 进行比较。如果找到匹配项，则会存储索引，并在消息页脚中报告（如果已配置）（请参阅节 25.7.1）。如果未找到匹配项，则返回的索引为 0xFF。

源匹配过程也可用于查找要在自动确认帧中传输的挂起数据位（请参阅节 25.5.4.1.3）。如果 frameFiltOpt.autoPendEn 为 1 并且找到了源匹配项，则挂起的数据位会设置为 srcPendEn 中对应于匹配索引的位的值。如果未找到匹配项或 frameFiltOpt.autoPendEn 为 0，则待处理数据位设置为等于 frameFiltOpt.defaultPend。如果 frameFiltOpt.bPendDataReqOnly 为 1，则无线电 CPU 会研究该帧以确定它是否是命令帧标识符设置为数据请求的 MAC 命令帧。否则，无论源匹配结果和 frameFiltOpt.defaultPend 的值如何，自动 ACK 的挂起数据位都会设置为 0。

25.5.4.1.1.2 帧接收

完成帧过滤后，数据包的其余部分会被接收并存储在接收队列中。PHY 数据包的最后两个字节是 MAC 帧尾，亦称 FCS（帧校验序列），是一个经过校验的 CRC 值。仅当 rxConfig.bIncludeCrc 为 1 时，该 CRC 才会存储到队列中。

收到的帧的状态取决于帧过滤结果和 CRC 校验结果。必须维护两个状态位：bCrcErr 和 bIgnore。如果已配置这些位，它们存在于 RX 队列条目的状态字节中。如果帧存在 CRC 错误，则 bCrcErr 位为 1，否则为 0。如果启用了帧过滤且该帧被帧过滤拒绝，则 bIgnore 位为 1，否则为 0。

备注

如果 frameFiltOpt.frameFiltStop 为 1，则永远不会观察到 bIgnore 等于 1 的帧，因为接收会停止，收到的字节也不会存入队列。如果 rxConfig.bAutoFlushCrc 为 1，则接收完成后，bCrcErr 等于 1 的数据包将从队列中移除；如果 rxConfig.bAutoFlushIgn 为 1，则接收完成后，bIgnore 等于 1 的数据包将从队列中移除。

数据包接收完毕后，会引发一个中断，并且 pOutput 中的某个计数器会递增。表 25-83 列出了这些条件。

表 25-83. RX 操作中递增计数器和引发中断的条件

条件	递增的计数器	产生的中断
收到帧，CRC 正确且帧过滤已禁用	nRxData	RX_OK
收到帧，CRC 错误	nRxNok	RX_NOK
收到帧，但无法放入 RX 队列	nRxBufFull	RX_BUF_FULL
收到信标帧，CRC 正确且 bIgnore = 0	nRxBeacon	RX_OK
收到 ACK 帧，CRC 正确且 bIgnore = 0	nRxAck	RX_OK
收到数据帧，CRC 正确且 bIgnore = 0	nRxData	RX_OK
收到 MAC 命令帧，CRC 正确且 bIgnore = 0	nRxMacCmd	RX_OK
收到具有保留帧类型的帧，CRC 正确且 bIgnore = 0	nRxReserved	RX_OK
收到帧，CRC 正确且 bIgnore = 1	nRxIgnored	RX_IGNORED
RX 队列中的第一个 RX 数据条目将状态更改为“已完成”	—	RX_ENTRY_DONE

收到帧时，在接收该帧过程中观察到的 RSSI 将写入

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

pOutput->lastRssi。如果该帧是信标帧，已被帧过滤接受且 CRC 正确，那么帧开头的时间戳将写入 pOutput->beaconTimeStamp。如果时间戳附加到 RX 条目元素中（请参阅节 25.7.1），对于信标帧，这两个时间戳是相同的。

数据包接收完毕后，无线电 CPU 要么重新开始同步搜索，要么发送确认帧。发生后者的条件如节 25.5.4.1.3 所示。

25.5.4.1.3 ACK 发送

接收到数据包后，如果满足以下所有条件，无线电 CPU 就会启动确认帧的传输：

- 通过 frameFiltOpt.autoAckEn = 1 启用自动 ACK
- 帧滤波接受帧 (bIgnore = 0)
- 帧是数据帧或 MAC 命令帧
- 目标地址不是广播地址
- FCF 的 ACK 请求位被置位
- 通过 CRC 校验 (bCrcErr = 0)
- 该帧适合接收队列

ACK 数据包的发送时间由无线电 CPU 计时，具体取决于 frameFiltOpt.slottedAckEn。如果该位为 0，则 ACK 数据包在接收到的数据包结束后传输 192µs。否则使用时隙 ACK。假设接收到的数据包在退避时隙边界上启动。然后，ACK 帧在接收帧结束后至少一个 TurnaroundTime 符号周期后的第一个退避边界处，开始一整数个退避周期。

自动传输的 ACK 帧的内容如下：

- PHY 标头为 0x05
- PHY 有效载荷由一个 3 字节 MAC 标头及一个 2 字节 MAC 标尾组成
- MAC 标头从 2 字节 FCF 开始，具有以下字段：
  - 帧类型子字段为 010b
  - 帧挂起子字段的设置如节 25.5.4.1.1.2 中所述
  - 其余子字段设置为全零
- MAC 标头中的下一个字节是序列号，序列号设置为等于接收到的帧的序列号
- MAC 页脚是自动计算的 FCS

发送 ACK 帧后，发出 TX_ACK 中断。然后，无线电 CPU 再次启用接收器。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.5.4.1.4 接收操作结束

接收操作可因由 endTrigger 和 endTime 给出的结束触发条件而结束，或者通过命令结束。可以结束接收操作的命令有：即时命令 CMD_ABORT 和 CMD_STOP，以及前台级无线电操作命令 CMD_IEEE_ABORT_BG。结束触发条件和 CMD_STOP 命令使接收器持续运行直至帧结束，或者如果在接收数据包时观察到该指令，则使接收过程在原本应停止的位置停止。CMD_ABORT 和 CMD_IEEE_ABORT_BG 命令使接收器以能够实现的尽可能快的方式停止。

接收操作因表 25-84 中列出的某一原因而结束。命令结束后，命令结构中的状态字段会指示操作结束的原因。在所有情况下，都会引发 COMMAND_DONE 中断。在每种情况下，结果均指示为 TRUE、FALSE 或 ABORT。接收器根据结果决定是启动 pNextOp 中指示的下一条命令（如有），还是返回 IDLE 状态。在接收操作结束前，无线电 CPU 会将接收操作期间观察到的最大 RSSI 写入 pOutput->maxRssi。

如果在前台启动了传输操作，接收操作将暂停。接收器会像被中止时一样停止，但在切换到传输模式时，频率合成器会尽可能保持开启。接收器停止后，命令结构的状态字段会设置为 IEEE_SUSPENDED。传输命令完成后，接收器会重新启动，命令结构的状态字段会设回 RUNNING。

表 25-84. 接收操作结束

条件	状态代码	结果
观察到结束触发条件并完成任何正在进行的接收	IEEE_DONE_OK	真
收到 CMD_STOP	IEEE_DONE_STOPPED	否
收到 CMD_ABORT 或 CMD_IEEE_ABORT_BG	IEEE_DONE_ABORT	ABORT
观察到非法参数	IEEE_ERROR_PAR	ABORT

25.5.4.1.5 CCA 监测

在接收器运行期间，无线电 CPU 会监测一些信号以用于空闲通道评估。该监测由 ccaOpt 控制。CCA 有三种来源：RSSI 高于阈值状态 (ccaEnergy)、基于相关值的载波侦听 (ccaCorr) 和基于同步检测的载波侦听 (ccaSync)。每种来源都可能具有 BUSY、IDLE 或 INVALID 三种状态。

RSSI 高于阈值状态通过监测 RSSI 来维护。如果 RSSI 大于或等于 ccaRssiThr，则 ccaEnergy 为 BUSY。如果 RSSI 小于 ccaRssiThr，则 ccaEnergy 为 IDLE。当由于接收器已启动而导致 RSSI 计算尚未完成时，ccaEnergy 为 INVALID。

基于相关值的载波侦听利用接收器中为 SFD 搜索算法所定义的相关峰值来进行监测。如果在最后 8 个符号周期 (32µs) 中观察到的相关峰值数量大于 ccaOpt.corrThr，则 ccaCorr 为 BUSY；否则 ccaCorr 为 IDLE。ccaOpt.corrThr 的取值范围为 0 到 3。当接收器正在接收帧时，无论观察到的相关峰值如何，ccaCorr 都为 BUSY。如果从接收器启动至今不足 8 个符号周期，且自接收器启动以来观察到的相关峰值数量小于或等于 ccaOpt.corrThr，则 ccaCorr 为 INVALID。

基于同步检测的载波侦听由无线电 CPU 维护，如下所示。如果接收器获得同步，无线电 CPU 会检查 PHY 标头以确定帧长度。无线电 CPU 在此帧的持续时间内认为通道处于 BUSY 状态。即使由于帧过滤导致该帧的接收停止并重新开始同步搜索，此检查也会执行。如果在通道被视为 BUSY 期间再次检测到同步，则通道将被视为 BUSY，直到这两个帧根据各自观察到的帧长度都结束为止。ccaSync 不使用 INVALID 状态。

如果无线电正在发送 ACK 或因执行 TX 操作而被挂起，则 ccaEnergy、ccaCorr 和 ccaSync 全为 BUSY。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

整体 CCA 状态 ccaState 取决于 ccaOpt 的 ccaEnEnergy、ccaEnCorr 和 ccaEnSync 位，以及 ccaCorrOp 和 ccaSyncOp 位。确定 ccaState 的规则如下（描述中 ccaTmp 为一个帮助程序状态）：

- 如果 ccaEnEnergy = 0 且 ccaEnCorr = 0 且 ccaEnSync = 0，则 ccaState = IDLE
- 如果 ccaEnEnergy = 1 且 ccaEnCorr = 0，则 ccaTmp = ccaEnergy
- 如果 ccaEnEnergy = 0 且 ccaEnCorr = 1，则 ccaTmp = ccaCorr
- 如果 ccaEnEnergy = 1 且 ccaEnCorr = 1 且 ccaCorrOp = 0，则：
  - 如果 ccaEnergy 或 ccaCorr 中任一个为 BUSY，则 ccaTmp = BUSY；
  - 否则，如果 ccaEnergy 或 ccaCorr 中任一个为 INVALID，则 ccaTmp = INVALID；
  - 否则，ccaTmp = IDLE
- 如果 ccaEnEnergy = 1 且 ccaEnCorr = 1 且 ccaCorrOp = 1，则：
  - 如果 ccaEnergy 或 ccaCorr 中任一个为 IDLE，则 ccaTmp = IDLE；
  - 否则，如果 ccaEnergy 或 ccaCorr 中任一个为 INVALID，则 ccaTmp = INVALID；
  - 否则，ccaTmp = BUSY
- 如果 ccaEnEnergy = 0 且 ccaEnCorr = 0 且 ccaEnSync = 1，则 ccaState = ccaSync
- 否则，如果 ccaEnSync = 1 且 ccaSyncOp = 0，则：
  - 如果 ccaTmp 或 ccaSync 中任一个为 BUSY，则 ccaState = BUSY；
  - 否则，如果 ccaTmp 为 INVALID，则 ccaState = INVALID；
  - 否则，ccaState = IDLE
- 否则，如果 ccaEnSync = 1 且 ccaSyncOp = 1，则：
  - 如果 ccaTmp 或 ccaSync 中任一个为 IDLE，则 ccaState = IDLE；
  - 否则，如果 ccaTmp 为 INVALID，则 ccaState = INVALID；
  - 否则，ccaState = BUSY

根据 IEEE 802.15.4 标准，要使整体 CCA 状态为 IDLE，ccaSync 的 CCA 状态必须为 IDLE。因此，为了符合标准，ccaEnSync 设为 1，cceSyncOp 设为 0。

IEEE 802.15.4 标准中定义的 CCA 模式 1 通过设置 ccaEnEnergy = 1 和 ccaEnCorr = 0 来实现。CCA 模式 2 通过设置 ccaEnEnergy = 0 和 ccaEnCorr = 1 来实现。CCA 模式 3 通过设置 ccaEnEnergy = 1 和 ccaEnCorr = 1 来实现。在 CCA 模式 3 下，ccaCorrOp 可以为 0 或 1；这区分了逻辑运算符 AND (1) 和 OR (0)，如 IEEE 802.15.4 标准所述。

系统 CPU 可以通过发出即时命令 CMD_IEEE_CCA_REQ 来读取 CCA 状态和当前 RSSI。如果 CMD_IEEE_CSMA 操作正在前台运行，无线电 CPU 也会自主监测 CCA。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.5.4.2 能量检测扫描操作

能量检测扫描无线电操作是一个后台级别的操作，由 CMD_IEEE_ED_SCAN 命令启动，并使用如表 25-63 所示的命令结构。

在 RX 操作开始时，无线电 CPU 会等待开始触发器，然后根据通道参数对频率进行编程。如果通道为 0xFF，则操作继续在已配置的通道上运行。这要求在另一个接收操作或者合成器编程操作之后进行操作。如果频率合成器未运行，则运行以错误结束。对频率进行编程后，无线电 CPU 将接收器配置为接收 IEEE 802.15.4 数据包，但不存储任何接收到的数据。

当接收器运行时，CCA 会按照节 25.5.4.1.5 中的描述进行更新。当解调器在帧上获得同步时，将读取 PHY 标头。这仅用于基于找到的同步来确定载波侦听，之后会立即重新开始同步搜索。

能量检测扫描操作在与 RX 操作相同的条件下结束，如节 25.5.4.1.4 所述。在操作结束之前，无线电 CPU 会将能量检测扫描操作期间观察到的最大 RSSI 写入 maxRssi。

25.5.4.3 CSMA-CA 操作

CSMA-CA 操作是在接收或能量检测扫描操作之上运行的前台操作。如果在能量检测扫描操作的基础上运行，则不执行能量检测扫描过程，而是在无需接收数据包的情况下启动接收器。此操作从 CMD_IEEE_CSMA 命令开始，并使用中表 25-64 提供的命令结构。

在 CSMA-CA 操作开始时，无线电 CPU 会等待启动触发。

无线电 CPU 维护一个变量 *CW*，该变量初始化为 csmaConfig.initCW。

如果 remainingPeriods 在命令启动时为非零，则无线电 CPU 会延迟从启动触发开始测量的该退避周期数（默认 320µs），然后再继续。否则，无线电 CPU 将绘制一个 0 到 $2^{(BE)} - 1$ 范围内的假随机数，其中在表 25-64 列出了 BE。然后，无线电 CPU 会等待从启动触发开始算起的退避周期数，然后再继续。

等待时间过后，无线电 CPU 将从后台操作检查 CCA 状态，如节 25.5.4.1.5 中所述。如果 CCA 状态为 INVALID，则无线电 CPU 会等待，然后重试。如果 csmaConfig.bSlotted = 1，则等待一个退避周期，否则将等待，直到 RSSI 结果可用。如果 CCA 状态为 IDLE，则无线电 CPU 将 CW 递减 1，如果这导致值为零，则 CSMA-CA 操作将成功结束。如果这导致出现非零值，则无线电 CPU 会等待一个从等待时间结束计时的退避周期，然后如上所述再次检查 CCA 状态。

如果在检查 CCA 状态时通道处于 BUSY 状态，则无线电 CPU 按如下方式更新变量：

CW = csmaConfig.initCW；NB + = 1；BE + = min (BE + 1, macMaxBE)；

如果此更新后的 NB 大于 macMaxCSMABackoffs，则 CSMA-CA 操作以失败结束。否则，无线电 CPU 会像前文所述那样绘制退避周期的随机数以等待，并像以前一样继续。如果 csmaConfig.bSlotted = 1，则等待从上一个等待时间结束后的下一个退避周期开始；否则，等待从上一个等待时间结束后的可配置时间开始。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

图 25-7 展示了 CSMA-CA 操作的流程图。

除了前面所述以成功或失败结束的 CSMA-CA 操作之外，该操作还可以根据 endTrigger 和 endTime 或命令给出的结束触发条件而结束。可结束 CSMA-CA 操作的命令为立即命令 CMD_ABORT、CMD_STOP、CMD_IEEE-ABORT_FG 及 CMD_IEEE-STOP_FG。当 CSMA-CA 操作结束时，无线电 CPU 向 lastTimeStamp 写入 CCA 检查完成前最近等待周期结束时的计时器值，以及向 lastRssi 写入当时的 RSSI 值。如果操作因超时或停止命令而结束，则无线电 CPU 会写入 remainingPeriods，其中包含等待时间剩余的退避周期数。否则，无线电 CPU 会向 remainingPeriods 写入 0。

假随机算法基于最大长度 16 位线性反馈移位寄存器 (LFSR)。种子 randomState 以提供。操作结束时，无线电 CPU 会将当前状态写回至此字段。如果 randomState 为 0，则无线电 CPU 通过将 LFSR 初始化为无线电计时器的 16 个 LSB 来自行供电。此值存在一定的随机性，但这是有限的，尤其是对于时隙 CSMA-CA，因此建议由系统 CPU 播种真随机数（或基于真随机种子的假随机数）。如果无线电计时器的 16 个 LSB 全部为 0，则替换为另一个固定值。

根据 csmaConfig.rxOffMode，如果有足够的时间，底层 RX 操作可能会在另一个 CCA 检查之前的退避期间暂停。不同的值具有以下含义：

- rxOffMode = 0：在 CSMA 回退期间，无线电保持开启状态。
- rxOffMode = 1：如果正在接收帧，正在传输 ACK，或者正在这些帧之间的转换，则无线电保持开启状态。否则，无线电将关闭，直至退避周期结束。
- rxOffMode = 2：如果正在接收帧，正在传输 ACK，或者正在这些帧之间的转换，则无线电保持开启状态，直到数据包完全接收且 ACK 被传输（如果适用）。之后，无线电会关闭，直到退避周期结束。
- rxOffMode = 3：在回退周期开始时，无线电立即关闭。这将中止正在接收的帧或者正在传输的 ACK。无线电保持关闭状态，直至回退周期结束。

如果无线电以这种方式关闭，接收器会提前重启，以便完成下一个 CCA 操作，并且无线电只会在有足够时间时关闭它。此特性可以在并不总是需要处于 RX 状态的系统中用于节能。除模式 0 之外的所有模式都可能导致帧丢失，并且概率会增加。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

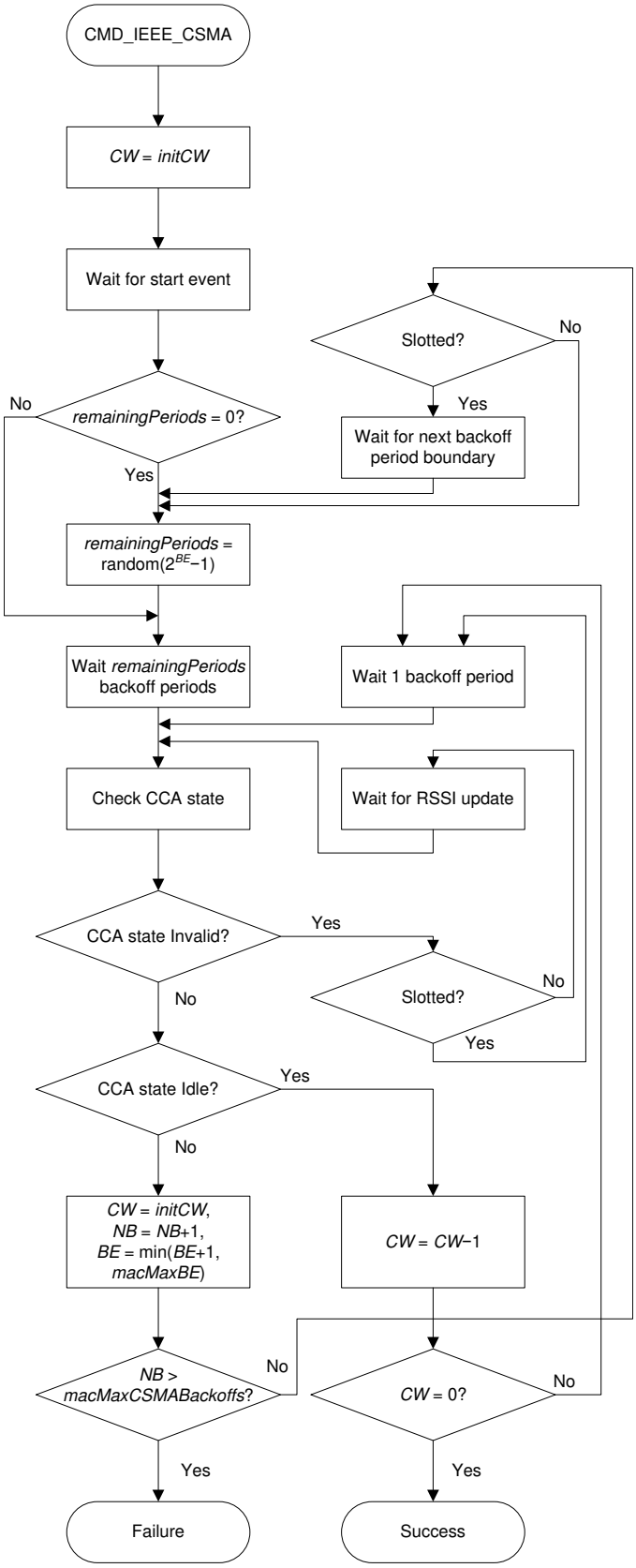


图 25-7. CSMA-CA 操作

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

对于根据 IEEE 802.15.4 进行的操作，在开始新的 CSMA-CA 操作之前，必须按如下方式初始化参数：

1. 必须将 randomState 设置为随机值
2. 对于时隙 CSMA-CA，csmaConfig.initCW 必须设置为 2；对于非时隙 CSMA-CA，必须设置为 1
3. 对于时隙 CSMA-CA，必须将 csmaConfig.bSlotted 设置为 1，对于非时隙 CSMA-CA，必须将其设置为 0
4. NB 必须设置为 0
5. BE 必须设置为 macMinBE，但具有延长的电池寿命的时隙 CSMA-CA 除外，其中 BE 必须设置为 min (2, macMinBE)
6. remainingPeriods 必须设置为 0
7. 必须将 macMaxBE 和 macMaxCSMABackoffs 设置为其相应的 MAC PIB 属性

对于时隙 CSMA-CS，必须将 startTrigger 设置为在退避时隙边界上出现。对于时隙 CSMA-CA，endTrigger 必须设置为在最晚的时间发生，并且事务可在超级帧内完成，如 IEEE 802.15.4 标准中所规定。如果 CSMA-CA 由于超时而结束，则可以在下一个超级帧处重新启动 CSMA，而无需修改参数（结束时间可能除外）。

表 25-85 列出了 CSMA-CA 操作结束的原因。命令结束后命令结构的状态字段指示操作结束的原因。在所有情况下都会产生 FG_COMMAND_DONE 中断。在每种情况下，都会指示结果是 TRUE、FALSE 还是 ABORT。这指示是在 pNextOp 中启动下一个命令（如果有），还是返回到 IDLE 状态。

表 25-85. CSMA-CA 操作结束

条件	状态代码	结果
CSMA-CA 操作成功完成	IEEE_DONE_OK	TRUE
CSMA-CA 操作完成，但未能成功	IEEE_DONE_BUSY	FALSE
发生结束触发	IEEE_DONE_TIMEOUT	FALSE
收到 CMD_STOP 或 CMD_IEEE-STOP_FG	IEEE_DONE_STOPPED	FALSE
收到的 CMD_ABORT 或 CMD_IEEE-ABORT_FG	IEEE_DONE_ABORT	ABORT
后台操作已结束	IEEE_DONE_BGEND	ABORT
观察到的非法参数	IEEE_ERROR_PAR	ABORT

当操作结束时，最后一次 CCA 检查的时间（即写入 lastTimeStamp 的时间）被定义为事件 1，可用于为后续链接操作计时。

25.5.4.4 发送操作

发送操作是前台操作，用于发送一个数据包。该操作使用 CMD_IEEE-TX 命令启动，并使用表 25-65 中给出的命令结构。

当无线电 CPU 接收到命令时，它会等待启动触发。在该等待时间内，任何后台级操作都会继续运行。当启动触发时，无线电 CPU 暂停接收器并配置发送器。假设合成器通电并经过校准，因此如果未运行后台级操作，则必须在 TX 操作之前添加校准合成器命令。如果频率合成器未运行，则运行以错误结束。

发送器将在缓冲器指针中找到的有效载荷发送到 pPayload（由 payloadLen 字节组成）。如果 txOpt.payloadLenMsb 为非零值，则将该字段乘以 256 并添加到 payloadLen，以创建不符合 IEEE 802.15.4 标准的长帧（用于测试）。如果 txOpt.bIncludePhyHdr 为 0，则无线电 CPU 会自动插入根据有效载荷长度计算得出的 PHY 标头。否则，无线电 CPU 不会插入 PHY 标头，因此为了符合 IEEE 802.15.4 标准，有效载荷缓冲器中的第一个字节必须是 PHY 标头。

然后，有效载荷按在有效载荷缓冲器中发现的情况传输。如果 txOpt.bIncludeCrc 为 0，则无线电 CPU 会附加两个 CRC 字节（根据 IEEE 802.15.4 标准计算）。否则不会附加 CRC，因此为了符合 IEEE 802.15.4 MAC 标准，有效载荷缓冲器中的最后两个字节必须是 MAC 页脚。发送操作可以由立即命令 CMD_ABORT、CMD_STOP、CMD_IEEE-ABORT_FG 和 CMD_IEEE-STOP_FG 结束。如果接收到 CMD_ABORT 或 CMD_IEEE-ABORT_FG，则传输在数据包中间尽快结束。如果在无线电 CPU 等待启动触发时接收到 CMD_STOP 或

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

CMD_IEEE-STOP_BG，则操作会在不进行任何传输的情况下结束；否则，传输完成，但结束状态和结果会有所不同，如下所述。

当数据包的传输开始时，用于启动调制解调器的触发器 RAT 时间由无线电 CPU 写入 timeStamp 字段。与 CMD_IEEE-TX 命令的配置启动时间相比，此时间戳因固件定义的参数 startToTXRatOffset 而延迟。如果发送器和接收器已同步 RAT 计时器，则该时间戳与附加到 RX Entry 元素的时间戳相同，如节 25.7.1 所示，尽管接收器侧存在估计不确定性。

当操作结束时，发送帧的结束时间被定义为事件 1，可用于为后续链接操作计时。

表 25-86 列出了发送操作结束的原因。命令结束后命令结构的状态字段指示操作结束的原因。在所有情况下都会产生 FG_COMMAND_DONE 中断。在每种情况下，都会指示结果是 TRUE、FALSE 还是 ABORT。这指示是在 pNextOp 中启动下一个命令（如果有），还是返回到 IDLE 状态。

表 25-86. 发送操作结束

条件	状态代码	结果
已发送数据包	IEEE_DONE_OK	TRUE
接收到 CMD_STOP 或 CMD_IEEE-STOP_FG，然后完成发送（如果已启动）	IEEE_DONE_STOPPED	FALSE
收到的 CMD_ABORT 或 CMD_IEEE-ABORT_FG	IEEE_DONE_ABORT	ABORT
观察到的非法参数	IEEE_ERROR_PAR	ABORT

25.5.4.5 接收确认操作

接收 ACK 操作是一个在接收操作之上运行的前台级操作。该操作从 CMD_IEEE-RX_ACK 命令开始，并使用表 25-66 中列出的命令结构。

在接收 ACK 操作开始时，无线电 CPU 会等待启动触发。如果在接收 ACK 操作之前由于 TX 操作而暂停接收器，则在发生启动触发之前不会恢复后台级 RX 操作。

当接收 ACK 操作运行时，后台级 RX 操作正常运行。但是，除了查找数据包之外，操作还会查找 seqNo 中给出了序列号的 ACK 数据包。仅当数据包配置为在后台级接收操作 (frameTypes.bAcceptFt2Ack = 1) 中时，数据包才会存储在接收队列中。如果在后台 RX 操作中滤除 ACK 数据包，则对于 ACK 数据包，将接收序列号，如果匹配，还将接收 FCS。

如果接收到带有所请求序列号的 ACK 数据包，则会检查 FCS。如果 CRC 为 OK，则接收 ACK 操作结束，否则它会继续。如果 ACK 接收到 OK，则检查标头的挂起数据位。

除了如前所述在接收到 ACK 后结束的接收 ACK 操作之外，该操作还可以根据 endTrigger 和 endTime 或命令给出的结束触发条件而结束。可结束接收 ACK 操作的命令为立即命令 CMD_ABORT、CMD_STOP、CMD_IEEE-ABORT_FG 及 CMD_IEEE-STOP_FG。

接收 ACK 操作由于表 25-87 中列出的原因之一而结束。命令结束后命令结构的状态字段指示操作结束的原因。在所有情况下都会产生 FG_COMMAND_DONE 中断。在每种情况下，都会指示结果是 TRUE、FALSE 还是 ABORT。这指示是在 pNextOp 中启动下一个命令（如果有），还是返回到 IDLE 状态。

表 25-87. 接收 ACK 操作结束

条件	状态代码	结果
已成功接收到请求的 ACK 并清除挂起数据位	IEEE_DONE_ACK	FALSE
在设置了挂起数据位的情况下成功接收到请求的 ACK	IEEE_DONE_ACKPEND	TRUE
发生结束触发	IEEE_DONE_TIMEOUT	FALSE
收到 CMD_STOP 或 CMD_IEEE-STOP_FG	IEEE_DONE_STOPPED	FALSE
收到的 CMD_ABORT 或 CMD_IEEE-ABORT_FG	IEEE_DONE_ABORT	ABORT
后台操作已结束	IEEE_DONE_BGEND	ABORT

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-87. 接收 ACK 操作结束（续）

条件	状态代码	结果
观察到的非法参数	IEEE_ERROR_PAR	ABORT

25.5.4.6 中止后台级操作命令

中止后台级操作命令是一条前台级命令，用于停止正在后台运行的命令。中止后台级操作命令定义为前台操作命令，以使其具有起始时间，并可与其他前台操作命令进行链式组合。该命令通过 CMD_IEEE_ABORT_BG 命令执行，并使用仅包含最少参数集的命令结构。

在中止后台级操作开始时，无线电 CPU 等待启动触发条件，然后中止正在进行的后台级接收或能量检测扫描操作。

在等待启动触发条件期间，可通过命令停止该操作。可停止该操作的命令包括 CMD_ABORT、CMD_STOP、CMD_IEEE_ABORT_FG 和 CMD_IEEE_STOP_FG。前两条命令无论如何都会导致后台级操作停止。

中止后台级操作会因表 25-88 中列出的某一原因而结束。命令结束后，命令结构中的状态字段会指示操作结束的原因。在所有情况下，都会引发 FG_COMMAND_DONE 中断。每种情况都会指明结果为 TRUE、FALSE 还是 ABORT。这会指示启动 pNextOp 中的下一条命令（如有）还是返回 IDLE 状态。

表 25-88. 后台级操作以 ABORT 结果结束

条件	状态代码	结果
后台级操作中止	IEEE_DONE_OK	真
接收到 CMD_STOP 或 CMD_IEEE_STOP_FG	IEEE_DONE_STOPPED	否
接收到 CMD_ABORT 或 CMD_IEEEE_ABORT_FG	IEEE_DONE_ABORT	ABORT

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.5.5 即时命令

25.5.5.1 修改 CCA 参数命令

CMD_IEEE_MOD_CCA 命令采用表 25-67 中定义的命令结构。

只能在 RX 或者能量检测扫描操作正在运行时发送 CMD_IEEE_MOD_CCA。接收时，无线电 CPU 将正在运行的进程的 ccaRssiThr 和 ccaOpt 的值分别修改为 newCcaRssiThr 和 newCcaOpt 给出的值。无线电 CPU 会更新命令结构。新设置用于将来的 CCA 请求。

如果发出该命令时没有活动或暂停的后台级操作，则无线电 CPU 在 CMDSTA 中返回结果 ContextError。如果输入的任何参数是非法的，无线电 CPU 将在 CMDSTA 中返回结果 ParError。否则，无线电 CPU 返回 DONE。

25.5.5.2 修改帧过滤参数命令

CMD_IEEE-MOD_FILT 命令采用表 25-68 中定义的命令结构。

CMD_IEEE_MOD_FILT 只能在 RX 操作运行期间发送。收到该命令后，无线电 CPU 会将正在运行的进程的 frameFiltOpt 和 frameTypes 值分别修改为 newFrameFiltOpt 和 newFrameTypes 给定的值。无线电 CPU 更新命令结构。

帧过滤选项的新值将从下一次启动帧过滤时开始使用。如果 autoAckEn 或 slottedAckEn 被更改，更改将在下一次数据包接收结束时生效。

如果该命令在无活动或挂起的后台级 RX 操作时发出，无线电 CPU 会在 CMDSTA 中返回结果 ContextError。如果输入的任何参数非法，无线电 CPU 会在 CMDSTA 中返回结果 ParError。否则，无线电 CPU 返回 DONE。

25.5.5.3 启用或禁用源匹配条目命令

CMD_IEEE_MOD_SRC_MATCH 命令采用表 25-68 中定义的命令结构。CMD_IEEE_MOD_SRC_MATCH 只能在 RX 操作运行期间发送。收到该命令后，无线电 CPU 根据命令结构中指示的信息启用或禁用对应的源匹配条目。如果 options.entryType 为 0，则该条目是由 pExtEntryList 指向的结构中的扩展地址条目；如果 options.entryType 为 1，则该条目是由 pShortEntryList 指向的结构中的短地址条目。条目的索引在 entryNo 中指示。如果 options.bEnable 为 0，则禁用该条目；如果为 1，则启用该条目。相应的源挂起位设置为 options.srcMatch 的值。新的使能值从下一次执行源匹配时开始生效。系统 CPU 可以修改已禁用条目的地址，但不能修改已启用条目的地址。如果该命令在无活动或挂起的后台级 RX 操作时发出，无线电 CPU 会在 CMDSTA 中返回结果 ContextError。如果输入的任何参数非法，例如指向不存在的条目，无线电 CPU 将在 CMDSTA 中返回结果 ParError。否则，无线电 CPU 返回 DONE。

25.5.5.4 中止前台级操作命令

CMD_IEEE_ABORT_FG 是一条即时命令，不带任何参数，因此可用作直接命令。

CMD_IEEE_ABORT_FG 命令会中止前台级操作，而后台级操作继续运行。有关更多详细信息，请参阅表 25-59 中对前台级操作的说明。

如果没有运行前台级无线电操作命令，则不执行任何操作。在所有情况下，CMDSTA 中报告的结果均为 DONE。如果正在运行前台级无线电操作命令，则 CMDSTA 可能会在无线电操作结束之前更新。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.5.5.5 停止前台级操作命令

CMD_IEEE_STOP_FG 是一条即时命令，不带任何参数，因此可用作直接命令。

CMD_IEEE_STOP_FG 命令会使前台级操作平稳停止，而后台级操作继续运行。有关更多详细信息，请参阅表 25-59 中对前台级操作的说明。

如果没有运行前台级无线电操作命令，则不执行任何操作。在所有情况下，CMDSTA 中指示的结果均为 DONE。如果正在运行前台级无线电操作命令，则 CMDSTA 可能会在无线电操作结束之前更新。

25.5.5.6 请求 CCA 及 RSSI 信息命令

CMD_IEEE-CCA_REQ 命令采用表 25-70 中定义的命令结构。

只能在 RX 或者能量检测扫描操作正在运行时发送 CMD_IEEE-CCA_REQ。接收时，无线电 CPU 会将以下数字写回命令结构中：

- currentRssi 设置为当前可以从解调器获取的 RSSI 编号
- maxRssi 设置为观察到的最大 RSSI，因为后台操作已启动
- 根据当前的 CCA 选项，ccaState 设置为 CCA 状态，请参阅节 25.5.4.1.5
- 根据节 25.5.4.1.5，将 ccaEnergy 设置为能量检测 CCA 状态
- 根据节 25.5.4.1.5，将 ccaCorr 设置为基于相关器的载波感应 CCA 状态
- 根据节 25.5.4.1.5，将 ccaSync 设置为基于发现的同步的载波感应 CCA 状态

如果发送请求时找不到有效的 RSSI，则返回的 currentRssi 和 maxRssi 通过使用特殊值 (0x80) 来指示这一点。

如果发出该命令时没有活动或暂停的后台级 RX 操作，则无线电 CPU 在 CMDSTA 中返回结果 ContextError。否则，无线电 CPU 返回 DONE。

25.6 低功耗 Bluetooth®

本节介绍了特定于低功耗 Bluetooth 的命令结构、数据处理、无线电操作命令和即时命令。

25.6.1 低功耗 Bluetooth® 命令

表 25-89 中定义了传统模式的低功耗蓝牙专用无线电操作命令。应当使用这些命令来运行蓝牙版本 4.0、4.1、4.2 以及蓝牙 5.0 的传统特性。

表 25-89. 传统低功耗 Bluetooth® 无线电操作命令

ID	命令名称	说明
0x1801	CMD_BLE_SLAVE	启动从器件操作。
0x1802	CMD_BLE_MASTER	启动主器件操作。
0x1803	CMD_BLE_ADV	启动可连接的非定向广播器操作。
0x1804	CMD_BLE_ADV_DIR	启动可连接的定向广播器操作。
0x1805	CMD_BLE_ADV_NC	启动非可连接的广播器操作。
0x1806	CMD_BLE_ADV_SCAN	启动可扫描的非定向广播器操作。
0x1807	CMD_BLE_SCANNER	开始扫描仪操作。
0x1808	CMD_BLE_INITIATOR	开始启动器操作。
0x1809	CMD_BLE_GENERIC_RX	接收通用数据包（用于 PHY 测试或者数据包嗅探）。
0x180A	CMD_BLE_TX_TEST	发送 PHY 测试数据包。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-90 定义了蓝牙 5 模式的低功耗蓝牙专用无线电操作命令。

表 25-90. 低功耗 Bluetooth® 5 无线电操作命令

ID	命令名称	说明
0x1820	CMD_BLE5_RADIO_SETUP	将无线电设置为蓝牙 5 模式，并能够在 PHY 之间切换。
0x1821	CMD_BLE5_SLAVE	启动从器件操作。
0x1822	CMD_BLE5_MASTER	启动主器件操作。
0x1823	CMD_BLE5_ADV_EXT	在主通道上启动扩展广播器操作。
0x1824	CMD_BLE5_ADV_AUX	在辅助通道上启动扩展广播器操作。
0x1827	CMD_BLE5_SCANNER	开始扫描仪操作。
0x1828	CMD_BLE5_INITIATOR	开始启动器操作。
0x1829	CMD_BLE5_GENERIC_RX	接收通用数据包（用于 PHY 测试或者数据包嗅探）。
0x182A	CMD_BLE5_TX_TEST	发送 PHY 测试数据包。

表 25-91 定义低功耗蓝牙专用即时命令。

表 25-91. 低功耗 Bluetooth® 即时命令

ID	命令名称	说明
0x1001	CMD_BLE_ADV_PAYLOAD	修改广播器操作中使用的有效载荷。

25.6.1.1 命令数据定义

本节定义了在描述系统 CPU 和无线电 CPU 之间的通信所使用的数据结构时采用的数据类型。这些数据结构以表格形式列出。字节索引是相对于指向该结构的指针的偏移量。多字节字段为小端字节序，且需要半字或字对齐。对于位编号，0 表示最低有效位。R/W 列的含义如下：

- R：系统 CPU 可以读回结果；无线电 CPU 不会读取该字段。
- W: 系统 CPU 应写入一个值，无线电 CPU 将读取该值且不会修改它。
- R/W：系统 CPU 应写入初始值，无线电 CPU 可能会修改该初始值。

25.6.1.1.1 低功耗 Bluetooth® 命令结构

表 25-92. 传统低功耗 Bluetooth® 无线电操作命令结构 (1)

字节索引	字段名称	位	位字段名称	类型	说明
0-1	commandNo			W	命令 ID 编号
2-3	状态			R/W	指示命令状态的整数。此值在运行期间通过无线电 CPU 更新，可随时由系统 CPU 读取。
4-7	pNextOp			W	指向此操作完成后要运行的下一个操作的指针
8-11	startTime			W	绝对或相对开始时间（取决于 startTrigger 的值）
12	startTrigger				启动操作的触发器的标识。
13	条件	0-3	规则	W	运行下一个命令的条件：继续操作的规则。
		4-7	nSkip	W	如果规则涉及跳过，则跳过次数 + 1。 0：相同 1：下一个 2：跳过下一个 ...

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-92. 传统低功耗 Bluetooth® 无线电操作命令结构 (1) （续）

字节索引	字段名称	位	位字段名称	类型	说明
14	通道			W	要使用的通道 0 – 39：低功耗蓝牙广播/数据通道编号 60 – 207：自定义频率；（2300 + 通道）MHz 255：使用现有频率 其他：保留
15	白化	0-6	init	W	如果 bOverride = 1 或使用了自定义频率： 0：请勿使用白化 其他值：7 位 LFSR 白化器初始化
		7	bOverride	W	0：对低功耗蓝牙广播/数据通道使用默认白化 1：使用 init 的值覆盖白化初始化
16-19	pParams			W	指向命令特定参数结构的指针
20-23	pOutput			W	指向命令特定输出结构的指针

(1) 此命令结构应用于表 25-89 中列出的用于支持传统低功耗蓝牙的所有无线电操作命令。

表 25-93. 低功耗 Bluetooth® 5 无线电操作命令结构 (1)

字节索引	字段名称	位	位字段名称	类型	说明
0-1	commandNo			W	命令 ID 号 0x1821
2-3	状态			R/W	指示命令状态的整数。此值在运行期间通过无线电 CPU 更新，可随时由系统 CPU 读取。
4-7	pNextOp			W	指向此操作完成后要运行的下一个操作的指针
8-11	startTime			W	绝对或相对开始时间（取决于 startTrigger 的值）
12	startTrigger				启动操作的触发器的标识
13	条件	0-3	规则	W	运行下一个命令的条件：继续操作的规则
		4-7	nSkip	W	如果规则涉及跳过，则跳过次数 + 1。 0：相同 1：下一个 2：跳过下一个 ...
14	通道			W	要使用的通道 0 – 39：低功耗蓝牙广播/数据通道编号 60 – 207：自定义频率；（2300 + 通道）MHz 255：使用现有频率 其他：保留
15	白化	0-6	init	W	如果 bOverride = 1 或使用了自定义频率： 0：请勿使用白化 其他值：7 位 LFSR 白化器初始化
		7	bOverride	W	0：对低功耗蓝牙广播/数据通道使用默认白化 1：使用 init 的值覆盖白化初始化
16	phyMode	0-1	mainMode	W	要使用的 PHY： 0：1Mbps 1：2Mbps 2：已编码 3：保留
		2-7	编码	W	选择了编码的 PHY 时要用于 TX 的编码。请参阅 节 25.8.3
17	rangeDelay			W	在 T_IFS 之后添加至侦听时间的 RAT 节拍数
18-19	txPower			W	要使用的发射功率（覆盖无线电设置中指定的功率）。 0x0000：使用默认 TX 功率。
20-23	pParams			W	指向命令特定参数结构的指针

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-93. 低功耗 Bluetooth® 5 无线电操作命令结构 (1) （续）

字节索引	字段名称	位	位字段名称	类型	说明
24-27	pOutput			W	指向命令特定输出结构的指针

(1) 此命令结构应用于支持低功耗蓝牙 5 的所有无线电操作命令（表 25-90 中列出），CMD_BLE5_RADIO_SETUP 除外。

表 25-94. 低功耗 Bluetooth® 5 无线电设置命令结构 (1)

字节索引	字段名称	位	位字段名称	类型	说明
0-1	commandNo			W	命令 ID 号 0x1820
2-3	状态			R/W	指示命令状态的整数。此值在运行期间通过无线电 CPU 更新，可随时由系统 CPU 读取。
4-7	pNextOp			W	指向此操作完成后要运行的下一个操作的指针
8-11	startTime			W	绝对或相对开始时间（取决于 startTrigger 的值）
12	startTrigger				启动操作的触发器的标识
13	条件	0-3	规则	W	运行下一个命令的条件：继续操作的规则
		4-7	nSkip	W	如果规则涉及跳过，则跳过次数 + 1。 0：相同 1：下一个 2：跳过下一个 ...
14	defaultPhy	0-1	mainMode	W	要用于非 BLE 命令的 PHY： 0：1Mbps 1：2Mbps 2：已编码 3：保留
		2	编码	W	为非 BLE 命令选择了编码的 PHY 时要用于 TX 的编码 0：S = 8 (125kbps) 1：S = 2 (500kbps)
		3-7			保留
15					保留
16-17	config				配置选项
18-19	txPower				默认发射功率
20-23	pRegOverrideCommon			W	指向在通用初始化期间要覆盖的硬件及配置寄存器列表的指针。如果为 NULL，不使用覆盖。
24-27	pRegOverride1Mbps			W	选择 1Mbps PHY 模式时要覆盖的硬件及配置寄存器列表的指针。如果为 NULL，不使用覆盖。
28-31	pRegOverride2Mbps			W	指向在选择 2Mbps PHY 模式时要覆盖的硬件及配置寄存器列表的指针。如果为 NULL，不使用覆盖。
32-35	pRegOverrideCoded			W	指向在选择编码的 PHY 模式时要覆盖的硬件及配置寄存器列表的指针。如果为 NULL，不使用覆盖。

(1) 此命令结构应当用于 CMD_BLE5_RADIO_SETUP。

表 25-95. 更新广播有效载荷命令 (1)

字节索引	字段名称	类型	说明
0-1	commandNo	W	命令 ID 号 0x1001
2	payloadType	W	0：广播数据 1：扫描响应数据
3	newLen	W	新有效载荷的长度
4-7	pNewData	W	指向包含新数据的缓冲器的指针

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-95. 更新广播有效载荷命令 (1) （续）

字节索引	字段名称	类型	说明
8-11	pParams	W	指向要更新的参数结构的指针

(1) 此命令结构应当用于 CMD_BLE_ADV_PAYLOAD。

25.6.1.2 参数结构

表 25-96. 传统从器件命令 (1)

字节索引	字段名称	类型	说明
0-3	pRxQ	W	指向接收队列的指针
4-7	pTxQ	W	指向发送队列的指针
8	rxConfig	W	接收队列条目的配置位（请参阅表 25-116）
9	seqStat	R/W	序列号状态（请参阅表 25-117）
10	maxNack	W	操作结束前接收到的最大 NACK 数。 0：无限制
11	maxPkt	W	操作结束前传输的最大数据包数。 0：无限制
12-15	accessAddress	W	连接上使用的访问地址
16	crcInit0	W	连接上使用的 CRC 初始化值 – 最低有效字节
17	crcInit1	W	连接上使用的 CRC 初始化值 – 中间字节
18	crcInit2	W	连接上使用的 CRC 初始化值 – 最高有效字节
19	timeoutTrigger	W	定义第一个接收操作超时的触发
20-23	timeoutTime	W	用于定义第一个接收操作超时的与 timeoutTrigger 一起使用的时间
24-26			保留
27	endTrigger	W	触发器，使器件在允许的情况下尽快结束连接事件
28-31	endTime	W	与 endTrigger 一起使用的时间，使器件在允许的情况下尽快结束连接事件

(1) 此参数结构用于旧从属命令，CMD_BLE_SLAVE。

表 25-97. 旧主器件命令 (1)

字节索引	字段名称	类型	说明
0-3	pRxQ	W	指向接收队列的指针
4-7	pTxQ	W	指向发送队列的指针
8	rxConfig	W	接收队列条目的配置位（请参阅表 25-116）
9	seqStat	R/W	序列号状态（请参阅表 25-117）
10	maxNack	W	操作结束前接收到的最大 NACK 数。 0：无限制
11	maxPkt	W	操作结束前传输的最大数据包数。 0：无限制
12-15	accessAddress	W	连接上使用的访问地址
16	crcInit0	W	连接上使用的 CRC 初始化值 – 最低有效字节
17	crcInit1	W	连接上使用的 CRC 初始化值 – 中间字节
18	crcInit2	W	连接上使用的 CRC 初始化值 – 最高有效字节
19	endTrigger	W	触发器，使器件在允许的情况下尽快结束连接事件
20-23	endTime	W	与 endTrigger 一起使用的时间，使器件在允许的情况下尽快结束连接事件

(1) 此参数结构用于旧主器件命令，CMD_BLE_MASTER。

Copyright © 2026 Texas Instruments Incorporated

表 25-98. 旧广播器命令 (1)

字节索引	字段名称	位	位字段名称	类型	说明
0-3	pRxQ			W	指向接收队列的指针
4	rxConfig			W	接收队列条目的配置位（请参阅表 25-116）
5	advConfig	0-1	advFilterPolicy	W	广播器滤波器策略 0：处理来自所有器件的扫描和连接请求 1：处理来自所有设备的连接请求，仅扫描来自白名单中设备的请求 2：处理来自所有设备的扫描请求，仅连接来自白名单中设备的请求 3：仅处理来自白名单中设备的扫描及连接请求
		2	deviceAddrType	W	器件地址的类型： 公共 (0) 或随机 (1)
		3	peerAddrType	W	定向广告器：对等地址的类型： 公共 (0) 或随机 (1)
		4	bStrictLenFilter	W	0：接受任何具备有效广播数据包长度的数据包 1：对于给定的数据包类型，丢弃长度非法的消息
		5	chSel	W	0：不报告通道选择算法 2 的支持 1：报告通道选择算法 2 的支持
		6			保留
		7	rpaMode	W	可解析私有地址模式 0：正常运行 1：对接收的 RPA 使用白名单，而不考虑滤波器策略
6	advLen			W	广播器数据的大小
7	scanRspLen			W	扫描响应数据的大小
8-11	pAdvData			W	指向包含 ADV*_IND 数据的缓冲器的指针
12-15	pScanRspData			W	指向包含 SCAN_RSP 数据的缓冲器的指针
16-19	pDeviceAddress			W	指向用于该器件的器件地址的指针（最低有效位设置为 0）。如果最低有效位为 1，则 advConfig.deviceAddrType 给出的地址类型被反转。
20-23	pWhiteList			W	指向白名单或对等地址（定向广播器）的指针（最低有效位设置为 0）。如果最低有效位为 1，则 advConfig.peerAddrType 给出的地址类型被反转。
24-26					保留
27	endTrigger			W	触发器，使器件在允许的情况下尽快结束广播器事件
28-31	endTime			W	与 endTrigger 一起使用的时间，使器件在允许的情况下尽快结束广播器事件

(1) 此参数结构用于所有旧广播器命令 CMD_BLE_ADV、CMD_BLE_ADV_DIR、CMD_BLE_ADV_NC 和 CMD_BLE_ADV_SCAN。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-99. 传统扫描仪命令 (1)

字节索引	字段名称	位	位字段名称	类型	说明
0-3	pRxQ			W	指向接收队列的指针
4	rxConfig			W	接收队列条目的配置位
5	scanConfig	0	scanFilterPolicy	W	扫描与广播器地址相关的滤波器策略 0：接受所有广播包。 1：仅接受来自广播器地址位于白名单中的设备的广播包。
		1	bActiveScan	W	0：被动扫描 1：活动扫描
		2	deviceAddrType	W	器件地址的类型： 公共 (0) 或随机 (1)
		3	rpaFilterPolicy	W	ADV_DIRECT_IND 消息的 TargetA 的滤波器策略 0：仅接受与自己地址匹配的 TargetA。 1：也接受所有可解析的私有地址。
		4	bStrictLenFilter	W	0：接受任何具备有效广播数据包长度的数据包 1：对于给定的数据包类型，丢弃长度非法的消息
		5	bAutoWlIgnore	W	0：请勿在无线电 CPU 的白名单中设置忽略位 1：在白名单中自动设置忽略位
		6	bEndOnRpt	W	0：在每次报告 ADV*_IND 或发送 SCAN_RSP 后继续扫描仪操作 1：在每次报告 ADV*_IND 并且可能是 SCAN_RSP 之后结束扫描仪操作
		7	rpaMode	W	可解析私有地址模式 0：正常运行 1：对接收的 RPA 使用白名单，而不考虑滤波器策略
6-7	randomState			R/W	退避过程中使用的假随机数生成的状态
8-9	backoffCount			R/W	回退过程中使用的参数 backoffCount（请参阅相关文档中的蓝牙规范文档）。
10	backoffPar	0-3	logUpperLimit	R/W	扫描仪退避过程中使用的参数 upperLimit 的二进制对数
		4	bLastSucceeded	R/W	如果最后一个 SCAN_RSP 已成功接收且 upperLimit 未更改，则为 1
		5	bLastFailed	R/W	如果接收最后一个 SCAN_RSP 失败且 upperLimit 未更改，则为 1
		6-7			保留
11	scanReqLen			W	扫描请求数据的大小
12-15	pScanReqData			W	指向包含 SCAN_REQ 数据的缓冲器的指针
16-19	pDeviceAddress			W	指向用于该器件的器件地址的指针（最低有效位设置为 0）。如果最低有效位为 1，则 scanConfig.deviceAddrType 给出的地址类型被反转。
20-23	pWhiteList			W	指向白名单的指针
24-25					保留
26	timeoutTrigger			W	触发器，使器件在允许的情况下停止接收
27	endTrigger			W	触发器，使器件在允许的情况下停止接收
28-31	timeoutTime			W	与 timeoutTrigger 一起使用的时间，使设备在允许的情况下尽快停止接收，以 BLE_DONE_RXTIMEOUT 结束
32-35	endTime			W	与 endTrigger 一起使用的时间，使设备在允许的情况下尽快停止接收，以 BLE_DONE_ENDED 结束

(1) 此参数结构用于旧扫描仪命令，CMD_BLE_SCANNER。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-100. 传统启动器命令 (1)

字节索引	字段名称	位	位字段名称	类型	说明
0-3	pRxQ			W	指向接收队列的指针
4	rxConfig			W	接收队列条目的配置位（请参阅表 25-116）
5	initConfig	0	bUseWhiteList	W	启动器滤波器策略 0：使用特定对等地址 1：使用白名单
		1	bDynamicWinOffset		0：无动态 WinOffset 插入 1：使用动态 WinOffset 插入
		2	deviceAddrType	W	器件地址的类型： 公共 (0) 或随机 (1)
		3	peerAddrType	W	对等地址的类型： 公共 (0) 或随机 (1)
		4	bStrictLenFilter	W	0：接受任何具备有效广播数据包长度的数据包 1：对于给定的数据包类型，丢弃长度非法的消息
		5	chSel	W	0：不报告通道选择算法 2 的支持 1：报告通道选择算法 2 的支持
		6			保留
		7			保留
6					保留
7	connectReqLen			W	连接请求数据的大小
8-11	pConnectReqData			W	指向缓冲器的指针，该缓冲器包含要进入 CONNECT_IND (CONNECT_REQ) 的 LLData
12-15	pDeviceAddress			W	指向用于该器件的器件地址的指针（最低有效位设置为 0）。如果最低有效位为 1，则 initConfig.deviceAddrType 给出的地址类型被反转。
16-19	pWhiteList			W	指向白名单或对等地址的指针（最低有效位设置为 0）。如果最低有效位为 1，则 initConfig.peerAddrType 给出的地址类型被反转。
20-23	connectTime			R/W	指示第一个连接事件的第一个可能开始时间的计时器值。如果已建立连接，则设置为计算值；如果未建立连接、则设置为下一个可能的连接时间（请参阅表 25-112）。
24-25					保留
26	timeoutTrigger			W	触发器，使器件在允许的情况下停止接收
27	endTrigger			W	触发器，使器件在允许的情况下停止接收
28-31	timeoutTime			W	与 timeoutTrigger 一起使用的时间，使设备在允许的情况下尽快停止接收，以 BLE_DONE_RXTIMEOUT 结束
32-35	endTime			W	与 endTrigger 一起使用的时间，使设备在允许的情况下尽快停止接收，以 BLE_DONE_ENDED 结束

(1) 此参数结构用于传统启动器命令，CMD_BLE_INITIATOR。

表 25-101. 通用 RX 命令 (1)

字节索引	字段名称	类型	说明
0-3	pRxQ	W	指向接收队列的指针。可能为 NULL；如果为 NULL，则不存储接收到的数据包。
4	rxConfig	W	接收队列条目的配置位（请参阅表 25-116）
5	bRepeat	W	0：接收到数据包后结束操作 1：接收数据包后重新启动接收器
6-7			保留
8-11	accessAddress	W	连接上使用的访问地址

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-101. 通用 RX 命令 (1) （续）

字节索引	字段名称	类型	说明
12	crcInit0	W	连接上使用的 CRC 初始化值 – 最低有效字节
13	crcInit1	W	连接上使用的 CRC 初始化值 – 中间字节
14	crcInit2	W	连接上使用的 CRC 初始化值 – 最高有效字节
15	endTrigger	W	导致器件结束 RX 操作的触发器
16-19	endTime	W	与 endTrigger 一起使用的时间，使器件结束 RX 操作。

(1) 此参数结构同时用于传统和低功耗蓝牙 5 通用接收器命令，CMD_BLE_GENERIC_RX 和 CMD_BLE5_GENERIC_RX。

表 25-102. TX 测试命令 (1)

字节索引	字段名称	位	位字段名称	类型	说明
0-1	numPackets			W	要传输的数据包数 0：传输的数据包数量不受限
2	payloadLength			W	每个数据包中的有效载荷字节数。
3	packetType			W	要使用的数据包类型，根据蓝牙 5.0 规范进行编码（请参阅相关文档）。
4-7	period			W	每个数据包开始之间的无线电计时器周期数
8	config	0	bOverrideDefault	W	0：使用默认数据包编码 1：覆盖数据包内容
		1	bUsePrbs9	W	如果 bOverride 为 1： 0：无数据包的 PRBS15 编码 1：使用数据包的 PRBS9 编码
		2	bUsePrbs15	W	如果 bOverride 为 1： 0：无数据包的 PRBS15 编码 1：使用数据包的 PRBS15 编码
		3-7			保留
9	byteVal			W	如果 config.bOverride 为 1，则为要发送的每个字节的值
10					保留
11	endTrigger			W	导致器件结束测试 TX 操作的触发器
12-15	endTime			W	与 endTrigger 一起使用的时间，使器件结束测试 TX 操作

(1) 此参数结构同时用于传统和低功耗蓝牙 5 发送器测试命令，CMD_BLE_TX_TEST 和 CMD_BLE5_TX_TEST。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-103. Bluetooth®5 从器件命令 (1)

字节索引	字段名称	类型	说明
0-3	pRxQ	W	指向接收队列的指针
4-7	pTxQ	W	指向发送队列的指针
8	rxConfig	W	接收队列条目的配置位（请参阅表 25-116）
9	seqStat	RW	序列号状态（请参阅表 25-117）
10	maxNack	W	操作结束前接收到的最大 NACK 数。 0：无限制
11	maxPkt	W	操作结束前传输的最大数据包数。 0：无限制
12-15	accessAddress	W	连接上使用的访问地址
16	crcInit0	W	连接上使用的 CRC 初始化值 – 最低有效字节
17	crcInit1	W	连接上使用的 CRC 初始化值 – 中间字节
18	crcInit2	W	连接上使用的 CRC 初始化值 – 最高有效字节
19	timeoutTrigger	W	定义第一个接收操作超时的触发
20-23	timeoutTime	W	用于定义第一个接收操作超时的与 timeoutTrigger 一起使用的时间
24	maxRxPktLen	W	当前允许在连接上接收的数据包的最大数据包长度
25	maxLenLowRate	W	发送时允许使用 S = 8 (125kbps) 的最大数据包长度。 0：无限制
26			保留
27	endTrigger	W	触发器，使器件在允许的情况下尽快结束连接事件
28-31	endTime	W	与 endTrigger 一起使用的时间，使器件在允许的情况下尽快结束连接事件

(1) 此参数结构用于蓝牙 5 从器件，CMD_BLE5_SLAVE。

表 25-104. Bluetooth® 5 主命令 (1)

字节索引	字段名称	类型	说明
0-3	pRxQ	W	指向接收队列的指针
4-7	pTxQ	W	指向发送队列的指针
8	rxConfig	W	接收队列条目的配置位（请参阅表 25-116）
9	seqStat	RW	序列号状态（请参阅表 25-117）
10	maxNack	W	操作结束前接收到的最大 NACK 数。 0：无限制
11	maxPkt	W	操作结束前传输的最大数据包数。 0：无限制
12-15	accessAddress	W	连接上使用的访问地址
16	crcInit0	W	连接上使用的 CRC 初始化值 – 最低有效字节
17	crcInit1	W	连接上使用的 CRC 初始化值 – 中间字节
18	crcInit2	W	连接上使用的 CRC 初始化值 – 最高有效字节
19	endTrigger	W	触发器，使器件在允许的情况下尽快结束连接事件
20-23	endTime	W	与 endTrigger 一起使用的时间，使器件在允许的情况下尽快结束连接事件
24	maxRxPktLen	W	当前允许在连接上接收的数据包的最大数据包长度
25	maxLenLowRate	W	发送时允许使用 S = 8 (125kbps) 的最大数据包长度。 0：无限制

(1) 此参数结构用于蓝牙 5 主器件，CMD_BLE5_MASTER。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-105. 扩展的广播器命令 (1)

字节索引	字段名称	位	位字段名称	类型	说明
0	advConfig	0-1			保留
		2	deviceAddrType	W	器件地址的类型 公共 (0) 或随机 (1)
		3-7			保留
1-2					保留
3	auxPtrTargetType			W	表示 auxPtrTargetTime 基准的数字。使用与触发类型相同的值，但仅允许 TRIG_ABSTIME 和 TRIG_REL_*。
4-7	auxPtrTargetTime			W	auxPtr 指向的数据包的开始时间
8-11	pAdvPkt			W	指向 ADV_EXT_IND 数据包的扩展广播数据包的指针
12-15	pDeviceAddress			W	指向用于该器件的器件地址的指针（最低有效位设置为 0）。如果最低有效位为 1，则 advConfig.deviceAddrType 给出的地址类型被反转。

(1) 此参数结构用于低功耗蓝牙 5 扩展广播器命令，CMD_BLE5_ADV_EXT。有关 pAdvPkt 指向的结构的定义，请参阅表 25-123。

表 25-106. 辅助通道广播器命令 (1)

字节索引	字段名称	位	位字段名称	类型	说明
0-3	pRxQ			W	指向接收队列的指针
4	rxConfig			W	接收队列条目的配置位（请参阅表 25-116）
5	advConfig	0-1	advFilterPolicy	W	广播器滤波器策略 0：处理来自所有器件的扫描和连接请求。 1：处理来自所有设备的连接请求，仅扫描来自白名单中设备的请求。 2：处理来自所有设备的扫描请求，仅连接来自白名单中设备的请求。 3：仅处理来自白名单中设备的扫描及连接请求。
		2	deviceAddrType	W	器件地址的类型 公共 (0) 或随机 (1)
		3	targetAddrType	W	定向辅助广播器：目标地址的类型 公共 (0) 或随机 (1)
		4	bStrictLenFilter	W	0：接受任何具备有效广播数据包长度的数据包。 1：对于给定的数据包类型，丢弃长度非法的消息。
		5	bDirected	W	0：广告商未定向：pWhiteList 指向白名单 1：广告商被定向：pWhiteList 指向单个器件地址。
		6			保留
		7	rpaMode	W	可解析私有地址模式 0：正常运行 1：对接收的 RPA 使用白名单，而不考虑滤波器策略。
6					保留
7	auxPtrTargetType			W	表示 auxPtrTargetTime 基准的数字。使用与触发类型相同的值，但仅允许 TRIG_ABSTIME 和 TRIG_REL_*。
8-11	auxPtrTargetTime			W	auxPtr 指向的数据包的开始时间
12-15	pAdvPkt			W	指向 ADV_AUX_IND 数据包的扩展广播数据包的指针
16-19	pRspPkt			W	指向 AUX_SCAN_RSP 或 AUX_CONNECT_RSP 数据包的扩展广播数据包的指针（如果不适用，可能为 NULL）
20-23	pDeviceAddress			W	指向用于该器件的器件地址的指针（最低有效位设置为 0）。如果最低有效位为 1，则 advConfig.deviceAddrType 给出的地址类型被反转。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-106. 辅助通道广播器命令 (1) （续）

字节索引	字段名称	位	位字段名称	类型	说明
24-27	pWhiteList			W	指向白名单或对等地址（定向广播器）的指针（最低有效位设置为 0）。如果最低有效位为 1，则 advConfig.peerAddrType 给出的地址类型被反转。

(1) 此参数结构用于低功耗蓝牙 5 辅助通道广播器命令，CMD_BLE5_ADV_AUX。有关 pAdvPkt 和 pRspPkt 指向的结构的定义，请参阅表 25-123。

表 25-107. Bluetooth®5 扫描仪命令 (1)

字节索引	字段名称	位	位字段名称	类型	说明
0-3	pRxQ			W	指向接收队列的指针
4	rxConfig			W	接收队列条目的配置位（请参阅表 25-116）
5	scanConfig	0	scanFilterPolicy	W	扫描与广播器地址相关的滤波器策略 0：接受所有广播包。 1：仅接受来自广播器地址位于白名单中的设备的广播包。
		1	bActiveScan	W	0：被动扫描 1：活动扫描
		2	deviceAddrType	W	器件地址的类型 公共 (0) 或随机 (1)
		3	rpaFilterPolicy	W	ADV_DIRECT_IND 消息的 initA 的滤波器策略 0：仅接受与自身地址匹配的 initA。 1：也接受所有可解析的私有地址。
		4	bStrictLenFilter	W	0：接受任何具备有效广播数据包长度的数据包。 1：对于给定的数据包类型，丢弃长度非法的消息。
		5	bAutoWlIgnore	W	0：请勿为旧数据包设置来自无线电 CPU 的白名单中的忽略位。 1：为旧数据包自动设置白名单中的忽略位。
		6	bEndOnRpt	W	0：每次报告 ADV*_IND 或发送 SCAN_RSP 后，继续扫描仪操作。 1：在每次报告 ADV*_IND 并且可能是 SCAN_RSP 之后结束扫描仪操作。
		7	rpaMode	W	可解析私有地址模式 0：正常运行 1：对接收的 RPA 使用白名单，而不考虑滤波器策略。
6-7	randomState			R/W	退避过程中使用的假随机数生成的状态
8-9	backoffCount			R/W	回退过程中使用的参数 backoffCount（请参阅相关文档中所列的蓝牙规范文档）。
10	backoffPar	0-3	logUpperLimit	R/W	扫描仪退避过程中使用的参数 upperLimit 的二进制对数
		4	bLastSucceeded	R/W	如果最后一个 SCAN_RSP 已成功接收且 upperLimit 未更改，则为 1
		5	bLastFailed	R/W	如果接收最后一个 SCAN_RSP 失败且 upperLimit 未更改，则为 1
		6-7			保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-107. Bluetooth®5 扫描仪命令 (1) （续）

字节索引	字段名称	位	位字段名称	类型	说明
11	extFilterConfig	0	bCheckAdi	W	0：请勿执行 ADI 滤波。 1：对存在 ADI 的数据包执行 ADI 滤波。
		1	bAutoAdiUpdate	W	0：请勿更新无线电 CPU 中的 ADI 条目。 1：使用 AdvDataIndex 自动更新已接收数据包的 ADI 条目。
		2	bApplyDuplicateFiltering	W	0：不要对扩展的广播器数据包应用基于器件地址的重复滤波。 1：针对没有 ADI 字段的扩展广播器数据包应用基于器件地址的重复滤波。
		3	bAutoWlIgnore	W	0：请勿为扩展广播数据包设置来自无线电 CPU 的白名单中的忽略位。 1：自动为扩展广播数据包设置白名单中的忽略位。
		4-7			保留
12-15					保留
16-19	pDeviceAddress			W	指向用于该器件的器件地址的指针（最低有效位设置为 0）。如果最低有效位为 1，则 scanConfig.deviceAddrType 给出的地址类型被反转。
20-23	pWhiteList			W	指向白名单的指针
24-27	pAdiList			W	指向 advDataInfo 列表的指针
28-29	maxWaitTimeForAuxCh			W	在命令内切换至次要扫描的最长等待时间。如果事件开始的时间大于此值，命令将以 BLE_DONE_AUX 结束。如果尺寸更小，无线电将自动切换到正确的信道和 PHY。
30	timeoutTrigger			W	触发器，使器件在允许的情况下停止接收
31	endTrigger			W	触发器，使器件在允许的情况下停止接收
32-35	timeoutTime			W	与 timeoutTrigger 一起使用的时间，使设备在允许的情况下尽快停止接收，以 BLE_DONE_RXTIMEOUT 结束
36-39	endTime			W	与 endTrigger 一起使用的时间，使设备在允许的情况下尽快停止接收，以 BLE_DONE_ENDED 结束
40-43	rxStartTime			R	启动 RX 以接收数据包所需的时间
44-45	rxListenTime			R	接收数据包进行侦听所需的时间。 0：无 AUX 数据包
46	channelNo			R	用于辅助广播的通道编号
47	phyMode			R	要在辅助通道上使用的 PHY： 0：1Mbps 1：2Mbps 2：编码 其他：保留

(1) 此参数结构用于蓝牙 5 扫描仪命令，CMD_BLE5_SCANNER。

表 25-108. Bluetooth®5 启动器命令 (1)

字节索引	字段名称	位	位字段名称	类型	说明
0-3	pRxQ			W	指向接收队列的指针
4	rxConfig			W	接收队列条目的配置位（请参阅表 25-116）

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-108. Bluetooth®5 启动器命令 (1) （续）

字节索引	字段名称	位	位字段名称	类型	说明
5	initConfig	0	bUseWhiteList	W	启动器滤波器策略 0：使用特定对等地址。 1：使用白名单。
		1	bDynamicWinOffset		1：使用动态 WinOffset 插入。
		2	deviceAddrType	W	器件地址的类型 公共 (0) 或随机 (1)
		3	peerAddrType	W	对等地址的类型 公共 (0) 或随机 (1)
		4	bStrictLenFilter	W	0：接受任何具备有效广播数据包长度的数据包。 1：对于给定的数据包类型，丢弃长度非法的消息。
		5	chSel	W	0：不报告 CONNECT_IND 中通道选择算法 2 的支持。 1：报告 CONNECT_IND 中通道选择算法 2 的支持。
		6			保留
		7			保留
6-7	randomState			R/W	退避过程中使用的假随机数生成的状态
8-9	backoffCount			R/W	回退过程中使用的参数 backoffCount（请参阅相关文档中所列的蓝牙规范文档）。
10	backoffPar	0-3	logUpperLimit	R/W	扫描仪退避过程中使用的参数 upperLimit 的二进制对数
		4	bLastSucceeded	R/W	如果最后一个 SCAN_RSP 已成功接收且 upperLimit 未更改，则为 1
		5	bLastFailed	R/W	如果接收最后一个 SCAN_RSP 失败且 upperLimit 未更改，则为 1
		6-7			保留
11	connectReqLen			W	连接请求数据的大小
12-15	pConnectReqData			W	指向缓冲器的指针，该缓冲器包含要进入 CONNECT_IND 或 AUX_CONNECT_REQ 的 LLData
16-19	pDeviceAddress			W	指向用于该器件的器件地址的指针（最低有效位设置为 0）。如果最低有效位为 1，则 initConfig.deviceAddrType 给出的地址类型被反转。
20-23	pWhiteList			W	指向白名单或对等地址的指针（最低有效位设置为 0）。如果最低有效位为 1，则 initConfig.peerAddrType 给出的地址类型被反转。
24-27	connectTime			R/W	指示第一个连接事件的第一个可能开始时间的计时器值。如果已建立连接，则设置为计算值；如果未建立连接、则设置为下一个可能的连接时间。
28-29	maxWaitTimeForAuxCh			W	在命令内切换至次要扫描的最长等待时间。如果事件开始的时间大于此值，命令将以 BLE_DONE_AUX 结束。如果尺寸更小，无线电将自动切换到正确的信道和 PHY。
30	timeoutTrigger			W	触发器，使器件在允许的情况下停止接收
31	endTrigger			W	触发器，使器件在允许的情况下停止接收
32-35	timeoutTime			W	与 timeoutTrigger 一起使用的时间，使设备在允许的情况下尽快停止接收，以 BLE_DONE_RXTIMEOUT 结束
36-39	endTime			W	与 endTrigger 一起使用的时间，使设备在允许的情况下尽快停止接收，以 BLE_DONE_ENDED 结束

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-108. Bluetooth®5 启动器命令 (1) （续）

字节索引	字段名称	位	位字段名称	类型	说明
40-43	rxStartTime			R	启动 RX 以接收数据包所需的时间
44-45	rxListenTime			R	接收数据包进行侦听所需的时间。 0：无 AUX 数据包
46	channelNo			R	用于辅助广播的通道编号
47	phyMode			R	要在辅助通道上使用的 PHY： 0：1Mbps 1：2Mbps 2：编码 其他：保留
40	auxChRes				

(1) 此参数结构用于蓝牙 5 启动器命令，CMD_BLE5_INITIATOR。

25.6.1.3 输出结构

表 25-109. 主从命令

字节索引	字段名称	类型	说明
0	nTx	R/W	已传输的数据包总数（包括自动空包和重传）
1	nTxAck	R/W	已传输且已被确认的数据包总数（包括自动空包）
2	nTxCtrl	R/W	从 TX 队列传输的唯一 LL 控制包数量
3	nTxCtrlAck	R/W	TX 队列中已完成（已确认）的 LL 控制包数量
4	nTxCtrlAckAck	R/W	已确认并且已发送 ACK 作为响应的 LL 控制包数量
5	nTxRetrans	R/W	已执行的重传次数
6	nTxEntryDone	R/W	TX 队列中已完成（已确认）的数据包数量
7	nRxOk	R/W	已收到且包含有效载荷、CRC 正确且未被忽略的数据包数量
8	nRxCtrl	R/W	已收到、CRC 正确且未被忽略的 LL 控制包数量
9	nRxCtrlAck	R/W	已收到、CRC 正确且未被忽略，然后得到确认的 LL 控制包数量
10	nRxNok	R/W	已收到但 CRC 错误的数据包数量
11	nRxIgnored	R/W	已收到、CRC 正确但因序列号重复而被忽略的数据包数量
12	nRxEmpty	R/W	已收到、CRC 正确且无载荷的数据包数量
13	nRxBufFull	R/W	已收到但因缓冲区空间不足而被丢弃的数据包数量
14	lastRssi	R	最后收到的数据包的 RSSI（有符号数）
15	pktStatus	R/W	收到的数据包的状态（参阅表 25-122）
16-19	timeStamp	R	从器件操作：收到的第一个数据包的时间戳

表 25-110. 广播器命令

字节索引	字段名称	类型	说明
0-1	nTxAdvInd	R/W	完整传输的 ADV*_IND 数据包数量
2	nTxScanRsp	R/W	传输的 AUX_SCAN_RSP 或 SCAN_RSP 数据包数量
3	nRxScanReq	R/W	正确接收且未被忽略的 AUX_SCAN_REQ 或 SCAN_REQ 数据包数量
4	nRxConnectReq	R/W	正确接收且未被忽略的 AUX_CONNECT_REQ 或 CONNECT_IND (CONNECT_REQ) 数据包数量
5	nTxConnectRsp	R/W	传输的 AUX_CONNECT_RSP 数据包数量
6-7	nRxNok	R/W	已收到但 CRC 错误的数据包数量
8-9	nRxIgnored	R/W	已收到、CRC 正确但被忽略的数据包数量
10	nRxBufFull	R/W	已收到但无法放入 RX 队列的数据包数量
11	lastRssi	R	最后收到的数据包的 RSSI（有符号数）

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-110. 广播器命令（续）

字节索引	字段名称	类型	说明
12-15	timeStamp	R	最后收到的数据包的时间戳

表 25-111. 传统扫描器命令

字节索引	字段名称	类型	说明
0-1	nTxScanReq	R/W	传输的 SCAN_REQ 数据包数量
2-3	nBackedOffScanReq	R/W	因退避过程而未发送的 SCAN_REQ 数据包数量
4-5	nRxAdvOk	R/W	已收到、CRC 正确且未被忽略的 ADV*_IND 数据包数量
6-7	nRxAdvIgnored	R/W	已收到、CRC 正确但被忽略的 ADV*_IND 数据包数量
8-9	nRxAdvNok	R/W	已收到但 CRC 错误的 ADV*_IND 数据包数量
10-11	nRxScanRspOk	R/W	已收到、CRC 正确且未被忽略的 SCAN_RSP 数据包数量
12-13	nRxScanRspIgnored	R/W	已收到、CRC 正确但被忽略的 SCAN_RSP 数据包数量
14-15	nRxScanRspNok	R/W	已收到但 CRC 错误的 SCAN_RSP 数据包数量
16	nRxAdvBufFull	R/W	已收到但无法放入 RX 队列的 ADV*_IND 数据包数量
17	nRxScanRspBufFull	R/W	已收到但无法放入 RX 队列的 SCAN_RSP 数据包数量
18	lastRssi	R	最后收到的数据包的 RSSI（有符号数）
19			保留
20-23	timeStamp	R	最后成功接收且未被忽略的 ADV*_IND 数据包的时间戳

表 25-112. 传统发起器命令

字节索引	字段名称	类型	说明
0	nTxConnectReq	R/W	传输的 CONNECT_IND (CONNECT_REQ) 数据包数量
1	nRxAdvOk	R/W	已收到、CRC 正确且未被忽略的 ADV*_IND 数据包数量
2-3	nRxAdvIgnored	R/W	已收到、CRC 正确但被忽略的 ADV*_IND 数据包数量
4-5	nRxAdvNok	R/W	已收到但 CRC 错误的 ADV*_IND 数据包数量
6	nRxAdvBufFull	R/W	已收到但无法放入 RX 队列的 ADV*_IND 数据包数量
7	lastRssi	R	最后收到的数据包的 RSSI（有符号数）
8-11	timeStamp	R	导致发送 CONNECT_IND (CONNECT_REQ) 的已收到 ADV*_IND 数据包的时间戳

表 25-113. 低功耗 Bluetooth® 5 扫描器和发起器命令

字节索引	字段名称	类型	说明
0-1	nTxReq	R/W	传输的 AUX_SCAN_REQ、SCAN_REQ、AUX_CONNECT_REQ 或 CONNECT_IND 数据包数量
2-3	nBackedOffReq	R/W	因退避过程而未发送的 AUX_SCAN_REQ、SCAN_REQ 或 AUX_CONNECT_REQ 数据包数量
4-5	nRxAdvOk	R/W	已收到、CRC 正确且未被忽略的 ADV*_IND 数据包数量
6-7	nRxAdvIgnored	R/W	已收到、CRC 正确但被忽略的 ADV*_IND 数据包数量
8-9	nRxAdvNok	R/W	已收到但 CRC 错误的 ADV*_IND 数据包数量
10-11	nRxRspOk	R/W	已收到、CRC 正确且未被忽略的 AUX_SCAN_RSP、SCAN_RSP 或 AUX_CONNECT_RSP 数据包数量
12-13	nRxRspIgnored	R/W	已收到、CRC 正确但被忽略的 AUX_SCAN_RSP、SCAN_RSP 或 AUX_CONNECT_RSP 数据包数量
14-15	nRxRspNok	R/W	已收到但 CRC 错误的 AUX_SCAN_RSP、SCAN_RSP 或 AUX_CONNECT_RSP 数据包数量
16	nRxAdvBufFull	R/W	已收到但无法放入 RX 队列的 ADV*_IND 数据包数量
17	nRxRspBufFull	R/W	已收到但无法放入 RX 队列的 AUX_SCAN_RSP、SCAN_RSP 或 AUX_CONNECT_RSP 数据包数量

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-113. 低功耗 Bluetooth® 5 扫描器和发起器命令（续）

字节索引	字段名称	类型	说明
18	lastRssi	R	最后收到的数据包的 RSSI（有符号数）
19			保留
20-23	timeStamp	R	最后成功接收且未被忽略的 *ADV*_IND 数据包的时间戳

表 25-114. 通用 RX 命令

字节索引	字段名称	类型	说明
0-1	nRxOk	R/W	已收到且 CRC 正确的数据包数量
2-3	nRxNok	R/W	已收到但 CRC 错误的数据包数量
4-5	nRxBufFull	R/W	已收到但因缓冲区空间不足而被丢弃的数据包数量
6	lastRssi	R	最后收到的数据包的 RSSI（有符号数）
7			保留
8-11	timeStamp	R	最后收到的数据包的时间戳

表 25-115. 测试 TX 命令

字节索引	字段名称	类型	说明
0-1	nTx	R/W	传输的数据包数量

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.6.1.4 其他结构及位字段

表 25-116. 接收队列条目配置位字段 (1)

位索引	位字段名称	类型	说明
0	bAutoFlushIgnored	W	如果为 1，则自动从 RX 队列中删除被忽略的数据包。
1	bAutoFlushCrcErr	W	如果为 1，则自动从 RX 队列中删除存在 CRC 错误的数据包。
2	bAutoFlushEmpty	W	如果为 1，则自动从 RX 队列中删除空数据包。
3	bIncludeLenByte	W	如果为 1，则将接收到的长度字节包含在存储的数据包中；否则丢弃该字节。
4	bIncludeCrc	W	如果为 1，则将接收到的 CRC 字段包含在存储的数据包中；否则丢弃该字段。
5	bAppendRssi	W	如果为 1，则将 RSSI 字节附加到 RX 队列中的数据包。
6	bAppendStatus	W	如果为 1，则将一个状态字附加到 RX 队列中的数据包。
7	bAppendTimestamp	W	如果为 1，则向 RX 队列中的数据包附加时间戳。

(1) 此位字段用于参数结构的 rxConfig 字节。

表 25-117. 序列号状态位字段 (1)

位索引	位字段名称	类型	说明
0	lastRxSn	R/W	与 CRC OK 一同接收的最后一个数据包的标头的 SN 位
1	lastTxSn	R/W	最后发送的数据包标头的 SN 位
2	nextTxSn	R/W	要发送的下一个数据包标头的 SN 位
3	bFirstPkt	R/W	对于从器件：如果在连接上发送数据包，则为 0；否则为 1
4	bAutoEmpty	R/W	如果最后发送的数据包是自动空的数据包，则为 1
5	bLlCtrlTx	R/W	如果最后发送的数据包是 LL 控制数据包 (LLID = 11)，则为 1
6	bLlCtrlAckRx	R/W	如果最后接收到的数据包是 LL 控制数据包的 ACK，则为 1
7	bLlCtrlAckPending	R/W	如果最后成功接收到的数据包是 LL 控制数据包，则为 1，该数据包尚未进行 ACK 处理

(1) 该位字段用于主器件和从器件参数结构的 seqStat 字节。

白名单结构具有数组形式。每个元素由 8 个字节组成，如表 25-118 所示。第一个元素的第一个字节告知条目数，而该字节保留在其余条目中。第二个字节包含一些配置位，其余 6 个字节包含地址。

表 25-118. 白名单结构

字节索引	字段名称	位	位字段名称	类型	说明
0	大小			W	while 列表条目数。仅用于列表的第一个条目。
1	conf	0	bEnable	W	如果该条目正在使用中，则为 1；如果该条目未在使用中，则为 0
		1	addrType	W	条目中的类型地址 公开 (0) 或随机 (1)
		2	bWlIgn	R/W	如果不存在 AdvDataInfo 字段，扫描器将忽略该条目，从而为 1，否则为 0。用于屏蔽已扫描和报告的条目。
		3			保留
		4	bIrkValid	R/W	如果存在有效的 IRK，则由启动器忽略该条目，从而为 1；否则为 0
		5-7			保留
2-3	地址			W	条目中包含的地址的 16 个最低有效位
4-7	addressHi			W	条目中包含的地址的最高有效 32 位

ADI 列表的形式是一个数组，该数组由表 25-119 中提供的类型的 16 个条目组成。

表 25-119. 广播数据 ID 条目结构

位索引	位字段名称	类型	说明
0-11	advDataId	R/W	如果模式 = 1：和数组中的条目编号相对应的广播集 ID (SID) 的上次广播数据 ID (DID)

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-119. 广播数据 ID 条目结构（续）

位索引	位字段名称	类型	说明
12-13	mode	R/W	0：条目无效（始终接收具有给定 SID 的数据包） 1：条目有效（忽略具有给定 SID 的数据包，其中 DID 等于 advDataId） 2：条目被阻止（始终忽略具有给定 SID 的数据包） 3：保留
14-15			保留

表 25-120. 旧命令的接收状态字节位字段 (1)

位索引	位字段名称	类型	说明
0-5	通道	R	接收数据包的通道，前提是通道在 0 – 39 范围内；否则为 0x3F
6	bIgnore	R	如果数据包被标记为忽略，则为 1；否则为 0
7	bCrcErr	R	如果接收到的数据包存在 CRC 错误，则为 1；否则为 0

(1) 如果已配置，该位字段的一个字节会附加到旧命令接收到的条目中。

表 25-121. 低功耗 Bluetooth®5 命令的接收状态字位字段 (1)

位索引	位字段名称	类型	说明
0-5	通道	R	接收数据包的通道，前提是通道在 0 – 39 范围内；否则为 0x3F
6	bIgnore	R	如果数据包被标记为忽略，则为 1；否则为 0
7	bCrcErr	R	如果接收到的数据包存在 CRC 错误，则为 1；否则为 0
8-9	phyMode	R	接收数据包的 PHY 0：1Mbps 1：2Mbps 2：编码，S = 8 (125kbps) 3：编码，S = 2 (500kbps)
10-15			保留

(1) 如果配置了低功耗蓝牙 5 命令，该位字段的一个 16 位字会附加到接收到的条目中。

主和从输出结构字段 pktStatus 具有表 25-122 中列出的格式。如果将时间戳写入输出结构，则无线电 CPU 会在操作开始时将 bTimeStampValid 位设置为 0；如果将时间戳写入输出结构，则设置为 1（这仅在从器件运行模式下发生）。当一个数据包被完全接收时，根据 CRC 结果设置 bLastCrcErr 位；如果没有接收到数据包，则该位不受影响。当在 CRC OK 的情况下接收数据包时，其余位被设置；如果没有正确接收数据包、则这些位不受影响。

表 25-122. 主器件和从器件数据包状态字节

位索引	位字段名称	类型	说明
0	bTimeStampValid	R/W	如果将有效时间戳写入 timeStamp，则为 1；否则为 0
1	bLastCrcErr	R/W	如果最后接收到的数据包有 CRC 错误，则为 1；否则为 0
2	bLastIgnored	R/W	如果最后接收到的具有 CRC OK 的数据包被忽略，则为 1；否则为 0
3	bLastEmpty	R/W	如果最后接收到的具有 CRC OK 的数据包为空，则为 1；否则为 0
4	bLastCtrl	R/W	如果最后接收到的具有 CRC OK 的数据包是 LL 控制数据包，则为 1；否则为 0
5	bLastMd	R/W	如果最后接收到的具有 CRC OK 的数据包的 MD = 1，则为 1；否则为 0
6	bLastAck	R/W	如果最后接收到的具有 CRC OK 的数据包是已传输数据包的 ACK，则为 1；否则为 0
7			保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-123. 通用扩展数据包条目格式 (1)

字节索引	字段名称	位	位字段名称	类型	说明
0	extHdrInfo	0-5	length	W	扩展标头长度
		6-7	advMode	W	低功耗蓝牙中定义的广播器模式： 0：不可连接，不可扫描 1：可连接，不可扫描 2：不可连接，可扫描 3：保留
1	extHdrFlags			W	低功耗蓝牙中定义的扩展标头标志
2	extHdrConfig	0	bSkipAdvA	W	0：如果在 extHdrFlags 中配置了 AdvA，则扩展有效载荷中存在该项。 1：如果在 extHdrFlags 中配置了 AdvA，则该项会自动从命令结构中插入，并在扩展标头中被省略。
		1	bSkipTargetA	W	0：如果在 extHdrFlags 中配置了 TargetA，则扩展有效载荷中存在该项。对于响应消息，发送时该值将替换为接收到的地址。 1：如果在 extHdrFlags 中配置，TargetA 会从命令结构或接收到的地址自动插入，并且在扩展标头中被省略。
		2	deviceAddrType	W	如果 bSkipAdvA = 0：扩展标头缓冲器中的器件地址类型 公共 (0) 或随机 (1)
		3	targetAddrType	W	如果 bSkipAdvA = 0：扩展标头缓冲器中的目标地址类型 公共 (0) 或随机 (1)
		4-7			保留
3	advDataLen			W	有效载荷缓冲器的大小
4-7	pExtHeader			W	指向包含扩展标头的缓冲器的指针。如果不存在除扩展标头标志、自动广播器地址或自动目标地址之外的任何字段，指针可能为 NULL。
8-11	pAdvData			W	指向包含 advData 的缓冲器的指针。如果 advDataLen = 0，则指针可以为 NULL。

(1) 此结构用于通用扩展广播有效载荷格式的发送数据包的数据包格式（请参阅相关文档中列出的蓝牙系统规范 5.0 版），与 CMD_BLE5_ADV_EXT 和 CMD_BLE5_ADV_AUX 一起使用。

25.6.2 中断

无线电 CPU 使用固件定义的中断将事件发送回系统 CPU。表 25-124 列出了低功耗蓝牙命令使用的中断。每个中断都可在系统 CPU 中单独启用。节 25.8 介绍了有关何时生成中断的详细信息。

表 25-124. 适用于低功耗 Bluetooth® 的中断定义

中断号	中断名称	说明
0	Command_Done	无线电操作命令已完成
1	Last_Command_Done	命令链中的最后一个无线电操作命令已完成
4	Tx_Done	传输了数据包
5	Tx_Ack	在发送的数据包上接收到确认
6	Tx_Ctrl	传输的 LL 控制数据包
7	Tx_Ctrl_Ack	在发送的 LL 控制数据包上接收到确认
8	Tx_Ctrl_Ack_Ack	在传输的 LL 控制数据包上接收到确认，并针对该数据包发送确认
9	Tx_Retrans	已重新传输数据包
10	Tx_Entry_Done	TX 队列数据条目状态更改为“已完成”
11	Tx_Buffer_Changed	缓冲器更改在 CMD_BLE_ADV_PAYLOAD 之后完成
16	Rx_Ok	接收到的数据包，具有 CRC OK、有效载荷并且不能被忽略
17	Rx_Nok	接收到的数据包，具有 CRC 错误
18	Rx_Ignored	接收到的数据包，具有 CRC OK，但将被忽略

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-124. 适用于低功耗 Bluetooth® 的中断定义（续）

中断号	中断名称	说明
19	Rx_Empty	接收到的数据包，具有 CRC OK，但不会被忽略，无有效载荷
20	Rx_Ctrl	接收到 LL 控制数据，具有 CRC OK，不能忽略
21	Rx_Ctrl_Ack	接收到 LL 控制数据，具有 CRC OK，不能忽略，然后发送确认
22	Rx_Buf_Full	接收的数据包不适合 RX 队列
23	Rx_Entry_Done	RX 队列数据条目，将状态更改为已完成
29	Modules_Unlocked	在启动过程中，Arm Cortex-M0 处理器已打开对 RF 内核模块和存储器的访问权限
30	Boot_Done	RF 内核 CPU 引导完成
31	Internal_Error	无线电 CPU 观察到意外错误

25.7 数据处理

对于所有低功耗 Bluetooth 命令，通过无线方式接收的数据存储在接收队列中。对于主从操作，要发送的数据从传输队列中获取；而对于非连接操作，数据则从特定缓冲区获取，或由无线电 CPU 完全基于其他可用信息创建。

25.7.1 接收缓冲器

正在接收的数据包存储在接收缓冲器中。首先，如果 config.lenSz 在 RX 条目中配置了长度字节或字，则会存储一个长度字节或词，如节 25.3.2.7.2 中所述。这根据通过空中接收的长度和附加信息的配置而计算。

在可选长度字段之后，接收到的标头和有效载荷会以无线方式存储。如果 rxConfig.bIncludeLenByte 为 1，则会存储完整的 16 位标头（包括接收到的长度字段），尽管长度字段是冗余信息（如果存在长度字节或字）。如果 rxConfig.bIncludeLenByte 为 0，则仅存储标头的第一个字节，以便第二个字节（仅包含冗余长度字段）以及一些 RFU 位被丢弃（取决于蓝牙版本）。

如果 rxConfig.bIncludeCrc 为 1，则接收到的 CRC 值将存储在 RX 缓冲器中。如果 rxConfig.bAppendRssi 为 1，则附加一个指示接收到的 RSSI 值的有符号字节。如果 rxConfig.bAppendStatus 为 1，则会附加一个状态字。对于传统的低功耗蓝牙命令，这是表 25-120 中定义的 1 字节字段，而对于低功耗蓝牙 5 命令，这是表 25-121 中定义的 2 字节字段。如果 rxConfig.bAppendTimeStamp 为 1，则附加一个指示数据包开始的时间戳。此时间戳对应于 ratmr_t 数据类型，该数据类型是小端字节序格式的 32 位值。不应进行填充以确保多字节字段（时间戳及低功耗蓝牙 5 状态）进行特定对齐。因此，必须按字节写入和读取多字节字段。

图 25-8 显示了 RX 队列中条目元素的格式。

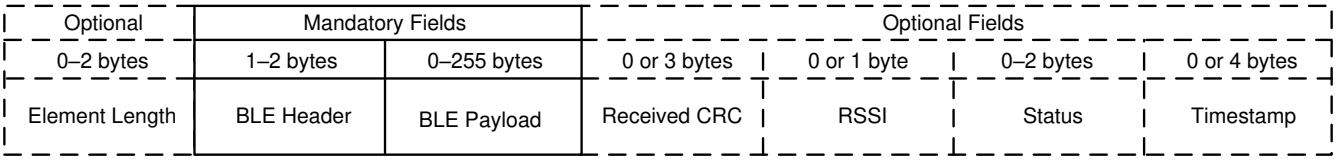


图 25-8. 接收缓冲器条目元素

25.7.2 发送缓冲器

对于主从操作，发送缓冲器在缓冲队列中设置。数据包的长度由数据条目中的长度字段定义。数据条目的第一个字节提供进入数据通道数据包标头的 LLID。NESN、SN 和 MD 位应由无线电 CPU 自动插入，RFU 位应设置为 0，长度字段应根据数据条目的长度计算。

对于传统广播通道数据包和低功耗蓝牙扫描仪和发起方发送的数据包，无线电 CPU 应自动生成有效载荷的标头和地址字段。每种消息类型的地址字段之后的数据由指向数据缓冲器的指针提供。此缓冲器中的字节数在一个单独的参数中给出。如果没有要发送的数据字节，可以通过将长度设置为 0 来表示这种情况。在这种情况下，指针应被忽略，并可设置为 NULL。为了符合低功耗蓝牙标准，ADV_DIRECT_IND 和 SCAN_REQ 消息应该没有有效载荷（但为了能够覆盖这一点，传统命令中仍然存在数据缓冲器）。根据蓝牙规范，使用低功耗蓝牙 5 扫描仪命令

Copyright © 2026 Texas Instruments Incorporated

时，始终在没有有效载荷的情况下发送 SCAN_REQ 和 AUX_SCAN_REQ 消息。对于 CONNECT_IND 和 AUX_CONNECT_REQ 消息，数据必须具有长度 22 才能符合低功耗蓝牙标准，但实现应允许任何长度。

对于低功耗蓝牙 5 广播命令，发送的数据包使用通用扩展广播格式。表 25-123 列出了用于描述此类数据包的结构。PDU 标头和扩展标头的部分应由基于此结构的无线电 CPU 自动生成。发送的数据包、PDU 标头或扩展标头都有指向包含扩展标头其余部分的缓冲器和包含有效载荷的另一个缓冲器的指针。

25.8 无线电操作命令说明

25.8.1 Bluetooth®5 无线电设置命令

运行蓝牙 5 时，应使用 CMD_BLE5_RADIO_SETUP 命令设置无线电。此命令将使用与 CMD_RADIO_SETUP 命令类似的方式设置无线电，但可以在不同的受支持 PHY 之间切换命令。

如果该命令与非低功耗蓝牙 RX 或 TX 命令一同使用，则参数 *defaultPhy* 可让 PHY 使用。对于低功耗蓝牙命令，defaultPhy 参数将被忽略。低功耗蓝牙 5 命令中包含多个 PHY 选项，如果使用 CMD_BLE5_RADIO_SETUP 命令配置了无线电，则传统低功耗蓝牙命令将始终使用 1Mbps PHY。

参数 *config* 和 *txPower* 同 CMD_RADIO_SETUP 命令的含义相同。

该命令有四个用于覆盖列表的指针。这些覆盖列表的处理方式和 CMD_RADIO_SETUP 命令相同。仅在运行设置命令时才会处理覆盖列表 pRegOverrideCommon，并且应包含适用于所有 PHY 的任何覆盖。在切换到给定的 PHY 时，会处理覆盖列表 pRegOverride1Mbps、pRegOverrideCoded 和 pRegOverride2Mbps。无线电 CPU 将存储这些覆盖指针并在每个交换机上读取覆盖列表，因此这些覆盖列表必须保留在内存中。如果不需要覆盖，则任何覆盖指针都可以为 NULL。

若要运行仅支持 1Mbps 的传统低功耗蓝牙模式，模式 0 的 CMD_RADIO_SETUP 可供使用，和在 CC26x0 器件上使用时一样。

25.8.2 用于低功耗 Bluetooth® 数据包传输的无线电操作命令

在运行本文档中介绍的任何无线电操作命令（CMD_BLE5_RADIO_SETUP 命令除外）之前，必须使用 CMD_BLE5_RADIO_SETUP 命令或 CMD_RADIO_SETUP 命令将无线电设置为低功耗蓝牙模式。否则，操作将以错误结束。运行任何低功耗蓝牙 5 命令时，必须使用 CMD_BLE5_RADIO_SETUP 命令。

操作从系统 CPU 发出的无线电操作命令开始。操作的实际开始由无线电 CPU 根据命令结构中的 startTrigger 及 startTime 进行设置。此时，无线电 CPU 开始配置发送器或接收器，具体取决于操作类型。在计算运行开始时间时，系统 CPU 必须考虑发送器或接收器的设置时间。

无线电 CPU 根据通道参数设置通道。如果通道在 0 – 39 范围内，则表示数据通道索引或广播通道索引。在这种情况下，主机和主器件命令中只允许值 0 – 36，在广播器、扫描器和启动器命令中只允许值 37 – 39。如果通道在 60 – 207 范围内，则表示偏移为 2300MHz 的 RF 频率。如果通道为 255，则无线电 CPU 不会对任何频率字进行编程，但会保持已使用 CMD_FS 编程的频率。如果通道为 255 且频率合成器未运行，则操作结束时会出现错误。

白化参数指示用于在低功耗蓝牙模式下进行数据白化的 7 位 LFSR 的初始化。如果 whitening.bOverride 为 0 且通道在 0 – 39 范围内，则 LFSR 使用（0x40 | 通道）进行初始化。否则，LFSR 会用 whitening.init 进行初始化。如果 whitening.init 在这种情况下为 0，则不使用白化。

使用低功耗蓝牙无线电操作命令传输的所有数据包都应当附加符合低功耗蓝牙标准的 CRC。对使用低功耗蓝牙无线电操作命令接收到的所有数据包，应执行低功耗蓝牙兼容 CRC 校验。为每条命令定义了 CRC 寄存器的初始化。

低功耗蓝牙 5 命令具有一些附加参数。phyMode 参数用于选择将哪个 PHY 用于该命令。对于编码的 PHY，phyMode.coding 提供了一条用于选择每个数据包所用编码的规则（有关详细信息，请参阅节 25.8.3）。txPower 参数可用于选择在此命令中使用的特定 TX 功率，该功率会覆盖无线电设置命令中设置的功率和使用 CMD_SET_TX_POWER 命令设置的功率。如果 txPower 设置为 0x0000，则使用设置命令或最后一条

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

CMD_SET_TX_POWER 命令的 TX 功率。rangeDelay 参数提供了 T_IFS 之后接收器使用的额外侦听时间。给定的 RAT 节拍数会添加至侦听窗口的末尾。

为了满足 T_IFS 的要求，无线电 CPU 对接收之后的发送进行计时和同步。对于发送后立即接收，无线电 CPU 将在 RX 和超时开始时计时，以便它始终在通过低功耗蓝牙标准设定的限制范围内接收发送的数据包，但没有过多裕度，以避免在广播通道上进行错误同步。对于从器件命令中的第一个接收操作，无线电 CPU 设置在 pParams->timeoutTrigger 和 pParams->timeoutTime 中定义的超时。无论是在从器件还是对等主器件中，该触发时间都取决于睡眠时钟的不确定性。

当接收器运行时，消息将被接收到 RX 条目中，如节 25.3.2.7.2 中所述。无线电 CPU 应具有标志 bCrcErr 和 bIgnore，这些标志将写入 RX 条目的状态字节的相应字段（如果存在）。如果接收到的数据包中存在 CRC 错误，则应设置 bCrcErr 标志。如果 CRC 为 OK，则可以根据为每个角色定义的原则设置 bIgnore 标志。此标志意味着系统 CPU 可能会忽略数据包。接收到数据包后，无线电 CPU 应向系统 CPU 发出中断。

如果接收到的数据包的长度字段大于下面定义的最大长度，则停止接收，并将其视为未在数据包上获得同步。当使用 CMD_RADIO_SETUP 命令设置无线电时，数据通道数据包中的长度字段配置为 5 位，广播通道数据包中的长度字段配置为 6 位，这与蓝牙 4.0 和蓝牙 4.1 中的规范相对应。在蓝牙 4.2 中，数据通道数据包具有 8 位长度的字段，这可以通过覆盖进行配置，请参阅节 25.8.4。当使用 CMD_BLE5_RADIO_SETUP 命令设置无线电时，所有长度字段都配置为 8 位，这与蓝牙 5.0 中的规范相对应。

默认情况下，旧广播通道数据包的最大允许有效载荷长度为 37，扩展广播通道数据包的最大允许有效载荷长度为 255。对于旧主主器件命令，接收到的数据通道数据包的默认最大允许长度为 31（使用 CMD_RADIO_SETUP 命令的默认设置不会违反该长度，因为这种情况下的长度字段为 5 位）。对于低功耗蓝牙 5 主器件和从器件命令，接收到的数据包的最大数据包长度在命令结构 pParams->maxRxPktLen 中设置。

如果设置了 bCrcErr 或 bIgnore 标志，或者如果数据包为空（根据每个操作中的定义），则可在发出中断之前从 RX 条目中删除数据包，具体取决于 pParams->rxConfig 的 bAutoFlushIgnored、bAutoFlushCrc 和 bAutoFlushEmpty 位。

应在操作期间更新所发出命令的状态字段。提交命令时，系统 CPU 应将 IDLE 状态写入此字段。在操作过程中，无线电 CPU 应更新字段以指示操作模式。操作完成后，无线电 CPU 应写入一个状态，指示操作已完成。表 25-125 列出了低功耗蓝牙无线电操作要使用的状态代码。

表 25-125. 低功耗 Bluetooth® 无线电操作状态代码

编号	名称	说明
操作未完成		
0x0000	IDLE	操作未开始
0x0001	PENDING	等待启动触发
0x0002	ACTIVE	运行操作
操作正常完成		
0x1400	BLE_DONE_OK	操作正常结束
0x1401	BLE_DONE_RXTIMEOUT	从器件操作的第一个 RX 超时或扫描窗口结束
0x1402	BLE_DONE_NOSYNC	后续 RX 超时
0x1403	BLE_DONE_RXERR	由于接收错误（CRC 或其他），操作结束
0x1404	BLE_DONE_CONNECT	已接收或发送 CONNECT_IND 或 AUX_CONNECT_RSP
0x1405	BLE_DONE_MAXNACK	超过最大重新发送数
0x1406	BLE_DONE_ENDED	操作在结束触发后停止
0x1407	BLE_DONE_ABORT	操作由中止命令中止
0x1408	BLE_DONE_STOPPED	停止命令后操作停止
0x1409	BLE_DONE_AUX	在收到指示需较长时间的 AuxPtr 后，操作结束
0x140A	BLE_DONE_CONNECT_CHSEL0	接收或发送的 CONNECT_IND；对等不支持通道选择算法 2
操作完成。存在错误		

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-125. 低功耗 Bluetooth® 无线电操作状态代码（续）

编号	名称	说明
0x1800	BLE_ERROR_PAR	非法参数
0x1801	BLE_ERROR_RXBUF	没有可用的 RX 缓冲器（广播器、扫描器、启动器）
0x1802	BLE_ERROR_NO_SETUP	未在低功耗蓝牙模式下设置无线电
0x1803	BLE_ERROR_NO_FS	运行 RX 或 TX 时未对合成器进行编程
0x1804	BLE_ERROR_SYNTH_PROG	合成器编程失败
0x1805	BLE_ERROR_RXOVF	在操作期间观察到 RX 溢出
0x1806	BLE_ERROR_TXUNF	运行期间观察到 TX 下溢
0x1807	BLE_ERROR_AUX	AUX 指针目标距离未来太远

列出了每个操作给出每个状态的条件。表 25-125 中列出的一些错误原因不会在这些列表中重复出现。在某些情况下，可能会出现一般错误原因。在所有这些情况下，操作的结果都是 ABORT。

25.8.3 编码 PHY 的编码选择

选择编码 PHY 后，会使用蓝牙 5 命令的 phyMode.coding 参数，该参数决定了发送数据包的编码速率。phyMode.coding 参数对于链路层连接命令（主器件和从器件）和其他命令具有不同的含义。

对于主器件命令和从器件命令，pParams->maxLenLowRate 参数也用于确定编码率。如果发送的数据包的长度大于 pParams->maxLenLowRate，则无论 phyMode.coding 如何，在 S = 2 (500kbps) 的情况下传输数据包。此特性可用于确保不违反最大数据包持续时间。在其他情况下，使用 phyMode.coding 参数，位的含义在表 25-126 中进行了说明。

表 25-126. 主器件和从器件命令的编码选择

位编号	说明
0	0：默认为 S = 8 (125 kbps) 1：默认为 S = 2 (500 kbps)
1	0：请勿根据上次接收到的数据包来修改默认速率 1：如果在同一连接事件中最后接收到的数据包使用 S = 8 (125kbps) 进行编码，则始终使用 S = 8 (125kbps)
2	0：使用空数据包的默认速率 1：对于空数据包，始终使用 S = 8 (125 kbps)
3	0：使用默认速率进行重新传输 1：务必使用 S = 8 (125kbps) 进行重新传输
4-5	保留

对于其他蓝牙 5 命令，表 25-127 中给出了 phyMode.coding 的含义。对于 CMD_BLE5_ADV_EXT 和 CMD_BLE5_TX_TEST 命令，位 1 和位 2 不适用。对于 CMD_BLE5_SCANNER 和 CMD_BLE5_INITIATOR 命令，位 0 不适用。对于 CMD_BLE5_ADV_AUX 命令，位 0、位 1 和位 2 都适用。

表 25-127. 广播器、扫描仪和启动器命令的编码选择

位编号	说明
0	用于广播指示（ADV_EXT_IND、AUX_ADV_IND、AUX_CHAIN_IND）和 TX 测试数据包的码率 0：使用 S = 8 (125 kbps) 1：使用 S = 2 (500 kbps)
1	请求和响应数据包使用的默认码速率（AUX_SCAN_REQ、AUX_CONNECT_REQ、AUX_SCAN_RSP、AUX_CONNECT_RSP） 0：默认为 S = 8 (125 kbps) 1：默认为 S = 2 (500 kbps)
2	0：使用请求及响应数据包的默认速率 1：如果最后接收到的数据包使用 S = 8 (125kbps) 进行编码，则始终将 S = 8 (125kbps) 用于请求或响应数据包

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-127. 广播器、扫描仪和启动器命令的编码选择（续）

位编号	说明
3-5	保留

25.8.4 参数覆盖

使用 CMD_BLE5_RADIO_SETUP、CMD_RADIO_SETUP、CMD_UPDATE_RADIO_SETUP 或 CMD_WRITE_FWPAR 命令仍可覆盖几个不可以通过参数结构配置的参数。

在蓝牙低功耗模式下可以覆盖的参数在相关文档中列出的蓝牙规范文档中进行了说明。例如，这些参数可用于提高性能或创建在某些情况下可能有用的不合规行为。示例如下：

- 修改 T_IFS 和其他时序
- 修改最大 RX 数据包长度（蓝牙 4.2 中的 LE 数据包长度扩展功能需要）
- 修改自动生成的标头等
- 对 CRC 进行修改，从而生成用于测试的位错误并使用不同的 CRC

25.8.5 链路层连接

在主器件或从器件操作开始时，无线电 CPU 应等待启动触发条件。然后，它应根据命令结构的通道参数来编程频率。通道参数不允许设为 37、38 或 39，因为它们不是数据通道。对于低功耗 Bluetooth 5 命令，还应设置 phyMode.mainMode 中给出的 PHY 模式。无线电 CPU 应设置 pParams->accessAddress 中定义的访问地址，并应使用 pParams->crcInit 中定义的 CRC 初始化值。应按照白化参数中的定义设置白化器。然后，无线电 CPU 应配置接收器或传输器。该操作交替进行接收与发送，直至满足某个命令结束条件为止。

当解调器在一条消息上获得同步时，该消息将接收到第一个能够容纳该数据包的可用 RX 缓冲区中。标志 *bCrcErr* 和 *bIgnore* 根据表 25-128 进行设置，具体取决于 CRC 结果以及标头中的 SN 字段是否与上次成功接收的数据包的 SN 字段相同。有效载荷长度为 0 的已接收数据包应视为空包，这意味着如果 pParams->rxConfig.bAutoflushEmpty 为 1 且 bCrcErr 和 bIgnore 均为 0，则该数据包将从 RX 缓冲区中被移除。

表 25-128. 对收到的数据包执行的操作

CRC 结果	SN 与前一个不同	bCrcErr	bIgnore
OK	是	0	0
OK	否	0	1
NOK	X	1	0

如果没有可用的 RX 缓冲区具有足够空间存放收到的数据包，则收到的数据应被丢弃。但是，仍应接收该数据包以便检查 CRC。当数据包收到时，无线电 CPU 应设置序列位，以请求重传丢失的数据包（即 NACK），除非该数据包因 pParams->rxConfig 的设置无论如何都会从 RX 队列中丢弃。

根据低功耗 Bluetooth 规范的要求，如果收到的连续两个数据包都出现 CRC 错误，则命令应结束。

当需要传输或重新传输数据包时，将从 TX 队列中的当前数据条目中读取，除非 TX 队列为空或必须重新发送一个自动空包。无线电 CPU 应按如下方式创建标头。

- LLID 位应从 TX 数据条目的第一个字节插入。
- SN 和 NESN 位将根据低功耗 Bluetooth 协议设置（后文将讨论）
- MD 位将按照后文所述自动计算，除非被覆盖（请参阅节 25.8.4）。
- 如果 TX 队列为空，则传输一个空包（LLID = 0x1，长度 = 0）。这种数据包称为自动空包。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

有许多中断可以在不同的条件下引发。pOutput 结构包含与这些中断对应的多个计数器。表 25-129 列出了递增每个计数器或引发中断的条件。在传输或接收数据包后，可能会满足不止一个条件。在条件列表中，术语*确认*用于定义成功接收的数据包，其标头中的 NESN 值与上次传输的数据包的 SN 值不同。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-129. 主器件和从器件命令中递增计数器和引发中断的条件

条件	递增的计数器	产生的中断
传输了数据包	nTx	Tx_Done
传输了数据包且收到确认	nTxAck	Tx_Ack
传输了 LLID = 11b 的数据包	nTxCtrl	Tx_Ctrl
传输了 LLID = 11b 的数据包且收到确认	nTxCtrlAck	Tx_Ctrl_Ack
传输了 LLID = 11b 的数据包，收到确认，并且已发送确认	nTxCtrlAckAck	Tx_Ctrl_Ack_Ack
传输了数据包且其 SN 与前一个已传输数据包相同	nTxRetrans	Tx_Retrans
传输了带有效载荷的数据包且收到确认	nTxEntryDone	Tx_Entry_Done
收到 bCrcErr = 0、bIgnore = 0 且有效载荷长度 > 0 的数据包	nRxOk	Rx_Ok
收到存在 CRC 错误 (bCrcErr = 1) 的数据包	nRxNok	Rx_Nok
收到 bCrcErr = 0 且 bIgnore = 1 的数据包	nRxIgnored	Rx_Ignored
收到 bCrcErr = 0、bIgnore = 0 且有效载荷长度 = 0 的数据包	nRxEmpty	Rx_Empty
收到 LLID = 11b、bCrcErr = 0 且 bIgnore = 0 的数据包	nRxCtrl	Rx_Ctrl
收到 LLID = 11b、bCrcErr = 0 且 bIgnore = 0 的数据包，并且已发送确认	nRxCtrlAck	Rx_Ctrl_Ack
收到的数据包无法放入 RX 缓冲区且不应被清除	nRxBufFull	Rx_Buf_Full
RX 队列中的第一个 RX 数据条目将状态更改为“已完成”	—	Rx_Entry_Done

无线电 CPU 应维护两个计数器：一个数据包计数器 *nPkt* 和一个 NACK 计数器 *nNack*。在主器件或从器件无线电操作开始时，两个计数器应分别初始化为 pParams->maxPkt 和 pParams->maxNack。每次传输数据包时，数据包计数器 *nPkt* 都应递减。如果收到的数据包不包含对上次传输的数据包的确认，则 NACK 计数器 *nNack* 应递减；否则，如果收到确认，应重置为 pParams->maxNack。如果任一计数器计数到 0，则操作应结束。对于主器件，该操作应在收到一个数据包后发生；对于从器件，应在传输一个数据包后发生。将 pParams->maxPkt 或 pParams->maxNack 设置为 0 将禁用相应的计数器功能。

结束操作的触发条件由 pParams->endTrigger 和 pParams->endTime 设置。如果该参数定义的触发条件发生，无线电操作应尽快结束。无线电操作结束后，任何传输的数据包的 MD 位都应为 0，并且对于从器件，连接事件应在发送下一个数据包后结束；对于主器件，则在收到下一个数据包后结束。如果无线电 CPU 收到即时命令 *CMD_STOP*，其含义与发生结束触发条件时相同（但结束后，状态代码应为 CMD_DONE_STOPPED）。对于从器件命令，pParams->endTrigger.triggerType 可取值为 BLE_TRIG_REL_SYNC，即 15。此值表示 pParams->endTime 相对于第一个收到的数据包的时间戳。如果没有收到数据包，则结束触发条件不会发生（应使用超时触发条件来处理这种情况）。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

寄存器 *pParams->seqStat* 包含一些位，这些位应由无线电 CPU 在操作期间更新，并应被用于确保 SN、NESN 和重新发送操作的正确性。无线电 CPU 必须遵循的规则如下：

- 在连接上执行第一个操作前，系统 CPU 应将 pParams->seqStat 中的位设置如下：
  - lastRxSn = 1
  - lastTxSn = 1
  - nextTxSn = 0
  - bFirstPkt = 1
  - bAutoEmpty = 0
  - bLlCtrlRx = 0
  - bLlCtrlAckRx = 0
  - bLlCtrlAckPending = 0
- 在判断标头中的 SN 字段是否与上次成功收到的数据包的 SN 字段相同时，将收到的 SN 位与 pParams->seqStat.lastRxSn 进行比较。
- 如果收到的数据包具有正确的 CRC 且可放入 RX 缓冲区，则收到的 SN 应存储到 pParams->seqStat.lastRxSn 中。如果该数据包为 LL 控制包 (LLID = 11b) 且不应被忽略，则 pParams->seqStat.bLlCtrlAckPending 应设置为 1，并引发 Rx_Ctrl 中断。
- 如果收到的数据包具有正确的 CRC 且收到的 NESN 与 pParams->seqStat.lastTxSn 不同，则 pParams->seqStat.nextTxSn 应设置为收到的 NESN 值（无论该数据包是否能放入 RX 缓冲区）。
- 如果 pParams->seqStat.bFirstPkt = 0：
  - 如果在收到数据包后 pParams->seqStat.nextTxSn 已更新且变得与 pParams->seqStat.lastTxSn 不同，则 nNack 应设置为 pParams->maxNack，并引发 Tx_Ack 中断。
  - 否则，nNack 应递减。
  - 如果在收到数据包后 pParams->seqStat.nextTxSn 已更新且变得与 pParams->seqStat.lastTxSn 不同，并且 pParams->seqStat.bAutoEmpty = 0，则当前 TX 队列条目应完成，下一个条目应设为活动条目（请参阅节 25.3.2.7.2），并引发 Tx_Entry_Done 中断。如果 pParams->seqStat.bLlCtrlTx = 1，则应引发 Tx_Ctrl_Ack 中断，并将 pParams->seqStat.bLlCtrlAckRx 设置为 1。
  - 如果在收到数据包后 pParams->seqStat.nextTxSn 已更新且变得与 pParams->seqStat.lastTxSn 不同，则应将 pParams->seqStat.bAutoEmpty 设置为 0。
- 如果 TX 队列中没有可用的缓冲区，或者在将要发送数据包时 pParams->seqStat.nextTxSn 等于 pParams->seqStat.lastTxSn 且 pParams->seqStat.bAutoEmpty = 1，则应传输一个自动空包。不应从 TX 队列读取任何内容。传输的数据包应从 TX 队列的第一个条目中读取。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

- 在传输的数据包的标头中，SN 位应设置为 pParams->seqStat.nextTxSn 的值，NESN 位应设置为 pParams->seqStat.lastRxSn 的取反值。
- 数据包传输后：
  - 如果 pParams->seqStat.nextTxSn 等于 pParams->seqStat.lastTxSn，则应引发 Tx_Retrans 中断。
  - 如果发送后 pParams->seqStat.nextTxSn 与 pParams->seqStat.lastTxSn 不同且传输的数据包的 LLID = 11b，则应引发 Tx_Ctrl 中断。
  - 如果传输后 pParams->seqStat.nextTxSn 与 pParams->seqStat.lastTxSn 不同且 pParams->seqStat.bLlCtrlAckPending = 1，则应引发 Rx_Ctrl_Ack 中断。
  - 如果传输后 pParams->seqStat.nextTxSn 与 pParams->seqStat.lastTxSn 不同且 pParams->seqStat.bLlCtrlAckRx = 1，则应引发 Tx_Ctrl_Ack_Ack 中断。
  - pParams->seqStat.lastTxSn 应设置为 pParams->seqStat.nextTxSn 的值。
  - 如果数据包不是从 TX 队列读取的，则 pParams->seqStat.bAutoEmpty 应设置为 1，否则设置为 0。
  - 如果传输的数据包的 LLID = 11，则 pParams->seqStat.bLlCtrlTx 应设置为 1，否则设置为 0。
  - pParams->seqStat.firstPkt、pParams->seqStat.bLlCtrlAckPending 和 pParams->seqStat.bLlCtrlAckRx 均应设置为 0。
  - 应引发 Tx_Done 中断。
  - nPkt 应递减。

当按照前述列表引发中断时，对应的计数器（如表 25-129 所示）应递增。

在传输的数据包的标头中，应根据以下规则设置 MD 位：

- 如果传输队列为空，或者正在传输的数据包是传输队列的最后一个数据包，则 MD 应设置为 0。
- 如果 pParams->endTrigger 中描述的触发条件已发生，则 MD 应设置为 0。
- 如果计数器 nPkt 为 1，则 MD 应设置为 0。
- 否则，MD 应设置为 1。

pOutput 结构包含计数器，无线电 CPU 应按照前述列表以及表 25-129 对这些计数器进行更新。无线电 CPU 不会初始化这些字段，因此当需要复位计数器时，必须由系统 CPU 完成此操作。除计数器外，无线电 CPU 还应设置以下字段：

- 如果收到数据包，则 lastRssi 设置为该数据包的 RSSI。
- 对于从器件命令，timeStamp 应设置为第一个收到的数据包开始的时间戳（如果收到任何数据包）。bValidTimeStamp 字段应在操作开始时设置为 0，如果收到数据包并因此写入 timeStamp，则设置为 1。

为确保正确操作，命令开始时的 pParams->seqStat 值应与同一连接的上一个操作结束时的值相同。在同一连接上运行的命令之间，TX 队列也应保持不变（只能向队列附加数据包）。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.8.6 从器件命令

从无线电操作通过 CMD_BLE_SLAVE 或 CMD_BLE5_SLAVE 命令启动。在命令结构中，它应具有一个在表 25-96 或表 25-103 中定义的类型的 pParams 参数和一个在表 25-109 中定义的类型的 pOutput 参数。操作从接收开始。参数 pParams->timeoutTrigger 和 pParams->timeoutTime 定义在解调器未找到同步的情况下结束操作的时间。startTrigger 和 pParams->timeoutTrigger 共同定义从器件的接收窗口。

应以特殊方式处理从器件上新 LL 连接的第一个接收到的数据包。这由系统 CPU 发出信号，方法是在开始新连接的第一个从操作时将 pParams->seqStat.bFirstPkt 设置为 1。设置此标志时，接收到的数据包不应视为先前发送的数据包的 ACK 或 NACK。传输数据包时，无线电 CPU 应清除 pParams->seqStat.bFirstPkt。

无线电 CPU 应将无线电操作的第一个所接收数据包的时间戳写入 pOutput->timeStamp。系统 CPU 可将采集的时间用作锚点，以计算未来从器件命令的开始。此时间也应定义为*事件 1*。这可以用于为后续链接的操作计时。如果未找到锚点，则事件 1 将是从器件操作开始的时间。

如果接收到带有 CRC 错误的数据包，则如果同一无线电操作中的前一个数据包也带有 CRC 错误，则无线电 CPU 应结束无线电操作（请参阅表 25-130）。否则，如果接收到数据包、无线电 CPU 应启动发送器并从 TX 队列进行发送，或者如果 TX 队列为空，则发送自动空的数据包。传输可以是重传。除非操作以表 25-130 中列出的标准结束，否则接收器将在传输完成后启动。

从属操作应以表 25-130 中列出的原因之一结束。操作结束后，命令结构的状态字段会指示操作结束的原因。在所有情况下，都会产生 Command_Done 中断。在每种情况下，结果都指示为 TRUE、FALSE 或 ABORT，这将决定下一个操作。

表 25-130. 从器件运行结束

条件	状态代码	结果
成功接收到标头的 MD 位为 0 的数据包后，MD = 0 的已发送数据包	BLE_DONE_OK	TRUE
在接收到不适合 RX 队列的数据包后，MD = 0 的已发送数据包	BLE_DONE_OK	TRUE
已完成发送数据包，nPkt 计数为 0	BLE_DONE_OK	TRUE
由 pParams->timeoutTrigger 指示的触发发生在启动命令后获得解调器同步之前	BLE_DONE_RXTIMEOUT	FALSE
发送后在接收操作时未获得同步	BLE_DONE_NOSYNC	TRUE
在相同操作中接收的两个后续数据包存在 CRC 错误	BLE_DONE_RXERR	TRUE
在内部计数器 nNack 倒计时至 0 后完成数据包发送	BLE_DONE_MAXNACK	TRUE
在观察到 pParams->endTrigger 指示的触发后完成数据包发送	BLE_DONE_ENDED	FALSE
在观察到 CMD_STOP 后完成数据包发送	BLE_DONE_STOPPED	FALSE
收到了 CMD_ABORT	BLE_DONE_ABORT	ABORT
通道值非法	BLE_ERROR_PAR	ABORT
TX 数据条目长度字段具有非法值	BLE_ERROR_PAR	ABORT

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.8.7 主器件命令

CMD_BLE_MASTER 或者 CMD_BLE5_MASTER 命令会启动主无线电操作。在命令结构中，它应具有一个在表 25-97 中定义的类型的 *pParams* 参数和一个在表 25-109 中定义的类型的 *pOutput* 参数。操作从传输开始。每次传输后，接收器都会启动。

主操作应以表 25-131 中列出的原因之一结束。操作结束后，命令结构的状态字段指示操作结束的原因。在所有情况下，都会产生 Command_Done 中断。在每种情况下，都会指示 TRUE、FALSE 或 ABORT 的结果，这些结果将决定下一个操作。

表 25-131. 主器件操作结束

条件	状态代码	结果
在发送 MD = 0 的数据包后，成功接收到 MD = 0 的数据包	BLE_DONE_OK	TRUE
在发送 MD = 0 的数据包后，接收到的数据包不适合 RX 队列	BLE_DONE_OK	TRUE
在 nPkt 计数到 0 后收到数据包	BLE_DONE_OK	TRUE
发送后在接收操作时未获得同步	BLE_DONE_NOSYNC	TRUE
在相同操作中接收的两个后续数据包存在 CRC 错误	BLE_DONE_RXERR	TRUE
接收到数据包后，内部计数器 nNack 倒计时至 0	BLE_DONE_MAXNACK	TRUE
在观察到 pParams->endTrigger 指示的触发后收到了一个数据包	BLE_DONE_ENDED	FALSE
在观察 CMD_STOP 之后收到了一个数据包	BLE_DONE_STOPPED	FALSE
收到了 CMD_ABORT	BLE_DONE_ABORT	ABORT
通道值非法	BLE_ERROR_PAR	ABORT
TX 数据条目长度字段具有非法值	BLE_ERROR_PAR	ABORT

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.8.8 传统广播器

在任何类型的广播器操作开始时，无线电 CPU 会等待启动触发条件，然后根据命令结构的通道参数设置频率。通道参数不允许在 0 – 36 范围内，因为该范围为数据通道。无线电 CPU 设置广播通道访问地址，并使用 CRC 初始化值 0x55 5555。按照白化参数中的定义设置白化器。然后，无线电 CPU 配置发送器。除了不可连接的广播器外，该操作在发送后会继续进入接收状态，并且如果收到 SCAN_REQ，可能会再发送一次 SCAN_RSP。

在低功耗 Bluetooth 模式下，广播通常在所有三个广播通道上进行。为实现此目的，可以使用 pNextOp 参数将三个命令结构链接起来。通常，所有通道的参数和输出结构可以相同。

发送的第一个数据包始终是 ADV*_IND 数据包。该数据包包含标头、广播器地址和广播数据（定向广播中使用的 ADV_DIRECT_IND 数据包除外）。无线电 CPU 按如下方式构造这些数据包（ADV_DIRECT_IND 数据包如节 25.8.8.2 所述）。

- 在标头中，PDU 类型位如表 25-132 所示。
- TXAdd 位如 pParams->advConfig.deviceAddrType 所示。
- 长度根据广播数据的大小计算，即 pParams->advLen + 6。
- RXAdd 位未使用且为 0，RFU 位也是。
- 有效载荷以 6 字节的器件地址开始，该地址从 pParams->pDeviceAddress 读取。
- 有效载荷的其余部分从 pParams->pAdvData 缓冲区读取（如果 pParams->advLen 非零）。

表 25-132. 不同广播器命令的 PDU 类型

命令	广播数据包类型	标头中 PDU 类型位的值
CMD_BLE_ADV	ADV_IND	0000b
CMD_BLE_ADV_DIR	ADV_DIRECT_IND	0001b
CMD_BLE_ADV_NC	ADV_NONCONN_IND	0010b
CMD_BLE_ADV_SCAN	ADV_SCAN_IND	0110b

除不可连接的广播器外，在发送完 ADV*_IND 数据包后，接收器应启动。根据广播器操作的类型，接收器应侦听 SCAN_REQ 和（或）CONNECT_IND（在相关文档所列的 Bluetooth 4.0、4.1 和 4.2 规范中称为 *CONNECT_REQ*）。如果解调器获得同步，应在收到标头时对其进行检查，如果它不是 SCAN_REQ 或 CONNECT_IND 消息，则应立即停止解调器。

SCAN_REQ 或 CONNECT_IND 消息接收到 pParams->pRxQ 给出的 RX 队列中，如节 25.10.4.1 所述。bCrcErr 和 bIgnore 位根据 CRC 结果和收到的消息进行设置。消息中的 AdvA 字段和收到的标头的 TxAdd 位会分别与 pParams->pDeviceAddress 数组和 pParams->advConfig.deviceAddrType 进行比较，以确定该消息是否发往此广播器。然后，根据 pParams->advConfig.advFilterPolicy 给出的广播过滤策略，将收到的 ScanA 或 InitA 字段以及收到的标头的 RxAdd 位与白名单（请参阅节 25.8.14）进行核对，但对于定向广播器，将收到的标头与对端地址进行比较（请参阅节 25.8.8.2）。如果（由 pParams->advConfig.rpaMode 给出的）可解析私有地址 (RPA) 模式为非零值，则会执行额外检查，以确定对端地址是否为可解析私有地址。如果收到的 TxAdd 为 1，且收到的 ScanA 或 InitA 字段的两个最高有效位为 01b，则该地址为 RPA。如果是这样，无论过滤策略如何，都会执行白名单核对。根据收到的数据包，应采取的操作如表 25-133 所述，其中每个操作的定义（包括将在 bCrcErr 和 bIgnore 上使用的值）在表 25-134 中给出。如果 pParams->advConfig.bStrictLenFilter 为 1，则只有符合蓝牙低耗能规范的长度字段才被视为有效。对于 SCAN_REQ，这意味着长度字段为 12；对于 CONNECT_IND，这意味着长度字段为 34。如果 pParams->advConfig.bStrictLenFilter 为 0，则收到的所有长度字段小于或等于广播器数据包最大长度的数据包均被视为有效。如果长度无效，则应停止接收器。

表 25-133. 广播器根据收到的数据包采取的操作

PDU 类型	CRC 结果	高级类型	有效长度	AdvA 匹配自己的地址	过滤策略	RPA 模式	可解析私有地址	白名单中存在 ScanA 或 InitA	操作编号
SCAN_REQ	OK	C、S	是	否	X	X	X	X	1
SCAN_REQ	OK	C、S	是	是	1 或 3	X	X	否	1
SCAN_REQ	OK	C、S	是	是	1 或 3	X	X	是	2

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-133. 广播器根据收到的数据包采取的操作（续）

PDU 类型	CRC 结果	高级类型	有效长度	AdvA 匹配自己的地址	过滤策略	RPA 模式	可解析私有地址	白名单中存在 ScanA 或 InitA	操作编号
SCAN_REQ	OK	C、S	是	是	0 或 2	0	X	X	2
SCAN_REQ	OK	C、S	是	是	0 或 2	1	否	X	2
SCAN_REQ	OK	C、S	是	是	0 或 2	1	是	否	1
SCAN_REQ	OK	C、S	是	是	0 或 2	1	是	是	2
SCAN_REQ	NOK	C、S	是	X	X	X	X	X	3
SCAN_REQ	X	C、S	否	X	X	X	X	X	5
SCAN_REQ	X	D	X	X	X	X	X	X	5
CONNECT_IND	OK	C、D	是	否	X	X	X	X	1
CONNECT_IND	OK	C、D	是	是	2 或 3	X	X	否	1
CONNECT_IND	OK	C、D	是	是	2 或 3	X	X	是	4
CONNECT_IND	OK	C、D	是	是	0 或 1	0	X	X	4
CONNECT_IND	OK	C、D	是	是	0 或 1	1	否	X	4
CONNECT_IND	OK	C、D	是	是	0 或 1	1	是	否	1
CONNECT_IND	OK	C、D	是	是	0 或 1	1	是	是	4
CONNECT_IND	NOK	C、D	是	X	X	X	X	X	3
CONNECT_IND	X	C、D	否	X	X	X	X	X	5
CONNECT_IND	X	S	X	X	X	X	X	X	5
其他	X	X	X	不适用	X	X	不适用	不适用	5
未收到数据包	不适用	X	不适用	不适用	X	X	不适用	不适用	5

表 25-134. 要对收到的数据包执行的操作说明

操作编号	bCrcErr	bIgnore	说明
1	0	1	以 BLE_DONE_OK 状态结束操作
2	0	0	发送 SCAN_RSP 消息
3	1	0	以 BLE_DONE_RXERR 状态结束操作
4	0	0	以 BLE_DONE_CONNECT 或 BLE_DONE_CONNECT_CHSEL0 状态结束操作
5	—	—	立即停止接收器，并以 BLE_DONE_NOSYNC 状态结束操作

如果接收到长度为 12（或更短）的 SCAN_REQ 数据包，应将其视为空数据包。这意味着，如果 pParams->rxConfig.bAutoflushEmpty 为 1，且 bCrcErr 和 bIgnore 均为 0，则该数据包会从 RX 缓冲区中移除。如果数据包具有 bIgnore 或 bCrcErr 标记，也可根据 pParams->rxConfig 中的位将其移除。

如果正在接收的数据包无法放入 RX 队列，则该数据包会被接收到末尾，但收到的字节不会存储。如果根据 pParams->rxConfig 中的位，该数据包通常不会从 RX 队列中丢弃，则命令应结束。

如果正确接收到 CONNECT_IND 数据包（请参阅表 25-134 中的操作 4）且 pParams->advConfig.chSel 为 1，则检查收到的标头的 ChSel 位。如果该位为 0（表示对端不支持通道选择算法 2），则状态设置为 BLE_DONE_CONNECT_CHSEL0 而非 BLE_DONE_CONNECT。。

如果（根据表 25-133 和表 25-134）下一个操作是发送 SCAN_RSP 数据包，无线电 CPU 应启动发送器来发送该数据包。该数据包由标头、广播器地址和广播数据组成。

无线电 CPU 应按如下方式构建这些数据包：

- 在标头中，PDU 类型位应为 0100b。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

- TxAdd 位应与 pParams->advConfig.deviceAddrType 中一致。
- 长度应根据扫描响应数据的大小计算，即应为 pParams->scanRspLen + 6。
- RxAdd 和 ChSel 位未使用，应为 0。
- RFU 位应为 0。
- 有效载荷应以 6 字节的器件地址开始，该地址应从 pParams->pDeviceAddress 读取。
- 如果是这样，有效载荷的其余部分应从 pParams->pScanRspData buffer 缓冲区读取。
- 发送 SCAN_RSP 后，命令就应结束。

结束操作的触发条件由 pParams->endTrigger 设置。如果发生该参数定义的触发条件，无线电操作应继续直至完成，但结束后的状态码应为 BLE_DONE_ENDED，结果应为 FALSE。可使用该触发条件来停止执行，而不利用命令结构中的条件继续执行链中的下一个操作。如果无线电 CPU 收到即时命令 CMD_STOP，其含义与发生结束触发条件时相同（但结束后的状态代码应为 CMD_DONE_STOPPED）。

输出结构 pOutput 包含的字段提供有关正在运行的命令的信息。无线电 CPU 不会初始化这些字段，因此当需要复位计数器时，必须由系统 CPU 完成此操作。无线电 CPU 将按照下方列表所述更新这些字段。该列表还指明了何时应在系统 CPU 中引发中断。

- 当 ADV*_IND 数据包发送后，nTxAdvInd 递增并引发 Tx_Done 中断。
- 如果发送了 SCAN_RSP 数据包，则之后 nTxScanRsp 递增并引发 Tx_Done 中断。
- 如果收到 SCAN_REQ 数据包，CRC 正确且 bIgnore 标志被清除，则 nRxScanReq 递增。如果有效载荷长度为 12 或更短，则引发 Rx_Empty 中断。如果有效载荷长度大于 12，则引发 Rx_Ok 中断。
- 如果收到 CONNECT_IND 数据包，CRC 正确且 bIgnore 标志被清除，则 nRxConnectReq 递增并引发 Rx_OK 中断。
- 如果收到数据包但 CRC 错误，则 nRxNok 递增并引发 Rx_Nok 中断。
- 如果收到数据包且 bIgnore 标志置位，则 nRxIgnored 递增并引发 Rx_Ignored 中断。
- 如果收到无法放入 RX 队列的数据包，则 nRxBufFull 递增并引发 Rx_Buf_Full 中断。
- 如果收到数据包，则 lastRssi 设置为该数据包的 RSSI。
- 如果收到数据包，则 timeStamp 设置为该数据包开始的时间戳。对于 CONNECT_IND 数据包，该时间戳可用于计算第一个数据包的锚点。
- 如果在收到数据包后 RX 队列中的第一个 RX 数据条目将状态更改为“已完成”，则引发 Rx_Entry_Done 中断。

25.8.8.1 可连接的非定向广播器命令

可连接的非定向广播器操作由 CMD_BLE_ADV 命令启动。在命令结构中，它应具有一个类型为表 25-98 中定义的 pParams 参数和一个类型为表 25-110 中定义的 pOutput 参数。该操作以传输开始，并按照节 25.8.8 中的描述运行。

可连接的非定向广播器操作应以表 25-135 中列出的状态之一结束。操作结束后，命令结构的状态字段指示操作结束的原因。在所有情况下，都会产生 Command_Done 中断。在每种情况下，都会指示 TRUE、FALSE 或 ABORT 的结果，这些结果将决定下一个操作。

表 25-135. 可连接的非定向广播器操作结束

条件	状态代码	结果
运行接收器后执行了操作 1	BLE_DONE_OK	真
执行了操作 2 并发送了 SCAN_RSP	BLE_DONE_OK	真
运行接收器后执行了操作 3	BLE_DONE_RXERR	真
运行接收器后执行了操作 4（如果 pParams->advConfig.chSel = 0 或接收到的 CONNECT_IND 的 ChSel 位为 1）	BLE_DONE_CONNECT	否
运行接收器后执行了操作 4（如果 pParams->advConfig.chSel = 1 或接收到的 CONNECT_IND 的 ChSel 位为 0）	BLE_DONE_CONNECT_CHSEL0	否
运行接收器后执行了操作 5	BLE_DONE_NOSYNC	真
观察到由 pParams->endTrigger 指示的触发，然后执行了操作 1、2、3 或 5	BLE_DONE_ENDED	否
观察到 CMD_STOP，然后执行了操作 1、2、3 或 5	BLE_DONE_STOPPED	否

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-135. 可连接的非定向广播器操作结束（续）

条件	状态代码	结果
收到了 CMD_ABORT	BLE_DONE_ABORT	ABORT
RX 缓冲区中没有空间来存储接收到的数据包	BLE_ERROR_RXBUF	否
通道值非法	BLE_ERROR_PAR	ABORT
广播数据或扫描响应数据长度字段的值非法	BLE_ERROR_PAR	ABORT

25.8.8.2 可连接的定向广播器命令

可连接的定向广播器操作通过 CMD_BLE_ADV_DIR 命令启动。在命令结构中，它应具有一个在表 25-98 中定义的类型的 pParams 参数和一个在表 25-110 中定义的类型的 pOutput 参数。该操作从传输开始，并按照节 25.8.8 中所述运行，进行了一些修改，如以下列表所述。

对于定向广播器，pParams->pWhiteList 应指向仅包含要连接的设备的设备地址的缓冲器。对等方的地址类型应在 pParams->advConfig.peerAddrType 中给出。

第一个发送操作将发送 ADV_DIRECT_IND 数据包。无线电 CPU 应按如下方式构建此数据包：

- 在标题中，PDU 类型位应为 0001b，如表 25-132 中所示。
- TxAdd 位应与 pParams->advConfig.deviceAddrType 中的相同。
- RxAdd 位应与 pParams->advConfig.peerAddrType 中的一样（如果 pParams->pWhiteList 的最低有效位为 1，则 RxAdd 位反转，请参阅节 25.8.17）。
- 如果 pParams->advConfig.chSel 为 1，则 ChSel 位应为 1；否则，ChSel 应为 0。
- 长度应根据广播数据的大小计算得出，这意味着它应为 pParams->advLen + 12。
- RFU 位应为 0。
- 有效载荷应以 6 字节器件地址开始，该地址应从 pParams->pDeviceAddress 读取，然后是从 pParams->pWhiteList 读取的 6 字节对等地址。
- 根据低功耗蓝牙规范，不应再有有效载荷，但可以通过将 pParams->advLen 设置为非零值来构建不符合要求的消息。如果是这样，则应从 pParams->pAdvData 缓冲器读取有效载荷的其余部分。

在按照节 25.8.8 中所述传输 ADV_DIRECT_IND 数据包后启动接收器，并按照其中所述处理收到的数据包。接收到的 TargetA 字段和 TxAdd 位将分别根据 pParams->pWhiteList 和 pParams->advConfig.peerAddrType 进行检查，而不是根据白名单进行检查。

定向广播器操作应以表 25-136 中列出的状态之一结束。操作结束后，命令结构的状态字段指示操作结束的原因。在所有情况下，都会产生 Command_Done 中断。在每种情况下，都会指示 TRUE、FALSE 或 ABORT 的结果，这些结果将决定下一个操作。

表 25-136. 定向广播器操作结束

条件	状态代码	结果
运行接收器后已执行操作 1	BLE_DONE_OK	TRUE
运行接收器后已执行操作 3	BLE_DONE_RXERR	TRUE
如果 pParams->advConfig.chSel = 0 或接收到的 CONNECT_IND 的 ChSel 位为 1，则在运行接收器后执行操作 4	BLE_DONE_CONNECT	FALSE
如果 pParams->advConfig.chSel = 1 并且接收到的 CONNECT_IND 的 ChSel 位为 0，则在运行接收器后执行了操作 4	BLE_DONE_CONNECT_CHSEL0	FALSE
运行接收器后已执行操作 5	BLE_DONE_NOSYNC	TRUE
观察到的触发由 pParams->endTrigger 指示，然后执行操作 1、3 或 5	BLE_DONE_ENDED	FALSE
观察到 CMD_STOP，然后执行操作 1、3 或 5	BLE_DONE_STOPPED	FALSE
收到了 CMD_ABORT	BLE_DONE_ABORT	ABORT
RX 缓冲器中没有空间来存储接收到的数据包	BLE_ERROR_RXBUF	FALSE
通道值非法	BLE_ERROR_PAR	ABORT

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-136. 定向广播器操作结束 （续）

条件	状态代码	结果
广播数据长度字段具有非法值	BLE_ERROR_PAR	ABORT

25.8.8.3 不可连接的广播器命令

不可连接的广播器操作由 CMD_BLE_ADV_NC 命令启动。在命令结构中，它应具有一个在表 25-98 中定义的类型的 pParams 参数和一个在表 25-110 中定义的类型的 pOutput 参数。操作从变速箱开始，并按节 25.8.8 中所述进行操作。传输 ADV_NONCONN_IND 数据包后，操作应在没有任何接收操作的情况下结束。

不可连接的广播器操作应以表 25-137 中列出的状态之一结束。操作结束后，命令结构的状态字段指示操作结束的原因。在所有情况下，都会产生 Command_Done 中断。在每种情况下，都会指示 TRUE、FALSE 或 ABORT 的结果，这些结果将决定下一个操作。

表 25-137. 不可连接的广播器操作结束

条件	状态代码	结果
发送了 ADV_NONCONN_IND	BLE_DONE_OK	TRUE
观察到的触发信号由 pParams->endTrigger 指示，然后完成 ADV_NONCONN_IND 发送	BLE_DONE_ENDED	FALSE
观察到 CMD_STOP，然后完成 ADV_NONCONN_IND 发送	BLE_DONE_STOPPED	FALSE
收到了 CMD_ABORT	BLE_DONE_ABORT	ABORT
通道值非法	BLE_ERROR_PAR	ABORT
广播数据长度字段具有非法值	BLE_ERROR_PAR	ABORT

25.8.8.4 可扫描非定向广播器命令

可扫描非定向广播器操作由 CMD_BLE_ADV_SCAN 命令启动。在命令结构中，它应包含具有表 25-98 中定义类型的 pParams 参数和具有表 25-110 中定义类型的 pOutput 参数。该操作从发送操作开始，运行过程如节 25.8.8 所述。

可扫描非定向广播器操作应以表 25-138 中列出的状态之一结束。操作结束后，命令结构中的状态字段指示操作结束的原因。在所有情况下，都会引发 Command_Done 中断。每种情况都会指示 TRUE、FALSE 或 ABORT 结果，这将决定下一步的操作。

表 25-138. 可扫描非定向广播器操作结束

条件	状态代码	结果
运行接收器后执行了操作 1	BLE_DONE_OK	真
执行了操作 2 并传输了 SCAN_RSP	BLE_DONE_OK	真
运行接收器后执行了操作 3	BLE_DONE_RXERR	真
运行接收器后执行了操作 5	BLE_DONE_NOSYNC	真
观察到 pParams->endTrigger 所指示的触发条件，然后执行了操作 1、2、3 或 5	BLE_DONE_ENDED	否
观察到 CMD_STOP，然后执行了操作 1、2、3 或 5	BLE_DONE_STOPPED	否
收到了 CMD_ABORT	BLE_DONE_ABORT	ABORT
RX 缓冲区中没有空间来存储收到的数据包	BLE_ERROR_RXBUF	否
通道值非法	BLE_ERROR_PAR	ABORT
广播数据或扫描响应数据长度字段具有非法值	BLE_ERROR_PAR	ABORT

25.8.9 Bluetooth® 5 广播器命令

在扩展通道或次级通道广播器开始时，无线电 CPU 应等待启动触发。然后，它应根据命令结构的通道参数对频率进行编程。通道参数必须指示广播器通道的正确类型，因此对于扩展广播器，该参数必须为 37、38 或 39，对于次级通道广播器，该参数必须在 0 – 36 范围内。无线电 CPU 还应设置 phyMode.mainMode 中给出的 PHY 模式。此外，它应设置广播通道访问地址并使用 CRC 初始化值 0x55 5555。白化器须按照白化参数的定义进行配

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

置。然后，无线电 CPU 应配置发送器。对于次级通道广播器，如果发送的数据包可扫描或可连接，则该操作将在发送后接收，如果接收到请求数据包，则随后可能会传输响应数据包。

在低功耗蓝牙中，广播通常在全部三个主要广播通道上完成，然后在一个或多个次级通道上进行广播。要设置这一点，可以使用 pNextOp 参数链接四个（或更多）命令结构。三个扩展广播器命令的参数结构可以相同，并且所有通道的输出结构可以相同。

发送的数据包始终使用通用的扩展广播格式。这样的数据包由无线电 CPU 基于如表 25-105 中给出的描述符中的信息构造。数据包处理的详细信息在节 25.8.9.1 中进行了描述。第一个传输的数据包由 pParams->pAdvPkt 指向的描述符给出。

如果传输的数据包含有 AuxPtr 字段，则无线电 CPU 应修改偏移单位和辅助偏移（见相关文档中列出的蓝牙系统规范 5.0 版）字段，使其指向在 pParams->auxPtrType 和 pParams->auxPtrTime 给出的时间传输的数据包。pParams->auxPtrType 允许的值与命令中通常使用的触发器相同，但只允许 TRIG_ABSTIME、TRIG_NEVER 和各种相对时间；TRIG_NEVER 意味着 AuxPtr 不应由无线电 CPU 进行修改。无线电 CPU 应计算从发送数据包开始到给定时间的时间，并找到可用于表示该时间的最小偏移单位。辅助偏移应设置为时间差除以向下舍入的单位。

节 25.8.9.2 和节 25.8.9.3 介绍了 CMD_ADV_EXT 和 CMD_ADV_AUX 的操作详情。

25.8.9.1 通用扩展广播数据包

使用通用扩展广播格式传输的数据包包含以下内容：

- 标头
- 扩展的标头长度和广播器模式
- 可选的扩展标头
- 可选的有效载荷

在标头中，所有相关类型的 PDU 类型均设置为 0111b，但 AUX_CONNECT_RSP 除外，其 PDU 类型为 1000b。TxAdd 和 RxAdd 位按如下所述自动插入。长度由无线电 CPU 根据扩展标头长度和有效载荷长度计算得出。扩展标头长度和广播器模式字段由 extHdrInfo 的相应字段插入。如果扩展标头长度大于 0，则扩展标头的第一个字节是扩展标头标志。该字节由无线电 CPU 从 extHdrFlags 插入。扩展标头的其余部分位于 pExtHeader 所指向的缓冲区中。

缓冲区中可以省略扩展标头中的地址（在这种情况下，pExtHdr 指向的缓冲区应短 6 或 12 字节）。如果 extHdrConfig.bSkipAdvA 为 1 且 extHdrInfo 中的 AdvA 位为 1，则广播器地址从参数结构中的 pDeviceAddress 所指向的缓冲区插入。在这种情况下，标头的 TxAdd 位根据参数结构位 advConfig.deviceAddrType 设置。如果 extHdrConfig.bSkipAdvA 为 0 且 extHdrInfo 中的 AdvA 位为 1，则标头的 TxAdd 位根据 extHdrConfig.deviceAddrType 设置。如果 extHdrInfo 中的 AdvA 位为 0，则标头的 TxAdd 位为 0。类似地，对于辅助通道上的定向广播数据包，如果 extHdrConfig.bSkipTargetA 为 1 且 extHdrInfo 中的 TargetA 位为 1，则目标地址从参数结构中的 pWhiteList 所指向的缓冲区插入。在这种情况下，标头的 RxAdd 位根据参数结构位 advConfig.peerAddrType 设置（如果 pWhiteList 的 LSB 为 1，则取反，如节 25.8.17 中所述）。如果 extHdrConfig.bSkipTargetA 为 0 且 extHdrInfo 中的 TargetA 位为 1，则标头的 RxAdd 位根据 extHdrConfig.targetAddrType 设置。如果 extHdrInfo 中的 TargetA 位为 0，则标头的 RxAdd 位为 0。

对于 AUX_CONNECT_RSP 数据包，TargetA 字段和标头的 RxAdd 位始终从收到的 AUX_CONNECT_REQ 地址插入。如果 extHdrConfig.bSkipTargetA 为 0，pExtHeader 指向的缓冲区应包含目标地址的占位符，而如果 extHdrConfig.bSkipTargetA 为 1，则应省略 TargetA 字段。

结束操作的触发条件由 pParams->endTrigger 设置。如果发生该参数定义的触发条件，无线电操作应继续直至完成，但结束后的状态码应为 BLE_DONE_ENDED，结果应为 FALSE。例如，可使用该触发条件来停止执行，而不利用命令结构中的条件继续执行链中的下一个操作。如果无线电 CPU 收到即时命令 *CMD_STOP*，其含义与发生结束触发条件时相同，但结束后的状态代码应为 CMD_DONE_STOPPED。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

输出结构 pOutput 包含的字段提供有关正在运行的命令的信息。无线电 CPU 不会初始化这些字段，因此当需要复位计数器时，必须由系统 CPU 完成此操作。无线电 CPU 将按照下方列表所述更新这些字段。该列表还指明了何时应在系统 CPU 中引发中断。

- 当 ADV_EXT_IND、AUX_ADV_IND 或 AUX_CHAIN_IND 数据包传输后，nTxAdvInd 递增并引发 Tx_Done 中断。
- 如果传输了 AUX_SCAN_RSP 数据包，则之后 nTxScanRsp 递增并引发 Tx_Done 中断。
- 如果传输了 AUX_CONNECT_RSP 数据包，则之后 nTxConnectRsp 递增并引发 Tx_Done 中断。
- 如果收到 AUX_SCAN_REQ 数据包，CRC 正确且 bIgnore 标志被清除，则 nRxScanReq 递增。如果有效载荷长度为 12 或更短，则引发 Rx_Empty 中断。如果有效载荷长度大于 12，则引发 Rx_Ok 中断。
- 如果收到 AUX_CONNECT_REQ 消息，CRC 正确且 bIgnore 标志被清除，则 nRxConnectReq 递增并引发 Rx_OK 中断。
- 如果收到数据包但 CRC 错误，则 nRxNok 递增并引发 Rx_Nok 中断。
- 如果收到数据包且 bIgnore 标志置位，则 nRxIgnored 递增并引发 Rx_Ignored 中断。
- 如果收到无法放入 RX 队列的数据包，则 nRxBufFull 递增并引发 Rx_Buf_Full 中断。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

- 如果收到数据包，则 lastRssi 设置为该数据包的 RSSI。
- 如果收到数据包，则 timeStamp 设置为该数据包开始的时间戳。对于 AUX_CONNECT_REQ 消息，该时间戳可用于计算第一个数据包的锚点
- 如果在收到数据包后 RX 队列中的第一个 RX 数据条目将状态更改为“已完成”，则引发 Rx_Entry_Done 中断。

25.8.9.2 扩展的广播器命令

扩展的广播器操作由 CMD_BLE5_ADV_EXT 命令启动。在命令结构中，它应具有一个在表 25-105 中定义的类型的 pParams 参数和一个在表 25-110 中定义的类型的 pOutput 参数。该操作包括在主要广播通道上传输一个数据包。传输 ADV_EXT_IND 后，操作应在没有任何接收操作的情况下结束。

扩展广播器操作应以表 25-139 中列出的状态之一结束。操作结束后，命令结构的状态字段指示操作结束的原因。在所有情况下，都会产生 Command_Done 中断。在每种情况下，都会指示 TRUE、FALSE 或 ABORT 的结果，这些结果将决定下一个操作。

表 25-139. 扩展广播器操作结束

条件	状态代码	结果
发送了 ADV_EXT_IND	BLE_DONE_OK	TRUE
观察到的触发信号由 pParams->endTrigger 指示，然后完成 ADV_EXT_IND 发送	BLE_DONE_ENDED	FALSE
观察到 CMD_STOP，然后完成 ADV_EXT_IND 发送	BLE_DONE_STOPPED	FALSE
收到了 CMD_ABORT	BLE_DONE_ABORT	ABORT
参数值非法	BLE_ERROR_PAR	ABORT
Aux 指针目标时间给出的 Aux 偏移量高于可以数字方式表示的该量	BLE_ERROR_AUX	ABORT

25.8.9.3 辅助通道广播器命令

辅助通道广播器操作由 CMD_BLE5_ADV_AUX 命令启动。在命令结构中，它应包含具有表 25-106 中定义类型的 pParams 参数和具有表 25-110 中定义类型的 pOutput 参数。操作以发送开始。

在 AUX_ADV_IND 数据包传输之后，如果所发送的数据包是可连接或可扫描的，则应启动接收器。如果传输的广播数据包是可扫描的，接收器应查找 AUX_CONNECT_REQ 消息。如果传输的广播数据包是可连接的，接收器应查找 AUX_SCAN_REQ 消息。

AUX_SCAN_REQ 或 AUX_CONNECT_REQ 消息接收到 pParams->pRxQ 给出的 RX 队列中，如节 25.10.4.1 所述。bCrcErr 和 bIgnore 位根据 CRC 结果和收到的消息进行设置。消息中的 AdvA 字段连同收到的标头的 TxAdd 位会分别与 pParams->pDeviceAddress 数组和 pParams->advConfig.deviceAddrType 进行比较，以确定该消息是否发往此广播器。收到的 ScanA 或 InitA 字段连同收到的标头的 RxAdd 位按如下方式进行检查：

- 如果 pParams->advConfig.bDirected 为 1，则将 ScanA 或 InitA 与 pParams->pWhiteList 指向的单个地址进行核对，并将 RxAdd 与 pParams->advConfig.peerAddrType 进行核对
- 如果 pParams->advConfig.bDirected 为 0，则根据 pParams->advConfig.advFilterPolicy 给出的广播过滤策略，将收到的 ScanA 或 InitA 字段连同收到的标头的 RxAdd 位，按照节 25.8.14 所述与白名单进行核对。如果由 pParams->advConfig.rpaMode 给出的可解析私有地址 (RPA) 模式为非零值，则会执行额外检查，以确定对端地址是否为可解析私有地址。如果收到的 TxAdd 为 1，且收到的 ScanA 或 InitA 字段的两个最高有效位为 01b，则该地址为 RPA。如果是这样，无论过滤策略如何，都会执行白名单核对。

根据收到的数据包，应采取的操作如表 25-140 和表 25-141 中所述，其中每个操作的定义（包括将在 bCrcErr 和 bIgnore 上使用的值）在表 25-142 中给出。如果 pParams->advConfig.bStrictLenFilter 为 1，则只有符合蓝牙低耗能规范的长度字段才被视为有效。对于 AUX_SCAN_REQ 数据包，这意味着长度字段为 12；对于 AUX_CONNECT_REQ 数据包，这意味着长度字段为 34。如果 pParams->advConfig.bStrictLenFilter 为 0，则收到的所有长度字段小于或等于广播器数据包最大长度（255，但可覆盖）的数据包均被视为有效。如果长度无效，则应停止接收器。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-140. 可扫描广播器根据收到的数据包采取的操作 (extHdrInfo.advMode = 2) (1)

PDU 类型	CRC 结果	定向	有效长度	AdvA 匹配	过滤策略	RPA 模式	可解析私有地址	ScanA 或 InitA 匹配	操作编号
AUX_SCAN_REQ	OK	X	是	否	X	X	X	X	1
AUX_SCAN_REQ	OK	否	是	是	2 或 3	X	X	否	1
AUX_SCAN_REQ	OK	否	是	是	2 或 3	X	X	是	2
AUX_SCAN_REQ	OK	否	是	是	0 或 1	0	X	X	2
AUX_SCAN_REQ	OK	否	是	是	0 或 1	1	否	X	2
AUX_SCAN_REQ	OK	否	是	是	0 或 1	1	是	否	1
AUX_SCAN_REQ	OK	否	是	是	0 或 1	1	是	是	2
AUX_SCAN_REQ	OK	是	是	是	X	X	X	否	1
AUX_SCAN_REQ	OK	是	是	是	X	X	X	是	2
AUX_SCAN_REQ	NOK	X	是	X	X	X	X	X	3
AUX_SCAN_REQ	X	X	否	X	X	X	X	X	5
其他	X	X	X	不适用	X	X	不适用	不适用	5
未收到数据包	不适用	X	不适用	不适用	X	X	不适用	不适用	5

(1) X = 不用考虑。N/A = 不适用。

表 25-141. 可连接广播器根据收到的数据包采取的操作 (extHdrInfo.advMode = 1) (1)

PDU 类型	CRC 结果	定向	有效长度	AdvA 匹配	过滤策略	RPA 模式	可解析私有地址	ScanA 或 InitA 匹配	操作编号
AUX_CONNECT_REQ	OK	X	是	否	X	X	X	X	1
AUX_CONNECT_REQ	OK	否	是	是	2 或 3	X	X	否	1
AUX_CONNECT_REQ	OK	否	是	是	2 或 3	X	X	是	4
AUX_CONNECT_REQ	OK	否	是	是	0 或 1	0	X	X	4
AUX_CONNECT_REQ	OK	否	是	是	0 或 1	1	否	X	4
AUX_CONNECT_REQ	OK	否	是	是	0 或 1	1	是	否	1
AUX_CONNECT_REQ	OK	否	是	是	0 或 1	1	是	是	4
AUX_CONNECT_REQ	OK	是	是	是	X	X	X	否	1
AUX_CONNECT_REQ	OK	是	是	是	X	X	X	是	4
AUX_CONNECT_REQ	NOK	X	是	X	X	X	X	X	3
AUX_CONNECT_REQ	X	X	否	X	X	X	X	X	5
其他	X	X	X	不适用	X	X	不适用	不适用	5
未收到数据包	不适用	X	不适用	不适用	X	X	不适用	不适用	5

(1) X = 不用考虑。N/A = 不适用。

表 25-142. 要对收到的数据包执行的操作说明

操作编号	bCrcErr	bIgnore	说明
1	0	1	以 BLE_DONE_OK 状态结束操作
2	0	0	传输 AUX_SCAN_RSP 消息
3	1	0	以 BLE_DONE_RXERR 状态结束操作
4	0	0	传输 AUX_CONNECT_RSP 消息
5	—	—	立即停止接收器，并以 BLE_DONE_NOSYNC 状态结束操作

如果接收到长度为 12（或更短）的 AUX_SCAN_REQ 数据包，应将其视为空数据包。这意味着，如果 pParams->rxConfig.bAutoflushEmpty 为 1，且 bCrcErr 和 bIgnore 均为 0，则该数据包会从 RX 缓冲区中移除。如果数据包具有 bIgnore 或 bCrcErr 标记，也可根据 pParams->rxConfig 中的位将其移除。

如果正在接收的数据包无法放入 RX 队列，则该数据包会被接收到末尾，但收到的字节不会存储。如果根据 pParams->rxConfig 中的位，该数据包通常不会从 RX 队列中丢弃，则命令应结束。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

如果根据表 25-142 的规定，下一个操作是传输 AUX_SCAN_RSP 或 AUX_CONNECT_RSP 消息，则无线电 CPU 应启动发送器，从 pParams->pRspPkt 中的描述符发送该数据包，如节 25.8.9.1 中所述。

辅助通道广播器操作应以表 25-143 中列出的状态之一结束。操作结束后，命令结构中的状态字段指示操作结束的原因。在所有情况下，都会引发 Command_Done 中断。每种情况都会指示 TRUE、FALSE 或 ABORT 结果，这将决定下一步的操作。

表 25-143. 辅助通道广播器操作结束

条件	状态代码	结果
运行接收器后执行了操作 1	BLE_DONE_OK	真
执行了操作 2 并传输了 AUX_SCAN_RSP	BLE_DONE_OK	真
运行接收器后执行了操作 3	BLE_DONE_RXERR	真
执行了操作 4 并传输了 AUX_CONNECT_RSP	BLE_DONE_CONNECT	否
运行接收器后执行了操作 5	BLE_DONE_NOSYNC	真
观察到 pParams->endTrigger 所指示的触发条件，然后执行了操作 1、2、3 或 5	BLE_DONE_ENDED	否
观察到 CMD_STOP，然后执行了操作 1、2、3 或 5	BLE_DONE_STOPPED	否
收到了 CMD_ABORT	BLE_DONE_ABORT	ABORT
RX 缓冲区中没有空间来存储收到的数据包	BLE_ERROR_RXBUF	否
非法参数	BLE_ERROR_PAR	ABORT
辅助指针目标时间产生的辅助偏移量超出了数值所能表示的范围	BLE_ERROR_AUX	ABORT

25.8.10 扫描器命令

扫描器操作由 CMD_BLE_SCANNER 或 CMD_BLE5_SCANNER 命令启动。在命令结构中，它应具有一个类型为 表 25-99 或 表 25-107 中定义的 pParams 参数，以及一个类型为 表 25-99 或 表 25-113 中定义的 pOutput 参数。在扫描器操作开始时，无线电 CPU 将等待启动触发器。然后，它将根据命令结构的通道参数对频率进行编程。对于 CMD_BLE_SCANNER，不允许通道参数在 0 – 36 范围内。因为它们不是主广播通道。对于 CMD_BLE5_SCANNER，它还应设置 phyMode.mainMode 中给出的 PHY 模式。无线电 CPU 将设置广播通道访问地址，并使用 CRC 初始化值 0x55 5555。白化器将按照白化参数中的定义进行设置。然后，无线电 CPU 将配置接收器。

在调谐到正确的通道后，无线电 CPU 将开始侦听广播通道数据包。如果在解调器上获得同步，则消息将被接收到 RX 队列中。将检查报头，如果它不是广播数据包，则将停止接收并重新开始同步搜索。接受的数据包以及后续操作取决于运行的命令、通道类型和 PHY 模式（请参阅节 25.8.10.1、节 25.8.10.2 和节 25.8.10.3 中的说明）。所有扫描器命令都按照节 25.8.10.5 中的描述结束。

25.8.10.1 扫描器在主通道上接收传统广播数据包

如果在主广播通道上以 1Mbps PHY 运行 CMD_BLE_SCANNER 或 CMD_BLE5_SCANNER，则会接受传统广播数据包（ADV_IND、ADV_NONCONN_IND、ADV_SCAN_IND 和 ADV_DIRECT_IND）。操作描述如下：

- bCrcErr 和 bIgnore 位根据 CRC 结果和收到的消息进行设置。
- 根据 pParams->scanConfig.scanFilterPolicy 给出的扫描过滤策略，将消息中收到的 AdvA 字段连同收到的标头的 TxAdd 位按照节 25.8.14 所述与白名单进行核对。如果由 pParams->scanConfig.rpaMode 给出的可解析私有地址 (RPA) 模式为非零值，则会执行额外检查，以确定对端地址是否为可解析私有地址。
- 如果收到的 TxAdd 为 1，且收到的 AdvA 字段的两个最高有效位为 01b，则该地址为 RPA。如果是这样，无论过滤策略如何，都会执行白名单核对。对 AdvA 的完整检查按照表 25-144 中列出的方式进行。对于 ADV_DIRECT_IND 消息，收到的 TargetA 字段和 RxAdd 位将分别与 pParams->pDeviceAddress 和 pParams->scanConfig.deviceAddrType 进行核对。如果 pParams->scanConfig.rpaFilterPolicy 给出的 RPA 过滤策略为 1，且 InitA 和 RxAdd 指示为 RPA，也会接受数据包。ADV_DIRECT_IND 消息的完整过滤如表 25-145 所示。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

- 根据收到的数据包以及扫描是主动扫描还是被动扫描（由 pParams->scanConfig.bActiveScan 指示），应采取的操作如表 25-146 中所述，其中每个操作的定义（包括将在 bCrcErr 和 bIgnore 上使用的值）在表 25-147 中给出。
- 如果 pParams->scanConfig.bStrictLenFilter 为 1，则只有符合蓝牙低耗能规范的长度字段才被视为有效。对于 ADV_DIRECT_IND，这意味着长度字段为 12；对于其他 ADV*_IND 消息，这意味着长度字段在 6 到 37 之间。
- 如果 pParams->advConfig.bStrictLenFilter 为 0，则收到的所有长度字段小于或等于广播器数据包最大长度的数据包均被视为有效。如果长度无效，则应停止接收器。

表 25-144. 过滤收到的广播器地址

过滤策略	RPA 模式	AdvA 为 RPA	AdvA 匹配 bWIIgn = 1 的白名单条目	AdvA 匹配 bEnable = 1 的白名单条目	AdvA 过滤结果	如果 bAutoWIIgnore = 1，则执行自动忽略
1	X	X	否	否	拒绝	否
1	X	X	否	是	接受	是
1	X	X	是	X	拒绝	否
0	0	X	否	X	接受	否
0	0	X	是	X	拒绝	否
0	1	否	否	X	接受	否
0	1	否	是	X	拒绝	否
0	1	是	否	否	拒绝	否
0	1	是	否	是	接受	是
0	1	是	是	X	拒绝	否

表 25-145. 过滤 ADV_DIRECT_IND 数据包中收到的发起器地址

RPA 过滤策略	TargetA 为 RPA	TargetA 等于器件地址	TargetA 匹配
0	X	否	否
0	X	是	是
1	否	否	否
1	否	是	是
1	是	X	是

表 25-146. 扫描器对收到的数据包执行的操作

PDU 类型	CRC 结果	AdvA 过滤结果	TargetA 匹配	活动扫描	操作编号
ADV_IND	OK	拒绝	不适用	X	1
ADV_IND	OK	接受	不适用	否	2
ADV_IND	OK	接受	不适用	是	3
ADV_IND	NOK	X	不适用	X	4
ADV_SCAN_IND	OK	拒绝	不适用	X	1
ADV_SCAN_IND	OK	接受	不适用	否	2
ADV_SCAN_IND	OK	接受	不适用	是	3
ADV_SCAN_IND	NOK	X	不适用	X	4
ADV_NONCONN_IND	OK	拒绝	不适用	X	1
ADV_NONCONN_IND	OK	接受	不适用	X	2
ADV_NONCONN_IND	NOK	X	不适用	X	4
ADV_DIRECT_IND	OK	拒绝	X	X	1
ADV_DIRECT_IND	OK	接受	否	X	1
ADV_DIRECT_IND	OK	接受	是	X	2
ADV_DIRECT_IND	NOK	X	X	X	4
ADV*_IND 长度无效	X	X	X	X	5

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-146. 扫描器对收到的数据包执行的操作 （续）

PDU 类型	CRC 结果	AdvA 过滤结果	TargetA 匹配	活动扫描	操作编号
其他（另请参阅节 25.8.10.2）	X	X	不适用	X	5

表 25-147. 扫描器对收到的数据包执行的操作说明

操作编号	bCrcErr	bIgnore	说明
1	0	1	继续扫描
2	0	0	继续扫描或以 BLE_DONE_OK 状态结束操作
3	0	0	执行退避流程，在适用时发送 SCAN_REQ 并接收 SCAN_RSP。然后继续扫描或结束操作
4	1	0	继续扫描
5	—	—	停止接收数据包，然后继续扫描

如果正在接收的数据包无法放入 RX 队列，则该数据包应被接收到末尾，但收到的字节不会存储。如果该数据包通常不会从 RX 缓冲区中丢弃，操作应结束。

如果从收到的数据包得到的操作是 3，则在退避流程允许的情况下，应发送 SCAN_REQ 消息。此过程从递减 pParams->backoffCount 开始。如果该变量在递减后变为 0，则应发送 SCAN_REQ 消息。否则，操作应结束。

如果从收到的数据包得到的操作是 2 或 3，则下一步操作可能是继续扫描或结束操作。这通过 pParams->scanConfig.bEndOnRpt 配置；如果为 1，操作应结束，否则应继续扫描。

发送 SCAN_REQ 消息时，无线电 CPU 应构建此数据包。在标头中，PDU 类型位应为 0011b。TxAdd 位应如 pParams->scanConfig.deviceAddrType 中所设（如果 pParams->pDeviceAddress 的最低有效位为 1，则 TxAdd 位取反，请参阅节 25.8.17）。RxAdd 位应与收到的 ADV_IND 或 ADV_SCAN_IND 消息标头中的 TxAdd 字段相同。长度应根据扫描请求数据的大小计算，即应为 pParams->scanReqLen + 12。ChSel 和 RFU 位应为 0。有效载荷应以 6 字节的器件地址开始，该地址从 pParams->pDeviceAddress 中读取，其后是从收到的消息的 AdvA 字段读取的 6 字节对端地址。根据低功耗 Bluetooth 规范，不应再有更多有效载荷，但在使用 CMD_BLE_SCANNER 命令时，可通过将 pParams->scanReqLen 设置为非零值来构建不合规的消息。如果是这样，有效载荷的其余部分应从 pParams->pScanData 缓冲区读取。

SCAN_REQ 消息发送后，无线电 CPU 应配置接收器并查找来自 SCAN_REQ 所发向的广播器的 SCAN_RSP。如果解调器获得同步，一旦接收到标头就应进行检查，如果不是 SCAN_RSP 消息，应立即停止解调器。如果是 SCAN_RSP 消息，则应将其接收到 RX 队列中。根据收到的 SCAN_RSP 消息，bCrcErr 和 bIgnore 的值应如表 25-148 中所示。如果 pParams->scanConfig.bStrictLenFilter 为 1，则只有符合蓝牙低耗能规范的长度字段才被视为有效。对于 SCAN_RSP，这意味着长度字段在 6 到 37 之间。如果 pParams->scanConfig.bStrictLenFilter 为 0，则收到的所有长度字段小于或等于广播器数据包最大长度的数据包均被视为有效。如果长度无效，则应停止接收器。

表 25-148. 扫描器在发送 SCAN_REQ 后对收到的数据包的操作

PDU 类型	CRC 结果	AdvA 与请求中相同	bCrcErr	bIgnore	响应数据包结果
SCAN_RSP	OK	否	0	1	失败
SCAN_RSP	OK	是	0	0	成功
SCAN_RSP	NOK	X	1	0	失败
SCAN_RSP 长度无效	X	X	—	—	失败
其他	X	不适用	—	—	失败
未收到数据包	不适用	不适用	—	—	失败

接收或尝试接收 SCAN_RSP 后，无线电 CPU 应根据节 25.8.15 中所述，基于表 25-148 的*响应数据包结果*列中给出的结果以及退避参数的旧值来更新退避参数。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

如果 pParams->scanConfig.bAutoWlIgnore 为 1，则会启用自动忽略功能。在表 25-154 最右列所述的情况下，无线电 CPU 应自动设置白名单条目的 bWlIgn 位，该条目对应于接收 ADV*_IND 消息的源地址。此操作应在执行操作 2 之后，或在执行操作 3 后收到 SCAN_RSP 且结果为 Success 之后完成。此功能可用于避免报告来自同一器件的多条广播消息，并避免重复扫描同一器件。

25.8.10.2 扫描器在主通道上接收扩展广播数据包

如果在主广播通道 (ADV_EXT_IND) 上运行 CMD_BLE5_SCANNER，则在跟随 AuxPtr 之前，接受主通道上的扩展广播包。

根据 CRC 结果和接收到的消息来设置 bCrcErr 和 bIgnore 位。如果扩展标头标志指示 AdvA 存在，则将根据白名单检查消息中的已接收 AdvA 字段以及已接收标头的 TxAdd 位，如节 25.8.14 中所述。如果 pParams->scanConfig.rpaMode 给出的可解析私有地址 (RPA) 模式为非零，则会进行额外检查以查看对等地址是否为可解析私有地址。如果接收到的 TxAdd 为 1，并且接收到的 AdvA 字段的两个最高有效位为 01b，则地址是 RPA。如果是，则无论过滤器策略如何，都会执行白名单检查。对 AdvA 的完整检查如表 25-144 中所示执行；但是，只有在扩展标头中没有 ADI 字段并且 pParams->extFilterConfig.bApplyDuplicateFiltering 为 1 时，白名单中设置的 bWlIgn 位的规则才会应用。可以按照节 25.8.10.4 中的说明找到 ADI 结果。如果扩展标头标志指示 AdvA 不存在，则始终接受 AdvA 筛选器结果。如果扩展标头标志指示 TargetA 存在（即，发送数据包），则会分别根据 pParams->pDeviceAddress 和 pParams->scanConfig.deviceAddrType 检查接收到的 TargetA 字段和 RxAdd 位。如果 pParams->scanConfig.rpaFilterPolicy 给出的 RPA 滤波器策略为 1，则如果 TargetA 和 RxAdd 指示 RPA，则也接受数据包。表 25-145 中给出了定向消息的完整滤波。非定向消息始终具有 TargetA 匹配。根据接收到的数据包，所采取的操作应如表 25-149 中所示，其中每个操作的定义（包括将在 bCrcErr 和 bIgnore 上使用的值）在表 25-150 中给出。默认情况下，接收到的 ADV_EXT_IND 数据包的数据包长度始终有效，但可以通过覆盖固件定义的参数 maxAdvExtLen 来配置最大长度。此外，还会检查扩展标头长度和标志。如果扩展标头长度太大而无法容纳在数据包中，或者如果扩展标头长度太小而无法容纳所配置的数据包，则长度无效。如果 pParams->scanConfig.bStrictLenFilter 为 1，则考虑所有已定义的字段，而如果 pParams->scanConfig.bStrictLenFilter 为 0，则在查找允许的最小扩展标头长度时，CPE 未自动检查的字段 (SyncInfo 和 TxPower) 将被忽略。

表 25-149. 由扫描器对收到的扩展广播器数据包执行的操作

PDU 类型	CRC 结果	AdvA 滤波器结果	TargetA 匹配	ADI 结果	存在 AuxPtr	操作编号
ADV_EXT_IND	OK	拒绝	X	X	X	1
ADV_EXT_IND	OK	接受	否	X	X	1
ADV_EXT_IND	OK	接受	是	拒绝	X	1
ADV_EXT_IND	OK	接受	是	接受	否	2
ADV_EXT_IND	OK	接受	是	接受	是	6
ADV_EXT_IND	NOK	X	X	X	X	4
长度无效的 ADV_EXT_IND	X	X	X	X	X	5
其他（另请参阅节 25.8.10.1）	X	X	X	X	X	5

表 25-150. 对扫描器接收到的扩展广播器数据包执行的操作说明

操作编号	bCrcErr	bIgnore	说明
1	0	1	继续扫描
2	0	0	以 BLE_DONE_OK 状态继续扫描或结束操作
4	1	0	继续扫描
5	—	—	停止接收数据包，然后继续扫描
6	0	0	跟随 Aux 指针并在新通道上接收，或使用 BLE_DONE_AUX 结束操作

如果所接收的数据包不适合 RX 队列，则应在末尾接收数据包，但不存储接收到的字节。如果通常不会从 RX 缓冲器中丢弃数据包，则操作应结束

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

如果接收到的数据包中的操作为 2，则下一个操作可能是继续扫描或结束操作。这通过 pParams->scanConfig.bEndOnRpt 进行配置；如果为 1，则操作将结束，否则扫描将继续。

如果 pParams->extFilterConfig.bAutoWlIgnore 为 1，则会启用自动忽略功能。如果表 25-144 的最右列中有所规定，并且扩展标头中不存在 ADI 字段，则无线电 CPU 应自动设置白名单条目的 bWlIgn 位，该白名单条目与接收 ADV_EXT_IND 消息的地址相对应。应在执行操作 2 之后或即将执行操作 6 时执行此操作。这可用于避免报告来自同一器件的多条广播消息，并避免重复扫描同一器件。

如果接收到的数据包中的操作为 6，则无线电 CPU 应遵循节 25.8.16 中的过程。这将导致接收器查找节 25.8.10.3 中所述的次级广播包，或者以 BLE_DONE_AUX 状态结束操作。

25.8.10.3 扫描器在次级通道上接收扩展广播包

如果在次级广播通道上运行 CMD_BLE5_SCANNER，或者在 AuxPtr 之后运行，则接受次级通道（AUX_ADV_IND 和 AUX_CHAIN_IND）上的扩展广播包。

根据 CRC 结果和接收到的消息来设置 bCrcErr 和 bIgnore 位。根据 pParams->scanConfig.scanFilterPolicy 提供的扫描滤波器策略，消息中接收的 AdvA 字段以及已接收的标头的 TxAdd 位将根据白名单进行检查，如节 25.8.14 中所述。如果 pParams->scanConfig.rpaMode 给出的可解析私有地址 (RPA) 模式为非零，则会进行额外检查以查看对等地址是否为可解析私有地址。如果接收到的 TxAdd 为 1，并且接收到的 AdvA 字段的两个最高有效位为 01b，则地址是 RPA。如果是，则无论过滤器策略如何，都会执行白名单检查。对 AdvA 的完整检查如表 25-154 中所示执行；但是，只有在扩展标头中没有 ADI 字段并且 pParams->extFilterConfig.bApplyDuplicateFiltering 为 1 时，白名单中设置了 bWlIgn 位的条目才会得到考虑。可以按照节 25.8.10.4 中的说明找到 ADI 结果。如果扩展标头标志指示 TargetA 存在（即，发送数据包），则会分别根据 pParams->pDeviceAddress 和 pParams->scanConfig.deviceAddrType 检查接收到的 TargetA 字段和 RxAdd 位（另请参阅节 25.8.17）。如果 pParams->scanConfig.rpaFilterPolicy 给出的 RPA 滤波器策略为 1，则如果 TargetA 和 RxAdd 指示 RPA，则也接受数据包。表 25-145 中给出了定向消息的完整滤波。非定向消息始终具有 TargetA 匹配。根据接收到的数据包，以及扫描是活动的还是被动的（在 pParams->scanConfig.bActiveScan 中发出信号），所采取的操作应如表 25-151 中所示，其中每个操作的定义（包括将在 bCrcErr 和 bIgnore 上使用的值）在表 25-152 中提供。默认情况下，接收到的 AUX_ADV_IND 或 AUX_CHAIN_IND 数据包的数据包长度始终有效，但可以通过覆盖固件定义的参数 maxAdvExtLen 来配置最大长度。此外，还会检查扩展标头长度和标志。如果扩展标头长度太大而无法容纳在数据包中，或者如果扩展标头长度太小而无法容纳所配置的数据包，则长度无效。如果 pParams->scanConfig.bStrictLenFilter 为 1，则考虑所有已定义的字段，而如果 pParams->scanConfig.bStrictLenFilter 为 0，则在查找允许的最小扩展标头长度时，CPE 未自动检查的字段 (SyncInfo 和 TxPower) 将被忽略。

表 25-151. 扫描器对收到的次级通道数据包执行的操作

PDU 类型	AdvMode	CRC 结果	AdvA 滤波器结果	TargetA 匹配	ADI 结果	存在 AuxPtr	活动扫描	操作编号
AUX_ADV_IND	X	OK	拒绝	X	X	X	X	1
AUX_ADV_IND	X	OK	接受	否	X	X	X	1
AUX_ADV_IND	X	OK	接受	是	拒绝	X	X	1
AUX_ADV_IND	00	OK	接受	是	接受	否	X	2
AUX_ADV_IND	00	OK	接受	是	接受	是	X	6
AUX_ADV_IND	01、11	OK	接受	是	接受	X	X	2
AUX_ADV_IND	10	OK	接受	是	接受	X	否	2
AUX_ADV_IND	10	OK	接受	是	接受	X	是	3
AUX_ADV_IND	X	NOK	X	X	X	X	X	4
长度无效的 AUX_ADV_IND	X	X	X	X	X	X	X	5
其他	X	X	X	X	X	X	X	5

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-152. 对扫描器接收到的次级通道数据包执行的操作的说明

操作编号	bCrcErr	bIgnore	说明
1	0	1	以 BLE_DONE_RXERR 状态结束操作
2	0	0	以 BLE_DONE_OK 状态结束操作
3	0	0	执行退避过程并发送 AUX_SCAN_REQ，然后接收 AUX_SCAN_RSP（如果适用）。然后结束操作
4	1	0	以 BLE_DONE_RXERR 状态结束操作
5	—	—	停止接收数据包，然后继续扫描
6	0	0	跟随 Aux 指针并在新通道上接收，或使用 BLE_DONE_AUX 结束操作

如果所接收的数据包不适合 RX 队列，则应在末尾接收数据包，但不存储接收到的字节。如果通常不会从 RX 缓冲器中丢弃数据包，则操作应结束

如果接收到的数据包的操作为 3，则在退避过程之后允许时，将发送 AUX_SCAN_REQ。此过程从递减 pParams->backoffCount 开始。如果此变量在递减后变为 0，则应发送 AUX_SCAN_REQ。否则，该操作将结束。

当传输 AUX_SCAN_REQ 时，无线电 CPU 应构建此数据包。在标头中，PDU 类型位应为 0011b。TxAdd 位应与 pParams->scanConfig.deviceAddrType 中 的一样（如果 pParams->pDeviceAddress 的最低有效位为 1，则 TxAdd 位被反转，请参阅节 25.8.17）。RxAdd 位应与收到的 AUX_ADV_IND 消息标头的 TxAdd 字段中的相同。长度应为 12。ChSel 和 RFU 位应为 0。有效载荷应以 6 字节器件地址开始，该地址应从 pParams->pDeviceAddress 读取、后跟从接收到的消息的 AdvA 字段读取的 6 字节对等地址。

发送 AUX_SCAN_REQ 消息后，无线电 CPU 应配置接收器并查找 AUX_SCAN_RSP。将使用表 25-151 中的规则接收数据包，但不执行操作 3，而改为执行操作 2，在这种情况下，操作 5 是在 BLE_DONE_NOSYNC 状态下结束。这意味着即使数据包的 AdvA 与第一个 AUX_ADV_IND 数据包不同，也可以接受该数据包。要纠正此问题，可以在系统 CPU 中进行检查，或者可以考虑使用插入码。

在接收或尝试接收 AUX_SCAN_RSP 后，无线电 CPU 应根据表 25-148 的*响应数据包结果*列中提供的结果和回退参数的旧值更新回退参数。

如果接收到的数据包中的操作为 6，则无线电 CPU 应遵循节 25.8.16 中的过程。这将导致接收器查找本部分中所述的次级广播包，或者以 BLE_DONE_AUX 状态结束操作。

25.8.10.4 ADI 滤波

如果接收到的扩展广播器数据包在扩展标头中包含 AdvDataIndex 字段并且 pParams->extFilterConfig.bCheckAdi 为 1，则会根据 pParams->pAdiList 指向的列表条目检查 AdvDataIndex 字段。此列表由表 25-119 中定义的类型的 16 个条目组成。

AdvDataIndex 字段由一个广播集 ID (SID) 及一个广播数据 ID (DID) 字段组成。根据列表进行检查时，将使用索引由接收的 SID 提供的条目。首先检查该条目的模式字段。如果该位为 0，则不会对该 DID 使用滤波，并且接受 ADI。如果模式为 2，则始终拒绝 ADI。如果模式为 1，则根据条目的 advDataId 字段检查接收到的 DID。如果它们相等，则拒绝 ADI，否则接受。

如果接收到的数据包带有 CRC OK（未忽略），且缓冲器未满，并且 pParams->extFilterConfig.bCheckAdi 和 pParams->extFilterConfig.bAutoAdiUpdate 均为 1，则无线电 CPU 应自动更新 ADI 列表。然后，它将接收到的 DID 字段复制到接收到的 SID 提供的条目的 advDataId 字段中。

备注

如果模式为 0，则复制完成，但不会自动修改模式，未来具有相同 DID 的数据包将被忽略，必须由系统 CPU 修改模式。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.8.10.5 扫描器命令结束

用于结束操作的两个触发条件分别由 pParams->endTrigger/pParams->endTime 和 pParams->timeoutTrigger/pParams->timeoutTime 设置。触发具有以下行为：

- 如果在等待广播器数据包上的同步时发生超时触发，则应立即结束操作
- 如果在主通道上运行时另一次发生超时触发，则应继续操作，直到在主通道上恢复扫描，然后结束。
- 如果跟随 Aux 指针，则会忽略任何超时触发（过去或未来）。
- 如果在等待广播器数据包上的同步时发生结束触发，则应立即结束操作。
- 如果结束触发发生在另一个时间，则应继续操作，直至遵循 Aux 指针，否则将在主通道上恢复扫描，然后结束。

如果无线电 CPU 接收到立即命令 CMD_STOP，它应与发生的结束触发器具有相同的含义，但结束后的状态代码应为 CMD_DONE_STOPPED。通常，timeoutTrigger 可以在扫描窗口结束时使用，而 endTrigger 可以在扫描完全结束时使用。

输出结构 pOutput 包含的字段提供有关正在运行的命令的信息。无线电 CPU 不应初始化字段，因此当需要复位计数器时，必须由系统 CPU 执行此操作。这些字段应由无线电 CPU 更新，如下所述。该列表还指示何时应当在系统 CPU 中发出中断。

- 如果发送了 SCAN_REQ 或 AUX_SCAN_REQ 数据包，则 nTxScanReq (CMD_BLE_SCANNER) 或 nTxReq (CMD_BLE5_SCANNER) 会递增，并产生 Tx_Done 中断。
- 如果由于退避过程而未发送 SCAN_REQ 或 AUX_SCAN_REQ，则 nBackedOffScanReq (CMD_BLE_SCANNER) 或 nBackedOffReq (CMD_BLE5_SCANNER) 会递增
- 如果在“CRC OK”的情况下接收 *ADV*_IND 或 AUX_CHAIN_IND 数据包并且清除 bIgnore 标志，则 nRxAdvOk 递增，产生 Rx_Ok 中断，并将 timeStamp 设置为数据包开始的时间戳。
- 如果在 CRC OK 的情况下接收到 *ADV*_IND 或 AUX_CHAIN_IND 数据包并且设置了 bIgnore 标志，则 nRxAdvIgnored 递增并产生 Rx_Ignored 中断。
- 如果接收到 *ADV*_IND 或 AUX_CHAIN_IND 数据包时出现 CRC 错误，则 nRxAdvNok 递增，并产生 Rx_Nok 中断。
- 如果收到 *ADV*_IND 或 AUX_CHAIN_IND 数据表并且不适合 RX 队列，则 nRxAdvBufFull 递增，并产生 Rx_Buf_Full 中断。
- 如果接收到 SCAN_RSP 或 AUX_SCAN_RSP 数据包且 CRC OK 并且清除了 bIgnore 标志，则 nRxScanRspOk (CMD_BLE_SCANNER) 或 nRxRspOk (CMD_BLE5_SCANNER) 会递增，并产生 Rx_Ok 中断。
- 如果在 CRC OK 的情况下接收到 SCAN_RSP 或 AUX_SCAN_RSP 数据包，并且设置了 bIgnore 标志，则 nRxScanRspIgnored (CMD_BLE_SCANNER) 或 nRxRspIgnored (CMD_BLE5_SCANNER) 会递增，并产生 Rx_Ignored 中断。
- 如果接收到 SCAN_RSP 或 AUX_SCAN_RSP 数据包但有 CRC 错误，则 nRxScanRspNok (CMD_BLE_SCANNER) 或 nRxRspNok (CMD_BLE5_SCANNER) 会递增，并产生 Rx_Nok 中断。
- 如果接收到 SCAN_RSP 或 AUX_SCAN_RSP 数据包但不适合 RX 队列，则 nRxScanRspBufFull (CMD_BLE_SCANNER) 或 nRxRspBufFull (CMD_BLE5_SCANNER) 将递增，并产生 Rx_Buf_Full 中断。
- 如果接收到一个数据包，则 lastRssi 设置为该数据包的 RSSI。
- 如果 RX 队列中的第一个 RX 数据条目在接收到数据包后将状态更改为完成，则会产生 Rx_Entry_Done 中断。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

扫描器操作应以表 25-153 中列出的状态之一结束。操作结束后，命令结构的状态字段指示操作结束的原因。在所有情况下，都会产生 Command_Done 中断。在每种情况下，都会指示 TRUE、FALSE 或 ABORT 的结果，这些结果将决定下一个操作。

表 25-153. 扫描器操作结束

条件	状态代码	结果
使用 pParams->scanConfig.bEndOnRpt = 1 或在次级通道上接收时执行操作 2	BLE_DONE_OK	TRUE
在 pParams->scanConfig.bEndOnRpt = 1 时或在次级通道上执行操作 3，并且由于退避而未发送 SCAN_REQ 或 AUX_SCAN_REQ	BLE_DONE_OK	TRUE
在 pParams->scanConfig.bEndOnRpt = 1 时或在次级通道上执行操作 3，发送了 SCAN_REQ 或 AUX_SCAN_REQ，并接收到 SCAN_RSP 或 AUX_SCAN_RSP，其中 bCrcErr = 0 且 bIgnore = 0	BLE_DONE_OK	TRUE
在次级通道上执行操作 1 或操作 4	BLE_DONE_RXERR	TRUE
在 pParams->scanConfig.bEndOnRpt = 1 时或在次级通道上执行操作 3，发送了 SCAN_REQ 或 AUX_SCAN_REQ 并接收到 SCAN_RSP 或 AUX_SCAN_RSP，其中 bCrcErr = 1 或 bIgnore = 1	BLE_DONE_RXERR	TRUE
在 pParams->scanConfig.bEndOnRpt = 1 时或在次级通道上执行操作 3，发送了 SCAN_REQ 或 AUX_SCAN_REQ，但未实现同步或发现错误的数据包类型或无效的长度	BLE_DONE_NOSYNC	TRUE
等待广播器数据包同步时，观察到由 pParams->timeoutTrigger 指示的触发	BLE_DONE_RXTIMEOUT	TRUE
在主通道上观察到的触发由 pParams->timeoutTrigger 指示，然后执行操作 1、2、3、4 或 5。	BLE_DONE_RXTIMEOUT	TRUE
等待广播器数据包同步时，观察到了由 pParams->endTrigger 指示的触发	BLE_DONE_ENDED	FALSE
观察到由 pParams->endTrigger 指示的触发，然后执行了操作 1、2、3、4、5 或 6。	BLE_DONE_ENDED	FALSE
在广播器数据包等待同步时观察到 CMD_STOP	BLE_DONE_STOPPED	FALSE
观察到 CMD_STOP，然后执行了操作 1、2、3、4、5 或 6。	BLE_DONE_STOPPED	FALSE
执行操作 6，等待 AUX 数据包的时间大于 pParams->maxWaitTimeForAuxCh。	BLE_DONE_AUX	FALSE
收到了 CMD_ABORT	BLE_DONE_ABORT	ABORT
RX 缓冲器中没有空间来存储接收到的数据包	BLE_ERROR_RXBUF	FALSE
通道值非法	BLE_ERROR_PAR	ABORT
扫描请求数据长度字段具有非法值	BLE_ERROR_PAR	ABORT

25.8.11 启动器命令

CMD_BLE_INITIATOR 或 CMD_BLE5_INITIATOR 命令会将启动器操作启动。在命令结构中，它应具有一个在表 25-100 或表 25-108 中定义的类型的 pParams 参数和一个在表 25-100 或表 25-113 中定义的类型的 pOutput 参数。在启动器操作开始时，无线电 CPU 应等待启动触发。然后，它应根据命令结构的通道参数对频率进行编程。对于 CMD_BLE_INITIATOR，不允许通道参数处于 0 – 36 范围内，因为它们不是主广播通道。对于 CMD_BLE5_INITIATOR，它还应设置 phyMode.mainMode 中给出的 PHY 模式。无线电 CPU 应设置广播通道访问地址并使用 CRC 初始化值 0x55 5555。白化器须按照白化参数的定义进行配置。然后，无线电 CPU 应配置接收器。

调谐到正确的通道后，无线电 CPU 应开始侦听广播通道数据包。如果在解调器上获得同步，则会将消息接收到 RX 队列中。检查报头，如果不是可连接的广播包，则应停止接收并重新启动同步搜索。接受的数据包和进一步的运行取决于命令运行、通道类型及 PHY 模式。

25.8.11.1 在主通道上接收传统广播数据包的启动器

如果在具有 1Mbps PHY 的主要广播通道上运行 CMD_BLE_INITIATOR 或 CMD_BLE5_INITIATOR，则接受传统广播数据包 (ADV_IND and ADV_DIRECT_IND)。操作如下表所示：

- 根据 CRC 结果和接收到的消息来设置 bCrcErr 和 bIgnore 位。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

- 参数 pParams->initConfig.bUseWhiteList 确定启动器是应该尝试连接到特定器件，还是依据白名单来连接。如果此参数为 0，则不使用白名单，并且 pParams->pWhiteList 应指向仅包含要连接的器件的器件地址的缓冲器。
- 对等方的地址类型应在 pParams->advConfig.peerAddrType 中给出（另请参阅节 25.8.17）。否则，pParams->pWhiteList 应指向白名单，并且将根据节 25.8.14 中所述白名单检查消息中的已接收 AdvA 字段以及已接收标头的 TxAdd 位。
- 表 25-154 中汇总了筛选。
- 对于 ADV_DIRECT_IND 消息，将分别根据 pParams->pDeviceAddress 和 pParams->initConfig.deviceAddrType 检查收到的 TargetA 字段和 RxAdd 位（另请参阅节 25.8.17）。根据这点，所采取的操作应如表 25-155 中所示，其中每个操作的定义（包括将在 bCrcErr 和 bIgnore 上使用的值）在表 25-156 中给出。
- 如果 pParams->initConfig.bStrictLenFilter 为 1，则只有符合低功耗蓝牙规范的长度字段才应视为有效。
- 对于 ADV_DIRECT_IND，这意味着长度字段为 12，而对于 ADV_IND 消息，这意味着长度字段的范围为 6 – 37。
- 如果 pParams->initConfig.bStrictLenFilter 为 0，则所有收到的长度字段小于或等于广播器数据包最大长度的数据包均应视为有效。如果长度无效，应停止接收器。

表 25-154. 筛选接收到的广播器地址

bUseWhiteList	AdvA 与特定的对等地址匹配	AdvA 将 bEnable = 1 与白名单条目匹配	AdvA 将 bIrkValid = 1 与白名单条目匹配	AdvA 滤波器结果
0	否	不适用	不适用	拒绝
0	是	不适用	不适用	接受
1	不适用	否	X	拒绝
1	不适用	是	否	接受
1	不适用	是	是	拒绝

表 25-155. 由启动器对收到的数据包执行的操作

PDU 类型	CRC 结果	AdvA 滤波器结果	TargetA 匹配	操作编号
ADV_IND	OK	拒绝	不适用	1
ADV_IND	OK	接受	不适用	3
ADV_IND	NOK	X	不适用	4
ADV_DIRECT_IND	OK	拒绝	X	1
ADV_DIRECT_IND	OK	接受	否	1
ADV_DIRECT_IND	OK	接受	是	3
ADV_DIRECT_IND	NOK	X	X	4
长度无效的 ADV*_IND	X	X	X	5
其他（另请参阅节 25.8.11.2）	X	不适用	不适用	5

表 25-156. 对启动器接收到的数据包执行的操作说明

操作编号	bCrcErr	bIgnore	说明
1	0	1	继续扫描
3	0	0	发送 CONNECT_IND 和结束操作
4	1	0	继续扫描
5	—	—	停止接收数据包，然后继续扫描

如果所接收的数据包不适合 RX 队列，则应在末尾接收数据包，但不存储接收到的字节。如果通常不会从 RX 缓冲器中丢弃数据包，则操作应结束

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

如果接收到的数据包的操作为 2，则应传输 CONNECT_IND（在相关文档中列出的蓝牙 4.0、4.1 和 4.2 规范中称为 CONNECT_REQ）数据包。当传输 CONNECT_IND 时，无线电 CPU 应构建此数据包。在标头中，PDU 类型位应为 0101b。TxAdd 位应与 pParams->initConfig.deviceAddrType 中 的一样（如果 pParams->pDeviceAddress 的最低有效位为 1，则 TxAdd 位被反转，请参阅节 25.8.17）。RxAdd 位应与收到的 ADV_IND 或 ADV_DIRECT_IND 消息标头的 TxAdd 字段相同。如果 pParams->initConfig.chSel 为 1，则 ChSel 位应为 1；否则，ChSel 应为 0。长度应根据 LLData 的长度计算得出，这意味着它应为 pParams->connectReqLen + 12。RFU 位应为 0。有效载荷应以 6 字节器件地址开始，该地址应从 pParams->pDeviceAddress 读取、后跟从接收到的消息的 AdvA 字段读取的 6 字节对等地址。有效载荷的其余部分应从 pParams->pConnectData 缓冲器读取。如果 pParams->initConfig.bDynamicWinOffset 为 1，则无线电 CPU 应将 WinSize 和 WinOffset 位置中的字节（请参阅相关文档中列出的蓝牙系统规范 5.0 版）替换节 25.8.11.4 为中所述的计算值。发送 CONNECT_IND 消息后，操作应结束。如果 pParams->initConfig.chSel 为 1，则应检查收到的 ADV_IND 或 ADV_DIRECT_IND 数据包的 ChSel 位以确定结束状态；请参阅节 25.8.11.5。

25.8.11.2 启动器在主通道上接收扩展广播数据包

如果在主广播通道 (ADV_EXT_IND) 上运行 CMD_BLE5_SCANNER，则在跟随 AuxPtr 之前，接受主通道上的扩展广播包。

根据 CRC 结果和接收到的消息来设置 bCrcErr 和 bIgnore 位。启动器仅接受可连接的数据包。参数 pParams->initConfig.bUseWhiteList 确定启动器是应该尝试连接到特定器件，还是依据白名单来连接。如果此参数为 0，则不使用白名单，并且 pParams->pWhiteList 应指向仅包含要连接的器件的器件地址的缓冲器。对等方的地址类型应在 pParams->advConfig.peerAddrType 中给出（另请参阅节 25.8.17）。否则，pParams->pWhiteList 应指向白名单，并且将根据节 25.8.14 中所述白名单检查消息中的已接收 AdvA 字段以及已接收标头的 TxAdd 位。对 AdvA 执行的完整检查如表 25-144 中所示。如果扩展标头标志指示 AdvA 不存在，则始终接受 AdvA 筛选器结果。如果扩展标头标志指示 TargetA 存在（即数据包已定向），则会分别根据 pParams->pDeviceAddress 和 pParams->initConfig.deviceAddrType 检查接收到的 TargetA 字段和 RxAdd 位（另请参阅节 25.8.17），并且如果它们相等，则进行匹配。非定向消息始终具有 TargetA 匹配。根据接收到的数据包，所采取的操作应如表 25-157 中所示，其中每个操作的定义（包括将在 bCrcErr 和 bIgnore 上使用的值）在表 25-158 中给出。默认情况下，接收到的 ADV_EXT_IND 数据包的数据包长度始终有效，但可以通过覆盖固件定义的参数 maxAdvExtLen 来配置最大长度。此外，还会检查扩展标头长度和标志。如果扩展标头长度太大而无法容纳在数据包中，或者如果扩展标头长度太小而无法容纳所配置的数据包，则长度无效。如果 pParams->initConfig.bStrictLenFilter 为 1，则考虑所有已定义的字段，而如果 pParams->initConfig.bStrictLenFilter 为 0，则在查找允许的最小扩展标头长度时，CPE 未自动检查的字段 (SyncInfo 和 TxPower) 将被忽略。

表 25-157. 由启动器对收到的扩展广播器数据包执行的操作

PDU 类型	CRC 结果	AdvMode	AdvA 滤波器结果	TargetA 匹配	存在 AuxPtr	操作编号
ADV_EXT_IND	OK	01	拒绝	X	X	1
ADV_EXT_IND	OK	01	接受	否	X	1
ADV_EXT_IND	OK	01	接受	是	否	2
ADV_EXT_IND	OK	01	接受	是	是	6
ADV_EXT_IND	NOK	01	X	X	X	4
ADV_EXT_IND	X	00、10 或 11	X	X	X	5
ADV_EXT_IND，长度无效	X	X	X	X	X	5
其他（另请参阅节 25.8.10.1）	X	X	X	X	X	5

表 25-158. 对启动器接收到的扩展广播器数据包执行的操作说明

操作编号	bCrcErr	bIgnore	说明
1	0	1	继续扫描
2	0	0	继续扫描
4	1	0	继续扫描
5	—	—	停止接收数据包，然后继续扫描

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-158. 对启动器接收到的扩展广播器数据包执行的操作说明（续）

操作编号	bCrcErr	bIgnore	说明
6	0	0	跟随 Aux 指针并在新通道上接收，或使用 BLE_DONE_AUX 结束操作

如果所接收的数据包不适合 RX 队列，则应在末尾接收数据包，但不存储接收到的字节。如果通常不会从 RX 缓冲器中丢弃数据包，则操作应结束。

如果接收到的数据包的操作为 6，则无线电 CPU 将查找次要广播包或以 BLE_DONE_AUX 状态结束操作。

25.8.11.3 启动器在次级通道上接收扩展广播数据包

如果在次级广播通道上运行 CMD_BLE5_INITIATOR 或者在 AuxPtr 之后运行，则可接受次级通道 (AUX_ADV_IND) 上的扩展广播包。

根据 CRC 结果和接收到的消息来设置 bCrcErr 和 bIgnore 位。启动器仅接受可连接的数据包。参数 pParams->initConfig.bUseWhiteList 确定启动器是应该尝试连接到特定器件，还是依据白名单来连接。如果此参数为 0，则不使用白名单，并且 pParams->pWhiteList 应指向仅包含要连接的器件的器件地址的缓冲器。对等方的地址类型应在 pParams->advConfig.peerAddrType 中给出（另请参阅节 25.8.17）。否则，pParams->pWhiteList 应指向白名单，并且将根据节 25.8.14 中所述白名单检查消息中的已接收 AdvA 字段以及已接收标头的 TxAdd 位。对 AdvA 执行的完整检查如表 25-144 中所示。如果扩展标头标志指示 AdvA 不存在，则始终接受 AdvA 筛选器结果。如果扩展标头标志指示 TargetA 存在（即数据包已定向），则会分别根据 pParams->pDeviceAddress 和 pParams->initConfig.deviceAddrType 检查接收到的 TargetA 字段和 RxAdd 位（另请参阅节 25.8.17），并且如果它们相等，则进行匹配。非定向消息始终具有 TargetA 匹配。根据接收到的数据包，所采取的操作应如表 25-159 中所示，其中每个操作的定义（包括将在 bCrcErr 和 bIgnore 上使用的值）在表 25-160 中给出。默认情况下，接收到的 AUX_ADV_IND 数据包的数据包长度始终有效，但可以通过覆盖固件定义的参数 maxAdvExtLen 来配置最大长度。此外，还会检查扩展标头长度和标志。如果扩展标头长度太大而无法容纳在数据包中，或者如果扩展标头长度太小而无法容纳所配置的数据包，则长度无效。如果 pParams->initConfig.bStrictLenFilter 为 1，则考虑所有已定义的字段，而如果 pParams->initConfig.bStrictLenFilter 为 0，则在查找允许的最小扩展标头长度时，CPE 未自动检查的字段 (SyncInfo 和 TxPower) 将被忽略。

表 25-159. 启动器对接收到的次级通道数据包执行的操作

PDU 类型	CRC 结果	AdvMode	AdvA 滤波器结果	TargetA 匹配	操作编号
AUX_ADV_IND	OK	01	拒绝	X	1
AUX_ADV_IND	OK	01	接受	否	1
AUX_ADV_IND	OK	01	接受	是	3
AUX_ADV_IND	NOK	01	X	X	4
AUX_ADV_IND	X	00、10 或 11	X	X	5
长度无效的 AUX_ADV_IND	X	X	X	X	5
其他	X	X	X	X	5

表 25-160. 对启动器接收到的次级通道数据包执行的操作说明

操作编号	bCrcErr	bIgnore	说明
1	0	1	以 BLE_DONE_RXERR 状态结束操作
3	0	0	执行退避程序并发送 AUX_CONNECT_REQ 和接收 AUX_CONNECT_RSP（如果适用）；然后结束操作
4	1	0	以 BLE_DONE_RXERR 状态结束操作
5	—	—	停止接收数据包，然后继续扫描

如果所接收的数据包不适合 RX 队列，则应在末尾接收数据包，但不存储接收到的字节。如果通常不会从 RX 缓冲器中丢弃数据包，则操作应结束

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

如果接收到的数据包的操作为 3，则在退避过程之后允许时，将发送 AUX_CONNEXT_REQ。此过程从递减 pParams->backoffCount 开始。如果此变量在递减后变为 0，则应发送 AUX_CONNECT_REQ。否则，该操作将结束。

当传输 AUX_CONNECT_REQ 时，无线电 CPU 应构建此数据包。在标头中，PDU 类型位应为 0101b。TxAdd 位应与 pParams->initConfig.deviceAddrType 中 的一样（如果 pParams->pDeviceAddress 的最低有效位为 1，则 TxAdd 位被反转，请参阅节 25.8.17）。RxAdd 位应与收到的 AUX_ADV_IND 消息标头的 TxAdd 字段中的相同。长度应根据 LLData 的长度计算得出，这意味着它应为 pParams->connectReqLen + 12。ChSel 和 RFU 位应为 0。有效载荷应以 6 字节器件地址开始，该地址应从 pParams->pDeviceAddress 读取、后跟从接收到的消息的 AdvA 字段读取的 6 字节对等地址。有效载荷的其余部分应从 pParams->pConnectData 缓冲器读取。如果 pParams->initConfig.bDynamicWinOffset 为 1，则无线电 CPU 将用计算值替换 WinSize 和 WifOffset 位置中的字节（请参阅相关文档中列出的蓝牙系统规范 5.0 版）。

发送 AUX_CONNECT_REQ 消息后，无线电 CPU 应配置接收器并查找 AUX_CONNECT_RSP。将使用表 25-161 和表 25-162 中的规则接收数据包。表 25-161 中的规则意味着即使数据包的 AdvA 与第一个 AUX_ADV_IND 数据包不同，也可以接受该数据包。要纠正此问题，可以在系统 CPU 中进行检查，或者可以考虑使用插入码。

表 25-161. 启动器对接收到的 AUX_CONNECT_RSP 数据包执行的操作

PDU 类型	CRC 结果	AdvA 滤波器结果	TargetA 匹配	操作编号
AUX_CONNECT_RSP	OK	拒绝	X	1
AUX_CONNECT_RSP	OK	接受	否	1
AUX_CONNECT_RSP	OK	接受	是	3
AUX_CONNECT_RSP	NOK	X	X	4
长度无效的 AUX_CONNECT_RSP	X	X	X	5
其他	X	X	X	5

表 25-162. 对启动器接收到的次级通道数据包执行的操作说明

操作编号	bCrcErr	bIgnore	说明
1	0	1	以 BLE_DONE_RXERR 状态结束操作
3	0	0	以 BLE_DONE_CONNECT 状态结束操作
4	1	0	以 BLE_DONE_RXERR 状态结束操作
5	—	—	停止接收数据包，然后以 BLE_DONE_NOSYNC 状态结束运行

在接收或尝试接收 AUX_CONNECT_RSP 后，无线电 CPU 应根据表 25-148 的*响应数据包结果*列中提供的结果和回退参数的旧值更新回退参数，如节 25.8.15 中所述。

25.8.11.4 自动窗口偏移插入

如果 pParams->initConfig.bDynamicWinOffset 为 1，则无线电 CPU 应在发送的 CONNECT_IND 或 AUX_CONNECT_REQ 消息中自动计算 WinSize 和 WinOffset 参数。WinSize 是有效载荷的字节 7，WinOffset 是字节 8 和 9（请参阅相关文档中列出的蓝牙规范文档）。无线电 CPU 应从 pParams->connectTime 参数和连接间隔中查找第一个连接事件的可能开始时间，该间隔应以待传输有效载荷的间隔字段（字节 10 和 11）的 1.25ms 为单位（请参阅相关文档中列出的蓝牙规范文档）。第一个连接事件的可能时间是 pParams->connectTime 的连接间隔的任意整数倍，从启动器命令开始，可能是过去或未来的时间。

无线电 CPU 应在传输的 CONNECT_IND 或 AUX_CONNECT_RSP 中插入 WinOffset 参数，以便在第一个连接事件发生时发出第一个连接事件信号，该连接事件在 CONNECT_IND 或 AUX_CONNEXT_REQ 数据包结束后足够长的时间内发生。这意味着从 CONNECT_IND 或 AUX_CONNECT_REQ 数据包结束到连接中第一个数据包开始的时间是介于 transmitWindowDelay + WinOffset 和 transmitWindowDelay + WinOffset + WinSize 之间，其中，当使用了 CONNECT_IND PDU 时，transmitWindowDelay 为 1.25ms，当在 LE 未编码 PHY 上使用了 AUX_CONNECT_REQ PDU 时为 2.5 ms，而当在 LE 编码 PHY 上使用了 AUX_CONNECT_REQ PDU 时为

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

3.75 ms（有关详细信息，请参阅相关文档中列出的蓝牙系统规范 5.0 版）。无线电 CPU 应设置发射窗口（WinOffset 和 WinSize），以便在发射窗口的开始和第一个主数据包的开始之间以及第一个主数据包的开始和发射窗口的结束之间存在余量（请参阅相关文档中列出的蓝牙系统规范 5.0 版）。插入的 WinSize 应为 1 或 2；确保有这样的裕度。通过 BLE 栈中的覆盖将裕度设置为正确的值（请参阅节 25.8.4）。

无线电 CPU 应将 WinSize 和 WinOffset 的计算值写入 pParams->pConnectData 缓冲器中的相应位置。无线电 CPU 应在 pParams->connectTime 中写回用于在信号传输窗口内传输第一个数据包的第一个连接事件的开始时间。如果未建立连接，无线电 CPU 应将多个连接间隔添加到 pParams->connectTime，因而这是操作结束后首次可能发生连接事件的时间。

25.8.11.5 启动器命令结束

用于结束操作的两个触发条件分别由 pParams->endTrigger/pParams->endTime 和 pParams->timeoutTrigger/pParams->timeoutTime 设置。如果发生这些触发中的任何一个，无线电操作应尽快结束。如果在等待 ADV*_IND 数据包的同步时发生该情况，则操作应立即结束。如果在另一个时间发生，则应继续操作，直至扫描恢复，然后结束。如果无线电 CPU 接收到立即命令 CMD_STOP（请参阅节 25.3.3.2.2），它应与发生的结束触发器具有相同的含义，但结束后的状态代码应为 CMD_DONE_STOPPED。两个触发器之间的差异在于操作结束时的状态和结果。通常，timeoutTrigger 可以在扫描窗口结束时使用，而 endTrigger 可以在扫描完全结束时使用。

输出结构 pOutput 包含的字段提供有关正在运行的命令的信息。无线电 CPU 不应初始化字段，因此当需要复位计数器时，必须由系统 CPU 执行此操作。这些字段应由无线电 CPU 按后续列表中所述进行更新。该列表还指示何时应当在系统 CPU 中发出中断。

- 如果发送了 CONNECT_IND 或 AUX_CONNECT_REQ 数据包，nTxConnectReq (CMD_BLE_INITIATOR) 或 nTxReq (CMD_BLE5_INITIATOR) 会递增，并引发 Tx_Done 中断。
- 如果由于退避过程而未传输 AUX_CONNECT_REQ，则 nBackedOffReq 会递增
- 如果在“CRC OK”的情况下接收 *ADV*_IND 数据包并且清除 bIgnore 标志，则 nRxAdvOk 递增，产生 Rx_Ok 中断，并将 timeStamp 设置为数据包开始的时间戳。
- 如果在 CRC OK 的情况下接收到 *ADV*_IND 数据包并且设置了 bIgnore 标志，则 nRxAdvIgnored 递增并产生 Rx_Ignored 中断。
- 如果接收到 *ADV*_IND 数据包时出现 CRC 错误，则 nRxAdvNok 递增，并产生 Rx_Nok 中断。
- 如果收到 *ADV*_IND 并且不适合 RX 队列，则 nRxAdvBufFull 递增，并产生 Rx_Buf_Full 中断。
- 如果在“CRC OK”的情况下接收到 AUX_CONNECT_RSP 数据包并且清除了 bIgnore 标志，则 nRxRspOk 会递增，并产生 Rx_Ok 中断。
- 如果在“CRC OK”的情况下接收到 AUX_CONNECT_RSP 数据包并且设置了 bIgnore 标志，则 nRxRspIgnored 递增并产生 Rx_Ignored 中断。
- 如果接收到 AUX_CONNECT_RSP 数据包时出现 CRC 错误，则 nRxRspNok 递增，并产生 Rx_Nok 中断。
- 如果接收到 AUX_SCAN_RSP 数据包并且不适合 RX 队列，则 nRxRspBufFull 会递增，并产生 Rx_Buf_Full 中断。
- 如果接收到一个数据包，则 lastRssi 设置为该数据包的 RSSI。
- 如果 RX 队列中的第一个 RX 数据条目在接收到数据包后将状态更改为完成，则会产生 Rx_Entry_Done 中断。

启动器操作应以表 25-163 中列出的状态之一结束。操作结束后，命令结构的状态字段指示操作结束的原因。在所有情况下，都会产生 Command_Done 中断。在每种情况下，都会指示 TRUE、FALSE 或 ABORT 的结果，这些结果将决定下一个操作，如节 25.3.2.5.2 中所定义。

表 25-163. 启动器操作结束

条件	状态代码	结果
如果 pParams->advConfig.chSel = 0 或者收到的 ADV_IND 或 ADV_DIRECT_IND 的 ChSel 位为 1，则在收到旧数据包（已传输 CONNECT_IND）后执行了操作 3	BLE_DONE_CONNECT	FALSE

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-163. 启动器操作结束（续）

条件	状态代码	结果
如果 pParams->advConfig.chSel = 1 并且收到的 ADV_IND 或 ADV_DIRECT_IND 的 ChSel 位为 0，则在收到旧数据包（已传输 CONNECT_IND）后执行了操作 3	BLE_DONE_CONNECT_CHSEL0	FALSE
发送 AUX_CONNECT_REQ 后正确接收到 AUX_CONNECT_RSP	BLE_DONE_CONNECT	FALSE
收到了可连接 ADV_EXT_IND 数据包，该数据表在没有 AUX 指针的情况下不被忽略	BLE_DONE_OK	TRUE
在辅助通道上执行了操作 3，并且由于退避，未发送 AUX_CONNECT_REQ	BLE_DONE_OK	TRUE
在辅助通道上执行了操作 1 或者操作 4	BLE_DONE_RXERR	TRUE
在辅助通道上执行了操作 3，发送 AUX_CONNECT_REQ 并接收到 AUX_CONNECT_RSP，其中 bCrcErr = 1 或 bIgnore = 1	BLE_DONE_RXERR	TRUE
在辅助通道上执行了操作 3，发送了 AUX_CONNECT_REQ，但未实现同步或找到错误数据包类型或无效长度	BLE_DONE_NOSYNC	TRUE
等待广播器数据包同步时，观察到由 pParams->timeoutTrigger 指示的触发	BLE_DONE_RXTIMEOUT	TRUE
观察到由 pParams->timeoutTrigger 指示的触发，然后执行了操作 1、2、3、4 或 5。	BLE_DONE_RXTIMEOUT	TRUE
等待广播器数据包同步时，观察到了由 pParams->endTrigger 指示的触发	BLE_DONE_ENDED	FALSE
观察到由 pParams->endTrigger 指示的触发，然后执行了操作 1、2、3、4、5 或 6。	BLE_DONE_ENDED	FALSE
在广播器数据包等待同步时观察到 CMD_STOP	BLE_DONE_STOPPED	FALSE
观察到 CMD_STOP，然后执行了操作 1、2、3、4、5 或 6。	BLE_DONE_STOPPED	FALSE
执行操作 6，等待 AUX 数据包的时间大于 pParams->maxWaitTimeForAuxCh。	BLE_DONE_AUX	FALSE
收到了 CMD_ABORT	BLE_DONE_ABORT	ABORT
RX 缓冲器中没有空间来存储接收到的数据包	BLE_ERROR_RXBUF	FALSE
通道值非法	BLE_ERROR_PAR	ABORT
LLData 长度字段具有非法值	BLE_ERROR_PAR	ABORT

25.8.12 通用接收器命令

通用接收器命令可用于接收物理层测试数据包，也可用于接收任何数据包，例如在数据包嗅探器应用中。

通用接收器操作由 CMD_BLE_GENERIC_RX 或 CMD_BLE5_GENERIC_RX 命令启动。在命令结构中，它应包含具有表 25-101 中定义类型的 pParams 参数和具有表 25-114 中定义类型的 pOutput 参数。在通用接收器操作开始时，无线电 CPU 应等待启动触发条件。然后，它应根据命令结构的通道参数来编程频率。对于 CMD_BLE5_GENERIC_RX，还应设置 phyMode.mainMode 中给出的 PHY 模式。无线电 CPU 应设置 pParams->accessAddress 中定义的访问地址，并使用 pParams->crcInit 中定义的 CRC 初始化值。应按照白化参数中的定义设置白化器。然后，无线电 CPU 应配置接收器。

在通用接收器操作中，对数据包格式的唯一假设是，第二个收到的字节是一个长度字段，指示该字节之后的载荷长度，并且数据包末尾附加了一个标准的 BLE 类型 CRC。假设长度字段中的位数与广播数据包相同，即如果使用 CMD_BLE5_RADIO_SETUP 配置，则默认为 8 位，否则为 6 位。

调谐到正确的通道后，无线电 CPU 应开始侦听数据包。如果在解调器上获得同步，则将消息接收到 RX 队列中（如果有）。如果长度大于 BLE 传统广播数据包允许的最大长度（37，但可覆盖），则应停止接收并重新开始。

如果 pParams->pRxQ 为 NULL，则不应存储收到的数据包。计数器仍应更新，并产生中断。

如果收到的数据包存在 CRC 错误，应置位 bCrcErr 位。对于通用 RX 命令，bIgnored 标志永远不会置位。

如果正在接收的数据包无法放入 RX 队列，则该数据包应被接收到末尾，但收到的字节不会存储。如果该数据包通常不会从 RX 缓冲区中丢弃，操作应结束。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

结束操作的触发条件由 pParams->endTrigger 和 pParams->endTime 设置。如果该参数定义的触发条件发生，无线电操作应尽快结束。如果触发条件在等待同步时发生，操作应立即结束。如果在其他时间发生，操作应继续执行到当前数据包完整接收，然后结束。如果无线电 CPU 收到即时命令 CMD_STOP（请参阅节 25.3.3.2.2），其含义与发生结束触发条件时相同，但结束后的状态代码应为 CMD_DONE_STOPPED。

输出结构 pOutput 包含的字段提供有关正在运行的命令的信息。无线电 CPU 不会初始化这些字段，因此当需要复位计数器时，必须由系统 CPU 完成此操作。无线电 CPU 将按照下方列表所述更新这些字段。该列表还指明了何时应在系统 CPU 中引发中断。

- 如果收到数据包且 CRC 正确，则 nRxOk 递增并引发 Rx_Ok 中断。
- 如果收到数据包但 CRC 错误，则 nRxNok 递增并引发 Rx_Nok 中断。
- 如果收到数据包但无法放入 RX 队列，则 nRxBufFull 递增并引发 Rx_Buf_Full 中断。
- 如果收到数据包，则 lastRssi 设置为该数据包的 RSSI。
- 如果收到数据包，则 timeStamp 设置为该数据包开始的时间戳。
- 如果在收到数据包后 RX 队列中的第一个 RX 数据条目将状态更改为“已完成”，则引发 Rx_Entry_Done 中断。

收到数据包后，如果 pParams->bRepeat = 1、未观察到结束事件且数据包可放入接收队列，则应在同一通道上重新开始接收。如果 pParams->bRepeat = 0，则操作始终应在收到数据包后结束。

通用 RX 操作应以表 25-164 中列出的状态之一结束。操作结束后，命令结构中的状态字段指示操作结束的原因。在所有情况下，都会引发 Command_Done 中断。每种情况都会指示 TRUE、FALSE 或 ABORT 结果，这将决定下一步的操作，如节 25.3.2.5.2 中所定义。通用 RX 命令结构的 pNextOp 字段可指向同一命令结构。这样，可以持续执行 RX，直到结束触发条件发生或 RX 缓冲区填满。

表 25-164. 通用 RX 操作结束

条件	状态代码	结果
收到数据包，CRC 正确且 pParams->bRepeat = 0	BLE_DONE_OK	真
收到数据包，CRC 错误且 pParams->bRepeat = 0	BLE_DONE_RXERR	真
等待同步时，观察到 pParams->endTrigger 指示的触发条件	BLE_DONE_ENDED	否
观察到 pParams->endTrigger 指示的触发条件，随后完成数据包接收	BLE_DONE_ENDED	否
等待同步时，观察到 CMD_STOP	BLE_DONE_STOPPED	否
观察到 CMD_STOP，随后完成数据包接收	BLE_DONE_STOPPED	否
收到了 CMD_ABORT	BLE_DONE_ABORT	ABORT
RX 缓冲区中没有空间来存储收到的数据包	BLE_ERROR_RXBUF	否
通道值非法	BLE_ERROR_PAR	ABORT

25.8.13 PHY 测试发送命令

测试数据包发送器命令可用于发送物理层测试数据包。

CMD_BLE_TX_TEST 或者 CMD_BLE5_TX_TEST 命令会启动测试数据包发送器操作。在命令结构中，它应具有一个在表 25-102 中定义的类型的 pParams 参数和一个在表 25-115 中定义的类型的 pOutput 参数。在测试 TX 操作开始时，无线电 CPU 应等待启动触发。然后，它应根据命令结构的通道参数对频率进行编程。对于 CMD_BLE5_TX_TEST，它还应设置 phyMode.mainMode 中给出的 PHY 模式。无线电 CPU 应设置测试数据包同步字（请参阅相关文档中列出的蓝牙规范文档）并使用 CRC 初始化值 0x55 5555。白化器须按照白化参数的定义进行配置。要生成符合蓝牙系统规范（相关文档中列出的 5.0 版文档）的 PHY 测试数据包，应禁用白化器。

无线电 CPU 应发送 pParams->numPackets 数据包，然后结束操作。如果 pParams->numPackets 为 0，传输应继续，直到操作因另一个原因（超时、停止或中止命令）结束。每个数据包开始之间的时间（无线电计时器节拍数）应由 pParams->period 提供。如果该时间小于数据包的持续时间，则应尽快发送每个数据包。每个数据包应按如下方式由无线电 CPU 汇编。第一个字节为标头字节。它应包含 pParams->packetType 的值，前提是这是表 25-165 中列出的值之一。下一个字节是长度字节，这应为 pParams->payloadLength 的值。此字段后面应有数个

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

有效载荷字节，这些字节应如表 25-165 中所列。有效载荷字节数应等于 pParams->payloadLength。如果 pParams->packetType 为 0，则字节应来自相关文档中列出的蓝牙系统文档的规范中定义的 PRBS9 序列。否则，所有字节都应相同，如表 25-165 中所列。根据相关文档中列出的低功耗蓝牙规范，应附加一个 3 字节 CRC。

表 25-165. 支持的 PHY 测试数据包类型

packetType 的值	发送的字节
0	PRBS9 序列
1	重复 0x0F
2	重复 0x55
3	PRBS15 序列
4	重复 0xFF
5	重复 0x00
6	重复 0xF0
7	重复 0xAA

在相关文档中列出的蓝牙规范文档中定义的 PRBS15 有效载荷类型，对应于有效载荷类型 3，应使用多项式 x15 + x14 + 1 来实现。对于发送的第一个数据包，应从无线电计时器进行初始化，而不是对于后续数据包重新初始化。

如果 pParams->config.overrideDefault 为 1，则数据包应为非标准数据包。标头应包含 pParams->packetType 中给出的值，传输的每个字节应在 pParams->byteVal 中提供。如果 pParams->config.bUsePrbs9 为 1，则应通过 XOR 运算生成序列，其中用于数据包类型 0 的 PRBS9 序列的每个字节与 pParams->byteVal 进行该运算。如果 pParams->config.bUsePrbs15 为 1，则应通过 XOR 运算生成序列，其中用于数据包类型 3 的 PRBS15 序列的每个字节与 pParams->byteVal 进行该运算。

如果使用任一 PRBS 序列，则无论白化参数中的设置如何，白化都将被禁用。

结束操作的触发器由 pParams->endTrigger 和 pParams->endTime 设置。如果发生此参数定义的触发，无线电操作应尽快结束。如果在等待数据包之间发生这种情况，则应立即结束操作。如果在另一个时间发生，则应继续操作，直到完全传输当前数据包，然后结束。如果无线电 CPU 接收到立即命令 CMD_STOP（请参阅节 25.3.3.2.2），它应与发生的结束触发器具有相同的含义，但结束后的状态代码应为 CMD_DONE_STOPPED。

输出结构 pOutput 仅包含字段 nTx。每次发送数据包时，nTx 字段都应递增。无线电 CPU 不应初始化字段，因此当需要复位计数器时，必须由系统 CPU 执行此操作。每次发送数据包时都应发出 Tx_Done 中断。

PHY 测试 TX 操作将以表 25-166 中列出的状态之一结束。操作结束后，命令结构的状态字段指示操作结束的原因。在所有情况下，都会产生 Command_Done 中断。在每种情况下，都会指示 TRUE、FALSE 或 ABORT 的结果，这些结果将决定下一个操作，如节 25.3.3.2.2 中所定义。

表 25-166. PHY 测试 TX 操作结束

条件	状态代码	结果
已发送的 pParams->numPackets 数据包	BLE_DONE_OK	TRUE
在数据包之间等待时，pParams->endTrigger 指示观察到触发	BLE_DONE_ENDED	FALSE
观察到的触发由 pParams->endTrigger 指示，然后完成发送数据包	BLE_DONE_ENDED	FALSE
在数据包之间等待时观察到 CMD_STOP	BLE_DONE_STOPPED	FALSE
观察到 CMD_STOP，然后完成数据包发送	BLE_DONE_STOPPED	FALSE
收到了 CMD_ABORT	BLE_DONE_ABORT	ABORT
通道值非法	BLE_ERROR_PAR	ABORT
pParams->packetType 的值非法	BLE_ERROR_PAR	ABORT

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.8.14 白名单处理

在广播器、扫描仪及启动器操作中使用白名单。白名单包含数量可配置的条目。白名单是表 25-118 中定义的类型的条目数组。数组的第一个条目应在大小字段中包含数组大小。

白名单数组中的最低条目数为 1，但如果不使用白名单，则 pParams->pWhiteList 可能为 NULL。条目的最大数量受到无线电 CPU 进行滤波时获得的性能的限制。测试表明，最多可以使用至少 24 个条目。

每个条目包含一个地址及三个配置位。如果启用了该条目，则 bEnable 位为 1，否则在执行白名单筛选时应忽略该地址。addrType 位指示该条目是公共地址还是随机地址。扫描仪可以使用 bIgnore 位来避免多次报告和扫描同一器件。

当要根据白名单检查地址时，会将地址与白名单中每个条目的地址字段进行比较。只有在以下情况下，地址才会出现在白名单中：

- bEnable 为 1
- addrType 等于要检查的地址的地址类型
- 地址数组的所有字节等于要检查的地址的字节
- 仅适用于扫描仪：bWIIgn 为 0
- 仅对于启动器：bIrkValid 为 0

对于扫描仪，可以在白名单中设置 bWIIgn 位，以指示即使白名单条目不匹配，器件也应被忽略。这可用于检查已扫描的广播器或已报告广播数据的位置。即使不执行白名单筛选，也可以使用此特性。对于与地址和地址类型匹配且 bWIIgn 为 1 的器件，应扫描白名单。应忽略此类器件。在这种情况下，不应检查 bEnable 位。接收旧数据包时始终执行检查，而对于扩展广播器数据包，检查仅在 pParams->extFilterConfig.bApplyDuplicateFiltering 为 1 且扩展标头中不存在 ADI 字段时执行。可以将无线电 CPU 配置为自动设置 bWIIgn 位，请参阅节 25.8.10。

对于启动器，可以在白名单中设置 bIrkValid 位，以指示器件应被忽略，即使白名单条目是匹配的。这可用于避免连接到存在有效身份解析密钥 (IRK) 的具有可解析私有地址的器件。bIrkValid 位绝对不应由系统 CPU 为不包含 RPA 的地址条目设置。

25.8.15 退避过程

在接收或尝试接收 SCAN_RSP、AUX_SCAN_RSP 或 AUX_CONNECT_RSP 数据包后，退避参数应由无线电 CPU 更新。更新取决于响应数据包结果和退避参数的旧值。pParams->backoffPar 中给出的退避参数应按表 25-167 中所示进行更新。更新完成后，无线电 CPU 应将 pParams->backoffCount 设置为介于 1 和 2 $^{pParams->backoffPar.logUpperLimit}$ 之间的假随机数。

表 25-167. 更新退避参数

响应数据包结果	旧 pParams->backoffPar		新建 pParams->backoffPar		
	bLastSucceeded	bLastFailed	bLastSucceeded	bLastFailed	logUpperLimit
失败	X	0	0	1	logUpperLimit
失败	0	1	0	0	min(logUpperLimit+1, 8)
成功	0	X	1	0	logUpperLimit
成功	1	0	0	0	max(logUpperLimit-1, 0)

假随机算法应基于最大长度的 16 位线性反馈移位寄存器 (LFSR)。种子应按照 pParams->randomState 中的规定。当操作结束时，无线电 CPU 应将当前状态写回此字段。如果 pParams->randomState 为 0，则无线电 CPU 应通过将 LFSR 初始化为无线电计时器的 16 个最低有效位来自行播种。只有在首次需要 LFSR（即收到 ADV*_IND 之后）时才应执行此操作，因此该值会有一定的随机性。如果无线电计时器的 16 个最低有效位全部为 0，则应替换另一个固定值。

当器件进入扫描或启动（仅限蓝牙 5）状态时，系统 CPU 应将 pParams->backoffCount 初始化为 1、将 pParams->backoffPar.logUpperLimit 初始化为 0、将 pParams->backoffPar.bLastSucceeded 和 pParams-

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

>backoffPar.bLastFailed 初始化为 0、将 pParams->randomState 初始化为真随机数字（或基于真随机种子初始化为假随机数）。当在保持扫描状态的情况下启动新的扫描仪操作时，系统 CPU 应将 pParams->randomState、pParams->backoffCount 和 pParams->backoffPar 保持为上次扫描仪操作结束时的值。

25.8.16 AUX 指针处理

如果接收到包含 Aux 指针的扩展广播器数据包，则无线电 CPU 应跟随 Aux 指针或返回有关在后续命令中如何遵循 Aux 指针的信息。在以下任何情况下均不会跟随 Aux 指针：

- 数据包存在 CRC 错误
- 忽略数据包
- 接收缓冲器已满
- 数据包是可扫描或可连接的 AUX_ADV_IND 数据包或 AUX_CONNECT_RSP 数据包（这些数据包不应有 Aux 指针）

如果遵循 Aux 指针，则读取 AUX 偏移量并将派生值存储在参数结构中。pParams->rxStartTime 设置为接收器为了接收数据包而应启动的时间，同时考虑到接收到的数据包的长度、时钟不准确性以及 PHY 用于下一个数据包的接收器启动时间。pParams->rxListenTime 设置为从 pParams->rxStartTime 到接收操作超时的时间。pParams->channelNo 设置为指向的次级通道，而 pParams->phyMode 指示要使用的 PHY。

如果从包含 Aux 指针的数据包结束到 pParams->rxStartTime 中给出的时间的时间大于 pParams->maxWaitTimeForAuxCh，该操作将在接收到状态为 BLE_DONE_AUX 的数据包后结束。然后由系统 CPU 在指定的时间（通常在处于低功耗模式之后）在次级通道上启动新的扫描仪或启动器命令。此命令应以 pParams->rxStartTime - startToSynthRatOffset 的绝对启动时间启动，其中 startToSynthRatOffset 是固件定义的参数。该命令应在 pParams->rxStartTime + pParams->rxListenTime 时具有一个超时触发器。

如果 pParams->rxStartTime 中给出的时间早于此时间，无线电 CPU 应自动安排通道和 PHY 切换。在合成器重新编程并切换 PHY 模式后，接收器将在使用 Aux 指针接收到数据包后停止，并在 pParams->rxStartTime 重新启动。如果从接收到的数据包结束到 pParams->rxStartTime 的时间太短，接收器将尽快启动。接收器将按照节 25.8.10.3 中为扫描器所述或节 25.8.11.3 中为启动器所述而运行。它将在 pParams->rxListenTime 之后有一个超时，如果超时，则操作以状态 BLE_DONE_RXTIMEOUT 结束。

25.8.17 器件地址的动态更改

对于广播器、扫描器和启动器命令，可以在命令运行时更改器件地址。如果将新指针写入广播器、扫描器或启动器参数结构的 pDeviceAddress 或 pWhiteList 字段，则该指针将在以下时刻被用于设备地址或对等地址：

- 对于广播器：下一次启动广播器命令（在命令链中）时。对于已在运行的命令，地址不会更新。
- 对于扫描器和启动器：下一次在广播者数据包上发现同步时。

在使用新的指针之前，上一个指针将用于器件地址。

advConfig、scanConfig 和 initConfig 字段与器件地址指针同时被重新读取。但是，不建议在命令或命令链运行时更改这些字段中的器件地址类型或对等地址类型，因为地址类型与地址不同步的可能性很小。作为替代方案，可以将 pDeviceAddress 的最低有效位设置为 1，以指示器件地址类型应与 advConfig、scanConfig 或 initConfig 中的类型相反。类似地，可以将 pWhiteList 的最低有效位设置为 1，以指示对端地址类型应与 advConfig 或 initConfig 中的类型相反。这意味着地址和地址类型可以在一个原子操作中进行修改。在查找存放地址的数组的地址时，pDeviceAddress 和 pWhiteList 的最低有效位将被忽略；无线电 CPU 将始终从半字对齐的地址读取。

备注

在命令运行时，不应修改 pDeviceAddress 或 pWhiteList（当指向单个地址时）所指向的数组的内容。如果命令结构中的指针已更新，则在确认 pDeviceAddress 或 pWhiteList 旧值所指向的地址不再被无线电 CPU 使用之前，不应修改该数组。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.9 即时命令

除其他可用的即时命令外，还应支持即时命令*更新广播有效载荷命令*（有关更多详细信息，请参阅节 25.9.1）。

25.9.1 更新广播有效载荷命令

CMD_BLE_ADV_PAYLOAD 命令可用于更改旧广播命令的有效载荷缓冲器。无论广播命令是否正在运行，都可能发出此消息。

命令结构应具有表 25-95 中给出的格式。收到后，无线电 CPU 应使用即时命令结构的 pParams 中给出的参数结构检查广播器无线电操作命令是否正在运行。否则，无线电 CPU 应立即更新参数结构。如果正在运行使用待更新的参数结构的无线电操作命令，则无线电 CPU 只能在当前未传输待更改的有效载荷时才应修改参数结构。如果正在传输，则无线电 CPU 应在数据包传输完成后立即存储请求并进行更新。

更新参数结构时，要更改的有效载荷取决于命令结构的 payloadType 参数。如果 payloadType 为 0，则无线电 CPU 应将 pParams->advLen 设置为等于 newLen，将 pParams->pAdvData 设置为等于 newData。如果 payloadType 为 1，则无线电 CPU 应将 pParams->scanRspLen 设置为等于 newLen，并将 pParams->pScanRspData 设置为等于 newData。更新发生后，无线电 CPU 应发出 Tx_Buffer_Changed 中断，请参阅表 25-124。无论更新是否延迟，都应发出此中断。

如果任何参数是非法的，无线电 CPU 应在 CMDSTAT 中以 ParError 响应，而不执行任何更新。否则，无线电 CPU 应在 CMDSTAT 中以 Done 进行响应。这可以在更新之前完成。

提供此命令是为了提供有效载荷和长度的原子变化。对于蓝牙 5 广播器命令，不需要此命令。可以随时安全地更改指针 pAdvPkt 和 pRspPkt；但是，在命令结束之前，不得修改指向的数据包条目。

25.10 专有无线电

本部分介绍专有无线电命令结构、数据处理、无线电操作命令和立即命令。这些命令定义了灵活的数据包处理，兼容 CC110x、CC111x、CC112x、CC120x、CC2500 和 CC251x 器件，并支持其他传统模式。

25.10.1 数据包格式

为了与现有 TI 器件兼容，在大多数情况下可以使用图 25-9 中给出的数据包格式。该数据包格式通过使用命令 CMD_PROP_TX 和 CMD_PROP_RX 来支持。

1 bit to 32 bytes	8 to 32 bits	0 or 1 byte	0 or 1 byte	0 to 255 bytes	0 or 16 bits (0 to 32 bits)
Preamble	Sync word	Length field	Address	Payload	CRC

图 25-9. 标准数据包格式

还支持一种更灵活的数据包格式，如图 25-10 中所定义。此格式由命令 CMD_PROP_RX_ADV 和 CMD_PROP_TX_ADV 支持。此格式的示例见图 25-9。

1 bit to 32 bytes or repetition	8 to 32 bits	0 to 32 bits	0 to 8 bytes	Arbitrary	0 or 16 bits (0 to 32 bits)
Preamble	Sync word	Header	Address	Payload	CRC

图 25-10. 高级数据包格式

25.10.2 命令

表 25-168 定义专有无线电操作命令。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-168. 专有无线电操作命令

ID	命令名称	说明
0x3801	CMD_PROP_TX	传输数据包
0x3802	CMD_PROP_RX	接收一个或多个数据包
0x3803	CMD_PROP_TX_ADV	使用高级模式传输数据包
0x3804	CMD_PROP_RX_ADV	接收一个或多个具有高级模式的数据包
0x3805	CMD_PROP_CS	运行载波传感命令
0x3806	CMD_PROP_RADIO_SETUP	以专有模式设置无线电
0x3807	CMD_PROP_RADIO_DIV_SETUP	以专有模式设置无线电
0x3808	CMD_PROP_RX_SNIFF	接收一个或多个支持监听模式的数据包
0x3809	CMD_PROP_RX_ADV_SNIFF	接收能够支持高级模式和监听模式的一个或多个数据包

表 25-169 定义专有立即命令。

表 25-169. 专有立即数命令

ID	命令名称	说明
0x3401	CMD_PROP_SET_LEN	设置接收的数据包的长度
0x3402	CMD_PROP_RESTART_RX	停止接收数据包，然后返回同步搜索

25.10.2.1 命令数据定义

本节定义了在描述系统 CPU 和无线电 CPU 之间的通信所使用的数据结构时采用的数据类型。这些数据结构以表格形式列出。字节索引是相对于指向该结构的指针的偏移量。多字节字段为小端字节序，且需要半字或字对齐。对于位编号，0 表示 LSB。R/W 列的含义如下：

R：系统 CPU 可以读回结果；无线电 CPU 不读取该字段。

W: 系统 CPU 写入一个值，无线电 CPU 读取该值且不会修改。

R/W：系统 CPU 写入初始值，无线电 CPU 可能会修改该初始值。

25.10.2.1.1 命令结构

对于所有无线电操作命令，前 14 个字节的定义见表 25-9。

表 25-170. CMD_PROP_TX 命令结构

字节索引	字段名称	位	位字段名称	类型	说明
14	pktConf	0	bFsOff	W	0：执行命令后使频率合成器保持打开状态。 1：执行命令后关闭频率合成器。
		1-2			保留
		3	bUseCrc	W	0：不附加 CRC。 1：附加 CRC。
		4	bVarLen	W	0：固定长度 1：在第一个字节中传输长度
		5-7			保留
15	pktLen			W	数据包长度
16-19	syncWord			W	要传输的同步字
20-23	pPkt			W	指向数据包的指针

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-171. CMD_PROP_TX_ADV 命令结构

字节索引	字段名称	位	位字段名称	类型	说明
14	pktConf	0	bFsOff	W	0：执行命令后使频率合成器保持打开状态。 1：执行命令后关闭频率合成器。
		1-2			保留
		3	bUseCrc	W	0：不附加 CRC。 1：附加 CRC。
		4	bCrcIncSw	W	0：在 CRC 计算中不包括同步字。 1：在 CRC 计算中包括同步字。
		5	bCrcIncHdr	W	0：在 CRC 计算中不包括标头。 1：在 CRC 计算中包括标头。
		6-7			保留
15	numHdrBits			W	标头中的位数 (0 – 32)
16-17	pktLen			W	数据包长度。0：无限
18	startConf	0	bExtTxTrig	W	0：命令启动触发条件发生后的固定时间开始发送数据包。 1：通过外部触发条件启动数据包发送（请联系 TI 启用此特性）。
		1-2	inputMode	W	如果使用外部触发条件启动 TX，则为输入模式。 00：上升沿 01：下降沿 10：两个边沿 11：保留
		3-7	source	W	如果使用外部触发条件启动 TX，则为用于捕获的 RAT 输入事件编号。
19	preTrigger			W	从前导码切换到同步字的触发条件。如果设置为“now”，则发送一个在设置中配置的前导码。否则，重复发送前导码直到检测到此触发条件。
20-23	preTime			W	preTrigger 的时间参数
24-27	syncWord			W	要传输的同步字
28-31	pPkt			W	指向数据包的指针，或无限长度的 TX 队列

表 25-172. CMD_PROP_RX 和 CMD_PROP_RX_SNIFF 命令结构

字节索引	字段名称	位	位字段名称	类型	说明
14	pktConf	0	bFsOff	W	0：执行命令后使频率合成器保持打开状态。 1：执行命令后关闭频率合成器。
		1	bRepeatOk	W	0：正确接收数据包后结束操作。 1：正确接收数据包后返回同步搜索。
		2	bRepeatNok	W	0：收到包含 CRC 错误的数据包后结束操作。 1：收到包含 CRC 错误的数据包后返回同步搜索。
		3	bUseCrc	W	0：不检查 CRC。 1：检查 CRC。
		4	bVarLen	W	0：固定长度 1：在第一个字节中接收长度。
		5	bChkAddress	W	0：不检查地址。 1：检查地址。
		6	endType	W	0：如果在获得同步后发生结束触发条件，则接收数据包直至结束。 1：如果发生结束触发条件，则停止接收数据包。
		7	filterOp	W	0：在地址不匹配时停止接收器并重新启动同步搜索。 1：在地址不匹配时接收数据包并将其标记为忽略。
15	rxConf			W	RX 配置，请参阅表 25-179 了解详情
16-19	syncWord			W	要侦听的同步字

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-172. CMD_PROP_RX 和 CMD_PROP_RX_SNIFF 命令结构（续）

字节索引	字段名称	位	位字段名称	类型	说明
20	maxPktLen			W	固定长度时为数据包长度，可变长度时为最大数据包长度 0：长度不受限制或未知
21	address0			W	地址
22	address1			W	地址（设置为等于 address0 以仅接受一个地址。如果是 0xFF，也接受 0x00）
23	endTrigger			W	用于结束操作的触发条件分类器
24-27	endTime			W	结束操作的时间
28-31	pQueue			W	指向接收队列的指针
32-35	pOutput			W	指向输出结构的指针
36-47	仅限 CMD_PROP_RX_SNIFF：表 25-178 中给出的载波侦听选项				

表 25-173. CMD_PROP_RX_ADV 和 CMD_PROP_RX_ADV_SNIFF 命令结构

字节索引	字段名称	位	位字段名称	类型	说明
14	pktConf	0	bFsOff	W	0：执行命令后使频率合成器保持打开状态。 1：执行命令后关闭频率合成器。
		1	bRepeatOk	W	0：正确接收数据包后结束操作。 1：正确接收数据包后返回同步搜索。
		2	bRepeatNok	W	0：收到包含 CRC 错误的数据包后结束操作。 1：收到包含 CRC 错误的数据包后返回同步搜索。
		3	bUseCrc	W	0：不检查 CRC。 1：检查 CRC。
		4	bCrcIncSw	W	0：在 CRC 计算中不包括同步字。 1：在 CRC 计算中包括同步字。
		5	bCrcIncHdr	W	0：在 CRC 计算中不包括标头。 1：在 CRC 计算中包括标头。
		6	endType	W	0：如果在获得同步后发生结束触发条件，则接收数据包直至结束。 1：如果发生结束触发条件，则停止接收数据包。
		7	filterOp	W	0：在地址不匹配时停止接收器并重新启动同步搜索。 1：在地址不匹配时接收数据包并将其标记为忽略。
15	rxConf			W	RX 配置（有关详细信息，请参阅表 25-179）
16-19	syncWord0			W	要侦听的同步字
20-23	syncWord1			W	替代同步字（如果非零）
24-25	maxPktLen			W	收到的数据包的最大长度 0：长度不受限制或未知
26-27	hdrConf	0-5	numHdrBits	W	标头中的位数 (0 – 32)
		6-10	lenPos	W	标头中长度字段的位置 (0 – 31)
		11-15	numLenBits	W	长度字段中的位数 (0 – 16)
28-29	addrConf	0	addrType	W	0：地址位于标头后 1：地址位于标头中
		1-5	addrSize	W	如果 addrType = 0：地址大小（以字节为单位）。如果 addrType = 1：地址大小（以位为单位）。
		6-10	addrPos	W	如果 addrType = 1：标头中地址的位位置 如果 addrType = 0：非零则用同步字标识符来扩展地址
		11-15	numAddr	W	地址列表中的地址数量
30	lenOffset			W	要添加到长度字段的有符号值
31	endTrigger			W	用于结束操作的触发条件分类器

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-173. CMD_PROP_RX_ADV 和 CMD_PROP_RX_ADV_SNIFF 命令结构（续）

字节索引	字段名称	位	位字段名称	类型	说明
32-35	endTime			W	结束操作的时间
36-39	pAddr			W	指向地址列表的指针
40-43	pQueue			W	指向接收队列的指针
44-47	pOutput			W	指向输出结构的指针
48-59	仅限 CMD_PROP_RX_ADV_SNIFF：表 25-178 中给出的载波侦听选项				

表 25-174. CMD_PROP_CS 命令结构

字节索引	字段名称	位	位字段名称	类型	说明
14	csFsConf	0	bFsOffIdle	W	0：如果命令以通道空闲状态结束，则保持合成器运行。 1：如果命令以通道空闲状态结束，则关闭合成器。
		1	bFsOffBusy	W	0：如果命令以通道忙碌状态结束，则保持合成器运行。 1：如果命令以通道忙碌状态结束，则关闭合成器。
15					保留
16-27	表 25-178 中给出的载波侦听选项。				

表 25-175. CMD_PROP_RADIO_SETUP 和 CMD_PROP_RADIO_DIV_SETUP 命令结构

字节索引	字段名称	位	位字段名称	类型	说明
14-15	modulation	0-2	modType	W	0：FSK 1：GFSK 其他：保留
		3-13	deviation	W	偏差（步长由 deviationStepSz 给出）
		14-15	deviationStepSz		偏差步长： 0：250Hz 1：1000Hz 2：15.625Hz 3：62.5Hz
16-19	symbolRate	0-7	preScale	W	预分频器值（请参阅节 25.10.5.2）
		8-28	rateWord	W	速率字（请参阅节 25.10.5.2）
		29-31			保留，设置为 0
20	rxBw			W	接收器带宽（请参阅表 25-183） 1 – 18：传统模式（带宽 87 – 4,236kHz） 32 – 52：正常模式（带宽 45 – 4,236kHz） 64 – 103：增强模式（带宽 4.8 – 4,236kHz）
21	preamConf	0-5	nPreamBytes	W	0：1 个前导码位 1-16：前导码字节数 18、20、...、30：前导码字节数 31：4 个前导码位 32：32 个前导码字节 其他：保留
		6-7	preamMode	W	00：发送 0 作为第一个前导码位 01：发送 1 作为第一个前导码位 10：前导码和同步字中发送相同的第一位 11：前导码和同步字中发送不同的第一位

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-175. CMD_PROP_RADIO_SETUP 和 CMD_PROP_RADIO_DIV_SETUP 命令结构（续）

字节索引	字段名称	位	位字段名称	类型	说明
22-23	formatConf	0-5	nSwBits	W	同步字位数。有效值为 8 – 32
		6	bBitReversal	W	0：对 1 使用正偏差 1：对 0 使用正偏差
		7	bMsbFirst	W	0：首先传输最低有效位 1：首先传输最高有效位
		8-11	fecMode	W	选择编码 0000：未编码的二进制调制 1000：远距离模式 1010：曼彻斯特编码的二进制调制 (FSK/GFSK) 其他：保留
		12			保留
		13-15	whitenMode	W	000：无白化 001：符合 CC1101 和 CC2500 标准的白化 010：PN9 白化，无字节反转 011：保留 100：无白化器、32 位、兼容 IEEE 802.15.4g 的 CRC 101：兼容 IEEE 802.15.4g 白化器和 32 位 CRC 110：无白化器、动态、兼容 IEEE 802.15.4g 的 16 位或 32 位 CRC 111：动态、兼容 IEEE 802.15.4g 的白化器以及 16 位或 32 位 CRC
24-25	config	0-2	保留	W	
		3	保留	W	
		4-9	保留	W	
		10	保留	W	
		11	保留	W	
		12	保留	W	
		13-15	保留		
26-27	txPower			W	输出功率设置，使用来自 SmartRF Studio 的值。有关更多详细信息，请参阅节 25.3.3.2.16。 0xFFFF：使用 20dBm PA（仅限“P”型号器件）
28-31	pRegOverride			W	指向要覆盖的硬件和配置寄存器列表的指针。如果为 NULL，则不使用覆盖。
32-33	centerFreq			W	仅限 CMD_PROP_RADIO_DIV_SETUP：频带的中心频率。用于初始参数计算。
34-35	intFreq			W	仅限 CMD_PROP_RADIO_DIV_SETUP：用于 RX 的中频，以 MHz 为单位，采用 4.12 有符号格式。如果支持，TX 将使用相同的中频，否则为 0。 0x8000：使用默认值
36	loDivider			W	仅限 CMD_PROP_RADIO_DIV_SETUP：要使用的分频器设置。有关每个器件和频带的建议设置，请参阅 Smart RF Studio。

表 25-176. CMD_PROP_SET_LEN 命令结构

字节索引	字段名称	位	位字段名称	类型	说明
0-1	RXLen			W	使用的有效载荷长度

25.10.2.2 输出结构

表 25-177. 接收命令

字节索引	字段名称	类型	说明
0-1	nRxOk	R/W	已收到且包含有效载荷、CRC 正确且未被忽略的数据包数量

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-177. 接收命令（续）

字节索引	字段名称	类型	说明
2-3	nRxNok	R/W	已收到但 CRC 错误的数据包数量
4	nRxIgnored	R/W	已收到、CRC 正确但因地址不匹配而被忽略的数据包数量
5	nRxStopped	R/W	长度非法或地址不匹配且 pktConf.filterOp = 1 而未收到的数据包数量
6	nRxBufFull	R/W	已收到但因缓冲区空间不足而被丢弃的数据包数量
7	lastRssi	R	最后收到的数据包的 RSSI。RSSI 在找到同步字时捕获。
8-11	timeStamp	R	最后收到的数据包的时间戳

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.10.2.3 其他结构和位字段

表 25-178. CMD_PROP_RX_SNIFF、CMD_PROP_RX_ADV_SNIFF 和 CMD_PROP_CS 的载波侦听字段

字节索引	字段名称	位	位字段名称	类型	说明
0	csConf	0	bEnaRssi	W	如果为 1，则启用 RSSI 作为标准。
		1	bEnaCorr	W	如果为 1，则启用关联值作为标准。
		2	操作	W	0：如果 RSSI 或关联值指示 BUSY，则忙碌。 1：如果 RSSI 和关联值都指示 BUSY，则忙碌。
		3	busyOp	W	0：在通道处于 BUSY 状态时继续执行载波侦听。 1：在通道处于 BUSY 状态时结束载波侦听。 对于 RX 命令，接收器在载波侦听结束时继续运行，不会在通道进入 IDLE 状态时结束。
		4	idleOp	W	0：在通道处于 IDLE 状态时继续运行。 1：在通道处于 IDLE 状态时结束。
		5	timeoutRes	W	0：超时且通道状态 INVALID 视为 BUSY。 1：超时且通道状态 INVALID 视为 IDLE。
1	rssiThr			W	RSSI 阈值
2	numRssiIdle			W	通道声明 IDLE 之前所需的阈值以下连续 RSSI 测量次数。
3	numRssiBusy			W	通道声明 BUSY 之前所需的阈值以上连续 RSSI 测量次数。
4-5	corrPeriod			W	关联观察期间的 RAT 节拍数
6	corrConfig	0-3	numCorrInv	W	从 IDLE 变为 INVALID 所需的、彼此间隔最大 corrPeriod 个 RAT 节拍的连续关联峰值的数量。
		4-7	numCorrBusy	W	从 INVALID 变为 BUSY 所需的、彼此间隔最大 corrPeriod 个 RAT 节拍的连续关联峰值的数量。
7	csEndTrigger			W	用于结束载波侦听的触发条件分类器
8-11	csEndTime			W	结束载波侦听时间

表 25-179. 接收队列条目配置位字段 (1)

位	位字段名称	说明
0	bAutoFlushIgnored	如果为 1，则自动从 RX 队列中丢弃忽略的数据包。
1	bAutoFlushCrcErr	如果为 1，则自动从 RX 队列中丢弃 CRC 错误的数据包。
2		保留
3	bIncludeHdr	如果为 1，则在存储的数据包中包含收到的标头或长度字节；否则将其丢弃。
4	bIncludeCrc	如果为 1，则在存储的数据包中包含收到的 CRC 字段；否则将其丢弃。这要求 pktConf.bUseCrc 为 1。
5	bAppendRssi	如果为 1，则将 RSSI 字节附加到 RX 队列中的数据包上。
6	bAppendTimestamp	如果为 1，则将时间戳附加到 RX 队列中的数据包上。
7	bAppendStatus	如果为 1，则状态字节附加到 RX 队列中的数据包上。

(1) 该位字段用于参数结构的 rxConf 字节。

表 25-180. 接收状态字节位字段 (1)

位	位字段名称	说明
0-4	addressInd	找到的地址索引（如果不适用，则为 0）
5	syncWordId	0 表示主要同步字，1 表示备用同步字。
6-7	结果	00：收到的数据包正确，未忽略。 01：收到的数据包存在 CRC 错误。 10：收到的数据包正确，但可以忽略。 11：数据包接收中止。

(1) 如果配置了该位字段，则会将其中一个字节附加到收到的条目中。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.10.3 中断

无线电 CPU 使用固件定义的中断将事件信号发送回系统 CPU。表 25-181 列出了专有命令要使用的中断。每个中断都可在系统 CPU 中单独启用。节 25.10.4 和节 25.10.5 详细说明何时生成中断。

表 25-181. 中断定义

中断号	中断名称	说明
0	COMMAND_DONE	无线电操作命令已完成
1	LAST_COMMAND_DONE	命令链中的最后一个无线电操作命令已完成
10	TX_ENTRY_DONE	对于长度不受限制的数据包传输：从 TX 条目进行的读取完成
16	RX_OK	接收到的数据包，具有 CRC OK、有效载荷并且不能被忽略
17	RX_NOK	接收到的数据包，具有 CRC 错误
18	RX_IGNORED	接收到的数据包，具有 CRC OK，但将被忽略
22	RX_BUF_FULL	接收到的数据包，不适合 RX 缓冲器
23	RX_ENTRY_DONE	RX 队列数据条目，将状态更改为已完成
24	RX_DATA_WRITTEN	写入部分读取 RX 缓冲器的数据
25	RX_N_DATA_WRITTEN	写入部分读取 RX 缓冲器的指定字节数
26	RX_ABORTED	数据包接收在数据包完成之前停止
28	SYNTH_NO_LOCK	合成器已报告失去锁定
29	MODULES_UNLOCKED	在启动过程中，Arm Cortex-M0 处理器已打开对 RF 内核模块和存储器的访问权限
30	BOOT_DONE	RF 内核 CPU 引导完成
31	INTERNAL_ERROR	无线电 CPU 观察到意外错误

25.10.4 数据处理

对于专有模式 TX 命令，通过无线方式接收的数据将存储在接收队列中。支持部分读取 RX 缓冲器，对于无限长度来说是必需的。发送的数据从特定缓冲器中获取。

25.10.4.1 接收缓冲区

正在接收的数据包存储在接收缓冲区中。首先，根据 RX 条目中 config.lenSz 的配置，会存储一个长度字节或长度字。该长度是根据无线接收到的长度以及附加信息的配置计算得出的；对于部分读取 RX 缓冲区，则初始化为该段的最大可能大小，并在完成后设置为一个缓冲区中的段长度。

在可选长度字段之后，如果 rxConf.bIncludeHdr 为 1，则接收到的标头将按无线接收到的原始形式存储。此标头是 CMD_PROP_RX 的长度字节，对于 CMD_PROP_RX_ADV 来说是一个最多 32 位的字段。在后一种情况下，如果位数不是 8 的倍数，则标头的最后一个字节在 MSB 中补零填充，后跟接收到的地址（如果已配置）和有效载荷。

如果 rxConf.bIncludeCrc 为 1，则接收到的 CRC 值将存储在 RX 缓冲区中；否则，它不会进行存储，而仅用于检查 CRC 结果。如果 rxConf.bAppendRssi 为 1，则会附加一个字节，指示接收到的 RSSI 值。如果 rxConf.bAppendStatus 为 1，则会附加一个状态字节，其类型定义见表 25-180。如果 rxConf.bAppendTimeStamp 为 1，则附加一个指示数据包起始时间的时间戳。该时间戳对应于 ratmr_t 数据类型。虽然时间戳是多字节的，但不会进行字地址对齐，因此必须按字节写入和读取时间戳。

如果数据包的接收被中止，则数据包将立即从接收队列中移除，除非使用了部分读取 RX 条目。如果配置了 RSSI、时间戳和状态字段（除非没有更多的缓冲区空间可用），则会附加这些字段，并且状态字节会指示接收已中止。

图 25-11 显示了 RX 队列中条目元素的格式。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

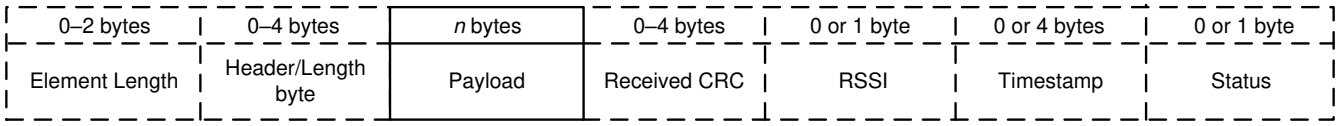


图 25-11. 接收缓冲区条目元素

当 RX 条目将其状态更改为“已完成”时，会产生 RX_ENTRY_DONE 中断。根据使用的 RX 条目类型，这意味着：

- 对于通用条目或指针条目，在数据包完全接收后会产生 RX_ENTRY_DONE 中断，除非数据包自动刷新。
- 对于多元素条目，当分配新的缓冲区并将新的条目投入使用时，或者当缓冲区完成并填充整个条目时，会产生 RX_ENTRY_DONE 中断。
- 对于部分读取条目，当 RX 条目已满时，会产生 RX_ENTRY_DONE 中断，因此必须在下一个条目中继续写入。

对于部分读取条目，每当将数据写入接收缓冲区时，会产生 RX_Data_Written 中断。自数据包开始以来，每当写入 config.irqIntv (在数据条目中给出) 字节的倍数时，会产生 RX_N_Data_Written 中断。

25.10.4.2 发送缓冲器

发送操作包含一个缓冲器，其中包含要发送的数据。此缓冲器中的字节数由 pktLen 给出。对于 CMD_PROP_TX 命令，如果 pktConf.bVarLen 为 1，则 pktLen 中给出的长度作为第一个字节发送，接着后跟发送缓冲器的内容。

对于 CMD_PROP_TX_ADV，如果标头长度大于 0，则缓冲器的前几个字节包含标头。字节数为标头中的位数除以 8 并且进行四舍五入。如果位数未除以 8，则不会发送最后标头字节的 MSB。如果要使用 CMD_PROP_TX_ADV 发送长度字段，则必须从系统端将其作为标头的一部分显式给出。

如果配置了不受限制的长度，则使用 TX 队列而不是一个缓冲器。在这种情况下，有效载荷传输会继续进行，直到队列清空。每次从一个条目完成传输时，意味着从下一个条目继续读取，或将整个有效载荷输入到调制解调器中时，都会产生 TX_ENTRY_DONE 中断。

25.10.5 无线电操作命令说明

在运行任何专有的 RX 或 TX 无线电操作命令之前，必须使用命令 CMD_PROP_RADIO_SETUP 或 CMD_PROP_RADIO_DIV_SETUP 将无线电设置为专有模式，或使用 CMD_RADIO_SETUP 将无线电设置为另一种兼容模式。否则，操作以错误结束。RX 和 TX 命令还需要使用 CMD_FS 命令对合成器进行编程，该命令通常由命令链完成，RX 或 TX 命令紧跟在 CMD_FS 之后。

25.10.5.1 操作结束

所发出命令的状态字段会在操作期间更新。提交命令时，系统 CPU 必须将该字段写为 IDLE 状态。操作期间，无线电 CPU 会更新该字段以指示操作模式。操作完成时，无线电 CPU 会写入一个状态来指示操作完成。表 25-182 列出了专有无线电操作所使用的状态代码。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-182. 专有无线电操作状态代码

编号	名称	说明
操作未完成		
0x0000	空闲	操作未开始
0x0001	挂起	等待启动触发条件
0x0002	运行	正在运行操作
操作正常完成		
0x3400	PROP_DONE_OK	操作正常结束
0x3401	PROP_DONE_RXTIMEOUT	在等待同步期间，在收到结束触发条件后停止操作
0x3402	PROP_DONE_BREAK	由于数据包中间发生超时，接收停止
0x3403	PROP_DONE_ENDED	在接收期间，在收到结束触发条件后停止操作
0x3404	PROP_DONE_STOPPED	收到停止命令后停止操作
0x3405	PROP_DONE_ABORT	收到中止命令后中止操作
0x3406	PROP_DONE_RXERR	收到包含 CRC 错误的数据包后结束操作
0x3407	PROP_DONE_IDLE	由于通道空闲，结束载波侦听操作
0x3408	PROP_DONE_BUSY	由于通道忙碌，结束载波侦听操作
0x3409	PROP_DONE_IDLETIMEOUT	由于超时且 csConf.timeoutRes = 1，结束载波侦听操作
0x340A	PROP_DONE_BUSYTIMEOUT	由于超时且 csConf.timeoutRes = 0，结束载波侦听操作
操作完成并出错		
0x3800	PROP_ERROR_PAR	非法参数
0x3801	PROP_ERROR_RXBUF	在数据包开始时，没有足够大的 RX 缓冲区来存放收到的数据
0x3802	PROP_ERROR_RXFULL	在部分读取缓冲区的接收过程中，RX 缓冲区空间不足
0x3803	PROP_ERROR_NO_SETUP	无线电未设置为专有模式
0x3804	PROP_ERROR_NO_FS	运行 RX 或 TX 时未对合成器进行编程
0x3805	PROP_ERROR_RXOVF	运行期间观察到 TX 溢出
0x3806	PROP_ERROR_TXUNF	运行期间观察到 TX 下溢

相关章节列出了每种操作处于每种给定状态的条件。表 25-182 中列出的一些错误原因在这些列表中不再重复。如果在等待启动触发条件时收到 CMD_STOP 或 CMD_ABORT，则结束原因为 DONE_STOPPED 或 DONE_ABORT，结束结果分别为 FALSE 和 ABORT。在某些情况下，可能会出现一般性错误原因。对于所有这些错误情况，操作的结果均为 ABORT。

25.10.5.2 专有模式设置命令

对于专有模式无线电，使用 CMD_PROP_RADIO_SETUP 和 CMD_PROP_RADIO_DIV_SETUP 命令来代替 CMD_RADIO_SETUP。当 CMD_PROP_RADIO_SETUP 或 CMD_PROP_RADIO_DIV_SETUP 执行时，会从 FCFG1 读取修整值、除非在其他位置提供了这些值（有关更多详细信息，请参阅节 25.3.3.1.2）。

在启动时，无线电 CPU 使用表 25-175 中提供的参数设置专有模式的参数。modulation.modType 参数选择 GFSK 和未成形 FSK。对于 FSK 和 GFSK，modulation.deviation 提供了可在 modulation.stepSize 中编程的步长偏差。无线电 CPU 使用此参数来计算适合在 TX 中使用的适当形状。

备注

由于硬件限制，偏差的精度可能小于编程的步长。

符号速率使用 symbolRate 进行编程。这些参数直接传递给调制解调器，并可使用外部工具进行计算。符号速率由方程式 15 提供。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

$$f_{baud} = (R \times f_{clk}) / (p \times 2^{21}) \quad (15)$$

其中

- f_{baud} 是获得的波特率
- f_{clk} 为 48 MHz 系统时钟频率
- R 是由 symbolRate.rateWord 提供的速率字
- P 是由 symbolRate.preScale 提供的预分频器值，范围为 4 到 255

rxBw 参数可提供接收器带宽。值 64 – 108 提供支持的带宽以及建议的设置。值 32 – 52 提供了其中的一个子集，支持与 CC13x0 的向后兼容。值 1 – 18 提供的带宽与设置 35 – 52 相同，以实现与 CC13x2 和 CC26x2 器件平台的兼容性。表 25-183 中总结了支持的值和相应的设置。接收器带宽针对 868MHz（LO 分频器 5）、915MHz（LO 分频器 5）和 2432MHz（LO 分频器 2）的 RF 频率进行规定。带宽与 RF 频率乘以 LO 分频器成正比。

表 25-183. 接收器带宽设置

设置（传统）	设置（正常）	设置（增强）	接收器带宽 (868MHz)	接收器带宽 (915MHz)	接收器带宽 (2432MHz)	默认中频 (kHz)
—	—	64	4.3	4.5	4.8	62.5
—	—	65	4.9	5.1	5.4	62.5
—	—	66	6.1	6.5	6.9	62.5
—	—	67	7.4	7.8	8.2	62.5
—	—	68	8.5	9.0	9.6	125
—	—	69	9.7	10.2	10.9	125
—	—	70	12.2	12.9	13.7	125
—	—	71	14.7	15.5	16.5	125
—	—	72	17.1	18.0	19.1	125
—	—	73	19.4	20.5	21.8	125
—	—	74	24.5	25.8	27.4	125
—	—	75	29.4	31.0	33.0	125
—	—	76	34.1	36.0	38.3	250
—	32	77	38.9	41.0	43.5	250
—	33	78	49.0	51.6	54.9	250
—	34	79	58.9	62.1	66.0	250
—	—	80	68.3	72.0	76.5	250
1	35	81	77.7	81.9	87.1	250
2	36	82	98.0	103.3	109.8	250
3	37	83	117.7	124.1	131.9	250
—	—	84	136.6	144.0	153.1	500
4	38	85	155.4	163.8	174.2	500
5	39	86	195.9	206.5	219.6	500
6	40	87	235.5	248.2	263.9	500
—	—	88	273.1	287.9	306.1	1000
7	41	89	310.8	327.6	348.3	1000
8	42	90	391.8	413.0	439.1	1000
9	43	91	470.9	496.4	527.8	1000
—	-	92	546.3	575.8	612.2	1000
10	44	93	621.6	655.3	696.7	1000
11	45	94	783.6	826.0	878.2	1000
12	46	95	941.8	992.8	1055.6	1000
—	—	96	1092.5	1151.7	1224.4	1000

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-183. 接收器带宽设置（续）

设置（传统）	设置（正常）	设置（增强）	接收器带宽 (868MHz)	接收器带宽 (915MHz)	接收器带宽 (2432MHz)	默认中频 (kHz)
13	47	97	1243.2	1310.5	1393.3	1000
14	48	98	1567.2	1652.1	1756.4	1000
15	49	99	1883.7	1985.7	2111.1	1000
—	—	100	2185.1	2303.4	2448.9	1000
16	50	101	2486.5	2621.1	2786.7	1000
17	51	102	3134.4	3304.2	3512.9	1000
18	52	103	3767.4	3971.4	4222.2	1000
其他			保留			

CMD_PROP_RADIO_DIV_SETUP 命令包含频段和中频的设置。要使用的频带的中心频率由 centerFreq 提供，并用于计算发送器整形滤波器和 TX IF。要在合成器中使用的分频器由 loDivider 提供。用户必须确保该设置与给定频率兼容。仅支持在 2.4GHz 频段运行的器件允许值 0 或 2，仅支持在低于 1GHz 频段运行的器件允许值大于 2。在 CMD_PROP_RADIO_SETUP 命令中，centerFreq 默认为 2432MHz，loDivider 默认为 0。

对于 CMD_PROP_RADIO_DIV_SETUP，可以通过 intFreq 参数指定中间频率，该参数计算调制解调器中的 RX 设置并写入配置参数区域。如果此参数为 0x8000，并且若针对的是 CMD_PROP_RADIO_SETUP，则使用表 25-183 中提供的默认中间频率。该器件用于检查 TX 是否支持已配置的中频。否则，TX 中间频率被设置为 0。如果中间频率大于 RF 中心频率除以 15500，就会发生这种情况。这导致合成器重新编程，而无需在 RX 和 TX 之间重新校准。这可能会增加必要的周转时间，或者在某些情况下导致频率合成器解除锁定，这意味着在 RX 和 TX 之间切换时需要重新校准。

preamConf 设置提供前导码。前导码是一个 1010...或 0101... 序列，其中 preamConf.preamMode 提供第一个传输位。如果超过 16 个字节，则只支持偶数个字节。设置 preamConf.nPreamBytes = 31 会产生 4 位前导码，设置 preamConf.nPremBytes = 0 会产生 1 位前导码。

formatConf 设置用于数据包格式的各种设置。同步字长由 nSwBits 提供，最高可达 32 位。FSK 类型调制的位极性由 bBitReversal 提供，它必须为 1 才能与 CC1101 兼容。位排序由 bMsbFirst 提供，其中 1 提供了与 CC1101 器件的兼容性，以此类推。whitenMode 设置可以选择一种白化方案。使用超控设置可获得其他白化器。有关 IEEE 802.15.4g 设置的详细信息，请参阅节 25.10.5.2.1。fecMode 设置可用于更改传输或接收信号的编码。对于远距离模式 (fecMode = 8)，nSwBits 设置和在 RX 和 TX 命令中编程的同步字将被忽略，并使用性能良好的硬编码 64 位同步字。将 fecMode 设置为 10 将启用曼彻斯特编码。仅支持有效载荷及 CRC 的编码/解码。0 将编码为 01b，1 将编码为 10b。

该命令使用多项式 $x^{16} + x^{15} + x^2 + 1$ 及全 1 初始化设置 16 位 CRC。此项与 CC1101 器件兼容。其他多项式、长度和初始化可以通过参数覆盖来获得。

pRegOverride 参数提供指向覆盖结构的指针，与为 CMD_RADIO_SETUP 提供的指针一样。该参数可用于覆盖通过主器件命令中的其他设置以及其他参数计算出的参数。如果值为 NULL，则不使用覆盖。

25.10.5.2.1 IEEE 802.15.4g 数据包格式

IEEE 802.15.4g PHY（包括标头）由 CMD_PROP_RX_ADV 和 CMD_PROP_TX_ADV 命令提供支持。

使用 CMD_PROP_RADIO_DIV_SETUP 命令，将 formatConf.whitenMode 字段设置为 4、5、6 或 7，并将 formatConf.bMsbFirst 设置为 1，即可将无线电配置为 IEEE 802.15.4g 模式。对于 CMD_PROP_TX_ADV 和 CMD_PROP_RX_ADV 命令，pktConf.bCrcIncSw 和 pktConf.bCrcIncHdr 均必须设置为 0。对于 CMD_PROP_RX_ADV，hdrConf.numHdrBits 必须设置为 16，hdrConf.lenPos 必须设置为 0，hdrConf.numLenBits 必须设置为 11，lenOffset 必须为 −4。

当 formatConf.whitenMode 为 5 或 7 时，配置无线电以生成 IEEE 802.15.4g 中定义的 32 位 CRC 并进行白化处理。当 formatConf.whitenMode 为 6 或 7 时，无线电还会在接收和传输过程中按如下方式处理标头：

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

- 如果标头的第 15 位（从 LSB 算起）为 1，则假定该帧仅由标头组成，不包含有效载荷或 CRC。
- 如果标头的第 12 位（从 LSB 算起）为 1，则假定使用 IEEE 802.15.4g 中定义的 16 位 CRC，而非 32 位 CRC。对于 TX，长度偏移会加上 2 以考虑此情况，前提是接收帧长度中包含 CRC。
- 对于模式 7：如果标头的第 11 位（从 LSB 算起）为 1，则启用白化；否则禁用白化。

备注

IEEE 802.15.4g PHY 标头必须以 MSB 优先的方式提供给射频内核。在 IEEE 802.15.4g 规范中，有效载荷部分为 LSB 优先，但物理层标头 (PHR) 中的有效载荷长度信息为 MSB 优先。这意味着需要在系统 CPU 中对有效载荷进行翻转。翻转可以通过系统 CPU 汇编指令 RBIT 来实现。

下面的示例展示了如何使用自动标头处理功能 (formatConf.whitenMode = 7) 发送启用了白化的 CRC-32 IEEE 802.15.4g 帧。

```
/*
 * Prepare the .15.4g PHY header
 * MS=0, Length MSBits=0, DW and CRC settings read from 15.4g header (PHDR) by RF core.
 * Total length = transmit_len (payload) + CRC length
 *
 * The Radio will flip the bits around, so tx_buf[0] must have the
 * length LSBs (PHR[15:8] and tx_buf[1] will have PHR[7:0]
 */
/* Length in .15.4g PHY HDR includes the CRC but not the HDR itself */
```

```
uint16_t total_length;
```

```
total_length = transmit_len + CRC_LEN; /* CRC_LEN is 2 for CRC-16 and 4 for CRC-32 */
```

```
tx_buf[0] = total_length & 0xFF;
tx_buf[1] = (total_length >> 8) + 0x08 + 0x0; /* Whitening and CRC-32 bits */
tx_buf[2] = data;
```

要支持 FEC、模式切换或 IEEE 802.15.4g PHY 的其他高级功能，需要使用 MCE 补丁。

25.10.5.3 发送器命令

有两个用于发送数据包的命令：CMD_PROP_TX 和 CMD_PROP_TX_ADV。后者在数据包形成方式上提供了更大的灵活性。其详细信息分别如节 25.10.5.3.1 和节 25.10.5.3.2 所述。

这两个命令都要求无线电以兼容模式（例如专有模式）进行设置，并且使用 CMD_FS 对合成器进行编程。

对于这两个命令，在数据包发送完毕后，如果 pktConf.bFsOff 为 1，则命令结束时频率合成器会被关闭。如果 pktConf.bFsOff 为 0，则合成器保持运行，因此该命令之后必须跟随一个 RX 或 TX 命令（在同一频率上运行），或者跟随一个 CMD_FS_OFF 命令来关闭合成器。

表 25-184 中列出了 CMD_PROP_TX 和 CMD_PROP_TX_ADV 的结束状态。该状态决定下一个操作，请参阅节 25.10.5.1。

表 25-184. 无线电 CMD_PROP_TX 和 CMD_PROP_TX_ADV 命令的结束状态

条件	状态代码	结果
已发送数据包	PROP_DONE_OK	真
在发送数据包时收到 CMD_STOP 并完成了数据包发送	PROP_DONE_STOPPED	否
发送数据包时收到 CMD_ABORT	PROP_DONE_ABORT	ABORT
观察到非法参数	PROP_ERROR_PAR	ABORT
未使用 CMD_PROP_RADIO_SETUP 或 CMD_RADIO_SETUP 在受支持模式下设置无线电即发送命令	PROP_ERROR_NO_SETUP	ABORT
未对合成器进行编程即发送命令	PROP_ERROR_NO_FS	ABORT

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-184. 无线电 CMD_PROP_TX 和 CMD_PROP_TX_ADV 命令的结束状态（续）

条件	状态代码	结果
操作期间发生 TX 下溢	PROP_ERROR_TXUNF	ABORT

25.10.5.3.1 标准传输命令 CMD_PROP_TX

CMD_PROP_TX 命令传输一个符合表 25-172 中格式的数据包。参数如表 25-168 中所示。

数据包发送在发生给定的启动触发条件时开始，并具有固定延迟。调制解调器首先按照配置传输前导码和同步字。要传输的同步字在 syncWord 字段中给出，如果使用的位数少于 32 位，则该字段位于 LSB 中。该字按照无线电中编程的位序传输。

如果 pktConf.bVarLen 为 1，则接下来发送一个等于 pktLen 值的长度字节。此后，发送 pPkt 所指向的缓冲区的内容。该缓冲区由 pktLen 中给定的字节数组成。如果需要图 25-9 中所示的地址字节，则必须将其作为第一个有效载荷字节发送。

如果 pktConf.bUseCrc 为 1，则在末尾计算并传输 CRC。CRC 位数、多项式和初始化值遵循无线电中的配置。对长度字节（如果存在）和 pPkt 指向的整个缓冲区内容计算 CRC。

如果启用了白化，则可选的长度字节、pPkt 指向的整个缓冲区内容以及 CRC 都将进行白化处理。白化在数据用于 CRC 计算之后进行。

25.10.5.3.2 高级发送命令，CMD_PROP_TX_ADV

CMD_PROP_TX_ADV 命令发送具有图 25-10 中格式的数据包。在特殊情况下，用户可以如图 25-9 中所示设置数据包。无线电必须设置为兼容模式（例如专有模式）并且使用 CMD_FS 对合成器进行编程。参数如表 25-173 中所示。

数据包传输从给定的启动触发开始，延迟固定。或者，如果 startConf.bExtTXTrig 为 1，则数据包传输在 RF 内核的外部触发时开始。触发器被识别为无线电计时器的输入之一，可以通过 startConf 参数配置为上升沿、下降沿或两个边沿。系统必须确保该触发信号在启动触发后发出，否则将丢失。启动触发后的最小延迟取决于实现，并取决于特性。

调制解调器首先传输配置的前导码和同步字。如果 preTrigger 不是 TRIG_NOW，则会重复配置的前导码，直到观察到该触发（与 preTime 结合使用时看到）。观察到触发后，传输下配置的前导码在同步字传输开始之前完成。如果 preTrigger 为 TRIG_NOW，则前导码将发送一次，然后发送同步字。要传输的同步字在 syncWord 字段中给出（如果使用的位少于 32 位，则在 LSB 中给出），并按照在无线电中编程的位顺序进行传输。

如果 numHdrBits 大于 0，则接下来会发送 numHdrBits 的标头。标头可包含长度字段或者地址。如果是，这些字段必须正确插入数据包缓冲器中。要传输的标头是 pPkt 指向的缓冲器的前几个字节。如果 numHdrBits 未除以 8，则标头最后一个字节的 MSB 将被忽略。

标头作为无线电中编程的位顺序中的一个字段来传输。如果标头具有 8 位以上，则始终按照小端字节序的顺序从发送缓冲器读取。如果将无线电配置为首先传输 MSB，则首先发送 TX 缓冲器的最后一个标头字节。

在标头之后，传输 pPkt 指向的缓冲器中的剩余字节。有效载荷是按字节传输的，因此在标头之后，无论通过无线方式进行的位排序如何，都不会发生字节交换。此缓冲器中的总字节数（包括标头）由 pktLen 给出。如果该长度太小而无法适应标头，则操作结束时将 PROP_ERROR_PAR 作为状态。如果需要如图 25-10 所示的标头之后的地址字段，则必须将其作为第一个有效载荷字节发送。

如果 pktLen 为 0，则使用不受限制的长度。在这种情况下，pPkt 指向传输队列而不是缓冲器，请参阅节 25.5.3.2。

如果 pktConf.bUseCrc 为 1，则计算 CRC 并在末尾发送。CRC 位的数量、多项式和初始化在无线电中配置。如果 pktConf.bCrcIncSw 为 1，则发送的同步字将包含在计算 CRC 的数据集中。如果 pktConf.bCrcIncHdr 为 1，则发送的标头将包含在计算 CRC 的数据集中。有效载荷始终用于计算 CRC。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

如果启用了白化，则如果 pktConf.bCrcIncHdr 为 1，则可选标头会进行白化处理。启用时，整个有效载荷和 CRC 始终会进行白化处理。在将数据用于 CRC 计算后进行白化。

25.10.5.4 接收器命令

有两条接收数据包的命令：CMD_PROP_RX 和 CMD_PROP_RX_ADV。后者在数据包形成方式上提供了更大的灵活性。有关详细信息，请分别参阅节 25.10.5.4.1 和节 25.10.5.4.2。

对于这两条命令，无线电必须设置为兼容模式（例如专有模式），并且在命令发送到无线电内核之前，必须使用 CMD_FS 对频率合成器进行编程。

这两条命令都有一个由 endTrigger 和 endTime 指定的结束触发条件。如果在接收器搜索同步时发生该触发条件，则操作以 PROP_DONE_RXTIMEOUT 状态结束。如果在接收数据包时发生该触发条件，则操作取决于 pktConf.endType。如果 pktConf.endType = 0，则数据包将接收完，然后操作以 PROP_DONE_ENDED 状态结束。如果 pktConf.endType = 1，则数据包接收中止，操作以 PROP_DONE_BREAK 状态结束。无线电分别根据节 25.10.5.4.1 和节 25.10.5.4.2 中给出的详细信息接收数据包。收到一个数据包后，就会引发中断。如果 pOutput 不为 NULL，则 pOutput 指向的输出结构（如表 25-173 所示）也会更新。表 25-185 中给出了要引发的中断和要更新的字段。该表还给出了要写入接收缓冲区状态字段的结果（如果启用）。节 25.10.5.4.1 和节 25.10.5.4.2 分别介绍了忽略数据包的条件。

表 25-185. 收到的数据包的中断、计数器和结果字段 (1)

条件	引发的中断	递增的计数器	状态字节的结果字段
数据包完整接收，CRC 正确且未忽略	RX_OK	nRxOk	0
数据包完成接收，存在 CRC 错误	RX_NOK	nRxNok	1
数据包完整接收，CRC 正确但地址不匹配 (pktConf.filterOp = 1)	RX_IGNORED	nRxIgnored	2
数据包接收因超时 (pktConf.endType = 1)、CMD_ABORT、CMD_PROP_SET_LEN 中的长度过短或 CMD_PROP_RESTART_RX 而中止	RX_ABORTED	nRxStopped	3 (1)
数据包接收因非法长度或地址不匹配而中止 (pktConf.filterOp = 0)	RX_ABORTED	nRxStopped	-
数据包因缺少缓冲区空间而无法存储	RX_BUF_FULL	nRxBufFull	3 (1)

(1) 提供的部分读取条目已使用，数据已写入缓冲区。

对于这两类命令，可以通过将 maxPktLen 设置为 0，以在数据包接收开始时将数据包长度配置为无限或未知。该模式只能与部分读取 RX 缓冲器配合使用。如果随后确定了长度，可通过即时或直接命令 CMD_PROP_SET_LEN 进行设置，该命令给出了标头（如有）与 CRC 之间的字节数。除通过这种方式设置长度外，还可通过以下方式停止数据包接收（以下情况不执行 CRC 校验）：

- 调用 CMD_PROP_SET_LEN 时给定的字节数小于已接收的字节数
- 已给出 CMD_PROP_RESTART_RX
- 没有更多的 RX 缓冲区可用
- 结束触发条件发生且 pktConf.endType 为 1
- 通过 CMD_ABORT 中止了命令

对于被忽略的数据包和存在 CRC 错误的数据包，可配置接收缓冲区自动刷新。在这种情况下，数据包在接收后会从接收缓冲区中移除，因此下一个数据包会将其覆盖，且计数器不会更新以反映该数据包已接收。

备注

部分读取 RX 条目不支持自动刷新。如果 rxConf.bAutoFlushCrcErr 为 1，则存在 CRC 错误（即引发 RX_NOK 中断）的数据包会自动刷新。

如果 rxConf.bAutoFlushIgnored 为 1，则被忽略（即引发 RX_IGNORED 中断）的数据包会自动刷新。收到数据包后，下一步操作取决于 pktConf.bRepeat。如果该值为 0，则命令结束。否则，它返回 RX，除非存在导致命令

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

结束的其他条件。命令结束时，如果 pktConf.bFsOff 为 1，则频率合成器关闭。如果 pktConf.bFsOff 为 0，则合成器保持运行，因此该命令之后必须跟一条 RX 或 TX 命令（工作在同一频率）或一条 CMD_FS_OFF 命令来关闭合成器。

表 25-186 中列出了 CMD_PROP_RX 和 CMD_PROP_RX_ADV 的结束状态。该状态决定下一个操作，请参阅节 25.10.5.1。

表 25-186. 无线电 CMD_PROP_RX 和 CMD_PROP_RX_ADV 命令结束

条件	状态代码	结果
收到数据包，CRC 正确且 pktConf.bRepeatOk = 0	PROP_DONE_OK	真
收到数据包，CRC 错误且 pktConf.bRepeatNok = 0	PROP_DONE_RXERR	否
在同步搜索期间观察到结束触发条件	PROP_DONE_RXTIMEOUT	否
接收数据包期间观察到结束触发条件且 pktConf.endType = 1	PROP_DONE_BREAK	否
接收数据包期间观察到结束触发条件之后收到数据包，pktConf.endType = 0	PROP_DONE_ENDED	否
命令启动后收到 CMD_STOP，且如果找到同步字，则数据包已接收	PROP_DONE_STOPPED	否
命令启动后收到 CMD_ABORT	PROP_DONE_ABORT	ABORT
数据包开始时无可用 RX 缓冲区	PROP_ERROR_RXBUF	否
在部分读取缓冲区的接收过程中，RX 缓冲区空间不足	PROP_ERROR_RXFULL	否
观察到非法参数	PROP_ERROR_PAR	ABORT
发送命令前未使用 CMD_PROP_RADIO_SETUP 或 CMD_RADIO_SETUP 将无线电设置为受支持的模式	PROP_ERROR_NO_SETUP	ABORT
发送命令前未对频率合成器进行编程	PROP_ERROR_NO_FS	ABORT
运行期间观察到 TX 溢出	PROP_ERROR_RXOVF	ABORT

25.10.5.4.1 标准接收命令 CMD_PROP_RX

CMD_PROP_RX 命令用于接收符合
图 25-9 所示格式的数据包。其参数如表 25-174 中所示。

调制解调器配置接收器并开始侦听同步字。要侦听的同步字在 syncWord 字段中给出，如果使用的位少于 32 位，则位于 LSB 中。该字遵循无线电中编程的位顺序。

如果找到同步字，无线电 CPU 开始接收数据。如果 pktConf.bVarLen 为 1 且 maxPktLen 为非零，则假定下一个字节为长度字节。此长度字节与 maxPktLen 进行比较，如果大于该值，则停止接收并重新开始同步搜索。否则，这表示长度字节之后和 CRC 之前的字节数。如果 pktConf.bVarLen 为 0，则长度是固定的，接收器假定在同步字之后和 CRC 之前有 maxPktLen 个字节。如果 maxPktLen 为零，则长度不受限制，如节 25.10.5.4 开头所述。

如果 pktConf.bChkAddress 为 1，则接下来将检查地址字节。该地址字节将与 address0 和 address1 的值进行比较。如果只需要一个地址，则这两个字段必须设置为相同的值。如果 address1 为 0xFF，则还会与值 0x00 进行比较。要在不检查 0x00 的情况下检查 0xFF，address0 必须设置为 0xFF。如果地址字节与配置的地址不匹配，则进一步的处理取决于 pktConf.filterOp。如果该值为 0，则停止接收并重新开始同步搜索。如果该值为 1，则如同地址匹配一样接收该数据包，但会将其标记为“已忽略”。

如果数据包正在接收中，则数据会被放入接收缓冲区。此接收缓冲区可从 pQueue 指向的接收队列中找到。如果 pQueue 为 NULL，则永远不会存储数据包。

如果 pktConf.bUseCrc 为 1，则会在末尾接收并检查 CRC。CRC 的位数、多项式和初始化值取决于无线电中的配置。CRC 的计算覆盖长度字节（如果存在）、可选地址和有效载荷。如果 pktConf.bUseCrc 为 0，则按 CRC 正确的处理方式对待。

如果启用了白化，则可选长度字节、有效载荷（包括可选地址）和接收到的 CRC 都会进行解白化处理。解白化在 CRC 评估之前完成。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

如果将状态字节附加到数据包（rxConf.bAppendStatus 为 1），则其格式如下（请参阅表 25-180）。如果 pktConf.addressMode 为非零，则 addressInd 字段值如下：如果地址匹配 address0，则为 0；如果匹配 address1，则为 1；如果匹配 0x00 且启用了此地址，则为 2；如果匹配 0xFF 且启用了此地址，则为 3。否则，addressInd 为 0。对于 CMD_PROP_RX，syncWordId 字段始终为 0。结果字段根据表 25-186 写入。

25.10.5.4.2 高级接收命令 CMD_PROP_RX_ADV

CMD_PROP_RX_ADV 用于接收格式符合图 25-10 的数据包。作为特例，用户可以按图 25-9 中所示设置数据包。参数如表 25-175 中所示。

调制解调器配置接收器并开始监听同步字。要监听的同步字在 syncWord0 字段中给出，如果使用的位数少于 32 位，则该字段位于 LSB 中。该字按照无线电中编程的位序排列。如果 syncWord1 非零且 MCE 支持，接收器还会监听 syncWord1 字段中给出的同步字（格式相同）。当使用 CMD_PROP_RX_ADV_SNIFF 且 csConf.bEnaCorr 设置为 1 时，不能使用两个同步字。

如果检测到同步字，则无线电 CPU 开始接收数据。数据包可包含一个标头，其位数由 hdrConf.numHdrBits 指定，最多可达 32 位。如果标头中的位数不能被 8 整除，则认为标头由足够数量的字节组成，包含所有存储的位，如节 25.5.3.1 所示。此标头可包含长度字段或地址。

收到的数据包可以是固定长度或可变长度。如果 hdrConf.numLenBits 为 0 且 maxPktLen 非零，则数据包为固定长度，由标头之后到 CRC 之前的 maxPktLen 个字节组成。如果 hdrConf.numLenBits 大于 0，则从标头 LSB 起始的第 hdrConf.lenPos 位开始，读取长度为 hdrConf.numLenBits 的字段，并将其作为长度字段。将有符号数 lenOffset 与收到的长度相加，得出标头之后到 CRC 之前的字节数。如果此数值小于或等于 maxPktLen，则接收该数据包。如果 maxPktLen 为零，则长度不受限制，如节 25.10.5.4 开头所述。CMD_PROP_RX_ADV 的数据包长度定义与 CMD_PROP_TX_ADV 的定义不同，请参阅节 25.10.5.4.2，其中标头包含在数据包长度内。

支持两种地址。第一种选择是地址作为标头的一部分。在这种情况下，地址大小可以为 1 到 31 位。另一种选择是将地址放在标头之后。如果这样，地址由 1 到 8 个字节组成。要使用包含在标头中的地址，必须将 addrConf.addrType 设置为 1。地址的位数由 addrConf.addrSize 给出。这些位从标头的第一位起，按 addrConf.addrPos 指定的位序号进行读取。要使用标头之后的地址，必须将 addrConf.addrType 设置为 0。在这种情况下，地址的字节数由 addrConf.addrSize 给出。

收到的地址将与 pAddr 指向的地址列表进行比较。要与此列表比较的地址即为收到的地址。此外，如果满足以下条件之一，会将一个用于标识同步字的位作为 MSB 与地址相拼接：

- syncWord1 ≠ 0 且 addrConf.addrType = 1
- syncWord1 ≠ 0、addrConf.addrType = 0 且 addrConf.addrPos ≠ 0

如果收到的同步字为 syncWord0，则此额外位为 0；如果收到的同步字为 syncWord1，则为 1。地址列表中的条目大小为 8、16、32 或 64 位；选择能够容纳地址大小（如果适用，包括同步字标识位）的最小大小。地址列表中的条目数量由 addrConf.numAddr 给出。无线电 CPU 扫描地址列表中的地址，并将其与收到的地址进行比较。如果不匹配，则进一步的处理取决于 pktConf.filterOp。如果此位为 0，则停止接收并重新开始搜索同步字。如果此位为 1，则如同地址已匹配一样接收数据包，但将其标记为忽略。

如果 addrConf.addrSize 为零，则不使用任何地址。在这种情况下，pAddr 被忽略且必须为 NULL。

如果正在接收数据包，数据将放入接收缓冲区，如节 25.5.3.1 所示。此接收缓冲区位于 pQueue 指向的接收队列中。如果 pQueue 为 NULL，则从不存储数据包。

标头作为一个字段接收，遵循无线电中编程的位序。如果标头超过 8 位且 rxConf.bIncludeHdr 为 1，则标头始终以小端字节序写入接收缓冲区。如果无线电配置为先接收 MSB，则接收缓冲区中存储的最后一个标头字节最先被接收。有效载荷逐字节存储，因此在标头之后，无论无线传输的位序如何，都不会发生字节交换。

如果 pktConf.bUseCrc 为 1，则在末尾接收并校验 CRC。CRC 位数、多项式和初始化值遵循无线电中的配置。如果 pktConf.bCrcIncSw 为 1，则收到的同步字（假设它与 syncWord0 或 syncWord1 完全相同）包含在用于计算 CRC 的数据集中。如果 pktConf.bCrcIncHdr 为 1，则收到的标头包含在用于计算 CRC 的数据集中。有效载荷（包括标头之后的可选地址）始终用于计算 CRC。如果 pktConf.bUseCrc 为 0，则处理方式与 CRC OK 相同。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

如果启用了数据白化，只有当 pktConf.bCrcIncHdr 为 1 时，可选的标头才会进行解白化处理。启用后，有效载荷（包括标头之后的可选地址）和收到的 CRC 始终会进行解白化处理。解白化处理在 CRC 评估之前完成。

如果将状态字节附加到数据包（rxConf.bAppendStatus 为 1），其格式如下（请参阅“接收缓冲区”

表 25-180）。如果 addrConf.addrSize 非零，addressInd 字段是地址列表中第一个与接收地址匹配的索引（如果存在地址匹配情况）。否则，addressInd 为 0。如果收到的同步字为 syncWord0，则 syncWordId 字段为 0；如果同步字为 syncWord1，则为 1。结果字段根据表 25-185 写入。

25.10.5.5 载波侦听操作

载波侦听操作用于检测是否存在信号，其主要用途如下：

- 在没有信号时关闭无线电，而不是进行接收
- 仅当没有信号时才将无线电切换到发送状态

载波侦听操作可以与命令 CMD_PROP_CS 配合使用，以与另一操作（例如发射操作）链接，或者与命令 CMD_PROP_RX_SNIFF 或 CMD_PROP_RX_ADV_SNIFF 配合使用，以与常规接收操作结合实现嗅探模式。有关这些命令的详细信息，请参阅节 25.10.5.5.1、节 25.10.5.5.2 和节 25.10.5.5.3。

25.10.5.5.1 常见载波检测说明

节 25.10.2.3 给出了所有命令通用的载波检测操作的参数。节 25.10.2.3 中也给出了与用于载波检测参数的第一个字节的偏移量。

通道可以处于以下三种状态之一：BUSY、IDLE 或 INVALID。BUSY 表示通道上有信号。IDLE 表示通道上不存在信号。INVALID 表示无法确定状态。通道信息有两个来源：RSSI 和相关性，并为每个来源维持单独的状态。

当无线电设置为接收模式时，操作开始。根据使能位 csConf.bEnaRssi 和 csConf.bEnaCorr 监测 RSSI 或相关性。如果 csConf.bEnaRssi 为 1，则会监控 RSSI。如果 csConf.bEnaCorr 为 1，则将相关性设置为与前导码关联。无法将两个使能位都设置为 0。

如果 csConf.bEnaRssi 为 1，则每次无线电提供新值时都会监控 RSSI。每次更新时，将 RSSI 与有符号值 rssiThr 进行比较。如果 RSSI 低于 rssiThr 且已观察到 numRssiIdle 连续 RSSI 测量值低于阈值，则 RSSI 状态为 IDLE 状态。如果 RSSI 高于 rssiThr 且已观察到 numRssiBusy 连续 RSSI 测量值高于阈值，则 RSSI 状态为 BUSY 状态。否则，RSSI 状态为 INVALID。

如果 csConf.bEnaCorr 为 1，无线电 CPU 将监视调制解调器的相关性峰值。无线电启动时，状态为 INVALID。如果在载波侦听命令启动后，在 corrPeriod RAT 发出节拍之前未观察到相关性顶部，则状态变为 IDLE 状态。如果状态为 IDLE，并且至少观察到 corrConfig.numCorrInv 个相关性顶部，并且它们之间至多观察到 corrPeriod RAT 节拍，则该状态变为无效。如果状态为 INVALID，并且至少观察到 corrConfig.numCorrBusy 个相关性顶部，并且它们之间至多观察到 corrPeriod RAT 节拍，则该状态变为 BUSY。如果 corrConfig.numCorrBusy 为 0，则该状态从 IDLE 状态直接进入 BUSY 状态。corrConfig.numCorrIdle 的值必须大于 0。如果状态不是空闲且 corrTime RAT 节拍在上次相关性顶部结束后经过，则状态再次变为 IDLE。

如果只有一个使能位为 1，则通道状态等于相应源的状态。如果两个使能位都为 1，则通道状态取决于两个源的状态和 csConf.operation 位，如表 25-187 所示。

表 25-187. 启用两个源时的通道状态

csConf.operation = 0			
RSSI 状态	关联状态		
	INVALID	IDLE	BUSY
INVALID	INVALID	INVALID	BUSY
IDLE	INVALID	IDLE	BUSY
BUSY	BUSY	BUSY	BUSY
csConf.operation = 1			

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-187. 启用两个源时的通道状态（续）

csConf.operation = 0			
RSSI 状态	关联状态		
	INVALID	IDLE	BUSY
INVALID	INVALID	IDLE	INVALID
IDLE	IDLE	IDLE	IDLE
BUSY	INVALID	IDLE	BUSY

如果通道状态更改为 BUSY，则操作取决于 csConf.busyOp 和正在运行的命令。如果 csConf.busyOp 为 0，操作将继续。如果 csConf.busyOp 为 1 且命令为 CMD_PROP_CS，则操作以 PROP_DONE_BUSY 状态结束。如果 csConf.busyOp 为 1 并且命令为 CMD_PROP_RX_SNIFF 或 CMD_PROP_RX_ADV_SNIFF，则接收操作继续，但载波检测停止，因此如果通道状态稍后更改为 IDLE，则操作不受影响。

如果通道的状态更改为 IDLE，该操作取决于 csConf.idleOp。如果该字段的值为 0，则接收器和载波检测操作继续。如果该位为 1，则操作以 PROP_DONE_IDLE 状态结束。

如果观察到 csEndTrigger 和 csEndTime 给出的触发条件，则操作取决于正在运行的命令和当时的通道状态。节 25.10.5.5.2 和节 25.10.5.5.3 说明了相关详情。

25.10.5.5.2 载波检测命令，CMD_PROP_CS

当载波侦听命令启动时，将无线电设置为接收模式，并执行节 25.10.5.5.1 中所述的操作。无线电必须设置为兼容模式（例如专有模式）并且使用 CMD_FS 对合成器进行编程。

如果观察到 csEndTrigger 和 csEndTime 给出的触发条件，操作结束，并检查当前通道状态。如果状态为 BUSY 或 IDLE，则状态分别为 PROP_DONE_BUSY 或 PROP_DONE_IDLE。如果状态为 INVALID，则状态取决于 csConf.timeoutRes。如果为 0，则状态为 PROP_DONE_BUSYTIMEOUT；如果为 1，则为 PROP_DONE_IDLETIMEOUT。

当 CMD_PROP_CS 结束且状态为 PROP_DONE_BUSY 或 PROP_DONE_BUSYTIMEOUT 时，如果 csFsConf.bFsOffBusy 为 1，合成器将关闭。如果命令结束且状态为 PROP_DONE_IDLE 或 PROP_DONE_IDLETIMEOUT，则在 csFsConf.bFsOffIdle 为 1 时关闭合成器。如果它以另一个状态结束，则合成器会在这些位中的任何一个为 1 时关闭。

与 CMD_PROP_CS 一起使用的结束状态在表 25-188 中汇总。此状态决定下一个操作，如节 25.10.5.1 中所示。

表 25-188. CMD_PROP_CS 命令结束

条件	状态代码	结果
当 csConf.busyOp = 1 时观察到的繁忙通道状态	PROP_DONE_BUSY	TRUE
csConf.idleOp = 1 时观察到的空闲通道状态	PROP_DONE_IDLE	FALSE
在通道状态繁忙时观察到的超时触发	PROP_DONE_BUSY	TRUE
在通道状态空闲时观察到的超时触发	PROP_DONE_IDLE	FALSE
在通道状态无效且 csConf.timeoutRes = 0 时观察到的超时触发	PROP_DONE_BUSYTIMEOUT	TRUE
在通道状态无效且 csConf.timeoutRes = 1 时观察到超时触发	PROP_DONE_IDLETIMEOUT	FALSE
在命令启动后收到的 CMD_STOP	PROP_DONE_STOPPED	FALSE
在命令启动后接收到的 CMD_ABORT	PROP_DONE_ABORT	ABORT
观察到的非法参数	PROP_ERROR_PAR	ABORT
使用 CMD_PROP_RADIO_SETUP 或 CMD_RADIO_SETUP 在不在支持的模式下设置无线电的情况下发送的命令	PROP_ERROR_NO_SETUP	ABORT
在不对合成器进行编程的情况下发送的命令	PROP_ERROR_NO_FS	ABORT

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.10.5.5.3 嗅探模式接收命令 CMD_PROP_RX_SNIFF 和 CMD_PROP_RX_ADV_SNIFF

命令 CMD_PROP_RX_SNIFF 和 CMD_PROP_RX_ADV_SNIFF 的行为分别类似于命令 CMD_PROP_RX 和 CMD_PROP_RX_ADV，但它们在同步字搜索期间执行载波侦听操作。

启动时，这些命令会执行节 25.10.5.5.1 中所述的载波侦听操作。如前文所述，如果通道状态变为 IDLE，操作可能会结束。

如果观察到由 csEndTrigger 和 csEndTime 给出的触发条件，表明正在检查当前通道状态。如果状态为 BUSY，接收器会继续运行，但若通道状态变为 IDLE 且 csConf.busyOp 为 0，则稍后可能结束。如果通道状态为 IDLE，则操作结束（即使 csConf.idleOp 为 0），状态为 PROP_DONE_IDLE。如果通道状态为 INVALID，操作取决于 csConf.timeoutRes。如果为 0，接收操作继续执行，如果 csConf.busyOp 为 1，则不再检查载波侦听。如果 csConf.timeoutRes 为 1，则操作结束，状态为 PROP_DONE_IDLETIMEOUT。

如果找到了同步字，则接收器按照节 25.10.5.4 中所述运行。如果在收到数据包后或因长度字段无效或地址不匹配而停止接收之后，重新开始同步字搜索，并且在找到同步字时载波侦听操作正在运行，则恢复该载波侦听操作。

表 25-186 和表 25-189 列出了用于 CMD_PROP_RX_SNIFF 和 CMD_PROP_RX_ADV_SNIFF 的结束状态。此状态决定下一个操作，如节 25.10.5.1 中所示。

表 25-189. CMD_PROP_RX_SNIFF 和 CMD_PROP_RX_ADV_SNIFF 的其他结束状态

条件	状态代码	结果
观察到通道状态 IDLE，且 csConf.idleOp = 1	PROP_DONE_IDLE	否
观察到超时触发条件，且通道状态为 IDLE	PROP_DONE_IDLE	否
观察到超时触发条件，通道状态为 INVALID 且 csConf.timeoutRes = 1	PROP_DONE_IDLETIMEOUT	否

25.10.6 即时命令

25.10.6.1 设置数据包长度命令 CMD_PROP_SET_LEN

CMD_PROP_SET_LEN 命令采用表 25-176 中定义的命令结构。

CMD_PROP_SET_LEN 只能在运行配置为无限数据包长度的 CMD_PROP_RX 或 CMD_PROP_RX_ADV 命令时发送。接收到该命令后，无线电 CPU 将标头和 CRC 之间要接收的字节数设置为 RXLen。如果已经接收到至少这个字节数，则接收将中止，如节 25.10.5.4 和节 25.10.5.4.2 所述。

如果要设置的有效载荷长度为 255 字节或更少，则该命令可以作为直接命令发送。在这种情况下，RXLen 参数写入 CMDR 的第 8 – 16 位，并且该参数的 8 个 MSB 为 0。

如果在没有运行 CMD_PROP_RX 或 CMD_PROP_RX_ADV 命令的情况下发出该命令，或者此类命令未配置为无限长度，则无线电 CPU 在 CMDSTA 中返回结果 ContextError。否则，无线电 CPU 返回 DONE。

25.10.6.2 重新启动数据包 RX 命令 CMD_PROP_RESTART_RX

CMD_PROP_RESTART_RX 是一个直接命令，不包含任何参数。

CMD_PROP_RESTART_RX 只能在运行 CMD_PROP_RX 或 CMD_PROP_RX_ADV 命令时发送。如果正在接收数据包，则会按照节 25.10.5.4 所述中止接收，数据包返回同步搜索状态。

如果在没有 RX 命令运行的情况下发出该命令，无线电 CPU 在 CMDSTA 中返回结果 ContextError。否则，无线电 CPU 返回 DONE。

25.11 无线电寄存器

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.11.1 RFC_RAT 寄存器

表 25-190 列出了 RFC_RAT 寄存器的存储器映射寄存器。表 25-190 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

表 25-190. RFC_RAT 寄存器

偏移	首字母缩写词	寄存器名称	部分
4h	RATCNT	射频计时器计数器值	RATCNT 寄存器（偏移 = 4h）[复位 = 00000000h]
80h	RATCH0VAL	计时器通道 0 采集/比较寄存器	RATCH0VAL 寄存器（偏移 = 80h）[复位 = 00000000h]
84h	RATCH1VAL	计时器通道 1 采集/比较寄存器	RATCH1VAL 寄存器（偏移 = 84h）[复位 = 00000000h]
88h	RATCH2VAL	计时器通道 2 采集/比较寄存器	RATCH2VAL 寄存器（偏移 = 88h）[复位 = 00000000h]
8Ch	RATCH3VAL	计时器通道 3 采集/比较寄存器	RATCH3VAL 寄存器（偏移 = 8Ch）[复位 = 00000000h]
90h	RATCH4VAL	计时器通道 4 采集/比较寄存器	RATCH4VAL 寄存器（偏移 = 90h）[复位 = 00000000h]
94h	RATCH5VAL	计时器通道 5 采集/比较寄存器	RATCH5VAL 寄存器（偏移 = 94h）[复位 = 00000000h]
98h	RATCH6VAL	计时器通道 6 采集/比较寄存器	RATCH6VAL 寄存器（偏移 = 98h）[复位 = 00000000h]
9Ch	RATCH7VAL	计时器通道 7 采集/比较寄存器	RATCH7VAL 寄存器（偏移 = 9Ch）[复位 = 00000000h]

复杂的位访问类型经过编码可适应小型表单元。表 25-191 展示了适用于此部分中访问类型的代码。

表 25-191. RFC_RAT 访问类型代码

访问类型	代码	说明
读取类型		
R	R	读取
写入类型		
W	W	写入
复位或默认值		
-n		复位后的值或默认值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.11.1.1 RATCNT 寄存器（偏移 = 4h）[复位 = 00000000h]

图 25-12 展示了 RATCNT，表 25-192 中对此进行了介绍。

返回到汇总表。

射频计时器计数器值

图 25-12. RATCNT 寄存器

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

表 25-192. RATCNT 寄存器字段说明

位	字段	类型	复位	说明
31-0	CNT	R/W	0h	计数器值。此项在使能射频计时器计数器时不可写入。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.11.1.2 RATCH0VAL 寄存器（偏移 = 80h）[复位 = 00000000h]

图 25-13 展示了 RATCH0VAL，表 25-193 对其进行了介绍。

返回到汇总表。

计时器通道 0 采集/比较寄存器

图 25-13. RATCH0VAL 寄存器

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

表 25-193. RATCH0VAL 寄存器字段说明

位	字段	类型	复位	说明
31-0	值	R/W	0h	采集/比较值。仅在通道配置为比较模式时可写入。在比较模式下，向该寄存器写入将自动启动通道。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.11.1.3 RATCH1VAL 寄存器（偏移 = 84h）[复位 = 00000000h]

图 25-14 展示了 RATCH1VAL，表 25-194 对其进行了介绍。

返回到汇总表。

计时器通道 1 采集/比较寄存器

图 25-14. RATCH1VAL 寄存器

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

表 25-194. RATCH1VAL 寄存器字段说明

位	字段	类型	复位	说明
31-0	值	R/W	0h	采集/比较值。仅在通道配置为比较模式时可写入。在比较模式下，向该寄存器写入将自动启动通道。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.11.1.4 RATCH2VAL 寄存器（偏移 = 88h）[复位 = 00000000h]

图 25-15 展示了 RATCH2VAL，表 25-195 对其进行了介绍。

返回到汇总表。

计时器通道 2 采集/比较寄存器

图 25-15. RATCH2VAL 寄存器

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

表 25-195. RATCH2VAL 寄存器字段说明

位	字段	类型	复位	说明
31-0	值	R/W	0h	采集/比较值。仅在通道配置为比较模式时可写入。在比较模式下，向该寄存器写入将自动启动通道。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.11.1.5 RATCH3VAL 寄存器（偏移 = 8Ch）[复位 = 00000000h]

图 25-16 展示了 RATCH3VAL，表 25-196 对其进行了介绍。

返回到汇总表。

计时器通道 3 采集/比较寄存器

图 25-16. RATCH3VAL 寄存器

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

表 25-196. RATCH3VAL 寄存器字段说明

位	字段	类型	复位	说明
31-0	值	R/W	0h	采集/比较值。仅在通道配置为比较模式时可写入。在比较模式下，向该寄存器写入将自动启动通道。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.11.1.6 RATCH4VAL 寄存器（偏移 = 90h）[复位 = 00000000h]

图 25-17 展示了 RATCH4VAL，表 25-197 对其进行了介绍。

返回到汇总表。

计时器通道 4 采集/比较寄存器

图 25-17. RATCH4VAL 寄存器

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

表 25-197. RATCH4VAL 寄存器字段说明

位	字段	类型	复位	说明
31-0	值	R/W	0h	采集/比较值。仅在通道配置为比较模式时可写入。在比较模式下，向该寄存器写入将自动启动通道。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.11.1.7 RATCH5VAL 寄存器（偏移 = 94h）[复位 = 00000000h]

图 25-18 展示了 RATCH5VAL，表 25-198 对其进行了介绍。

返回到汇总表。

计时器通道 5 采集/比较寄存器

图 25-18. RATCH5VAL 寄存器

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

表 25-198. RATCH5VAL 寄存器字段说明

位	字段	类型	复位	说明
31-0	值	R/W	0h	采集/比较值。仅在通道配置为比较模式时可写入。在比较模式下，向该寄存器写入将自动启动通道。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.11.1.8 RATCH6VAL 寄存器（偏移 = 98h）[复位 = 00000000h]

图 25-19 展示了 RATCH6VAL，表 25-199 对其进行了介绍。

返回到汇总表。

计时器通道 6 采集/比较寄存器

图 25-19. RATCH6VAL 寄存器

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

表 25-199. RATCH6VAL 寄存器字段说明

位	字段	类型	复位	说明
31-0	值	R/W	0h	采集/比较值。仅在通道配置为比较模式时可写入。在比较模式下，向该寄存器写入将自动启动通道。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.11.1.9 RATCH7VAL 寄存器（偏移 = 9Ch）[复位 = 00000000h]

图 25-20 展示了 RATCH7VAL，表 25-200 对其进行了介绍。

返回到汇总表。

计时器通道 7 采集/比较寄存器

图 25-20. RATCH7VAL 寄存器

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

表 25-200. RATCH7VAL 寄存器字段说明

位	字段	类型	复位	说明
31-0	值	R/W	0h	采集/比较值。仅在通道配置为比较模式时可写入。在比较模式下，向该寄存器写入将自动启动通道。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.11.2 RFC_DBELL 寄存器

表 25-201 列出了 RFC_DBELL 寄存器的存储器映射寄存器。表 25-201 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

表 25-201. RFC_DBELL 寄存器

偏移	首字母缩写词	寄存器名称	部分
0h	CMDR	门铃命令寄存器	CMDR 寄存器（偏移 = 0h）[复位 = 00000000h]
4h	CMDSTA	门铃命令状态寄存器	CMDSTA 寄存器（偏移 = 4h）[复位 = 00000000h]
8h	RFHWIFG	来自射频硬件模块的中断标志	RFHWIFG 寄存器（偏移 = 8h）[复位 = 00000000h]
Ch	RFHWIEN	射频硬件模块的中断使能	RFHWIEN 寄存器（偏移 = Ch）[复位 = 00000000h]
10h	RFCPEIFG	命令和数据包引擎所生成中断的中断标志	RFCPEIFG 寄存器（偏移 = 10h）[复位 = 00000000h]
14h	RFCPEIEN	命令和数据包引擎所生成中断的中断使能	RFCPEIEN 寄存器（偏移 = 14h）[复位 = FFFFFFFFh]
18h	RFCPEISL	命令和数据包引擎所生成中断的中断的中断向量选择	RFCPEISL 寄存器（偏移 = 18h）[复位 = FFFF0000h]
1Ch	RFACKIFG	门铃命令应答中断标志	RFACKIFG 寄存器（偏移 = 1Ch）[复位 = 00000000h]
20h	SYSGPOCTL	射频内核通用输出控制	SYSGPOCTL 寄存器（偏移 = 20h）[复位 = 00000000h]

复杂的位访问类型经过编码可适应小型表单元。表 25-202 展示了适用于此部分中访问类型的代码。

表 25-202. RFC_DBELL 访问类型代码

访问类型	代码	说明
读取类型		
R	R	读取
写入类型		
W	W	写入
复位或默认值		
-n		复位后的值或默认值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.11.2.1 CMDR 寄存器（偏移 = 0h）[复位 = 00000000h]

图 25-21 展示了 CMDR，表 25-203 中对此进行了介绍。

返回到汇总表。

门铃命令寄存器

图 25-21. CMDR 寄存器

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

表 25-203. CMDR 寄存器字段说明

位	字段	类型	复位	说明
31-0	CMD	R/W	0h	命令寄存器。写入时向命令和数据包引擎 (CPE) 触发中断。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.11.2.2 CMDSTA 寄存器（偏移 = 4h）[复位 = 00000000h]

图 25-22 展示了 CMDSTA，表 25-204 中对此进行了介绍。

返回到汇总表。

门铃命令状态寄存器

图 25-22. CMDSTA 寄存器

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

表 25-204. CMDSTA 寄存器字段说明

位	字段	类型	复位	说明
31-0	STAT	R	0h	上次使用命令的状态

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.11.2.3 RFHWIFG 寄存器（偏移 = 8h）[复位 = 00000000h]

图 25-23 展示了 RFHWIFG，表 25-205 对其进行了介绍。

返回到汇总表。

来自射频硬件模块的中断标志

图 25-23. RFHWIFG 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED				RATCH7	RATCH6	RATCH5	RATCH4
R-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RATCH3	RATCH2	RATCH1	RATCH0	RFESOFT2	RFESOFT1	RFESOFT0	RFEDONE
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED	TRCTK	MDMSOFT	MDMOUT	MDMIN	MDMDONE	FSCA	RESERVED
R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

表 25-205. RFHWIFG 寄存器字段说明

位	字段	类型	复位	说明
31-20	RESERVED	R	0h	保留
19	RATCH7	R/W	0h	射频计时器通道 7 中断标志。写入零可清除标志。写入一无影响。
18	RATCH6	R/W	0h	射频计时器通道 6 中断标志。写入零可清除标志。写入一无影响。
17	RATCH5	R/W	0h	射频计时器通道 5 中断标志。写入零可清除标志。写入一无影响。
16	RATCH4	R/W	0h	射频计时器通道 4 中断标志。写入零可清除标志。写入一无影响。
15	RATCH3	R/W	0h	射频计时器通道 3 中断标志。写入零可清除标志。写入一无影响。
14	RATCH2	R/W	0h	射频计时器通道 2 中断标志。写入零可清除标志。写入一无影响。
13	RATCH1	R/W	0h	射频计时器通道 1 中断标志。写入零可清除标志。写入一无影响。
12	RATCH0	R/W	0h	射频计时器通道 0 中断标志。写入零可清除标志。写入一无影响。
11	RFESOFT2	R/W	0h	射频引擎软件定义的中断 2 标志。写入零可清除标志。写入一无影响。
10	RFESOFT1	R/W	0h	射频引擎软件定义的中断 1 标志。写入零可清除标志。写入一无影响。
9	RFESOFT0	R/W	0h	射频引擎软件定义的中断 0 标志。写入零可清除标志。写入一无影响。
8	RFEDONE	R/W	0h	射频引擎命令完成中断标志。写入零可清除标志。写入一无影响。
7	RESERVED	R	0h	保留
6	TRCTK	R/W	0h	调试跟踪器系统节拍中断标志。写入零可清除标志。写入一无影响。
5	MDMSOFT	R/W	0h	调制解调器软件定义的中断标志。写入零可清除标志。写入一无影响。
4	MDMOUT	R/W	0h	调制解调器 FIFO 输出中断标志。写入零可清除标志。写入一无影响。
3	MDMIN	R/W	0h	调制解调器 FIFO 输入中断标志。写入零可清除标志。写入一无影响。
2	MDMDONE	R/W	0h	调制解调器命令完成中断标志。写入零可清除标志。写入一无影响。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-205. RFHWIFG 寄存器字段说明（续）

位	字段	类型	复位	说明
1	FSCA	R/W	0h	频率合成器校准加速器中断标志。写入零可清除标志。写入一无影响。
0	RESERVED	R	0h	保留

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.11.2.4 RFHWIEN 寄存器（偏移 = Ch）[复位 = 00000000h]

图 25-24 展示了 RFHWIEN，表 25-206 对其进行了介绍。

返回到汇总表。

射频硬件模块的中断使能

图 25-24. RFHWIEN 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED				RATCH7	RATCH6	RATCH5	RATCH4
R-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RATCH3	RATCH2	RATCH1	RATCH0	RFESOFT2	RFESOFT1	RFESOFT0	RFEDONE
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED	TRCTK	MDMSOFT	MDMOUT	MDMIN	MDMDONE	FSCA	RESERVED
R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

表 25-206. RFHWIEN 寄存器字段说明

位	字段	类型	复位	说明
31-20	RESERVED	R	0h	保留
19	RATCH7	R/W	0h	RFHWIFG.RATCH7 的中断使能。
18	RATCH6	R/W	0h	RFHWIFG.RATCH6 的中断使能。
17	RATCH5	R/W	0h	RFHWIFG.RATCH5 的中断使能。
16	RATCH4	R/W	0h	RFHWIFG.RATCH4 的中断使能。
15	RATCH3	R/W	0h	RFHWIFG.RATCH3 的中断使能。
14	RATCH2	R/W	0h	RFHWIFG.RATCH2 的中断使能。
13	RATCH1	R/W	0h	RFHWIFG.RATCH1 的中断使能。
12	RATCH0	R/W	0h	RFHWIFG.RATCH0 的中断使能。
11	RFESOFT2	R/W	0h	RFHWIFG.RFESOFT2 的中断使能。
10	RFESOFT1	R/W	0h	RFHWIFG.RFESOFT1 的中断使能。
9	RFESOFT0	R/W	0h	RFHWIFG.RFESOFT0 的中断使能。
8	RFEDONE	R/W	0h	RFHWIFG.RFEDONE 的中断使能。
7	RESERVED	R	0h	保留
6	TRCTK	R/W	0h	RFHWIFG.TRCTK 的中断使能。
5	MDMSOFT	R/W	0h	RFHWIFG.MDMSOFT 的中断使能。
4	MDMOUT	R/W	0h	RFHWIFG.MDMOUT 的中断使能。
3	MDMIN	R/W	0h	RFHWIFG.MDMIN 的中断使能。
2	MDMDONE	R/W	0h	RFHWIFG.MDMDONE 的中断使能。
1	FSCA	R/W	0h	RFHWIFG.FSCA 的中断使能。
0	RESERVED	R	0h	保留

Copyright © 2026 Texas Instruments Incorporated

25.11.2.5 RFCPEIFG 寄存器（偏移 = 10h）[复位 = 00000000h]

图 25-25 展示了 RFCPEIFG，表 25-207 对其进行了介绍。

返回到汇总表。

命令和数据包引擎所生成中断的中断标志

图 25-25. RFCPEIFG 寄存器

31	30	29	28	27	26	25	24
INTERNAL_ERROR	BOOT_DONE	MODULES_UNLOCKED	SYNTH_NO_LOCK	IRQ27	RX_ABORTED	RX_N_DATA_WRITTEN	RX_DATA_WRITTEN
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
RX_ENTRY_DONE	RX_BUF_FULL	RX_CTRL_ACK	RX_CTRL	RX_EMPTY	RX_IGNORED	RX_NOK	RX_OK
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
IRQ15	IRQ14	FG_COMMAND_STARTED	COMMAND_STARTED	TX_BUFFER_CHANGED	TX_ENTRY_DONE	TX_RETRANS	TX_CTRL_ACK_ACK
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
TX_CTRL_ACK	TX_CTRL	TX_ACK	TX_DONE	LAST_FG_COMMAND_DONE	FG_COMMAND_DONE	LAST_COMMAND_DONE	COMMAND_DONE
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 25-207. RFCPEIFG 寄存器字段说明

位	字段	类型	复位	说明
31	INTERNAL_ERROR	R/W	0h	中断标志 31。命令和数据包引擎 (CPE) 观测到意外错误。需要复位 CPE。这可通过在 PRCM:PDCTL1RFC 中关闭然后再开启射频内核电源域来实现。写入零可清除标志。写入一无影响。
30	BOOT_DONE	R/W	0h	中断标志 30。命令和数据包引擎 (CPE) 引导完成。写入零可清除标志。写入一无影响。
29	MODULES_UNLOCKED	R/W	0h	中断标志 29。在命令和数据包引擎 (CPE) 引导过程中，其开放了对射频内核模块和存储器的访问权限。写入零可清除标志。写入一无影响。
28	SYNTH_NO_LOCK	R/W	0h	中断标志 28。频率合成器中的锁相环已报告失锁。写入零可清除标志。写入一无影响。
27	IRQ27	R/W	0h	中断标志 27。写入零可清除标志。写入一无影响。
26	RX_ABORTED	R/W	0h	中断标志 26。数据包接收在数据包完成之前停止。写入零可清除标志。写入一无影响。
25	RX_N_DATA_WRITTEN	R/W	0h	中断标志 25。已将指定字节数写入部分读取 Rx 缓冲区。写入零可清除标志。写入一无影响。
24	RX_DATA_WRITTEN	R/W	0h	中断标志 24。数据已写入部分读取 Rx 缓冲区。写入零可清除标志。写入一无影响。
23	RX_ENTRY_DONE	R/W	0h	中断标志 23。Rx 队列数据条目状态变更为“已完成”。写入零可清除标志。写入一无影响。
22	RX_BUF_FULL	R/W	0h	中断标志 22。接收的数据包无法放入 Rx 队列。BLE 模式：接收的数据包无法放入 Rx 队列。IEEE 802.15.4 模式：接收的帧无法放入 Rx 队列。写入零可清除标志。写入一无影响。
21	RX_CTRL_ACK	R/W	0h	中断标志 21。仅 BLE 模式：已接收 LL 控制数据包，CRC 校验通过，不可忽略，随后发送了应答。写入零可清除标志。写入一无影响。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-207. RFCPEIFG 寄存器字段说明（续）

位	字段	类型	复位	说明
20	RX_CTRL	R/W	0h	中断标志 20。仅 BLE 模式：已接收 LL 控制数据包，CRC 校验通过，不可忽略。写入零可清除标志。写入一无影响。
19	RX_EMPTY	R/W	0h	中断标志 19。仅 BLE 模式：已接收数据包，CRC 校验通过，不可忽略，无有效载荷。写入零可清除标志。写入一无影响。
18	RX_IGNORED	R/W	0h	中断标志 18。已接收数据包，但可忽略。BLE 模式：已接收数据包，CRC 校验通过，但不可忽略。IEEE 802.15.4 模式：已接收帧，忽略标志置位。写入零可清除标志。写入一无影响。
17	RX_NOK	R/W	0h	中断标志 17。已接收数据包，存在 CRC 错误。BLE 模式：已接收数据包，存在 CRC 错误。IEEE 802.15.4 模式：已接收帧，存在 CRC 错误。写入零可清除标志。写入一无影响。
16	RX_OK	R/W	0h	中断标志 16。已正确接收数据包。BLE 模式：已接收数据包，CRC 校验通过，有效载荷且不可忽略。IEEE 802.15.4 模式：已接收帧，CRC 校验通过。写入零可清除标志。写入一无影响。
15	IRQ15	R/W	0h	中断标志 15。写入零可清除标志。写入一无影响。
14	IRQ14	R/W	0h	中断标志 14。写入零可清除标志。写入一无影响。
13	FG_COMMAND_STARTED	R/W	0h	中断标志 13。仅 IEEE 802.15.4 模式：前台射频操作命令已进入激活状态。
12	COMMAND_STARTED	R/W	0h	中断标志 12。射频操作命令已进入激活状态。
11	TX_BUFFER_CHANGED	R/W	0h	中断标志 11。仅 BLE 模式：在 CMD_BLE_ADV_PAYLOAD 之后，缓冲区更改完成。写入零可清除标志。写入一无影响。
10	TX_ENTRY_DONE	R/W	0h	中断标志 10。Tx 队列数据条目状态更改为“已完成”。写入零可清除标志。写入一无影响。
9	TX_RETRANS	R/W	0h	中断标志 9。仅 BLE 模式：数据包已重新发送。写入零可清除标志。写入一无影响。
8	TX_CTRL_ACK_ACK	R/W	0h	中断标志 8。仅 BLE 模式：在发送的 LL 控制数据包上接收应答，并发送该数据包的应答。写入零可清除标志。写入一无影响。
7	TX_CTRL_ACK	R/W	0h	中断标志 7。BLE 模式：在发送的 LL 控制数据包上接收应答。写入零可清除标志。写入一无影响。
6	TX_CTRL	R/W	0h	中断标志 6。BLE 模式：发送的 LL 控制数据包。写入零可清除标志。写入一无影响。
5	TX_ACK	R/W	0h	中断标志 5。BLE 模式：在发送的数据包上接收应答。IEEE 802.15.4 模式：发送的自动 ACK 帧。写入零可清除标志。写入一无影响。
4	TX_DONE	R/W	0h	中断标志 4。数据包已发送。（BLE 模式：数据包已发送。）（IEEE 802.15.4 模式：帧已发送）。写入零可清除标志。写入一无影响。
3	LAST_FG_COMMAND_DONE	R/W	0h	中断标志 3。仅 IEEE 802.15.4 模式：命令链中的最后一条前台射频操作命令已完成。写入零可清除标志。写入一无影响。
2	FG_COMMAND_DONE	R/W	0h	中断标志 2。仅 IEEE 802.15.4 模式：前台射频操作命令已完成。写入零可清除标志。写入一无影响。
1	LAST_COMMAND_DONE	R/W	0h	中断标志 1。命令链中的最后一条射频操作命令已完成。（IEEE 802.15.4 模式：命令链中的最后一条后台级射频操作命令已完成。）写入零可清除标志。写入一无影响。
0	COMMAND_DONE	R/W	0h	中断标志 0。射频操作已完成。（IEEE 802.15.4 模式：后台级射频操作命令已完成。）写入零可清除标志。写入一无影响。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.11.2.6 RFCPEIEN 寄存器（偏移 = 14h）[复位 = FFFFFFFFh]

图 25-26 展示了 RFCPEIEN，表 25-208 对其进行了介绍。

返回到汇总表。

命令和数据包引擎所生成中断的中断使能

图 25-26. RFCPEIEN 寄存器

31	30	29	28	27	26	25	24
INTERNAL_ERROR	BOOT_DONE	MODULES_UNLOCKED	SYNTH_NO_LOCK	IRQ27	RX_ABORTED	RX_N_DATA_WRITTEN	RX_DATA_WRITTEN
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
23	22	21	20	19	18	17	16
RX_ENTRY_DONE	RX_BUF_FULL	RX_CTRL_ACK	RX_CTRL	RX_EMPTY	RX_IGNORED	RX_NOK	RX_OK
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
15	14	13	12	11	10	9	8
IRQ15	IRQ14	FG_COMMAND_STARTED	COMMAND_STARTED	TX_BUFFER_CHANGED	TX_ENTRY_DONE	TX_RETRANS	TX_CTRL_ACK_ACK
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
7	6	5	4	3	2	1	0
TX_CTRL_ACK	TX_CTRL	TX_ACK	TX_DONE	LAST_FG_COMMAND_DONE	FG_COMMAND_DONE	LAST_COMMAND_DONE	COMMAND_DONE
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h

表 25-208. RFCPEIEN 寄存器字段说明

位	字段	类型	复位	说明
31	INTERNAL_ERROR	R/W	1h	RFCPEIFG.INTERNAL_ERROR 的中断使能。
30	BOOT_DONE	R/W	1h	RFCPEIFG.BOOT_DONE 的中断使能。
29	MODULES_UNLOCKED	R/W	1h	RFCPEIFG.MODULES_UNLOCKED 的中断使能。
28	SYNTH_NO_LOCK	R/W	1h	RFCPEIFG.SYNTH_NO_LOCK 的中断使能。
27	IRQ27	R/W	1h	RFCPEIFG.IRQ27 的中断使能。
26	RX_ABORTED	R/W	1h	RFCPEIFG.RX_ABORTED 的中断使能。
25	RX_N_DATA_WRITTEN	R/W	1h	RFCPEIFG.RX_N_DATA_WRITTEN 的中断使能。
24	RX_DATA_WRITTEN	R/W	1h	RFCPEIFG.RX_DATA_WRITTEN 的中断使能。
23	RX_ENTRY_DONE	R/W	1h	RFCPEIFG.RX_ENTRY_DONE 的中断使能。
22	RX_BUF_FULL	R/W	1h	RFCPEIFG.RX_BUF_FULL 的中断使能。
21	RX_CTRL_ACK	R/W	1h	RFCPEIFG.RX_CTRL_ACK 的中断使能。
20	RX_CTRL	R/W	1h	RFCPEIFG.RX_CTRL 的中断使能。
19	RX_EMPTY	R/W	1h	RFCPEIFG.RX_EMPTY 的中断使能。
18	RX_IGNORED	R/W	1h	RFCPEIFG.RX_IGNORED 的中断使能。
17	RX_NOK	R/W	1h	RFCPEIFG.RX_NOK 的中断使能。
16	RX_OK	R/W	1h	RFCPEIFG.RX_OK 的中断使能。
15	IRQ15	R/W	1h	RFCPEIFG.IRQ15 的中断使能。
14	IRQ14	R/W	1h	RFCPEIFG.IRQ14 的中断使能。
13	FG_COMMAND_STARTED	R/W	1h	RFCPEIFG.FG_COMMAND_STARTED 的中断使能。
12	COMMAND_STARTED	R/W	1h	RFCPEIFG.COMMAND_STARTED 的中断使能。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-208. RFCPEIEN 寄存器字段说明（续）

位	字段	类型	复位	说明
11	TX_BUFFER_CHANGED	R/W	1h	RFCPEIFG.TX_BUFFER_CHANGED 的中断使能。
10	TX_ENTRY_DONE	R/W	1h	RFCPEIFG.TX_ENTRY_DONE 的中断使能。
9	TX_RETRANS	R/W	1h	RFCPEIFG.TX_RETRANS 的中断使能。
8	TX_CTRL_ACK_ACK	R/W	1h	RFCPEIFG.TX_CTRL_ACK_ACK 的中断使能。
7	TX_CTRL_ACK	R/W	1h	RFCPEIFG.TX_CTRL_ACK 的中断使能。
6	TX_CTRL	R/W	1h	RFCPEIFG.TX_CTRL 的中断使能。
5	TX_ACK	R/W	1h	RFCPEIFG.TX_ACK 的中断使能。
4	TX_DONE	R/W	1h	RFCPEIFG.TX_DONE 的中断使能。
3	LAST_FG_COMMAND_DONE	R/W	1h	RFCPEIFG.LAST_FG_COMMAND_DONE 的中断使能。
2	FG_COMMAND_DONE	R/W	1h	RFCPEIFG.FG_COMMAND_DONE 的中断使能。
1	LAST_COMMAND_DONE	R/W	1h	RFCPEIFG.LAST_COMMAND_DONE 的中断使能。
0	COMMAND_DONE	R/W	1h	RFCPEIFG.COMMAND_DONE 的中断使能。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.11.2.7 RFCPEISL 寄存器（偏移 = 18h）[复位 = FFFF0000h]

图 25-27 展示了 RFCPEISL，表 25-209 对其进行了介绍。

返回到汇总表。

命令和数据包引擎所生成中断的中断的中断向量选择

图 25-27. RFCPEISL 寄存器

31	30	29	28	27	26	25	24
INTERNAL_ERROR	BOOT_DONE	MODULES_UNLOCKED	SYNTH_NO_LOCK	IRQ27	RX_ABORTED	RX_N_DATA_WRITTEN	RX_DATA_WRITTEN
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
23	22	21	20	19	18	17	16
RX_ENTRY_DONE	RX_BUF_FULL	RX_CTRL_ACK	RX_CTRL	RX_EMPTY	RX_IGNORED	RX_NOK	RX_OK
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
15	14	13	12	11	10	9	8
IRQ15	IRQ14	FG_COMMAND_STARTED	COMMAND_STARTED	TX_BUFFER_CHANGED	TX_ENTRY_DONE	TX_RETRANS	TX_CTRL_ACK_ACK
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
TX_CTRL_ACK	TX_CTRL	TX_ACK	TX_DONE	LAST_FG_COMMAND_DONE	FG_COMMAND_DONE	LAST_COMMAND_DONE	COMMAND_DONE
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 25-209. RFCPEISL 寄存器字段说明

位	字段	类型	复位	说明
31	INTERNAL_ERROR	R/W	1h	选择 RFCPEIFG.INTERNAL_ERROR 中断应使用的 CPU 中断向量。 0h = 将此中断线路与 INT_RF_CPE0 中断向量相关联 1h = 将此中断线路与 INT_RF_CPE1 中断向量相关联
30	BOOT_DONE	R/W	1h	选择 RFCPEIFG.BOOT_DONE 中断应使用的 CPU 中断向量。 0h = 将此中断线路与 INT_RF_CPE0 中断向量相关联 1h = 将此中断线路与 INT_RF_CPE1 中断向量相关联
29	MODULES_UNLOCKED	R/W	1h	选择 RFCPEIFG.MODULES_UNLOCKED 中断应使用的 CPU 中断向量。 0h = 将此中断线路与 INT_RF_CPE0 中断向量相关联 1h = 将此中断线路与 INT_RF_CPE1 中断向量相关联
28	SYNTH_NO_LOCK	R/W	1h	选择 RFCPEIFG.SYNTH_NO_LOCK 中断应使用的 CPU 中断向量。 0h = 将此中断线路与 INT_RF_CPE0 中断向量相关联 1h = 将此中断线路与 INT_RF_CPE1 中断向量相关联
27	IRQ27	R/W	1h	选择 RFCPEIFG.IRQ27 中断应使用的 CPU 中断向量。 0h = 将此中断线路与 INT_RF_CPE0 中断向量相关联 1h = 将此中断线路与 INT_RF_CPE1 中断向量相关联
26	RX_ABORTED	R/W	1h	选择 RFCPEIFG.RX_ABORTED 中断应使用的 CPU 中断向量。 0h = 将此中断线路与 INT_RF_CPE0 中断向量相关联 1h = 将此中断线路与 INT_RF_CPE1 中断向量相关联
25	RX_N_DATA_WRITTEN	R/W	1h	选择 RFCPEIFG.RX_N_DATA_WRITTEN 中断应使用的 CPU 中断向量。 0h = 将此中断线路与 INT_RF_CPE0 中断向量相关联 1h = 将此中断线路与 INT_RF_CPE1 中断向量相关联

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-209. RFCPEISL 寄存器字段说明（续）

位	字段	类型	复位	说明
24	RX_DATA_WRITTEN	R/W	1h	选择 RFCPEIFG.RX_DATA_WRITTEN 中断应使用的 CPU 中断向量。 0h = 将此中断线路与 INT_RF_CPE0 中断向量相关联 1h = 将此中断线路与 INT_RF_CPE1 中断向量相关联
23	RX_ENTRY_DONE	R/W	1h	选择 RFCPEIFG.RX_ENTRY_DONE 中断应使用的 CPU 中断向量。 0h = 将此中断线路与 INT_RF_CPE0 中断向量相关联 1h = 将此中断线路与 INT_RF_CPE1 中断向量相关联
22	RX_BUF_FULL	R/W	1h	选择 RFCPEIFG.RX_BUF_FULL 中断应使用的 CPU 中断向量。 0h = 将此中断线路与 INT_RF_CPE0 中断向量相关联 1h = 将此中断线路与 INT_RF_CPE1 中断向量相关联
21	RX_CTRL_ACK	R/W	1h	选择 RFCPEIFG.RX_CTRL_ACK 中断应使用的 CPU 中断向量。 0h = 将此中断线路与 INT_RF_CPE0 中断向量相关联 1h = 将此中断线路与 INT_RF_CPE1 中断向量相关联
20	RX_CTRL	R/W	1h	选择 RFCPEIFG.RX_CTRL 中断应使用的 CPU 中断向量。 0h = 将此中断线路与 INT_RF_CPE0 中断向量相关联 1h = 将此中断线路与 INT_RF_CPE1 中断向量相关联
19	RX_EMPTY	R/W	1h	选择 RFCPEIFG.RX_EMPTY 中断应使用的 CPU 中断向量。 0h = 将此中断线路与 INT_RF_CPE0 中断向量相关联 1h = 将此中断线路与 INT_RF_CPE1 中断向量相关联
18	RX_IGNORED	R/W	1h	选择 RFCPEIFG.RX_IGNORED 中断应使用的 CPU 中断向量。 0h = 将此中断线路与 INT_RF_CPE0 中断向量相关联 1h = 将此中断线路与 INT_RF_CPE1 中断向量相关联
17	RX_NOK	R/W	1h	选择 RFCPEIFG.RX_NOK 中断应该使用的 CPU 中断向量。 0h = 将此中断线路与 INT_RF_CPE0 中断向量相关联 1h = 将此中断线路与 INT_RF_CPE1 中断向量相关联
16	RX_OK	R/W	1h	选择 RFCPEIFG.RX_OK 中断应使用的 CPU 中断向量。 0h = 将此中断线路与 INT_RF_CPE0 中断向量相关联 1h = 将此中断线路与 INT_RF_CPE1 中断向量相关联
15	IRQ15	R/W	0h	选择 RFCPEIFG.IRQ15 中断应使用的 CPU 中断向量。 0h = 将此中断线路与 INT_RF_CPE0 中断向量相关联 1h = 将此中断线路与 INT_RF_CPE1 中断向量相关联
14	IRQ14	R/W	0h	选择 RFCPEIFG.IRQ14 中断应使用的 CPU 中断向量。 0h = 将此中断线路与 INT_RF_CPE0 中断向量相关联 1h = 将此中断线路与 INT_RF_CPE1 中断向量相关联
13	FG_COMMAND_STARTED	R/W	0h	选择 RFCPEIFG.FG_COMMAND_STARTED 中断应使用的 CPU 中断向量。 0h = 将此中断线路与 INT_RF_CPE0 中断向量相关联 1h = 将此中断线路与 INT_RF_CPE1 中断向量相关联
12	COMMAND_STARTED	R/W	0h	选择 RFCPEIFG.COMMAND_STARTED 中断应使用的 CPU 中断向量。 0h = 将此中断线路与 INT_RF_CPE0 中断向量相关联 1h = 将此中断线路与 INT_RF_CPE1 中断向量相关联
11	TX_BUFFER_CHANGED	R/W	0h	选择 RFCPEIFG.TX_BUFFER_CHANGED 中断应使用的 CPU 中断向量。 0h = 将此中断线路与 INT_RF_CPE0 中断向量相关联 1h = 将此中断线路与 INT_RF_CPE1 中断向量相关联
10	TX_ENTRY_DONE	R/W	0h	选择 RFCPEIFG.TX_ENTRY_DONE 中断应使用的 CPU 中断向量。 0h = 将此中断线路与 INT_RF_CPE0 中断向量相关联 1h = 将此中断线路与 INT_RF_CPE1 中断向量相关联
9	TX_RETRANS	R/W	0h	选择 RFCPEIFG.TX_RETRANS 中断应该使用的 CPU 中断向量。 0h = 将此中断线路与 INT_RF_CPE0 中断向量相关联 1h = 将此中断线路与 INT_RF_CPE1 中断向量相关联

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-209. RFCPEISL 寄存器字段说明（续）

位	字段	类型	复位	说明
8	TX_CTRL_ACK_ACK	R/W	0h	选择 RFCPEIFG.TX_CTRL_ACK_ACK 中断应使用的 CPU 中断向量。 0h = 将此中断线路与 INT_RF_CPE0 中断向量相关联 1h = 将此中断线路与 INT_RF_CPE1 中断向量相关联
7	TX_CTRL_ACK	R/W	0h	选择 RFCPEIFG.TX_CTRL_ACK 中断应使用的 CPU 中断向量。 0h = 将此中断线路与 INT_RF_CPE0 中断向量相关联 1h = 将此中断线路与 INT_RF_CPE1 中断向量相关联
6	TX_CTRL	R/W	0h	选择 RFCPEIFG.TX_CTRL 中断应使用的 CPU 中断向量。 0h = 将此中断线路与 INT_RF_CPE0 中断向量相关联 1h = 将此中断线路与 INT_RF_CPE1 中断向量相关联
5	TX_ACK	R/W	0h	选择 RFCPEIFG.TX_ACK 中断应使用的 CPU 中断向量。 0h = 将此中断线路与 INT_RF_CPE0 中断向量相关联 1h = 将此中断线路与 INT_RF_CPE1 中断向量相关联
4	TX_DONE	R/W	0h	选择 RFCPEIFG.TX_DONE 中断应使用的 CPU 中断向量。 0h = 将此中断线路与 INT_RF_CPE0 中断向量相关联 1h = 将此中断线路与 INT_RF_CPE1 中断向量相关联
3	LAST_FG_COMMAND_DONE	R/W	0h	选择 RFCPEIFG.LAST_FG_COMMAND_DONE 中断应使用的 CPU 中断向量。 0h = 将此中断线路与 INT_RF_CPE0 中断向量相关联 1h = 将此中断线路与 INT_RF_CPE1 中断向量相关联
2	FG_COMMAND_DONE	R/W	0h	选择 RFCPEIFG.FG_COMMAND_DONE 中断应使用的 CPU 中断向量。 0h = 将此中断线路与 INT_RF_CPE0 中断向量相关联 1h = 将此中断线路与 INT_RF_CPE1 中断向量相关联
1	LAST_COMMAND_DONE	R/W	0h	选择 RFCPEIFG.LAST_COMMAND_DONE 中断应使用的 CPU 中断向量。 0h = 将此中断线路与 INT_RF_CPE0 中断向量相关联 1h = 将此中断线路与 INT_RF_CPE1 中断向量相关联
0	COMMAND_DONE	R/W	0h	选择 RFCPEIFG.COMMAND_DONE 中断应使用的 CPU 中断向量。 0h = 将此中断线路与 INT_RF_CPE0 中断向量相关联 1h = 将此中断线路与 INT_RF_CPE1 中断向量相关联

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.11.2.8 RFACKIFG 寄存器（偏移 = 1Ch）[复位 = 00000000h]

图 25-28 展示了 RFACKIFG，表 25-210 对其进行了介绍。

返回到汇总表。

门铃命令应答中断标志

图 25-28. RFACKIFG 寄存器

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

表 25-210. RFACKIFG 寄存器字段说明

位	字段	类型	复位	说明
31-1	RESERVED	R	0h	保留
0	ACKFLAG	R/W	0h	命令 ACK 的中断标志

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.11.2.9 SYSGPOCTL 寄存器（偏移 = 20h）[复位 = 00000000h]

图 25-29 展示了 SYSGPOCTL，表 25-211 对其进行了介绍。

返回到汇总表。

射频内核通用输出控制

图 25-29. SYSGPOCTL 寄存器

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
GPOCTL3				GPOCTL2				GPOCTL1				GPOCTL0			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

表 25-211. SYSGPOCTL 寄存器字段说明

位	字段	类型	复位	说明
31-16	RESERVED	R	0h	保留
15-12	GPOCTL3	R/W	0h	射频内核 GPO 控制位 3。选择要在射频内核 GPO 线路 3 上输出的信号。 0h = CPE GPO 线路 0 1h = CPE GPO 线路 1 2h = CPE GPO 线路 2 3h = CPE GPO 线路 3 4h = MCE GPO 线路 0 5h = MCE GPO 线路 1 6h = MCE GPO 线路 2 7h = MCE GPO 线路 3 8h = RFE GPO 线路 0 9h = RFE GPO 线路 1 Ah = RFE GPO 线路 2 Bh = RFE GPO 线路 3 Ch = RAT GPO 线路 0 Dh = RAT GPO 线路 1 Eh = RAT GPO 线路 2 Fh = RAT GPO 线路 3
11-8	GPOCTL2	R/W	0h	射频内核 GPO 控制位 2。选择要在射频内核 GPO 线路 2 上输出的信号。 0h = CPE GPO 线路 0 1h = CPE GPO 线路 1 2h = CPE GPO 线路 2 3h = CPE GPO 线路 3 4h = MCE GPO 线路 0 5h = MCE GPO 线路 1 6h = MCE GPO 线路 2 7h = MCE GPO 线路 3 8h = RFE GPO 线路 0 9h = RFE GPO 线路 1 Ah = RFE GPO 线路 2 Bh = RFE GPO 线路 3 Ch = RAT GPO 线路 0 Dh = RAT GPO 线路 1 Eh = RAT GPO 线路 2 Fh = RAT GPO 线路 3

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

表 25-211. SYSGPOCTL 寄存器字段说明（续）

位	字段	类型	复位	说明
7-4	GPOCTL1	R/W	0h	射频内核 GPO 控制位 1。选择要在射频内核 GPO 线路 1 上输出的信号。 0h = CPE GPO 线路 0 1h = CPE GPO 线路 1 2h = CPE GPO 线路 2 3h = CPE GPO 线路 3 4h = MCE GPO 线路 0 5h = MCE GPO 线路 1 6h = MCE GPO 线路 2 7h = MCE GPO 线路 3 8h = RFE GPO 线路 0 9h = RFE GPO 线路 1 Ah = RFE GPO 线路 2 Bh = RFE GPO 线路 3 Ch = RAT GPO 线路 0 Dh = RAT GPO 线路 1 Eh = RAT GPO 线路 2 Fh = RAT GPO 线路 3
3-0	GPOCTL0	R/W	0h	射频内核 GPO 控制位 0。选择要在射频内核 GPO 线路 0 上输出的信号。 0h = CPE GPO 线路 0 1h = CPE GPO 线路 1 2h = CPE GPO 线路 2 3h = CPE GPO 线路 3 4h = MCE GPO 线路 0 5h = MCE GPO 线路 1 6h = MCE GPO 线路 2 7h = MCE GPO 线路 3 8h = RFE GPO 线路 0 9h = RFE GPO 线路 1 Ah = RFE GPO 线路 2 Bh = RFE GPO 线路 3 Ch = RAT GPO 线路 0 Dh = RAT GPO 线路 1 Eh = RAT GPO 线路 2 Fh = RAT GPO 线路 3

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.11.3 RFC_PWR 寄存器

表 25-212 列出了 RFC_PWR 寄存器的存储器映射寄存器。表 25-212 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

表 25-212. RFC_PWR 寄存器

偏移	首字母缩写词	寄存器名称	部分
0h	PWMCLKEN	射频内核电源管理和时钟使能	PWMCLKEN 寄存器 (偏移 =0h) [复位 = 00000001h]

复杂的位访问类型经过编码可适应小型表单元。表 25-213 展示了适用于此部分中访问类型的代码。

表 25-213. RFC_PWR 访问类型代码

访问类型	代码	说明
读取类型		
R	R	读取
写入类型		
W	W	写入
复位或默认值		
-n		复位后的值或默认值

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

25.11.3.1 PWMCLKEN 寄存器（偏移 =0h）[复位 = 00000001h]

图 25-30 展示了 PWMCLKEN，表 25-214 中对此进行了介绍。

返回到汇总表。

射频内核电源管理和时钟使能

图 25-30. PWMCLKEN 寄存器

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED					RFCTRC	FSCA	PHA
R-0h					R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RAT	RFERAM	RFE	MDMRAM	MDM	CPERAM	CPE	RFC
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-1h

表 25-214. PWMCLKEN 寄存器字段说明

位	字段	类型	复位	说明
31-11	RESERVED	R	0h	保留
10	RFCTRC	R/W	0h	使能射频内核跟踪器 (RFCTRC) 模块的时钟。
9	FSCA	R/W	0h	使能频率合成器校准加速器 (FSCA) 模块的时钟。
8	PHA	R/W	0h	使能数据包处理加速器 (PHA) 模块的时钟。
7	RAT	R/W	0h	使能射频计时器 (RAT) 模块的时钟。
6	RFERAM	R/W	0h	使能射频引擎 RAM 模块的时钟。
5	RFE	R/W	0h	使能射频引擎 (RFE) 模块的时钟。
4	MDMRAM	R/W	0h	使能调制解调器 RAM 模块的时钟。
3	MDM	R/W	0h	使能调制解调器 (MDM) 模块的时钟。
2	CPERAM	R/W	0h	使能命令和数据包引擎 (CPE) RAM 模块的时钟。在射频内核初始化过程中，将该位与 CPE 位一起设置以使能 CPE 引导。
1	CPE	R/W	0h	使能命令和数据包引擎 (CPE) 的处理器时钟 (hclk)。在射频内核初始化过程中，将该位与 CPERAM 位一起设置以使 CPE 能够引导。
0	RFC	R	1h	使能射频内核接口的基本时钟。这包括互连、射频门铃 DBELL 命令接口，电源管理 (PWR) 时钟控制模块和 CPE 的总线时钟 (sclk)。为避免锁死射频内核，该位无法被清除。如果需要禁用射频内核的所有时钟，请参阅 PRCM:RFCCLKG.CLK_EN 寄存器。

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

注：以前版本的页码可能与当前版本的页码不同

Changes from Revision F (April 2023) to Revision G (June 2024)	Page
• 添加了 CC1312PSIP 器件产品页面	66

Changes from Revision E (February 2022) to Revision F (April 2023)	Page
• CC1352R 和 CC1352P 器件采用重新格式化的 表 13-2 脚注	1090

English Document: SWCU185
Copyright © 2026 Texas Instruments Incorporated

This page intentionally left blank.

Copyright © 2026 Texas Instruments Incorporated

重要通知和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做出任何明示或暗示的担保，包括但不限于对适销性、与某特定用途的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他安全、安保法规或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的相关应用。严禁以其他方式对这些资源进行复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。对于因您对这些资源的使用而对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，您将全额赔偿，TI 对此概不负责。

TI 提供的产品受 TI 销售条款)、TI 通用质量指南 或 ti.com 上其他适用条款或 TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。 除非德州仪器 (TI) 明确将某产品指定为定制产品或客户特定产品，否则其产品均为按确定价格收入目录的标准通用器件。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

版权所有 © 2026，德州仪器 (TI) 公司

最后更新日期：2025 年 10 月