GigaDevice Semiconductor Inc.

GD32F1x0
ARM® Cortex®-M3 32-bit MCU

适用于 GD32F130xx、GD32F150xx 系列

用户手册

4.0 版本

(2025 年 8 月)

目录

目录 2
图索引 14
表索引 20
1. 系统及存储器架构 22
1.1. ARM Cortex-M3 处理器 22
1.2. 系统架构 23
1.3. 存储器映射 24
1.3.1. 位带操作 26
1.3.2. 片上 SRAM 27
1.3.3. 片上闪存 27
1.4. 引导配置 28
1.5. 系统配置寄存器（SYSCFG） 29
1.5.1. 系统配置寄存器 0（SYSCFG_CFG0） 29
1.5.2. EXTI 源选择寄存器 0（SYSCFG_EXTISS0） 30
1.5.3. EXTI 源选择寄存器 1（SYSCFG_EXTISS1） 31
1.5.4. EXTI 源选择寄存器 2（SYSCFG_EXTISS2） 32
1.5.5. EXTI 源选择寄存器 3（SYSCFG_EXTISS3） 33
1.5.6. 系统配置寄存器 2（SYSCFG_CFG2） 35
1.6. 设备电子签名 36
1.6.1. 存储容量信息 36
1.6.2. 设备唯一 ID（96 位/位域） 36
2. 闪存控制器（FMC） 38
2.1. 简介 38
2.2. 主要特性 38
2.3. 功能描述 38
2.3.1. 闪存结构 38
2.3.2. 读操作 39
2.3.3. FMC_CTL 寄存器解锁 39
2.3.4. 页擦除 39
2.3.5. 整片擦除 40
2.3.6. 主存储闪存块编程 41
2.3.7. 选项字节擦除 42
2.3.8. 选项字节编程 43
2.3.9. 选项字节说明 43
2.3.10. 页擦除/编程保护 44
2.3.11. 安全保护 45

2.4. FMC 寄存器 46
2.4.1. 等待状态寄存器 (FMC_WS) 46
2.4.2. 解锁寄存器 (FMC_KEY) 46
2.4.3. 选项字节解锁寄存器 (FMC_OBKEY) 47
2.4.4. 状态寄存器 (FMC_STAT) 47
2.4.5. 控制寄存器 (FMC_CTL) 48
2.4.6. 地址寄存器 0 (FMC_ADDR) 49
2.4.7. 选项字节状态寄存器 (FMC_OBSTAT) 50
2.4.8. 写保护寄存器 (FMC_WP) 50
2.4.9. 等待状态使能寄存器 (FMC_WSEN) 51
2.4.10. 产品 ID 寄存器 (FMC_PID) 51
3. 电源管理单元（PMU） 53
3.1. 简介 53
3.2. 主要特性 53
3.3. 功能描述 53
3.3.1. 电池备份域 55
3.3.2. V_{DD} / V_{DDA} 电源域 55
3.3.3. 1.2V 电源域 57
3.3.4. 省电模式 57
3.4. PMU 寄存器 60
3.4.1. 控制寄存器（PMU_CTL） 60
3.4.2. 电源控制和状态寄存器(PMU_CS) 61
4. 复位和时钟单元（RCU） 63
4.1. 复位控制单元（RCTL） 63
4.1.1. 简介 63
4.1.2. 功能描述 63
4.2. 时钟控制单元（CCTL） 64
4.2.1. 简介 64
4.2.2. 主要特性 65
4.2.3. 功能描述 66
4.3. RCU 寄存器 70
4.3.1. 控制寄存器 0（RCU_CTL0） 70
4.3.2. 配置寄存器 0（RCU_CFG0） 71
4.3.3. 中断寄存器（RCU_INT） 75
4.3.4. APB2 复位寄存器（RCU_APB2RST） 77
4.3.5. APB1 复位寄存器（RCU_APB1RST） 79
4.3.6. AHB 使能寄存器（RCU_AHBEN） 81
4.3.7. APB2 使能寄存器（RCU_APB2EN） 82
4.3.8. APB1 使能寄存器（RCU_APB1EN） 84
4.3.9. 备份域控制寄存器（RCU_BDCTL） 86
4.3.10. 复位源/时钟寄存器（RCU_RSTSCK） 87

4.3.11. AHB 复位寄存器（RCU_AHBRST）.....89
4.3.12. 配置寄存器 1（RCU_CFG1）.....90
4.3.13. 配置寄存器 2（RCU_CFG2）.....91
4.3.14. 控制寄存器 1（RCU_CTL1）.....91
4.3.15. APB1 附加使能寄存器（RCU_ADDAPB1EN）.....92
4.3.16. APB1 附加复位寄存器（RCU_ADDAPB1RST）.....93
4.3.17. 电源解锁寄存器（RCU_VKEY）.....93
4.3.18. RCU 深度睡眠模式电压寄存器（RCU_DSV).....94
4.3.19. RCU 掉电电压选择寄存器（RCU_PDVSEL）.....94
5. 中断和事件控制器（EXTI）.....96
5.1. 简介.....96
5.2. 主要特性.....96
5.3. 中断功能描述.....96
5.4. 外部中断及事件（EXTI）结构框图.....99
5.5. 外部中断及事件功能概述.....99
5.6. EXTI 寄存器.....102
5.6.1. 中断使能寄存器（EXTI_INTEN）.....102
5.6.2. 事件使能寄存器（EXTI_EVEN）.....102
5.6.3. 上升沿触发使能寄存器（EXTI_RTEN）.....103
5.6.4. 下降沿触发使能寄存器（EXTI_FTEN）.....103
5.6.5. 软件中断事件寄存器（EXTI_SWIEV）.....104
5.6.6. 挂起寄存器（EXTI_PD）.....105
6. 通用输入/输出接口（GPIO）.....106
6.1. 简介.....106
6.2. 主要特性.....106
6.3. 功能描述.....106
6.3.1. GPIO 管脚配置.....107
6.3.2. 备用功能（AF）.....108
6.3.3. 附加功能.....108
6.3.4. 输入配置.....108
6.3.5. 输出配置.....108
6.3.6. 模拟配置.....109
6.3.7. 备用功能（AF）配置.....109
6.3.8. GPIO 锁定功能.....110
6.3.9. GPIO 单周期输出翻转功能.....110
6.4. GPIO 寄存器.....111
6.4.1. 端口控制寄存器（GPIOx_CTL, x=A..D,F）.....111
6.4.2. 端口输出模式寄存器（GPIOx_OMODE, x=A..D,F）.....112
6.4.3. 端口输出速度寄存器（GPIOx_OSPD, x=A..D,F）.....114
6.4.4. 端口上拉/下拉寄存器（GPIOx_PUD, x=A..D,F）.....116

6.4.5. 端口输入状态寄存器（GPIOx_ISTAT, x=A..D,F）.....117
6.4.6. 端口输出控制寄存器（GPIOx_OCTL, x=A..D,F）.....118
6.4.7. 端口位操作寄存器（GPIOx_BOP, x=A..D,F）.....118
6.4.8. 端口配置锁定寄存器（GPIOx_LOCK, x=A,B）.....119
6.4.9. 备用功能选择寄存器 0（GPIOx_AFSEL0, x=A,B,C）.....120
6.4.10. 备用功能选择寄存器 1（GPIOx_AFSEL1, x=A,B,C）.....121
6.4.11. 位清除寄存器（GPIOx_BC, x=A..D,F）.....122
7. 循环冗余校验管理管理单元（CRC）.....123
7.1. 简介.....123
7.2. 主要特性.....123
7.3. 功能描述.....124
7.4. CRC 寄存器.....125
7.4.1. 数据寄存器（CRC_DATA）.....125
7.4.2. 独立数据寄存器（CRC_FDATA）.....125
7.1.1. 控制寄存器（CRC_CTL）.....126
7.1.2. 初值寄存器（CRC_IDATA）.....126
8. DMA 控制器（DMA）.....128
8.1. 简介.....128
8.2. 主要特性.....128
8.3. 结构框图.....129
8.4. 功能描述.....129
8.4.1. DMA 操作.....129
8.4.2. 外设握手.....130
8.4.3. 仲裁.....131
8.4.4. 地址生成.....131
8.4.5. 循环模式.....131
8.4.6. 存储器到存储器模式.....132
8.4.7. 通道配置.....132
8.4.8. 中断.....132
8.4.9. DMA 请求映射.....133
8.5. DMA 寄存器.....136
8.5.1. 中断标志位寄存器（DMA_INTF）.....136
8.5.2. 中断标志位清除寄存器（DMA_INTC）.....136
8.5.3. 通道 x 控制寄存器（DMA_CHxCTL）.....137
8.5.4. 通道 x 计数寄存器（DMA_CHxCNT）.....139
8.5.5. 通道 x 外设基地址寄存器（DMA_CHxPADDR）.....140
8.5.6. 通道 x 存储器基地址寄存器（DMA_CHxMADDR）.....140
9. 调试（DBG）.....142
9.1. 简介.....142

9.2. 串行调试接口简介 142
9.2.1. 引脚分配 142
9.2.2. JEDEC-106 ID code 142
9.3. 调试保持功能描述 142
9.3.1. 低功耗模式调试支持 142
9.3.2. TIMER, I2C, RTC, WWDGT 和 FWDGT 的外设调试支持 143
9.4. DBG 寄存器 144
9.4.1. ID 寄存器 (DBG_ID) 144
9.4.2. 控制寄存器 0 (DBG_CTL0) 144
9.4.3. 控制寄存器 1 (DBG_CTL1) 146
10. 模拟数字转换器（ADC） 148
10.1. 简介 148
10.2. 主要特性 148
10.3. 引脚和内部信号 148
10.4. 功能说明 149
10.4.1. 前置校准功能 149
10.4.2. 双时钟域架构 150
10.4.3. ADCON 使能 150
10.4.4. 常规序列 150
10.4.5. 运行模式 150
10.4.6. 转换结果阈值监测功能 153
10.4.7. 数据存储模式 153
10.4.8. 采样时间配置 154
10.4.9. 外部触发配置 154
10.4.10. DMA 请求 155
10.4.11. ADC 内部通道 155
10.4.12. 电池电压监测 155
10.4.13. ADC 中断 156
10.5. ADC 寄存器 157
10.5.1. 状态寄存器 (ADC_STAT) 157
10.5.2. 控制寄存器 0 (ADC_CTL0) 157
10.5.3. 控制寄存器 1 (ADC_CTL1) 159
10.5.4. 采样时间寄存器 0 (ADC_SAMPT0) 160
10.5.5. 采样时间寄存器 1 (ADC_SAMPT1) 161
10.5.6. 看门狗高阈值寄存器 (ADC_WDHT) 162
10.5.7. 看门狗低阈值寄存器 (ADC_WDLT) 162
10.5.8. 常规序列寄存器 0 (ADC_RSQ0) 163
10.5.9. 常规序列寄存器 1 (ADC_RSQ1) 163
10.5.10. 常规序列寄存器 2 (ADC_RSQ2) 164
10.5.11. 常规数据寄存器 (ADC_RDATA) 165
11. 数模转换器（DAC） 166

11.1. 简介 166
11.2. 主要特性 166
11.3. 功能描述 167
11.3.1. DAC 使能 167
11.3.2. DAC 输出缓冲 167
11.3.3. DAC 数据配置 167
11.3.4. DAC 触发 168
11.3.5. DAC 转换 168
11.3.6. DAC 输出电压 168
11.3.7. DMA 请求 168
11.4. DAC 寄存器 169
11.4.1. DACx 控制寄存器 (DAC_CTL0) 169
11.4.2. DACx 软件触发寄存器 (DAC_SWT) 170
11.4.3. DAC_OUT0 12 位右对齐数据保持寄存器 (DAC_OUT0_R12DH) 170
11.4.4. DAC_OUT0 12 位左对齐数据保持寄存器 (DAC_OUT0_L12DH) 171
11.4.5. DAC_OUT0 8 位右对齐数据保持寄存器 (DAC_OUT0_R8DH) 171
11.4.6. DAC_OUT0 数据输出寄存器 (DAC_OUT0_DO) 171
11.4.7. DAC 状态寄存器 0 (DAC_STAT0) 172
12. 比较器（CMP） 173
12.1. 简介 173
12.2. 主要特征 173
12.3. 功能描述 173
12.3.1. 比较器时钟 174
12.3.2. 比较器 I / O 配置 174
12.3.3. 比较器供电模式 175
12.3.4. 比较器窗口模式 175
12.3.5. 比较器迟滞 175
12.3.6. 比较器寄存器写保护 176
12.3.7. 比较器中断 176
12.4. CMP 寄存器 177
12.4.1. CMP 控制状态寄存器（CMP_CS） 177
13. 看门狗定时器（WDGT） 181
13.1. 独立看门狗定时器（FWDGT） 181
13.1.1. 简介 181
13.1.2. 主要特性 181
13.1.3. 功能描述 181
13.1.4. FWDGT 寄存器 184
13.2. 窗口看门狗定时器（WWDGT） 187
13.2.1. 简介 187
13.2.2. 主要特性 187

13.2.3. 功能描述 187
13.2.4. WWDGT 寄存器 189
14. 实时时钟（RTC） 191
14.1. 简介 191
14.2. 主要特性 191
14.3. 功能描述 192
14.3.1. 结构框图 192
14.3.2. 时钟源和预分频 192
14.3.3. 影子寄存器 193
14.3.4. 位域可屏蔽可配置的闹钟 193
14.3.5. RTC 初始化和配置 193
14.3.6. 读取日历 194
14.3.7. RTC 复位 195
14.3.8. RTC 移位功能 196
14.3.9. RTC 参考时钟检测 196
14.3.10. RTC 数字平滑校准 197
14.3.11. 时间戳功能 198
14.3.12. 侵入检测 199
14.3.13. 校准时钟输出 200
14.3.14. 闹钟输出 200
14.3.15. RTC 省电模式管理 200
14.3.16. RTC 中断 201
14.4. RTC 寄存器 202
14.4.1. 时间寄存器 (RTC_TIME) 202
14.4.2. 日期寄存器 (RTC_DATE) 202
14.4.3. 控制寄存器 (RTC_CTL) 203
14.4.4. 状态寄存器 (RTC_STAT) 205
14.4.5. 预分频寄存器 (RTC_PSC) 207
14.4.6. 闹钟 0 时间日期寄存器 (RTC_ALRM0TD) 207
14.4.7. 写保护钥匙寄存器 (RTC_WPK) 208
14.4.8. 亚秒寄存器(RTC_SS) 209
14.4.9. 移位控制寄存器 (RTC_SHIFTCTL) 209
14.4.10. 时间戳时间寄存器 (RTC_TTS) 210
14.4.11. 时间戳日期寄存器 (RTC_DTS) 211
14.4.12. 时间戳亚秒寄存器 (RTC_SSTS) 211
14.4.13. 高精度频率补偿寄存器 (RTC_HRFC) 212
14.4.14. 侵入寄存器 (RTC_TAMP) 213
14.4.15. 闹钟 0 亚秒寄存器 (RTC_ALRM0SS) 215
14.4.16. 备份寄存器 (RTC_BKPx) (x=0..4) 216
15. 定时器（TIMER） 217
15.1. 高级定时器（TIMERx,x=0） 218
15.1.1. 简介 218

15.1.2. 主要特性 218
15.1.3. 结构框图 218
15.1.4. 功能描述 219
15.1.5. TIMERx 寄存器(x=0) 244
15.2. 通用定时器 L0（TIMERx, x=1, 2） 269
15.2.1. 简介 269
15.2.2. 主要特性 269
15.2.3. 结构框图 269
15.2.4. 功能描述 270
15.2.5. TIMERx 寄存器（x=1,2） 283
15.3. 通用定时器 L2（TIMERx, x=13） 307
15.3.1. 简介 307
15.3.2. 主要特性 307
15.3.3. 结构框图 307
15.3.4. 功能描述 307
15.3.5. TIMERx 寄存器(x=13) 315
15.4. 通用定时器 L3（TIMERx,x=14） 324
15.4.1. 简介 324
15.4.2. 主要特性 324
15.4.3. 结构框图 324
15.4.4. 功能描述 325
15.4.5. TIMERx 寄存器(x=14) 340
15.5. 通用定时器 L4（TIMERx,x=15,16） 358
15.5.1. 简介 358
15.5.2. 主要特性 358
15.5.3. 结构框图 358
15.5.4. 功能描述 359
15.5.5. TIMERx 寄存器(x=15,16) 371
15.6. 基本定时器（TIMERx, x=5） 385
15.6.1. 简介 385
15.6.2. 主要特性 385
15.6.3. 结构框图 385
15.6.4. 功能描述 385
15.6.5. TIMERx 寄存器(x=5) 389
16. 红外线接口（IFRP） 394
16.1. 简介 394
16.2. 主要特性 394
16.3. 功能描述 394
17. 通用同步异步收发器（USART） 396
17.1. 简介 396

17.2. 主要特性 396
17.3. 功能描述 397
17.3.1. USART 帧格式 398
17.3.2. 波特率发生 399
17.3.3. USART 发送器 399
17.3.4. USART 接收器 400
17.3.5. DMA 方式访问数据缓冲区 401
17.3.6. 硬件流控制 403
17.3.7. 多处理器通信 404
17.3.8. LIN 模式 405
17.3.9. 同步通信模式 405
17.3.10. 串行红外（IrDA SIR）编解码功能模块 406
17.3.11. 半双工通信模式 407
17.3.12. 智能卡（ISO7816-3）模式 408
17.3.13. ModBus 通信 409
17.3.14. 从 Deepsleep 模式唤醒 409
17.3.15. USART 中断 410
17.4. USART 寄存器 412
17.4.1. USART 控制寄存器 0（USART_CTL0） 412
17.4.2. USART 控制寄存器 1（USART_CTL1） 414
17.4.3. USART 控制寄存器 2（USART_CTL2） 416
17.4.4. USART 波特率寄存器（USART_ BAUD） 419
17.4.5. USART 保护时间和预分频器寄存器（USART_GP） 420
17.4.6. USART 接收超时寄存器（USART_RT） 420
17.4.7. USART 请求寄存器（USART_CMD） 421
17.4.8. USART 状态寄存器（USART_STAT） 422
17.4.9. USART 中断标志清除寄存器（USART_INTC） 425
17.4.10. USART 数据接收寄存器 (USART_RDATA) 426
17.4.11. USART 数据发送寄存器（USART_TDATA） 427
18. 内部集成电路总线接口（I2C） 428
18.1. 简介 428
18.2. 主要特征 428
18.3. 功能说明 428
18.3.1. SDA 线和 SCL 线 429
18.3.2. 数据有效性 429
18.3.3. 开始和停止信号 430
18.3.4. 时钟同步 430
18.3.5. 仲裁 430
18.3.6. I2C 通讯流程 431
18.3.7. 软件编程模型 432
18.3.8. SCL 线控制 439
18.3.9. DMA 模式下数据传输 440

18.3.10. 报文错误校验 440
18.3.11. SMBus 支持 440
18.3.12. 状态、错误和中断 442
18.4. I2C 寄存器 443
18.4.1. 控制寄存器 0（I2C_CTL0） 443
18.4.2. 控制寄存器 1（I2C_CTL1） 444
18.4.3. 从机地址寄存器 0（I2C_SADDR0） 446
18.4.4. 从机地址寄存器 1（I2C_SADDR1） 446
18.4.5. 传输缓冲区寄存器（I2C_DATA） 447
18.4.6. 传输状态寄存器 0（I2C_STAT0） 447
18.4.7. 传输状态寄存器 1（I2C_STAT1） 449
18.4.8. 时钟配置寄存器（I2C_CKCFG） 450
18.4.9. 上升时间寄存器（I2C_RT） 451
19. 串行外设接口/片上音频接口（SPI/I2S） 452
19.1. 简介 452
19.2. 主要特性 452
19.2.1. SPI 主要特性 452
19.2.2. I2S 主要特性 452
19.3. SPI 功能描述 453
19.3.1. SPI 结构框图 453
19.3.2. SPI 信号线描述 453
19.3.3. SPI 时序和数据帧格式 453
19.3.4. NSS 功能 454
19.3.5. SPI 运行模式 455
19.3.6. DMA 功能 459
19.3.7. CRC 功能 459
19.3.8. SPI 中断 460
19.4. I2S 功能说明 461
19.4.1. I2S 结构框图 461
19.4.2. I2S 信号线描述 461
19.4.3. I2S 音频标准 461
19.4.4. I2S 时钟 469
19.4.5. 运行 470
19.4.6. DMA 功能 474
19.4.7. I2S 中断 474
19.5. SPI/I2S 寄存器 476
19.5.1. 控制寄存器 0（SPI_CTL0） 476
19.5.2. 控制寄存器 1（SPI_CTL1） 478
19.5.3. 状态寄存器（SPI_STAT） 478
19.5.4. 数据寄存器（SPI_DATA） 480
19.5.5. CRC 多项式寄存器（SPI_CRCPOLY） 480
19.5.6. 接收 CRC 寄存器（SPI_RCRC） 481

19.5.7. 发送 CRC 寄存器（SPI_TCRC）.....481
19.5.8. I2S 控制寄存器（SPI_I2SCTL）.....482
19.5.9. I2S 时钟预分频寄存器（SPI_I2SPSC）.....483
20. HDMI-CEC 控制器（HDMI-CEC）.....485
20.1. 简介.....485
20.2. 主要特性.....485
20.3. 功能描述.....485
20.3.1. CEC 总线引脚.....485
20.3.2. 信息说明.....486
20.3.3. 位时序说明.....487
20.3.4. 仲裁.....487
20.3.5. SFTOPT 位说明.....488
20.3.6. 错误定义.....489
20.3.7. HDMI-CEC 中断.....491
20.4. HDMI-CEC 寄存器.....493
20.4.1. 控制寄存器 (CEC_CTL).....493
20.4.2. 配置寄存器 (CEC_CFG).....493
20.4.3. 数据发送寄存器 (CEC_TDATA).....495
20.4.4. 数据接收寄存器 (CEC_RDATA).....496
20.4.5. 中断标志寄存器 (CEC_INTF).....496
20.4.6. 中断使能寄存器 (CEC_INTEN).....498
21. 触摸传感控制器（TSI）.....500
21.1. 简介.....500
21.2. 主要特性.....500
21.3. 功能描述.....500
21.3.1. TSI 框图.....500
21.3.2. 触摸传感技术概述.....501
21.3.3. 电荷转移序列.....501
21.3.4. 电荷转移序列状态机.....503
21.3.5. 状态时钟和持续时间.....504
21.3.6. 引脚模式和 TSI 控制.....505
21.3.7. 模拟开关和 I/O 迟滞模式.....505
21.3.8. TSI 操作流.....505
21.3.9. TSI 标志和中断.....506
21.3.10. TSI GPIOs.....506
21.4. TSI 寄存器.....508
21.4.1. 控制寄存器（TSI_CTL）.....508
21.4.2. 中断使能寄存器（TSI_INTEN）.....510
21.4.3. 中断标志位清除寄存器（TSI_INTC）.....510
21.4.4. 中断标志位寄存器（TSI_INTF）.....511

21.4.5. 引脚迟滞模式寄存器（TSI_PHM） 511
21.4.6. 模拟开关寄存器（TSI_ASW） 512
21.4.7. 采样配置寄存器（TSI_SAMPCFG） 512
21.4.8. 通道配置寄存器（TSI_CHCFG） 513
21.4.9. 组控制寄存器（TSI_GCTL） 513
21.4.10. 组 x 周期数寄存器（TSI_GxCYCN）（x = 0…5） 514
22. 通用串行总线全速设备接口（USBD） 515
22.1. 简介 515
22.2. 主要特性 515
22.3. 模块图 515
22.4. 信号描述 515
22.5. 时钟配置 516
22.6. 功能说明 516
22.6.1. USB 端点 516
22.6.2. USB 传输 518
22.6.3. USB 事件与中断 520
22.6.4. 操作指南 521
22.7. USBD 寄存器 523
22.7.1. USBD 控制寄存器 (USBD_CTL) 523
22.7.2. USBD 中断标志寄存器 (USBD_INTF) 524
22.7.3. USBD 状态寄存器 (USBD_STAT) 525
22.7.4. USBD 设备地址寄存器 (USBD_ADDR) 526
22.7.5. USBD 缓冲器地址寄存器 (USBD_BADDR) 527
22.7.6. USBD 端点 x 控制/状态寄存器 (USB_EPxCS), x=[0..7] 527
22.7.7. USBD 端点 x 发送缓冲地址寄存器 (USBD_EPxTBADDR), x=[0…7] 529
22.7.8. USBD 端点 x 发送缓冲区字节数目寄存器 (USBD_EPxTBCNT) x=[0…7] 529
22.7.9. USBD 端点 x 接收缓冲器地址寄存器 (USBD_EPxRBADDR) x=[0…7] 530
22.7.10. USBD 端点 x 接收缓冲区字节数目寄存器 n (USBD_EPxRBCNT) x=[0…7] 530
23. 附录 532
23.1. 寄存器表中使用的缩写列表 532
23.2. 术语表 532
23.3. 可用外设 532
24. 版本历史 533

图索引

图 1-1. Cortex®-M3 框图 23
图 1-2. GD32F130xx 和 GD32F150xx 产品的系统结构 24
图 2-1. 页擦除操作流程 40
图 2-2. 整片擦除操作流程 41
图 2-3. 字编程操作流程 42
图 3-1.电源域概览 54
图 3-2. 上电复位/掉电复位波形图 56
图 3-3. LVD 阈值波形图 57
图 4-1. 系统复位电路 64
图 4-2.产品的时钟树 65
图 4-3. HXTAL 接线图 66
图 5-1. EXTI 框图 99
图 6-1. GPIO 端口位的基本结构 107
图 6-2. 输入配置的基本结构 108
图 6-3. 输出配置的基本结构 109
图 6-4. 模拟配置的基本结构 109
图 6-5. 备用功能配置的基本结构 110
图 7-1. CRC 管理单元框图 123
图 8-1. DMA 结构框图 129
图 8-2. 握手机制 131
图 8-3. DMA 中断逻辑图 133
图 8-4. DMA 请求映射 134
图 10-1. ADC 模块框图 149
图 10-2. 单次运行模式 151
图 10-3. 连续运行模式 151
图 10-4. 扫描运行模式，连续运行模式禁用 152
图 10-5. 扫描运行模式，连续运行模式使能 152
图 10-6. 间断运行模式 153
图 10-7. 12 位分辨率的数据存储模式 154
图 10-8. 10 位分辨率的数据存储模式 154
图 10-9. 8 位分辨率的数据存储模式 154
图 10-10. 6 位分辨率的数据存储模式 154
图 11-1. DAC 结构框图 166
图 12-1. 比较器框图 174
图 12-2. 比较器迟滞 176
图 13-1. 独立看门狗定时器框图 182
图 13-2. 窗口看门狗定时器框图 187
图 13-3. 窗口看门狗定时器时序图 188
图 14-1. RTC 结构框图 192
图 15-1. 高级定时器结构框图 219
图 15-2. 内部时钟分频为 1 时，计数器的时序图 220

图 15-3. 当 PSC 数值从 0 变到 2 时，计数器的时序图 221
图 15-4. 向上计数时序图，PSC=0/2 222
图 15-5. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值 222
图 15-6. 向下计数时序图，PSC=0/2 223
图 15-7. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值 224
图 15-8. 中央计数模式计数器时序图 225
图 15-9. 中央计数模式下计数器重复时序图 226
图 15-10. 在向上计数模式下计数器重复时序图 226
图 15-11. 在向下计数模式下计数器重复时序图 227
图 15-12. 通道输入捕获原理 227
图 15-13. 通道输出比较原理（带有互补输出的通道，x=0，1，2） 228
图 15-14. 通道输出比较原理 229
图 15-15. 三种输出比较模式 230
图 15-16. EAPWM 时序图 231
图 15-17. CAPWM 时序图 231
图 15-18. 带死区时间的通道互补输出 234
图 15-19. 通道响应中止输入（高电平有效）时，输出信号的行为 235
图 15-20.在正交译码器模式 2 且 CI0FE0 极性不反相时计数器行为 236
图 15-21.在正交译码器模式 2 且 CI0FE0 极性反相时计数器行为 236
图 15-22. 霍尔传感器用在 BLDC 电机控制中 237
图 15-23. 两个定时器之间的霍尔传感器时序图 238
图 15-24. 复位模式下的控制电路 239
图 15-25. 暂停模式下的控制电路 239
图 15-26. 事件模式下的控制电路 240
图 15-27. 单脉冲模式，TIMERx_CHxCV = 4 TIMERx_CAR=99 240
图 15-28. 定时器 0 主/从模式的例子 241
图 15-29. 用定时器 2 的使能信号触发定时器 0 242
图 15-30. 用定时器 2 的 CI0 信号来触发定时器 0 和定时器 2 243
图 15-31. 通用定时器 L0 结构框图 270
图 15-32. 内部时钟分频为 1 时，计数器的时序图 271
图 15-33. 当 PSC 数值从 0 变到 2 时，计数器的时序图 272
图 15-34. 向上计数时序图，PSC=0/2 273
图 15-35. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值 273
图 15-36. 向下计数时序图，PSC=0/2 274
图 15-37. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值 275
图 15-38. 中央计数模式计数器时序图 276
图 15-39. 通道输入捕获原理 277
图 15-40. 通道输出比较原理（x=0，1，2，3） 278
图 15-41. 三种输出比较模式 279
图 15-42. EAPWM 时序图 280
图 15-43. CAPWM 时序图 280
图 15-44. 通用定时器 L2 结构框图 307
图 15-45. 内部时钟分频为 1 时，计数器的时序图 308
图 15-46. 当 PSC 数值从 0 变到 2 时，计数器的时序图 309

图 15-47. 向上计数时序图，PSC=0/2 310
图 15-48. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值 310
图 15-49. 通道输入捕获原理 311
图 15-50. 通道输出比较原理 312
图 15-51. 三种输出比较模式 313
图 15-52. PWM 时序图 314
图 15-53. 通用定时器 L3 结构框图 325
图 15-54. 内部时钟分频为 1 时，计数器的时序图 326
图 15-55. 当 PSC 数值从 0 变到 2 时，计数器的时序图 327
图 15-56. 向上计数时序图，PSC=0/2 328
图 15-57. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值 328
图 15-58. 在向上计数模式下计数器重复时序图 329
图 15-59. 通道输入捕获原理 330
图 15-60. 通道输出比较原理（带有互补输出的通道，x=0） 331
图 15-61. 通道输出比较原理 331
图 15-62. 三种输出比较模式 332
图 15-63. PWM 时序图 333
图 15-64. 带死区时间的互补输出 335
图 15-65. 通道响应中止输入（高电平有效）时，输出信号的行为 336
图 15-66. 复位模式下的控制电路 337
图 15-67. 暂停模式下的控制电路 337
图 15-68. 事件模式下的控制电路 337
图 15-69. 单脉冲模式，TIMERx_CHxCV = 4 TIMERx_CAR=99 338
图 15-70. 通用定时器 L4 结构框图 359
图 15-71. 内部时钟分频为 1 时，计数器的时序图 360
图 15-72. 当 PSC 数值从 0 变到 2 时，计数器的时序图 360
图 15-73. 向上计数时序图，PSC=0/2 361
图 15-74. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值 362
图 15-75. 在向上计数模式下计数器重复时序图 363
图 15-76. 通道输入捕获原理 363
图 15-77. 三种输出比较模式 365
图 15-78. PWM 时序图 366
图 15-79. 带死区时间的互补输出 368
图 15-80. 通道响应中止输入（高电平有效）时，输出信号的行为 369
图 15-81. 单脉冲模式，TIMERx_CHxCV = 4 TIMERx_CAR=99 370
图 15-82. 基本定时器结构框图 385
图 15-83. 内部时钟分频为 1 时，计数器的时序图 386
图 15-84. 当 PSC 数值从 0 变到 2 时，计数器的时序图 386
图 15-85. 向上计数时序图，PSC=0/2 387
图 15-86. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值 388
图 16-1. IFRP 输出时序图 1 394
图 16-2. IFRP 输出时序图 2 395
图 16-3. IFRP 输出时序图 3 395
图 17-1. USART 模块内部框图 398

图 17-2. USART 字符帧 (8 数据位和 1 停止位) 398
图 17-3. USART 发送步骤 400
图 17-4. 过采样方式接收一个数据位（OSB=0） 401
图 17-5. 采用 DMA 方式实现 USART 数据发送配置步骤 402
图 17-6. 采用 DMA 方式实现 USART 数据接收配置步骤 403
图 17-7. 两个 USART 之间的硬件流控制 403
图 17-8. 硬件流控制 404
图 17-9. 空闲状态下检测断开帧 405
图 17-10. 数据传输过程中检测断开帧 405
图 17-11. 同步模式下的 USART 示例 406
图 17-12. 8-bit 格式的 USART 同步通信波形（CLEN=1） 406
图 17-13. IrDA SIR ENDEC 模块 407
图 17-14. IrDA 数据调制 407
图 17-15. ISO7816-3 数据帧格式 408
图 17-16. USART 中断映射框图 411
图 18-1. I2C 模块框图 429
图 18-2. 数据有效性 430
图 18-3. 起始和停止信号 430
图 18-4. 时钟同步 430
图 18-5. SDA 线仲裁 431
图 18-6. 7 位地址的 I2C 通讯流程 431
图 18-7. 10 位地址的 I2C 通讯流程（主机发送） 431
图 18-8. 10 位地址的 I2C 通讯流程（主机接收） 431
图 18-9. 从机发送模式（10 位地址模式） 433
图 18-10. 从机接收模式（10 位地址模式） 434
图 18-11. 主机发送模式（10 位地址模式） 435
图 18-12. 主机接收使用方案 A 模式（10 位地址模式） 437
图 18-13. 主机接收使用方案 B 模式（10 位地址模式） 439
图 19-1. SPI 结构框图 453
图 19-2. 常规模式下的 SPI 时序图 454
图 19-3. 典型的全双工模式连接 456
图 19-4. 典型的单工模式连接（主机：接收，从机：发送） 457
图 19-5. 典型的单工模式连接（主机：只发送，从机：接收） 457
图 19-6. 典型的双向线连接 457
图 19-7. I2S 结构框图 461
图 19-8. I2S 飞利浦标准时序图（DTLEN=00, CHLEN=0, CKPL=0） 462
图 19-9. I2S 飞利浦标准时序图（DTLEN=00, CHLEN=0, CKPL=1） 462
图 19-10. I2S 飞利浦标准时序图（DTLEN=10, CHLEN=1, CKPL=0） 462
图 19-11. I2S 飞利浦标准时序图（DTLEN=10, CHLEN=1, CKPL=1） 462
图 19-12. I2S 飞利浦标准时序图（DTLEN=01, CHLEN=1, CKPL=0） 463
图 19-13. I2S 飞利浦标准时序图（DTLEN=01, CHLEN=1, CKPL=1） 463
图 19-14. I2S 飞利浦标准时序图（DTLEN=00, CHLEN=1, CKPL=0） 463
图 19-15. I2S 飞利浦标准时序图（DTLEN=00, CHLEN=1, CKPL=1） 463
图 19-16. MSB 对齐标准时序图（DTLEN=00, CHLEN=0, CKPL=0） 464

图 19-17. MSB 对齐标准时序图（DTLEN=00, CHLEN=0, CKPL=1）..... 464
图 19-18. MSB 对齐标准时序图（DTLEN=10, CHLEN=1, CKPL=0）..... 464
图 19-19. MSB 对齐标准时序图（DTLEN=10, CHLEN=1, CKPL=1）..... 464
图 19-20. MSB 对齐标准时序图（DTLEN=01, CHLEN=1, CKPL=0）..... 464
图 19-21. MSB 对齐标准时序图（DTLEN=01, CHLEN=1, CKPL=1）..... 464
图 19-22. MSB 对齐标准时序图（DTLEN=00, CHLEN=1, CKPL=0）..... 465
图 19-23. MSB 对齐标准时序图（DTLEN=00, CHLEN=1, CKPL=1）..... 465
图 19-24. LSB 对齐标准时序图（DTLEN=01, CHLEN=1, CKPL=0）..... 465
图 19-25. LSB 对齐标准时序图（DTLEN=01, CHLEN=1, CKPL=1）..... 465
图 19-26. LSB 对齐标准时序图（DTLEN=00, CHLEN=1, CKPL=0）..... 466
图 19-27. LSB 对齐标准时序图（DTLEN=00, CHLEN=1, CKPL=1）..... 466
图 19-28. PCM 标准短帧同步模式时序图（DTLEN=00, CHLEN=0, CKPL=0）..... 466
图 19-29. PCM 标准短帧同步模式时序图（DTLEN=00, CHLEN=0, CKPL=1）..... 466
图 19-30. PCM 标准短帧同步模式时序图（DTLEN=10, CHLEN=1, CKPL=0）..... 466
图 19-31. PCM 标准短帧同步模式时序图（DTLEN=10, CHLEN=1, CKPL=1）..... 467
图 19-32. PCM 标准短帧同步模式时序图（DTLEN=01, CHLEN=1, CKPL=0）..... 467
图 19-33. PCM 标准短帧同步模式时序图（DTLEN=01, CHLEN=1, CKPL=1）..... 467
图 19-34. PCM 标准短帧同步模式时序图（DTLEN=00, CHLEN=1, CKPL=0）..... 467
图 19-35. PCM 标准短帧同步模式时序图（DTLEN=00, CHLEN=1, CKPL=1）..... 467
图 19-36. PCM 标准长帧同步模式时序图（DTLEN=00, CHLEN=0, CKPL=0）..... 467
图 19-37. PCM 标准长帧同步模式时序图（DTLEN=00, CHLEN=0, CKPL=1）..... 468
图 19-38. PCM 标准长帧同步模式时序图（DTLEN=10, CHLEN=1, CKPL=0）..... 468
图 19-39. PCM 标准长帧同步模式时序图（DTLEN=10, CHLEN=1, CKPL=1）..... 468
图 19-40. PCM 标准长帧同步模式时序图（DTLEN=01, CHLEN=1, CKPL=0）..... 468
图 19-41. PCM 标准长帧同步模式时序图（DTLEN=01, CHLEN=1, CKPL=1）..... 468
图 19-42. PCM 标准长帧同步模式时序图（DTLEN=00, CHLEN=1, CKPL=0）..... 468
图 19-43. PCM 标准长帧同步模式时序图（DTLEN=00, CHLEN=1, CKPL=1）..... 469
图 19-44. I2S 时钟生成结构框图..... 469
图 19-45. I2S 初始化流程..... 471
图 19-46. I2S 主机接收禁能流程..... 473
图 20-1. HDMI-CEC 控制器框图..... 486
图 20-2. 信息结构..... 486
图 20-3. 起始位时序..... 487
图 20-4. 数据位时序..... 487
图 20-5. CEC 线仲裁过程..... 488
图 20-6. 信号空闲时间..... 488
图 20-7. 错误位周期..... 489
图 20-8. 长错误位时序..... 490
图 20-9. 传输错误监测..... 491
图 21-1. TSI 模块框图..... 500
图 21-2. 通道引脚与采样引脚的框图..... 501
图 21-3. 电荷序列转移期间的采样引脚的电压..... 503
图 21-4. 电荷转移序列的有限状态机的状态转移图..... 503
图 22-1. USBD 模块图..... 515

图 22-2. 缓冲描述符表的用法示例 (USBD_BADDR = 0)..... 517

表索引

表 1-1. GD32F130xx 和 GD32F150xx 产品的存储器映射 25
表 1-2. Flash 模块组织 27
表 1-3. 引导模式 28
表 2-1. 闪存的基地址和大小 38
表 2-2. 选项字节 43
表 2-3. OB_WP 位对应页保护 44
表 3-1. 节电模式总结 58
表 4-1. 时钟源的选择 68
表 4-2. 深度睡眠模式下内核电压选择 69
表 5-1. Cotrex®-M3 中的 NVIC 异常类型 96
表 5-2.中断向量表 97
表 5-3. EXTI 触发源 100
表 6-1. GPIO 配置表 106
表 8-1. DMA 传输操作 130
表 8-2. 中断事件 132
表 8-3. DMA 各通道请求表 135
表 10-1. ADC 内部输入信号 149
表 10-2. ADC 输入引脚定义 149
表 10-3. ADC 常规序列的外部触发源 155
表 11-1. DAC 引脚 167
表 11-2. DAC 触发与输出 167
表 11-3. DAC 外部触发 168
表 12-1 CMP 的输入和输出总结 175
表 13-1. 独立看门狗定时器在 40KHz（IRC40K）时的最小 / 最大超时周期 182
表 13-2. 在 36MHz（f_{PCLK1}）时的最大 / 最小超时值 188
表 14-1. 省电模式管理 200
表 14-2. 中断控制 201
表 15-1. 定时器（TIMERx）分为六种类型 217
表 15-2. 由参数控制的互补输出表 232
表 15-3. 不同正交译码器模式下的计数方向 235
表 15-4. 从模式例子列表 238
表 15-5. 定时器 0 内部互联 241
表 15-6. TIMERx(x=1,2)定时器内部互连 281
表 15-7. 由参数控制的互补输出表 334
表 15-8. 从模式例子列表 336
表 15-9. TIMERx(x=14)定时器内部互连 338
表 15-10. 由参数控制的互补输出表 367
表 17-1. USART 重要管脚描述 397
表 17-2. 停止位配置 398
表 17-3. USART 中断请求 410
表 18-1. I2C 总线术语说明（参考飞利浦 I2C 规范） 429

表 18-2. 事件状态标志位 442
表 18-3. 错误标志位 442
表 19-1. SPI 信号描述 453
表 19-2. 从机模式 NSS 功能 454
表 19-3. 主机模式 NSS 功能 455
表 19-4. SPI 运行模式 455
表 19-5. SPI 中断请求 460
表 19-6. I2S 比特率计算公式 469
表 19-7. 音频采样频率计算公式 469
表 19-8. 各种运行模式下 I2S 接口信号的方向 470
表 19-9. I2S 中断 475
表 20-1. 帧结构 486
表 20-2. 数据位时序参数表 487
表 20-3. 信号空闲时间的大小与应用场景的关系 488
表 20-4. 错误处理时序参数表 490
表 20-5. 时序参数表 491
表 20-6. HDMI-CEC 中断 492
表 21-1. 电荷转移序列中的引脚和模拟开关状态 501
表 21-2. 充电扩展状态的持续时间 504
表 21-3. TSI 错误和标志位 506
表 21-4. TSI 引脚 506
表 22-1. USBD 信号描述 515
表 22-2. 双缓冲标志定义 518
表 22-3. 双缓冲的用法 518
表 23-1. 寄存器功能位访问属性 532
表 23-2. 术语 532
表 24-1. 版本历史 533

1. 系统及存储器架构

GD32F1x0系列器件是基于ARM® Cortex®-M3处理器的32位通用微控制器。ARM® Cortex®-M3处理器包括三条AHB总线，分别称为I-CODE总线、D-Code总线和系统总线。Cortex®-M3处理器的所有存储访问，根据不同的目的和目标存储空间，都会在这三条总线上执行。存储器的组织采用了哈佛结构，预先定义的存储器映射和高达4 GB的存储空间，充分保证了系统的灵活性和可扩展性。

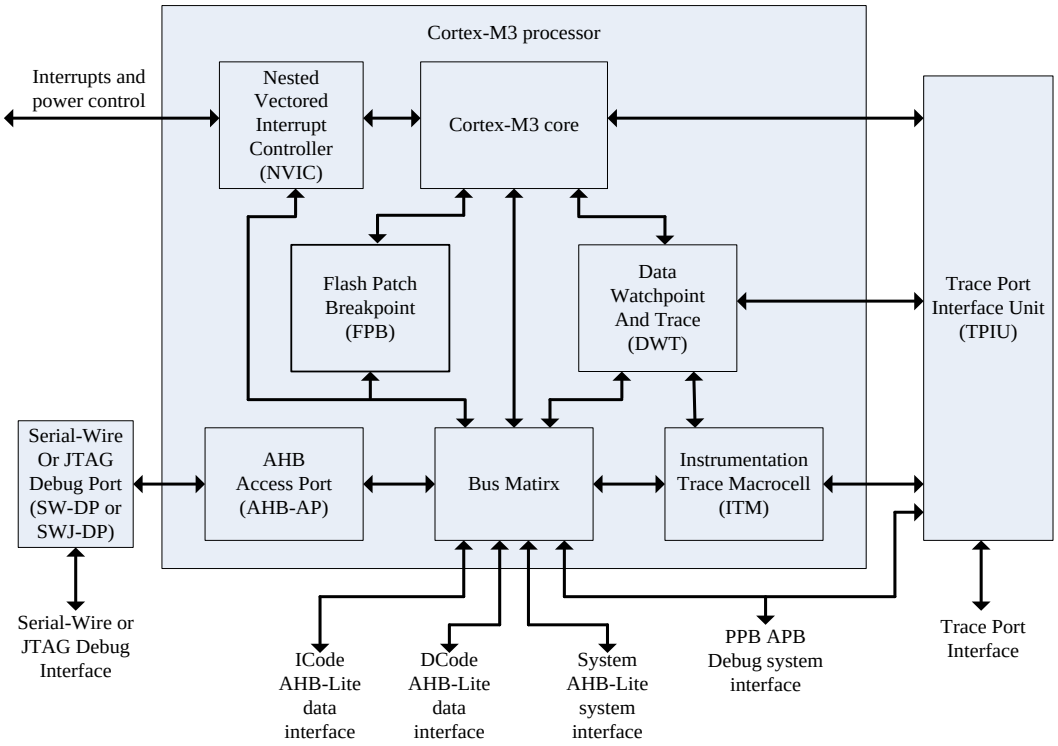
1.1. ARM Cortex-M3 处理器

Cortex®-M3处理器是一个具有低中断延迟时间和低成本调试特性的32位处理器。高集成度和增强的特性使Cortex®-M3处理器适合于那些需要高性能和低功耗微控制器的市场领域。Cortex®-M3处理器基于ARMv7架构，并且支持一个强大且可扩展的指令集，包括通用数据处理I/O控制任务、增强的数据处理位域操作。下面列出由Cortex®-M3提供的一些系统外设：

- 内部总线矩阵，用于实现I-Code总线、D-Code总线、系统总线、专用总线（PPB）以及调试专用总线（AHB-AP）的互联；
- 嵌套式向量型中断控制器（NVIC）；
- 闪存地址重载及断点单元（FPB）；
- 数据观测点及跟踪单元（DWT）；
- 指令跟踪宏单元（ITM）；
- 串行线和JTAG调试接口（SWJ-DP）；
- 跟踪端口接口单元（TPIU）；

下图显示了Cortex®-M3处理器结构框图。欲了解更多信息，请参阅ARM® Cortex®-M3技术参考手册。

图 1-1. Cortex®-M3 框图



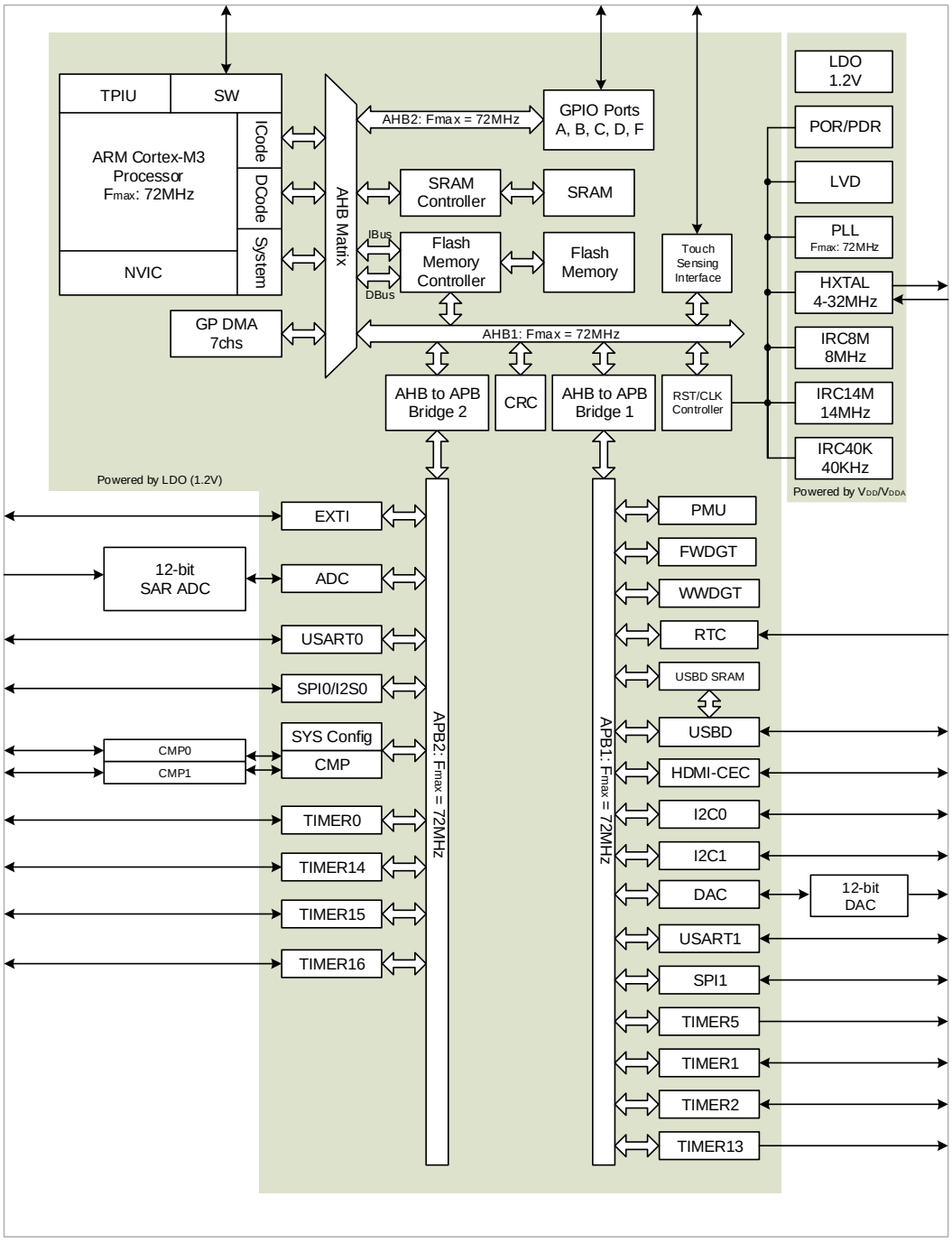
1.2. 系统架构

GD32F1x0系列器件的系统架构如下图所示。该AHB矩阵是一个基于AMBA 3.0 AHB-LITE的多层总线，这个结构使得系统中的多个主机和从机之间的并行通信成为可能。该AHB矩阵中包含属于Cortex®-M3内核的I-Code总线、D-Code总线和系统总线，以及内核外的DMA共4个主机。

I-Code总线是Cortex®-M3内核的指令总线，用于从代码区域(0x0000 0000～0x1FFF FFFF)中获取向量。D-Code总线用于加载和存储数据，以及代码区域的调试访问。类似的，系统总线用于指令和向量获取、数据加载和存储以及系统区域的调试访问。系统区域包括内部SRAM区域和外设区域。该AHB矩阵还连接了5个从机，分别为：FMC的I-Code、FMC的D-Code、内部SRAM、AHB1和AHB2。

AHB2连接GPIO端口。AHB1连接AHB外设，包括2个AHB-APB总线桥。AHB-APB总线桥提供了AHB1和两条APB总线之间的全同步连接。两条APB总线连接了所有的APB外设。

图 1-2. GD32F130xx 和 GD32F150xx 产品的系统结构



1.3. 存储器映射

ARM® Cortex®-M3处理器采用哈佛结构，可以使用相互独立的总线来读取指令和加载/存储数据。指令代码和数据都位于相同的存储器地址空间，但在不同的地址范围。程序存储器，数据存储器，寄存器和I/O端口都在同一个线性的4 GB的地址空间之内。这是Cortex®-M3的最大地址范围，因为它的地址总线宽度是32位。此外，为了降低不同客户在相同应用时的软件复杂度，存储映射是按Cortex®-M3处理器提供的规则预先定义的。同时，一部分地址空间由ARM® Cortex®-M3的系统外设所占用。下表显示了GD32F1x0系列器件的存储器映射，包括代码、

SRAM、外设和其他预先定义的区域。几乎每个外设都分配了1KB的地址空间，这样可以简化每个外设的地址译码。

表 1-1. GD32F130xx 和 GD32F150xx 产品的存储器映射

预先定义的地址空间	总线	地址范围	外设
		0xE000 0000 - 0xE00F FFFF	Cortex M3 内部外设
外部设备		0xA000 0000 - 0xDFFF FFFF	保留
外部 RAM		0x6000 0000 - 0x9FFF FFFF	保留
外设	AHB1	0x5000 0000 - 0x5FFF FFFF	保留
	AHB2	0x4800 1800 - 0x4FFF FFFF	保留
		0x4800 1400 - 0x4800 17FF	GPIOF
		0x4800 1000 - 0x4800 13FF	保留
		0x4800 0C00 - 0x4800 0FFF	GPIOD
		0x4800 0800 - 0x4800 0BFF	GPIOC
		0x4800 0400 - 0x4800 07FF	GPIOB
		0x4800 0000 - 0x4800 03FF	GPIOA
	AHB1	0x4002 4400 - 0x47FF FFFF	保留
		0x4002 4000 - 0x4002 43FF	TSI
		0x4002 3400 - 0x4002 3FFF	保留
		0x4002 3000 - 0x4002 33FF	CRC
		0x4002 2400 - 0x4002 2FFF	保留
		0x4002 2000 - 0x4002 23FF	FMC
		0x4002 1400 - 0x4002 1FFF	保留
		0x4002 1000 - 0x4002 13FF	RCU
		0x4002 0400 - 0x4002 0FFF	保留
		0x4002 0000 - 0x4002 03FF	DMA
	APB2	0x4001 4C00 - 0x4001 FFFF	保留
		0x4001 4800 - 0x4001 4BFF	TIMER16
		0x4001 4400 - 0x4001 47FF	TIMER15
		0x4001 4000 - 0x4001 43FF	TIMER14
		0x4001 3C00 - 0x4001 3FFF	保留
		0x4001 3800 - 0x4001 3BFF	USART0
		0x4001 3400 - 0x4001 37FF	保留
		0x4001 3000 - 0x4001 33FF	SPI0/I2S0
		0x4001 2C00 - 0x4001 2FFF	TIMER0
		0x4001 2800 - 0x4001 2BFF	保留
		0x4001 2400 - 0x4001 27FF	ADC
		0x4001 0800 - 0x4001 23FF	保留
		0x4001 0400 - 0x4001 07FF	EXTI
		0x4001 0000 - 0x4001 03FF	SYSCFG+CMP
	APB1	0x4000 C400 - 0x4000 FFFF	保留
		0x4000 C000 - 0x4000 C3FF	保留

预先定义的 地址空间	总线	地址范围	外设
		0x4000 8000 - 0x4000 BFFF	保留
		0x4000 7C00 - 0x4000 7FFF	保留
		0x4000 7800 - 0x4000 7BFF	CEC
		0x4000 7400 - 0x4000 77FF	DAC
		0x4000 7000 - 0x4000 73FF	PMU
		0x4000 6400 - 0x4000 6FFF	保留
		0x4000 6000 - 0x4000 63FF	USB SRAM
		0x4000 5C00 - 0x4000 5FFF	USB registers
		0x4000 5800 - 0x4000 5BFF	I2C1
		0x4000 5400 - 0x4000 57FF	I2C0
		0x4000 4800 - 0x4000 53FF	保留
		0x4000 4400 - 0x4000 47FF	USART1
		0x4000 4000 - 0x4000 43FF	保留
		0x4000 3C00 - 0x4000 3FFF	保留
		0x4000 3800 - 0x4000 3BFF	SPI1
		0x4000 3400 - 0x4000 37FF	保留
		0x4000 3000 - 0x4000 33FF	FWDGT
		0x4000 2C00 - 0x4000 2FFF	WWDGT
		0x4000 2800 - 0x4000 2BFF	RTC
		0x4000 2400 - 0x4000 27FF	保留
		0x4000 2000 - 0x4000 23FF	TIMER13
		0x4000 1400 - 0x4000 1FFF	保留
		0x4000 1000 - 0x4000 13FF	TIMER5
		0x4000 0800 - 0x4000 0FFF	保留
		0x4000 0400 - 0x4000 07FF	TIMER2
		0x4000 0000 - 0x4000 03FF	TIMER1
SRAM		0x2000 2000 - 0x3FFF FFFF	保留
		0x2000 0000 - 0x2000 1FFF	SRAM
代码		0x1FFF F810 - 0x1FFF FFFF	保留
		0x1FFF F800 - 0x1FFF F80F	Option bytes
		0x1FFF EC00 - 0x1FFF F7FF	System memory
		0x0801 0000 - 0x1FFF EBFF	保留
		0x0800 0000 - 0x0800 FFFF	Main Flash memory
		0x0000 0000 - 0x07FF FFFF	Aliased to Flash or system memory

1.3.1. 位带操作

为了减少“读-改-写”操作的次数，Cortex®-M3处理器提供了一个可以执行单原子比特操作的位带功能。存储器映射包含了两个支持位带操作的区域，分别位于SRAM和外设中。位带区域将存储器别名区的每个字映射到存储器位带区的某个位上。

下面的映射公式表明了别名区中的每个字如何对应位带区的相应比特或目标比特。

bit_word_addr =bit_band_base +(byte_offset×32)+(bit_number×4) (式1-1)

其中：

- bit_word_addr指的是映射到位带区目标比特的别名存储器区字地址；
- bit_band_base指的是别名区的起始地址；
- byte_offset指的是位带区目标比特所在的字节的字节地址偏移量；
- bit_number指的是目标比特在对应字节中的位置（0-7）。

例如，要想访问0x2000 0200地址的第7位，可访问的位带别名区地址是：

bit_word_addr = 0x2200 0000 + (0x200 * 32)+(7 * 4)= 0x2200 401C (式1-2)

如果对0x2200 401C进行写操作，那么0x2000 0200的第7位将会相应变化；如果对0x2200 401C进行读操作，那么视0x2000 0200的第7位状态而返回0x01或0x00。

1.3.2. 片上 SRAM

GD32F1x0系列微控制器含有高达8KB的片上SRAM，起始地址为0x2000 0000，支持字节、半字（16比特）和整字（32比特）访问。存储器支持奇偶校验来提高鲁棒性。用户可以通过用户选项字节（请参考第2.3.9章节*选项字节说明*）的SRAM_PARITY_CHECK位来启用奇偶校验功能。当启用时，如果校验失败，产生一个NMI中断。SRAM奇偶校验错误标志在系统配置寄存器 2 （ SYSCFG_CFG2 ） 之 中 。 如 果 系 统 配 置 寄 存 器 2 （ SYSCFG_CFG2 ） 的SRAM_PARITY_ERROR_LOCK位置1，错误标志将被连接到定时器0/定时器14/定时器15/定时器16的break输入端。

SRAM的真实数据宽度为36位，包括32位数据和4位奇偶校验（每字节1位）位。在写入时，奇偶校验位被计算并存储到SRAM。当读取时，奇偶校验位会用SRAM读出的数据再计算一遍。计算出的奇偶校验位将与读出的奇偶校验位（写入访问期间计算并存储的奇偶校验位）进行比较。如果它们不相同，则奇偶校验失败。

注意：如果启用了SRAM奇偶校验，建议通过软件在代码的开始初始化整个SRAM存储器，以避免读取未初始化的位置时，得到的奇偶校验错误。

1.3.3. 片上闪存

该系列微控制器提供高达64KB的片上闪存。片上闪存包括高达64KB的主闪存块和3KB容量的用于存储引导装载程序（boot loader）的信息块。主存储块分为64页，每页的容量为1KB。下表显示详细信息。

表 1-2. Flash 模块组织

模块	名称	地址	尺寸
主闪存块	Page 0	0x0800 0000 - 0x0800 03FF	1 Kbytes
	Page 1	0x0800 0400 - 0x0800 07FF	1 Kbytes
	Page 2	0x0800 0800 - 0x0800 0BFF	1 Kbytes
	.	.	.
	.	.	.
	Page 63	0x0800 FC00 - 0x0800 FFFF	1 Kbytes

模块	名称	地址	尺寸
信息块	系统存储器	0x1FFF EC00 - 0x1FFF F7FF	3 Kbytes
选项字节块	选项字节	0x1FFF F800 - 0x1FFF F80F	16 bytes

针对前32页的读访问可实现每个周期无任何等待状态读出32比特数据的效率。读访问支持字节、半字（16比特）和整字（32比特0；写访问（编程）只支持半字（16比特）和整字（32比特）。片上闪存的每一页都可以单独被擦除，整个主闪存块也可以同时被擦除。

1.4. 引导配置

GD32F1x0系列微控制器提供了三种引导源，可以通过用户选项字节BOOT1_n位（请参考第2.3.9章节*选项字节说明*）和BOOT0引脚进行选择。BOOT0引脚的电平值是在复位后第4个系统时钟上升沿锁存的。用户可自行选择所需要的引导源，通过设置上电复位或系统复位后的BOOT1_n位和BOOT0的引脚电平值。下表描述了详细的引导模式信息。

表 1-3. 引导模式

引导源选择	启动模式选择引脚	
	BOOT1	BOOT0
主FLASH存储器	x	0
系统存储器	0	1
片上SRAM	1	1

注意：BOOT1值与BOOT1_n值相反。

上电序列或系统复位后，ARM® Cortex®-M3处理器先从0x0000 0000地址获取栈顶值，再从0x0000 0004地址获得引导代码的基地址，然后从引导代码的基地址开始执行程序。

根据所选的引导源，片上闪存的主存（开始于0x0800 0000的存储空间）或系统存储器（开始于0x1FFF EC00的存储空间）会被映射到引导空间，即从0x0000 0000开始的地址空间。如果片上SRAM（开始于0x2000 0000的存储空间）被选为引导源，用户必须在应用程序初始化代码中通过修改NVIC异常向量表和偏移寄存器将向量表重置到SRAM中。

芯片内嵌的引导装载程序位于系统存储器中，用来对片上闪存的主存进行重编程。该引导装载程序可通过以下串行接口之一工作：USART0或USART1。

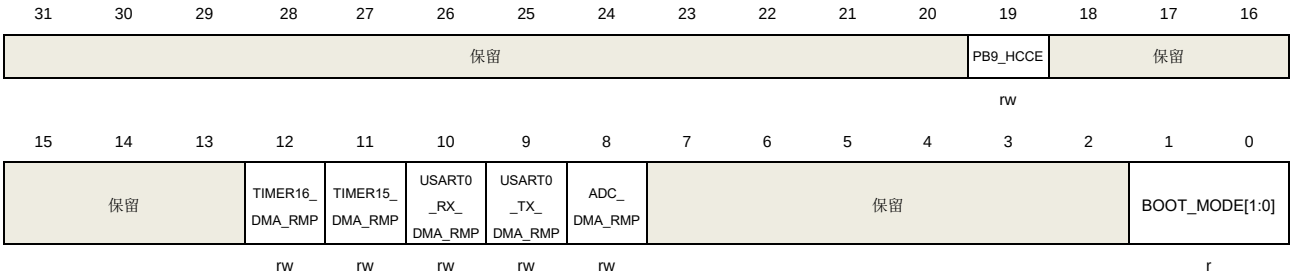
1.5. 系统配置寄存器（SYSCFG）

SYSCFG基地址：0x4001 0000

1.5.1. 系统配置寄存器 0（SYSCFG_CFG0）

地址偏移：0x00
复位值：0x0000 000X（根据BOOT0引脚的状态和用户选项字节的BOOT1_n的值，X表示BOOT_MODE[1:0]可能为任意值）。

该寄存器只能按字（32位）访问。



位/位域	名称	描述
31:20	保留	必须保持复位值。
19	PB9_HCCE	PB9引脚大电流能力使能 当该位为1时，PB9引脚可以直接用来控制红外发光二极管。 0：PB9引脚大电流能力关闭 1：PB9引脚大电流能力开启，同时该引脚的速度控制被忽略
18:13	保留	必须保持复位值。
12	TIMER16_DMA_RMP	TIMER16 DMA请求重映射使能 0：不重映射（TIMER16_CH0和TIMER16_UP DMA被映射在DMA通道0） 1：重映射（TIMER16_CH0和TIMER16_UP DMA被映射在DMA通道1）
11	TIMER15_DMA_RMP	TIMER15 DMA请求重映射使能 0：不重映射（TIMER15_CH0和TIMER15_UP DMA被映射在DMA通道2） 1：重映射（TIMER15_CH0和TIMER15_UP DMA被映射在DMA通道3）
10	USART0_RX_DMA_RMP	USART0_RX DMA请求重映射使能 0：不重映射（USART0_RX DMA被映射在DMA通道2） 1：重映射（USART0_RX DMA被映射在DMA通道4）
9	USART0_TX_DMA_RMP	USART0_TX DMA请求重映射使能 0：不重映射（USART0_TX DMA被映射在DMA通道1） 1：重映射（USART0_TX DMA被映射在DMA通道3）
8	ADC_DMA_RMP	ADC DMA请求重映射使能 0：不重映射（ADC DMA被映射在DMA通道0）

		1：重映射（ADC DMA被映射在DMA通道1）
7:2	保留	必须保持复位值。
1:0	BOOT_MODE[1:0]	引导模式（详细请参考第1.4章节 *引导配置*） bit0 映射到BOOT0引脚；bit1的值与BOOT1_n的值相反。 x0：从片上闪存的主存引导启动 01：从片上闪存的系统存储器引导启动 11：从片上SRAM引导启动

1.5.2. EXTI 源选择寄存器 0（SYSCFG_EXTISS0）

地址偏移：0x08

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI3_SS [3:0]				EXTI2_SS [3:0]				EXTI1_SS [3:0]				EXTI0_SS [3:0]			
rw				rw				rw				rw			

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:12	EXTI3_SS[3:0]	EXTI 3源选择 X000：PA3引脚 X001：PB3引脚 X010：PC3引脚 X011：保留 X100：保留 X101：保留 X110：保留 X111：保留
11:8	EXTI2_SS[3:0]	EXTI 2源选择 X000：PA2引脚 X001：PB2引脚 X010：PC2引脚 X011：PD2引脚 X100：保留 X101：保留 X110：保留 X111：保留
7:4	EXTI1_SS[3:0]	EXTI 1源选择

		X000：PA1引脚 X001：PB1引脚 X010：PC1引脚 X011：保留 X100：保留 X101：PF1引脚 X110：保留 X111：保留
3:0	EXTI0_SS[3:0]	EXTI 0 源选择 X000：PA0引脚 X001：PB0引脚 X010：PC0引脚 X011：保留 X100：保留 X101：PF0引脚 X110：保留 X111：保留

1.5.3. EXTI 源选择寄存器 1（SYSCFG_EXTISS1）

地址偏移：0x0C
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI7_SS [3:0]				EXTI6_SS [3:0]				EXTI5_SS [3:0]				EXTI4_SS [3:0]			
rw				rw				rw				rw			

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:12	EXTI7_SS[3:0]	EXTI 7源选择 X000：PA7引脚 X001：PB7引脚 X010：PC7引脚 X011：保留 X100：保留 X101：PF7引脚 X110：保留 X111：保留
11:8	EXTI6_SS[3:0]	EXTI 6源选择

		X000：PA6引脚 X001：PB6引脚 X010：PC6引脚 X011：保留 X100：保留 X101：PF6引脚 X110：保留 X111：保留
7:4	EXTI5_SS[3:0]	EXTI 5源选择 X000：PA5引脚 X001：PB5引脚 X010：PC5引脚 X011：保留 X100：保留 X101：PF5引脚 X110：保留 X111：保留
3:0	EXTI4_SS[3:0]	EXTI 4源选择 X000：PA4引脚 X001：PB4引脚 X010：PC4引脚 X011：保留 X100：保留 X101：PF4引脚 X110：保留 X111：保留

1.5.4. EXTI 源选择寄存器 2（SYSCFG_EXTISS2）

地址偏移：0x10

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI11_SS [3:0]				EXTI10_SS [3:0]				EXTI9_SS [3:0]				EXTI8_SS [3:0]			
rw				rw				rw				rw			

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:12	EXTI11_SS[3:0]	EXTI 11源选择

		X000：PA11引脚 X001：PB11引脚 X010：PC11引脚 X011：保留 X100：保留 X101：保留 X110：保留 X111：保留
11:8	EXTI10_SS[3:0]	EXTI 10源选择 X000：PA10引脚 X001：PB10引脚 X010：PC10引脚 X011：保留 X100：保留 X101：保留 X110：保留 X111：保留
7:4	EXTI9_SS[3:0]	EXTI 9源选择 X000：PA9引脚 X001：PB9引脚 X010：PC9引脚 X011：保留 X100：保留 X101：保留 X110：保留 X111：保留
3:0	EXTI8_SS[3:0]	EXTI 8源选择 X000：PA8引脚 X001：PB8引脚 X010：PC8引脚 X011：保留 X100：保留 X101：保留 X110：保留 X111：保留

1.5.5. EXTI 源选择寄存器 3（SYSCFG_EXTISS3）

地址偏移：0x14
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

保留

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI15_SS [3:0]				EXTI14_SS [3:0]				EXTI13_SS [3:0]				EXTI12_SS [3:0]			
rw				rw				rw				rw			

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:12	EXTI15_SS[3:0]	EXTI 15源选择 X000：PA15引脚 X001：PB15引脚 X010：PC15引脚 X011：保留 X100：保留 X101：保留 X110：保留 X111：保留
11:8	EXTI14_SS[3:0]	EXTI 14源选择 X000：PA14引脚 X001：PB14引脚 X010：PC14引脚 X011：保留 X100：保留 X101：保留 X110：保留 X111：保留
7:4	EXTI13_SS[3:0]	EXTI 13源选择 X000：PA13引脚 X001：PB13引脚 X010：PC13引脚 X011：保留 X100：保留 X101：保留 X110：保留 X111：保留
3:0	EXTI12_SS[3:0]	EXTI 12源选择 X000：PA12引脚 X001：PB12引脚 X010：PC12引脚 X011：保留 X100：保留 X101：保留

X110：保留

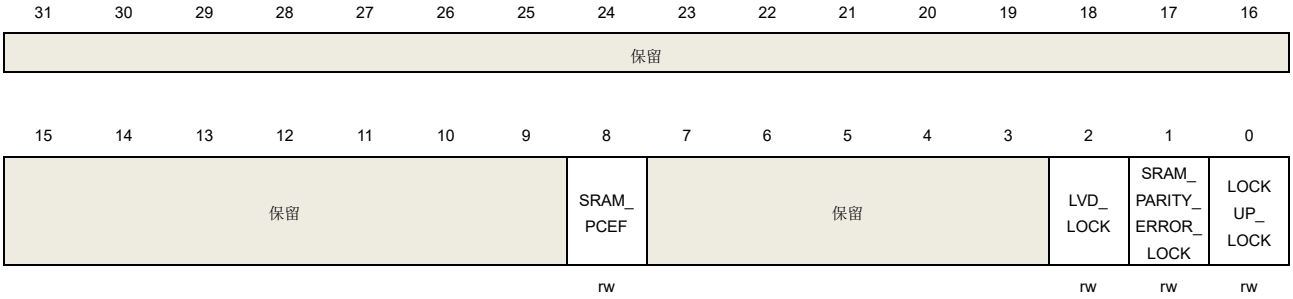
X111：保留

1.5.6. 系统配置寄存器 2（SYSCFG_CFG2）

地址偏移：0x18

复位值：0x0000 0000

该寄存器只能按字（32位）访问。



位/位域	名称	描述
31:9	保留	必须保持复位值。
8	SRAM_PCEF	SRAM奇偶校验错误标志 当SRAM奇偶校验错误发生时，该位由硬件置1。该位由软件写1清零。 0：未检测到SRAM奇偶校验错误 1：检测到SRAM奇偶校验错误
7:3	保留	必须保持复位值
2	LVD_LOCK	LVD锁定 该位由软件置1，在系统复位时才能清零。 0：LVD中断从TIMER0 / 14 / 15 / 16的break输入端断开。PMU_CTL寄存器的LVDEN和LVDT[2:0]可以被设置。 1：LVD中断与TIMER0 / 14 / 15 / 16的break输入端连接。PMU_CTL寄存器的LVDEN和LVDT[2:0]仅仅可读。
1	SRAM_PARITY_ ERROR_LOCK	SRAM奇偶校验错误锁定 该位由软件置1，在系统复位时才能清零。 0：SRAM奇偶校验错误从TIMER0 / 14 / 15 / 16的break输入端断开 1：SRAM奇偶校验错误与TIMER0 / 14 / 15 / 16的break输入端连接
0	LOCKUP_LOCK	Cortex®-M3 LOCKUP输出锁定 该位由软件置1，在系统复位时才能清零。 0：Cortex®-M3 LOCKUP输出从TIMER0 / 14 / 15 / 16的break输入端断开 1：Cortex®-M3 LOCKUP输出与TIMER0 / 14 / 15 / 16的break输入端连接

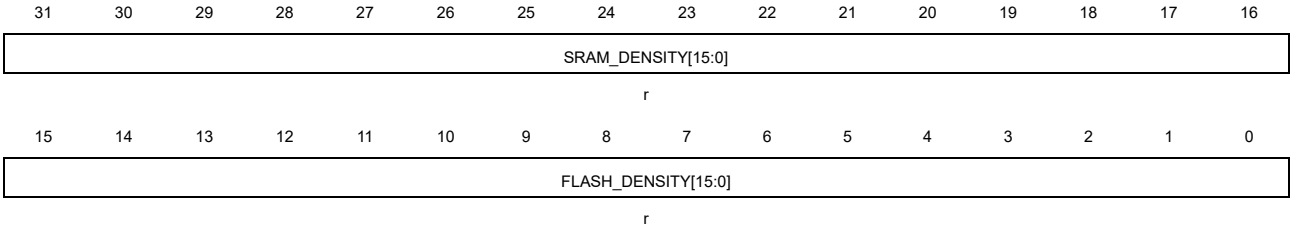
1.6. 设备电子签名

设备的电子签名中包含的存储容量信息和96位的唯一设备ID。它被存储在片上闪存的信息模块中。96位唯一设备ID对于每颗芯片而言都是唯一的。它可以用作序列号，或安全密钥的一部分，等等。

1.6.1. 存储容量信息

基地址：0x1FFF F7E0
该值是原厂设定的，不能由用户修改。

该寄存器只能按字（32位）访问。

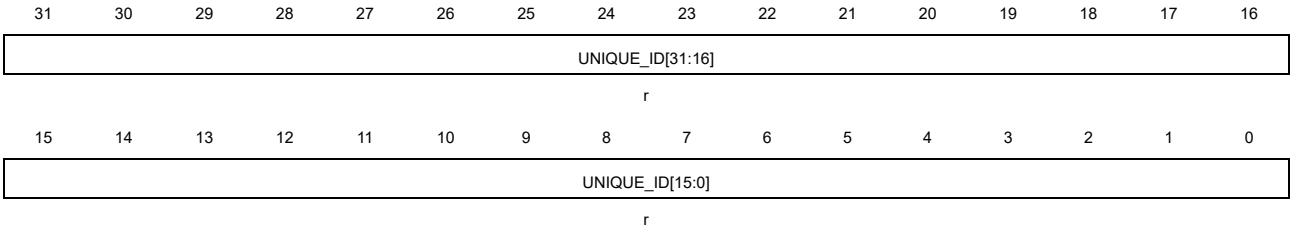


位/位域	名称	描述
31:16	SRAM_DENSITY [15:0]	SRAM存储器容量 该值表明芯片的片上SRAM存储器容量，以Kbytes为单位。 例如：0x0008表示8Kbytes。
15:0	FLASH_DENSITY [15:0]	Flash存储器容量 该值表明芯片的片上Flash容量，以Kbytes为单位。 例如：0x0020表示32Kbytes。

1.6.2. 设备唯一 ID（96 位/位域）

基地址：0x1FFF F7AC
该值是原厂设定的，不能由用户修改。

该寄存器只能按字（32位）访问。



位/位域	名称	描述
31:0	UNIQUE_ID[31:0]	设备唯一ID。

基地址：0x1FFF F7B0
该值是原厂设定的，不能由用户修改。

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UNIQUE_ID[63:48]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNIQUE_ID[47:32]															
r															

位/位域	名称	描述
31:0	UNIQUE_ID[63:32]	设备唯一ID。

基地址：0x1FFF F7B4
该值是原厂设定的，不能由用户修改。

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UNIQUE_ID[95:80]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNIQUE_ID[79:64]															
r															

位/位域	名称	描述
31:0	UNIQUE_ID[95:64]	唯一设备ID。

2. 闪存控制器（FMC）

2.1. 简介

闪存控制器（FMC），提供了片上闪存需要的所有功能。在闪存的前32K字节空间内，CPU执行指令零等待。FMC也提供了页擦除、整片擦除以及32位整字或16位半字编程等操作。

2.2. 主要特性

- 高达64KB字节的片上闪存可用于存储指令或数据；
- 在闪存的前32K字节空间内，CPU执行指令零等待；
- 从闪存的32K ~ 64K地址空间内取数据有比较长的延迟；
- 3K字节引导装载程序信息块；
- 16字节的选项字节块用于用户需求；
- 每页大小为1K字节；
- 32位字或16位半字编程，支持页擦除和整片擦除；
- 有闪存读保护功能，阻止非法代码或数据进入；
- 有页擦除和页编程保护功能，阻止意外操作；

2.3. 功能描述

2.3.1. 闪存结构

闪存存储器包括一个高达64K字节的主闪存块（按64页每页1K字节分块）和一个3K字节的用于引导装载程序的信息块。主闪存存储器的64页中每页都可以单独擦除。闪存存储器的基地址和大小见下表。

表 2-1. 闪存的基地址和大小

闪存块	名称	地址范围	大小(字节)
主闪存块	Page 0	0x0800 0000 - 0x0800 03FF	1KB
	Page 1	0x0800 0400 - 0x0800 07FF	1KB
	Page 2	0x0800 0800 - 0x0800 0BFF	1KB
	.	.	.
	.	.	.
	.	.	.
	Page 63	0x0800 FC00 - 0x0800 FFFF	1KB
信息块	引导装载程序	0x1FFF EC00 - 0x1FFF F7FF	3KB
选项字节块	选项字节	0x1FFF F800 - 0x1FFF F80F	16B

注意: 信息块存储了引导装载程序（boot loader），不能被用户编程或擦除。

2.3.2. 读操作

闪存可以像普通存储空间一样直接寻址访问。对闪存取指令和取数据分别使用CPU的IBUS或DBUS总线。

2.3.3. FMC_CTL 寄存器解锁

复位后，FMC_CTL寄存器不可以用写模式进行访问（OBRLD位除外，此位用于重加载选项字节）并且FMC_CTL寄存器中的LK位被置为1。一个包含对FMC_KEY寄存器进行两次写操作的解锁序列可以解锁FMC_CTL寄存器的访问，分别是先后写入0x45670123和0xCDEF89AB。两次写操作后，FMC_CTL寄存器的LK位将被硬件清0。可以通过软件设置FMC_CTL寄存器的LK位为1再次锁定FMC_CTL寄存器。任何对FMC_KEY寄存器的错误操作都会将LK位置1，从而锁定FMC_CTL寄存器，并且引发一个总线错误。

FMC_CTL寄存器的OBPG位和OBER位可以被FMC_OBKEY寄存器锁定。解锁序列是向FMC_OBKEY寄存器先后写入0x45670123和0xCDEF89AB，然后将FMC_CTL寄存器的OBWEN位置1。软件可以将FMC_CTL的OBWEN位清0来锁定FMC_CTL的OBPG位和OBER位。

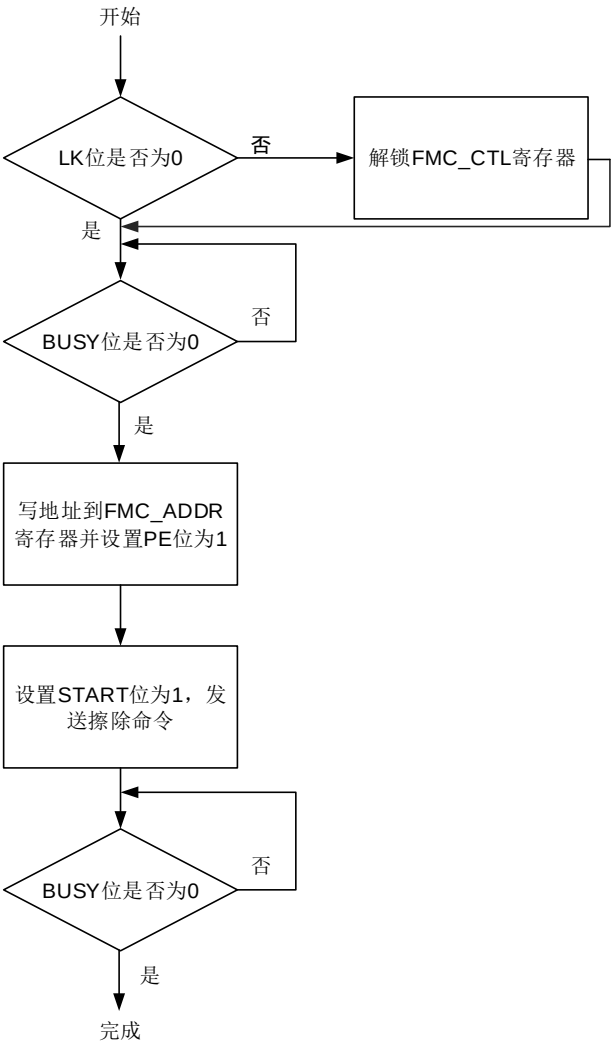
2.3.4. 页擦除

FMC的页擦除功能将一个主闪存存储页的内容初始化为高电平。每一页都可以被独立擦除，不影响其他页的内容。FMC页擦除的操作步骤如下：

- 确保FMC_CTL寄存器不处于锁定状态；
- 检查FMC_STAT寄存器的BUSY位来确保闪存存储器没有正在进行中的操作，即BUSY位为0。否则等待该操作完成；
- 写页地址到FMC_ADDR寄存器；
- 写页擦除命令到FMC_CTL寄存器的PER位（置1）；
- 通过将FMC_CTL寄存器的START位置1来发送页擦除命令到FMC；
- 通过检查FMC_STAT寄存器的BUSY位来判断擦除指令是否执行完毕，若未完成则需等待BUSY位为0；
- 如果有需要，可以使用DBUS读和验证该页的内容。

当页擦除成功执行，FMC_STAT寄存器的ENDF位将会被置1，并且如果FMC_CTL寄存器的ENDIE位之前已经被置1，那么FMC将触发一个中断。需要注意的是，一定要确保目标页地址的正确性。否则当错误的目标擦除页被用来取指令或访问数据时，软件可能失去控制。另一方面，在擦除/编程保护的页进行页擦除操作将会无效。如果FMC_CTL寄存器的ERRIE位被置1，FMC将触发一次闪存操作错误中断。软件可以检查FMC_STAT寄存器的PGERR位来监测中断处理器的状况。FMC_STAT寄存器的ENDF位指示该操作的结束。下图显示了页擦除操作的流程：

图 2-1. 页擦除操作流程



2.3.5. 整片擦除

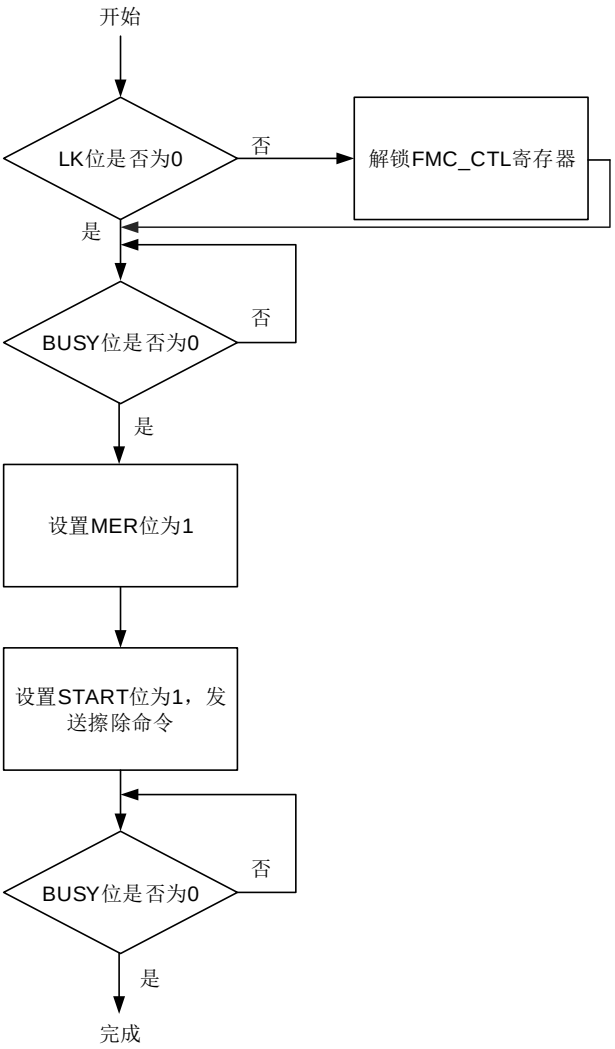
FMC提供了整片擦除功能可以初始化主闪存块的内容。整片擦除操作的寄存器设置具体步骤如下：

- 确保FMC_CTL寄存器不处于锁定状态；
- 检查FMC_STAT寄存器的BUSY位来确保闪存存储器没有正在进行中的操作，即BUSY位为0。否则等待该操作完成；
- 写整片擦除命令到FMC_CTL寄存器的MER位（置1）；
- 通过将FMC_CTL寄存器的START位置1来发送整片擦除命令到FMC；
- 通过检查FMC_STAT寄存器的BUSY位来判断擦除指令是否执行完毕，若未完成则需等待BUSY位为0；
- 如果有需要，可以使用DBUS读和验证该闪存的内容。

当整片擦除成功执行，FMC_STAT寄存器的ENDF位将会被置1，并且如果FMC_CTL寄存器的ENDIE位之前已经被置1，那么FMC将触发一个中断。由于所有的闪存数据都将被复位为0xFFFF FFFF，可以通过运行在SRAM中的程序或使用调试工具直接访问FMC寄存器来实现

整片擦除操作。FMC_STAT寄存器的ENDF位指示该操作的结束。（编程操作的起始地址需要是0x0800 0000）。下图显示了整片擦除操作的流程：

图 2-2. 整片擦除操作流程



2.3.6. 主存储闪存块编程

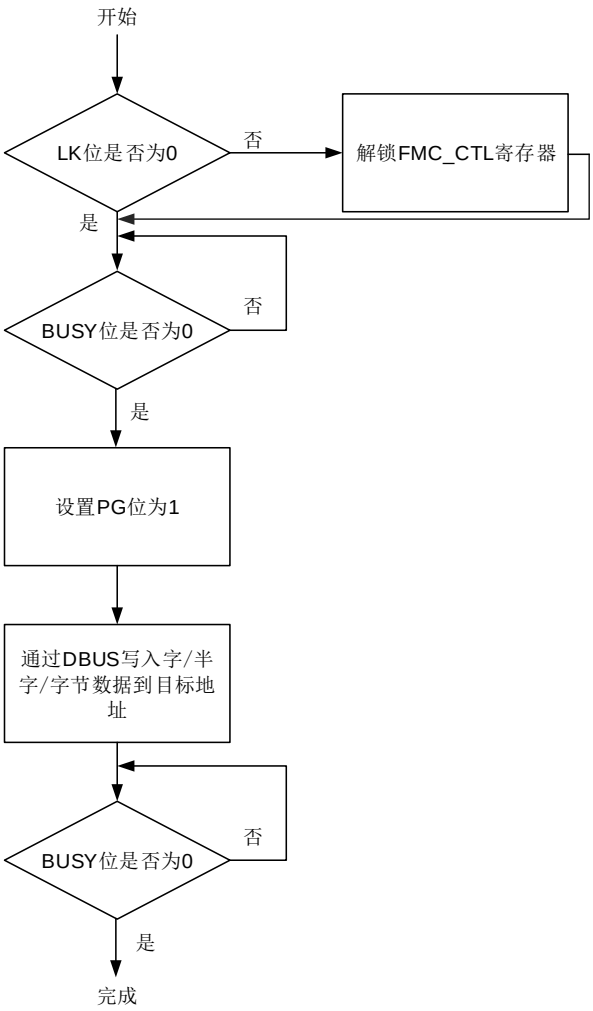
FMC提供了一个32位整字/16位半字编程功能，用来修改主闪存存储器的内容。以下步骤显示了字编程操作寄存器的过程：

- 确保FMC_CTL寄存器不处于锁定状态；
- 检查FMC_STAT寄存器的BUSY位来确保闪存存储器没有正在进行中的操作，即BUSY位为0。否则等待该操作完成；
- 写字编程命令到FMC_CTL寄存器的PG位；
- DBUS写一个32位字/16位半字到预期地址；
- 通过检查FMC_STAT寄存器的BUSY位来判断擦除指令是否执行完毕，若未完成则需等待BUSY位为0；
- 如果有需要，可以使用DBUS读并验证该闪存存储器是否编程成功。

当主存储块编程成功执行，FMC_STAT 寄存器的 ENDF 位将会被置 1，并且如果 FMC_CTL

寄存器的 ENDIE 位之前已经被置 1，那么 FMC 将触发一个中断。需要注意的是在执行字/半字编程操作之前需要检查目的地址是否已经被擦除过，如果没有被擦除过，PGERR 位将被置 1，该页上的编程操作无效（编程内容为 0x0 的情况除外）。另一方面，在擦除/编程保护的页进行编程操作将会无效。如果 FMC_CTL 寄存器的 ERRIE 位被置 1，FMC 将触发一次闪存操作错误中断。软件可以检查 FMC_STAT 寄存器的 PGERR 位来监测中断处理器的状况。FMC_STAT 寄存器的 ENDF 位指示操作的结束。下图显示了主存储块字编程操作的流程：

图 2-3. 字编程操作流程



2.3.7. 选项字节擦除

FMC提供了一个擦除功能用来初始化闪存中的选项字节块。下面的步骤显示了选项字节块的擦除过程：

- 确保FMC_CTL寄存器不处于锁定状态；
- 确保FMC_CTL寄存器的OBWEN位处于使能状态；
- 检查FMC_STAT寄存器的BUSY位来确保闪存存储器没有正在进行中的操作，即BUSY位为0。否则等待该操作完成；
- 写选项字节擦除命令到FMC_CTL寄存器的OBER位（置1）；
- 通过将FMC_CTL寄存器的START位置1来发送选项字节擦除命令到FMC；

- 通过检查FMC_STAT寄存器的BUSY位来判断擦除指令是否执行完毕，若未完成则需等待BUSY位为0；
- 如果有需要，可以使用DBUS读并验证是否擦除成功。

当选项字节擦除成功执行，FMC_STAT寄存器的ENDF位将会被置1，并且如果FMC_CTL寄存器的ENDIE位之前已经被置1，那么FMC将触发一个中断。FMC_STAT寄存器的ENDF位置1指示该操作的结束。

2.3.8. 选项字节编程

FMC提供了一个32位字/16位半字编程功能，用来修改选项字节块的内容。下面的步骤显示了选项字节编程的操作过程：

- 确保FMC_CTL寄存器不处于锁定状态；
- 确保FMC_CTL寄存器的OBWEN位处于使能状态；
- 检查FMC_STAT寄存器的BUSY位来确保闪存存储器没有正在进行中的操作，即BUSY位为0。否则等待该操作完成；
- 写选项字节编程命令到FMC_CTL寄存器的OBPG位；
- DBUS写一个32位字/16位半字到预期地址；
- 通过检查FMC_STAT寄存器的BUSY位来判断擦除指令是否执行完毕，若未完成则需等待BUSY位为0；
- 如果有需要，可以使用DBUS读并验证是否编程成功。

当选项字节编程操作成功执行，FMC_STAT寄存器的ENDF位将会被置1，并且如果FMC_CTL寄存器的ENDIE位之前已经被置1，那么FMC将触发一个中断。需要注意的是在执行字/半字编程操作之前需要检查目的地址是否已经被擦除过，如果没有被擦除过，PGERR位将被置1，该页上的编程操作无效（编程内容为0x0的情况除外）。FMC_STAT寄存器的ENDF位指示操作的结束。

2.3.9. 选项字节说明

每次系统复位或将 FMC_CTL 寄存器的 OBRLD 位置 1 时，闪存存储器的选项字节块会被重加载到 FMC_OBSTAT 和 FMC_WP 寄存器，之后选项字节生效。选项字节的补码和选项字节相反。当选项字节被重加载，如果选项字节的补码和选项字节不匹配，FMC_OBSTAT 寄存器的 OBERR 位将被置 1，选项字节将被设置为 0xFF。选项字节详情见下表：

表 2-2. 选项字节

地址	名称	说明
0x1fff f800	OB_SPC	选项字节安全保护码 0xA5：无保护 除0xA5或0xCC之外的任意值：低级别保护 0xCC：高级别保护
0x1fff f801	OB_SPC_N	OB_SPC补码值
0x1fff f802	OB_USER	用户定义的选项字节 [7]：保留 [6]：SRAM_PARITY_CHECK

地址	名称	说明
		0：使能SRAM奇偶校验 1：失能SRAM奇偶校验 [5]：VDDA_VISOR 0：失能V_{DDA}监视器 1：使能V_{DDA}监视器 [4]：BOOT1_n 0：BOOT1位是1 1：BOOT1位是0 [3]：保留 [2]：nRST_STDBY 0：设置待机模式时产生复位而不是进入待机模式 1：设置待机模式时进入待机模式而不是产生复位 [1]：nRST_DPSLP 0：设置深度睡眠模式时产生复位而不进入深度睡眠模式 1：设置深度睡眠模式时进入深度睡眠模式而不产生复位 [0]：nWDG_SW 0：硬件自动设置独立看门狗定时器 1：软件设置独立看门狗定时器
0x1fff f803	OB_USER_N	OB_USER补码值
0x1fff f804	OB_DATA[7:0]	用户定义数据位7到0位
0x1fff f805	OB_DATA_N[7:0]	OB_DATA补码值的7到0位
0x1fff f806	OB_DATA[15:8]	用户定义数据位15到8位
0x1fff f807	OB_DATA_N[15:8]	OB_DATA补码值的15到8位
0x1fff f808	OB_WP[7:0]	页擦除/编程保护位的7到0位
0x1fff f809	OB_WP_N[7:0]	OB_WP补码值的7到0位
0x1fff f80a	OB_WP[15:8]	页擦除/编程保护位的15到8位
0x1fff f80b	OB_WP_N[15:8]	OB_WP补码值的15到8位

2.3.10. 页擦除/编程保护

FMC的页擦除/编程保护功可以阻止对闪存存储器的意外操作。当FMC对被保护的页进行页擦除或编程操作时，操作本身无效且FMC_STAT寄存器的WPERR位将被置1。如果WPERR位被置1且ERRIE位也被置1来使能相应的中断，FMC将触发闪存操作错误中断来引起CPU重视。配置选项字节的OB_WP[15:0]位为0可以单独使能每个页的保护功能。如果在选项字节区域执行了页擦除操作，所有的闪存存储器页保护功能都将被禁止。当对选项字节的OB_WP位置1或清0时，需要将FMC_CTL寄存器的OBRLD位置1或系统复位来重加载OB_WP位。下表显示了通过设置OB_WP[15:0]寄存器可以保护哪些页：

表 2-3. OB_WP 位对应页保护

OB_WP位	页保护
OB_WP[0]	页0 ~ 页3
OB_WP[1]	页4 ~ 页7
OB_WP[2]	页8 ~ 页11

OB_WP位	页保护
. . .	. . .
OB_WP[14]	页56 ~ 页59
OB_WP[15]	页60 ~ 页63

2.3.11. 安全保护

FMC提供了一个安全保护功能来阻止闪存上非法的代码或数据访问。此功能可以很好地保护软件和固件免受非法用户的操作。安全保护等级划分为以下三等：

无保护状态：当将OB_SPC字节和它的补码值设置为0x5AA5，则不执行保护。主存储块和选项字节块可以被所有操作访问。

保护等级低：当设置OB_SPC字节和它的补码值为除了0x5AA5或0x33CC之外的任意值，则执行低级别保护。主闪存仅仅能被用户代码访问。在调试模式下，从SRAM或boot loader模式启动时，所有对主存储块的操作都被禁止，并且如果是读操作将会产生一个总线错误，如果是编程或擦除操作将会导致FMC_STAT寄存器的PGERR位被置1。在低级别保护状态下，选项字节块可以被任意操作访问。如果将OB_SPC字节和它的补码值设置为0x5AA5，进入无保护级别，那么将执行一次主存储块整片擦除操作。

保护等级高：当设置OB_SPC字节和它的补码值为0x33CC，则执行高级别保护。在调试模式下，从SRAM或boot loader模式启动都将被禁止。主闪存块可由用户代码的所有操作进行访问。选项字节不能被擦除，OB_SPC字节和它的补码值不能被重编程。因此，如果设置保护级别为高，不能转换为低级别保护和无保护。

2.4. FMC 寄存器

FMC 基地址：0x4002 2000

2.4.1. 等待状态寄存器 (FMC_WS)

地址偏移：0x00
复位值：0x0000 0000

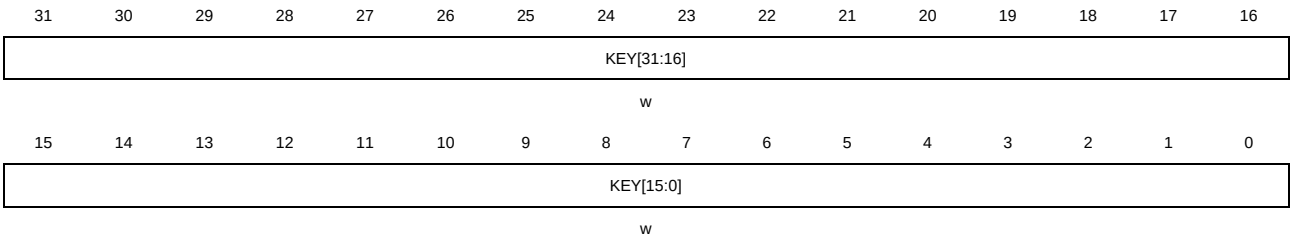
该寄存器只能按字(32位)访问。

位/位域	名称	描述
31:3	保留	必须保持复位值。
2:0	WSCNT[2:0]	等待状态计数寄存器。 硬件置 1 和清 0。WSEN 位被置 1 时 WSCNT 位有效。 000：不增加等待状态 001：增加 1 个等待状态 010：增加 2 个等待状态 011 ~ 111：保留

2.4.2. 解锁寄存器 (FMC_KEY)

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字(32位)访问。



位/位域	名称	描述
31:0	KEY[31:0]	FMC_CTL 解锁寄存器 这些位仅能被软件写。

写解锁值到 KEY[31:0]来解锁 FMC_CTL 寄存器。

2.4.3. 选项字节解锁寄存器 (FMC_OBKEY)

地址偏移：0x08
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OBKEY[31:16]															
w															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OBKEY[15:0]															
w															

位/位域	名称	描述
31:0	OBKEY[31:0]	FMC_CTL 选项字节解锁寄存器 这些位仅能被软件写 写解锁值到 OBKEY[31:0]来解锁 FMC_CTL 寄存器的选项字节命令

2.4.4. 状态寄存器 (FMC_STAT)

地址偏移：0x0C
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留										ENDF	WPERR	保留	PGERR	保留	BUSY
										rc_w1	rc_w1		rc_w1		r

位/位域	名称	描述
31:6	保留	必须保持复位值。
5	ENDF	操作结束标志位 操作成功执行后，此位被硬件置 1。软件可以通过写 1 来清 0 该位。
4	WPERR	擦除/编程保护错误标志位 在受保护的页上进行擦除/编程操作时，此位会被硬件置 1。软件可以通过写 1 来清 0 该位。
3	保留	必须保持复位值。

2	PGERR	编程错误标志位 当在值不为 0xFFFF 的闪存上进行编程时，此位会被硬件置 1。软件可以通过写 1 来清 0 该位。
1	保留	必须保持复位值。
0	BUSY	闪存忙标志位 当有闪存操作正在进行时，此位被置 1。当操作结束或者出错时，此位被清 0。

2.4.5. 控制寄存器 (FMC_CTL)

地址偏移：0x10
复位值：0x0000 0080

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		OBRLD	ENDIE	保留	ERRIE	OBWEN	保留	LK	START	OBER	OBPG	保留	MER	PER	PG
		rw	rw		rw	rw		rw	rw	rw	rw		rw	rw	rw

位/位域	名称	描述
31:14	保留	必须保持复位值。
13	OBRLD	选项字节重加载位 软件置 1。 0：没有作用 1：强制选项字节重装载，并产生一次系统复位
12	ENDIE	操作结束中断使能位 软件置 1 或清 0。 0：无硬件中断产生 1：使能操作结束中断
11	保留	必须保持复位值。
10	ERRIE	错误中断使能位 软件置 1 或清 0。 0：无硬件中断产生 1：使能错误中断
9	OBWEN	选项字节擦除/编程使能位 当正确的序列写入 FMC_OBKEY 寄存器，此位由硬件置 1。此位可以被软件清 0。
8	保留	必须保持复位值。
7	LK	锁 FMC_CTL 寄存器标志位

		当正确的序列写入 FMC_KEY 寄存器，此位由硬件清 0。此位可以由软件置 1。
6	START	发送擦除命令到 FMC 位 软件置 1 后将发送擦除命令到 FMC。当 BUSY 位被清 0 后，此位将被硬件清 0。
5	OBER	选项字节擦除命令位 软件置 1 和清 0。 0：无作用 1：选项字节擦除命令
4	OBPG	选项字节编程命令位 软件置 1 和清 0。 0：无作用 1：选项字节编程命令
3	保留	必须保持复位值。
2	MER	主存储块整片擦除命令位 软件置 1 和清 0。 0：无作用 1：主存储块整片擦除命令
1	PER	主存储块页擦除命令位 软件置 1 和清 0。 0：无作用 1：主存储块页擦除命令
0	PG	主存储块编程命令位 软件置 1 和清 0 0：无作用 1：主存储块编程命令

2.4.6. 地址寄存器 0 (FMC_ADDR)

地址偏移：0x14
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR[15:0]															
rw															

位/位域	名称	描述
31:0	ADDR[31:0]	闪存命令地址位

软件置位

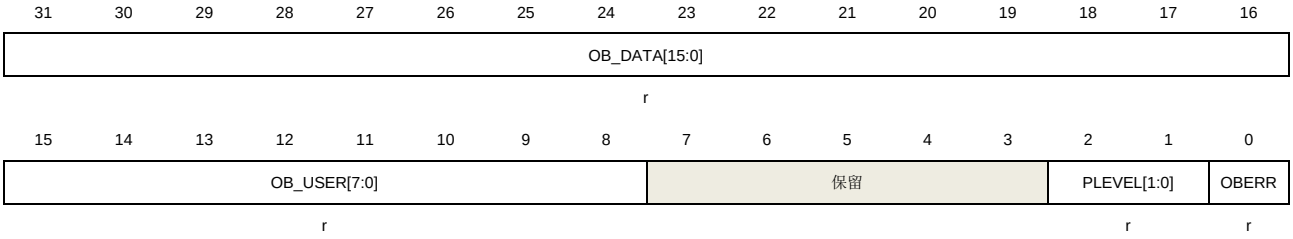
ADDR 位是闪存擦除命令的地址。

2.4.7. 选项字节状态寄存器 (FMC_OBSTAT)

地址偏移：0x1C

复位值：0xXXXX XX0X

该寄存器只能按字(32位)访问。



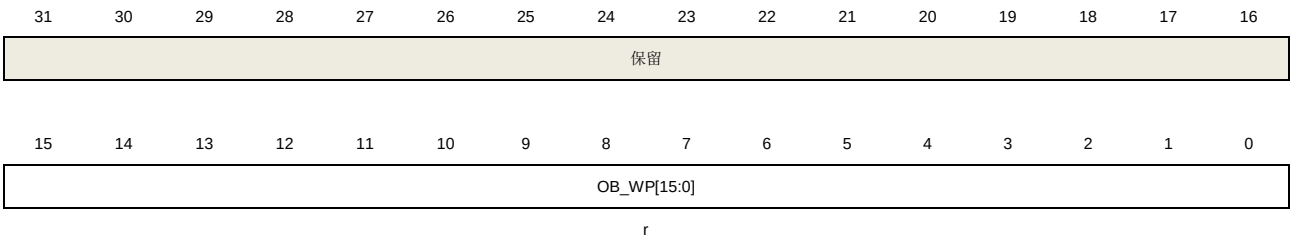
位/位域	名称	描述
31:16	OB_DATA[15:0]	系统复位后保存选项字节块的 OB_DATA[15:0]部分
15:8	OB_USER[7:0]	系统复位后保存选项字节块的 OB_USER 字节
7:3	保留	必须保持复位值。
2:1	PLEVEL[1:0]	安全保护级别 00：无保护 01：低保护级别 11：高保护级别
0	OBERR	选项字节读错误位 当选项字节和它的补码不匹配时此位由硬件置 1，并且选项字节被设置为 0xFF。

2.4.8. 写保护寄存器 (FMC_WP)

地址偏移：0x20

复位值：0x0000 XXXX

该寄存器只能按字(32位)访问。



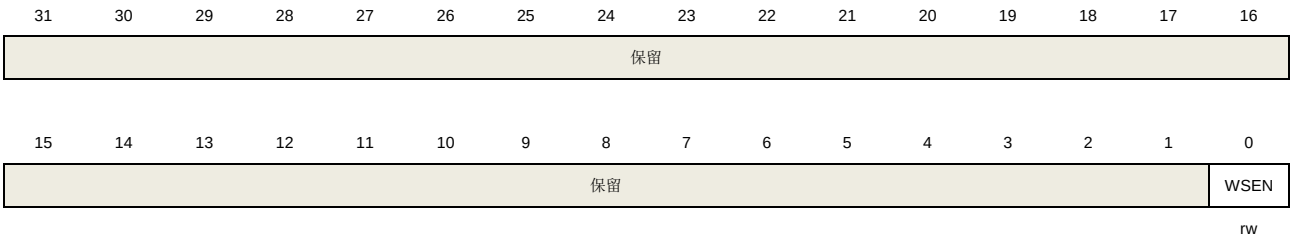
位/位域	名称	描述

31:16	保留	必须保持复位值。
15:0	OB_WP[15:0]	系统复位后保存选项字节块的 OB_WP[15:0]部分 0：保护生效 1：未保护

2.4.9. 等待状态使能寄存器 (FMC_WSEN)

地址偏移：0xFC
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

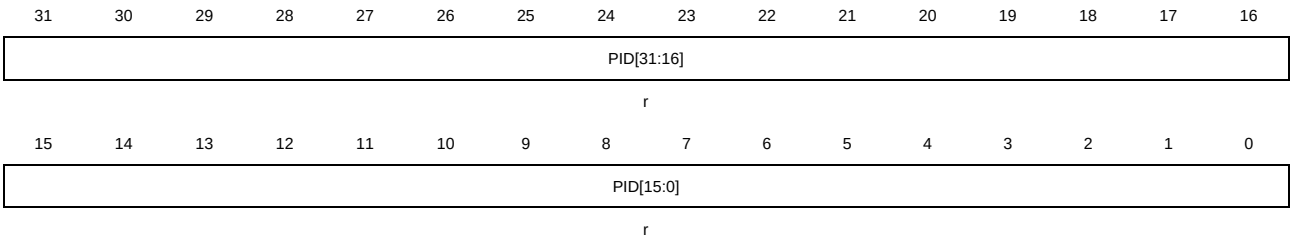


位/位域	名称	描述
31:1	保留	必须保持复位值。
0	WSEN	FMC等待状态使能寄存器。 此位由软件置1和清0。此位也被FMC_KEY寄存器保护。软件需要写0x45670123和0xCDEF89AB到FMC_KEY寄存器。 0：从闪存取指无等待状态增加。 1：从闪存取指增加等待状态。

2.4.10. 产品 ID 寄存器 (FMC_PID)

地址偏移：0x100
复位值：0xXXXX XXXX

该寄存器只能按字(32位)访问。



位/位域	名称	描述
31:0	PID1[31:0]	产品保留 ID 寄存器 该寄存器为只读

上电后这些位始终不会改变，该寄存器在生产过程中被一次性编程。

3. 电源管理单元（PMU）

3.1. 简介

功耗设计是GD32F1x0系列产品比较注重的问题之一。电源管理单元提供了三种省电模式，包括睡眠模式，深度睡眠模式和待机模式。这些模式能减少电源能耗，且使得应用程序可以在CPU运行时间要求、速度和功耗的相互冲突中获得最佳折衷。如*图3-1. 电源域概览*所示设备有三个电源域，包括V_{DD} / V_{DDA}域、1.2V域和备份域。V_{DD} / V_{DDA}域由电源直接供电。在V_{DD} / V_{DDA}域中嵌入了一个LDO，用来产生1.2V为1.2V域供电。在备份域中有一个电源切换器，当V_{DD}电源关闭时，电源切换器可以将备份域的电源切换到V_{BAT}引脚，此时备份域由V_{BAT}引脚（电池）供电。

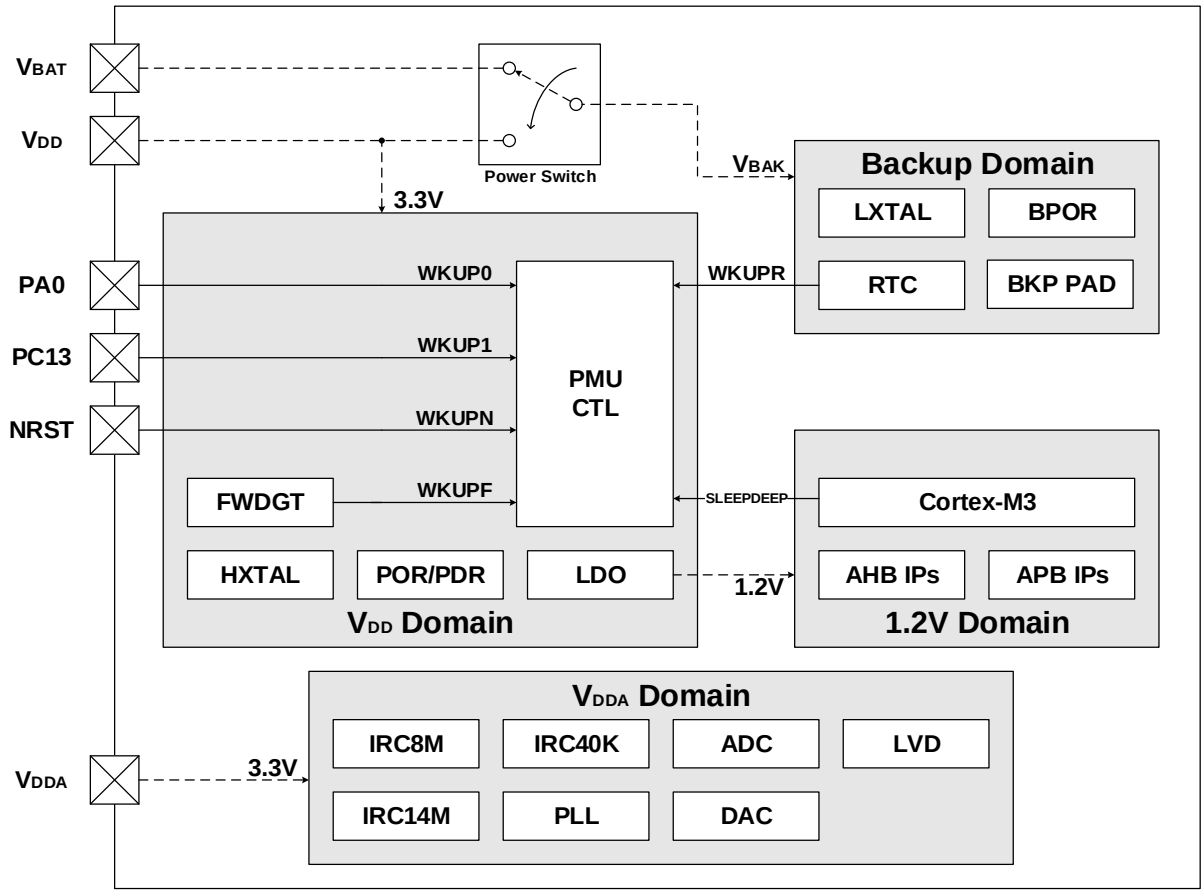
3.2. 主要特性

- 三个电源域：备份域、V_{DD} / V_{DDA}和1.2V电源域；
- 三种省电模式：睡眠模式、深度睡眠模式和待机模式；
- 内部调压器（LDO）提供1.2V电源；
- 提供低电压检测器（LVD），当电压低于所设定的阈值时能发出中断或事件；
- 当V_{DD}关闭的时，由V_{BAT}（电池）为备份域供电。

3.3. 功能描述

*图3-1. 电源域概览*提供了PMU及相关电源域的结构框图。

图 3-1.电源域概览



LVD：低压检测器　　LDO：电压调节器　　BPOR：备份域上电复位

POR：上电复位　　PDR：掉电复位

3.3.1. 电池备份域

电池备份域由内部电源切换器来选择V_{DD}供电或V_{BAT}（电池）供电，然后由V_{BAK}为备份域供电，该备份域包含RTC（实时时钟）、LXTAL（低速外部晶体振荡器）、BPOR（备份域上电复位），以及PC13至PC15共3个BKP PAD。为了确保备份域中寄存器的内容及RTC正常工作，当V_{DD}关闭时，V_{BAT}引脚可以连接至电池或其他电源等备份源供电。电源切换器是由V_{DD}/V_{DDA}域掉电复位电路控制的。对于没有外部电池的应用，建议将V_{BAT}引脚通过100nF的外部陶瓷去耦电容连接到V_{DD}引脚上。

备份域的复位源包括备份域上电复位和备份域软件复位。在V_{BAK}没有完全上电时，BPOR信号强制设备处于复位状态。应用软件可以通过设置RCU_BDCTL寄存器BKPRST位来触发备份域软件复位。

RTC的时钟源可以是内部40KHz的RC振荡器（IRC40K）或低速晶体振荡器（LXTAL），或高速晶体振荡器（HXTAL）时钟32分频。当V_{DD}被关闭时，RTC只能选择LXTAL作时钟源。在通过WFI/WFE指令进入省电模式之前，Cortex®-M3需要通过RTC寄存器设置预期的闹钟时间并启用闹钟功能，以实现RTC定时器唤醒事件。进入省电模式一定时间之后，当经过的时间与预设的唤醒时间匹配时，RTC将唤醒设备。RTC的配置和操作的细节将在*实时时钟（RTC）*章节描述。

当备份域由V_{DD}电源供电时（V_{BAK}连接至V_{DD}），以下功能可用：

- PC13可以作为通用I/O口或RTC功能引脚（参见*实时时钟（RTC）*）；
- PC14和PC15可以作为通用I/O口或LXTAL晶振引脚。

当备份域由V_{BAT}电源供电时（V_{BAK}连接至V_{BAT}），以下功能可用：

- PC13仅可以作为RTC功能引脚（参见*实时时钟（RTC）*）；
- PC14和PC15仅可作为LXTAL晶振引脚。

注意：由于PC13至PC15是通过电源切换器供电的，电源切换器仅可通过小电流，因此当PC13至PC15的GPIO口在输出模式时，其工作的速度不能超过2MHz（最大负载为30pF）。

3.3.2. V_{DD} / V_{DDA} 电源域

V_{DD} / V_{DDA}域包括V_{DD}域和V_{DDA}域两部分。V_{DD}域包括HXTAL（高速外部晶体振荡器）、LDO（电压调节器）、POR / PDR（上电/掉电复位）、FWDGT（独立看门狗定时器）和除PC13、PC14和PC15之外的所有PAD等等。V_{DDA}域包括ADC / DAC（AD / DA转换器）、IRC8M（内部8M RC振荡器）、IRC14M（内部14M RC振荡器）、IRC40K（内部40KHz RC振荡器）、PLLs（锁相环）和LVD（低电压检测器）等等。

V_{DD} 域

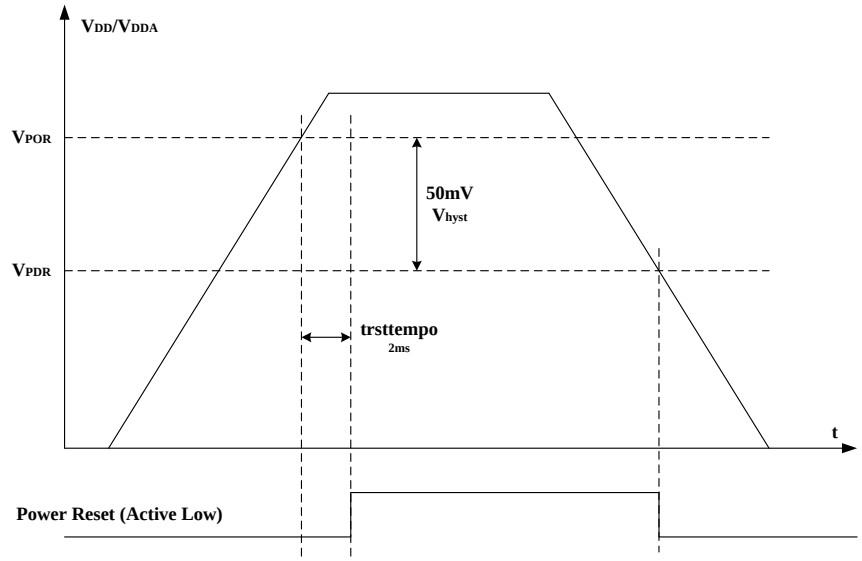
LDO用来给1.2V域供电，其复位后保持使能。可以被配置为三种不同的工作状态：包括睡眠模式（全供电状态）、深度睡眠模式（全供电或低功耗状态）和待机模式（关闭状态）。

POR/PDR（上电/掉电复位）电路检测 V_{DD} / V_{DDA} 并在电压低于特定阈值时产生电源复位信号复位除备份域之外的整个芯片。POR 电路只检测 V_{DD} 供电电压，PDR 电路检测 V_{DD} 和 V_{DDA} 供电电压。

注意：如果用户能确保V_{DDA}电压大于等于V_{DD}，可以通过清零选项字节寄存器OB_USER的位VDDA_VISOR，关闭由PDR电路实现的V_{DDA}监视器以减小功耗。

*图3-2. 上电复位/掉电复位波形图*显示了供电电压和电源复位信号之间的关系。V_{POR}表示上电复位的阈值电压，V_{PDR}表示掉电复位的阈值电压。V_{PDR}是可配置的，通过RCU_PDVSEL寄存器的PDVSEL位可以从两个值中选择一个作为V_{PDR}(参见4.3.20章节*RCU寄存器*)。当选择较低的值时，强烈建议不要对片上闪存做编程或擦除操作，因为这些操作在电压低至接近该阈值时可能会失败。迟滞电压V_{hyst}值约为50mV。

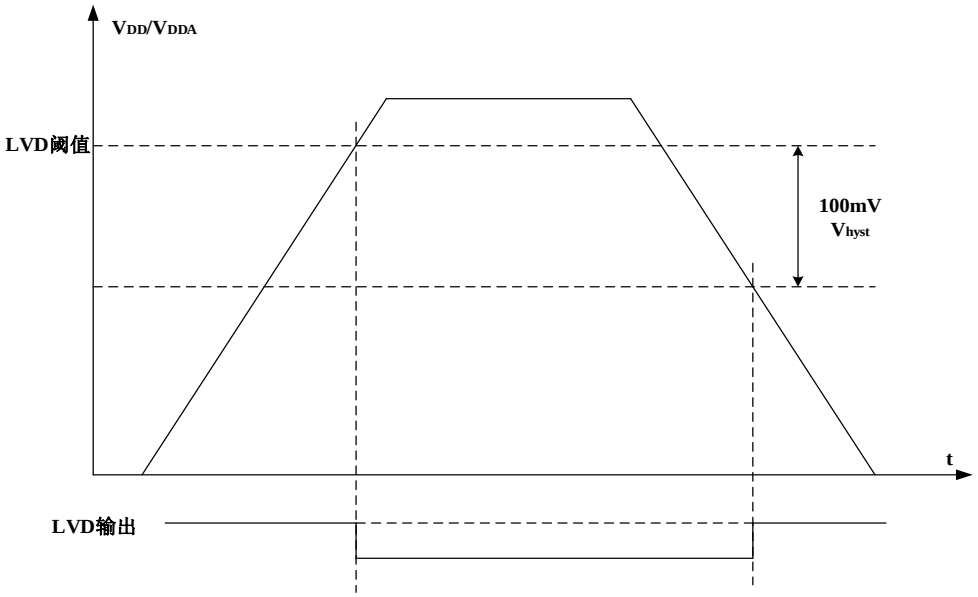
图 3-2. 上电复位/掉电复位波形图



V_{DDA} 域

LVD的功能是检测V_{DD} / V_{DDA}供电电压是否低于低电压检测阈值该阈值由电源控制寄存器（PMU_CTL）中的LVDT[2:0]位进行配置。LVD通过LVDEN置位使能，位于电源状态寄存器（PMU_CS）中的LVDF位表示低电压事件是否出现，该事件连接至EXTI的第16线，用户可以通过配置EXTI的第16线产生相应的中断。*图3-3. LVD阈值波形图*显示了V_{DD}/V_{DDA}供电电压和LVD输出信号的关系。（LVD中断信号依赖于EXTI第16线的上升或下降沿配置）。迟滞电压V_{hyst}值为100mV。

图3-3. LVD阈值波形图



一般来说，数字电路由 V_{DD} 供电，而大多数的模拟电路由 V_{DDA} 供电。为了提高 ADC 和 DAC 的转换精度，为 V_{DDA} 独立供电可使模拟电路达到更好的特性。为避免噪声，V_{DDA} 通过外部滤波电路连接至 V_{DD}，相应的 V_{SSA} 通过特定电路连接至 V_{SS}。否则，当 V_{DD}/V_{DDA} 不是同一个电源（差值小于 0.3V）提供时，在上电过程中 V_{DDA} 必须大于等于 V_{DD}。

3.3.3. 1.2V 电源域

主要功能包括Cortex®-M3内核逻辑、AHB / APB外设、备份域和V_{DD} / V_{DDA}域的APB接口等。当1.2V电压上电后，POR将在1.2V域中产生一个复位序列，复位完成后，要进入指定的省电模式，须先配置相关的控制位，之后一旦执行WFI或WFE指令，设备便进入该省电模式。关于这方面的详细内容，将在以下章节予以说明。

3.3.4. 省电模式

系统复位或电源复位后，GD32F1x0处于全功能状态且电源域全部处于供电状态。实现较低的功耗的方法有两种：减慢系统时钟（HCLK，PCLK1，PCLK2），关闭未使用的外设的时钟。此外，三种省电模式可以实现更低的功耗，它们是睡眠模式、深度睡眠模式和待机模式。

睡眠模式

睡眠模式与Cortex®-M3的休眠模式相对应。休眠模式只关闭Cortex®-M3的时钟。如需进入睡眠模式，只要清除Cortex®-M3系统控制寄存器中的SLEEPDEEP位，并执行一条WFI或WFE指令即可。如果睡眠模式是通过执行WFI指令进入的，任何中断都可以唤醒系统。如果睡眠模式是通过执行WFE指令进入的，任何唤醒事件都可以唤醒系统（如果SEVONPEND为1，任何中断都可以唤醒系统，请参考Cortex®-M3技术手册）。由于无需在进入或退出中断上浪费时间，该模式所需的唤醒时间最短。

根据Cortex®-M3中SCR（系统控制寄存器）的SLEEPONEXIT位，有两种睡眠进入机制可选：

- Sleep-now：如果SLEEPONEXIT位被清零，一旦执行WFI或WFE指令，MCU立即进入睡

眠模式。

- Sleep-on-exit：如果SLEEPONEXIT位被置位，当系统从最低优先级的中断处理程序离开后，MCU立即进入睡眠模式。

深度睡眠模式

深度睡眠模式与Cortex®-M3的SLEEPDEEP模式相对应。深度睡眠模式会关闭1.2V域中的所有时钟以及包括IRC8M、IRC14M、HXTAL和PLLs也全部被禁用。PMU_CTL寄存器的LDOLP位可以控制LDO工作在正常模式或低功耗模式。进入深度睡眠模式之前，先将Cortex®-M3系统控制寄存器的SLEEPDEEP位置1，再清除PMU_CTL寄存器的STBMOD位，然后执行WFI或WFE指令即可进入深度睡眠模式。如果睡眠模式是通过执行WFI指令进入的，任何来自EXTI的中断可以将系统从深度睡眠模式中唤醒。如果睡眠模式是通过执行WFE指令进入的，任何来自EXTI的任何中断或唤醒事件可以将系统从深度睡眠模式中唤醒（如果SEVONPEND为1，任何中断都可以唤醒系统，请参考Cortex®-M3技术手册）。刚退出深度睡眠模式时，IRC8M被选中作为系统时钟。请注意，如果LDO工作在低功耗模式，那么唤醒时需额外的延时时间。

注意：为了顺利进入深度睡眠模式，所有 EXTI 线上的挂起状态（在 EXTI_PD 寄存器中）和相关外设标志位必须被复位，参考*表 5-3. EXTI 触发源*。否则，程序将直接跳过深度睡眠模式进入过程而继续执行下面的程序。

待机模式

待机模式是基于Cortex®-M3的SLEEPDEEP模式实现的。待机模式会关闭整个1.2V域的供电，同时LDO和包括IRC8M、IRC14M、HXTAL和PLLs也会被关闭。进入待机模式前，先将Cortex®-M3系统控制寄存器的SLEEPDEEP位置1，再将PMU_CTL寄存器的STBMOD位置1，再清除PMU_CS寄存器的WUF位，然后执行WFI或WFE指令，系统进入待机模式，PMU_CS寄存器的STBF位状态表示MCU是否曾进入过待机模式。待机模式有五个唤醒源，包括来自NRST引脚的外部复位、RTC闹钟/时间戳/侵入事件、FWDGT复位、WKUP0或WKUP1引脚的上升沿。待机模式可以达到最低的功耗，但唤醒时间最长。另外，一旦进入待机模式，SRAM和1.2V电源域寄存器的内容都会丢失。退出待机模式时，会发生上电复位，复位之后Cortex®-M3将从0x00000000地址开始执行指令代码。

表 3-1. 节电模式总结

模式	睡眠	深度睡眠	待机
描述	仅关闭 CPU 时钟	1. 关闭 1.2V 电源域的所有时钟 2. 关闭 IRC8M、IRC14MHXTAL 和 PLL	1. 关闭 1.2V 电源域的供电 2. 关闭 IRC8M、IRC14MHXTAL 和 PLL
LDO 状态	开启（正常功耗模式）	开启（正常功耗模式或低功耗模式）	关闭
配置	SLEEPDEEP = 0	SLEEPDEEP = 1 STBMOD = 0	SLEEPDEEP = 1 STBMOD = 1, WURST=1
进入指令	WFI 或 WFE	WFI 或 WFE	WFI 或 WFE
唤醒	若通过 WFI 进入，则任何中断均可唤醒；	若通过 WFI 进入，则任何中断均可唤醒；	1. NRST 引脚 2. RTC

模式	睡眠	深度睡眠	待机
	若通过 WFE 进入，则任何事件（或 SEVONPEND = 1 时的中断）均可唤醒	若通过 WFE 进入，来自 EXTI 的任何事件（或 SEVONPEND = 1 时的中断）均可唤醒	3. FWDGT 复位 4. WKUP0 引脚 5. WKUP1 引脚
唤醒延迟	无	IRC8M 唤醒时间 如果 LDO 处于低功耗模式的话，需增加 LDO 唤醒时间	上电序列

注意：在待机模式下，除了 NRST 引脚，配置为 RTC 功能的 PC13，用作 LXTAL 晶振引脚的 PC14 和 PC15，使能的 WKUP 引脚，其他所有 I / O 都处于高阻态。

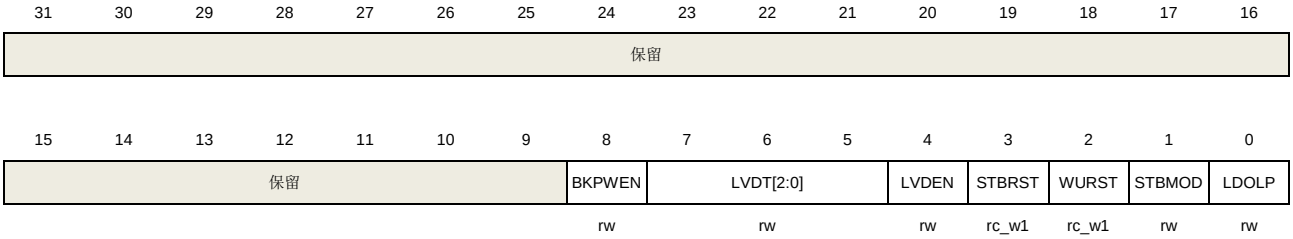
3.4. PMU 寄存器

PMU 基地址：0x4000 7000

3.4.1. 控制寄存器（PMU_CTL）

地址偏移：0x00
复位值：0x0000 0000（从待机模式唤醒后复位）

该寄存器可以按半字（16位）或字（32位）访问。



位/位域	名称	描述
31:9	保留	必须保持复位值。
8	BKPWEN	备份域写使能 0：禁止对备份域寄存器的写操作 1：允许对备份域寄存器的写操作 复位之后，任何对备份域寄存器的写操作都将被禁止。如需对备份域寄存器做写操作，需先将该位置1。
7:5	LVDT[2:0]	低电压检测器阈值 000：2.2V 001：2.3V 010：2.4V 011：2.5V 100：2.6V 101：2.7V 110：2.8V 111：2.9V
4	LVDEN	低电压检测器使能 0：关闭低电压检测器 1：开启低电压检测器 **注意：**当SYSCFG_CFG2寄存器里的LVD_LOCK位被置1时，LVDEN和LVDT[1:0]仅可读。
3	STBRST	待机标志复位 0：无影响 1：复位待机标志

读该位，始终返回0。

2	WURST	唤醒标志复位 0：无影响 1：复位唤醒标志 读该位，始终返回0。
1	STBMOD	待机模式 0：当Cortex®-M3进入深度睡眠模式时，系统进入深度睡眠模式 1：当Cortex®-M3进入深度睡眠模式时，系统进入待机模式
0	LDOLP	LDO低功耗模式 0：当系统进入深度睡眠模式时，LDO仍正常工作 1：当系统进入深度睡眠模式时，LDO进入低功耗模式 **注意：**在深度睡眠模式下，个别外设可能会开启IRC8M时钟来做一些工作。在这种情况下，如果LDO正处于低功耗模式，LDO会自动从低功耗模式切换到正常工作模式，并保持正常工作模式，直到外设工作完毕。

3.4.2. 电源控制和状态寄存器(PMU_CS)

地址偏移：0x04
复位值：0x0000 0000（从待机模式唤醒后不复位）

该寄存器可以按半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						WUPEN1	WUPEN0	保留					LVDF	STBF	WUF
						rw	rw						r	r	r

位/位域	名称	描述
31:10	保留	必须保持复位值。
9	WUPEN1	WKUP1引脚唤醒使能 0：关闭WKUP1引脚唤醒功能 1：开启WKUP1引脚唤醒功能 如果WUPEN1在进入待机模式之前置1，WKUP1引脚的上升沿会将系统从待机模式唤醒。由于WKUP1引脚为高电平有效，WKUP1引脚内部被配置为输入下拉模式。当置位该控制位后，当输入为高的时候，将会触发一个唤醒事件。
8	WUPEN0	WKUP0 引脚唤醒使能 0：关闭WKUP0引脚唤醒功能 1：开启WKUP0引脚唤醒功能 如果WUPEN0在进入待机模式之前置1，WKUP0引脚的上升沿会将系统从待机模式唤醒。由于WKUP0引脚为高电平有效，WKUP0引脚内部被配置为输入下拉模式。当

		置位该控制位后，当输入为高的时候，将会触发一个唤醒事件。
7:3	保留	必须保持复位值。
2	LVDF	低电压状态标志 0：低电压事件没出现（V_{DD}高于设定的LVD阈值） 1：低电压事件出现（V_{DD}等于或低于LVD阈值） **注意：**LVD功能在待机模式下被禁用。
1	STBF	待机标志 0：设备没进入过待机模式 1：设备曾进入过待机模式 该位只能由POR/PDR或通过设置PMU_CTL寄存器的STBRST位来清零。
0	WUF	唤醒标志 0：没有收到唤醒事件 1：收到来自WKUP引脚或RTC唤醒事件，包括RTC侵入事件、RTC闹钟事件、RTC时间戳事件。 该位只能由POR/PDR或通过设置PMU_CTL寄存器的STBRST位来清零。

4. 复位和时钟单元（RCU）

4.1. 复位控制单元（RCTL）

4.1.1. 简介

GD32F1x0复位控制包括三种复位控制：电源复位、系统复位和备份域复位。电源复位又称为冷复位，电源启动时复位除了备份域的所有系统。除了SW-DP控制器和备份域，系统复位将复位处理器内核和外设IP部分。备份域复位复位备份区域。复位被外部信号、内部事件和复位发生器触发。接下章节将详细介绍这些复位。

4.1.2. 功能描述

电源复位

当以下事件中之一发生时，产生电源复位：1、上电/掉电复位(POR/PDR 复位) 2、从待机模式中返回后由内部复位发生器产生。电源复位复位所有的寄存器除了备份域。电源复位为低电平有效，当内部LDO电源基准准备好提供1.2V电压时，电源复位电平将变为无效。复位入口矢量被固定在存储器映射地址0x0000_0004。

系统复位

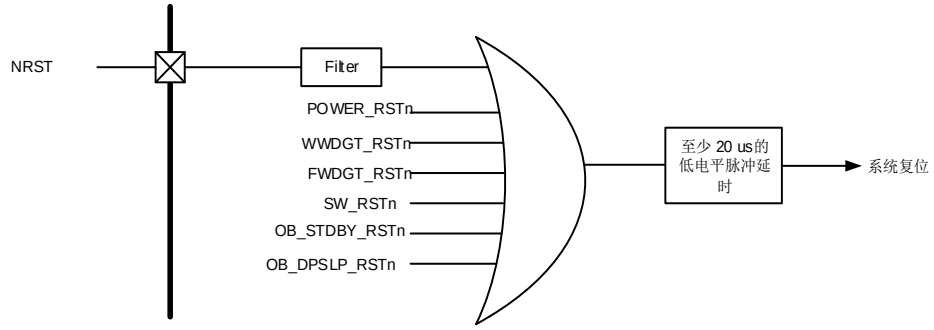
当发生以下任一事件时，产生一个系统复位：

- 电源复位（POWER_RSTn）；
- 外部引脚复位（NRST）；
- 窗口看门狗定时器计数终止（WWDGT_RSTn）；
- 独立看门狗定时器计数终止（FWDGT_RSTn）；
- Cortex®-M3的中断应用和复位控制寄存器中的SYSRESETREQ位置‘1’（SW_RSTn）；
- 可选负载字节复位（OBL_RSTn）；
- 用户选择字节寄存器nRST_STDBY设置为0，并且进入待机模式时（OB_STDBY_RSTn）；
- 用户选择字节寄存器nRST_DPSLP设置为0，并且进入深度睡眠模式时将产生复位（OB_DPSLP_RSTn）。

除了SW-DP控制器和备份域，系统复位将复位处理器内核和外设IP部分。

系统复位脉冲发生器保证每一个复位源（外部或内部）都能有至少20μs的低电平脉冲延时。

图 4-1. 系统复位电路



备份域复位

当以下事件之一发生时，产生备份域复位：

1、设置备份域控制寄存器中的BKPRST位为‘1’；

2、备份域电源上电复位（在V_{DD}和V_{BAT}两者掉电的前提下，V_{DD}或V_{BAT}上电）。

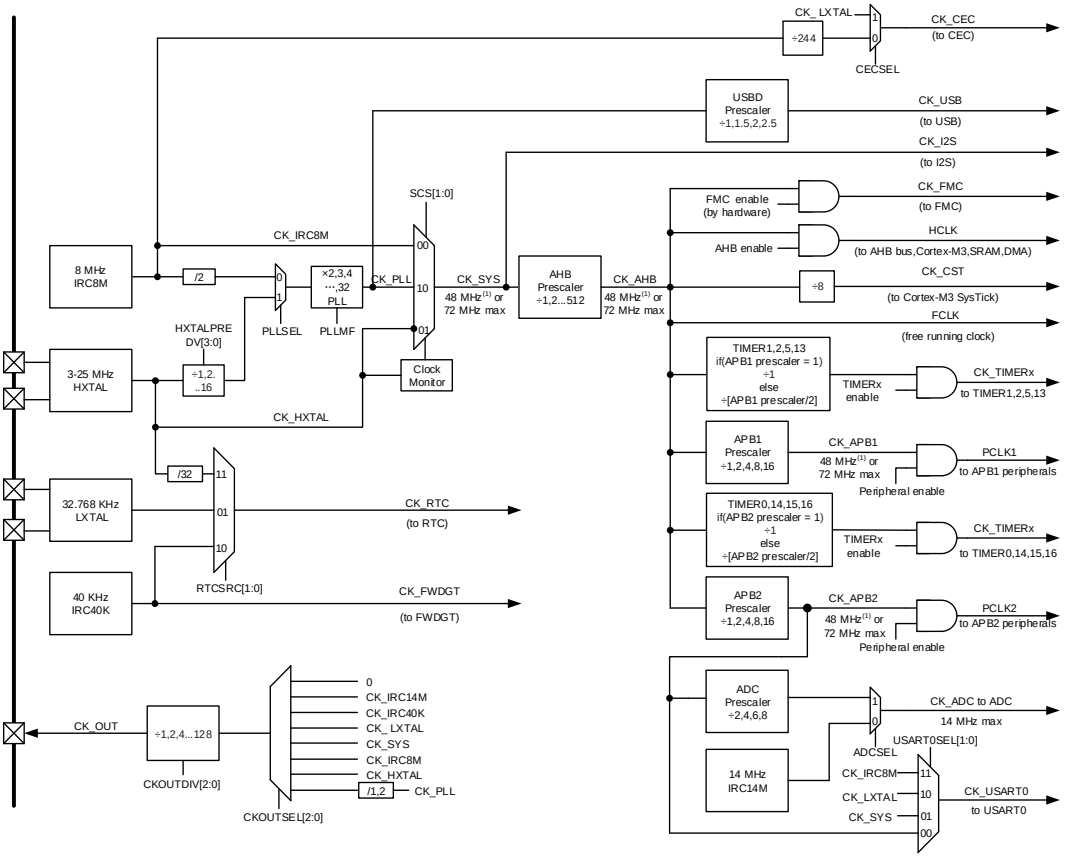
4.2. 时钟控制单元（CCTL）

4.2.1. 简介

时钟控制单元提供了一系列频率的时钟功能，包括一个内部8M RC振荡器时钟（IRC8M）、一个内部高速14M RC振荡器时钟（IRC14M）、一个外部高速晶体振荡器时钟（HXTAL）、一个内部低速RC振荡器时钟（IRC40K）、一个外部低速晶体振荡器时钟（LXTAL）、一个锁相环（PLL）、一个HXTAL时钟监视器、时钟预分频器、时钟多路复用器和时钟选通电路。

AHB、APB和Cortex®-M3时钟都源自系统时钟（CK_SYS），系统时钟的时钟源为IRC8M、HXTAL或PLL。系统时钟的最大运行时钟频率可以达到72MHz，GD32F130xx产品系统时钟的最大运行时钟频率可以达到48MHz。

图4-2.产品的时钟树



注意：GD32F130xx产品系统时钟的最大运行时钟频率为48MHz，GD32F150xx产品系统时钟的最大运行时钟频率为72MHz。

预分频器可以配置AHB、APB2和APB1域的时钟频率。GD32F150xx产品AHB和APB2/APB1域的最高时钟频率为72MHz，GD32F130xx产品AHB和APB2/APB1域的最高时钟频率为48MHz。但当使用I2C外设时，APB1时钟需保证不大于36MHz。RCU通过AHB时钟（HCLK）8分频后作为Cortex系统定时器（SysTick）的外部时钟。通过对SysTick控制与状态寄存器的设置，可选择上述时钟或Cortex（HCLK）时钟作为SysTick时钟。

产品中ADC时钟由APB2时钟经2、4、6、8分频或IRC14M时钟获得，是通过设置配置寄存器2（RCU_CFG2）的ADCSEL位来选择ADC时钟源的。USART0的时钟可以选择IRC8M时钟、LXTAL时钟或APB2时钟，通过设置配置寄存器2（RCU_CFG2）的USART0SEL位来选择。CEC时钟可以选择IRC8M时钟244分频或LXTAL时钟，通过设置配置寄存器2（RCU_CFG2）的CECSEL位来选择。

FWDGT时钟可以选择IRC40K时钟，当FWDGT启动时强制选择。

如果APB时钟分频系数为1，定时器的时钟频率与所在APB总线频率一致。否则，定时器的时钟频率被设为与其相连的APB总线频率的2倍。

4.2.2. 主要特性

- 3到25 MHz外部高速晶体振荡器（HXTAL）；
- 8 MHz内部高速RC振荡器（IRC8M）；

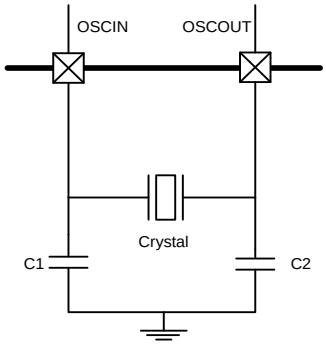
- 14 MHz内部高速RC振荡器（IRC14M）；
- 32,768 Hz外部低速晶体振荡器（LXTAL）；
- 40 kHz内部低速RC振荡器（IRC40K）；
- PLL时钟源可以是HXTAL或IRC8M；
- HXTAL时钟可监视。

4.2.3. 功能描述

高速外部晶体振荡器时钟（HXTAL）

3到25MHz的外部振荡器可为系统提供更为精确的主时钟。带有特定频率的晶体必须靠近两个HXTAL的引脚。和晶体连接的外部电阻和电容必须根据所选择的振荡器来调整。

图4-3. HXTAL接线图



HXTAL晶体可以通过设置时钟控制寄存器RCU_CTL0的HXTALEN位来启动或关闭，在时钟控制寄存器RCU_CTL0中的HXTALSTB位用来指示高速外部振荡器是否已稳定。在启动时，直到这一位被硬件置‘1’，时钟才被释放出来。这个特定的延迟时间又称启动时间。当HXTAL时钟稳定后，如果在时钟中断寄存器RCU_INT中的相应中断使能位HXTALSTBIE位被置‘1’，将会产生相应中断。在这一点上，HXTAL时钟可以被直接用作系统时钟源或者PLL输入时钟。

将时钟控制寄存器RCU_CTL0的HXTALBPS和HXTALEN位置‘1’可以设置外部时钟旁路模式。CK_HXTAL等于驱动OSCIN管脚的外部时钟。

高速内部 8MHz RC 振荡器时钟（IRC8M）

高速内部8MHz RC振荡器时钟，简称IRC8M时钟，拥有8MHz的固定频率，设备上电后CPU默认选择的时钟源就是IRC8M时钟。IRC8M RC振荡器能够在不需要任何外部器件的条件下提供更低成本类型的时钟源。IRC8M晶体可以通过设置时钟控制寄存器（RCU_CTL0）中的IRC8MEN位被启动和关闭。时钟控制寄存器RCU_CTL0中的IRC8MSTB位用来指示IRC8M内部RC振荡器是否稳定。IRC8M振荡器的启动时间比HXTAL晶体振荡器要更短。如果时钟中断寄存器RCU_INT中的相应中断使能位IRC8MSTBIE被置‘1’，在IRC8M稳定以后，将产生一个中断。IRC8M时钟也可用作PLL输入时钟。

工厂会校准IRC8M时钟频率的精度，但是它的精度仍然比HXTAL时钟要差。用户需求、环境条件和成本将决定选择哪个时钟作为系统时钟源。

如果HXTAL或者PLL是系统时钟源，为了最大程度减小系统从深度睡眠模式启动的时间，系统

从深度睡眠模式初始唤醒的时候硬件强制IRC8M时钟作为系统时钟。

锁相环（PLL）

内部锁相环PLL通过对输入参考频率为4~32MHz的时钟基准2 ~32倍频，可以提供16~72 MHz的时钟输出。

PLL可以通过设置时钟控制寄存器0（RCU_CTL0）中的PLLEN位被启动和关闭。时钟控制寄存器RCU_CTL0中的PLLSTB位用来指示PLL时钟是否稳定。如果时钟中断寄存器RCU_INT中的相应中断使能位PLLSTBIE被置‘1’，在PLL稳定以后，将产生一个中断。

高速内部 14M RC 振荡器时钟（IRC14M）

高速内部14M RC振荡器时钟（IRC14M）有一个固定的频率14MHz，专门用作ADC时钟。IRC14M可以通过设置时钟控制寄存器1（RCU_CTL1）中的IRC14MEN位被启动和关闭。时钟控制寄存器1（RCU_CTL1）中的IRC14MSTB位用来指示IRC14M时钟是否已稳定。如果时钟中断寄存器RCU_INT中的相应中断使能位IRC14MSTBIE被置‘1’，在IRC14M稳定以后，将产生一个中断。

低速外部晶体振荡器时钟（LXTAL）

LXTAL晶体是一个32.768kHz的低速外部晶体或陶瓷谐振器。它为实时时钟电路提供一个低功耗且精确的时钟源。LXTAL时钟可以通过设置备份域控制寄存器（RCU_BDCTL）中的LXTALEN位被启动和关闭。备份域控制寄存器RCU_BDCTL中的LXTALSTB位用来指示LXTAL时钟是否稳定。如果时钟中断寄存器RCU_INT中的相应中断使能位LXTALSTBIE被置‘1’，在LXTAL稳定以后，将产生一个中断。

将备份域控制寄存器RCU_BDCTL的LXTALBPS和LXTALEN位置‘1’可以选择外部时钟旁路模式。CK_LXTAL与连到OSC32IN脚上外部时钟信号一致。

低速内部 RC 振荡器时钟（IRC40K）

IRC40K RC振荡器时钟担当一个低功耗时钟源的角色，它的时钟频率大约40 kHz，为独立看门狗定时器和实时时钟电路提供时钟。IRC40K提供低成本的时钟源，因为不需要外部器件。IRC40K RC振荡器可以通过设置控制/状态寄存器RCU_RSTSCK中的IRC40KEN位被启动和关闭。控制/状态寄存器RCU_RSTSCK中的IRC40KSTB位用来指示IRC40K时钟是否已稳定。如果控制/状态寄存器RCU_RSTSCK中的相应中断使能位IRC40KSTBIE被置‘1’，在IRC40K稳定以后，将产生一个中断。

系统时钟（CK_SYS）选择

系统复位后，IRC8M时钟被选为系统时钟，改变时钟配置寄存器RCU_CFG0 中的系统时钟变换位SCS可以切换系统时钟源为HXTAL或PLL。当SCS的值改变，系统时钟将使用原来的时钟源继续运行直到转换的目标时钟源稳定。当一个时钟源被直接或通过PLL间接作为系统时钟时，它将不能被停止。

HXTAL 时钟监视器（CKM）

设置时钟控制寄存器RCU_CTL0中的HXTAL时钟监视使能位CKMEN，HXTAL可以使能时钟监视功能。该功能必须在HXTAL启动延迟完毕后使能，在HXTAL停止后禁止。一旦监测到HXTAL故障，HXTAL将自动被禁止，时钟中断寄存器RCU_INT中的HXTAL时钟阻塞标志位CKMIF将被置‘1’，产生HXTAL故障事件。这个故障引发的中断和Cortex-M3的不可屏蔽中断相连。如果HXTAL被选作系统或PLL的时钟源，HXTAL故障将促使选择IRC8M为系统时钟源且PLL将被自动禁止。

时钟输出功能

GD32F130xx产品时钟输出功能输出从32kHz到48MHz的时钟，GD32F150xx产品时钟输出功能输出从32kHz到72MHz的时钟。通过设置时钟配置寄存器RCU_CFG0中的CK_OUT时钟源选择位CKOUTSEL[2:0]能够选择不同的时钟信号。相应的GPIO引脚应该被配置成复用功能I/O（AFIO）模式来输出选择的时钟信号。

表 4-1. 时钟源的选择

时钟源选择位	时钟源
000	无时钟
001	CK_IRC14M
010	CK_IRC40K
011	CK_LXTAL
100	CK_SYS
101	CK_IRC8M
110	CK_HXTAL
111	CK_PLL 或 CK_PLL/2

通过配置时钟配置寄存器RCU_CFG0的CKOUTDIV[2:0]位，可以将输出时钟按比例分频，进而降低CK_OUT频率。

深度睡眠模式时钟控制

当MCU工作在深度睡眠模式时，HDMI CEC或USART0能唤醒MCU，前提是它们的时钟是由LXTAL时钟提供且LXTAL时钟被使能。

如果HDMI CEC或USART0时钟选择IRC8M时钟并且工作在深度睡眠模式，它们有能力开启或关闭IRC8M时钟，HDMI CEC或USART0采用IRC8M时钟作为工作时钟来唤醒深度睡眠模式。

电压控制

掉电选择寄存器（RCU_PDVSEL）中的PDRVS位可以控制掉电复位。如果PDRVS位是0，V_{DD}低于2.6V的时候电源掉电复位生效。如果PDRVS位是1，V_{DD}低于1.8V的时候掉电复位生效。在PDRVS位是1时，并且V_{DD}低于2.6V时，闪存编程和擦除不能使用。

深度睡眠模式电压寄存器（RCU_DSV）中的DSLPVS[2:0]位可以控制内核在深度睡眠模式下的电压。

表 4-2. 深度睡眠模式下内核电压选择

DSLPVS[2:0]	深度睡眠模式电压（V）
000	1.2
001	1.1
010	1.0
011	0.9
100 ~ 111	保留

RCU_PDVSEL和RCU_DSV寄存器被电源解锁寄存器（RCU_VKEY）保护。只有在写0x1A2B3C4D到RCU_VKEY后，RCU_PDVSEL和RCU_DSV寄存器才能被写入。

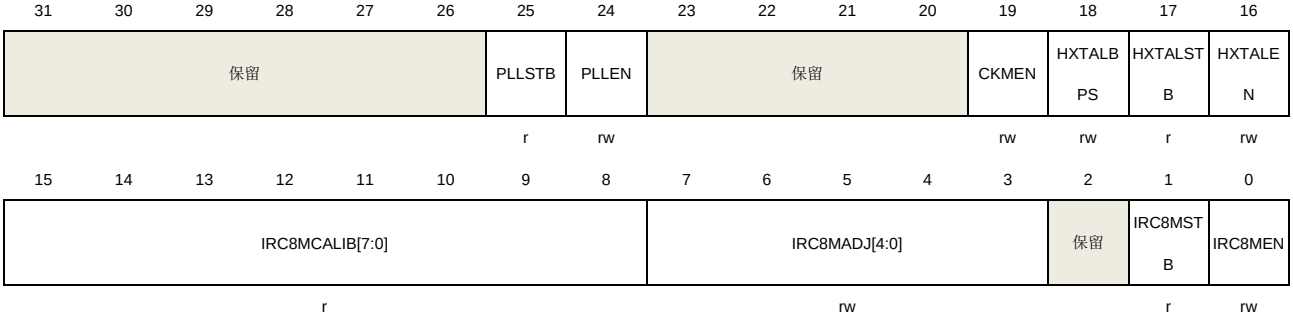
4.3. RCU 寄存器

RCU 基地址：0x4002 1000

4.3.1. 控制寄存器 0（RCU_CTL0）

地址偏移：0x00
复位值：0x0000 XX83 X表示未定义。

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。



位/位域	名称	描述
31:26	保留	必须保持复位值。
25	PLLSTB	PLL 时钟稳定标志位 硬件置‘1’来指示 PLL 输出时钟是否稳定待用。 0：PLL 没稳定 1：PLL 稳定
24	PLLEN	PLL 使能 软件置位或复位。如果 PLL 时钟作为系统时钟的时候该位不能被复位。进入深度睡眠或待机模式时硬件自动复位。 0：PLL 被关闭 1：PLL 被打开
23:20	保留	必须保持复位值。.
19	CKMEN	HXTAL 时钟监视使能 0：禁止外部晶体振荡器（HXTAL）时钟监视器 1：使能外部晶体振荡器（HXTAL）时钟监视器 当硬件监测到 HXTAL 时钟一直停留在低或者高的状态，内部硬件将切换系统时钟到 IRC8M RC 时钟。恢复原来系统时钟的方式有以下几种：外部复位，上电复位，软件清 CKMIF 位。 **注意：**使能 HXTAL 时钟监视器以后，硬件无视控制位 IRC8MEN 的状态，自动使能 IRC8M 时钟。
18	HXTALBPS	外部晶体振荡器（HXTAL）时钟旁路模式使能 只有在 HXTALEN 位为 0 时 HXTALBPS 位才可写。

		0：禁止 HXTAL 旁路模式 1：使能 HXTAL 旁路模式 HXTAL 输出时钟等于输入时钟
17	HXTALSTB	外部晶体振荡器（HXTAL）时钟稳定状态标志位 硬件置‘1’来指示 HXTAL 振荡器时钟是否稳定待用。 0：HXTAL 振荡器未稳定 1：HXTAL 振荡器已稳定
16	HXTALEN	外部高速振荡器时钟使能 软件置‘1’或清‘0’。如果 HXTAL 时钟或者 PLL 输入时钟作为系统时钟，该位不能被复位。进入深度睡眠或待机模式时硬件自动复位。 0：禁止外部晶体振荡器 1：使能外部晶体振荡器
15:8	IRC8MCALIB[7:0]	高速内部振荡器校准值寄存器 上电时自动加载这些位
7:3	IRC8MADJ[4:0]	高速内部振荡器时钟调整值 这些位由软件置位，最终调整值为 IRC8MADJ 当前值加上 IRC8MCALIB[7:0]位的值。最终调整值应该调整 IRC8M 到 8 MHz ± 1%。
2	保留	必须保持复位值。.
1	IRC8MSTB	高速内部（IRC8M）时钟稳定状态标志位 硬件置‘1’来指示 IRC8M 振荡器时钟是否稳定待用。 0：IRC8M 振荡器未稳定 1：IRC8M 振荡器已稳定
0	IRC8MEN	高速内部振荡器使能 软件复位置位。如果 IRC8M 时钟用作系统时钟时该位不能被复位。当从待机和深度睡眠模式返回或用作系统时钟的 HXTAL 振荡器发生故障时，该位由硬件置 1 来启动 IRC8M 振荡器。 0：内部 8 MHz RC 振荡器关闭 1：内部 8 MHz RC 振荡器开启

4.3.2. 配置寄存器 0（RCU_CFG0）

地址偏移：0x04
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PLLDV		CKOUTDIV[2:0]		PLLMF[4]	CKOUTSEL[2:0]			USBDPSC[1:0]		PLLMF[3:0]				PLLPRE DV	PLLSEL
rw		rw		rw	rw			rw		rw				rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCPSC[1:0]		APB2PSC[2:0]			APB1PSC[2:0]			AHBPSC[3:0]				SCSS[1:0]		SCS[1:0]	
rw		rw			rw			rw				r		rw	

位/位域	名称	描述
31	PLLDV	CK_PLL 1 或 2 分频来用作 CK_OUT 0：CK_PLL 2 分频用作 CK_OUT 1： CK_PLL 用作 CK_OUT
30:28	CKOUTDIV[2:0]	CK_OUT 分频器，来降低 CK_OUT 频率 CK_OUT 的选择参考 RCU_CFG0 的 26:24 位。 000：CK_OUT 不分频 001：CK_OUT 2 分频 010：CK_OUT 4 分频 011：CK_OUT 8 分频 100：CK_OUT 16 分频 101：CK_OUT 32 分频 110：CK_OUT 64 分频 111：CK_OUT 128 分频
27	PLLMF[4]	PLLMF 寄存器第 4 位 见 RCU_CFG0 的 21:18 位
26:24	CKOUTSEL[2:0]	CK_OUT 时钟源选择 软件置位或清零。 000：没有时钟被选择 001：选择内部 14M RC 振荡器时钟 010：选择内部 40K RC 振荡器时钟 011：选择外部低速振荡器时钟 100：选择系统时钟 101：选择内部 8M RC 振荡器时钟 110：选择外部高速振荡器时钟 111：依赖于 PLLDV 选择（CK_PLL / 2）或 CK_PLL
23:22	USBDPSC[1:0]	USBD 时钟预分频选择 软件置位或清零来控制 USBD 时钟预分频值。USBD 时钟必须为 48MHz.。如果 USBD 时钟使能这些位不能被复位。 00：选择（CK_PLL / 1.5） 01：选择 CK_PLL 10：选择（CK_PLL / 2.5） 11：选择（CK_PLL / 2）
21:18	PLLMF[3:0]	PLL 倍频因子 软件写这些位包括 RCU_CFG0 的 27 位来确定 PLL 的倍频因子。 00000：（PLL 时钟源 x 2） 00001：（PLL 时钟源 x 3） 00010：（PLL 时钟源 x 4） 00011：（PLL 时钟源 x 5） 00100：（PLL 时钟源 x 6） 00101：（PLL 时钟源 x 7）

00110：（PLL 时钟源 x 8）
00111：（PLL 时钟源 x 9）
01000：（PLL 时钟源 x 10）
01001：（PLL 时钟源 x 11）
01010：（PLL 时钟源 x 12）
01011：（PLL 时钟源 x 13）
01100：（PLL 时钟源 x 14）
01101：（PLL 时钟源 x 15）
01110：（PLL 时钟源 x 16）
01111：（PLL 时钟源 x 16）
10000：（PLL 时钟源 x 17）
10001：（PLL 时钟源 x 18）
10010：（PLL 时钟源 x 19）
10011：（PLL 时钟源 x 20）
10100：（PLL 时钟源 x 21）
10101：（PLL 时钟源 x 22）
10110：（PLL 时钟源 x 23）
10111：（PLL 时钟源 x 24）
11000：（PLL 时钟源 x 25）
11001：（PLL 时钟源 x 26）
11010：（PLL 时钟源 x 27）
11011：（PLL 时钟源 x 28）
11100：（PLL 时钟源 x 29）
11101：（PLL 时钟源 x 30）
11110：（PLL 时钟源 x 31）
11111：（PLL 时钟源 x 32）
注意：GD32F130xx 产品 PLL 输出频率不能超过 48MHz，GD32F150xx 产品 PLL 输出频率不能超过 72MHz。

17	PLLPREDV	HXTAL 分频器作为 PLL 输入。该位与时钟配置寄存器 1（RCU_CFG1）中的 HXTALPREDV[0]位是一样的。参考 RCU_CFG1 的 HXTALPREDV 位说明。 由软件置 1 或清 0 来分频或不分频 HXTAL 后，当作 PLL 输入时钟源。 0：选择 HXTAL 时钟 1：HXTAL 时钟二分频
16	PLLSEL	PLL 时钟源选择 软件置 1 或清 0 来控制 PLL 时钟源 0：选择 IRC8M 二分频为 PLL 时钟源 1：选择 HXTAL 为 PLL 时钟源
15:14	ADCPSC[1:0]	ADC 时钟预分频选择 软件清 0 和置 1。 00：选择 APB2 时钟 2 分频 01：选择 APB2 时钟 4 分频 10：选择 APB2 时钟 6 分频

		11：选择 APB2 时钟 8 分频
13:11	APB2PSC[2:0]	APB2 预分频选择 软件置 1 和清 0 来控制 APB2 时钟分频因子。 0xx：选择 AHB 时钟不分频 100：选择 AHB 时钟 2 分频 101：选择 AHB 时钟 4 分频 110：选择 AHB 时钟 8 分频 111：选择 AHB 时钟 16 分频
10:8	APB1PSC[2:0]	APB1 预分频选择 软件设置和清除来控制 APB1 时钟分频因子。 0xx：选择 AHB 时钟不分频 100：选择 AHB 时钟 2 分频 101：选择 AHB 时钟 4 分频 110：选择 AHB 时钟 8 分频 111：选择 AHB 时钟 16 分频
7:4	AHBPSC[3:0]	AHB 预分频选择 软件设置和清除来控制 AHB 时钟分频因子。 0xxx：选择 CK_SYS 系统时钟不分频 1000：选择 CK_SYS 系统时钟 2 分频 1001：选择 CK_SYS 系统时钟 4 分频 1010：选择 CK_SYS 系统时钟 8 分频 1011：选择 CK_SYS 系统时钟 16 分频 1100：选择 CK_SYS 系统时钟 64 分频 1101：选择 CK_SYS 系统时钟 128 分频 1110：选择 CK_SYS 系统时钟 256 分频 1111：选择 CK_SYS 系统时钟 512 分频
3:2	SCSS[1:0]	系统时钟转换状态 硬件设置和清除指示系统当前时钟源 00：选择 CK_IRC8M 作为 CK_SYS 系统时钟源 01：选择 CK_HXTAL 作为 CK_SYS 系统时钟源 10：选择 CK_PLL 作为 CK_SYS 系统时钟源 11：保留
1:0	SCS[1:0]	系统时钟转换 软件设置选择系统时钟源。由于 CK_SYS 的改变有固有的延迟，需要软件读 SCSS 位来确保转换是否结束。在从深度睡眠或待机模式中返回时，或作为系统时钟或 PLL 时钟源的 HXTAL 出现故障时，强制选择 IRC8M 作为系统时钟。 00：选择 IRC8M 时钟作为 CK_SYS 系统时钟源 01：选择 HXTAL 时钟作为 CK_SYS 系统时钟源 10：选择 PLL 作为 CK_SYS 系统时钟源 11：保留

4.3.3. 中断寄存器（RCU_INT）

地址偏移：0x08

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								CKMIC	保留	IRC14MSTBIC	PLLSTBIC	HXTALSTBIC	IRC8MSTBIC	LXTALSTBIC	IRC40KSTBIC
								w		w	w	w	w	w	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		IRC14MSTBIE	PLLSTBIE	HXTALSTBIE	IRC8MSTBIE	LXTALSTBIE	IRC40KSTBIE	CKMIF	保留	IRC14MSTBIF	PLLSTBIF	HXTALSTBIF	IRC8MSTBIF	LXTALSTBIF	IRC40KSTBIF
		rw	rw	rw	rw	rw	rw	r		r	r	r	r	r	r

位/位域	名称	描述
31:24	保留	必须保持复位值。
23	CKMIC	HXTAL 时钟阻塞中断清除 软件写 1 复位 CKMIF 标志位。 0：不复位 CKMIF 标志位 1：复位 CKMIF 标志位
22	保留	必须保持复位值。
21	IRC14MSTBIC	IRC14M 时钟稳定中断清除 软件写 1 复位 IRC14MSTBIF 标志位。 0：不复位 IRC14MSTBIF 标志位 1：复位 IRC14MSTBIF 标志位
20	PLLSTBIC	PLL 稳定中断清除 软件写 1 复位 PLLSTBIF 标志位。 0：不复位 PLLSTBIF 标志位 1：复位 PLLSTBIF 标志位
19	HXTALSTBIC	HXTAL 时钟稳定中断清除 软件写 1 复位 HXTALSTBIF 标志位。 0：不复位 HXTALSTBIF 标志位 1：复位 HXTALSTBIF 标志位
18	IRC8MSTBIC	IRC8M 时钟稳定中断清除 软件写 1 复位 IRC8MSTBIF 标志位。 0：不复位 IRC8MSTBIF 标志位 1：复位 IRC8MSTBIF 标志位
17	LXTALSTBIC	LXTAL 时钟稳定中断清除 软件写 1 复位 LXTALSTBIF 标志位。 0：不复位 LXTALSTBIF 标志位

		1：复位 LXTALSTBIF 标志位
16	IRC40KSTBIC	IRC40K 时钟稳定中断清除 软件写 1 复位 IRC40KSTBIF 标志位。 0：不复位 IRC40KSTBIF 标志位 1：复位 IRC40KSTBIF 标志位
15:14	保留	必须保持复位值。
13	IRC14MSTBIE	IRC14M 时钟稳定中断使能 软件置 1 和清 0 来使能/禁止 IRC14M 时钟稳定中断。 0：禁止 IRC14M 时钟稳定中断 1：使能 IRC14M 时钟稳定中断
12	PLLSTBIE	PLL 时钟稳定中断使能 软件置 1 和清 0 来使能/禁止 PLL 时钟稳定中断。 0：禁止 PLL 时钟稳定中断 1：使能 PLL 时钟稳定中断
11	HXTALSTBIE	HXTAL 时钟稳定中断使能 软件置 1 和清 0 来使能/禁止 HXTAL 时钟稳定中断。 0：禁止 HXTAL 时钟稳定中断 1：使能 HXTAL 时钟稳定中断
10	IRC8MSTBIE	IRC8M 时钟稳定中断使能 软件置 1 和清 0 来使能/禁止 IRC8M 时钟稳定中断。 0：禁止 IRC8M 时钟稳定中断 1：使能 IRC8M 时钟稳定中断
9	LXTALSTBIE	LXTAL 时钟稳定中断使能 LXTAL 时钟稳定中断使能/禁止控制。 0：禁止 LXTAL 时钟稳定中断 1：使能 LXTAL 时钟稳定中断
8	IRC40KSTBIE	IRC40K 时钟稳定中断使能 IRC40K 时钟稳定中断使能/禁止控制。 0：禁止 IRC40K 时钟稳定中断 1：使能 IRC40K 时钟稳定中断
7	CKMIF	HXTAL 时钟阻塞中断标志位 当 HXTAL 时钟阻塞时硬件置 1。 软件置 CKMIC=1 时清除该位。 0：时钟运行正常 1：HXTAL 时钟阻塞
6	保留	必须保持复位值。
5	IRC14MSTBIF	IRC14M 时钟稳定中断标志位 当 IRC14M 时钟稳定且 IRC14MSTBIE 位被置 1 时由硬件置 1。 软件置 IRC14MSTBIC=1 时清除该位。

0：无 IRC14M 时钟稳定中断产生
1：IRC14M 时钟稳定中断发生

4	PLLSTBIF	PLL 时钟稳定中断标志位 当 PLL 时钟稳定且 PLLSTBIE 位被置 1 时由硬件置 1。 软件置 PLLSTBIC=1 时清除该位。 0：无 PLL 时钟稳定中断产生 1：产生 PLL 时钟稳定中断
3	HXTALSTBIF	HXTAL 时钟稳定中断标志位 当外部 4 ~ 32 MHz 晶体振荡器时钟稳定且 HXTALSTBIE 位被置 1 时由硬件置 1。 软件置 HXTALSTBIC=1 时清除该位。 0：无 HXTAL 时钟稳定中断发生 1：发生 HXTAL 时钟稳定中断
2	IRC8MSTBIF	IRC8M 时钟稳定中断标志位 当内部 8 MHz RC 振荡器时钟稳定且 IRC8MSTBIE 位被置 1 时由硬件置 1。 软件置 IRC8MSTBIC=1 时清除该位。 0：无 IRC8M 时钟稳定中断产生 1：产生 IRC8M 时钟稳定中断
1	LXTALSTBIF	LXTAL 时钟稳定中断标志位 当外部 32.768KHz 晶体振荡器时钟稳定且 LXTALSTBIE 为被置 1 时由硬件置 1。 软件置 LXTALSTBIC=1 时清除该位。 0：无 LXTAL 时钟稳定中断发生 1：发生 LXTAL 时钟稳定中断
0	IRC40KSTBIF	IRC40K 时钟稳定中断标志位 当内部 32kHz RC 振荡器时钟稳定且 IRC40KSTBIE 位被置 1 时由硬件置 1。 软件置 IRC40KSTBIC =1 时清除该位。 0：无 IRC40K 时钟稳定中断产生 1：产生 IRC40K 时钟稳定中断

4.3.4. APB2 复位寄存器（RCU_APB2RST）

地址偏移：0x0C
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留													TIMER16 RST	TIMER15 RST	TIMER14 RST
													rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	USART0 RST	保留.	SPI0RST	TIMER0 RST	保留.	ADCRST	保留								CFGCMP RST

rw rw rw rw rw

位/位域	名称	描述
31:19	保留	必须保持复位值。
18	TIMER16RST	TIMER16 定时器复位 由软件置 1 或清 0。 0：无复位 1：复位 TIMER16 定时器
17	TIMER15RST	TIMER15 定时器复位 由软件置 1 或清 0。 0：无复位 1：复位 TIMER15 定时器
16	TIMER14RST	TIMER14 定时器复位 由软件置 1 或清 0。 0：无复位 1：复位 TIMER14 定时器
14	USART0RST	USART0 复位 由软件置 1 或清 0。 0：无复位 1：复位 USART0
13	保留	必须保持复位值。
12	SPI0RST	SPI0 复位 由软件置 1 或清 0。 0：无复位 1：复位 SPI0
11	TIMER0RST	TIMER0 定时器复位 由软件置 1 或清 0。 0：无复位 1：复位 TIMER0 定时器
10	保留	必须保持复位值。
9	ADCRST	ADC 复位 由软件置 1 或清 0。 0：无复位 1：复位 ADC
8:1	保留	必须保持复位值。
0	CFGCMPRST	系统配置和比较器复位 由软件置 1 或清 0。 0：无复位

1：复位系统配置模块和比较器

4.3.5. APB1 复位寄存器（RCU_APB1RST）

地址偏移：0x10
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	CECRST	DACRST	PMURST	保留				USBDRST	I2C1RST	I2C0RST	保留			USART1RST	保留.
	rw	rw	rw					rw	rw	rw				rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI2RST	SPI1RST	保留		WWDGTRST	保留		TIMER13RST	保留			TIMER5RST	保留		TIMER2RST	TIMER1RST
rw	rw			rw			rw				rw			rw	rw

位/位域	名称	描述
31	保留	必须保持复位值。
30	CECRST	HDMI CEC 复位 由软件置 1 或清 0。 0：无复位 1：复位 HDMI CEC 单元
29	DACRST	DAC 复位 由软件置 1 或清 0。 0：无复位 1：复位 DAC 单元
28	PMURST	电源控制复位 由软件置 1 或清 0。 0：无复位 1：复位电源控制单元
27:24	保留	必须保持复位值。
23	USBDRST	USBD 复位 由软件置 1 或清 0。 0：无复位 1：复位 USBD
22	I2C1RST	I2C1 复位 由软件置 1 或清 0。 0：无复位 1：复位 I2C1
21	I2C0RST	I2C0 复位

		由软件置 1 或清 0。 0：无复位 1：复位 I2C0
20:18	保留	必须保持复位值。
17	USART1RST	USART1 复位 由软件置 1 或清 0。 0：无复位 1：复位 USART1
16	保留	必须保持复位值。
15	SPI2RST	SPI2 复位 由软件置 1 或清 0。 0：无复位 1：复位 SPI2
14	SPI1RST	SPI1 复位 由软件置 1 或清 0。 0：无复位 1：复位 SPI1
13:12	保留	必须保持复位值。
11	WWDGTRST	窗口看门狗定时器复位 由软件置 1 或清 0。 0：无复位 1：复位窗口看门狗定时器
10:9	保留	必须保持复位值。
8	TIMER13RST	TIMER13 定时器复位 由软件置 1 或清 0。 0：无复位 1：复位 TIMER13 定时器
7:5	保留	必须保持复位值。
4	TIMER5RST	TIMER5 定时器复位 由软件置 1 或清 0。 0：无复位 1：复位 TIMER5 定时器
3:2	保留	必须保持复位值。
1	TIMER2RST	TIMER2 定时器复位 由软件置 1 或清 0。 0：无复位 1：复位 TIMER2 定时器

0	TIMER1RST	TIMER1 定时器复位 由软件置 1 或清 0。 0：无复位 1：复位 TIMER1 定时器

4.3.6. AHB 使能寄存器（RCU_AHBEN）

地址偏移：0x14

复位值：0x0000 0014

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留							TSIEN	保留	PFEN	保留	PDEN	PCEN	PBEN	PAEN	保留
							rw		rw		rw	rw	rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留.									CRCEN	保留	FMC SPEN	保留	SRAM SPEN	保留	DMAEN
									rw		rw		rw		rw

位/位域	名称	描述
31:25	保留	必须保持复位值。
24	TSIEN	TSI 时钟使能 由软件置 1 或清 0。 0：TSI 时钟关闭 1：TSI 时钟开启
23	保留	必须保持复位值。
22	PFEN	GPIOF 时钟使能 由软件置 1 或清 0。 0：GPIOF 时钟关闭 1：GPIOF 时钟开启
21	保留	必须保持复位值。
20	PDEN	GPIOD 时钟使能 由软件置 1 或清 0。 0：GPIOD 时钟关闭 1：GPIOD 时钟开启
19	PCEN	GPIOC 时钟使能 由软件置 1 或清 0。 0：GPIOC 时钟关闭 1：GPIOC 时钟开启
18	PBEN	GPIOB 时钟使能 由软件置 1 或清 0。

		0：GPIOB 时钟关闭 1：GPIOB 时钟开启
17	PAEN	GPIOA 时钟使能 由软件置 1 或清 0。 0：GPIOA 时钟关闭 1：GPIOA 时钟开启
16:7	保留	必须保持复位值。
6	CRCEN	CRC 时钟使能 由软件置 1 或清 0。 0：CRC 时钟关闭 1：CRC 时钟开启
5	保留	必须保持复位值。
4	FMCSPEN	在睡眠模式下 FMC 时钟使能 由软件置 1 或清 0 来开启/关闭在睡眠模式下的 FMC 时钟。 0：关闭睡眠模式下的 FMC 时钟 1：开启睡眠模式下的 FMC 时钟
3	保留	必须保持复位值。
2	SRAMSPEN	在睡眠模式下 SRAM 接口时钟使能 由软件置 1 或清 0 来开启/关闭在睡眠模式下的 SRAM 时钟。 0：关闭睡眠模式下的 SRAM 接口时钟 1：开启睡眠模式下的 SRAM 接口时钟
1	保留	必须保持复位值。
0	DMAEN	DMA 时钟使能 由软件置 1 或清 0。 0：关闭 DMA 时钟 1：开启 DMA 时钟

4.3.7. APB2 使能寄存器（RCU_APB2EN）

地址偏移：0x18
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留													TIMER16EN	TIMER15EN	TIMER14EN
													rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	USART0EN	保留	SPI0EN	TIMER0EN	保留	ADCEN	保留								CFGCMPEN

rw rw rw rw rw

位/位域	名称	描述
31:19	保留	必须保持复位值。
18	TIMER16EN	TIMER16 定时器时钟使能 由软件置 1 或清 0。 0：关闭 TIMER16 定时器时钟 1：开启 TIMER16 定时器时钟
17	TIMER15EN	TIMER15 定时器时钟使能 由软件置 1 或清 0。 0：关闭 TIMER15 定时器时钟 1：开启 TIMER15 定时器时钟
16	TIMER14EN	TIMER14 定时器时钟使能 由软件置 1 或清 0。 0：关闭 TIMER14 定时器时钟 1：开启 TIMER14 定时器时钟
15	保留	必须保持复位值。
14	USART0EN	USART0 时钟使能 由软件置 1 或清 0。 0：关闭 USART0 时钟 1：开启 USART0 时钟
13	保留	必须保持复位值。
12	SPI0EN	SPI0 时钟使能 由软件置 1 或清 0。 0：关闭 SPI0 时钟 1：开启 SPI0 时钟
11	TIMER0EN	TIMER0 定时器时钟使能 由软件置 1 或清 0。 0：关闭 TIMER0 定时器时钟 1：开启 TIMER0 定时器时钟
10	保留	必须保持复位值。
9	ADCEN	ADC 接口时钟使能 由软件置 1 或清 0。 0：关闭 ADC 接口时钟 1：开启 ADC 接口时钟
8:1	保留	必须保持复位值。
0	CFGCMPEN	系统配置与比较器时钟使能 由软件置 1 或清 0。

0：关闭系统配置与比较器模块时钟
1：开启系统配置与比较器模块时钟

4.3.8. APB1 使能寄存器（RCU_APB1EN）

地址偏移：0x1C
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	CECEN	DACEN	PMUEN	保留				USBDEN	I2C1EN	I2C0EN	保留			USART1EN	保留
	rw	rw	rw					rw	rw	rw				rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI2EN	SPI1EN	保留		WWDGTEN	保留		TIMER13EN	保留			TIMER5EN	保留		TIMER2EN	TIMER1EN
rw	rw			rw			rw				rw			rw	rw

位/位域	名称	描述
31	保留	必须保持复位值。
30	CECEN	HDMI CEC 接口时钟使能 由软件置 1 或清 0。 0：关闭 HDMI CEC 接口时钟 1：开启 HDMI CEC 接口时钟
29	DACEN	DAC 接口时钟使能 由软件置 1 或清 0。 0：关闭 DAC 接口时钟 1：开启 DAC 接口时钟
28	PMUEN	电源接口时钟使能 由软件置 1 或清 0。 0：关闭电源接口时钟 1：开启电源接口时钟
27:24	保留	必须保持复位值。
23	USBDEN	USBD 时钟使能 由软件置 1 或清 0。 0：关闭 USBD 时钟 1：开启 USBD 时钟
22	I2C1EN	I2C1 时钟使能 由软件置 1 或清 0。 0：关闭 I2C1 时钟 1：开启 I2C1 时钟

21	I2C0EN	I2C0 时钟使能 由软件置 1 或清 0。 0：关闭 I2C0 时钟 1：开启 I2C0 时钟
20:18	保留	必须保持复位值。
17	USART1EN	USART1 时钟使能 由软件置 1 或清 0。 0:关闭 USART1 时钟 1:开启 USART1 时钟
16	保留	必须保持复位值。
15	SPI2EN	SPI2 时钟使能 由软件置 1 或清 0。 0：关闭 SPI2 时钟 1：开启 SPI2 时钟
14	SPI1EN	SPI1 时钟使能 由软件置 1 或清 0。 0：关闭 SPI1 时钟 1：开启 SPI1 时钟
13:12	保留	必须保持复位值。
11	WWDGTEN	窗口看门狗定时器时钟使能 由软件置 1 或清 0。 0：关闭窗口看门狗定时器时钟 1：开启窗口看门狗定时器时钟
10:9	保留	必须保持复位值。
8	TIMER13EN	TIMER13 定时器时钟使能 由软件置 1 或清 0。 0：关闭 TIMER13 定时器时钟 1：开启 TIMER13 定时器时钟
7:5	保留	必须保持复位值。
4	TIMER5EN	TIMER5 定时器时钟使能 由软件置 1 或清 0。 0：关闭 TIMER5 定时器时钟 1：开启 TIMER5 定时器时钟
3:2	保留	必须保持复位值。
1	TIMER2EN	TIMER2 定时器时钟使能 由软件置 1 或清 0。 0：关闭 TIMER2 定时器时钟

		1：开启 TIMER2 定时器时钟
0	TIMER1EN	TIMER1 定时器时钟使能 由软件置 1 或清 0。 0：关闭 TIMER1 定时器时钟 1：开启 TIMER1 定时器时钟

4.3.9. 备份域控制寄存器（RCU_BDCTL）

地址偏移：0x20
复位值：0x0000 0018，由备份域复位电路复位

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。
注意：备份域控制寄存器(BDCTL)的LXTALEN, LXTALBPS, RTCSRC和RTCEN位仅在备份域复位后才清0。只有在电源控制寄存器（PMU_CTL）中的BKPWEN位置1后才能对这些位进行改动。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															BKPRST
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTCEN	保留					RTCSRC[1:0]		保留			LXTALDRI[1:0]		LXTALBPS	LXTALSTB	LXTALEN
rw						rw					rw		rw	r	rw

位/位域	名称	描述
31:17	保留	必须保持复位值。
16	BKPRST	备份域复位 由软件置 1 或清 0。 0：无复位 1：复位备份域
15	RTCEN	RTC 时钟使能 由软件置 1 或清 0。 0：关闭 RTC 时钟 1：开启 RTC 时钟
14:10	保留	必须保持复位值。
9:8	RTCSRC[1:0]	RTC 时钟入口选择 软件置位或清除来控制 RTC 时钟源。 00：没有时钟 01：选择 LXTAL 时钟作为 RTC 时钟源 10：选择 IRC40K 时钟作为 RTC 时钟源 11：选择 HXTAL 时钟 32 分频作为 RTC 时钟源
7:5	保留	必须保持复位值。

4:3	LXTALDRI[1:0]	LXTAL 驱动能力 软件置位或清除。当复位备份域时，会重装载缺省值。 00：弱驱动能力 01：中低驱动能力 10：中高驱动能力 11：强驱动能力 (复位后的缺省值) 注：LXTALDRI 在旁路模式下无效
2	LXTALBPS	LXTAL 旁路模式使能 软件置 1 和清 0。 0：禁止 LXTAL 旁路模式 1：使能 LXTAL 旁路模式
1	LXTALSTB	外部低速振荡器稳定状态位 硬件置 1 来指示 LXTAL 输出时钟是否稳定待用。 0：LXTAL 未稳定 1：LXTAL 已稳定
0	LXTALEN	LXTAL 使能 软件置 1 和清 0。 0：关闭 LXTAL 1：开启 LXTAL

4.3.10. 复位源/时钟寄存器（RCU_RSTSCK）

地址偏移：0x24

复位值：0x0C00 0000，除复位标志外由系统复位清除，复位标志只能由电源复位清除。

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPRSTF	WWDGTRSTF	FWDGTRSTF	SWRSTF	PORRSTF	EPRSTF	OBLRSTF	RSTFC	V12RSTF	保留						
rw	rw	rw	rw	rw	rw	rw	rw	rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留														IRC40KSTB	IRC40KEN
														r	rw

位/位域	名称	描述
31	LPRSTF	低功耗复位标志位 深度睡眠/待机复位发生时由硬件置 1。 由软件通过写 RSTFC 位来清除该位。 0：无低功耗管理复位发生 1：发生低功耗管理复位
30	WWDGTRSTF	窗口看门狗定时器复位标志位

		窗口看门狗定时器复位发生时由硬件置 1。 由软件通过写 RSTFC 位来清除该位。 0：无窗口看门狗定时器复位发生 1：发生窗口看门狗定时器复位
29	FWDGTRSTF	独立看门狗定时器复位标志位 独立看门狗复位发生时由硬件置 1。 由软件通过写 RSTFC 位来清除该位。 0：无独立看门狗定时器复位发生 1：发生独立看门狗定时器复位
28	SWRSTF	软件复位标志位 软件复位发生时由硬件置 1。 由软件通过写 RSTFC 位来清除该位。 0：无软件复位发生 1：发生软件复位
27	PORRSTF	电源复位标志位 电源复位发生时由硬件置 1。 由软件通过写 RSTFC 位来清除该位。 0：无电源复位发生 1：发生电源复位
26	EPRSTF	外部引脚复位标志位 当有外部引脚复位发生时由硬件置 1。 由软件通过写 RSTFC 位来清除该位。 0：无外部引脚复位发生 1：发生外部引脚复位
25	OBLRSTF	可选字节装载器复位标志位 可选字节装载器装载字节时由硬件置 1。 由软件通过写 RSTFC 位来清除该位。 0：无选项字节装载器复位发生 1：发生选项字节装载器复位
24	RSTFC	清除复位标志位 由软件置 1 来清除所有复位标志位。 0：无作用 1：清除复位标志位
23	V12RSTF	1.2V 域电源复位标志位 当有 1.2V 域电源复位发生时由硬件置 1。 由软件通过写 RSTFC 位来清除该位。 0：无 1.2V 域电源复位发生 1：发生 1.2V 域电源复位
22:2	保留	必须保持复位值。.
1	IRC40KSTB	IRC40K 时钟稳定状态位

		该位由硬件置 1 指示 IRC40K 输出时钟是否稳定待用。 0：IRC40K 时钟未稳定 1：IRC40K 时钟已稳定
0	IRC40KEN	IRC40K 时钟使能 软件置 1 和清 0。 0：关闭 IRC40K 时钟 1：开启 IRC40K 时钟

4.3.11. AHB 复位寄存器（RCU_AHBRST）

地址偏移：0x28
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留							TSIRST	保留	PFRST	保留	PDRST	PCRST	PBRST	PARST	保留
							rw		rw		rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留.															

位/位域	名称	描述
31:25	保留	必须保持复位值。
24	TSIRST	TSI 单元复位 由软件置 1 或清 0。 0：无作用 1：复位 TSI 单元
23	保留	必须保持复位值。
22	PFRST	GPIOF 复位 由软件置 1 或清 0。 0：无作用 1：复位 GPIOF 口
21	保留	必须保持复位值。
20	PDRST	GPIOD 复位 由软件置 1 或清 0. 0：无作用 1：复位 GPIOD 口
19	PCRST	GPIOC 复位 由软件置 1 或清 0。 0：无作用

		1：复位 GPIOC 口
18	PBRST	GPIOB 复位 由软件置 1 或清 0。 0：无作用 1：复位 GPIOB 口
17	PARST	GPIOA 复位 由软件置 1 或清 0。 0：无作用 1：复位 GPIOA 口
16:0	保留	必须保持复位值。

4.3.12. 配置寄存器 1（RCU_CFG1）

地址偏移：0x2C
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留												HXTALPREDV[3:0]			
												rw			

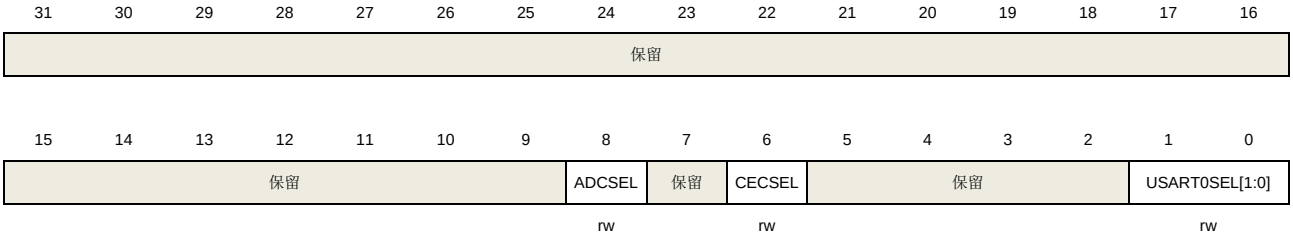
位/位域	名称	描述
31:4	保留	必须保持复位值。
3:0	HXTALPREDV[3:0]	HXTAL 时钟作为 PLL 输入源分频因子 由软件置 1 或清 0。这些位仅能在 PLL 关闭时改写。 注：HXTALPREDV 的位 0 与 RCU_CFG0 的位 17 的功能相同，修改 RCU_CFG0 的位 17 同时改变这里的位 0，HXTAL 时钟分频因子为（HXTALPREDV + 1） 0000：HXTAL 作为 PLL 的输入，不分频 0001：HXTAL 作为 PLL 的输入 2 分频 0010：HXTAL 作为 PLL 的输入 3 分频 0011：HXTAL 作为 PLL 的输入 4 分频 0100：HXTAL 作为 PLL 的输入 5 分频 0101：HXTAL 作为 PLL 的输入 6 分频 0110：HXTAL 作为 PLL 的输入 7 分频 0111：HXTAL 作为 PLL 的输入 8 分频 1000：HXTAL 作为 PLL 的输入 9 分频 1001：HXTAL 作为 PLL 的输入 10 分频 1010：HXTAL 作为 PLL 的输入 11 分频 1011：HXTAL 作为 PLL 的输入 12 分频

1100：HXTAL 作为 PLL 的输入 13 分频
1101：HXTAL 作为 PLL 的输入 14 分频
1110：HXTAL 作为 PLL 的输入 15 分频
1111：HXTAL 作为 PLL 的输入 16 分频

4.3.13. 配置寄存器 2（RCU_CFG2）

地址偏移：0x30
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。



位/位域	名称	描述
31:9	保留	必须保持复位值。
8	ADCSEL	ADC 时钟源选择 由软件置 1 或清 0。 0：ADC 时钟源选择 IRC14M 时钟 1：ADC 时钟源选择 APB2 时钟 2/4/6/8 分频
7	保留	必须保持复位值。
6	CECSEL	CEC 时钟源选择 由软件置 1 或清 0。 0：CEC 时钟选择 IRC8M 时钟 244 分频 1：CEC 时钟选择 LXTAL 时钟
5:2	保留	必须保持复位值。
1:0	USART0SEL[1:0]	USART0 时钟源选择 由软件置 1 或清 0。 00：USART0 时钟选择 APB2 时钟 01：USART0 时钟选择系统时钟 10：USART0 时钟选择 LXTAL 时钟 11：USART0 时钟选择 IRC8M 时钟

4.3.14. 控制寄存器 1（RCU_CTL1）

地址偏移：0x34
复位值：0x0000 XX80 X表示未定义

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

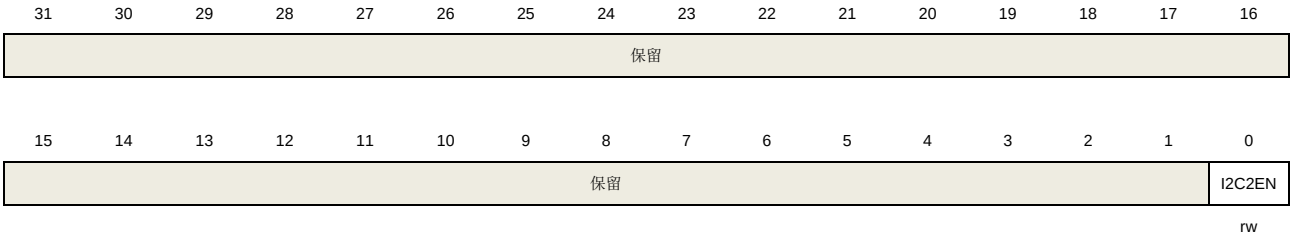
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:8	IRC14MCALIB[7:0]	IRC14M 时钟校准 启动时这些位会被自动初始化为出厂校准参数。
7:3	IRC14MADJ[4:0]	IRC14M 时钟调整 这些位由软件设定，最终调整值为(IRC14MADJ)位的值加上 IRC14MCALIB[7:0]位的值。最终调整值应该把 IRC14M 调整到 14MHz ±1%。
2	保留	必须保持复位值。
1	IRC14MSTB	IRC14M 时钟稳定标志位 由硬件置 1 来指示内部 IRC14M 振荡器已经稳定可用。 0：IRC14M 振荡器未稳定 1：IRC14M 振荡器已稳定
0	IRC14MEN	IRC14M 时钟使能 软件置 1 和清 0。 0：关闭 IRC14M 时钟 1：开启 IRC14M 时钟

4.3.15. APB1 附加使能寄存器（RCU_ADDAPB1EN）

地址偏移：0xF8

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。



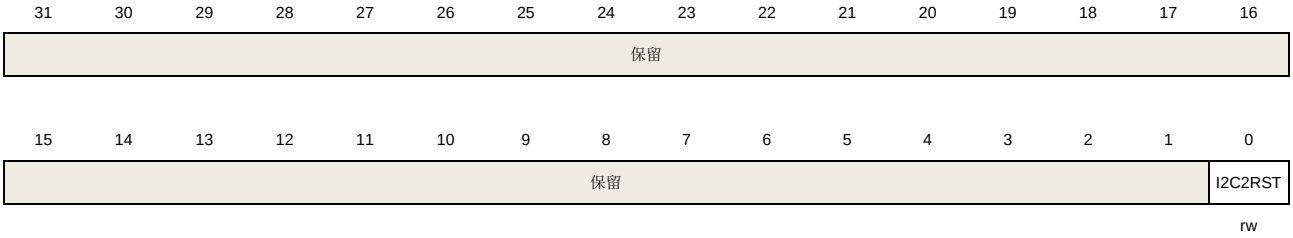
位/位域	名称	描述
31:1	保留	必须保持复位值。

0	I2C2EN	I2C2 时钟单元使能 由软件置 1 或清 0。 0：关闭 I2C2 时钟单元 1：开启 I2C2 时钟单元

4.3.16. APB1 附加复位寄存器（RCU_ADDAPB1RST）

地址偏移：0xFC
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

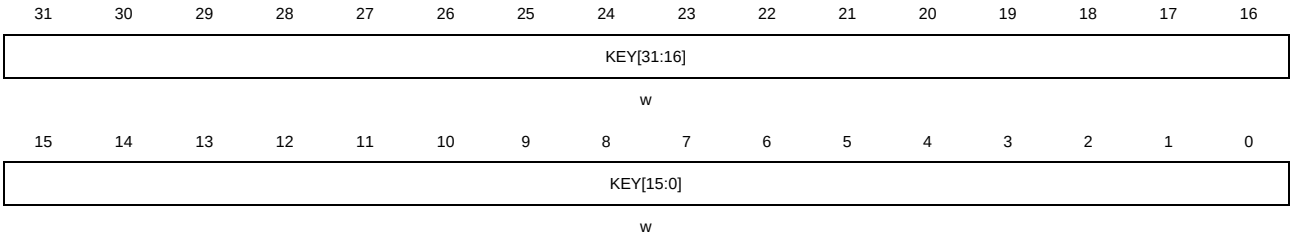


位/位域	名称	描述
31:1	保留	必须保持复位值。
0	I2C2RST	I2C2 单元复位 由软件置 1 或清 0。 0：无作用 1：复位 I2C2 单元

4.3.17. 电源解锁寄存器（RCU_VKEY）

地址偏移：0x100
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

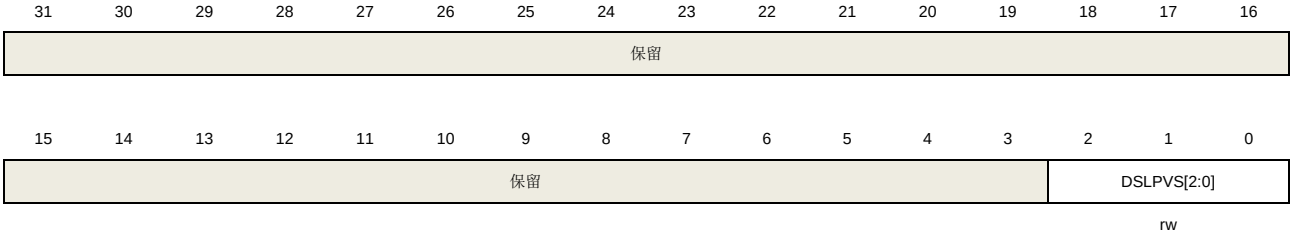


位/位域	名称	描述
31:0	KEY[31:0]	RCU_PDVSEL 和 RCU_DSV 寄存器解锁 这些位只能被软件写，读的话全是 0。只有在向 RCU_VKEY 寄存器写 0x1A2B3C4D 后，RCU_PDVSEL 和 RCU_DSV 寄存器才能被写。

4.3.18. RCU 深度睡眠模式电压寄存器 (RCU_DSV)

地址偏移：0x134
复位值：0x0000 0000

只有在向RCU_VKEY寄存器写0x1A2B3C4D后，RCU_DSV寄存器才能被写。
该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

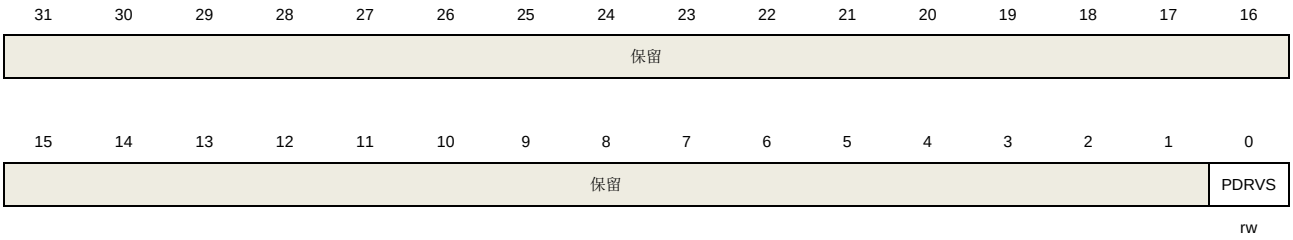


位/位域	名称	描述
31:3	保留	必须保持复位值。
2:0	DSLPVS[2:0]	深度睡眠模式电压选择 这些位由软件置位和清除。 000：在深度睡眠模式下内核电压为 1.2V 001：在深度睡眠模式下内核电压为 1.1V 010：在深度睡眠模式下内核电压为 1.0V 011：在深度睡眠模式下内核电压为 0.9V 100~111：保留

4.3.19. RCU 掉电电压选择寄存器（RCU_PDVSEL）

地址偏移：0x138
复位值：0x0000 0000

只有在向RCU_VKEY 寄存器写0x1A2B3C4D后，RCU_PDVSEL寄存器才能被写。
该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。



位	位域	说明
31:1	保留	必须保持复位值。
0	PDRVS	掉电电压选择 由软件置 1 或清 0。 0：掉电电压为 2.6V

1：掉电电压为 1.8V

5. 中断和事件控制器（EXTI）

5.1. 简介

Cortex®-M3 集成了嵌套式矢量型中断控制器（Nested Vectored Interrupt Controller（NVIC））来实现高效的异常和中断处理。NVIC 实现了低延迟的异常和中断处理，以及电源管理控制。它和内核是紧密耦合的。更多关于 NVIC 的说明请参考《Cortex®-M3 技术参考手册》。

EXTI（中断/事件控制器）包括 23 个相互独立的边沿检测电路并且能够向处理器内核产生中断请求或唤醒事件。EXTI 有三种触发类型：上升沿触发、下降沿触发和双边沿触发。EXTI 中的每一个边沿检测电路都可以独立配置和屏蔽。

5.2. 主要特性

- Cortex®-M3系统异常；
- 多达52种可屏蔽的外设中断；
- 4位中断优先级配置位—16个中断优先等级；
- 高效的中断处理；
- 支持异常抢占和咬尾中断；
- 将系统从省电模式唤醒；
- EXTI中有多达23个相互独立的边沿检测电路；
- 3种触发类型：上升沿触发，下降沿触发和任意沿触发；
- 软件中断或事件触发；
- 可配置的触发源。

5.3. 中断功能描述

ARM Cortex-M3 处理器和嵌套式矢量型中断控制器（NVIC）在处理（Handler）模式下对所有异常进行优先级区分以及处理。当异常发生时，系统自动将当前处理器工作状态压栈，在执行完中断服务子程序（ISR）后自动将其出栈。

取向量是和当前工作态压栈并行进行的，从而提高了中断入口效率。处理器支持咬尾中断，可实现背靠背中断，大大削减了反复切换工作态所带来的开销。*表 5-1. Cotrex®-M3 中的 NVIC 异常类型*和*表 5-2. 中断向量表*列出了所有的异常类型。

表 5-1. Cotrex®-M3 中的 NVIC 异常类型

异常类型	向量编号	优先级（a）	向量地址	描述
-	0	-	0x0000_0000	保留
复位	1	-3	0x0000_0004	复位
NMI	2	-2	0x0000_0008	不可屏蔽中断
硬件故障	3	-1	0x0000_000C	各种硬件级别的故障
存储器管理	4	可编程设置	0x0000_0010	存储器管理

异常类型	向量编号	优先级（a）	向量地址	描述
总线故障	5	可编程设置	0x0000_0014	预取指故障，存储器访问故障
用法故障	6	可编程设置	0x0000_0018	未定义的指令或非法状态
-	7-10	-	0x0000_001C - 0x0000_002B	保留
服务调用	11	可编程设置	0x0000_002C	通过 SWI 指令实现系统的服务调用
调试监控	12	可编程设置	0x0000_0030	调试监控器
-	13	-	0x0000_0034	保留
挂起服务	14	可编程设置	0x0000_0038	可挂起的系统服务请求
系统节拍	15	可编程设置	0x0000_003C	系统节拍定时器

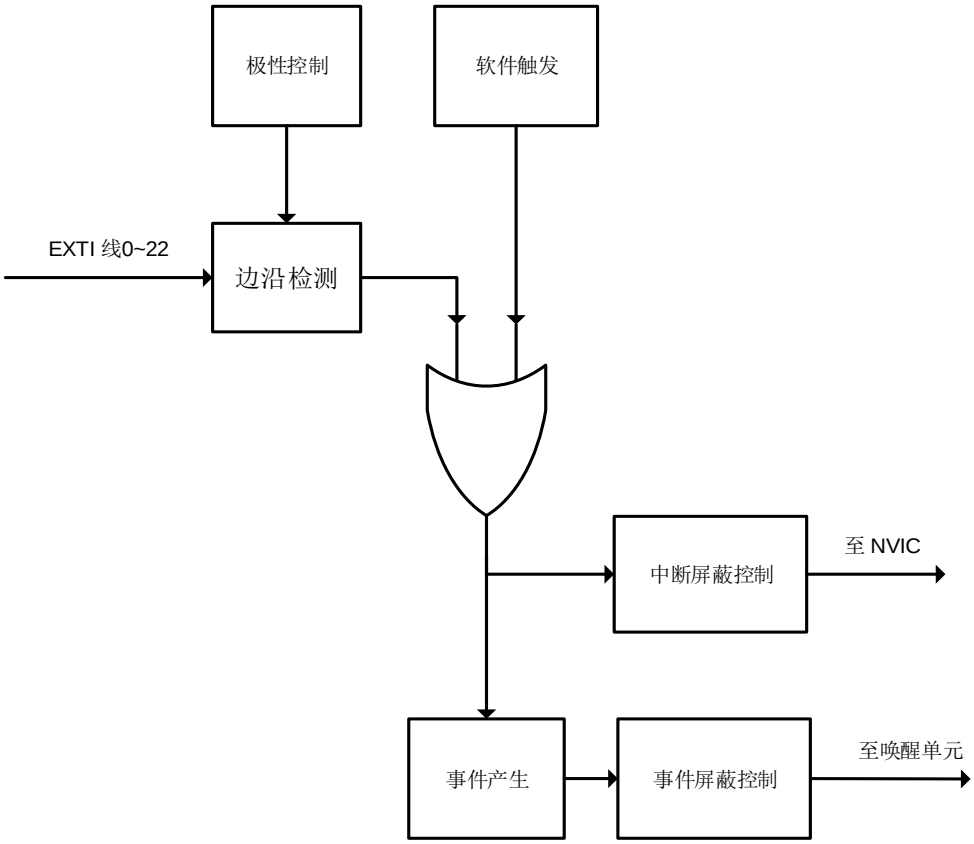
表 5-2.中断向量表

中断编号	向量编号	外设中断描述	向量地址
IRQ 0	16	窗口看门狗定时器中断	0x0000_0040
IRQ 1	17	连接到 EXTI 线的 LVD 中断	0x0000_0044
IRQ 2	18	RTC 全局中断	0x0000_0048
IRQ 3	19	FMC 全局中断	0x0000_004C
IRQ 4	20	RCU 全局中断	0x0000_0050
IRQ 5	21	EXTI 线 0-1 中断	0x0000_0054
IRQ 6	22	EXTI 线 2-3 中断	0x0000_0058
IRQ 7	23	EXTI 线 4-15 中断	0x0000_005C
IRQ 8	24	TSI 全局中断	0x0000_0060
IRQ 9	25	DMA 通道 0 全局中断	0x0000_0064
IRQ 10	26	DMA 通道 1-2 全局中断	0x0000_0068
IRQ 11	27	DMA 通道 3-4 全局中断	0x0000_006C
IRQ 12	28	ADC 和 CMP0-1 中断	0x0000_0070
IRQ 13	29	TIMER0 中止，更新，触发和换相中断	0x0000_0074
IRQ 14	30	TIMER0 捕获比较中断	0x0000_0078
IRQ 15	31	TIMER1 全局中断	0x0000_007C
IRQ 16	32	TIMER2 全局中断	0x0000_0080
IRQ 17	33	TIMER5 和 DAC 全局中断	0x0000_0084
IRQ 18	34	保留	0x0000_0088
IRQ 19	35	TIMER13 全局中断	0x0000_008C
IRQ 20	36	TIMER14 全局中断	0x0000_0090
IRQ 21	37	TIMER15 全局中断	0x0000_0094
IRQ 22	38	TIMER16 全局中断	0x0000_0098
IRQ 23	39	I2C0 事件中断	0x0000_009C
IRQ 24	40	I2C1 事件中断	0x0000_00A0
IRQ 25	41	SPI0 全局中断	0x0000_00A4
IRQ 26	42	SPI1 全局中断	0x0000_00A8

中断编号	向量编号	外设中断描述	向量地址
IRQ 27	43	USART0 全局中断	0x0000_00AC
IRQ 28	44	USART1 全局中断	0x0000_00B0
IRQ 29	45	保留	0x0000_00B4
IRQ 30	46	CEC 全局中断	0x0000_00B8
IRQ 31	47	保留	0x0000_00BC
IRQ 32	48	I2C0 错误中断	0x0000_00C0
IRQ 33	49	保留	0x0000_00C4
IRQ 34	50	I2C1 错误中断	0x0000_00C8
IRQ 35	51	I2C2 事件中断	0x0000_00CC
IRQ 36	52	I2C2 错误中断	0x0000_00D0
IRQ 37	53	USBD 低优先级中断	0x0000_00D4
IRQ 38	54	USBD 高优先级中断	0x0000_00D8
IRQ 39-41	55-57	保留	0x0000_00DC-0x0000_00E4
IRQ 42	58	连接到 EXTI 线的 USBD 唤醒中断	0x0000_00E8
IRQ 43-47	59-63	保留	0x0000_00EC-0x0000_00FC
IRQ 48	64	DMA 通道 5-6 全局中断	0x0000_0100
IRQ 49-50	65-66	保留	0x0000_0104-0x0000_0108
IRQ 51	67	SPI2 全局中断	0x0000_010C

5.4. 外部中断及事件（EXTI）结构框图

图 5-1. EXTI 框图



5.5. 外部中断及事件功能概述

EXTI 包含多达 23 个相互独立的边沿检测电路并且可以向处理器产生中断请求或事件唤醒。EXTI 提供 3 种触发类型：上升沿触发、下降沿触发和任意沿触发。EXTI 中每个边沿检测电路都可以分别予以配置或屏蔽。

EXTI 触发源包括来自 I/O 管脚的 16 根线以及来自内部模块的 8 根线，具体细节*表 5-3. EXTI 触发源*。通过配置 SYSCFG 模块的 SYSCFG_EXTISSx 寄存器，所有的 GPIO 管脚都可能选作 EXTI 的触发源，具体细节请参考*系统配置寄存器（SYSCFG）*。

除了中断，EXTI 还可以向处理器提供事件信号。Cortex®-M3 内核完全支持等待中断（WFI），等待事件（WFE）和发送事件（SEV）指令。芯片内部有一个唤醒中断控制器（WIC），用户可以放心的让处理器和 NVIC 进入功耗极低的休眠模式，由 WIC 来识别中断和事件以及判断优先级。当某些预期的事件发生时，EXTI 能唤醒处理器及整个系统，例如一个特定的 I/O 管脚电平翻转或者 RTC 闹钟动作。

硬件触发

硬件触发被用来检测外部或内部信号的电压变化。软件需要按如下步骤配置来使用这项功能：

1. 根据应用需要配置 SYSCFG 模块中的 EXTI 触发源；
2. 配置 EXTI_RTEN 寄存器和 EXTI_FTEN 寄存器以使能相应引脚的上升沿或下降沿检测（软件应当同时配置引脚对应的 RTENx 和 FTENx 位以检测该引脚上升沿和下降沿的变化）；
3. 通过配置引脚对应的 EXTI_INTEN 或 EXTI_EVEN 位，使能中断或事件；
4. EXTI 开始检测被配置的引脚上的电平变化，当这些引脚上期望的变化被检测到时，使能的中断或事件将被触发。如果为中断触发，则对应的 PD 位将立刻被置 1；如果为事件触发，则对应的 PD 位不被置 1。软件需要响应该中断或事件并清除相应 PDx 位。

软件触发

按照如下步骤软件也可以触发 EXTI 中断或事件：

1. 配置对应的 EXTI_INTEN 或 EXTI_EVEN 位使能中断或事件；
2. 配置 EXTI_SWIEV 寄存器的对应 SWIEVx 位，使能的中断或事件将被立即触发。如果为中断触发，则对应的 PD 位将立刻被置 1；如果为事件触发，则对应的 PD 位不被置 1。软件需要响应该中断或事件并清除相应 PDx 位。

表 5-3. EXTI 触发源

EXTI 线编号	触发源
0	PA0 / PB0 / PC0 / PF0
1	PA1 / PB1 / PC1 / PF1
2	PA2 / PB2 / PC2 / PD2
3	PA3 / PB3 / PC3
4	PA4 / PB4 / PC4 / PF4
5	PA5 / PB5 / PC5 / PF5
6	PA6 / PB6 / PC6 / PF6
7	PA7 / PB7 / PC7 / PF7
8	PA8 / PB8 / PC8
9	PA9 / PB9 / PC9
10	PA10 / PB10 / PC10
11	PA11 / PB11 / PC11
12	PA12 / PB12/ PC12
13	PA13 / PB13 / PC13
14	PA14 / PB14 / PC14
15	PA15 / PB15 / PC15
16	LVD
17	RTC 闹钟
18	USBD 唤醒
19	RTC 干预和时间戳
20	保留
21	比较器 0 输出
22	比较器 1 输出
23	保留
24	保留

EXTI 线编号	触发源
25	USART0 唤醒
26	保留
27	CEC 唤醒

5.6. EXTI 寄存器

EXTI 基地址：0x4001 0400

5.6.1. 中断使能寄存器（EXTI_INTEN）

地址偏移：0x00
复位值：0x0F90 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				INTEN27	INTEN26	INTEN25	INTEN24	INTEN23	INTEN22	INTEN21	INTEN20	INTEN19	INTEN18	INTEN17	INTEN16
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTEN15	INTEN14	INTEN13	INTEN12	INTEN11	INTEN10	INTEN9	INTEN8	INTEN7	INTEN6	INTEN5	INTEN4	INTEN3	INTEN2	INTEN1	INTEN0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:28	保留	必须保持复位值。
27:0	INTENx	中断使能位 x（x = 0…27） 0：第 x 线中断被禁用 1：第 x 线中断被使能

5.6.2. 事件使能寄存器（EXTI_EVEN）

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				EVEN27	EVEN26	EVEN25	EVEN24	EVEN23	EVEN22	EVEN21	EVEN20	EVEN19	EVEN18	EVEN17	EVEN16
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVEN15	EVEN14	EVEN13	EVEN12	EVEN11	EVEN10	EVEN9	EVEN8	EVEN7	EVEN6	EVEN5	EVEN4	EVEN3	EVEN2	EVEN1	EVEN0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:28	保留	必须保持复位值。
27:0	EVENx	事件屏蔽控制 x（x = 0...27） 0：第 x 线事件被禁用 1：第 x 线事件被使能

5.6.3. 上升沿触发使能寄存器（EXTI_RTEN）

地址偏移：0x08

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留									RTEN22	RTEN21	保留	RTEN19	RTEN18	RTEN17	RTEN16
									rw	rw		rw		rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTEN15	RTEN14	RTEN13	RTEN12	RTEN11	RTEN10	RTEN9	RTEN8	RTEN7	RTEN6	RTEN5	RTEN4	RTEN3	RTEN2	RTEN1	RTEN0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:23	保留	必须保持复位值。
22:21	RTENx	上升沿触发使能x（x = 21, 22） 0：第x线上升沿触发无效 1：第x线上升沿触发有效（中断/事件）
20	保留	必须保持复位值。
19	RTENx	上升沿触发使能x（x = 19） 0：第x线上升沿触发无效 1：第x线上升沿触发有效（中断/事件）
18	RTENx	上升沿触发使能x（x = 18） 该位仅适用于GD32F150xx产品。 0：第x线上升沿触发无效 1：第x线上升沿触发有效（中断/事件）
17:0	RTENx	上升沿触发使能x(x = 0…17) 0：第x线上升沿触发无效 1：第x线上升沿触发有效（中断/事件）

5.6.4. 下降沿触发使能寄存器（EXTI_FTEN）

地址偏移：0x0C

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留									FTEN22	FTEN21	保留	FTEN19	FTEN18	FTEN17	FTEN16
									rw	rw		rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FTEN15	FTEN14	FTEN13	FTEN12	FTEN11	FTEN10	FTEN9	FTEN8	FTEN7	FTEN6	FTEN5	FTEN4	FTEN3	FTEN2	FTEN1	FTEN0

rw rw rw rw rw rw rw rw rw rw rw rw rw rw rw rw

位/位域	名称	描述
31:23	保留	必须保持复位值。
22:21	FTENx	下降沿触发使能（x = 21, 22） 0：第 x 线下降沿触发无效 1：第 x 线下降沿触发有效（中断/事件）
20	保留	必须保持复位值。
19	FTENx	下降沿触发使能 x（x = 19） 0：第 x 线下降沿触发无效 1：第 x 线下降沿触发有效（中断/事件）
18	FTENx	下降沿触发使能 x = 18） 该位仅适用于 GD32F150xx 产品。 0：第 x 线下降沿触发无效 1：第 x 线下降沿触发有效（中断/事件）
17:0	FTENx	下降沿触发使能（x = 0…17） 0：第 x 线下降沿触发无效 1：第 x 线下降沿触发有效（中断/事件）

5.6.5. 软件中断事件寄存器（EXTI_SWIEV）

地址偏移：0x10
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留									SWIEV22	SWIEV21	保留	SWIEV19	SWIEV18	SWIEV17	SWIEV16
									rw	rw		rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWIEV15	SWIEV14	SWIEV13	SWIEV12	SWIEV11	SWIEV10	SWIEV9	SWIEV8	SWIEV7	SWIEV6	SWIEV5	SWIEV4	SWIEV3	SWIEV2	SWIEV1	SWIEV0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:23	保留	必须保持复位值。
22:21	SWIEVx	中断/事件软件触发（x = 21, 22） 0：禁用 EXTI 线 x 软件中断/事件请求 1：激活 EXTI 线 x 软件中断/事件请求
20	保留	必须保持复位值。
19:0	SWIEVx	中断/事件软件触发（x = 0, 19）

0：禁用 EXTIx 软件中断/事件请求

1：激活 EXTIx 软件中断/事件请求

5.6.6. 挂起寄存器（EXTI_PD）

地址偏移：0x14

复位值：0xXXXX XXXX

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留									PD22	PD21	保留	PD19	PD18	PD17	PD16
									rc_w1	rc_w1		rc_w1	rc_w1	rc_w1	rc_w1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PD15	PD14	PD13	PD12	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位/位域	名称	描述
31:23	保留	必须保持复位值。
22:21	PDx	中断挂起状态（x = 21, 22） 0：EXTI 线 x 没有被触发 1：EXTI 线 x 被触发 对这些位写 1，可将其清 0。
20	保留	必须保持复位值。
19:0	PDx	中断挂起状态（x = 0…19） 0：EXTI 线 x 没有被触发 1：EXTI 线 x 被触发 对这些位写 1，可将其清 0。

6. 通用输入/输出接口（GPIO）

6.1. 简介

最多可支持 55 个通用 I/O 引脚（GPIO），分别为 PA0 ~ PA15，PB0 ~ PB15，PC0 ~ PC15，PD2，PF0，PF1，PF4 ~ PF7，各片上设备用其来实现逻辑输入/输出功能。每个 GPIO 端口有相关的控制和配置寄存器以满足特定应用的需求。

GPIO 端口和其他备用功能（AFs）的备用引脚，在特定的封装下获得最大的的灵活性。GPIO 引脚通过配置相关的寄存器可以用作备用功能输入/输出引脚。

每个 GPIO 引脚可以由软件配置为输出（推挽或开漏）、输入、外设备用功能或者模拟模式。每个 GPIO 引脚都可以配置为上拉、下拉或无上拉/下拉。除模拟模式外，所有的 GPIO 引脚都具备大电流驱动能力。

6.2. 主要特性

- 输入/输出方向控制；
- 施密特触发输入功能使能控制；
- 每个引脚都具有弱上拉/下拉功能；
- 推挽/开漏输出使能控制；
- 置位/复位输出使能；
- 输出驱动速度选择；
- 模拟输入/输出配置；
- 备用功能输入/输出配置；
- 端口锁定配置；

6.3. 功能描述

每个通用I/O端口都可以通过32位控制寄存器（GPIOx_CTL）配置为GPIO输入，GPIO输出，AF功能或模拟模式。引脚AFIO输入/输出是通过AFIO功能使能来选择。当端口配置为输出（GPIO输出或AFIO输出）时，可以通过GPIO输出模式寄存器（GPIOx_OMODE）配置为推挽或开漏模式。输出端口的最大速度可以通过GPIO输出速度寄存器（GPIOx_OSPD）配置。每个端口可以通过GPIO上/下拉寄存器（GPIOx_PUD）配置为浮空（无上拉或下拉），上拉或下拉功能。

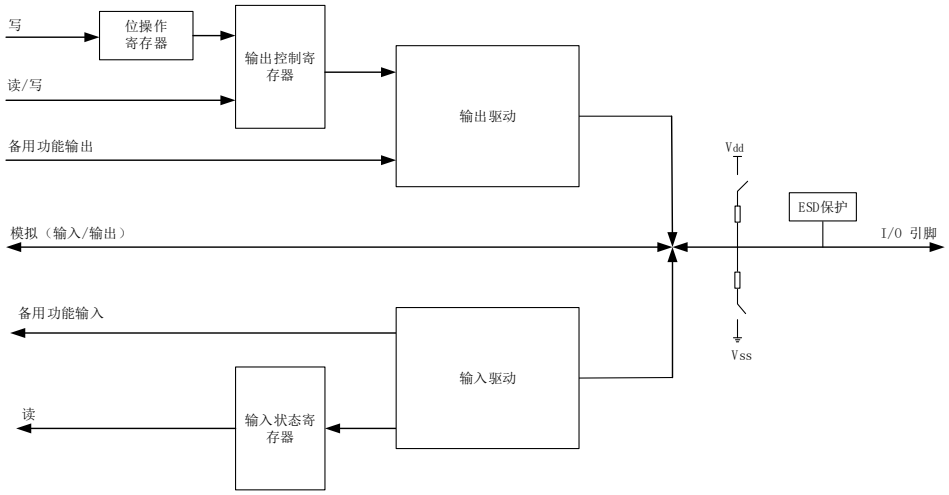
表 6-1. GPIO 配置表

PAD TYPE			CTLn	OMn	PUDn
GPIO 输入	X	悬空	00	X	00
		上拉			01
		下拉			10
GPIO 输出	推挽	悬空	01	0	00
		上拉			01

PAD TYPE			CTLn	OMn	PUDn
		下拉			10
	开漏	悬空			00
		上拉		1	01
		下拉			10
AFIO 输入	X	悬空	10	X	00
		上拉			01
		下拉			10
AFIO 输出	推挽	悬空	10	0	00
		上拉			01
		下拉			10
	开漏	悬空		1	00
		上拉			01
		下拉			10
ANALOG	X	X	11	X	XX

*图6-1. GPIO端口位的基本结构*为标准I/O端口位的基本结构图。

图 6-1. GPIO 端口位的基本结构



6.3.1. GPIO 管脚配置

在复位期间或复位之后，备用功能并未激活，所有 GPIO 端口都被配置成输入浮空模式，这种输入模式禁用上拉（PU）/下拉（PD）电阻。但是复位后，串行线调试为输入 PU/PD 模式。

PA14：SWCLK 为 AF 下拉模式
PA13：SWDIO 为 AF 上拉模式

GPIO 管脚可以配置为输入或输出。并且所有的 GPIO 管脚都有一个内部的弱上拉和弱下拉可以选择。当 GPIO 管脚可配置为输入管脚时，外部管脚上的数据在每个 AHB 时钟周期时都会装载到端口输入状态寄存器（GPIOx_ISTAT）。

当 GPIO 引脚配置为输出引脚，用户可以配置端口的输出速度和选择输出驱动模式：推挽或开漏模式。端口输出控制寄存器（GPIOx_OCTL）的值将会从相应 I/O 引脚上输出。

当需要对 GPIOx_OCTL 进行按位写操作时不需关中断，用户可以通过写‘1’到位操作寄存器（GPIOx_BOP，或用于清 0 的 GPIOx_BC，或用于翻转操作的 GPIOx_TG）修改一位或几位，该过程仅需要一个最小的 AHB 写访问周期，而其他位不受影响。

6.3.2. 备用功能（AF）

当端口配置为AFIO（设置GPIOx_CTL寄存器中的CTLy值为“10”）时，该端口用作外设备用功能。通过配置GPIO备用功能选择寄存器（GPIOx_AFSELy(y=0..1)），每个端口可以配置16个备用功能。端口备用功能分配的详细介绍见芯片数据手册。

6.3.3. 附加功能

有些引脚具有附加功能，它们优先于标准GPIO寄存器中的配置。当用作ADC或DAC附加功能时，引脚必须配置成模拟模式。当引脚用作RTC、WKUPx和振荡器附加功能时，端口类型通过相关的RTC、PMU和RCU寄存器自动设置。当附加功能禁用时，这些端口可用作普通GPIO。

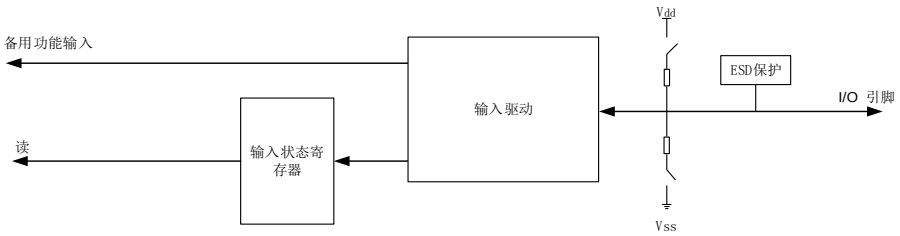
6.3.4. 输入配置

当GPIO引脚配置为输入时：

- 施密特触发输入使能；
- 可选择的弱上拉和下拉电阻；
- 当前I/O引脚上的数据在每个AHB时钟周期都会被采样并存入端口输入状态寄存器；
- 输出缓冲器禁用。

*图6-2. 输入配置的基本结构*是I/O引脚的输入配置。

图 6-2. 输入配置的基本结构



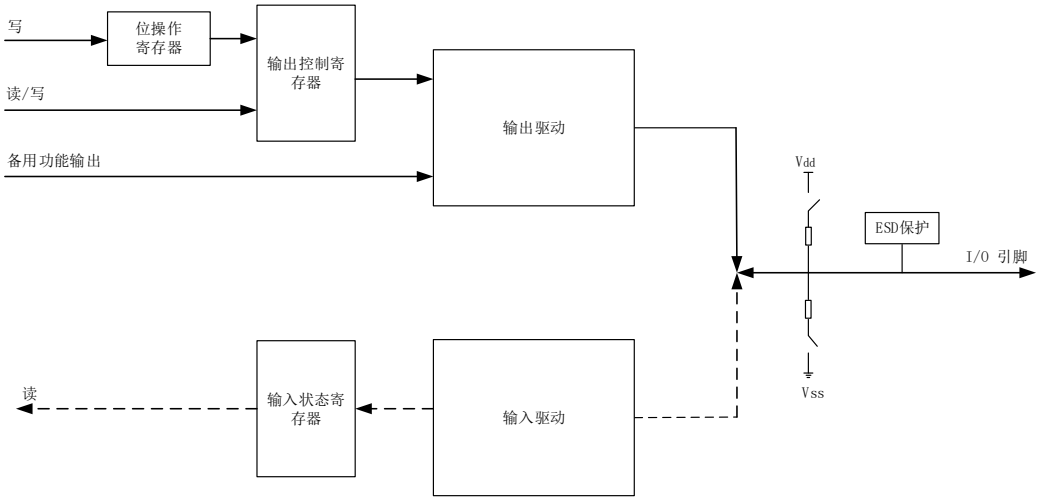
6.3.5. 输出配置

当GPIO配置为输出时：

- 施密特触发输入使能；
- 可选择的弱上拉和下拉电阻；
- 输出缓冲器使能
  - 开漏模式：输出控制寄存器设置为“0”时，相应引脚输出“0”；输出控制寄存器设置为“1”，相应管脚处于高阻状态；
  - 推挽模式：输出控制寄存器设置为“0”时，相应引脚输出“0”；输出控制寄存器设置为“1”，相应引脚输出“1”。
- 在推挽模式下，对端口输出控制寄存器的读访问将返回上次写入的值；
- 在开漏模式下，对端口输入状态寄存器的读访问将返回I/O的状态。

*图 6-3. 输出配置的基本结构*是 I/O 端口位的输出配置。

图 6-3. 输出配置的基本结构



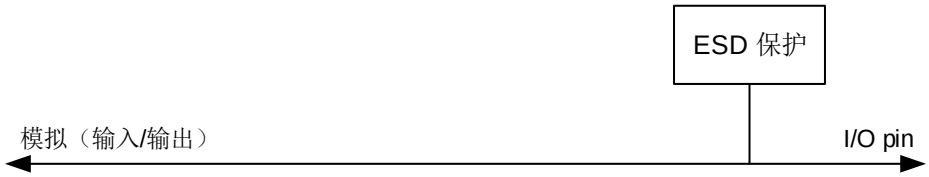
6.3.6. 模拟配置

当GPIO引脚用于模拟模式时：

- 弱上拉和下拉电阻禁用；
- 输出缓冲器禁用；
- 施密特触发输入禁用；
- 读端口输入状态寄存器返回“0”。

*图6-4. 模拟配置的基本结构*是I/O端口的模拟高阻配置。

图 6-4. 模拟配置的基本结构



6.3.7. 备用功能（AF）配置

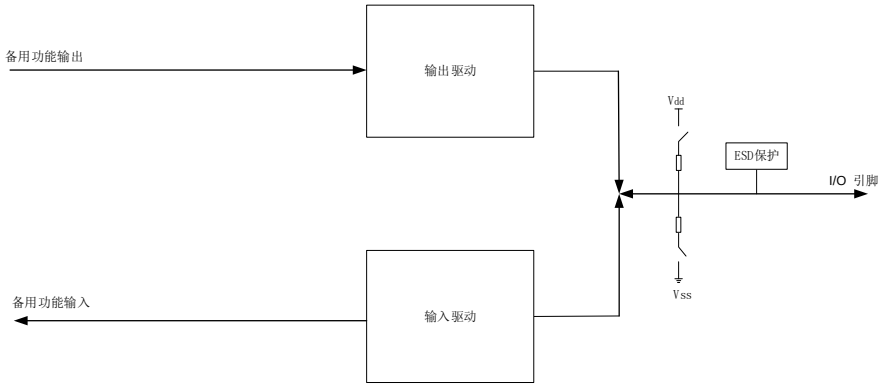
为了适应不同的器件封装，GPIO端口支持软件配置将一些备用功能应用到其他引脚上。

当引脚配置为备用功能时：

- 输出缓冲器启用开漏或者推挽功能；
- 输出缓冲器由外设驱动；
- 施密特触发输入使能；
- 可选择的弱上拉/下拉电阻；
- I/O引脚上的数据在每个AHB时钟周期采样并存入端口输入状态寄存器；
- 在开漏模式下对端口输入状态寄存器进行读操作，将获得I/O口的状态；
- 在推挽输出模式下对端口输出控制寄存器进行读操作，将返回上次写入的值。

*图6-5. 备用功能配置的基本结构*是I/O端口备用功能配置图。

图 6-5. 备用功能配置的基本结构



6.3.8. GPIO 锁定功能

GPIO 的锁定机制可以保护 I/O 端口的配置。

被保护的寄存器有：GPIOx_CTL，GPIOx_OMODE，GPIOx_OSPD，GPIOx_PUD 和 GPIOx_AFSELy(y=0..1)。通过配置 32 位锁定寄存器（GPIOx_LOCK）可以锁定 I/O 端口的配置。当 LOCK 序列已经被应用在相应端口位上，直到下一次复位前，不能改变锁定寄存器的值。建议在电源驱动模块驱动的配置时使用锁定功能。

6.3.9. GPIO 单周期输出翻转功能

通过将GPIOx_TG寄存器中对应的位写1，GPIO可以在一个AHB时钟周期内翻转I/O的输出电平。输出信号的频率可以达到AHB时钟的一半。

6.4. GPIO 寄存器

GPIOA 基地址：0x4800 0000
GPIOB 基地址：0x4800 0400
GPIOC 基地址：0x4800 0800
GPIOD 基地址：0x4800 0C00
GPIOF 基地址：0x4800 1400

6.4.1. 端口控制寄存器（GPIOx_CTL, x=A..D,F）

地址偏移：0x00
复位值：端口 A 0x2800 0000；其他端口 0x0000 0000

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）写访问。
该寄存器只能按字（32 位）读访问。

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16
CTL15[1:0]	CTL14[1:0]	CTL13[1:0]	CTL12[1:0]	CTL11[1:0]	CTL10[1:0]	CTL9[1:0]	CTL8[1:0]
rw	rw	rw	rw	rw	rw	rw	rw

15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
CTL7[1:0]	CTL6[1:0]	CTL5[1:0]	CTL4[1:0]	CTL3[1:0]	CTL2[1:0]	CTL1[1:0]	CTL0[1:0]
rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:30	CTL15[1:0]	Pin 15 配置位 该位由软件置位和清除。 参照 CTL0[1:0]的描述
29:28	CTL14[1:0]	Pin 14 配置位 该位由软件置位和清除。 参照 CTL0[1:0]的描述
27:26	CTL13[1:0]	Pin 13 配置位 该位由软件置位和清除。 参照 CTL0[1:0]的描述
25:24	CTL12[1:0]	Pin 12 配置位 该位由软件置位和清除。 参照 CTL0[1:0]的描述
23:22	CTL11[1:0]	Pin 11 配置位 该位由软件置位和清除。 参照 CTL0[1:0]的描述
21:20	CTL10[1:0]	Pin 10 配置位 该位由软件置位和清除。 参照 CTL0[1:0]的描述

19:18	CTL9[1:0]	Pin 9 配置位 该位由软件置位和清除。 参照 CTL0[1:0]的描述
17:16	CTL8[1:0]	Pin 8 配置位 该位由软件置位和清除。 参照 CTL0[1:0]的描述
15:14	CTL7[1:0]	Pin 7 配置位 该位由软件置位和清除。 参照 CTL0[1:0]的描述
13:12	CTL6[1:0]	Pin 6 配置位 该位由软件置位和清除。 参照 CTL0[1:0]的描述
11:10	CTL5[1:0]	Pin 5 配置位 该位由软件置位和清除。 参照 CTL0[1:0]的描述
9:8	CTL4[1:0]	Pin 4 配置位 该位由软件置位和清除。 参照 CTL0[1:0]的描述
7:6	CTL3[1:0]	Pin 3 配置位 该位由软件置位和清除。 参照 CTL0[1:0]的描述
5:4	CTL2[1:0]	Pin 2 配置位 该位由软件置位和清除。 参照 CTL0[1:0]的描述
3:2	CTL1[1:0]	Pin 1 配置位 该位由软件置位和清除。 参照 CTL0[1:0]的描述
1:0	CTL0[1:0]	Pin 0 配置位 该位由软件置位和清除。 00：输入模式（复位值） 01：GPIO 输出模式 10：备用功能模式 11：模拟模式

6.4.2. 端口输出模式寄存器（GPIOx_OMODE, x=A..D,F）

地址偏移：0x04
复位值：0x0000 0000

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）写访问。

该寄存器只能按字（32 位）读访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OM15	OM14	OM13	OM12	OM11	OM10	OM9	OM8	OM7	OM6	OM5	OM4	OM3	OM2	OM1	OM0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15	OM15	Pin 15 输出模式位 该位由软件置位和清除。 参考 OM0 的描述
14	OM14	Pin 14 输出模式位 该位由软件置位和清除。 参考 OM0 的描述
13	OM13	Pin 13 输出模式位 该位由软件置位和清除。 参考 OM0 的描述
12	OM12	Pin 12 输出模式位 该位由软件置位和清除。 参考 OM0 的描述
11	OM11	Pin 11 输出模式位 该位由软件置位和清除。 参考 OM0 的描述
10	OM10	Pin 10 输出模式位 该位由软件置位和清除。 参考 OM0 的描述
9	OM9	Pin 9 输出模式位 该位由软件置位和清除。 参考 OM0 的描述
8	OM8	Pin 8 输出模式位 该位由软件置位和清除。 参考 OM0 的描述
7	OM7	Pin 7 输出模式位 该位由软件置位和清除。 参考 OM0 的描述
6	OM6	Pin 6 输出模式位 该位由软件置位和清除。

		参考 OM0 的描述
5	OM5	Pin 5 输出模式位 该位由软件置位和清除。 参考 OM0 的描述
4	OM4	Pin 4 输出模式位 该位由软件置位和清除。 参考 OM0 的描述
3	OM3	Pin 3 输出模式位 该位由软件置位和清除。 参考 OM0 的描述
2	OM2	Pin 2 输出模式位 该位由软件置位和清除。 参考 OM0 的描述
1	OM1	Pin 1 输出模式位 该位由软件置位和清除。 参考 OM0 的描述
0	OM0	Pin 0 输出模式位 该位由软件置位和清除。 0：输出推挽模式（复位值） 1：输出开漏模式

6.4.3. 端口输出速度寄存器（GPIOx_OSPD, x=A..D,F）

地址偏移：0x08
复位值：端口 A 0x0C00 0000；其他端口 0x0000 0000

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）写访问。
该寄存器只能按字（32 位）读访问。

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16
OSPD15[1:0]	OSPD14[1:0]	OSPD13[1:0]	OSPD12[1:0]	OSPD11[1:0]	OSPD10[1:0]	OSPD9[1:0]	OSPD8[1:0]
rw	rw	rw	rw	rw	rw	rw	rw
15 14	**13 12**	**11 10**	**9 8**	**7 6**	**5 4**	**3 2**	**1 0**
OSPD7[1:0]	OSPD6[1:0]	OSPD5[1:0]	OSPD4[1:0]	OSPD3[1:0]	OSPD2[1:0]	OSPD1[1:0]	OSPD0[1:0]
rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:30	OSPD15[1:0]	Pin 15 输出最大速度位 该位由软件置位和清除。 参考 OSPD0[1:0]的描述
29:28	OSPD14[1:0]	Pin 14 输出最大速度位 该位由软件置位和清除。

		参考 OSPD0[1:0]的描述
27:26	OSPD13[1:0]	Pin 13 输出最大速度位 该位由软件置位和清除。 参考 OSPD0[1:0]的描述
25:24	OSPD12[1:0]	Pin 12 输出最大速度位 该位由软件置位和清除。 参考 OSPD0[1:0]的描述
23:22	OSPD11[1:0]	Pin 11 输出最大速度位 该位由软件置位和清除。 参考 OSPD0[1:0]的描述
21:20	OSPD10[1:0]	Pin 10 输出最大速度位 该位由软件置位和清除。 参考 OSPD0[1:0]的描述
19:18	OSPD9[1:0]	Pin 9 输出最大速度位 该位由软件置位和清除。 参考 OSPD0[1:0]的描述
17:16	OSPD8[1:0]	Pin 8 输出最大速度位 该位由软件置位和清除。 参考 OSPD0[1:0]的描述
15:14	OSPD7[1:0]	Pin 7 输出最大速度位 该位由软件置位和清除。 参考 OSPD0[1:0]的描述
13:12	OSPD6[1:0]	Pin 6 输出最大速度位 该位由软件置位和清除。 参考 OSPD0[1:0]的描述
11:10	OSPD5[1:0]	Pin 5 输出最大速度位 该位由软件置位和清除。 参考 OSPD0[1:0]的描述
9:8	OSPD4[1:0]	Pin 4 输出最大速度位 该位由软件置位和清除。 参考 OSPD0[1:0]的描述
7:6	OSPD3[1:0]	Pin 3 输出最大速度位 该位由软件置位和清除。 参考 OSPD0[1:0]的描述
5:4	OSPD2[1:0]	Pin 2 输出最大速度位 该位由软件置位和清除。 参考 OSPD0[1:0]的描述
3:2	OSPD1[1:0]	Pin 1 输出最大速度位

该位由软件置位和清除。
参考 OSPD0[1:0]的描述

1:0	OSPD0[1:0]	Pin 0 输出最大速度位 该位由软件置位和清除。 x0：输出最大速度 2M（复位值） 01：输出最大速度 10M 11：输出最大速度 50M

6.4.4. 端口上拉/下拉寄存器（GPIOx_PUD, x=A..D,F）

地址偏移：0x0C
复位值：端口 A 0x2400 0000；其他端口 0x0000 0000

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）写访问。
该寄存器只能按字（32 位）读访问。

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16
PUD15[1:0]	PUD14[1:0]	PUD13[1:0]	PUD12[1:0]	PUD11[1:0]	PUD10[1:0]	PUD9[1:0]	PUD8[1:0]
rw	rw	rw	rw	rw	rw	rw	rw

15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
PUD7[1:0]	PUD6[1:0]	PUD5[1:0]	PUD4[1:0]	PUD3[1:0]	PUD2[1:0]	PUD1[1:0]	PUD0[1:0]
rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:30	PUD15[1:0]	Pin 15 上拉或下拉位 该位由软件置位和清除。 参照 PUD0[1:0]的描述
29:28	PUD14[1:0]	Pin 14 上拉或下拉位 该位由软件置位和清除。 参照 PUD0[1:0]的描述
27:26	PUD13[1:0]	Pin 13 上拉或下拉位 该位由软件置位和清除。 参照 PUD0[1:0]的描述
25:24	PUD12[1:0]	Pin 12 上拉或下拉位 该位由软件置位和清除。 参照 PUD0[1:0]的描述
23:22	PUD11[1:0]	Pin 11 上拉或下拉位 该位由软件置位和清除。 参照 PUD0[1:0]的描述
21:20	PUD10[1:0]	Pin 10 上拉或下拉位 该位由软件置位和清除。 参照 PUD0[1:0]的描述

19:18	PUD9[1:0]	Pin 9 上拉或下拉位 该位由软件置位和清除。 参照 PUD0[1:0]的描述
17:16	PUD8[1:0]	Pin 8 上拉或下拉位 该位由软件置位和清除。 参照 PUD0[1:0]的描述
15:14	PUD7[1:0]	Pin 7 上拉或下拉位 该位由软件置位和清除。 参照 PUD0[1:0]的描述
13:12	PUD6[1:0]	Pin 6 上拉或下拉位 该位由软件置位和清除。 参照 PUD0[1:0]的描述
11:10	PUD5[1:0]	Pin 5 上拉或下拉位 该位由软件置位和清除。 参照 PUD0[1:0]的描述
9:8	PUD4[1:0]	Pin 4 上拉或下拉位 该位由软件置位和清除。 参照 PUD0[1:0]的描述
7:6	PUD3[1:0]	Pin 3 上拉或下拉位 该位由软件置位和清除。 参照 PUD0[1:0]的描述
5:4	PUD2[1:0]	Pin 2 上拉或下拉位 该位由软件置位和清除。 参照 PUD0[1:0]的描述
3:2	PUD1[1:0]	Pin 1 上拉或下拉位 该位由软件置位和清除。 参照 PUD0[1:0]的描述
1:0	PUD0[1:0]	Pin 0 上拉或下拉位 该位由软件置位和清除。 00：悬空模式，无上拉和下拉（复位值） 01：端口上拉模式 10：端口下拉模式 11：保留

6.4.5. 端口输入状态寄存器（GPIOx_ISTAT, x=A..D,F）

地址偏移：0x10
复位值：0x0000 XXXX

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISTAT15	ISTAT14	ISTAT13	ISTAT12	ISTAT11	ISTAT10	ISTAT9	ISTAT8	ISTAT7	ISTAT6	ISTAT5	ISTAT4	ISTAT3	ISTAT2	ISTAT1	ISTAT0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	ISTATy[15:0]	端口输入状态位(y=0..15) 这些位由硬件置位和清除。 0：引脚输入信号为低电平 1：引脚输入信号为高电平

6.4.6. 端口输出控制寄存器（GPIOx_OCTL, x=A..D,F）

地址偏移：0x14
复位值：0x0000 0000

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）写访问。
该寄存器只能按字（32 位）读访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OCTL15	OCTL14	OCTL13	OCTL12	OCTL11	OCTL10	OCTL9	OCTL8	OCTL7	OCTL6	OCTL5	OCTL4	OCTL3	OCTL2	OCTL1	OCTL0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	OCTLy[15:0]	端口输出控制位(y=0..15) 该位由软件置位和清除。 0：引脚输出低电平 1：引脚输出高电平

6.4.7. 端口位操作寄存器（GPIOx_BOP, x=A..D,F）

地址偏移：0x18
复位值：0x0000 0000

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CR15	CR14	CR13	CR12	CR11	CR10	CR9	CR8	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0

w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BOP15	BOP14	BOP13	BOP12	BOP11	BOP10	BOP9	BOP8	BOP7	BOP6	BOP5	BOP4	BOP3	BOP2	BOP1	BOP0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位/位域	名称	描述
31:16	CRy	端口清除位 y(y=0..15) 该位由软件置位和清除。 0：相应的 OCTLy 位没有改变 1：清除相应的 OCTLy 位为 0
15:0	BOPy[15:0]	端口置位位 y(y=0..15) 该位由软件置位和清除。 0：相应的 OCTLy 位没有改变 1：设置相应的 OCTLy 位为 1

6.4.8. 端口配置锁定寄存器（GPIOx_LOCK, x=A,B）

地址偏移：0x1C
复位值：0x0000 0000

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）写访问。
该寄存器只能按字（32 位）读访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															LKK
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LK15	LK14	LK13	LK12	LK11	LK10	LK9	LK8	LK7	LK6	LK5	LK4	LK3	LK2	LK1	LK0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:17	保留	必须保持复位值
16	LKK	锁定键 该位只能通过 Lock Key 写序列置位，始终可读。 0：GPIOx_LOCK 寄存器和端口配置没有锁定 1：直到下一次 MCU 复位前，GPIOx_LOCK 寄存器被锁定 LOCK key 写序列： 写 1→写 0→写 1→读 0→读 1 注意：在 LOCK Key 写序列期间，LK y(y=0..15)的值必须保持。
15:0	LKy	端口锁定位 y(y=0..15) 该位由软件置位和清除。 0：端口配置没有锁定

1：端口配置锁定

6.4.9. 备用功能选择寄存器 0（GPIOx_AFSEL0, x=A,B,C）

地址偏移：0x20

复位值：0x0000 0000

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）写访问。

该寄存器只能按字（32 位）读访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEL7[3:0]				SEL6[3:0]				SEL5[3:0]				SEL4[3:0]			
rw				rw				rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEL3[3:0]				SEL2[3:0]				SEL1[3:0]				SEL0[3:0]			
rw				rw				rw				rw			

位/位域	名称	描述
31:28	SEL7[3:0]	Pin 7 选择备用功能 该位由软件置位和清除。 参照 SEL0 [3:0]的描述
27:24	SEL6[3:0]	Pin 6 选择备用功能 该位由软件置位和清除。 参照 SEL0 [3:0]的描述
23:20	SEL5[3:0]	Pin 5 选择备用功能 该位由软件置位和清除。 参照 SEL0 [3:0]的描述
19:16	SEL4[3:0]	Pin 4 选择备用功能 该位由软件置位和清除。 参照 SEL0 [3:0]的描述
15:12	SEL3[3:0]	Pin 3 选择备用功能 该位由软件置位和清除。 参照 SEL0 [3:0]的描述
11:8	SEL2[3:0]	Pin 2 选择备用功能 该位由软件置位和清除。 参照 SEL0 [3:0]的描述
7:4	SEL1[3:0]	Pin 1 选择备用功能 该位由软件置位和清除。 参照 SEL0 [3:0]的描述
3:0	SEL0[3:0]	Pin 0 选择备用功能 该位由软件置位和清除。 0000：选择 AF0 功能（复位值）

0001：选择 AF1 功能
0010：选择 AF2 功能
0011：选择 AF3 功能
0100：选择 AF4 功能（Port A，B only）
0101：选择 AF5 功能（Port A，B only）
0110：选择 AF6 功能（Port A，B only）
0111：选择 AF7 功能（Port A，B only）
1000 ~ 1111：保留

6.4.10. 备用功能选择寄存器 1（GPIOx_AFSEL1, x=A,B,C）

地址偏移：0x24
复位值：0x0000 0000

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）写访问。
该寄存器只能按字（32 位）读访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEL15[3:0]				SEL14[3:0]				SEL13[3:0]				SEL12[3:0]			
rw				rw				rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEL11[3:0]				SEL10[3:0]				SEL9[3:0]				SEL8[3:0]			
rw				rw				rw				rw			

位/位域	名称	描述
31:28	SEL15[3:0]	Pin 15 选择备用功能 该位由软件置位和清除。 参照 SEL8[3:0]的描述
27:24	SEL14[3:0]	Pin 14 选择备用功能 该位由软件置位和清除。 参照 SEL8[3:0]的描述
23:20	SEL13[3:0]	Pin 13 选择备用功能 该位由软件置位和清除。 参照 SEL8[3:0]的描述
19:16	SEL12[3:0]	Pin 12 选择备用功能 该位由软件置位和清除。 参照 SEL8[3:0]的描述
15:12	SEL11[3:0]	Pin 11 选择备用功能 该位由软件置位和清除。 参照 SEL8[3:0]的描述
11:8	SEL10[3:0]	Pin 10 选择备用功能 该位由软件置位和清除。

		参照 SEL8[3:0] 的描述
7:4	SEL9[3:0]	Pin 9 选择备用功能 该位由软件置位和清除。 参照 SEL8[3:0]的描述
3:0	SEL8[3:0]	Pin 8 选择备用功能 该位由软件置位和清除。 0000：选择 AF0 功能（复位值） 0001：选择 AF1 功能 0010：选择 AF2 功能 0011：选择 AF3 功能 0100：选择 AF4 功能（Port A，B only） 0101：选择 AF5 功能（Port A，B only） 0110：选择 AF6 功能（Port A，B only） 0111：选择 AF7 功能（Port A，B only） 1000 ~ 1111：保留

6.4.11. 位清除寄存器（GPIOx_BC, x=A..D,F）

地址偏移：0x28

复位值：0x0000 0000

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CR15	CR14	CR13	CR12	CR11	CR10	CR9	CR8	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CRy	端口清除位 y(y=0..15) 该位由软件置位和清除。 0：相应 OCTLy 位没有改变 1：清除相应的 OCTLy 位

7. 循环冗余校验管理管理单元（CRC）

7.1. 简介

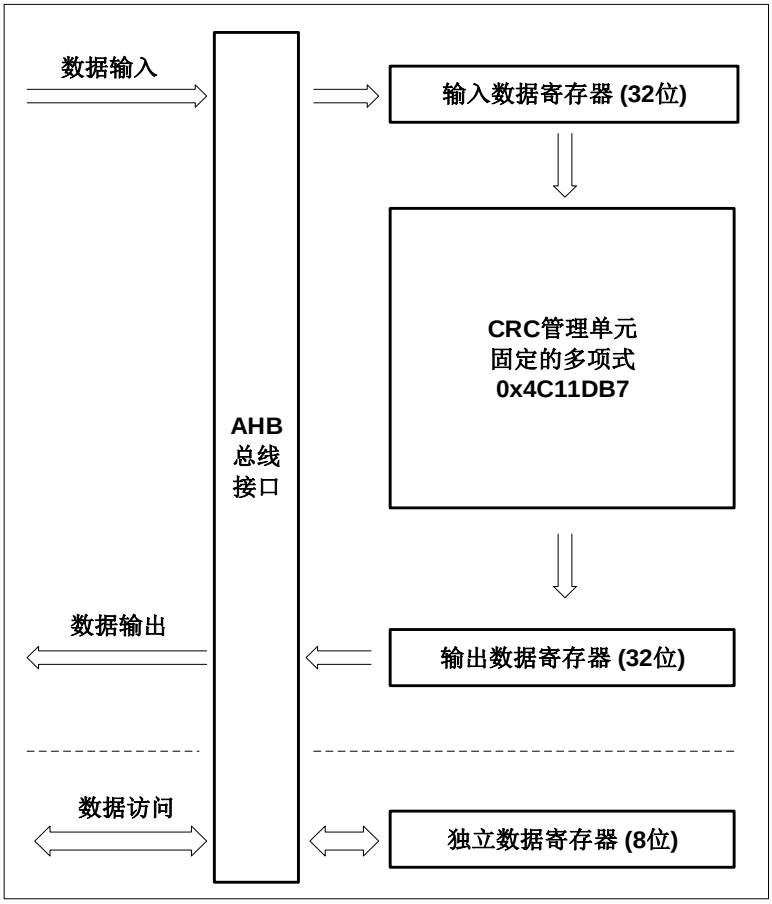
循环冗余校验码是一种用在数字网络和存储设备上的差错校验码，可以校验原始数据的偶然误差。

CRC 管理单元能用固定的多项式来计算 32/16/8 位的 CRC 校验码。

7.2. 主要特性

- 支持8/16/32位数据输入；
- 对于8/16/32位的输入数据长度，计算周期分别为1/2/4个AHB时钟周期；
- 固定的计算多项式：0x4C11DB7
  X32 + X26 + X23 + X22 + X16 + X12 + X11 + X10 +X8 + X7 + X5 + X4 + X2+ X +1
  该32位CRC多项式是一个与以太网计算多项式相同的多项式。
- CRC复位后，用户可以配置计算初值；
- 配有与计算无关的独立8位寄存器，可以供其他任何外设使用；
- 用户可设置的计算初值，可使计算更灵活。

图 7-1. CRC 管理单元框图



7.3. 功能描述

- CRC管理单元可以用来计算32位的原始数据，CRC_DATA寄存器接收原始数据并存储计算结果；
  如果不通过软件设置CRC_CTL寄存器的方式来清除CRC_DATA寄存器，新输入的原始数据将会基于前一次CRC_DATA寄存器中的结果进行计算；
  对于32/16/8位的数据长度，CRC的计算分别要花费4/2/1个AHB的时钟周期。在此期间，因为32位输入缓存的原因，AHB总线将不会被挂起。
- 此模块提供了一个8位的独立寄存器CRC_FDATA，CRC_FDATA与CRC计算无关，任何时候都可以进行独立的读写操作；
- 逆序功能可以交换输入输出数据的位序。
  输入数据可选择三种逆序形式。
  以原始数据0x3456CDEF为例：
  1) 按字节逆序：
  32位数据被分成四组，组内完成颠倒。逆序后的数据为：0x2C6AB3F7
  2) 按半字逆序：
  32位数据被分成两组，组内完成颠倒。逆序后的数据为：0x6A2CF7B3
  3) 按字逆序：
  32位数据被分成一组，组内完成颠倒，逆序后的数据为：0xF7B36A2C
  对于输出数据来说，逆序形式为按字逆序。
  例如：当REV_O=1，计算结果0x3344CCDD将被逆序成0xBB3322CC。
- 多重输入数据大小支持功能可以使用户在组合计算数据上有更大的灵活度。
  例如：6位的输入数据能组合成一个字或一个半字，同时也能被组合成3个半字。
- 用户可配置的初始计算数据功能可让CRC以用户设置值开始计算。

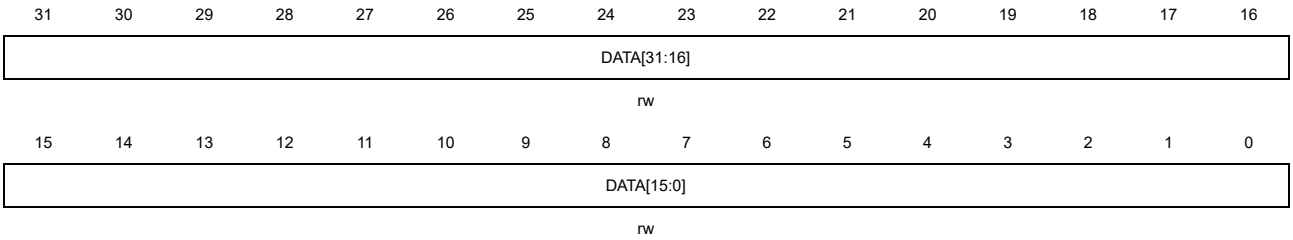
7.4. CRC 寄存器

CRC基地址：0x4002 3000

7.4.1. 数据寄存器 (CRC_DATA)

地址偏移：0x00
复位值：0xFFFF FFFF

该寄存器只能按字(32 位)访问。

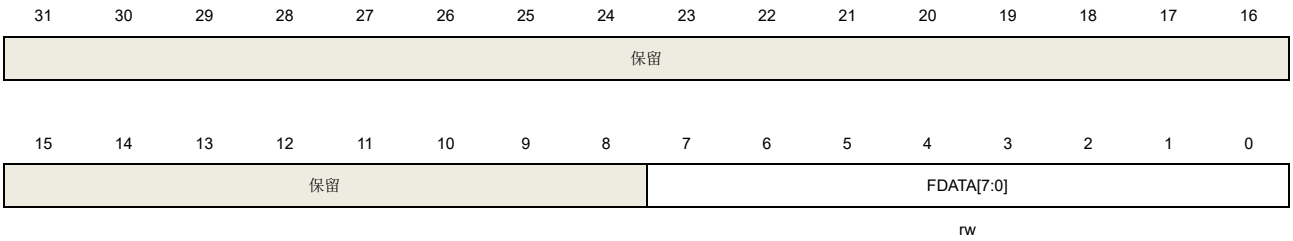


位/位域	名称	描述
31:0	DATA[31:0]	CRC 计算结果位 软件可读可写。 该寄存器用于接收待计算的新数据，直接将其写入即可。刚写入的数据不能被读出来因为读取该寄存器得到的是上次 CRC 计算的结果。

7.4.2. 独立数据寄存器 (CRC_FDATA)

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

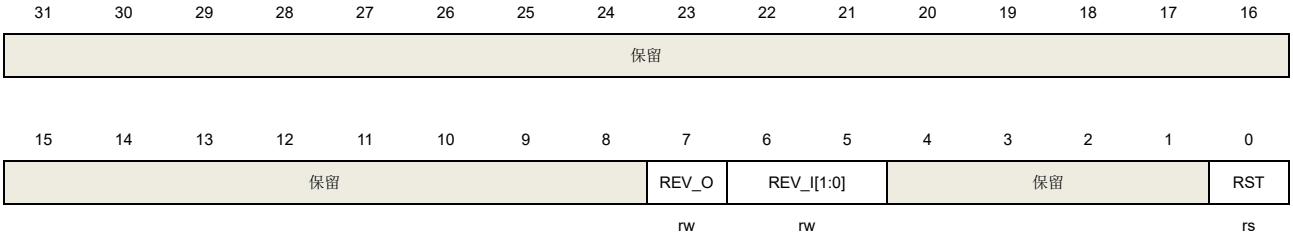


位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	FDATA[7:0]	独立数据寄存器位 软件可读可写。 这些位与 CRC 计算无关。该字节能被任何其他外设用于其他任何目的。该字节不受 CRC_CTL 寄存器的影响。

7.1.1. 控制寄存器 (CRC_CTL)

地址偏移：0x08
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

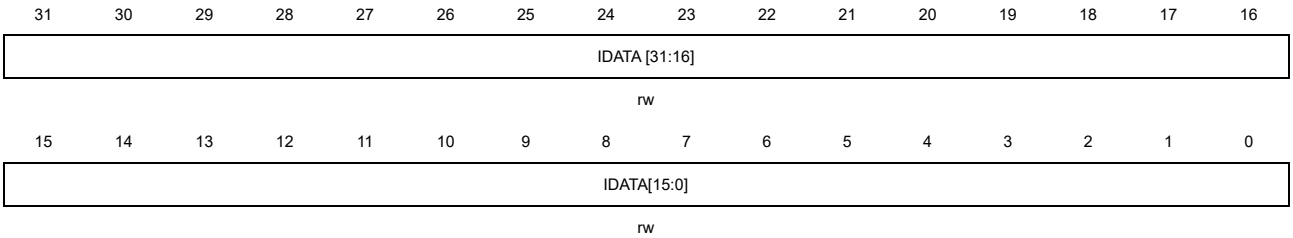


位/位域	名称	描述
31:8	保留	必须保持复位值。
7	REV_O	按位顺序翻转输出数据功能 0：输出数据不翻转 1：输出数据按位顺序翻转
6:5	REV_I[1:0]	翻转输入数据功能 0：输入数据不翻转 1：输入数据按字节翻转 2：输入数据按半字翻转 3：输入数据按字翻转
4:1	保留	必须保持复位值。
0	RST	该位用来复位 CRC_DATA 寄存器，并将 CRC_DATA 寄存器中的更新到 0xFFFFFFFF 中，然后自动清零。该位对 CRC_FDATA 寄存器没有影响。 软件可读写

7.1.2. 初值寄存器 (CRC_IDATA)

地址偏移：0x10
复位值：0xFFFF FFFF

该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31:0	IDATA[31:0]	配置 CRC 初值 CRC_CTL 寄存器的 RST 位置位后，CRC_DATA 寄存器的值将被更新为此寄存器的值。

8. DMA 控制器（DMA）

8.1. 简介

DMA 控制器提供了一种硬件的方式在外设和存储器之间或者存储器和存储器之间传输数据，而无需 CPU 的介入，从而使 CPU 可以专注在处理其他系统功能上。DMA 控制器有 7 个通道。每个通道都是专门用来处理一个或多个外设的存储器访问请求的。DMA 控制器内部实现了一个仲裁器，用来仲裁多个 DMA 请求的优先级。

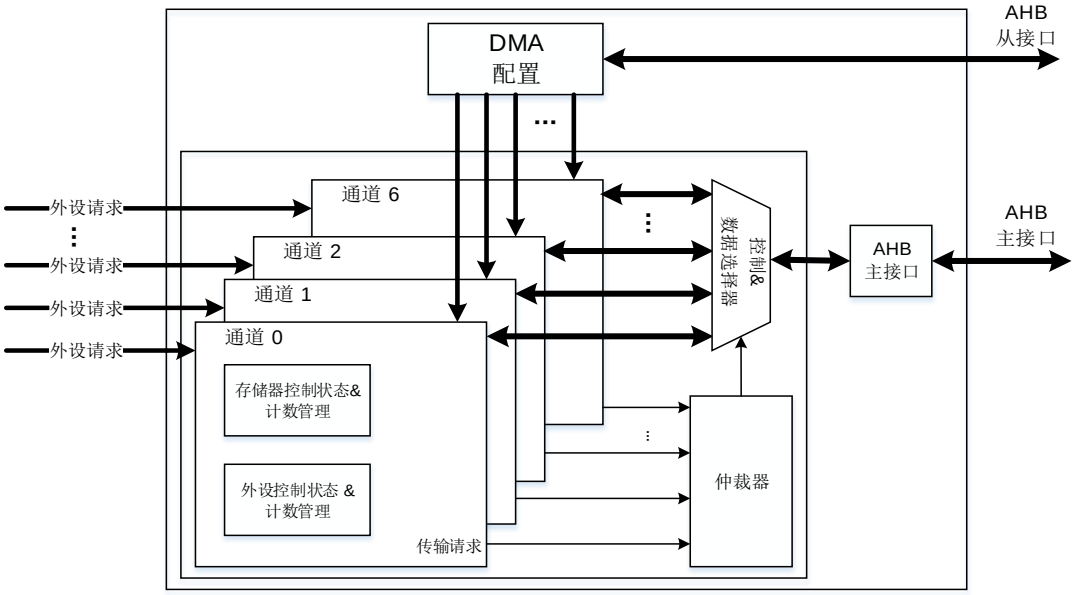
DMA 控制器和 Cortex®-M3 内核共享系统总线。当 DMA 和 CPU 访问同样的地址空间时，DMA 访问可能会阻挡 CPU 访问系统总线几个总线周期。总线矩阵中实现了循环仲裁算法来分配 DMA 与 CPU 的访问权，它可以确保 CPU 得到至少一半的系统总线带宽。

8.2. 主要特性

- 传输数据长度可编程配置，最大到 65536；
- 7 个通道，并且每个通道都可配置；
- AHB 和 APB 外设，片上闪存和 SRAM 都可以作为访问的源端和目的端；
- 每个通道连接固定的硬件 DMA 请求；
- 支持软件优先级（低、中、高、极高）和硬件优先级（通道号越低，优先级越高）；
- 存储器和外设的数据传输宽度可配置：字节，半字，字；
- 存储器和外设的数据传输支持固定寻址和增量式寻址；
- 支持循环传输模式；
- 支持外设到存储器，存储器到外设，存储器到存储器的数据传输；
- 每个通道有 3 种类型的事件标志和独立的中断，支持中断的使能和清除；
- 支持中断使能和清除。

8.3. 结构框图

图 8-1. DMA 结构框图



由*图 8-1. DMA 结构框图*所示，DMA 控制器由 4 部分组成：

- AHB 从接口配置 DMA；
- AHB 主接口进行数据传输；
- 仲裁器进行 DMA 请求的优先级管理；
- 数据处理和计数。

8.4. 功能描述

8.4.1. DMA 操作

DMA 传输分为两步操作：从源地址读取数据，之后将读取的数据存储到目的地址。DMA 控制器基于 DMA_CHxPADDR、DMA_CHxMADDR、DMA_CHxCTL 寄存器的值计算下一次操作的源/目的地址。DMA_CHxCNT 寄存器用于控制传输的次数。DMA_CHxCTL 寄存器的 PWIDTH 和 MWIDTH 位域决定每次发送和接收的字节数（字节/半字/字）。

假设 DMA_CHxCNT 寄存器的值为 4，并且 PNAGA 和 MNAGA 位均置位。结合 PWIDTH 和 MWIDTH 的各种配置，DMA 传输的操作详见*表 8-1. DMA 传输操作*。

表 8-1. DMA 传输操作

传输宽度		传输操作	
源	目标	源	目标
32 bits	32 bits	1: Read B3B2B1B0[31:0] @0x0 2: Read B7B6B5B4[31:0] @0x4 3: Read BBBAB9B8[31:0] @0x8 4: Read BFBEBDBC[31:0] @0xC	1: Write B3B2B1B0[31:0] @0x0 2: Write B7B6B5B4[31:0] @0x4 3: Write BBBAB9B8[31:0] @0x8 4: Write BFBEBDBC[31:0] @0xC
32 bits	16 bits	1: Read B3B2B1B0[31:0] @0x0 2: Read B7B6B5B4[31:0] @0x4 3: Read BBBAB9B8[31:0] @0x8 4: Read BFBEBDBC[31:0] @0xC	1: Write B1B0[7:0] @0x0 2: Write B5B4[7:0] @0x2 3: Write B9B8[7:0] @0x4 4: Write BDBC[7:0] @0x6
32 bits	8 bits	1: Read B3B2B1B0[31:0] @0x0 2: Read B7B6B5B4[31:0] @0x4 3: Read BBBAB9B8[31:0] @0x8 4: Read BFBEBDBC[31:0] @0xC	1: Write B0[7:0] @0x0 2: Write B4[7:0] @0x1 3: Write B8[7:0] @0x2 4: Write BC[7:0] @0x3
16 bits	32 bits	1: Read B1B0[15:0] @0x0 2: Read B3B2[15:0] @0x2 3: Read B5B4[15:0] @0x4 4: Read B7B6[15:0] @0x6	1: Write 0000B1B0[31:0] @0x0 2: Write 0000B3B2[31:0] @0x4 3: Write 0000B5B4[31:0] @0x8 4: Write 0000B7B6[31:0] @0xC
16 bits	16 bits	1: Read B1B0[15:0] @0x0 2: Read B3B2[15:0] @0x2 3: Read B5B4[15:0] @0x4 4: Read B7B6[15:0] @0x6	1: Write B1B0[15:0] @0x0 2: Write B3B2[15:0] @0x2 3: Write B5B4[15:0] @0x4 4: Write B7B6[15:0] @0x6
16 bits	8 bits	1: Read B1B0[15:0] @0x0 2: Read B3B2[15:0] @0x2 3: Read B5B4[15:0] @0x4 4: Read B7B6[15:0] @0x6	1: Write B0[7:0] @0x0 2: Write B2[7:0] @0x1 3: Write B4[7:0] @0x2 4: Write B6[7:0] @0x3
8 bits	32 bits	1: Read B0[7:0] @0x0 2: Read B1[7:0] @0x1 3: Read B2[7:0] @0x2 4: Read B3[7:0] @0x3	1: Write 000000B0[31:0] @0x0 2: Write 000000B1[31:0] @0x4 3: Write 000000B2[31:0] @0x8 4: Write 000000B3[31:0] @0xC
8 bits	16 bits	1: Read B0[7:0] @0x0 2: Read B1[7:0] @0x1 3: Read B2[7:0] @0x2 4: Read B3[7:0] @0x3	1, Write 00B0[15:0] @0x0 2, Write 00B1[15:0] @0x2 3, Write 00B2[15:0] @0x4 4, Write 00B3[15:0] @0x6
8 bits	8 bits	1: Read B0[7:0] @0x0 2: Read B1[7:0] @0x1 3: Read B2[7:0] @0x2 4: Read B3[7:0] @0x3	1, Write B0[7:0] @0x0 2, Write B1[7:0] @0x1 3, Write B2[7:0] @0x2 4, Write B3[7:0] @0x3

DMA_CHxCNT寄存器的CNT位域必须在CHEN位置位前被配置，其控制传输的次数。在传输过程中，CNT位域的值表示还有多少次数据传输将被执行。

将 DMA_CHxCTL 寄存器的 CHEN 位清零，可以停止 DMA 传输。

- 若 CHEN 位被清零时 DMA 传输还未完成，重新使能 CHEN 位将分两种情况：
  - 在重新使能 DMA 通道前，未对该通道的相关寄存器进行操作，则 DMA 将继续完成上次的传输；
  - 在重新使能 DMA 通道前，对任意相关寄存器进行了操作，则 DMA 将开始一次新的传输。
- 若清零 CHEN 位时，DMA 传输已经完成，之后未对任意寄存器进行操作前便使能 DMA 通道，则不会触发任何 DMA 传输。

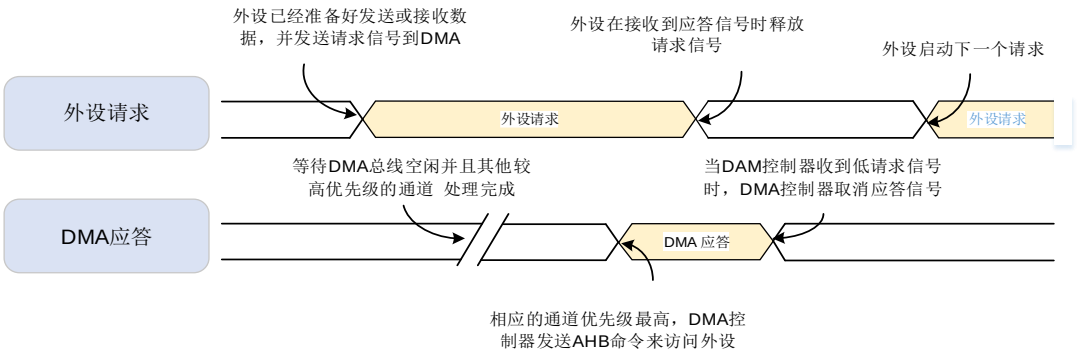
8.4.2. 外设握手

为了保证数据的有效传输，DMA控制器中引入了外设和存储器的握手机制，包括请求信号和应答信号：

- 请求信号：由外设发出，表明外设已经准备好发送或接收数据；
- 应答信号：由 DMA 控制器响应，表明 DMA 控制器已经发送 AHB 命令去访问外设。

*图8-2. 握手机制*中详细描述了DMA控制器与外设之间的握手机制。

图 8-2. 握手机制



8.4.3. 仲裁

当DMA控制器在同一时间接收到多个外设请求时，仲裁器将根据外设请求的优先级来决定响应哪一个外设请求。优先级包括软件优先级和硬件优先级，优先级规则如下：

- 软件优先级：分为4级，低，中，高和极高。可以通过寄存器DMA_CHxCTL的PRIO位域来配置；
- 硬件优先级：当通道具有相同的软件优先级时，编号低的通道优先级高。例：通道0和通道2配置为相同的软件优先级时，通道0的优先级高于通道2。

8.4.4. 地址生成

存储器和外设都独立的支持两种地址生成算法：固定模式和增量模式。寄存器DMA_CHxCTL的PNAGA和MNAGA位用来设置存储器和外设的地址生成算法。

在固定模式中，地址一直固定为初始化的基地址（DMA_CHxPADDR，DMA_CHxMADDR）。

在增量模式中，下一次传输数据的地址是当前地址加1（或者2，4），这个值取决于数据传输宽度。

8.4.5. 循环模式

循环模式用来处理连续的外设请求(如ADC扫描模式)。将DMA_CHxCTL寄存器的CMEN位置位可以使能循环模式。

在循环模式中，当每次DMA传输完成后，CNT值会被重新载入，且传输完成标志位会被置1。DMA会一直响应外设的请求，直到通道使能位（DMA_CHxCTL寄存器的CHEN位）被清0。

8.4.6. 存储器到存储器模式

将DMA_CHxCTL寄存器的M2M位置位可以使能存储器到存储器模式。在此模式下，DMA通道传输数据时不依赖外设的请求信号。一旦DMA_CHxCTL寄存器的CHEN位被置1，DMA通道就立即开始传输数据，直到DMA_CHxCNT寄存器达到0，DMA通道才会停止。

8.4.7. 通道配置

要启动一次新的 DMA 数据传输，建议遵循以下步骤进行操作：

1. 读取 CHEN 位，如果为 1（通道已使能），清零该位。当 CHEN 为 0 时，请按照下列步骤配置 DMA 开始新的传输；
2. 配置 DMA_CHxCTL 寄存器的 M2M 及 DIR 位，选择传输模式；
3. 配置 DMA_CHxCTL 寄存器的 CMEN 位，选择是否使能循环模式；
4. 配置 DMA_CHxCTL 寄存器的 PRIO 位域，选择该通道的软件优先级；
5. 通过 DMA_CHxCTL 寄存器配置存储器和外设的传输宽度以及存储器和外设地址生成算法；
6. 通过 DMA_CHxCTL 寄存器配置传输完成中断，半传输完成中断，传输错误中断的使能位；
7. 通过 DMA_CHxPADDR 寄存器配置外设基地址；
8. 通过 DMA_CHxMADDR 寄存器配置存储器基地址；
9. 通过 DMA_CHxCNT 寄存器配置数据传输总量；
10. 将 DMA_CHxCTL 寄存器的 CHEN 位置 1，使能 DMA 通道。

8.4.8. 中断

每个DMA通道都有一个专用的中断。中断事件有三种类型：传输完成，半传输完成和传输错误。

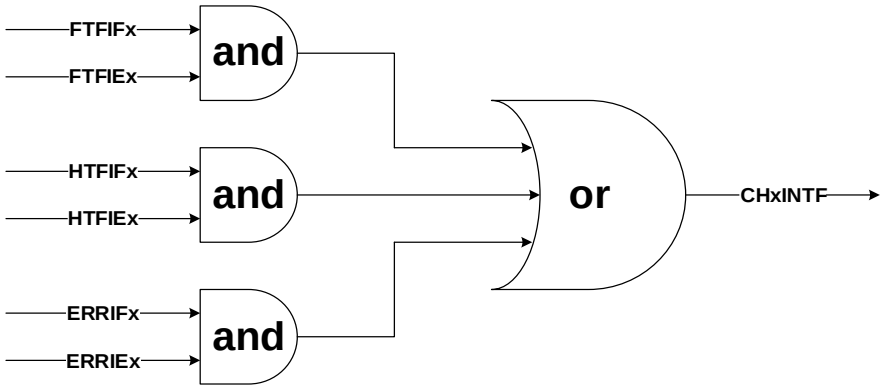
每一个中断事件在DMA_INTF寄存器中有专用的标志位，在DMA_INTC寄存器中有专用的清除位，在DMA_CHxCTL寄存器中有专用的使能位。*表8-2. 中断事件*描述了其对应关系。

表 8-2. 中断事件

中断事件	标志位	清除位	使能位
	DMA_INTF	DMA_INTC	DMA_CHxCTL
传输完成	FTFIF	FTFIFC	FTFIE
传输半完成	HTFIF	HTFIFC	HTFIE
传输错误	ERRIF	ERRIFC	ERRIE

DMA中断逻辑如*表8-3. DMA各通道请求表*所示，任何类型中断使能时，产生了相应中断事件均会产生中断。

图 8-3. DMA 中断逻辑图



注意：“x”表示通道数（对应x=0…6）

8.4.9. DMA 请求映射

多个外设请求被映射到同一个 DMA 通道。这些请求信号在经过逻辑或后进入 DMA。详情可见 *图 8-4. DMA 请求映射*。通过配置对应外设的寄存器，每个外设的请求均可以独立的开启或关闭。用户必须确保同一时间，在同一个通道上仅有一个外设的请求被开启。*表 8-3. DMA 各通道请求表*列举了 DMA 的每个通道所支持的外设请求。

图 8-4. DMA 请求映射

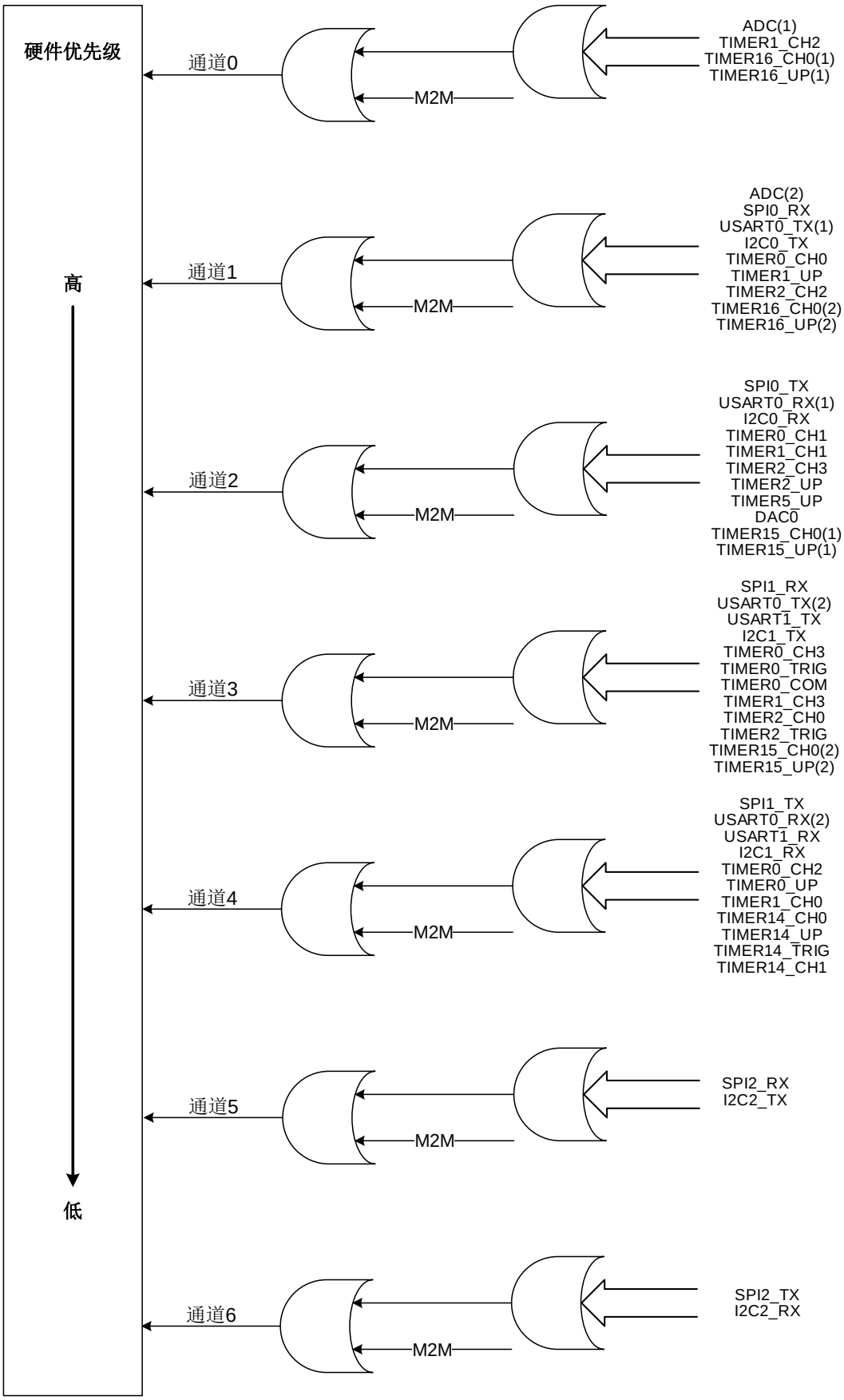

硬件优先级
高
低
通道0
M2M
ADC(1)
TIMER1_CH2
TIMER16_CH0(1)
TIMER16_UP(1)
通道1
M2M
ADC(2)
SPI0_RX
USART0_TX(1)
I2C0_TX
TIMER0_CH0
TIMER1_UP
TIMER2_CH2
TIMER16_CH0(2)
TIMER16_UP(2)
通道2
M2M
SPI0_TX
USART0_RX(1)
I2C0_RX
TIMER0_CH1
TIMER1_CH1
TIMER2_CH3
TIMER2_UP
TIMER5_UP
DAC0
TIMER15_CH0(1)
TIMER15_UP(1)
通道3
M2M
SPI1_RX
USART0_TX(2)
USART1_TX
I2C1_TX
TIMER0_CH3
TIMER0_TRIG
TIMER0_COM
TIMER1_CH3
TIMER2_CH0
TIMER2_TRIG
TIMER15_CH0(2)
TIMER15_UP(2)
通道4
M2M
SPI1_TX
USART0_RX(2)
USART1_RX
I2C1_RX
TIMER0_CH2
TIMER0_UP
TIMER1_CH0
TIMER14_CH0
TIMER14_UP
TIMER14_TRIG
TIMER14_CH1
通道5
M2M
SPI2_RX
I2C2_TX
通道6
M2M
SPI2_TX
I2C2_RX

表8-3. DMA各通道请求表

外设	通道 0	通道 1	通道 2	通道 3	通道 4	通道 5	通道 6
ADC	ADC(1)	ADC(2)	•	•	•	•	•
SPI/I2S	•	SPI/I2S0_RX	SPI/I2S0_TX	SPI1_RX	SPI1_TX	SPI2_RX	SPI2_TX
USART	•	USART0_TX(1)	USART0_RX(1)	USART0_TX(2) USART1_TX	USART0_RX(2) USART1_RX	•	•
I²C	•	I2C0_TX	I2C0_RX	I2C1_TX	I2C1_RX	I²C2_TX	I²C2_RX
TINER0	•	TIMER0_CH0	TIMER0_CH1	TIMER0_CH3 TIMER0_TRIG TIMER0_COM	TIMER0_CH2 TIMER0_UP	•	•
TIMER1	TIMER1_CH2	TIMER1_UP	TIMER1_CH1	TIMER1_CH3	TIMER1_CH0	•	•
TIMER2	•	TIMER2_CH2	TIMER2_CH3 TIMER2_UP	TIMER2_CH0 TIMER2_TRIG	•	•	•
TIMER5/ DAC	•	•	TIMER5_UP DAC	•	•	•	•
TIMER14	•	•	•	•	TIMER14_CH0 TIMER14_UP TIMER14_TRIG TIMER14_CH1	•	•
TIMER15	•	•	TIMER15_CH0(1) TIMER15_UP(1)	TIMER15_CH0(2) TIMER15_UP(2)	•	•	•
TIMER16	TIMER16_CH0(1) TIMER16_UP(1)	TIMER16_CH0(2) TIMER16_UP(2)	•	•	•	•	•

1. 当 SYSCFG_CFGR0 寄存器的相应重映射位被清零时，请求被映射到该通道；
2. 当 SYSCFG_CFGR0 寄存器的相应重映射位被置位时，请求被映射到该通道。

8.5. DMA 寄存器

DMA 基地址：0x4002 0000

8.5.1. 中断标志位寄存器 (DMA_INTF)

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				ERRIF6	HTFIF6	FTFIF6	GIF6	ERRIF5	HTFIF5	FTFIF5	GIF5	ERRIF4	HTFIF4	FTFIF4	GIF4
				r	r	r	r	r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRIF3	HTFIF3	FTFIF3	GIF3	ERRIF2	HTFIF2	FTFIF2	GIF2	ERRIF1	HTFIF1	FTFIF1	GIF1	ERRIF0	HTFIF0	FTFIF0	GIF0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位/位域	名称	描述
31:28	保留	必须保持复位值。
27/23/19/ 15/11/7/3	ERRIFx	通道 x 错误标志位(x=0…6) 硬件置位，软件写 DMA_INTC 相应位为 1 清零 0：通道 x 未发生传输错误 1：通道 x 发生传输错误
26/22/18/ 14/10/6/2	HTFIFx	通道 x 半传输完成标志位(x=0…6) 硬件置位，软件写 DMA_INTC 相应位为 1 清零 0：通道 x 半传输未完成 1：通道 x 半传输完成
25/21/17/ 13/9/5/1	FTFIFx	通道 x 传输完成标志位(x=0…6) 硬件置位，软件写 DMA_INTC 相应位为 1 清零 0：通道 x 传输未完成 1：通道 x 传输完成
24/20/16/ 12/8/4/0	GIFx	通道 x 全局中断标志位(x=0…6) 硬件置位，软件写 DMA_INTC 相应位为 1 清零 0：通道 x ERRIF，HTFIF 或 FTFIF 标志位未置位 1：通道 x 至少发生 ERRIF，HTFIF 或 FTFIF 之一置位

8.5.2. 中断标志位清除寄存器 (DMA_INTC)

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				ERRIFC6	HTFIFC6	FTFIFC6	GIFC6	ERRIFC5	HTFIFC5	FTFIFC5	GIFC5	ERRIFC4	HTFIFC4	FTFIFC4	GIFC4
				w	w	w	w	w	w	w	w	w	w	w	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRIFC3	HTFIFC3	FTFIFC3	GIFC3	ERRIFC2	HTFIC2	FTFIFC2	GIFC2	ERRIFC1	HTFIFC1	FTFIFC1	GIFC1	ERRIFC0	HTFIFC0	FTFIFC0	GIFC0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位/位域	名称	描述
31:28	保留	必须保持复位值。
27/23/19/ 15/11/7/3	ERRIFCx	清除通道 x(x=0…6)的错误标志位 0：无影响 1：清零 DMA_INTF 寄存器的 ERRIFx 位
26/22/18/ 14/10/6/2	HTFIFCx	清除通道 x(x=0…6)的半传输完成标志位 0：无影响 1：清零 DMA_INTF 寄存器的 HTFIFx 位
25/21/17/ 13/9/5/1	FTFIFCx	清除通道 x(x=0…6)的传输完成标志位 0：无影响 1：清零 DMA_INTF 寄存器的 FTFIFx 位
24/20/16/ 12/8/4/0	GIFCx	清除通道 x(x=0…6)的全局中断标志位 0：无影响 1：清零 DMA_INTF 寄存器的 GIFx，ERRIFx，HTFIFx 和 FTFIFx 位

8.5.3. 通道 x 控制寄存器 (DMA_CHxCTL)

x = 0...6, x为通道序号

地址偏移：0x08 + 0x14 × x

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	M2M	PRIO[1:0]		MWIDTH[1:0]		PWIDTH[1:0]		MNAGA	PNAGA	CMEN	DIR	ERRIE	HTFIE	FTFIE	CHEN
	rw	rw		rw		rw		rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:15	保留	必须保持复位值。
14	M2M	存储器到存储器模式 软件置位和清零

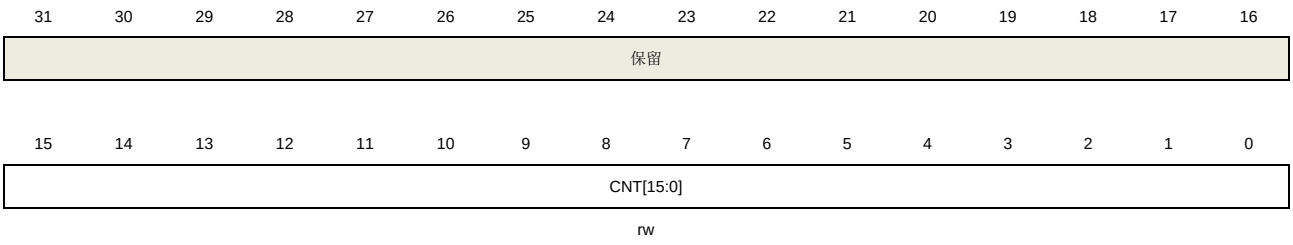
		0：禁止存储器到存储器模式 1：使能存储器到存储器模式 CHEN 位为 1 时，该位不能被配置
13:12	PRIO[1:0]	软件优先级 软件置位和清零 00：低 01：中 10：高 11：极高 CHEN 位为 1 时，该位域不能被配置
11:10	MWIDTH[1:0]	存储器的传输数据宽度 软件置位和清零 00：8-bit 01：16-bit 10：32-bit 11：保留 CHEN 位为 1 时，该位域不能被配置
9:8	PWIDTH[1:0]	外设的传输数据宽度 软件置位和清零 00：8-bit 01：16-bit 10：32-bit 11：保留 CHEN 位为 1 时，该位域不能被配置
7	MNAGA	存储器的地址生成算法 软件置位和清零 0：固定地址模式 1：增量地址模式 CHEN 位为 1 时，该位不能被配置
6	PNAGA	外设的地址生成算法 软件置位和清零 0：固定地址模式 1：增量地址模式 CHEN 位为 1 时，该位不能被配置
5	CMEN	循环模式使能 软件置位和清零 0：禁止循环模式 1：使能循环模式 CHEN 位为 1 时，该位不能被配置
4	DIR	传输方向 软件置位和清零

		0：从外设读出并写入存储器 1：从存储器读出并写入外设 CHEN 位为 1 时，该位不能被配置
3	ERRIE	通道错误中断使能位 软件置位和清零 0：禁止通道错误中断 1：使能通道错误中断
2	HTFIE	通道半传输完成中断使能位 软件置位和清零 0：禁止通道半传输完成中断 1：使能通道半传输完成中断
1	FTFIE	通道传输完成中断使能位 软件置位和清零 0：禁止通道传输完成中断 1：使能通道传输完成中断
0	CHEN	通道使能 软件置位和清零 0：禁止该通道 1：使能该通道

8.5.4. 通道 x 计数寄存器 (DMA_CHxCNT)

x = 0...6, x为通道序号
地址偏移：0x0C + 0x14 × x
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CNT[15:0]	传输计数 CHEN 位为 1 时，该位域不能被配置 该寄存器标明还有多少数据等待被传输。一旦通道使能，该寄存器为只读的，并在每个 DMA 传输之后值减 1。如果该寄存器的值为 0，无论通道开启与否，都不会有数据传输。如果该通道工作在循环模式下，一旦通道的传输任务完成，该寄存器会

被自动重装载为初始设置值。

8.5.5. 通道 x 外设基地址寄存器 (DMA_CHxPADDR)

x = 0...6, x为通道序号

地址偏移：0x10 + 0x14 × x

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

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

位/位域	名称	描述
31:0	PADDR[31:0]	外设基地址 CHEN 位为 1 时，该位域不能被配置 当 PWIDTH 位域的值为 01 (16-bit),，PADDR[0]被忽略，访问自动与 16 位地址对齐。 当 PWIDTH 位域的值为 10 (32-bit)，PADDR [1:0]被忽略，访问自动与 32 位地址对齐。

8.5.6. 通道 x 存储器基地址寄存器 (DMA_CHxMADDR)

x = 0...6, x为通道序号

地址偏移：0x14 + 0x14 × x

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

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

位/位域	名称	描述
31:0	MADDR[31:0]	存储器基地址 CHEN 位为 1 时，该位域不能被配置 当 MWIDTH 位域的值为 01(16-bit)时，MADDR[0]被忽略，访问自动与 16 位地址对齐。

当 MWIDTH 位域的值为 10 (32-bit)时，MADDR [1:0]被忽略，访问自动与 32 位地址对齐。

9. 调试（DBG）

9.1. 简介

GD32F1x0系列产品提供了各种各样的调试，跟踪和测试功能。这些功能通过ARM CoreSight™组件的标准配置和链状连接的TAP控制器来实现的。调试和跟踪功能集成在ARM Cortex-M3内核中。调试系统支持串行调试（SWD）和跟踪功能。调试和跟踪功能请参考下列文档：

- Cortex-M3技术参考手册；
- ARM调试接口v5结构规范。

调试系统帮助调试者在低功耗模式下调试以及一些外设调试，包括：TIMER、I2C、RTC、WWDGT和FWDGT。当相应的位被置1，调试系统会在低功耗模式下提供时钟，或者为一些外设保持当前状态，这些外设包括：TIMER、I2C、RTC、WWDGT和FWDGT。

9.2. 串行调试接口简介

调试工具可以通过串行调试接口（SWD）来访问调试功能。

9.2.1. 引脚分配

串行调试（SWD）提供两个引脚的接口：数据输入输出引脚（SWDIO）和时钟引脚（SWCLK）。

调试引脚分配：

PA14： SWCLK
PA13： SWDIO

如果SWD没有使用，这两个引脚均释放作为普通GPIO功能。两个引脚具体配置请参考*GPIO管脚配置*。

9.2.2. JEDEC-106 ID code

Cortex-M3集成了JEDEC-106 ID代码。位于ROM表中，映射地址为0xE00FF000_0xE00FFFFF。

9.3. 调试保持功能描述

9.3.1. 低功耗模式调试支持

当DBG控制寄存器0（DBG_CTL0）的STB_HOLD位置1并且进入待机模式，AHB总线时钟和系统时钟由CK_IRC8M提供，可以在待机模式下调试。当退出待机模式后，产生系统复位。

当DBG控制寄存器0（DBG_CTL0）的DSLP_HOLD位置1并且进入深度睡眠模式，AHB总线时钟和系统时钟由CK_IRC8M提供，可以在深度睡眠模式下调试。

当DBG控制寄存器0（DBG_CTL0）的SLP_HOLD位置1并且进入睡眠模式，AHB总线时钟没有关闭，可以在睡眠模式下调试。

9.3.2. TIMER, I2C, RTC, WWDGT 和 FWDGT 的外设调试支持

当内核停止，并且DBG控制寄存器0（DBG_CTL0）或DBG控制寄存器1（DBG_CTL1）中的相应位置1。对于不同外设，有不同动作：

对于TIMER外设，TIMER计数器停止并进行调试；

对于I2C外设，SMBUS保持状态并进行调试；

对于RTC外设，计数器停止并进行调试；

对于WWDGT或者FWDGT外设，计数器时钟停止并进行调试。

9.4. DBG 寄存器

DBG基地址：0xE004 2000

9.4.1. ID 寄存器 (DBG_ID)

地址偏移：0x00

该寄存器只能按字(32 位)访问。

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

位/位域	名称	描述
31:0	ID_CODE[15:0]	DBG ID 寄存器 这些位由软件读取，这些位是不变的常数。

9.4.2. 控制寄存器 0 (DBG_CTL0)

地址偏移：0x04
复位值：0x0000 0000，仅上电复位

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				TIMER13_HOLD	保留							TIMER5_HOLD	保留	I2C2_HOLD	I2C1_HOLD
				rw								rw		wr	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2C0_HOLD	保留		TIMER2_HOLD	TIMER1_HOLD	TIMER0_HOLD	WWDGT_HOLD	FWDGT_HOLD	保留					STB_HOLD	DSLP_HOLD	SLP_HOLD
rw			rw	rw	rw	rw	rw						rw	rw	rw

位/位域	名称	描述
31:28	保留	必须保持复位值
27	TIMER13_HOLD	TIMER13 保持位 该位由软件置位和复位。 0：无影响 1：当内核停止时保持定时器 13 计数器不变，用于调试
26:20	保留	必须保持复位值
19	TIMER5_HOLD	TIMER5 保持位

		该位由软件置位和复位。 0：无影响 1：当内核停止时保持定时器 5 计数器不变，用于调试
18	保留	必须保持复位值
17	I2C2_HOLD	I2C2 保持位 该位由软件置位和复位。 0：无影响 1：当内核停止时保持 I2C2 SMBUS 状态不变，用于调试
16	I2C1_HOLD	I2C1 保持位 该位由软件置位和复位。 0：无影响 1：当内核停止时保持 I2C1 SMBUS 状态不变，用于调试
15	I2C0_HOLD	I2C0 保持位 该位由软件置位和复位。 0：无影响 1:当内核停止时保持 I2C0 SMBUS 状态不变，用于调试
14:13	保留	必须保持复位值
12	TIMER2_HOLD	TIMER2 保持位 该位由软件置位和复位。 0：无影响 1：当内核停止时保持定时器 2 计数器不变，用于调试
11	TIMER1_HOLD	TIMER1 保持位 该位由软件置位和复位。 0：无影响 1：当内核停止时保持定时器 1 计数器不变，用于调试
10	TIMER0_HOLD	TIMER0 保持位 该位由软件置位和复位。 0：无影响 1：当内核停止时保持定时器 0 计数器不变，用于调试
9	WWDGT_HOLD	WWDGT 保持位 该位由软件置位和复位。 0：无影响 1：当内核停止时保持 WWDGT counter 不变，用于调试
8	FWDGT_HOLD	FWDGT 保持位 该位由软件置位和复位 0：无影响 1：当内核停止时保持 FWDGT counter 不变，用于调试
7:3	保留	必须保持复位值

2	STB_HOLD	待机模式保持位 该位由软件置位和复位。 0：无影响 1：在待机模式下，系统时钟和 HCLK 由 CK_IRC8M 提供，当退出待机模式时，产生系统复位
1	DSLP_HOLD	深度睡眠模式保持位 该位由软件置位和复位。 0：无影响 1：在深度睡眠模式下，系统时钟和 HCLK 由 CK_IRC8M 提供
0	SLP_HOLD	睡眠模式保持位 该位由软件置位和复位 0：无影响 1：在睡眠模式下，HCLK 继续运行

9.4.3. 控制寄存器 1 (DBG_CTL1)

地址偏移：0x08
复位值：0x0000 0000，仅仅上电复位

该寄存器只能按字(32 位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留													TIMER16_HOLD	TIMER15_HOLD	TIMER14_HOLD
													rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					RTC_HOLD	保留									
					rw										

位/位域	名称	描述
31:19	保留	必须保持复位值
18	TIMER16_HOLD	TIMER16 保持位 该位由软件置位和复位。 0：无影响 1：当内核停止时保持定时器 16 计数器不变，用于调试
17	TIMER15_HOLD	TIMER15 保持位 该位由软件置位和复位。 0：无影响 1：当内核停止时保持定时器 15 计数器不变，用于调试
16	TIMER14_HOLD	TIMER14 保持位 该位由软件置位和复位。 0：无影响

		1：当内核停止时保持定时器 14 计数器不变，用于调试
15:11	保留	必须保持复位值
10	RTC_HOLD	RTC 保持位 该位由软件置位和复位。 0：无影响 1：当内核停止时保持 RTC 计数器不变，用于调试
9:0	保留	必须保持复位值

10. 模拟数字转换器（ADC）

10.1. 简介

MCU片上集成了12位逐次逼近式模数转换器模块（ADC），可以采样来自于16个外部通道、2个内部通道和电池电压（V_{BAT}）通道的模拟信号。这19个ADC采样通道都支持多种运行模式，采样转换后，转换结果可以按照最低有效位对齐或最高有效位对齐的方式保存在相应的数据寄存器中。

10.2. 主要特性

- 高性能：
  - ADC 采样分辨率：12 位分辨率；
  - 前置校准功能；
  - 可编程的采样时间；
  - 数据存储模式：最高有效位对齐和最低有效位对齐；
  - DMA 请求。
- 双时钟域架构(APB时钟和ADC时钟)。
- 模拟输入通道：
  - 16 个外部模拟输入通道；
  - 1 个内部温度传感通道（V_{SENSE}）；
  - 1 个内部参考电压输入通道（V_{REFINT}）；
  - 1 个监测外部 V_{BAT} 供电引脚的内部输入通道。
- 转换开始的触发：
  - 软件方式；
  - 硬件触发。
- 运行模式：
  - 转换单个通道，或者扫描一序列的通道；
  - 单次运行模式，每次触发转换一个选择的输入通道；
  - 连续运行模式，连续转换所选择的输入通道；
  - 间断运行模式。
- 中断的产生：
  - 常规序列转换结束；
  - 模拟看门狗事件。
- 转换结果阈值监测器功能：模拟看门狗。
- 通道输入范围：$V_{SSA} \le V_{IN} \le V_{DDA}$。

10.3. 引脚和内部信号

*图 10-1. ADC 模块框图*给出了 ADC 模块框图。*表 10-1. ADC 内部输入信号*，*表 10-2. ADC 输入引脚定义*给出了 ADC 内部信号和引脚定义。

表 10-1. ADC 内部输入信号

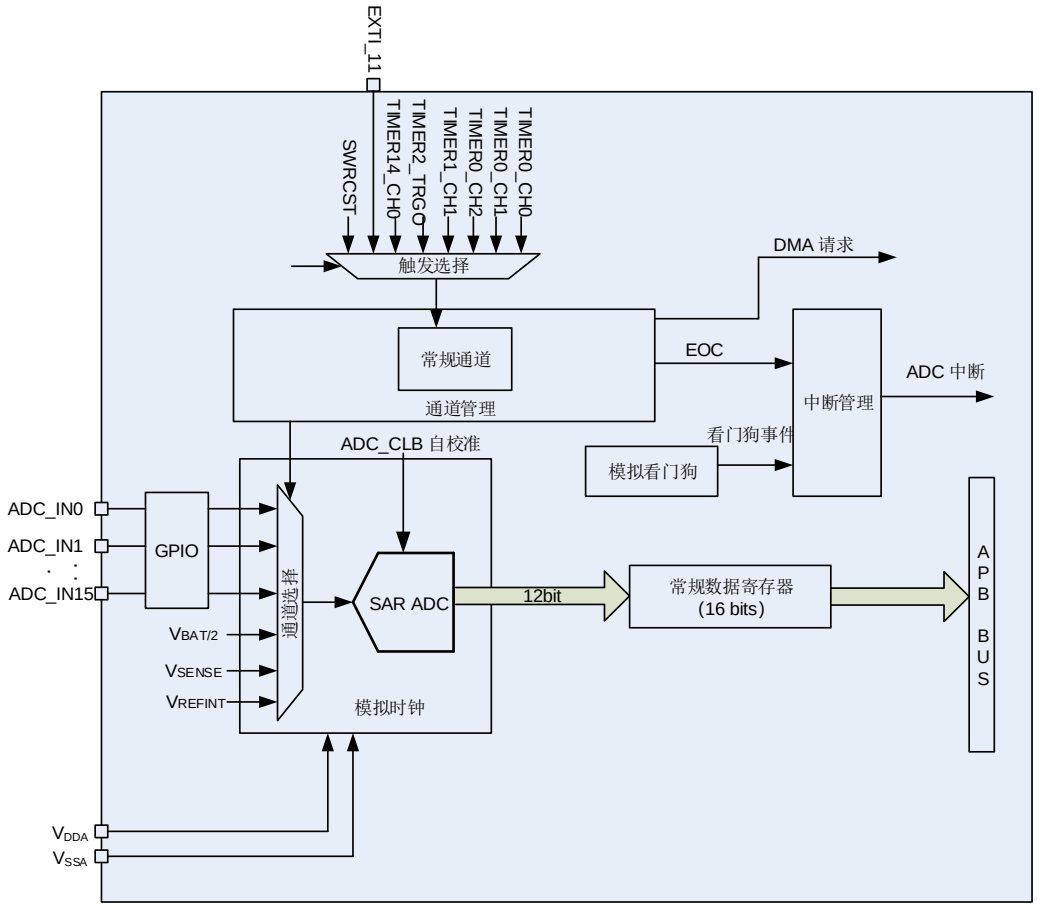
内部信号名称	说明
V_{SENSE}	内部温度传感器输出电压
V_{REFINT}	内部电压参考输出电压
$V_{BAT}/2$	V_{BAT} 引脚输入电压除以 2

表 10-2. ADC 输入引脚定义

名称	说明
V_{DDA}	模拟电源输入等于V_{DD}
V_{SSA}	模拟地，等于 V_{SS}
ADCx_IN [15:0]	多达 16 路外部通道

10.4. 功能说明

图 10-1. ADC 模块框图



10.4.1. 前置校准功能

在前置校准期间，ADC计算一个校准因子，这个系数应用于ADC的内部，直到ADC下次掉电才无效。在校准期间，应用程序不能使用ADC，必须等到ADC校准完成。在A/D开始转换前应执

行校准操作，清除比较器的偏移误差和电容不匹配误差的校准过程需要83个时钟周期。由于生产过程存在中的差异，每一片芯片的这些误差都各不相同。

通过软件设置 CLB=1 启动校准。在校准期间 CLB 位会一直保持 1，一旦校准完成，该位由硬件清 0。

当ADC运行条件改变(V_{DDA}改变是ADC偏移变化的最主要因素，其次是温度的改变)，建议重新做一次校准操作。

内部的模拟校准通过设置ADC_CTL1寄存器的RSTCLB位来重置。

软件校准过程：

1. 确保ADCON=1
2. 延迟 14 个 CK_ADC 以等待 ADC 稳定；
3. 设置RSTCLB (可选的)
4. 设置CLB=1
5. 等待直到CLB=0

10.4.2. 双时钟域架构

除了APB接口时钟，ADC的子模块时钟还可以由ADC时钟提供。ADC时钟和APB时钟异步，并独立于APB时钟。

应用程序能够在低功耗运行时降低PLCK时钟频率，但是ADC仍能保持最佳运行状态。

想要更多ADC时钟产生的信息，可以参考RCU章节*4.2.1简介*部分。

10.4.3. ADCON 使能

ADC_CTL1寄存器中的ADCON位是ADC模块的使能开关。如果ADCON位为0，则ADC模块保持复位状态。为了省电，当ADCON位为0时，ADC模拟子模块将会进入掉电模式。ADC使能后需要等待t_{su}时间后才能采样，t_{su}数值详见芯片数据手册。

10.4.4. 常规序列

通道管理电路可以将采样通道组织成一个序列：常规序列。常规序列支持最多16个通道，每个通道称为常规通道。

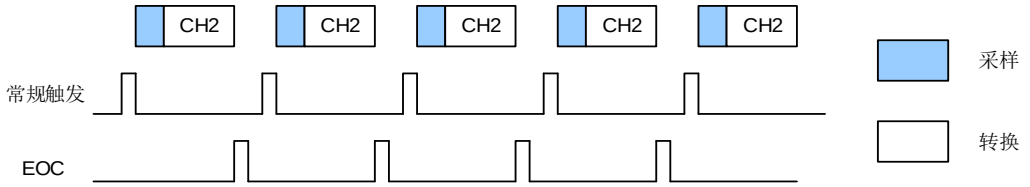
ADC_RSQ0 寄存器的 RL[3:0] 位规定了整个常规序列转换序列的长度。ADC_RSQ0~ADC_RSQ2寄存器规定了常规序列的通道选择。

10.4.5. 运行模式

单次运行模式

单次运行模式下，ADC_RSQ2 寄存器的 RSQ0[4:0]位（常规序列）规定了 ADC 的转换通道。当 ADCON 位被置 1 时，一旦相应的软件触发或者外部触发发生，ADC 就会采样和转换一个通道。

图 10-2. 单次运行模式



常规通道单次转换结束后，转换数据将被存放于 ADC_RDATA 寄存器中，EOC 位置 1。如果 EOCIE 位被置 1，将产生一个中断。

常规序列单次运行模式的软件流程：

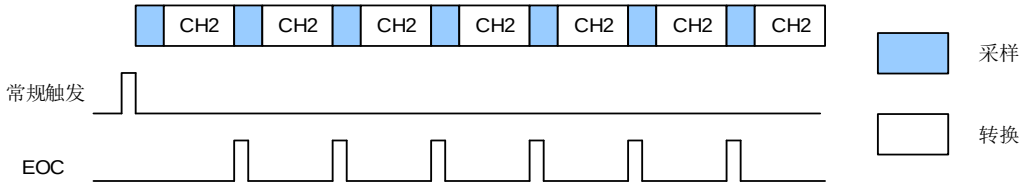
1. 确保ADC_CTL0寄存器中的DISRC位和SM位以及ADC_CTL1寄存器中的CTN位为0；
2. 根据模拟通道编号来配置RSQ0；
3. 配置ADC_SAMPTx寄存器；
4. 如果有需要，可以配置ADC_CTL1寄存器中的ETERC位和ETSRC位；
5. 设置SWRCST位，或者为常规序列产生一个外部触发信号；
6. 等待EOC标志位置1；
7. 延迟一个CK_ADC后，从ADC_RDATA寄存器中读ADC转换结果；
8. 写0清除EOC标志位。

注意：当EOC置1后，需延迟一个CK_ADC再读取ADC转换结果。

连续运行模式

当 ADC_CTL1 寄存器中的 CTN 位置 1 时，可以使能连续运行模式。该模式下，ADC 执行由 RSQ0[4:0]规定的转换通道。当 ADCON 位被置 1，一旦相应的软件触发或者外部触发产生，ADC 就会采样和转换规定的通道。转换数据保存在 ADC_RDATA 寄存器中。

图 10-3. 连续运行模式



常规序列连续运行模式的软件流程：

1. 设置ADC_CTL1寄存器中的CTN位为1；
2. 根据模拟通道编号来配置RSQ0；
3. 配置ADC_SAMPTx寄存器；
4. 如果有需要，配置ADC_CTL1寄存器的ETERC和ETSRC位；
5. 设置SWRCST位，或者给常规序列产生一个外部触发信号；
6. 等待EOC标志位置1；
7. 延迟一个CK_ADC后，从ADC_RDATA寄存器中读ADC转换结果；
8. 写0清除EOC标志位；
9. 如果需要进行连续转换，重复步骤6~8。

注意：当EOC置1后，需延迟一个CK_ADC再读取ADC转换结果。

可以使用DMA来传输转换数据，不需循环查询EOC标志位：

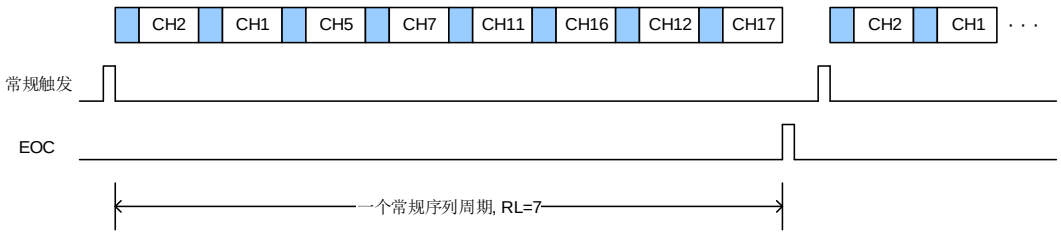
1. 设置ADC_CTL1寄存器的CTN位为1；
2. 根据模拟通道编号配置RSQ0；
3. 配置ADC_SAMPTx寄存器；
4. 如果有需要，配置ADC_CTL1寄存器的ETERC位和ETSRC位；
5. 准备*DMA控制器（DMA*模块，用于传输ADC_RDATA寄存器的数据；
6. 设置SWRCST位，或者给常规序列产生一个外部触发。

扫描运行模式

扫描运行模式可以通过将 ADC_CTL0 寄存器的 SM 位置 1 来使能。该模式下，ADC 扫描转换所有被 ADC_RSQ0~ADC_RSQ2 寄存器选中的通道。一旦 ADCON 位被置 1，当相应的软件触发或者外部触发产生，ADC 就会一个接一个的采样和转换常规序列通道。转换数据存储在 ADC_RDATA 寄存器中。常规序列转换结束后，EOC 位将被置 1。如果 EOCIE 位被置 1，将产生一个中断。当常规序列通道工作在扫描模式下时，ADC_CTL1 寄存器的 DMA 位必须设置为 1。

如果 ADC_CTL1 寄存器的 CTN 位也被置 1，则在常规序列转换完之后，这个转换自动重新开始。

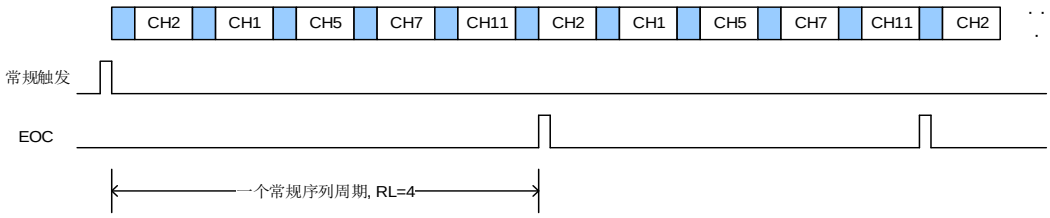
图 10-4. 扫描运行模式，连续运行模式禁用



常规序列扫描运行模式的软件流程：

1. 设置ADC_CTL0寄存器的SM位和ADC_CTL1寄存器的DMA位为1；
2. 配置ADC_RSQx和ADC_SAMPTx寄存器；
3. 如果有需要，配置ADC_CTL1寄存器中的ETERC和ETSRC位；
4. 准备*DMA控制器（DMA*模块，用于传输ADC_RDATA寄存器的数据；
5. 设置SWRCST位，或者给常规序列产生一个外部触发；
6. 等待EOC标志位置1；
7. 写0清除EOC标志位。

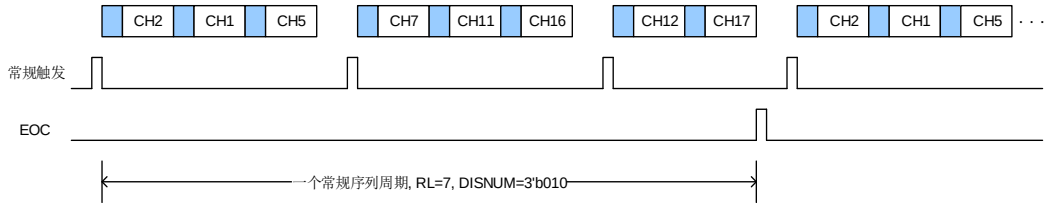
图 10-5. 扫描运行模式，连续运行模式使能



间断运行模式

对于常规序列，将 ADC_CTL0 寄存器的 DISRC 位置 1，常规序列使能间断运行模式。该模式下，可以执行一次 n 个通道的短序列转换（n 不超过 8），该转换是由 ADC_RSQ0~RSQ2 寄存器所选择的转换序列的一部分。数值 n 由 ADC_CTL0 寄存器的 DISCNUM[2:0]位配置。当相应的软件触发或外部触发产生时，ADC 就会采样和转换在 ADC_RSQ0~RSQ2 寄存器所选择通道中剩下来的 n 个通道，直到常规序列中所有的通道转换完成。每个常规序列短序列转换周期结束后，EOC 位将被置 1。如果 EOCIE 位被置 1，将产生一个中断。

图 10-6. 间断运行模式



常规序列间断运行模式的软件流程：

1. 设置ADC_CTL0寄存器的DISRC位和ADC_CTL1寄存器的DMA位为1；
2. 配置ADC_CTL0寄存器的DISNUM[2:0]位；
3. 配置ADC_RSQx和ADC_SAMPTx寄存器；
4. 如果有需要，配置ADC_CTL1寄存器中的ETERC位和ETSRC位；
5. 准备*DMA控制器（DMA*模块，用于传输ADC_RDATA寄存器中的数据；
6. 设置SWRCST位，或者给常规序列产生一个外部触发；
7. 如果有需要，重复步骤6；
8. 等待EOC标志位置1；
9. 写0清除EOC标志位。

10.4.6. 转换结果阈值监测功能

ADC_CTL0 寄存器中的 RWDEN 位置 1 时，将使能常规序列的模拟看门狗功能。该功能用于监测转换结果是否超过设定的阈值。如果 ADC 转换的模拟电压低于低阈值或高于高阈值时，ADC_STAT 状态寄存器的 WDE 位将被置 1。如果 WDEIE 位被置 1，将产生中断。ADC_WDHT 和 ADC_WDLT 寄存器用来设定高低阈值。内部数据的比较在对齐之前完成，因此阀值与 ADC_CTL1 寄存器中 DAL 位确定的对齐方式无关。ADC_CTL0 寄存器的 RWDEN，WDSC 和 WDCHSEL[4:0]位可以用来选择模拟看门狗监控单一通道或多个通道。

10.4.7. 数据存储模式

ADC_CTL1寄存器的DAL位确定转换后数据存储的对齐方式。

在最高有效位对齐中，12/10/8 位数据按半字方式对齐，而 6 位数据按照字节的方式对齐的，如下*图 10-7. 12 位分辨率的数据存储模式*，*图 10-8. 10 位分辨率的数据存储模式*，*图 10-9. 8 位分辨率的数据存储模式*和*图 10-10. 6 位分辨率的数据存储模式*所示。

图 10-7. 12 位分辨率的数据存储模式

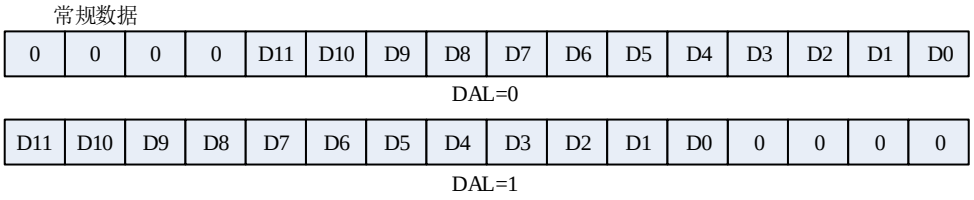


图 10-8. 10 位分辨率的数据存储模式

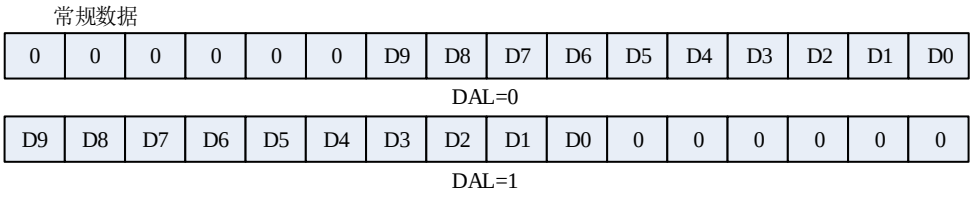


图 10-9. 8 位分辨率的数据存储模式

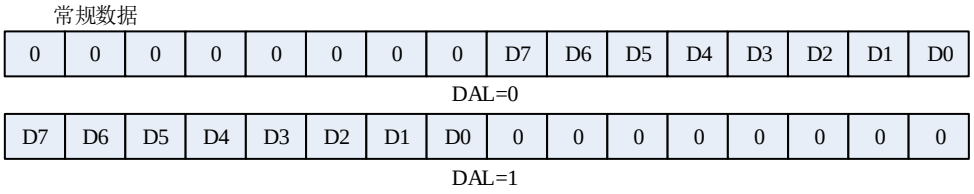
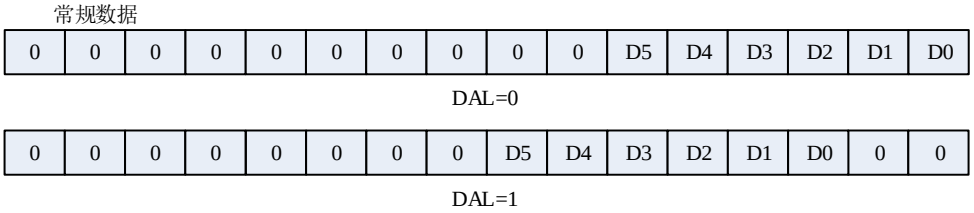


图 10-10. 6 位分辨率的数据存储模式



10.4.8. 采样时间配置

ADC 使用多个 ADC_CLK 周期对输入电压采样，采样周期数目可以通过 ADC_SAMPT0 和 ADC_SAMPT1 寄存器的 SPTn[2:0]位设置。每个通道可以用不同的采样周期。例如，在 12 位分辨率的情况下，总转换时间=采样时间+12.5 个 CK_ADC 周期。

例如：

CK_ADC = 14MHz，采样时间为 1.5 个周期，那么总的转换时间为：“1.5+12.5”个 CK_ADC 周期，即 1us。

10.4.9. 外部触发配置

外部触发输入的上升沿可以触发常规序列的转换。常规序列的外部触发源由 ADC_CTL1 寄存器的 ETSRC[2:0]位控制。

表 10-3. ADC 常规序列的外部触发源

ETSRC[2:0]	触发源	触发类型
000	TIMER0_CH0	硬件触发
001	TIMER0_CH1	
010	TIMER0_CH2	
011	TIMER1_CH1	
100	TIMER2_TRGO	
101	TIMER14_CH0	
110	EXTI11	
111	SWRCST	软件触发

10.4.10. DMA 请求

DMA请求，可以通过设置ADC_CTL1寄存器的DMA位来使能，用来传输常规序列多个通道的转换结果。ADC在常规序列一个通道转换结束后产生一个DMA请求，DMA接受到请求后可以将转换的数据从ADC_RDATA寄存器传输到用户指定的目的地址。

10.4.11. ADC 内部通道

将 ADC_CTL1 寄存器的 TSVREN 位置 1，可以使能温度传感器通道(ADC_IN16)和 V_{REFINT} 通道(ADC_IN17)。温度传感器可以用来测量器件周围的温度。传感器输出电压能被 ADC 转换成数字量。建议设置温度传感器的采样时间为 t_{S_temp} μs。温度传感器不用时，复位 TSVREN 位可以将其置于掉电模式。

温度传感器的输出电压随温度会发生线性变化，由于芯片生产过程的多样化，温度变化曲线的偏差在芯片间会有不同(最多相差 45°C)。内部温度传感器更适用于检测温度的变化，而不是用于测量绝对温度。如果需要测量精确的温度，应该使用一个外置的温度传感器来校准这个偏移错误。

内部电压参考(V_{REFINT})提供了一个稳定的（带隙基准）电压输出给 ADC 和比较器。V_{REFINT} 内部连接到 ADC_IN17 输入通道。

使用温度传感器：

1. 配置温度传感器通道（ADC_IN16）的转换序列和采样时间为t_{S_temp} us。
2. 置位ADC_CTL1寄存器的TSVREN位，使能温度传感器。
3. 置位ADC_CTL1寄存器的ADCON位，或者由外部触发ADC转换。
4. 读取内部温度传感器输出电压$V_{temperature}$，并由下面公式计算出实际温度：

$$温度\ (°C) = \{(V_{25} - V_{temperature}) / Avg_Slope\} + 25$$

V_{25}：内部温度传感器在 25°C 下的电压，典型值请参考相关型号 datasheet。
Avg_Slope：温度与内部温度传感器输出电压曲线的均值斜率，典型值请参考相关型号 datasheet。

10.4.12. 电池电压监测

V_{BAT}通道由于监测从V_{BAT}引脚过来的备份电池电压。当ADC_CTL1寄存器中的VBATEN位置1

时，使能V_{BAT}通道（ADC_IN18），同时一个集成在V_{BAT}引脚上的2分压桥也随之自动被使能。由于V_{BAT}可能比V_{DDA}高，所以使用这个2分压桥用来确保ADC正确操作。它将ADC_IN18输入通道连接到V_{BAT}/2，所以，ADC_IN18输入通道转换的值是V_{BAT}/2。为了防止不必要的电池能量消耗，推荐仅在需要时才使能2分压桥。

10.4.13. ADC 中断

以下任一个事件发生都可以产生中断：

- 常规序列转换结束；
- 模拟看门狗事件（模拟看门狗状态位置1）；

单独的中断使能位用于灵活设置ADC中断。

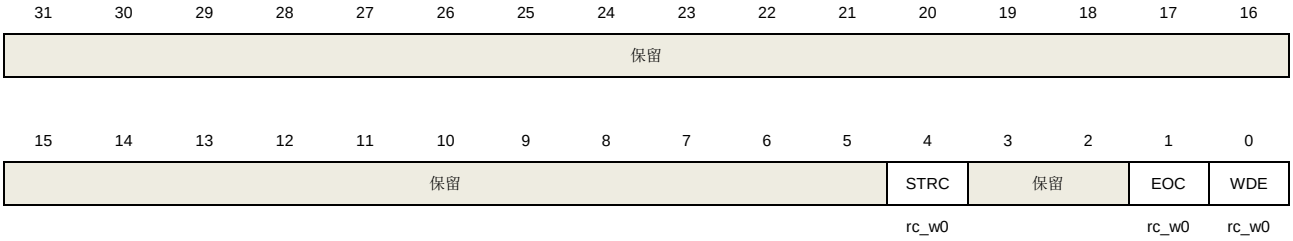
10.5. ADC 寄存器

ADC 基地址：0x4001 2400

10.5.1. 状态寄存器 (ADC_STAT)

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

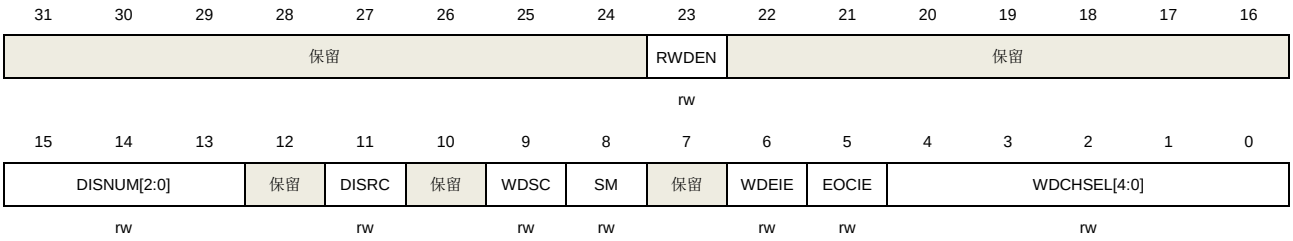


位/位域	名称	描述
31:5	保留	必须保持复位值。
4	STRC	常规序列转换开始标志 0：转换没有开始 1：转换开始 常规序列转换开始时硬件置位。软件写 0 清除。
3:2	保留	必须保持复位值。
1	EOC	常规序列转换结束标志 0：转换没有结束 1：转换结束 常规序列转换结束时硬件置位。 软件写 0 或读 ADC_RDATA 寄存器清除。
0	WDE	模拟看门狗事件标志 0：没有模拟看门狗事件 1：产生模拟看门狗事件 转换电压超过 ADC_WDLT 和 ADC_WDHT 寄存器中设定的阈值时，该位由硬件置 1，软件写 0 清除。

10.5.2. 控制寄存器 0 (ADC_CTL0)

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字(32位)访问。



位/位域	名称	描述
31:24	保留	必须保持复位值。
23	RWDEN	常规序列模拟看门狗使能 0：常规序列模拟看门狗禁止 1：常规序列模拟看门狗使能
22:16	保留	必须保持复位值
15:13	DISNUM[2:0]	间断模式下的转换数目 触发后即将被转换的通道数目将变成 DISNUM[2:0]+1
12	保留	必须保持复位值。
11	DISRC	常规序列间断模式 0：间断运行模式禁止 1：间断运行模式使能
10	保留	必须保持复位值。
9	WDSC	扫描模式下，模拟看门狗在通道配置 0：模拟看门狗在所有通道有效 1：模拟看门狗在单通道有效
8	SM	扫描模式 0：扫描运行模式禁止 1：扫描运行模式使能
7	保留	必须保持复位值。
6	WDEIE	WDE 中断使能 0：中断禁止 1：中断使能
5	EOCIE	EOC 中断使能 0：中断禁止 1：中断使能
4:0	WDCHSEL[4:0]	模拟看门狗通道选择 00000：ADC 通道 0 00001：ADC 通道 1 00010：ADC 通道 2 ……

01111：ADC 通道 15
10000：ADC 通道 16
10001：ADC 通道 17
10010：ADC 通道 18
注意：ADC 的模拟输入通道 16，通道 17 和通道 18 分别连接到温度传感器，V_{REFINT} 和 V_{BAT} 模拟输入。

10.5.3. 控制寄存器 1 (ADC_CTL1)

地址偏移：0x08
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留							VBATEN	TSVREN	SWRCST	保留	ETERC	ETSRC[2:0]			保留
							rw	rw	rw		rw	rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				DAL	保留		DMA	保留				RSTCLB	CLB	CTN	ADCON
				rw			rw					rw	rw	rw	rw

位/位域	名称	描述
31:25	保留	必须保持复位值。
24	VBATEN	该位由软件置位和清零，用于使能和禁止 V_{BAT} 通道 0：V_{BAT} 通道禁止 1：V_{BAT} 通道使能
23	TSVREN	ADC 的通道 16 和 17 使能 0：ADC 的通道 16 和 17 禁止 1：ADC 的通道 16 和 17 使能
22	SWRCST	软件触发常规序列转换开始 如果 ETSRC 是 111，该位置‘1’时开启常规序列转换。该位由软件置位，软件清零或转换开始由硬件清零。
21	保留	必须保持复位值。
20	ETERC	常规序列通道外部触发使能 0：常规序列通道外部触发禁止 1：常规序列通道外部触发使能
19:17	ETSRC [2:0]	常规序列通道外部触发选择 000：TIMER0 CH0 001：TIMER0 CH1 010：TIMER0 CH2 011：TIMER1 CH1

		100：TIMER2 TRGO 101：TIMER14 CH0 110：中断线 11 111：软件触发 SWRCST
16:12	保留	必须保持复位值。
11	DAL	数据对齐 0：最低有效位对齐 1：最高有效位对齐
10:9	保留	必须保持复位值。
8	DMA	DMA 请求使能 0：DMA 请求禁止 1：DMA 请求使能
7:4	保留	必须保持复位值。
3	RSTCLB	校准复位 该位由软件置位，在校准寄存器初始化后，该位硬件清零。 0：校准寄存器初始化结束 1：校准寄存器初始化开始
2	CLB	ADC 校准 0：校准结束 1：校准开始
1	CTN	连续模式 0：连续运行模式禁止 1：连续运行模式使能
0	ADCON	开启 ADC。该位从‘0’变成‘1’将在稳定时间结束后唤醒 ADC。当该位被置位以后，不改变寄存器的其他位仅仅对该位写‘1’，将开启转换。 0：禁能 ADC 并掉电 1：使能 ADC

10.5.4. 采样时间寄存器 0 (ADC_SAMPT0)

地址偏移：0x0C
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								SPT17[2:0]			SPT16[2:0]			SPT15[2:1]	
								rw			rw			rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPT15[0]	SPT14[2:0]			SPT13[2:0]			SPT12[2:0]			SPT11[2:0]			SPT10[2:0]		

rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:24	保留	必须保持复位值
23:21	SPT17[2:0]	参考 SPT10[2:0]的描述
20:18	SPT16[2:0]	参考 SPT10[2:0]的描述
17:15	SPT15[2:0]	参考 SPT10[2:0]的描述
14:12	SPT14[2:0]	参考 SPT10[2:0]的描述
11:9	SPT13[2:0]	参考 SPT10[2:0]的描述
8:6	SPT12[2:0]	参考 SPT10[2:0]的描述
5:3	SPT11[2:0]	参考 SPT10[2:0]的描述
2:0	SPT10[2:0]	通道采样时间 000：通道采样时间为1.5周期 001：通道采样时间为7.5周期 010：通道采样时间为13.5周期 011：通道采样时间为28.5周期 100：通道采样时间为41.5周期 101：通道采样时间为55.5周期 110：通道采样时间为71.5周期 111：通道采样时间为 239.5 周期 **注意**：通道 0 和通道 18 的采样时间都是通过 ADC_SAMPT1 的 SPT0[2:0]设置。

10.5.5. 采样时间寄存器 1 (ADC_SAMPT1)

地址偏移：0x10
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		SPT9[2:0]			SPT8[2:0]			SPT7[2:0]			SPT6[2:0]			SPT5[2:1]	
		rw			rw			rw			rw			rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPT5[0]		SPT4[2:0]			SPT3[2:0]			SPT2[2:0]			SPT1[2:0]			SPT0[2:0]	
rw		rw			rw			rw			rw			rw	

位/位域	名称	描述
31:30	保留	必须保持复位值
29:27	SPT9[2:0]	参考 SPT0[2:0]的描述

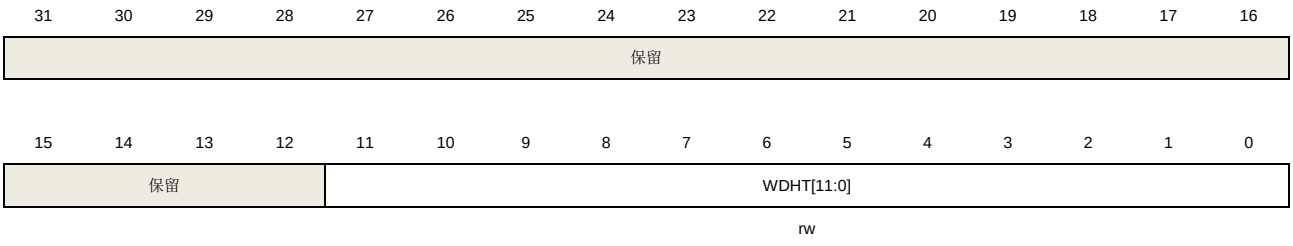
26:24	SPT8[2:0]	参考 SPT0[2:0]的描述
23:21	SPT7[2:0]	参考 SPT0[2:0]的描述
20:18	SPT6[2:0]	参考 SPT0[2:0]的描述
17:15	SPT5[2:0]	参考 SPT0[2:0]的描述
14:12	SPT4[2:0]	参考 SPT0[2:0]的描述
11:9	SPT3[2:0]	参考 SPT0[2:0]的描述
8:6	SPT2[2:0]	参考 SPT0[2:0]的描述
5:3	SPT1[2:0]	参考 SPT0[2:0]的描述
2:0	SPT0[2:0]	通道采样时间 000：通道采样时间为1.5周期 001：通道采样时间为7.5周期 010：通道采样时间为13.5周期 011：通道采样时间为28.5周期 100：通道采样时间为41.5周期 101：通道采样时间为55.5周期 110：通道采样时间为71.5周期 111：通道采样时间为 239.5 周期 **注意**：通道 0 和通道 18 的采样时间都是通过 ADC_SAMPT1 的 SPT0[2:0]设置。

10.5.6. 看门狗高阈值寄存器 (ADC_WDHT)

地址偏移：0x24

复位值：0x0000 0FFF

该寄存器只能按字(32位)访问。



位/位域	名称	描述
31:12	保留	必须保持复位值
11:0	WDHT[11:0]	模拟看门狗高侧阈值 这些位定义了模拟看门狗的高侧阈值。

10.5.7. 看门狗低阈值寄存器 (ADC_WDLT)

地址偏移：0x28

复位值：0x0000 0000

该寄存器只能按字(32位)访问。

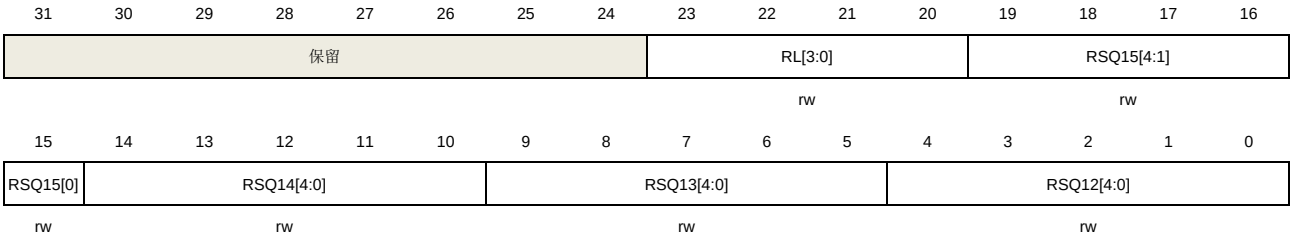
位/位域	名称	描述
31:12	保留	必须保持复位值
11:0	WDLT[11:0]	模拟看门狗低侧阈值 这些位定义了模拟看门狗的低侧阈值。

10.5.8. 常规序列寄存器 0 (ADC_RSQ0)

地址偏移：0x2C

复位值：0x0000 0000

该寄存器只能按字(32位)访问。



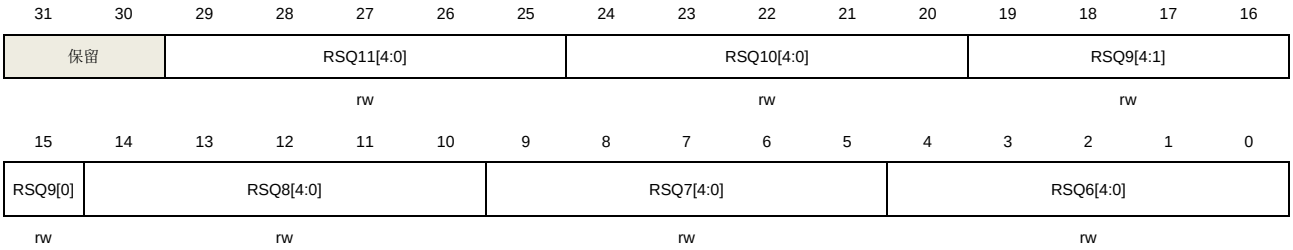
位/位域	名称	描述
31:24	保留	必须保持复位值
23:20	RL[3:0]	常规序列通道长度 常规序列通道转换序列中的总通道数目为 RL[3:0]+1.
19:15	RSQ15[4:0]	参考 RSQ0[4:0]的描述
14:10	RSQ14[4:0]	参考 RSQ0[4:0]的描述
9:5	RSQ13[4:0]	参考 RSQ0[4:0]的描述
4:0	RSQ12[4:0]	参考 RSQ0[4:0]的描述

10.5.9. 常规序列寄存器 1 (ADC_RSQ1)

地址偏移：0x30

复位值：0x0000 0000

该寄存器只能按字(32位)访问。

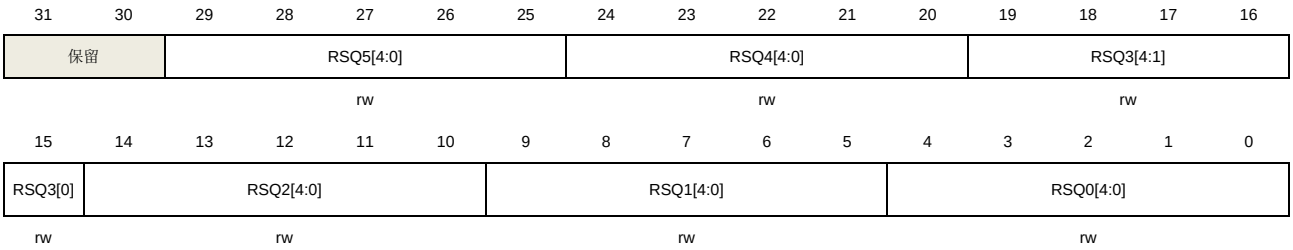


位/位域	名称	描述
31:30	保留	必须保持复位值
29:25	RSQ11[4:0]	参考 RSQ0[4:0]的描述
24:20	RSQ10[4:0]	参考 RSQ0[4:0]的描述
19:15	RSQ9[4:0]	参考 RSQ0[4:0]的描述
14:10	RSQ8[4:0]	参考 RSQ0[4:0]的描述
9:5	RSQ7[4:0]	参考 RSQ0[4:0]的描述
4:0	RSQ6[4:0]	参考 RSQ0[4:0]的描述

10.5.10. 常规序列寄存器 2 (ADC_RSQ2)

地址偏移：0x34
复位值：0x0000 0000

该寄存器只能按字(32位)访问。



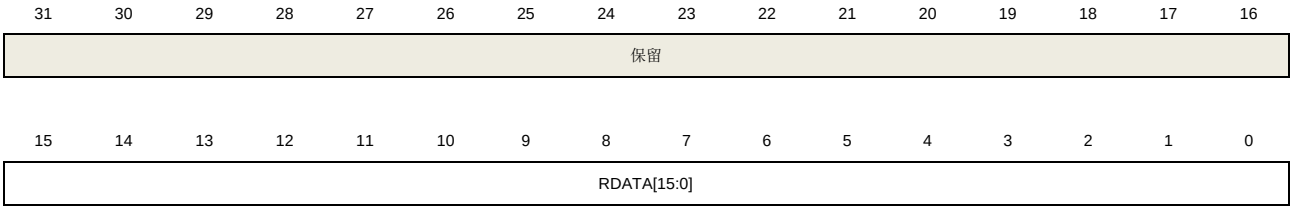
位/位域	名称	描述
31:30	保留	必须保持复位值
29:25	RSQ5[4:0]	参考 RSQ0[4:0]的描述
24:20	RSQ4[4:0]	参考 RSQ0[4:0]的描述
19:15	RSQ3[4:0]	参考 RSQ0[4:0]的描述
14:10	RSQ2[4:0]	参考 RSQ0[4:0]的描述
9:5	RSQ1[4:0]	参考 RSQ0[4:0]的描述

4:0	RSQ0[4:0]	通道编号（0..18）写入这些位来选择常规通道的第 n 个转换的通道

10.5.11. 常规数据寄存器 (ADC_RDATA)

地址偏移：0x4C
复位值：0x0000 0000

该寄存器只能按字(32位)访问。



位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	RDATA[15:0]	常规通道转换数据 这些位包含了常规通道的转换结果，只读。

11. 数模转换器（DAC）

11.1. 简介

数字/模拟转换器可以将 12 位的数字数据转换为外部引脚上的电压输出。数据可以采用 8 位或 12 位模式，左对齐或右对齐模式。当使能了外部触发，DMA 可被用于更新输入端数字数据。

在输出电压时，可以利用 DAC 输出缓冲区来获得更高的驱动能力。

11.2. 主要特性

DAC 的主要特征如下：

- 8 位或 12 位分辨率；
- 数据左对齐或右对齐；
- DMA 功能与欠载检测；
- 同步更新转换；
- 外部事件触发转换；
- 可配置的内部缓冲区；
- 输入参考电压 V_{DDA}；

*图 11-1. DAC 结构框图*为 DAC 的结构框图，*表 11-1. DAC 引脚*给出了引脚描述。

图 11-1. DAC 结构框图

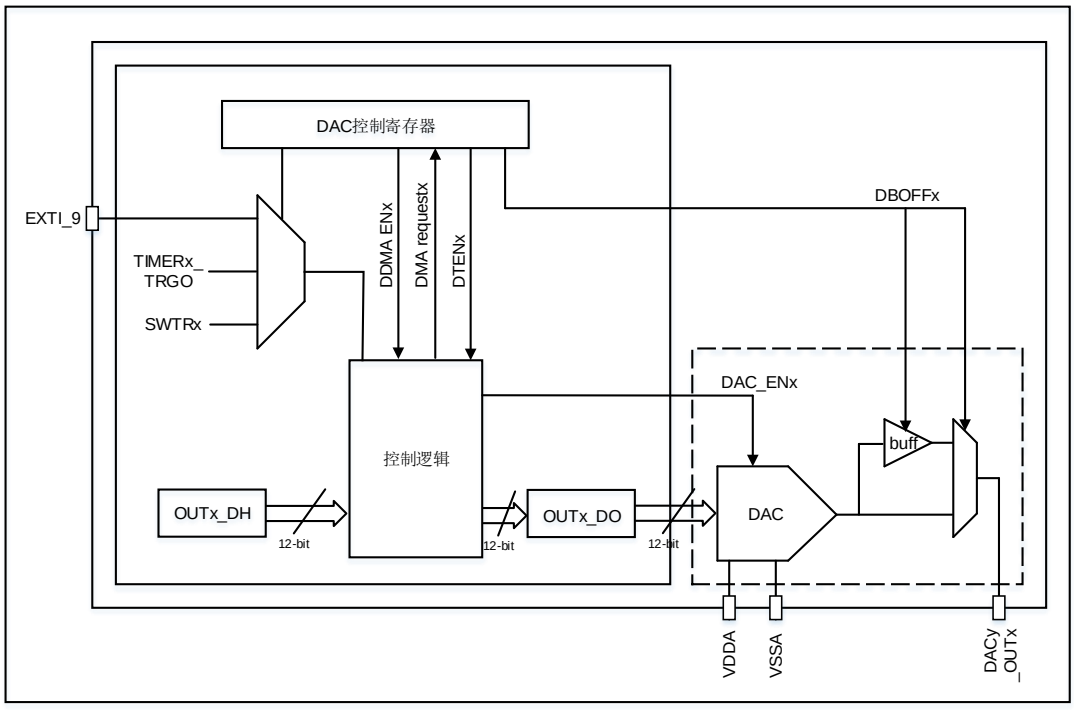


表 11-1. DAC 引脚

名称	描述	信号类型
V_{DDA}	模拟电源	输入，模拟电源
V_{SSA}	模拟电源地	输入，模拟电源地
DACy_OUTx	DAC 模拟输出	模拟输出信号

下表详细列出了 DAC 的触发与输出信号。

表 11-2. DAC 触发与输出

	DAC0
通道	通道 0
DAC 输出 I/O	PA4
DAC 输出 BUFFER 功能	•
软件触发功能	•
EXTI 触发信号	EXTI_9
TIMER 触发信号	TIMER1_TRGO TIMER2_TRGO TIMER5_TRGO TIMER14_TRGO

注意：在使能 DAC 模块前，GPIO 口（DAC 输出 I/O）应配置为模拟模式。

11.3. 功能描述

11.3.1. DAC 使能

将 DAC_CTL0 寄存器中的 DENx 位置 1，可以给 DAC 模块上电，DAC 子模块完全启动需要等待 t_{WAKEUP} 时间。

11.3.2. DAC 输出缓冲

为了降低输出阻抗，并在没有外部运算放大器的情况下驱动外部负载，每个 DAC 模块内部各集成了一个输出缓冲区。

缺省情况下，输出缓冲区是开启的，可以通过设置 DAC_CTL0 寄存器的 DBOFFx 位来开启或关闭缓冲区。

11.3.3. DAC 数据配置

对于 12 位的 DAC 保持数据（OUTx_DH），可以通过对 OUTx_R12DH、OUTx_L12DH 和 OUTx_R8DH 中的任意一个寄存器写入数据来配置。当数据被加载到 OUTx_R8DH 寄存器时，只有 8 位最高有效位是可配置，4 位最低有效位被强制置为 4’b0000。

11.3.4. DAC 触发

DAC 可以通过软件或者外部信号的上升沿触发。外部触发可以通过设置 DAC_CTL0 寄存器中 DTENx 位来使能。触发源可以通过 DAC_CTL0 寄存器中 DTSELx 位来进行选择，如*表 11-3. DAC 外部触发*所示。

表 11-3. DAC 外部触发

DTSELx[2:0]	触发源	触发类型
3b'000	TIMER5_TRGO	硬件触发
3b'001	TIMER2_TRGO	
3b'010	保留	
3b'011	TIMER14_TRGO	
3b'100	TIMER1_TRGO	
3b'101	保留	
3b'110	EXTI_9	
3b'111	SWTR	软件触发

TIMERx_TRGO 信号是由定时器生成的，而软件触发是通过设置 DAC_SWT 寄存器的 SWTRx 位生成的。

11.3.5. DAC 转换

如果使能了外部触发（通过设置 DAC_CTL0 寄存器的 DTENx 位），当已经选择的触发事件发生，DAC 保持数据（OUTx_DH）会被转移到 DAC 数据输出寄存器（OUTx_DO）。而在外部触发未使能的情况下，DAC 保持数据（OUTx_DH）会被自动转移到 DAC 数据输出寄存器（OUTx_DO）。

当 DAC 保持数据（OUTx_DH）加载到 OUTx_DO 寄存器时，经过 $t_{SETTLING}$ 时间之后，模拟输出变得有效，$t_{SETTLING}$ 的值与电源电压和模拟输出负载有关。

11.3.6. DAC 输出电压

DAC 引脚上的模拟输出电压取决于下面的等式：

$$V_{DAC_OUT}=V_{DDA}*OUTx_DO/4096 \quad (21\text{-}1)$$

数字输入被线性地转换成模拟输出电压，输出范围为 0 到 V_{DDA}。

11.3.7. DMA 请求

在外部触发使能的情况下，通过设置 DAC_CTL0 寄存器的 DDMAENx 位来使能 DMA 请求。当有外部硬件触发的时候（不是软件触发），则产生一个 DMA 请求。

如果在前一个请求响应之前第二个外部触发到达，则不响应新到的触发请求，并且发生欠载错误事件。DAC_STAT0 寄存器中的 DDUDRx 位置 1，如果 DAC_CTL0 寄存器中的 DDUDRIEx 位置 1，则会产生中断。

11.4. DAC 寄存器

DAC0 基地址：0x4000 7400

11.4.1. DACx 控制寄存器 (DAC_CTL0)

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		DDUDRIE0	DDMAEN0	保留						DTSEL0[2:0]			DTEN0	DBOFF0	DEN0
		rw	rw								rw		rw	rw	rw

位/位域	名称	描述
31:14	保留	必须保持复位值
13	DDUDRIE0	DACx_OUT0 DMA 欠载中断使能 0: DACx_OUT0 DMA 欠载中断禁能 1: DACx_OUT0 DMA 欠载中断使能
12	DDMAEN0	DACx_OUT0 DMA 使能 0：DACx_OUT0 DMA 模式禁能 1：DACx_OUT0 DMA模式使能
11:6	保留	必须保持复位值
5:3	DTSEL0[2:0]	DACx_OUT0 触发选择 这些位仅在 DTEN=1 并选择用于触发 DAC 的外部事件时使用。 000：TIMER5 TRGO 001：TIMER2 TRGO 010：Reserved 011：TIMER14 TRGO 100：TIMER1 TRGO 101：Reserved 110：外部中断线9 111：软件触发
2	DTEN0	DACx_OUT0 触发使能 0：DACx_OUT0 触发禁能 1：DACx_OUT0触发使能
1	DBOFF0	DACx_OUT0输出缓冲区关闭

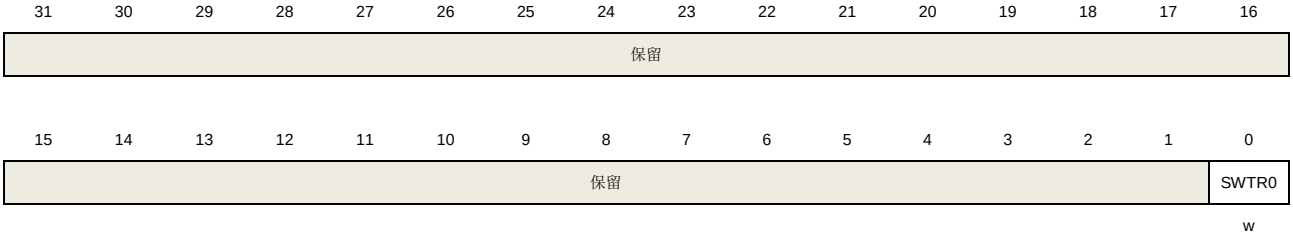
0：DACx_OUT0输出缓冲区打开，以降低输出阻抗，提高驱动能力
1：DACx_OUT0输出缓冲区关闭

0	DEN0	DACx_OUT0 使能 0：DACx_OUT0 禁能 1：DACx_OUT0使能

11.4.2. DACx 软件触发寄存器 (DAC_SWT)

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

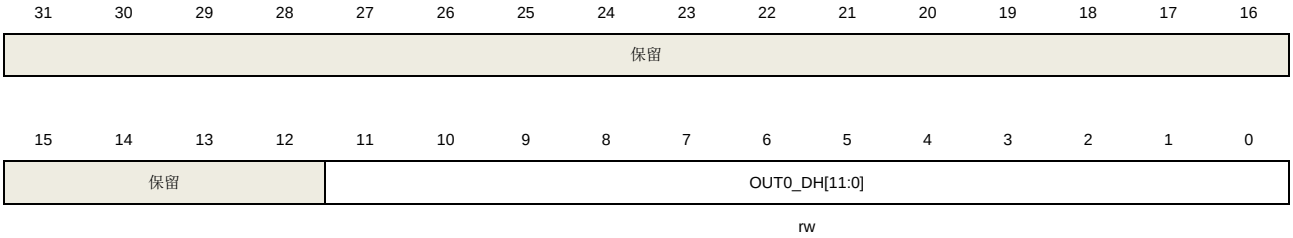


位/位域	名称	描述
31:1	保留	必须保持复位值
0	SWTR0	DACx_OUT0 软件触发，由硬件清除。 0：软件触发禁能 1：软件触发使能

11.4.3. DAC_OUT0 12 位右对齐数据保持寄存器 (DAC_OUT0_R12DH)

地址偏移：0x08
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

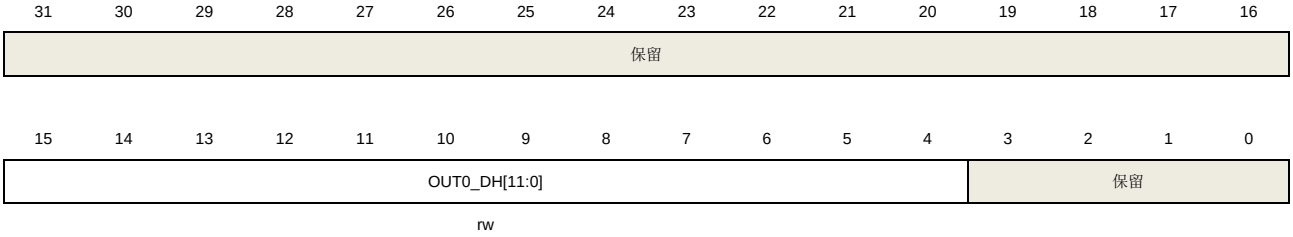


位/位域	名称	描述
31:12	保留	必须保持复位值
11:0	OUT0_DH[11:0]	DACx_OUT0 12 位右对齐数据 这些位指定了将由 DACx_OUT0 转换的数据。

11.4.4. DAC_OUT0 12 位左对齐数据保持寄存器 (DAC_OUT0_L12DH)

地址偏移：0x0C
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

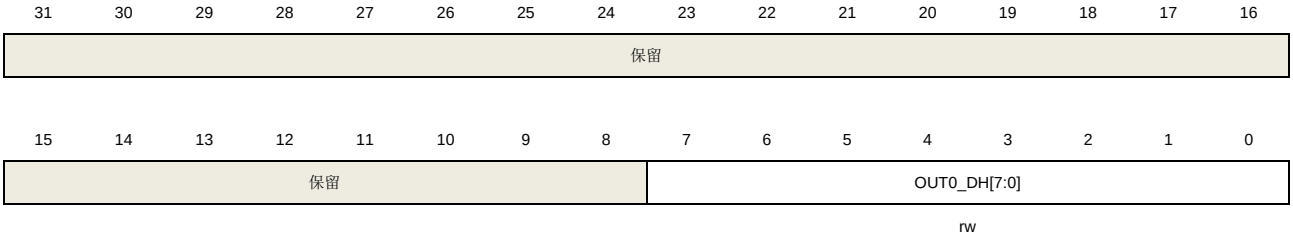


位/位域	名称	描述
31:16	保留	必须保持复位值
15:4	OUT0_DH[11:0]	DACx_OUT0 12 位左对齐数据 这些位指定了将由 DACx_OUT0 转换的数据。
3:0	保留	必须保持复位值

11.4.5. DAC_OUT0 8 位右对齐数据保持寄存器 (DAC_OUT0_R8DH)

地址偏移：0x10
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

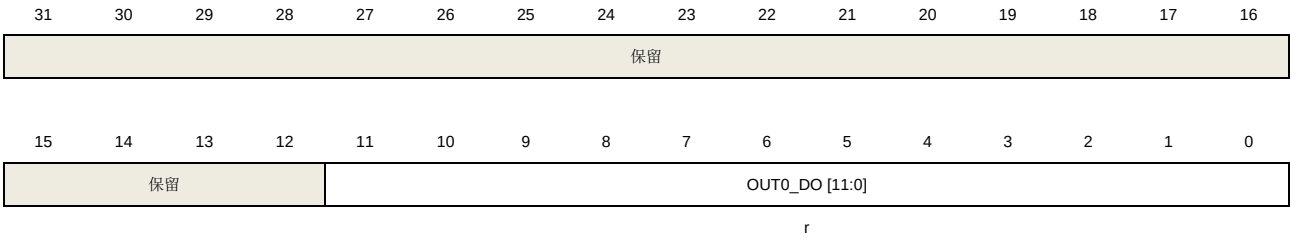


位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	OUT0_DH[7:0]	DACx_OUT0 8 位右对齐数据 这些位指定了将由 DACx_OUT0 转换的数据的最高 8 位有效位。

11.4.6. DAC_OUT0 数据输出寄存器 (DAC_OUT0_DO)

地址偏移：0x2C
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



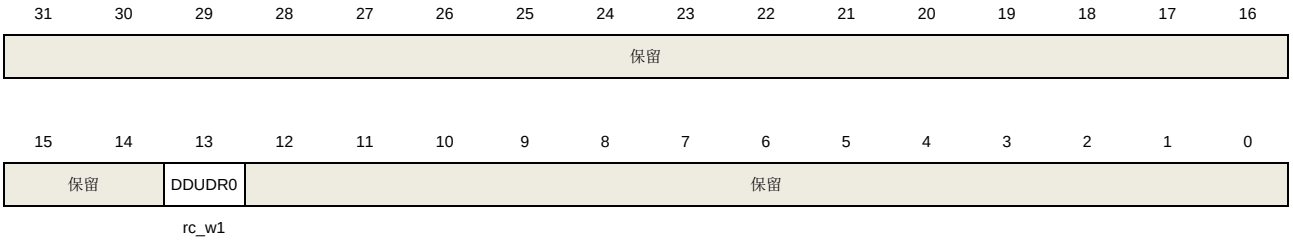
位/位域	名称	描述
31:12	保留	必须保持复位值
11:0	OUT0_DO [11:0]	DACx_OUT0 数据输出。 这些位为只读类型，存储由 DACx_OUT0 转换的数据。

11.4.7. DAC 状态寄存器 0 (DAC_STAT0)

地址偏移：0x34

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



位/位域	名称	描述
31:14	保留	必须保持复位值。
13	DDUDR0	DACx_OUT0 DMA 欠载标志位，硬件置位，软件写 1 清零。 0：没有欠载发生 1：发生欠载（DAC 触发产生速度快于 DMA 传输速度）
12:0	保留	必须保持复位值。

12. 比较器（CMP）

12.1. 简介

通用比较器可独立工作，其输出端可用于 I / O 口，也可和定时器结合使用。

比较器可通过模拟信号将 MCU 从低功耗模式中唤醒，在一定的条件下，可将模拟信号作为 TIMER 的触发源，结合 DAC 和 TIMER 的 PWM 输出，可以实现电流控制。

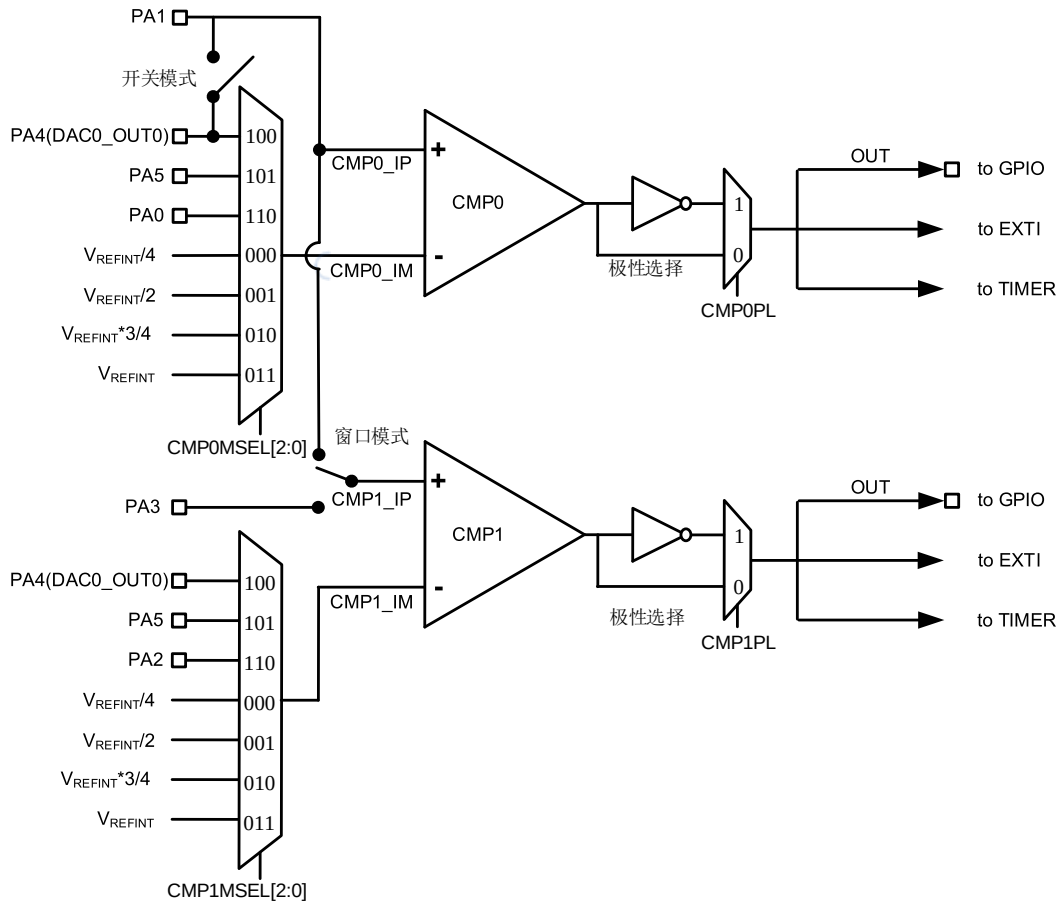
12.2. 主要特征

- 轨对轨比较器；
- 迟滞可配置；
- 速度、功耗可配置；
- 每个比较器可配置以下模拟信号作为输入源；
  - DAC 输出；
  - 多路复用 I / O 引脚；
  - 0.25、0.5、0.75、1 倍的内部参考电压；
- 窗口比较器；
- 输出到 I / O 口；
- 作为触发源输出到定时器；
- 输出到 EXTI。

12.3. 功能描述

比较器的框图展示如下：

图 12-1. 比较器框图



注意：V_{REFINT}为1.2V。

12.3.1. 比较器时钟

比较器与 APB 总线连接，时钟与 PCLK 同步。它与 SYSCFG 分享共同的复位和时钟使能位。

12.3.2. 比较器 I / O 配置

在被选为比较器输入端之前，相应管脚必须配置为模拟模式。

参考 Datasheet 的引脚定义，比较器输出必须连接到对应的复用 I / O 口。

比较器的输出可同时实现内部和外部输出。

比较器输出内部连接到定时器，他们的连接关系如下：

- CMP 输出连接到定时器输入通道；
- CMP 输出连接到定时器中止功能；
- CMP 输出连接到定时器 OCPRE_CLR。

为了在深度睡眠模式下工作，比较器端口的极性选择和输出重定向不会因为 PCLK 关闭。

*表 12-1 CMP 的输入和输出总结*详细描述了 CMP 的输入和输出。

表 12-1 CMP 的输入和输出总结

	CMP0	CMP1
CMP 同相输入连接到 I / O	PA1 PA4	PA1 PA3
CMP 反相输入连接到 I / O	PA0 PA4 PA5	PA2 PA4 PA5
CMP 反相输入连接到内部信号	V_{REFINT}/4 V_{REFINT}/2 V_{REFINT}*3/4 V_{REFINT} DAC0_OUT0	V_{REFINT}/4 V_{REFINT}/2 V_{REFINT}*3/4 V_{REFINT} DAC0_OUT0
CMP 输出连接到 I / O	PA0 PA6 PA11	PA2 PA7 PA12
CMP 输出连接到 EXTI	•	
CMP 输出连接到内部信号	TIMER0_CH0 TIMER1_CH3 TIMER2_CH0 TIMER0 OCPRE_CLR TIMER1 OCPRE_CLR TIMER2 OCPRE_CLR	TIMER0_CH0 TIMER1_CH3 TIMER2_CH0 TIMER0 OCPRE_CLR TIMER1 OCPRE_CLR TIMER2 OCPRE_CLR
CMP 输出（电机控制保护）	TIMER0 BRKIN	

12.3.3. 比较器供电模式

对于给定的程序，在比较器功耗和传输迟滞之间存在着权衡，可通过寄存器 CMP_CS 的位 CMPxM[1:0]的配置进行调整。当 CMPxM[1:0]为 2’b 00 时，比较器以运行速度最快和功耗最大模式工作，但当 CMPxM[1:0]为 2’b 11 时，比较器以运行速度最慢和功耗最小的模式工作。

12.3.4. 比较器窗口模式

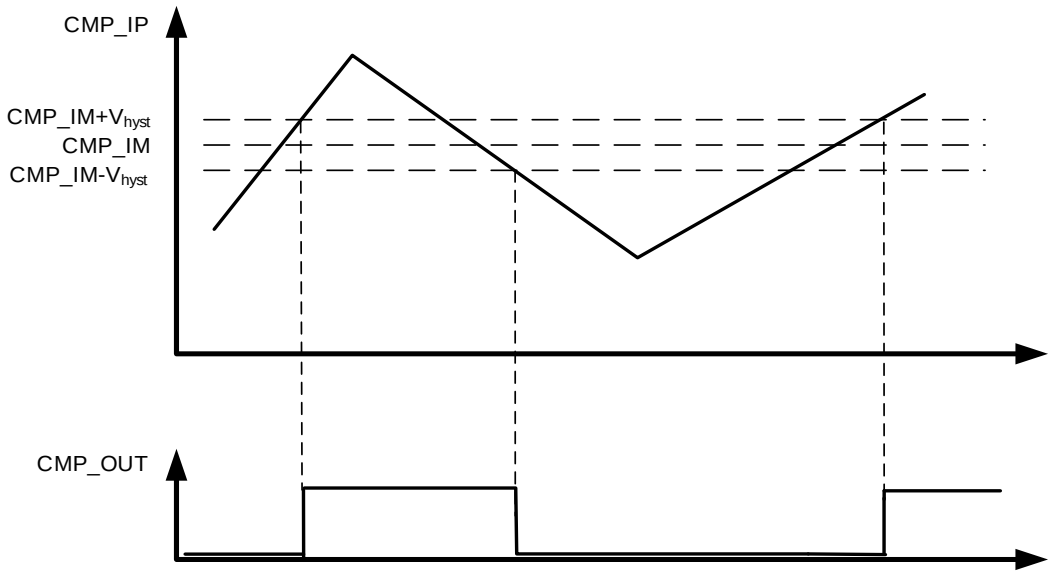
如果寄存器 CMP_CS 的 WNDEN 位被置位，比较器的窗口模式被使能，比较器 1 的正向输入端即与比较器 0 的正向输入端相连。如果 CMP0 和 CMP1 的反向输入端连接不同的内部电压，可以通过分析 CMP0 和 CMP1 的输出结果监测输入电压的范围，该范围的上下限由反向输入端所连接的内部电压值决定。

12.3.5. 比较器迟滞

为了避免噪声信号所引起的假输出，电路设计了可编程的迟滞功能，通过配置控制状态寄存器

来控制迟滞电压值。该功能可以在无需要时关闭。

图 12-2. 比较器迟滞



12.3.6. 比较器寄存器写保护

比较器的控制状态寄存器（CMP_CS）可通过设置 CMPxLK 位为 1 来进行写保护。CMP1LK 写 1 时，CMP_CS[31:16]位就会变为只读位，CMP0LK 写 1 时，CMP_CS[15:0]位就会变为只读位，只有在 MCU 复位时才可以复位。

12.3.7. 比较器中断

CMP 输出连接到 EXTI，EXTI 线对每个 CMP 都是独占的。通过这个功能，可以产生中断或者事件，用于退出省电模式。

12.4. CMP 寄存器

CMP 基地址：0x4001 001C

12.4.1. CMP 控制状态寄存器（CMP_CS）

地址偏移：0x00

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CMP1LK	CMP1O	CMP1HST[1:0]		CMP1PL	CMP1OSEL[2:0]			WNDEN	CMP1MSEL[2:0]			CMP1M[1:0]		保留	CMP1EN
rwo	r	rw		rw	rw			rw	rw			rw			rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP0LK	CMP0O	CMP0HST[1:0]		CMP0PL	CMP0OSEL[2:0]			保留	CMP0MSEL[2:0]			CMP0M[1:0]		CMP0SW	CMP0EN
rwo	r	rw		rw	rw				rw			rw		rw	rw

位/位域	名称	描述
31	CMP1LK	CMP1 写保护 该位可将 CMP1 的各控制位设为只读，该位只可通过软件置位一次，通过系统复位清除。 0：CMP_CS[31:16]是可读可写位 1：CMP_CS[31:16]是只读位
30	CMP1O	CMP1 输出 该位反映 CMP1 输出状态，是只读位。 0：正相输入端低于反相输入端，输出为低电平 1：正相输入端高于反相输入端，输出为高电平
29:28	CMP1HST[1:0]	CMP1 迟滞 该位域用于控制迟滞水平。 00：无迟滞 01：低迟滞 10：中迟滞 11：高迟滞
27	CMP1PL	CMP1 输出极性 该位用于控制 CMP1 输出极性。 0：输出是正相的 1：输出是反相的
26:24	CMP1OSEL[2:0]	CMP1 输出选择 该位域用于控制 CMP1 输出选择。 000：无选择 001：定时器 0 中止输入 010：定时器 0 通道 0 输入捕获

011：定时器 0 OCPRE_CLR 输入
100：定时器 1 通道 3 输入捕获
101：定时器 1 OCPRE_CLR 输入
110：定时器 2 通道 0 输入捕获
111：定时器 2 OCPRE_CLR 输入
注意：使用定时器捕获比较器的输出信号时，建议先使能 CMP，再配置定时器通道。

23	WNDEN	窗口模式使能 该位用于选择 CMP1_IP 输入源。 0：CMP1_IP 连接到 CMP1 的同相输入端 1：CMP1_IP 连接到 CMP0_IP
22:20	CMP1MSEL[2:0]	CMP1_IM 输入选择 该位域用于选择 CMP1 的输入端 CMP1_IM 的输入源。 000：V_{REFINT} / 4 001：V_{REFINT} / 2 010：V_{REFINT} * 3 / 4 011：V_{REFINT} 100：PA4（DAC0_OUT0） 101：PA5 110：PA2 111：保留
19:18	CMP1M[1:0]	CMP1 模式 该位域用于控制 CMP1 的运行模式以调整速度和功耗。 00：高速 / 全功耗 01：中速 / 中功耗 10：低速 / 低功耗 11：超低速 / 超低功耗
17	保留	必须保持复位值。
16	CMP1EN	CMP1 使能 0：CMP1 禁能。 1：CMP1 使能。
15	CMP0LK	CMP0 写保护 该位可将 CMP0 的各控制位设为只读。该位只可通过软件置位一次，通过系统复位清除。 0：CMP_CS[15:0]是可读可写位 1：CMP_CS[15:0]是只读位
14	CMP0O	CMP0 输出 该位反映 CMP0 输出状态，是只读位。 0：正相输入端低于反相输入端，输出为低电平 1：正相输入端高于反相输入端，输出为高电平

13:12	CMP0HST[1:0]	CMP0 迟滞 该位域用于控制迟滞水平。 00：无迟滞 01：低迟滞 10：中迟滞 11：高迟滞
11	CMP0PL	CMP0 输出极性 该位用于来控制 CMP0 输出极性。 0：输出是正相的 1：输出是反相的
10:8	CMP0OSEL[2:0]	CMP0 输出选择 该位域用于控制 CMP0 输出选择。 000：无选择 001：定时器 0 中止输入 010：定时器 0 通道 0 输入捕获 011：定时器 0 OCPRE_CLR 输入 100：定时器 1 通道 3 输入捕获 101：定时器 1 OCPRE_CLR 输入 110：定时器 2 通道 0 输入捕获 111：定时器 2 OCPRE_CLR 输入 **注意**：使用定时器捕获比较器的输出信号时，建议先使能 CMP，再配置定时器通道。
7	保留	必须保持复位值。
6:4	CMP0MSEL[2:0]	CMP0_IM 输入选择 该位域用于选择 CMP0 的输入端 CMP0_IM 的输入源。 000：V_{REFINT} / 4 001：V_{REFINT} / 2 010：V_{REFINT} * 3 / 4 011：V_{REFINT} 100：PA4（DAC0_OUT0） 101：PA5 110：PA0 111：保留
3:2	CMP0M[1:0]	CMP0 模式 该位域用于控制 CMP0 的运行模式以调整速度和功耗。 00：高速 / 全功耗 01：中速 / 中功耗 10：低速 / 低功耗 11：超低速 / 超低功耗
1	CMP0SW	CMP0 开关模式 该位用于开关 CMP0 正相输入端 PA1 与 PA4 之间的连接。

0：开关模式禁能
1：开关模式使能

0	CMP0EN	CMP0 使能 0：CMP0 禁能 1：CMP0 使能

13. 看门狗定时器（WDGT）

看门狗定时器（WDGT）是一个硬件计时电路，用来监测由软件故障导致的系统故障。片上有两个看门狗定时器外设，独立看门狗定时器（FWDGT）和窗口看门狗定时器（WWDGT）。它们使用灵活，并提供了很高的安全水平和精准的时间控制。两个看门狗定时器都是用来解决软件故障问题的。

看门狗定时器在内部计数值达到了预设的门限的时候，会触发一个复位。当处理器工作在调试模式的时候，看门狗定时器定时计数器可以暂停计数。

13.1. 独立看门狗定时器（FWDGT）

13.1.1. 简介

独立看门狗定时器（FWDGT）有独立的时钟源（IRC40K）。即使主时钟失效，FWDGT 依然能保持正常工作状态，适用于需要独立环境且对计时精度要求不高的场合。

当内部向下计数器的计数值达到 0，独立看门狗会产生一个系统复位。使能独立看门狗的寄存器写保护功能可以避免寄存器的值被意外的配置篡改。

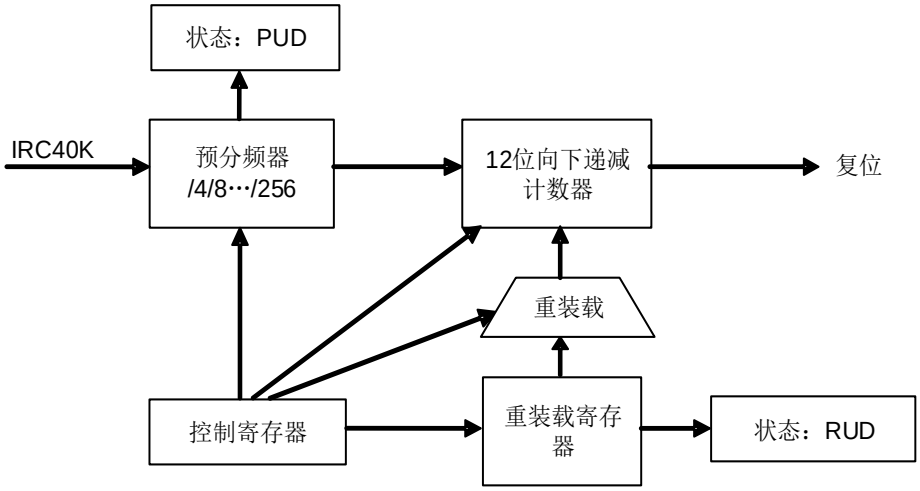
13.1.2. 主要特性

- 自由运行的12位向下计数器；
- 使能独立看门狗定时器，以下两种情况下产生复位：
  - 当计数器到0时产生复位；
  - 当计数器的值大于窗口寄存器的值时，更新计数器会产生复位。
- 独立时钟源，自由看门狗定时器在主时钟故障（例如待机和深度睡眠模式下）时仍能工作；
- 独立看门狗定时器硬件控制位，用来控制是否在上电时自动启动独立看门狗定时器；
- 可以配置独立看门狗定时器在调试模式下选择停止还是继续工作。

13.1.3. 功能描述

独立看门狗定时器带有一个 8 级预分频器和一个 12 位的向下递减计数器。参考*图 13-1. 独立看门狗定时器框图*的独立看门狗定时器的功能模块。

图 13-1. 独立看门狗定时器框图



向控制寄存器（FWDGT_CTL）中写 0xCCCC 可以开启独立看门狗定时器，计数器开始向下计数。当计数器计到 0x000，产生一次系统复位。

在任何时候向 FWDGT_CTL 中写 0xAAAA 都可以重装载计数器，重装载值来源于重装载寄存器（FWDGT_RLD）。软件可以在计数器计数值达到 0x000 之前通过重装载计数器来阻止看门狗定时器产生系统复位。

独立看门狗定时器也能够作为窗口看门狗定时器运行，只要在窗口寄存器（FWDGT_WND）中设置适当的窗口值即可。当重装载操作执行时，如果看门狗定时器计数器的值大于 FWDGT_WND 中存储的值，将会引起系统复位。FWDGT_WND 的默认值是 0x00000FFF，所以如果没有改写它，那么窗口选项默认是关闭的。窗口值一旦改变，立即就会引起看门狗定时器计数器的一次重加载动作，将向下递减计数器置为 FWDGT_RLD 中的值，并复位预分频计数器。

如果在可选字节中（OB_USER）打开了“硬件看门狗定时器”功能，那么在上电的时候看门狗定时器就被自动打开。为了避免系统复位，软件应该在计数器达到 0x000 之前重装载计数器。

预分频寄存器（FWDGT_PSC）、FWDGT_RLD 寄存器和 FWDGT_WND 寄存器有写保护功能。在写入数据到这些寄存器之前，需要写 0x5555 到 FWDGT_CTL 中。写其他任何值到 FWDGT_CTL 寄存器中将会再次启动对这些寄存器的写保护。当 FWDGT_PSC 寄存器或者 FWDGT_RLD 寄存器或者 FWDGT_WND 寄存器更新时，FWDGT_STAT 寄存器的状态位被置 1。

如果 DBG 中控制寄存器 0（DBG_CTL0）中的 FWDGT_HOLD 位被清 0，即使 Cortex®-M3 内核停止（调试模式下）独立看门狗定时器依然工作。如果 FWDGT_HOLD 位置 1，独立看门狗定时器将在调试模式下停止工作。

表 13-1. 独立看门狗定时器在 40KHz（IRC40K）时的最小 / 最大超时周期

预分频系数	PSC[2:0] 位	最小超时周期（ms） RLD[11:0]=0x000	最大超时周期（ms） RLD[11:0]=0xFFF
1/4	000	0.025	409.525
1 / 8	001	0.025	819.025

预分频系数	PSC[2:0] 位	最小超时周期（ms） RLD[11:0]=0x000	最大超时周期（ms） RLD[11:0]=0xFFF
1 / 16	010	0.025	1638.025
1 / 32	011	0.025	3276.025
1 / 64	100	0.025	6552.025
1 / 128	101	0.025	13104.025
1 / 256	110 or 111	0.025	26208.025

通过校准 IRC40 可以使独立看门狗定时器超时更加精确。

注意：当执行完喂狗 reload 操作之后，如需要立即进入 deepsleep / standby 模式时，必须通过软件设置，在 reload 命令及 deepsleep / standby 模式命令中间插入（3 个以上）IRC40K 时钟间隔。

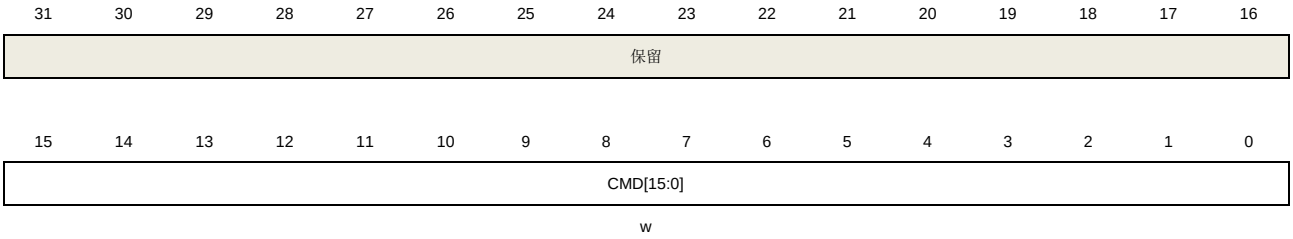
13.1.4. FWDGT 寄存器

FWDGT 基地址：0x4000 3000

控制寄存器（FWDGT_CTL）

地址偏移：0x00
复位值：0x0000 0000

该寄存器可以按半字（16 位）或字（32 位）访问。

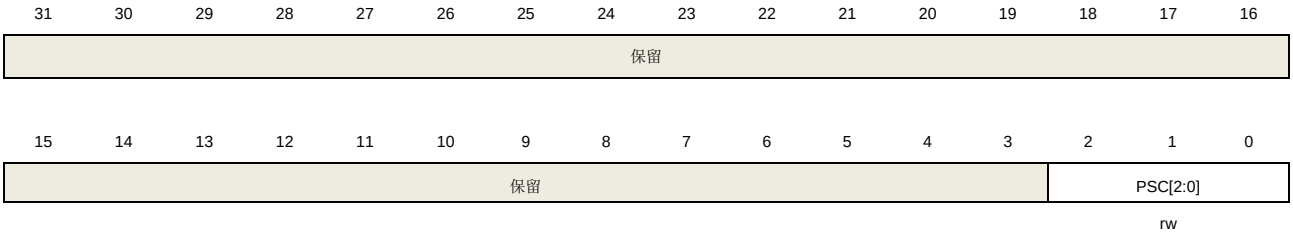


位/位域	名称	说明
31:16	保留	必须保持复位值。
15:0	CMD[15:0]	只可写，写入不同的值来产生不同的功能 0x5555：关闭FWDGT_PSC，FWDGT_RLD和FWDGT_WND寄存器的写保护 0xCCCC：开启独立看门狗定时器计数器。计数减到0时产生复位 0xAAAA：重装载计数器

预分频寄存器（FWDGT_PSC）

地址偏移：0x04
复位值：0x0000 0000

该寄存器可以按半字（16 位）或字（32 位）访问。



位/位域	名称	说明
31:3	保留	必须保持复位值。
2:0	PSC[2:0]	独立看门狗定时器计时预分频选择。写这些位之前要通过向FWDGT_CTL寄存器写0x5555去除写保护。在改写这个寄存器的过程中，FWDGT_STAT寄存器的PUD位被置1，此时读取此寄存器的值都是无效的。 000：1 / 4 001：1 / 8

010：1 / 16
011：1 / 32
100：1 / 64
101：1 / 128
110：1 / 256
111：1 / 256
如果应用需要使用不同的预分频系数，改变预分频值之前必须等到 PUD 位被清 0。更新预分频寄存器中的值后，在代码持续执行之前不必等待 PUD 值被清零（对于 GD32F130xx 和 GD32F150xx，在进入省电模式前需要等待 PUD 值清零）。

重加载寄存器（FWDGT_RLD）

地址偏移：0x08
复位值：0x0000 0FFF

该寄存器可以按半字（16 位）或字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				RLD [11:0]											
				rw											

位/位域	名称	说明
31:12	保留	必须保持复位值。
11:0	RLD[11:0]	独立看门狗定时器定时计数器重装载值。向 FWDGT_CTL 寄存器写入 0xAAAA 的时候，这个值会被更新到看门狗定时器计数器中。 这些位有写保护功能。在写这些位之前需向 FWDGT_CTL 寄存器中写 0x5555 解除写保护。在改写这个寄存器的过程中，FWDGT_STAT 寄存器的 RUD 位被置 1，此时寄存器中读取的任何值都是无效的。 如果应用需要使用不同的重装载值，改变重装载值之前必须等到 RUD 位被清 0。更新了重装载值之后，在代码持续执行之前不必等待 RUD 值被清零（对于 GD32F130xx 和 GD32F150xx，在进入省电模式前需要等待 RUD 值清零）。

状态寄存器（FWDGT_STAT）

地址偏移：0x0C
复位值：0x0000 0000

该寄存器可以按半字（16 位）或字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

保留	WUD	RUD	PUD
	r	r	r

位/位域	名称	说明
31:2	保留	必须保持复位值。
2	WUD	看门狗定时器定时计数器窗口值更新 FWDGT_WND 寄存器写操作时，该位被置 1，此时读取 FWDGT_WND 寄存器的任何值都是无效的。
1	RUD	独立看门狗定时器定时计数器重装载值更新 FWDGT_RLD 寄存器写操作时，该位被置 1，此时读取 FWDGT_RLD 寄存器的任何值都是无效的。
0	PUD	独立看门狗定时器预分频值更新 FWDGT_PSC 寄存器写操作时，该位被置 1，此时读取 FWDGT_PSC 寄存器的任何值都是无效的。

窗口寄存器（FWDGT_WND）

地址偏移：0x10
复位值：0x0000 0FFF

该寄存器可以按半字（16 位）或字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				WND[11:0]											
				rw											

位/位域	名称	说明
31:12	保留	必须保持复位值。
11:0	WND[11:0]	看门狗定时器定时计数器窗口值。这些位将用来将窗口值的上限值与向下递减计数器进行比较。当计数值大于WMD[11:0]中值，重装载操作会引起复位，若要改变重装载值，FWDGT_STAT寄存器中的WUD位必须保持复位状态。 这些位有写保护功能。在写这些位之前需向FWDGT_CTL寄存器中写0x5555。 如果应用需要使用几个窗口值，改变窗口值之前必须等到 WUD 位被清 0。除了在进入低功耗模式下，更新了窗口值后，在代码持续执行之前不必等待 WUD 值被清零。

13.2. 窗口看门狗定时器（WWDGT）

13.2.1. 简介

窗口看门狗定时器（WWDGT）用来监测由软件故障导致的系统故障。窗口看门狗定时器开启后，向下递减计数器值逐渐减小。计数值达到 0x3F 时会产生系统复位（CNT[6]位被清 0）。在计数器计数值达到窗口寄存器值之前，计数器的更新也会产生系统复位。因此软件需要在给定的窗口区间内更新计数器。窗口看门狗定时器在计数器计数值达到 0x40，会产生一个提前唤醒标志，如果使能中断也将会产生中断。

窗口看门狗定时器时钟是由 APB1 时钟预分频而来。窗口看门狗定时器适用于需要精确计时的场合。

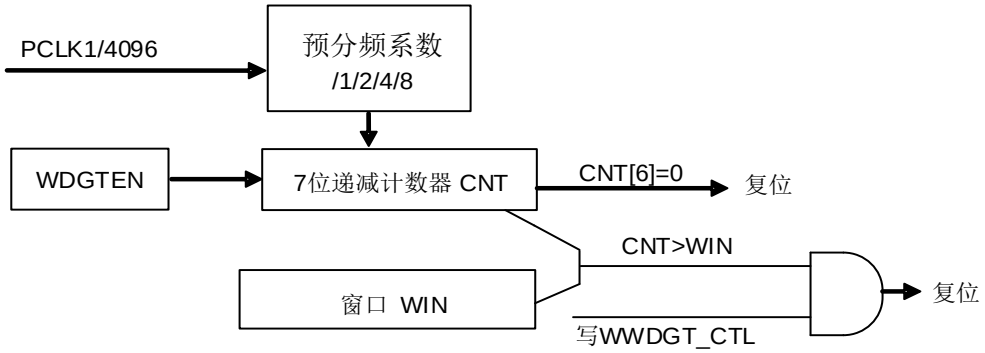
13.2.2. 主要特性

- 可编程的7位自由运行向下递减计数器；
- 当窗口看门狗使能后，有以下两种情况会产生复位：
  - 当计数器达到0x3F时产生复位；
  - 当计数器的值大于窗口寄存器的值时，更新计数器会产生复位。
- 提前唤醒中断（EWI）：看门狗定时器打开，中断使能，计数值达到0x40时会产生中断；
- 可以配置窗口看门狗定时器在调试模式下选择停止还是继续工作。

13.2.3. 功能描述

如果窗口看门狗定时器使能（将 WWDGT_CTL 寄存器的 WDGTEN 位置 1），计数值达到 0x3F 的时候产生复位（CNT[6]位被清 0）。或是在计数值达到窗口寄存器值之前，更新计数器也会产生复位。

图 13-2. 窗口看门狗定时器框图



上电复位之后窗口看门狗定时器总是关闭的。软件可以向 WWDGT_CTL 的 WDGTEN 写 1 开启窗口看门狗定时器。窗口看门狗定时器打开后，计数器始终递减计数，计数器配置的值应该大于 0x3F，也就是说 CNT[6]位应该被置 1。CNT[5:0]决定了两次重装载之间的最大间隔时间。计数器的递减速度取决于 APB1 时钟和预分频器（WWDGT_CFG 寄存器的 PSC[1:0]位）。

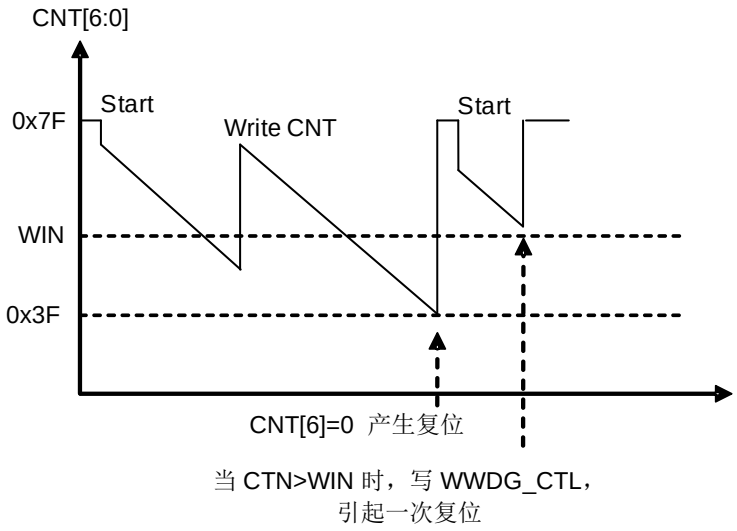
配置寄存器 （WWDGT_CFG）中的 WIN[6:0]位用来设定窗口值。当计数器的值小于窗口值，

且大于 0x3F 的时候，重装载向下计数器可以避免复位，否则引起复位。

对 WWDGT_CFG 寄存器的 EWIE 位置 1 可以使能提前唤醒中断（EWI），当计数值达到 0x40 的时候该中断产生。同时可以用相应的中断服务程序（ISR）来触发特定的行为（例如通信或数据记录），来分析软件故障的原因以及在器件复位的时候挽救重要数据。此外，在 ISR 中软件可以重装载计数器来管理软件系统检查等。在这种情况下，窗口看门狗定时器将永远不会复位但是可以用于其他地方。

通过将 WWDGT_STAT 寄存器的 EWIF 位写 0 可以清除 EWI 中断标志。

图 13-3. 窗口看门狗定时器时序图



窗口看门狗定时器超时的计算公式如下：

$$t_{WWDGT}=t_{PCLK1} \times 4096 \times 2^{PSC} \times (CNT[5:0]+1) \quad (ms) \qquad (13\text{-}1)$$

其中：

t_{WWDGT}：窗口看门狗定时器的超时时间

t_{PCLK1}： APB1以ms为单位的时钟周期

t_{WWDGT}的最大值和最小值请参考*表13-2. 在36MHz（fPCLK1）时的最大 / 最小超时值*。

表 13-2. 在 36MHz（f_{PCLK1}）时的最大 / 最小超时值

预分频系数	PSC[1:0]	最小超时 CNT[6:0] =0x40	最大超时 CNT[6:0]=0x7F
1 / 1	00	113 μs	7.28 ms
1 / 2	01	227 μs	14.56 ms
1 / 4	10	455 μs	29.12 ms
1 / 8	11	910 μs	58.25 ms

如果 DBG 控制寄存器 0（DBG_CTL0）中的 WWDGT_HOLD 位被清 0，即使 Cortex®-M3 内核停止工作（调试模式下），窗口看门狗定时器也可以继续工作。当 WWDGT_HOLD 位被置 1 时，窗口看门狗定时器在调试模式下停止。

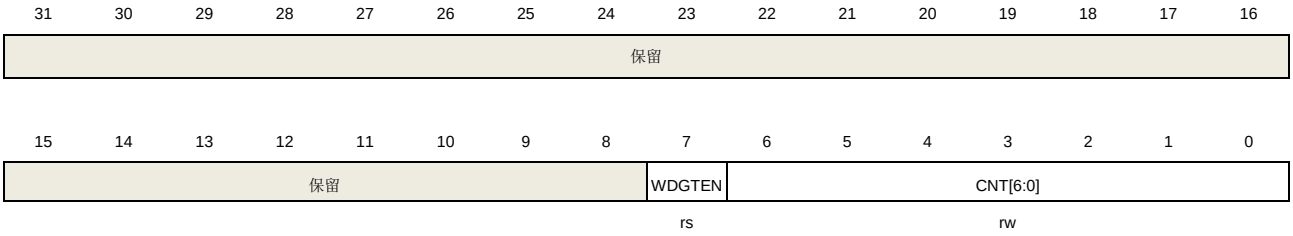
13.2.4. WWDGT 寄存器

WWDGT 基地址：0x4000 2C00

控制寄存器（WWDGT_CTL）

地址偏移：0x00
复位值：0x0000 007F

该寄存器可以按半字（16 位）或字（32 位）访问。

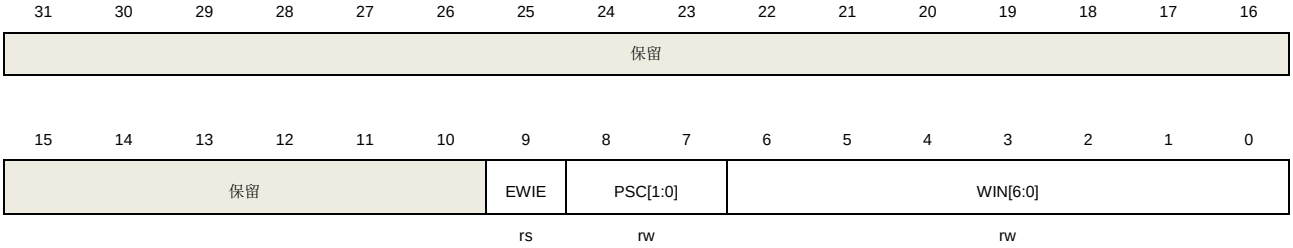


位/位域	名称	说明
31:8	保留	必须保持复位值。
7	WDGTEN	开启窗口看门狗定时器，硬件复位的时候清 0，写 0 无效。 0：关闭窗口看门狗定时器 1：开启窗口看门狗定时器
6:0	CNT[6:0]	看门狗定时器计数器的值。当计数值从 0x40 降到 0x3F 时，产生看门狗定时器复位。当计数器值高于窗口值的时候，写计数器可以产生看门狗定时器系统复位。

配置寄存器（WWDGT_CFG）

地址偏移：0x04
复位值：0x0000 007F

该寄存器可以按半字（16 位）或字（32 位）访问。



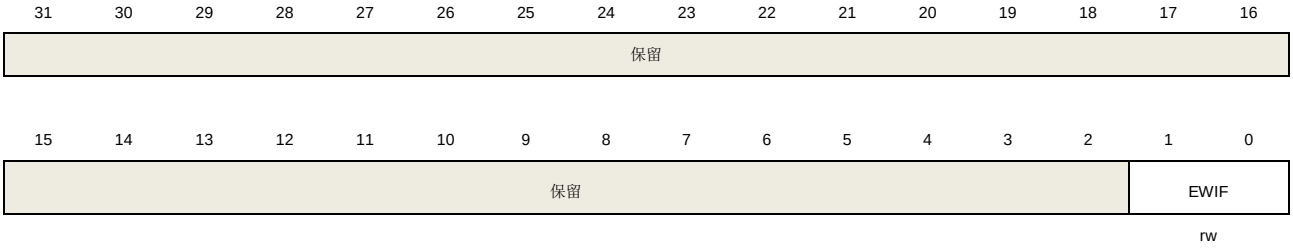
位/位域	名称	说明
31:10	保留	必须保持复位值。
9	EWIE	提前唤醒中断使能。如果该位被置 1，计数值达到 0x40 时触发中断。该位由硬件复位清 0，或通过置位 RCU 模块的 WWDGTRST 位进行软件复位。写 0 没有任何作用。

8:7	PSC[1:0]	看门狗定时器计数器的预分频系数 00：PCLK1 / 4096 / 1 01：PCLK1 / 4096 / 2 10：PCLK1 / 4096 / 4 11：PCLK1 / 4096 / 8
6:0	WIN[6:0]	窗口值，当看门狗定时器计数器的值大于窗口值时，写看门狗定时器计数器（WWDGT_CTL 的 CNT 位）会产生系统复位。

状态寄存器（WWDGT_STAT）

地址偏移：0x08
复位值：0x0000 0000

该寄存器可以按半字（16 位）或字（32 位）访问。



位/位域	名称	说明
31:1	保留	必须保持复位值。
0	EWIF	提前唤醒中断标志位。当计数值达到 0x40，即使中断没有被使能（WWDGT_CFG 中的 EWIE 位被清除）该位也会被硬件置 1。这个位可以通过写 0 清零，写 1 无效。

14. 实时时钟（RTC）

14.1. 简介

RTC 模块提供了一个包含日期（年/月/日）和时间（时/分/秒/亚秒）的日历功能。除亚秒用二进制码显示外，时间和日期都以 BCD 码的形式显示。RTC 可以进行夏令时补偿。RTC 可以工作在省电模式下，并通过软件配置来智能唤醒。RTC 支持外接更高精度的低频时钟，用以达到更高的日历精度。

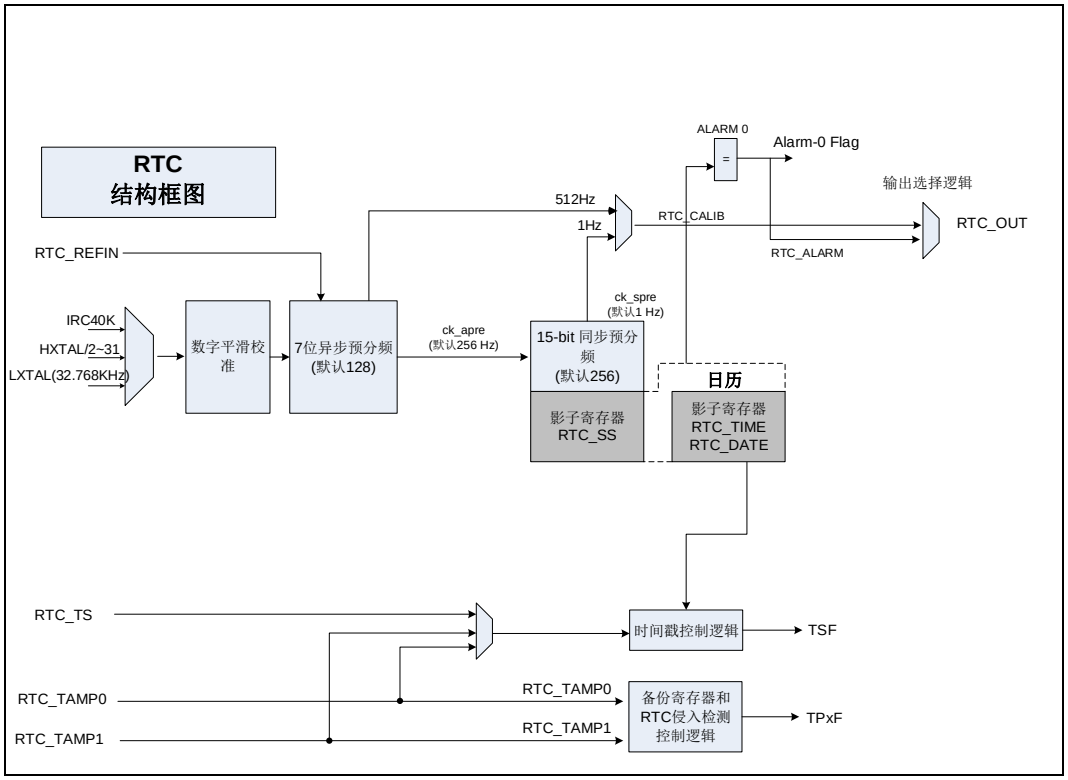
14.2. 主要特性

- 通过软件设置来实现夏令时补偿。
- 参考时钟检测功能：通过外接更高精度的低频率时钟源(50Hz或60Hz)来提高日历精度。
- 数字校准功能：通过调整最小时间单位（最大可调精度0.95ppm）来进行日历校准。
- 通过移位功能进行亚秒级调整。
- 记录事件时间的时间戳功能。
- 两个模式可配置的独立的侵入检测。
- 可编程的日历和一个位域可屏蔽的闹钟。
- 可屏蔽的中断源：
  - 闹钟 0；
  - 时间戳检测；
  - 侵入检测；
- 5个32位 (共20字节) 通用备份寄存器，能够在省电模式下保存数据。当有外部事件侵入时，备份寄存器将会复位。

14.3. 功能描述

14.3.1. 结构框图

图 14-1. RTC 结构框图



RTC 单元包括：

- 闹钟事件/中断。
- 侵入事件/中断。
- 32位备份寄存器。
- 可选的RTC输出功能：
  - 512Hz (默认预分频值)：RTC_OUT；
  - 1Hz(默认预分频值) ：RTC_ OUT；
  - 闹钟事件(极性可配置) ：RTC_ OUT。
- 可选的RTC输入功能：
  - 时间戳事件检测(RTC_TS)；
  - 侵入事件检测 0(RTC_TAMP0)；
  - 侵入事件检测 1(RTC_TAMP1)；
  - 参考时钟输入 RTC_REFIN(50 或 60Hz)。

14.3.2. 时钟源和预分频

RTC 单元有三个可选的独立时钟源：LXTAL、IRC40K 和 HXTAL 的 32 分频后的时钟。

在 RTC 单元，有两个预分频器用来实现日历功能和其他功能。一个分频器是 7 位异步预分频

器，另一个是 15 位同步预分频器。异步分频器主要用来降低功率消耗。 如果两个分频器都被使用，建议异步分频器的值尽可能大。

两个预分频器的频率计算公式如下：

$$f_{ck_apre}=\frac{f_{rtcclk}}{FACTOR_A+1} \quad (14\text{-}1)$$

$$f_{ck_spre}=\frac{f_{ck_apre}}{FACTOR_S+1}=\frac{f_{rtcclk}}{(FACTOR_A+1)*(FACTOR_S+1)} \quad (14\text{-}2)$$

ck_apre 用于为 RTC_SS 亚秒寄存器自减计数器提供时钟，该寄存器值为二进制，表示到达下一秒时间，该寄存器自减到 0 时，自动加载 FACTOR_S 的值。ck_spre 用于为日历寄存器提供时钟，每个时钟增加一秒。

14.3.3. 影子寄存器

当 APB 总线访问 RTC 日历寄存器 RTC_DATE、RTC_TIME 和 RTC_SS 时，BPSHAD 位决定是访问影子寄存器还是真实日历寄存器。默认情况下 BPSHAD 为 0，APB 总线访问影子日历寄存器。每两个 RTC 时钟，影子日历寄存器值会更新为真实日历寄存器的值，与此同时 RSYNF 位也会再次置位。在 Deep-sleep 和 Standby 模式下，影子寄存器不会更新。退出这两种模式时，软件必须清除 RSYNF 位。如果想要在 BPSHAD=0 的情况下读日历寄存器的值，须等待 RSYNF 置 1（最大的等待时间是 2 个 RTC 时钟周期）。

注意：在 BPSHAD=0 下，读日历寄存器(RTC_SS，RTC_TIME，RTC_DATE)的 APB 时钟的频率(f_{apb})必须至少是 RTC 时钟频率(f_{rtcclk})的七倍。

系统复位将复位影子寄存器。

14.3.4. 位域可屏蔽可配置的闹钟

RTC 闹钟功能被划分为多个位域并且每一个位域有一个该域的可屏蔽位。

RTC 闹钟功能的使能由 RTC_CTL 寄存器中的 ALRM0EN 位控制。当 ALRM0EN=1 并且闹钟所有位域的值与对应的日历时间值匹配，ALRM0F 标志位将会置位。

注意：当秒字段未被屏蔽时(RTC_ALRM0TD 寄存器的 MSKS=0)，为确保正常运行，RTC_PSC 寄存器的同步预分频系数（FACTOR_S）应大于等于 3。

如果一个位域被屏蔽，这个位域被认为在逻辑上匹配的。如果所有的位域被屏蔽，在 ALRM0EN 位被置位 3 个 RTC 时钟周期后，ALRM0F 位将置位。

14.3.5. RTC 初始化和配置

RTC 寄存器写保护

在默认情况下，PMU_CTL 寄存器的 BKPWEN 位被清 0。所以写 RTC 寄存器前需要软件提前设置 BKPWEN 位。

上电复位后，大多数 RTC 寄存器是被写保护的。写入这些寄存器的第一步是解锁这些保护。

通过下面的步骤，可以解锁这些保护：

1. 写‘0xCA’到RTC_WPK寄存器；
2. 写‘0x53’到RTC_WPK寄存器。

写一个错误的值到 RTC_WPK 会使写保护再次生效。写保护状态不受系统复位的影响。被写保护的寄存器如下：

RTC_TIME， RTC_DATE， RTC_CTL， RTC_STAT， RTC_PSC， RTC_ALRM0TD，RTC_SHIFTCTL，RTC_HRFC，RTC_ALRM0SS。

日历初始化和配置

通过以下步骤可以设置日历和预分频器的值：

1. 设置 INITM 位为 1 进入初始化模式。等待 INITF 位被置 1。
2. 在 RTC_PSC 寄存器中，设置同步和异步预分频器的分频系数。
3. 在影子寄存器(RTC_TIME 和 RTC_DATE)中写初始的日历值，并且通过设置 RTC_CTL 寄存器的 CS 位来配置时间的格式 (12 或 24 小时制)。
4. 清除 INITM 位退出初始化模式。

大约 4 个 RTC 时钟周期后，真正的日历寄存器将从影子寄存器载入时间和日期的设定值，同时日历计数器将要重新开始运行。

注意：初始化以后如果要读取日历寄存器(BPSHAD=0)，软件应该确保 RSYNF 位已经置 1。

YCM 标志表明日历是否完成初始化，该标志会硬件检查日历的年份值。

夏令时

通过 S1H，A1H 和 DSM 位配置，RTC 模块可以支持夏令时补偿调节功能。

当日历正在运行时，S1H 和 A1H 能使日历减去或加上 1 小时。S1H 和 A1H 功能可以重复设置，可以软件配置 DSM 位来记录这个调节操作。设置 S1H 或 A1H 位后，减或加 1 小时将在下一秒钟到来时生效。

闹钟功能操作步骤

为了避免意外的闹钟标记置位和亚稳态，闹钟功能的操作应遵循如下流程：

1. 清除寄存器 RTC_CTL 的 ALRM0EN 位，禁用闹钟；
2. 设置 Alarm 寄存器(RTC_ALRM0TD/RTC_ALRM0SS)；
3. 设置寄存器 RTC_CTL 的 ALRM0EN 位，使能闹钟功能。

14.3.6. 读取日历

当 BPSHAD=0 时，读日历寄存器

当 BPSHAD=0，从影子寄存器读日历的值。由于同步机制的存在，正常读取日历需要满足一个基本要求：APB1 总线时钟频率必须大于或等于 RTC 时钟频率的 7 倍。在任何情况下 APB1

总线时钟的频率都不能低于 RTC 的时钟频率。

当 APB1 总线时钟频率低于 7 倍 RTC 时钟频率时，日历的读取应该遵守以下流程：

1. 读取两次日历时间和日期寄存器；
2. 如果两次的值相等，那么这个值就是正确的；
3. 如果这两次的值不相等，应该再读一次；
4. 第三次的值可以认为是正确的。

RSYNF 每 2 个 RTC 时钟周期被置位一次。在这时，影子日历寄存器会更新为真实的日历时间和日期。

为了确保这 3 个值(RTC_SS，RTC_TIME，RTC_DATE)为同一时间，硬件上采取了如下一致性机制：

1. 读RTC_SS锁定RTC_TIME和RTC_DATE的更新；
2. 读RTC_TIME锁定RTC_DATE的更新；
3. 读 RTC_DATE 解锁 RTC_TIME 和 RTC_DATE 的更新。

如果想在一个很短的时间间隔内（少于 2 个 RTCCLK）读取日历，应先清除 RSYNF 位并等待其置位后再读取。

下面几种情况，软件须等待 RSYNF 置位后才能读日历寄存器（RTC_SS，RTC_TIME，RTC_DATE）：

1. 系统复位之后；
2. 日历初始化之后；
3. 一次移位操作之后。

特别是从低功耗模式唤醒后，软件必须清除 RSYNF 位并等待 RSYNF 再次置位后才能读取日历寄存器。

当 BPSHAD=1 时，读日历寄存器

当 BPSHAD=1，RSYNF 位会被硬件清 0，读日历寄存器不需考虑 RSYNF 位。当前真实的日历寄存器值会被直接读取。如此配置的好处是当从低功耗模式(Deep-sleep/Standby 模式)唤醒后，软件可以立即获取当前日历寄存器的值而无需加入任何等待延迟(此延迟最大为 2 个 RTC 时钟周期)。

由于没有 RSYNF 位周期性的置位，如果两次读日历寄存器之间出现 ck_apre 时钟边沿，不同寄存器(RTC_SS/RTC_TIME/RTC_DATE)的值可能并非同一时刻。

另外，如果日历寄存器的值正在发生变化的时刻被 APB 总线读取，那么有可能 APB 总线读取的值是不准确的。

为了确保日历值的正确性和一致性，读取时软件须如下操作：连续读取所有日历寄存器的值两次，如果上两次的值是一样的，那么这个值就是一致的且准确的。

14.3.7. RTC 复位

在 RTC 单元，有两个复位源可用：系统复位和备份域复位。

当系统复位有效时，日历影子寄存器和 RTC_STAT 寄存器的某些位将要复位到默认值。

备份域复位将会影响下面的寄存器，但系统复位不会对它们产生影响：

- RTC 真实的日历寄存器；
- RTC 控制寄存器 (RTC_CTL)；
- RTC 预分频寄存器 (RTC_PSC)；
- RTC 高精度频率补偿寄存器 (RTC_HRFC)；
- RTC 移位控制寄存器(RTC_SHIFTCTL)；
- RTC 时间戳寄存器 (RTC_SSTS/RTC_TTS/RTC_DTS)；
- RTC 侵入寄存器 (RTC_TAMP)；
- RTC 备份寄存器 (RTC_BKPx)；
- RTC 闹钟寄存器 (RTC_ALRM0SS/RTC_ALRM0TD)。

当系统复位或者进入省电模式的时候，RTC 单元将会继续运行。但是如果备份域复位，RTC 将会停止计数并且所有的寄存器会复位。

14.3.8. RTC 移位功能

当用户有一个高精度的远程时钟而且 RTC 1Hz 时钟（ck_spre）和远程时钟只有一个亚秒级的偏差，RTC 单元提供一个称作移位的功能去消除这个偏差来提高秒钟的精确性。

以二进制格式显示亚秒值，RTC 运行时该值是递减计数。 因此通过增加 RTC_SHIFTCTL 寄存器的 SFS[14:0]的值到 RTC_SS 同步预分频器计数器值 SSC[15:0])或通过增加 SFS[14:0]的值到同步预分频器计数器 SSC[15:0]并且同时置位 A1S 位，能分别延迟或提前下一秒到达的时间。

RTC_SS 的最大值取决于 RTC_PSC 寄存器的 FACTOR_S 的值。FACTOR_S 越大，调整的精度也就越高。

因为 1Hz 的时钟(ck_spre) 由 FACTOR_A 和 FACTOR_S 共同产生，越高的 FACTOR_S 值就意味着越低的 FACTOR_A 值，同时越低的 FACTOR_A 意味着越高的功耗。

注意：在使用移位功能之前，软件必须检查 RTC_SS 中 SSC 的第 15 位(SSC[15])并确保该位为 0。

写 RTC_SHIFTCTL 寄存器之后，RTC_STAT 寄存器的 SOPF 位将会再次置位。当同步移位操作完成时，SOPF 位被硬件清 0。系统复位不影响 SOPF 位。

当 REFEN=0 时，移位操作才能正确的工作。

如果 REFEN=1，软件禁止写入 RTC_SHIFTCTL。

14.3.9. RTC 参考时钟检测

RTC 参考时钟是另外一种提高 RTC 秒级精度的方法。为了使能这项功能，需要有一个相对于 LXTAL 有更高精度的外部参考时钟源 (50Hz 或 60 Hz)。

使能这项功能之后(REFEN=1)，每一个秒更新的时钟(1Hz)边沿将与最近的 RTC_REFIN 参考时钟沿进行对比。在大多数情况下，这两个时钟沿是对齐的。但当两个时钟沿由于 LXTAL 准

确度的原因没有对齐的时候，RTC 参考时钟的检测功能会偏移 1Hz 时钟沿一点相位，使得下一个 1Hz 时钟沿和参考时钟沿对齐。

当 REFEN=1，每一秒前后都会有一个进行检测的时间窗，处于不同的检测状态，时间窗时长也不同。

当检测状态处于检测第一个参考时钟边沿时，使用 7 个 ck_apre 时长的时间窗，当检测状态处于边沿对齐操作时，使用 3 个 ck_apre 时长的时间窗。

无论使用哪一种时间窗，当参考时钟在时间窗中被检测到的时候，同步预分频计数器会被强制重载。当两个时钟(ck_spre 和参考时钟) 边沿是对齐的，这个重载操作对 1Hz 日历更新没有任何影响。但是当两个时钟边沿没有对齐时，这个重载操作将会移动 ck_spre 时钟边沿，以使得 ck_spre(1Hz) 时钟边沿和参考时钟边沿对齐。

当参考检测功能正在运行中但外部参考时钟消失(在 3 个 ck_apre 时长时间窗内没有发现参考时钟边沿)，日历也能通过 LXTAL 继续自动更新。如果这个参考时钟重新恢复，参考时钟检测功能会先用 7 个 ck_apre 时长时间窗口去检测参考时钟，然后用 3 个 ck_apre 时长时间窗口去调节 ck_spre(1Hz)时钟边沿。

注意： 使能参考时钟检测功能之前(REFEN=1)，软件必须配置 FACTOR_A 为 0x7F，FACTOR_S 为 0xFF。

待机模式下时，参考时钟检测功能不可用。

14.3.10. RTC 数字平滑校准

RTC 平滑校准是一种用于校准 RTC 频率的方法，该方法通过调整校准周期内的 RTC 时钟脉冲个数的方式来实现校准。

完成一次这种校准相当于在一次校准周期内，RTC 时钟的脉冲个数增加或者减少了一定的数目。这种校准的分辨率大约为 0.954ppm，范围为从-487.1ppm 到+488.5ppm。

校准周期的时间可以配置到 $2^{20}/2^{19}/2^{18}$ RTC 时钟周期，如果 RTC 的输入频率是 32.768KHz，这些校准周期时间分别代表 32/16/8 秒。

高精度频率补偿寄存器(RTC_HRFC)指定了在校准周期内要屏蔽的 RTC 时钟数目，CMSK[8：0]位能屏蔽 0 到 511 个 RTC 时钟，这样 RTC 的频率最多降低 487.1ppm。

为了提高 RTC 频率可以设置 FREQI 位。如果 FREQI 位被置位，将会有 512 个额外的 RTC 时钟周期增加到校准周期(32/16/8 秒)时间期间，这意味着每 $2^{11}/2^{10}/2^{9}$ RTC 时钟插入一个 RTC 时钟周期。

因此使用 FREQI 可以使 RTC 频率增加 488.5ppm。

同时使用 CMSK 和 FREQI，每个周期时间可以调整-511 到+512 个 RTC 时钟周期。这意味着在 0.954ppm 分辨率的情况下，调整范围为从-487.1ppm 到+488.5ppm。

当数字平滑校准功能正在运行时，按如下公式计算输出校准频率：

$$f_{cal}=f_{rtcclk} \times(1+\frac{FREQI\times512-CMSK}{2^{N}+CMSK-FREQI\times512}) \quad (14\text{-}3)$$

注意： N=20/19/18 (32/16/8 秒)校准时间周期。

当 FACTOR_A < 3 时校准：

当异步预分频器值(FACTOR_A)被设置小于 3 时，若要使用校准功能，软件不能将 FREQI 位设置为 1。当 FACTOR_A<3，FREQI 位设置将会被忽略。

当 FACTOR_A 小于 3 时，FACTOR_S 值应小于标称值。假设 RTC 时钟频率是正常的 32.768KHz，对应的 FACTOR_S 应该按下面所示设置：
FACTOR_A = 2：FACTOR_S 减少 2(8189)
FACTOR_A = 1：FACTOR_S 减少 4(16379)
FACTOR_A = 0：FACTOR_S 减少 8(32759)

当 FACTOR_A 小于 3，CMSK 为 0x100，校准频率公式如下：

$$f_{cal}=f_{rtcclk} \times (1+\frac{256-CMSK}{2^N+CMSK-256}) \quad (14\text{-}4)$$

注意： N=20/19/18 (32/16/8 秒)校准时间周期。

验证 RTC 校准

提供 1Hz 校准时钟的输出用于协助软件测量并验证 RTC 的精度。

在有限的测量周期内测量 RTC 的频率，最高可能发生 2 个 RTCCLK 的测量误差。为了消除这一测量误差，测量周期应该和校准周期一致。

- 校准周期设为32秒(默认配置)

  用准确的 32 秒周期去测量 1Hz 校准输出的准确性能保证这个测量误差在 0.477ppm(在 32 秒周期内 0.5 个 RTCCLK)之内。

- 校准周期设为16秒(通过设置CWND16位)

  使用此配置，CMSK[0]被硬件置 0。

  用准确的 16 秒周期去测量 1Hz 校准输出的准确性能保证这个测量误差在 0.954ppm(在 16 秒周期内 0.5 个 RTCCLK)之内。

- 校准周期设为8秒(通过设置CWND8位)

  使用此配置，CMSK[1:0]被硬件置 0。用准确的 8 秒周期去测量 1Hz 校准输出的准确性能保证这个测量误差在 1.907ppm(在 8 秒周期内 0.5 个 RTCCLK)之内。

运行中重校准

当 INITF 位是 0，用下面的步骤，软件可以更新 RTC_HRFC：

1) 等待 SCPF 位置 0；
2) 写一个新的值到 RTC_HRFC 寄存器；
3) 3 个 ck_apre 时钟周期之后，新的校准设置开始生效。

14.3.11. 时间戳功能

时间戳功能由 RTC_TS 管脚输入，通过配置 TSEN 位来使能。

当 RTC_TS 管脚检测到时间戳事件发生时，会将日历的值保存在时间戳寄存器中(RTC_DTS/RTC_TTS/RTC_SSTS)，同时时间戳标志(TSF)也将由硬件置 1。如果时间戳中断使能被启用(TSIE)，时间戳事件会产生一个中断。

时间戳寄存器只会在时间戳事件第一次发生的时刻（TSF=0）记录日历时间，而当 TSF=1 时，时间戳事件的值不会被记录。

RTC 模块提供了一个可选的功能特性，来增加时间戳事件的触发源：设置 TPTS=1，使得侵入检测功能的侵入事件同时也作为时间戳事件的输入源。

注意：因为同步机制的原因，当时间戳事件发生时，TSF 会延迟 2 个 ck_apre 周期置位。

14.3.12. 侵入检测

RTC_TAMPx 管脚可以作为侵入事件检测功能输入管脚，检测模式有两种可供用户选择：边沿检测模式或者是带可配置滤波功能的电平检测模式。

RTC 备份寄存器(RTC_BKPx)

RTC 备份寄存器处于备份域中，即使 V_{DD} 电源被切断，该区域的寄存器的电源还可由 V_{BAT} 提供。从待机模式唤醒或系统复位操作都不会影响这些寄存器。

只有当被检测到有侵入事件和备份域复位时，这些寄存器复位。.

初始化侵入检测功能

TPxEN 位可以独立使能对应于不同管脚上的 RTC 侵入检测功能。使能 TPxEN 位启动侵入检测功能之前，需要设置好侵入检测的配置。当检测到侵入事件，相应的标志位(TPxF)将会置位。如果侵入事件中断使能被启用(TPIE)，侵入事件会产生一个中断。任何侵入事件都会导致备份寄存器(RTC_BKPx)复位。

侵入事件源的时间戳

使能 TPTS 位，能让侵入检测功能被用作时间戳功能。如果这位被设置为 1，当检测到侵入事件时，TSF 也将会被置位，如同使能了时间戳功能。当检测到侵入事件时，无论 TPTS 位的值如何，TPxF 位将置位。

侵入事件检测为边沿检测模式

当 FLT 位为 0x0 时，侵入检测被设置成边沿检测模式，TPxEG 位决定检测沿是上升沿还是下降沿。当侵入检测配置为边沿检测模式时，侵入检测输入管脚上的上拉电阻将会被禁用。

由于检测侵入事件会复位备份寄存器 (RTC_BKPx)，因此对备份寄存器写操作时应该确保侵入事件导致的复位和写操作不会同时发生。避免这种情形的推荐方法是先关闭侵入检测功能，在完成写操作后再重新启动该功能。

注意：PC13 上的侵入检测功能即使 V_{DD} 电源被关掉也依然可以运行。

侵入事件检测为带可配置滤波功能的电平检测模式

当 FLT 位没有被设置成 0x0 时，侵入检测被设置成电平检测模式，FLT 位决定有效电平需连续采样的次数(2，4 或者 8)。当 DISPU 被设置成 0(默认值)，内部的上拉电阻将会在每一次采样前预充电侵入管脚，这样侵入事件的输入管脚上就允许连接更大的电容。预充电的时间可以通过 PRCH 位来配置。越大的电容，所需的充电时间越长。

电平检测模式下每次采样之间的时间间隔是可配置的。通过调整采样频率(FREQ)，软件能在功耗和检测延迟之间取得一个平衡。

14.3.13. 校准时钟输出

如果 COEN 位设置为 1，RTC_OUT 管脚会输出参考校准时钟。

当 COS 位设置为 0(默认值)并且异步预分频器(FACTOR_A)设为 0x7F 时，RTC_CALIB 的频率是 $f_{rtcclk}/64$。因此若 RTCCLK 的频率为 32.768KHz，RTC_CALIB 对应的输出为 512Hz。因为下降沿存在轻微的抖动，因此推荐使用 RTC_CALIB 输出的上升沿。

当 COS 位设置为 1 时，RTC_CALIB 的频率计算公式为：

$$f_{rtc_calib} = \frac{f_{rtcclk}}{(FACTOR_A+1)\times(FACTOR_S+1)} \quad (14\text{-}5)$$

若 RTCCLK 为 32.768KHz，如果预分频器是默认值，那么 RTC_CALIB 对应的输出是 1Hz。

14.3.14. 闹钟输出

当 OS 控制位被设置为 0x01 时，RTC_ALARM 复用输出功能被启用。这个功能将直接输出 RTC_STAT 寄存器的 ALRM0F 值。

RTC_CTL 寄存器中的 OPOL 位可以配置 ALRM0F 位或者 WTF 位输出时候的极性，因此 RTC_ALARM 的输出电平有可能与相应的位值相反。

14.3.15. RTC 省电模式管理

表 14-1. 省电模式管理

模式	模式下能否工作	退出该模式的方法
睡眠模式	是	RTC 中断
深度睡眠模式	当时钟源是 LXTAL 或 IRC40K 时可以工作	RTC 闹钟/侵入事件/时间戳事件
待机模式	当时钟源是 LXTAL 或 IRC40K 时可以工作	RTC 闹钟/侵入事件/时间戳事件

14.3.16. RTC 中断

所有的 RTC 中断都被连接到 EXTI 控制器。

如果想使用 RTC 闹钟/侵入事件/时间戳中断，应按下面步骤操作：

1) 设置并使能对应的 EXTI 中连接到 RTC 闹钟/侵入事件/时间戳的中断线，然后配置该线为上升沿触发模式；
2) 配置并使能 RTC 闹钟/侵入事件/时间戳的全局中断；
3) 配置并使能 RTC 闹钟/侵入事件/时间戳功能。

表 14-2. 中断控制

中断	事件标志	控制位	退出睡眠模式	退出深度睡眠模式	退出待机模式
闹钟 0	ALRM0F	ALRM0IE	Y	Y(*)	Y(*)
时间戳	TSF	TSIE	Y	Y(*)	Y(*)
侵入 0	TP0F	TPIE	Y	Y(*)	Y(*)
侵入 1	TP1F	TPIE	Y	Y(*)	Y(*)

* ：仅当 RTC 时钟源是 LXTAL 或 IRC40K 时有效。

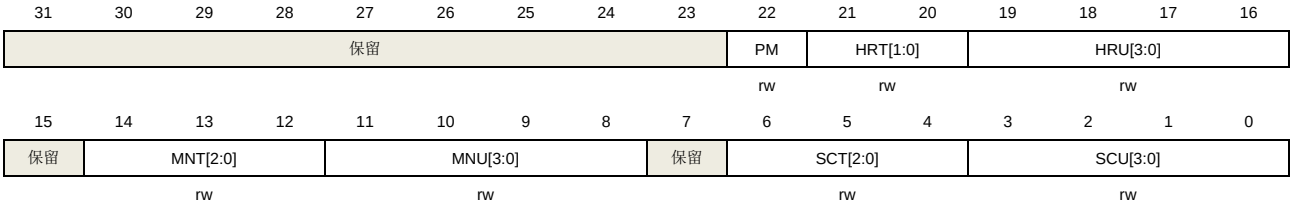
14.4. RTC 寄存器

RTC 基地址：0x4000 2800

14.4.1. 时间寄存器 (RTC_TIME)

地址偏移：0x00
系统复位值：当BPSHAD = 0，0x0000 0000
当BPSHAD = 1，无影响

写保护寄存器，仅在初始化状态可以进行写操作。
该寄存器只能按字(32位)访问。



位/位域	名称	描述
31:23	保留	必须保持复位值
22	PM	AM/PM 标志 0：AM 或 24 小时制 1：PM
21:20	HRT[1:0]	小时十位值，以 BCD 码形式存储
19:16	HRU[3:0]	小时个位值，以 BCD 码形式存储
15:	保留	必须保持复位值
14:12	MNT[2:0]	分钟十位值，以 BCD 码形式存储
11:8	MNU[3:0]	分钟个位值，以 BCD 码形式存储
7	保留	必须保持复位值
6:4	SCT[2:0]	秒钟十位值，以 BCD 码形式存储
3:0	SCU[3:0]	秒钟个位值，以 BCD 码形式存储

14.4.2. 日期寄存器 (RTC_DATE)

地址偏移：0x04
系统复位值：当 BPSHAD = 0，0x0000 2101
当BPSHAD = 1，无影响

写保护寄存器，仅在初始化状态可以进行写操作。

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								YRT[3:0]				YRU[3:0]			
								rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DOW[2:0]			MONT	MONU[2:0]				保留		DAYT[1:0]		DAYU[3:0]			
rw			rw	rw						rw		rw			

位/位域	名称	描述
31:24	保留	必须保持复位值
23:20	YRT[3:0]	年份十位值，以 BCD 码形式存储
19:16	YRU[3:0]	年份个位值，以 BCD 码形式存储
15:13	DOW[2:0]	星期 0x0：保留 0x1：星期一 … 0x7：星期日
12	MONT	月份十位值，以 BCD 码形式存储
11:8	MONU[2:0]	月份个位值，以 BCD 码形式存储
7:6	保留	必须保持复位值
5:4	DAYT[1:0]	日期十位值，以 BCD 码形式存储
3:0	DAYU[3:0]	日期个位值，以 BCD 码形式存储

14.4.3. 控制寄存器 (RTC_CTL)

地址偏移：0x08
系统复位：无影响
备份域复位值：0x0000 0000

写保护寄存器
该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								COEN	OS[1:0]		OPOL	COS	DSM	S1H	A1H
								rw	rw		rw	rw	rw	w	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSIE	保留		ALRM0IE	TSEN	保留		ALRM0EN	保留	CS	BPSHAD	REFEN	TSEG	保留		
rw			rw	rw			rw		rw	rw	rw	rw			

位/位域	名称	描述
31:24	保留	必须保持复位值

23	COEN	校准输出使能 0：关闭校准输出 1：使能校准输出
22:21	OS[1:0]	输出选择 该位用来选择输出的标志源。 0x0：禁用 RTC_ALARM 输出 0x1：启用闹钟 0 标志输出
20	OPOL	输出极性 该位用来反转 RTC_ALARM 输出。 0：禁用反转 RTC_ALARM 输出 1：启用反转 RTC_ALARM 输出
19	COS	校准输出选择 仅当 COEN=1 并且预分频器是默认值时有效。 0：校准输出是 512Hz 1：校准输出是 1Hz
18	DSM	夏令时屏蔽位 该位可以通过软件灵活使用。常用来记录夏令时调整。
17	S1H	减 1 小时(冬季时间变化) 当前时间非零的情况下，将当前时间减去一个小时。 0：没有影响 1：在下一个秒改变时，将减少一个小时
16	A1H	增加 1 小时(夏季时间变化) 将当前时间增加一个小时。 0：没有影响 1：在下一个秒改变时，将增加一个小时
15	TSIE	时间戳中断使能 0：禁用时间戳中断 1：启用时间戳中断
14:13	保留	必须保持复位值
13	ALRM1IE	RTC 闹钟 1 中断使能 0：禁用闹钟中断 1：启用闹钟中断
12	ALRM0IE	RTC 闹钟 0 中断使能 0：禁用闹钟中断 1：启用闹钟中断
11	TSEN	时间戳功能使能 0：禁用时间戳功能 1：启用时间戳功能

10:9	保留	必须保持复位值
8	ALRM0EN	闹钟 0 功能使能 0：禁用闹钟功能 1：启用闹钟功能
7	保留	必须保持复位值
6	CS	时间格式 0：24 小时制 1：12 小时制 **注意：**仅能在初始化状态进行写入
5	BPSHAD	禁止影子寄存器 0：读取的日历的值来自影子日历寄存器 1：读取的日历的值来自真正日历寄存器 **注意：**如果 APB1 时钟的频率小于 RTCCLK 频率的 7 倍，该位必须设为 1
4	REFEN	参考时钟检测功能使能 0：禁用参考时钟检测功能 1：启用参考时钟检测功能 **注意：**仅能在初始化状态进行写入并且 FACTOR_S 必须为 0x00FF
3	TSEG	时间戳事件有效检测边沿 0：上升沿是时间戳事件有效检测沿 1：下降沿是时间戳事件有效检测沿
2:0	保留	必须保持复位值

14.4.4. 状态寄存器 (RTC_STAT)

地址偏移：0x0C
系统复位：仅INITM，INITF和RSYNF位被置0，其他位无影响。
备份域复位值：0x0000 0007

写保护寄存器，除 RTC_STAT[14:8]外。
该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															SCPF
															r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TP1F	TP0F	TSOVRF	TSF	保留		ALRM0F	INITM	INITF	RSYNF	YCM	SOPF	保留		ALRM0WF
	rc_w0	rc_w0	rc_w0	rc_w0			rc_w0	rw	r	rc_w0	r	r			r

位/位域	名称	描述
31:17	保留	必须保持复位值
16	SCPF	平滑校准挂起标志

		对 RTC_HRFC 进行软件写操作，该位被硬件置 1。当平滑校准周期完成后，该位被硬件清 0
15	保留	必须保持复位值
14	TP1F	RTC_TAMP1 事件标志 当在 tamper1 输入管脚检测到侵入事件时，该位硬件置 1。可以通过向该位软件写 0 来清除。
13	TP0F	RTC_TAMP0 事件标志 当在 tamper0 输入管脚检测到侵入事件时，该位硬件置 1。可以通过向该位软件写 0 来清除。
12	TSOVRF	时间戳事件溢出标志 如果 TSF 位已经置位，当再次检测到时间戳事件时，该位会通过硬件置 1。 可以通过向该位软件写 0 来清除。
11	TSF	时间戳事件标志 当检测到一个时间戳事件时，该位会通过硬件置 1。可以通过向该位软件写 0 来清除。
10:9	保留	必须保持复位值
8	ALRM0F	Alarm 0 发生标志 当现在的时间/日期与闹钟 0 设置的时间/日期匹配的时候，该位会通过硬件置 1。可以通过向该位软件写 0 来清除。
7	INITM	进入初始化模式 0：自由运行模式 1：进入初始化模式设置时间/日期和预分频，计数器将停止运行
6	INITF	初始化状态标志 该位被硬件置 1，初始化状态时可以设置日历寄存器和预分频器。 0：日历寄存器和预分频器的值不能改变 1：日历寄存器和预分频器的值可以改变
5	RSYNF	寄存器同步标志 每 2 个 RTCCLK 将会由硬件置 1 一次，同时会复制当前日历时间/日期到影子日历寄存器。初始化模式(INITM)，移位操作挂起标志(SOPF)或者禁止影子寄存器模式(BPSHAD = 1)会清除该位。该位也可以通过软件写 0 清除。 0：影子寄存器未同步 1：影子寄存器已同步
4	YCM	年份配置标志 当日历寄存器的年份值不为 0 时硬件置 1 0：日历尚未初始化 1：日历已经初始化
3	SOPF	移位功能操作挂起标志 0：移位操作没有挂起

		1：移位操作挂起
2:1	保留	必须保持复位值
0	ALRM0WF	Alarm 0 配置可写标志 硬件置位和清零。ALRM0EN=0 时，标记 alarm 是否可写。 0：不允许修改 Alarm 寄存器设置 1：允许修改 Alarm 寄存器设置

14.4.5. 预分频寄存器 (RTC_PSC)

地址偏移：0x10
系统复位：无影响
备份域复位值：0x007F 00FF

写保护寄存器，仅在初始化状态可以进行写操作。
该寄存器只能按字(32位)访问。

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

位/位域	名称	描述
31:23	保留	必须保持复位值
22:16	FACTOR_A[6:0]	异步预分频系数 ck_apre 频率 = RTCCLK 频率/(FACTOR_A+1)
15	保留	必须保持复位值
14:0	FACTOR_S[14:0]	同步预分频系数 ck_spre 频率 = ck_apre 频率/(FACTOR_S+1)

14.4.6. 闹钟 0 时间日期寄存器 (RTC_ALRM0TD)

地址偏移：0x1C
系统复位：无影响
备份域复位值：0x0000 0000

写保护寄存器，仅在初始化状态可以进行写操作。
该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MSKD	DOWS	DAYT[1:0]		DAYU[3:0]				MSKH	PM	HRT[1:0]		HRU[3:0]			
rw	rw	rw		rw				rw	rw	rw		rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

MSKM	MNT[2:0]	MNU[3:0]	MSKS	SCT[2:0]	SCU[3:0]
rw	rw	rw	rw	rw	rw

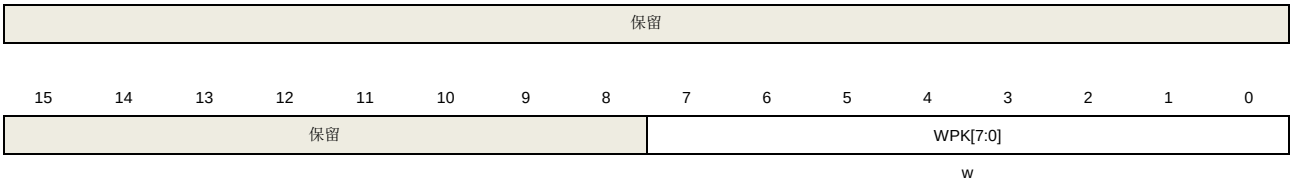
位/位域	名称	描述
31	MSKD	闹钟日期位域屏蔽位 0：不屏蔽日期/天位域 1：屏蔽日期/天位域
30	DOWS	星期选择 0：此时 DAYU[3:0] 代表日期个位值 1：此时 DAYU[3:0] 代表星期几，此时 DAYT[1:0]无意义
29:28	DAYT[1:0]	日期十位值，以 BCD 码格式存储
27:24	DAYU[3:0]	日期个位值或星期天数，以 BCD 码格式存储
23	MSKH	闹钟小时位域屏蔽位 0：不屏蔽小时位域 1：屏蔽小时位域
22	PM	AM/PM 标志 0：AM 或 24 小时制 1：PM
21:20	HRT[1:0]	小时十位值，以 BCD 码形式存储
19:16	HRU[3:0]	小时个位值，以 BCD 码形式存储
15	MSKM	闹钟分钟位域屏蔽位 0：不屏蔽分钟位域 1：屏蔽分钟位域
14:12	MNT[2:0]	分钟十位值，以 BCD 码形式存储
11:8	MNU[3:0]	分钟个位值，以 BCD 码形式存储
7	MSKS	闹钟秒位域屏蔽位 0：不屏蔽秒位域 1：屏蔽秒位域
6:4	SCT[2:0]	秒钟十位值，以 BCD 码形式存储
3:0	SCU[3:0]	秒钟个位值，以 BCD 码形式存储

14.4.7. 写保护钥匙寄存器 (RTC_WPK)

地址偏移：0x24
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

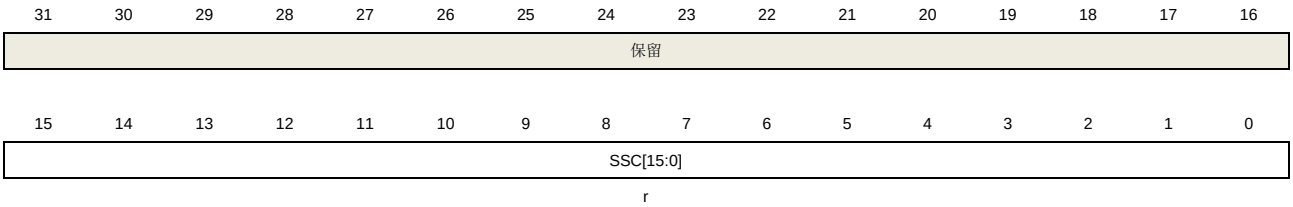


位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	WPK[7:0]	写保护的解锁值

14.4.8. 亚秒寄存器(RTC_SS)

地址偏移：0x28
系统复位值：当BPSHAD = 0，0x0000 0000。
当BPSHAD = 1，无影响。

该寄存器只能按字(32位)访问。

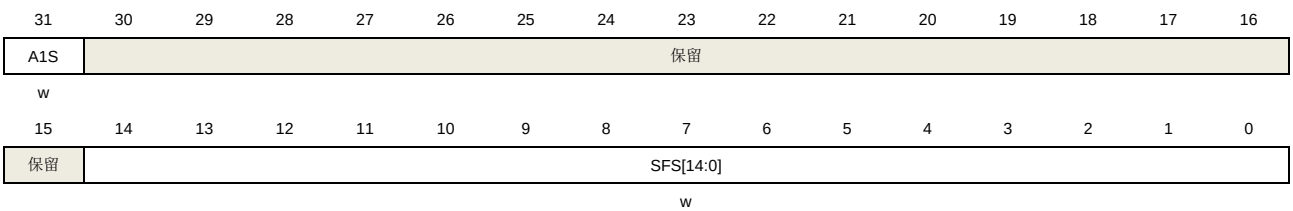


位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	SSC[15:0]	亚秒值 该位值是同步预分频计数器的值。秒的小数部分由下面公式给出： 秒的小数部分 = (FACTOR_S - SSC) / (FACTOR_S + 1)

14.4.9. 移位控制寄存器 (RTC_SHIFTCTL)

地址偏移：0x2C
系统复位：无影响
备份域复位值：0x0000 0000

写保护寄存器，仅当SOPF=0，该寄存器可写。
该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31	A1S	增加一秒 0：无影响 1：增加一秒到时钟/日历 该位与 SFS 位一起使用，增加小于一秒到当前时间。
30:15	保留	必须保持复位值
14:0	SFS[14:0]	减去小于一秒的一段时间 这位的值将增加到同步预分频计数器 当仅用 SFS 时，由于同步预分频器是一个递减计数器，所以时钟将会延迟。 延迟(秒) = SFS / (FACTOR_S + 1) 当 A1S 和 SFS 一起使用时，时钟将会提前 提前(秒) = (1 - (SFS / (FACTOR_S + 1)))

注意：写入此寄存器会导致 RSYNF 位被清 0。

14.4.10. 时间戳时间寄存器 (RTC_TTS)

地址偏移：0x30
备份域复位值：0x0000 0000
系统复位：无影响

当TSF被置1，该位用来记录日历时间。
清除TSF位也会清除此寄存器。
该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留									PM	HRT[1:0]		HRU[3:0]			
									r	r		r			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	MNT[2:0]			MNU[3:0]				保留	SCT[2:0]			SCU[3:0]			
	r			r					r			r			

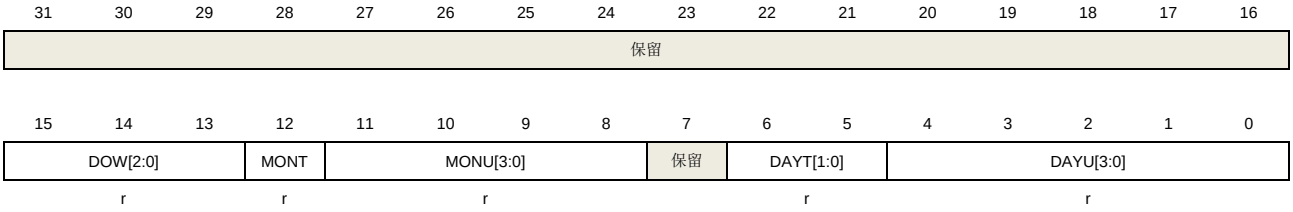
位/位域	名称	描述
31:23	保留	必须保持复位值
22	PM	AM/PM 标记 0：AM 或 24 小时制 1：PM
21:20	HRT[1:0]	小时十位值，以 BCD 码形式存储
19:16	HRU[3:0]	小时个位值，以 BCD 码形式存储
15	保留	必须保持复位值
14:12	MNT[2:0]	分钟十位值，以 BCD 码形式存储

11:8	MNU[3:0]	分钟个位值，以 BCD 码形式存储
7	保留	必须保持复位值
6:4	SCT[2:0]	秒钟十位值，以 BCD 码形式存储
3:0	SCU[3:0]	秒钟个位值，以 BCD 码形式存储

14.4.11. 时间戳日期寄存器 (RTC_DTS)

地址偏移：0x34
备份域复位值：0x0000 0000
系统复位：无影响

当TSF被置1，该位用来记录日历日期。
清除TSF位也会清除此寄存器。
该寄存器只能按字(32位)访问。

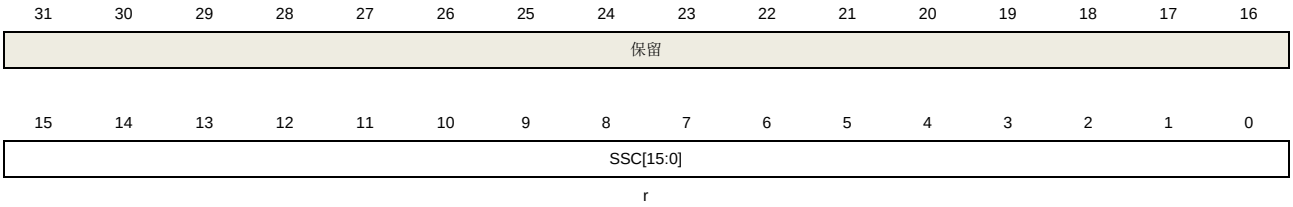


位/位域	名称	描述
31:16	保留	必须保持复位值
15:13	DOW[2:0]	星期数
12	MONT	月份十位值，以 BCD 码形式存储
11:8	MONU[3:0]	月份个位值，以 BCD 码形式存储
7	保留	必须保持复位值
6:5	DAYT[1:0]	日期十位值，以 BCD 码形式存储
4:0	DAYU[3:0]	日期个位值，以 BCD 码形式存储

14.4.12. 时间戳亚秒寄存器 (RTC_SSTS)

地址偏移：0x38
备份域复位：0x0000 0000
系统复位：无影响

当TSF被置1，该位用来记录日历时间。
清除TSF位也会清除此寄存器。
该寄存器只能按字(32 位)访问。

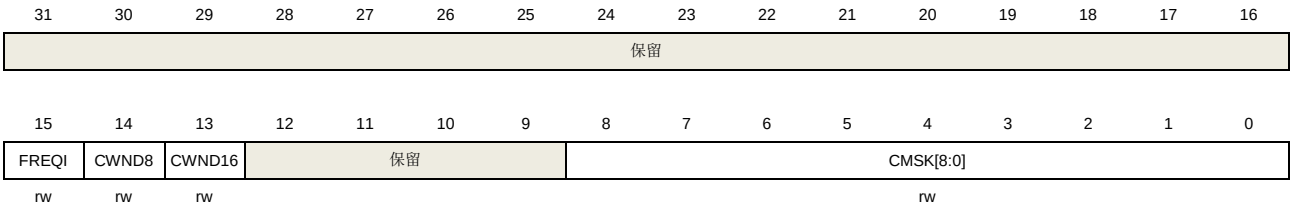


位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	SSC[15:0]	亚秒值 TSF 置 1 时记录当时的同步预分频计数器的值。

14.4.13. 高精度频率补偿寄存器 (RTC_HRFC)

地址偏移：0x3C
备份域复位：0x0000 0000
系统复位：无影响

写保护寄存器。
该寄存器只能按字(32位)访问。



位/位域	名称	描述
31:16	保留	必须保持复位值
15	FREQI	RTC 频率增加 488.5ppm 0：无影响 1：每 2^{11} 个脉冲增加一个 RTCCLK 脉冲 该位需与 CMSK 位一起使用。如果输入时钟频率是 32.768KHz，在 32s 校准窗期间，增加的 RTCCLK 脉冲数是 (512 * FREQI) - CMSK
14	CWND8	采用 8 秒校准周期 0：无影响 1：采用 8 秒校准周期 **注意：**当 CWND8=1，CMSK[1:0] 被锁定在“00”。
13	CWND16	采用 16 秒校准周期 0：无影响 1：采用 16 秒校准周期 **注意：**当 CWND16=1，CMSK[0] 被锁定在“0”.

12:9	保留	必须保持复位值
8:0	CMSK[8:0]	校准周期 RTCCLK 脉冲屏蔽数 在 2^{20} RTCCLK 脉冲之内屏蔽的脉冲数 此项功能可以以 0.9537 ppm 的分辨率来降低日历频率

14.4.14. 侵入寄存器 (RTC_TAMP)

地址偏移：0x40
备份域复位：0x0000 0000
系统复位：无影响

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								PC15MDE	PC15VAL	PC14MDE	PC14VAL	PC13MDE	PC13VAL	保留	
								rw	rw	rw	rw	rw	rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DISPU	PRCH[1:0]		FLT[1:0]		FREQ[2:0]			TPTS	保留		TP1EG	TP1EN	TPIE	TP0EG	TP0EN
rw	rw		rw		rw			rw			rw	rw	rw	rw	rw

位/位域	名称	描述
31:24	保留	必须保持复位值
23	PC15MDE	PC15 模式 0：无影响 1：如果 LXTAL 禁用，强制 PC15 推挽输出
22	PC15VAL	PC15 值 当 LXTAL 禁用且 PC15MDE=1 时，PC15 输出该位数据
21	PC14MDE	PC14 模式 0：无影响 1：如果 LXTAL 禁用，强制 PC14 推挽输出
20	PC14VAL	PC14 值 当 LXTAL 禁用且 PC14MDE=1 时，PC14 输出该位数据
19	PC13MDE	PC13 模式 0：无影响 1：当 RTC 所有备用功能禁用时，PC13 输出该位数据
18	PC13VAL	PC13 值或闹钟输出类型值 PC13 输出闹钟 0：PC13 开漏输出 1：PC13 推挽输出 当 RTC 所有备用功能禁用且 PC13MDE=1 时 0：PC13 输出 0

		1：PC13 输出 1
17:16	保留	必须保持复位值
15	DISPU	RTC_TAMPx 上拉禁用位 0：使能内部 RTC_TAMPx 引脚上的上拉电阻并在采样前进行预充电 1：禁用预充电功能
14:13	PRCH[1:0]	RTC_TAMPx 的预充电时间 该位设置决定了每次采样前的预充电时间 0x0：1 个 RTC 时钟 0x1：2 个 RTC 时钟 0x2：4 个 RTC 时钟 0x3：8 个 RTC 时钟
12:11	FLT[1:0]	RTC_TAMPx 过滤器计数设置 该位决定了侵入事件检测模式和在电平检测模式下连续采样的次数。 0x0：用边沿模式检测侵入事件，预充电功能被自动禁用。 0x1：用电平模式检测侵入事件。连续采样到 2 个有效电平时认为发生侵入事件 0x2：用电平模式检测侵入事件。连续采样到 4 个有效电平时认为发生侵入事件 0x3：用电平模式检测侵入事件。连续采样到 8 个有效电平时认为发生侵入事件
10:8	FREQ[2:0]	侵入事件电平模式检测的采样频率 0x0：每次采样间隔 32768 个 RTCCLK(若 RTCCLK=32.768KHz，频率为 1Hz) 0x1：每次采样间隔 16384 个 RTCCLK(若 RTCCLK=32.768KHz，频率为 2Hz) 0x2：每次采样间隔 8192 个 RTCCLK(若 RTCCLK=32.768KHz，频率为 4Hz) 0x3：每次采样间隔 4096 个 RTCCLK(若 RTCCLK=32.768KHz，频率为 8Hz) 0x4：每次采样间隔 2048 个 RTCCLK(若 RTCCLK=32.768KHz，频率为 16Hz) 0x5：每次采样间隔 1024 个 RTCCLK(若 RTCCLK=32.768KHz，频率为 32Hz) 0x6：每次采样间隔 512 个 RTCCLK(若 RTCCLK=32.768KHz，频率为 64Hz) 0x7：每次采样间隔 256 个 RTCCLK(若 RTCCLK=32.768KHz，频率为 128Hz)
7	TPTS	侵入事件时触发时间戳 0：无影响 1：当检测到侵入事件时，即使 TSEN=0，TSF 也会被置位
6:5	保留	必须保持复位值
4	TP1EG	TAMP1 输入管脚的侵入事件检测触发沿 如果侵入检测处于边沿模式(FLT =0)： 0：上升沿触发一个侵入检测事件 1：下降沿触发一个侵入检测事件 如果侵入检测处于电平模式(FLT !=0)： 0：低电平触发一个侵入检测事件 1：高电平触发一个侵入检测事件
3	TP1EN	Tamper1 检测使能位 0：禁用 Tamper1 检测功能

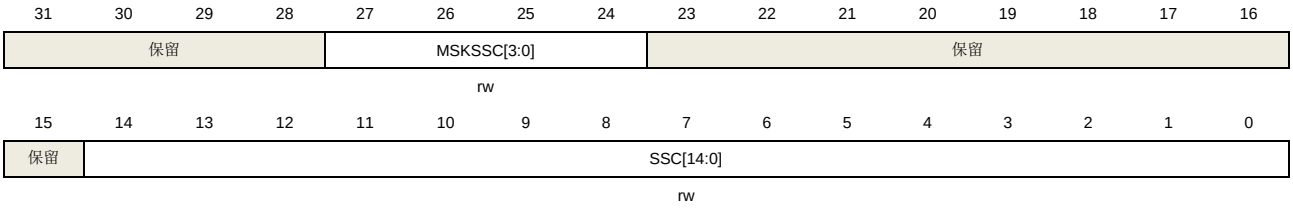
		1：启用 Tamper1 检测功能
2	TPIE	侵入检测中断使能 0：禁用侵入中断 1：启用侵入中断
1	TP0EG	TAMP0 输入管脚的侵入事件检测触发沿 如果侵入检测处于边沿模式(FLT =0)： 0：上升沿触发一个侵入检测事件 1：下降沿触发一个侵入检测事件 如果侵入检测处于电平模式(FLT !=0)： 0：低电平触发一个侵入检测事件 1：高电平触发一个侵入检测事件
0	TP0EN	Tamper0 检测使能位 0：禁用 Tamper0 检测功能 1：启用 Tamper0 检测功能

注意：强烈建议在改变侵入检测配置之前，应该复位 TPxEN 位。

14.4.15. 闹钟 0 亚秒寄存器 (RTC_ALRM0SS)

地址偏移：0x44
备份域复位：0x0000 0000
系统复位：无影响

写保护寄存器，仅当ALRM0EN=0或INITM=1，可以进行写操作。
该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31:28	保留	必须保持复位值
27:24	MSKSSC[3:0]	亚秒位域的屏蔽控制位 0x0：屏蔽闹钟亚秒设置。当所有其他的闹钟位域匹配的时候，闹钟将会在每一秒钟到达的时刻置 1。 0x1：SSC[0]位用于时间匹配，其他位被忽略。 0x2：SSC[1:0] 位用于时间匹配，其他位被忽略。 0x3：SSC[2:0] 位用于时间匹配，其他位被忽略。 0x4：SSC[3:0] 位用于时间匹配，其他位被忽略。 0x5：SSC[4:0] 位用于时间匹配，其他位被忽略。 0x6：SSC[5:0] 位用于时间匹配，其他位被忽略。

0x7：SSC[6:0] 位用于时间匹配，其他位被忽略。
0x8：SSC[7:0] 位用于时间匹配，其他位被忽略。
0x9：SSC[8:0] 位用于时间匹配，其他位被忽略。
0x10：SSC[9:0] 位用于时间匹配，其他位被忽略。
0x11：SSC[10:0] 位用于时间匹配，其他位被忽略。
0x12：SSC[11:0] 位用于时间匹配，其他位被忽略。
0x13：SSC[12:0] 位用于时间匹配，其他位被忽略。
0x14：SSC[13:0] 位用于时间匹配，其他位被忽略。
0x15：SSC[14:0] 位用于时间匹配，其他位被忽略。
注意：同步预分频计数器的第 15 位(RTC_SS 寄存器中的 SSC[15])从不被匹配。

23:15	保留	必须保持复位值
14:0	SSC[14:0]	闹钟亚秒值 该值为闹钟亚秒值，用于与同步预分频计数器匹配。 匹配位数由 MSKSSC 位控制。

14.4.16. 备份寄存器 (RTC_BKPx) (x=0..4)

地址偏移：0x50到0x60
备份域复位：0x0000 0000
系统复位：无影响

该寄存器只能按字(32位)访问。

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

位/位域	名称	描述
31:0	DATA[31:0]	数据 软件可读写寄存器。由于此寄存器可由 V_{BAT} 供电，因此寄存器值在省电模式下依然保持有效。当侵入检测标志位 TPxF 置 1 ，这些寄存器会被复位。当 FMC 读保护功能禁用时，这些寄存器会被复位。

15. 定时器（TIMER）

表 15-1. 定时器（TIMERx）分为六种类型

定时器	定时器 0	定时器 1/2	定时器 13	定时器 14	定时器 15/16	定时器 5
类型	高级	通用 L0	通用 L2	通用 L3	通用 L4	基本
预分频器	16 位	16 位	16 位	16 位	16 位	16 位
计数器	16 位	32 位(定时器 1) 16 位(定时器 2)	16 位	16 位	16 位	16 位
计数模式	向上，向下，中央对齐	向上，向下，中央对齐	只有向上	只有向上	只有向上	只有向上
可重复性	●	×	×	●	●	×
捕获/比较通道数	4	4	1	2	1	0
互补和死区时间	●	×	×	●	●	×
中止输入	●	×	×	●	●	×
单脉冲	●	●	×	●	●	●
正交译码器	●	●	×	×	×	×
主-从管理	●	●	×	●	×	×
内部连接	●(1)	●(2)	×	●(3)	×	TRGO TO DAC
DMA	●	●	×	●	●	●(4)
Debug 模式	●	●	●	●	●	●

(1) TIMER0 **ITI0:** TIMER14_TRGO **ITI1:** TIMER1_TRGO **ITI2:** TIMER2_TRGO **ITI3:** 0

(2) TIMER1 **ITI0:** TIMER0_TRGO **ITI1:** TIMER14_TRGO **ITI2:** TIMER2_TRGO **ITI3:** 0
TIMER2 **ITI0:** TIMER0_TRGO **ITI1:** TIMER1_TRGO **ITI2:** TIMER14_TRGO **ITI3:** 0

(3) TIMER14 **ITI0:** TIMER1_TRGO **ITI1:** TIMER2_TRGO **ITI2:** 0 **ITI3:** 0

(4) 只有更新事件可以产生 DMA 请求。但是定时器 5 中没有 DMA 配置寄存器。

15.1. 高级定时器（TIMERx,x=0）

15.1.1. 简介

高级定时器（TIMER0）是四通道定时器，支持输入捕获和输出比较。可以产生 PWM 信号控制电机和电源管理。高级定时器含有一个 16 位无符号计数器。

高级定时器是可编程的，可以被用来计数，其外部事件可以驱动其他定时器

高级定时器包含了一个死区时间插入模块，非常适合电机控制。

定时器和定时器之间是相互独立，但是他们可以被同步在一起形成一个更大的定时器，这些定时器的计数器一致地增加。

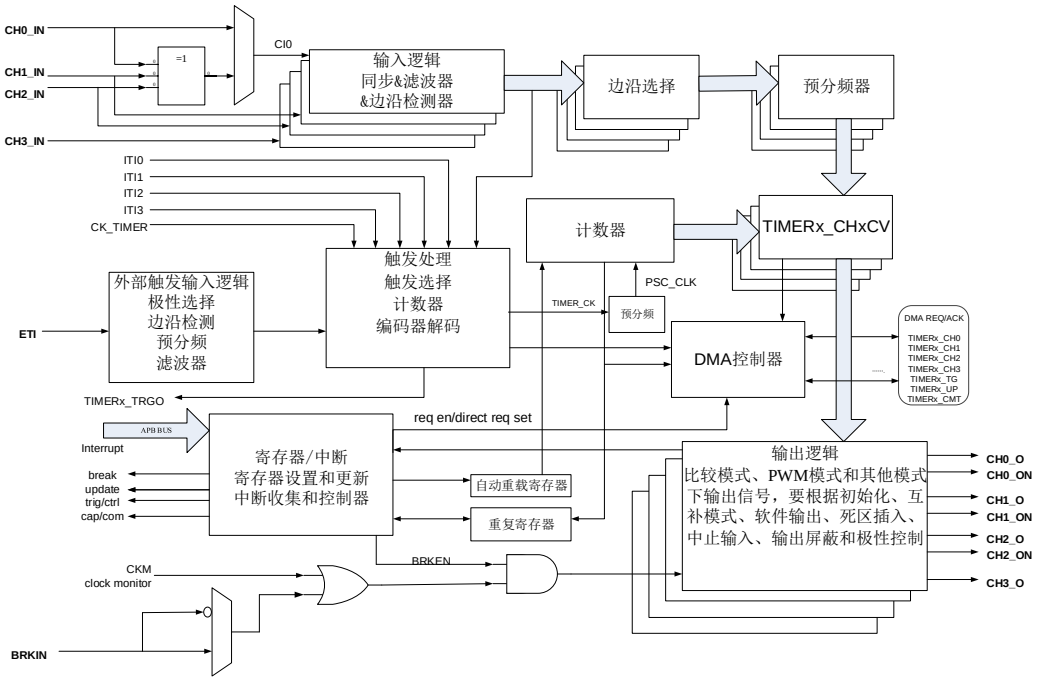
15.1.2. 主要特性

- 总通道数：4；
- 计数器宽度：16位；
- 时钟源可选：内部时钟，内部触发，外部输入，外部触发；
- 多种计数模式：向上计数，向下计数和中央计数；
- 正交译码器接口：被用来追踪运动和分辨旋转方向和位置；
- 霍尔传感器接口：用来做三相电机控制；
- 可编程的预分频器：16位，运行时可以被改变；
- 每个通道可配置：输入捕获模式，输出比较模式，可编程的PWM模式，单脉冲模式；
- 可编程的死区时间；
- 自动重装载功能；
- 可编程的计数器重复功能；
- 中止输入功能；
- 中断输出和DMA请求：更新事件，触发事件，比较/捕获事件，换相事件和中止事件；
- 多个定时器的菊链使得一个定时器可以同时启动多个定时器；
- 定时器的同步允许被选择的定时器在同一个时钟周期开始计数；
- 定时器主-从管理。

15.1.3. 结构框图

*图 15-1. 高级定时器结构框图*提供了高级定时器的内部配置细节

图 15-1. 高级定时器结构框图



15.1.4. 功能描述

时钟源配置

高级定时器可以由内部时钟源 CK_TIMER 或者由 SMC（TIMERx_SMCFG 寄存器位[2:0]）控制的复用时钟源驱动。

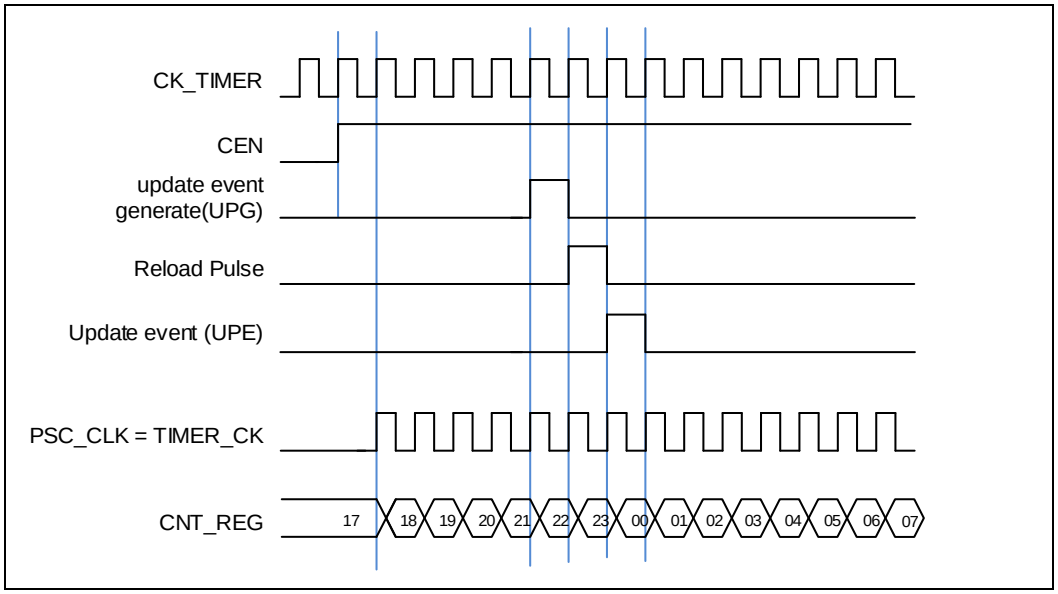
- SMC[2:0]==3’b000，定时器选择内部时钟源（连接到RCU模块的CK_TIMER）

如果 SMC[2:0]==3’b000，默认用来驱动计数器预分频器的是内部时钟源 CK_TIMER。当 CEN 置位，CK_TIMER 经过预分频器（预分频值由 TIMERx_PSC 寄存器确定）产生 PSC_CLK。

这种模式下，驱动预分频器计数的 TIMER_CK 等于来自于 RCU 模块的 CK_TIMER

如果将 TIMERx_SMCFG 寄存器的 SMC[2:0]设置为 0x1、0x2、0x3 和 0x7，预分频器被其他时钟源(由 TIMERx_SMCFG 寄存器的 TRGS [2:0]区域选择)驱动，在下文说明。当 SMC 位被设置为 0x4、0x5 和 0x6，计数器预分频器时钟源由内部时钟 CK_TIMER 驱动。

图 15-2. 内部时钟分频为 1 时，计数器的时序图



- SMC[2:0]==3'b111(外部时钟模式0)，定时器选择外部输入引脚作为时钟源

计数器预分频器可以在 TIMERx_CI0/ TIMERx_CI1 引脚的每个上升沿或下降沿计数。这种模式可以通过设置 SMC [2:0]为 0x7 同时设置 TRGS[2:0]为 0x4，0x5 或 0x6 来选择。

计数器预分频器也可以在内部触发信号 ITI0/1/2/3 的上升沿计数。这种模式可以通过设置 SMC [2:0]为 0x7 同时设置 TRGS [2:0]为 0x0, 0x1, 0x2 或者 0x3。

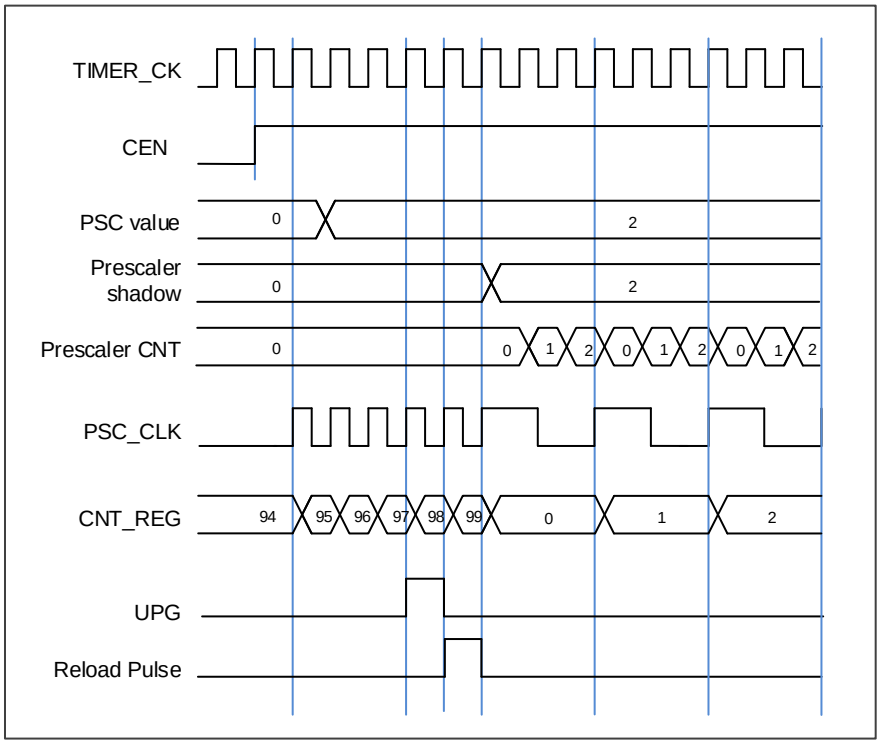
- SMC1==1'b1(外部时钟模式1)，定时器选择外部输入引脚ETI作为时钟源

计数器预分频器可以在外部引脚 ETI 的每个上升沿或下降沿计数。这种模式可以通过设置 TIMERx_SMCFG 寄存器中的 SMC1 位为 1 来选择。另一种选择 ETI 信号作为时钟源方式是，设置 SMC [2:0]为 0x7 同时设置 TRGS [2:0]为 0x7。注意 ETI 信号是通过数字滤波器采样 ETI 引脚得到的。如果选择 ETI 信号为时钟源，触发控制器包括边沿监测电路将在每个 ETI 信号上升沿产生一个时钟脉冲来为计数器预分频器提供时钟。

时钟预分频器

预分频器可以将定时器的时钟（TIMER_CK)频率按 1 到 65536 之间的任意值分频，分频后的时钟 PSC_CLK 驱动计数器计数。分频系数受预分频寄存器 TIMERx_PSC 控制，这个控制寄存器带有缓冲器，它能够在运行时被改变。新的预分频器的参数在下一次更新事件到来时被采用。

图 15-3. 当 PSC 数值从 0 变到 2 时，计数器的时序图



计数器向上计数模式

在这种模式，计数器的计数方向是向上计数。计数器从 0 开始向上连续计数到计数器自动加载值（定义在 TIMERx_CAR 寄存器中），一旦计数器计数到自动加载值，会重新从 0 开始向上计数，并且产生上溢事件。另外，在(TIMERx_CREP+1)次上溢后将会产生更新事件。在向上计数模式中，TIMERx_CTL0 寄存器中的计数方向控制位 DIR 应该被设置成 0。

当通过 TIMERx_SWEVG 寄存器的 UPG 位置 1 来设置更新事件时，计数值会被清 0，并产生更新事件。

如果 TIMERx_CTL0 寄存器的 UPDIS 置 1，则禁止更新事件。

当发生更新事件时，所有影子寄存器(重复计数器，计数器自动重载寄存器，预分频寄存器)都将被更新。

图 15-4. 向上计数时序图,PSC=0/2 和 *图 15-5. 向上计数时序图,在运行时改变 TIMERx_CAR 寄存器的值* 给出了一些例子，当 TIMERx_CAR=0x99 时，计数器在不同预分频因子下的行为。

图 15-4. 向上计数时序图，PSC=0/2

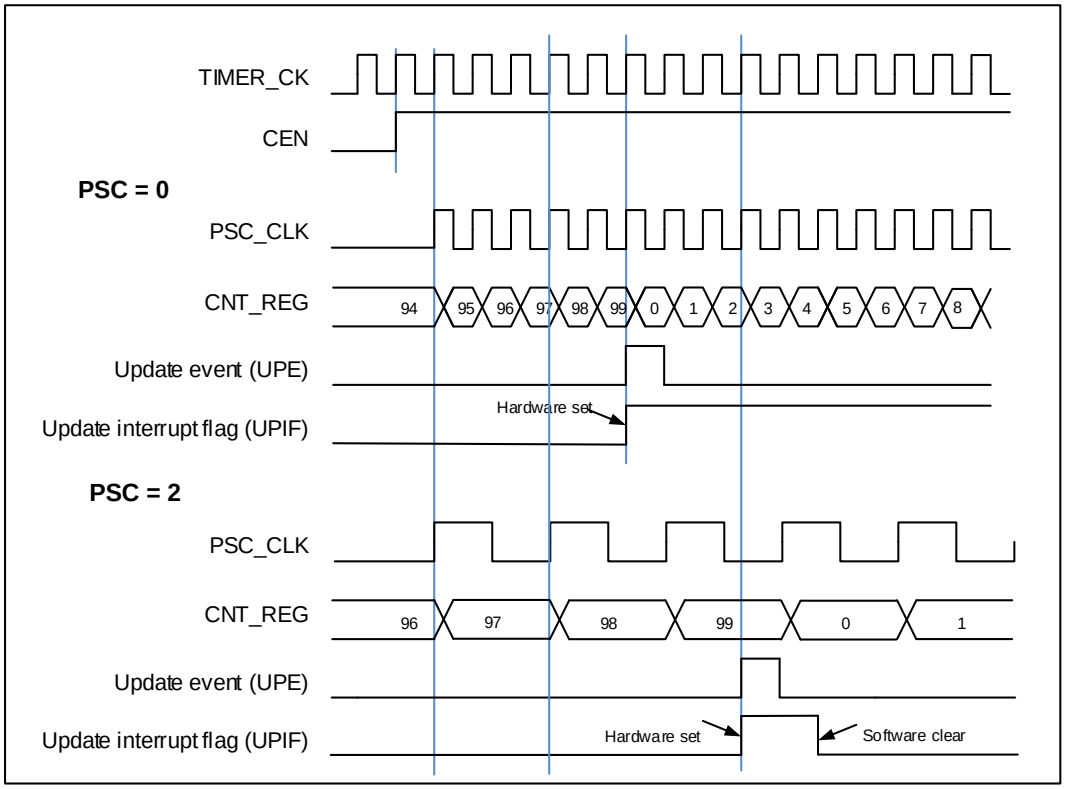
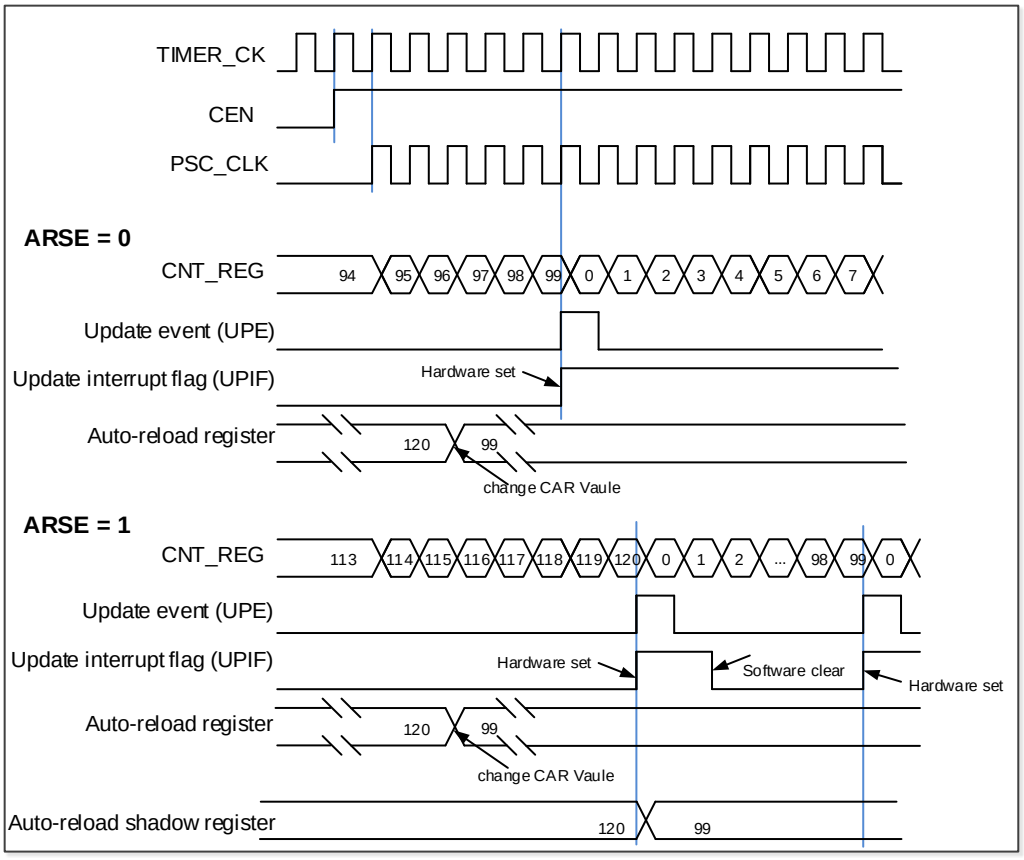


图 15-5. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值



计数器向下计数模式

在这种模式，计数器的计数方向是向下计数。计数器从自动加载值（定义在TIMERx_CAR寄存器中）向下连续计数到0。一旦计数器计数到0，计数器会重新从自动加载值开始计数并且产生下溢事件。另外，在(TIMERx_CREP+1)次下溢后将会产生更新事件。在向下计数模式中，TIMERx_CTL0寄存器中的计数方向控制位DIR应该被设置成1。

当通过 TIMERx_SWEVG 寄存器的 UPG 位置 1 来设置更新事件时，计数值会被初始化为自动加载值，并产生更新事件。

如果 TIMERx_CTL0 寄存器的 UPDIS 置 1，则禁止更新事件。

当发生更新事件时，所有影子寄存器（重复计数器，计数器自动重载寄存器，预分频寄存器）都将被更新。

*图15-6. 向下计数时序图,PSC=0/2*和*图15-7. 向下计数时序图,在运行时改变TIMERx_CAR寄存器值*给出了一些例子，当 TIMERx_CAR=0x99 时，计数器在不同时钟频率下的行为。

图 15-6. 向下计数时序图，PSC=0/2

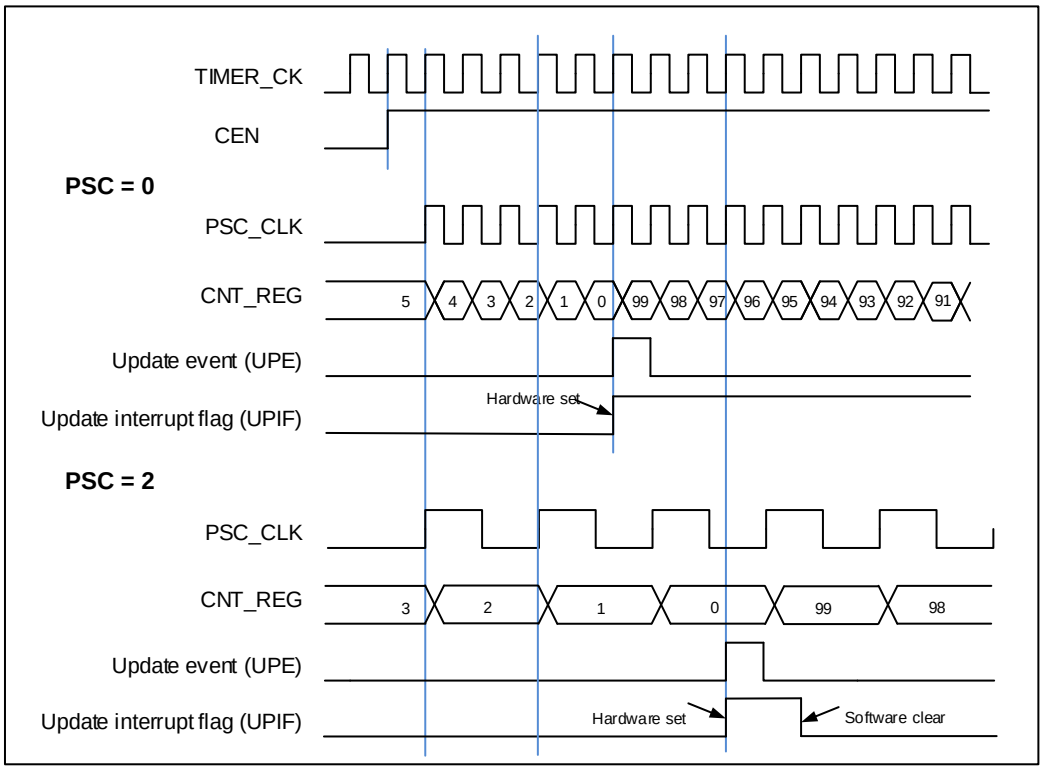
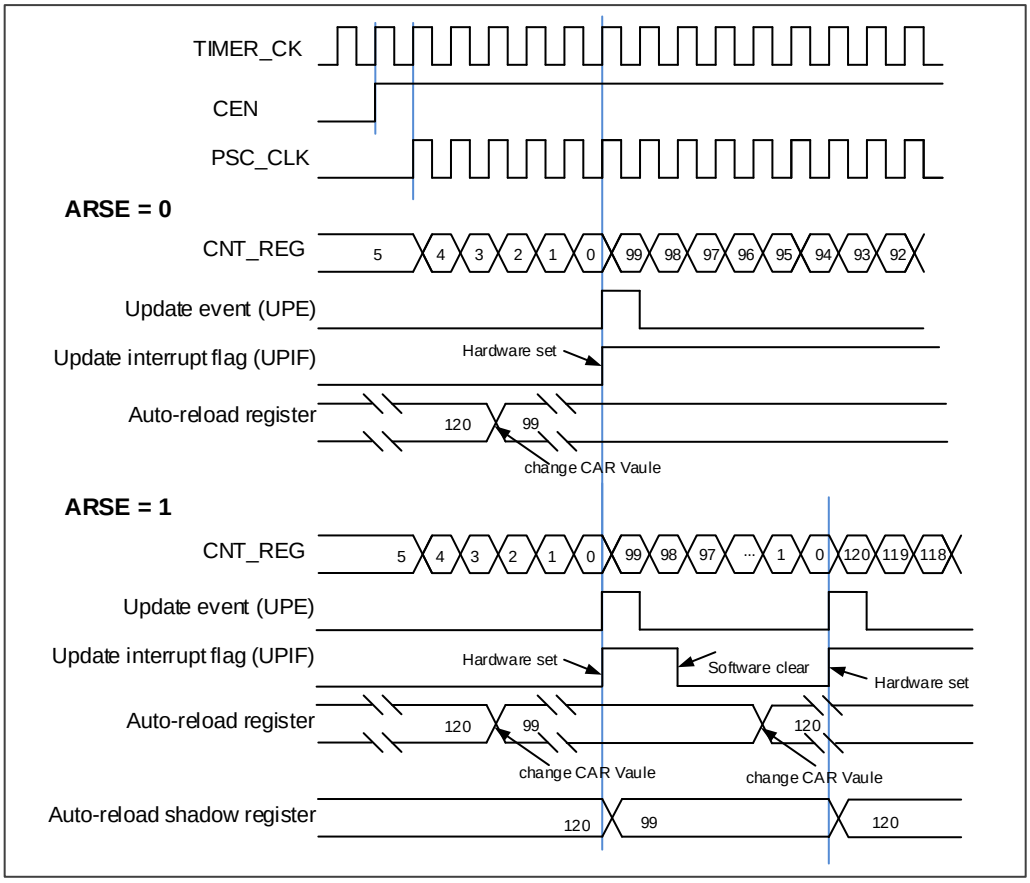


图 15-7. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值



计数器中央对齐模式

在中央对齐模式下，计数器交替的从 0 开始向上计数到自动加载值，然后再向下计数到 0。。向上计数模式中， 定时器模块在计数器计数到自动加载值-1 产生一个上溢事件；向下计数模式中，定时器模块在计数器计数到 1 时产生一个下溢事件。在中央计数模式中，TIMERx_CTL0 寄存器中的计数方向控制位 DIR 只读，表明了的计数方向。

将 TIMERx_SWEVG 寄存器的 UPG 位置 1 可以初始化计数值为 0，并产生一个更新事件，而无需考虑计数器在中央模式下是向上计数还是向下计数。

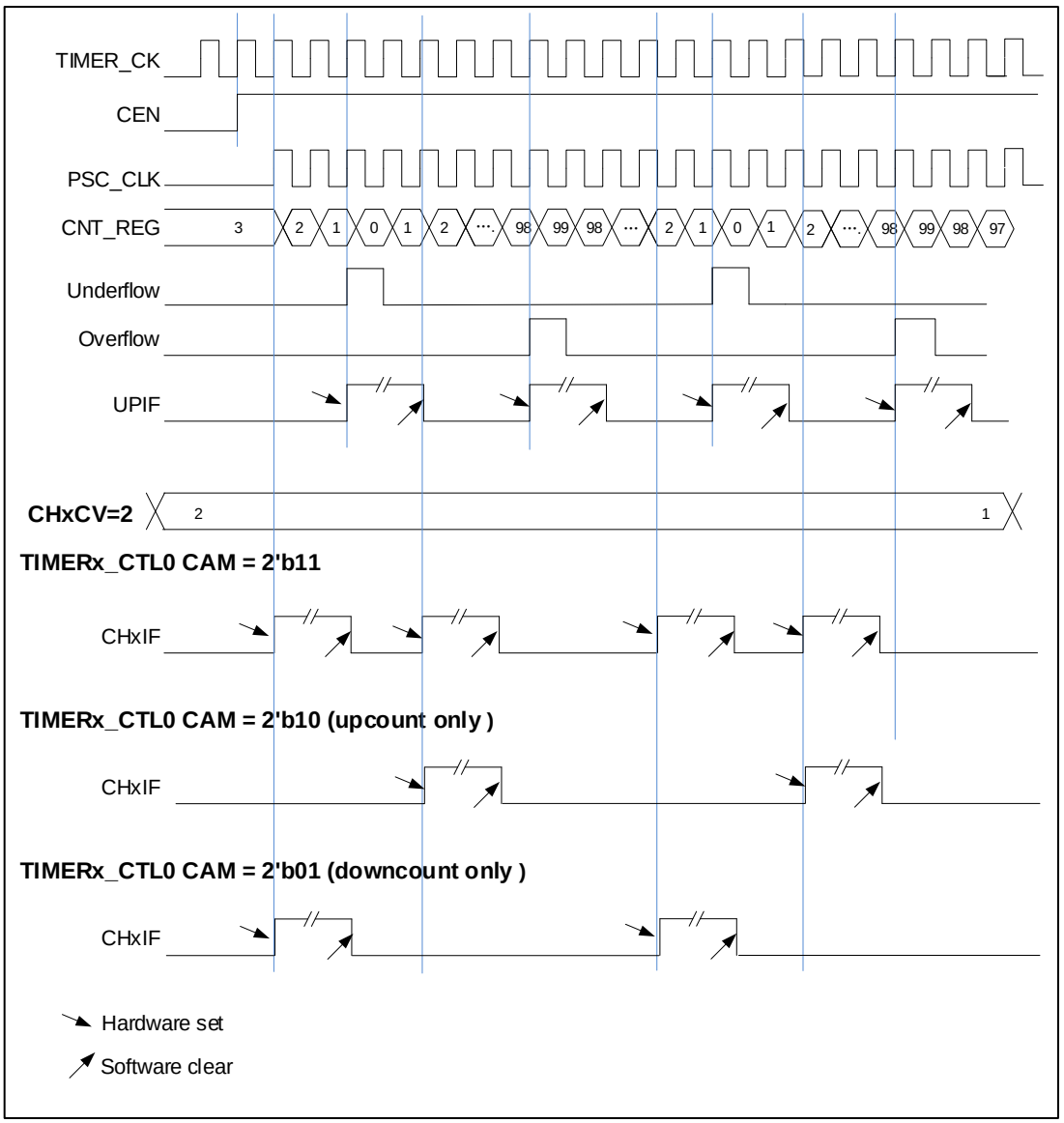
上溢或者下溢时，TIMERx_INTF 寄存器中的 UPIF 位都会被置 1，然而 CHxIF 位置 1 与 TIMERx_CTL0 寄存器中 CAM 的值有关。具体细节参考*图15-8. 中央计数模式计数器时序图*。

如果 TIMERx_CTL0 寄存器的 UPDIS 置 1，则禁止更新事件。

当发生更新事件时，所有影子寄存器（重复计数器，计数器自动重载寄存器，预分频寄存器）都将被更新。

*图 15-8. 中央计数模式计数器时序图*给出了一些例子，当 TIMERx_CAR=0x99，TIMERx_PSC=0x0 时，计数器的行为。

图 15-8. 中央计数模式计数器时序图



更新事件（来自上溢/下溢）频率配置

更新事件的生成频率（来自上溢和下溢事件）可以通过TIMERx_CREP寄存器进行配置。重复计数器是用来在N+1个计数周期之后产生更新事件，更新定时器的寄存器，N为TIMERx_CREP寄存器的CREP。重复计数器在每次计数器上溢和下溢时递减（向上计数模式中不存在下溢事件；向下计数模式中不存在上溢事件）。

将TIMERx_SWEVG寄存器的UPG位置1可以重载TIMERx_CREP寄存器中CREP的值并产生一个更新事件。

新写入的 CREP 值将在下一次更新事件到来时生效。当 CREP 的值为奇数，并且计数器在中央对齐模式下计数时，更新事件发生在上溢或下溢取决于写入的 CREP 值何时生效。如果在写入奇数到 CREP 寄存器后由软件生成更新事件，则在下溢时产生更新事件。如果在写入奇数到 CREP 寄存器后下一个更新事件发生在上溢，此后将在上溢时产生更新事件。

图 15-9. 中央计数模式下计数器重复时序图

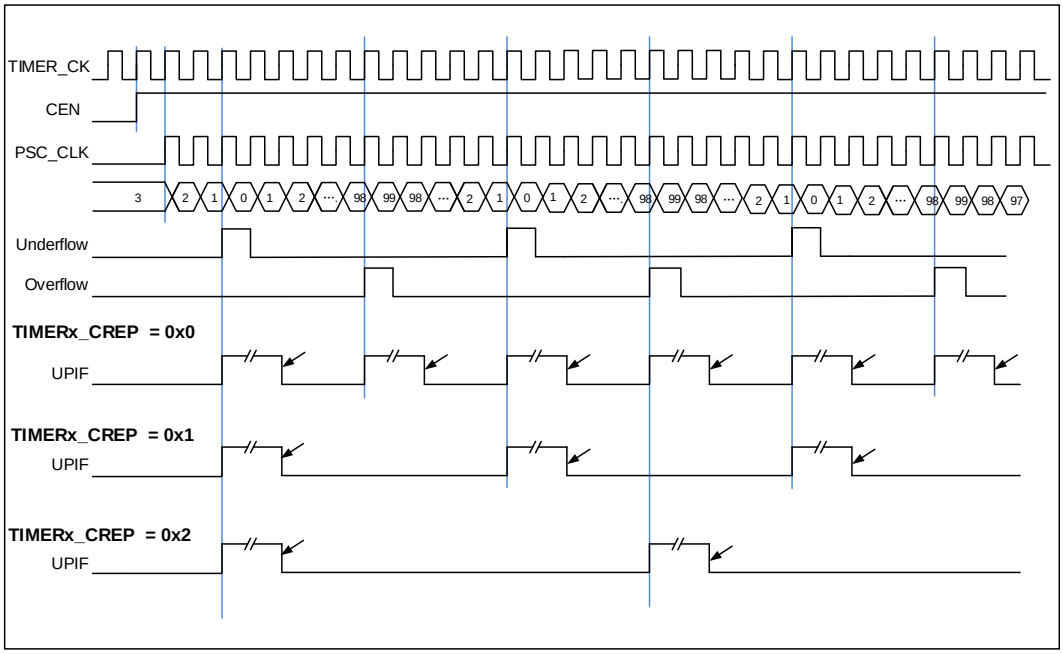


图 15-10. 在向上计数模式下计数器重复时序图

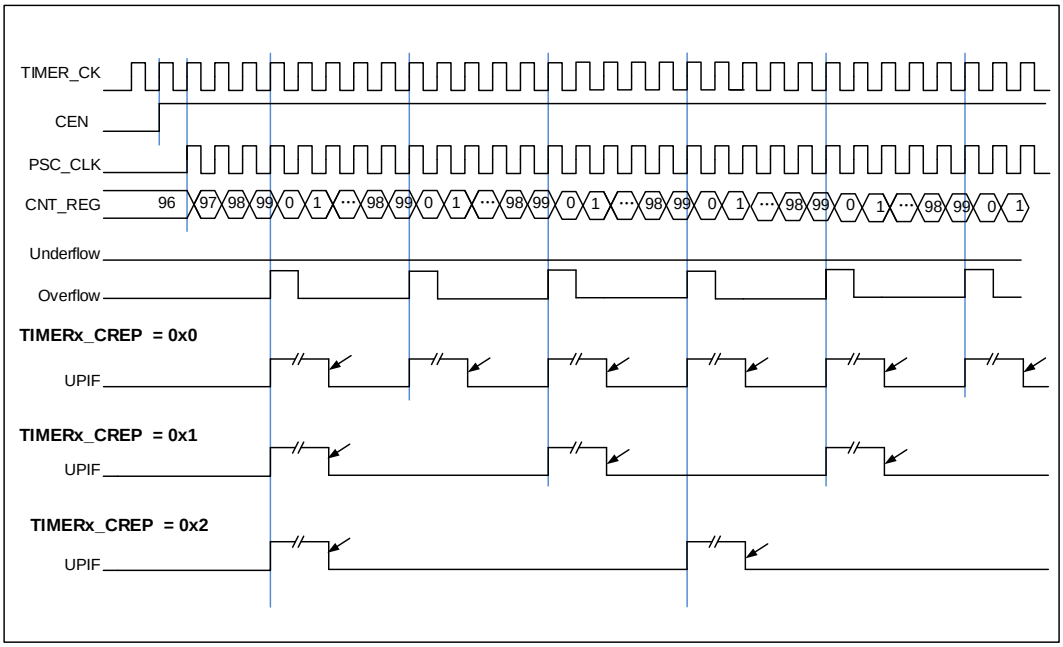
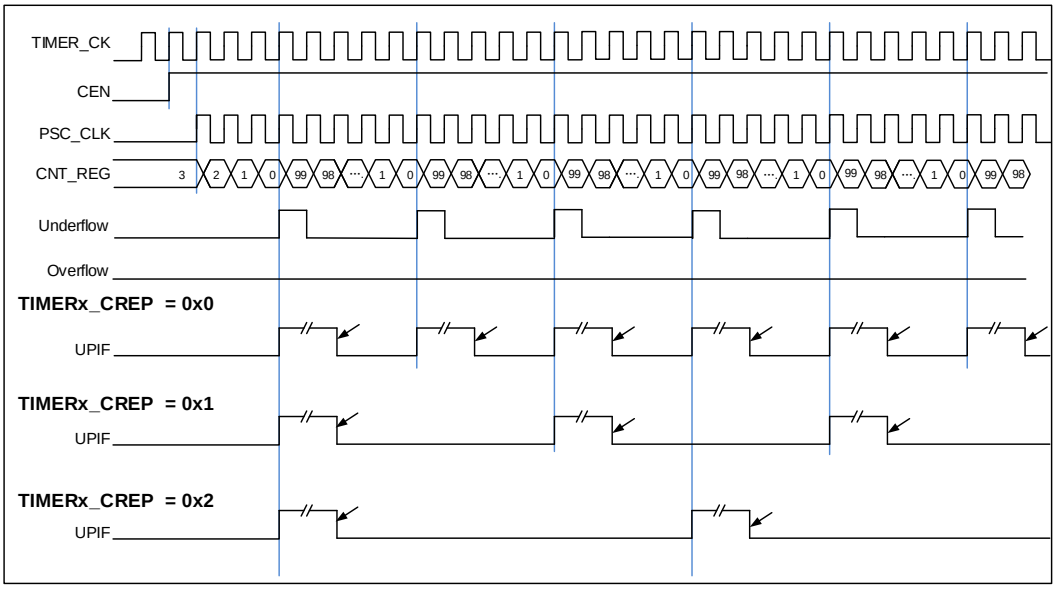


图 15-11. 在向下计数模式下计数器重复时序图



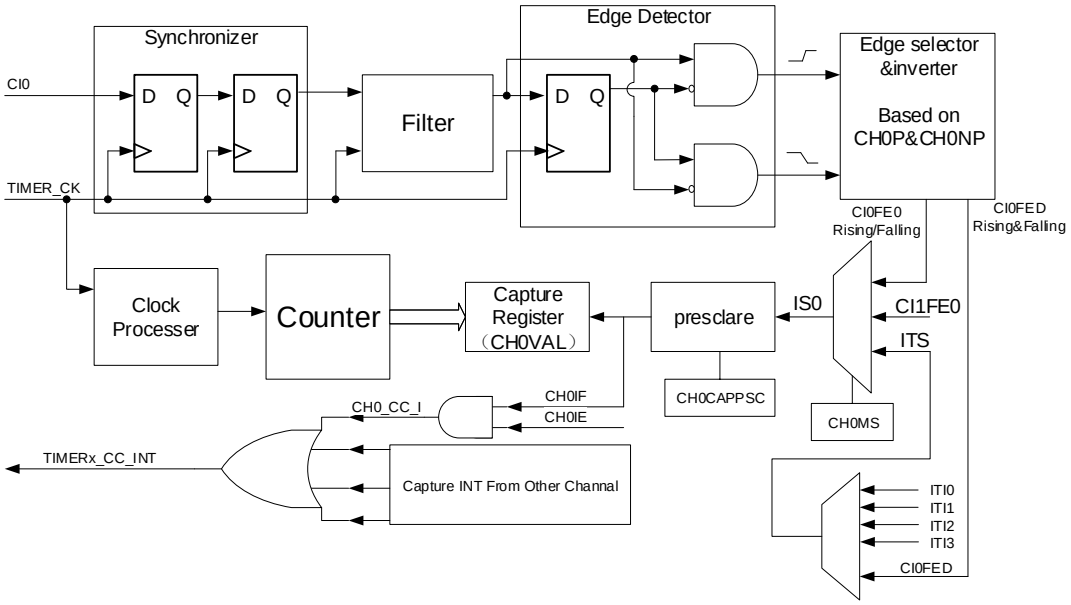
输入捕获和输出比较通道

高级定时器拥有四个独立的通道用于捕获输入或比较输出是否匹配。每个通道都围绕一个通道捕获比较寄存器建立，包括一个输入级，通道控制器和输出级。

通道输入捕获功能

通道输入捕获功能允许通道测量一个波形时序，频率，周期，占空比等。输入级包括一个数字滤波器，一个通道极性选择，边沿检测和一个通道预分频器。如果在输入引脚上出现被选择的边沿，TIMERx_CHxCV 寄存器会捕获计数器当前的值，同时 CHxIF 位被置 1，如果 CHxIE = 1 则产生通道中断。

图 15-12. 通道输入捕获原理



通道输入信号 CIx 有两种选择，一种是 TIMERx_CHx 信号，另一种是

TIMERx_CH0,TIMERx_CH1 和 TIMERx_CH2 异或之后的信号。通道输入信号 CIx 先被 TIMER_CK 信号同步，然后经过数字滤波器采样，产生一个被滤波后的信号。通过边沿检测器，可以选择检测上升沿或者下降沿。通过配置 CHxP 选择使用上升沿或者下降沿。配置 CHxMS.，可以选择其他通道的输入信号，内部触发信号。配置 IC 预分频器，使得若干个输入事件后才产生一个有效的捕获事件。捕获事件发生，CHxVAL 存储计数器的值。

配置步骤如下：

第一步：滤波器配置（TIMERx_CHCTL0寄存器中CHxCAPFLT）：
根据输入信号和请求信号的质量，配置相应的CHxCAPFLT。

第二步：边沿选择（TIMERx_CHCTL2寄存器中CHxP/CHxNP）：
配置CHxP/CHxNP选择上升沿或者下降沿。

第三步：捕获源选择（TIMERx_CHCTL0寄存器中CHxMS）：
一旦通过配置CHxMS选择输入捕获源，必须确保通道配置在输入模式（CHxMS!=0x0），而且TIMERx_CHxCV寄存器不能再被写。

第四步：中断使能（TIMERx_DMAINTEN寄存器中CHxIE和CHxDEN）：
使能相应中断，可以获得中断和DMA请求。

第五步：捕获使能（TIMERx_CHCTL2寄存器中CHxEN）。

结果：当期望的输入信号发生时，TIMERx_CHxCV被设置成当前计数器的值，CHxIF为置1。如果CHxIF位已经为1，则CHxOF位置1。根据TIMERx_DMAINTEN寄存器中CHxIE和CHxDEN的配置，相应的中断和DMA请求会被提出。

直接产生：软件设置CHxG位，会直接产生中断和DMA请求。

通道输入捕获功能也可用来测量 TIMERx_CHx 引脚上信号的脉冲波宽度。例如，一个 PWM 波连接到 CI0。配置 TIMERx_CHCTL0 寄存器中 CH0MS 为 2'b01，选择通道 0 的捕获信号为 CI0 并设置上升沿捕获。配置 TIMERx_CHCTL0 寄存器中 CH1MS 为 2'b10，选择通道 1 捕获信号为 CI0 并设置下降沿捕获。计数器配置为复位模式，在通道 0 的上升沿复位。TIMERX_CH0CV 寄存器测量 PWM 的周期值，TIMERx_CH1CV 寄存器测量 PWM 占空比值。

通道输出比较功能

图 15-13. 通道输出比较原理（带有互补输出的通道，x=0，1，2）

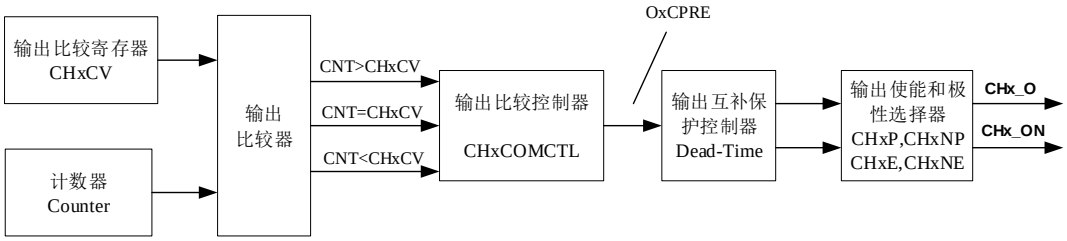


图 15-14. 通道输出比较原理

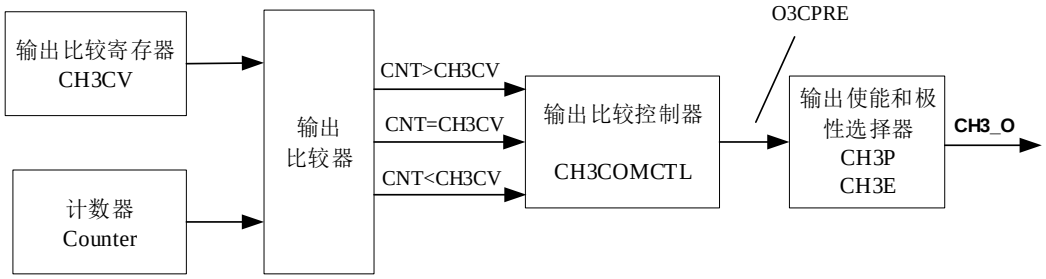


*图 15-13. 通道输出比较原理（带有互补输出的通道，x=0，1，2）*和*图 15-14. 通道输出比较原理*分别给出了输出比较原理电路。通道输出信号 CHx_O/CHx_ON 与 OxCPRE 信号的关系描述如下：OxCPRE 信号高电平有效，CHx_O/CHx_ON 的输出情况与 OxCPRE 信号，CHxP/CHxNP 位和 CHxE/CHxNE 位有关（具体情况请见 TIMERx_CHCTL2 寄存器中的描述）。例如：

1）当设置 CHxP=0（CHx_O 高电平有效，与 OxCPRE 输出极性相同）、CHxE=1（CHx_O 输出使能）时：

若 OxCPRE 输出有效（高）电平，则 CHx_O 输出有效（高）电平；
若 OxCPRE 输出无效（低）电平，则 CHx_O 输出无效（低）电平。

2）当设置 CHxNP=1（CHx_ON 低电平有效，与 OxCPRE 输出极性相反）、CHxNE=1（CHx_ON 输出使能）时：

若 OxCPRE 输出有效（高）电平，则 CHx_ON 输出有效（低）电平；
若 OxCPRE 输出无效（低）电平，则 CHx_ON 输出无效（高）电平。

当 CH0_O 和 CH0_ON 同时输出时，CH0_O 和 CH0_ON 的具体输出情况还与 TIMERx_CCHP 寄存器中的相关位（ROS、IOS、POE 和 DTCFG 等位）有关。。

在通道输出比较功能，TIMERx 可以产生时控脉冲，其位置，极性，持续时间和频率都是可编程的。当一个输出通道的 CHxVAL 寄存器与计数器的值匹配时，根据 CHxCOMCTL 的配置，这个通道的输出可以被置高电平，被置低电平或者反转。当计数器的值与 CHxVAL 寄存器的值匹配时，CHxIF 位被置 1，如果 CHxIE = 1 则会产生中断，如果 CHxDEN=1 则会产生 DMA 请求。

配置步骤如下：

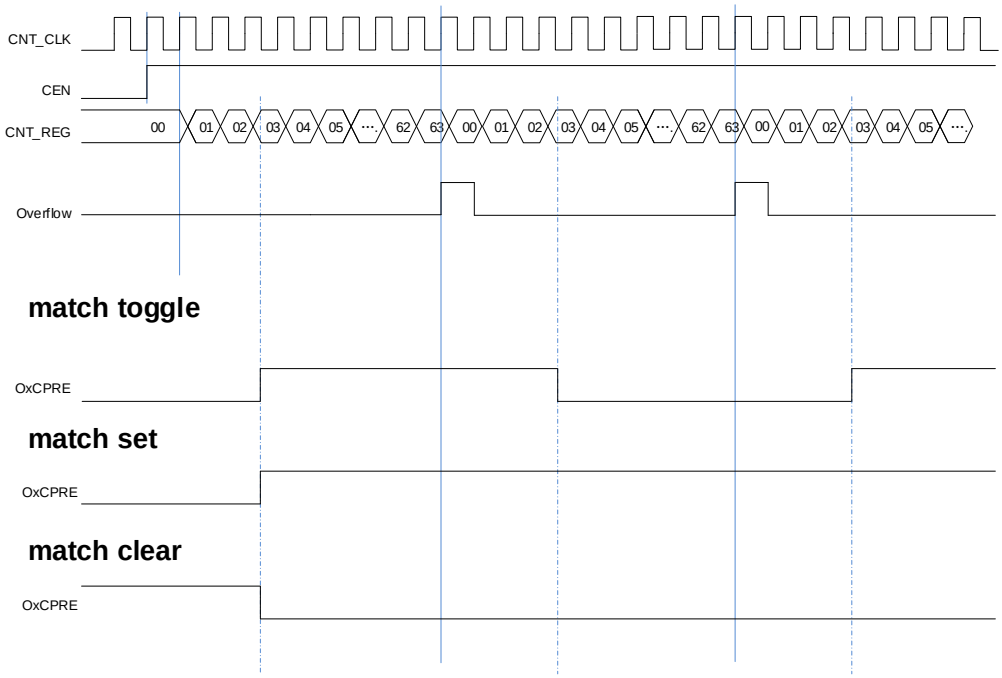
第一步：时钟配置：
配置定时器时钟源，预分频器等。
第二步：比较模式配置：
设置CHxCOMSEN位来配置输出比较影子寄存器；
设置CHxCOMCTL位来配置输出模式（置高电平/置低电平/反转）；
设置CHxP/CHxNP位来选择有效电平的极性；
设置CHxEN使能输出。
第三步：通过CHxIE/CHxDEN位配置中断/DMA请求使能。
第四步：通过TIMERx_CAR寄存器和TIMERx_CHxCV寄存器配置输出比较时基：

CHxVAL可以在运行时根据你所期望的波形而改变。

第五步：设置CEN位使能定时器。

*图 15-15. 三种输出比较模式*显示了三种比较输出模式：反转/置高电平/置低电平，CAR=0x63, CHxVAL=0x3。

图 15-15. 三种输出比较模式



输出 PWM 功能

在 PWM 输出模式下（PWM 模式 0 是配置 CHxCOMCTL 为 3'b110，PWM 模式 1 是配置 CHxCOMCTL 为 3'b111），通道根据 TIMERx_CAR 寄存器和 TIMERx_CHxCV 寄存器的值，输出 PWM 波形。

根据计数模式，我们可以分为两种 PWM 波：EAPWM(边沿对齐 PWM)和 CAPWM(中央对齐 PWM)。

EAPWM 的周期由 TIMERx_CAR 寄存器值决定，占空比由 TIMERx_CHxCV 寄存器值决定。*图 15-16. EAPWM 时序图*显示了 EAPWM 的输出波形和中断。

CAPWM 的周期由（2*TIMERx_CAR 寄存器值）决定，占空比由（2*TIMERx_CHxCV 寄存器值）决定。*图 15-17. CAPWM 时序图*显示了 CAPWM 的输出波形和中断。

在 PWM0 模式下(CHxCOMCTL==3'b110)，如果 TIMERx_CHxCV 寄存器的值大于 TIMERx_CAR 寄存器的值，通道输出一直为有效电平。

在 PWM0 模式下(CHxCOMCTL==3'b110)，如果 TIMERx_CHxCV 寄存器的值等于 0，通道输出一直为无效电平。

图 15-16. EAPWM 时序图

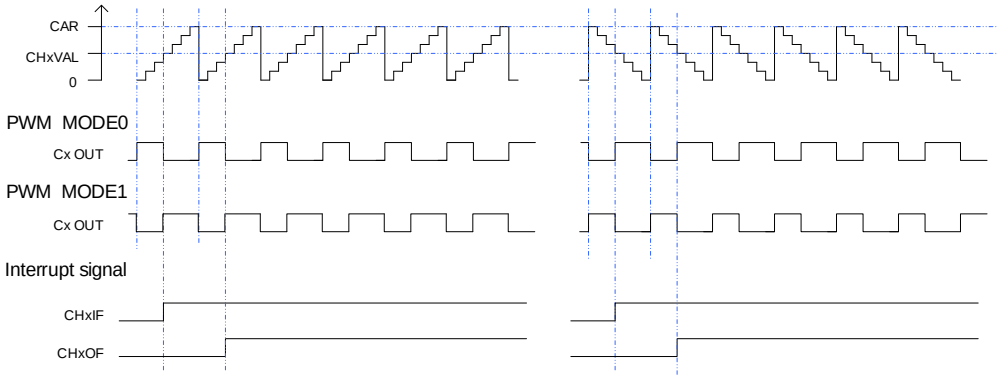
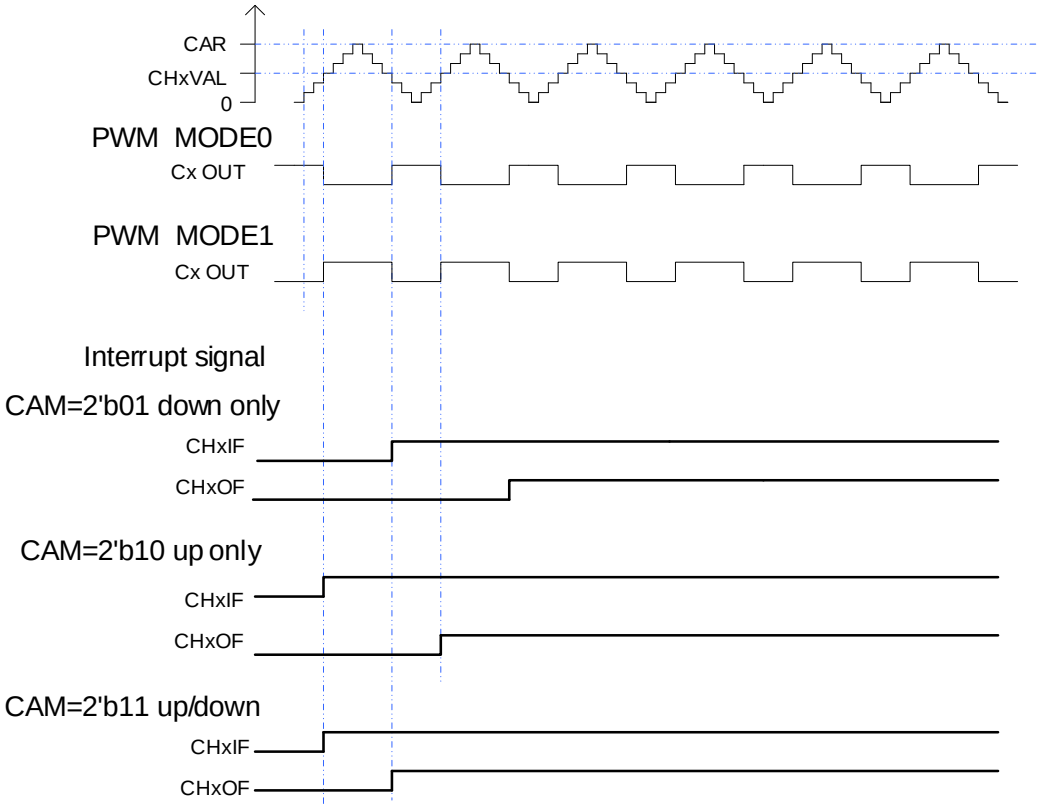


图 15-17. CAPWM 时序图



通道输出参考信号

根据*图 15-13. 通道输出比较原理（带有互补输出的通道，x=0，1，2）*,当 TIMERx 用于输出匹配比较模式下，设置 CHxCOMCTL 位可以定义 OxCPRE 信号(通道 x 准备信号)类型。OxCPRE 信号有若干类型的输出功能，包括，设置 CHxCOMCTL=0x00 可以保持原始电平；设置 CHxCOMCTL=0x01 可以将 OxCPRE 信号设置为高电平；设置 CHxCOMCTL=0x02 可以将 OxCPRE 信号设置为低电平；设置 CHxCOMCTL=0x03，在计数器值和 TIMERx_CHxCV 寄存器的值匹配时，可以翻转输出信号。

PWM 模式 0 和 PWM 模式 1 是 OxCPRE 的另一种输出类型，设置 CHxCOMCTL 位域位 0x06 或 0x07 可以配置 PWM 模式 0/PWM 模式 1。在这些模式中，根据计数器值和 TIMERx_CHxCV 寄存器值的关系以及计数方向，OxCPRE 信号改变其电平。具体细节描述，请参考相应的位。

设置 CHxCOMCTL =0x04 或 0x05 可以实现 OxCPRE 信号的强制输出功能。输出比较信号能够直接由软件强置为有效或无效状态，而不依赖于 TIMERx_CHxCV 的值和计数器值之间间的比较结果。

设置 CHxCOMCEN=1，当由外部 ETI 引脚信号产生的 ETIFE 信号为高电平时，OxCPRE 被强制为低电平。在下一次更新事件到来时，OxCPRE 信号才会回到有效电平状态。

通道输出互补 PWM

CHx_O 和 CHx_ON 是一对互补输出通道，这两个信号不能同时有效。TIMERx 有四路通道，只有前三路有互补输出通道。互补信号 CHx_O 和 CHx_ON 是由一组参数来决定：TIMERx_CHCTL2 寄存器中的 CHxEN 和 CHxNEN 位，TIMERx_CCHP 寄存器中和 TIMERx_CTL1 寄存器中的 POEN, ROS, IOS, ISOx 和 ISOxN 位。输出极性由 TIMERx_CHCTL2 寄存器中的 CHxP 和 CHxNP 位来决定。

表 15-2. 由参数控制的互补输出表

<table>
<tr><th colspan="5">互补参数</th><th colspan="2">输出状态</th></tr>
<tr><th>POEN</th><th>ROS</th><th>IOS</th><th>CHxEN</th><th>CHxNEN</th><th>CHx_O</th><th>CHx_ON</th></tr>
<tr><td rowspan="4">0</td><td rowspan="4">0/1</td><td rowspan="3">0</td><td rowspan="2">0</td><td>0</td><td colspan="2">CHx_O / CHx_ON = LOW
CHx_O / CHx_ON 输出禁能(1)</td></tr>
<tr><td>1</td><td colspan="2" rowspan="3">CHx_O/CHx_ON输出关闭状态(2)：
通道先输出无效电平：CHx_O = CHxP，CHx_ON = CHxNP）；如果死区产生时钟未失效，在死区时间之后：
CHx_O = ISOx，CHx_ON = ISOxN (3)</td></tr>
<tr><td rowspan="2">1</td><td>0</td></tr>
<tr><td>1</td></tr>
<tr><td>1</td><td>x</td><td>x</td><td colspan="2">CHx_O/CHx_ON输出关闭状态：
通道先输出无效电平：CHx_O = CHxP，CHx_ON = CHxNP）；如果死区产生时钟未失效，在死区时间之后：
CHx_O = ISOx，CHx_ON = ISOxN</td></tr>
<tr><td rowspan="6">1</td><td rowspan="4">0</td><td rowspan="5">0/1</td><td rowspan="2">0</td><td>0</td><td colspan="2">CHx_O/CHx_ON = LOW
CHx_O/CHx_ON输出禁能</td></tr>
<tr><td>1</td><td>CHx_O = LOW
CHx_O输出禁能</td><td>CHx_ON=OxCPRE⊕(4)CHxNP
CHx_ON输出使能</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td>CHx_O=OxCPRE⊕CHxP
CHx_O输出使能</td><td>CHx_ON = LOW
CHx_ON输出禁能</td></tr>
<tr><td>1</td><td>CHx_O=OxCPRE⊕CHxP
CHx_O输出使能</td><td>CHx_ON=(!OxCPRE)(5)⊕CHxNP
CHx_ON输出使能</td></tr>
<tr><td rowspan="2">1</td><td rowspan="2">0</td><td>0</td><td>CHx_O = CHxP
CHx_O输出关闭状态</td><td>CHx_ON = CHxNP
CHx_ON输出关闭状态</td></tr>
<tr><td>1</td><td>CHx_O = CHxP</td><td>CHx_O=OxCPRE⊕CHxNP</td></tr>
</table>

互补参数					输出状态	
POEN	ROS	IOS	CHxEN	CHxNEN	CHx_O	CHx_ON
					CHx_O输出关闭状态	CHx_ON输出使能
			1	0	CHx_O=OxCPRE⊕CHxP CHx_O输出使能	CHx_ON = CHxNP CHx_ON输出关闭状态
				1	CHx_O=OxCPRE⊕CHxP CHx_O输出使能	CHx_ON=（!OxCPRE）⊕CHxNP CHx_ON输出使能

注意：

（1） 输出禁能：CHx_O / CHx_ON 输出与对应引脚断开，对应引脚电平受 GPIO 上下拉配置控制，无上下拉时为悬空高阻态；

（2） 输出关闭状态：CHx_O / CHx_ON 输出无效电平（CHx_O = 0⊕CHxP = CHxP）；

（3） 详情见中止模式章节。

（4） ⊕：异或操作；

（5） (!OxCPRE)：OxCPRE 信号的互补信号。

互补 PWM 插入死区时间

设置 CHxEN 和 CHxNEN 为 1'b1 同时设置 POEN，死区插入就会被使能。DTCFG 位域定义了死区时间，死区时间对除了通道 3 以外通道有效。死区时间的细节，请参考 TIMERx_CCHP 寄存器。

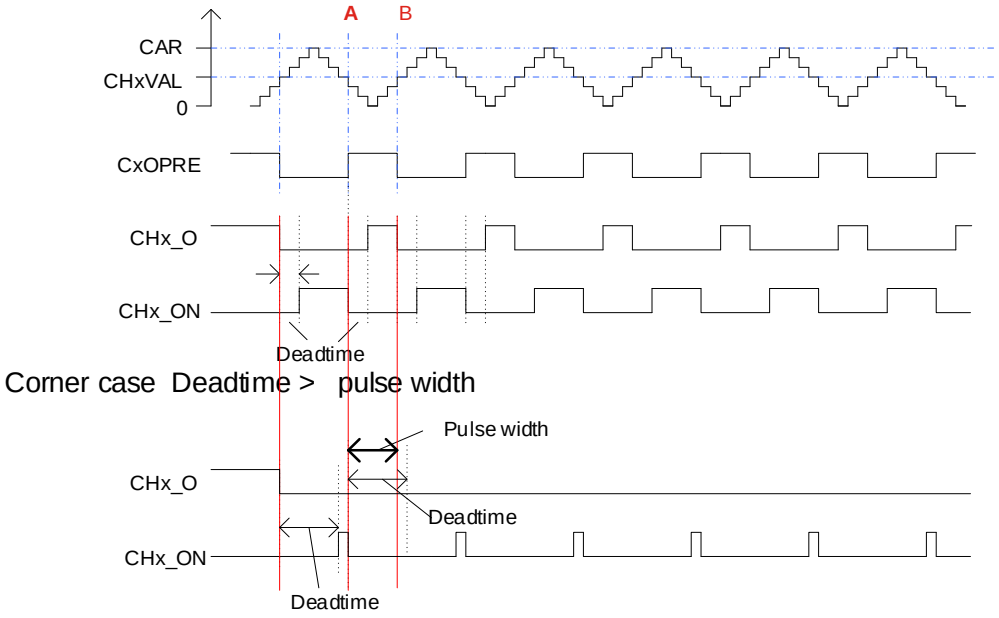
死区时间的插入，确保了通道互补的两路信号不会同时有效。

在 PWM0 模式，当通道 x 匹配发生时（TIMERx 计数器= CHxVAL），OxCPRE 反转。在*图 15-18. 带死区时间的通道互补输出*中的 A 点，CHx_O 信号在死区时间内为低电平，直到死区时间过后才变为高电平，而 CHx_ON 信号立刻变为低电平。同样，在 B 点，计数器再次匹配（TIMERx 计数器= CHxVAL），OxCPRE 信号被清 0，CHx_O 信号被立即清零，CHx_ON 信号在死区时间内仍然是低电平，在死区时间过后才变为高电平。

有时会有一些死角事件发生，例如：

- 如果死区延时大于或者等于 CHx_O 信号的占空比，CHx_O 信号一直为无效值（如*图15-18. 带死区时间的通道互补输出*）。
- 如果死区延时大于或者等于 CHx_ON 信号的占空比，CHx_ON 信号一直为无效值。

图 15-18. 带死区时间的通道互补输出



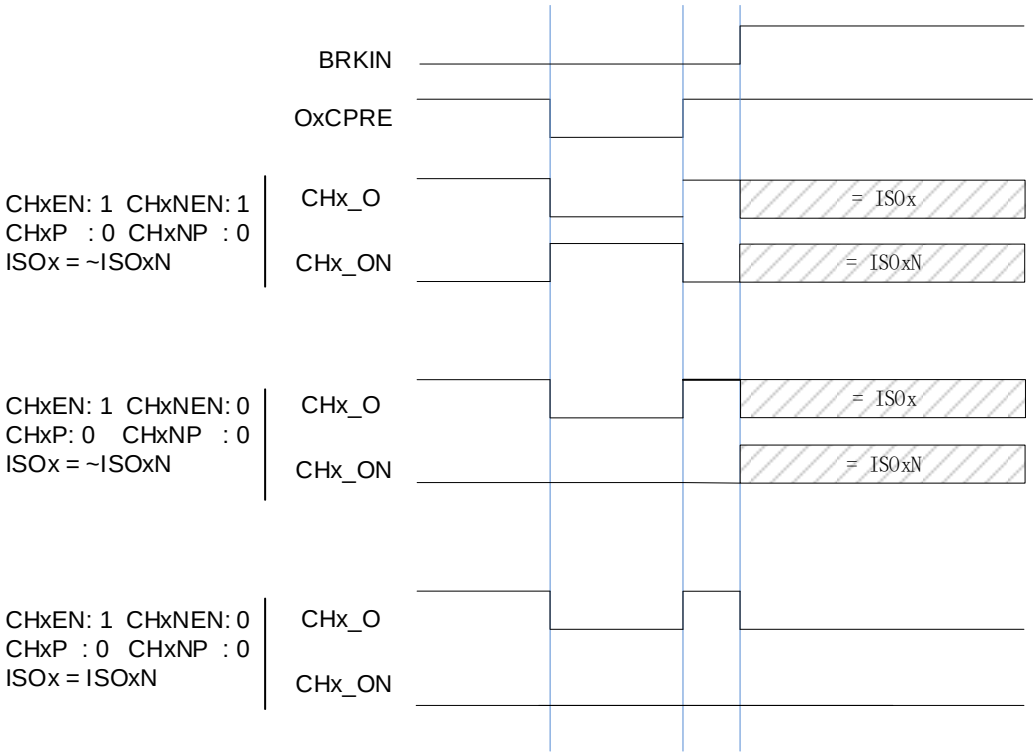
中止模式

使用中止模式时，输出 CHx_O 和 CHx_ON 信号电平被以下位控制，TIMERx_CCHP 寄存器的 POEN, IOS 和 ROS 位，TIMERx_CTL1 寄存器的 ISOx 和 ISOxN 位。当中止事件发生时，CHx_O 和 CHx_ON 信号输出不能同时设置为有效电平。中止源可以选择中止输入引脚，也可以选择 HXTAL 时钟失效事件。时钟失败事件由 RCU 中的时钟监视器(CKM)产生。将 TIMERx_CCHP 寄存器的 BRKEN 位置 1 可以使能中止功能。TIMERx_CCHP 寄存器的 BRKP 位决定了中止输入极性。

发生中止时，POEN 位被异步清除，一旦 POEN 位为 0，CHx_O 和 CHx_ON 被 TIMERx_CTL1 寄存器中的 ISOx 位和 ISOxN 驱动。如果 IOS=0，定时器释放输出使能，否则输出使能仍然为高。起初互补输出被置于复位状态，然后死区时间产生器重新被激活，以便在一个死区时间后驱动输出，输出电平由 ISOx 和 ISOxN 位配置。

发生中止时，TIMERx_INTF 寄存器的 BRKIF 位被置 1。如果 BRKIE=1，中断产生。

图 15-19. 通道响应中止输入（高电平有效）时，输出信号的行为



正交译码器

正交译码器功能使用由TIMERx_CH0和TIMERx_CH1引脚生成的CI0FE0和CI1FE1正交信号各自相互作用产生计数值。在每个输入源改变期间，DIR位会发生改变。输入源可以是只有CI0FE0，可以只有CI1FE1，或着可以同时有CI0FE0和CI1FE1，通过设置SMC=0x01, 0x02或0x03来选择使用哪种模式。计数器计数方向改变的机制如*表15-3. 不同正交译码器模式下的计数方向*所示。正交译码器可以当作一个带有方向选择的外部时钟，这意味着计数器会在0和自动加载值之间连续的计数。因此，用户必须在计数器开始计数前配置TIMERx_CAR寄存器。

表 15-3. 不同正交译码器模式下的计数方向

计数模式	电平	CI0FE0		CI1FE1	
		上升	下降	上升	下降
正交译码器模式0 SMC[2:0]=3'b001	CI1FE1=1	向下	向上	-	-
	CI1FE1=0	向上	向下	-	-
正交译码器模式1 SMC [2:0]=3'b010	CI0FE0=1	-	-	向上	向下
	CI0FE0=0	-	-	向下	向上
正交译码器模式2 SMC [2:0]=3'b011	CI1FE1=1	向下	向上	X	X
	CI1FE1=0	向上	向下	X	X
	CI0FE0=1	X	X	向上	向下
	CI0FE0=0	X	X	向下	向上

注意： "-" 意思是"无计数"; "X" 意思是不可能。"0" 意思是低电平, "1" 意思是高电平。

图 15-20.在正交译码器模式 2 且 CI0FE0 极性不反相时计数器行为

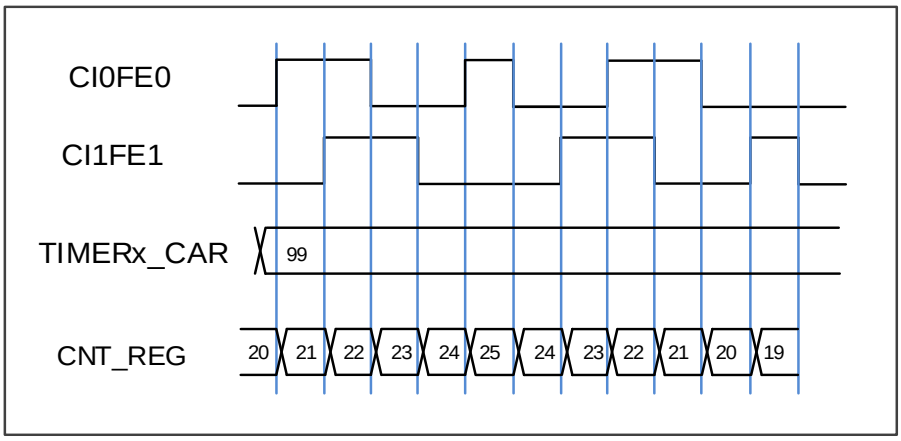
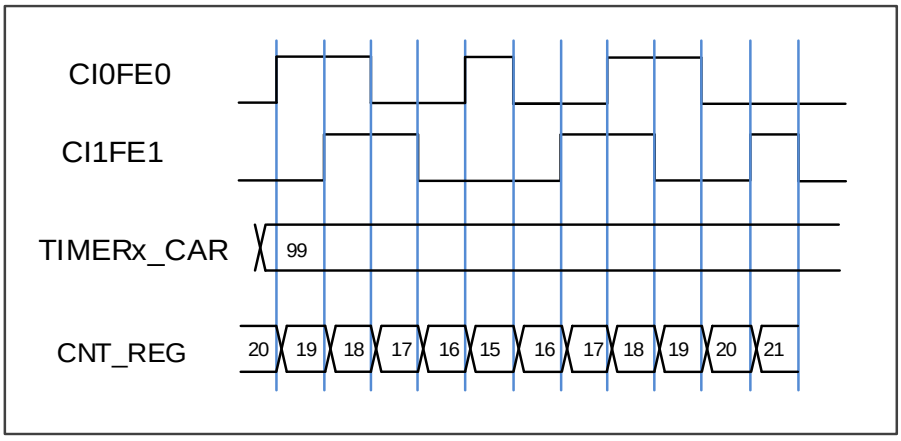


图 15-21.在正交译码器模式 2 且 CI0FE0 极性反相时计数器行为



霍尔传感器接口功能

高级定时器支持霍尔传感器接口功能，该功能可以用来控制 BLDC 电机。

*图 15-22. 霍尔传感器用在 BLDC 电机控制中*是定时器和电机的连接示意图。TIMER_in 定时器（可以是高级定时器或者通用 L0 定时器）接收霍尔传感器的三路信号。

三个霍尔传感器信号与 TIMER_in 定时器的三路输入捕获引脚一一对应连接，每个霍尔传感器输入一路波形到输入引脚，分析三路霍尔信号可以计算出转子的位置和速度。

通过定时器内部连接，例如 TRGO-ITIx，TIMER_in 定时器和 TIMER_out 定时器可以连接在一起。TIMER_out 定时器根据 ITIx 触发信号输出 PWM 波，驱动 BLDC 电机，控制 BLDC 电机的速度。这样，TIMER_in 定时器和 TIMER_out 定时器的连接形成了一个反馈电路，可以根据需求改变配置。

TIMER_in 定时器需要具备输入异或功能，所以可以选择高级定时器和通用 L0 定时器。

TIMER_out 定时器需要具备互补输出和死区插入功能，所以可以选择高级定时器。另外，根据定时器的内部互连关系，可以选择成对的互连定时器，例如:

TIMER_in (TIMER1) -> TIMER_out (TIMER0 ITI1)

等等。

选择好合适的互连定时器，定时器和 BLDC 的线路也已经连接好，我们就可以配置定时器了。有以下关键配置：

- 设置TI0S，使能异或功能。三路输入信号的任何一路发生变化，CI0都会反转，CH0VAL此时会捕获计数器的当前值。
- 设置CCUC和CCSE，使能ITIx直接连接到换相功能。
- 根据需求配置PWM参数。

图 15-22. 霍尔传感器用在 BLDC 电机控制中

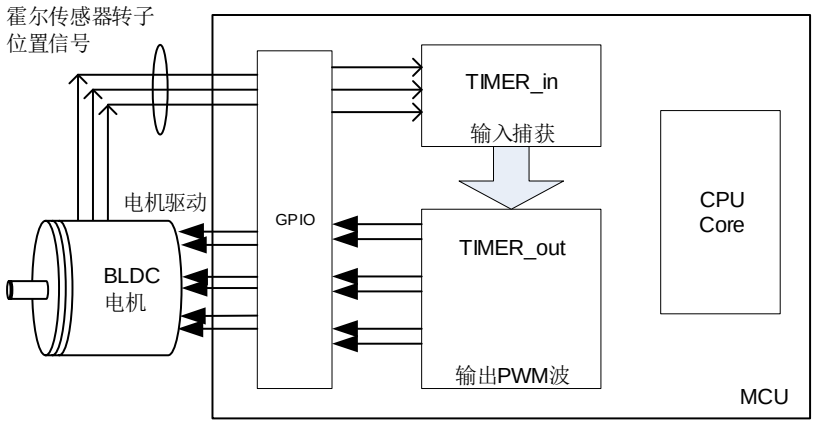
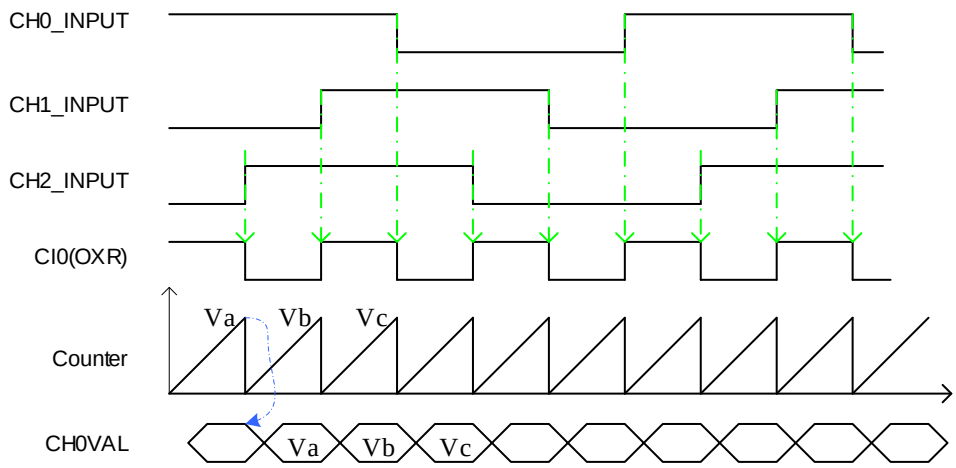
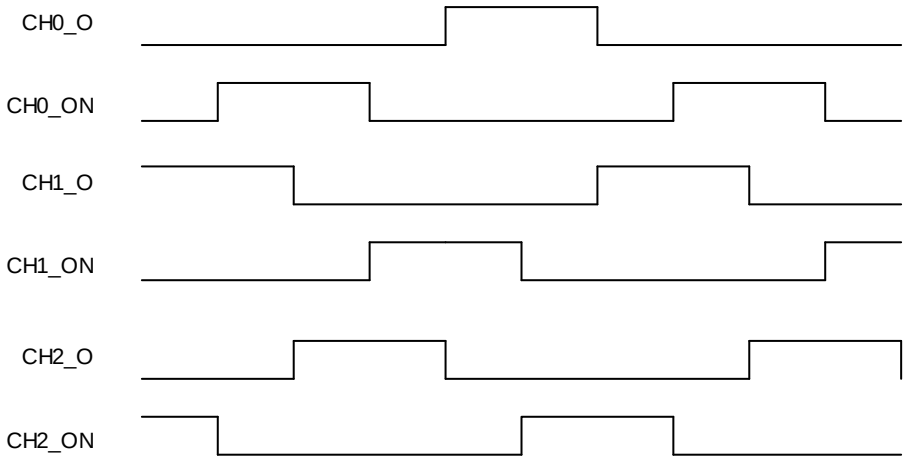


图 15-23. 两个定时器之间的霍尔传感器时序图

高级/通用 L0定时器 TIMER_in 工作在输入捕获模式



高级定时器 TIMER_out 工作在输出比较模式(带有死区的PWM)

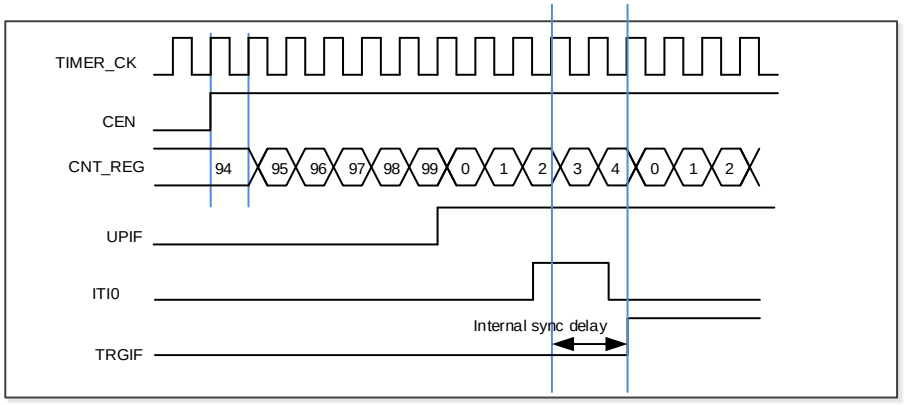
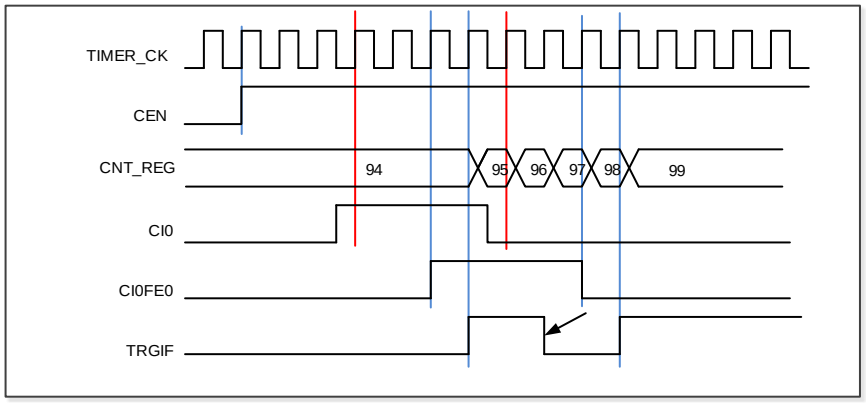


主-从管理

TIMERx 能在多种模式下同步外部触发，包括复位模式，暂停模式和事件模式，可以通过设置 TIMERx_SMCFG 寄存器中的 SMC[2:0]配置这些模式。这些模式的输入触发源可以通过设置 TIMERx_SMCFG 寄存器中的 TRGS[2:0]来选择。

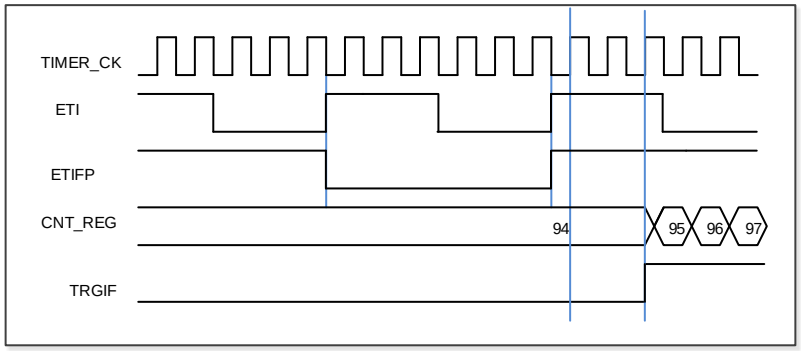
表 15-4. 从模式例子列表

	模式选择	触发源选择	极性选择	滤波和预分频
列举	SMC[2:0] 3'b100 （复位模式） 3'b101 （暂停模式） 3'b110 （事件模式）	TRGS[2:0] 000: ITI0 001: ITI1 010: ITI2 011: ITI3 100: CI0F_ED 101: CI0FE0 110: CI1FE1	如果触发源是CI0FE0或者CI1FE1，配置CHxP和 CHxNP来选择极性和反相 如果触发源是ETIF，配置ETP选择极性和反相	触发源ITIx，滤波和预分频不可用 触发源CIx，配置CHxCAPFLT设置滤波，分频不可用 触发源是ETIF，滤波和预分频不可用

<table>
<tr><th></th><th>模式选择</th><th>触发源选择</th><th>极性选择</th><th>滤波和预分频</th></tr>
<tr><td></td><td></td><td>111: ETIFP</td><td></td><td></td></tr>
<tr><td rowspan="2">例1</td><td>复位模式
当触发输入上升沿，计数器清零重启</td><td>TRGIS[2:0]=3’b000
选择ITI0为触发源</td><td>触发源是ITI0，极性选择不可用</td><td>触发源是ITI0，滤波和预分频不可用</td></tr>
<tr><td colspan="4">图 15-24. 复位模式下的控制电路
 
</td></tr>
<tr><td rowspan="2">例2</td><td>暂停模式
当触发输入为低的时候，计数器暂停计数</td><td>TRGIS[2:0]=3’b101
选择CI0FE0为触发源</td><td>TI0S=0（非异或）[CH0NP==0, CH0P==0]不反相.在上升沿捕获</td><td>在这个例子中滤波被旁路</td></tr>
<tr><td colspan="4">图 15-25. 暂停模式下的控制电路
 
</td></tr>
<tr><td>例3</td><td>事件模式
触发输入的上升沿计数器开始计数</td><td>TRGIS[2:0]=3’b111
选择ETIF为触发源</td><td>ETP = 0没有极性改变</td><td>ETPSC = 1，2分频.
ETFC = 0，无滤波</td></tr>
</table>

	模式选择	触发源选择	极性选择	滤波和预分频

图 15-26. 事件模式下的控制电路



单脉冲模式

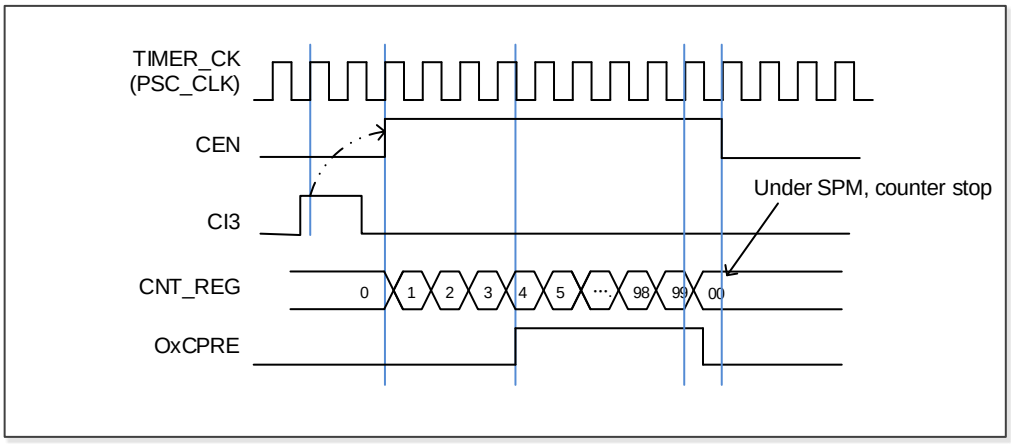
单脉冲模式与重复模式是相反的，设置 TIMERx_CTL0 寄存器的 SPM 位置 1，则使能单脉冲模式。当 SPM 置 1，计数器在下次更新事件到来后清零并停止计数。为了得到脉冲波，可以通过设置 CHxCOMCTL 配置 TIMERx 为 PWM 模式或者比较模式。

一旦设置定时器运行在单脉冲模式下，没有必要设置 TIMERx_CTL0 寄存器的定时器使能位 CEN=1 来使能计数器。触发信号沿或者软件写 CEN=1 都可以产生一个脉冲，此后 CEN 位一直保持为 1 直到更新事件发生或者 CEN 位被软件写 0。如果 CEN 位被软件清 0，计数器停止工作， 计数值被保持。

在单脉冲模式下，有效的外部触发边沿会将 CEN 位置 1，使能计数器。然而，执行计数值和 TIMERx_CHxCV 寄存器值的比较结果依然存在一些时钟延迟。为了最大限度减少延迟，用户可以将 TIMERx_CHCTL0/1 寄存器的 CHxCOMFEN 位置 1。单脉冲模式下，触发上升沿产生之后， OxCPRE 信号将被立即强制转换为与发生比较匹配时相同的电平，但是不用考虑比较结果。只有输出通道配置为 PWM0 或 PWM1 输出运行模式下时 CHxCOMFEN 位才可用，触发源来源于触发信号。

图 15-27. 单脉冲模式，TIMERx_CHxCV = 4 TIMERx_CAR=99 展示了一个例子。

图 15-27. 单脉冲模式，TIMERx_CHxCV = 4 TIMERx_CAR=99



定时器互连

定时器之间可配置为内部级联，一个定时器配置为主模式输出TRGO信号，另一个定时器配置为从模式，TRGO信号包括复位事件、使能事件、更新事件、捕获比较脉冲事件、比较事件。从定时器接收到ITIx信号，并执行对应的操作，包括内部时钟模式、正交编码模式、复位模式、暂停模式、事件模式、外部时钟模式。

*图 15-28. 定时器 0 主/从模式的例子*显示了一些主从模式触发选择的例子。

图 15-28. 定时器 0 主/从模式的例子

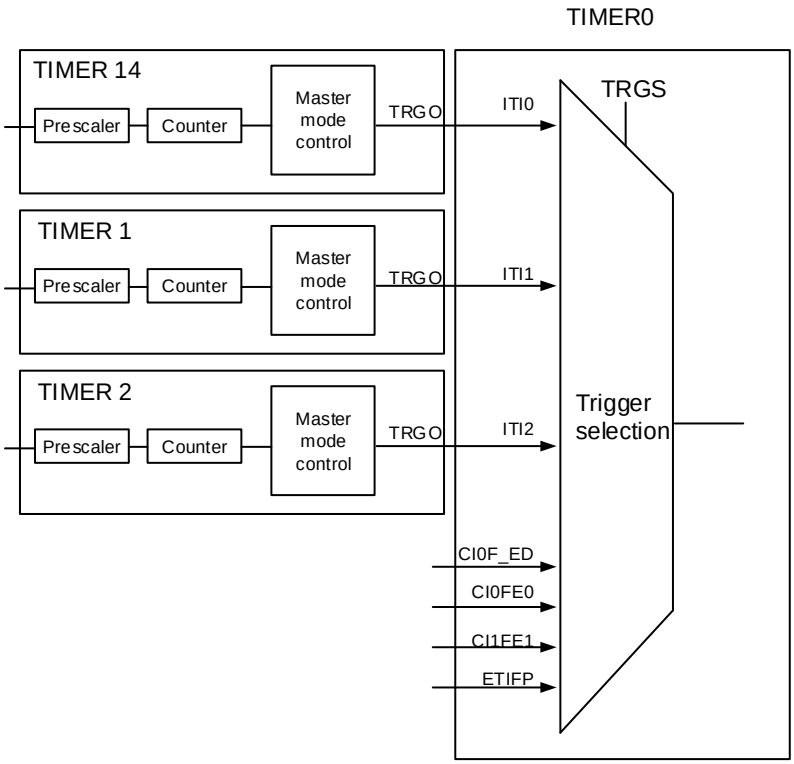


*表15-5. 定时器0内部互联*显示了内部触发。

表 15-5. 定时器 0 内部互联

Slave TIMER	ITI0(TRGS = 000)	ITI1(TRGS = 001)	ITI2(TRGS = 010)	ITI3(TRGS = 011)
TIMER0	TIMER14	TIMER1	TIMER2	reserved

注释: '-' 表示没有内部级联。

其他定时器互连的例子：

定时器 2 作为定时器 0 的预分频器

参考*图 15-28. 定时器 0 主/从模式的例子*连接配置定时器 2 为定时器 0 的预分频器，步骤如下：

1. 配置定时器2为主模式，选择其更新事件(UPE)为触发输出(配置TIMER2_CTL1寄存器的MMC=3'b010)。定时器2在每次计数器溢出产生更新事件时，输出一个周期信号；
2. 配置定时器2周期(TIMER2_CAR寄存器)；

3. 选择定时器0输入触发源为定时器2 (配置TIMERx_SMCFG寄存器的TRGS=3’b010)；
4. 配置定时器0在外部时钟模式0(配置TIMERx_SMCFG寄存器的SMC=3’b111)；
5. 写1到CEN位启动定时器0 (TIMER0_CTL0寄存器)；
6. 写1到CEN位启动定时器2 (TIMER2_CTL0寄存器)。

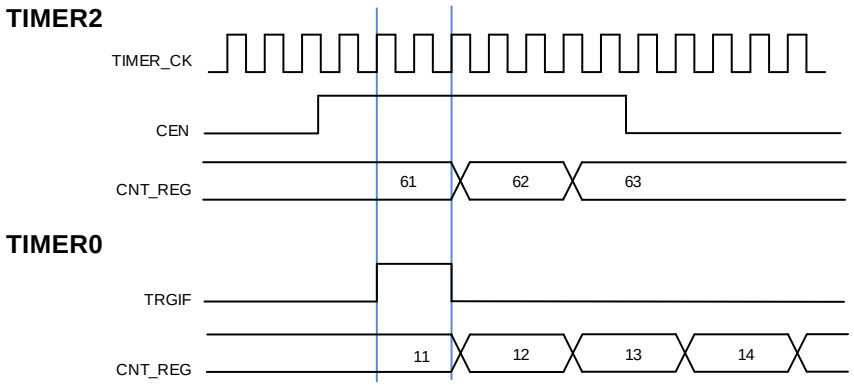
用定时器2的使能/更新信号来启动定时器0

用定时器 2 的使能信号来启动定时器 0，见*图 15-29. 用定时器 2 的使能信号触发定时器 0*。在定时器 2 使能信号输出后，定时器 0 按照分频后的内部时钟从当前值开始计数。

当定时器 0 接收到触发信号，它的 CEN 位置 1，计数器计数直到禁能定时器 0。两个定时器的计数器频率都是 TIMER_CK 经过预分频器 3 分频后频率（fCNT_CLK = fTIMER_CK/3）。步骤如下：

1. 配置定时器 2 为主模式，发送它的使能信号作为触发输出（配置 TIMER2_CTL1 寄存器的 MMC=3’b001）；
2. 配置定时器 0 选择输入触发来自定时器 2（配置 TIMER0_SMCFG 寄存器的 TRGS=3’b010）；
3. 配置定时器 0 在事件模式（配置 TIMER0_SMCFG 寄存器的 SMC=3’b 110）；
4. 写 1 到 CEN 来开启定时器 2（TIMER2_CTL0 寄存器）。

图 15-29. 用定时器 2 的使能信号触发定时器 0



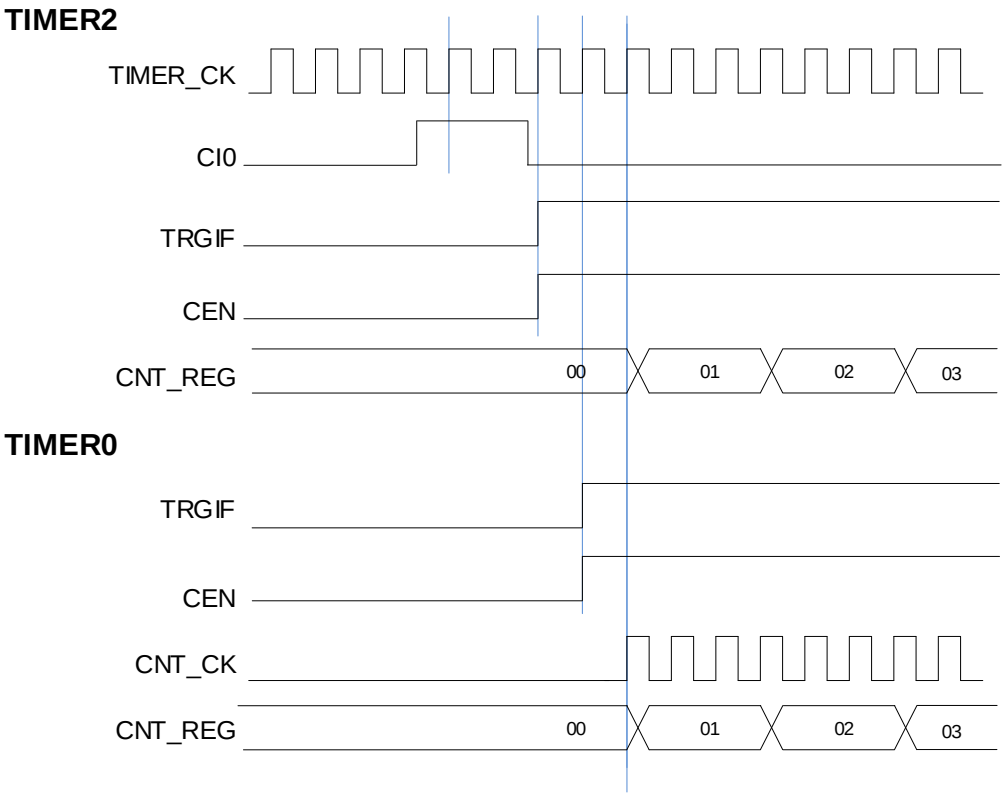
使用一个外部触发来同步两个定时器

配置定时器2的使能信号触发定时器0的开启，配置定时器2的CI0输入信号上升沿来触发定时器2。为了确保两个定时器同步开启，定时器2必须配置在主/从模式。步骤如下:

1. 配置定时器2工作在从模式，并选择CI0_ED作为触发输入(配置TIMER2_SMCFG寄存器的TRGS=3’b100)；
2. 配置定时器2工作在事件模式(配置TIMER2_SMCFG寄存器的SMC=3’b110)；
3. 写MSM=1(TIMER2_SMCFG寄存器)来配置定时器2工作在主/从模式；
4. 配置定时器0的触发输入为定时器2 (配置TIMERx_SMCFG 寄存器的TRGS=3’b010)；
5. 配置定时器0工作在事件模式(配置TIMER0_SMCFG寄存器的SMC=3’b110)。

当定时器 2 的 CI0 信号产生上升沿时，两个定时器的计数器在内部时钟下开始同步计数，二者的 TRGIF 标志位都被置 1。

图 15-30. 用定时器 2 的 CI0 信号来触发定时器 0 和定时器 2



定时器 DMA 模式

定时器 DMA 模式是指通过 DMA 模块配置定时器的寄存器。有两个跟定时器 DMA 模式相关的寄存器：TIMERx_DMACFG 和 TIMERx_DMATB。当然，必须要使能 DMA 请求，一些内部中断事件可以产生 DMA 请求。当中断事件发生，TIMERx 会给 DMA 发送请求。DMA 配置成 M2P 模式，PADDR 是 TIMERx_DMATB 寄存器地址，DMA 就会访问 TIMERx_DMATB 寄存器。实际上，TIMERx_DMATB 寄存器只是一个缓冲，定时器会将 TIMERx_DMATB 映射到一个内部寄存器，这个内部寄存器由 TIMERx_DMACFG 寄存器中的 DMATA 来指定。如果 TIMERx_DMACFG 寄存器的 DMATC 位域值为 0，表示 1 次传输，定时器的发送 1 个 DMA 请求就可以完成。如果 TIMERx_DMACFG 寄存器的 DMATC 位域值不为 1，例如其值为 3，表示 4 次传输，定时器就需要再多发 3 次 DMA 请求。在这 3 次请求下，DMA 对 TIMERx_DMATB 寄存器的访问会映射到访问定时器的 DMATA+0x4, DMATA+0x8, DMATA+0xc 寄存器。总之，发生一次 DMA 内部中断请求，定时器会连续发送（DMATC+1）次请求。

如果再来 1 次 DMA 请求事件，TIMERx 将会重复上面的过程。

定时器调试模式

当 Cortex®-M3 内核停止，DBG_CTL0 寄存器中的 TIMERx_HOLD 配置位被置 1，定时器计数器停止。

15.1.5. TIMERx 寄存器(x=0)

TIMER0 基地址：0x4001 2C00

控制寄存器 0 (TIMERx_CTL0)

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						CKDIV[1:0]		ARSE	CAM[1:0]		DIR	SPM	UPS	UPDIS	CEN
						rw		rw	rw		rw	rw	rw	rw	rw

位/位域	名称	描述
31:10	保留	必须保持复位值。
9:8	CKDIV[1:0]	时钟分频 通过软件配置CKDIV，规定定时器时钟(CK_TIMER) 与死区时间和数字滤波器采样时钟(DTS)之间的分频系数。 00：$f_{DTS}=f_{CK_TIMER}$ 01：$f_{DTS}= f_{CK_TIMER}$ /2 10：$f_{DTS}= f_{CK_TIMER}$ /4 11：保留
7	ARSE	自动重载影子使能 0：禁能 TIMERx_CAR 寄存器的影子寄存器 1：使能 TIMERx_CAR 寄存器的影子寄存器
6:5	CAM[1:0]	计数器对齐模式选择 00：无中央对齐计数模式(边沿对齐模式)。 DIR位指定了计数方向 01：中央对齐向下计数置1模式。计数器在中央计数模式计数，通道被配置在输出模式（TIMERx_CHCTL0寄存器中CHxMS=00），只有在向下计数时，CHxF位置1 10：中央对齐向上计数置1模式。计数器在中央计数模式计数，通道被配置在输出模式（TIMERx_CHCTL0寄存器中CHxMS=00），只有在向上计数时，CHxF位置1 11：中央对齐上下计数置1模式。计数器在中央计数模式计数，通道被配置在输出模式（TIMERx_CHCTL0寄存器中CHxMS=00），在向上和向下计数时，CHxF位都会置1 当计数器使能以后，该位不能从 0x00 切换到非 0x00
4	DIR	方向 0：向上计数 1：向下计数

当计数器配置为中央对齐计数模式或正交译码器模式时，该位只读。

3	SPM	单脉冲模式 0：单脉冲模式禁能。更新事件发生后，计数器继续计数 1：单脉冲模式使能。在下一次更新事件发生时，计数器停止计数
2	UPS	更新请求源 软件配置该位，选择更新事件源. 0：以下事件均会产生更新中断或DMA请求： UPG位被置1 计数器溢出/下溢 复位模式产生的更新 1：下列事件会产生更新中断或DMA请求： 计数器溢出/下溢
1	UPDIS	禁止更新. 该位用来使能或禁能更新事件的产生 0：更新事件使能. 更新事件发生时，相应的影子寄存器被装入预装载值，以下事件均会产生更新事件： UPG位被置1 计数器溢出/下溢 复位模式产生的更新 1：更新事件禁能. 注意：当该位被置 1 时，UPG 位被置 1 或者复位模式不会产生更新事件，但是计数器和预分频器被重新初始化
0	CEN	计数器使能 0：计数器禁能 1：计数器使能 在软件将 CEN 位置 1 后，外部时钟、暂停模式和正交译码器模式才能工作。

控制寄存器 1 (TIMERx_CTL1)

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留.	ISO3	ISO2N	ISO2	ISO1N	ISO1	ISO0N	ISO0	TI0S	MMC[2:0]			DMAS	CCUC	保留.	CCSE
	rw	rw	rw	rw	rw	rw	rw	rw	rw			rw	rw		rw

位/位域	名称	描述
31:15	保留	必须保持复位值。

14	ISO3	通道 3 的空闲状态输出 参考 ISO0 位
13	ISO2N	通道 2 的互补通道空闲状态输出 参考 ISO0N 位
12	ISO2	通道 2 的空闲状态输出 参考 ISO0 位
11	ISO1N	通道 1 的互补通道空闲状态输出 参考 ISO0N 位
10	ISO1	通道 1 的空闲状态输出 参考 ISO0 位
9	ISO0N	通道 0 的互补通道空闲状态输出 0：当 POEN 复位，CH0_ON 设置低电平. 1：当 POEN 复位，CH0_ON 设置高电平 此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0]位为 00 的时候可以被更改.
8	ISO0	通道 0 的空闲状态输出 0：当 POEN 复位，CH0_O 设置低电平 1：当 POEN 复位，CH0_O 设置高电平 如果 CH0_ON 生效，一个死区时间后 CH0_O 输出改变。此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0]位为 00 的时候可以被更改.
7	TI0S	通道 0 触发输入选择 0：选择 TIMERx_CH0 引脚作为通道 0 的触发输入 1：选择 TIMERx_CH0, CH1 和 CH2 引脚异或的结果作为通道 0 的触发输入
6:4	MMC[2:0]	主模式控制 l 这些位控制 TRGO 信号的选择，TRGO 信号由主定时器发给从定时器用于同步功能 000：当产生一个定时器复位事件后，输出一个TRGO信号，定时器复位源为： 主定时器产生一个复位事件 TIMERx_SWEVG寄存器中UPG位置1 001：当产生一个定时器使能事件后，输出一个TRGO信号，定时器使能源为： CEN位置1 在暂停模式下，触发输入置1 010：当产生一个定时器更新事件后，输出一个TRGO信号，更新事件源由UPDIS和UPS位决定 011：当通道0在发生一次捕获或一次比较成功时，主模式控制器产生一个TRGO脉冲 100：当产生一次比较事件时，输出一个TRGO信号，比较事件源来自O0CPRE 101：当产生一次比较事件时，输出一个TRGO信号，比较事件源来自O1CPRE 110：当产生一次比较事件时，输出一个TRGO信号，比较事件源来自O2CPRE 111：当产生一次比较事件时，输出一个 TRGO 信号，比较事件源来自 O3CPRE
3	DMAS	DMA请求源选择 0：当通道捕获/比较事件发生时，发送通道 x 的 DMA 请求 .

		1：当更新事件发生，发送通道 x 的 DMA 请求
2	CCUC	换相控制影子寄存器更新控制 当换相控制影子寄存器（CHxEN, CHxNEN 和 CHxCOMCTL 位）使能(CCSE=1)，这 些影子寄存器更新控制如下： 0：CMTG 位被置 1 时更新影子寄存器 1：当 CMTG 位被置 1 或检测到 TRIGI 上升沿时，影子寄存器更新 当通道没有互补输出时，此位无效。
1	保留	必须保持复位值。
0	CCSE	换相控制影子使能 0：影子寄存器 CHxEN, CHxNEN 和 CHxCOMCTL 位禁能. 1：影子寄存器 CHxEN, CHxNEN 和 CHxCOMCTL 位使能. 如果这些位已经被写入了，换相事件到来时这些位才被更新 当通道没有互补输出时，此位无效

从模式配置寄存器 （TIMERx_SMCFG）

地址偏移：0x08

复位值: 0x0000 0000

该寄存器通过字访问（32位）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	SMC1	ETPSC[1:0]		ETFC[3:0]				MSM	TRGS[2:0]			OCRC	SMC[2:0]		
rw	rw	rw		rw				rw	rw			rw	rw		

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	ETP	外部触发极性 该位指定 ETI 信号的极性。 0：ETI 高电平或上升沿有效。 1：ETI 低电平或下降沿有效。
14	SMC1	SMC 的一部分为了使能外部时钟模式 1 在外部时钟模式 1，计数器由 ETIF 信号上的任意有效边沿驱动。 0：外部时钟模式 1 禁能。 1：外部时钟模式 1 使能。 当从模式配置为复位模式，暂停模式和事件模式时，定时器仍然可以工作在外部时钟模式 1。但是 TRGS 必须不能为 3’b111。 如果外部时钟模式 0 和外部时钟模式 1 同时被被配置，外部时钟的输入是 ETIF 注意：外部时钟模式 0 使能在寄存器的 SMC[2:0]位域。

13:12 ETPSC[1:0] 外部触发预分频

外部触发信号 ETI 的频率不能超过 TIMER_CK 频率的 1/4。当输入较快的外部时钟时，可以使用预分频降低 ETIP 的频率。

00：预分频禁能。

01：2 分频。

10：4 分频。

11：8 分频。

11:8 ETFC[3:0] 外部触发滤波控制

外部触发信号可以通过数字滤波器进行滤波，该位域定义了数字滤波器的滤波能力。数字滤波器的基本原理是：以 f_{SAMP} 频率连续采样外部触发信号，同时记录采样相同电平的次数。当该次数达到配置的滤波能力时，则认为是一个有效的电平信号。

EXTFC[3:0]	次数	f_{SAMP}
4'b0000	Filter disabled.	
4'b0001	2	f_{CK_TIMER}
4'b0010	4	
4'b0011	8	
4'b0100	6	$f_{DTS_CK}/2$
4'b0101	8	
4'b0110	6	$f_{DTS_CK}/4$
4'b0111	8	
4'b1000	6	$f_{DTS_CK}/8$
4'b1001	8	
4'b1010	5	$f_{DTS_CK}/16$
4'b1011	6	
4'b1100	8	
4'b1101	5	$f_{DTS_CK}/32$
4'b1110	6	
4'b1111	8	

7 MSM 主-从模式

该位被用来同步被选择的定时器同时开始计数。通过 TRIGI 和 TRGO，定时器被连接在一起，TRGO 用做启动事件。

0：主从模式禁能。

1：主从模式使能。

6:4 TRGS[2:0] 触发选择

该位域用来指定选择哪一个信号作为用来同步计数器的触发输入源。

000: ITI0

001: ITI1

010: ITI2

011: ITI3

100: CI0F_ED

101: CI0FE0

110: CI1FE1

111: ETIFP

从模式被使能后这些位不能改。

3	OCRC	OCPRE 清除源选择 0: OCPRE_CLR_INT 连接到 OCPRE_CLR 输入。 1: OCPRE_CLR_INT 连接到 ETIF。
2:0	SMC[2:0]	从模式控制 000：关闭从模式。如果 CEN=1，则预分频器直接由内部时钟驱动。 001：正交译码器模式 0。根据 CI0FE0 的电平，计数器在 CI1FE1 的边沿向上/下计数。 010：正交译码器模式 1。根据 CI1FE1 的电平，计数器在 CI0FE0 的边沿向上/下计数。 011：正交译码器模式 2。根据另一个信号的输入电平，计数器在 CI0FE0 和 CI1FE1 的边沿向上/下计数。 100：复位模式。选中的触发输入的上升沿重新初始化计数器，并且产生更新事件。 101：暂停模式。当触发输入为高时，计数器的时钟开启。一旦触发输入变为低，则计数器时钟停止。 110：事件模式。计数器在触发输入的上升沿启动。 111：外部时钟模式 0。选中的触发输入的上升沿驱动计数器。 由于 CI0F_ED 是一个脉冲波形，而暂停模式是检测触发信号的电平，所以，当 CI0F_ED 用作触发输入时，暂停模式必须禁能。

DMA 和中断使能寄存器 (TIMERx_DMAINTEN)

地址偏移：0x0C
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TRGDEN	CMTDEN	CH3DEN	CH2DEN	CH1DEN	CH0DEN	UPDEN	BRKIE	TRGIE	CMTIE	CH3IE	CH2IE	CH1IE	CH0IE	UPIE
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:15	保留	必须保持复位值。
14	TRGDEN	触发 DMA 请求使能 0：禁止触发 DMA 请求 1：使能触发 DMA 请求
13	CMTDEN	换相 DMA 更新请求使能 0：禁止换相 DMA 更新请求 1：使能换相 DMA 更新请求

12	CH3DEN	通道 3 比较/捕获 DMA 请求使能 0：禁止通道 3 比较/捕获 DMA 请求 1：使能通道 3 比较/捕获 DMA 请求
11	CH2DEN	通道 2 比较/捕获 DMA 请求使能 0：禁止通道 2 比较/捕获 DMA 请求 1：使能通道 2 比较/捕获 DMA 请求
10	CH1DEN	通道 1 比较/捕获 DMA 请求使能 0：禁止通道 1 比较/捕获 DMA 请求 1：使能通道 1 比较/捕获 DMA 请求
9	CH0DEN	通道 0 比较/捕获 DMA 请求使能 0：禁止通道 0 比较/捕获 DMA 请求 1：使能通道 0 比较/捕获 DMA 请求
8	UPDEN	更新 DMA 请求使能 0：禁止更新 DMA 请求 1：使能更新 DMA 请求
7	BRKIE	中止中断使能 0：禁止中止中断 1：使能中止中断
6	TRGIE	触发中断使能 0：禁止触发中断 1：使能触发中断
5	CMTIE	换相更新中断使能 0：禁止换相更新中断 1：使能换相更新中断
4	CH3IE	通道 3 比较/捕获中断使能 0：禁止通道 3 中断 1：使能通道 3 中断
3	CH2IE	通道 2 比较/捕获中断使能 0：禁止通道 2 中断 1：使能通道 2 中断
2	CH1IE	通道 1 比较/捕获中断使能 0：禁止通道 1 中断 1：使能通道 1 中断
1	CH0IE	通道 0 比较/捕获中断使能 0：禁止通道 0 中断 1：使能通道 0 中断
0	UPIE	更新中断使能 0：禁止更新中断

1：使能更新中断

中断标志寄存器 (TIMERx_INTF)

地址偏移：0x10
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留			CH3OF	CH2OF	CH1OF	CH0OF	保留	BRKIF	TRGIF	CMTIF	CH3IF	CH2IF	CH1IF	CH0IF	UPIF
			rc_w0	rc_w0	rc_w0	rc_w0	.	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

位/位域	名称	描述
31:13	保留	必须保持复位值。
12	CH3OF	通道 3 捕获溢出标志 参见 CH0OF 描述
11	CH2OF	通道 2 捕获溢出标志 参见 CH0OF 描述
10	CH1OF	通道 1 捕获溢出标志 参见 CH0OF 描述
9	CH0OF	通道 0 捕获溢出标志 当通道 0 被配置为输入模式时，在 CH0IF 标志位已经被置 1 后，捕获事件再次发生时，该标志位可以由硬件置 1。该标志位由软件清 0. 0：无捕获溢出中断发生 1：发生了捕获溢出中断
8	保留	必须保持复位值。
7	BRKIF	中止中断标志位 当中止输入有效时，由硬件对该位置‘1’。 当中止输入无效时，则该位可由软件清‘0’。 0：无中止事件产生 1：中止输入上检测到有效电平
6	TRGIF	触发中断标志 当发生触发事件时，此标志会置 1，此位由软件清 0。当暂停模式使能时，触发输入的任意边沿都可以产生触发事件。否则，其它模式时，仅在触发输入端检测到有效边沿，产生触发事件。 0：无触发事件产生 1：触发中断产生

5	CMTIF	通道换相更新中断标志 当通道换相更新事件发生时此标志位被硬件置 1，此位由软件清 0。 0：无通道换相更新中断发生 1：通道换相更新中断发生
4	CH3IF	通道 3 比较/捕获中断标志 参见 CH0IF 描述
3	CH2IF	通道 2 比较/捕获中断标志 参见 CH0IF 描述
2	CH1IF	通道 1 比较/捕获中断标志 参见 CH0IF 描述
1	CH0IF	通道 0 比较/捕获中断标志 此标志由硬件置 1 软件清 0。当通道 0 在输入模式下时，捕获事件发生时此标志位被置 1；当通道 0 在输出模式下时，此标志位在一个比较事件发生时被置 1。 当通道 0 在输入模式下时，读 TIMERx_CH0CV 会将此标志清零。 0：无通道 0 中断发生 1：通道 0 中断发生
0	UPIF	更新中断标志 此位在任何更新事件发生时由硬件置 1，软件清 0。 0：无更新中断发生 1：发生更新中断

软件事件产生寄存器 (TIMERx_SWEVG)

地址偏移：0x14
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								BRKG	TRGG	CMTG	CH3G	CH2G	CH1G	CH0G	UPG
								w	w	w	w	w	w	w	w

位/位域	名称	描述
31:8	保留	必须保持复位值。
7	BRKG	产生中止事件 该位由软件置 1，用于产生一个中止事件，由硬件自动清 0。当此位被置 1 时，POEN 位被清 0 且 BRKIF 位被置 1，若开启对应的中断和 DMA，则产生相应的中断和 DMA 传输。 0：不产生中止事件

1：产生中止事件

6	TRGG	触发事件产生 此位由软件置 1，由硬件自动清 0. 当此位被置 1，TIMERx_INTF 寄存器的 TRGIF 标志位被置 1，若开启对应的中断和 DMA，则产生相应的中断和 DMA 传输。 0：无触发事件产生 1：产生触发事件
5	CMTG	通道换相更新事件发生 此位由软件置 1，由硬件自动清 0. 当此位被置 1，通道捕获/比较控制寄存器 (CHxEN, CHxNEN 和 CHxCOMCTL) 的互补输出被更新。 0：不产生通道控制更新事件 1：产生通道控制更新事件
4	CH3G	通道 3 捕获或比较事件发生 参见 CH0G 描述
3	CH2G	通道 2 捕获或比较事件发生 参见 CH0G 描述
2	CH1G	通道 1 捕获或比较事件发生 参见 CH0G 描述
1	CH0G	通道 0 捕获或比较事件发生 该位由软件置 1，用于在通道 0 产生一个捕获/比较事件，由硬件自动清 0。当此位被置 1，CH0IF 标志位被置 1，若开启对应的中断和 DMA，则发出相应的中断和 DMA 请求。此外，如果通道 0 配置为输入模式，计数器的当前值被 TIMERx_CH0CV 寄存器捕获，如果 CH0IF 标志位已经为 1，则 CH0OF 标志位被置 1。 0：不产生通道 0 捕获或比较事件 1：发生通道 0 捕获或比较事件
0	UPG	更新事件产生 此位由软件置 1，被硬件自动清 0。当此位被置 1，如果选择了中央对齐或向上计数模式，计数器被清 0。否则(向下计数模式)计数器将载入自动重载值，预分频计数器将同时被清除。 0：无更新事件产生 1：产生更新事件

通道控制寄存器 0 (TIMERx_CHCTL0)

地址偏移：0x18
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

CH1COMCEN	CH1COMCTL[2:0]	CH1COMSEN	CH1COMFEN	CH1MS[1:0]	CH0COMCEN	CH0COMCTL[2:0]	CH0COMSEN	CH0COMFEN	CH0MS[1:0]
CH1CAPFLT[3:0]		CH1CAPPSC[1:0]			CH0CAPFLT[3:0]		CH0CAPPSC[1:0]		
rw		rw		rw	rw		rw		rw

输出比较模式：

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	CH1COMCEN	通道 1 输出比较清 0 使能 参见 CH0COMCEN 描述
14:12	CH1COMCTL[2:0]	通道 1 输出比较模式 参见 CH0COMCTL 描述
11	CH1COMSEN	通道 1 输出比较影子寄存器使能 参见 CH0COMSEN 描述
10	CH1COMFEN	通道 1 输出比较快速使能 参见 CH0COMFEN 描述
9:8	CH1MS[1:0]	通道 1 模式选择 这些位定义了通道的方向和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2 寄存器的 CH1EN 位被清 0)时这些位才可以写。 00：通道 1 配置为输出 01：通道 1 配置为输入，IS1 映射在 CI1FE1 上 10：通道 1 配置为输入，IS1 映射在 CI0FE1 上 11：通道 1 配置为输入，IS1 映射在 ITS 上 注意：当 CH1MS[1:0]=11 时，需要通过 TRGS 位（位于 TIMERx_SMCFG 寄存器）选择内部触发输入。
7	CH0COMCEN	通道 0 输出比较清 0 使能 当此位被置 1，当检测到 ETIFP 信号输入高电平时，O0CPRE 参考信号被清 0 0：禁止通道 0 输出比较清零 1：使能通道 0 输出比较清零
6:4	CH0COMCTL[2:0]	通道 0 输出比较模式 此位定义了输出准备信号 O0CPRE 的输出比较模式，而 O0CPRE 决定了 CH0_O、CH0_ON 的值。另外，O0CPRE 高电平有效，而 CH0_O、CH0_ON 通道的极性取决于 CH0P、CH0NP 位。 000：时基。输出比较寄存器 TIMERx_CH0CV 与计数器 TIMERx_CNT 间的比较对 O0CPRE 不起作用 001：匹配时设置为高。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 为高。 010：匹配时设置为低。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 为低。 011：匹配时翻转。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 翻转。

100：强制为低。强制 O0CPRE 为低电平
101：强制为高。强制 O0CPRE 为高电平
110：PWM 模式 0。在向上计数时，一旦计数器值小于 TIMERx_CH0CV 时，O0CPRE 为高电平，否则为低电平。在向下计数时，一旦计数器的值大于 TIMERx_CH0CV 时，O0CPRE 为低电平，否则为高电平。
111：PWM 模式 1。在向上计数时，一旦计数器值小于 TIMERx_CH0CV 时，O0CPRE 为低电平，否则为高电平。在向下计数时，一旦计数器的值大于 TIMERx_CH0CV 时，O0CPRE 为高电平，否则为低电平。
如果配置在 PWM 模式下，只有当输出比较模式从时基模式变为 PWM 模式或者比较结果改变时，O0CPRE 电平才改变。
当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 且 CH0MS =00（比较模式）时此位不能被改变。

3	CH0COMSEN	通道 0 输出比较影子寄存器使能 当此位被置 1，TIMERx_CH0CV 寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。 0：禁止通道 0 输出/比较影子寄存器 1：使能通道 0 输出/比较影子寄存器 仅在单脉冲模式下(SPM =1)，可以在未确认影子寄存器的情况下使用 PWM 模式 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 且 CH0MS =00 时此位不能被改变。
2	CH0COMFEN	通道 0 输出比较快速使能 当该位为 1 时，如果通道配置为 PWM0 模式或者 PWM1 模式，会加快捕获/比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，CH0_O 被设置为比较电平而与比较结果无关。 0：禁止通道 0 输出比较快速. 1：使能通道 0 输出比较快速。
1:0	CH0MS[1:0]	通道 0 I/O 模式选择 这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2 寄存器的 CH0EN 位被清 0)时这些位才可写。 00：通道 0 配置为输出 01：通道 0 配置为输入，IS0 映射在 CI0FE0 上 10：通道 0 配置为输入，IS0 映射在 CI1FE0 上 11：通道 0 配置为输入，IS0 映射在 ITS 上 注意：当 CH0MS[1:0]=11 时，需要通过 TRGS 位（位于 TIMERx_SMCFG 寄存器）选择内部触发输入。

输入捕获模式：

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:12	CH1CAPFLT[3:0]	通道 1 输入捕获滤波控制 参见 CH0CAPFLT 描述
11:10	CH1CAPPSC[1:0]	通道 1 输入捕获预分频器

参见 CH0CAPPSC 描述

9:8 CH1MS[1:0] 通道 1 模式选择
与输出模式相同

7:4 CH0CAPFLT[3:0] 通道 0 输入捕获滤波控制
CI0 输入信号可以通过数字滤波器进行滤波，该位域配置滤波参数。
数字滤波器的基本原理：根据 f_{SAMP} 对 CI0 输入信号进行连续采样，并记录信号相同电平的次数。达到该位配置的滤波参数后，认为是有效电平。
滤波器参数配置如下：

CH0CAPFLT [3:0]	采样次数	f_{SAMP}
4'b0000	无滤波器	
4'b0001	2	f_{CK_TIMER}
4'b0010	4	
4'b0011	8	
4'b0100	6	$f_{DTS}/2$
4'b0101	8	
4'b0110	6	$f_{DTS}/4$
4'b0111	8	
4'b1000	6	$f_{DTS}/8$
4'b1001	8	
4'b1010	5	$f_{DTS}/16$
4'b1011	6	
4'b1100	8	
4'b1101	5	$f_{DTS}/32$
4'b1110	6	
4'b1111	8	

3:2 CH0CAPPSC[1:0] 通道 0 输入捕获预分频器
这 2 位定义了通道 0 输入的预分频系数。当 TIMERx_CHCTL2 寄存器中的 CH0EN =0 时，则预分频器复位。
00：无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获
01：每 2 个事件触发一次捕获
10：每 4 个事件触发一次捕获
11：每 8 个事件触发一次捕获

1:0 CH0MS[1:0] 通道 0 模式选择
与输出比较模式相同

通道控制寄存器 1 (TIMERx_CHCTL1)

地址偏移：0x1C
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

保留

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3COMCEN	CH3COMCTL[2:0]			CH3COMSEN	CH3COMFEN	CH3MS[1:0]		CH2COMCEN	CH2COMCTL[2:0]			CH2COMSEN	CH2COMFEN	CH2MS[1:0]	
CH3CAPFLT[3:0]				CH3CAPPSC[1:0]				CH2CAPFLT[3:0]				CH2CAPPSC[1:0]			
rw				rw		rw		rw				rw		rw	

输出比较模式：

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	CH3COMCEN	通道 3 输出比较清 0 使能 参见 CH0COMCEN 描述
14:12	CH3COMCTL[2:0]	通道 3 输出比较模式 参见 CH0COMCTL 描述
11	CH3COMSEN	通道 3 输出比较影子寄存器使能 参见 CH0COMSEN 描述
10	CH3COMFEN	通道 3 输出比较快速使能 参见 CH0COMFEN 描述
9:8	CH3MS[1:0]	通道 3 模式选择 这些位定义了通道的方向和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2 寄存器的 CH3EN 位被清 0)时这些位才可以写。 00：通道 3 配置为输出 01：通道 3 配置为输入，IS3 映射在 CI3FE3 上 10：通道 3 配置为输入，IS3 映射在 CI2FE3 上 11：通道 3 配置为输入，IS3 映射在 ITS 上 注意：当 CH3MS[1:0]=11 时，需要通过 TRGS 位（位于 TIMERx_SMCFG 寄存器）选择内部触发输入。
7	CH2COMCEN	通道 2 输出比较清 0 使能 当此位被置 1，当检测到 ETIFP 输入高电平时，O2CPRE 参考信号被清 0 0：使能通道 2 输出比较清零 1：禁止通道 2 输出比较清零
6:4	CH2COMCTL[2:0]	通道 2 输出比较模式 此位定义了输出准备信号 O2CPRE 的输出比较模式，而 O2CPRE 决定了 CH2_O、CH2_ON 的值。另外，O2CPRE 高电平有效，而 CH2_O、CH2_ON 通道的极性取决于 CH2P、CH2NP 位。 000：时基。输出比较寄存器 TIMERx_CH2CV 与计数器 TIMERx_CNT 间的比较对 O2CPRE 不起作用 001：匹配时设置为高。当计数器的值与捕获/比较值寄存器 TIMERx_CH2CV 相同时，强制 O2CPRE 为高。 010：匹配时设置为低。当计数器的值与捕获/比较值寄存器 TIMERx_CH2CV 相同

		时，强制 O2CPRE 为低。 011：匹配时翻转。当计数器的值与捕获/比较值寄存器 TIMERx_CH2CV 相同时，强制 O2CPRE 翻转。 100：强制为低。强制 O2CPRE 为低电平 101：强制为高。强制 O2CPRE 为高电平 110：PWM 模式 0。在向上计数时，一旦计数器值小于 TIMERx_CH2CV 时，O2CPRE 为高电平，否则为低电平。在向下计数时，一旦计数器的值大于 TIMERx_CH2CV 时，O2CPRE 为低电平，否则为高电平。 111：PWM 模式 1。在向上计数时，一旦计数器值小于 TIMERx_CH2CV 时，O2CPRE 为低电平，否则为高电平。在向下计数时，一旦计数器的值大于 TIMERx_CH2CV 时，O2CPRE 为高电平，否则为低电平。 如果配置在 PWM 模式下，只有当输出比较模式从时基模式变为 PWM 模式或者比较结果改变时，O2CPRE 电平才改变。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 且 CH2MS =00（比较模式）时此位不能被改变。
3	CH2COMSEN	通道 0 输出比较影子寄存器使能 当此位被置 1，TIMERx_CH2CV 寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。 0：禁止通道 2 输出/比较影子寄存器 1：使能通道 2 输出/比较影子寄存器 仅在单脉冲模式下(SPM =1)，可以在未确认影子寄存器情况下使用 PWM 模式 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 且 CH2MS =00 时此位不能被改变。
2	CH2COMFEN	通道 2 输出比较快速使能 当该位为 1 时，如果通道配置为 PWM0 模式或者 PWM1 模式，会加快捕获/比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，CH2_O 被设置为比较电平而与比较结果无关。 0：禁止通道 2 输出比较快速. 1：使能通道 2 输出比较快速。
1:0	CH2MS[1:0]	通道 2 I/O 模式选择 这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2 寄存器的 CH2EN 位被清 0)时这些位才可写。 00：通道 2 配置为输出 01：通道 2 配置为输入，IS2 映射在 CI2FE2 上 10：通道 2 配置为输入，IS2 映射在 CI3FE2 上 11：通道 2 配置为输入，IS2 映射在 ITS 上. 注意：当 CH2MS[1:0]=11 时，需要通过 TRGS 位（位于 TIMERx_SMCFG 寄存器）选择内部触发输入。

输入捕获模式：

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:12	CH3CAPFLT[3:0]	通道 3 输入捕获滤波控制

参见 CH0CAPFLT 描述

11:10 CH3CAPPSC[1:0] 通道 3 输入捕获预分频器
参见 CH0CAPPSC 描述

9:8 CH3MS[1:0] 通道 3 模式选择
与输出模式相同

7:4 CH2CAPFLT[3:0] 通道 2 输入捕获滤波控制
CI2 输入信号可以通过数字滤波器进行滤波，该位域配置滤波参数。
数字滤波器的基本原理：根据 f_{SAMP} 对 CI2 输入信号进行连续采样，并记录信号相同电平的次数。达到该位配置的滤波参数后，认为是有效电平。
滤波器参数配置如下：

CH2CAPFLT [3:0]	采样次数	f_{SAMP}
4'b0000	无滤波器	
4'b0001	2	f_{CK_TIMER}
4'b0010	4	
4'b0011	8	
4'b0100	6	$f_{DTS}/2$
4'b0101	8	
4'b0110	6	$f_{DTS}/4$
4'b0111	8	
4'b1000	6	$f_{DTS}/8$
4'b1001	8	
4'b1010	5	$f_{DTS}/16$
4'b1011	6	
4'b1100	8	
4'b1101	5	$f_{DTS}/32$
4'b1110	6	
4'b1111	8	

3:2 CH2CAPPSC[1:0] 通道 2 输入捕获预分频器
这 2 位定义了通道 2 输入的预分频系数。当 TIMERx_CHCTL2 寄存器中的 CH2EN =0 时，则预分频器复位。
00：无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获
01：每 2 个事件触发一次捕获
10：每 4 个事件触发一次捕获
11：每 8 个事件触发一次捕获

1:0 CH2MS[1:0] 通道 2 模式选择
与输出比较模式相同

通道控制寄存器 2 (TIMERx_CHCTL2)

地址偏移：0x20
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留.		CH3P	CH3EN	CH2NP	CH2NEN	CH2P	CH2EN	CH1NP	CH1NEN	CH1P	CH1EN	CH0NP	CH0NEN	CH0P	CH0EN
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:14	保留	必须保持复位值。
13	CH3P	通道 3 极性 参考 CH0P 描述
12	CH3EN	通道 3 使能 参考 CH0EN 描述
11	CH2NP	通道 2 互补输出极性 参考 CH0NP 描述
10	CH2NEN	通道 2 互补输出使能 参考 CH0NEN 描述
9	CH2P	通道 2 极性 参考 CH0P 描述
8	CH2EN	通道 2 使能 参考 CH0EN 描述
7	CH1NP	通道 1 互补输出极性 参考 CH0NP 描述
6	CH1NEN	通道 1 互补输出使能 参考 CH0NEN 描述
5	CH1P	通道 1 极性 参考 CH0P 描述
4	CH1EN	通道 1 使能 参考 CH0EN 描述
3	CH0NP	通道 0 互补输出极性 当通道 0 配置为输出模式，此位定义了互补输出信号的极性。 0：通道0互补输出高电平为有效电平 1：通道0互补输出低电平为有效电平 当通道 0 配置为输入模式时，此位和 CH0P 联合使用，作为输入信号 CI0 的极性选择控制信号。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 或 10 时此位不能被更改。
2	CH0NEN	通道 0 互补输出使能

当通道 0 配置为输出模式时，将此位置 1 使能通道 0 的互补输出。
0：禁止通道 0 互补输出
1：使能通道 0 互补输出

1	CH0P	通道 0 极性 当通道 0 配置为输出模式时，此位定义了输出信号极性。 0：通道0高电平为有效电平 1：通道0低电平为有效电平 当通道 0 配置为输入模式时，此位定义了 CI0 信号极性 [CH0NP, CH0P] 将选择 CI0FE0 或者 CI1FE0 的有效边沿或者捕获极性 [CH0NP==0, CH0P==0]：把 CIxFE0 的上升沿作为捕获或者从模式下触发的有效信号，并且 CIxFE0 不会被翻转。 [CH0NP==0, CH0P==1]：把 CIxFE0 的下降沿作为捕获或者从模式下触发的有效信号，并且 CIxFE0 会被翻转。 [CH0NP==1, CH0P==0]：保留。 [CH0NP==1, CH0P==1]：把 CIxFE0 的上升沿和下降沿都作为捕获或者从模式下触发的有效信号，并且 CIxFE0 不会被翻转。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 或 10 时此位不能被更改。
0	CH0EN	通道 0 捕获/比较使能 当通道 0 配置为输出模式时，将此位置 1 使能 CH0_O 信号有效。当通道 0 配置为输入 模式时，将此位置 1 使能通道 0 上的捕获事件。 0：禁止通道 0 1：使能通道 0

计数器寄存器 (TIMERx_CNT)

地址偏移：0x24
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CNT[15:0]	这些位是当前的计数值。写操作能改变计数器值。

预分频寄存器 (TIMERx_PSC)

地址偏移：0x28
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	PSC[15:0]	计数器时钟预分频值 计数器时钟等于 TIMER_CK 时钟除以(PSC+1)，每次当更新事件产生时，PSC 的值被装入到对应的影子寄存器。

计数器自动重载寄存器 (TIMERx_CAR)

地址偏移：0x2C
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

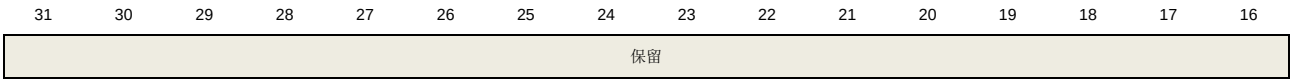
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CARL[15:0]															
rw															

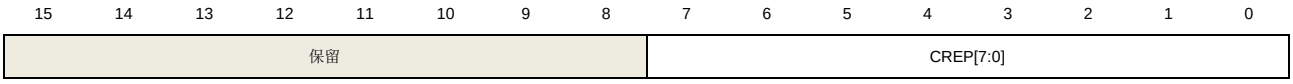
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CARL[15:0]	计数器自动重载值 这些位定义了计数器的自动重载值。

重复计数寄存器 (TIMERx_CREP)

地址偏移：0x30
复位值：0x0000 0000

该寄存器只能按字(32位)访问。



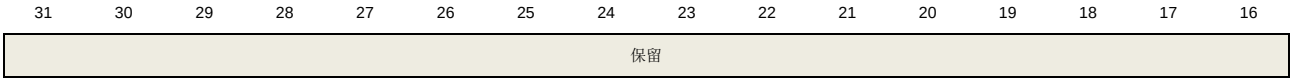


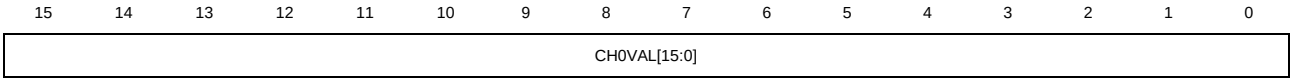
位/位域	名称	描述
31:8	保留	必须保持复位值。.
7:0	CREP[7:0]	重复计数器的值 这些位定义了更新事件的产生速率。重复计数器计数值减为 0 时产生更新事件。影子寄存器的更新速率也会受这些位影响(前提是影子寄存器被使能)。

通道 0 捕获/比较寄存器 (TIMERx_CH0CV)

地址偏移：0x34
复位值：0x0000 0000

该寄存器只能按字(32位)访问。



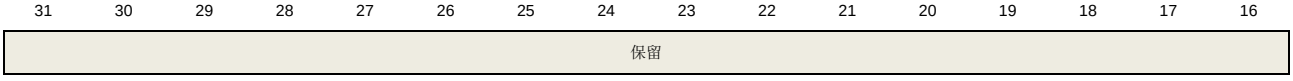


位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CH0VAL[15:0]	通道 0 的捕获或比较值 当通道 0 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 0 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

通道 1 捕获/比较寄存器 (TIMERx_CH1CV)

地址偏移：0x38
复位值：0x0000 0000

该寄存器只能按字(32位)访问。



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1VAL[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CH1VAL[15:0]	通道 1 的捕获或比较值 当通道 1 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 1 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

通道 2 捕获/比较寄存器 (TIMERx_CH2CV)

地址偏移：0x3C
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH2VAL[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CH2VAL[15:0]	通道 2 的捕获或比较值 当通道 2 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 2 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

通道 3 捕获/比较寄存器 (TIMERx_CH3CV)

地址偏移：0x40
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

CH3VAL[15:0]
rw

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CH3VAL[15:0]	通道 3 的捕获或比较值 当通道 3 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 3 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

互补通道保护寄存器 (TIMERx_CCHP)

地址偏移：0x44

复位值: 0x0000 0000

该寄存器通过字访问（32位）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POEN	OAEN	BRKP	BRKEN	ROS	IOS	PROT[1:0]		DTCFG[7:0]							
rw	rw	rw	rw	rw	rw	rw		rw							

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	POEN	所有的通道输出使能 该位通过以下方式置 1： –写 1 置位 –如果 OAEN=1，则在下一次更新事件发生时置 1. 该位通过以下方式清 0： –写 0 清 0 –有效的中止输入（异步） 如果一个通道配置为输出模式，如果设置了相应的使能位（TIMERx_CHCTL2 寄存器的 CHxEN, CHxNEN 位），则开启 CHx_O 和 CHx_ON 输出。 0：禁止通道输出 1：使能通道输出 注意：仅当 CHxMS[1:0]=2’b00 时该位有效。
14	OAEN	自动输出使能 0：POEN 位只能使用软件方式置 1。 1：如果中止输入无效，下一次更新事件发生时，POEN 位将会置 1。 此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0] =00 时才可修改。
13	BRKP	中止极性

此位定义了中止输入信号 BRKIN 的极性。
0：中止输入低电平有效。
1：中止输入高电平有效。

12 BRKEN

中止使能
此位置 1 使能中止事件和 CCS 时钟失败事件输入。
0：禁能中止输入。
1：使能中止输入。
此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0] =00 时才可修改。

11 ROS

运行模式下“关闭状态”使能
当 POEN 位被置 1（运行模式），此位可以被置 1 来使能通道(带有互补输出且配置为输出模式)的输出“关闭状态”。
0：输出“关闭状态”禁能。当 CHxEN 或者 CHxNEN 位被清零，对应通道为输出“禁能状态”。
1：输出“关闭状态”使能。当 CHxEN 或者 CHxNEN 位被清零，对应通道为输出“关闭状态”。
此位在 TIMERx_CCHP 寄存器的 PROT [1:0]=10 或 11 时不能被更改。

10 IOS

空闲模式下“关闭状态”使能
当 POEN 位被清 0（空闲模式），此位可以被置 1 来使能通道(带有互补输出且配置为输出模式)的输出“关闭状态”。
0：输出“关闭状态”禁能。当 CHxEN 和 CHxNEN 位均被清零，对应通道为输出“禁能状态”。
1：输出“关闭状态”使能。不论 CHxEN 和 CHxNEN 位的值，对应通道为输出“关闭状态”。
此位在 TIMERx_CCHP 寄存器的 PROT [1:0]=10 或 11 时不能被更改。

9:8 PROT[1:0]

互补寄存器保护控制
这两位定义了寄存器的写保护特性。
00：禁能保护模式。无写保护。
01：PROT 模式 0。TIMERx_CTL1 寄存器中 ISOx/ISOxN 位，TIMERx_CCHP 寄存器中 BRKEN/BRKP/OAEN/DTCFG 位写保护。
10：PROT 模式 1。除了 PROT 模式 0 下的寄存器写保护外，还有 TIMERx_CHCTL2 寄存器中 CHxP/CHxNP 位（如果相应通道配置为输出模式），TIMERx_CCHP 寄存器中 ROS/IOS 位。
11：PROT 模式 2.。除了 PROT 模式 1 下的寄存器写保护外，还有 TIMERx_CHCTL0/1 中 CHxCOMCTL/ CHxCOMSEN 位（如果相关通道配置为输出模式）写保护。
系统复位后这两位只能被写一次，一旦 TIMERx_CCHP 寄存器被写入，这两位被写保护。

7:0 DTCFG[7:0]

死区时间控制
DTCFG 值和死区时间的关系如下：

DTCFG[7:5]	The duration of dead-time
3’b0xx	DTCFG[7:0] * t_{DTS_CK}
3’b10x	(64+ DTCFG[5:0]) * t_{DTS_CK} *2
3’b110	(32+ DTCFG[4:0]) * t_{DTS_CK} *8

3’b111	(32+ DTCFG[4:0]) * t_{DTS_CK} *16

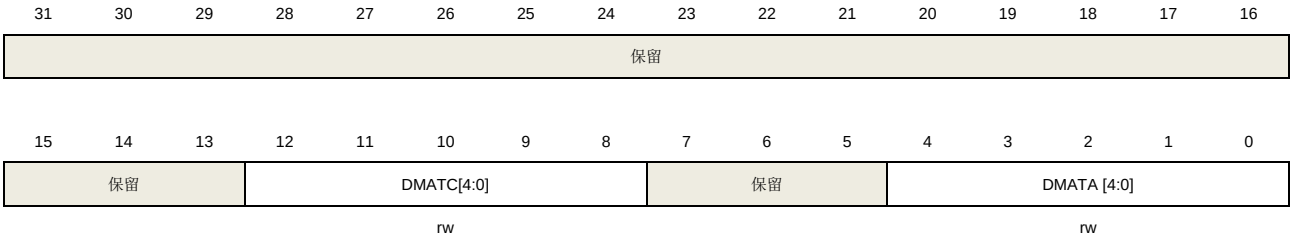
注意：

1. t_{DTS_CK} 是 DTS_CK 的周期，由 TIMERx_CTL0 中的 CKDIC[1:0]定义。
2. 此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0]=00 时才可修改。

DMA 配置寄存器 (TIMERx_DMACFG)

地址偏移：0x48
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

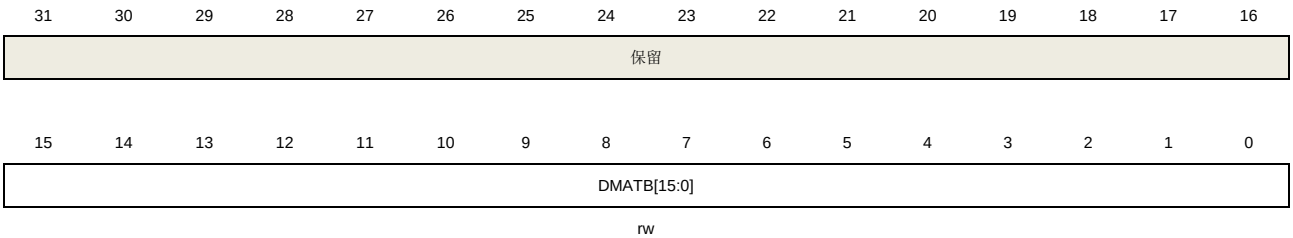


位/位域	名称	描述
31:13	保留	必须保持复位值。
12:8	DMATC [4:0]	DMA 传输计数 该位域定义了 DMA 访问（读写）TIMERx_DMATB 寄存器的数量 n，n = (DMATC [4:0] +1). DMATC [4:0] 从 5’b0_0000 到 5’b1_0001.
7:5	保留	必须保持复位值。
4:0	DMATA [4:0]	DMA 传输起始地址 该位域定义了 DMA 访问 TIMERx_DMATB 寄存器的第一个地址。当通过 TIMERx_DMA 第一次访问时，访问的就是该位域指定的地址。第二次访问 TIMERx_DMATB 时，将访问起始地址+0x4。

DMA 发送缓冲区寄存器 (TIMERx_DMATB)

地址偏移：0x4C
复位值：0x0000 0000

该寄存器只能按字(32位)访问。



位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	DMATB [15:0]	DMA 发送缓冲 对这个寄存器的读或写，（起始地址+传输次数*4）地址范围内的寄存器会被访问 传输次数由硬件计算，范围为 0 到 DMATC。

15.2. 通用定时器 L0（TIMERx, x=1, 2）

15.2.1. 简介

通用定时器 L0 是 4 通道定时器，支持输入捕获，输出比较，产生 PWM 信号控制电机和电源管理。通用定时器 L0 计数器是 16 位无符号计数器。

通用定时器 L0 是可编程的，可以被用来计数，其外部事件可以驱动其他定时器。

定时器和定时器之间是相互独立，但是他们可以被同步在一起形成一个更大的定时器，这些定时器的计数器一致地增加。

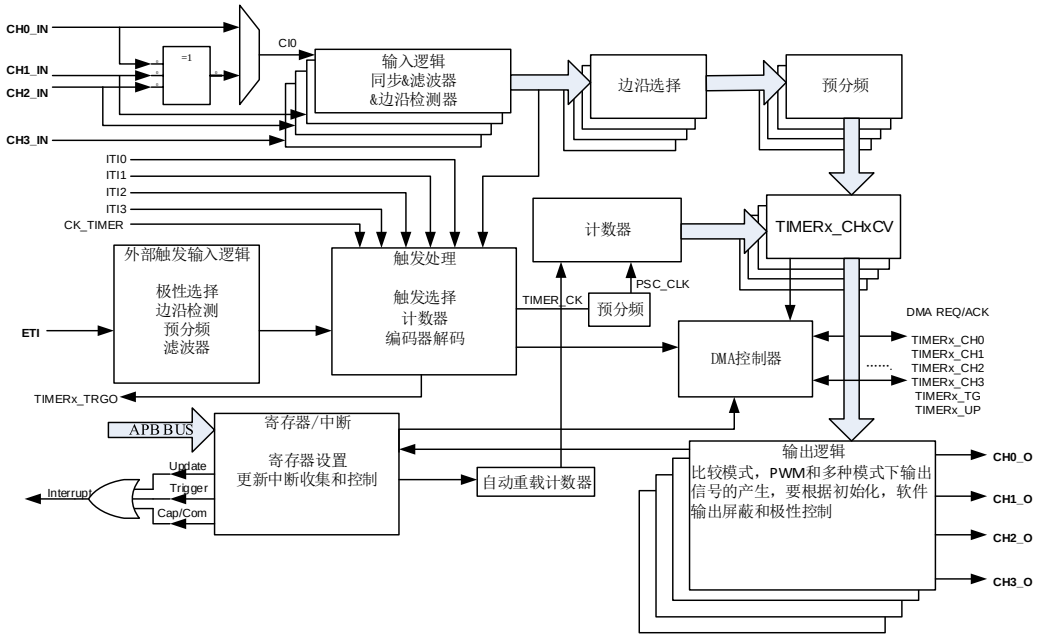
15.2.2. 主要特性

- 总通道数：4；
- 计数器宽度：16位(TIMER2)，32位(TIMER1)；
- 时钟源可选：内部时钟，内部触发，外部输入，外部触发；
- 多种计数模式：向上计数，向下计数和中央计数；
- 正交正交译码器接口：被用来追踪运动和分辨旋转方向和位置；
- 霍尔传感器接口：用来做三相电机控制；
- 可编程的预分频器：16位，运行时可以被改变；
- 每个通道可配置：输入捕获模式，输出比较模式，可编程的PWM模式，单脉冲模式；
- 自动重装载功能；
- 中断输出和DMA请求：更新事件，触发事件，比较/捕获事件；
- 多个定时器的菊链使得一个定时器可以同时启动多个定时器；
- 定时器的同步允许被选择的定时器在同一个时钟周期开始计数；
- 定时器主-从管理。

15.2.3. 结构框图

*图 15-31. 通用定时器 L0 结构框图*提供了通用定时器 L0 的内部细节

图 15-31. 通用定时器 L0 结构框图



15.2.4. 功能描述

时钟源配置

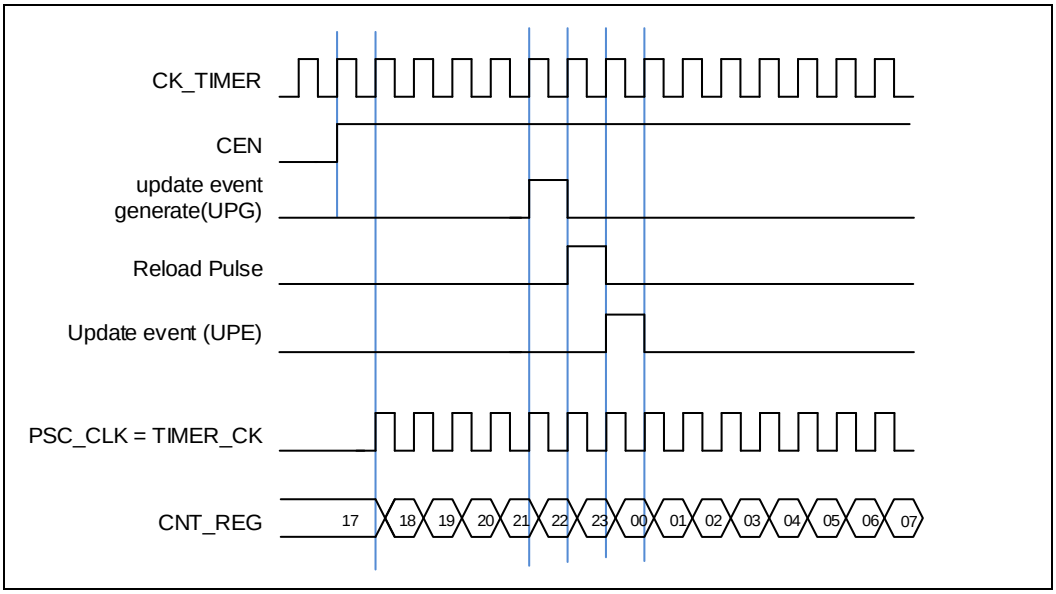
通用定时器 L0 可以由内部时钟源 CK_TIMER 或者由 SMC（TIMERx_SMCFG 寄存器位[2:0]）控制的复用时钟源驱动。

- SMC[2:0]==3’b000，定时器选择内部时钟源（连接到RCU模块的CK_TIMER）

如果 SMC[2:0]==3’b000，默认用来驱动计数器预分频器的是内部时钟源 CK_TIMER。当 CEN 置位，CK_TIMER 经过预分频器（预分频值由 TIMERx_PSC 寄存器确定）产生 PSC_CLK。

如果将 TIMERx_SMCFG 寄存器的 SMC[2:0]设置为 0x1、0x2、0x3 和 0x7，预分频器被其他时钟源(由 TIMERx_SMCFG 寄存器的 TRGS [2:0]区域选择)驱动，在下文说明。当 SMC 位被设置为 0x4、0x5 和 0x6，计数器预分频器时钟源由内部时钟 CK_TIMER 驱动。

图 15-32. 内部时钟分频为 1 时，计数器的时序图



- SMC[2:0]==3'b111(外部时钟模式0)，定时器选择外部输入引脚作为时钟源

计数器预分频器可以在 TIMERx_CI0/ TIMERx_CI1 引脚的每个上升沿或下降沿计数。这种模式可以通过设置 SMC [2:0]为 0x7 同时设置 TRGS [2:0]为 0x4，0x5 或 0x6 来选择。CIx 是 TIMERx_CIx 通过数字滤波器采样后的信号。

计数器预分频器也可以在内部触发信号 ITI0/1/2/3 的上升沿计数。这种模式可以通过设置 SMC [2:0]为 0x7 同时设置 TRGS [2:0]为 0x0, 0x1, 0x2 或者 0x3。

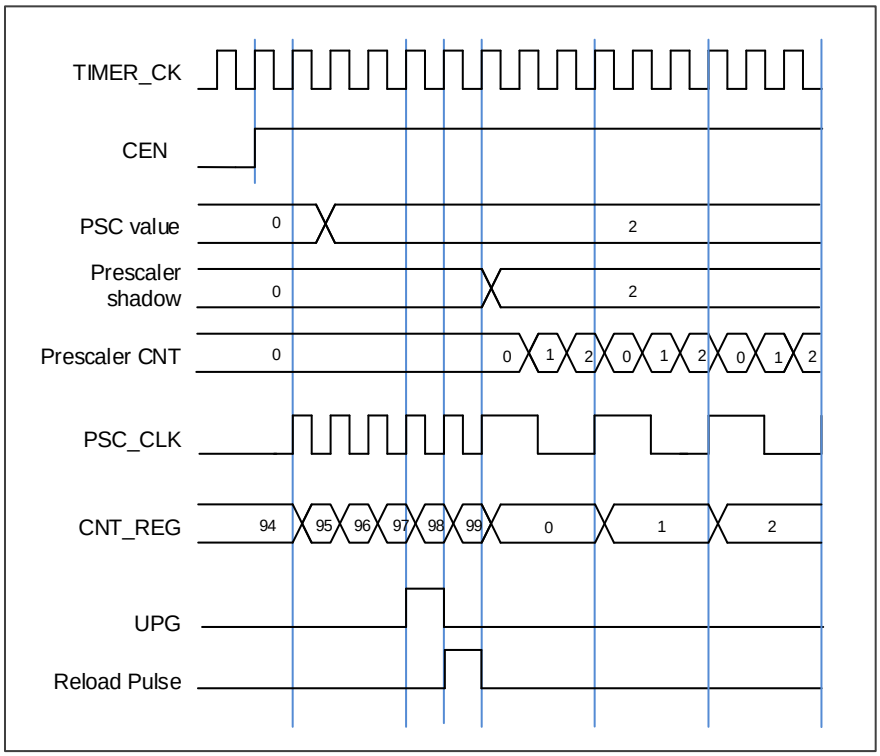
- SMC==1'b1(外部时钟模式1)，定时器选择外部输入引脚ETI作为时钟源

计数器预分频器可以在外部引脚 ETI 的每个上升沿或下降沿计数。这种模式可以通过设置 TIMERx_SMCFG 寄存器中的 SMC1 位为 1 来选择。另一种选择 ETI 信号作为时钟源方式是，设置 SMC [2:0]为 0x7 同时设置 TRGS [2:0]为 0x7。注意 ETI 信号是通过数字滤波器采样 ETI 引脚得到的。。如果选择 ETIF 信号为时钟源，触发控制器包括边沿监测电路将在每个 ETI 信号上升沿产生一个时钟脉冲来为计数器预分频器提供时钟。

时钟预分频器

预分频器可以将定时器的时钟（TIMER_CK)频率按 1 到 65536 之间的任意值分频，分频后的时钟 PSC_CLK 驱动计数器计数。分频系数受预分频寄存器 TIMERx_PSC 控制，这个控制寄存器带有缓冲器，它能够在运行时被改变。新的预分频器的参数在下一次更新事件到来时被采用。

图 15-33. 当 PSC 数值从 0 变到 2 时，计数器的时序图



计数器向上计数模式

在这种模式，计数器的计数方向是向上计数。计数器从0开始向上连续计数到自动加载值（定义在TIMERx_CAR寄存器中），一旦计数器计数到自动加载值，会重新从0开始向上计数并产生上溢事件。在向上计数模式中，TIMERx_CTL0寄存器中的计数方向控制位DIR应该被设置成0。

当通过TIMERx_SWEVG寄存器的UPG位置1来设置更新事件时，计数值会被清0，并产生更新事件。

如果TIMERx_CTL0寄存器的UPDIS置1，则禁止更新事件。

当发生更新事件时，所有影子寄存器（计数器自动重载寄存器，预分频寄存器）都将被更新。

图 15-34. 向上计数时序图，PSC=0/2 和 *图 15-35. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值*给出了一些例子，当 TIMERx_CAR=0x99 时，计数器在不同预分频因子下的行为。

图 15-34. 向上计数时序图，PSC=0/2

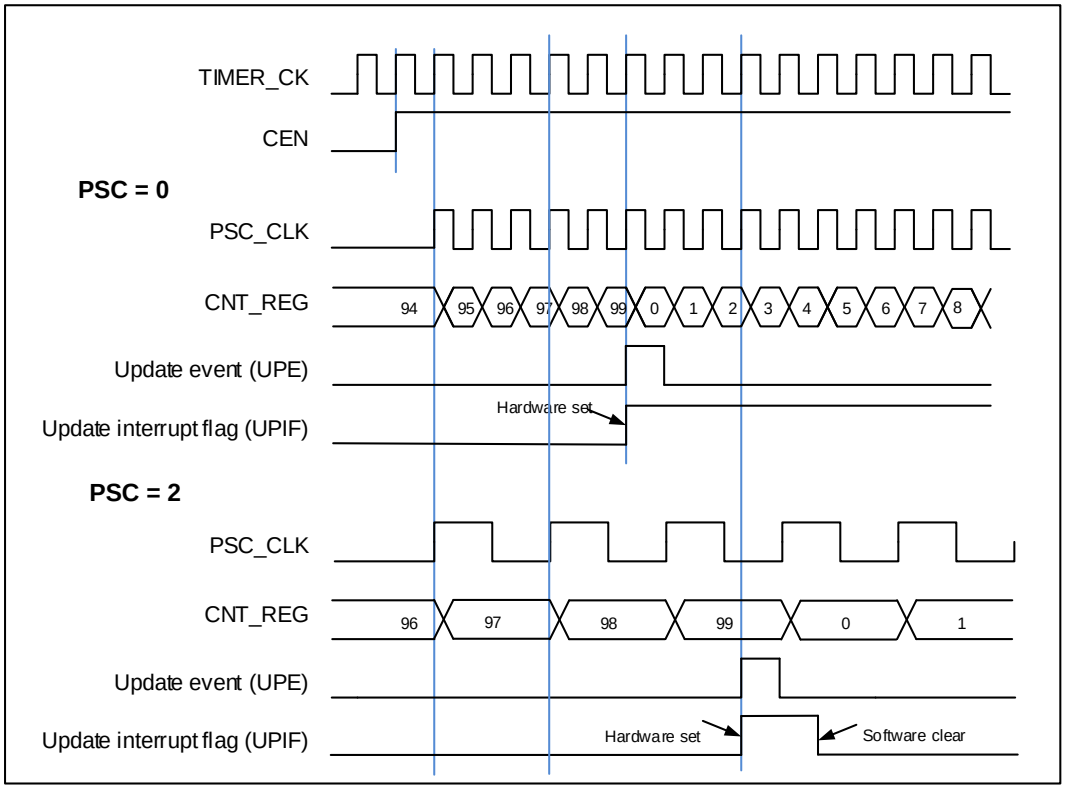
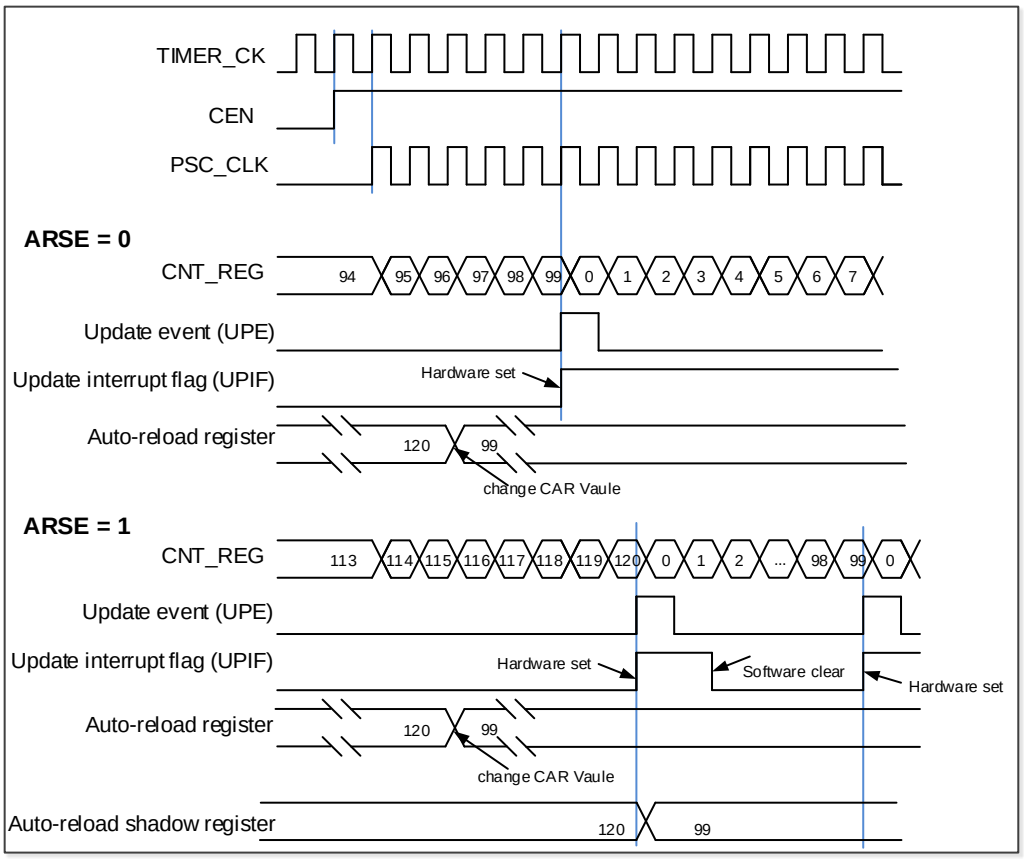


图 15-35. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值



计数器向下计数模式

在这种模式，计数器的计数方向是向下计数。计数器从自动加载值（定义在 TIMERx_CAR 寄存器中）向下连续计数到 0。一旦计数器计数到 0，计数器会重新从自动加载值开始计数并产生下溢事件。在向下计数模式中，TIMERx_CTL0 寄存器中的计数方向控制位 DIR 应该被设置成 1。

当通过 TIMERx_SWEVG 寄存器的 UPG 位置 1 来设置更新事件时，计数值会被初始化为自动加载值，并产生更新事件。

如果 TIMERx_CTL0 寄存器的 UPDIS 置 1，则禁止更新事件。

当发生更新事件时，所有影子寄存器（计数器自动重载寄存器，预分频寄存器）都将被更新。

图 15-36. 向下计数时序图，PSC=0/2 和 *图 15-37. 向下计数时序图，在运行时改变 TIMERx CAR 寄存器值*给出了一些例子，当 TIMERx_CAR=0x99 时，计数器在不同时钟频率下的行为。

图 15-36. 向下计数时序图，PSC=0/2

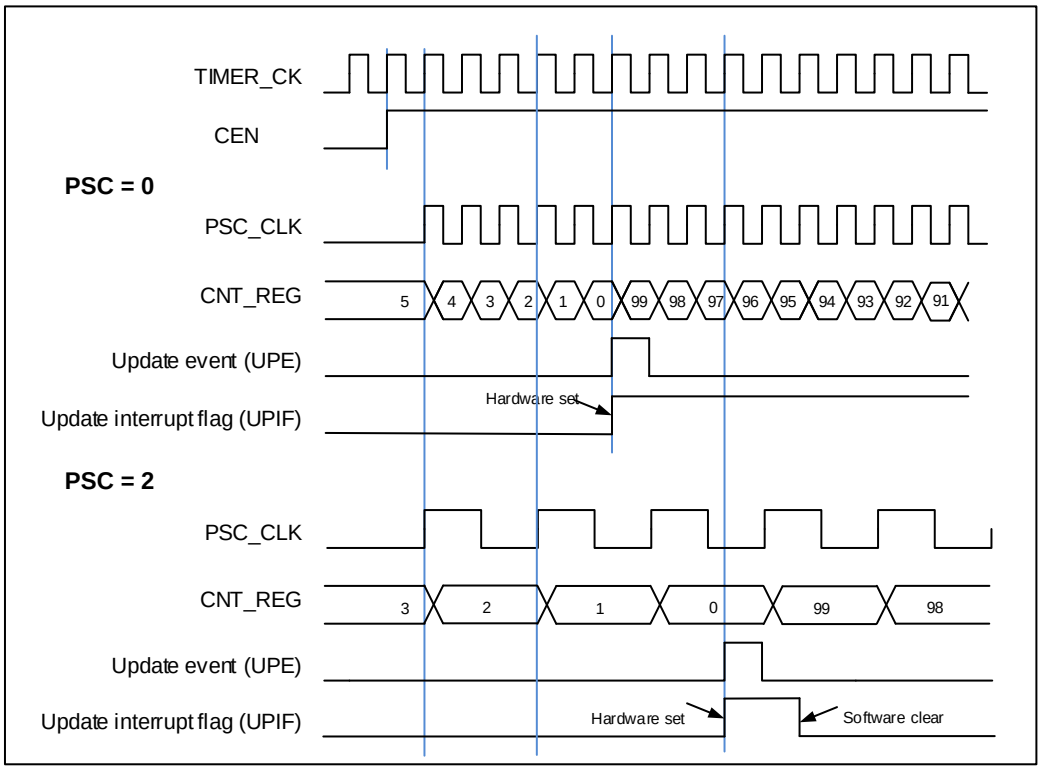
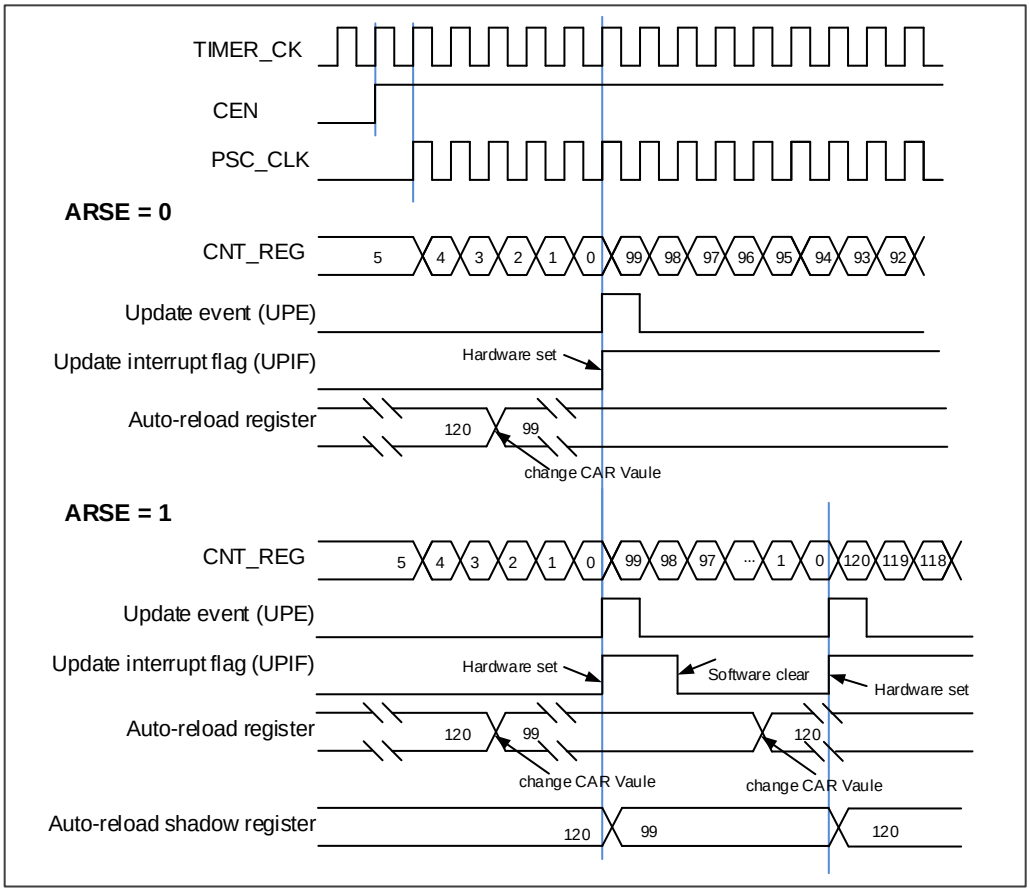


图 15-37. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值



计数器中央对齐模式

在中央对齐模式下，计数器交替的从 0 开始向上计数到自动加载值，然后再向下计数到 0。。向上计数模式中， 定时器模块在计数器计数到（自动加载值-1）产生一个上溢事件；向下计数模式中，定时器模块在计数器计数到 1 时产生一个下溢事件。在中央计数模式中，TIMERx_CTL0 寄存器中的计数方向控制位 DIR 只读，表明了的计数方向。

将 TIMERx_SWEVG 寄存器的 UPG 位置 1 可以初始化计数值为 0，并产生一个更新事件，而无需考虑计数器在中央模式下是向上计数还是向下计数。

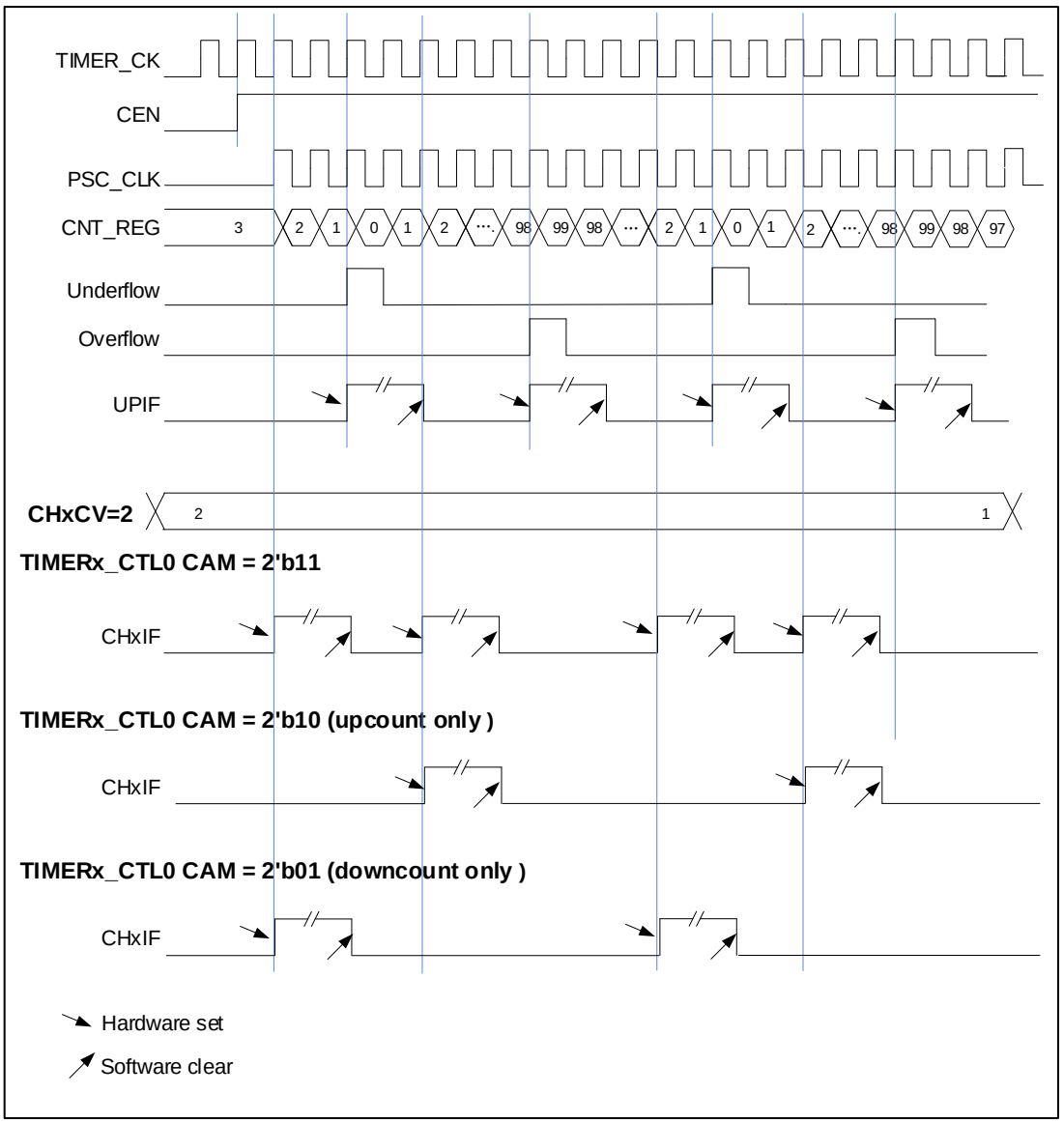
上溢或者下溢时，TIMERx_INTF 寄存器中的 UPIF 位都会被置 1，然而 CHxIF 位置 1 与 TIMERx_CTL0 寄存器中 CAM 的值有关。具体细节参考*图 15-38. 中央计数模式计数器时序图*

如果 TIMERx_CTL0 寄存器的 UPDIS 置 1，则禁止更新事件。

当发生更新事件时，所有影子寄存器（计数器自动重载寄存器，预分频寄存器）都将被更新。

*图 15-38. 中央计数模式计数器时序图*给出了一些例子，当 TIMERx_CAR=0x99，TIMERx_PSC=0x0 时，计数器的行为。

图 15-38. 中央计数模式计数器时序图



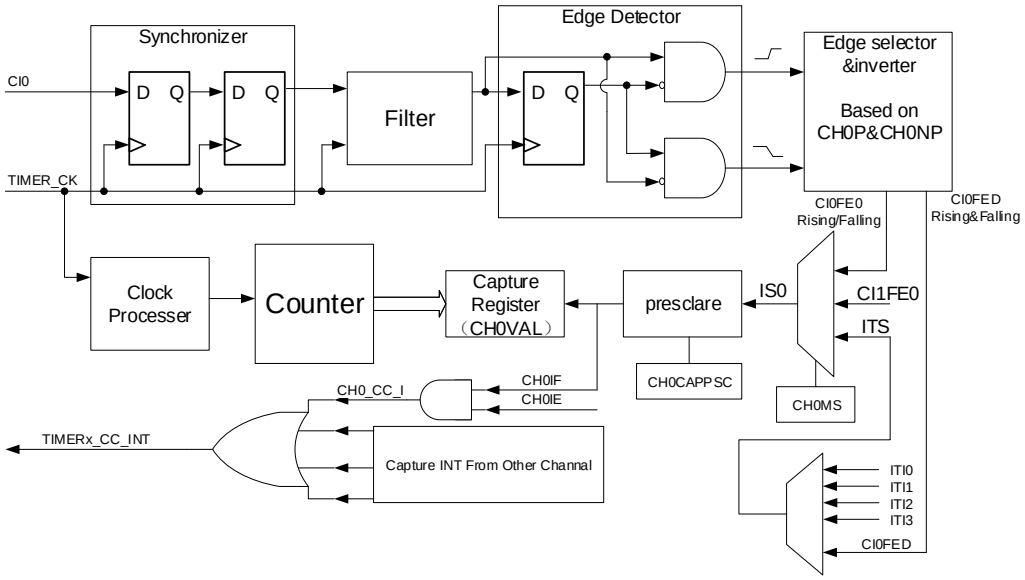
输入捕获和输出比较通道

通用定时器 L0 拥有四个独立的通道用于捕获输入或比较输出是否匹配。每个通道都围绕一个通道捕获比较寄存器建立，包括一个输入级，通道控制器和输出级。

■ 通道输入捕获功能

通道输入捕获功能允许通道测量一个波形时序，频率，周期，占空比等。输入级包括一个数字滤波器，一个通道极性选择，边沿检测和一个通道预分频器。如果在输入引脚上出现被选择的边沿，TIMERx_CHxCV 寄存器会捕获计数器当前的值，同时 CHxIF 位被置 1，如果 CHxIE = 1 则产生通道中断。

图 15-39. 通道输入捕获原理



通道输入信号CIx有两种选择，一种是TIMERx_CHx信号，另一种是TIMERx_CH0,TIMERx_CH1和TIMERx_CH2异或之后的信号。通道输入信号CIx先被TIMER_CK信号同步，然后经过数字滤波器采样，产生一个被滤波后的信号。通过边沿检测器，可以选择检测上升沿或者下降沿。通过配置CHxP选择使用上升沿或者下降沿。配置CHxMS.，可以选择其他通道的输入信号，内部触发信号。配置IC预分频器，使得若干个输入事件后才产生一个有效的捕获事件。捕获事件发生，CHxVAL存储计数器的值。

配置步骤如下：

第一步：滤波器配置（TIMERx_CHCTL0寄存器中CHxCAPFLT）：
根据输入信号和请求信号的质量，配置相应的CHxCAPFLT。

第二步：边沿选择（TIMERx_CHCTL2寄存器中CHxP/CHxNP）：
配置CHxP/CHxNP选择上升沿或者下降沿。

第三步：捕获源选择（TIMERx_CHCTL0寄存器中CHxMS）：
一旦通过配置CHxMS选择输入捕获源，必须确保通道配置在输入模式（CHxMS!=0x0），而且TIMERx_CHxCV寄存器不能再被写。

第四步：中断使能（TIMERx_DMAINTEN寄存器中CHxIE和CHxDEN）：
使能相应中断，可以获得中断和DMA请求。

第五步：捕获使能（TIMERx_CHCTL2寄存器中CHxEN）。

结果：当期望的输入信号发生时，TIMERx_CHxCV被设置成当前计数器的值，CHxIF为置1。如果CHxIF位已经为1，则CHxOF位置1。根据TIMERx_DMAINTEN寄存器中CHxIE和CHxDEN的配置，相应的中断和DMA请求会被提出。

直接产生：软件设置CHxG位，会直接产生中断和DMA请求。

输入捕获模式也可用来测量TIMERx_CHx引脚上信号的脉冲波宽度。例如，一个PWM波连接到CI0。配置TIMERx_CHCTL0寄存器中CH0MS为2’b01，选择通道0的捕获信号为CI0并设置上升沿捕获。配置TIMERx_CHCTL0寄存器中CH1MS为2’b10，选择通道1捕获信号为CI0并设

置下降沿捕获。计数器配置为复位模式，在通道0的上升沿复位。TIMERX_CH0CV寄存器测量PWM的周期值，TIMERx_CH1CV寄存器测量PWM占空比值。

通道输出比较功能

图 15-40. 通道输出比较原理（x=0，1，2，3）

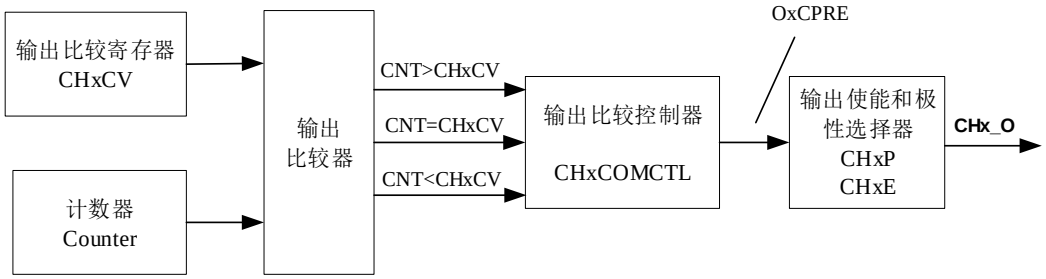


*图 15-40. 通道输出比较原理(x=0,1,2,3)*给出了输出比较的原理电路。通道输出信号 CHx_O 与 OxCPRE 信号的关系描述如下：OxCPRE 信号高电平有效，CHx_O 的输出情况与 OxCPRE 信号，CHxP 位和 CHxE 位有关（具体情况请见 TIMERx_CHCTL2 寄存器中的描述）。例如，当设置 CHxP=0（CHx_O 高电平有效，与 OxCPRE 输出极性相同）、CHxE=1（CHx_O 输出使能）时：

若OxCPRE输出有效（高）电平，则CHx_O输出有效（高）电平；

若OxCPRE输出无效（低）电平，则CHx_O输出无效（低）电平。

在通道输出比较功能，TIMERx 可以产生时控脉冲，其位置，极性，持续时间和频率都是可编程的。当一个输出通道的 CHxVAL 寄存器与计数器的值匹配时，根据 CHxCOMCTL 的配置，这个通道的输出可以被置高电平，被置低电平或者反转。当计数器的值与 CHxVAL 寄存器的值匹配时，CHxIF 位被置 1，如果 CHxIE = 1 则会产生中断，如果 CHxDEN=1 则会产生 DMA 请求。

配置步骤如下：

第一步：时钟配置：

配置定时器时钟源，预分频器等。

第二步：比较模式配置：

设置CHxCOMSEN位来配置输出比较影子寄存器；

设置CHxCOMCTL位来配置输出模式（置高电平/置低电平/反转）；

设置CHxP/CHxNP位来选择有效电平的极性；

设置CHxEN使能输出。

第三步：通过CHxIE/CHxDEN位配置中断/DMA请求使能。

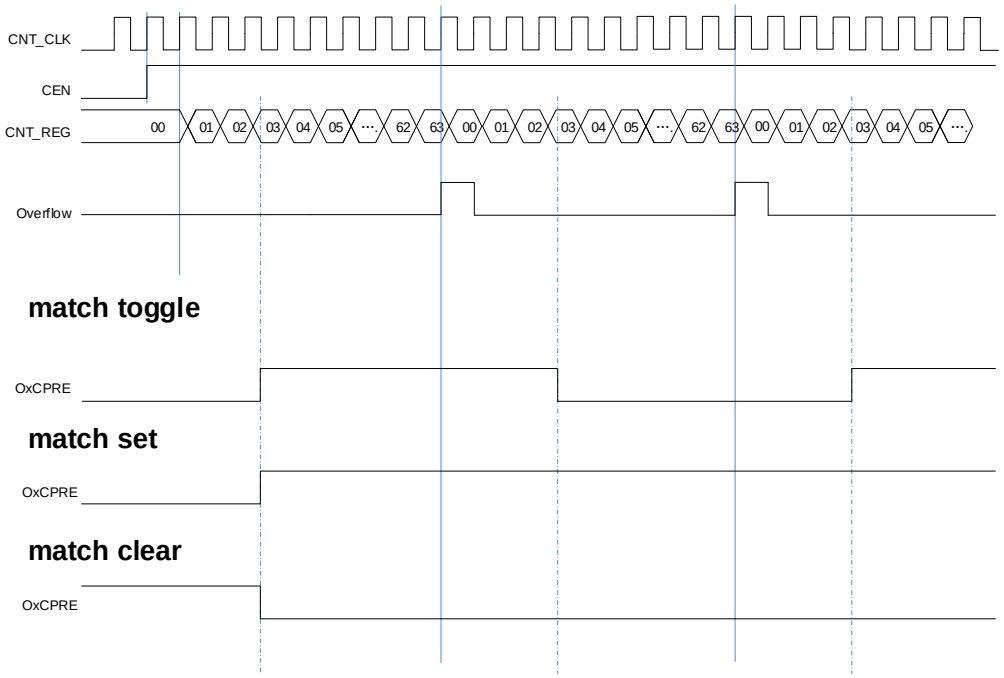
第四步：通过TIMERx_CAR寄存器和TIMERx_CHxCV寄存器配置输出比较时基：

CHxVAL可以在运行时根据你所期望的波形而改变。

第五步：设置CEN位使能定时器。

*图 15-41. 三种输出比较模式*显示了三种比较输出模式：反转/置高电平/置低电平，CAR=0x63, CHxVAL=0x3。

图 15-41. 三种输出比较模式



输出 PWM 功能

在 PWM 输出模式下（PWM 模式 0 是配置 CHxCOMCTL 为 3'b110，PWM 模式 1 是配置 CHxCOMCTL 为 3'b111），通道根据 TIMERx_CAR 寄存器和 TIMERx_CHxCV 寄存器的值，输出 PWM 波形。

根据计数模式，我们可以分为两种 PWM 波：EAPWM(边沿对齐 PWM)和 CAPWM(中央对齐 PWM)。

EAPWM 的周期由 TIMERx_CAR 寄存器值决定，占空比由 TIMERx_CHxCV 寄存器值决定。*图 15-42. EAPWM 时序图*显示了 EAPWM 的输出波形和中断。

CAPWM 的周期由（2*TIMERx_CAR 寄存器值）决定，占空比由（2*TIMERx_CHxCV 寄存器值）决定。*图 15-43. CAPWM 时序图*显示了 CAPWM 的输出波形和中断。

在 PWM0 模式下(CHxCOMCTL==3'b110)，如果 TIMERx_CHxCV 寄存器的值大于 TIMERx_CAR 寄存器的值，通道输出一直为有效电平。

在 PWM0 模式下(CHxCOMCTL==3'b110)，如果 TIMERx_CHxCV 寄存器的值等于 0，通道输出一直为无效电平。

图 15-42. EAPWM 时序图

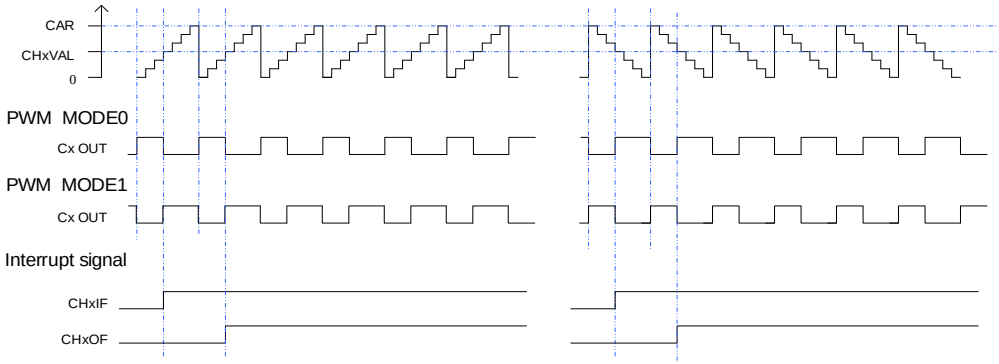
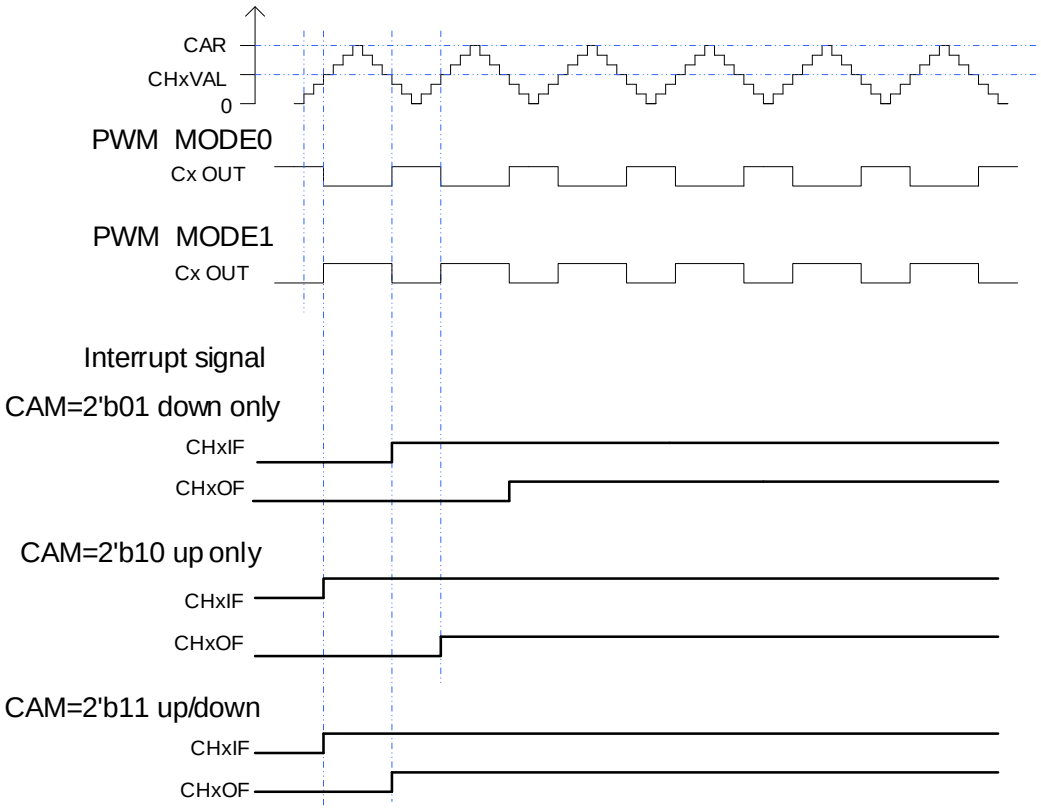


图 15-43. CAPWM 时序图



通道输出参考信号

根据*图 15-40. 通道输出比较原理（x=0，1，2，3）*所示，当 TIMERx 用于输出匹配比较模式下，设置 CHxCOMCTL 位可以定义 OxCPRE 信号(通道 x 准备信号)类型。OxCPRE 信号有若干类型的输出功能，包括，设置 CHxCOMCTL=0x00 可以保持原始电平；设置 CHxCOMCTL=0x01 可以将 OxCPRE 信号设置为高电平；设置 CHxCOMCTL=0x02 可以将 OxCPRE 信号设置为低电平；设置 CHxCOMCTL=0x03，在计数器值和 TIMERx_CHxCV 寄存器的值匹配时，可以翻转输出信号。

PWM 模式 0 和 PWM 模式 1 是 OxCPRE 的另一种输出类型，设置 CHxCOMCTL 位域位 0x06

或 0x07 可以配置 PWM 模式 0/PWM 模式 1。在这些模式中，根据计数器值和 TIMERx_CHxCV 寄存器值的关系以及计数方向，OxCPRE 信号改变其电平。具体细节描述，请参考相应的位。

设置 CHxCOMCTL =0x04 或 0x05 可以实现 OxCPRE 信号的强制输出功能。输出比较信号能够直接由软件强置为有效或无效状态，而不依赖于 TIMERx_CHxCV 的值和计数器值之间间的比较结果。

设置 CHxCOMCEN=1，当由外部 ETI 引脚信号产生的 ETIFE 信号为高电平时，OxCPRE 被强制为低电平。在下一次更新事件到来时，OxCPRE 信号才会回到有效电平状态。

正交译码器

参考*正交译码器*。

霍尔传感器接口功能

参考*霍尔传感器接口功能*。

主-从管理

参考*主-从管理*。

单脉冲模式

参考*单脉冲模式*。

定时器互连

参考*高级定时器互连（TIMERx,x=0）*

表 15-6. TIMERx(x=1,2)定时器内部互连

Slave TIMER	ITI0(TRGS = 000)	ITI1(TRGS = 001)	ITI2(TRGS = 010)	ITI3(TRGS = 011)
TIMER1	TIMER0	TIMER14	TIMER2	保留
TIMER2	TIMER0	TIMER1	TIMER14	保留

定时器 DMA 模式

定时器 DMA 模式是指通过 DMA 模块配置定时器的寄存器。有两个跟定时器 DMA 模式相关的寄存器：TIMERx_DMACFG 和 TIMERx_DMATB。当然，必须要使能 DMA 请求，一些内部中断事件可以产生 DMA 请求。当中断事件发生，TIMERx 会给 DMA 发送请求。DMA 配置成 M2P 模式，PADDR 是 TIMERx_DMATB 寄存器地址，DMA 就会访问 TIMERx_DMATB 寄存器。实际上，TIMERx_DMATB 寄存器只是一个缓冲，定时器会将 TIMERx_DMATB 映射到一个内部寄存器，这个内部寄存器由 TIMERx_DMACFG 寄存器中的 DMATA 来指定。如果 TIMERx_DMACFG 寄存器的 DMATC 位域值为 0，表示 1 次传输，定时器的发送 1 个 DMA 请求就可以完成。如果 TIMERx_DMACFG 寄存器的 DMATC 位域值不为 1，例如其值为 3，表示 4 次传输，定时器就需要再多发 3 次 DMA 请求。在这 3 次请求下，DMA 对 TIMERx_DMATB 寄存器的访问会映射到访问定时器的 DMATA+0x4, DMATA+0x8, DMATA+0xc 寄存器。总之，

发生一次 DMA 内部中断请求，定时器会连续发送（DMATC+1）次请求。

如果再来 1 次 DMA 请求事件，TIMERx 将会重复上面的过程。

定时器调试模式

当 Cortex®-M3 内核停止，DBG_CTL0 寄存器中的 TIMERx_HOLD 配置位被置 1，定时器计数器停止。

15.2.5. TIMERx 寄存器（x=1,2）

TIMER1 基地址：0x4000 0000
TIMER2 基地址：0x4000 0400

控制寄存器 0 (TIMERx_CTL0)

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						CKDIV[1:0]		ARSE	CAM[1:0]		DIR	SPM	UPS	UPDIS	CEN
						rw		rw	rw		rw	rw	rw	rw	rw

位/位域	名称	描述
31:10	保留	必须保持复位值。
9:8	CKDIV[1:0]	时钟分频 通过软件配置CKDIV，规定定时器时钟(CK_TIMER) 与死区时间和数字滤波器采样时钟(DTS)之间的分频系数。 00：$f_{DTS}=f_{CK_TIMER}$ 01：$f_{DTS}= f_{CK_TIMER}$ /2 10：$f_{DTS}= f_{CK_TIMER}$ /4 11：保留
7	ARSE	自动重载影子使能 0：禁能TIMERx_CAR寄存器的影子寄存器 1：使能 TIMERx_CAR 寄存器的影子寄存器
6:5	CAM[1:0]	计数器对齐模式选择 00：无中央对齐计数模式(边沿对齐模式)。 DIR位指定了计数方向 01：中央对齐向下计数置1模式。计数器在中央计数模式计数，通道被配置在输出模式（TIMERx_CHCTL0寄存器中CHxMS=00），只有在向下计数时，CHxF位置1 10：中央对齐向上计数置1模式。计数器在中央计数模式计数，通道被配置在输出模式（TIMERx_CHCTL0寄存器中CHxMS=00），只有在向上计数时，CHxF位置1 11：中央对齐上下计数置1模式。计数器在中央计数模式计数，通道被配置在输出模式（TIMERx_CHCTL0寄存器中CHxMS=00），在向上和向下计数时，CHxF位都会置1 当计数器使能以后，该位不能从 0x00 切换到非 0x00
4	DIR	方向 0：向上计数

1：向下计数

当计数器配置为中央对齐计数模式或正交译码器模式时，该位只读。

3	SPM	单脉冲模式 0：单脉冲模式禁能。更新事件发生后，计数器继续计数 1：单脉冲模式使能。在下一次更新事件发生时，计数器停止计数
2	UPS	更新请求源 软件配置该位，选择更新事件源. 0：以下事件均会产生更新中断或DMA请求： UPG位被置1 计数器溢出/下溢 复位模式产生的更新 1：下列事件会产生更新中断或DMA请求： 计数器溢出/下溢
1	UPDIS	禁止更新. 该位用来使能或禁能更新事件的产生 0：更新事件使能. 更新事件发生时，相应的影子寄存器被装入预装载值，以下事件均会产生更新事件： UPG位被置1 计数器溢出/下溢 复位模式产生的更新 1：更新事件禁能. 注意：当该位被置 1 时，UPG 位被置 1 或者复位模式不会产生更新事件，但是计数器和预分频器被重新初始化
0	CEN	计数器使能 0：计数器禁能 1：计数器使能 在软件将CEN位置1后，外部时钟、暂停模式和正交译码器模式才能工作。

控制寄存器 1 (TIMERx_CTL1)

地址偏移：0x04

复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								TI0S	MMC[2:0]			DMAS	保留.		
								rw	rw			rw			

位/位域	名称	描述

31:8	保留	必须保持复位值。
7	TI0S	通道0触发输入选择 0：选择 TIMERx_CH0 引脚作为通道 0 的触发输入 1：选择 TIMERx_CH0, CH1 和 CH2 引脚异或的结果作为通道 0 的触发输入
6:4	MMC[2:0]	主模式控制 这些位控制TRGO信号的选择，TRGO信号由主定时器发给从定时器用于同步功能 000：当产生一个定时器复位事件后，输出一个TRGO信号，定时器复位源为： 主定时器产生一个复位事件 TIMERx_SWEVG寄存器中UPG位置1 001：当产生一个定时器使能事件后，输出一个TRGO信号，定时器使能源为： CEN位置1 在暂停模式下，触发输入置1 010：当产生一个定时器更新事件后，输出一个TRGO信号，更新事件源由UPDIS和UPS位决定 011：当通道0在发生一次捕获或一次比较成功时，主模式控制器产生一个TRGO脉冲 100：当产生一次比较事件时，输出一个TRGO信号，比较事件源来自O0CPRE 101：当产生一次比较事件时，输出一个TRGO信号，比较事件源来自O1CPRE 110：当产生一次比较事件时，输出一个TRGO信号，比较事件源来自O2CPRE 111：当产生一次比较事件时，输出一个 TRGO 信号，比较事件源来自 O3CPRE
3	DMAS	DMA 请求源选择 0：当通道捕获/比较事件发生时，发送通道 x 的 DMA 请求 . 1：当更新事件发生，发送通道 x 的 DMA 请求
2:0	保留	必须保持复位值。

从模式配置寄存器 (TIMERx_SMCFG)

地址偏移：0x08

复位值: 0x0000 0000

该寄存器通过字访问（32位）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	SMC1	ETPSC[1:0]		ETFC[3:0]				MSM	TRGS[2:0]			OCRC	SMC[2:0]		
rw	rw	rw		rw				rw	rw			rw	rw		

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	ETP	外部触发极性 该位指定 ETI 信号的极性。 0：ETI 高电平或上升沿有效。

1：ETI 低电平或下降沿有效。

14 SMC1

SMC 的一部分为了使能外部时钟模式 1

在外部时钟模式 1，计数器由 ETIF 信号上的任意有效边沿驱动。

0：外部时钟模式 1 禁能。

1：外部时钟模式 1 使能。

当从模式配置为复位模式，暂停模式和事件模式时，定时器仍然可以工作在外部时钟模式 1。但是 TRGS 必须不能为 3'b111。

如果外部时钟模式 0 和外部时钟模式 1 同时被被配置，外部时钟的输入是 ETIF

注意：外部时钟模式 0 使能在寄存器的 SMC[2:0]位域。

13:12 ETPSC[1:0]

外部触发预分频

外部触发信号 ETI 的频率不能超过 TIMER_CK 频率的 1/4。当输入较快的外部时钟时，可以使用预分频降低 ETIP 的频率。

00：预分频禁能。

01：2 分频。

10：4 分频。

11：8 分频。

11:8 ETFC[3:0]

外部触发滤波控制

外部触发信号可以通过数字滤波器进行滤波，该位域定义了数字滤波器的滤波能力。数字滤波器的基本原理是：以 f_{SAMP} 频率连续采样外部触发信号，同时记录采样相同电平的次数。当该次数达到配置的滤波能力时，则认为是一个有效的电平信号。

EXTFC[3:0]	次数	f_{SAMP}
4'b0000	Filter disabled.	
4'b0001	2	f_{CK_TIMER}
4'b0010	4	
4'b0011	8	
4'b0100	6	$f_{DTS_CK}/2$
4'b0101	8	
4'b0110	6	$f_{DTS_CK}/4$
4'b0111	8	
4'b1000	6	$f_{DTS_CK}/8$
4'b1001	8	
4'b1010	5	$f_{DTS_CK}/16$
4'b1011	6	
4'b1100	8	
4'b1101	5	$f_{DTS_CK}/32$
4'b1110	6	
4'b1111	8	

7 MSM

主-从模式

该位被用来同步被选择的定时器同时开始计数。通过 TRIGI 和 TRGO，定时器被连接在一起，TRGO 用做启动事件。

0：主从模式禁能。

1：主从模式使能。

6:4	TRGS[2:0]	触发选择 该位域用来指定选择哪一个信号作为用来同步计数器的触发输入源。 000: ITI0 001: ITI1 010: ITI2 011: ITI3 100: CI0F_ED 101: CI0FE0 110: CI1FE1 111: ETIFP 从模式被使能后这些位不能改。
3	OCRC	OCPRE 清除源选择 0: OCPRE_CLR_INT 连接到 OCPRE_CLR 输入。 1: OCPRE_CLR_INT 连接到 ETIF。
2:0	SMC[2:0]	从模式控制 000：关闭从模式。如果 CEN=1，则预分频器直接由内部时钟驱动。 001：正交译码器模式 0。根据 CI0FE0 的电平，计数器在 CI1FE1 的边沿向上/下计数。 010：正交译码器模式 1。根据 CI1FE1 的电平，计数器在 CI0FE0 的边沿向上/下计数。 011：正交译码器模式 2。根据另一个信号的输入电平，计数器在 CI0FE0 和 CI1FE1 的边沿向上/下计数。 100：复位模式。选中的触发输入的上升沿重新初始化计数器，并且产生更新事件。 101：暂停模式。当触发输入为高时，计数器的时钟开启。一旦触发输入变为低，则计数器时钟停止。 110：事件模式。计数器在触发输入的上升沿启动。 111：外部时钟模式 0。选中的触发输入的上升沿驱动计数器。 由于 CI0F_ED 是一个脉冲波形，而暂停模式是检测触发信号的电平，所以，当 CI0F_ED 用作触发输入时，暂停模式必须禁能。

DMA 和中断使能寄存器 (TIMERx_DMAINTEN)

地址偏移：0x0C
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TRGDEN	保留	CH3DEN	CH2DEN	CH1DEN	CH0DEN	UPDEN	保留	TRGIE	保留	CH3IE	CH2IE	CH1IE	CH0IE	UPIE
	rw		rw	rw	rw	rw	rw		rw		rw	rw	rw	rw	rw

位/位域	名称	描述
31:15	保留	必须保持复位值。
14	TRGDEN	触发 DMA 请求使能 0：禁止触发 DMA 请求 1：使能触发 DMA 请求
13	保留	必须保持复位值。
12	CH3DEN	通道 3 比较/捕获 DMA 请求使能 0：禁止通道 3 比较/捕获 DMA 请求 1：使能通道 3 比较/捕获 DMA 请求
11	CH2DEN	通道 2 比较/捕获 DMA 请求使能 0：禁止通道 2 比较/捕获 DMA 请求 1：使能通道 2 比较/捕获 DMA 请求
10	CH1DEN	通道 1 比较/捕获 DMA 请求使能 0：禁止通道 1 比较/捕获 DMA 请求 1：使能通道 1 比较/捕获 DMA 请求
9	CH0DEN	通道 0 比较/捕获 DMA 请求使能 0：禁止通道 0 比较/捕获 DMA 请求 1：使能通道 0 比较/捕获 DMA 请求
8	UPDEN	更新 DMA 请求使能 0：禁止更新 DMA 请求 1：使能更新 DMA 请求
7	保留	必须保持复位值。
6	TRGIE	触发中断使能 0：禁止触发中断 1：使能触发中断
5	保留	必须保持复位值。
4	CH3IE	通道 3 比较/捕获中断使能 0：禁止通道 3 中断 1：使能通道 3 中断
3	CH2IE	通道 2 比较/捕获中断使能 0：禁止通道 2 中断 1：使能通道 2 中断
2	CH1IE	通道 1 比较/捕获中断使能 0：禁止通道 1 中断 1：使能通道 1 中断
1	CH0IE	通道 0 比较/捕获中断使能 0：禁止通道 0 中断

1：使能通道 0 中断

0	UPIE	更新中断使能 0：禁止更新中断 1：使能更新中断

中断标志寄存器 (TIMERx_INTF)

地址偏移：0x10
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留			CH3OF	CH2OF	CH1OF	CH0OF	保留		TRGIF	保留	CH3IF	CH2IF	CH1IF	CH0IF	UPIF
			rc_w0	rc_w0	rc_w0	rc_w0	.		rc_w0		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

位/位域	名称	描述
31:13	保留	必须保持复位值。
12	CH3OF	通道 3 捕获溢出标志 参见 CH0OF 描述
11	CH2OF	通道 2 捕获溢出标志 参见 CH0OF 描述
10	CH1OF	通道 1 捕获溢出标志 参见 CH0OF 描述
9	CH0OF	通道 1 捕获溢出标志 当通道 0 被配置为输入模式时，在 CH0IF 标志位已经被置 1 后，捕获事件再次发生时， 该标志位可以由硬件置 1。该标志位由软件清 0. 0：无捕获溢出中断发生 1：发生了捕获溢出中断
8:7	保留	必须保持复位值。
6	TRGIF	触发中断标志 当发生触发事件时，此标志会置 1，此位由软件清 0。当暂停模式使能时，触发输入的任意边沿都可以产生触发事件。否则，其它模式时，仅在触发输入端检测到有效边沿，产生触发事件。 0：无触发事件产生 1：触发中断产生

5	保留	必须保持复位值。
4	CH3IF	通道 3 比较/捕获中断标志 参见 CH0IF 描述
3	CH2IF	通道 2 比较/捕获中断标志 参见 CH0IF 描述
2	CH1IF	通道 1 比较/捕获中断标志 参见 CH0IF 描述
1	CH0IF	通道 0 比较/捕获中断标志 此标志由硬件置 1 软件清 0。当通道 0 在输入模式下时，捕获事件发生时此标志位被置 1；当通道 0 在输出模式下时，此标志位在一个比较事件发生时被置 1。 当通道 0 在输入模式下时，读 TIMERx_CH0CV 会将此标志清零。 0：无通道 0 中断发生 1：通道 0 中断发生
0	UPIF	更新中断标志 此位在任何更新事件发生时由硬件置 1，软件清 0。 0：无更新中断发生 1：发生更新中断

软件事件产生寄存器 (TIMERx_SWEVG)

地址偏移：0x14
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									TRGG	保留.	CH3G	CH2G	CH1G	CH0G	UPG
									w		w	w	w	w	w

位/位域	名称	描述
31:7	保留	必须保持复位值。
6	TRGG	触发事件产生 此位由软件置 1，由硬件自动清 0. 当此位被置 1，TIMERx_INTF 寄存器的 TRGIF 标志 位被置 1，若开启对应的中断和 DMA，则产生相应的中断和 DMA 传输。 0：无触发事件产生 1：产生触发事件
5	保留	必须保持复位值。

4	CH3G	通道 3 捕获或比较事件发生 参见 CH0G 描述
3	CH2G	通道 2 捕获或比较事件发生 参见 CH0G 描述
2	CH1G	通道 1 捕获或比较事件发生 参见 CH0G 描述
1	CH0G	通道 0 捕获或比较事件发生 该位由软件置 1，用于在通道 0 产生一个捕获/比较事件，由硬件自动清 0。当此位被置 1，CH0IF 标志位被置 1，若开启对应的中断和 DMA，则发出相应的中断和 DMA 请求。此外，如果通道 0 配置为输入模式，计数器的当前值被 TIMERx_CH0CV 寄存器捕获，如果 CH0IF 标志位已经为 1，则 CH0OF 标志位被置 1。 0：不产生通道 0 捕获或比较事件 1：发生通道 0 捕获或比较事件
0	UPG	更新事件产生 此位由软件置 1，被硬件自动清 0。当此位被置 1，如果选择了中央对齐或向上计数模式，计数器被清 0。否则(向下计数模式)计数器将载入自动重载值，预分频计数器将同时被清除。 0：无更新事件产生 1：产生更新事件

通道控制寄存器 0 (TIMERx_CHCTL0)

地址偏移：0x18
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1COMCEN	CH1COMCTL[2:0]			CH1COMSEN	CH1COMFEN	CH1MS[1:0]		CH0COMCEN	CH0COMCTL[2:0]			CH0COMSEN	CH0COMFEN	CH0MS[1:0]	
CH1CAPFLT[3:0]				CH1CAPPSC[1:0]				CH0CAPFLT[3:0]				CH0CAPPSC[1:0]			
rw				rw		rw		rw				rw		rw	

输出比较模式：

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	CH1COMCEN	通道 1 输出比较清 0 使能 参见 CH0COMCEN 描述
14:12	CH1COMCTL[2:0]	通道 1 输出比较模式

参见 CH0COMCTL 描述

11	CH1COMSEN	通道 1 输出比较影子寄存器使能 参见 CH0COMSEN 描述
10	CH1COMFEN	通道 1 输出比较快速使能 参见 CH0COMFEN 描述
9:8	CH1MS[1:0]	通道 1 模式选择 这些位定义了通道的方向和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2 寄存器的 CH1EN 位被清 0)时这些位才可以写。 00：通道 1 配置为输出 01：通道 1 配置为输入，IS1 映射在 CI1FE1 上 10：通道 1 配置为输入，IS1 映射在 CI0FE1 上 11：通道 1 配置为输入，IS1 映射在 ITS 上 注意：当 CH1MS[1:0]=11 时，需要通过 TRGS 位（位于 TIMERx_SMCFG 寄存器）选择内部触发输入。
7	CH0COMCEN	通道 0 输出比较清 0 使能 当此位被置 1，当检测到 ETIFP 信号输入高电平时，O0CPRE 参考信号被清 0 0：禁止通道 0 输出比较清零 1：使能通道 0 输出比较清零
6:4	CH0COMCTL[2:0]	通道 0 输出比较模式 此位定义了输出准备信号 O0CPRE 的输出比较模式，而 O0CPRE 决定了 CH0_O、CH0_ON 的值。另外，O0CPRE 高电平有效，而 CH0_O、CH0_ON 通道的极性取决于 CH0P、CH0NP 位。 000：时基。输出比较寄存器 TIMERx_CH0CV 与计数器 TIMERx_CNT 间的比较对 O0CPRE 不起作用 001：匹配时设置为高。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 为高。 010：匹配时设置为低。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 为低。 011：匹配时翻转。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 翻转。 100：强制为低。强制 O0CPRE 为低电平 101：强制为高。强制 O0CPRE 为高电平 110：PWM 模式 0。在向上计数时，一旦计数器值小于 TIMERx_CH0CV 时，O0CPRE 为高电平，否则为低电平。在向下计数时，一旦计数器的值大于 TIMERx_CH0CV 时，O0CPRE 为低电平，否则为高电平。 111：PWM 模式 1。在向上计数时，一旦计数器值小于 TIMERx_CH0CV 时，O0CPRE 为低电平，否则为高电平。在向下计数时，一旦计数器的值大于 TIMERx_CH0CV 时，O0CPRE 为高电平，否则为低电平。 如果配置在 PWM 模式下，只有当输出比较模式从时基模式变为 PWM 模式或者比较结果改变时，O0CPRE 电平才改变。
3	CH0COMSEN	通道 0 输出比较影子寄存器使能

当此位被置 1，TIMERx_CH0CV 寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。
0：禁止通道 0 输出/比较影子寄存器
1：使能通道 0 输出/比较影子寄存器
仅在单脉冲模式下(SPM =1)，可以在未确认影子寄存器的情况下使用 PWM 模式

2 CH0COMFEN 通道 0 输出比较快速使能
当该位为 1 时，如果通道配置为 PWM0 模式或者 PWM1 模式，会加快捕获/比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，CH0_O 被设置为比较电平而与比较结果无关。
0：禁止通道 0 输出比较快速.
1：使能通道 0 输出比较快速。

1:0 CH0MS[1:0] 通道 0 I/O 模式选择
这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2 寄存器的 CH0EN 位被清 0)时这些位才可写。
00：通道 0 配置为输出
01：通道 0 配置为输入，IS0 映射在 CI0FE0 上
10：通道 0 配置为输入，IS0 映射在 CI1FE0 上
11：通道 0 配置为输入，IS0 映射在 ITS 上
注意：当 CH0MS[1:0]=11 时，需要通过 TRGS 位（位于 TIMERx_SMCFG 寄存器）选择内部触发输入。

输入捕获模式：

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:12	CH1CAPFLT[3:0]	通道 1 输入捕获滤波控制 参见 CH0CAPFLT 描述
11:10	CH1CAPPSC[1:0]	通道 1 输入捕获预分频器 参见 CH0CAPPSC 描述
9:8	CH1MS[1:0]	通道 1 模式选择 与输出模式相同
7:4	CH0CAPFLT[3:0]	通道 0 输入捕获滤波控制 CI0 输入信号可以通过数字滤波器进行滤波，该位域配置滤波参数。 数字滤波器的基本原理：根据 f_{SAMP} 对 CI0 输入信号进行连续采样，并记录信号相同电平的次数。达到该位配置的滤波参数后，认为是有效电平。 滤波器参数配置如下：

CH0CAPFLT [3:0]	采样次数	f_{SAMP}
4’b0000	无滤波器	
4’b0001	2	f_{CK_TIMER}
4’b0010	4	
4’b0011	8	

4’b0100	6	f_{DTS}/2
4’b0101	8	
4’b0110	6	f_{DTS}/4
4’b0111	8	
4’b1000	6	f_{DTS}/8
4’b1001	8	
4’b1010	5	f_{DTS}/16
4’b1011	6	
4’b1100	8	
4’b1101	5	f_{DTS}/32
4’b1110	6	
4’b1111	8	

3:2 CH0CAPPSC[1:0] 通道 0 输入捕获预分频器

这 2 位定义了通道 0 输入的预分频系数。当 TIMERx_CHCTL2 寄存器中的 CH0EN =0 时，则预分频器复位。

00：无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获

01：每 2 个事件触发一次捕获

10：每 4 个事件触发一次捕获

11：每 8 个事件触发一次捕获

1:0 CH0MS[1:0] 通道 0 模式选择

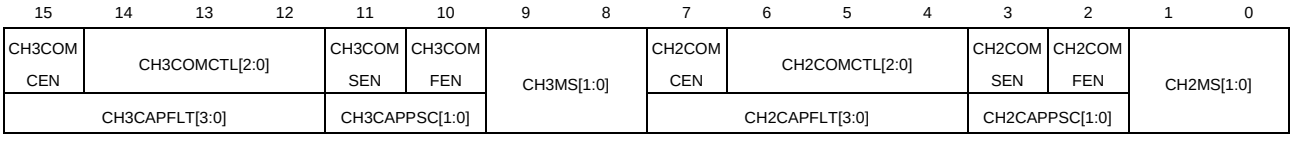
与输出比较模式相同

通道控制寄存器 1 (TIMERx_CHCTL1)

地址偏移：0x1C

复位值：0x0000 0000

该寄存器只能按字(32位)访问。



输出比较模式：

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	CH3COMCEN	通道 3 输出比较清 0 使能 参见 CH0COMCEN 描述
14:12	CH3COMCTL[2:0]	通道 3 输出比较模式

		参见 CH0COMCTL 描述
11	CH3COMSEN	通道 3 输出比较影子寄存器使能 参见 CH0COMSEN 描述
10	CH3COMFEN	通道 3 输出比较快速使能 参见 CH0COMFEN 描述
9:8	CH3MS[1:0]	通道 3 模式选择 这些位定义了通道的方向和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2 寄存器的 CH3EN 位被清 0)时这些位才可以写。 00：通道 3 配置为输出 01：通道 3 配置为输入，IS3 映射在 CI3FE3 上 10：通道 3 配置为输入，IS3 映射在 CI2FE3 上 11：通道 3 配置为输入，IS3 映射在 ITS 上 注意：当 CH3MS[1:0]=11 时，需要通过 TRGS 位（位于 TIMERx_SMCFG 寄存器）选择内部触发输入。
7	CH2COMCEN	通道 2 输出比较清 0 使能 当此位被置 1，当检测到 ETIFP 输入高电平时，O2CPRE 参考信号被清 0 0：使能通道 2 输出比较清零 1：禁止通道 2 输出比较清零
6:4	CH2COMCTL[2:0]	通道 2 输出比较模式 此位定义了输出准备信号 O2CPRE 的输出比较模式，而 O2CPRE 决定了 CH2_O、CH2_ON 的值。另外，O2CPRE 高电平有效，而 CH2_O、CH2_ON 通道的极性取决于 CH2P、CH2NP 位。 000：时基。输出比较寄存器 TIMERx_CH2CV 与计数器 TIMERx_CNT 间的比较对 O2CPRE 不起作用 001：匹配时设置为高。当计数器的值与捕获/比较值寄存器 TIMERx_CH2CV 相同时，强制 O2CPRE 为高。 010：匹配时设置为低。当计数器的值与捕获/比较值寄存器 TIMERx_CH2CV 相同时，强制 O2CPRE 为低。 011：匹配时翻转。当计数器的值与捕获/比较值寄存器 TIMERx_CH2CV 相同时，强制 O2CPRE 翻转。 100：强制为低。强制 O2CPRE 为低电平 101：强制为高。强制 O2CPRE 为高电平 110：PWM 模式 0。在向上计数时，一旦计数器值小于 TIMERx_CH2CV 时，O2CPRE 为高电平，否则为低电平。在向下计数时，一旦计数器的值大于 TIMERx_CH2CV 时，O2CPRE 为低电平，否则为高电平。 111：PWM 模式 1。在向上计数时，一旦计数器值小于 TIMERx_CH2CV 时，O2CPRE 为低电平，否则为高电平。在向下计数时，一旦计数器的值大于 TIMERx_CH2CV 时，O2CPRE 为高电平，否则为低电平。 如果配置在 PWM 模式下，只有当输出比较模式从时基模式变为 PWM 模式或者比较结果改变时，O2CPRE 电平才改变。
3	CH2COMSEN	通道 0 输出比较影子寄存器使能

当此位被置 1，TIMERx_CH2CV 寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。
0：禁止通道 2 输出/比较影子寄存器
1：使能通道 2 输出/比较影子寄存器
仅在单脉冲模式下(SPM =1)，可以在未确认影子寄存器情况下使用 PWM 模式

2	CH2COMFEN	通道 2 输出比较快速使能 当该位为 1 时，如果通道配置为 PWM0 模式或者 PWM1 模式，会加快捕获/比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，CH2_O 被设置为比较电平而与比较结果无关。 0：禁止通道 2 输出比较快速. 1：使能通道 2 输出比较快速。
1:0	CH2MS[1:0]	通道 2 I/O 模式选择 这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2 寄存器的 CH2EN 位被清 0)时这些位才可写。 00：通道 2 配置为输出 01：通道 2 配置为输入，IS2 映射在 CI2FE2 上 10：通道 2 配置为输入，IS2 映射在 CI3FE2 上 11：通道 2 配置为输入，IS2 映射在 ITS 上. 注意：当 CH2MS[1:0]=11 时，需要通过 TRGS 位（位于 TIMERx_SMCFG 寄存器）选择内部触发输入。

输入捕获模式：

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:12	CH3CAPFLT[3:0]	通道 3 输入捕获滤波控制 参见 CH0CAPFLT 描述
11:10	CH3CAPPSC[1:0]	通道 3 输入捕获预分频器 参见 CH0CAPPSC 描述
9:8	CH3MS[1:0]	通道 3 模式选择 与输出模式相同
7:4	CH2CAPFLT[3:0]	通道 2 输入捕获滤波控制 CI2 输入信号可以通过数字滤波器进行滤波，该位域配置滤波参数。 数字滤波器的基本原理：根据 f_{SAMP} 对 CI2 输入信号进行连续采样，并记录信号相同电平的次数。达到该位配置的滤波参数后，认为是有效电平。 滤波器参数配置如下：

CH2CAPFLT [3:0]	采样次数	f_{SAMP}
4’b0000	无滤波器	
4’b0001	2	f_{CK_TIMER}
4’b0010	4	
4’b0011	8	

4’b0100	6	f_{DTS}/2
4’b0101	8	
4’b0110	6	f_{DTS}/4
4’b0111	8	
4’b1000	6	f_{DTS}/8
4’b1001	8	
4’b1010	5	f_{DTS}/16
4’b1011	6	
4’b1100	8	
4’b1101	5	f_{DTS}/32
4’b1110	6	
4’b1111	8	

3:2 CH2CAPPSC[1:0] 通道 2 输入捕获预分频器

这 2 位定义了通道 2 输入的预分频系数。当 TIMERx_CHCTL2 寄存器中的 CH2EN =0 时，则预分频器复位。

00：无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获

01：每 2 个事件触发一次捕获

10：每 4 个事件触发一次捕获

11：每 8 个事件触发一次捕获

1:0 CH2MS[1:0] 通道 2 模式选择

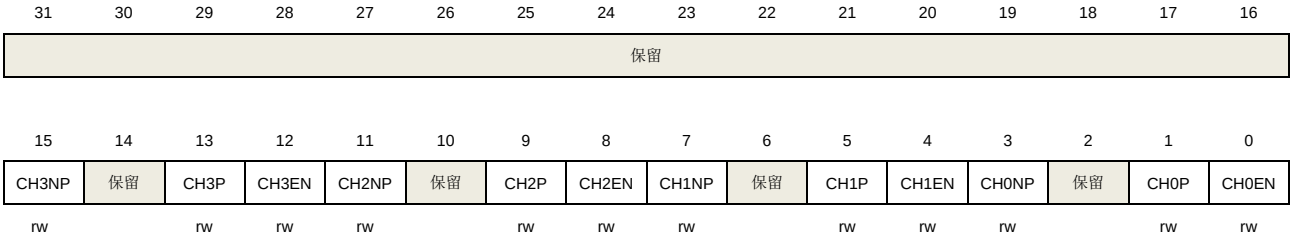
与输出比较模式相同

通道控制寄存器 2 (TIMERx_CHCTL2)

地址偏移：0x20

复位值：0x0000 0000

该寄存器只能按字(32位)访问。



位/位域	名称	描述
31:16	保留	必须保持复位值。
15	CH3NP	通道 3 互补输出极性 参考 CH0NP 描述
14	保留	必须保持复位值。
13	CH3P	通道 3 极性

		参考 CH0P 描述
12	CH3EN	通道 3 使能 参考 CH0EN 描述
11	CH2NP	通道 2 互补输出极性 参考 CH0NP 描述
10	保留	必须保持复位值。
9	CH2P	通道 2 极性 参考 CH0P 描述
8	CH2EN	通道 2 使能 参考 CH0EN 描述
7	CH1NP	通道 1 互补输出极性 参考 CH0NP 描述
6	保留	必须保持复位值。
5	CH1P	通道 1 极性 参考 CH0P 描述
4	CH1EN	通道 1 使能 参考 CH0EN 描述
3	CH0NP	通道 0 互补输出极性 当通道 0 配置为输出模式，该位保持 0。 当通道 0 配置为输入模式时，此位和 CH0P 联合使用，作为输入信号 CI0 的极性选择控制信号。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 或 10 时此位不能被更改。
2	保留	必须保持复位值。
1	CH0P	通道 0 极性 当通道 0 配置为输出模式时，此位定义了输出信号极性。 0：通道0高电平为有效电平 1：通道0低电平为有效电平 当通道 0 配置为输入模式时，此位定义了 CI0 信号极性 [CH0NP, CH0P] 将选择 CI0FE0 或者 CI1FE0 的有效边沿或者捕获极性 [CH0NP==0, CH0P==0]：把 CIxFE0 的上升沿作为捕获或者从模式下触发的有效信号，并且 CIxFE0 不会被翻转。 [CH0NP==0, CH0P==1]：把 CIxFE0 的下降沿作为捕获或者从模式下触发的有效信号，并且 CIxFE0 会被翻转。 [CH0NP==1, CH0P==0]：保留。 [CH0NP==1, CH0P==1]：把 CIxFE0 的上升沿和下降沿都作为捕获或者从模式下触发的有效信号，并且 CIxFE0 不会被翻转。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 或 10 时此位不能被更改。
0	CH0EN	通道 0 捕获/比较使能

当通道 0 配置为输出模式时，将此位置 1 使能 CH0_O 信号有效。当通道 0 配置为输入
模式时，将此位置 1 使能通道 0 上的捕获事件。
0：禁止通道 0
1：使能通道 0

计数器寄存器 (TIMERx_CNT)(x=1)

地址偏移：0x24
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

位/位域	名称	描述
31:0	CNT[31:0]	这些位是当前的计数值。写操作能改变计数器值。

计数器寄存器 (TIMERx_CNT)(x=2)

地址偏移：0x24
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CNT[15:0]	这些位是当前的计数值。写操作能改变计数器值。

预分频寄存器 (TIMERx_PSC)

地址偏移： 0x28
复位值： 0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	PSC[15:0]	计数器时钟预分频值 计数器时钟等于 TIMER_CK 时钟除以(PSC+1)，每次当更新事件产生时，PSC 的值被装入到对应的影子寄存器。

计数器自动重载寄存器 (TIMERx_CAR)(x=1)

地址偏移：0x2C
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CARL[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CARL[15:0]															
rw															

位/位域	名称	描述
31:0	CARL[31:0]	计数器自动重载值 这些位定义了计数器的自动重载值。

计数器自动重载寄存器 (TIMERx_CAR)(x=2)

地址偏移：0x2C
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CARL[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CARL[15:0]	计数器自动重载值 这些位定义了计数器的自动重载值。

通道 0 捕获/比较值寄存器 (TIMERx_CH0CV)（x=1）

地址偏移：0x34
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH0VAL[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0VAL[15:0]															
rw															

位/位域	名称	描述
31:0	CH0VAL[31:0]	通道 0 的捕获或比较值 当通道 0 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 0 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

通道 0 捕获/比较值寄存器 (TIMERx_CH0CV)(x=2)

地址偏移：0x34
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0VAL[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CH0VAL[15:0]	通道 0 的捕获或比较值 当通道 0 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器

为只读。
当通道 0 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

通道 1 捕获/比较值寄存器 (TIMERx_CH1CV)（x=1）

地址偏移：0x38
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH1VAL[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1VAL[15:0]															
rw															

位/位域	名称	描述
31:0	CH1VAL[31:0]	通道 1 的捕获或比较值 当通道 1 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 1 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

通道 1 捕获/比较值寄存器 (TIMERx_CH1CV)(x=2)

地址偏移：0x38
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1VAL[15:0]															
rw															

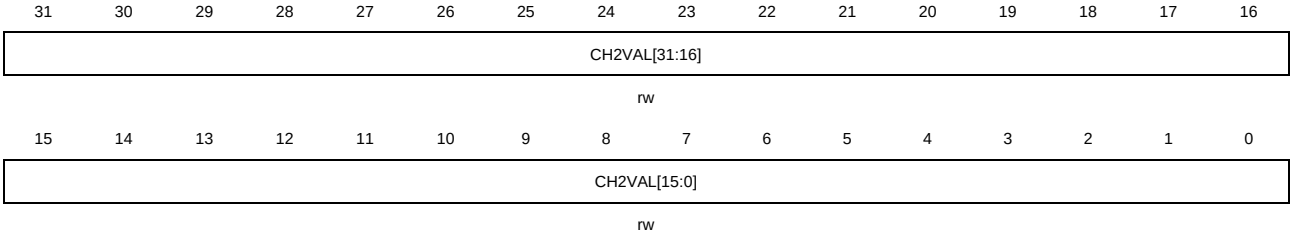
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CH1VAL[15:0]	通道 1 的捕获或比较值 当通道 1 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 1 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄

存器后，影子寄存器值随每次更新事件更新。

通道 2 捕获/比较值寄存器 (TIMERx_CH2CV)（x=1）

地址偏移：0x3C
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

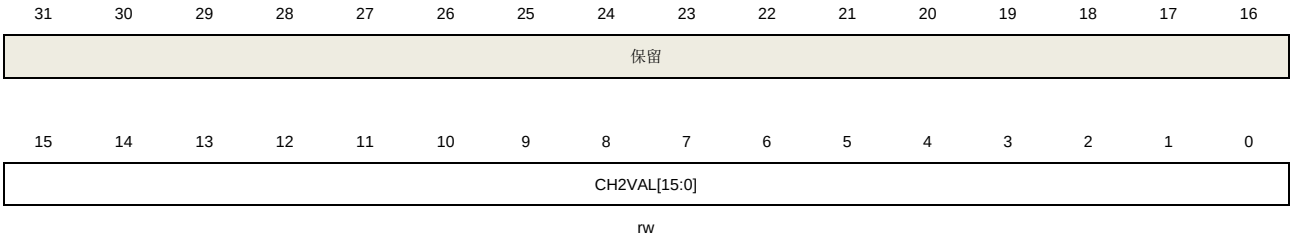


位/位域	名称	描述
31:0	CH2VAL[31:0]	通道 2 的捕获或比较值 当通道 2 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 2 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

通道 2 捕获/比较值寄存器 (TIMERx_CH2CV)(x=2)

地址偏移：0x3C
复位值：0x0000 0000

该寄存器只能按字(32位)访问。



位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CH2VAL[15:0]	通道 2 的捕获或比较值 当通道 2 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 2 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

通道 3 捕获/比较值寄存器 (TIMERx_CH3CV)（x=1）

地址偏移：0x40
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH3VAL[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3VAL[15:0]															
rw															

位/位域	名称	描述
31:0	CH3VAL[31:0]	通道 3 的捕获或比较值 当通道 3 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 3 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

通道 3 捕获/比较值寄存器 (TIMERx_CH3CV)(x=2)

地址偏移：0x40
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

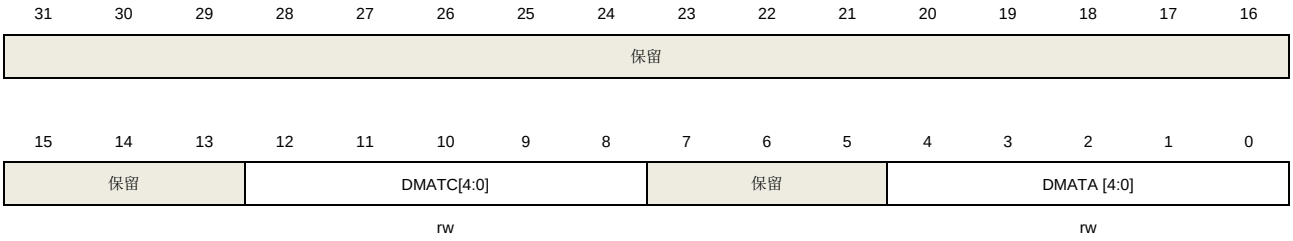
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3VAL[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CH3VAL[15:0]	通道 3 的捕获或比较值 当通道 3 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 3 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

DMA 配置寄存器 (TIMERx_DMACFG)

地址偏移：0x48
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

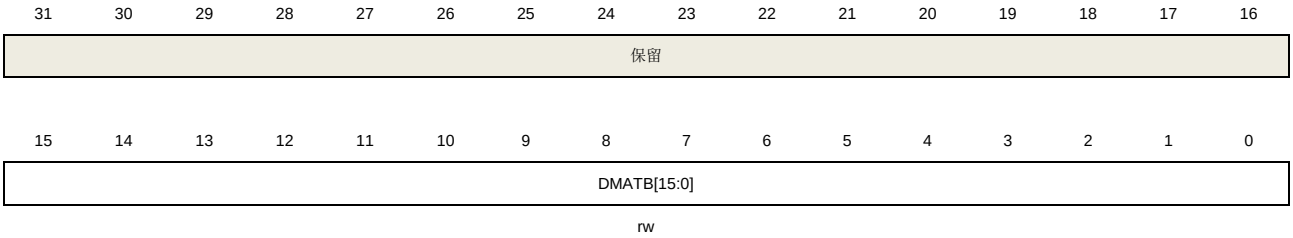


位/位域	名称	描述
31:13	保留	必须保持复位值。.
12:8	DMATC [4:0]	DMA 传输计数 该位域定义了 DMA 访问（读写）TIMERx_DMATB 寄存器的数量 n，n = (DMATC [4:0] +1). DMATC [4:0] 从 5'b0_0000 到 5'b1_0001
7:5	保留	必须保持复位值。
4:0	DMATA [4:0]	DMA 传输起始地址 该位域定义了 DMA 访问 TIMERx_DMATB 寄存器的第一个地址。当通过 TIMERx_DMA 第一次访问时，访问的就是该位域指定的地址。第二次访问 TIMERx_DMATB 时，将访问起始地址+0x4。

DMA 发送缓冲区寄存器 (TIMERx_DMATB)

地址偏移：0x4C
复位值：0x0000 0000

该寄存器只能按字(32位)访问。



位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	DMATB [15:0]	DMA 发送缓冲 对这个寄存器的读或写，（起始地址+传输次数*4）地址范围内的寄存器会被访问

传输次数由硬件计算，范围为 0 到 DMATC。

15.3. 通用定时器 L2（TIMERx, x=13）

15.3.1. 简介

通用定时器 L2 (TIMERx, x=13)是单通道定时器，支持输入捕获和输出比较，产生 PWM 信号控制电机和电源管理。通用定时器 L2 含有一个 16 位无符号计数器。

通用定时器 L2 是可编程的，可以被用来计数，其外部事件可以驱动其他定时器

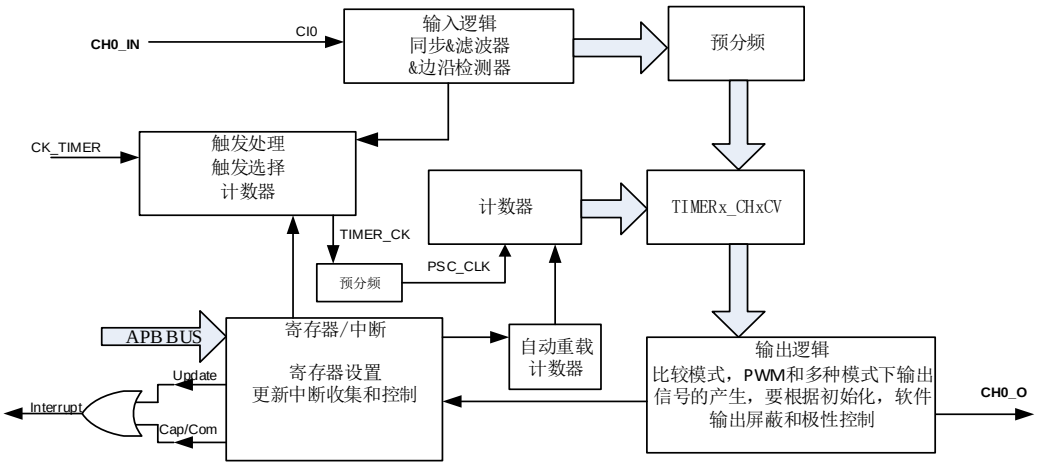
15.3.2. 主要特性

- 总通道数：1
- 计数器宽度：16位
- 时钟源可选：内部时钟
- 计数模式：向上计数
- 可编程的预分频器：16位，运行时可以被改变
- 每个通道可配置：输入捕获模式，输出比较模式，可编程的PWM模式
- 自动重装载功能.
- 中断输出：更新事件，比较/捕获事件

15.3.3. 结构框图

*图 15-44. 通用定时器 L2 结构框图*提供了通用定时器 L2 的内部配置细节

图 15-44. 通用定时器 L2 结构框图



15.3.4. 功能描述

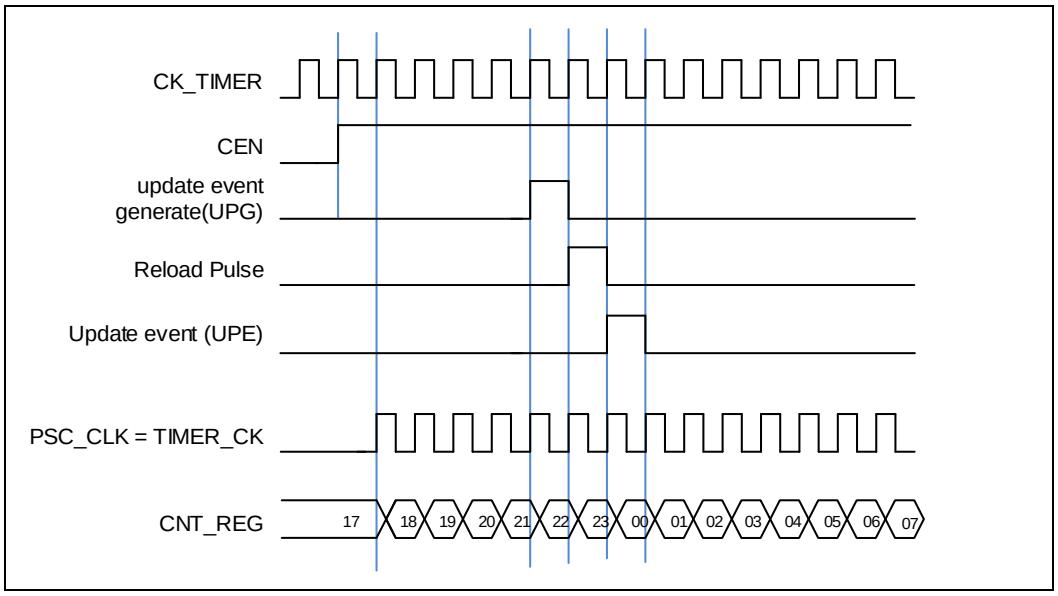
时钟源配置

通用定时器 L2 由内部时钟源 CK_TIMER 驱动

- 定时器时钟TIMER_CK连接到RCU模块的CK_TIMER。

通用定时器 L2 仅有一个时钟源 CK_TIMER，用来驱动计数器预分频器。当 CEN 置位，CK_TIMER 经过预分频器（预分频值由 TIMERx_PSC 寄存器确定）产生 PSC_CLK。

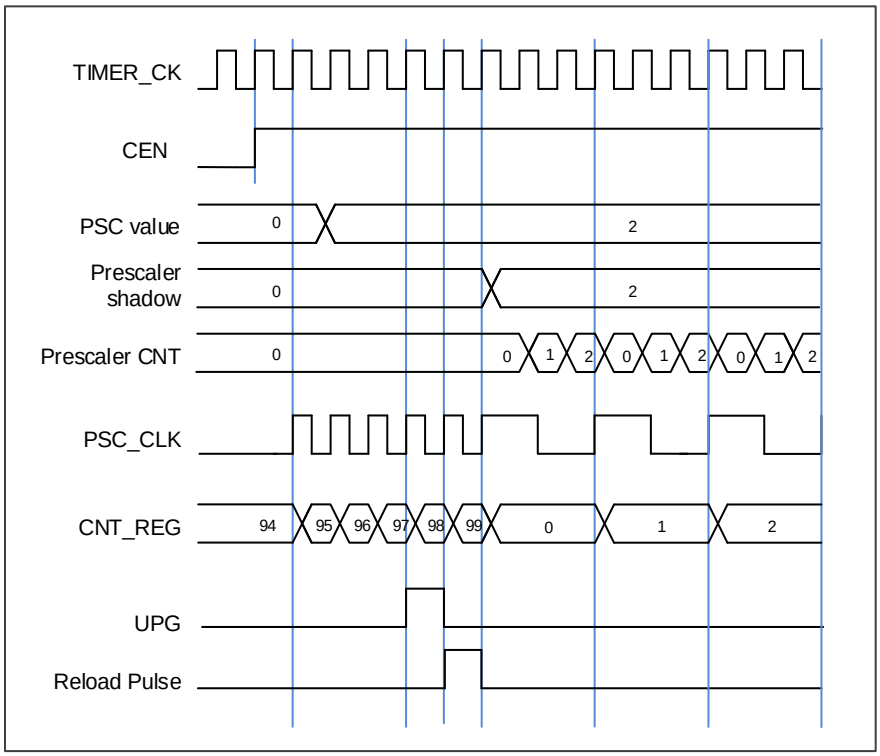
图 15-45. 内部时钟分频为 1 时，计数器的时序图



时钟预分频器

预分频器可以将定时器的时钟（TIMER_CK)频率按 1 到 65536 之间的任意值分频，分频后的时钟 PSC_CLK 驱动计数器计数。分频系数受预分频寄存器 TIMERx_PSC 控制，这个控制寄存器带有缓冲器，它能够在运行时被改变。新的预分频器的参数在下一次更新事件到来时被采用。

图 15-46. 当 PSC 数值从 0 变到 2 时，计数器的时序图



计数器向上计数模式

在这种模式，计数器的计数方向是向上计数。计数器从 0 开始向上连续计数到自动加载值（定义在 TIMERx_CAR 寄存器中），一旦计数器计数到自动加载值，会重新从 0 开始向上计数并产生上溢事件。在向上计数模式中，TIMERx_CTL0 寄存器中的计数方向控制位 DIR 应该被设置成 0。

当通过 TIMERx_SWEVG 寄存器的 UPG 位置 1 来设置更新事件时，计数值会被清 0，并产生更新事件。

如果 TIMERx_CTL0 寄存器的 UPDIS 置 1，则禁止更新事件。

当发生更新事件时，所有影子寄存器（计数器自动重载寄存器，预分频寄存器）都将被更新。

图 15-47. 向上计数时序图，PSC=0/2 和 *图 15-48. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值*给出了一些例子，当 TIMERx_CAR=0x99 时，计数器在不同预分频因子下的行为。

图 15-47. 向上计数时序图，PSC=0/2

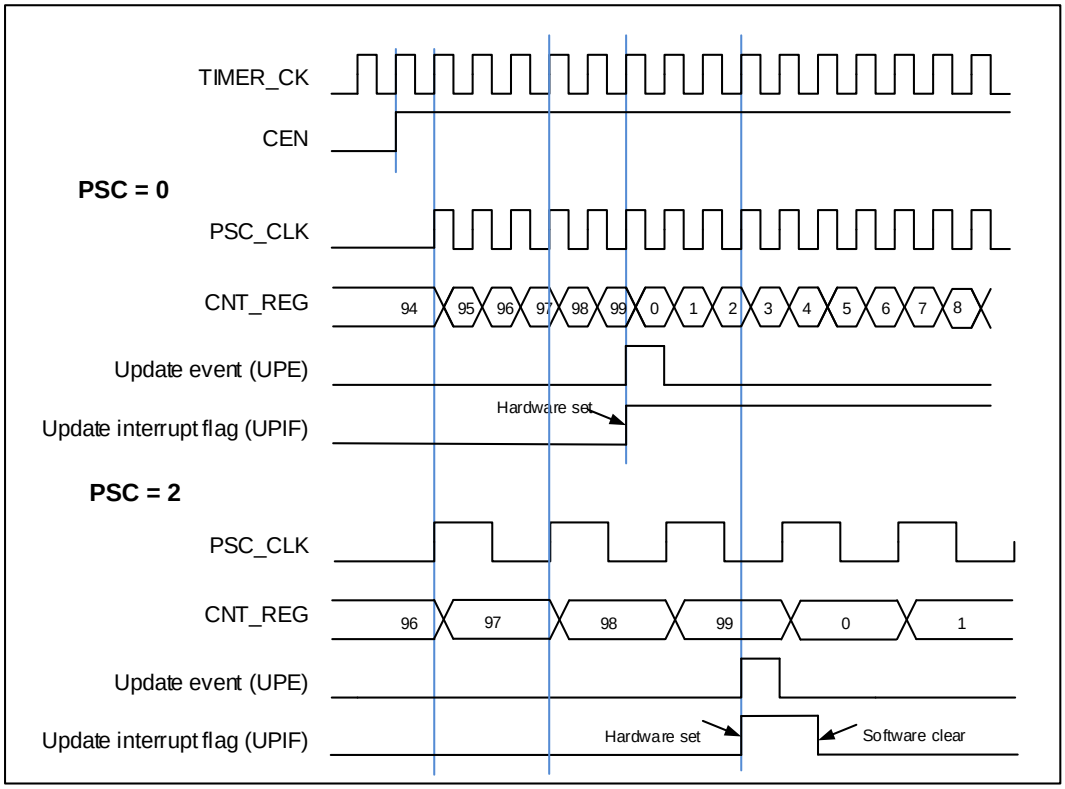
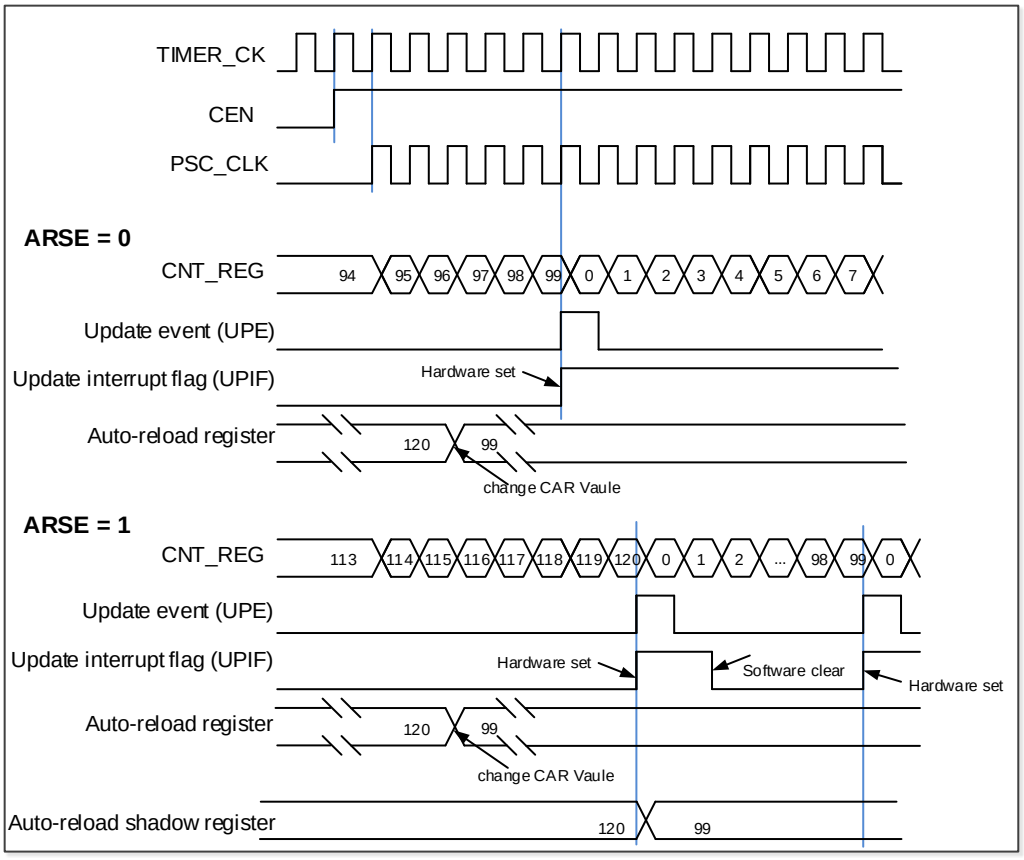


图 15-48. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值



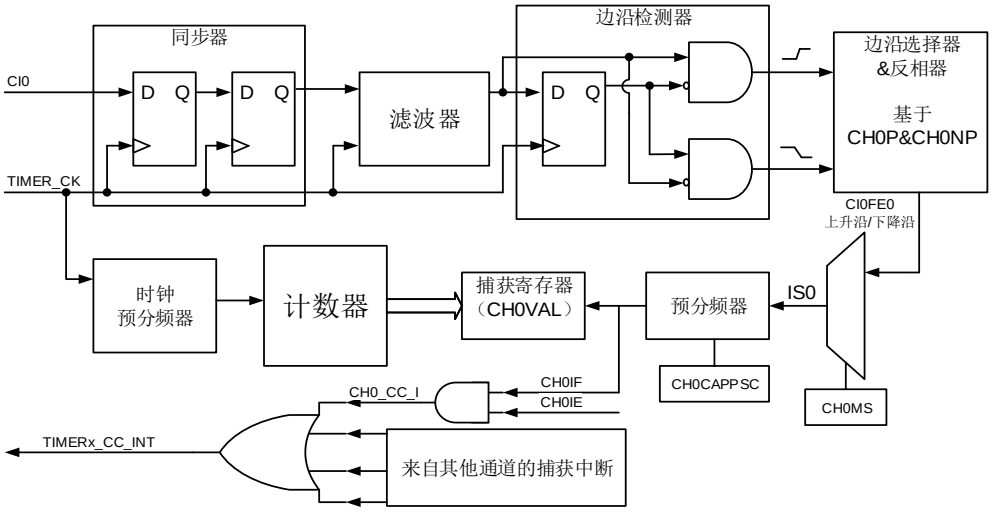
输入捕获和输出比较通道

通用定时器 L2 只有一个独立的通道用于捕获输入或比较输出是否匹配。该通道通道都围绕一个通道捕获比较寄存器建立，包括一个输入级，通道控制器和输出级。

通道输入捕获功能

通道输入捕获功能允许通道测量一个波形时序，频率，周期，占空比等。输入级包括一个数字滤波器，一个通道极性选择，边沿检测和一个通道预分频器。如果在输入引脚上出现被选择的边沿，TIMERx_CHxCV 寄存器会捕获计数器当前的值，同时 CHxIF 位被置 1，如果 CHxIE = 1 则产生通道中断。

图 15-49. 通道输入捕获原理



通道输入信号 CIx 先被 TIMER_CK 信号同步，然后经过数字滤波器采样，产生一个被滤波后的信号。通过边沿检测器，可以选择检测上升沿或者下降沿。通过配置 CHxP 选择使用上升沿或者下降沿。配置 CHxMS.，可以选择其他通道的输入信号，内部触发信号。配置 IC 预分频器，使得若干个输入事件后才产生一个有效的捕获事件。捕获事件发生，CHxVAL 存储计数器的值。

配置步骤如下：

第一步：滤波器配置（TIMERx_CHCTL0寄存器中CHxCAPFLT）：
根据输入信号和请求信号的质量，配置相应的CHxCAPFLT。
第二步：边沿选择（TIMERx_CHCTL2寄存器中CHxP/CHxNP）：
配置CHxP/CHxNP选择上升沿或者下降沿。
第三步：捕获源选择（TIMERx_CHCTL0寄存器中CHxMS）：
一旦通过配置CHxMS选择输入捕获源，必须确保通道配置在输入模式（CHxMS!=0x0），而且TIMERx_CHxCV寄存器不能再被写。
第四步：中断使能（TIMERx_DMAINTEN寄存器中CHxIE）：
使能相应中断，可以获得中断。
第五步：捕获使能（TIMERx_CHCTL2寄存器中CHxEN）。

结果：当期望的输入信号发生时，TIMERx_CHxCV被设置成当前计数器的值，CHxIF为置1。如果CHxIF位已经为1，则CHxOF位置1。根据TIMERx_DMAINTEN寄存器中CHxIE的配置，相应的中断会被提出。

直接产生：软件设置CHxG位，会直接产生中断。

通道输出比较模式

图 15-50. 通道输出比较原理

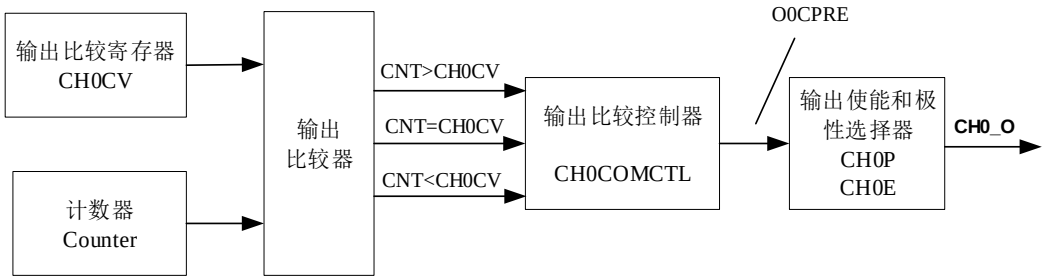


*图 15-50. 通道输出比较原理*给出了输出比较的逻辑电路。通道输出信号 CHx_O 与 OxCPRE 信号的关系描述：OxCPRE 信号高电平有效，CHx_O 的输出情况与 OxCPRE 信号，CHxP 位和 CHxE 位有关（具体情况请见 TIMERx_CHCTL2 寄存器中的描述）。例如，当设置 CHxP=0（CHx_O 高电平有效，与 OxCPRE 输出极性相同）、CHxE=1（CHx_O 输出使能）时：

若 OxCPRE 输出有效（高）电平，则 CHx_O 输出有效（高）电平；

若 OxCPRE 输出无效（低）电平，则 CHx_O 输出无效（低）电平。

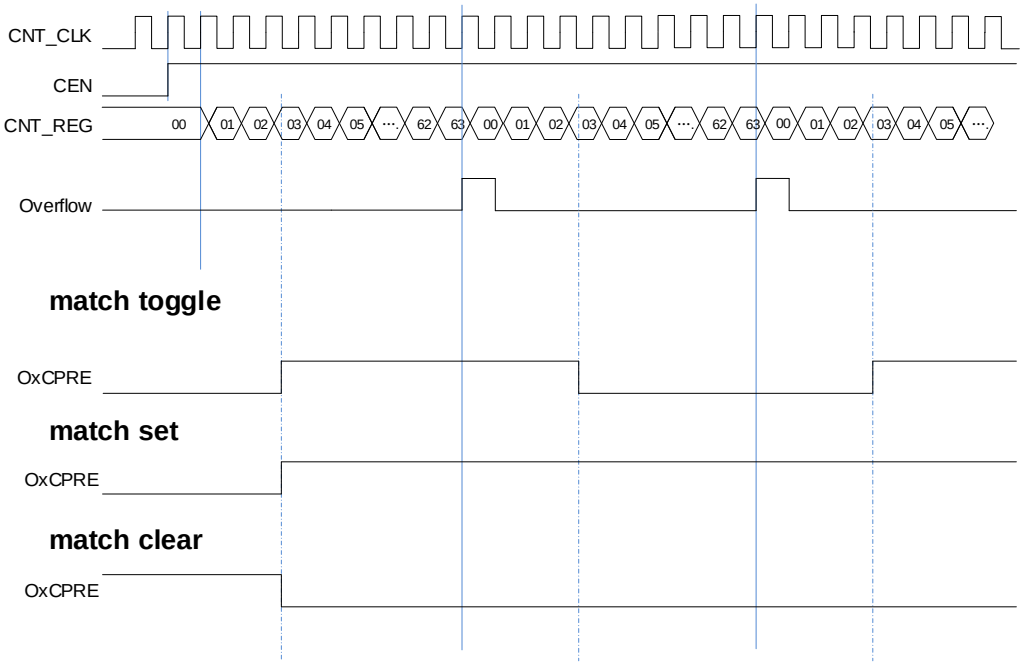
在输出比较模式，TIMERx可以产生时控脉冲，其位置，极性，持续时间和频率都是可编程的。当一个输出通道的CHxVAL寄存器与计数器的值匹配时，根据CHxCOMCTL的配置，这个通道的输出可以被置高电平，被置低电平或者反转。当计数器的值与CHxVAL寄存器的值匹配时，CHxIF位被置1，如果CHxIE = 1则会产生中断，如果CHxDEN=1则会产生DMA请求。

配置步骤如下：

第一步：时钟配置：
配置定时器时钟源，预分频器等。
第二步：比较模式配置：
设置CHxCOMSEN位来配置输出比较影子寄存器；
设置CHxCOMCTL位来配置输出模式（置高电平/置低电平/反转）；
设置CHxP/CHxNP位来选择有效电平的极性；
设置CHxEN使能输出。
第三步：通过CHxIE位配置中断使能。
第四步：通过TIMERx_CAR寄存器和TIMERx_CHxCV寄存器配置输出比较时基：
CHxVAL可以在运行时根据你所期望的波形而改变。
第五步：设置CEN位使能定时器。

*图 15-51. 三种输出比较模式*显示了三种比较输出模式：反转/置高电平/置低电平，CAR=0x63, CHxVAL=0x3。

图 15-51. 三种输出比较模式



输出 PWM 功能

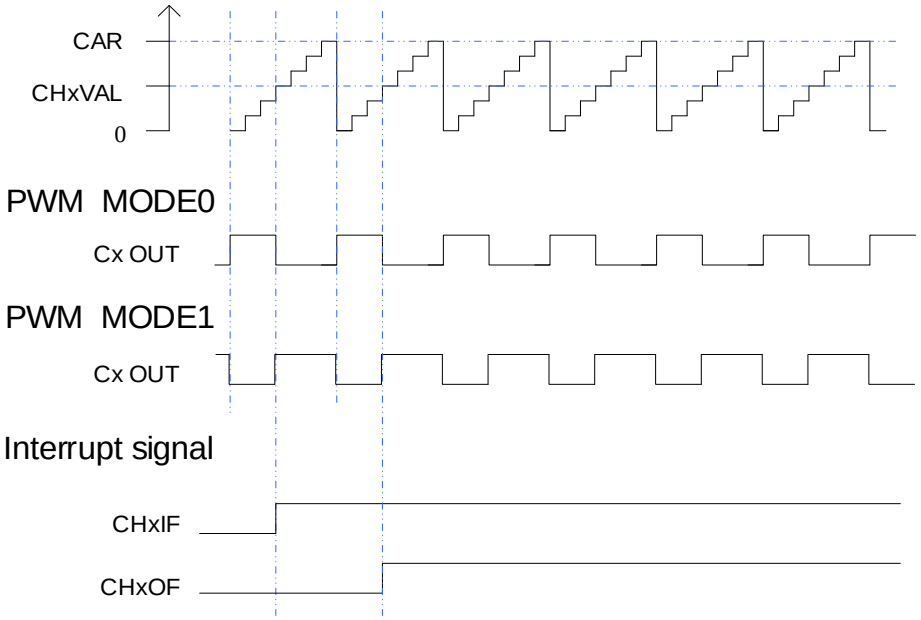
在 PWM 输出模式下（PWM 模式 0 是配置 CHxCOMCTL 为 3'b110，PWM 模式 1 是配置 CHxCOMCTL 为 3'b111），通道根据 TIMERx_CAR 寄存器和 TIMERx_CHxCV 寄存器的值，输出 PWM 波形。

PWM 的周期由 TIMERx_CAR 寄存器值决定，占空比由 TIMERx_CHxCV 寄存器值决定。*图 15-52. PWM 时序图*显示了 EAPWM 的输出波形和中断。

在 PWM0 模式下(CHxCOMCTL==3'b110)，如果 TIMERx_CHxCV 寄存器的值大于 TIMERx_CAR 寄存器的值，通道输出一直为有效电平。

在 PWM0 模式下(CHxCOMCTL==3'b110)，如果 TIMERx_CHxCV 寄存器的值等于 0，通道输出一直为无效电平。

图 15-52. PWM 时序图



通道输出参考信号

根据*图 15-50. 通道输出比较原理*所示，当 TIMERx 用于输出匹配比较模式下，在通道输出信号之前会产生一个中间信号 OxCPRE 信号(通道 x 输出准备信号)。设置 CHxCOMCTL 位可以定义 OxCPRE 信号类型。当 TIMERx 用于输出匹配比较模式下，设置 CHxCOMCTL 位可以定义 OxCPRE 信号(通道 x 输出准备信号)类型。OxCPRE 信号有若干类型的输出功能，包括，设置 CHxCOMCTL=0x00 可以保持原始电平；设置 CHxCOMCTL=0x01 可以将 OxCPRE 信号设置为高电平；设置 CHxCOMCTL=0x02 可以将 OxCPRE 信号设置为低电平；设置 CHxCOMCTL=0x03，在计数器值和 TIMERx_CHxCV 寄存器的值匹配时，可以翻转输出信号。

PWM 模式 0 和 PWM 模式 1 是 OxCPRE 的另一种输出类型，设置 CHxCOMCTL 位域为 0x06 或 0x07 可以配置 PWM 模式 0/PWM 模式 1。在这些模式中，根据计数器值和 TIMERx_CHxCV 寄存器值的关系以及计数方向，OxCPRE 信号改变其电平。具体细节描述，请参考相应的位。

设置 CHxCOMCTL=0x04 或 0x05 可以实现 OxCPRE 信号的强制输出功能。输出比较信号能够直接由软件强置为有效或无效状态，而不依赖于 TIMERx_CHxCV 的值和计数器值之间的比较结果。

定时器调试模式

当 Cortex®-M3 内核停止，DBG_CTL0 寄存器中的 TIMERx_HOLD 配置位被置 1，定时器计数器停止。

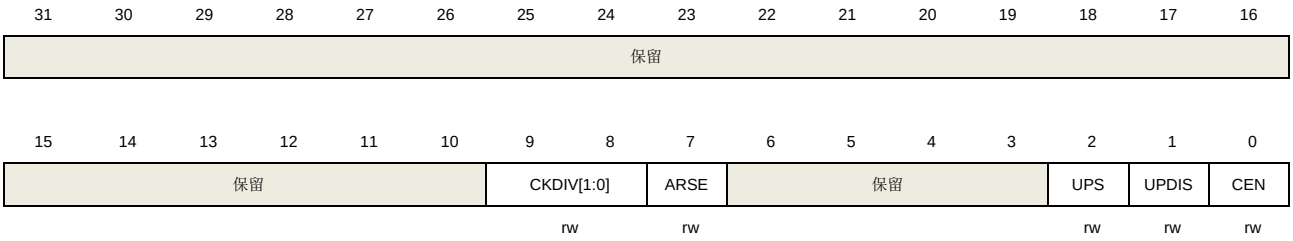
15.3.5. TIMERx 寄存器(x=13)

TIMER13 基地址：0x4000 2000

控制寄存器 0 (TIMERx_CTL0)

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字(32位)访问。



位/位域	名称	描述
31:10	保留	必须保持复位值。
9:8	CKDIV[1:0]	时钟分频 通过软件配置CKDIV，规定定时器时钟(CK_TIMER) 与死区时间和数字滤波器采样时钟(DTS)之间的分频系数。 00：$f_{DTS}=f_{CK_TIMER}$ 01：$f_{DTS}= f_{CK_TIMER}$ /2 10：$f_{DTS}= f_{CK_TIMER}$ /4 11：保留
7	ARSE	自动重载影子使能 0：禁能 TIMERx_CAR 寄存器的影子寄存器。 1：使能 TIMERx_CAR 寄存器的影子寄存器。
6:3	保留	必须保持复位值。
2	UPS	更新请求源 软件配置该位，选择更新事件源. 0：以下事件均会产生更新中断或DMA请求： UPG位被置1 计数器溢出/下溢 复位模式产生的更新 1：下列事件会产生更新中断或DMA请求： 计数器溢出/下溢
1	UPDIS	禁止更新. 该位用来使能或禁能更新事件的产生 0：更新事件使能. 更新事件发生时，相应的影子寄存器被装入预装载值，以下事件均会产生更新事件：

UPG位被置1
计数器溢出/下溢
复位模式产生的更新
1：更新事件禁能.
注意：当该位被置 1 时，UPG 位被置 1 或者复位模式不会产生更新事件，但是计数器和预分频器被重新初始化

0	CEN	计数器使能 0：计数器禁能 1：计数器使能 在软件将CEN位置1后，外部时钟、暂停模式和正交译码器模式才能工作。

DMA 和中断使能寄存器 (TIMERx_DMAINTEN)

地址偏移：0x0C
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留														CH0IE	UPIE
														rw	rw

位/位域	名称	描述
31:2	保留	必须保持复位值。
1	CH0IE	通道 0 比较/捕获中断使能 0：禁止通道 0 中断 1：使能通道 0 中断
0	UPIE	更新中断使能 0：禁止更新中断 1：使能更新中断

中断标志寄存器 (TIMERx_INTF)

地址偏移：0x10
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						CH0OF	保留							CH0IF	UPIF
						rc_w0								rc_w0	rc_w0

位/位域	名称	描述
31:10	保留	必须保持复位值。
9	CH0OF	通道 0 捕获溢出标志 当通道 0 被配置为输入模式时，在 CH0IF 标志位已经被置 1 后，捕获事件再次发生时，该标志位可以由硬件置 1。该标志位由软件清 0. 0：无捕获溢出中断发生 1：发生了捕获溢出中断
8:2	保留	必须保持复位值。
1	CH0IF	通道 0 比较/捕获中断标志 此标志由硬件置 1 软件清 0。当通道 0 在输入模式下时，捕获事件发生时此标志位被置 1；当通道 0 在输出模式下时，此标志位在一个比较事件发生时被置 1。 当通道 0 在输入模式下时，读 TIMERx_CH0CV 会将此标志清零。 0：无通道 0 中断发生 1：通道 0 中断发生
0	UPIF	更新中断标志 此位在任何更新事件发生时由硬件置 1，软件清 0。 0：无更新中断发生 1：发生更新中断

软件事件产生寄存器 (TIMERx_SWEVG)

地址偏移：0x14
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留														CH0G	UPG
														w	w

位/位域	名称	描述
31:2	保留	必须保持复位值。
1	CH0G	通道 0 捕获或比较事件发生 该位由软件置 1，用于在通道 0 产生一个捕获/比较事件，由硬件自动清 0。当此位被置 1，CH0IF 标志位被置 1，若开启对应的中断和 DMA，则发出相应的中断和 DMA

请求。此外，如果通道 0 配置为输入模式，计数器的当前值被 TIMERx_CH0CV 寄存器捕获，如果 CH0IF 标志位已经为 1，则 CH0OF 标志位被置 1。
0：不产生通道 0 捕获或比较事件
1：发生通道 0 捕获或比较事件

0	UPG	更新事件产生 此位由软件置 1，被硬件自动清 0。当此位被置 1 并且向上计数模式，计数器被清 0，预分频计数器将同时被清除。 0：无更新事件产生 1：产生更新事件

通道控制寄存器 0 (TIMERx_CHCTL0)

地址偏移：0x18
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								保留	CH0COMCTL[2:0]			CH0COMSEN	保留	CH0MS[1:0]	
								CH0CAPFLT[3:0]				CH0CAPPSC[1:0]			
									rw			rw		rw	

输出比较模式：

位/位域	名称	描述
31:7	保留	必须保持复位值。
6:4	CH0COMCTL[2:0]	通道 0 输出比较模式 此位定义了输出准备信号 O0CPRE 的输出比较模式，而 O0CPRE 决定了 CH0_O、CH0_ON 的值。另外，O0CPRE 高电平有效，而 CH0_O、CH0_ON 通道的极性取决于 CH0P、CH0NP 位。 000：时基。输出比较寄存器 TIMERx_CH0CV 与计数器 TIMERx_CNT 间的比较对 O0CPRE 不起作用 001：匹配时设置为高。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 为高。 010：匹配时设置为低。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 为低。 011：匹配时翻转。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 翻转。 100：强制为低。强制 O0CPRE 为低电平 101：强制为高。强制 O0CPRE 为高电平 110：PWM 模式 0。在向上计数时，一旦计数器值小于 TIMERx_CH0CV 时，O0CPRE 为高电平，否则为低电平。在向下计数时，一旦计数器的值大于 TIMERx_CH0CV 时，

O0CPRE 为低电平，否则为高电平。
111：PWM 模式 1。在向上计数时，一旦计数器值小于 TIMERx_CH0CV 时，O0CPRE 为低电平，否则为高电平。在向下计数时，一旦计数器的值大于 TIMERx_CH0CV 时，O0CPRE 为高电平，否则为低电平。
如果配置在 PWM 模式下，只有当输出比较模式从时基模式变为 PWM 模式或者比较结果改变时，O0CPRE 电平才改变。

3	CH0COMSEN	通道 0 输出比较影子寄存器使能 当此位被置 1，TIMERx_CH0CV 寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。 0：禁止通道 0 输出/比较影子寄存器 1：使能通道 0 输出/比较影子寄存器 仅在单脉冲模式下(SPM =1)，可以在未确认影子寄存器的情况下使用 PWM 模式
2	保留	必须保持复位值。
1:0	CH0MS[1:0]	通道 0 I/O 模式选择 这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2 寄存器的 CH0EN 位被清 0)时这些位才可写。 00：通道 0 配置为输出 01：通道 0 配置为输入，IS0 映射在 CI0FE0 上 10：保留 11：保留

输入捕获模式：

位/位域	名称	描述
31:8	保留	必须保持复位值。
7:4	CH0CAPFLT[3:0]	通道 0 输入捕获滤波控制 CI0 输入信号可以通过数字滤波器进行滤波，该位域配置滤波参数。 数字滤波器的基本原理：根据 f_{SAMP} 对 CI0 输入信号进行连续采样，并记录信号相同电平的次数。达到该位配置的滤波参数后，认为是有效电平。 滤波器参数配置如下：

CH0CAPFLT [3:0]	采样次数	f_{SAMP}
4’b0000	无滤波器	
4’b0001	2	f_{CK_TIMER}
4’b0010	4	
4’b0011	8	
4’b0100	6	$f_{DTS}/2$
4’b0101	8	
4’b0110	6	$f_{DTS}/4$
4’b0111	8	
4’b1000	6	$f_{DTS}/8$
4’b1001	8	
4’b1010	5	$f_{DTS}/16$

		4’b1011	6
		4’b1100	8
		4’b1101	5
		4’b1110	6
		4’b1111	8

$f_{DTS}/32$

3:2 CH0CAPPSC[1:0] 通道 0 输入捕获预分频器

这 2 位定义了通道 0 输入的预分频系数。当 TIMERx_CHCTL2 寄存器中的 CH0EN =0 时，则预分频器复位。

00：无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获

01：每 2 个事件触发一次捕获

10：每 4 个事件触发一次捕获

11：每 8 个事件触发一次捕获

1:0 CH0MS[1:0] 通道 0 模式选择

与输出比较模式相同

通道控制寄存器 2 (TIMERx_CHCTL2)

地址偏移：0x20

复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留..												CH0NP	保留	CH0P	CH0EN
												rw		rw	rw

位/位域	名称	描述
31:4	保留	必须保持复位值。
3	CH0NP	通道 0 互补输出极性 当通道 0 配置为输出模式，此位定义了互补输出信号的极性。 0：通道0互补输出高电平为有效电平 1：通道0互补输出低电平为有效电平 当通道 0 配置为输入模式时，此位和 CH0P 联合使用，作为输入信号 CI0 的极性选择控制信号。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 或 10 时此位不能被更改。
2	保留	必须保持复位值。
1	CH0P	通道 0 极性 当通道 0 配置为输出模式时，此位定义了输出信号极性。 0：通道0高电平为有效电平 1：通道0低电平为有效电平

		当通道 0 配置为输入模式时，此位定义了 CI0 信号极性 [CH0NP, CH0P] 将选择 CI0FE0 或者 CI1FE0 的有效边沿或者捕获极性 [CH0NP==0, CH0P==0]：把 CIxFE0 的上升沿作为捕获或者从模式下触发的有效信号，并且 CIxFE0 不会被翻转。 [CH0NP==0, CH0P==1]：把 CIxFE0 的下降沿作为捕获或者从模式下触发的有效信号，并且 CIxFE0 会被翻转。 [CH0NP==1, CH0P==0]：保留。 [CH0NP==1, CH0P==1]：把 CIxFE0 的上升沿和下降沿都作为捕获或者从模式下触发的有效信号，并且 CIxFE0 不会被翻转。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 或 10 时此位不能被更改。
0	CH0EN	通道 0 捕获/比较使能 当通道 0 配置为输出模式时，将此位置 1 使能 CH0_O 信号有效。当通道 0 配置为输入模式时，将此位置 1 使能通道 0 上的捕获事件。 0：禁止通道 0 1：使能通道 0

计数器寄存器 (TIMERx_CNT)

地址偏移：0x24
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CNT[15:0]	这些位是当前的计数值。写操作能改变计数器值。

预分频寄存器 (TIMERx_PSC)

地址偏移： 0x28
复位值： 0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

PSC[15:0]

rw

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	PSC[15:0]	计数器时钟预分频值 计数器时钟等于 TIMER_CK 时钟除以(PSC+1)，每次当更新事件产生时，PSC 的值被装入到对应的影子寄存器。

计数器自动重载寄存器 (TIMERx_CAR)

地址偏移：0x2C
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CARL[15:0]															

rw

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CARL[15:0]	计数器自动重载值 这些位定义了计数器的自动重载值。

通道 0 捕获/比较值寄存器 (TIMERx_CH0CV)

地址偏移：0x34
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0VAL[15:0]															

rw

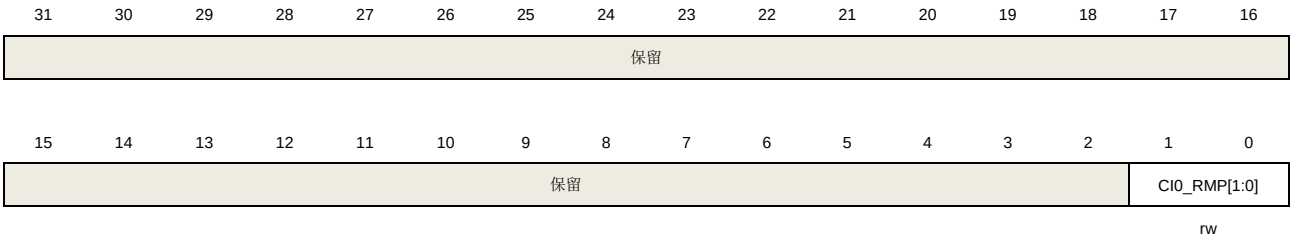
位/位域	名称	描述
31:16	保留	必须保持复位值。

15:0	CH0VAL[15:0]	通道 0 的捕获或比较值 当通道 0 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 0 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

通道输入重映射寄存器(TIMERx_IRMP)

地址偏移： 0x50
复位值： 0x0000 0000

该寄存器只能按字(32位)访问。



位/位域	名称	描述
31:2	保留	必须保持复位值。
1:0	CI0_RMP[1:0]	通道 0 输入重映射 00: 通道 0 输入连接到 GPIO(TIMER13_CH0) 01: 通道 0 输入连接到 RTCCLK 10: 通道 0 输入连接到 HXTAL/32 clock 11: 通道 0 输入连接到 CKOUTSEL

15.4. 通用定时器 L3（TIMERx,x=14）

15.4.1. 简介

通用定时器 L3（TIMER14）是两通道定时器，支持输入捕获和输出比较。可以产生 PWM 信号控制电机和电源管理。通用定时器 L3 含有一个 16 位无符号计数器。

通用定时器 L3 是可编程的，可以被用来计数，其外部事件可以驱动其他定时器

通用定时器 L3 包含了一个死区时间插入模块，非常适合电机控制。

定时器和定时器之间是相互独立，但是他们可以被同步在一起形成一个更大的定时器，这些定时器的计数器一致地增加。

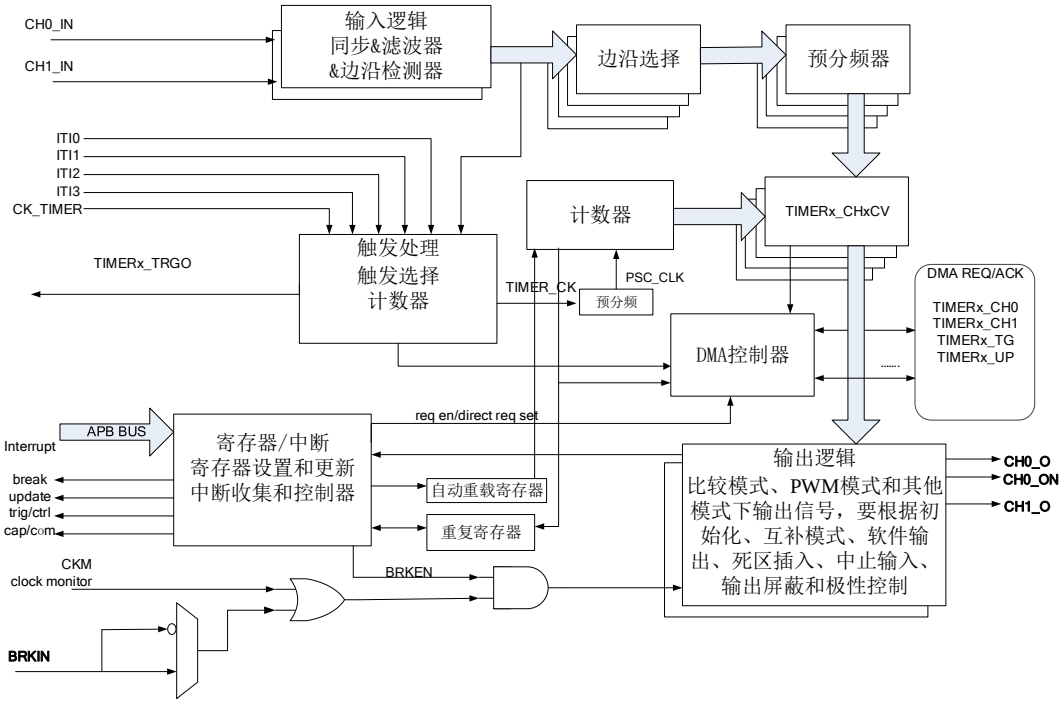
15.4.2. 主要特性

- 总通道数：2；
- 计数器宽度：16位；
- 时钟源可选：内部时钟，内部触发，外部输入；
- 计数模式：向上计数；
- 可编程的预分频器：16位，运行时可以被改变；
- 每个通道可配置：输入捕获模式，输出比较模式，可编程的PWM模式，单脉冲模式；
- 可编程的死区时间；
- 自动重装载功能；
- 可编程的计数器重复功能；
- 中止输入功能；
- 中断输出和DMA请求：更新事件，比较/捕获事件，换相事件和中止事件；
- 多个定时器的菊链使得一个定时器可以同时启动多个定时器；
- 定时器的同步允许被选择的定时器在同一个时钟周期开始计数；
- 定时器主-从管理。

15.4.3. 结构框图

*图 15-53. 通用定时器 L3 结构框图*提供了通用定时器 L3 的内部配置细节

图 15-53. 通用定时器 L3 结构框图



15.4.4. 功能描述

时钟源配置

通用定时器 L3 可以由内部时钟源 CK_TIMER 或者由 SMC（TIMERx_SMCFG 寄存器位[2:0]）控制的复用时钟源驱动。

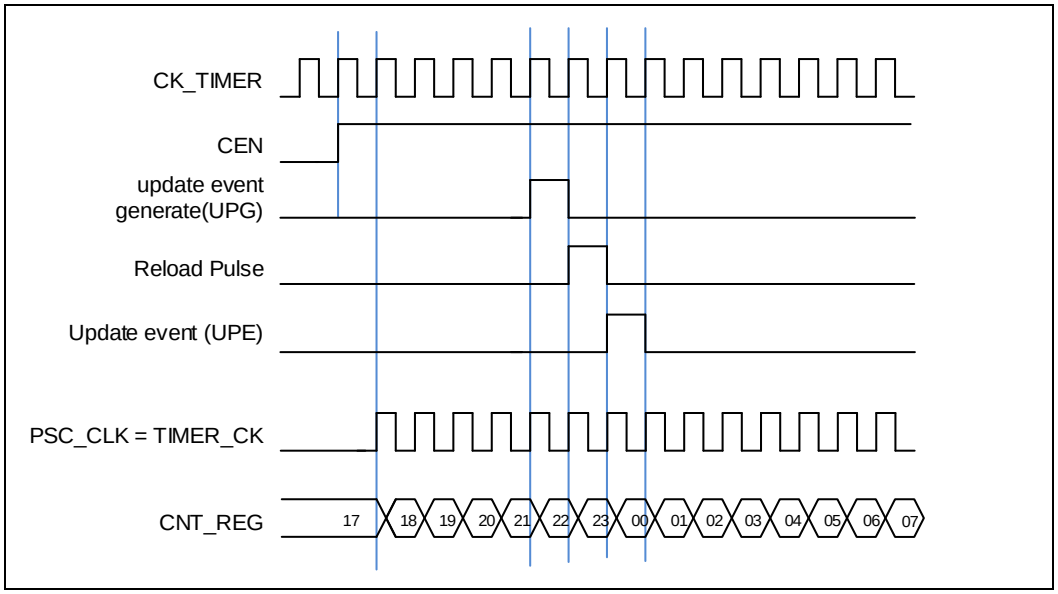
- SMC[2:0]==3’b000，定时器选择内部时钟源（连接到RCU模块的CK_TIMER）

如果 SMC[2:0]==3’b000，默认用来驱动计数器预分频器的是内部时钟源 CK_TIMER。当 CEN 置位，CK_TIMER 经过预分频器（预分频值由 TIMERx_PSC 寄存器确定）产生 PSC_CLK。

这种模式下，驱动预分频器计数的 TIMER_CK 等于来自于 RCU 模块的 CK_TIMER

如果将 TIMERx_SMCFG 寄存器的 SMC[2:0]设置为 0x7，预分频器被其他时钟源(由 TIMERx_SMCFG 寄存器的 TRGS [2:0]区域选择)驱动，在下文说明。当 SMC 位被设置为 0x4、0x5 和 0x6，计数器预分频器时钟源由内部时钟 CK_TIMER 驱动。

图 15-54. 内部时钟分频为 1 时，计数器的时序图



- SMC[2:0]==3'b111(外部时钟模式0)，定时器选择外部输入引脚作为时钟源

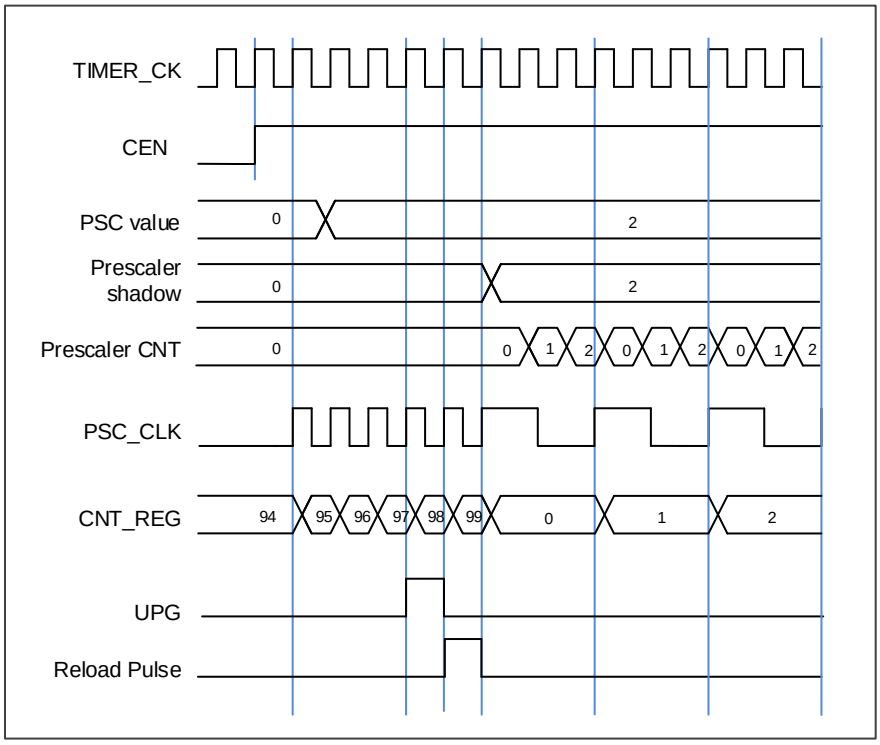
计数器预分频器可以在 TIMERx_CI0/ TIMERx_CI1 引脚的每个上升沿或下降沿计数。这种模式可以通过设置 SMC [2:0]为 0x7 同时设置 TRGS[2:0]为 0x4，0x5 或 0x6 来选择。

计数器预分频器也可以在内部触发信号 ITI0/1/2/3 的上升沿计数。这种模式可以通过设置 SMC [2:0]为 0x7 同时设置 TRGS [2:0]为 0x0, 0x1, 0x2 或者 0x3。

时钟预分频器

预分频器可以将定时器的时钟（TIMER_CK)频率按 1 到 65536 之间的任意值分频，分频后的时钟 PSC_CLK 驱动计数器计数。分频系数受预分频寄存器 TIMERx_PSC 控制，这个控制寄存器带有缓冲器，它能够在运行时被改变。新的预分频器的参数在下一次更新事件到来时被采用。

图 15-55. 当 PSC 数值从 0 变到 2 时，计数器的时序图



计数器向上计数模式

在这种模式，计数器的计数方向是向上计数。计数器从0开始向上连续计数到计数器自动加载值（定义在TIMERx_CAR寄存器中），一旦计数器计数到自动加载值，会重新从0开始向上计数，并且产生上溢事件。另外，在(TIMERx_CREP+1)次上溢后将会产生更新事件。在向上计数模式中，TIMERx_CTL0寄存器中的计数方向控制位DIR应该被设置成0。

当通过 TIMERx_SWEVG 寄存器的 UPG 位置 1 来设置更新事件时，计数值会被清 0，并产生更新事件。

如果 TIMERx_CTL0 寄存器的 UPDIS 置 1，则禁止更新事件。

当发生更新事件时，所有影子寄存器（重复计数器，计数器自动重载寄存器，预分频寄存器）都将被更新。

图 15-56. 向上计数时序图，PSC=0/2 和 *图 15-57. 向上计数时序图，在运行时改变 TIMERx CAR 寄存器的值*给出了一些例子，当 TIMERx_CAR=0x99 时，计数器在不同预分频因子下的行为。

图 15-56. 向上计数时序图，PSC=0/2

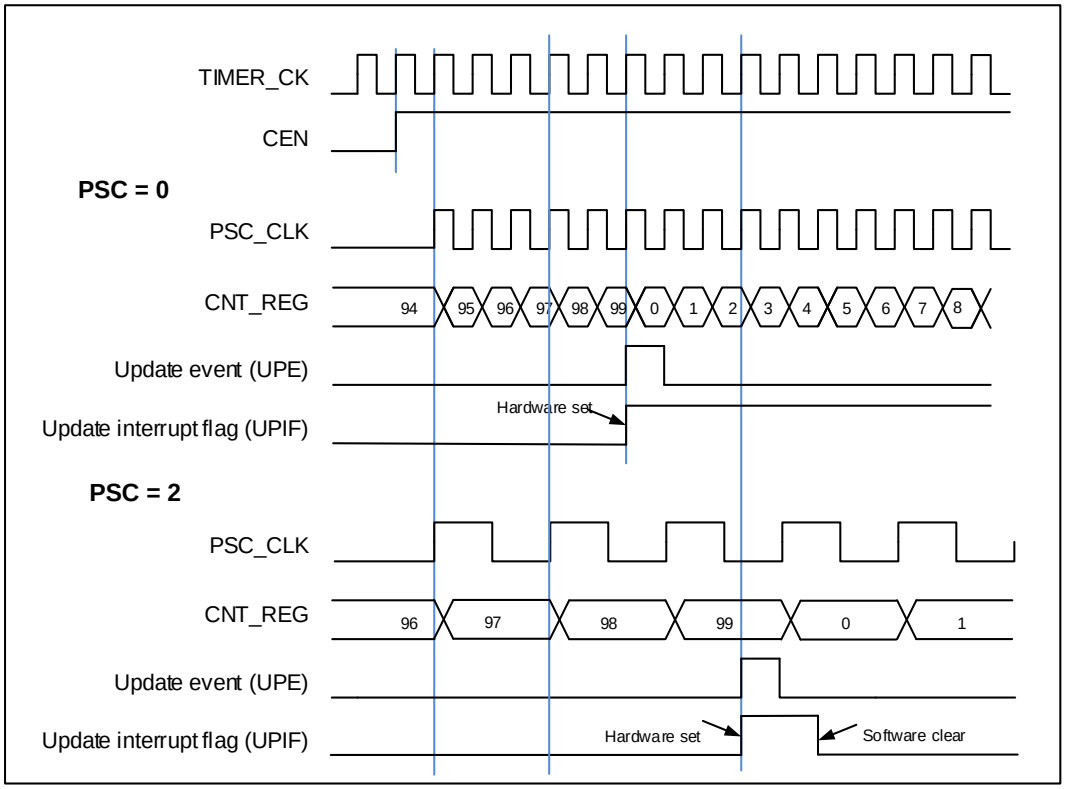
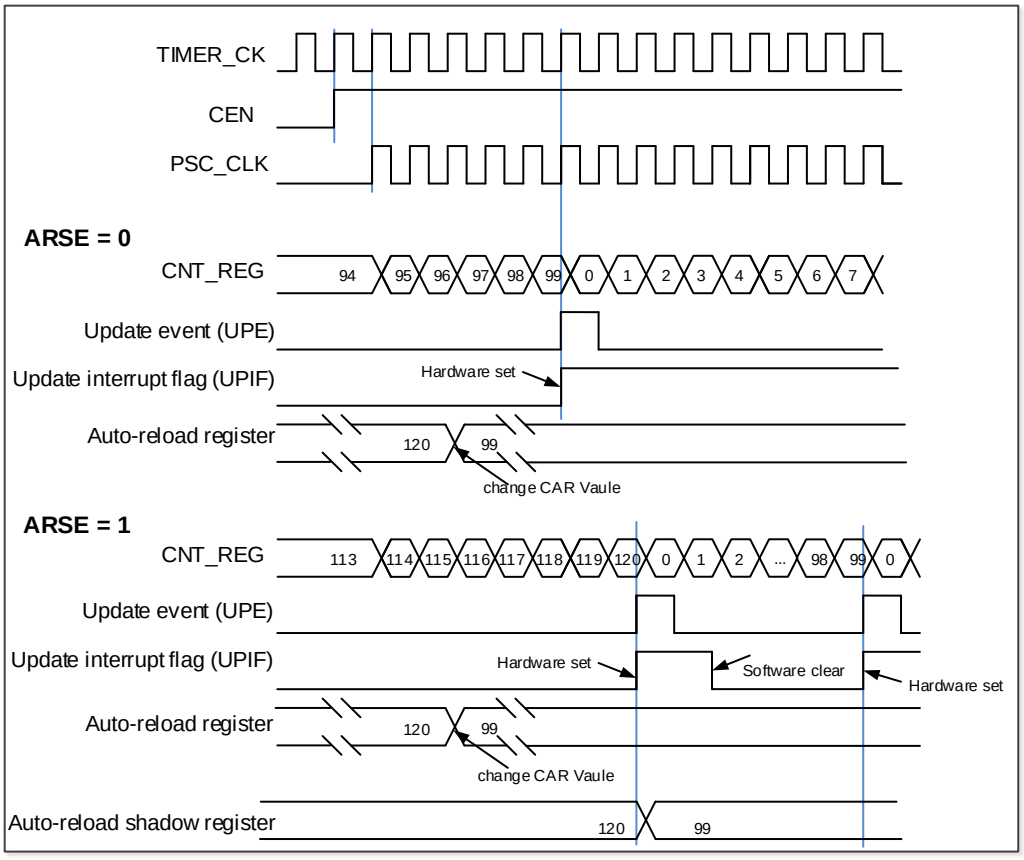


图 15-57. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值

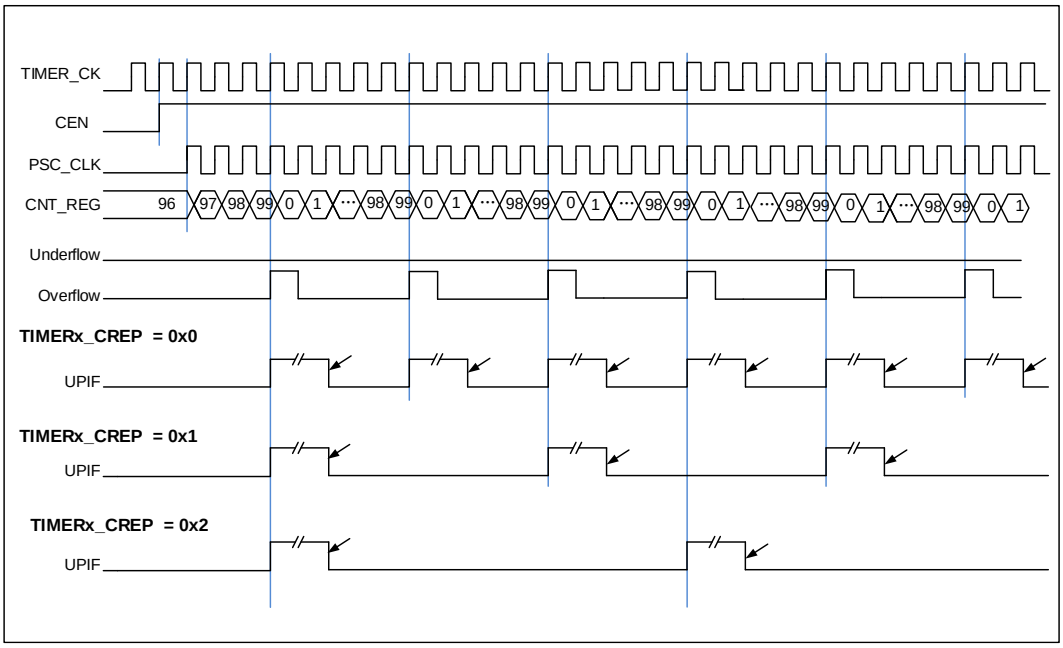


更新事件（来自上溢/下溢）频率配置

重复计数器是用来在 N+1 个计数周期之后产生更新事件，更新定时器的寄存器，N 为 TIMERx_CREP 寄存器的 CREP。向上计数模式下，重复计数器在每次计数器上溢时递减。

将 TIMERx_SWEVG 寄存器的 UPG 位置 1 可以重载 TIMERx_CREP 寄存器中 CREP 的值并产生一个更新事件。

图 15-58. 在向上计数模式下计数器重复时序图



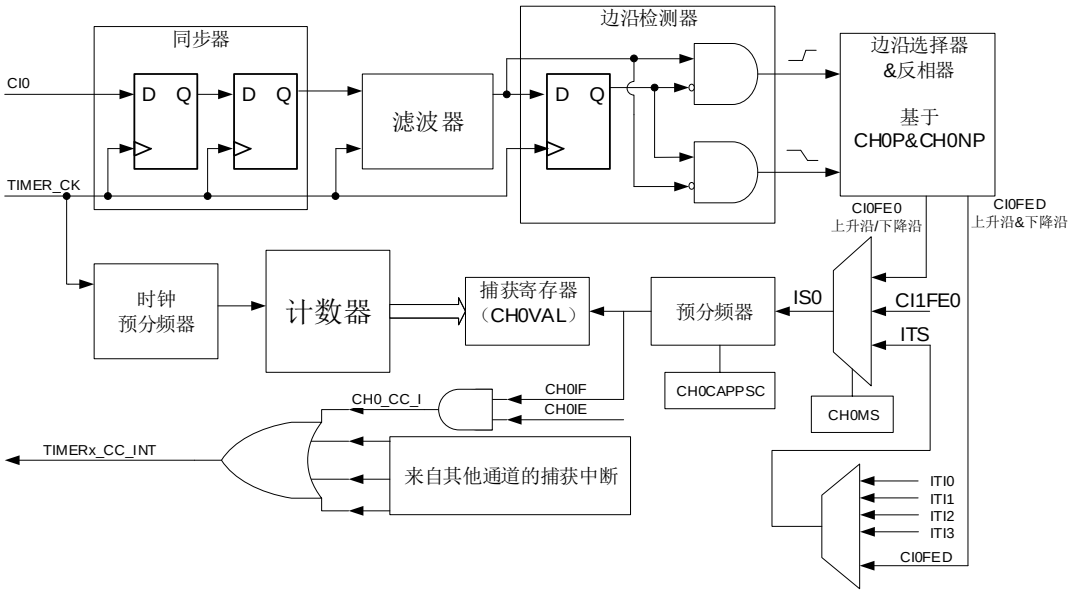
输入捕获和输出比较通道

通用定时器 L3 拥有两个独立的通道用于捕获输入或比较输出是否匹配。每个通道都围绕一个通道捕获比较寄存器建立，包括一个输入级，通道控制器和输出级。

通道输入捕获功能

通道输入捕获功能允许通道测量一个波形时序，频率，周期，占空比等。输入级包括一个数字滤波器，一个通道极性选择，边沿检测和一个通道预分频器。如果在输入引脚上出现被选择的边沿，TIMERx_CHxCV 寄存器会捕获计数器当前的值，同时 CHxIF 位被置 1，如果 CHxIE = 1 则产生通道中断。

图 15-59. 通道输入捕获原理



通道输入信号 CIx 有两种选择，一种是 TIMERx_CHx 信号，另一种是 TIMERx_CH0,TIMERx_CH1 和 TIMERx_CH2 异或之后的信号。通道输入信号 CIx 先被 TIMER_CK 信号同步，然后经过数字滤波器采样，产生一个被滤波后的信号。通过边沿检测器，可以选择检测上升沿或者下降沿。通过配置 CHxP 选择使用上升沿或者下降沿。配置 CHxMS.，可以选择其他通道的输入信号，内部触发信号。配置 IC 预分频器，使得若干个输入事件后才产生一个有效的捕获事件。捕获事件发生，CHxVAL 存储计数器的值。

配置步骤如下：

第一步：滤波器配置（TIMERx_CHCTL0寄存器中CHxCAPFLT）：
根据输入信号和请求信号的质量，配置相应的CHxCAPFLT。

第二步：边沿选择（TIMERx_CHCTL2寄存器中CHxP/CHxNP）：
配置CHxP/CHxNP选择上升沿或者下降沿。

第三步：捕获源选择（TIMERx_CHCTL0寄存器中CHxMS）：
一旦通过配置CHxMS选择输入捕获源，必须确保通道配置在输入模式（CHxMS!=0x0），而且TIMERx_CHxCV寄存器不能再被写。

第四步：中断使能（TIMERx_DMAINTEN寄存器中CHxIE和CHxDEN）：
使能相应中断，可以获得中断和DMA请求。

第五步：捕获使能（TIMERx_CHCTL2寄存器中CHxEN）。

结果：当期望的输入信号发生时，TIMERx_CHxCV被设置成当前计数器的值，CHxIF为置1。如果CHxIF位已经为1，则CHxOF位置1。根据TIMERx_DMAINTEN寄存器中CHxIE和CHxDEN的配置，相应的中断和DMA请求会被提出。

直接产生：软件设置CHxG位，会直接产生中断和DMA请求。

通道输入捕获功能也可用来测量 TIMERx_CHx 引脚上信号的脉冲波宽度。例如，一个 PWM 波连接到 CI0。配置 TIMERx_CHCTL0 寄存器中 CH0MS 为 2’b01，选择通道 0 的捕获信号为 CI0 并设置上升沿捕获。配置 TIMERx_CHCTL0 寄存器中 CH1MS 为 2’b10，选择通道 1 捕获信号为 CI0 并设置下降沿捕获。计数器配置为复位模式，在通道 0 的上升沿复位。

TIMERX_CH0CV 寄存器测量 PWM 的周期值，TIMERx_CH1CV 寄存器测量 PWM 占空比值。

通道输出比较功能

图 15-60. 通道输出比较原理（带有互补输出的通道，x=0）

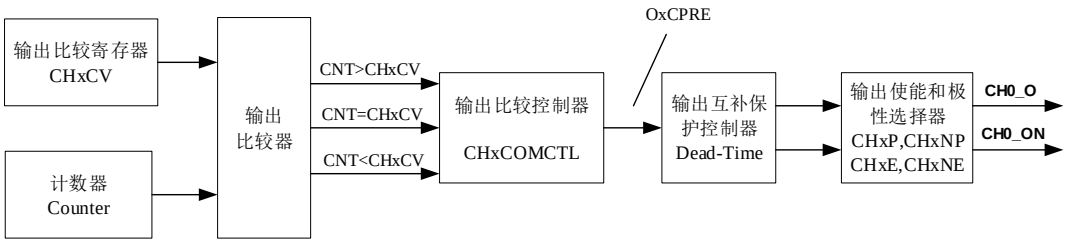


图 15-61. 通道输出比较原理

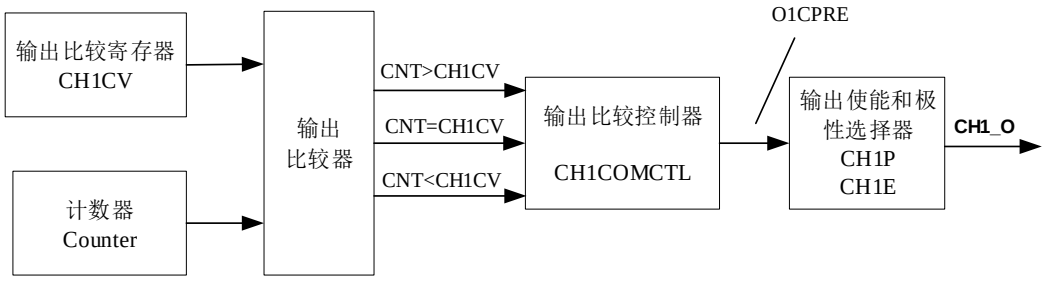


*图 15-60. 通道输出比较原理（带有互补输出的通道，x=0）*和*图 15-61. 通道输出比较原理*分别给出了输出比较的原理。通道输出信号 CHx_O/CHx_ON 与 OxCPRE 信号的关系描述如下：OxCPRE 信号高电平有效，CHx_O/CHx_ON 的输出情况与 OxCPRE 信号，CHxP/CHxNP 位和 CHxE/CHxNE 位有关（具体情况请见 TIMERx_CHCTL2 寄存器中的描述）。例如：

1）当设置CHxP=0（CHx_O高电平有效，与OxCPRE输出极性相同）、CHxE=1（CHx_O输出使能）时：

若OxCPRE输出有效（高）电平，则CHx_O输出有效（高）电平；

若OxCPRE输出无效（低）电平，则CHx_O输出无效（低）电平。

2）当设置CHxNP=1（CHx_ON低电平有效，与OxCPRE输出极性相反）、CHxNE=1（CHx_ON输出使能）时：

若OxCPRE输出有效（高）电平，则CHx_ON输出有效（低）电平；

若OxCPRE输出无效（低）电平，则CHx_ON输出无效（高）电平。

当 CH0_O 和 CH0_ON 同时输出时，CH0_O 和 CH0_ON 的具体输出情况还与 TIMERx_CCHP 寄存器中的相关位（ROS、IOS、POE 和 DTCFG 等位）有关。

在通道输出比较功能，TIMERx 可以产生时控脉冲，其位置，极性，持续时间和频率都是可编程的。当一个输出通道的 CHxVAL 寄存器与计数器的值匹配时，根据 CHxCOMCTL 的配置，这个通道的输出可以被置高电平，被置低电平或者反转。当计数器的值与 CHxVAL 寄存器的值匹配时，CHxIF 位被置 1，如果 CHxIE = 1 则会产生中断，如果 CHxDEN=1 则会产生 DMA 请求。

配置步骤如下：

第一步：时钟配置：

配置定时器时钟源，预分频器等。

第二步：比较模式配置：

设置CHxCOMSEN位来配置输出比较影子寄存器；

设置CHxCOMCTL位来配置输出模式（置高电平/置低电平/反转）；

设置CHxP/CHxNP位来选择有效电平的极性；

设置CHxEN使能输出。

第三步：通过CHxIE/CHxDEN位配置中断/DMA请求使能。

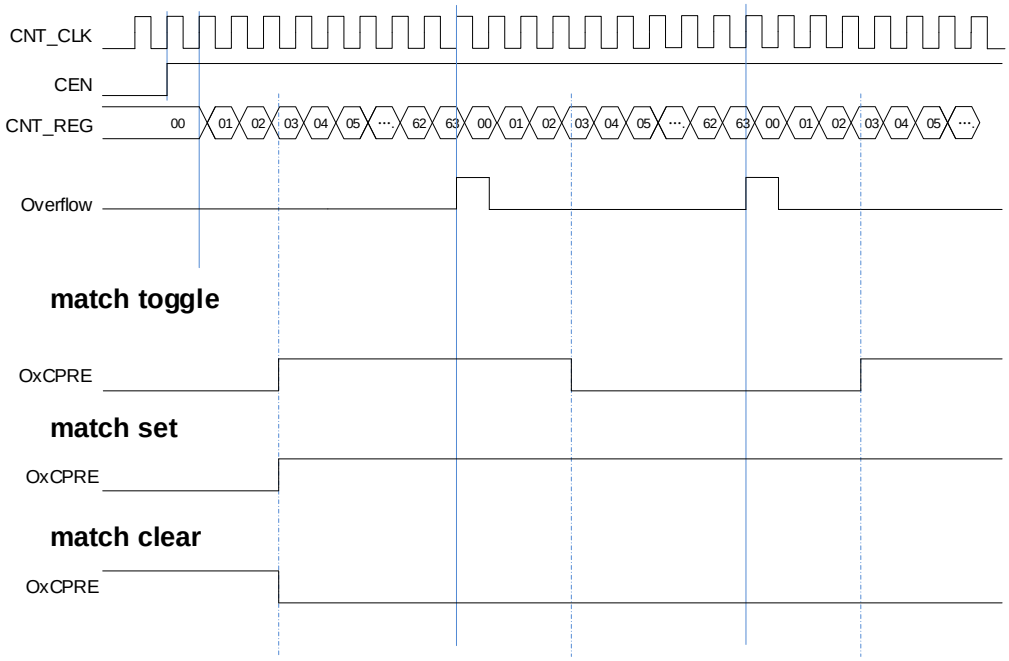
第四步：通过TIMERx_CAR寄存器和TIMERx_CHxCV寄存器配置输出比较时基：

CHxVAL可以在运行时根据你所期望的波形而改变。

第五步：设置CEN位使能定时器。

*图 15-62. 三种输出比较模式*显示了三种比较输出模式：反转/置高电平/置低电平，CAR=0x63, CHxVAL=0x3。

图 15-62. 三种输出比较模式



输出 PWM 功能

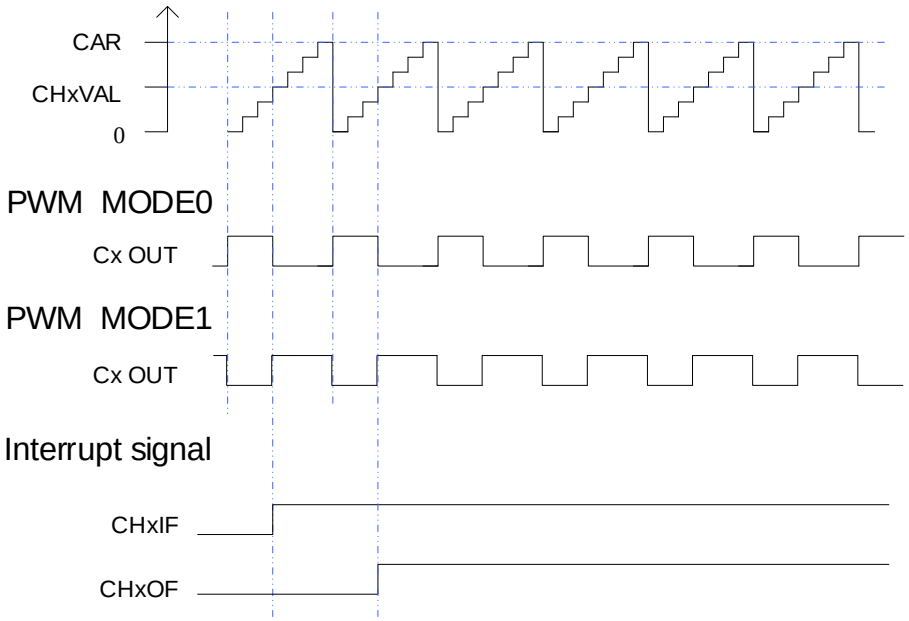
在 PWM 输出模式下（PWM 模式 0 是配置 CHxCOMCTL 为 3'b110，PWM 模式 1 是配置 CHxCOMCTL 为 3'b111），通道根据 TIMERx_CAR 寄存器和 TIMERx_CHxCV 寄存器的值，输出 PWM 波形。

PWM 的周期由 TIMERx_CAR 寄存器值决定，占空比由 TIMERx_CHxCV 寄存器值决定。*图 15-63. PWM 时序图*显示了 PWM 的输出波形和中断。

在 PWM0 模式下(CHxCOMCTL==3'b110)，如果 TIMERx_CHxCV 寄存器的值大于 TIMERx_CAR 寄存器的值，通道输出一直为有效电平。

在 PWM0 模式下(CHxCOMCTL==3'b110)，如果 TIMERx_CHxCV 寄存器的值等于 0，通道输出一直为无效电平。

图 15-63. PWM 时序图



通道输出参考信号

当 TIMERx 用于输出匹配比较模式下，设置 CHxCOMCTL 位可以定义 OxCPRE 信号(通道 x 准备信号)类型。OxCPRE 信号有若干类型的输出功能，包括，设置 CHxCOMCTL=0x00 可以保持原始电平；设置 CHxCOMCTL=0x01 可以将 OxCPRE 信号设置为高电平；设置 CHxCOMCTL=0x02 可以将 OxCPRE 信号设置为低电平；设置 CHxCOMCTL=0x03，在计数器值和 TIMERx_CHxCV 寄存器的值匹配时，可以翻转输出信号。

PWM 模式 0 和 PWM 模式 1 是 OxCPRE 的另一种输出类型，设置 CHxCOMCTL 位域位 0x06 或 0x07 可以配置 PWM 模式 0/PWM 模式 1。在这些模式中，根据计数器值和 TIMERx_CHxCV 寄存器值的关系以及计数方向，OxCPRE 信号改变其电平。具体细节描述，请参考相应的位。

设置 CHxCOMCTL =0x04 或 0x05 可以实现 OxCPRE 信号的强制输出功能。输出比较信号能够直接由软件强置为有效或无效状态，而不依赖于 TIMERx_CHxCV 的值和计数器值之间间的比较结果。

通道输出互补 PWM

CHx_O 和 CHx_ON 是一对互补输出通道，这两个信号不能同时有效。TIMERx 有两路通道，只有第一路有互补输出通道。互补信号 CHx_O 和 CHx_ON 是由一组参数来决定：TIMERx_CHCTL2 寄存器中的 CHxEN 和 CHxNEN 位，TIMERx_CCHP 寄存器中和 TIMERx_CTL1 寄存器中的 POEN, ROS, IOS, ISOx 和 ISOxN 位。输出极性由 TIMERx_CHCTL2 寄存器中的 CHxP 和 CHxNP 位来决定。

表 15-7. 由参数控制的互补输出表

互补参数					输出状态	
POEN	ROS	IOS	CHxEN	CHxNEN	CHx_O	CHx_ON
0	0/1	0	0	0	CHx_O / CHx_ON = LOW CHx_O / CHx_ON 输出禁能(1)	
				1	CHx_O/CHx_ON输出关闭状态(2)： 通道先输出无效电平：CHx_O = CHxP，CHx_ON = CHxNP）；如果死区产生时钟未失效，在死区时间之后： CHx_O = ISOx，CHx_ON = ISOxN (3)	
			1	0		
				1		
		1	x	x	CHx_O/CHx_ON输出关闭状态： 通道先输出无效电平：CHx_O = CHxP，CHx_ON = CHxNP）；如果死区产生时钟未失效，在死区时间之后： CHx_O = ISOx，CHx_ON = ISOxN	
1	0	0/1	0	0	CHx_O/CHx_ON = LOW CHx_O/CHx_ON输出禁能	
				1	CHx_O = LOW CHx_O输出禁能	CHx_ON=OxCPRE⊕(4)CHxNP CHx_ON输出使能
			1	0	CHx_O=OxCPRE⊕CHxP CHx_O输出使能	CHx_ON = LOW CHx_ON输出禁能
				1	CHx_O=OxCPRE⊕CHxP CHx_O输出使能	CHx_ON=(!OxCPRE)(5)⊕CHxNP CHx_ON输出使能
	1		0	0	CHx_O = CHxP CHx_O输出关闭状态	CHx_ON = CHxNP CHx_ON输出关闭状态
				1	CHx_O = CHxP CHx_O输出关闭状态	CHx_O=OxCPRE⊕CHxNP CHx_ON输出使能
			1	0	CHx_O=OxCPRE⊕CHxP CHx_O输出使能	CHx_ON = CHxNP CHx_ON输出关闭状态
				1	CHx_O=OxCPRE⊕CHxP CHx_O输出使能	CHx_ON=（!OxCPRE）⊕CHxNP CHx_ON输出使能

注意：

（6） 输出禁能：CHx_O / CHx_ON 输出与对应引脚断开，对应引脚电平受 GPIO 上下拉配置控制，无上下拉时为悬空高阻态；

（7） 输出关闭状态：CHx_O / CHx_ON 输出无效电平（CHx_O = 0⊕CHxP = CHxP）；

（8） 详情见中止模式章节。

（9） ⊕：异或操作；

（10） (!OxCPRE)：OxCPRE 信号的互补信号。

互补 PWM 插入死区时间

设置 CHxEN 和 CHxNEN 为 1’b1 同时设置 POEN，死区插入就会被使能。DTCFG 位域定义

了死区时间，死区时间对通道 0 有效。死区时间的细节，请参考 TIMERx_CCHP 寄存器。

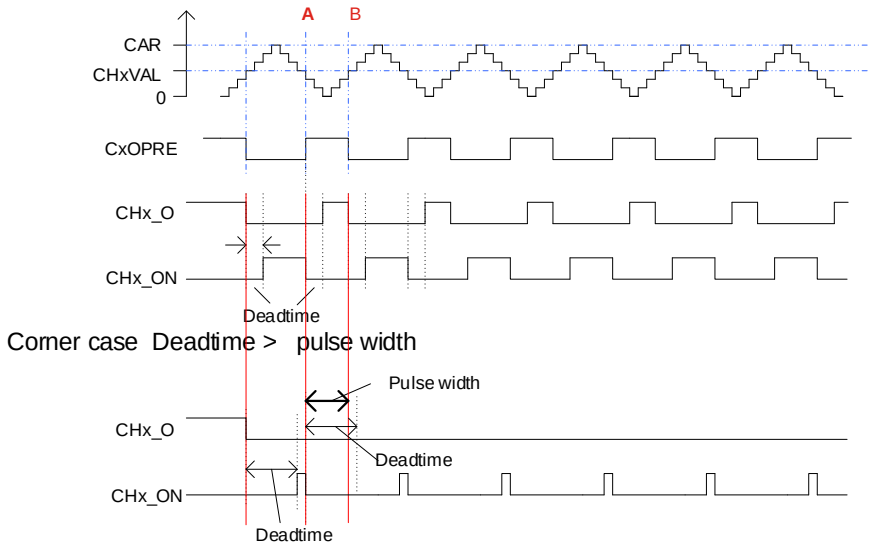
死区时间的插入，确保了通道互补的两路信号不会同时有效。

在 PWM0 模式，当通道 x 匹配发生时（TIMERx 计数器= CHxVAL），OxCPRE 反转。在*图 15-64. 带死区时间的互补输出*中的 A 点，CHx_O 信号在死区时间内为低电平，直到死区时间过后才变为高电平，而 CHx_ON 信号立刻变为低电平。同样，在 B 点，计数器再次匹配（TIMERx 计数器= CHxVAL），OxCPRE 信号被清 0，CHx_O 信号被立即清零，CHx_ON 信号在死区时间内仍然是低电平，在死区时间过后才变为高电平。

有时会有一些死角事件发生，例如：

- 如果死区延时大于或者等于 CHx_O 信号的占空比，CHx_O 信号一直为无效值（如*图 15-64. 带死区时间的互补输出*）。
- 如果死区延时大于或者等于 CHx_ON 信号的占空比，CHx_ON 信号一直为无效值。

图 15-64. 带死区时间的互补输出



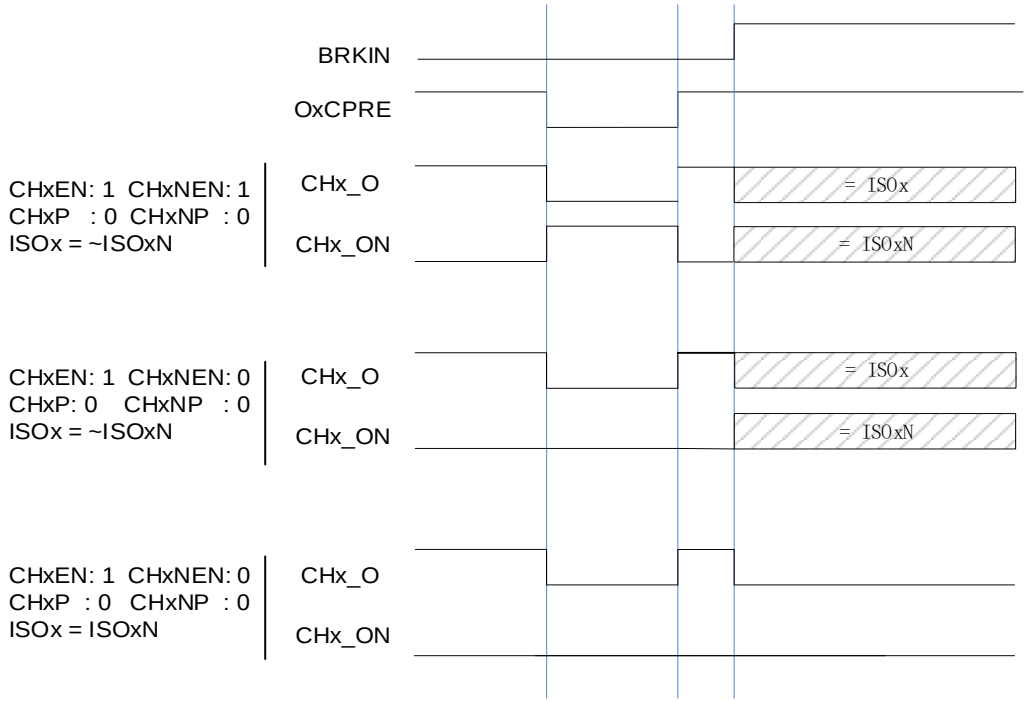
中止模式

使用中止模式时，输出 CHx_O 和 CHx_ON 信号电平被以下位控制，TIMERx_CCHP 寄存器的 POEN, IOS 和 ROS 位，TIMERx_CTL1 寄存器的 ISOx 和 ISOxN 位。当中止事件发生时，CHx_O 和 CHx_ON 信号输出不能同时设置为有效电平。中止源可以选择中止输入引脚，也可以选择 HXTAL 时钟失效事件。时钟失败事件由 RCU 中的时钟监视器(CKM)产生。将 TIMERx_CCHP 寄存器的 BRKEN 位置 1 可以使能中止功能。TIMERx_CCHP 寄存器的 BRKP 位决定了中止输入极性。

发生中止时，POEN 位被异步清除，一旦 POEN 位为 0，CHx_O 和 CHx_ON 被 TIMERx_CTL1 寄存器中的 ISOx 位和 ISOxN 驱动。如果 IOS=0，定时器释放输出使能，否则输出使能仍然为高。起初互补输出被置于复位状态，然后死区时间产生器重新被激活，以便在一个死区时间后驱动输出，输出电平由 ISOx 和 ISOxN 位配置。

发生中止时，TIMERx_INTF 寄存器的 BRKIF 位被置 1。如果 BRKIE=1，中断产生。

图 15-65. 通道响应中止输入（高电平有效）时，输出信号的行为

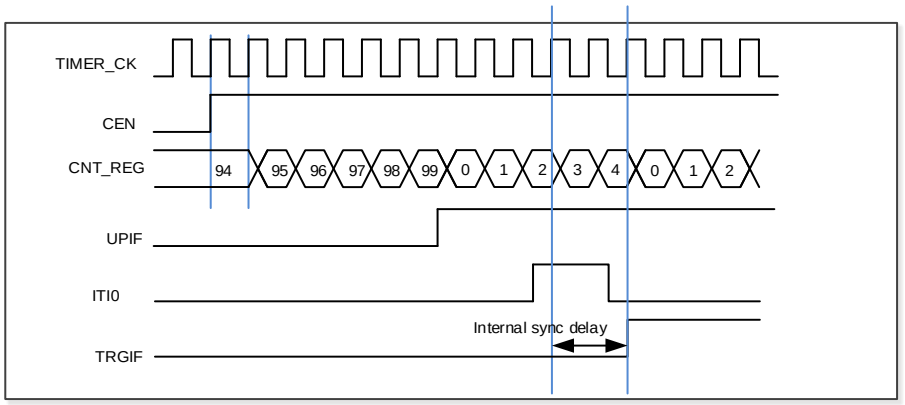
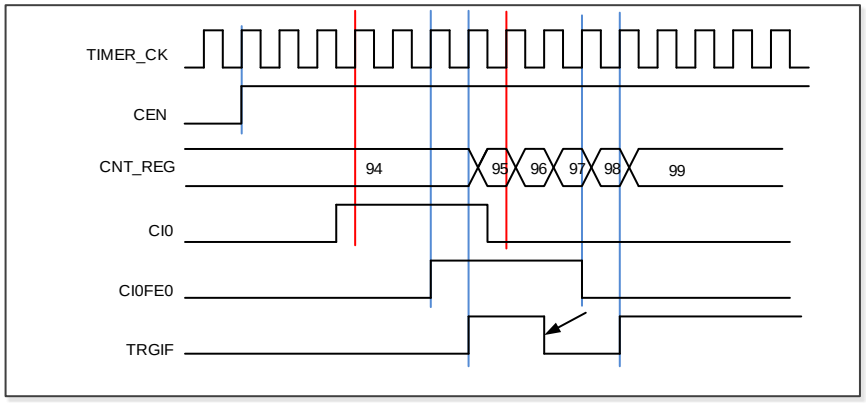
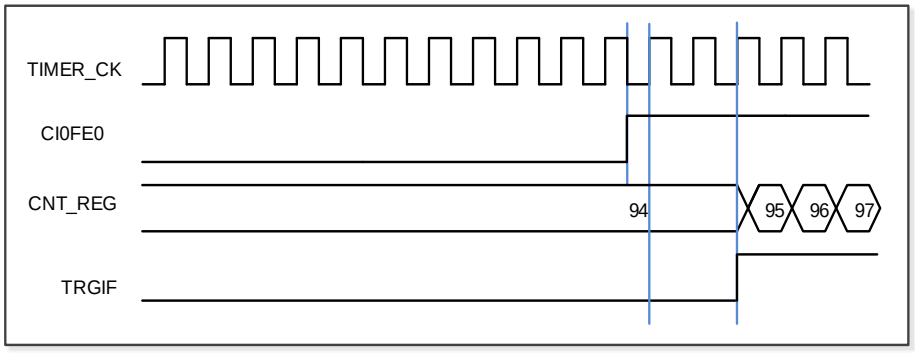


主-从管理

TIMERx 能在多种模式下同步外部触发，包括复位模式，暂停模式和事件模式，可以通过设置 TIMERx_SMCFG 寄存器中的 SMC [2:0]配置这些模式。这些模式的输入触发源可以通过设置 TIMERx_SMCFG 寄存器中的 TRGS [2:0]来选择。

表 15-8. 从模式例子列表

	模式选择	触发源选择	极性选择	滤波和预分频
列举	SMC[2:0] 3'b100（复位模式） 3'b101（暂停模式） 3'b110（事件模式）	TRGS[2:0] 000: ITI0 001: ITI1 010: ITI2 011: ITI3 100: CI0F_ED 101: CI0FE0 110: CI1FE1 111: 保留	如果触发源是CI0FE0或者CI1FE1，配置CHxP和 CHxNP来选择极性和反相	触发源ITIx，滤波和预分频不可用 触发源CIx，配置CHxCAPFLT设置滤波，分频不可用
例1	复位模式 当触发输入上升沿，计数器清零重启	TRGIS[2:0]=3'b000 选择ITI0为触发源	触发源是ITI0，极性选择不可用	触发源是ITI0，滤波和预分频不可用

<table>
<tr><th></th><th>模式选择</th><th>触发源选择</th><th>极性选择</th><th>滤波和预分频</th></tr>
<tr><td></td><td colspan="4">图 15-66. 复位模式下的控制电路
 
</td></tr>
<tr><td rowspan="2">例2</td><td>暂停模式
当触发输入为低的时候，计数器暂停计数</td><td>TRGIS[2:0]=3’b101
选择CI0FE0为触发源</td><td>TI0S=0.（非异或）
[CH0NP==0,
CH0P==0]不反相.在上升沿捕获</td><td>在这个例子中滤波被旁路</td></tr>
<tr><td colspan="4">图 15-67. 暂停模式下的控制电路
 
</td></tr>
<tr><td rowspan="2">例3</td><td>事件模式
触发输入的上升沿计数器开始计数</td><td>TRGIS[2:0]=3’b101
选择CI0FE0为触发源.</td><td>TI0S=0（非异或）
[CH0NP==0,CH0P==0]
不反相</td><td>在这个例子中滤波被旁路</td></tr>
<tr><td colspan="4">图 15-68. 事件模式下的控制电路
 
</td></tr>
</table>

单脉冲模式

单脉冲模式与重复模式是相反的，设置 TIMERx_CTL0 寄存器的 SPM 位置 1，则使能单脉冲

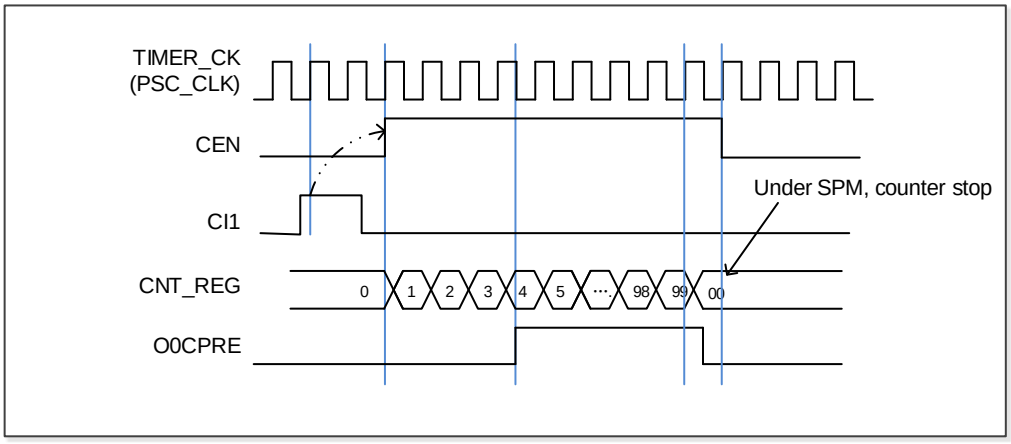
模式。当 SPM 置 1，计数器在下次更新事件到来后清零并停止计数。为了得到脉冲波，可以通过设置 CHxCOMCTL 配置 TIMERx 为 PWM 模式或者比较模式。

一旦设置定时器运行在单脉冲模式下，没有必要设置 TIMERx_CTL0 寄存器的定时器使能位 CEN=1 来使能计数器。触发信号沿或者软件写 CEN=1 都可以产生一个脉冲，此后 CEN 位一直保持为 1 直到更新事件发生或者 CEN 位被软件写 0。如果 CEN 位被软件清 0，计数器停止工作，计数值被保持。

在单脉冲模式下，有效的外部触发边沿会将 CEN 位置 1，使能计数器。然而，执行计数值和 TIMERx_CHxCV 寄存器值的比较结果依然存在一些时钟延迟。为了最大限度减少延迟，用户可以将 TIMERx_CHCTL0 寄存器的 CHxCOMFEN 位置 1。单脉冲模式下，触发上升沿产生之后，OxCPRE 信号将被立即强制转换为与发生比较匹配时相同的电平，但是不用考虑比较结果。只有输出通道配置为 PWM0 或 PWM1 输出运行模式下时 CHxCOMFEN 位才可用，触发源来源于触发信号。

图 15-69. 单脉冲模式，TIMERx_CHxCV = 4 TIMERx_CAR=99 展示了一个例子。

图 15-69. 单脉冲模式，TIMERx_CHxCV = 4 TIMERx_CAR=99



定时器互连

参考*高级定时器（TIMERx,x=0）*

表 15-9. TIMERx(x=14)定时器内部互连

Slave TIMER	ITI0(TRGS = 000)	ITI1(TRGS = 001)	ITI2(TRGS = 010)	ITI3(TRGS = 011)
TIMER14	TIMER1	TIMER2	保留	保留

定时器 DMA 模式

定时器 DMA 模式是指通过 DMA 模块配置定时器的寄存器。有两个跟定时器 DMA 模式相关的寄存器：TIMERx_DMACFG 和 TIMERx_DMATB。当然，必须要使能 DMA 请求，一些内部中断事件可以产生 DMA 请求。当中断事件发生，TIMERx 会给 DMA 发送请求。DMA 配置成 M2P 模式，PADDR 是 TIMERx_DMATB 寄存器地址，DMA 就会访问 TIMERx_DMATB 寄存器。实际上，TIMERx_DMATB 寄存器只是一个缓冲，定时器会将 TIMERx_DMATB 映射到一个内部寄存器，这个内部寄存器由 TIMERx_DMACFG 寄存器中的 DMATA 来指定。如果 TIMERx_DMACFG 寄存器的 DMATC 位域值为 0，表示 1 次传输，定时器的发送 1 个 DMA 请求就可以完成。如果 TIMERx_DMACFG 寄存器的 DMATC 位域值不为 1，例如其值为 3，

表示 4 次传输，定时器就需要再多发 3 次 DMA 请求。在这 3 次请求下，DMA 对 TIMERx_DMATB 寄存器的访问会映射到访问定时器的 DMATA+0x4, DMATA+0x8, DMATA+0xc 寄存器。总之，发生一次 DMA 内部中断请求，定时器会连续发送（DMATC+1）次请求。

如果再来 1 次 DMA 请求事件，TIMERx 将会重复上面的过程。

定时器调试模式

当 Cortex®-M3 内核停止，DBG_CTL1 寄存器中的 TIMERx_HOLD 配置位被置 1，定时器计数器停止。

15.4.5. TIMERx 寄存器(x=14)

TIMER14基地址：0x4001 4000

控制寄存器 0 (TIMERx_CTL0)

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						CKDIV[1:0]		ARSE	保留			SPM	UPS	UPDIS	CEN
						rw		rw				rw	rw	rw	rw

位/位域	名称	描述
31:10	保留	必须保持复位值。
9:8	CKDIV[1:0]	时钟分频 通过软件配置CKDIV，规定定时器时钟(CK_TIMER) 与死区时间和数字滤波器采样时钟(DTS)之间的分频系数。 00：$f_{DTS}=f_{CK_TIMER}$ 01：$f_{DTS}= f_{CK_TIMER}$ /2 10：$f_{DTS}= f_{CK_TIMER}$ /4 11：保留
7	ARSE	自动重载影子使能 0：禁能 TIMERx_CAR 寄存器的影子寄存器。 1：使能 TIMERx_CAR 寄存器的影子寄存器。
6:4	保留	必须保持复位值。
3	SPM	单脉冲模式 0：单脉冲模式禁能。更新事件发生后，计数器继续计数 1：单脉冲模式使能。在下一次更新事件发生时，计数器停止计数
2	UPS	更新请求源 软件配置该位，选择更新事件源. 0：以下事件均会产生更新中断或DMA请求： UPG位被置1 计数器溢出/下溢 复位模式产生的更新 1：下列事件会产生更新中断或DMA请求： 计数器溢出/下溢

1	UPDIS	禁止更新. 该位用来使能或禁能更新事件的产生 0：更新事件使能. 更新事件发生时，相应的影子寄存器被装入预装载值，以下事件均会产生更新事件： UPG位被置1 计数器溢出/下溢 复位模式产生的更新 1：更新事件禁能. 注意：当该位被置 1 时，UPG 位被置 1 或者复位模式不会产生更新事件，但是计数器和预分频器被重新初始化
0	CEN	计数器使能 0：计数器禁能 1：计数器使能 在软件将 CEN 位置 1 后，外部时钟、暂停模式和正交译码器模式才能工作。

控制寄存器 1 (TIMERx_CTL1)

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					ISO1	ISO0N	ISO0	保留	MMC[2:0]			DMAS	CCUC	保留	CCSE
					rw	rw	rw		rw			rw	rw		rw

位/位域	名称	描述
31:11	保留	必须保持复位值。
10	ISO1	通道 1 的空闲状态输出 参考 ISO0 位
9	ISO0N	通道 0 的互补通道空闲状态输出 0：当 POEN 复位，CH0_ON 设置低电平. 1：当 POEN 复位，CH0_ON 设置高电平 此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0]位为 00 的时候可以被更改.
8	ISO0	通道 0 的空闲状态输出 0：当 POEN 复位，CH0_O 设置低电平 1：当 POEN 复位，CH0_O 设置高电平 如果 CH0_ON 生效，一个死区时间后 CH0_O 输出改变。此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0]位为 00 的时候可以被更改.

7	保留	必须保持复位值。
6:4	MMC[2:0]	主模式控制 这些位控制 TRGO 信号的选择，TRGO 信号由主定时器发给从定时器用于同步功能 000：当产生一个定时器复位事件后，输出一个TRGO信号，定时器复位源为： 主定时器产生一个复位事件 TIMERx_SWEVG寄存器中UPG位置1 001：当产生一个定时器使能事件后，输出一个TRGO信号，定时器使能源为： CEN位置1 在暂停模式下，触发输入置1 010：当产生一个定时器更新事件后，输出一个TRGO信号，更新事件源由UPDIS和UPS位决定 011：当通道0在发生一次捕获或一次比较成功时，主模式控制器产生一个TRGO脉冲 100：当产生一次比较事件时，输出一个TRGO信号，比较事件源来自O0CPRE 101：当产生一次比较事件时，输出一个TRGO信号，比较事件源来自O1CPRE 110：当产生一次比较事件时，输出一个TRGO信号，比较事件源来自O2CPRE 111：当产生一次比较事件时，输出一个 TRGO 信号，比较事件源来自 O3CPRE
3	DMAS	DMA请求源选择 0：当通道捕获/比较事件发生时，发送通道 x 的 DMA 请求 . 1：当更新事件发生，发送通道 x 的 DMA 请求
2	CCUC	换相控制影子寄存器更新控制 当换相控制影子寄存器（CHxEN, CHxNEN 和 CHxCOMCTL 位）使能(CCSE=1)，这些影子寄存器更新控制如下： 0：CMTG 位被置 1 时更新影子寄存器 1：当 CMTG 位被置 1 或检测到 TRIGI 上升沿时，影子寄存器更新 当通道没有互补输出时，此位无效。
1	保留	必须保持复位值。
0	CCSE	换相控制影子使能 0：影子寄存器 CHxEN, CHxNEN 和 CHxCOMCTL 位禁能. 1：影子寄存器 CHxEN, CHxNEN 和 CHxCOMCTL 位使能. 如果这些位已经被写入了，换相事件到来时这些位才被更新 当通道没有互补输出时，此位无效

从模式配置寄存器 (TIMERx_SMCFG)

地址偏移：0x08

复位值: 0x0000 0000

该寄存器通过字访问（32位）。

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

保留	MSM	TRGS[2:0]	保留	SMC[2:0]
	rw	rw		rw

位/位域	名称	描述
31:8	保留	必须保持复位值。
7	MSM	主-从模式 该位被用来同步被选择的定时器同时开始计数。通过 TRIGI 和 TRGO，定时器被连接在一起，TRGO 用做启动事件。 0：主从模式禁能。 1：主从模式使能。
6:4	TRGS[2:0]	触发选择 该位域用来指定选择哪一个信号作为用来同步计数器的触发输入源。 000: ITI0 001: ITI1 010: ITI2 011: ITI3 100: CI0F_ED 101: CI0FE0 110: CI1FE1 111：保留 从模式被使能后这些位不能改。
3	保留	必须保持复位值
2:0	SMC[2:0]	从模式控制 000：关闭从模式. 如果 CEN=1，则预分频器直接由内部时钟驱动。 001：保留。 010：保留。 011：保留。 100：复位模式。选中的触发输入的上升沿重新初始化计数器，并且产生更新事件。 101：暂停模式。当触发输入为高时，计数器的时钟开启。一旦触发输入变为低，则计数器时钟停止。 110：事件模式。计数器在触发输入的上升沿启动。 111：外部时钟模式 0。选中的触发输入的上升沿驱动计数器。 由于 CI0F_ED 是一个脉冲波形，而暂停模式是检测触发信号的电平，所以，当 CI0F_ED 用作触发输入时，暂停模式必须禁能。

DMA 和中断使能寄存器 (TIMERx_DMAINTEN)

地址偏移：0x0C
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

保留

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TRGDEN	保留			CH1DEN	CH0DEN	UPDEN	BRKIE	TRGIE	CMTIE	保留		CH1IE	CH0IE	UPIE
	rw				rw	rw	rw	rw	rw	rw			rw	rw	rw

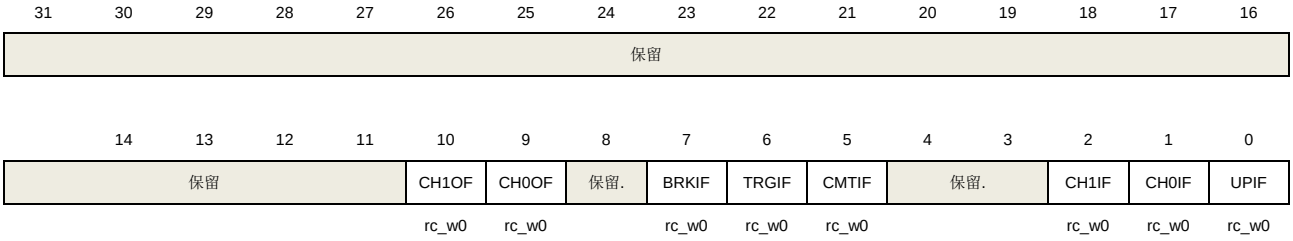
位/位域	名称	描述
31:15	保留	必须保持复位值。
14	TRGDEN	触发 DMA 请求使能 0：禁止触发 DMA 请求 1：使能触发 DMA 请求
13:11	保留	必须保持复位值。
10	CH1DEN	通道 1 比较/捕获 DMA 请求使能 0：禁止通道 1 比较/捕获 DMA 请求 1：使能通道 1 比较/捕获 DMA 请求
9	CH0DEN	通道 0 比较/捕获 DMA 请求使能 0：禁止通道 0 比较/捕获 DMA 请求 1：使能通道 0 比较/捕获 DMA 请求
8	UPDEN	更新 DMA 请求使能 0：禁止更新 DMA 请求 1：使能更新 DMA 请求
7	BRKIE	中止中断使能 0：禁止中止中断 1：使能中止中断
6	TRGIE	触发中断使能 0：禁止触发中断 1：使能触发中断
5	CMTIE	换相更新中断使能 0：禁止换相更新中断 1：使能换相更新中断
4:3	保留	必须保持复位值。
2	CH1IE	通道 1 比较/捕获中断使能 0：禁止通道 1 中断 1：使能通道 1 中断
1	CH0IE	通道 0 比较/捕获中断使能 0：禁止通道 0 中断 1：使能通道 0 中断
0	UPIE	更新中断使能

0：禁止更新中断
1：使能更新中断

中断标志寄存器 (TIMERx_INTF)

地址偏移：0x10
复位值：0x0000 0000

该寄存器只能按字(32位)访问。



位/位域	名称	描述
31:11	保留	必须保持复位值。
10	CH1OF	通道 1 捕获溢出标志 参见 CH0OF 描述
9	CH0OF	通道 0 捕获溢出标志 当通道 0 被配置为输入模式时，在 CH0IF 标志位已经被置 1 后，捕获事件再次发生时，该标志位可以由硬件置 1。该标志位由软件清 0. 0：无捕获溢出中断发生 1：发生了捕获溢出中断
8	保留	必须保持复位值。
7	BRKIF	中止中断标志位 当中止输入有效时，由硬件对该位置‘1’。 当中止输入无效时，则该位可由软件清‘0’。 0：无中止事件产生 1：中止输入上检测到有效电平
6	TRGIF	触发中断标志 当发生触发事件时，此标志会置 1，此位由软件清 0。当暂停模式使能时，触发输入的任意边沿都可以产生触发事件。否则，其它模式时，仅在触发输入端检测到有效边沿，产生触发事件。 0：无触发事件产生 1：触发中断产生
5	CMTIF	通道换相更新中断标志 当通道换相更新事件发生时此标志位被硬件置 1，此位由软件清 0。 0：无通道换相更新中断发生

		1：通道换相更新中断发生
4:3	保留	必须保持复位值。
2	CH1IF	通道 1 比较/捕获中断标志 参见 CH0IF 描述
1	CH0IF	通道 0 比较/捕获中断标志 此标志由硬件置 1 软件清 0。当通道 0 在输入模式下时，捕获事件发生时此标志位被置 1；当通道 0 在输出模式下时，此标志位在一个比较事件发生时被置 1。 当通道 0 在输入模式下时，读 TIMERx_CH0CV 会将此标志清零。 0：无通道 0 中断发生 1：通道 0 中断发生
0	UPIF	更新中断标志 此位在任何更新事件发生时由硬件置 1，软件清 0。 0：无更新中断发生 1：发生更新中断

软件事件产生寄存器 (TIMERx_SWEVG)

地址偏移：0x14
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								BRKG	TRGG	CMTG	保留		CH1G	CH0G	UPG
								w	w	w			w	w	w

位/位域	名称	描述
31:8	保留	必须保持复位值。
7	BRKG	产生中止事件 该位由软件置 1，用于产生一个中止事件，由硬件自动清 0。当此位被置 1 时，POEN 位被清 0 且 BRKIF 位被置 1，若开启对应的中断和 DMA，则产生相应的中断和 DMA 传输。 0：不产生中止事件 1：产生中止事件
6	TRGG	触发事件产生 此位由软件置 1，由硬件自动清 0. 当此位被置 1，TIMERx_INTF 寄存器的 TRGIF 标志位被置 1，若开启对应的中断和 DMA，则产生相应的中断和 DMA 传输。 0：无触发事件产生

1：产生触发事件

5	CMTG	通道换相更新事件发生 此位由软件置 1，由硬件自动清 0. 当此位被置 1，通道捕获/比较控制寄存器(CHxEN, CHxNEN 和 CHxCOMCTL) 的互补输出被更新（根据 TIMERx_CTL1 中 CCSE 值）。 0：不产生通道控制更新事件 1：产生通道控制更新事件
4:3	保留	必须保持复位值。
2	CH1G	通道 1 捕获或比较事件发生 参见 CH0G 描述
1	CH0G	通道 0 捕获或比较事件发生 该位由软件置 1，用于在通道 0 产生一个捕获/比较事件，由硬件自动清 0。当此位被置 1，CH0IF 标志位被置 1，若开启对应的中断和 DMA，则发出相应的中断和 DMA 请求。此外，如果通道 0 配置为输入模式，计数器的当前值被 TIMERx_CH0CV 寄存器捕获，如果 CH0IF 标志位已经为 1，则 CH0OF 标志位被置 1。 0：不产生通道 0 捕获或比较事件 1：发生通道 0 捕获或比较事件
0	UPG	更新事件产生 此位由软件置 1，被硬件自动清 0。当此位被置 1，如果选择了中央对齐或向上计数模式，计数器被清 0。否则(向下计数模式)计数器将载入自动重载值，预分频计数器将同时被清除。 0：无更新事件产生 1：产生更新事件

通道控制寄存器 0 (TIMERx_CHCTL0)

地址偏移：0x18
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	CH1COMCTL[2:0]			CH1COMSEN	CH1COMFEN	CH1MS[1:0]		保留	CH0COMCTL[2:0]			CH0COMSEN	CH0COMFEN	CH0MS[1:0]	
CH1CAPFLT[3:0]				CH1CAPPSC[1:0]				CH0CAPFLT[3:0]				CH0CAPPSC[1:0]			
rw				rw		rw		rw				rw		rw	

输出比较模式：

位/位域	名称	描述
31:15	保留	必须保持复位值。

14:12	CH1COMCTL[2:0]	通道 1 输出比较模式 参见 CH0COMCTL 描述
11	CH1COMSEN	通道 1 输出比较影子寄存器使能 参见 CH0COMSEN 描述
10	CH1COMFEN	通道 1 输出比较快速使能 参见 CH0COMFEN 描述
9:8	CH1MS[1:0]	通道 1 模式选择 这些位定义了通道的方向和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2 寄存器的 CH1EN 位被清 0)时这些位才可以写。 00：通道 1 配置为输出 01：通道 1 配置为输入，IS1 映射在 CI1FE1 上 10：通道 1 配置为输入，IS1 映射在 CI0FE1 上 11：通道 1 配置为输入，IS1 映射在 ITS 上 注意：当 CH1MS[1:0]=11 时，需要通过 TRGS 位（位于 TIMERx_SMCFG 寄存器）选择内部触发输入。
7	保留	必须保持复位值。
6:4	CH0COMCTL[2:0]	通道 0 输出比较模式 此位定义了输出准备信号 O0CPRE 的输出比较模式，而 O0CPRE 决定了 CH0_O、CH0_ON 的值。另外，O0CPRE 高电平有效，而 CH0_O、CH0_ON 通道的极性取决于 CH0P、CH0NP 位。 000：时基。输出比较寄存器 TIMERx_CH0CV 与计数器 TIMERx_CNT 间的比较对 O0CPRE 不起作用 001：匹配时设置为高。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 为高。 010：匹配时设置为低。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 为低。 011：匹配时翻转。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 翻转。 100：强制为低。强制 O0CPRE 为低电平 101：强制为高。强制 O0CPRE 为高电平 110：PWM 模式 0。在向上计数时，一旦计数器值小于 TIMERx_CH0CV 时，O0CPRE 为高电平，否则为低电平。在向下计数时，一旦计数器的值大于 TIMERx_CH0CV 时，O0CPRE 为低电平，否则为高电平。 111：PWM 模式 1。在向上计数时，一旦计数器值小于 TIMERx_CH0CV 时，O0CPRE 为低电平，否则为高电平。在向下计数时，一旦计数器的值大于 TIMERx_CH0CV 时，O0CPRE 为高电平，否则为低电平。 如果配置在 PWM 模式下，只有当输出比较模式从时基模式变为 PWM 模式或者比较结果改变时，O0CPRE 电平才改变。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 且 CH0MS =00（比较模式）时此位不能被改变。
3	CH0COMSEN	通道 0 输出比较影子寄存器使能

当此位被置 1，TIMERx_CH0CV 寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。
0：禁止通道 0 输出/比较影子寄存器
1：使能通道 0 输出/比较影子寄存器
仅在单脉冲模式下(SPM =1)，可以在未确认影子寄存器的情况下使用 PWM 模式
当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 且 CH0MS =00 时此位不能被改变。

2	CH0COMFEN	通道 0 输出比较快速使能 当该位为 1 时，如果通道配置为 PWM0 模式或者 PWM1 模式，会加快捕获/比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，CH0_O 被设置为比较电平而与比较结果无关。 0：禁止通道 0 输出比较快速. 1：使能通道 0 输出比较快速。
1:0	CH0MS[1:0]	通道 0 I/O 模式选择 这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2 寄存器的 CH0EN 位被清 0)时这些位才可写。 00：通道 0 配置为输出 01：通道 0 配置为输入，IS0 映射在 CI0FE0 上 10：通道 0 配置为输入，IS0 映射在 CI1FE0 上 11：通道 0 配置为输入，IS0 映射在 ITS 上. 注意：当 CH0MS[1:0]=11 时，需要通过 TRGS 位（位于 TIMERx_SMCFG 寄存器）选择内部触发输入。

输入捕获模式：

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:12	CH1CAPFLT[3:0]	通道 1 输入捕获滤波控制 参见 CH0CAPFLT 描述
11:10	CH1CAPPSC[1:0]	通道 1 输入捕获预分频器 参见 CH0CAPPSC 描述
9:8	CH1MS[1:0]	通道 1 模式选择 与输出模式相同
7:4	CH0CAPFLT[3:0]	通道 0 输入捕获滤波控制 CI0 输入信号可以通过数字滤波器进行滤波，该位域配置滤波参数。 数字滤波器的基本原理：根据 f_{SAMP} 对 CI0 输入信号进行连续采样，并记录信号相同电平的次数。达到该位配置的滤波参数后，认为是有效电平。 滤波器参数配置如下：

CH0CAPFLT [3:0]	采样次数	f_{SAMP}
4’b0000	无滤波器	
4’b0001	2	f_{CK_TIMER}
4’b0010	4	

4’b0011	8	
4’b0100	6	$f_{DTS}/2$
4’b0101	8	
4’b0110	6	$f_{DTS}/4$
4’b0111	8	
4’b1000	6	$f_{DTS}/8$
4’b1001	8	
4’b1010	5	$f_{DTS}/16$
4’b1011	6	
4’b1100	8	
4’b1101	5	$f_{DTS}/32$
4’b1110	6	
4’b1111	8	

3:2 CH0CAPPSC[1:0] 通道 0 输入捕获预分频器

这 2 位定义了通道 0 输入的预分频系数。当 TIMERx_CHCTL2 寄存器中的 CH0EN =0 时，则预分频器复位。

00：无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获

01：每 2 个事件触发一次捕获

10：每 4 个事件触发一次捕获

11：每 8 个事件触发一次捕获

1:0 CH0MS[1:0] 通道 0 模式选择

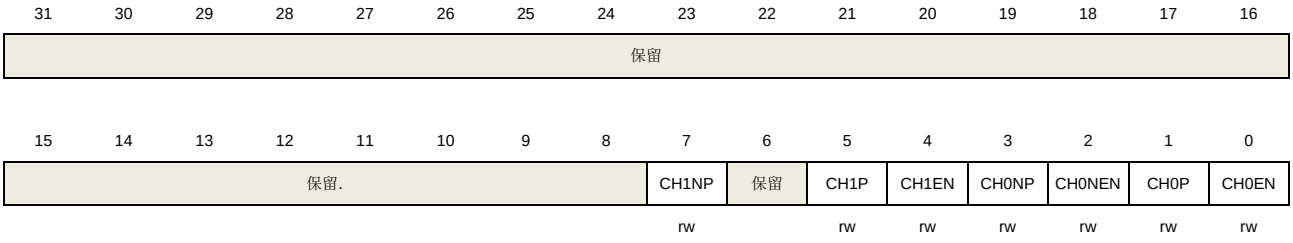
与输出比较模式相同

通道控制寄存器 2 (TIMERx_CHCTL2)

地址偏移：0x20

复位值：0x0000 0000

该寄存器只能按字(32位)访问。



位/位域	名称	描述
31:8	保留	必须保持复位值。
7	CH1NP	通道 1 互补输出极性 参考 CH0NP 描述
6	保留	必须保持复位值。
5	CH1P	通道 1 极性

参考 CH0P 描述

4	CH1EN	通道 1 使能 参考 CH0EN 描述
3	CH0NP	通道 0 互补输出极性 当通道 0 配置为输出模式，此位定义了互补输出信号的极性。 0：通道 0 高电平有效 1：通道 0 低电平有效 当通道 0 配置为输入模式时，此位和 CH0P 联合使用，作为输入信号 CI0 的极性选择控制信号。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 或 10 时此位不能被更改。
2	CH0NEN	通道 0 互补输出使能 当通道 0 配置为输出模式时，将此位置 1 使能通道 0 的互补输出。 0：禁止通道 0 互补输出 1：使能通道 0 互补输出
1	CH0P	通道 0 极性 当通道 0 配置为输出模式时，此位定义了输出信号极性。 0：通道 0 高电平有效 1：通道 0 低电平有效 当通道 0 配置为输入模式时，此位定义了 CI0 信号极性 [CH0NP, CH0P] 将选择 CI0FE0 或者 CI1FE0 的有效边沿或者捕获极性 [CH0NP==0, CH0P==0]：把 CIxFE0 的上升沿作为捕获或者从模式下触发的有效信号，并且 CIxFE0 不会被翻转。 [CH0NP==0, CH0P==1]：把 CIxFE0 的下降沿作为捕获或者从模式下触发的有效信号，并且 CIxFE0 会被翻转。 [CH0NP==1, CH0P==0]：保留。 [CH0NP==1, CH0P==1]：把 CIxFE0 的上升沿和下降沿都作为捕获或者从模式下触发的有效信号，并且 CIxFE0 不会被翻转。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 或 10 时此位不能被更改。
0	CH0EN	通道 0 捕获/比较使能 当通道 0 配置为输出模式时，将此位置 1 使能 CH0_O 信号有效。当通道 0 配置为输入模式时，将此位置 1 使能通道 0 上的捕获事件。 0：禁止通道 0 1：使能通道 0

计数器寄存器 (TIMERx_CNT)

地址偏移：0x24
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CNT[15:0]	这些位是当前的计数值。写操作能改变计数器值。

预分频寄存器 (TIMERx_PSC)

地址偏移：0x28
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	PSC[15:0]	计数器时钟预分频值 计数器时钟等于 TIMER_CK 时钟除以(PSC+1)，每次当更新事件产生时，PSC 的值被装入到对应的影子寄存器。

计数器自动重载寄存器 (TIMERx_CAR)

地址偏移：0x2C
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CARL[15:0]															
rw															

位/位域	名称	描述

31:16	保留	必须保持复位值。
15:0	CARL[15:0]	计数器自动重载值 这些位定义了计数器的自动重载值。

重复计数寄存器 (TIMERx_CREP)

地址偏移：0x30
复位值：0x0000 0000

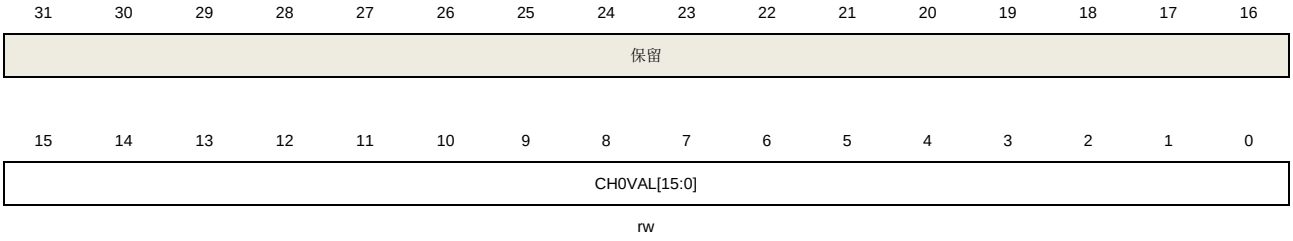
该寄存器只能按字(32位)访问。

位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	CREP[7:0]	重复计数器的值 这些位定义了更新事件的产生速率。重复计数器计数值减为 0 时产生更新事件。影子寄存器的更新速率也会受这些位影响(前提是影子寄存器被使能)。

通道 0 捕获/比较寄存器 (TIMERx_CH0CV)

地址偏移：0x34
复位值：0x0000 0000

该寄存器只能按字(32位)访问。



位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CH0VAL[15:0]	通道 0 的捕获或比较值 当通道 0 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 0 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄

存器后，影子寄存器值随每次更新事件更新。

通道 1 捕获/比较寄存器 (TIMERx_CH1CV)

地址偏移：0x38
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1VAL[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CH1VAL[15:0]	通道 1 的捕获或比较值 当通道 1 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 1 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

互补通道保护寄存器 (TIMERx_CCHP)

地址偏移：0x44
复位值: 0x0000 0000

该寄存器通过字访问（32位）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POEN	OAEN	BRKP	BRKEN	ROS	IOS	PROT[1:0]		DTCFG[7:0]							
rw	rw	rw	rw	rw	rw	rw		rw							

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	POEN	所有的通道输出使能 该位通过以下方式置 1： –写 1 置位 –如果 OAEN=1，则在下一次更新事件发生时置 1. 该位通过以下方式清 0： –写 0 清 0

–有效的中止输入（异步）
如果一个通道配置为输出模式，如果设置了相应的使能位（TIMERx_CHCTL2 寄存器的 CHxEN, CHxNEN 位），则开启 CHx_O 和 CHx_ON 输出。
0：禁止通道输出。
1：使能通道输出。

14 OAEN
自动输出使能
0：POEN 位只能使用软件方式置 1。
1：如果中止输入无效，下一次更新事件发生时，POEN 位将会置 1。
此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0] =00 时才可修改。

13 BRKP
中止极性
此位定义了中止输入信号 BRKIN 的极性。
0：中止输入低电平有效。
1：中止输入高电平有效。

12 BRKEN
中止使能
此位置 1 使能中止事件和 CCS 时钟失败事件输入。
0：禁能中止输入。
1：使能中止输入。
此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0] =00 时才可修改。

11 ROS
运行模式下“关闭状态”使能
当 POEN 位被置 1（运行模式），此位可以被置 1 来使能通道(带有互补输出且配置为输出模式)的输出“关闭状态”。
0：输出“关闭状态”禁能。当 CHxEN 或者 CHxNEN 位被清零，对应通道为输出“禁能状态”。
1：输出“关闭状态”使能。当 CHxEN 或者 CHxNEN 位被清零，对应通道为输出“关闭状态”。
此位在 TIMERx_CCHP 寄存器的 PROT [1:0]=10 或 11 时不能被更改。

10 IOS
空闲模式下“关闭状态”使能
当 POEN 位被清 0（空闲模式），此位可以被置 1 来使能通道(带有互补输出且配置为输出模式)的输出“关闭状态”。
0：输出“关闭状态”禁能。当 CHxEN 和 CHxNEN 位均被清零，对应通道为输出“禁能状态”。
1：输出“关闭状态”使能。不论 CHxEN 和 CHxNEN 位的值，对应通道为输出“关闭状态”。
此位在 TIMERx_CCHP 寄存器的 PROT [1:0]=10 或 11 时不能被更改。

9:8 PROT[1:0]
互补寄存器保护控制
这两位定义了寄存器的写保护特性。
00：禁能保护模式。无写保护。
01：PROT 模式 0。TIMERx_CTL1 寄存器中 ISOx/ISOxN 位，TIMERx_CCHP 寄存器中 BRKEN/BRKP/OAEN/DTCFG 位写保护。
10：PROT 模式 1。除了 PROT 模式 0 下的寄存器写保护外，还有 TIMERx_CHCTL2 寄存器中 CHxP/CHxNP 位（如果相应通道配置为输出模式），TIMERx_CCHP 寄

存器中 ROS/IOS 位。
11：PROT 模式 2.。除了 PROT 模式 1 下的寄存器写保护外，还有 TIMERx_CHCTL0 中 CHxCOMCTL/ CHxCOMSEN 位（如果相关通道配置为输出模式）写保护。
系统复位后这两位只能被写一次，一旦 TIMERx_CCHP 寄存器被写入，这两位被写保护。

7:0	DTCFG[7:0]	死区时间控制 这些位定义了插入互补输出之间的死区持续时间。DTCFG 值和死区时间的关系如下： DTCFG [7:5] =3’b0xx：DTvalue =DTCFG [7:0]x tDT, tDT=tDTS. DTCFG [7:5] =3’b 10x：DTvalue = （64+DTCFG [5:0]）xtDT, tDT =tDTS*2. DTCFG [7:5] =3’b 110：DTvalue = （32+DTCFG [4:0]）xtDT, tDT=tDTS*8. DTCFG [7:5] =3’b 111：DTvalue = （32+DTCFG [4:0]）xtDT, tDT =tDTS*16. 此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0]=00 时才可修改。

DMA 配置寄存器 (TIMERx_DMACFG)

地址偏移：0x48
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

位/位域	名称	描述
31:13	保留	必须保持复位值。
12:8	DMATC [4:0]	DMA 传输计数 该位域定义了 DMA 访问（读写）TIMERx_DMATB 寄存器的数量 n，n = (DMATC [4:0] +1). DMATC [4:0] 从 5’b0_0000 到 5’b1_0001.
7:5	保留	必须保持复位值。
4:0	DMATA [4:0]	DMA 传输起始地址 该位域定义了 DMA 访问 TIMERx_DMATB 寄存器的第一个地址。当通过 TIMERx_DMA 第一次访问时，访问的就是该位域指定的地址。第二次访问 TIMERx_DMATB 时，将访问起始地址+0x4。

DMA 发送缓冲区寄存器 (TIMERx_DMATB)

地址偏移：0x4C
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMATB[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	DMATB [15:0]	DMA 发送缓冲 对这个寄存器的读或写，（起始地址+传输次数*4）地址范围内的寄存器会被访问 传输次数由硬件计算，范围为 0 到 DMATC。

15.5. 通用定时器 L4（TIMERx,x=15,16）

15.5.1. 简介

通用定时器 L4（TIMER15/16）是单通道定时器，支持输入捕获和输出比较。可以产生 PWM 信号控制电机和电源管理。通用定时器 L4 含有一个 16 位无符号计数器。

通用定时器 L4 是可编程的，可以被用来计数，其外部事件可以驱动其他定时器

通用定时器 L4 包含了一个死区时间插入模块，非常适合电机控制。

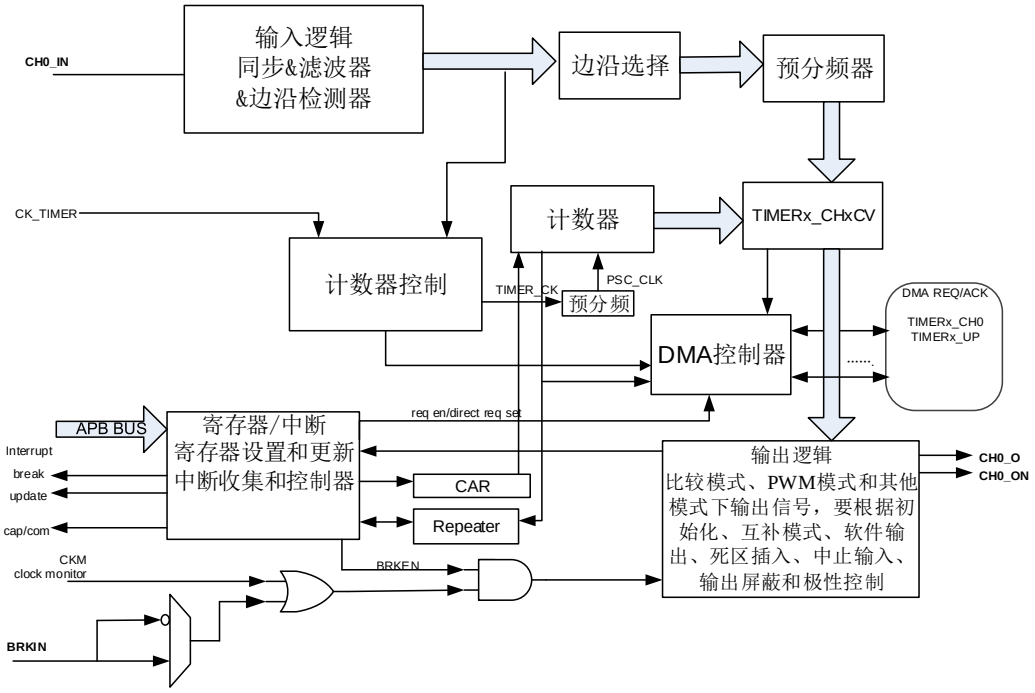
15.5.2. 主要特性

- 总通道数：1；
- 计数器宽度：16位；
- 时钟源可选：内部时钟；
- 计数模式：向上计数；
- 可编程的预分频器：16位，运行时可以被改变；
- 每个通道可配置：输入捕获模式，输出比较模式，可编程的PWM模式，单脉冲模式；
- 可编程的死区时间；
- 自动重装载功能；
- 可编程的计数器重复功能；
- 中止输入功能；
- 中断输出和DMA请求：更新事件，比较/捕获事件，换相事件和中止事件；

15.5.3. 结构框图

*图 15-70. 通用定时器 L4 结构框图*提供了通用定时器 L4 的内部配置细节

图 15-70. 通用定时器 L4 结构框图



15.5.4. 功能描述

时钟源配置

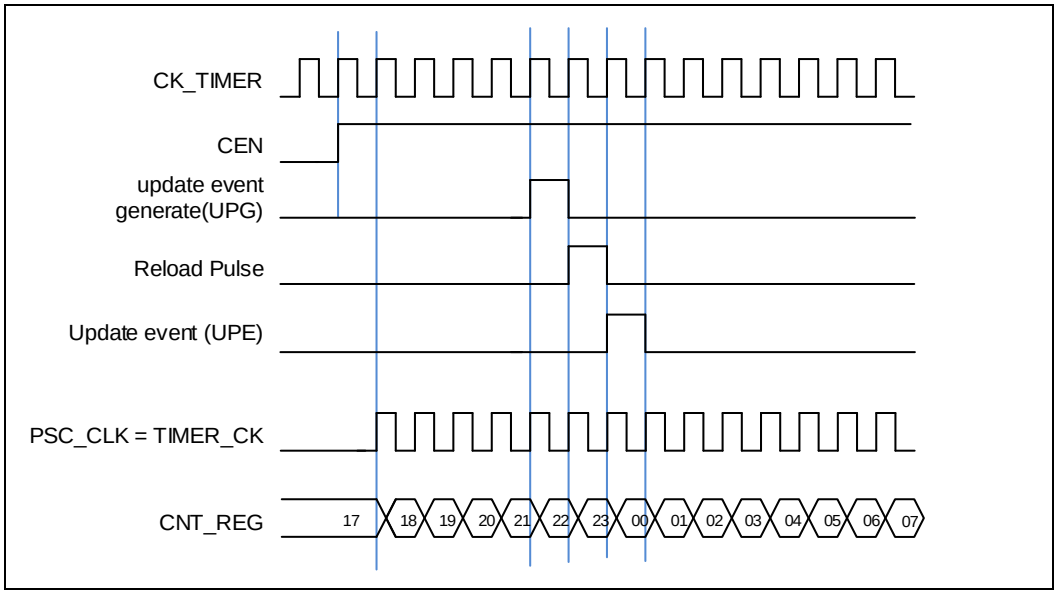
通用定时器 L4 由内部时钟源 CK_TIMER.

■ 定时器选择内部时钟源（连接到RCU模块的CK_TIMER）

通用定时器 L4 只有一个时钟源：内部时钟源。用来驱动计数器预分频器的是内部时钟源 CK_TIMER。当 CEN 置位，CK_TIMER 经过预分频器（预分频值由 TIMERx_PSC 寄存器确定）产生 PSC_CLK。

驱动预分频器计数的 TIMER_CK 等于来自于 RCU 模块的 CK_TIMER

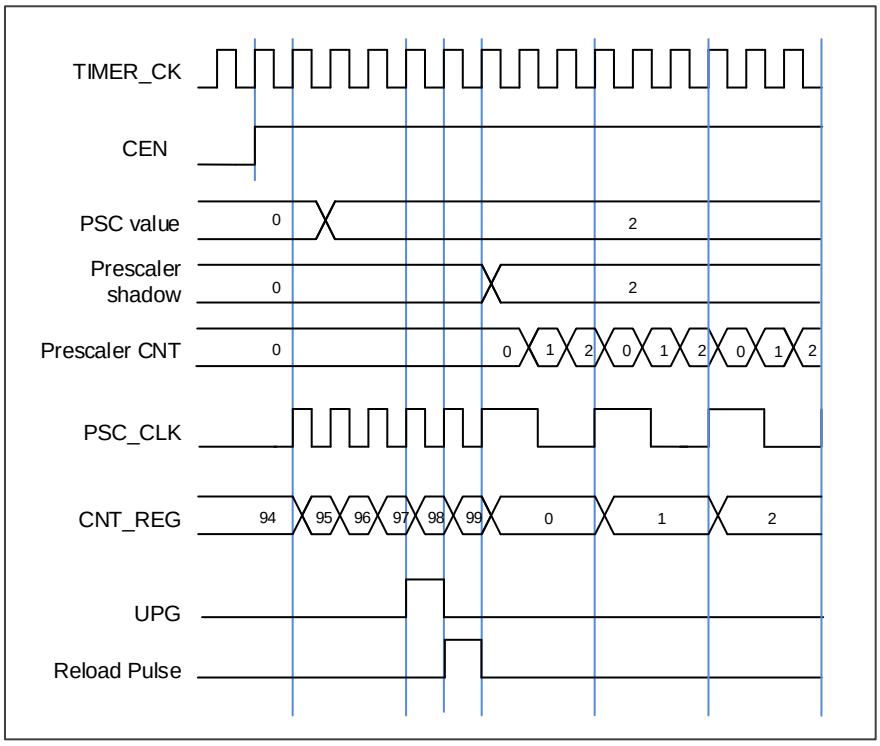
图 15-71. 内部时钟分频为 1 时，计数器的时序图



时钟预分频器

预分频器可以将定时器的时钟（TIMER_CK)频率按 1 到 65536 之间的任意值分频，分频后的时钟 PSC_CLK 驱动计数器计数。分频系数受预分频寄存器 TIMERx_PSC 控制，这个控制寄存器带有缓冲器，它能够在运行时被改变。新的预分频器的参数在下一次更新事件到来时被采用。

图 15-72. 当 PSC 数值从 0 变到 2 时，计数器的时序图



计数器向上计数模式

在这种模式，计数器的计数方向是向上计数。计数器从0开始向上连续计数到计数器自动加载值（定义在TIMERx_CAR寄存器中），一旦计数器计数到自动加载值，会重新从0开始向上计数，并且产生上溢事件。另外，在(TIMERx_CREP+1)次上溢后将会产生更新事件。在向上计数模式中，TIMERx_CTL0寄存器中的计数方向控制位DIR应该被设置成0。

当通过 TIMERx_SWEVG 寄存器的 UPG 位置 1 来设置更新事件时，计数值会被清 0，并产生更新事件。

如果 TIMERx_CTL0 寄存器的 UPDIS 置 1，则禁止更新事件。

当发生更新事件时，所有影子寄存器（重复计数器，计数器自动重载寄存器，预分频寄存器）都将被更新。

图 15-73. 向上计数时序图，PSC=0/2 给出了一些例子，当 TIMERx_CAR=0x99 时，计数器在不同预分频因子下的行为。

图 15-73. 向上计数时序图，PSC=0/2

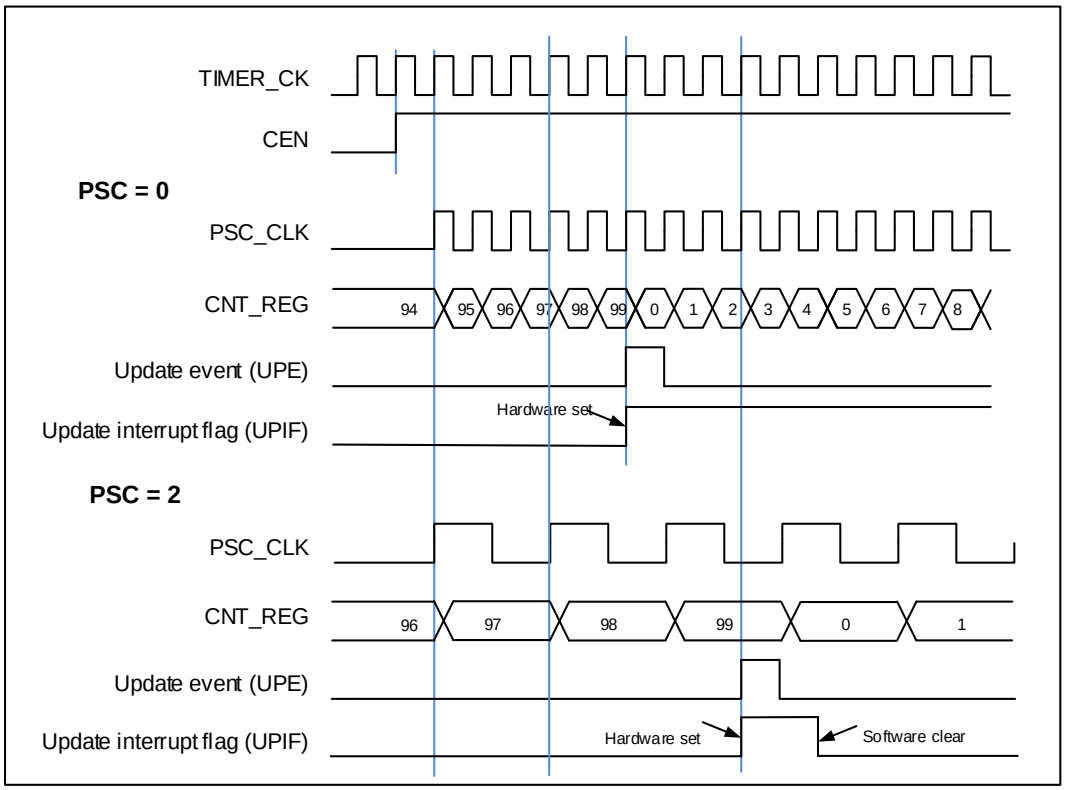
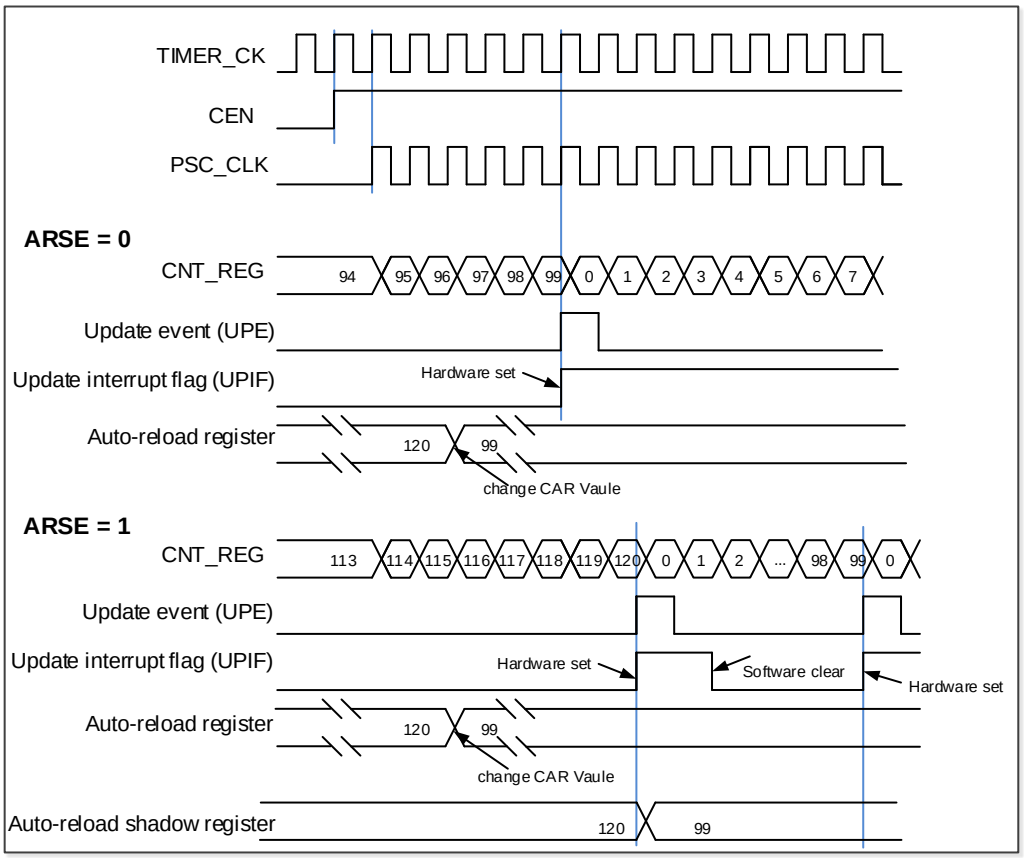


图 15-74. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值

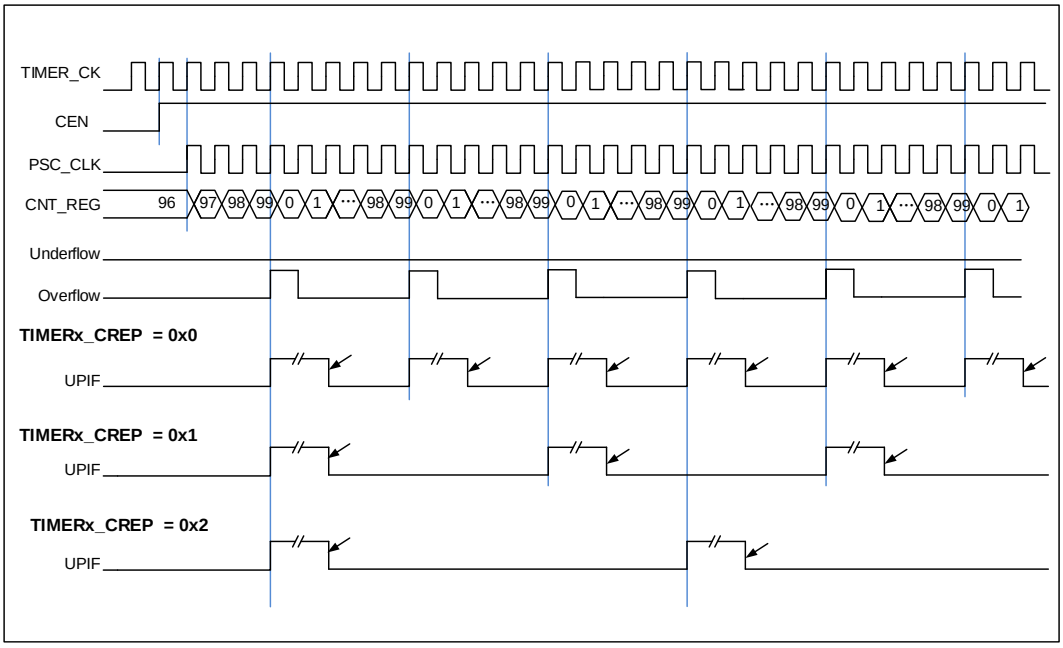


更新事件（来自上溢/下溢）频率配置

重复计数器是用来在 N+1 个计数周期之后产生更新事件，更新定时器的寄存器，N 为 TIMERx_CREP 寄存器的 CREP。向上计数模式下，重复计数器在每次计数器上溢时递减。

将 TIMERx_SWEVG 寄存器的 UPG 位置 1 可以重载 TIMERx_CREP 寄存器中 CREP 的值并产生一个更新事件。

图 15-75. 在向上计数模式下计数器重复时序图



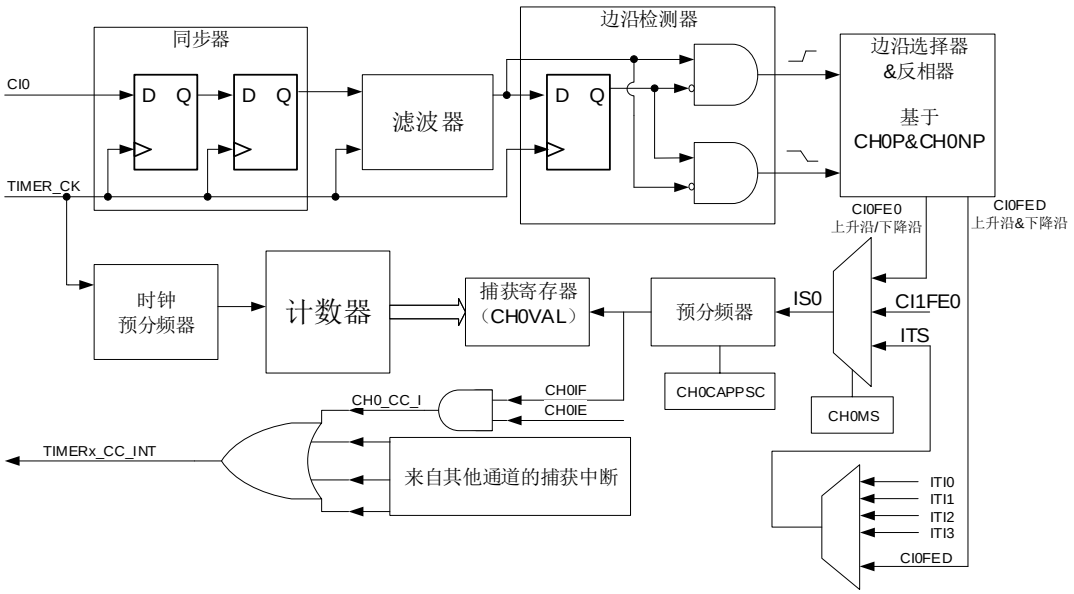
输入捕获和输出比较通道

通用定时器 L4 拥有一个独立的通道用于捕获输入或比较输出是否匹配。每个通道都围绕一个通道捕获比较寄存器建立，包括一个输入级，通道控制器和输出级。

通道输入捕获功能

通道输入捕获功能允许通道测量一个波形时序，频率，周期，占空比等。输入级包括一个数字滤波器，一个通道极性选择，边沿检测和一个通道预分频器。如果在输入引脚上出现被选择的边沿，TIMERx_CHxCV 寄存器会捕获计数器当前的值，同时 CHxIF 位被置 1，如果 CHxIE = 1 则产生通道中断。

图 15-76. 通道输入捕获原理



通道输入信号 CIx 有两种选择，一种是 TIMERx_CHx 信号，另一种是 TIMERx_CH0,TIMERx_CH1 和 TIMERx_CH2 异或之后的信号。通道输入信号 CIx 先被 TIMER_CK 信号同步，然后经过数字滤波器采样，产生一个被滤波后的信号。通过边沿检测器，可以选择检测上升沿或者下降沿。通过配置 CHxP 选择使用上升沿或者下降沿。配置 CHxMS.，可以选择其他通道的输入信号，内部触发信号。配置 IC 预分频器，使得若干个输入事件后才产生一个有效的捕获事件。捕获事件发生，CHxVAL 存储计数器的值。

配置步骤如下：

第一步：滤波器配置（TIMERx_CHCTL0寄存器中CHxCAPFLT）：
根据输入信号和请求信号的质量，配置相应的CHxCAPFLT。
第二步：边沿选择（TIMERx_CHCTL2寄存器中CHxP/CHxNP）：
配置CHxP/CHxNP选择上升沿或者下降沿。
第三步：捕获源选择（TIMERx_CHCTL0寄存器中CHxMS）：
一旦通过配置CHxMS选择输入捕获源，必须确保通道配置在输入模式（CHxMS!=0x0），而且TIMERx_CHxCV寄存器不能再被写。
第四步：中断使能（TIMERx_DMAINTEN寄存器中CHxIE和CHxDEN）：
使能相应中断，可以获得中断和DMA请求。
第五步：捕获使能（TIMERx_CHCTL2寄存器中CHxEN）。

结果：当期望的输入信号发生时，TIMERx_CHxCV被设置成当前计数器的值，CHxIF为置1。如果CHxIF位已经为1，则CHxOF位置1。根据TIMERx_DMAINTEN寄存器中CHxIE和CHxDEN的配置，相应的中断和DMA请求会被提出。

直接产生：软件设置CHxG位，会直接产生中断和DMA请求。

通道输出比较功能

在通道输出比较功能，TIMERx 可以产生时控脉冲，其位置，极性，持续时间和频率都是可编程的。当一个输出通道的 CHxVAL 寄存器与计数器的值匹配时，根据 CHxCOMCTL 的配置，这个通道的输出可以被置高电平，被置低电平或者反转。当计数器的值与 CHxVAL 寄存器的值匹配时，CHxIF 位被置 1，如果 CHxIE = 1 则会产生中断，如果 CHxDEN=1 则会产生 DMA 请求。

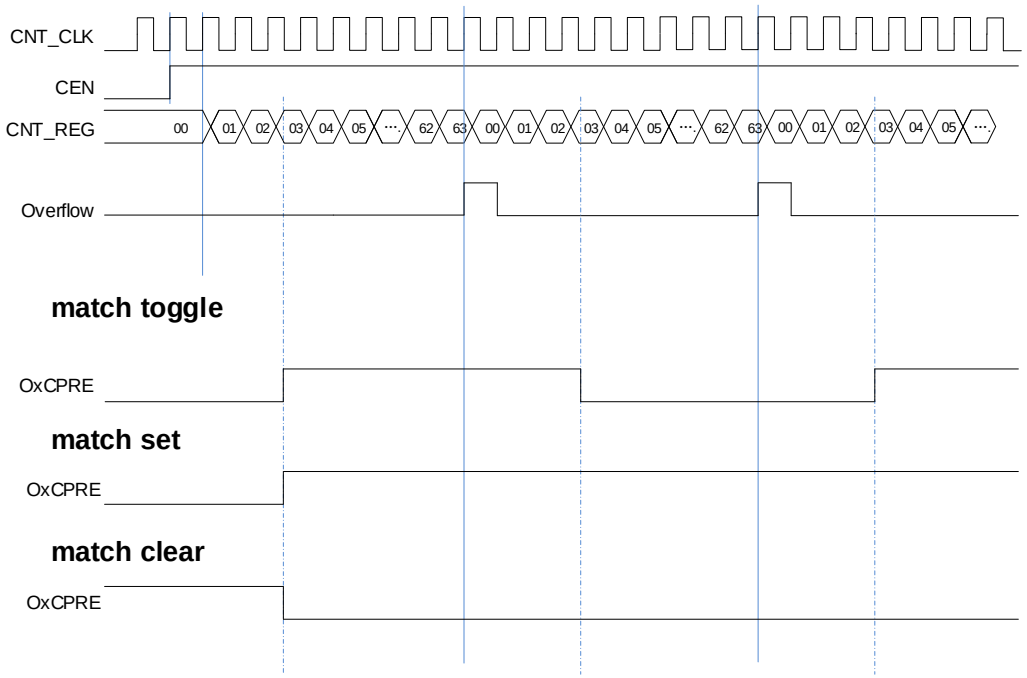
配置步骤如下：

第一步：时钟配置：
配置定时器时钟源，预分频器等。
第二步：比较模式配置：
设置CHxCOMSEN位来配置输出比较影子寄存器；
设置CHxCOMCTL位来配置输出模式（置高电平/置低电平/反转）；
设置CHxP/CHxNP位来选择有效电平的极性；
设置CHxEN使能输出。
第三步：通过CHxIE/CHxDEN位配置中断/DMA请求使能。
第四步：通过TIMERx_CAR寄存器和TIMERx_CHxCV寄存器配置输出比较时基：
CHxVAL可以在运行时根据你所期望的波形而改变。
第五步：设置CEN位使能定时器。

*图 15-77. 三种输出比较模式*显示了三种比较输出模式：反转/置高电平/置低电平，CAR=0x63,

CHxVAL=0x3。

图 15-77. 三种输出比较模式



输出 PWM 功能

在 PWM 输出模式下（PWM 模式 0 是配置 CHxCOMCTL 为 3'b110，PWM 模式 1 是配置 CHxCOMCTL 为 3'b111），通道根据 TIMERx_CAR 寄存器和 TIMERx_CHxCV 寄存器的值，输出 PWM 波形。

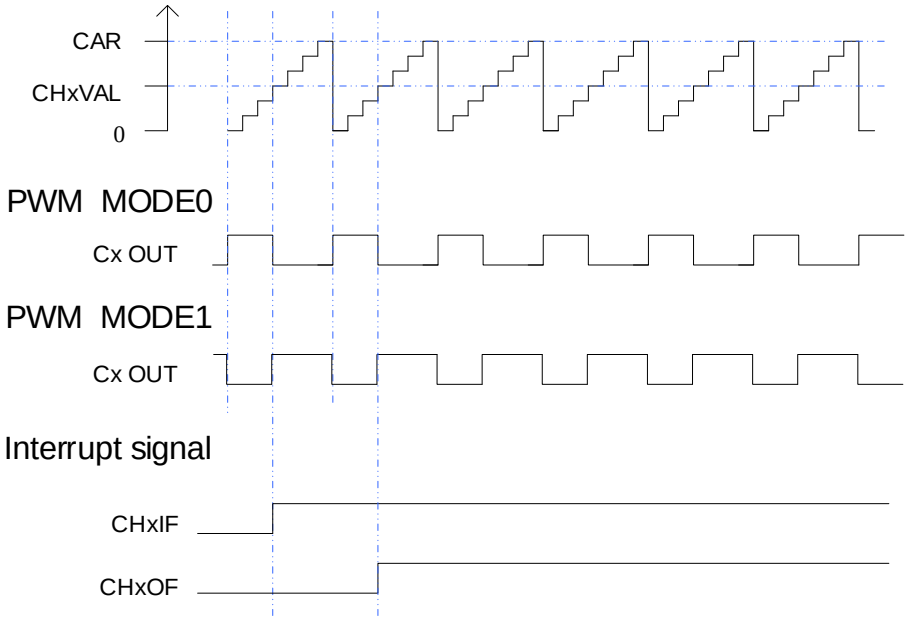
根据计数模式，我们可以分为两种 PWM 波：EAPWM(边沿对齐 PWM)和 CAPWM(中央对齐 PWM)。

EAPWM 的周期由 TIMERx_CAR 寄存器值决定，占空比由 TIMERx_CHxCV 寄存器值决定。*图 15-78. PWM 时序图*显示了 PWM 的输出波形和中断。

在 PWM0 模式下(CHxCOMCTL==3'b110)，如果 TIMERx_CHxCV 寄存器的值大于 TIMERx_CAR 寄存器的值，通道输出一直为有效电平。

在 PWM0 模式下(CHxCOMCTL==3'b110)，如果 TIMERx_CHxCV 寄存器的值等于 0，通道输出一直为无效电平。

图 15-78. PWM 时序图



通道输出参考信号

当 TIMERx 用于输出匹配比较模式下，设置 CHxCOMCTL 位可以定义 OxCPRE 信号(通道 x 准备信号)类型。OxCPRE 信号有若干类型的输出功能，包括，设置 CHxCOMCTL=0x00 可以保持原始电平；设置 CHxCOMCTL=0x01 可以将 OxCPRE 信号设置为高电平；设置 CHxCOMCTL=0x02 可以将 OxCPRE 信号设置为低电平；设置 CHxCOMCTL=0x03，在计数器值和 TIMERx_CHxCV 寄存器的值匹配时，可以翻转输出信号。

PWM 模式 0 和 PWM 模式 1 是 OxCPRE 的另一种输出类型，设置 CHxCOMCTL 位域位 0x06 或 0x07 可以配置 PWM 模式 0/PWM 模式 1。在这些模式中，根据计数器值和 TIMERx_CHxCV 寄存器值的关系以及计数方向，OxCPRE 信号改变其电平。具体细节描述，请参考相应的位。

设置 CHxCOMCTL =0x04 或 0x05 可以实现 OxCPRE 信号的强制输出功能。输出比较信号能够直接由软件强置为有效或无效状态，而不依赖于 TIMERx_CHxCV 的值和计数器值之间间的比较结果。

通道输出互补 PWM

CHx_O 和 CHx_ON 是一对互补输出通道，这两个信号不能同时有效。TIMERx 有两路通道，只有一路有互补输出通道。互补信号 CHx_O 和 CHx_ON 是由一组参数来决定：TIMERx_CHCTL2 寄存器中的 CHxEN 和 CHxNEN 位，TIMERx_CCHP 寄存器中和 TIMERx_CTL1 寄存器中的 POEN, ROS, IOS, ISOx 和 ISOxN 位。输出极性由 TIMERx_CHCTL2 寄存器中的 CHxP 和 CHxNP 位来决定。

表 15-10. 由参数控制的互补输出表

<table>
<tr><th colspan="5">互补参数</th><th colspan="2">输出状态</th></tr>
<tr><th>POEN</th><th>ROS</th><th>IOS</th><th>CHxEN</th><th>CHxNEN</th><th>CHx_O</th><th>CHx_ON</th></tr>
<tr><td rowspan="4">0</td><td rowspan="4">0/1</td><td rowspan="3">0</td><td rowspan="2">0</td><td>0</td><td colspan="2">CHx_O / CHx_ON = LOW
CHx_O / CHx_ON 输出禁能(1)</td></tr>
<tr><td>1</td><td colspan="2" rowspan="3">CHx_O/CHx_ON输出关闭状态(2)：
通道先输出无效电平：CHx_O = CHxP，CHx_ON = CHxNP）；如果死区产生时钟未失效，在死区时间之后：
CHx_O = ISOx，CHx_ON = ISOxN (3)</td></tr>
<tr><td rowspan="2">1</td><td>0</td></tr>
<tr><td>1</td><td>x</td><td>x</td></tr>
<tr><td colspan="5">CHx_O/CHx_ON输出关闭状态：
通道先输出无效电平：CHx_O = CHxP，CHx_ON = CHxNP）；如果死区产生时钟未失效，在死区时间之后：
CHx_O = ISOx，CHx_ON = ISOxN</td></tr>
<tr><td rowspan="8">1</td><td rowspan="4">0</td><td rowspan="4">0/1</td><td rowspan="2">0</td><td>0</td><td colspan="2">CHx_O/CHx_ON = LOW
CHx_O/CHx_ON输出禁能</td></tr>
<tr><td>1</td><td>CHx_O = LOW
CHx_O输出禁能</td><td>CHx_ON=OxCPRE⊕(4)CHxNP
CHx_ON输出使能</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td>CHx_O=OxCPRE⊕CHxP
CHx_O输出使能</td><td>CHx_ON = LOW
CHx_ON输出禁能</td></tr>
<tr><td>1</td><td>CHx_O=OxCPRE⊕CHxP
CHx_O输出使能</td><td>CHx_ON=(!OxCPRE)(5)⊕CHxNP
CHx_ON输出使能</td></tr>
<tr><td rowspan="4">1</td><td rowspan="2">0</td><td>0</td><td>CHx_O = CHxP
CHx_O输出关闭状态</td><td>CHx_ON = CHxNP
CHx_ON输出关闭状态</td></tr>
<tr><td>1</td><td>CHx_O = CHxP
CHx_O输出关闭状态</td><td>CHx_O=OxCPRE⊕CHxNP
CHx_ON输出使能</td></tr>
<tr><td rowspan="2">1</td><td rowspan="2">1</td><td>0</td><td>CHx_O=OxCPRE⊕CHxP
CHx_O输出使能</td><td>CHx_ON = CHxNP
CHx_ON输出关闭状态</td></tr>
<tr><td>1</td><td>CHx_O=OxCPRE⊕CHxP
CHx_O输出使能</td><td>CHx_ON=（!OxCPRE）⊕CHxNP
CHx_ON输出使能</td></tr>
</table>

注意：

（11） 输出禁能：CHx_O / CHx_ON 输出与对应引脚断开，对应引脚电平受 GPIO 上下拉配置控制，无上下拉时为悬空高阻态；

（12） 输出关闭状态：CHx_O / CHx_ON 输出无效电平（CHx_O = 0⊕CHxP = CHxP）；

（13） 详情见中止模式章节。

（14） ⊕：异或操作；

（15） (!OxCPRE)：OxCPRE 信号的互补信号。

互补 PWM 插入死区时间

设置 CHxEN 和 CHxNEN 为 1'b1 同时设置 POEN，死区插入就会被使能。DTCFG 位域定义了死区时间，死区时间对通道 0 有效。死区时间的细节，请参考 TIMERx_CCHP 寄存器。

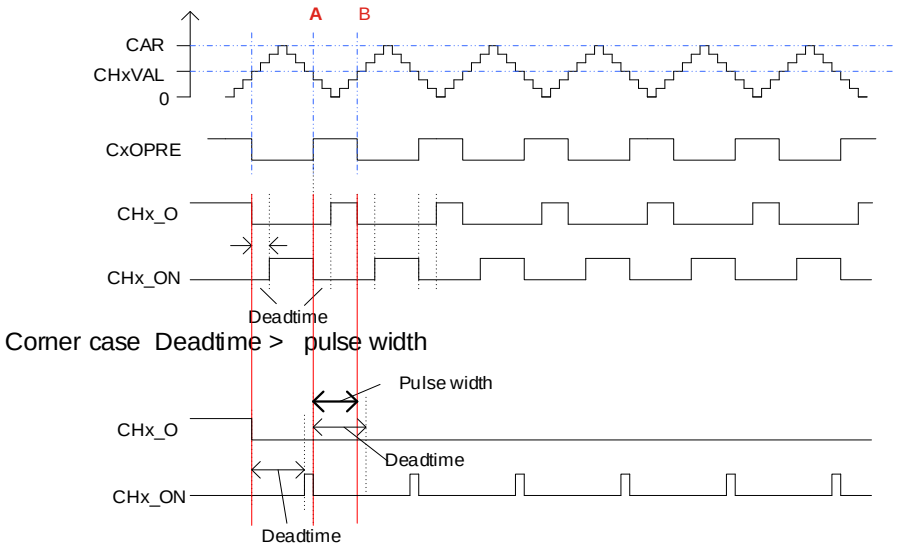
死区时间的插入，确保了通道互补的两路信号不会同时有效。

在 PWM0 模式，当通道 x 匹配发生时（TIMERx 计数器= CHxVAL），OxCPRE 反转。在*图 15-79. 带死区时间的互补输出*中的 A 点，CHx_O 信号在死区时间内为低电平，直到死区时间过后才变为高电平，而 CHx_ON 信号立刻变为低电平。同样，在 B 点，计数器再次匹配（TIMERx 计数器= CHxVAL），OxCPRE 信号被清 0，CHx_O 信号被立即清零，CHx_ON 信号在死区时间内仍然是低电平，在死区时间过后才变为高电平。

有时会有一些死角事件发生，例如：

- 如果死区延时大于或者等于 CHx_O 信号的占空比，CHx_O 信号一直为无效值（如*图 15-79. 带死区时间的互补输出*）。
- 如果死区延时大于或者等于 CHx_ON 信号的占空比，CHx_ON 信号一直为无效值。

图 15-79. 带死区时间的互补输出



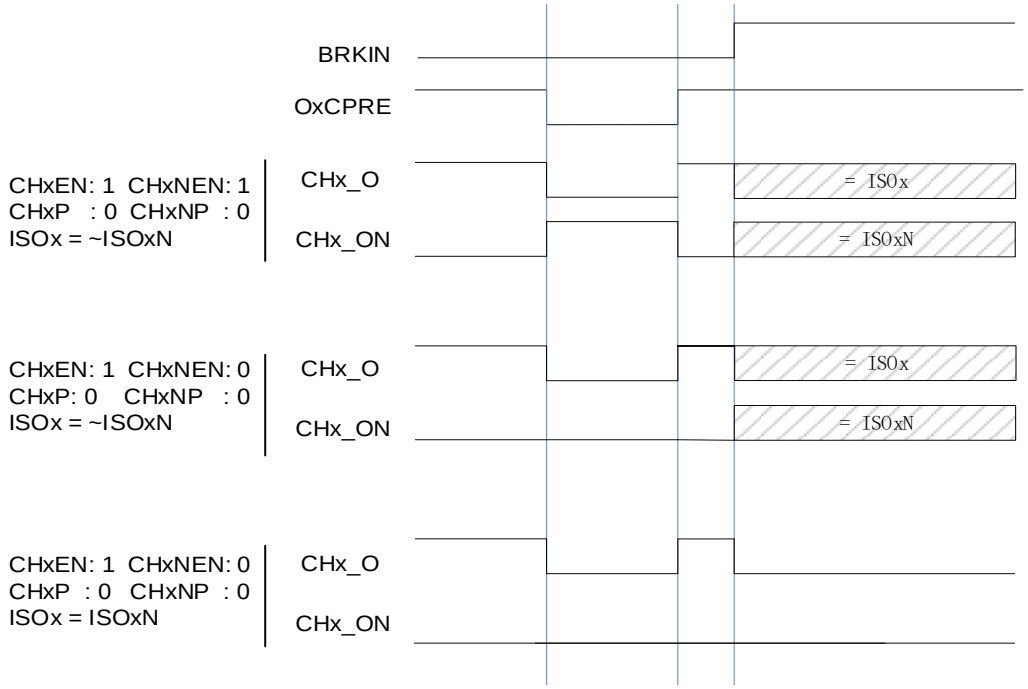
中止模式

使用模式时，输出 CHx_O 和 CHx_ON 信号电平被以下位控制，TIMERx_CCHP 寄存器的 POEN, IOS 和 ROS 位，TIMERx_CTL1 寄存器的 ISOx 和 ISOxN 位。当中止事件发生时，CHx_O 和 CHx_ON 信号输出不能同时设置为有效电平。中止源可以选择中止输入引脚，也可以选择 HXTAL 时钟失效事件。时钟失败事件由 RCU 中的时钟监视器(CKM)产生。将 TIMERx_CCHP 寄存器的 BRKEN 位置 1 可以使能中止功能。TIMERx_CCHP 寄存器的 BRKP 位决定了中止输入极性。

发生中止时，POEN 位被异步清除，一旦 POEN 位为 0，CHx_O 和 CHx_ON 被 TIMERx_CTL1 寄存器中的 ISOx 位和 ISOxN 驱动。如果 IOS=0，定时器释放输出使能，否则输出使能仍然为高。起初互补输出被置于复位状态，然后死区时间产生器重新被激活，以便在一个死区时间后驱动输出，输出电平由 ISOx 和 ISOxN 位配置。

发生中止时，TIMERx_INTF 寄存器的 BRKIF 位被置 1。如果 BRKIE=1，中断产生。

图 15-80. 通道响应中止输入（高电平有效）时，输出信号的行为



单脉冲模式

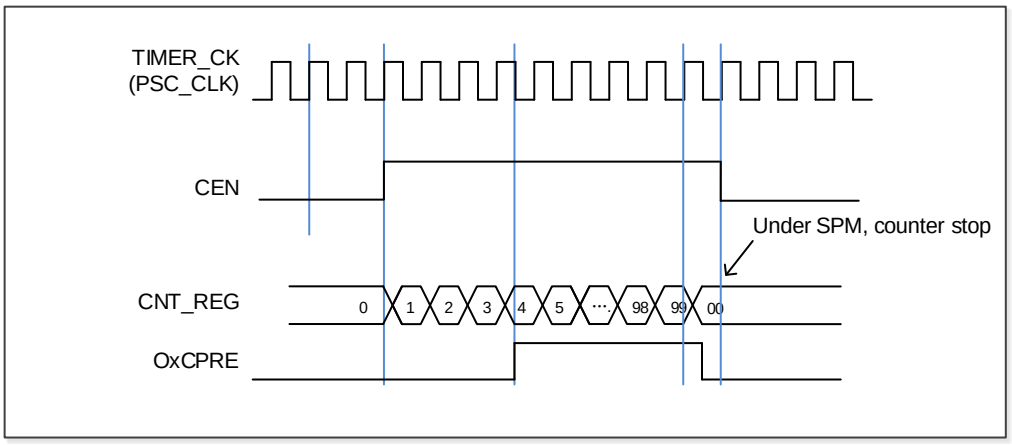
单脉冲模式与重复模式是相反的，设置 TIMERx_CTL0 寄存器的 SPM 位置 1，则使能单脉冲模式。当 SPM 置 1，计数器在下次更新事件到来后清零并停止计数。为了得到脉冲波，可以通过设置 CHxCOMCTL 配置 TIMERx 为 PWM 模式或者比较模式。

一旦设置定时器运行在单脉冲模式下，需要设置 TIMERx_CTL0 寄存器的定时器使能位 CEN=1 来使能计数器。软件写 CEN=1 都可以产生一个脉冲，此后 CEN 位一直保持为 1 直到更新事件发生或者 CEN 位被软件写 0。如果 CEN 位被软件清 0，计数器停止工作，计数值被保持。

在单脉冲模式下，有效的外部触发边沿会将 CEN 位置 1，使能计数器。然而，执行计数值和 TIMERx_CHxCV 寄存器值的比较结果依然存在一些时钟延迟。为了最大限度减少延迟，用户可以将 TIMERx_CHCTL0/1 寄存器的 CHxCOMFEN 位置 1。单脉冲模式下，触发上升沿产生之后， OxCPRE 信号将被立即强制转换为与发生比较匹配时相同的电平，但是不用考虑比较结果。只有输出通道配置为 PWM0 或 PWM1 输出运行模式下时 CHxCOMFEN 位才可用，触发源来源于触发信号。

图 15-81. 单脉冲模式，TIMERx_CHxCV = 4 TIMERx_CAR=99 展示了一个例子。

图 15-81. 单脉冲模式，TIMERx_CHxCV = 4 TIMERx_CAR=99



定时器 DMA 模式

定时器 DMA 模式是指通过 DMA 模块配置定时器的寄存器。有两个跟定时器 DMA 模式相关的寄存器：TIMERx_DMACFG 和 TIMERx_DMATB。当然，必须要使能 DMA 请求，一些内部中断事件可以产生 DMA 请求。当中断事件发生，TIMERx 会给 DMA 发送请求。DMA 配置成 M2P 模式，PADDR 是 TIMERx_DMATB 寄存器地址，DMA 就会访问 TIMERx_DMATB 寄存器。实际上，TIMERx_DMATB 寄存器只是一个缓冲，定时器会将 TIMERx_DMATB 映射到一个内部寄存器，这个内部寄存器由 TIMERx_DMACFG 寄存器中的 DMATA 来指定。如果 TIMERx_DMACFG 寄存器的 DMATC 位域值为 0，表示 1 次传输，定时器的发送 1 个 DMA 请求就可以完成。如果 TIMERx_DMACFG 寄存器的 DMATC 位域值不为 1，例如其值为 3，表示 4 次传输，定时器就需要再多发 3 次 DMA 请求。在这 3 次请求下，DMA 对 TIMERx_DMATB 寄存器的访问会映射到访问定时器的 DMATA+0x4, DMATA+0x8, DMATA+0xc 寄存器。总之，发生一次 DMA 内部中断请求，定时器会连续发送（DMATC+1）次请求。

如果再来 1 次 DMA 请求事件，TIMERx 将会重复上面的过程。

定时器调试模式

当 Cortex®-M3 内核停止，DBG_CTL1 寄存器中的 TIMERx_HOLD 配置位被置 1，定时器计数器停止。

15.5.5. TIMERx 寄存器(x=15,16)

TIMER15 基地址：0x4001 4400
TIMER16 基地址：0x4001 4800

控制寄存器 0 (TIMERx_CTL0)

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						CKDIV[1:0]		ARSE	保留			SPM	UPS	UPDIS	CEN
						rw		rw				rw	rw	rw	rw

位/位域	名称	描述
31:10	保留	必须保持复位值。
9:8	CKDIV[1:0]	时钟分频 通过软件配置CKDIV，规定定时器时钟(CK_TIMER) 与死区时间和数字滤波器采样时钟(DTS)之间的分频系数。 00：$f_{DTS}=f_{CK_TIMER}$ 01：$f_{DTS}= f_{CK_TIMER}$ /2 10：$f_{DTS}= f_{CK_TIMER}$ /4 11：保留
7	ARSE	自动重载影子使能 0：禁能 TIMERx_CAR 寄存器的影子寄存器。 1：使能 TIMERx_CAR 寄存器的影子寄存器。
6:4	保留	必须保持复位值。
3	SPM	单脉冲模式 0：单脉冲模式禁能。更新事件发生后，计数器继续计数 1：单脉冲模式使能。在下一次更新事件发生时，计数器停止计数
2	UPS	更新请求源 软件配置该位，选择更新事件源. 0：以下事件均会产生更新中断或DMA请求： UPG位被置1 计数器溢出/下溢 复位模式产生的更新 1：下列事件会产生更新中断或DMA请求：

		计数器溢出/下溢
1	UPDIS	禁止更新. 该位用来使能或禁能更新事件的产生 0：更新事件使能. 更新事件发生时，相应的影子寄存器被装入预装载值，以下事件均会产生更新事件： UPG位被置1 计数器溢出/下溢 复位模式产生的更新 1：更新事件禁能. 注意：当该位被置 1 时，UPG 位被置 1 或者复位模式不会产生更新事件，但是计数器和预分频器被重新初始化
0	CEN	计数器使能 0：计数器禁能 1：计数器使能 在软件将 CEN 位置 1 后，外部时钟、暂停模式和正交译码器模式才能工作。

控制寄存器 1 (TIMERx_CTL1)

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						ISO0N	ISO0	保留				DMAS	CCUC	保留	CCSE
						rw	rw					rw	rw		rw

位/位域	名称	描述
31:10	保留	必须保持复位值。
9	ISO0N	通道 0 的互补通道空闲状态输出 0：当 POEN 复位，CH0_ON 设置低电平. 1：当 POEN 复位，CH0_ON 设置高电平 此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0]位为 00 的时候可以被更改.
8	ISO0	通道 0 的空闲状态输出 0：当 POEN 复位，CH0_O 设置低电平 1：当 POEN 复位，CH0_O 设置高电平 如果 CH0_ON 生效，一个死区时间后 CH0_O 输出改变。此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0]位为 00 的时候可以被更改.
7:4	保留	必须保持复位值。

3	DMAS	DMA请求源选择 0：当通道捕获/比较事件发生时，发送通道 x 的 DMA 请求 . 1：当更新事件发生，发送通道 x 的 DMA 请求
2	CCUC	换相控制影子寄存器更新控制 当换相控制影子寄存器（CHxEN, CHxNEN 和 CHxCOMCTL 位）使能(CCSE=1)，这些影子寄存器更新控制如下： 0：CMTG 位被置 1 时更新影子寄存器 1：当 CMTG 位被置 1 或检测到 TRIGI 上升沿时，影子寄存器更新 当通道没有互补输出时，此位无效。
1	保留	必须保持复位值。.
0	CCSE	换相控制影子使能 0：影子寄存器 CHxEN, CHxNEN 和 CHxCOMCTL 位禁能. 1：影子寄存器 CHxEN, CHxNEN 和 CHxCOMCTL 位使能. 如果这些位已经被写入了，换相事件到来时这些位才被更新 当通道没有互补输出时，此位无效

DMA 和中断使能寄存器 (TIMERx_DMAINTEN)

地址偏移：0x0C
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						CH0DEN	UPDEN	BRKIE	保留	CMTIE	保留			CH0IE	UPIE
						rw	rw	rw		rw				rw	rw

位/位域	名称	描述
31:10	保留	必须保持复位值。
9	CH0DEN	通道 0 比较/捕获 DMA 请求使能 0：禁止通道 0 比较/捕获 DMA 请求 1：使能通道 0 比较/捕获 DMA 请求
8	UPDEN	更新 DMA 请求使能 0：禁止更新 DMA 请求 1：使能更新 DMA 请求
7	BRKIE	中止中断使能 0：禁止中止中断 1：使能中止中断

6	保留	必须保持复位值。
5	CMTIE	换相更新中断使能 0：禁止换相更新中断 1：使能换相更新中断
4:2	保留	必须保持复位值。
1	CH0IE	通道 0 比较/捕获中断使能 0：禁止通道 0 中断 1：使能通道 0 中断
0	UPIE	更新中断使能 0：禁止更新中断 1：使能更新中断

中断标志寄存器 (TIMERx_INTF)

地址偏移：0x10
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						CH0OF	保留.	BRKIF	保留	CMTIF	保留.			CH0IF	UPIF
						rc_w0		rc_w0		rc_w0				rc_w0	rc_w0

位/位域	名称	描述
31:10	保留	必须保持复位值。
9	CH0OF	通道 0 捕获溢出标志 当通道 0 被配置为输入模式时，在 CH0IF 标志位已经被置 1 后，捕获事件再次发生时，该标志位可以由硬件置 1。该标志位由软件清 0. 0：无捕获溢出中断发生 1：发生了捕获溢出中断
8	保留	必须保持复位值。
7	BRKIF	中止中断标志位 当中止输入有效时，由硬件对该位置‘1’。 当中止输入无效时，则该位可由软件清‘0’。 0：无中止事件产生 1：中止输入上检测到有效电平
6	保留	必须保持复位值。

5	CMTIF	通道换相更新中断标志 当通道换相更新事件发生时此标志位被硬件置 1，此位由软件清 0。 0：无通道换相更新中断发生 1：通道换相更新中断发生
4:2	保留	必须保持复位值。
1	CH0IF	通道 0 比较/捕获中断标志 此标志由硬件置 1 软件清 0。当通道 0 在输入模式下时，捕获事件发生时此标志位被置 1；当通道 0 在输出模式下时，此标志位在一个比较事件发生时被置 1。 当通道 0 在输入模式下时，读 TIMERx_CH0CV 会将此标志清零。 0：无通道 0 中断发生 1：通道 0 中断发生
0	UPIF	更新中断标志 此位在任何更新事件发生时由硬件置 1，软件清 0。 0：无更新中断发生 1：发生更新中断

软件事件产生寄存器 (TIMERx_SWEVG)

地址偏移：0x14
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								BRKG	保留	CMTG	保留			CH0G	UPG
								w		w				w	w

位/位域	名称	描述
31:8	保留	必须保持复位值。
7	BRKG	产生中止事件 该位由软件置 1，用于产生一个中止事件，由硬件自动清 0。当此位被置 1 时，POEN 位被清 0 且 BRKIF 位被置 1，若开启对应的中断和 DMA，则产生相应的中断和 DMA 传输。 0：不产生中止事件 1：产生中止事件
6	保留	必须保持复位值。
5	CMTG	通道换相更新事件发生 此位由软件置 1，由硬件自动清 0. 当此位被置 1，通道捕获/比较控制寄存器 (CHxEN, CHxNEN 和 CHxCOMCTL) 的互补输出被更新（根据 TIMERx_CTL1 中

CCSE 值）。
0：不产生通道控制更新事件
1：产生通道控制更新事件

4:2	保留	必须保持复位值。
1	CH0G	通道 0 捕获或比较事件发生 该位由软件置 1，用于在通道 0 产生一个捕获/比较事件，由硬件自动清 0。当此位被置 1，CH0IF 标志位被置 1，若开启对应的中断和 DMA，则发出相应的中断和 DMA 请求。此外，如果通道 0 配置为输入模式，计数器的当前值被 TIMERx_CH0CV 寄存器捕获，如果 CH0IF 标志位已经为 1，则 CH0OF 标志位被置 1。 0：不产生通道 0 捕获或比较事件 1：发生通道 0 捕获或比较事件
0	UPG	更新事件产生 此位由软件置 1，被硬件自动清 0。当此位被置 1，如果选择了中央对齐或向上计数模式，计数器被清 0。否则(向下计数模式)计数器将载入自动重载值，预分频计数器将同时被清除。 0：无更新事件产生 1：产生更新事件

通道控制寄存器 0 (TIMERx_CHCTL0)

地址偏移：0x18
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								保留	CH0COMCTL[2:0]			CH0COMSEN	保留	CH0MS[1:0]	
								CH0CAPFLT[3:0]				CH0CAPPSC[1:0]			
									rw			rw		rw	

输出比较模式：

位/位域	名称	描述
31:7	保留	必须保持复位值。
6:4	CH0COMCTL[2:0]	通道 0 输出比较模式 此位定义了输出准备信号 O0CPRE 的输出比较模式，而 O0CPRE 决定了 CH0_O、CH0_ON 的值。另外，O0CPRE 高电平有效，而 CH0_O、CH0_ON 通道的极性取决于 CH0P、CH0NP 位。 000：时基。输出比较寄存器 TIMERx_CH0CV 与计数器 TIMERx_CNT 间的比较对 O0CPRE 不起作用 001：匹配时设置为高。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同

时，强制 O0CPRE 为高。
010：匹配时设置为低。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 为低。
011：匹配时翻转。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 翻转。
100：强制为低。强制 O0CPRE 为低电平
101：强制为高。强制 O0CPRE 为高电平
110：PWM 模式 0。在向上计数时，一旦计数器值小于 TIMERx_CH0CV 时，O0CPRE 为高电平，否则为低电平。在向下计数时，一旦计数器的值大于 TIMERx_CH0CV 时，O0CPRE 为低电平，否则为高电平。
111：PWM 模式 1。在向上计数时，一旦计数器值小于 TIMERx_CH0CV 时，O0CPRE 为低电平，否则为高电平。在向下计数时，一旦计数器的值大于 TIMERx_CH0CV 时，O0CPRE 为高电平，否则为低电平。
如果配置在 PWM 模式下，只有当输出比较模式从时基模式变为 PWM 模式或者比较结果改变时，O0CPRE 电平才改变。
当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 且 CH0MS =00（比较模式）时此位不能被改变。

3	CH0COMSEN	通道 0 输出比较影子寄存器使能 当此位被置 1，TIMERx_CH0CV 寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。 0：禁止通道 0 输出/比较影子寄存器 1：使能通道 0 输出/比较影子寄存器 仅在单脉冲模式下(SPM =1)，可以在未确认影子寄存器的情况下使用 PWM 模式 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 且 CH0MS =00 时此位不能被改变。
2	保留	必须保持复位值。
1:0	CH0MS[1:0]	通道 0 I/O 模式选择 这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2 寄存器的 CH0EN 位被清 0)时这些位才可写。 00：通道 0 配置为输出 01：通道 0 配置为输入，IS0 映射在 CI0FE0 上 10：保留 11：保留

输入捕获模式：

位/位域	名称	描述
31:8	保留	必须保持复位值。
7:4	CH0CAPFLT[3:0]	通道 0 输入捕获滤波控制 CI0 输入信号可以通过数字滤波器进行滤波，该位域配置滤波参数。 数字滤波器的基本原理：根据 f_{SAMP} 对 CI0 输入信号进行连续采样，并记录信号相同电平的次数。达到该位配置的滤波参数后，认为是有效电平。 滤波器参数配置如下：

CH0CAPFLT [3:0]	采样次数	f_{SAMP}
4’b0000	无滤波器	
4’b0001	2	f_{CK_TIMER}
4’b0010	4	
4’b0011	8	
4’b0100	6	$f_{DTS}/2$
4’b0101	8	
4’b0110	6	$f_{DTS}/4$
4’b0111	8	
4’b1000	6	$f_{DTS}/8$
4’b1001	8	
4’b1010	5	$f_{DTS}/16$
4’b1011	6	
4’b1100	8	
4’b1101	5	$f_{DTS}/32$
4’b1110	6	
4’b1111	8	

3:2 CH0CAPPSC[1:0] 通道 0 输入捕获预分频器

这 2 位定义了通道 0 输入的预分频系数。当 TIMERx_CHCTL2 寄存器中的 CH0EN =0 时，则预分频器复位。

00：无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获

01：每 2 个事件触发一次捕获

10：每 4 个事件触发一次捕获

11：每 8 个事件触发一次捕获

1:0 CH0MS[1:0] 通道 0 模式选择

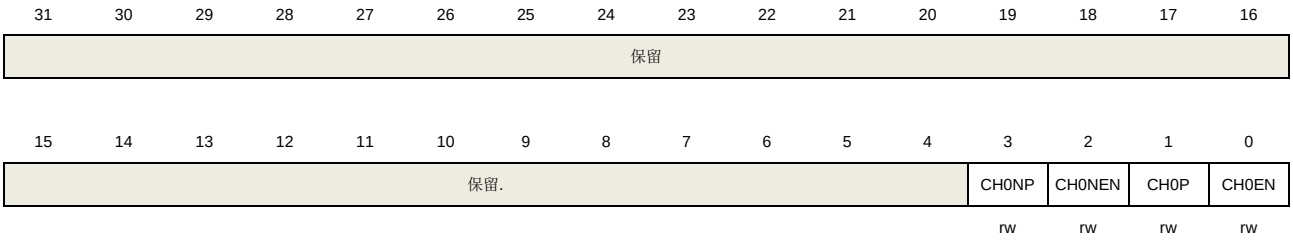
与输出比较模式相同

通道控制寄存器 2 (TIMERx_CHCTL2)

地址偏移：0x20

复位值：0x0000 0000

该寄存器只能按字(32位)访问。



位/位域	名称	描述
31:4	保留	必须保持复位值。

3	CH0NP	通道 0 互补输出极性 当通道 0 配置为输出模式，此位定义了互补输出信号的极性。 0：通道0互补输出高电平为有效电平 1：通道0互补输出低电平为有效电平 当通道 0 配置为输入模式时，此位和 CH0P 联合使用，作为输入信号 CI0 的极性选择控制信号。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 或 10 时此位不能被更改。
2	CH0NEN	通道 0 互补输出使能 当通道 0 配置为输出模式时，将此位置 1 使能通道 0 的互补输出。 0：禁止通道 0 互补输出 1：使能通道 0 互补输出
1	CH0P	通道 0 极性 当通道 0 配置为输出模式时，此位定义了输出信号极性。 0：通道0高电平为有效电平 1：通道0低电平为有效电平 当通道 0 配置为输入模式时，此位定义了 CI0 信号极性 [CH0NP, CH0P] 将选择 CI0FE0 或者 CI1FE0 的有效边沿或者捕获极性 [CH0NP==0, CH0P==0]：把 CIxFE0 的上升沿作为捕获或者从模式下触发的有效信号，并且 CIxFE0 不会被翻转。 [CH0NP==0, CH0P==1]：把 CIxFE0 的下降沿作为捕获或者从模式下触发的有效信号，并且 CIxFE0 会被翻转。 [CH0NP==1, CH0P==0]：保留。 [CH0NP==1, CH0P==1]：把 CIxFE0 的上升沿和下降沿都作为捕获或者从模式下触发的有效信号，并且 CIxFE0 不会被翻转。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 或 10 时此位不能被更改。
0	CH0EN	通道 0 捕获/比较使能 当通道 0 配置为输出模式时，将此位置 1 使能 CH0_O 信号有效。当通道 0 配置为输入模式时，将此位置 1 使能通道 0 上的捕获事件。 0：禁止通道 0 1：使能通道 0

计数器寄存器 (TIMERx_CNT)

地址偏移：0x24
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CNT[15:0]	这些位是当前的计数值。写操作能改变计数器值。

预分频寄存器 (TIMERx_PSC)

地址偏移： 0x28
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	PSC[15:0]	计数器时钟预分频值 计数器时钟等于 TIMER_CK 时钟除以(PSC+1)，每次当更新事件产生时，PSC 的值被装入到对应的影子寄存器。

计数器自动重载寄存器 (TIMERx_CAR)

地址偏移：0x2C
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CARL[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CARL[15:0]	计数器自动重载值 这些位定义了计数器的自动重载值。

重复计数寄存器 (TIMERx_CREP)

地址偏移：0x30
复位值：0x0000 0000

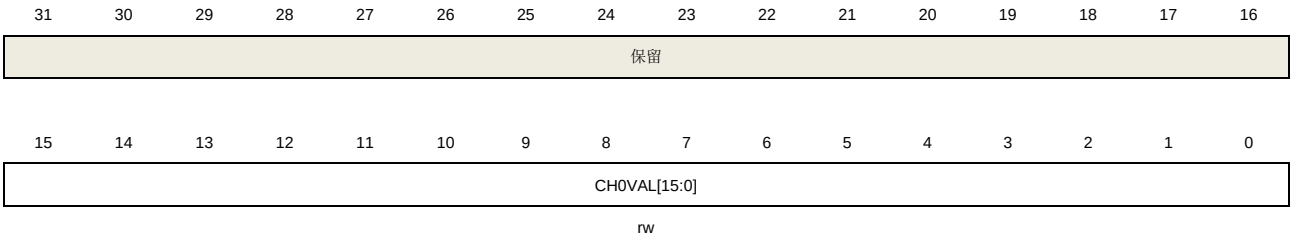
该寄存器只能按字(32位)访问。

位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	CREP[7:0]	重复计数器的值 这些位定义了更新事件的产生速率。重复计数器计数值减为 0 时产生更新事件。影子寄存器的更新速率也会受这些位影响(前提是影子寄存器被使能)。

通道 0 捕获/比较寄存器 (TIMERx_CH0CV)

地址偏移：0x34
复位值：0x0000 0000

该寄存器只能按字(32位)访问。



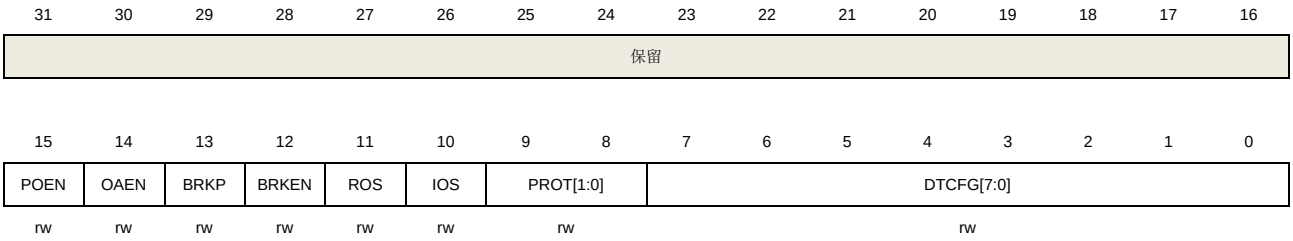
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CH0VAL[15:0]	通道 0 的捕获或比较值 当通道 0 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 0 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

互补通道保护寄存器 (TIMERx_CCHP)

地址偏移：0x44

复位值: 0x0000 0000

该寄存器通过字访问（32位）。



位/位域	名称	描述
31:16	保留	必须保持复位值。
15	POEN	所有的通道输出使能 该位通过以下方式置 1： –写 1 置位 –如果 OAEN=1，则在下一次更新事件发生时置 1. 该位通过以下方式清 0： –写 0 清 0 –有效的中止输入（异步） 如果一个通道配置为输出模式，如果设置了相应的使能位（TIMERx_CHCTL2 寄存器的 CHxEN, CHxNEN 位），则开启 CHx_O 和 CHx_ON 输出。 0：禁止通道输出 1：使能通道输出 注意：仅当 CHxMS[1:0]=2’b00 时该位有效
14	OAEN	自动输出使能 0：POEN 位只能使用软件方式置 1。 1：如果中止输入无效，下一次更新事件发生时，POEN 位将会置 1。 此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0] =00 时才可修改。
13	BRKP	中止极性 此位定义了中止输入信号 BRKIN 的极性。 0：中止输入低电平有效。 1：中止输入高电平有效。
12	BRKEN	中止使能 此位置 1 使能中止事件和 CCS 时钟失败事件输入。 0：禁能中止输入。 1：使能中止输入。 此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0] =00 时才可修改。
11	ROS	运行模式下“关闭状态”使能 当 POEN 位被置 1（运行模式），此位可以被置 1 来使能通道(带有互补输出且配置为输出模式)的输出“关闭状态”。 0：输出“关闭状态”禁能。当 CHxEN 或者 CHxNEN 位被清零，对应通道为输出“禁能状态”。 1：输出“关闭状态”使能。当 CHxEN 或者 CHxNEN 位被清零，对应通道为输出

“关闭状态”。
此位在 TIMERx_CCHP 寄存器的 PROT [1:0]=10 或 11 时不能被更改。

10 IOS
空闲模式下“关闭状态”使能
当 POEN 位被清 0（空闲模式），此位可以被置 1 来使能通道(带有互补输出且配置为输出模式)的输出“关闭状态”。
0：输出“关闭状态”禁能。当 CHxEN 和 CHxNEN 位均被清零，对应通道为输出“禁能状态”。
1：输出“关闭状态”使能。不论 CHxEN 和 CHxNEN 位的值，对应通道为输出“关闭状态”。
此位在 TIMERx_CCHP 寄存器的 PROT [1:0]=10 或 11 时不能被更改。

9:8 PROT[1:0]
互补寄存器保护控制
这两位定义了寄存器的写保护特性。
00：禁能保护模式。无写保护。
01：PROT 模式 0。TIMERx_CTL1 寄存器中 ISOx/ISOxN 位，TIMERx_CCHP 寄存器中 BRKEN/BRKP/OAEN/DTCFG 位写保护。
10：PROT 模式 1。除了 PROT 模式 0 下的寄存器写保护外，还有 TIMERx_CHCTL2 寄存器中 CHxP/CHxNP 位（如果相应通道配置为输出模式），TIMERx_CCHP 寄存器中 ROS/IOS 位。
11：PROT 模式 2.。除了 PROT 模式 1 下的寄存器写保护外，还有 TIMERx_CHCTL0 中 CHxCOMCTL/ CHxCOMSEN 位（如果相关通道配置为输出模式）写保护。
系统复位后这两位只能被写一次，一旦 TIMERx_CCHP 寄存器被写入，这两位被写保护。

7:0 DTCFG[7:0]
死区时间控制
DTCFG 值和死区时间的关系如下：

DTCFG[7:5]	The duration of dead-time
3’b0xx	DTCFG[7:0] * t_{DTS_CK}
3’b10x	(64+ DTCFG[5:0]) * t_{DTS_CK} *2
3’b110	(32+ DTCFG[4:0]) * t_{DTS_CK} *8
3’b111	(32+ DTCFG[4:0]) * t_{DTS_CK} *16

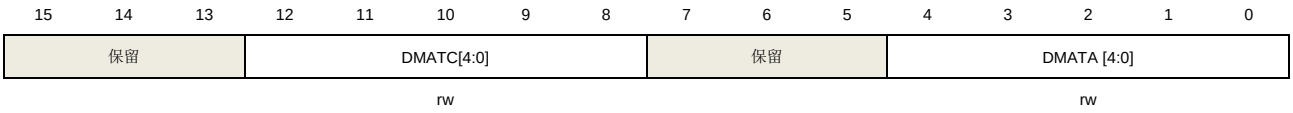
注意：
1. t_{DTS_CK} 是 DTS_CK 的周期，由 TIMERx_CTL0 中的 CKDIC[1:0]定义。
2. 此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0]=00 时才可修改。

DMA 配置寄存器 (TIMERx_DMACFG)

地址偏移：0x48
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

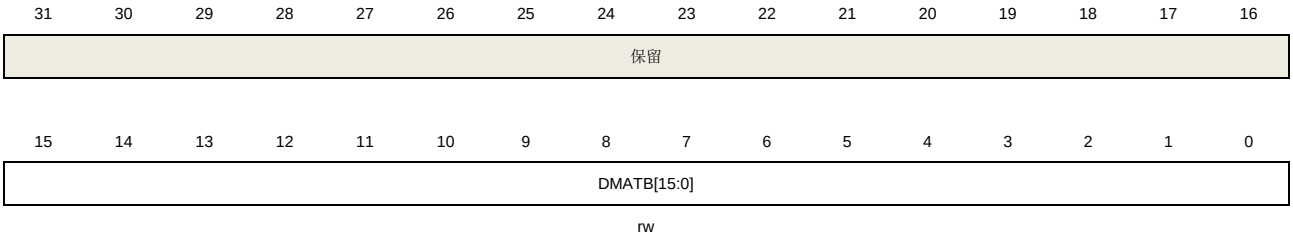


位/位域	名称	描述
31:13	保留	必须保持复位值。
12:8	DMATC [4:0]	DMA 传输计数 该位域定义了 DMA 访问（读写）TIMERx_DMATB 寄存器的数量 n，n = (DMATC [4:0] +1). DMATC [4:0] 从 5'b0_0000 到 5'b1_0001.
7:5	保留	必须保持复位值。
4:0	DMATA [4:0]	DMA 传输起始地址 该位域定义了 DMA 访问 TIMERx_DMATB 寄存器的第一个地址。当通过 TIMERx_DMA 第一次访问时，访问的就是该位域指定的地址。第二次访问 TIMERx_DMATB 时，将访问起始地址+0x4。

DMA 发送缓冲区寄存器 (TIMERx_DMATB)

地址偏移：0x4C
复位值：0x0000 0000

该寄存器只能按字(32位)访问。



位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	DMATB [15:0]	DMA 发送缓冲 对这个寄存器的读或写，（起始地址+传输次数*4）地址范围内的寄存器会被访问传输次数由硬件计算，范围为 0 到 DMATC。

15.6. 基本定时器（TIMERx, x=5）

基本定时器仅仅适用于 GD32F150 系列芯片。

15.6.1. 简介

基本定时器(Timer5)包含一个无符号 16 位计数器。可以被用作通用定时器和为 DAC (数字到模拟转换器)提供时钟。基本定时器可以配置产生 DMA 请求，TRGO 触发连接到 DAC。

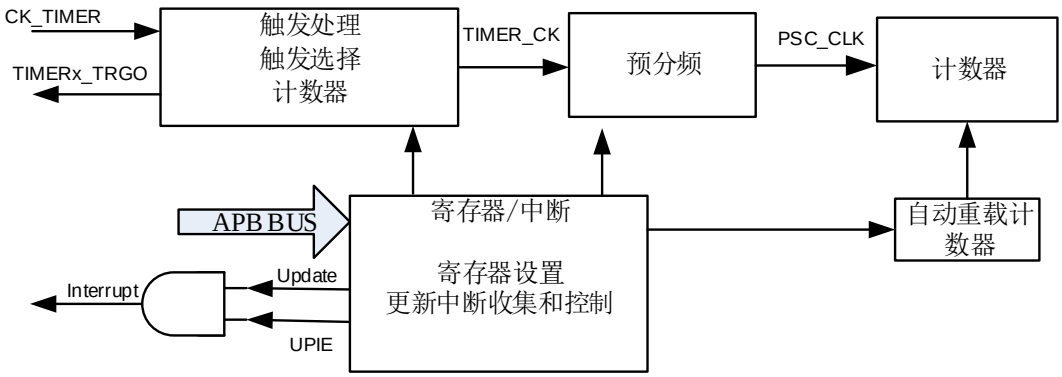
15.6.2. 主要特性

- 计数器宽度：16位
- 时钟源只有内部时钟
- 计数模式：向上计数
- 可编程的预分频器：16位，运行时可以被改变
- 自动重装载功能.
- 中断输出和DMA请求：更新事件

15.6.3. 结构框图

*图 15-82. 基本定时器结构框图*提供了基本定时器内部配置的细节

图 15-82. 基本定时器结构框图



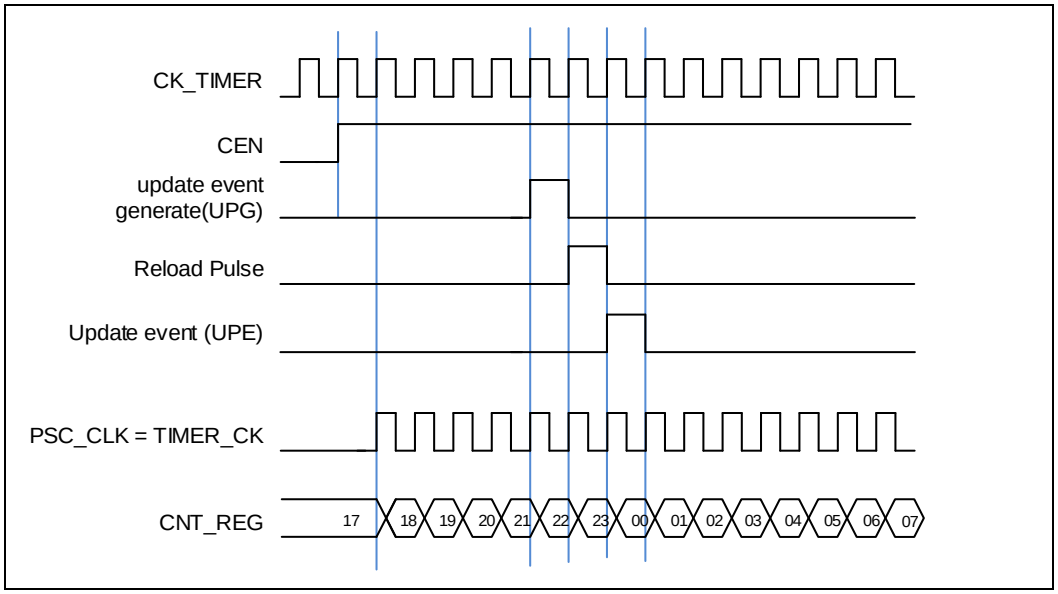
15.6.4. 功能描述

时钟源配置

基本定时器可以由内部时钟源 CK_TIMER 驱动 。

基本定时器仅有一个时钟源 CK_TIMER，用来驱动计数器预分频器。当 CEN 置位，CK_TIMER 经过预分频器（预分频值由 TIMERx_PSC 寄存器确定）产生 PSC_CLK。

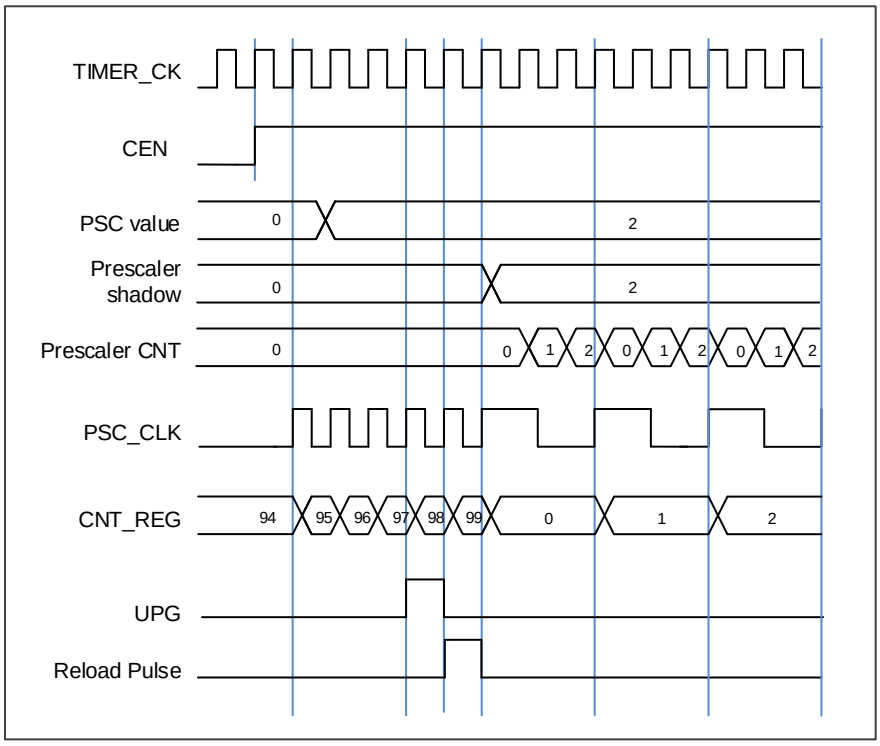
图 15-83. 内部时钟分频为 1 时，计数器的时序图



时钟预分频器

预分频器可以将定时器的时钟（TIMER_CK)频率按 1 到 65536 之间的任意值分频，分频后的时钟 PSC_CLK 驱动计数器计数。分频系数受预分频寄存器 TIMERx_PSC 控制，这个控制寄存器带有缓冲器，它能够在运行时被改变。新的预分频器的参数在下一次更新事件到来时被采用。

图 15-84. 当 PSC 数值从 0 变到 2 时，计数器的时序图



计数器向上计数模式

在这种模式，计数器的计数方向是向上计数。计数器从 0 开始向上连续计数到自动加载值（定义在 TIMERx_CAR 寄存器中），一旦计数器计数到自动加载值，会重新从 0 开始向上计数并产生上溢事件。在向上计数模式中，TIMERx_CTL0 寄存器中的计数方向控制位 DIR 应该被设置成 0。

当通过 TIMERx_SWEVG 寄存器的 UPG 位置 1 来设置更新事件时，计数值会被清 0，并产生更新事件。

如果 TIMERx_CTL0 寄存器的 UPDIS 置 1，则禁止更新事件。

当发生更新事件时，所有影子寄存器（计数器自动重载寄存器，预分频寄存器）都将被更新。

图 15-85. 向上计数时序图，PSC=0/2 和 *图 15-86. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值*给出了一些例子，当 TIMERx_CAR=0x99 时，计数器在不同预分频因子下的行为。

图 15-85. 向上计数时序图，PSC=0/2

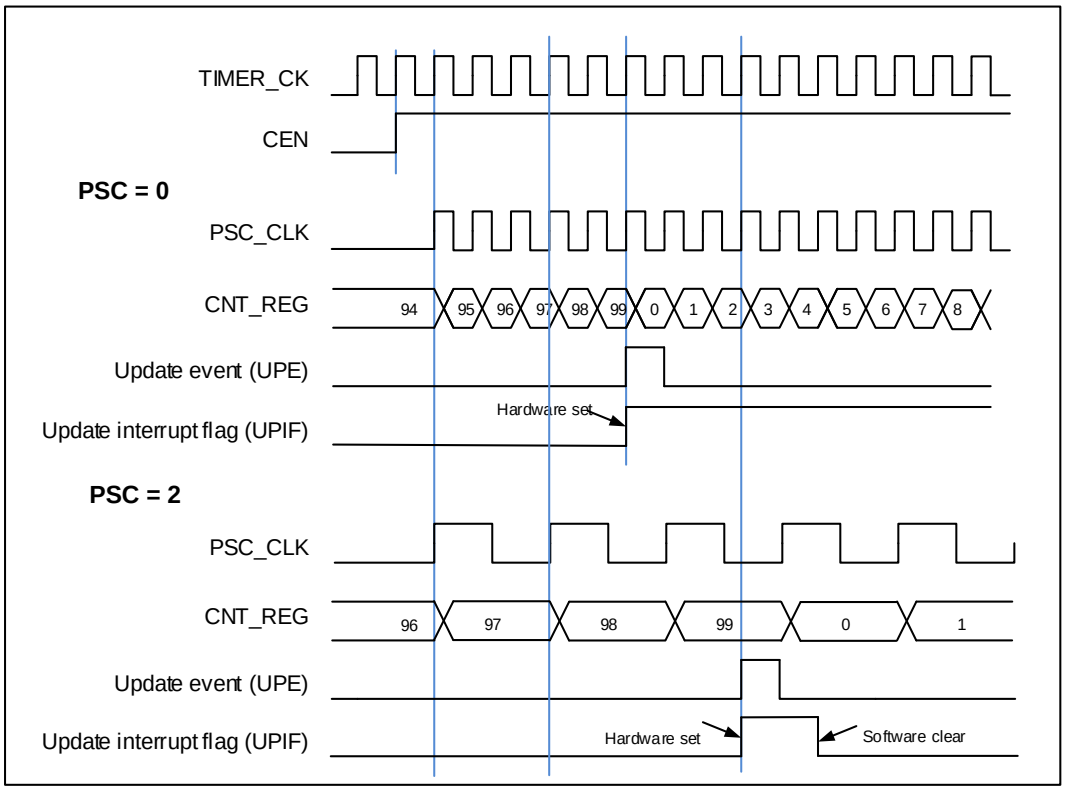
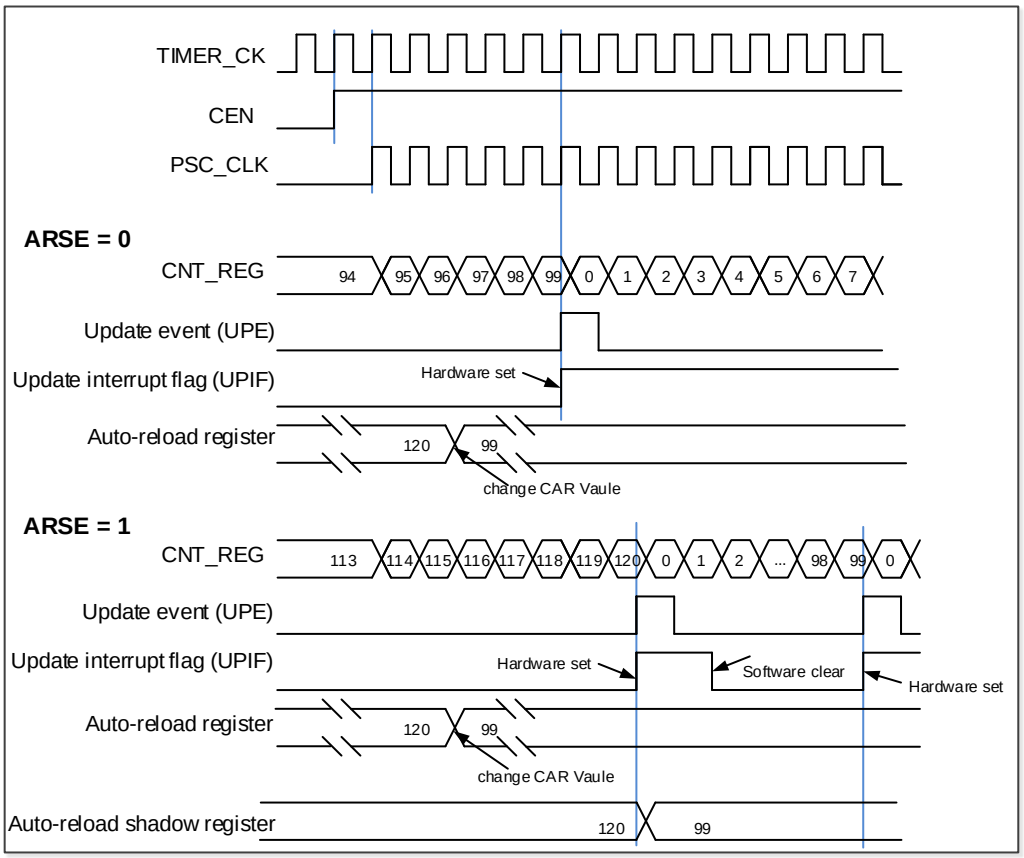


图 15-86. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值



单脉冲模式

单脉冲模式与重复模式是相反的，设置TIMERx_CTL0寄存器的SPM位置1，则使能单脉冲模式。当SPM置1，计数器在下次更新事件到来后清零并停止计数。

一旦设置定时器运行在单脉冲模式下，需要设置TIMERx_CTL0寄存器的定时器使能位CEN=1来使能计数器，此后CEN位一直保持为1直到更新事件发生或者CEN位被软件写0。如果CEN位被软件清0，计数器停止工作，计数值被保持。

定时器调试模式

当 Cortex®-M3 内核停止，DBG_CTL0 寄存器中的 TIMERx_HOLD 配置位被置 1，定时器计数器停止。

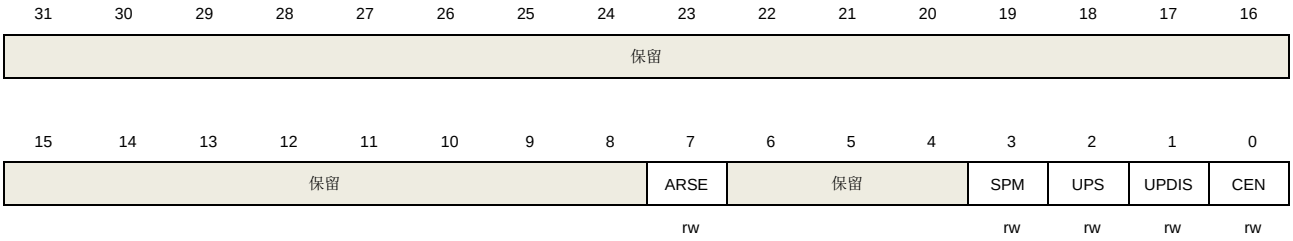
15.6.5. TIMERx 寄存器(x=5)

TIMER5 基地址：0x4000 1000

控制寄存器 0 (TIMERx_CTL0)

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字(32位)访问。



位/位域	名称	描述
31:8	保留	必须保持复位值。
7	ARSE	自动重载影子使能 0: 禁能 TIMERx_CAR 寄存器的影子寄存器 1: 使能 TIMERx_CAR 寄存器的影子寄存器
6:4	保留	必须保持复位值。
3	SPM	单脉冲模式 0：单脉冲模式禁能。更新事件发生后，计数器继续计数 1：单脉冲模式使能。在下一次更新事件发生时，计数器停止计数
2	UPS	更新请求源 软件配置该位，选择更新事件源. 0：以下事件均会产生更新中断或DMA请求： UPG位被置1 计数器溢出/下溢 复位模式产生的更新 1：下列事件会产生更新中断或DMA请求： 计数器溢出/下溢
1	UPDIS	禁止更新. 该位用来使能或禁能更新事件的产生 0：更新事件使能. 更新事件发生时，相应的影子寄存器被装入预装载值，以下事件均会产生更新事件： UPG位被置1 计数器溢出/下溢 复位模式产生的更新 1：更新事件禁能.

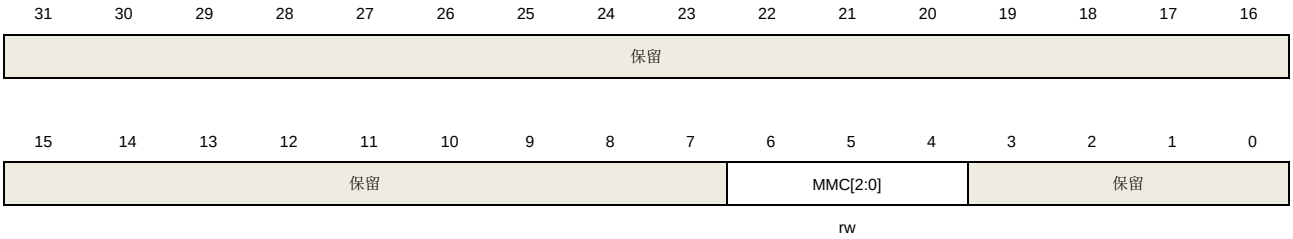
注意：当该位被置 1 时，UPG 位被置 1 或者复位模式不会产生更新事件，但是计数器和预分频器被重新初始化

0	CEN	计数器使能 0: 计数器禁能 1: 计数器使能 在软件将 CEN 位置 1 后，外部时钟、暂停模式和正交译码器模式才能工作。

控制寄存器 1 (TIMERx_CTL1)

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

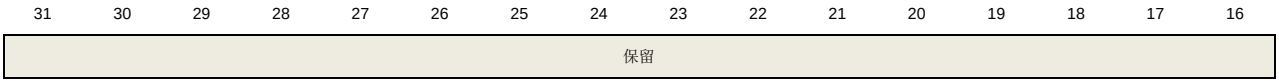


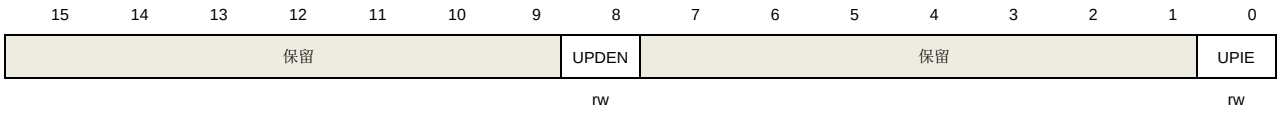
位/位域	名称	描述
31:7	保留	必须保持复位值。
6:4	MMC[2:0]	这些位控制 TRGO 信号的选择，TRGO 信号由主定时器发给从定时器用于同步功能 000：当产生一个定时器复位事件后，输出一个TRGO信号，定时器复位源为： 主定时器产生一个复位事件 TIMERx_SWEVG寄存器中UPG位置1 001：当产生一个定时器使能事件后，输出一个TRGO信号，定时器使能源为： CEN位置1 在暂停模式下，触发输入置1 010：当产生一个定时器更新事件后，输出一个TRGO信号，更新事件源由UPDIS和UPS位决定
3:0	保留	必须保持复位值。

DMA 和中断使能寄存器 (TIMERx_DMAINTEN)

地址偏移：0x0C
复位值：0x0000 0000

该寄存器只能按字(32位)访问。



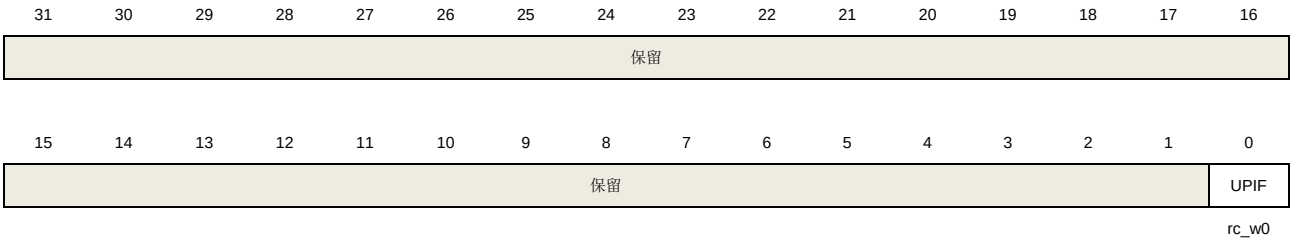


位/位域	名称	描述
31:9	保留	必须保持复位值。
8	UPDEN	更新 DMA 请求使能 0：禁止更新 DMA 请求 1：使能更新 DMA 请求
7:1	保留	必须保持复位值。
0	UPIE	更新中断使能 0：禁止更新中断 1：使能更新中断

中断标志寄存器 (TIMERx_INTF)

地址偏移：0x10
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

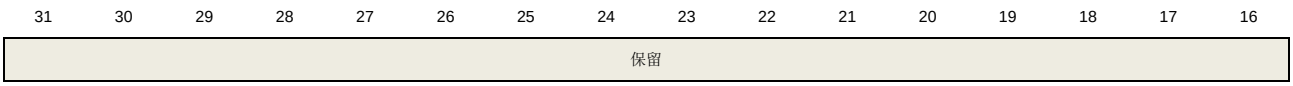


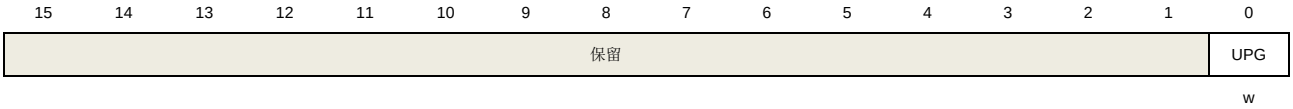
位/位域	名称	描述
31:1	保留	必须保持复位值。
0	UPIF	更新中断标志 此位在任何更新事件发生时由硬件置 1，软件清 0。 0：无更新中断发生 1：发生更新中断

软件事件产生寄存器 (TIMERx_SWEVG)

地址偏移：0x14
复位值：0x0000 0000

该寄存器只能按字(32位)访问。



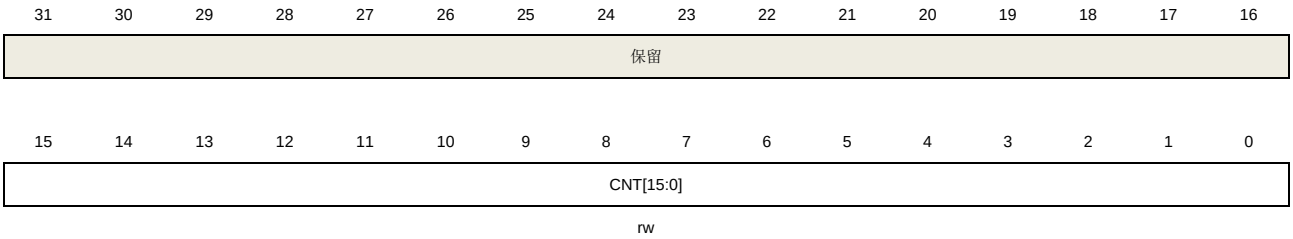


位/位域	名称	描述
31:1	保留	必须保持复位值。
0	UPG	更新事件产生 此位由软件置 1，被硬件自动清 0。当此位被置 1 并且向上计数模式，计数器被清 0，预分频计数器将同时被清除。 0：无更新事件产生 1：产生更新事件

计数器寄存器 (TIMERx_CNT)

地址偏移：0x24
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

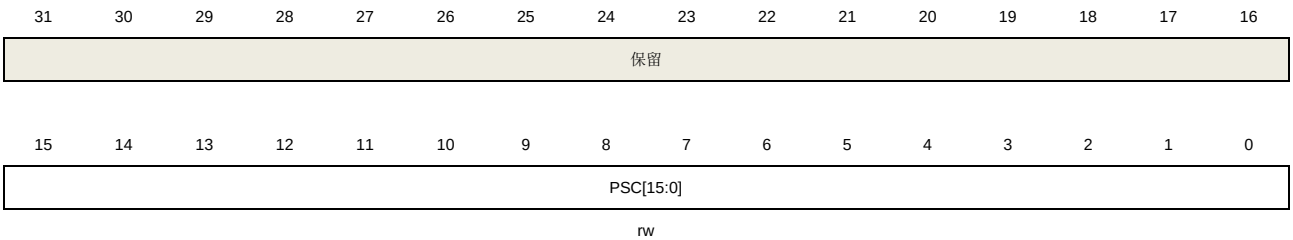


位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CNT[15:0]	这些位是当前的计数值。写操作能改变计数器值。

预分频寄存器 (TIMERx_PSC)

地址偏移：0x28
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

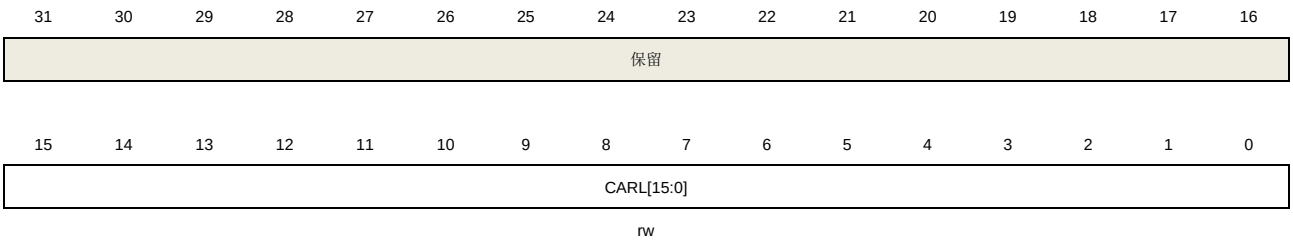


位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	PSC[15:0]	计数器时钟预分频值 计数器时钟等于 TIMER_CK 时钟除以(PSC+1)，每次当更新事件产生时，PSC 的值被装入到对应的影子寄存器。

计数器自动重载寄存器 (TIMERx_CAR)

地址偏移：0x2C
复位值：0x0000 0000

该寄存器只能按字(32位)访问。



位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CARL[15:0]	计数器自动重载值 这些位定义了计数器的自动重载值。

16. 红外线接口（IFRP）

16.1. 简介

红外线接口（IFRP）用来控制发射红外光的LED，该LED可发射红外数据来实现红外遥控。

该模块无寄存器，受TIMER15定时器和TIMER16定时器控制。通过设置GPIO引脚为高速模式，可以提高模块的输出电流能力。

16.2. 主要特性

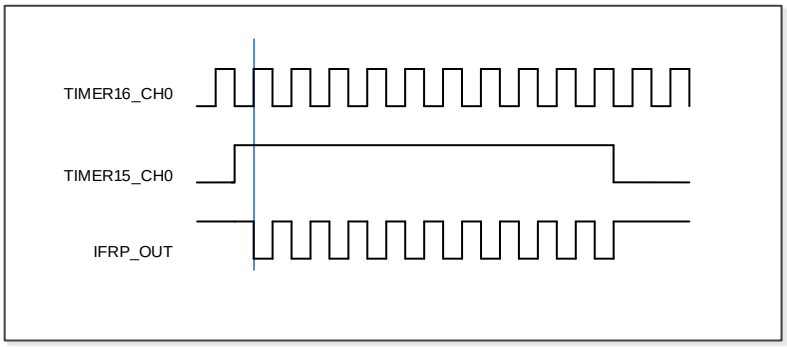
- IFRP输出信号由TIMER15定时器的通道0和TIMER16定时器的通道0决定
- 为了获取正确的红外信号，TIMER15定时器应该产生低频调制包络信号，TIMER16应该产生高频载波信号
- 通过设置SYSCFG_CFG0中的PB9_HCCE，红外线接口输出（PB9）能够提供高电流输出驱动LED接口

16.3. 功能描述

IFRP 模块整合了 TIMER15 定时器和 TIMER16 定时器的输出来产生红外信号。

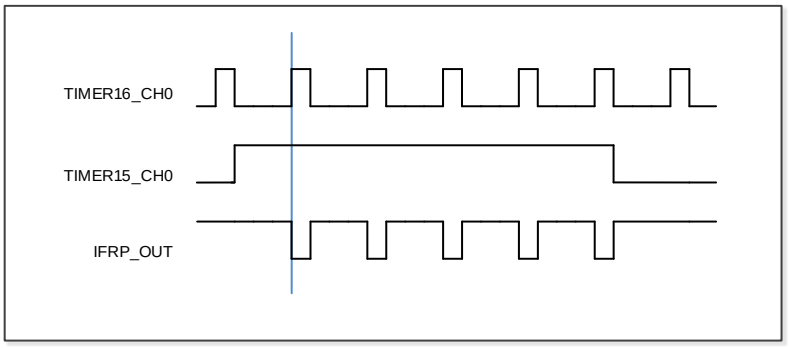
1. 通过对 TIMER15 定时器的通道 0 编程输出低频 PWM 信号来产生调制包络信号，对 TIMER16 定时器的通道 0 编程输出高频 PWM 信号来产生载波信号。产生信号之前需要开启这些通道。
2. 配置 GPIO 口为复用，并使能这些引脚。
3. 如果你想获得更高的驱动电流输出，需要将 IFRP_OUT 映射到 PB9 口上，并且在 SYS_CFG 模式下通过相应寄存器设置 PB9 口为高速模式。

图 16-1. IFRP 输出时序图 1



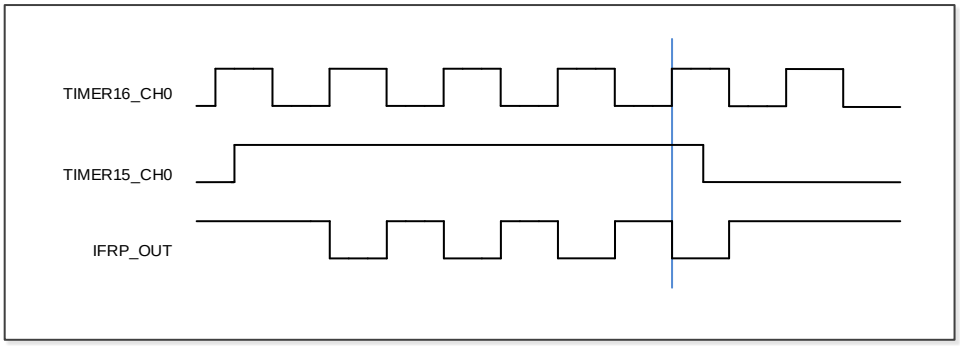
注：IFRP_OUT比TIMER16_CH0时钟通道1输出信号有一个APB时钟延迟。

图 16-2. IFRP 输出时序图 2



注：载波（TIMER15_CH0）的占空比可以被改变，在TIMER15定时器的通道0输出电平为高时，IFRP_OUT和TIMER16定时器的通道0输出信号反相。

图 16-3. IFRP 输出时序图 3



注：IFRP_OUT将保持TIMER16定时器通道0输出信号的完整，即使TIMER15定时器的包络信号无效。

17. 通用同步异步收发器（USART）

17.1. 简介

通用同步异步收发器（USART）提供了一个灵活方便的串行数据交换接口。数据帧可以通过全双工或半双工，同步或异步的方式进行传输。USART 提供了可编程的波特率发生器，能对UCLK（PCLK1，PCLK2）进行分频产生 USART 发送和接收所需的特定频率。

USART 不仅支持标准的异步收发模式，还实现了一些其他类型的串行数据交换模式，如红外编码规范，SIR，智能卡协议，LIN，半双工以及同步模式。它还支持多处理器通信和 Modem 流控操作（CTS/RTS）。数据帧支持从 LSB 或者 MSB 开始传输。数据位的极性和 TX/RX 引脚都可以灵活配置。

所有 USART 都支持 DMA 功能，来实现高速率的数据通信。

17.2. 主要特性

- NRZ 标准格式(Mark/Space)
- 全双工异步通信
- 半双工单线通信
- 双时钟域
  - 互为异步关系的APB时钟和USART时钟
  - 不依赖于PCLK设置的波特率设置
- 可编程的波特率产生器，当时钟频率为72MHz，过采样为8时，最高速度可达9 MBits/s
- 完全可编程的串口特性：
  - 数据位（8或9位）低位或高位在前
  - 偶校验位，奇校验位，无校验位的生成/检测
  - 产生1，1.5或者2个停止位
- 可互换的Tx/Rx引脚
- 可配置的数据极性
- 支持硬件Modem流控操作（CTS/RTS）和RS485驱动使能
- 借助集中式DMA，可实现可配置的多级缓存通信
- 发送器和接收器可分别使能
- 奇偶校验位控制：
  - 发送奇偶校验位
  - 检测接收的数据字节的奇偶校验位
- LIN断开帧的产生和检测
- 支持红外数据协议（IrDA）
- 同步传输模式以及为同步传输输出发送时钟
- 支持兼容ISO7816-3的智能卡接口：
  - 字节模式（T=0）
  - 块模式（T=1）
  - 直接和反向转换

- 多处理器通信：
  - 如果地址不匹配，则进入静默模式
  - 通过线路空闲检测或者地址标记检测从静默模式唤醒
- 支持ModBus通信：
  - 超时功能
  - CR/LF字符识别
- 从深度睡眠模式唤醒：
  - 通过标准的RBNE中断
  - 通过WUF中断
- 多种状态标志：
  - 传输检测标志：接收缓冲区不为空（RBNE），发送缓冲区为空（TBE），传输完成（TC）
  - 错误检测标志：过载错误（ORERR），噪声错误（NERR），帧格式错误（FERR），奇偶校验错误（PERR）
  - 硬件流控操作标志：CTS变化（CTSF）
  - LIN模式标志：LIN断开检测（LBDF）
  - 多处理器通信模式标志：IDLE帧检测（IDLEF）
  - ModBus通信标志：地址/字符匹配（AMF），接收超时（RTF）
  - 智能卡模式标志：块结束（EBF）和接收超时（RTF）
  - 从深度睡眠模式唤醒标志
  - 若相应的中断使能，这些事件发生将会触发中断

USART0 完全实现上述功能，但是 USART1 只实现了上面所介绍功能的部分，下面这些功能在 USART1 中没有实现：

- 智能卡模式
- IrDA SIR ENDEC模块
- LIN模式
- 双时钟域和从深度睡眠模式唤醒
- 接收超时中断
- Modbus通信

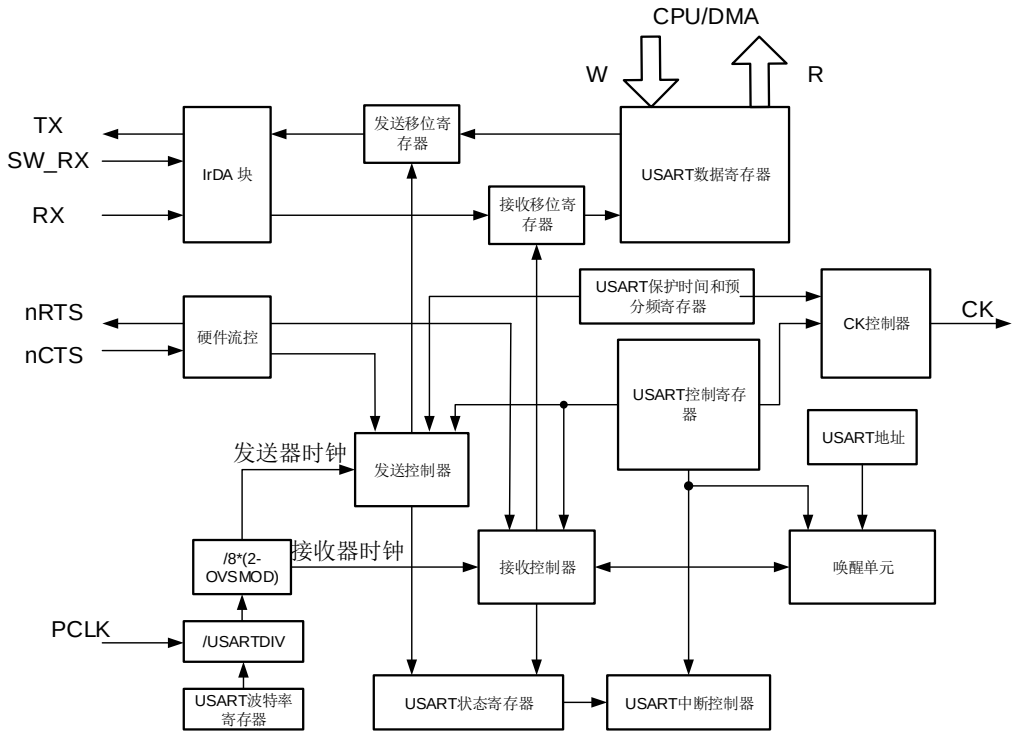
17.3. 功能描述

USART 接口通过*表 17-1. USART 重要管脚描述*中主要管脚从外部连接到其他设备。

表 17-1. USART 重要管脚描述

管脚名称	类型	描述
RX	输入	接收数据
TX	输出 I/O (单线模式/智能卡模式)	发送数据当 USART 使能后，若无数据发送，默认为高电平
CK	输出	用于同步通信的串行时钟信号
nCTS	输入	硬件流控模式发送使能信号
nRTS	输出	硬件流控模式发送请求信号

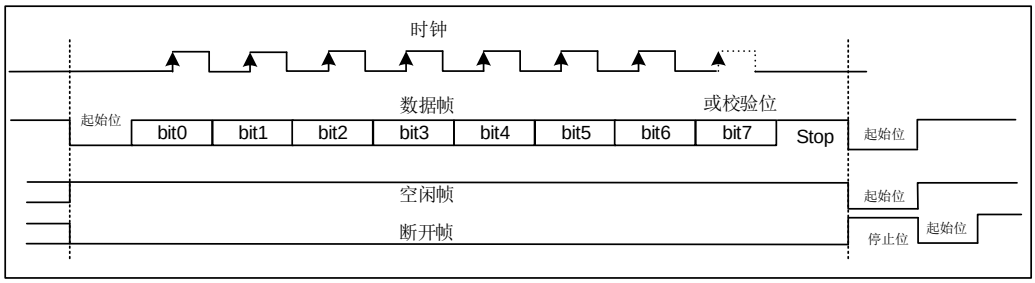
图 17-1. USART 模块内部框图



17.3.1. USART 帧格式

USART 数据帧开始于起始位，结束于停止位。USART_CTL0 寄存器中 WL 位可以设置数据长度。将 USART_CTL0 寄存器中 PCEN 置位，最后一个数据位可以用作校验位。若 WL 位为 0，第七位为校验位。若 WL 位置 1，第八位为校验位。USART_CTL0 寄存器中 PM 位用于选择校验位的计算方法。

图17-2. USART字符帧 (8数据位和1停止位)



在发送和接收中，停止位可以在 USART_CTL1 寄存器中 STB[1:0]位域中配置。

表 17-2. 停止位配置

STB[1:0]	停止位长度（位）	功能描述
00	1	默认值
01	保留	保留
10	2	标准 USART 和单线模式
11	1.5	智能卡模式发送和接收

一个空闲帧中，所有位都为 1。数据帧长度与正常 USART 数据帧长度相同。

紧随停止位后多个低电平为中断帧。USART 数据帧的传输速度由 UCLK 时钟频率，波特率发生器的配置，以及过采样模式共同决定。

17.3.2. 波特率发生

波特率分频系数是一个 16 位的数字，包含 12 位整数部分和 4 位小数部分。波特率发生器使用这两部分组合所得的数值来确定波特率。由于具有小数部分的波特率分频系数，将使 USART 能够产生所有标准波特率。

波特率分频系数（USARTDIV）与 UCLK 具有如下关系：

如果过采样率是 16，公式为：

$$\text{USARTDIV}=\frac{\text{UCLK}}{16\times\text{Baud Rate}} \quad (17\text{-}1)$$

如果过采样是 8，公式为：

$$\text{USARTDIV}=\frac{\text{UCLK}}{8\times\text{Baud Rate}} \quad (17\text{-}2)$$

例如，当过采样是16：

1. 由USART_BAUD寄存器的值得到USARTDIV：
   假设USART_BAUD=0x21D，则INTDIV=33（0x21），FRADIV=13（0xD）。
   UASRTDIV=33+13/16=33.81。
2. 由USARTDIV得到USART_BAUD寄存器的值：
   假设要求UASRTDIV=30.37，INTDIV=30（0x1E）
   16*0.37=5.92，接近整数6，所以FRADIV=6（0x6）
   USART_BAUD=0x1E6。

注意：若取整后 FRADIV=16（溢出），则进位必须加到整数部分。

17.3.3. USART 发送器

如果 USART_CTL0 寄存器的发送使能位（TEN）被置位，当发送数据缓冲区不为空时，发送器将会通过 TX 引脚发送数据帧。TX 引脚的极性可以通过 USART_CTL1 寄存器中 TINV 位来配置。时钟脉冲通过 CK 引脚输出。

在传输被破坏的情况下，只要传输在继续，便不得禁用 TEN 位。

系统上电后，TBE 默认为高电平。在 USART_STAT 寄存器中 TBE 置位时，数据可以在不覆盖前一个数据的情况下写入 USART_TDATA 寄存器。当数据写入 USART_TDATA 寄存器，TBE 位将被清 0。在数据由 USART_TDATA 移入移位寄存器后，该位由硬件置 1。如果数据在一个发送过程正在进行时被写入 USART_TDATA 寄存器，它将首先被存入发送缓冲区，在当前发送过程完成时传输到发送移位寄存器中。如果数据在写入 USART_TDATA 寄存器时，没有发送过程正在进行，TBE 位将被清零然后迅速置位，原因是数据被立刻传输到发送移位寄

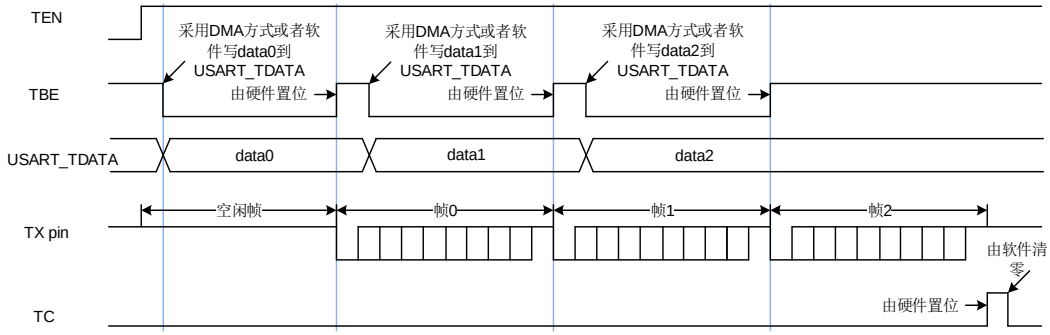
存器。

假如一帧数据已经被发送出去，并且 TBE 位已被置位，那么 USART_STAT 寄存器中 TC 位将被置 1。如果 USART_CTL0 寄存器中的中断使能位（TCIE）为 1，将会产生中断。

USART 发送按下列步骤进行：

1. 通过USART_CTL0寄存器的WL设置字长；
2. 在USART_CTL1寄存器中写STB[1:0]位来设置停止位的长度；
3. 如果选择了多级缓存通信方式，应该在USART_CTL2寄存器中使能DMA（DENT位）；
4. 在USART_BAUD寄存器中设置波特率；
5. 在USART_CTL0寄存器中置位UEN位，使能USART；
6. 在USART_CTL0寄存器中设置TEN位；
7. 等待TBE置位；
8. 向USART_TDATA寄存器写数据；
9. 等待TC=1，发送完成。

图 17-3. USART 发送步骤



在禁用 USART 或进入低功耗状态之前，必须等待 TC 置位。

通过向 USART_INTC 寄存器的 TCC 位写 1 可将 TC 位清 0。当 SBKCMD 置位时，会发送一个断开帧，发送完成后，SBKCMD 将被清 0。

17.3.4. USART 接收器

上电后，按以下步骤使能 USART 接收器：

1. 写USART_CTL0寄存器的WL位去设置字长；
2. 在USART_CTL1寄存器中写STB[1:0]位来设置停止位的长度；
3. 如果选择了多级缓存通信方式，应该在USART_CTL2寄存器中使能DMA（DENR位）；
4. 在USART_BAUD寄存器中设置波特率；
5. 在USART_CTL0寄存器中置位UEN位，使能USART；
6. 在USART_CTL0中设置REN位。

接收器在使能后，若检测到一个有效的起始脉冲，接收器开始接收码流。在接收一个数据帧的过程中会进行噪声错误，奇偶校验错误，帧错误和过载错误的检测。

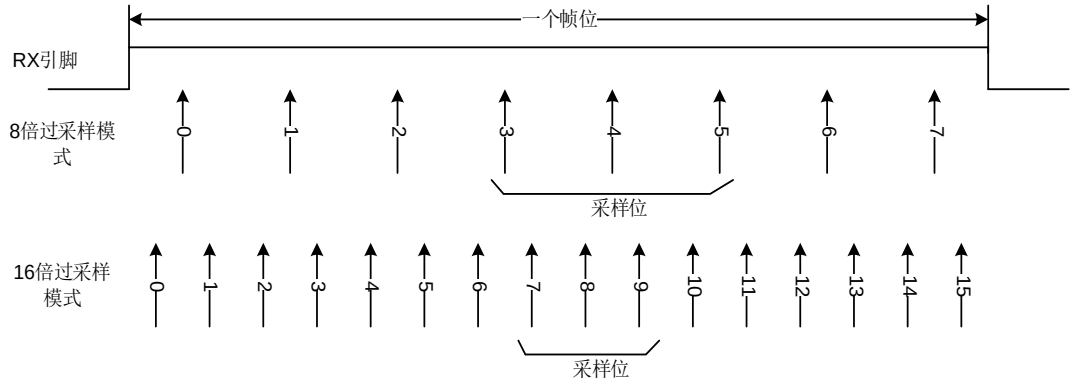
当一个数据帧接收完成，USART_STAT 寄存器中的 RBNE 置位，如果设置了 USART_CTL0 寄存器中相应的中断使能位 RBNEIE，将会产生中断。在 USART_STAT 寄存器中可以观察接收状态标志。

软件可以通过读 USART_RDATA 寄存器或者 DMA 方式获取接收到的数据。不管是直接读寄存器还是通过 DMA，只要是对 USART_RDATA 寄存器的一个读操作都可以清除 RBNE 位。

在接收过程中，需使能 REN 位，不然当前的数据帧将会丢失。

在默认情况下，接收器通过获取三个采样点的值来估计该位的值。如果是 8 倍过采样模式，选择第 3、4、5 个采样点；如果是 16 倍过采样模式，选择第 7、8、9 个采样点。如果在 3 个采样点中有 2 个或 3 个为 0，该数据位被视为 0，否则为 1。如果 3 个采样点中有一个采样点的值与其他两个不同，不管是起始位，数据位，奇偶校验位或者停止位，都将产生噪声错误（NERR）。如果使能 DMA，并置位 USART_CTL2 寄存器中 ERRIE，将会产生中断。如果在 USART_CTL2 中置位 OSB，接收器将仅获取一个采样点来估计一个数据位的值。在这种情况下将不会检测到噪声错误。

图 17-4. 过采样方式接收一个数据位（OSB=0）



通过置位 USART_CTL0 寄存器中的 PCEN 位，奇偶校验功能被使能，接收器在接收一个数据帧时计算预期奇偶校验值，并将其与接收到的奇偶校验位进行比较。如果不相等，USART_STAT 寄存器中 PERR 被置位。如果设置了 USART_CTL0 寄存器中的 PERRIE 位，将产生中断。

如果在停止位传输过程中 RX 引脚为 0，将产生帧错误，USART_STAT 寄存器中 FERR 置位。如果使能 DMA 并置位 USART_CTL2 寄存器中 ERRIE 位，将产生中断。

当接收到一帧数据，而 RBNE 位还没有被清零，随后的数据帧将不会存储在数据接收缓冲区中。USART_STAT 寄存器中的溢出错误标志位 ORERR 将置位。如果使能 DMA 并置位 USART_CTL2 寄存器中 ERRIE 位或者置位 RBNEIE，将产生中断。

若接收过程中，产生了噪声错误（NERR）、校验错误（PERR）、帧错误（FERR）或溢出错误（ORERR），则NERR、PERR、FERR或ORERR将和RBNE同时置位。如果没有使能DMA，RBNE中断发生时，软件需检查是否有噪声错误、校验错误、帧错误或溢出错误产生。

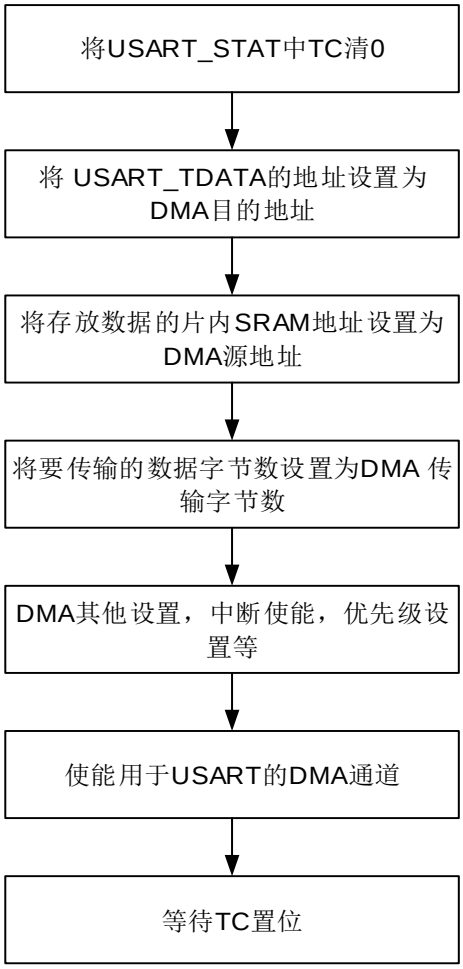
17.3.5. DMA 方式访问数据缓冲区

为减轻处理器的负担，可以采用 DMA 访问发送缓冲区或者接收缓冲区。置位 USART_CTL2 寄存器中 DENT 位可以使能 DMA 发送，置位 USART_CTL2 寄存器中 DENR 位可以使能 DMA 接收。

当 DMA 用于 USART 发送时，DMA 将数据从片内 SRAM 传送到 USART 的数据缓冲区。配

置步骤如*图 17-5. 采用 DMA 方式实现 USART 数据发送配置步骤*所示。

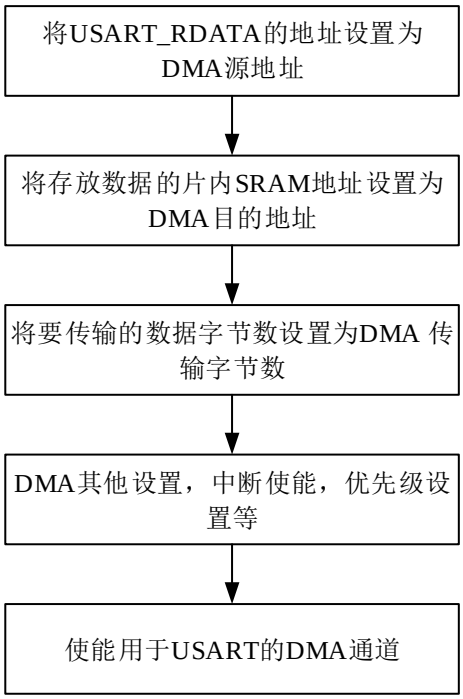
图 17-5. 采用 DMA 方式实现 USART 数据发送配置步骤



所有数据帧都传输完成后，USART_STAT 寄存器中 TC 位置 1。如果 USART_CTL0 寄存器中 TCIE 置位，将产生中断。

当 DMA 用于 USART 接收时，DMA 将数据从接收缓冲区传送到片内 SRAM。配置步骤如*图 17-6. 采用 DMA 方式实现 USART 数据接收配置步骤*所示。如果将 USART_CTL2 寄存器中 ERRIE 位置 1，USART_STAT 寄存器中的错误标志位（FERR、ORERR 和 NERR）被置位时将产生中断。

图 17-6. 采用 DMA 方式实现 USART 数据接收配置步骤

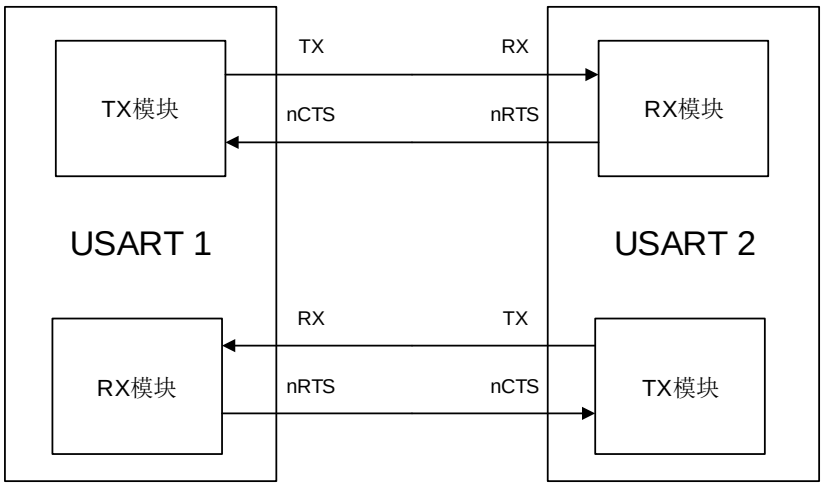


当 USART 接收到的数据数量达到了 DMA 传输数据数量，DMA 模块将产生传输完成中断。

17.3.6. 硬件流控制

硬件流控制功能通过 nCTS 和 nRTS 引脚来实现。通过将 USART_CTL2 寄存器中 RTSEN 位置 1 来使能 RTS 流控，将 USART_CTL2 寄存器中 CTSEN 位置 1 来使能 CTS 流控。

图 17-7. 两个 USART 之间的硬件流控制



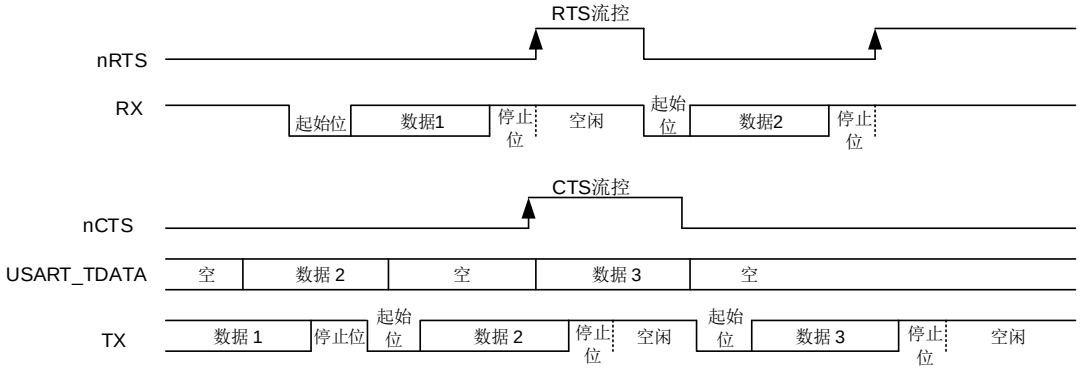
RTS 流控

仅当 nRTS 信号低电平时，USART 接收器才能接收数据。在接收期间，信号保持低电平。接收完成之后，电平变高。当 nRTS 信号再次变低时，开始下一次接收。如果接收寄存器已满，信号将保持高电平。

CTS 流控

如果 USART_STAT 寄存器的 TBE 位是'0'，且 nCTS 信号是低电平，发送器发送数据帧。在发送期间，若 nCTS 信号变为高电平，发送器在发送完当前数据帧后停止发送。

图 17-8. 硬件流控制



RS485 驱动使能

驱动使能功能通过设置 USART_CTL2 控制寄存器的 DEM 位来打开。它允许用户通过 DE（Driver Enable）信号激活外部收发器控制。提前时间是驱动使能信号和第一个字节的起始位之间的时间间隔。这个时间可以在 USART_CTL0 控制器的 DEA[4:0]位域中进行设置。滞后时间是一个发送信息最后一个字节的停止位与释放 DE 信号之间的时间间隔。这个时间可以在 USART_CTL0 控制寄存器的 DED[4:0]位域中进行设置。DE 信号的极性可以通过 USART_CTL2 控制寄存器的 DEP 位进行设置。

17.3.7. 多处理器通信

在多处理器通信中，多个 USART 被连接成一个网络。对于一个设备来说，监视所有来自 RX 引脚的消息，是一种巨大的负担。为减轻设备负担，软件可以通过将 USART_CMD 寄存器中 MMCMD 位置 1 使 USART 进入静默模式。

如果 USART 处于静默模式，所有的接收状态标志位将不会被置位。此外，USART 可以由硬件用以下两种方式中的一种来唤醒：空闲总线检测和地址匹配检测。

设备默认使用空闲总线检测方法唤醒 USART。当在 RX 引脚检测到空闲帧时，硬件会将 RWU 清零，从而退出静默模式，但 USART_STAT 寄存器中 IDLEF 位不会被置 1。

当 USART_CTL0 寄存器中 WM 被置位，数据最高位会被认为是地址标志位。如果地址标志位为 1，该字节被认为是地址字节。如果地址标志位是 0，该字节被认为是数据字节。如果地址字节的低 4 位或低 7 位与 USART_CTL1 寄存器中的 ADDR 位相同，硬件会将 RWU 清零，并退出静默模式。接收到将 USART 唤醒的数据帧，RBNE 将置位。状态标志可以从 USART_STAT 寄存器中获取。如果地址字节的低 4 位或低 7 位与 USART_CTL1 寄存器中的 ADDR 位不相同，硬件会置位 RWU 并自动进入静默模式。在这种情况下，RBNE 不会被置位。

如果 USART_CTL0 寄存器中 PCEN 位被置位，地址字节最高位被视为校验位，其余位被视为地址。如果 ADDM 位被置位，且接收帧为 7 位的数据，其中最低的 6 位将与 ADDR[5:0]比较。如果 ADDM 位被置位，且接收帧为 9 位的数据，其中最低 8 位将与 ADDR[7:0]进行比较。

17.3.8. LIN 模式

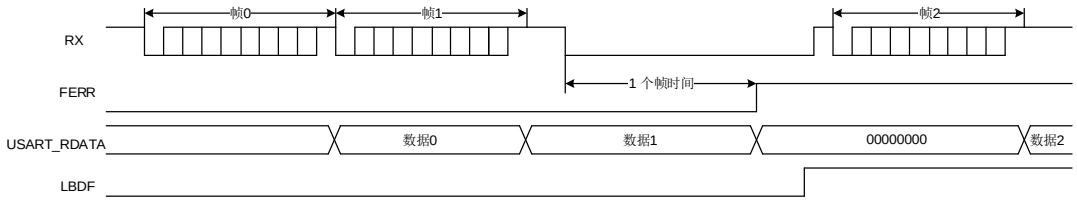
将 USART_CTL1 寄存器的 LMEN 置位即可使能本地互联网络模式。在 LIN 模式下，USART_CTL1 寄存器中 CKEN，STB[1:0]和 USART_CTL2 的 SCEN，HDEN，IREN 位须被清 0。

LIN 发送过程与普通发送过程基本相同。数据位的长度只能为 8。一个停止位后连续 13 个 0 为断开帧。

断开检测功能完全独立于普通 USART 接收器。因此，在空闲状态下或帧传输状态下都可以进行断开帧检测。USART_CTL1 寄存器中 LBLEN 位可以选择断开帧的长度。如果在 RX 引脚检测到大于或等于与预期的断开帧长度的 0（LBLEN=0 时，10 个 0；LBLEN=1 时，11 个 0），USART_STAT 寄存器中 LBDF 置位。如果 USART_CTL1 寄存器中 LBDIE 被置位，将产生中断。

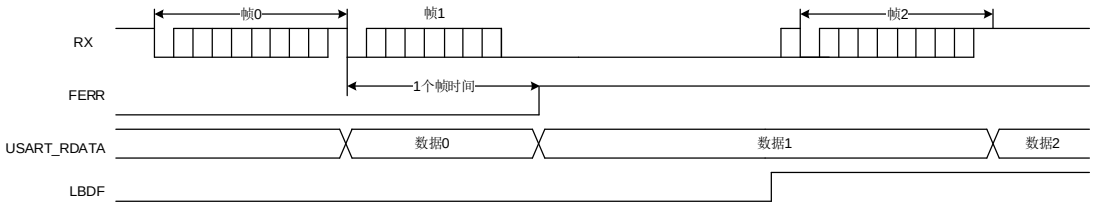
如*图 17-9. 空闲状态下检测断开帧*所示，如果断开帧发生在空闲状态下，USART 接收器会接收到一个全 0 数据帧，同时 FERR 置位。

图 17-9. 空闲状态下检测断开帧



如*图 17-10. 数据传输过程中检测断开帧*所示，如果断开帧发生在数据传输过程中，当前传输帧发生错误，FERR 置位。

图 17-10. 数据传输过程中检测断开帧



17.3.9. 同步通信模式

USART 支持主机模式下的全双工同步串行通信，可以通过置位 USART_CTL1 的 CKEN 位来使能。在同步模式下，USART_CTL1 的 LMEN 和 USART_CTL2 的 SCEN，HDEN，IREN 位应被清 0。CK 引脚作为 USART 同步发送器的时钟输出，仅仅当 TEN 位被使能时，它才被激活。在起始位和停止位传送期间，不会从 CK 引脚输出时钟脉冲。USART_CTL1 的 CLEN 位用来决定在最低位（地址索引位）发送期间是否有时钟信号输出。在空闲状态和断开帧的发送过程中，也不会有时钟信号产生。USART_CTL1 的 CPH 位用来决定数据在第一个时钟沿被采样还是在第二个时钟沿被采样。USART_CTL1 的 CPL 位用来决定在 USART 同步模式空闲状态下，时钟引脚的电平。

在发送器或接收器使能的情况下，不能改变这三位（CPL,CPH,CLEN）的值。

时钟与已发送的数据同步。同步模式下的接收器按照发送器的时钟进行采样，没有任何过采样。

图 17-11. 同步模式下的 USART 示例

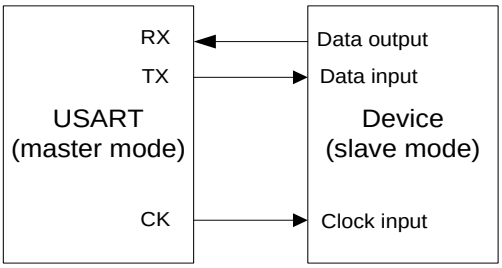
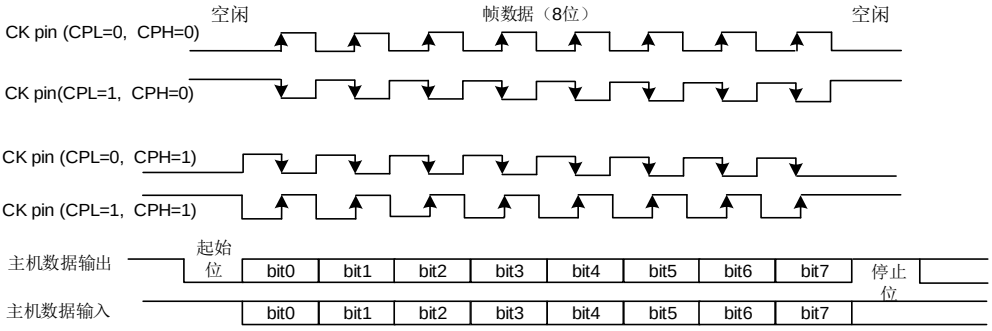


图 17-12. 8-bit 格式的 USART 同步通信波形（CLEN=1）

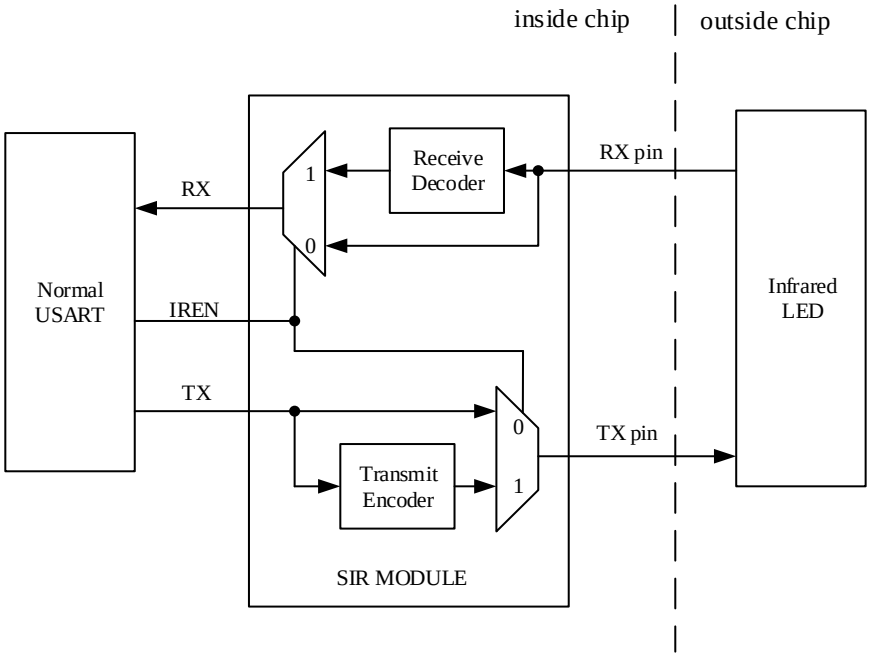


17.3.10. 串行红外（IrDA SIR）编解码功能模块

通过置位 USART_CTL2 寄存器中 IREN，使能 IrDA 模式。在 IrDA 模式下，USART_CTL1 寄存器的 LMEN，STB[1:0]，CKEN 位和 USART_CTL2 寄存器的 HDEN，SCEN 位将被清 0。

在 IrDA 模式下，USART 数据帧由 SIR 发送正交译码器进行调制，再被发送至红外 LED，调制后的信号经由红外 LED 进行发送，经解调后将数据发送至 USART 接收器。对于正交译码器而言，波特率应小于 115200。

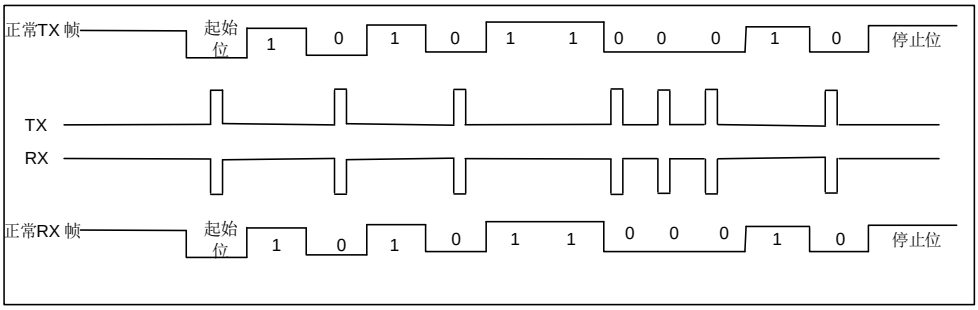
图 17-13. IrDA SIR ENDEC 模块



在 IrDA 模式下，TX 引脚与 RX 引脚电平不同。TX 引脚通常为低电平，RX 引脚通常为高电平。IrDA 引脚电平保持稳定代表逻辑‘1’，红外光源脉冲（RTZ 信号）代表逻辑‘0’。其脉冲宽度通常占一个位时间的 3/16。IrDA 无法检测到宽度小于一个 1 个 PSC 时钟的脉冲。如果脉冲宽度大于 1 但是小于 2 倍 PSC 时钟，IrDA 则无法可靠地检测到。

由于 IrDA 是一种半双工协议，因此在 IrDA SIR ENDEC 模块中，发送和接收不得同时进行。

图 17-14. IrDA 数据调制



将 USART_CTL2 寄存器中 IRLP 置位可以使 SIR 子模块工作在低功耗模式下。发送正交译码器由 PCLK 分频得到的低速时钟来驱动。分频系数在 USART_GP 寄存器中 PSC[7:0]位配置。TX 引脚脉冲宽度是低速时钟的 3 个周期。接收解码器工作模式与正常 IrDA 模式相同。

17.3.11. 半双工通信模式

通过设置 USART_CTL2 寄存器的 HDEN 位，可以使能半双工模式。在半双工通信模式下，USART_CTL1 寄存器的 LMEN，CKEN 位和 USART_CTL2 寄存器的 SCEN，IREN 位清零。

半双工模式下仅用单线通信。TX 引脚和 RX 引脚从内部连接到一起，RX 引脚不再使用。TX 引脚应被配置为开漏模式，通信冲突由软件处理。当 TEN 被置位时，在数据寄存器中的数据将会被发送。

17.3.12. 智能卡（ISO7816-3）模式

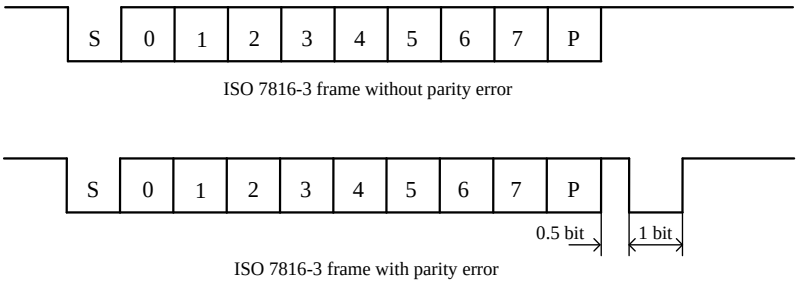
智能卡模式是一种异步通信模式，支持 ISO7816-3 协议。支持字节模式（T=0）和块模式（T=1）。将 USART_CTL2 寄存器的 SCEN 位置位可使能智能卡模式。在智能卡模式下，USART_CTL1 的 LMEN 位和 USART_CTL2 的 HDEN，IREN 位应该清零。

如果 CKEN 位被置位，USART 将向智能卡提供一个时钟。该时钟可以分频用于其他用途。

智能卡模式下的帧格式为：1 起始位+9 数据位（包括 1 个奇偶校验位）+1.5 停止位。

智能卡模式是一种半双工通信协议模式。当与智能卡连接时，TX 引脚须被设置成开漏模式，这个引脚将会与智能卡驱动同一条双向连线。

图 17-15. ISO7816-3 数据帧格式



字节模式（T=0）

相较于正常操作模式下的时序，从发送移位寄存器到 TX 引脚的传递时间延迟了半个波特率时钟，并且 TC 标志的置位将根据 USART_GP 寄存器的 GUAT[7:0]设置延迟某一特定时间。根据协议，USART 能自动重发数据，重发次数在 SCRTNUM 中设置。在上一个重复的字节接收结束，TC 位将会被立即置位而没有一个保护时间。如果 USART 连续收到 NACK 信号，那么在设定好的重发次数后，它将停止发送并将该错误标记为帧格式错误。USART_CMD 寄存器的 TXFCMD 位可将 TBE 位清 0。

在 USART 接收期间，在一帧数据的接收过程中，如果检测到有奇偶校验错误，TX 引脚将拉低一个波特率时钟。这个信号是发送‘NACK’信号到智能卡。然后智能卡一侧会产生一个帧错误。如果接收到的字节是错误的，RBNE 中断和接收 DMA 请求都不会被激活。根据协议，智能卡将重新发送数据。如果在最大的重新发送次数后（这个次数的具体值在 SCRTNUM 位域），接收到的字符仍然是错误的，USART 停止发送 NACK 信号并标注这个错误为奇偶校验错误。

若 USART_CTL2 中 NKEN 被置位，‘NACK’信号将被发送到 USART。USART 不会将‘NACK’误当作起始位。

空闲帧和断开帧在智能卡模式下不适用。

块模式（T=1）

在 T=1（块模式）下，USART_CTL2 寄存器的 NKEN 位应该清零来关闭校验错误发送。

当要从智能卡读取数据时，软件必须将 USART_RT 寄存器设置成 BWT（块等待）-11 的值，并将 RBNEIE 置位。如果到了这个时间，还没有从智能卡收到应答，将引起超时中断。如果在超时之前收到了第一个字节，则会引起 RBNE 中断。块模式下，如果用 DMA 从智能卡读取数

据，也只能在第一个字节接收完后再去使能 DMA。

在接收到第一个字节之后（RBNE 中断）必须将 USART_RT 寄存器设置为 CWT（字节等待时间）–11 的值（这个时间以波特时间作为单位），这是为了在两个连续的字符之间自动检测最大等待时间。如果智能卡在前一个字符发送结束后到设定的 CWT 周期之间没有发送字符，USART 会通过 RTF 标志提醒软件，当 RTIE 被置位时，会引起中断。

USART 用一个块长度计数器去统计收到的所有字符的长度。这个计数器在 USART 开始发送的时候自动清 0（TBE=0）。这个块长度信息位于智能卡发出数据的第三个字节（序言部分）。这个值必须写入 USART_RT 寄存器的 BL。当使用 DMA 模式时，在块开始之前，这个寄存器必须被设定为最小值（0x0）。有了这个值，在收到第四个字节后，会引起一个中断。软件必须从接收缓冲区读取第三个字节作为块长度。

在中断驱动接收模式，块的长度可以由软件提取出来并做检测或者通过设置 BL 的值得到。 但是在块开始之前，BL（0xFF）可以被设置为最大值。实际值则要在接收到第三个字节后写到寄存器中。

整个块的长度（包括序言区，收尾区和信息区）等于 BL+4。块尾通过 EBF 标志和相应中断提醒给软件（当 EBIE 位置 1 时）。如果块长度出错，将会引起一个 RT 中断。

直接和反向转换

智能卡协议定义了两种转换方式：直接转换和反向转换。

如果选择直接转换，从数据帧的最低位开始传输，TX 引脚高电平代表逻辑 ‘1’，偶校验。在这种情况下，MSBF 位和 DINV 位都设置为 0（默认值）。

如果选择反向转换，从数据帧的最高位开始传输，TX 引脚低电平代表逻辑 ‘1’，偶校验。在这种情况下，MSBF 位和 DINV 位都设置为 1。

17.3.13. ModBus 通信

通过实现块尾检测功能，USART 提供对 ModBus/RTU 和 ModBus/ASCII 协议实现的基本支持。

ModBus/RTU 这个模式下，块尾通过一个超过 2 个字符长度的空闲状态来识别。这个功能是通过一个可设置的超时检测功能来实现的。

为了检测空闲状态，USART_CTL1 寄存器的 RTEN 位和 USART_CTL0 寄存器的 RTIE 位必须置位。USART_RT 寄存器必须被设置成与 2 个字节超时所对应的值。在最后一个停止位被接收后，当接收线在这期间是空闲的，将产生一个中断，通知软件当前块接收已经完成。

在 ModBus/ASCII 模式下，块尾被认为是一个特定的字符（CR/LF）串。USART 用字符匹配机制实现这个功能。具体是通过将 LF 的 ASCII 码配置到 ADDR 区域并激活地址匹配中断（AMIE=1）来实现。软件将在收到 LF 或可以在 DMA 缓存中查找到 CR/LF 时得到提示。

17.3.14. 从 Deepsleep 模式唤醒

通过标准 RBNE 中断或 WUM 中断 USART 能从深度睡眠模式唤醒 MCU。

UESM 位必须置 1 并且 USART 时钟必须设置为 IRC8M 或 LXTAL（请参考 RCU 部分）。

当使用 RBNE 标准中断时，必须在进入深度睡眠模式前将 RBNEIE 位置位。

当使用 WUIE 中断时，WUIE 中断源可以通过 WUM 位来选择。

在进入深度睡眠模式前，必须禁用 DMA。在进入深度睡眠模式前，软件必须检测 USART 没有正在传送数据。这可以通过 USART_STAT 寄存器中的 BSY 标志来判断。REA 位必须被检测以确保 USART 是使能的。

当检测到唤醒事件时，无论 MCU 工作在深度睡眠模式还是正常模式，WUF 标志位通过硬件被置 1，并且在 WUIE 被置位的情况下，触发一个唤醒中断。

17.3.15. USART 中断

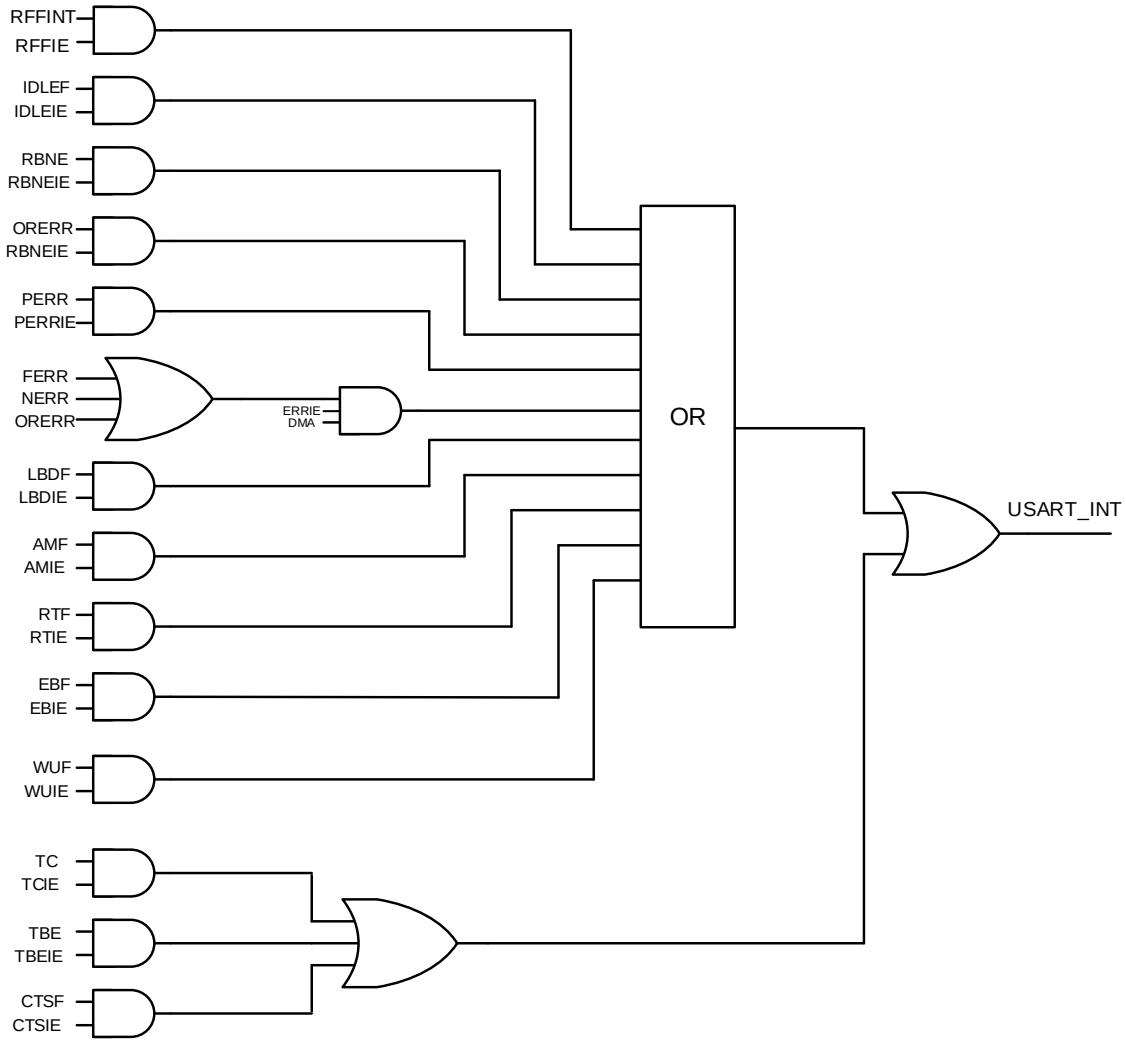
USART 中断事件和标志如*表 17-3. USART 中断请求*所示：

表 17-3. USART 中断请求

中断事件	事件标志	使能控制位
发送数据寄存器空	TBE	TBEIE
CTS标志	CTSF	CTSIE
发送结束	TC	TCIE
接收到的数据可以读取	RBNE	RBNEIE
检测到过载错误	ORERR	
检测到线路空闲	IDLEF	IDLEIE
奇偶校验错误	PERR	PERRIE
LIN模式下，检测到断开标志	LBDF	LBDIE
接收错误（噪声错误、溢出错误、帧错误）当DMA接收使能时	NERR或ORERR或FERR	ERRIE
字符匹配	AMF	AMIE
接收超时错误	RTF	RTIE
发现块尾	EBF	EBIE
从Deepsleep模式唤醒	WUF	WUIE

在发送给中断控制器之前，所有的中断事件是逻辑或的关系。因此在任何时候 USART 只能向控制器产生一个中断请求。不过软件可以在一个中断服务程序里处理多个中断事件。

图 17-16. USART 中断映射框图



17.4. USART 寄存器

USART0 基地址：0x4001 3800
USART1 基地址：0x4000 4400

17.4.1. USART 控制寄存器 0（USART_CTL0）

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				EBIE	RTIE	DEA[4:0]					DED[4:0]				
				rw	rw	rw					rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVSMOD	AMIE	MEN	WL	WM	PCEN	PM	PERRIE	TBEIE	TCIE	RBNEIE	IDLEIE	TEN	REN	UESM	UEN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:28	保留	必须保持复位值。
27	EBIE	块尾中断使能。 0：中断禁止 1：中断使能 在 USART1，该位保留。
26	RTIE	接收超时中断使能。 0：中断禁止 1：中断使能 在 USART1，该位保留。
25:21	DEA[4:0]	驱动使能置位时间。 这些数字用来定义 DE（驱动使能）信号的置位与第一个字节的起始位之间的时间间隔。它以采样时间为单位（1/8 或 1/16 位时间），可以通过 OVSMOD 位来配置。 当 USART 被使能（UEN=1）时，该位域不能被改写。
20:16	DED[4:0]	驱动使能置低时间 这些位用来定义一个发送信息最后一个字节的停止位与置低 DE（驱动使能）信号之间的时间间隔。它以采样时间为单位（1/8 或 1/16 位时间），可以通过 OVSMOD 位来配置。 当 USART 被使能（UEN=1）时，该位域不能被改写。
15	OVSMOD	过采样模式 0：16 倍过采样 1：8 倍过采样 在 LIN，IrDA 和智能卡模式，该位保持清 0。

		当 USART 被使能（UEN=1）时，该位域不能被改写。
14	AMIE	ADDR 字符匹配中断使能。 0：ADDR 字符匹配中断禁用。 1：ADDR 字符匹配中断使能。
13	MEN	静默模式使能。 0：静默模式禁用。 1：静默模式被使能。
12	WL	字长 0：8 数据位 1：9 数据位 当 USART 被使能（UEN=1）时，该位域不能被改写。
11	WM	从静默模式唤醒方法。 0：空闲线 1：地址标记 当 USART 被使能（UEN=1）时，该位域不能被改写。
10	PCEN	校验控制使能。 0：校验控制禁用。 1：校验控制被使能。 当 USART 被使能（UEN=1）时，该位域不能被改写。
9	PM	校验模式 0：偶校验 1：奇校验 当 USART 被使能（UEN=1）时，该位域不能被改写。
8	PERRIE	校验错误中断使能。 0：校验错误中断禁用。 1：当 USART_STAT 寄存器的 PERR 位置位时，将触发中断。
7	TBEIE	发送寄存器空中断使能。 0：中断禁止。 1：当 USART_STAT 寄存器的 TBE 位置位时，将触发中断。
6	TCIE	发送完成中断使能。 0：发送完成中断禁用。 1：当 USART_STAT 寄存器的 TC 位置位时，将触发中断。
5	RBNEIE	读数据缓冲区非空中断和过载错误中断使能。 0：读数据缓冲区非空中断和过载错误中断禁用。 1：当 USART_STAT 寄存器的 ORERR 或 RBNE 位置位时，将触发中断。
4	IDLEIE	IDLE 线检测中断使能。 0：IDLE 线检测中断禁用。 1：当 USART_STAT 寄存器的 IDLEF 位置位时，将触发中断。

3	TEN	发送器使能。 0：发送器关闭。 1：发送器打开。
2	REN	接收器使能。 0：接收器关闭。 1：接收器打开并且开始搜索起始位。
1	UESM	USART 在深度睡眠模式下使能。 0：USART 不能从深度睡眠模式唤醒 MCU。 1：USART 能从深度睡眠模式唤醒 MCU。条件是 USART 的时钟源必须是 IRC8M 或 LXTAL。 在 USART1，该位保留。
0	UEN	USART 使能 0：USART 预分频器和输出禁用。 1：USART 预分频器和输出被使能。

17.4.2. USART 控制寄存器 1（USART_CTL1）

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR[7:0]								RTEN	保留			MSBF	DINV	TINV	RINV
rw								rw	rw			rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STRP	LMEN	STB[1:0]		CKEN	CPL	CPH	CLEN	保留	LBDIE	LBLEN	ADDM	保留			
rw	rw	rw		rw	rw	rw	rw		rw	rw	rw				

位/位域	名称	描述
31:24	ADDR[7:0]	USART 的节点地址。 这些位给出 USART 的节点地址。 在多处理器通信并且静默模式或者深度睡眠模式期间，这些位用来唤醒进行地址匹配的检测。接收到的最高位为 1 的数据帧将和这些位进行比较。当 ADDM 位被清零时，仅仅 ADDR[3:0]被用来比较。 在正常的接收期间，这些位也用来进行字符检测。所有接收到的字符（8 位）与 ADDR[7:0]的值进行比较，如果匹配，AMF 标志将被置位。 当接收器（REN=1）和 USART（UEN=1）被使能时，该位域不能被改写。
23	RTEN	接收器超时使能。 0：接收器超时功能禁用。 1：接收器超时功能被使能。 在 USART1，该位保留。

22:20	保留	必须保持复位值。
19	MSBF	高位在前 0：数据发送/接收，采用低位在前。 1：数据发送/接收，采用高位在前。 USART 被使能(UEN=1)时，该位域不能被改写。
18	DINV	数据位反转。 0：数据位信号值没有反转。 1：数据位信号值被反转。 USART 被使能(UEN=1)时，该位域不能被改写。
17	TINV	TX 管脚电平反转。 0：TX 管脚信号值没有反转。 1：TX 管脚信号值被反转。 USART 被使能(UEN=1)时，该位域不能被改写。
16	RINV	RX 管脚电平反转。 0：RX 管脚信号值没有反转。 1：RX 管脚信号值被反转。 USART 被使能(UEN=1)时，该位域不能被改写。
15	STRP	交换 TX/RX 管脚。 0：TX 和 RX 管脚功能不被交换。 1：TX 和 RX 管脚功能被交换。 当 USART 被使能（UEN=1）时，该位域不能改写。
14	LMEN	LIN 模式使能。 0：LIN 模式关闭。 1：LIN 模式开启。 USART 被使能（UEN=1）时，该位域不能被改写。 在 USART1，该位保留。
13:12	STB[1:0]	STOP 位长 00：1 停止位 01：0.5 停止位 10：2 停止位 11：1.5 停止位 USART 被使能（UEN=1）时，该位域不能被改写。
11	CKEN	CK 管脚使能 0：CK 管脚禁用 1：CK 管脚被使能 USART 被使能（UEN=1）时，该位域不能被改写。 在 USART1，该位保留。
10	CPL	时钟极性 0：在同步模式下，CK 管脚不对外发送时保持为低电平 1：在同步模式下，CK 管脚不对外发送时保持为高电平

		USART 被使能（UEN=1）时，该位域不能被改写。
9	CPH	时钟相位 0：在同步模式下，在首个时钟边沿采样第一个数据。 1：在同步模式下，在第二个时钟边沿采样第一个数据。 USART 被使能（UEN=1）时，该位域不能被改写。
8	CLEN	CK 长度 0：在同步模式下，最后一位(MSB)的时钟脉冲不输出到 CK 管脚。 1：在同步模式下，最后一位(MSB)的时钟脉冲输出到 CK 管脚。 USART 被使能（UEN=1）时，该位域不能被改写。
7	保留	必须保持复位值。
6	LBDIE	LIN 断开信号检测中断使能。 0：断开信号检测中断禁用。 1：当 USART_STAT 的 LBDF 位置位，将产生中断。 在 USART1，该位保留。
5	LBDL	LIN 断开帧长度。 0：检测 10 位断开帧。 1：检测 11 位断开帧。 USART 被使能（UEN=1）时，该位域不能被改写。 在 USART1，该位保留。
4	ADDM	地址检测模式 这位用来选择 4 位地址检测还是全位地址检测。 0：4 位地址检测。 1：全位地址检测。在 7 位，8 位和 9 位数据模式下，地址检测分别按 6 位，7 位和 8 位地址（ADDR[5:0], ADDR[6:0] 和 ADDR[7:0]）执行。 USART 被使能（UEN=1）时，该位域不能被改写。
3:0	保留	必须保持复位值。

17.4.3. USART 控制寄存器 2（USART_CTL2）

地址偏移：0x08

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留									WUIE	WUM[1:0]		SCRTNUM[2:0]			保留
									rw	rw		rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEP	DEM	DDRE	OVRD	OSB	CTSIE	CTSEN	RTSEN	DENT	DENR	SCEN	NKEN	HDEN	IRLP	IREN	ERRIE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述

31:23	保留	必须保持复位值。
22	WUIE	从深度睡眠模式唤醒中断使能。 0：从深度睡眠模式唤醒中断禁用。 1：从深度睡眠模式唤醒中断被使能。 在 USART1，该位保留。
21:20	WUM[1:0]	从深度睡眠模式唤醒模式。 这个位域指定什么事件可以置位 USART_STAT 寄存器中的 WUF（从深度睡眠唤醒标志）标志。 00：WUF 在地址匹配的时候置位。如何实现地址匹配在 ADDR 和 ADDM 中定义。 01：保留 10：WUF 在检测到起始位时置位。 11：WUF 在检测到 RBNE 时置位。 USART 被使能（UEN=1）时，该位域不能被改写。 在 USART1，该位保留。
19:17	SCRTNUM[2:0]	智能卡自动重试数目 在智能卡模式下，这些位用来指定在发送和接收时重试的次数。在发送模式下，它指的是在产生发送错误（FERR 位置位）之前自动重试的发送次数。 在接收模式下，它指的是在产生接收错误（RBNE 位和 PERR 位置位）之前自动重试的接收次数。 当这些位被设置为 0x0 时，在发送模式下这些位将不会自动发送。 USART 被使能（UEN=1）时，该位域被清零，并停止重发。 在 USART1，该位保留。
16	保留	必须保持复位值。
15	DEP	驱动使能的极性选择模式。 0：DE 信号高有效 1：DE 信号低有效 USART 被使能（UEN=1）时，该位域不能被改写。
14	DEM	驱动使能模式 用户使能该位以后，可以通过 DE 信号对外部收发器进行控制。DE 信号是从 RTS 管脚输出的。 0：DE 功能禁用 1：DE 功能开启 USART 被使能（UEN=1）时，该位域不能被改写。
13	DDRE	在接收错误时屏蔽 DMA 请求 0: 在发生接收错误的情况下，不禁用 DMA。所有的错误数据不会产生 DMA 请求，以确保错误的数据不会被传输，但是下一个接收到的正确的数据会被传输。在发生接收错误时，RBNE 位保持 0 以阻止过载错误，但是相应错误标志位会被置位。这种模式可用于智能卡模式。 1: 在接收错误的情况下，DMA 请求会被屏蔽，直到相应的标志位被清 0。RBNE 标志和相应的错误标志位会被置位。软件在清除错误标志前，必须首先失能 DMA 接收（DMAR = 0）或清 RBNE。

		USART 被使能（UEN=1）时，该位域不能被改写。
12	OVRD	溢出禁止 0：溢出功能被使能。当接收到的数据在新数据到达前没有被读走，ORERR 错误标志位将被置位，并且新数据将会丢失。 1：溢出功能禁止。当接收到的数据在新数据到达前没有被读走，ORERR 错误标志位将不会被置位，新数据会将 USART_ RDATA 寄存器以前的内容覆盖。 USART 被使能（UEN=1）时，该位域不能被改写。
11	OSB	单次采样方式。 0：三次采样方法。 1：一次采样方法。 USART 被使能（UEN=1）时，该位域不能被改写。
10	CTSIE	CTS 中断使能 0：CTS 中断屏蔽 1：当 USART_STAT 的 CTS 位置位时，会产生中断。
9	CTSEN	CTS 使能 0：CTS 硬件流控禁用。 1：CTS 硬件流控被使能。 USART 被使能（UEN=1）时，该位域不能被改写。
8	RTSEN	RTS 使能 0：RTS 硬件流控禁用。 1：RTS 硬件流控被使能，只有当接收缓冲区有空间的时候，才会请求下一个数据。 USART 被使能（UEN=1）时，该位域不能被改写。
7	DENT	DMA 发送使能。 0：关闭 DMA 发送模式。 1：开启 DMA 发送模式。
6	DENR	DMA 接收使能。 0：关闭 DMA 接收模式。 1：开启 DMA 接收模式。
5	SCEN	智能卡模式使能。 0：智能卡模式禁用。 1：智能卡模式使能。 USART 被使能（UEN=1）时，该位域不能被改写。 在 USART1 中，该位保留。
4	NKEN	在智能卡模式 NACK 使能。 0：当出现校验错误时不发送 NACK。 1：当出现校验错误时发送 NACK。 USART 被使能（UEN=1）时，该位域不能被改写。 在 USART1 中，该位保留。
3	HDEN	半双工使能

		0：禁用半双工模式。 1：开启半双工模式。 USART 被使能（UEN=1）时，该位域不能被改写。
2	IRLP	IrDA 低功耗模式 0：正常模式 1：低功耗模式 USART 被使能（UEN=1）时，该位域不能被改写。
1	IREN	IrDA 模式使能 0：IrDA 禁用 1：IrDA 被使能 USART 被使能(UEN=1)时，该位域不能被改写。 在 USART1 中，该位保留。
0	ERRIE	多级缓存通信模式的错误中断使能 0：禁用错误中断。 1：在多级缓存通信时，当 USART_STAT 寄存器的 FERR 位，ORERR 位或 NERR 位被置位时，会产生中断。

17.4.4. USART 波特率寄存器（USART_ BAUD）

地址偏移：0x0C
复位值：0x0000 0000

该寄存器只能按字（32位）访问。
当USART（UEN=1）被使能时，该寄存器不能被改写。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

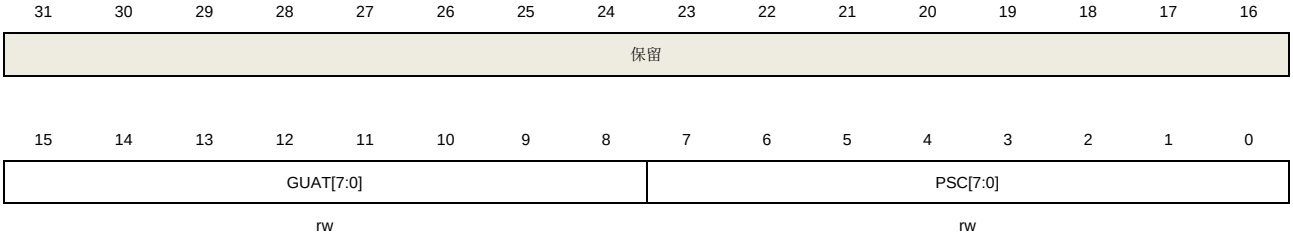
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BRR [15:4]												BRR[3:0]			
rw												rw			

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:4	BRR[15:4]	波特率分频系数的整数部分。 DIV_INT[11:0] = BRR[15:4]。
3:0	BRR[3:0]	波特率分频系数的小数部分。 如果 OVSMOD = 0，USARTDIV [3:0] = BRR [3:0]。 如果 OVSMOD = 1，USARTDIV [3:1] = BRR [2:0]，BRR [3]必须被置 0。

17.4.5. USART 保护时间和预分频器寄存器（USART_GP）

地址偏移：0x10
复位值：0x0000 0000

该寄存器只能按字（32位）访问。
USART被使能（UEN=1）时，该寄存器不能被改写。
在USART1中，该寄存器保留。

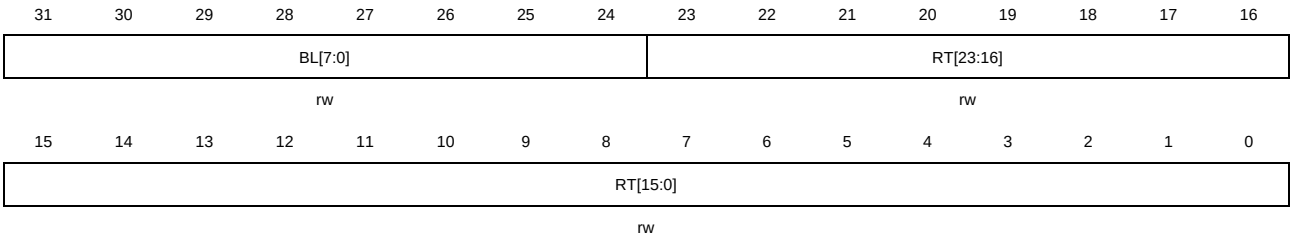


位/位域	名称	描述
31:16	保留	必须保持复位值。
15:8	GUAT[7:0]	在智能卡模式下的保护时间值。 USART 被使能（UEN=1）时，该位域不能被改写。
7:0	PSC[7:0]	预分频器值 在红外低功耗模式下，对系统时钟进行分频已获得低功耗模式下的频率。寄存器的值是分频系数。 00000000: 保留 – 不设置这个值 00000001: 1 分频 00000010: 2 分频 ... 在 IrDA 正常模式下的分频值 00000001: 仅仅能设为这个值 在智能卡模式下，对系统时钟进行分频的值存于 PSC[4:0]位域中。PSC[7:5]位保持为复位值。分频系数是寄存器中值的两倍。 00000：保留 -不设置这个值 00001：2 分频 00010：4 分频 00011：6 分频 ... USART 被使能（UEN=1）时，该位域不能被改写。

17.4.6. USART 接收超时寄存器（USART_RT）

地址偏移：0x14
复位值：0x0000 0000

该寄存器只能按字（32位）访问。
在USART1中，该寄存器保留。

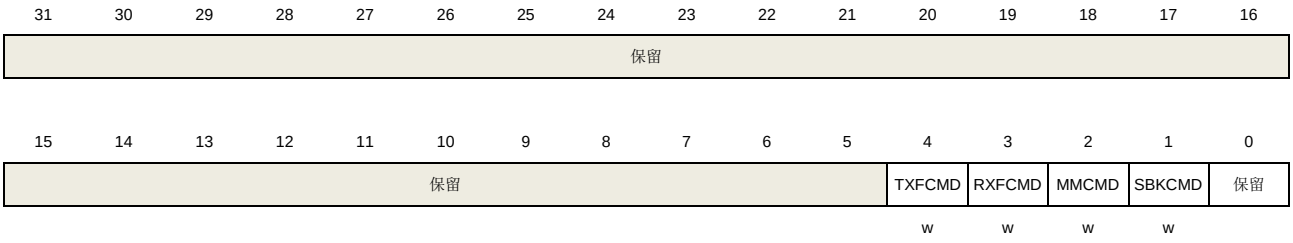


位/位域	名称	描述
31:24	BL[7:0]	块长度 这些位给出了智能卡 T=1 的接收时块的长度。它的值等于信息字节的长度+结束部分的长度（1-LEC/2-CRC）– 1。 这个值可以在块接收开始时设置（用于需要从块的序言提取块的长度的情形），这个只在每一个接收时钟周期只能设置一次。在智能卡模式下，当 TBE=0 时，块的长度计数器被清 0。 在其他模式下，当 REN=0（禁用接收器）并且/或者当 EBC 位被写 1 时块的长度计数器被清 0。
23:0	RT[23:0]	接收器超时门限。 该位域指定接收超时值，单位是波特时钟的时长。 标准模式下，如果在最后一个字节接收后，在 RT 规定的时长内，没有检测到新的起始位，RTF 标志被置位。 在智能卡模式，这个值被用来实现 CWT 和 BWT。在这种情况下，超时检测是从最后一个接收字节的起始位开始。 这些位可以在工作时改写。假如一个新数据到来的时间比 RT 规定的晚，RTF 标志会被置位。 对于每个接收字符，这个值只能改写一次。

17.4.7. USART 请求寄存器（USART_CMD）

地址偏移：0x18
复位值：0x0000 0000

该寄存器只能按字（32位）访问。



位/位域	名称	描述
31:5	保留	必须保持复位值。
4	TXFCMD	发送数据清空请求。 向该位写 1 去置位 TBE 标志位，以取消发送数据。

		在 USART1 中，该位保留。
3	RXFCMD	接收数据清空请求 向该位写 1 来清除 RBNE 标志位，以丢弃未读的接收数据。
2	MMCMD	静默模式请求。 向该位写 1 使 USART 进入静默模式并且置位 RWU 标志位。
1	SBKCMD	发送断开帧请求。 向该位写 1 置位 SBKF 标志并使 USART 在空闲时发送一个断开帧。
0	保留	必须保持复位值。

17.4.8. USART 状态寄存器（USART_STAT）

地址偏移：0x1C
复位值：0x0000 00C0

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留									REA	TEA	WUF	RWU	SBF	AMF	BSY
									r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留			EBF	RTF	CTS	CTSF	LBDF	TBE	TC	RBNE	IDLEF	ORERR	NERR	FERR	PERR
	r		r	r	r	r	r	r	r	r	r	r	r	r	r

位/位域	名称	描述
31:23	保留	必须保持复位值。
22	REA	接收使能通知标志。 这位反映了 USART 核心逻辑的接收使能状态，该位可以通过硬件设置。 0：USART 核心接收逻辑禁用。 1：USART 核心接收逻辑被使能。
21	TEA	发送使能通知标志。 这位反映了 USART 核心逻辑的发送使能状态，该位可以通过硬件设置。 0：USART 核心发送逻辑禁用。 1：USART 核心发送逻辑被使能。
20	WUF	从深度睡眠模式唤醒标志。 0：没有从深度睡眠模式唤醒。 1：已从深度睡眠模式唤醒， 如果在 USART_CTL2 寄存器的 WUFIE=1 并且 MCU 处于深度睡眠模式，将引发一个中断。 当检测到一个唤醒事件时，该位通过硬件置位，这个事件在 WUM 位域被定义。 向 USART_INTC 寄存器中的 WUC 写 1，该位被清 0。 当 UESM 被清 0 时，该位清 0。 在 USART1 中，该位保留。
19	RWU	接收器从静默模式唤醒。

		这位表示 USART 处于静默模式。 0：接收器在工作状态。 1：接收器在静默状态。 当在唤醒和静默模式切换时，它通过硬件清 0 或者置 1。静默模式控制（地址帧还是空闲帧）是用通过 USART_CTL0 寄存器的 WM 位选择。 如果选择空闲信号唤醒，只能通过向 USART_CMD 寄存器的 MMCMD 位写 1 来将该位置位。
18	SBF	断开信号发送标识。 0：没发送断开字符。 1：将要发送断开字符。 该位表示一个断开发送信号被请求。 通过向 USART_CMD 寄存器的 SBKCMD 写 1 来置位。 在断开帧的停止位发送期间，硬件清 0。
17	AMF	ADDR 匹配标志。 0：ADDR 和接收到的字符不匹配。 1：ADDR 和接收到的字符匹配，如果 USART_CTL0 寄存器的 AMIE=1，将引发一个中断。 当接收到 ADDR [7:0]中定义的字符时，硬件置位。 通过向 USART_ INTC 寄存器的 AMC 写 1 清 0。
16	BSY	忙标志 0：USART 处于空闲。 1：USART 正在接收。
15:13	保留	必须保持复位值。
12	EBF	块结束标志 0：块没有结束 1：块结束已到（足够的字节数），如果 USART_CTL1 寄存器的 EBIE=1，将引发一个中断。 当接收到的字节数（从块开始，包括序言部分）等于或大于 BLEN + 4，硬件置位。 通过向 USART_INTC 寄存器的 EBC 写 1 清 0。 在 USART1 中，该位保留。
11	RTF	接收超时标志。 0：尚未超时 1：已经超时，如果 USART_CTL1 寄存器的 RTIE 被置位，将会引发中断。 如果空闲的时间已经超过了在 USART_RT 寄存器中设定的 RT 值，通过硬件置 1。 通过向 USART_INTC 寄存器的 RTC 位写 1 清 0。 在智能卡模式，这个超时相当于 CWT 或 BWT 计时。 在 USART1 中，该位保留。
10	CTS	CTS 电平 这个值等于 nCTS 输入引脚电平的反向拷贝。 0：nCTS 输入引脚高电平。

		1：nCTS 输入引脚低电平。
9	CTSF	CTS 变化标志 0：nCTS 状态线没有变化。 1：nCTS 状态线发生变化，如果 USART_CTL2 寄存器的 CTSIE 位置位，将引发中断。 当 nCTS 输入变化时，由硬件置位。 通过向 USART_INTC 寄存器的 CTSC 位写 1，清零该位。
8	LBDF	LIN 断开检测标志。 0：没有检测到 LIN 断开字符。 1：检测到 LIN 断开字符。当 USART_CTL1 寄存器的 LBDIE 位被置位时，将会有中断产生。 当 LIN 断开帧被检测到的时候，硬件置位。 通过向 USART_INTC 寄存器的 LBDC 位写 1，清零该位。 在 USART1 中，该位保留。
7	TBE	发送数据寄存器空。 0：数据没有发送到移位寄存器。 1：数据发送到移位寄存器。如果 USART_CTL0 寄存器的 TBEIE 位置位，将会有中断产生。 当 USART_TDATA 寄存器的内容已经被转移到移位寄存器或者向 USART_CMD 寄存器的 TXFCMD 位写 1 时，由硬件置位。 通过向 USART_TDATA 寄存器中写数据来清 0。
6	TC	发送完成 0：发送没有完成。 1：发送完成。如果 USART_CTL0 寄存器的 TCIE 被置位，将会有中断产生。 如果一个包含数据的帧的发送完成且 TBE 被置位，该位由硬件置位。 通过向 USART_INTC 寄存器的 TCC 位写 1 清 0。
5	RBNE	读数据缓冲区非空。 0：没有接收到数据。 1：已接收到数据并且可以读取。当寄存器 USART_CTL0 的 RBNEIE 位被置位，将会有中断产生。 当接收移位寄存器的内容已经被转移到寄存器 USART_ RDATA，由硬件置位。 通过读 USART_RDATA 寄存器或向 USART_CMD 寄存器的 RXFCMD 位写 1 清 0。
4	IDLEF	空闲线检测标志。 0：没检测到空闲线。 1：检测到空闲线。,如果 USART_CTL0 寄存器的 IDLEIE 位置 1，将会有中断产生。 当检测到空闲线时，通过硬件置位。直到 RBNE 位置位，否则它不会被再次置位。 向 USART_INTC 寄存器的 IDLEC 位写 1 清 0。
3	ORERR	溢出错误 0：未检测到溢出错误。 1：检测到溢出错误。在多级缓存通信中，如果寄存器 USART_CTL0 的 RBNEIE 位置位，将会引发中断。如果寄存器 USART_CTL2 的 ERRIE 位置位也会引发中断。

在 RBNE 置位的情况下，如果接收移位寄存器的数据传递给 USART_RDATA 寄存器，将会由硬件置位。
向 USART_INTC 寄存器的 OREC 位写 1 清 0。

2	NERR	噪声错误标志 。 0：未检测到噪声错误。 1：检测到噪声错误。在多级缓存通信中，如果寄存器 USART_CTL2 的 ERRIE 位置位，将会有中断产生。 在接收帧的时候检测到噪声错误，将会由硬件置位。 向寄存器 USART_INTC 的 NEC 位写 1 清 0。
1	FERR	帧错误 0：未检测到帧错误。 1：检测到帧错误或者断开字符。在多级缓存通信中，如果寄存器 USART_CTL2 的 ERRIE 位置位，将会有中断产生。 当一个不同步，强噪声或者断开字符被检测到时，硬件置位。在智能卡模式下，当发送次数达到上限，仍然没有收到成功发送应答（卡一直响应 NACKs），该位也将被置位。 向 USART_INTC 寄存器的 FEC 位写 1 清 0。
0	PERR	校验错误 0：未检测到校验错误。 1：检测到校验错误 在多级缓存通信中，如果寄存器 USART_CTL0 的 PERRIE 位置位，将会有中断产生。 当在接收模式的时候检测到校验错误，将会由硬件置位。 向 USART_INTC 寄存器的 PEC 位写 1 清 0。

17.4.9. USART 中断标志清除寄存器（USART_INTC）

地址偏移：0x20
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留											WUC	保留		AMC	保留
											w			w	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留			EBC	RTC	保留	CTSC	LBDC	保留	TCC	保留	IDLEC	OREC	NEC	FEC	PEC
			w	w		w	w		w		w	w	w	w	w

位/位域	名称	描述
31:21	保留	必须保持复位值。
20	WUC	从深度睡眠模式唤醒标志的清除。 向该位写 1 清除 USART_STAT 寄存器的 WUF 位。 在 USART1 中，该位保留。

19:18	保留	必须保持复位值。
17	AMC	ADDR 匹配标志清除。 向该位写 1 清除 USART_STAT 寄存器的 AMF 位。
16:13	保留	必须保持复位值。
12	EBC	块结束标志清除。 向该位写 1 清除 USART_STAT 寄存器的 EBF 位。 在 USART1 中，该位保留。
11	RTC	接收超时标志清除。 向该位写 1 清除 USART_STAT 寄存器的 RTF 标志。 在 USART1 中，该位保留。
10	保留	必须保持复位值。
9	CTSC	CTS 变化标志清除。 向该位写 1 清除 USART_STAT 寄存器的 CTSF 位。
8	LBDC	LIN 断开字符检测标志清除 向该位写 1 清除 USART_STAT 寄存器的 LBDF 标志位。 在 USART1 中，该位保留。
7	保留	必须保持复位值。
6	TCC	发送完成标志清除。 向该位写 1 清除 USART_STAT 寄存器的 TC 位。
5	保留	必须保持复位值。
4	IDLEC	空闲线检测标志清除。 向该位写 1 清除 USART_STAT 寄存器的 IDLEF 位。
3	OREC	溢出标志清除 向该位写 1 清除 USART_STAT 寄存器的 ORERR 位。
2	NEC	噪声检测清除 向该位写 1 清除 USART_STAT 寄存器的 NERR 位。
1	FEC	帧格式错误标志清除。 向该位写 1 清除 USART_STAT 寄存器的 FERR 位。
0	PEC	校验错误标志清除。 向该位写 1 清除 USART_STAT 寄存器的 PERR 位。

17.4.10. USART 数据接收寄存器 (USART_RDATA)

地址偏移：0x24
复位值：未定义

该寄存器只能按字（32位）访问。

位/位域	名称	描述
31:9	保留	必须保持复位值。
8:0	RDATA[8:0]	接收数据的值。 包含接收到的数据字节。 如果接收到的数据打开了奇偶校验位（USART_CTL0 寄存器的 PCEN 置 1），那么接收到的数据的最高位（第 7 位或 8 位，取决于数据的长度）是奇偶校验位。

17.4.11. USART 数据发送寄存器（USART_TDATA）

地址偏移：0x28
复位值：未定义

该寄存器只能按字（32 位）访问。

位/位域	名称	描述
31:9	保留	必须保持复位值。
8:0	TDATA[8:0]	发送数据的值 包含发送的数据字节。 如果发送到的数据打开了奇偶校验位（USART_CTL0 寄存器的 PCEN 置 1），那么发送的数据的最高位（第 7 位或 8 位取决于数据的长度）将会被奇偶校验位替代。 只有当 USART_STAT 寄存器的 TBE 位被置位时，这个寄存器才可以改写。

18. 内部集成电路总线接口（I2C）

18.1. 简介

I2C（内部集成电路总线）模块提供了符合工业标准的两线串行制接口，可用于 MCU 和外部 I2C 设备的通讯。I2C 总线使用两条串行线：串行数据线 SDA 和串行时钟线 SCL。

I2C 接口模块实现了 I2C 协议的标速模式和快速模式，具备 CRC 计算和校验功能、支持 SMBus（系统管理总线）和 PMBus（电源管理总线），此外还支持多主机 I2C 总线架构。I2C 接口模块也支持 DMA 模式，可有效减轻 CPU 的负担。

18.2. 主要特征

- 并行总线至 I2C 总线协议的转换及接口。
- 同一接口既可实现主机功能又可实现从机功能。
- 主从机之间的双向数据传输。
- 支持 7 位和 10 位的地址模式和广播寻址。
- 支持 I2C 多主机模式。
- 支持标速模式（最高 100 kHz）和快速模式（最高 400 kHz）。
- 从机模式下可配置的 SCL 主动拉低。
- 支持 DMA 模式。
- 兼容 SMBus 2.0 和 PMBus。
- 两个中断：字节成功传输中断和错误事件中断。
- 可选择的 PEC（报文错误校验）生成和校验。

18.3. 功能说明

I2C 接口的内部结构如*图 18-1. I2C 模块框图*所示。

图 18-1. I2C 模块框图

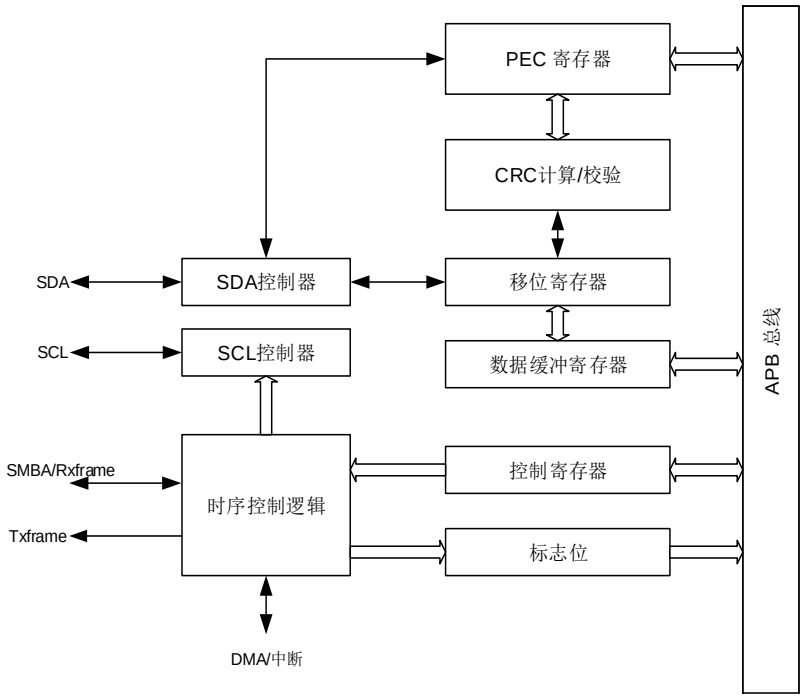


表 18-1. I2C 总线术语说明（参考飞利浦 I2C 规范）

术语	说明
发送器	发送数据到总线的设备
接收器	从总线接收数据的设备
主机	初始化数据传输，产生时钟信号和结束数据传输的设备
从机	由主机寻址的设备
多主	多个主机可以尝试在不破坏信息的前提下同时控制总线
同步	同步两个或更多设备之间的时钟信号的过程
仲裁	如果超过一个主机同时试图控制总线，只有一个主机被允许，且获胜主机的信息不被破坏

18.3.1. SDA 线和 SCL 线

I2C 模块有两条接口线：串行数据 SDA 线和串行时钟 SCL 线。连接到总线上的设备通过这两根线互相传递信息。

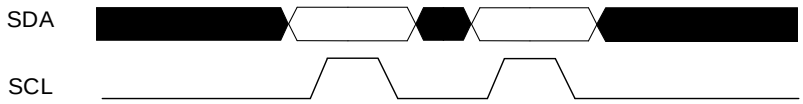
SDA 和 SCL 都是双向线，通过一个电流源或者上拉电阻接到电源正极。当总线空闲时，两条线都是高电平。连接到总线的设备输出极必须是开漏或者开集，以提供线与功能。I2C 总线上的数据在标准模式下可以达到 100 Kbit/s，在快速模式下可以达到 400 Kbit/s。由于 I2C 总线上可能会连接不同工艺的设备（CMOS，NMOS，双极性器件），逻辑‘0’和逻辑‘1’的电平并不是固定的，取决于 V_{DD} 的实际电平。

18.3.2. 数据有效性

时钟信号的高电平期间 SDA 线上的数据必须稳定。只有在时钟信号 SCL 变低的时候数据线

SDA 的电平状态才能跳变（如*图 18-2. 数据有效性*）。每个数据比特传输需要一个时钟脉冲。

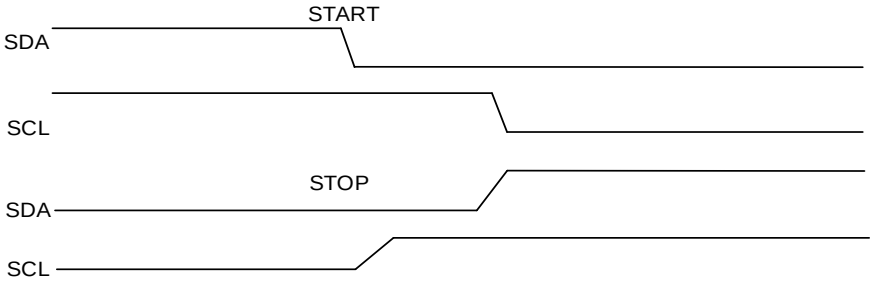
图 18-2. 数据有效性



18.3.3. 开始和停止信号

所有的数据传输起始于一个 START 结束于一个 STOP（参见*图 18-3. 起始和停止信号*）。START 信号定义为，在 SCL 为高时，SDA 线上出现一个从高到低的电平转换。STOP 信号定义为，在 SCL 为高时，SDA 线上出现一个从低到高的电平转换。

图 18-3. 起始和停止信号

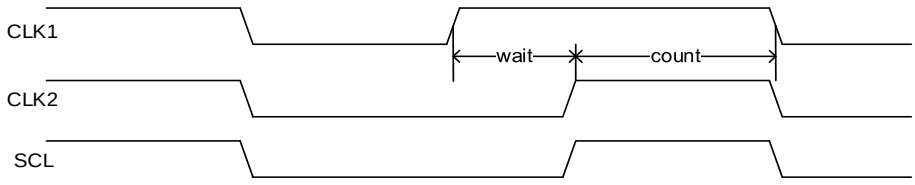


18.3.4. 时钟同步

两个主机可以同时在空闲总线上开始传送数据，因此必须通过一些机制来决定哪个主机获取总线的控制权并完成传输，这一般是通过时钟同步和仲裁来完成的。单主机系统下不需要时钟同步和仲裁机制。

时钟同步通过 SCL 线的线与来实现。这就是说 SCL 线的高到低切换会使器件开始计数它们的低电平周期，而且当主机的时钟变低电平时，它会使 SCL 线保持这种状态直到到达时钟的高电平（参见*图 18-4. 时钟同步*）。但是如果另一个时钟仍处于低电平周期，这个时钟的低到高切换不会改变 SCL 线的状态。因此 SCL 线被有最长低电平周期的器件保持低电平。此时低电平周期短的器件会进入高电平的等待状态。

图 18-4. 时钟同步



18.3.5. 仲裁

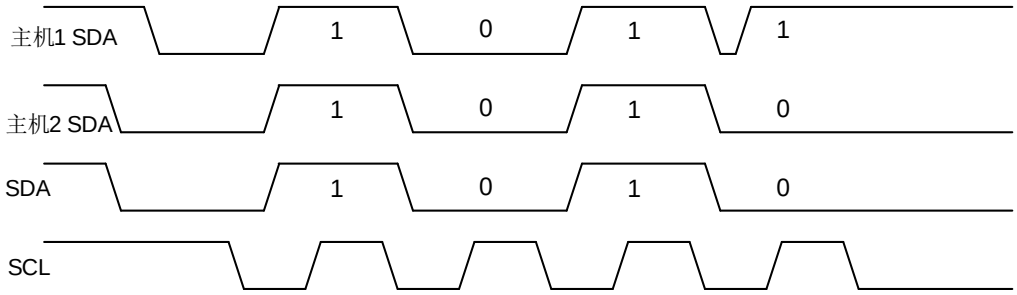
仲裁和同步一样，都是为了解决多主机情况下的总线控制冲突。仲裁的过程与从机无关。

只有在总线空闲的时候主机才可以启动传输。两个主机可能在START信号的最短保持时间内在

总线上产生一个有效的START信号，这种情况下需要仲裁来决定由哪个主机来完成传输。

仲裁逐位进行，在每一位的仲裁期间，当SCL为高时，每个主机都检查SDA电平是否和自己发送的相同。仲裁的过程需要持续很多位。理论上讲，如果两个主机所传输的内容完全相同，那么它们能够成功传输而不出现错误。如果一个主机发送高电平但检测到SDA电平为低，则认为自己仲裁失败并关闭自己的SDA输出驱动，而另一个主机则继续完成自己的传输。

图 18-5. SDA 线仲裁



18.3.6. I2C 通讯流程

每个I2C设备（不管是微控制器，LCD驱动，存储器或者键盘接口）都通过唯一的地址进行识别，根据设备功能，他们既可以是发送器也可作为接收器。

I2C从机检测到I2C总线上的START信号之后，就开始从总线上接收地址，之后会把从总线接收到的地址和自身的地址（通过软件编程）进行比较，当两个地址相同时，I2C从机将发送一个确认应答（ACK），并响应总线的后续命令：发送或接收所需数据。此外，如果软件开启了广播呼叫，则I2C从机始终对一个广播地址（0x00）发送确认应答。I2C模块始终支持7位和10位的地址。

I2C主机负责产生START信号和STOP信号来开始和结束一次传输，并且负责产生SCL时钟。

图 18-6. 7 位地址的 I2C 通讯流程

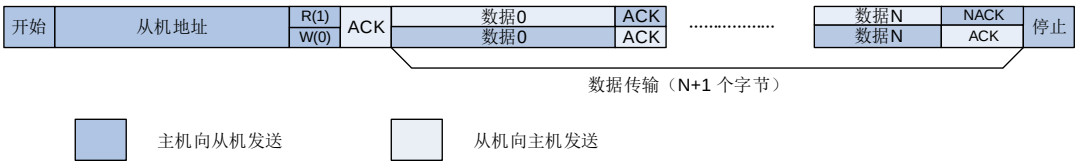


图 18-7. 10 位地址的 I2C 通讯流程（主机发送）

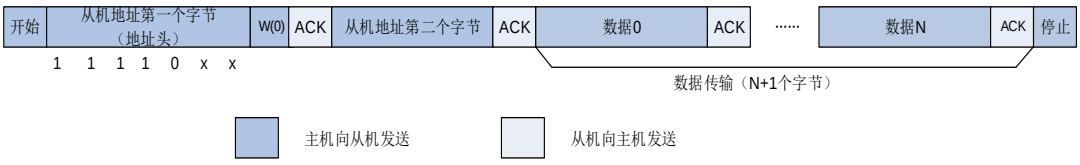
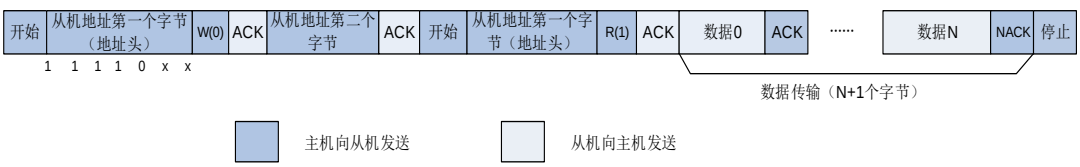


图 18-8. 10 位地址的 I2C 通讯流程（主机接收）



18.3.7. 软件编程模型

一个I2C设备例如LCD驱动器可能只是作为一个接收器，但是一个存储器既可以接收数据，也能发送数据。除了按照发送/接收方来区分，I2C设备也分为数据传输的主机和从机。主机是指负责初始化总线上数据的传输并产生时钟信号的设备，此时任何被寻址的设备都是从机。

不管I2C设备是主机还是从机，都可以发送或接收数据，因此，I2C设备有以下4种运行模式：

- 主机发送方
- 主机接收方
- 从机发送方
- 从机接收方

I2C模块支持以上四种模式。系统复位以后，I2C默认工作在从机模式下。通过软件配置使I2C在总线上发送一个START信号之后，I2C变为主机模式，软件配置在I2C总线上发送STOP信号后，I2C又变回从机模式。

从机发送模式下的软件流程

如*图18-9. 从机发送模式（10位地址模式）*所示，在从机模式下要发送数据，软件应该按照以下步骤来运行操作：

1. 首先，软件应该使能I2C外设时钟，以及配置I2C_CTL1中时钟相关寄存器来确保正确的I2C时序。使能和配置以后，I2C运行在默认的从机模式状态，等待I2C总线上的START信号和地址。
2. 当接收到一个START信号及随后的地址后，地址可以是7位格式也可以是10位格式，I2C硬件将I2C_STAT0寄存器的ADDSEND位置1，此位应该被软件查询或者中断监视，发现置位后，软件应该读I2C_STAT0寄存器然后读I2C_STAT1寄存器来清除ADDSEND位。如果地址是10位格式，I2C主机应该接着再产生一个START并发送一个地址头到I2C总线。从机在检测到START和紧接着的地址头之后会继续将ADDSEND位置1。软件可以通过读I2C_STAT0寄存器和接着读I2C_STAT1寄存器来第二次清除ADDSEND位。
3. 现在I2C进入数据发送状态，由于移位寄存器和数据寄存器I2C_DATA都是空的，硬件将TBE位置1。软件此时可以写入第一个字节数据到I2C_DATA寄存器，但是TBE位并没有被清0，因为写入I2C_DATA寄存器的字节被立即移入内部移位寄存器。当移位寄存器非空的时候，I2C开始发送数据到I2C总线。
4. 第一个字节的发送期间，软件可以写第二个字节到I2C_DATA，此时TBE位被清0，因为I2C_DATA寄存器和移位寄存器都不是空。
5. 第一个字节的发送完成之后，TBE被再次置起，软件可以写第三个字节到I2C_DATA，同时TBE位被清0。在此之后，任何时候TBE被置1，只要依然有数据待被发送，软件都可以写入一个字节到I2C_DATA寄存器。
6. 倒数第二个字节发送期间，软件写最后一个数据到I2C_DATA寄存器来清除TBE标志位，之后就再不用关心TBE的状态。TBE位会在倒数第二个字节发送完成后置起，直到检测到STOP信号时被清0。
7. 根据I2C协议，I2C主机将不会对接收到的最后一个字节发送应答，所以在最后一个字节发

送结束后，I2C从机的AERR（应答错误）会置起以通知软件发送结束。软件写0到AERR位可以清除此位。

图 18-9. 从机发送模式（10 位地址模式）

从机接收模式下的软件流程

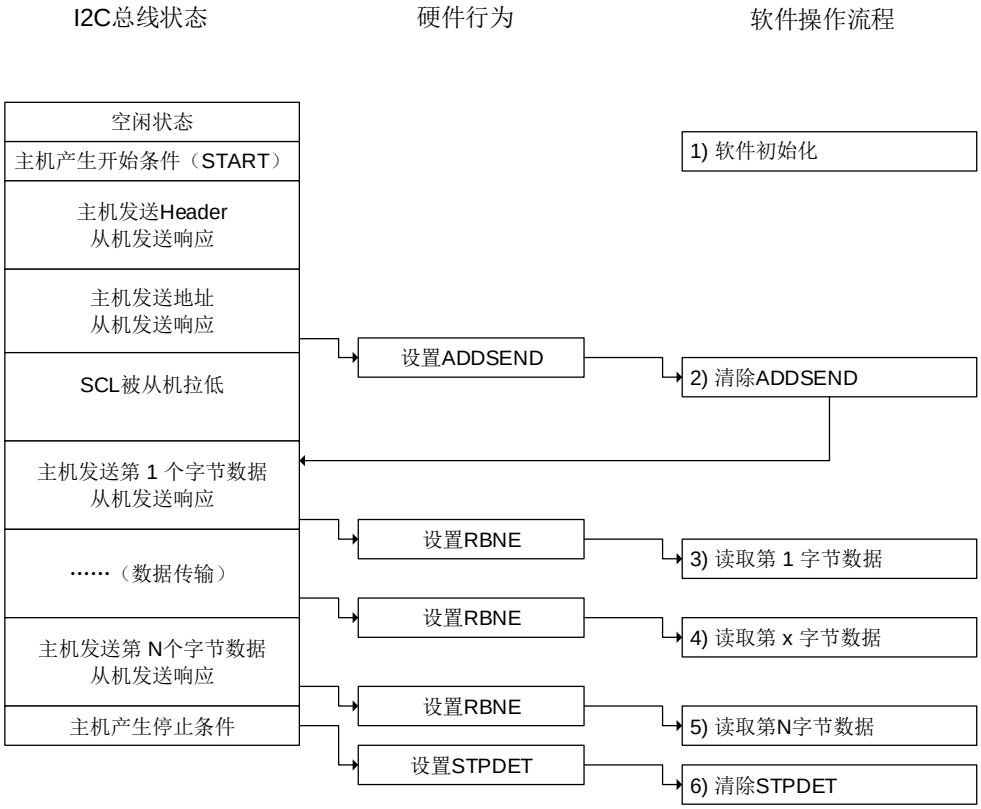
如*图 18-10. 从机接收模式（10 位地址模式）*所示，在从机模式下接收数据时，软件应该遵循这些步骤来操作：

1. 首先，软件应该使能I2C外设时钟，以及配置I2C_CTL1中时钟相关寄存器来确保正确的I2C时序。使能和配置以后，I2C运行在默认的从机模式状态，等待START信号以及地址。
2. 在接收到START起始信号和匹配的7位或10地址之后，I2C硬件将I2C状态寄存器0的ADDSEND位置1，此位应该通过软件轮询或者中断来检测，发现置起后，软件通过先读I2C_STAT0寄存器然后读I2C_STAT1寄存器来清除ADDSEND位。当ADDSEND位被清0时，I2C就开始接收来自I2C总线的数据。
3. 当接收到第一个字节时，RBNE位被硬件置1，软件可以读取I2C_DATA寄存器的第一个字节，此时RBNE位也被清0。
4. 任何时候RBNE被置1，软件可以从I2C_DATA寄存器读取一个字节。

5. 接收到最后一个字节后，RBNE被置1，软件可以读取最后的字节。

6. 当I2C检测到I2C总线上一个STOP信号，STPDET位被置1，软件通过先读I2C_STAT0寄存器再写I2C_CTL0寄存器来清除STPDET位。

图 18-10. 从机接收模式（10 位地址模式）



主机发送模式下的软件流程

如*图18-11. 主机发送模式（10位地址模式）*所示，在主机模式下发送数据到I2C总线时，软件应该遵循这些步骤来运行I2C模块：

1. 首先，软件应该使能I2C外设时钟，以及配置I2C_CTL1中时钟相关寄存器来确保正确的I2C时序。使能和配置以后，I2C运行在默认的从机模式状态，等待START信号，随后等待I2C总线寻址。

2. 软件将START位置1，在I2C总线上产生一个START信号。

3. 发送一个START信号后，I2C硬件将I2C_STAT0的SBSEND位置1然后进入主机模式。现在软件应该读I2C_STAT0寄存器然后写一个7位地址位或10位地址的地址头到I2C_DATA寄存器来清除SBSEND位。当SBSEND位被清0时，I2C就开始发送地址或者地址头到I2C总线。如果发送的地址是10位地址的地址头，硬件在发送地址头的时候会将ADD10SEND位置1，软件应该通过读I2C_STAT0寄存器然后写10位低地址到I2C_DATA来清除ADD10SEND位。

4. 7位或10位的地址位发送出去之后，I2C硬件将ADDSEND位置1，软件通过读I2C_STAT0寄存器然后读I2C_STAT1寄存器清除ADDSEND位。

5. I2C进入数据发送状态，因为移位寄存器和数据寄存器I2C_DATA都是空的，所以硬件将TBE位置1。此时软件可以写第一个字节数据到I2C_DATA寄存器，但是TBE位此时不会被清零，因为写入I2C_DATA寄存器的字节会被立即移入内部移位寄存器。当移位寄存器非空时，I2C就开始发送数据到总线。

6. 在第一个字节的发送过程中，软件可以写第二个字节到I2C_DATA，此时TBE会被清零，因为I2C_DATA寄存器和移位寄存器都不为空。

7. 任意时刻TBE被置1，软件都可以向I2C_DATA寄存器写入一个字节，只要还有数据待发送。

8. 在倒数第二个字节发送过程中，软件写入最后一个字节数据到I2C_DATA来清除TBE标志位，此后就不用关心TBE位的状态。TBE位会在倒数第二个字节发送完成后被置起，直到发送STOP信号时被清零。

9. 最后一个字节发送结束后，I2C主机将BTC位置起，因为移位寄存器和I2C_DATA寄存器此时都为空。软件此时应该配置STOP来发送一个STOP信号，此后TBE和BTC状态位都将被清0。

图 18-11. 主机发送模式（10 位地址模式）

主机接收模式下的软件流程

在主机接收模式下，主机需要为最后一个字节接收产生NACK，然后发送STOP信号。因此，需要特别注意以确保最后接收到数据的正确性。下面提供了两种针对主机接收模式的软件编程方案：方案A和B。方案A需要保证软件能对I2C事件进行快速响应，方案B则不需要。

方案 A

1. 首先，软件应该使能I2C外设时钟，以及配置I2C_CTL1中时钟相关寄存器来确保正确的I2C时序。使能和配置以后，I2C运行在默认的从机模式状态，等待START信号，随后等待I2C总线寻址。
2. 软件将START位置1，从而在I2C总线上产生一个START信号。
3. 发送一个START信号后，I2C硬件将I2C_STAT0寄存器的SBSEND位置1然后进入主机模式。现在软件应该读I2C_STAT0寄存器然后写一个7位地址位或10位地址的地址头到I2C_DATA寄存器来清除SBSEND位。当SBSEND位被清0时，I2C就开始发送地址或者地址头到I2C总线。如果发送的地址是10位地址的地址头，硬件在发送地址头的时候会先将ADD10SEND位置1，软件应该通过读I2C_STAT0寄存器然后写10位低地址到I2C_DATA来清除ADD10SEND位。
4. 7位或10位的地址位发送出去之后，I2C硬件将ADDSEND位置1，软件应该通过读I2C_STAT0寄存器然后读I2C_STAT1寄存器清除ADDSEND位。如果地址是10位格式，软件应该再次将START位置1来重新产生一个START。在START产生后，SBSEND位会被置1。软件应该通过先读I2C_STAT0然后写地址头到I2C_DATA来清除SBSEND位，然后地址头被发到I2C总线，ADDSEND再次被置1。软件应该再次通过先读I2C_STAT0然后读I2C_STAT1来清除ADDSEND位。
5. 当接收到第一个字节时，硬件会将RBNE位置1。此时软件可以从I2C_DATA寄存器读取第一个字节，之后RBNE位被清0。
6. 此后任何时候RBNE被置1，软件就可以从I2C_DATA寄存器读取一个字节。
7. 接收完倒数第二个字节（N-1）数据之后，软件应该立即将ACKEN位清0，并将STOP位置1，这一过程需要在最后一个字节接收完毕之前完成，以确保NACK发送给最后一个字节。
8. 最后一个字节接收完毕后，RBNE位被置1，软件可以读取最后一个字节。由于ACKEN已经在前一步骤中被清0，I2C不再为最后一个字节发送ACK，并在最后一个字节发送完毕后产生一个STOP信号。

以上步骤要求字节数目N>1，如果N=1，步骤7应该在步骤4之后就执行，且需要在字节接收完成之前完成。

图 18-12. 主机接收使用方案 A 模式（10 位地址模式）

方案 B

1. 首先，软件应该使能I2C外设时钟，配置I2C_CTL1中时钟相关寄存器来确保正确的I2C时序。初始化完成之后，I2C运行在默认的从机模式状态，等待START信号和地址。

2. 软件将START位置1，从而在I2C总线上产生一个START信号。

3. 发送一个START信号后，I2C硬件将I2C_STAT0寄存器的SBSEND位置1然后进入主机模式。现在软件应该读I2C_STAT0寄存器然后写一个7位地址位或10位地址的地址头到I2C_DATA寄存器来清除SBSEND位。当SBSEND位被清0时，I2C就开始发送地址或者地址头到I2C总线。如果发送的地址是10位地址的地址头，硬件在发送地址头之后会将ADD10SEND位置1，软件应该通过读I2C_STAT0寄存器然后写10位低地址到I2C_DATA来清除ADD10SEND位。

4. 7位或10位的地址位发送出去之后，I2C硬件将ADDSEND位置1，软件应该通过读

I2C_STAT0寄存器然后读I2C_STAT1寄存器清除ADDSEND位。如果地址是10位格式，软件应该接着将START位再次置1来产生一个开始信号，START被发送出去以后SBSEND位被再次置1。软件应该通过先读I2C_STAT0然后写地址头到I2C_DATA来清除SBSEND位，然后地址头被发到I2C总线，ADDSEND再次被置1。软件应该再次通过先读I2C_STAT0然后读I2C_STAT1来清除ADDSEND位。

5. 当第一个字节被接收时，RBNE位会被硬件置1。此时软件可从I2C_DATA寄存器读取出第一个字节，同时RBNE位被清0。

6. 此后任何时刻，只要RBNE位被置1，软件就可以从I2C_DATA寄存器读取一个字节的数据，直到主机接收了N-3个字节。

如*图 18-13. 主机接收使用方案 B 模式（10 位地址模式）*所示，第 N-2 个字节还没被软件读出，之后第 N-1 个字节被接收，此时 BTC 和 RBNE 都被置位，总线就会被主机锁死以阻止最后一个字节的接收。然后软件应该清除 ACKEN 位。

7. 软件从I2C_DATA读出倒数第三个（N-2）字节数据，同时也将BTC位清0。此后第N-1个字节从移位寄存器被移到I2C_DATA，总线得到释放然后开始接收最后一个字节，由于ACKEN已经被清除，因此主机不会给最后一个字节数据发送ACK响应。

8. 最后一个字节接收完毕后，硬件再次把BTC位和RBNE置1，并拉低SCL，软件将STOP位置1，主机发出一个STOP信号。

9. 软件读取第N-1个字节，清除BTC。此后最后一个字节从移位寄存器被移动到I2C_DATA。

10. 软件读取最后一个字节，清除RBNE。

以上步骤需要字节数字N>2，N=1和N=2的情况近似：

N=1

在第 4 步，软件应该在清除 ADDSEND 位之前将 ACKEN 位清 0，在清除 ADDSEND 位之后将 STOP 位置 1。当 N=1 时步骤 5 是最后一步。

N=2

在第2步，软件应该在START置1之前将POAP置1。在第4步，软件应该在清除ADDSEND位之前将ACKEN位清0。在第5步，软件应该一直等到BTC位被置1然后将STOP位置1且读取I2C_DATA两次。

图 18-13. 主机接收使用方案 B 模式（10 位地址模式）

18.3.8. SCL 线控制

SCL 线拉低功能是为了避免在接收时发生上溢错误以及在发送时发生下溢错误。如在软件编程模型中所示，在发送模式，当 TBE 和 BTC 被置位，发送器保持 SCL 线为低电平直到下一个发送数据写入传输缓冲区寄存器。在接收模式，当 RBNE 和 BTC 被置位，发送器保持 SCL 线为低电平直到传输缓冲区寄存器里的数据被读出。

当工作在从模式的时候，可以通过置位 I2C_CTL0 寄存器的 SS 位禁止 SCL 线拉低功能。如

果该位置位，软件要能足够快的处理 TBE，RBNE 和 BTC 状态，否则上溢或下溢的情况可能会发生。

18.3.9. DMA 模式下数据传输

按照前面的软件流程，每当 TBE 位或 RBNE 位被置 1 之后，软件都应该写或读一个字节，这样将导致 CPU 的负荷较重。I2C 的 DMA 功能可以在 TBE 或 RBNE 位置 1 时，自动进行一次写或读操作，从而减轻了 CPU 的负荷，具体 DMA 的配置请参看 DMA 相关章节。

DMA 请求通过 I2C_CTL1 寄存器的 DMAON 位使能。该位应该在清除 ADDSEND 状态位之后被置位。如果一个从机的 SCL 线延长功能被禁止，DMAON 位应该在 ADDSEND 事件前被置位。

参考 DMA 控制器的关于 DMA 的配置方法说明。DMA 必须在 I2C 传输开始之前配置和使能。当指定个数的字节已经传输完成，DMA 会发送一个传输结束（EOT）信号给 I2C 接口，并产生一个 DMA 传输完成中断。

当主机接收两个或两个以上字节时，需将 I2C_CTL1 寄存器的 DMALST 位置位。在接收到最后一个字节之后，I2C 主机发送 NACK。在 DMA 传输完成中断 ISR 中，通过置位 STOP 位，产生一个停止信号。

当主机仅接收一个字节时，清除ADDSEND状态前ACKEN位必须被清除。在清除ADDSEND状态后或在DMA传输完成中断ISR中，通过置位STOP位，产生一个停止信号。

18.3.10. 报文错误校验

I2C模块中有一个PEC（包错误检查）模块，它使用CRC-8计算器来执行I2C数据的报文校验，CRC多项式为x8 + x2 + x + 1，和SMBus协议兼容。将PECEN位置1就可以使能PEC功能。PEC将会计算所有通过I2C总线发送的数据（包括地址）。软件可以通过配置PECTRANS来控制I2C在最后一个字节发送完毕后发送PEC值，或者在接收完成后检查接收到的PEC值是否正确。在DMA模式下，如果PECEN位和PECTRANS位被置1，I2C将自动发送或者检查PEC值。

18.3.11. SMBus 支持

系统管理总线（System Management Bus，简写为SMBus或SMB）是一种结构简单的单端双线制总线，可实现轻量级的通信需求。一般来说，SMBus最常见于计算机主板，主要用于电源传输ON / OFF指令的通信。SMBus是I2C的一种衍生总线形式，主要用于计算机主板上的低带宽设备间通信，尤其是与电源相关的芯片，例如笔记本电脑的可充电电池子系统（参见Smart Battery Data）。

SMBus 协议

SMBus上每个报文交互都遵从SMBus协议中预定义的格式。SMBus是I2C规范中数据传输格式的子集。只要I2C设备可通过SMBus协议之一进行访问，便视为兼容SMBus规范。不符合这些协议的I2C设备，将无法被SMBus和ACPI规范所定义的标准方法访问。

地址解析协议

SMBus是基于I2C硬件实现的，它使用了I2C的硬件寻址方式，但在I2C的基础上增加了二级软件处理，建立自己独特的系统。比较特别的是SMBus规范包含一个地址解析协议，可用于实现动态地址分配。动态识别硬件和软件使得总线设备能够支持热插拔，无需重启系统便能即插即用。总线中的设备将被自动识别并分配唯一地址。这个优点非常有利于实现即插即用的用户接口。在此协议中，系统中的host与设备之间有一个重要的区别，即host具有分配地址的功能。

超时特性

SMBus有一种超时特性：假如某个通信耗时太久，便会自动复位设备。这就解释了为什么最小时钟周期为10kHz——为了防止长时间锁死总线。I2C在本质上可以视为一个“直流”总线，也就是说当主机正在访问从机的时候，假如从机正在执行一些子程序无法及时响应，从机可以拉住主机的时钟。这样便可以提醒主机：从机正忙，但并不想放弃当前的通信。从机的当前任务结束之后，将可以继续I2C通信。I2C总线协议中并没有限制这个延时的上限，但在SMBus系统中，这个时间被限定为25ms。按照SMBus协议的假定，如果某个通信耗时太久，就意味着总线出了问题，此时所有设备都应当复位以消除这种问题。这样就并不允许从设备将时钟拉低太长时间。

报文错误校验

SMBus 2.0以及1.1都采用了报文错误校验（Packet Error Checking，缩写为PEC）。在这种模式中，每次通信最后都将传输PEC字节。该字节是按照CRC-8校验和的方式计算的，计算范围包括整个报文，包括地址以及读/写位。所采用的多项式为x8+x2+x+1（CRC-8-ATM HEC算法，初始化为0）。

SMBus 警报

SMBus还有一个额外的中断信号，称为SMBALERT#。从机上发生事件后，可通过这个信号通知主机来访问从机。SMBus中还定义了较少见的“主机提醒协议”，基于I2C多主模式实现类似的提醒功能，但是可以传递更多数据。

SMBus 通讯流程

SMBus的通讯流程和标准I2C的流程相似。为了使用SMBus模式，在程序中需要配置几个SMBus特定的寄存器，响应一些SMBus特定标志位，实现那些在SMBus手册中介绍的上层协议。

1. 在通信之前，需要将I2C_CTL0中SMBEN位置1，并且根据需求，配置SMBSEL和ARPEN的值。
2. 为了支持ARP协议（ARPEN=1），在SMBus主机模式下（SMBSEL=1），软件需要响应标志位HSTSMB（在SMBus从机模式下，响应DEFSMB标志位），并实现ARP协议中的功能。
3. 为了支持SMBus警告模式，软件应该响应SMBALT标志位，并实现相应的功能。

18.3.12. 状态、错误和中断

I2C有一些状态、错误标志位，通过设置一些寄存器位，便可以从这些标志触发中断（详情参见*I2C寄存器*）。

表 18-2. 事件状态标志位

事件标志位名称	说明
SBSEND	主机发送 START 信号
ADDSEND	地址发送和接收
ADD10SEND	10 位地址模式中地址头发送
STPDET	监测到 STOP 信号
BTC	字节发送结束
TBE	发送时 I2C_DATA 为空
RBNE	接收时 I2C_DATA 非空

表 18-3. 错误标志位

错误名称	说明
BERR	总线错误
LOSTARB	仲裁丢失
OUERR	当禁用 SCL 拉低后，发生了上溢或下溢
AERR	没有接收到应答
PECERR	CRC 值不相同
SMBTO	SMBus 模式下总线超时
SMBALT	SMBus 警报

18.4. I2C 寄存器

I2C0基地址：0x4000 5400
I2C1基地址：0x4000 5800

18.4.1. 控制寄存器 0（I2C_CTL0）

地址偏移：0x00
复位值：0x0000 0000

该寄存器可以按半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRESET	保留	SALT	PECTRANS	POAP	ACKEN	STOP	START	SS	GCEN	PECEN	ARPEN	SMBSEL	保留	SMBEN	I2CEN
rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	SRESET	软件复位 I2C，软件应该在 I2C 总线被释放时复位 I2C。 0：I2C 未复位 1：I2C 复位
14	保留	必须保持复位值。
13	SALT	SMBus 警报 通过 SMBA 引脚发出警报。 软件置 1 和清 0，硬件清 0。 0：不通过 SMBA 发布警告 1：通过 SMBA 引脚发送警告
12	PECTRANS	PEC 传输 软件置 1 和清 0，硬件在以下条件下清除此位：PEC 传输完成，或监测到 START / STOP 信号，或 I2CEN=0。 0：不传输 PEC 值 1：传输的 PEC 值
11	POAP	ACK/PEC 的位置含义 软件置 1 和清 0，当 I2CEN=0 时，硬件清 0。 0：ACKEN 位决定对当前正在接收的字节是否发送 ACK，PECTRANS 位表明正在接收的字节是否为 PEC。 1：ACKEN 位决定是否对下一个字节发送 ACK，PECTRANS 位表明下一个即将被接收的字节是 PEC。

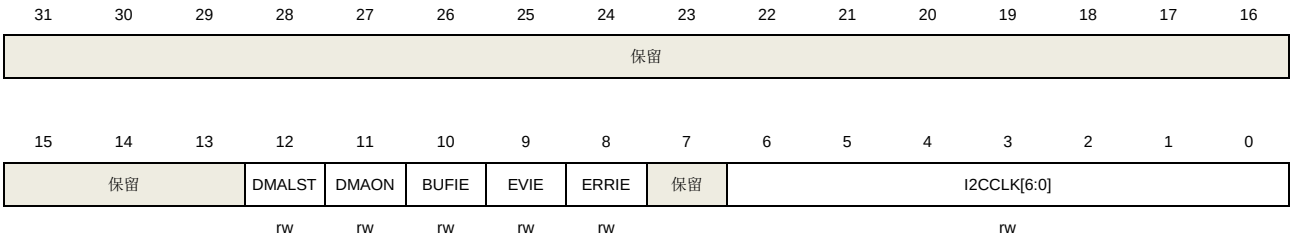
10	ACKEN	是否发送 ACK 软件置 1 和清 0，当 I2CEN=0 时硬件清 0。 0：不发送 ACK 1：发送 ACK
9	STOP	I2C 总线上产生一个 STOP 信号 软件置 1 和清 0，SMBus 超时时，硬件置 1，监测到 STOP 信号时，硬件清 0。 0：不发送 STOP 1：发送 STOP
8	START	I2C 总线上产生一个 START 信号 软件置 1 和清 0，当监测到 START 信号或 I2CEN=0 时由硬件清 0。 0：不发送 START 1：发送 START
7	SS	在从机模式下数据未就绪是否将 SCL 拉低 软件置 1 和清 0。 0：拉低 SCL 1：不拉低 SCL
6	GCEN	是否响应对地址（0x00）的广播呼叫 0：从机不响应广播呼叫 1：从机将响应广播呼叫
5	PECEN	PEC 计算使能 0：PEC 计算禁能 1：PEC 计算使能
4	ARPEN	SMBus 下 ARP 协议使能 0：ARP 禁能 1：ARP 使能
3	SMBSEL	SMBus 类型选择 0：从机 1：主机
2	保留	必须保持复位值
1	SMBEN	SMBus/I2C 模式开关 0：I2C 模式 1：SMBus 模式
0	I2CEN	I2C 外设使能 0：禁能 I2C 1：使能 I2C

18.4.2. 控制寄存器 1（I2C_CTL1）

地址偏移：0x04

复位值：0x0000 0000

该寄存器可以按半字（16位）或字（32位）访问。



位/位域	名称	描述
31:13	保留	必须保持复位值。
12	DMALST	DMA 最后传输配置 0：下一个 DMA EOT 不是最后传输 1：下一个 DMA EOT 是最后传输
11	DMAON	DMA 模式开关 0：DMA 模式关 1：DMA 模式开
10	BUFIE	缓冲区中断使能 0：禁用缓冲区中断，如果 EVIE=1，当 TBE = 1 或 RBNE = 1 时不产生中断。 1：使能缓冲区中断，如果 EVIE=1，当 TBE = 1 或 RBNE = 1 时产生中断。
9	EVIE	事件中断使能 0：禁用事件中断 1：使能事件中断，意味着当 SBSEND、ADDSEND、ADD10SEND、STPDET 或 BTC 标志位有效或当 BUFIE=1 时 TBE=1 或 RBNE=1 时产生中断。
8	ERRIE	错误中断使能 0：禁用错误中断 1：使能错误中断，意味着当 BERR、LOSTARB、AERR、OUERR、PECERR、SMBTO 或 SMBALT 标志位生效时产生中断。
7	保留	必须保持复位值。
6:0	I2CCLK[6:0]	I2C 外设时钟频率 I2CCLK[6:0]应该是输入 APB1 时钟频率，最低 2MHz。 0000000 - 0000001：无时钟 0000010 - 1001000：2MHz~72MHz 1001001 - 1111111：由于 APB1 时钟限制，无时钟 **注意：** 在标准模式下，APB1 时钟频率需大于或者等于 2MHz。在快速模式下，APB1 时钟频率需大于或者等于 8MHz。

18.4.3. 从机地址寄存器 0（I2C_SADDR0）

地址偏移：0x08
复位值：0x0000 0000

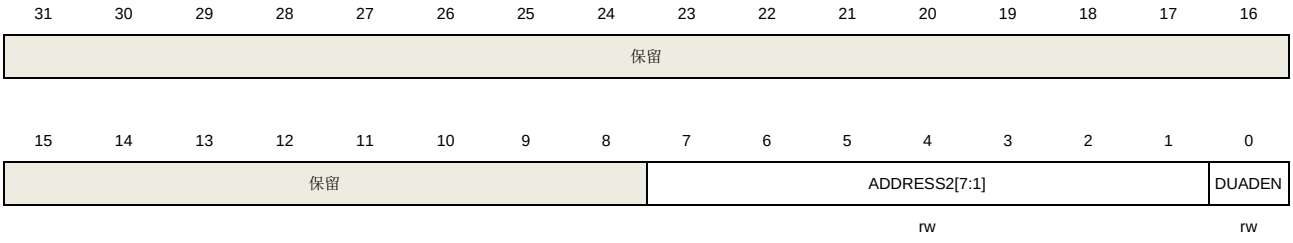
该寄存器可以按半字（16位）或字（32位）访问。

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	ADDFORMAT	I2C 从机地址格式 0：7 位地址 1：10 位地址
14:10	保留	必须保持复位值。
9:8	ADDRESS[9:8]	10 位地址的最高两位
7:1	ADDRESS[7:1]	7 位地址或者 10 位地址的第 7-1 位
0	ADDRESS0	10 位地址的第 0 位

18.4.4. 从机地址寄存器 1（I2C_SADDR1）

地址偏移：0x0C
复位值：0x0000 0000

该寄存器可以按半字（16位）或字（32位）访问。



位/位域	名称	描述
31:8	保留	必须保持复位值。
7:1	ADDRESS2[7:1]	从机在双重地址模式下第二个 I2C 地址
0	DUADEN	双重地址模式使能 0：禁用双重地址模式

1：使能双重地址模式

18.4.5. 传输缓冲区寄存器（I2C_DATA）

地址偏移：0x10
复位值：0x0000 0000

该寄存器可以按半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								TRB[7:0]							
								rw							

位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	TRB[7:0]	数据发送接收缓冲区

18.4.6. 传输状态寄存器 0（I2C_STAT0）

地址偏移：0x14
复位值：0x0000 0000

该寄存器可以按半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMBALT	SMBTO	保留	PECERR	OUERR	AERR	LOSTARB	BERR	TBE	RBNE	保留	STPDET	ADD10SEND	BTC	ADDSEND	SBSEND
rc_w0	rc_w0		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	r	r		r	r	r	r	r

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	SMBALT	SMBus 警报状态 硬件置 1，软件写 0 清 0。 0：SMBA 引脚未被拉低（从机模式）或未监测到警报（主机模式） 1：SMBA 引脚被拉低且接收到警报地址（从机模式）或监测到警报（主机模式）
14	SMBTO	SMBus 模式下超时信号 硬件置 1，软件写 0 清 0。 0：无超时错误 1：超时事件发生（SCL 被拉低达 25ms）

13	保留	必须保持复位值。
12	PECERR	接收数据时 PEC 错误 硬件置 1，软件写 0 清 0。 0：接收到 PEC 且校验正确 1：接收到 PEC 但检验错误，此时不管 ACKEN 位的值，I2C 将发送 NACK
11	OUERR	当禁用 SCL 拉低功能后，在从机模式下发生了上溢或下溢事件。在从机接收模式下，假如 I2C_DATA 中的最后一字节并未被读出，并且后续字节又接收完成，就会发生上溢错误。在从机发送模式下，假如当前字节已经发送完成，而 I2C_DATA 仍然为空，就会发生下溢错误。 硬件置 1，软件写 0 清 0。 0：无上溢或下溢错误发生 1：发生上溢或下溢错误
10	AERR	应答错误 硬件置 1，软件写 0 清 0。 0：未发生应答错误 1：发生了应答错误
9	LOSTARB	主机模式下仲裁丢失 硬件置 1，软件写 0 清 0。 0：无仲裁丢失 1：发生仲裁丢失，I2C 模块返回从机模式。
8	BERR	总线错误，表示 I2C 总线上发生了预料之外的 START 信号或 STOP 信号。 硬件置 1，软件写 0 清 0。 0：无总线错误 1：发生了总线错误
7	TBE	发送期间 I2C_DATA 为空 硬件从 I2C_DATA 寄存器移动一个字节到移位寄存器之后将此位置 1，软件写一个字节到 I2C_DATA 寄存器清除该位。如果移位寄存器和 I2C_DATA 寄存器都是空的，写 I2C_DATA 寄存器将不会清除 TBE 位（详见主机/从机发送模式下的软件操作流程） 0：I2C_DATA 非空 1：I2C_DATA 空，软件可以写
6	RBNE	接收期间 I2C_DATA 非空 硬件从移位寄存器移动一个字节到 I2C_DATA 寄存器之后将此位置 1，读 I2C_DATA 可以清除此位。如果 BTC 和 RBNE 都被置 1，读 I2C_DATA 将不会清除 RBNE，因为移位寄存器的字节将被立即移到 I2C_DATA。 0：I2C_DATA 为空 1：I2C_DATA 非空，软件可以读
5	保留	必须保持复位值。
4	STPDET	从机模式下监测到 STOP 信号 此位被硬件置 1，先读 I2C_STAT0 然后写 I2C_CTL0 可以清除此位。

0：从机模式下未监测到 STOP 信号
1：从机模式下监测到 STOP 信号

3	ADD10SEND	主机模式下 10 位地址的地址头被发送 该位由硬件置 1，软件读 I2C_STAT0 和写 I2C_DATA 清除此位。 0：主机模式下未发送 10 位地址的地址头 1：主机模式下发送 10 位地址的地址头
2	BTC	字节发送结束 接收模式下，如果一个字节已经被移位寄存器接收但是此时 I2C_DATA 寄存器仍然是满的；或者发送模式下，一个字节已经被移位寄存器发送但是 I2C_DATA 寄存器仍然是空的，如果使能了 SCL 拉低功能，位硬件就会置起 BTC 标志。 此位由硬件置 1。 可由以下三种方式清除： 1、软件清除：读 I2C_STAT0，然后读或写 I2C_DATA 寄存器清除此位 2、硬件清除：发送一个 STOP 或 START 信号 3、寄存器 I2C_CTL0 中 I2CEN=0 0：未发生 BTC 1：发生了 BTC
1	ADDSEND	主机模式下：成功发送了地址并收到 ACK 从机模式下：接收到的地址与自身的地址匹配 此位由硬件置 1，软件读 I2C_STAT0 寄存器和读 I2C_STAT1 清 0。 0：从机模式下，未收到地址或者收到的地址不匹配；主机模式下，无地址被发送或地址已发送但未收到从机回复的 ACK 1：从机模式下，接收到的地址与自身的地址匹配；主机模式下，地址已发送并收到 ACK
0	SBSEND	主机模式下发送 START 信号 此位由硬件置 1，软件读 I2C_STAT0 和写 I2C_DATA 清 0。 0：未发送 START 信号 1：START 信号被发送

18.4.7. 传输状态寄存器 1（I2C_STAT1）

地址偏移：0x18
复位值：0x0000 0000

该寄存器可以按半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PECV[7:0]								DUMODF	HSTSMB	DEFSMB	RXGC	保留	TR	I2CBSY	MASTER
r								r	r	r	r		r	r	r

位/位域	名称	描述

31:16	保留	必须保持复位值。
15:8	PECV[7:0]	当 PEC 使能后硬件计算出的 PEC 值。
7	DUMODF	从机模式下双标志位表明哪个地址和双地址模式匹配 STOP 或 START 信号产生后或 I2CEN=0 时此位由硬件清 0。 0：地址和 I2C_SADDR0 匹配 1：地址和 I2C_SADDR1 匹配
6	HSTSMB	从机模式下监测到 SMBus 主机地址头 STOP 或 START 信号产生后或 I2CEN=0 时此位由硬件清 0。 0：未监测到 SMBus 主机地址头 1：监测到 SMBus 主机地址头
5	DEFSMB	SMBus 设备缺省地址 STOP 或 START 信号产生后或 I2CEN=0 时此位由硬件清 0。 0：SMBus 设备没有接收到缺省地址 1：SMBus 设备接收到缺省地址
4	RXGC	是否接收到广播地址（0x00） STOP 或 START 信号产生后或 I2CEN=0 时此位由硬件清 0。 0：未接收到广播呼叫地址 1：接收到广播呼叫地址
3	保留	必须保持复位值。
2	TR	发送端或接收端 该位表明 I2C 作为发送端还是接收端。STOP 或 START 信号产生后或 I2CEN 或 LOSTARB=1 时此位由硬件清 0。 0：接收端 1：发送端
1	I2CBSY	忙标志 STOP 信号后硬件清 0。 0：无 I2C 通讯 1：I2C 正在通讯
0	MASTER	主机模式 表明 I2C 时钟在主机模式还是从机模式的标志位。 该位在 START 信号产生后由硬件置 1。 该位在 STOP 信号产生后或 I2CEN=0 或 LOSTARB=1 时此位由硬件清 0。 0：从机模式 1：主机模式

18.4.8. 时钟配置寄存器（I2C_CKCFG）

地址偏移：0x1C
复位值：0x0000 0000

该寄存器可以按半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FAST	DTCY	保留		CLKC[11:0]											
rw	rw			rw											

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	FAST	主机模式下 I2C 速度选择 0：标准速度 1：快速
14	DTCY	快速模式下占空比 0：$T_{low}/T_{high}=2$ 1：$T_{low}/T_{high}=16/9$
13:12	保留	必须保持复位值。
11:0	CLKC[11:0]	主机模式下 I2C 时钟控制 标准速度模式下：$T_{high}=T_{low}=CLKC*T_{PCLK1}$ 如果 DTCY=0，快速模式下： $T_{high}=CLKC*T_{PCLK1}$，$T_{low}=2*CLKC*T_{PCLK1}$ 如果 DTCY=1，快速模式下： $T_{high}=9*CLKC*T_{PCLK1}$，$T_{low}=16*CLKC*T_{PCLK1}$ **注意：**如果 DTCY=0，当 PCLK1 为 3 的整数倍时，波特率会比较准确。如果 DTCY=1，当 PCLK1 为 25 的整数倍时，波特率会比较准确。

18.4.9. 上升时间寄存器（I2C_RT）

地址偏移：0x20

复位值：0x0000 0002

该寄存器可以按半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									RISETIME[6:0]						
									rw						

位/位域	名称	描述
31:7	保留	必须保持复位值。
6:0	RISETIME[6:0]	主机模式下最大上升时间

RISETIME 值应该为 SCL 最大上升时间加 1

19. 串行外设接口/片上音频接口（SPI/I2S）

19.1. 简介

SPI/I2S模块可以通过SPI协议或I2S音频协议与外部设备进行通信。

串行外设接口（Serial Peripheral Interface，缩写为SPI）提供了基于SPI协议的数据发送和接收功能，可以工作于主机或从机模式。SPI接口支持具有硬件CRC计算和校验的全双工和单工模式。

片上音频接口（Inter-IC Sound，缩写为I2S）支持四种音频标准，分别是I2S飞利浦标准，MSB对齐标准，LSB对齐标准和PCM标准。它可以在四种模式下运行，包括主机发送模式，主机接收模式，从机发送模式和从机接收模式。

19.2. 主要特性

19.2.1. SPI 主要特性

- 具有全双工、半双工和单工模式的主从操作。
- 16位宽度，独立的发送和接收缓冲区。
- 8位或16位数据帧格式。
- 低位在前或高位在前的数据位顺序。
- 软件和硬件NSS管理。
- 硬件CRC计算、发送和校验。
- 发送和接收支持DMA模式。

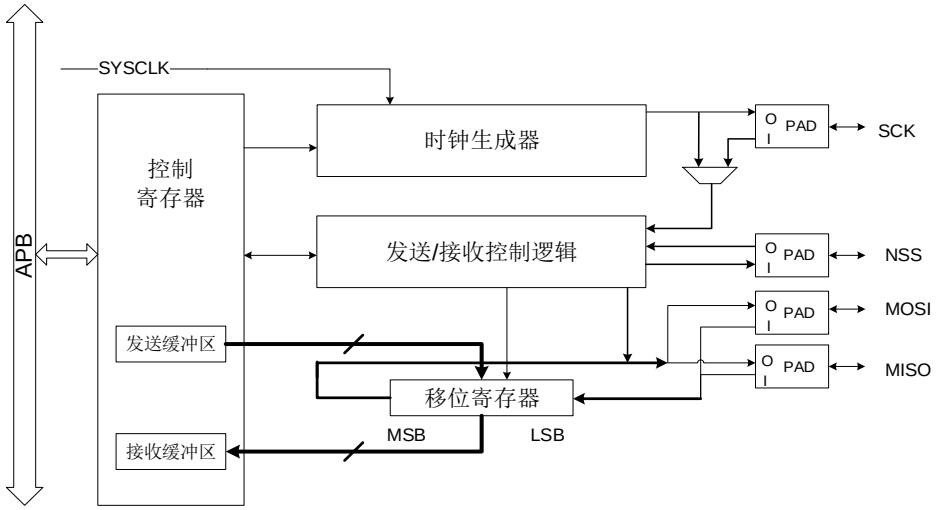
19.2.2. I2S 主要特性

- 具有发送和接收功能的主从操作。
- 支持四种I2S音频标准：飞利浦标准，MSB对齐标准，LSB对齐标准和PCM标准。
- 数据长度可以为16位，24位和32位。
- 通道长度为16位或32位。
- 16位缓冲区用于发送和接收。
- 通过I2S时钟分频器，可以得到8 kHz到192 kHz的音频采样频率。
- 可编程空闲状态时钟极性。
- 可以输出主时钟（MCK）。
- 发送和接收支持DMA功能。

19.3. SPI 功能描述

19.3.1. SPI 结构框图

图 19-1. SPI 结构框图



19.3.2. SPI 信号线描述

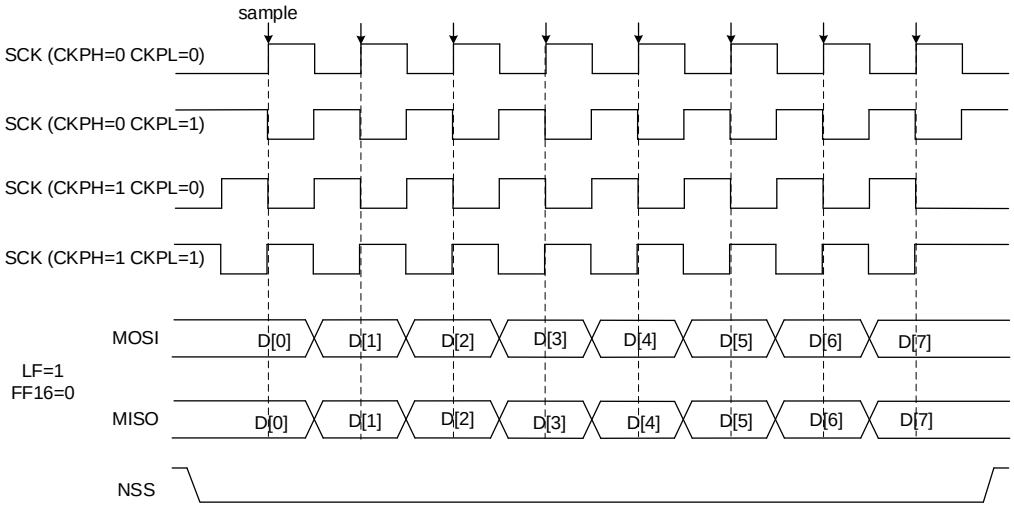
表 19-1. SPI 信号描述

引脚名称	方向	描述
SCK	I/O	主机：SPI 时钟输出 从机：SPI 时钟输入
MISO	I/O	主机：数据接收线 从机：数据发送线 主机双向线模式：不使用 从机双向线模式：数据发送和接收线
MOSI	I/O	主机：数据发送线 从机：数据接收线 主机双向线模式：数据发送和接收线 从机双向线模式：不使用
NSS	I/O	软件 NSS 模式：不使用 主机硬件 NSS 模式：NSSDRV=1 时，为 NSS 输出，适用于单主机模式；NSSDRV=0 时，为 NSS 输入，适用于多主机模式。 从机硬件 NSS 模式：为 NSS 输入，作为从机的片选信号。

19.3.3. SPI 时序和数据帧格式

SPI_CTL0寄存器中的CKPL位和CKPH位决定了SPI时钟和数据信号的时序。CKPL位决定了空闲状态时SCK的电平，CKPH位决定了第一个或第二个时钟跳变沿为有效采样边沿。

图 19-2. 常规模式下的 SPI 时序图



在常规模式中，通过SPI_CTL0中的FF16位配置数据长度，当FF16=1时，数据长度为16位，否则为8位。

通过设置SPI_CTL0中的LF位可以配置数据顺序，当LF=1时，SPI先发送LSB位，当LF=0时，则先发送MSB位。

19.3.4. NSS 功能

从机模式

当配置为从机模式（MSTMOD=0）时，在硬件NSS模式（SWNSSEN = 0）下，SPI从NSS引脚获取NSS电平，在软件NSS（SWNSSEN = 1）下，SPI根据SWNSS位得到NSS电平。只有当NSS为低电平时，发送或接收数据。在软件NSS模式下，不使用NSS引脚。

表 19-2. 从机模式 NSS 功能

模式	寄存器配置	描述
从机硬件 NSS 模式	MSTMOD = 0 SWNSSEN = 0	SPI 从机 NSS 电平从 NSS 引脚获取。
从机软件 NSS 模式	MSTMOD = 0 SWNSSEN = 1	SPI 从机 NSS 电平由 SWNSS 位决定。 SWNSS = 0：NSS 电平为低 SWNSS = 1：NSS 电平为高

主机模式

在主机模式（MSTMOD=1）下，如果应用程序使用多主机连接方式，NSS可以配置为硬件输入模式（SWNSSEN=0，NSSDRV=0）或者软件模式（SWNSSEN=1）。一旦NSS引脚（在硬件NSS模式下）或SWNSS位（在软件NSS模式下）被拉低，SPI将自动进入从机模式，并且产生主机配置错误，CONFERR位置1。

如果应用程序希望使用NSS引脚控制SPI从设备，NSS应该配置为硬件输出模式（SWNSSEN=0，NSSDRV=1）。使能SPI之后，NSS变为低电平。

应用程序可以使用一个通用I/O口作为NSS引脚，以实现更加灵活的NSS应用。

表 19-3. 主机模式 NSS 功能

模式	寄存器配置	描述
主机硬件 NSS 输出模式	MSTMOD = 1 SWNSSEN = 0 NSSDRV=1	适用于单主机模式，主机使用 NSS 引脚控制 SPI 从设备，此时 NSS 配置为硬件输出模式。使能 SPI 后 NSS 为低电平。
主机硬件 NSS 输入模式	MSTMOD = 1 SWNSSEN = 0 NSSDRV=0	适用于多主机模式，此时 NSS 配置为硬件输入模式，一旦 NSS 引脚被拉低，SPI 将自动进入从机模式，并且产生主机配置错误，CONFERR 位置 1。
主机软件 NSS 模式	MSTMOD = 1 SWNSSEN = 1 SWNSS = 0 NSSDRV：不要求	适用于多主机模式，一旦 SWNSS = 0，SPI 将自动进入从机模式，并且产生主机配置错误，CONFERR 位置 1。
	MSTMOD = 1 SWNSSEN = 1 SWNSS = 1 NSSDRV：不要求	从机可以使用硬件或软件 NSS 模式

19.3.5. SPI 运行模式

表 19-4. SPI 运行模式

模式	描述	寄存器配置	数据引脚用法
MFD	全双工主机模式	MSTMOD = 1 RO = 0 BDEN = 0 BDOEN：不要求	MOSI：发送 MISO：接收
MTU	单向线连接主机发送模式	MSTMOD = 1 RO = 0 BDEN = 0 BDOEN：不要求	MOSI：发送 MISO：不使用
MRU	单向线连接主机接收模式	MSTMOD = 1 RO = 1 BDEN = 0 BDOEN：不要求	MOSI：不使用 MISO：接收
MTB	双向线连接主机发送模式	MSTMOD = 1 RO = 0 BDEN = 1 BDOEN = 1	MOSI：发送 MISO：不使用

模式	描述	寄存器配置	数据引脚用法
MRB	双向线连接主机接收模式	MSTMOD = 1 RO = 0 BDEN = 1 BDOEN = 0	MOSI：接收 MISO：不使用
SFD	全双工从机模式	MSTMOD = 0 RO = 0 BDEN = 0 BDOEN：不要求	MOSI：接收 MISO：发送
STU	单向线连接从机发送模式	MSTMOD = 0 RO = 0 BDEN = 0 BDOEN：不要求	MOSI：不使用 MISO：发送
SRU	单向线连接从机接收模式	MSTMOD = 0 RO = 1 BDEN = 0 BDOEN：不要求	MOSI：接收 MISO：不使用
STB	双向线连接从机发送模式	MSTMOD = 0 RO = 0 BDEN = 1 BDOEN = 1	MOSI：不使用 MISO：发送
SRB	双向线连接从机接收模式	MSTMOD = 0 RO = 0 BDEN = 1 BDOEN = 0	MOSI：不使用 MISO：接收

图 19-3. 典型的全双工模式连接

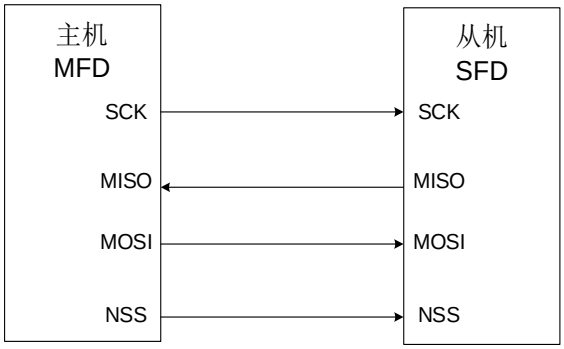


图 19-4. 典型的单工模式连接（主机：接收，从机：发送）

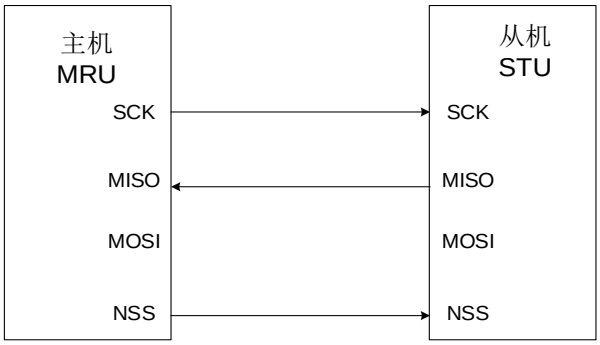


图 19-5. 典型的单工模式连接（主机：只发送，从机：接收）

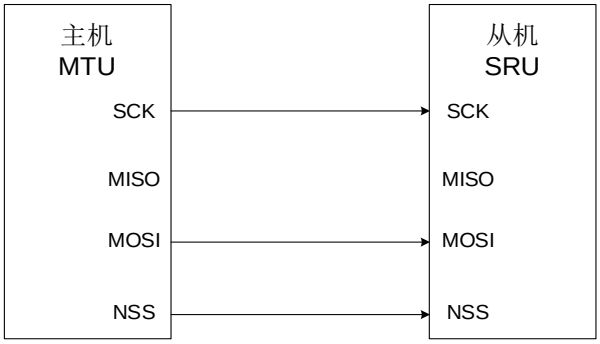
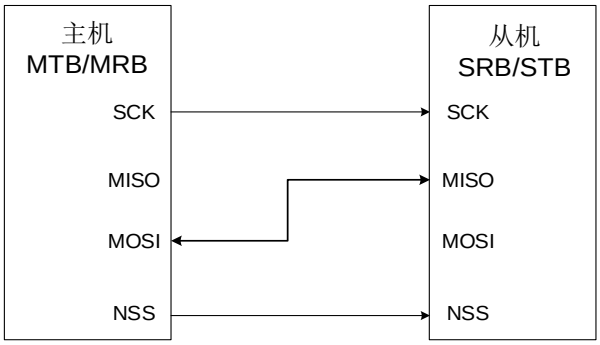


图 19-6. 典型的双向线连接



SPI 初始化流程

在发送或接收数据之前，应用程序应遵循如下的SPI初始化流程：

1. 如果工作在主机模式，配置SPI_CTL0中的PSC[2:0]位来生成预期波特率的SCK信号。否则，忽略此步骤。
2. 配置数据格式（SPI_CTL0中的FF16位）。
3. 配置时钟时序（SPI_CTL0中的CKPL位和CKPH位）。
4. 配置帧格式（SPI_CTL0中的LF位）。
5. 按照上文*NSS 功能*的描述，根据应用程序的需求，配置NSS模式（SPI_CTL0中的SWNSSEN位和NSSDRV位）。
6. 根据*表19-4. SPI运行模式*描述的运行模式，配置MSTMOD位、RO位、BDEN位和BDOEN位。
7. 使能SPI（将SPIEN位置1）。

注意：在通信过程中，不应更改CKPH、CKPL、MSTMOD、PSC[2:0]、LF位。

SPI 基本发送和接收流程

发送流程

在完成初始化过程之后，SPI模块使能并保持在空闲状态。在主机模式下，当软件写一个数据到发送缓冲区时，发送过程开始。在从机模式下，当SCK引脚上的SCK信号开始翻转，且NSS引脚电平为低，发送过程开始。所以，在从机模式下，应用程序必须确保在数据发送开始前，数据已经写入发送缓冲区中。

当SPI开始发送一个数据帧时，首先将这个数据帧从数据缓冲区加载到移位寄存器中，然后开始发送加载的数据。在数据帧的第一位发送之后，TBE（发送缓冲区空）位置1。TBE标志位置1，说明发送缓冲区为空，此时如果需要发送更多数据，软件应该继续写SPI_DATA寄存器。

在主机模式下，若想要实现连续发送功能，那么在当前数据帧发送完成前，软件应该将下一个数据写入SPI_DATA寄存器中。

接收流程

在最后一个采样时钟边沿之后，接收到的数据将从移位寄存器存入到接收缓冲区，且RBNE（接收缓冲区非空）位置1。软件通过读SPI_DATA寄存器获得接收的数据，此操作会自动清除RBNE标志位。在MRU和MRB模式中，为了接收下一个数据帧，硬件需要连续发送时钟信号，而在全双工主机模式（MFD）中，当发送缓冲区非空时，硬件才接收下一个数据帧。

SPI 不同模式下的操作流程

在全双工模式下，无论是MFD模式或者SFD模式，应用程序都应该监视RBNE标志位和TBE标志位，并且遵循上文描述的操作流程。

发送模式（MTU，MTB，STU或STB）与全双工模式中的发送流程类似，不同的是需要忽略RBNE位和RXORERR位。

相比于发送模式的情况，主机接收模式（MRU或MRB）与全双工的接收流程大不相同。在MRU模式或MRB模式下，在SPI使能后，SPI产生连续的SCK信号，直到SPI停止。所以，软件应该忽略TBE标志位，并且在RBNE位置1后，读出接收缓冲区内的数据，否则，将会产生接收过载错误。

除了忽略TBE标志位，且只执行上述的接收流程之外，从机接收模式（SRU或SRB）与全双工模式类似。

SPI 停止流程

不同运行模式下采用不同的流程来停止SPI功能。

MFD SFD

等待最后一个RBNE位并接收最后一个数据，等待TBE=1和TRANS=0，最后，通过清零SPIEN位关闭SPI。

MTU MTB STU STB

将最后一个数据写入SPI_DATA寄存器，等待TBE位置1，等待TRANS位清零，通过清零SPIEN位关闭SPI。

MRU MRB

等待倒数第二个RBNE位置1，从SPI_DATA寄存器读数据，等待一个SCK时钟周期，然后通过清零SPIEN位关闭SPI。等待最后一个RBNE位置1，并从SPI_DATA读数据。

SRU SRB

应用程序可以在任何时候关闭SPI功能，然后等待TRANS=0以确保当前通信过程结束。

19.3.6. DMA 功能

DMA功能在传输过程中将应用程序从数据读写过程中释放出来，从而提高了系统效率。

通过置位SPI_CTL1寄存器中的DMATEN位和DMAREN位，使能SPI模式的DMA功能。为了使用DMA功能，软件首先应当正确配置DMA模块，然后通过初始化流程配置SPI模块，最后使能SPI。

SPI使能后，如果DMATEN位置1，每当TBE=1时，SPI将会发出一个DMA请求，然后DMA应答该请求，并自动写数据到SPI_DATA寄存器。如果DMAREN位置1，每当RBNE=1时，发出一个DMA请求，然后DMA应答该请求，并自动从SPI_DATA寄存器读取数据。

19.3.7. CRC 功能

SPI模块包含两个CRC计算单元：分别用于发送数据和接收数据。CRC计算单元使用SPI_CRCPOLY寄存器中定义的多项式。

通过配置SPI_CTL0中的CRCEN位使能CRC功能。对于数据线上每个发送和接收的数据，CRC单元逐位计算CRC值，计算得到的CRC值可以从SPI_TCRC寄存器和SPI_RCRC寄存器中读取。

为了传输计算得到的CRC值，应用程序需要在最后一个数据写入发送缓冲区之后，设置SPI_CTL0中的CRCNT位。在全双工模式（MFD或SFD），当SPI发送一个CRC值并且准备校验接收到的CRC值时，会将最新接收到的数据当作CRC值。在接收模式（MRB，MRU，SRU和SRB）下，在倒数第二个数据帧被接收后，软件将CRCNT位置1。在CRC校验失败时，CRCERR错误标志位将会置1。

如果使能了DMA功能，软件不需要设置CRCNT位，硬件将会自动处理CRC传输和校验。

注意：当SPI处于从机模式且CRC功能使能时，无论SPI是否使能，CRC计算器都对输入SCK时钟敏感。只有当时钟稳定时，软件才能启用CRC，以避免错误的CRC计算。当SPI作为从机工作时，在数据阶段和CRC阶段之间，内部NSS信号需要保持低电平。

19.3.8. SPI 中断

状态标志位

■ **发送缓冲区空标志位（TBE）**

当发送缓冲区为空时，TBE置位。软件可以通过写SPI_DATA寄存器将下一个待发送数据写入发送缓冲区。

■ **接收缓冲区非空标志位（RBNE）**

当接收缓冲区非空时，RBNE置位，表示此时接收到一个数据，并已存入到接收缓冲区中，软件可以通过读SPI_DATA寄存器来读取此数据。

■ **SPI通信进行中标志位（TRANS）**

TRANS位是用来指示当前传输是否正在进行或结束的状态标志位，它由内部硬件置位和清除，无法通过软件控制。该标志位不会产生任何中断。

错误标志

■ **配置错误标志（CONFERR）**

在主机模式中，CONFERR位是一个错误标志位。在硬件NSS模式中，如果NSSDRV没有使能，当NSS被拉低时，CONFERR位被置1。在软件NSS模式中，当SWNSS位为0时，CONFERR位置1。当CONFERR位置1时，SPIEN位和MSTMOD位由硬件清除，SPI关闭，设备强制进入从机模式。

在CONFERR位清零之前，SPIEN位和MSTMOD位保持写保护，从机的CONFERR位不能置1。在多主机配置中，设备可以在CONFERR位置1时进入从机模式，这意味着发生了系统控制的多主冲突。

■ **接收过载错误（RXORERR）**

在RBNE位为1时，如果再有数据被接收，RXORERR位将会置1。这说明，上一帧数据还未被读出而新的数据已经接收了。接收缓冲区的内容不会被新接收的数据覆盖，所以新接收的数据丢失。

■ **CRC错误（CRCERR）**

当CRCEN位置1时，SPI_RCRC寄存器中接收到的数据的CRC计算值将会和紧随着最后一帧数据接收到的CRC值进行比较，当两者不同时，CRCERR位将会置1。

表 19-5. SPI 中断请求

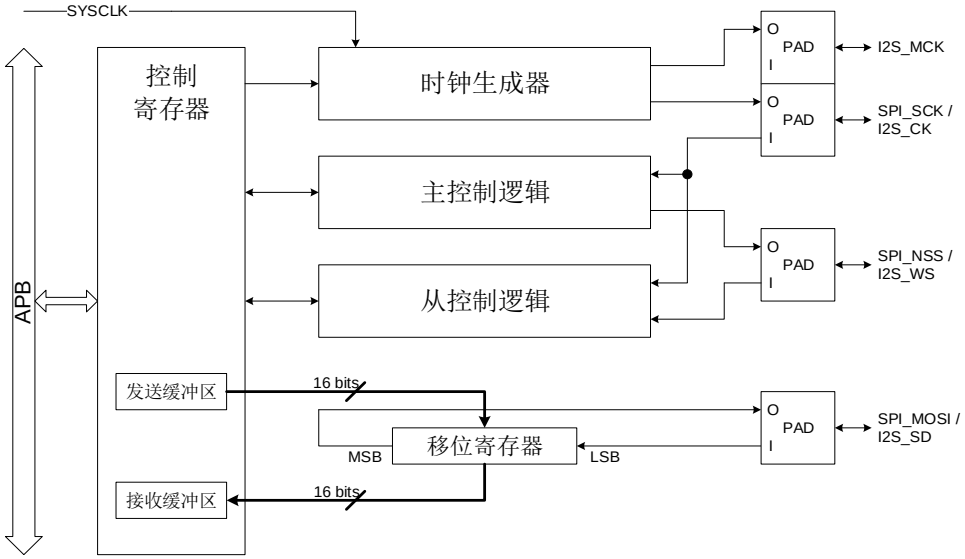
中断事件	描述	清除方式	中断使能位
TBE	发送缓冲区空	写 SPI_DATA 寄存器	TBEIE
RBNE	接收缓冲区非空	读 SPI_DATA 寄存器	RBNEIE
CONFERR	配置错误	读或写 SPI_STAT 寄存器，然后写 SPI_CTL0 寄存器	ERRIE
RXORERR	接收过载错误	读 SPI_DATA 寄存器，然后读	

中断事件	描述	清除方式	中断使能位
		SPI_STAT 寄存器	
CRCERR	CRC 错误	写 0 到 CRCERR 位	

19.4. I2S 功能说明

19.4.1. I2S 结构框图

图 19-7. I2S 结构框图



I2S功能有5个子模块，分别是控制寄存器、时钟生成器、主机控制逻辑、从机控制逻辑和移位寄存器。所有的用户可配置寄存器都在控制寄存器模块实现，其中包括发送缓冲区和接收缓冲区。时钟生成器用来在主机模式下生成I2S通信时钟。主机控制逻辑用来在主机模式下生成I2S_WS信号并控制通信。从机控制逻辑根据接收到的I2S_CK和I2S_WS信号来控制从机模式的通信。移位寄存器控制I2S_SD上的串行数据发送和接收。

19.4.2. I2S 信号线描述

I2S接口有4个引脚，分别是I2S_CK、I2S_WS、I2S_SD和I2S_MCK。I2S_CK是串行时钟信号，与SPI_SCK共享引脚。I2S_WS是数据帧控制信号，与SPI_NSS共享引脚。I2S_SD是串行数据信号，与SPI_MOSI共享引脚。I2S_MCK是主时钟信号，它最大可提供一个256倍于Fs的时钟频率，其中Fs是音频采样率。

19.4.3. I2S 音频标准

I2S音频标准是通过设置SPI_I2SCTL寄存器中的I2SSTD位来选择的，可以选择四种音频标准：I2S飞利浦标准，MSB对齐标准，LSB对齐标准和PCM标准。除PCM之外的所有标准都是两个通道（左通道和右通道）的音频数据分时复用I2S接口的，并通过I2S_WS信号来区分当前数据属于哪个通道。对于PCM标准，I2S_WS信号表示帧同步信息。

数据长度和通道长度可以通过SPI_I2SCTL寄存器中的DTLEN位和CHLEN位来设置。由于通道长度必须大于或等于数据长度，所以有四种数据包类型可供选择。它们分别是：16位数据打包成16位数据帧格式，16位数据打包成32位数据帧格式，24位数据打包成32位数据帧格式，32位数据打包成32位数据帧格式。用于发送和接收的数据缓冲区都是16位宽度。所以，要完成数据长度为24位或32位的数据帧传输，SPI_DATA寄存器需要被访问2次；而要完成数据长度为16位的数据帧传输，SPI_DATA寄存器只需被访问1次。如需将16位数据打包成32位数据帧，硬件会自动插入16位0将16位数据扩展为32位格式。

对于所有标准和数据包类型来说，数据的最高有效位总是最先被发送的。对于所有基于两通道分时复用的标准来说，总是先发送左通道，然后是右通道。

I2S 飞利浦标准

对于I2S飞利浦标准，I2S_WS和I2S_SD在I2S_CK的下降沿变化，I2S_WS在数据的前一个时钟开始有效。各种配置情况的时序图如下所示。

图 19-8. I2S 飞利浦标准时序图（DTLEN=00, CHLEN=0, CKPL=0）

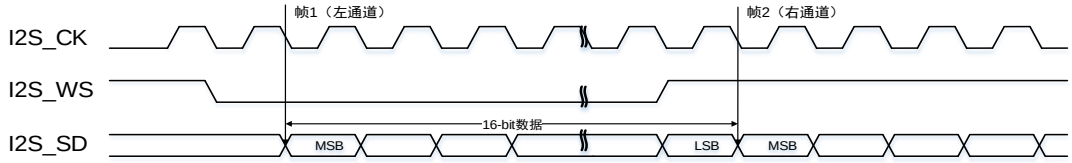
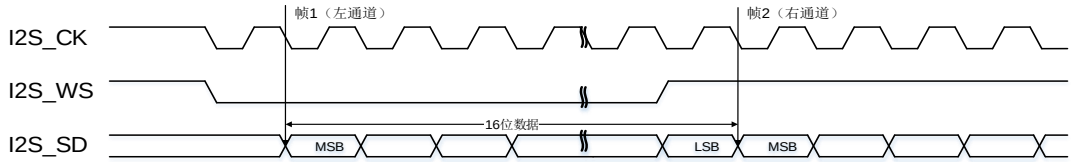


图 19-9. I2S 飞利浦标准时序图（DTLEN=00, CHLEN=0, CKPL=1）



当 16 位数据打包成 16 位数据帧时，每完成一帧数据的传输只需要访问 SPI_DATA 寄存器一次。

图 19-10. I2S 飞利浦标准时序图（DTLEN=10, CHLEN=1, CKPL=0）

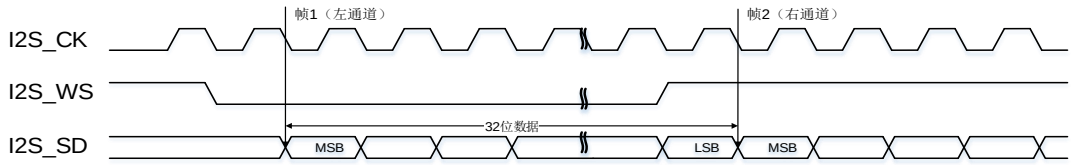
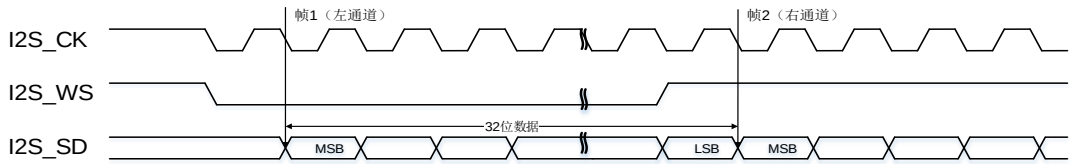


图 19-11. I2S 飞利浦标准时序图（DTLEN=10, CHLEN=1, CKPL=1）



当32位数据打包成32位数据帧的帧格式时，每完成1帧数据的传输需要访问SPI_DATA寄存器

2次。在发送模式下，如果要发送一个32位数据，第一个写入SPI_DATA寄存器的数据应该是高16位数据，第二个数据应该是低16位数据。在接收模式下，如果要接收一个32位数据，第一个从SPI_DATA寄存器读到的数据应该是高16位数据，第二个数据应该是低16位数据。

图 19-12. I2S 飞利浦标准时序图（DTLEN=01, CHLEN=1, CKPL=0）

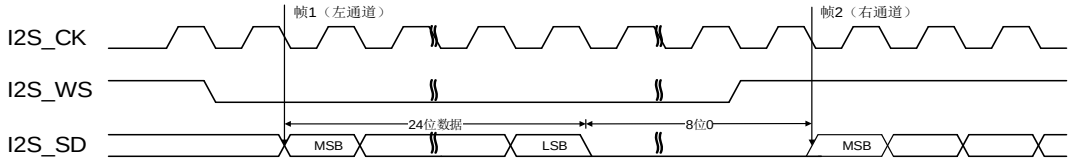
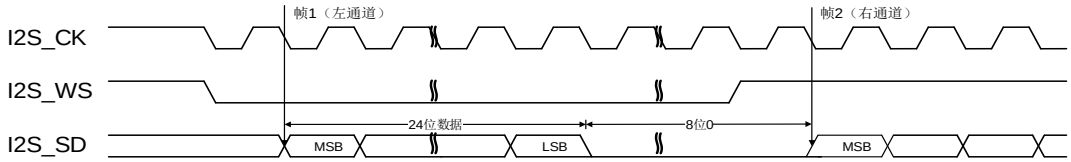


图 19-13. I2S 飞利浦标准时序图（DTLEN=01, CHLEN=1, CKPL=1）



当24位数据打包成32位数据帧的帧格式时，每完成1帧数据的传输需要访问SPI_DATA寄存器2次。在发送模式下，如果要发送一个24位数据D[23:0]，第一个写入SPI_DATA寄存器的数据应该是高16位数据D[23:8]，第二个数据应该是一个16位数据，该16位数据的高8位是D[7:0]，低8位数据可以是任意值。在接收模式下，如果要接收一个24位数据D[23:0]，第一个从SPI_DATA寄存器读到的数据应该是高16位数据D[23:8]，第二个数据应该是一个16位数据，该16位数据的高8位是D[7:0]，低8位数据全是0。

图 19-14. I2S 飞利浦标准时序图（DTLEN=00, CHLEN=1, CKPL=0）

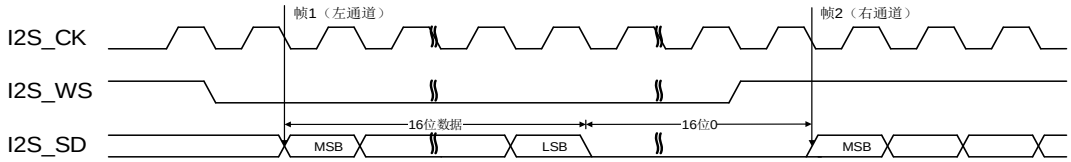
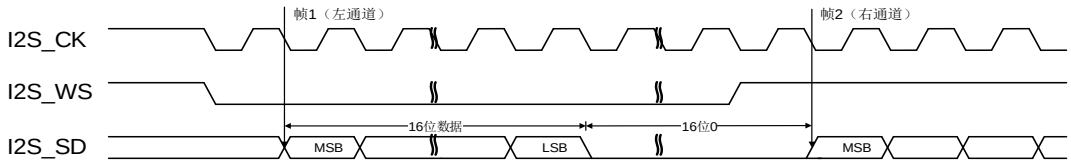


图 19-15. I2S 飞利浦标准时序图（DTLEN=00, CHLEN=1, CKPL=1）



当16位数据打包成32位数据帧时，每完成一帧数据的传输只需要访问SPI_DATA寄存器一次。为了将该16位数据扩展成32位数据，剩下的16位被硬件强制填充为0x0000。

MSB 对齐标准

对于MSB对齐标准，I2S_WS和I2S_SD在I2S_CK的下降沿变化。SPI_DATA寄存器的处理方式与I2S飞利浦标准完全相同。各个配置情况的时序图如下所示。

图 19-16. MSB 对齐标准时序图（DTLEN=00, CHLEN=0, CKPL=0）

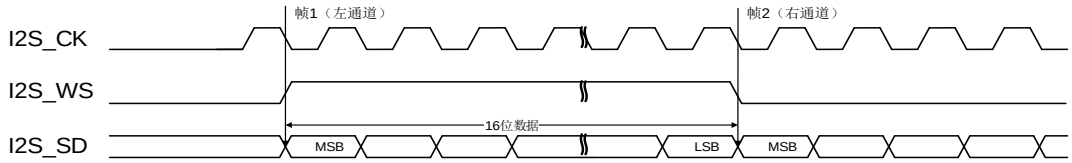


图 19-17. MSB 对齐标准时序图（DTLEN=00, CHLEN=0, CKPL=1）

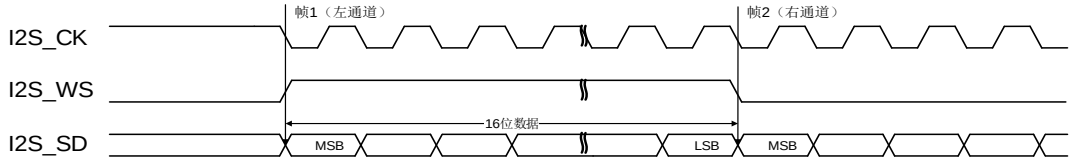


图 19-18. MSB 对齐标准时序图（DTLEN=10, CHLEN=1, CKPL=0）

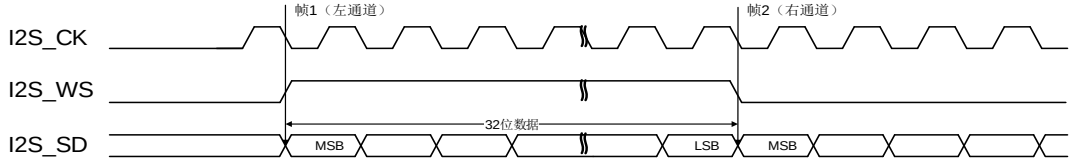


图 19-19. MSB 对齐标准时序图（DTLEN=10, CHLEN=1, CKPL=1）

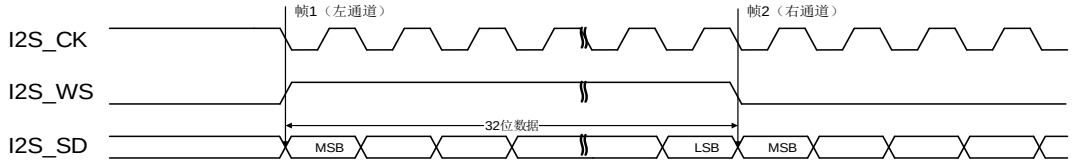


图 19-20. MSB 对齐标准时序图（DTLEN=01, CHLEN=1, CKPL=0）

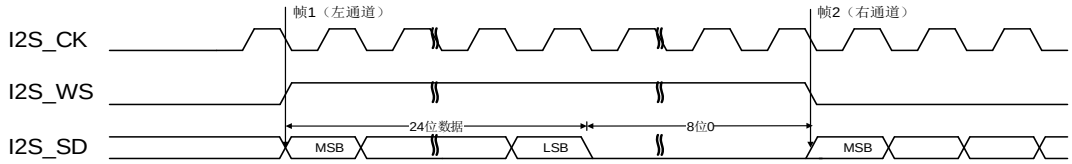


图 19-21. MSB 对齐标准时序图（DTLEN=01, CHLEN=1, CKPL=1）

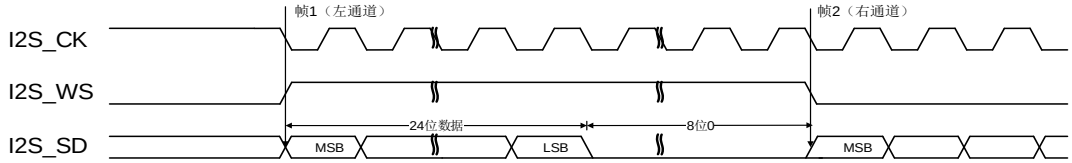


图 19-22. MSB 对齐标准时序图（DTLEN=00, CHLEN=1, CKPL=0）

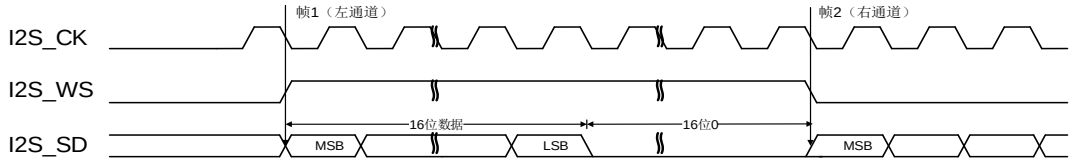
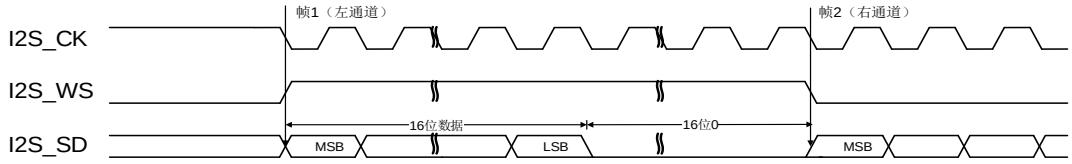


图 19-23. MSB 对齐标准时序图（DTLEN=00, CHLEN=1, CKPL=1）



LSB 对齐标准

对于LSB对齐标准，I2S_WS和I2S_SD在I2S_CK的下降沿变化。在通道长度与数据长度相同的情况下，LSB对齐标准和MSB对齐标准是完全相同的。对于通道长度大于数据长度的情况，LSB对齐标准的有效数据与最低位对齐，而MSB对齐标准的有效数据与最高位对齐。通道长度大于数据长度的各种配置情况时序图如下所示。

图 19-24. LSB 对齐标准时序图（DTLEN=01, CHLEN=1, CKPL=0）

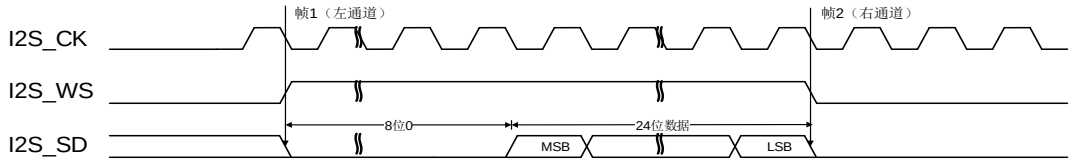
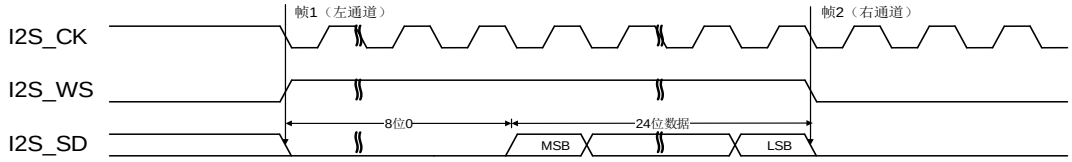


图 19-25. LSB 对齐标准时序图（DTLEN=01, CHLEN=1, CKPL=1）



当24位数据打包成32位数据帧的帧格式时，每完成1帧数据的传输需要访问SPI_DATA寄存器2次。在发送模式下，如果要发送一个24位数据D[23:0]，第一个写入SPI_DATA寄存器的数据应该是一个16位数据，该16位数据的高8位可以是任意值，低8位是D[23:16]，第二个数据应该是低16位数据D[15:0]。在接收模式下，如果要接收一个24位数据D[23:0]，第一个从SPI_DATA寄存器读到的数据应该是一个16位数据，该16位数据的高8位是0，低8位是D[23:16]，第二个数据应该是低16位数据D[15:0]。

图 19-26. LSB 对齐标准时序图（DTLEN=00, CHLEN=1, CKPL=0）

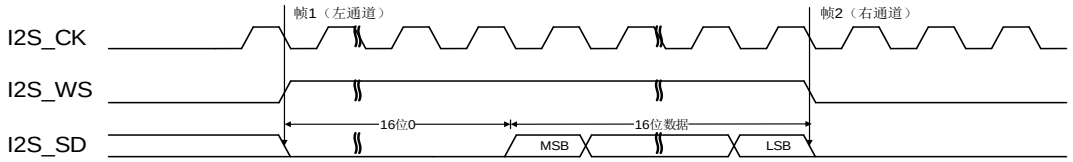
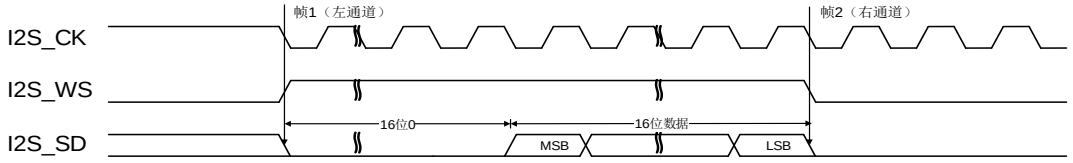


图 19-27. LSB 对齐标准时序图（DTLEN=00, CHLEN=1, CKPL=1）



当16位数据打包成32位数据帧时，每完成一帧数据的传输只需要访问SPI_DATA寄存器一次。为了将该16位数据扩展成32位数据，剩下的16位被硬件强制填充为0x0000。

PCM 标准

对于PCM标准，I2S_WS和I2S_SD在I2S_CK的上升沿变化，I2S_WS信号表示帧同步信息。可以通过SPI_I2SCTL寄存器的PCMSMOD位来选择短帧同步模式和长帧同步模式。SPI_DATA寄存器的处理方式与I2S飞利浦标准完全相同。短帧同步模式的各种配置情况时序图如下所示。

图 19-28. PCM 标准短帧同步模式时序图（DTLEN=00, CHLEN=0, CKPL=0）

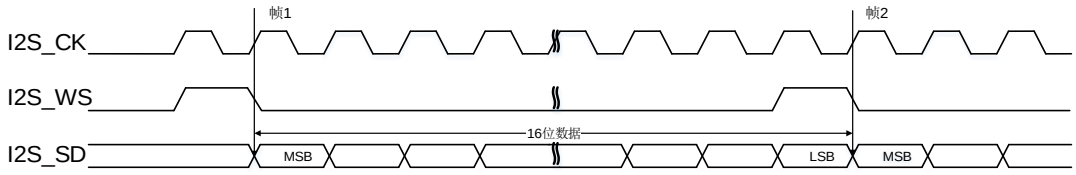


图 19-29. PCM 标准短帧同步模式时序图（DTLEN=00, CHLEN=0, CKPL=1）

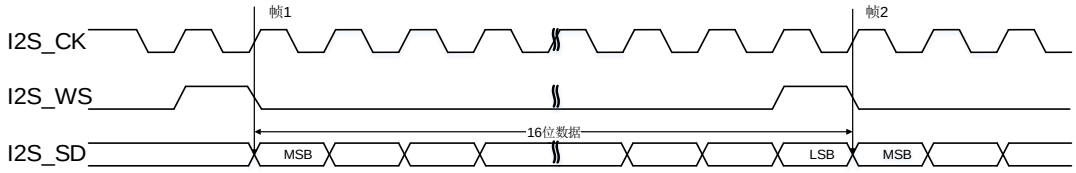


图 19-30. PCM 标准短帧同步模式时序图（DTLEN=10, CHLEN=1, CKPL=0）

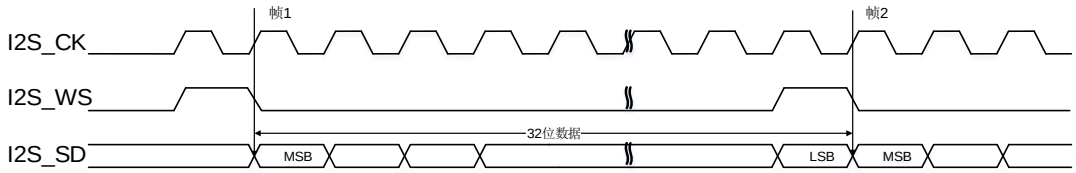


图 19-31. PCM 标准短帧同步模式时序图（DTLEN=10, CHLEN=1, CKPL=1）

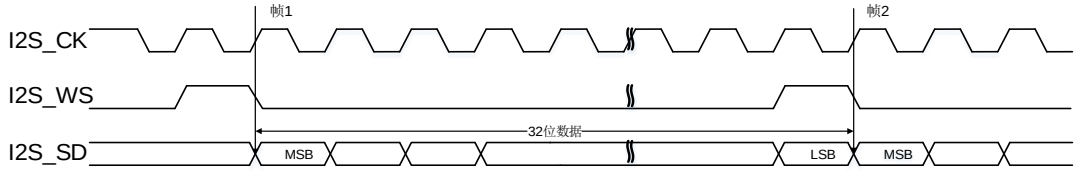


图 19-32. PCM 标准短帧同步模式时序图（DTLEN=01, CHLEN=1, CKPL=0）

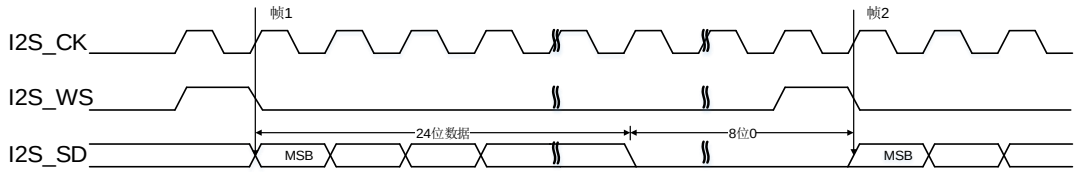


图 19-33. PCM 标准短帧同步模式时序图（DTLEN=01, CHLEN=1, CKPL=1）

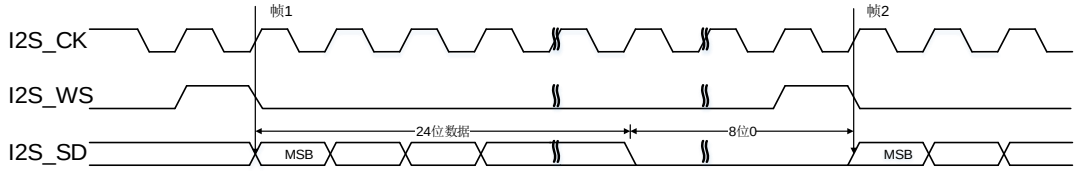


图 19-34. PCM 标准短帧同步模式时序图（DTLEN=00, CHLEN=1, CKPL=0）

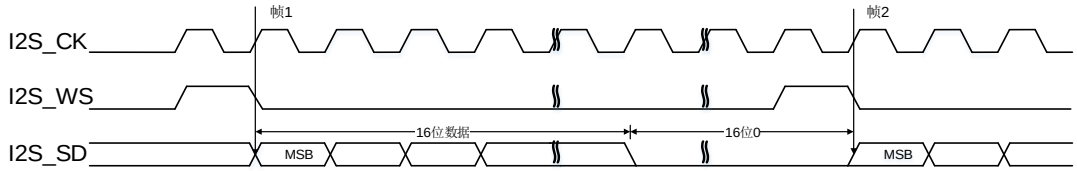
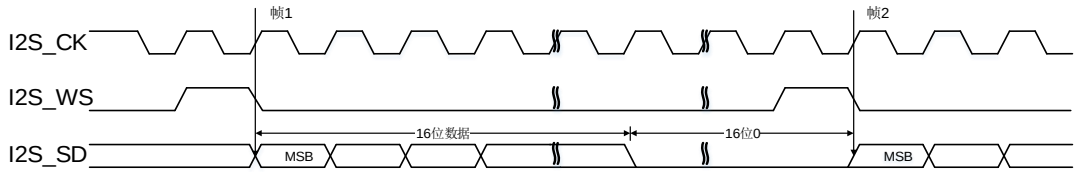


图 19-35. PCM 标准短帧同步模式时序图（DTLEN=00, CHLEN=1, CKPL=1）



长帧同步模式的各种配置情况时序图如下所示。

图 19-36. PCM 标准长帧同步模式时序图（DTLEN=00, CHLEN=0, CKPL=0）

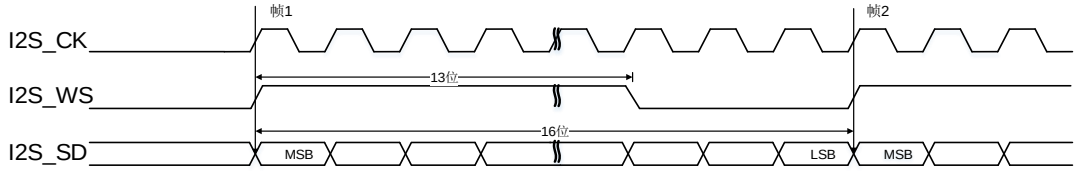


图 19-37. PCM 标准长帧同步模式时序图（DTLEN=00, CHLEN=0, CKPL=1）

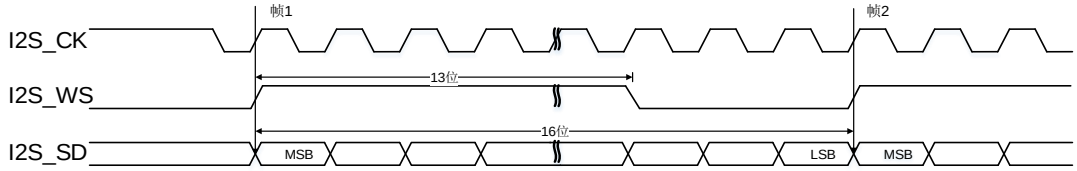


图 19-38. PCM 标准长帧同步模式时序图（DTLEN=10, CHLEN=1, CKPL=0）

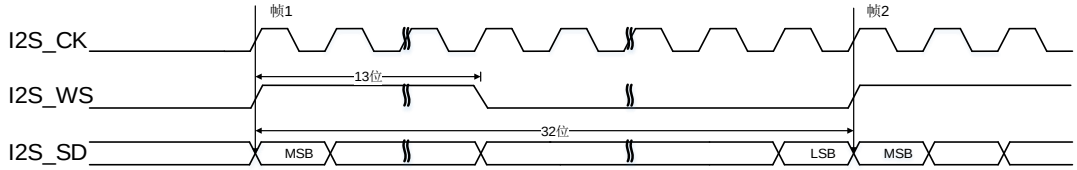


图 19-39. PCM 标准长帧同步模式时序图（DTLEN=10, CHLEN=1, CKPL=1）

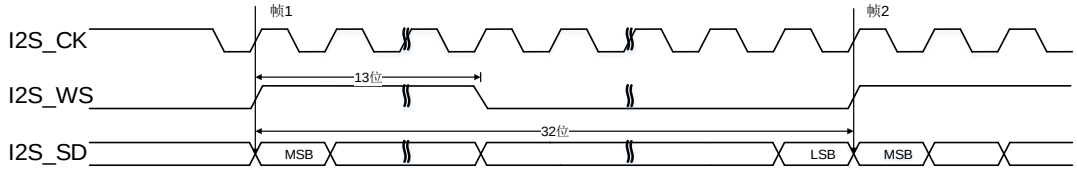


图 19-40. PCM 标准长帧同步模式时序图（DTLEN=01, CHLEN=1, CKPL=0）

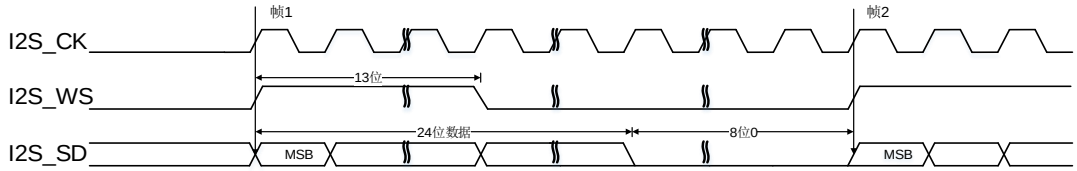


图 19-41. PCM 标准长帧同步模式时序图（DTLEN=01, CHLEN=1, CKPL=1）

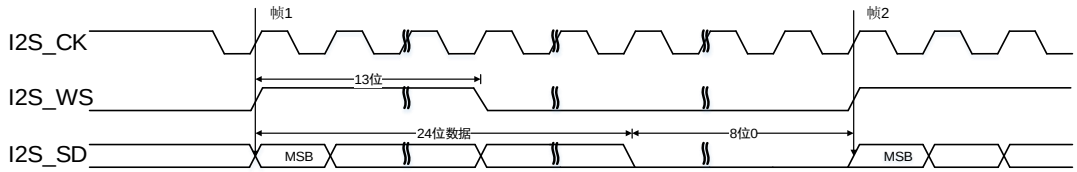


图 19-42. PCM 标准长帧同步模式时序图（DTLEN=00, CHLEN=1, CKPL=0）

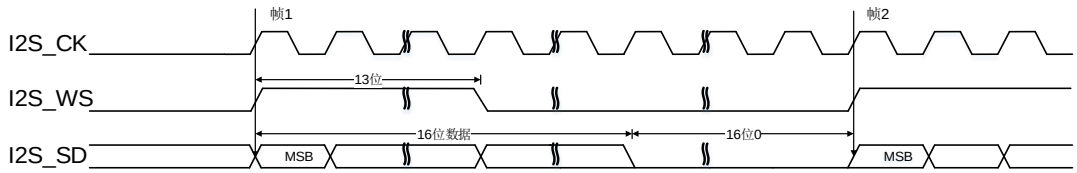
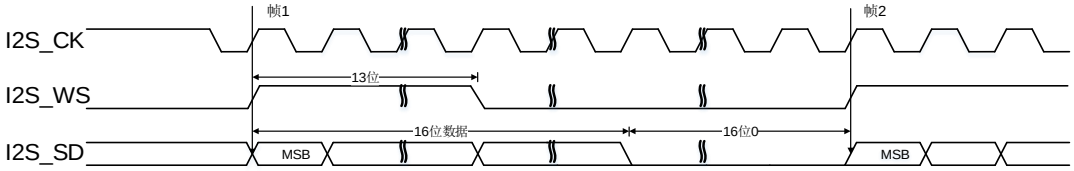
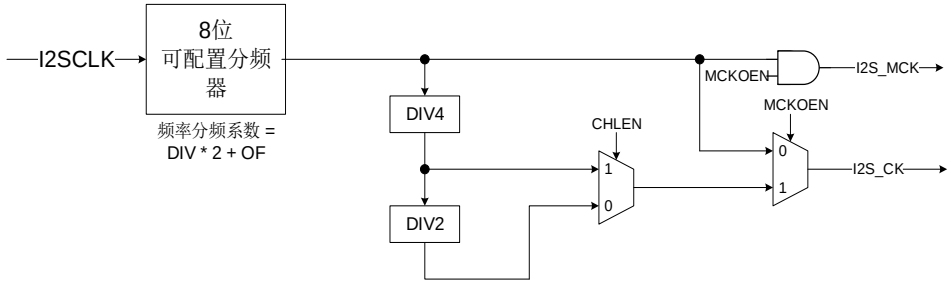


图 19-43. PCM 标准长帧同步模式时序图（DTLEN=00, CHLEN=1, CKPL=1）



19.4.4. I2S 时钟

图 19-44. I2S 时钟生成结构框图



I2S时钟生成器框图如*图19-44. I2S时钟生成结构框图*所示。I2S接口时钟是通过SPI_I2SPSC寄存器的DIV位，OF位和MCKOEN位以及SPI_I2SCTL寄存器的CHLEN位来配置的。时钟源是系统时钟（CK_SYS）。I2S比特率可以通过*表19-6. I2S比特率计算公式*所示的公式计算。

表 19-6. I2S 比特率计算公式

MCKOEN	CHLEN	公式
0	0	I2SCLK / (DIV * 2 + OF)
0	1	I2SCLK / (DIV * 2 + OF)
1	0	I2SCLK / (8 * (DIV * 2 + OF))
1	1	I2SCLK / (4 * (DIV * 2 + OF))

音频采样率（Fs）和I2S比特率的关系由如下公式定义：

Fs = I2S比特率 / （通道长度 * 通道数）

所以，为了得到期望的音频采样率，时钟生成器需要按*表19-7. 音频采样频率计算公式*所列的公式进行配置。

表 19-7. 音频采样频率计算公式

MCKOEN	CHLEN	公式
0	0	I2SCLK / (32 * (DIV * 2 + OF))
0	1	I2SCLK / (64 * (DIV * 2 + OF))
1	0	I2SCLK / (256 * (DIV * 2 + OF))
1	1	I2SCLK / (256 * (DIV * 2 + OF))

注意：I2S串行时钟的配置值需设置为低于PCLK时钟的1/6倍以下(不包含1/6)。

19.4.5. 运行

运行模式

运行模式是通过SPI_I2SCTL寄存器的I2SOPMOD[1:0]位域来选择的。共有四种运行模式可供选择：主机发送模式，主机接收模式，从机发送模式和从机接收模式。各种运行模式下I2S接口信号的方向如*表19-8. 各种运行模式下I2S接口信号的方向*所示。

表 19-8. 各种运行模式下 I2S 接口信号的方向

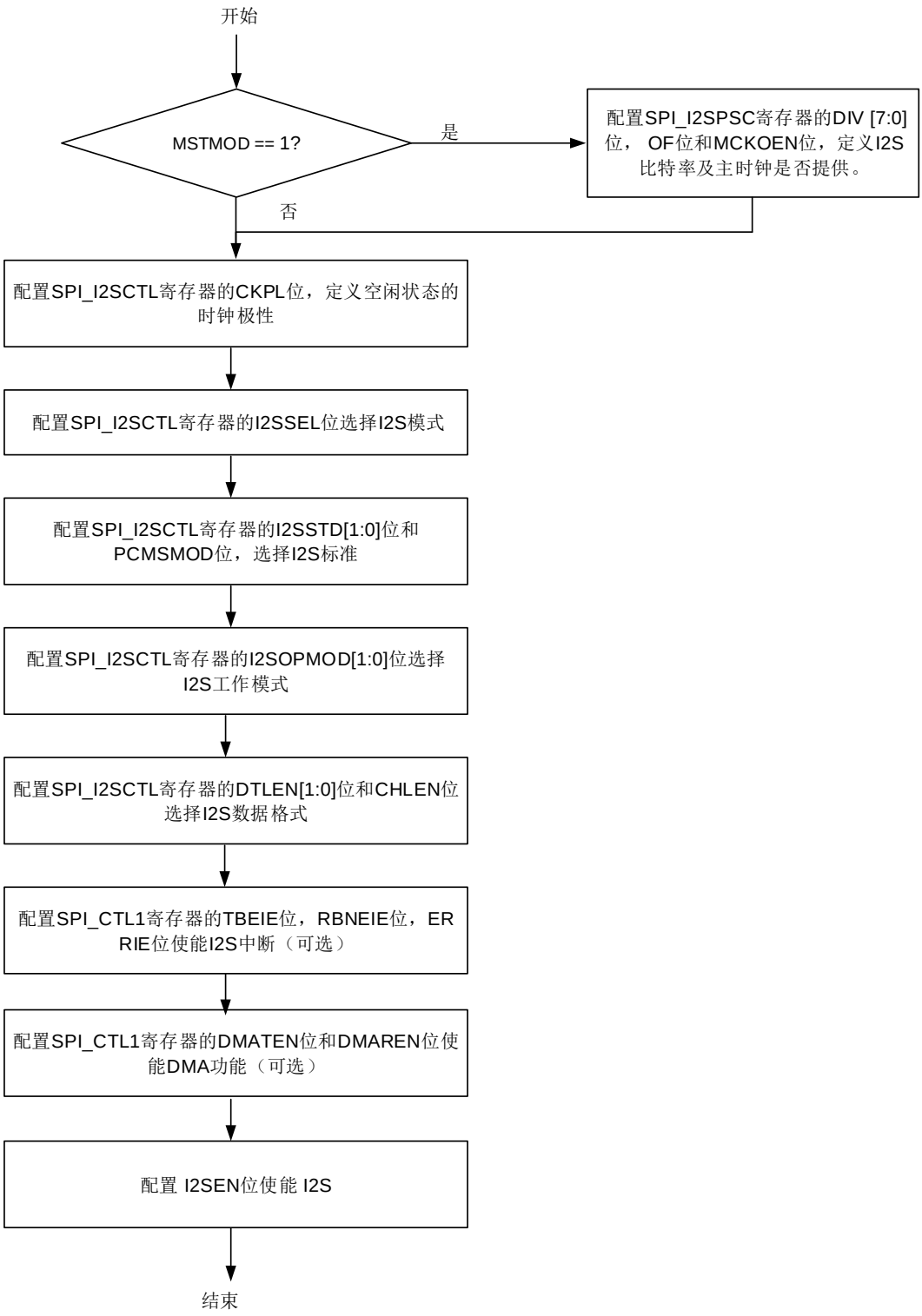
运行模式	I2S_MCK	I2S_CK	I2S_WS	I2S_SD
主机发送	输出或 NU(1)	输出	输出	输出
主机接收	输出或 NU(1)	输出	输出	输入
从机发送	输出或 NU(1)	输入	输入	输出
从机接收	输出或 NU(1)	输入	输入	输入

1. NU表示该引脚没有被I2S使用，可以用于其他功能。

I2S 初始化流程

I2S初始化过程如*图19-45. I2S初始化流程*所示。

图 19-45. I2S 初始化流程



I2S 主机发送流程

TBE标志位被用来控制发送流程。如前文所述，TBE标志位置位表示发送缓冲区空，此时，如果SPI_CTL1寄存器的TBEIE位为1，将产生中断。首先，发送缓冲区为空（TBE为1），且移位

寄存器中没有发送序列。当16位数据被写入SPI_DATA寄存器时（TBE变为0），数据立即从发送缓冲区装载到移位寄存器中（TBE变为1）。此时，发送序列开始。

数据是并行地装载到16位移位寄存器中的，然后串行地从I2S_SD引脚发出（高位先发）。下一个数据应该在TBE为1时写入SPI_DATA寄存器。数据写入SPI_DATA寄存器之后，TBE变为0。当前发送序列结束时，发送缓冲区的数据会自动装载到移位寄存器中，然后TBE标志变回1。为了保证连续的音频数据发送，下一个将要发送的数据必须在当前发送序列结束之前写入SPI_DATA寄存器。

对于除PCM标准外的所有标准，I2SCH标志用来区别当前传输数据所属的通道。I2SCH标志在每次TBE标志由0变1的时候更新。刚开始I2SCH标志为0，表示左通道的数据应该被写入SPI_DATA寄存器。

为了关闭I2S，I2SEN位必须在TBE标志为1且TRANS标志为0之后清零。

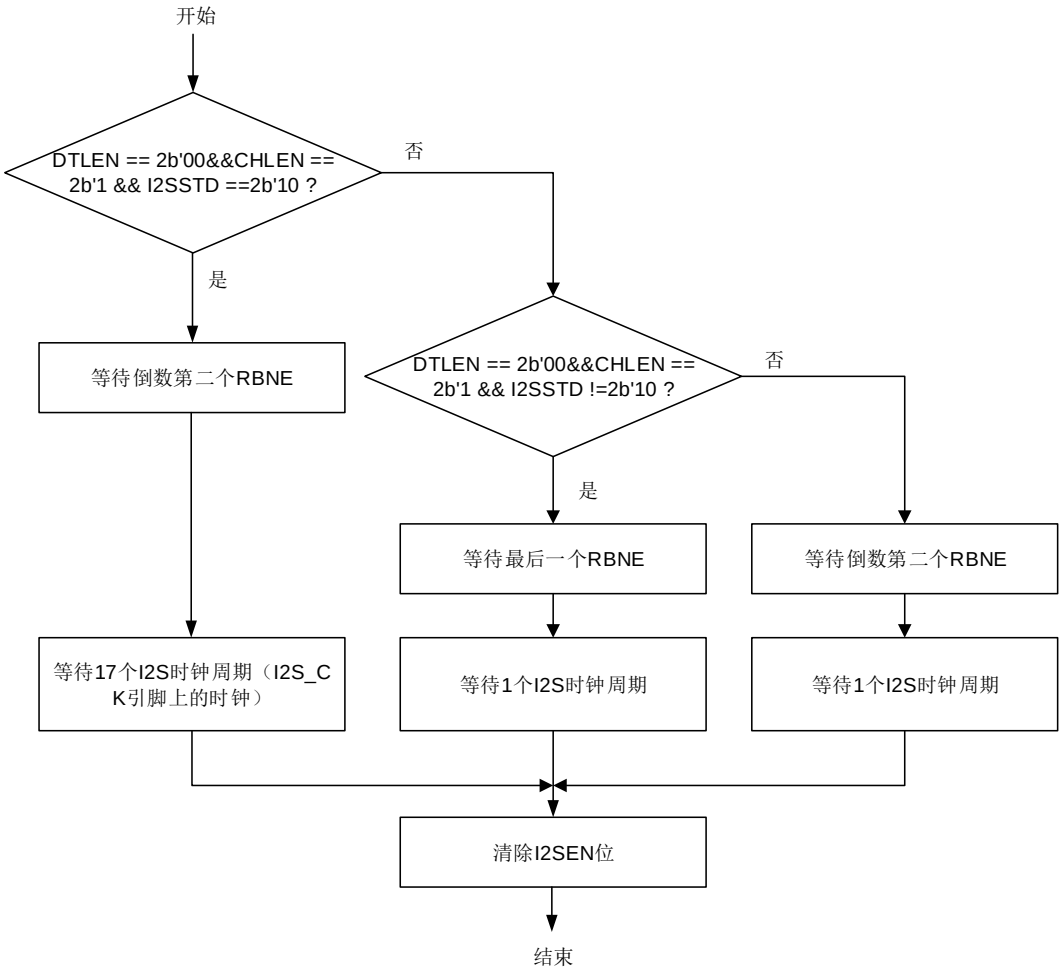
I2S 主机接收流程

RBNE标志被用来控制接收序列。如前文所述，RBNE标志置位表示接收缓冲区非空，如果SPI_CTL1寄存器的RBNEIE位为1，将产生中断。当SPI_I2SCTL寄存器的I2SEN位被置1时，接收流程立即开始。首先，接收缓冲区为空（RBNE为0）。当一个接收流程结束时，接收到的数据将从移位寄存器装载到接收缓冲区（RBNE变为1）。当RBNE为1时，用户应该将数据从SPI_DATA寄存器中读走。读操作完成后，RBNE变为0。必须在下一次接收结束之前读走SPI_DATA寄存器中的数据，否则将发生接收过载错误。此时RXORERR标志位会被置1，如果SPI_CTL1寄存器的ERRIE位为1，将会产生中断。这种情况下，必须先关闭I2S再打开I2S，然后再恢复通讯。

对于除PCM之外的所有标准来说，I2SCH标志用来区分当前传输数据所属的通道。I2SCH标志在每次RBNE标志由0变1时更新。

为了关闭I2S，不同的音频标准，数据长度和通道长度采用不同的操作步骤。每种情况的操作如*图19-46. I2S主机接收禁能流程*所示。

图 19-46. I2S 主机接收禁能流程



I2S 从机发送流程

从机发送流程和主机发送流程相似，不同之处如下：

在从机模式下，从机需要在外部主机开始通讯之前使能。当外部主机开始发送时钟信号且I2S_WS信号请求传输数据时，发送流程开始。数据需要在外部主机发起通讯之前写入SPI_DATA寄存器。为了确保音频数据的连续传输，必须在当前发送序列结束之前将下一个待发送的数据写入SPI_DATA寄存器，否则会产生发送欠载错误。此时TXURERR标志会置1，如果SPI_CTL1寄存器的ERRIE位为1，将会产生中断。这种情况下，必须先关闭I2S再打开I2S来恢复通讯。从机模式下，I2SCH标志是根据外部主机发送的I2S_WS信号而变化的。

为了关闭I2S，必须在TBE标志变为1且TRANS标志变为0之后，才能清除I2SEN位。

I2S 从机接收流程

从机接收流程与主机接收流程类似。不同之处如下。

在从机模式下，从机需要在外部主机开始通讯之前使能。当外部主机开始发送时钟信号且I2S_WS信号指示数据开始时，接收流程开始。从机模式下，I2SCH标志是根据外部主机发送

的I2S_WS信号而变化的。

为了关闭I2S，必须在收到最后一个RBNE之后立即清除I2SEN位。

19.4.6. DMA 功能

DMA功能与SPI模式完全一样，唯一不同的地方就是I2S模式不支持CRC功能。

19.4.7. I2S 中断

状态标志位

SPI_STAT寄存器中有4个可用的标志位，分别是TBE、RBNE、TRANS和I2SCH，用户通过这些标志位可以全面监视I2S总线的状态。

- **发生缓冲区空标志（TBE）**
  当发送缓冲区为空时，TBE置位。软件可以通过写SPI_DATA寄存器将下一个数据写入发送缓冲区。
- **接收缓冲区非空标志（RBNE）**
  接收缓冲区非空时，RBNE置位，表示此时接收到一个数据，并已存入接收缓冲区中，软件可以通过读SPI_DATA寄存器来读取此数据。
- **I2S通信进行中标志（TRANS）**
  TRANS是用来指示当前传输是否正在进行或结束的状态标志，它由内部硬件置位和清除，无法进行软件操作。该标志位不会产生任何中断。
- **I2S通道标志（I2SCH）**
  I2SCH用来表明当前传输数据的通道信息，对PCM音频标准来说没有意义。在发送模式下，I2SCH标志在每次TBE由0变1时更新，在接收模式下，I2SCH标志在每次RBNE由0变1时更新。该标志位不会产生任何中断。

错误标志

有三个错误标志：

- **发送欠载错误标志（TXURERR）**
  在从发送模式下，有效的SCK信号开始发送，当发送缓冲区为空时，发送欠载错误标志TXURERR置位。
- **接收过载错误标志（RXORERR）**
  当接收缓冲区已满且又接收到一个新的数据时，接收过载错误标志RXORERR置位。当接收过载发生时，接收缓冲区中的数据没有更新，新接收的数据丢失。
- **帧格式错误（FERR）**
  在从I2S模式下，I2S模块监视I2S_WS信号，如果I2S_WS信号在一个错误的位置发生翻转，将会置位FERR帧错误标志位。

*表19-9. I2S中断*总结了I2S中断事件和相应的使能位。

表 19-9. I2S 中断

中断标志	描述	清除方式	中断使能位
TBE	发送缓冲区空	写 SPI_DATA 寄存器	TBEIE
RBNE	接收缓冲区非空	读 SPI_DATA 寄存器	RBNEIE
TXURERR	发送欠载错误	读 SPI_STAT 寄存器	ERRIE
RXORERR	接收过载错误	读 SPI_DATA 寄存器，然后再读 SPI_STAT 寄存器	
FERR	I2S 帧错误	读 SPI_STAT 寄存器	

19.5. SPI/I2S 寄存器

SPI0/I2S0基地址：0x4001 3000

SPI1基地址：0x4000 3800

SPI2/I2S2基地址：0x4000 3C00

19.5.1. 控制寄存器 0（SPI_CTL0）

地址偏移：0x00
复位值：0x0000 0000

该寄存器可以按字节（8位）或半字（16位）或字（32位）访问。

该寄存器在I2S模式下没有意义。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BDEN	BDOEN	CRCEN	CRCNT	FF16	RO	SWNSSEN	SWNSS	LF	SPIEN	PSC [2:0]			MSTMOD	CKPL	CKPH
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	BDEN	双向数据模式 0：2 线单向传输模式。 1：1 线双向传输模式。数据在主机的 MOSI 引脚和从机的 MISO 引脚之间传输。
14	BDOEN	双向传输输出使能 当 BDEN 置位时，该位决定了数据的传输方向。 0：工作在只接收模式。 1：工作在只发送模式。
13	CRCEN	CRC计算使能 0：禁能CRC计算。 1：使能 CRC 计算。
12	CRCNT	下一次传输 CRC 0：下一次传输值为数据。 1：下一次传输值为 CRC 值（TCRC）。 当数据传输由 DMA 管理时，CRC 值由硬件传输，该位应该被清零。 在全双工和只发送模式下，当最后一个数据写入 SPI_DATA 寄存器后应将该位置 1。 在只接收模式下，在接收完倒数第二个数据后应将该位置 1。
11	FF16	数据帧格式

		0：8 位数据帧格式 1：16 位数据帧格式
10	RO	只接收模式 当 BDEN 清零时，该位决定了数据的传输方向。 0：全双工模式 1：只接收模式
9	SWNSSEN	NSS 软件模式选择 0：NSS 硬件模式，NSS 电平取决于 NSS 引脚。 1：NSS 软件模式，NSS 电平取决于 SWNSS 位。
8	SWNSS	NSS 软件模式下 NSS 引脚选择 0：NSS 引脚拉低。 1：NSS 引脚拉高。 只有在 SWNSSEN 置位时，该位有效。
7	LF	最低有效位先发模式 0：先发送最高有效位。 1：先发送最低有效位。
6	SPIEN	SPI使能 0：禁能SP 1：使能 SPI
5:3	PSC[2:0]	主时钟预分频选择 000：PCLK/2 001：PCLK/4 010：PCLK/8 011：PCLK/16 100：PCLK/32 101：PCLK/64 110：PCLK/128 111：PCLK/256 当使用 SPI0 时，PCLK=PCLK2，当使用 SPI1 时，PCLK=PCLK1。
2	MSTMOD	主从模式使能 0：从机模式 1：主机模式
1	CKPL	时钟极性选择 0：SPI 为空闲状态时，CLK 引脚拉低。 1：SPI 为空闲状态时，CLK 引脚拉高。
0	CKPH	时钟相位选择 0：在第一个时钟跳变沿采集第一个数据。 1：在第二个时钟跳变沿时钟跳变沿采集第一个数据。

19.5.2. 控制寄存器 1（SPI_CTL1）

地址偏移：0x04

复位值：0x0000 0000

该寄存器可以按字节（8位）或半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								TBEIE	RBNEIE	ERRIE	保留		NSSDRV	DMATEN	DMAREN
								rw	rw	rw			rw	rw	rw

位/位域	名称	描述
31:8	保留	必须保持复位值。
7	TBEIE	发送缓冲区空中断使能 0：禁能TBE中断 1：使能 TBE 中断。当 TBE 置位时，产生中断。
6	RBNEIE	接收缓冲区非空中断使能 0：禁能RBNE中断 1：使能 RBNE 中断。当 RBNE 置位时，产生中断。
5	ERRIE	错误中断使能 0：禁能错误中断 1：使能错误中断。当 CRCERR 位，CONFERR 位，RXORERR 位或者 TXURERR 位置 1 时，产生中断。
4:3	保留	必须保持复位值。
2	NSSDRV	NSS输出使能 0：禁能主机NSS输出 1：使能主机 NSS 输出
1	DMATEN	发送缓冲区 DMA 使能 0：禁能发送缓冲区DMA 1：使能发送缓冲区 DMA。当 SPI_STAT 中的 TBE 置位时，将会在相应的 DMA 通道上产生一个 DMA 请求。
0	DMAREN	接收缓冲区 DMA 使能 0：禁能接收缓冲区DMA 1：使能接收缓冲区 DMA。当 SPI_STAT 中的 RBNE 置位时，将会在相应的 DMA 通道上产生一个 DMA 请求。

19.5.3. 状态寄存器（SPI_STAT）

地址偏移：0x08

复位值：0x0000 0002

该寄存器可以按字节（8位）或半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留							FERR	TRANS	RXORERR	CONFERR	CRCERR	TXURERR	I2SCH	TBE	RBNE
							rc_w0	r	r	r	rc_w0	r	r	r	r

位/位域	名称	描述
31:9	保留	必须保持复位值。
8	FERR	帧错误 I2S 模式： 0：没有 I2S 帧错误发生 1：I2S 帧错误发生 该位由硬件置位，可以通过写 0 清除。
7	TRANS	通信进行中标志 0：SPI或I2S空闲 1：SPI或I2S当前正在发送且/或接收数据。 该位由硬件置位和清除。
6	RXORERR	接收过载错误标志 0：没有接收过载错误发生 1：接收过载错误发生 该位由硬件置位，软件序列清零。软件序列为：先读 SPI_DATA 寄存器，然后读 SPI_STAT 寄存器。
5	CONFERR	SPI 配置错误 0：无配置错误发生 1：配置错误发生（主机模式下，在硬件 NSS 模式时 NSS 引脚被拉低，或者软件 NSS 模式时 SWNSS 位为 0，都会产生 CONFERR 错误） 该位由硬件置位，软件序列清零。软件序列为：读或写 SPI_STAT 寄存器，然后写 SPI_CTL0 寄存器。 I2S 模式下不使用该位。
4	CRCERR	SPI CRC 错误标志 0：SPI_RCRC 值等于最后接收到的 CRC 值。 1：SPI_RCRC 值不等于最后接收到的 CRC 值该位由硬件置位，可以通过写 0 清除。 I2S 模式下不使用该位。
3	TXURERR	发送欠载错误标志 0：无发送欠载错误发生 1：发送欠载错误发生 该位由硬件置位，通过读 SPI_STAT 寄存器清除。

		SPI 模式下不使用该位。
2	I2SCH	I2S 通道标志 0：下一个将要发送或当前接收的数据属于左通道。 1：下一个要发送或当前接收的数据属于右通道。 该位由硬件置位和清除。 SPI 模式下不使用该位，I2S PCM 模式下该位没有意义。
1	TBE	发送缓冲区空 0：发送缓冲区非空 1：发送缓冲区空
0	RBNE	接收缓冲区非空 0：接收缓冲区空 1：接收缓冲区非空

19.5.4. 数据寄存器（SPI_DATA）

地址偏移：0x0C
复位值：0x0000 0000

该寄存器可以按字节（8位）或半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI_DATA[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	SPI_DATA[15:0]	数据传输寄存器值 硬件有两个缓冲区：发送缓冲区和接收缓冲区。向 SPI_DATA 写数据将会把数据存入发送缓冲区，从 SPI_DATA 读数据，将从接收缓冲区获得数据。 当数据帧格式为 8 位时，SPI_DATA[15:8]强制为 0，SPI_DATA[7:0]用来发送和接收数据，发送和接收缓冲区都是 8 位。如果数据帧格式为 16 位，SPI_DATA[15:0]用于发送和接收数据，发送和接收缓冲区也是 16 位。

19.5.5. CRC 多项式寄存器（SPI_CRCPOLY）

地址偏移：0x10
复位值：0x0000 0007

该寄存器可以按字节（8位）或半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRCPOLY[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CRCPOLY[15:0]	CRC 多项式寄存器值 该值包含了 CRC 多项式，用于 CRC 计算，默认值为 0007h。

19.5.6. 接收 CRC 寄存器（SPI_RCRC）

地址偏移：0x14
复位值：0x0000 0000

该寄存器可以按字节（8位）或半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RCRC[15:0]															
r															

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	RCRC[15:0]	接收 CRC 寄存器值 当 SPI_CTL0 中的 CRCEN 置位时，硬件计算接收数据的 CRC 值，并保存到 RCRC 寄存器中。如果是 8 位数据帧格式，CRC 计算基于 CRC8 标准进行，保存数据到 RCRC[7:0]。如果是 16 位数据帧格式，CRC 计算基于 CRC16 标准进行，保存数据到 RCRC[15:0]。 硬件在接收到每个数据位后都会计算 CRC 值，当 TRANS 置位时，读该寄存器将返回一个中间值。 当 SPI_CTL0 寄存器中的 CRCEN 位置 1 或 RCU 复位寄存器中的 SPIxRST 位置 1 时，该寄存器复位。

19.5.7. 发送 CRC 寄存器（SPI_TCRC）

地址偏移：0x18
复位值：0x0000 0000

该寄存器可以按字节（8位）或半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCRC[15:0]															
r															

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	TCRC[15:0]	发送 CRC 寄存器值 当 SPI_CTL0 中的 CRCEN 置位时，硬件计算发送数据的 CRC 值，并保存到 TCRC 寄存器中。如果是 8 位数据帧格式，CRC 计算基于 CRC8 标准进行，保存数据到 TCRC[7:0]。如果是 16 位数据帧格式，CRC 计算基于 CRC16 标准进行，保存数据到 TCRC[15:0]。 硬件在发送出每个数据位后都会计算 CRC 值，当 TRANS 置位时，读该寄存器将返回一个中间值。不同的数据帧格式（SPI_CTL0 中的 LF 位决定）将会得到不同的 CRC 值。 当 SPI_CTL0 寄存器中的 CRCEN 位置 1 或 RCU 复位寄存器中的 SPIxRST 位置 1 时，该寄存器复位。

19.5.8. I2S 控制寄存器（SPI_I2SCTL）

地址偏移：0x1C

复位值：0x0000 0000

该寄存器可以按字节（8位）或半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				I2SSEL	I2SEN	I2SOPMOD[1:0]		PCMSMOD	保留	I2SSTD[1:0]		CKPL	DTLEN[1:0]		CHLEN
				rw	rw	rw		rw		rw		rw	rw		rw

位/位域	名称	描述
31:12	保留	必须保持复位值。
11	I2SSEL	I2S 模式选择 0：SPI 模式 1：I2S 模式 当 SPI 模式或 I2S 模式关闭时配置该位。
10	I2SEN	I2S使能 0：禁能I2S 1：使能I2S SPI 模式不使用该位。

9:8	I2SOPMOD[1:0]	I2S 运行模式 00：从机发送模式 01：从机接收模式 10：主机发送模式 11：主机接收模式 当 I2S 模式关闭时配置该位。SPI 模式不使用该位。
7	PCMSMOD	PCM 帧同步模式 0：短帧同步 1：长帧同步 只有在 PCM 标准下，该位才有意义。 当 I2S 模式关闭时配置该位。SPI 模式不使用该位。
6	保留	必须保持复位值。
5:4	I2SSTD[1:0]	I2S 标准选择 00：I2S 飞利浦标准 01：MSB 对齐标准 10：LSB 对齐标准 11：PCM 标准 当 I2S 模式关闭时配置该位。SPI 模式不使用该位。
3	CKPL	空闲状态时钟极性 0：I2S_CK 空闲状态为低电平 1：I2S_CK 空闲状态为高电平 当 I2S 模式关闭时配置该位。SPI 模式不使用该位。
2:1	DTLEN[1:0]	数据长度 00：16 位 01：24 位 10：32 位 11：保留 当 I2S 模式关闭时配置该位。SPI 模式不使用该位。
0	CHLEN	通道长度 0：16 位 1：32 位 通道长度必须大于或等于数据长度。 当 I2S 模式关闭时配置该位。SPI 模式不使用该位。

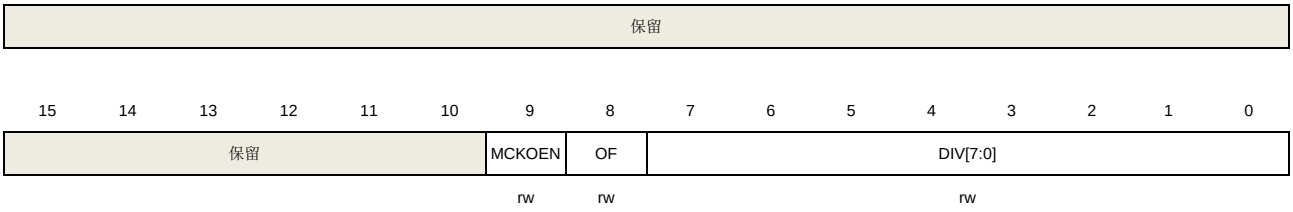
19.5.9. I2S 时钟预分频寄存器（SPI_I2SPSC）

地址偏移：0x20

复位值：0x0000 0002

该寄存器可以按字节（8位）或半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16



位/位域	名称	描述
31:10	保留	必须保持复位值。
9	MCKOEN	I2S_MCK输出使能 0：禁能I2S_MCK输出 1：使能I2S_MCK输出 当I2S关闭时配置该位。 SPI 模式不使用该位。
8	OF	预分频器的奇系数 0：实际分频系数为 DIV * 2 1：实际分频系数为 DIV * 2 + 1 当 I2S 模式关闭时配置该位。SPI 模式下不使用该位。
7:0	DIV[7:0]	预分频器的分频系数 实际分频系数是 DIV * 2 + OF。 DIV 不能为 0。 当 I2S 模式关闭时配置该位。SPI 模式下不使用该位。

20. HDMI-CEC 控制器（HDMI-CEC）

20.1. 简介

GD32F150xx 系列产品内部集成了支持 CEC 协议的 HDMI-CEC 控制器。消费电子控制（CEC）是 HDMI（高清多媒体接口）标准的一部分。CEC 作为一种协议，提供了在用户环境中各种音像制品之间的高级控制功能。用户可以通过 HDMI-CEC 控制器灵活地实现控制功能。

20.2. 主要特性

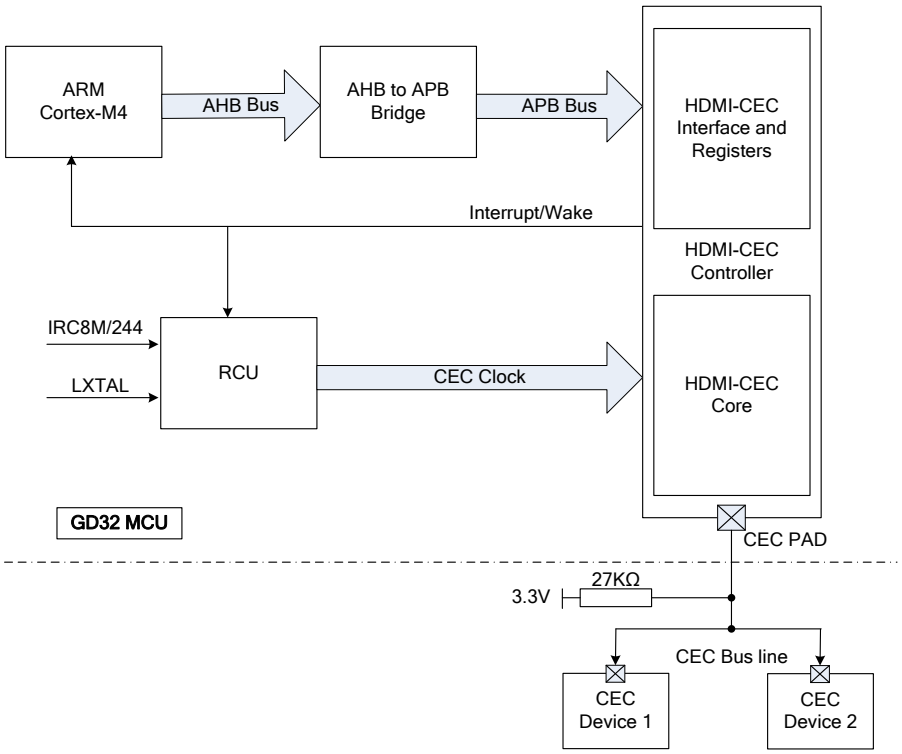
- HDMI-CEC 控制器符合 HDMI-CEC1.4 规范；
- 32.768KHz 的 CEC 时钟具有两个时钟源可供选择：
  - LXTAL 振荡器；
  - 固定分频后的 IRC8M 振荡器（IRC8M/244）；
- 为了超低功耗应用，CEC 控制器可工作在 Deepsleep 模式；
- 可配置信号空闲时间的仲裁优先级；
  - 用户配置；
  - 控制器根据 HDMI-CEC 协议规格自动配置；
- 可编程的私有地址（OAD）；
- 支持监听模式，在不干扰 CEC 总线情况下接收 CEC 总线上的数据；
- 接收位宽容度功能支持更高的兼容性；
- 支持检测多种错误状态的功能
  - 位错误：短位错误（BPSE）、长位错误（BPLE）、位上升沿错误（BRE）；
  - 传输错误（TERR）；
  - 传输欠载（TU）；
  - 接收过载（RO）；
  - 仲裁失败（ARBF）
- 可配置的错误位生成条件；
  - BPSE 检测将总会生成错误位；
  - BPLE 检测只有在 BPLEG=1 时会生成错误位；
  - BRE 检测只有在 BREG =1 时会生成错误位；

20.3. 功能描述

20.3.1. CEC 总线引脚

CEC 控制器用单根双向线来发送和接收数据。当 CEC 设备处于输出的状态，设备之间要实现“线与”的功能连接，则 CEC 的引脚必须配置为漏极开路或集电极开路模式，并且还需外接一个 27KΩ 的上拉电阻到+3.3V 电压。

图 20-1. HDMI-CEC 控制器框图



20.3.2. 信息说明

完整的信息包括一帧或者多帧信息，信息结构如下：

图 20-2. 信息结构

帧有两种类型：

1) **帧头**：信息中紧接着起始位的第一帧，包含信号源的逻辑地址和信号目的地的逻辑地址。帧头必须存在。

2) **数据帧**：帧结构中紧跟着帧头的部分。数据帧是可选的。

所有的帧长度都是 10 位并且有相同的基本结构如下：

表 20-1. 帧结构

帧结构									
7	6	5	4	3	2	1	0		
信息位								ENDOM	ACK

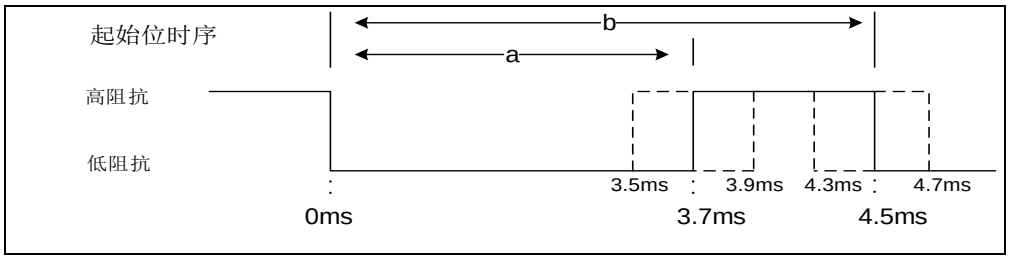
信息位是数据、指令还是地址取决于应用环境。控制位 ENDOM 和 ACK 在一帧中会始终出现，并且所代表的意义相同。

20.3.3. 位时序说明

信息中所有位的时序都被分成两种类型：起始位和数据位

1) 起始位：起始位通过一段低电平持续时间(a)的来表示有效，其总周期(b)如下：

图 20-3. 起始位时序



2) 数据位：有效数据位的时序约束如下：

图 20-4. 数据位时序

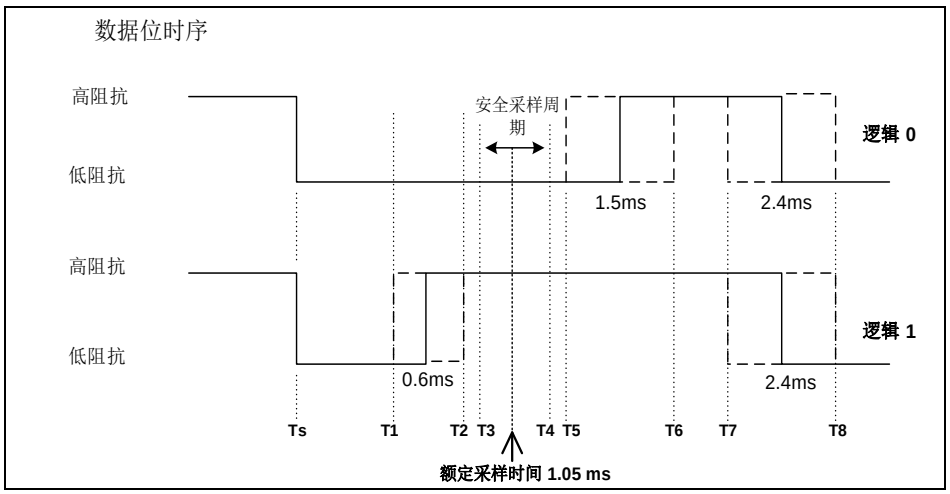


表 20-2. 数据位时序参数表

Ts	时间 (ms)	位起始
T1	0.4ms	传输逻辑 1 时，T1 为从低电平到高电平过渡的最早时间
T2	0.8ms	传输逻辑 1 时，T2 为从低电平到高电平过渡的最晚时间
T3	0.85ms	可以对信号线安全采样的最早时间
T4	1.25ms	可以对信号线安全采样的最晚时间
T5	1.3ms	T5 为逻辑 0 时序中允许设备返回高阻态的最早时间
T6	1.7ms	T6 为逻辑 0 时序中允许设备返回高阻态的最晚时间
T7	2.05ms	T7 为下一位开始的最早时间
	2.4ms	额定数据位周期总长
T8	2.75ms	T8 为下一位开始的最晚时间

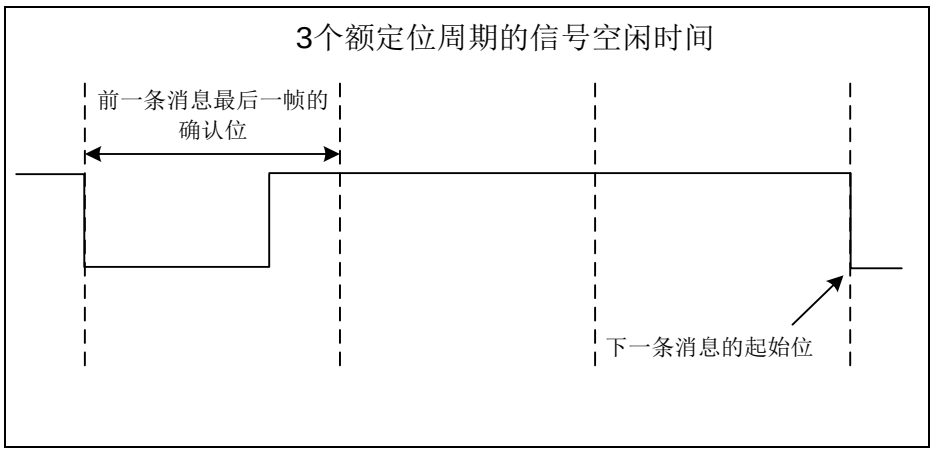
20.3.4. 仲裁

CEC 线仲裁阶段起始于起始位的前沿，结束于启动器地址位的末端。在此期间，启动器监视 CEC 线。如果此时它监测到非自身驱动的低阻态，那么就认为失去了仲裁。

图 20-5. CEC 线仲裁过程

在试图传输或者再次传输一帧之前，CEC 设备应该确保 CEC 总线信号已经空闲了一段时间。这个信号空闲时间的起点为先前帧结束位的开始时刻。

图 20-6. 信号空闲时间



信号空闲时间长度取决于当前控制信号线的状态和设备的初始值。如果 SFT=0x0，HDMI-CEC 控制器的 SFT 将有如下表现：

表 20-3. 信号空闲时间的大小与应用场景的关系

应用场景	信号空闲时间（额定信号位周期）
先前的启动器在发送之前信息后想立即发送另一个信息	≥7
新的启动器想发送一个信息	≥5
先前的信息发送不成功	≥3

这意味着在当前设备完成发送当前信息之后，其他设备有机会在以上空闲时间内进入 CEC 线并发送自己的信息。

如果 SFT 不是 0x0，会执行相应的用户所配置的 SFT 时间。

20.3.5. SFTOPT 位说明

SFT 选项位可通过设置更多的 SFT 计数器的启动时间点来节省总线闲置时间。

当 SFTOPT = 0 时，STAOM 位生效时，SFT 定时器将在控制器处于闲置状态下启动。

当 SFTOPT = 1 时，SFT 定时器将在 CEC 总线处于空闲状态下启动。此时如果在 SFT 结束后配置 STAOM，会节省 SFT 时间，因为控制器启动传输的时候不再需要等待时间。

当 SFTOPT = 1 时，某些事件也可能启动 SFT 计数器：

- 常规 TX/RX 收发结束（TEND/REND 有效）

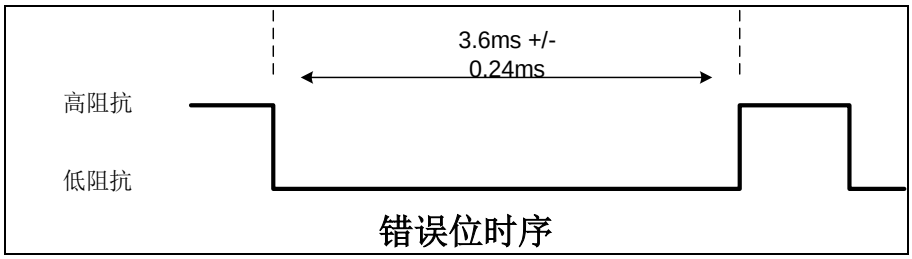
- 传输没有结束例如 TE，TAE 或 TU 有效
- 接收过程中，如果监测到错误且产生错误位， SFT 定时器在输出错误位结束后启动

20.3.6. 错误定义

错误位

如果发生错误且相应产生配置使能，HDMI-CEC 控制器将在 CEC 引脚上产生一个错误位来提示错误。错误位周期定义如下：

图 20-7. 错误位周期



帧错误

CEC 协议规定信息的每一帧需要受到确定信号来确保传输成功。对于广播（目的地址=0xF）来说，ACK 位应该为逻辑 1；对于单播（目的地址<0xF）来说，ACK 位应该为逻辑 0，否则将产生帧错误（TAERR/RAE 标志位生效）

另一个帧错误产生的情况是当 HDMI-CEC 控制器处于启动器阶段时，CEC 总线电平和 CEC 控制器输出电平不同(TERR 会置位)。

位上升错误（BRE）

BRE 位表示 BRE 检查窗口是否检测到上升沿。如果 BREIE=1，则 BRE 置位后产生 CEC 中断。

如果 BRES=1，控制器将停止接受信息， BREG=1，错误位将产生。

如果 BRES=1，在广播时 BRE 标志位会置位，并且会产生错误位来通知启动器有错误发生。如果不想产生监测 BRE 的错误位，可以配置 BREG=0 且 BCNG=1。

注：不可同时配置 BRES=0 和 BREG=1

短位错误（BPSE）

BPSE 位表明相邻的下降沿周期是否需要比预期短。如果 BPSEIE=1，则 BPSE 置位后产生 CEC 中断。

如果 BPSE 错误标志置位，一定会输出错误位，除了以下情况：

1) BCNG = 1
2) LMEN = 1
3) 接收广播

长位错误（BPLE）

BPLE 位表明相邻的下降沿周期是否需要比预期长。如果 BPLEIE =1，则 BPLE 置位后产生 CEC 中断。

BPLE 置位时，如果有以下一种情况出现，控制器将会停止接收信息并产生错误位：

1) 单次播放和广播模式下 BPLEG=1
2) 广播模式下 BCNG=0

图 20-8. 长错误位时序

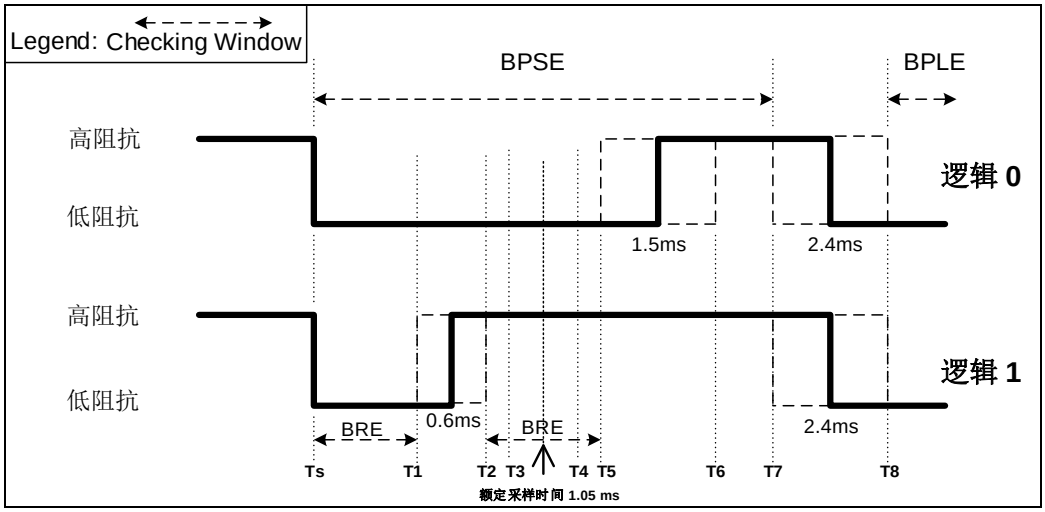


表 20-4. 错误处理时序参数表

标记	RTOL	时长(ms)	说明
Ts	-	0ms	开始传输时刻.
T1	1	0.3ms	传输逻辑 1 时低电平到高电平的转换最早时间
	0	0.4ms	
T2	0	0.8ms	传输逻辑 1 时低电平到高电平的转换最晚时间
	1	0.9ms	
T3	-	0.85ms	对信号线安全采样的最早时间
T4	-	1.25ms	对信号线安全采样的最晚时间
T5	1	1.2ms	逻辑 0 时允许设备返回到高阻态的最早时间
	0	1.3ms	
T6	0	1.7ms	逻辑 0 时允许设备返回到高阻态的最晚时间
	1	1.8ms	
T7	1	1.85ms	下一位开始传输的最早时间
	0	2.05ms	
		2.4ms	额定数据位周期
T8	0	2.75ms	下一位开始传输的最晚时间
	1	2.95ms	

传输错误监测（TERR）

当启动器在 CEC 总线传输高组态的时候监测到总线呈低阻态时，TERR 被置 1。如果 TERRIE=1 时，TERR 也会产生 CEC 中断。

当 TERR 置位时，传输停止，软件可以重启传输。

TERR 检查窗口依赖于帧的不同位状态，显示如下：

图 20-9. 传输错误监测

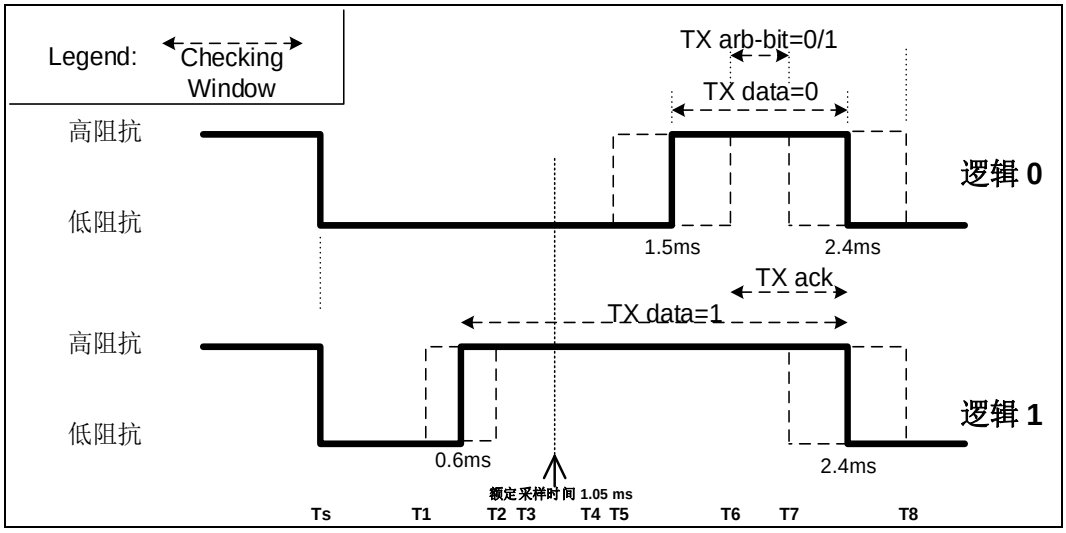


表 20-5. 时序参数表

标记	RTOL	时长(ms)	说明
Ts	-	0ms	开始传输时刻
T1	1	0.3ms	传输逻辑 1 时低电平到高电平的转换最早时间
	0	0.4ms	
T2	0	0.8ms	传输逻辑 1 时低电平到高电平的转换最晚时间
	1	0.9ms	
T3	-	0.85ms	对信号线安全采样的最早时间
T4	-	1.25ms	对信号线安全采样的最晚时间
T5	1	1.2ms	逻辑 0 时允许设备返回到高阻态的最早时间
	0	1.3ms	
T6	0	1.7ms	逻辑 0 时允许设备返回到高阻态的最晚时间
	1	1.8ms	
T7	1	1.85ms	下一位开始传输的最早时间
	0	2.05ms	
		2.4ms	额定数据位周期
T8	0	2.75ms	下一位开始传输的最晚时间
	1	2.95ms	

20.3.7. HDMI-CEC 中断

HDMI-CEC 控制器中有 13 个中断，每个中断由相应的标志位和中断使能位组成。

表 20-6. HDMI-CEC 中断

编号	HDMI-CEC 中断事件	事件标志位	中断使能位
1	仲裁失败	ARBF	ARBFIE
2	传输字节请求	TBR	TBRIE
3	传输结束	TEND	TENDIE
4	传输字节缓冲区欠载	TU	TUIE
5	传输错误	TERR	TERRIE
6	传输确认错误	TAERR	TAERRIE
7	字节接收	BR	BRIE
8	接收结束	REND	RENDIE
9	接收过载	RO	ROIE
10	位上升沿错误	BRE	BREIE
11	短位错误	BPSE	BPSEIE
12	长位错误	BPLE	BPLEIE
13	接收位确认错误	RAE	RAEIE

注: HDMI-CEC 的任何中断都将使芯片从深度睡眠模式唤醒。

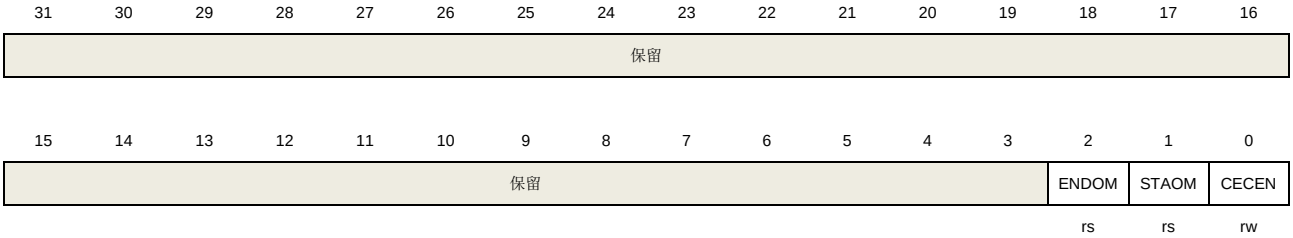
20.4. HDMI-CEC 寄存器

HDMI-CEC 基地址：0x4000 7800

20.4.1. 控制寄存器 (CEC_CTL)

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字（32位）访问。



位/位域	名称	描述
31:3	保留	必须保持复位值
2	ENDOM	发送模式下下一帧的 ENDOM 位值 当 CECEN=1 时 ENDOM 只能被软件写入。ENDOM 和 STAOM 被硬件清 0 的情况相同。 0：下一帧的 ENDOM 位发送 0 1：下一帧的 ENDOM 位发送 1
1	STAOM	启动发送一帧信息 当 CECEN=1 时 STAOM 只能被软件写入。在 TEND、TU、TAERR、TERR 中的任意一位置位或者 CECEN 位清零时，STAOM 位被硬件清零。 如果信息只有一帧，ENDOM 应该在配置 TDATA 之前被置 1。STAOM 被置 1 以后，SFT 计数器启动，当 SFT 计数结束，Start-bit 将输出到 CEC 总线上。软件可以在 STAOM =1 的时候通过清除 CECEN 位来取消发送。 0：无 CEC 传输 1：CEC 传输挂起或者正在进行
0	CECEN	使能/禁止 HDMI-CEC 控制位 CECEN 位由软件配置. 0：禁止 HDMI-CEC 控制器，取消任何信息发送并清除 ENDOM/STAOM 位 1： 使能 CEC 控制器，如果 STAOM=0，进入接收状态

20.4.2. 配置寄存器 (CEC_CFG)

地址偏移：0x04
复位值：0x0000 0000

注意：该寄存器只有在 CECEN=0 的时候才能写入

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LMEN	OAD [14:0]														
rw								rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留							SFTOPT	BCNG	BPLEG	BREG	BRES	RTOL	SFT[2:0]		
							rw	rw	rw	rw	rw	rw	rw		

位/位域	名称	描述
31	LMEN	监听模式使能位 由软件置 1 和清 0 0：只接收 OAD 中的单播信息或广播信息，并返回正确 ACK 1：接收 OAD 的信息时给出正确 ACK；接收不是 OAD 的信息，不发送 ACK
30:16	OAD[14:0]	自身地址 OAD 的每一位代表一个目的地址。例如，如果 OAD[0]=1，控制器将接收发送目的地址为 0x0 的信息。这意味着控制器可以配置为多个私有地址。广播信息始终被接收。接收到目的地址（帧头的后 4 位）以后，如果在 OAD 中声明了，控制器将收到正反馈信息，如果目的地址不在 OAD 中但 LMEN=1，控制器将接收信息但不反馈 ACK。如果在 OAD 中没有声明且 LMEN=0，控制器将不会接收该信息。
15:9	保留	必须保持复位值。
8	SFTOPT	SFT 开始选项位 软件置 1 和清 0 0：STAOM 置 1 后 SFT 计数器开始计数 1：发收/接收结束后 SFT 计数器自动启动
7	BCNG	广播信息模式下不产生错误位 软件置 1 和清 0 0：广播模式下，BRE 和 BPLE 将在 CEC 总线上产生错误位，当 LMEN =1 时 BPSE 也将产生错误位 1：以上条件下不产生错误位
6	BPLEG	在单播模式下监测到 BPLE 的时候产生错误位 软件置 1 和清 0 0：在单播模式下监测到 BPLE 时不产生错误位 1：在单播模式下监测到 BPLE 时产生错误位
5	BREG	在单次传播模式下监测到 BRE 的时候产生错误位 软件置 1 和清 0 0：在单播模式下监测到 BRE 的时候不产生错误位 1：在单播模式下监测到 BRE 的时候产生错误位
4	BRES	监测到 BRE 时是否停止接收信息 软件置 1 和清 0

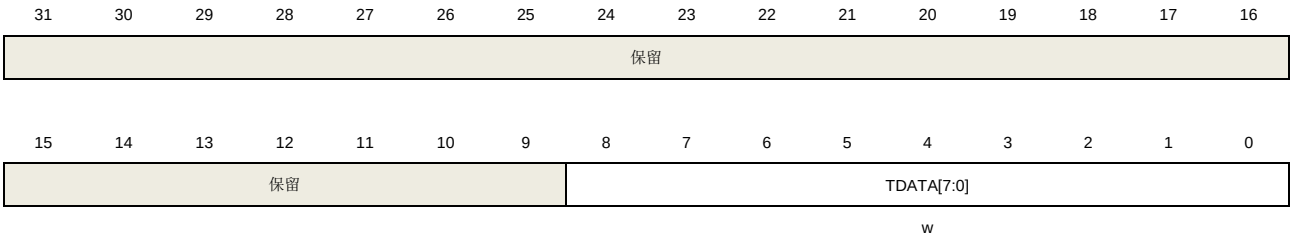
0：不停止接收 BRE 下的信息，数据位按额定时间采样（1.05ms）
1：停止接收 BRE 下的信息

3	RTOL	接收位时间宽容度 软件置 1 和清 0 0：标准接收位时间宽容度 1：扩展接收位时间宽容度
2:0	SFT[2:0]	信号空闲时间 软件置 1 和清 0 如果 SFT=0x0，SFT 时间跟 HDMI-CEC 协议描述的一样，否则，SFT 时间被软件固定配置。ACK 位的下降沿启动计时。 0x0： - 3 个额定数据位周期，如果 SFT 计数器是由于传输不成功而启动（ARBF=1，TERR=1，TU=1 或 TAERR=1） - 5 个额定数据位周期，如果 CEC 控制器是一次新的传输 - 7 个额定数据位周期，如果 CEC 控制器成功完成传输 0x1：1.5 个额定数据位周期 0x2：2.5 个额定数据位周期 0x3：3.5 个额定数据位周期 0x4：4.5 个额定数据位周期 0x5：5.5 个额定数据位周期 0x6：6.5 个额定数据位周期 0x7：7.5 个额定数据位周期

20.4.3. 数据发送寄存器 (CEC_TDATA)

地址偏移：0x08
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

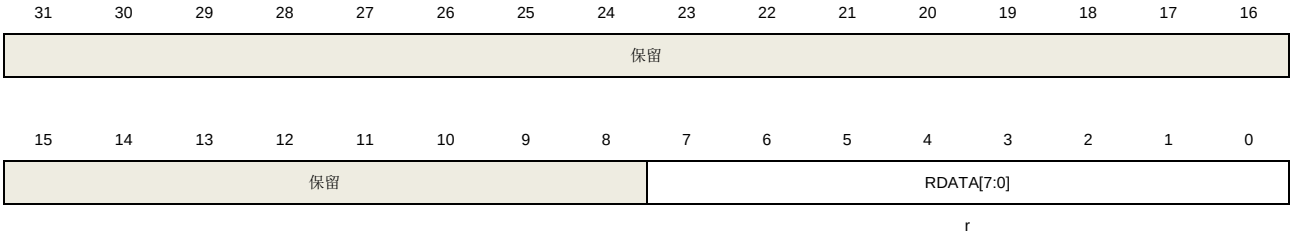


位域	名称	描述
31:8	保留	必须保持复位值。
7:0	TDATA[7:0]	发送数据寄存器 包含将要传输的数据位，只进行写操作。

20.4.4. 数据接收寄存器 (CEC_RDATA)

地址偏移：0x0C
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

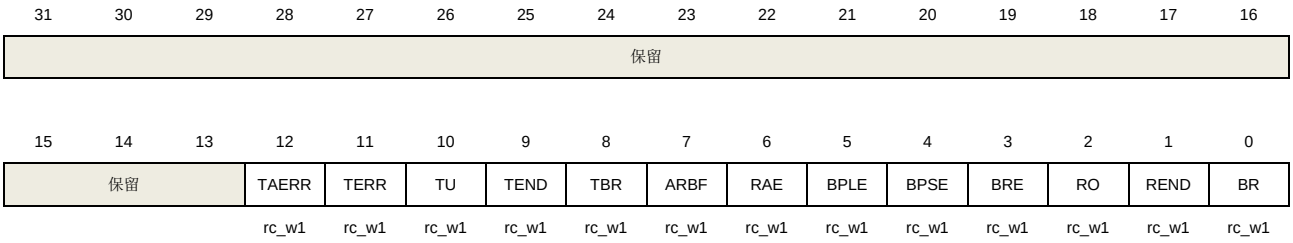


位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	RDATA[7:0]	接收数据寄存器 该位包含从 CEC 线接收的最后数据字节，只能进行读操作。

20.4.5. 中断标志寄存器 (CEC_INTF)

地址偏移：0x10
复位值：0x0000 0000

该寄存器只能按字（32位）访问。



位/位域	名称	描述
31:13	保留	必须保持复位值。
12	TAERR	发送 ACK 错误标志位 硬件置 1，软件写 1 清 0 单播模式下 ACK 位作为 1 被接收及广播模式下作为 0 被接收将使该标志位生效。 TAERR 将停止发送信息并清除 STAOM 和 ENDOM。
11	TERR	发送错误 该位由硬件置 1，软件写 1 清零 如果控制器在启动器阶段下且监测到 CEC 线上非控制器输出的低阻态，TERR 置位。 TERR 将停止发送信息且清除 STAOM 和 ENDOM。

10	TU	发送数据缓冲区欠载 该位由硬件置 1，软件写 1 清零 在发送下一字节之前如果软件没有写数据，TU 生效。TU 将停止发送信息并且清除 STAOM 和 ENDOM。
9	TEND	发送成功结束 该位由硬件置 1，软件写 1 清零 如果信息的所有帧成功传输，TEND 位有效，TEND 将清除 STAOM 和 ENDOM 位。
8	TBR	发送字节数据请求 该位由硬件置 1，软件写 1 清零 当前帧的第四位传输结束后，TBR 位生效，软件需要在 6 个额定数据位周期内写数据到 TDATA。
7	ARBF	仲裁失败 该位由硬件置 1，软件写 1 清零 以下任一情况发生时 ARBF 位生效：外部 CEC 设备在控制器处于 SFT 状态时为了发送起始位拉低 CEC 线，或者控制器和 CEC 设备同时发送起始位但是控制器的启动器地址优先级更低。 如果 ARBF 生效，控制器将进入接收状态，接收信息结束后控制器将重新尝试发送消息。发送和接收期间，STAOM 始终被置 1。
6	RAE	接收 ACK 错误 该位由硬件置 1，软件写 1 清零 以下情况 RAE 生效，在广播模式下如果 ACK=0；或者 LMEN=1 且目的地址不是 OAD 的单播模式下，ACK=1。RAE 将停止接收信息。
5	BPLE	长位周期错误 该位由硬件置 1，软件写 1 清零 数据位周期超过最大时间范围的时候 BPLE 生效。BPLE 将停止接收信息且在以下情况产生错误位：单次传播模式下 BPLEG=1，或广播模式下 BCNG=0。
4	BPSE	短位周期错误 该位由硬件置 1，软件写 1 清零 数据位周期小于最小周期的时候 BPSE 位生效。
3	BRE	位上升错误 该位由硬件置 1，软件写 1 清零 期望时间外产生上升沿时，BRE 生效。
2	RO	接收过载 该位由硬件置 1，软件写 1 清零 接收新的字节且 BR 被置 1 时 RO 仍有效。 RO 将停止接收信息并发送错误 ACK 位。
1	REND	接收结束 该位由硬件置 1，软件写 1 清零 当控制器完整接收到带有成功 ACK 反馈的全部信息的时候 REND 置位。REND 和

BR 同时生效。

0	BR	字节接收 该位由硬件置 1，软件写 1 清零 当控制器接收到带有成功 ACK 反馈的数据的时候 BR 生效，此时 RDATA 有效。

20.4.6. 中断使能寄存器 (CEC_INTEN)

地址偏移：0x14
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留			TAERRIE	TERRIE	TUIE	TXENDIE	TBRIE	ARBFIE	RAEIE	BPLEIE	BPSEIE	BREIE	ROIE	RENDIE	BRIE
			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:13	保留	必须保持复位值。
12	TAERRIE	TAERR 中断使能 软件置 1 和清 0 0：TAERR 中断禁止 1：TAERR 中断允许
11	TERRIE	TERR 中断使能 软件置 1 和清 0 0：TERR 中断禁止 1：TERR 中断允许
10	TUIE	TU 中断使能 软件置 1 和清 0 0：TU 中断禁止 1：TU 中断允许
9	TENDIE	TEND 中断使能 软件置 1 和清 0 0：TEND 中断禁止 1：TEND 中断允许
8	TBRIE	TBR 中断使能 软件置 1 和清 0 0：TBR 中断禁止 1：TBR 中断允许

7	ARBFIE	ARBF 中断使能 软件置 1 和清 0 0：ARBF 中断禁止 1：ARBF 中断允许
6	RAEIE	RAE 中断使能 软件置 1 和清 0 0：RAE 中断禁止 1：RAE 中断允许
5	BPLEIE	BPLE 中断使能 软件置 1 和清 0 0：BPLE 中断禁止 1：BPLE 中断允许
4	BPSEIE	BPSE 中断使能 软件置 1 和清 0 0：BPSE 中断禁止 1：BPSE 中断允许
3	BREIE	BRE 中断使能 软件置 1 和清 0 0：BRE 中断禁止 1：BRE 中断允许
2	ROIE	RO 中断使能 软件置 1 和清 0 0：RO 中断禁止 1：RO 中断允许
1	RENDIE	REND 中断使能 软件置 1 和清 0 0：REND 中断禁止 1：REND 中断允许
0	BRIE	BR 中断使能 软件置 1 和清 0 0：BR 中断禁止 1：BR 中断允许

21. 触摸传感控制器（TSI）

21.1. 简介

触摸传感控制器（TSI）为触摸按键、滑块、电容近距感测等应用提供了简易的解决方案。控制器基于电荷转移方法，当一个手指接近电极时会引起整个系统的电容变化，TSI可以通过电荷转移的方法来检测这种变化，从而感知到手指接近这一行为。

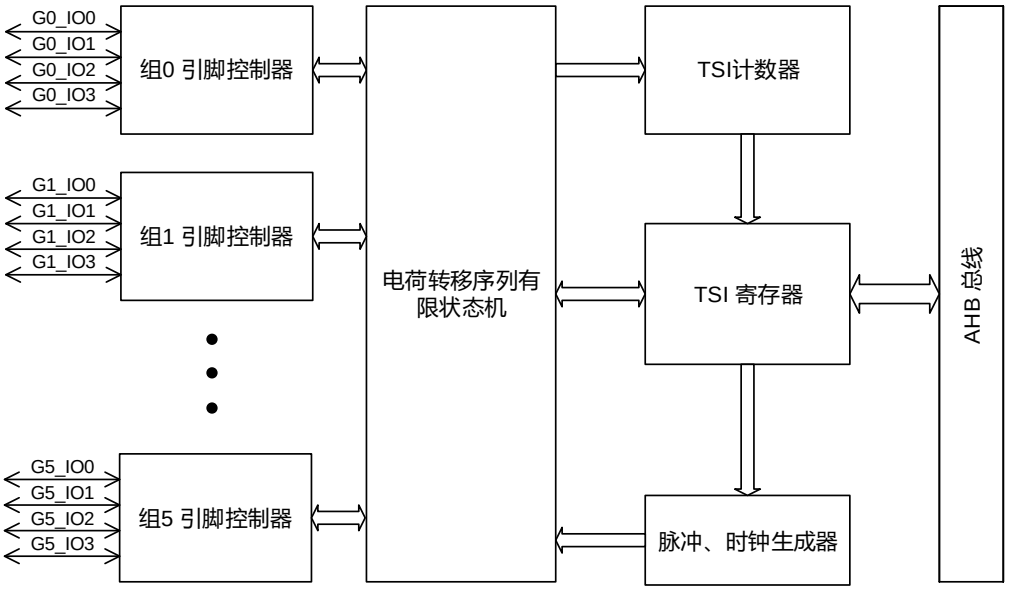
21.2. 主要特性

- 电荷转移序列完全由硬件控制；
- 包含6个完全并行的引脚组；
- 18个引脚可配置为电容感应通道引脚、6个引脚可配置为采样引脚；
- 电荷转移序列的频率可配置；
- 能够实现用户特定电荷转移序列；
- 序列结束标志、错误标志和可配置中断；
- 支持扩频功能。

21.3. 功能描述

21.3.1. TSI 框图

图 21-1. TSI 模块框图

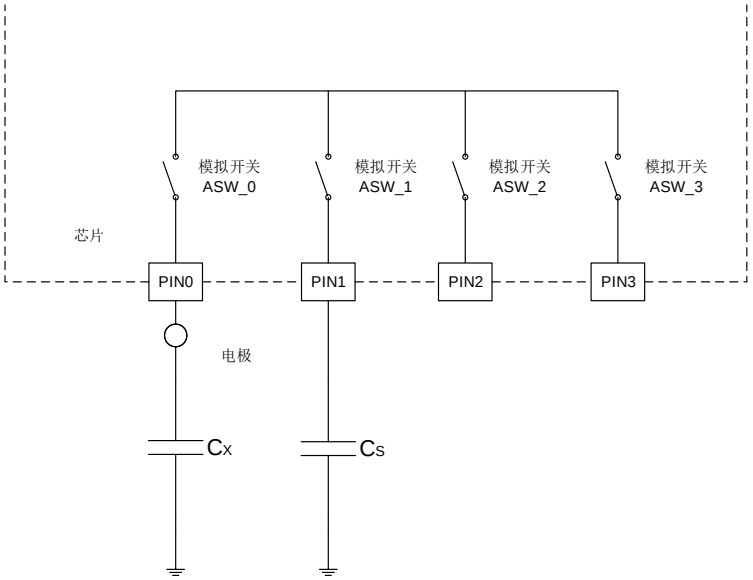


21.3.2. 触摸传感技术概述

触摸感知有多种不同的技术手段，例如光学、电阻式、电容式及应变式等，其中检测系统的变化是这些技术的关键问题和目标。TSI模块使用电荷转移的方法来检测由于触摸或手指靠近带来的系统的变化，尤其是触摸电极上电容的变化。为了检测到这种变化，TSI会运行电荷转移序列，这个序列包括数次充电、转移的步骤直至满足终止条件，这一过程中的充电、转移次数就表征了系统的电容特性，因此应用程序可以通过监测这一数值来感知系统的变化。

如在*图21-2. 通道引脚与采样引脚的框图*中，由4个引脚组成了一个组，4个引脚各自通过一个模拟开关连接到一个公共点上，这些模拟开关是实现电荷转移序列的关键组成部分。每一个引脚组中应该配置一个引脚为采样引脚，其他的3个引脚中应该至少配置一个为通道引脚。例如在*图21-2. 通道引脚与采样引脚的框图*中，PIN0配置为通道引脚，PIN1配置为采样引脚，PIN2和PIN3没有使用。PIN0作为通道引脚，它会连接一个电极，这个电极通常设计在PCB板上，而采样引脚PIN1上需要连接一个采样电容C_s。现在通道引脚PIN0的电容包括C_x和通过电极引入的电容，所以当手指触摸电极时，PIN0的电容会增大而PIN1的电容保持不变。TSI模块会使用电荷转移序列来检测到这一变化，具体实现方法会在下一节中详细说明。

图 21-2. 通道引脚与采样引脚的框图



21.3.3. 电荷转移序列

TSI模块会执行电荷转移序列来测量一个通道引脚的电容变化。整个序列的详细步骤如*表21-1. 电荷转移序列中的引脚和模拟开关状态*所示，其中PINs是按照*图21-2. 通道引脚与采样引脚的框图*来配置的，即PIN0是通道引脚，PIN1是采样引脚。

表 21-1. 电荷转移序列中的引脚和模拟开关状态

步骤序号	名称	ASW_0 状态	ASW_1 状态	PIN0 状态	PIN1 状态
1	放电	闭合	闭合	浮空输入	下拉
2	缓冲时间 1	断开	断开	浮空输入	浮空输入
3	充电	断开	断开	输出高电平	浮空输入
4	充电扩展	断开	断开	输出高电平	浮空输入
5	缓冲时间 2	断开	断开	浮空输入	浮空输入

步骤序号	名称	ASW_0 状态	ASW_1 状态	PIN0 状态	PIN1 状态
6	电荷转移	闭合	闭合	浮空输入	浮空输入
7	缓冲时间 3	断开	断开	浮空输入	浮空输入
8	对比	断开	断开	浮空输入	浮空输入

1. 放电

闭合ASW_0和ASW_1并将PIN0配置为浮空输入且PIN1配置为下拉，使C_x和C_s放电。这个步骤是初始化的操作，应该由软件在转移序列开始之前通过配置寄存器来实现。该步骤的放电时间需要保证C_x和C_s的电压被彻底放电至0。

2. 缓冲时间 1

ASW_0和ASW_1断开并将PIN0与PIN1配置为浮空输入。

3. 充电

ASW_0和ASW_1断开，将PIN0配置为输出高电平，给C_x充电，将PIN1配置为浮空输入。充电时间应该配置为可确保C_x充电到VDD。

4. 充电扩展

这是在电荷转移序列中的一个可选步骤，目的是实现扩频。此步骤中所有引脚和模拟开关的状态与步骤3完全相同，唯一差别是持续时间，该步骤的持续时间在每次循环都会变化，最大值可以TSI寄存器配置。

5. 缓冲时间 2

ASW_0和ASW_1断开并将PIN0与PIN1配置为浮空输入。

6. 电荷转移

ASW_0和ASW_1闭合并将PIN0与PIN1配置为浮空输入，电荷将从C_x向C_s转移。这个转移时间应该被合理配置以确保转移后采样引脚和通道引脚的电压值相等。

7. 缓冲时间 3

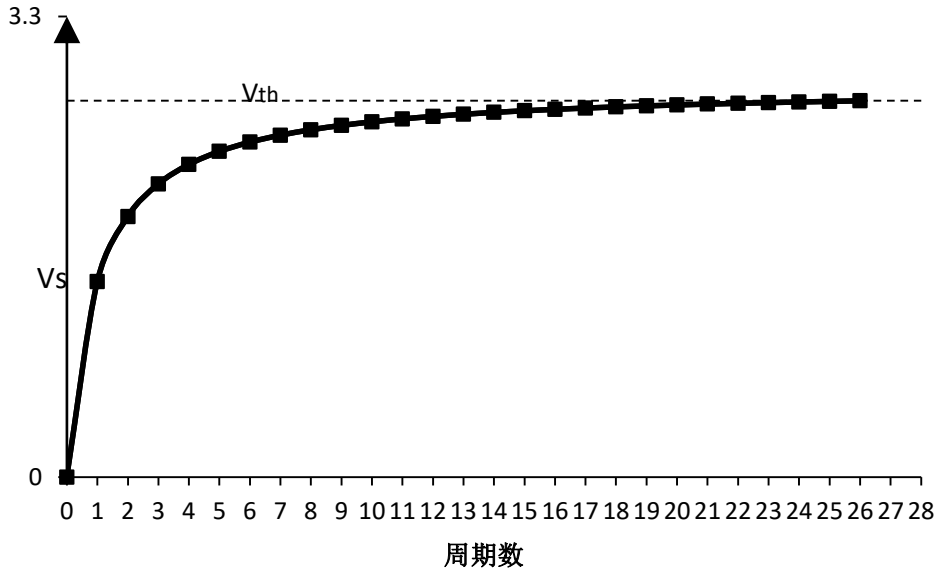
ASW_0和ASW_1断开并将PIN0与PIN1配置为浮空输入。

8. 对比

ASW_0、ASW_1、PIN0与PIN1仍然保留上一步骤的配置。在这一步将采样引脚PIN1的电压V_s和一个门限电压V_{th}进行比较。如果V_s比V_{th}低，那么整个序列返回到步骤2开始新的循环，否则整个序列结束。

在初始化步骤1完成之后，采样引脚的电压V_s是0，之后每经过一个充电、转移周期都会增大一些，如*图21-3. 电荷序列转移期间的采样引脚的电压*所示。根据电气学知识，如果电极上的电容越大，则每个周期V_s增大的值就会越大。电荷转移序列的停止条件是V_s达到V_{th}。每个序列都有一个计数器用来记录达到V_{th}的周期次数。在电荷转移序列的结束，序列计数器会被读出用于评估C_x的电容，例如，一个比较小的计数器值代表一个比较大的C_x电容值。

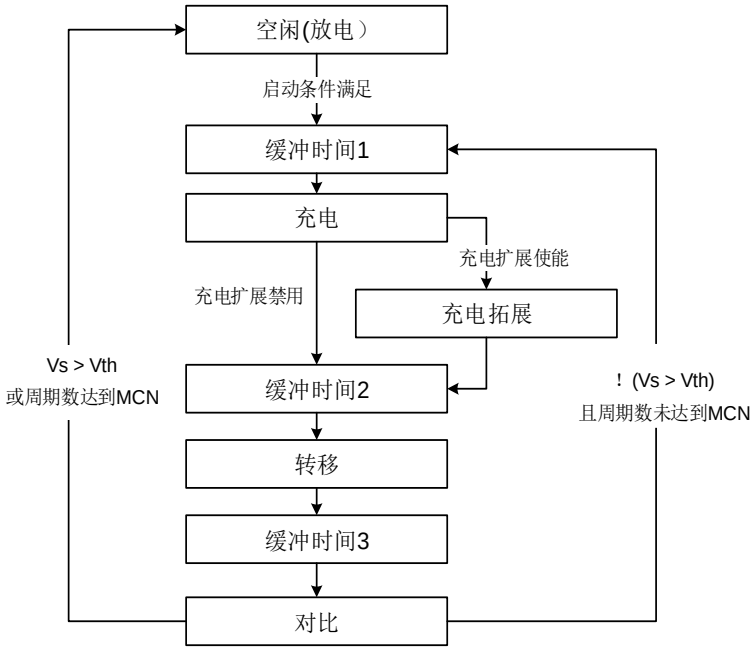
图 21-3. 电荷序列转移期间的采样引脚的电压



21.3.4. 电荷转移序列状态机

TSI内部有一个硬件实现的有限状态机，来执行前一节中描述的电荷转移序列，如*图21-4. 电荷转移序列的有限状态机的状态转移图*所示。

图 21-4. 电荷转移序列的有限状态机的状态转移图



复位之后，状态机处于默认的IDLE状态。状态机有2种启动条件（由TSI_CTL寄存器的TRGMOD位定义）：

TRGMOD = 0：软件触发模式。在这一模式下，软件通过将TSI_CTL寄存器的TSIS位写1来启动状态机。

TRGMOD = 1：硬件触发模式。在这种模式下，状态机会监测TSITG引脚，当发现上升沿或下降沿时自动启动。

状态机一旦启动就会按照上图所描述的跳转流程来运行。状态机如果在当前状态下的持续时间一旦达到配置值就会离开当前状态，并根据当前的条件进入下一个状态。

只有当TSI_CTL寄存器中的ECEN位被置1，才会出现充电扩展这一状态，这个状态用于实现扩频功能，该状态会根据当前FSM循环次数扩展脉冲高电平持续时间。换言之，充电频率将变得动态且不固定。在有干扰的应用环境下，启用该功能能够提高TSI的鲁棒性，同时系统的电磁干扰将会降低。

在比较状态，该状态机会比较每个已启用的引脚组中的采样引脚电压和阈值电压。如果所有采样引脚的电压都达到了阈值，状态机返回空闲状态并停止，否则，状态机返回到缓冲时间1状态，继续下一个周期。如*图21-3. 电荷序列转移期间的采样引脚的电压*所示，在27个周期之后，V_s（采样引脚的电压）到达V_{th}（阈值电压）。

在TSI_CTL寄存器的MCN位也定义了一个最大周期数，当周期数达到MCN，状态机将会在比较状态之后返回到IDLE状态并停止，不管此时V_s是否达到了V_{th}。

21.3.5. 状态时钟和持续时间

TSI模块里有3个时钟：HCLK、CTCLK和ECCLK。HCLK为系统时钟，它驱动TSI的寄存器和状态机。CTCLK是从HCLK分频得到的（分频系数由CTCDIV定义），它是用来计算充电状态和电荷转移状态持续时间的时钟。ECCLK也是从HCLK分频得到（分频系数由ECCDIV定义），它用来计算扩展充电状态的最大持续时间。ECCLK和CTCLK是相互独立的。

除扩展充电状态外，其他状态的持续时间在每一次循环周期中都是固定的（根据寄存器配置）。

缓冲时间1、缓冲时间2和缓冲时间3的持续时间固定为2个HCLK周期。充电状态、电荷转移状态的持续时间由CDT与CTDT寄存器定义（详细见TSI_CTL寄存器部分）。

通常，扩展充电频率的变化范围限制在10%到50%之间。在电荷转移状态机的每个周期中，扩展充电状态的最大持续时间由TSI_CTL寄存器的ECDT[6:0]控制位定义。如果扩展充电状态被使能，最长的充电时间是当周期数为ECDT+2时。扩展充电状态在每个周期的持续时间变化规律如*表21-2. 充电扩展状态的持续时间*所示：

表 21-2. 充电扩展状态的持续时间

周期数	充电扩展状态持续时间
1	0
2	1
…	
ECDT	ECDT-1
ECDT+1	ECDT
ECDT+2	ECDT+1
ECDT+3	ECDT
ECDT+4	ECDT-1
…	…
2*ECDT+1	2
2*ECDT+2	1
2*ECDT+3	0

周期数	充电扩展状态持续时间
2*ECDT+4	1
2*ECDT+5	2
…	…

21.3.6. 引脚模式和 TSI 控制

每一个引脚组包括4个引脚，任意一个引脚都能够被用作采样引脚或通道引脚。但是在一个引脚组内，只有一个引脚应配置为采样引脚，通道引脚可以是多于一个，但任何情况下采样引脚和通道引脚都不能配置为同一个引脚。

一旦在GPIO中将一个引脚配置为TSI引脚，并在TSI中将其配置为采样或通道引脚，则在电荷转移序列中，该引脚的模式就由TSI控制。一般情况下，每个引脚有3种模式：输入、输出高和输出低。

一个通道引脚或采样引脚在电荷转移序列期间的模式已在*表21-1. 电荷转移序列中的引脚和模拟开关状态*中具体描述，其中，PIN0代表一个通道引脚，PIN1代表一个采样引脚，即当序列在进行时，电荷转移状态机控制这些通道和采样引脚的模式以及相关模拟开关的状态。当序列处于空闲状态时，TSI_CTL寄存器中的PINMOD控制位定义了这些引脚的模式。此外，如果一个引脚在GPIO中配置为TSI引脚，但在TSI模块配置中它既不是采样引脚，也不是通道引脚，我们称之为自由引脚，该引脚的模式也由TSI_CTL寄存器的PINMOD位定义。

21.3.7. 模拟开关和 I/O 迟滞模式

当状态机正在运行时，一个通道或采样引脚的模拟开关由电荷转移序列控制，如*表21-1. 电荷转移序列中的引脚和模拟开关状态*所示。当状态机处于空闲状态时，这些引脚的模拟开关由TSI_ASW寄存器中对应的GxPy位控制，此外所有的自由引脚的模拟开关也由GxPy位控制。

TSI模块会始终控制引脚的模拟开关，即使这些引脚在GPIO中没有配置成TSI引脚。用户可以利用这个特性，通过软件读写GxPy位来控制这些模拟开关，同时通过GPIO方式直接配置引脚的输出/输入模式，这样就可以实现用户自定义的电荷转移序列。

TSI控制器具有GPIO最高的优先级，当TSI被使能，该配置变为可用，而不受GPIO寄存器、模式或其他外设的控制。

通过清除TSI_PHM寄存器中的GxPy控制位可以禁用TSI引脚的施密特触发器迟滞模式，该操作能够提高系统的抗干扰性。

21.3.8. TSI 操作流

TSI的正常软件运行流程如下所示：

1. 系统初始化，如系统时钟配置，TSI相关的GPIO配置等。
2. 按要求编程TSI_CTL，TSI_INTEN，TSI_CHCFG，TSI_SAMPCFG和TSI_GCTL寄存器。
3. 通过设置TSI_CTL寄存器的TSIEN位使能TSI。
4. 通过设置TSI_INTC寄存器清除CCTCF和CMNERR中断标志位。

5. 如果为软件触发模式（TRGMOD = 0），通过置位TSIS位以启动充电转移序列。如果为硬件触发模式（TRGMOD = 1），充电转移序列由触发引脚的下降沿/上升沿启动。

6. 等待TSI_INTF寄存器中的CTCF或MNERR标志位置位，并且通过置位TSI_INTCC寄存器中的CCTCF或CMNERR位清除这些标志位。

7. 读出TSI_GxCYCN寄存器的CYCN位。

21.3.9. TSI 标志和中断

表 21-3. TSI 错误和标志位

标识名称	描述	清零
CTCF	当所有使能引脚组的采样电压V_s都达到了阈值电压V_{th}时，TSI 停止	置位 TSI_INTC 寄存器的 CCTCF 位
MNERR	当周期数到达最大值时，TSI 停止	置位 TSI_INTC 寄存器的 CMNERR 位

21.3.10. TSI GPIOs

表 21-4. TSI 引脚

TSI 组	TSI 引脚	GPIO 引脚
引脚组 0	PIN0	PA0
	PIN1	PA1
	PIN2	PA2
	PIN3	PA3
引脚组 1	PIN0	PA4
	PIN1	PA5
	PIN2	PA6
	PIN3	PA7
引脚组 2	PIN0	PC5
	PIN1	PB0
	PIN2	PB1
	PIN3	PB2
引脚组 3	PIN0	PA9
	PIN1	PA10
	PIN2	PA11
	PIN3	PA12
引脚组 4	PIN0	PB3
	PIN1	PB4
	PIN2	PB6
	PIN3	PB7
引脚组 5	PIN0	PB11
	PIN1	PB12
	PIN2	PB13

TSI 组	TSI 引脚	GPIO 引脚
	PIN3	PB14

21.4. TSI 寄存器

TSI基地址：0x4002 4000

21.4.1. 控制寄存器（TSI_CTL）

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CDT[3:0]				CTDT[3:0]				ECDT[6:0]							ECEN
rw				rw				rw							rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECDIV	CTCDIV[2:0]			保留				MCN[2:0]			PINMOD	EGSEL	RGMOD	TSIS	TSIEN
rw	rw							rw			rw	rw	rw	rw	rw

位/位域	名称	描述
31:28	CDT[3:0]	充电状态持续时间 这些位由软件置位和清除。这些位控制在电荷转移序列中充电状态的持续时间。 0000：1×t_{CTCLK} 0001：2×t_{CTCLK} 0010：3×t_{CTCLK} … 1111：16×t_{CTCLK}
27:24	CTDT[3:0]	电荷转移状态持续时间 这些位由软件置位和清除。这些位控制在电荷转移序列中电荷转移状态的持续时间。 0000：1×t_{CTCLK} 0001：2×t_{CTCLK} 0010：3×t_{CTCLK} … 1111：16×t_{CTCLK}
23:17	ECDT[6:0]	扩展充电状态最大持续时间 这些位由软件置位和清除。这些位控制在电荷转移序列中充电扩展状态的最大持续时间。 0000000：1×t_{ECCLK} 0000001：2×t_{ECCLK} 0000010：3×t_{ECCLK} … 1111111：128×t_{ECCLK} **注意**：仅当 TSI_CTL 寄存器的 ECEN 位置 1 时才有充电扩展状态出现。
16	ECEN	扩展充电状态使能

		0：禁能扩展充电 1：使能扩展充电
15	ECDIV	ECCLK 时钟分频系数 ECCLK 由 HCLK 分频得到，ECDIV 定义了分频系数。 000：$f_{ECCLK}=f_{HCLK}$ 001：$f_{ECCLK}=f_{HCLK}/2$
14:12	CTCDIV[2:0]	CTCLK 时钟分频系数 CTCLK 由 HCLK 分频得到，CTCDIV 定义了该分频系数。 0000: $f_{CTCLK}=f_{HCLK}$ 0001: $f_{CTCLK}=f_{HCLK}/2$ 0010: $f_{CTCLK}=f_{HCLK}/4$ 0011: $f_{CTCLK}=f_{HCLK}/8$ … 0111: $f_{CTCLK}=f_{HCLK}/128$
11:8	保留	必须保持复位值。
7:5	MCN[2:0]	一个序列的最大周期数 这些位定义电荷转移序列的最大充电、转移周期数，一旦达到这个值，序列将会停止。 000：255 001：511 010：1023 011：2047 100：4096 101：8191 110：16383 111：保留
4	PINMOD	引脚模式 该位定义在状态机为空闲状态时 TSI 引脚的模式。 0：TSI 引脚输出低电平 1：TSI 引脚保持浮空输入模式
3	EGSEL	边沿类型选择 该位定义硬件触发模式下的边沿类型。 0：下降沿触发 1：上升沿触发
2	TRGMOD	触发模式选择 0：软件触发模式，当 TSIS 位置位后，序列将启动 1：硬件触发模式，在触发引脚检测到一个上升沿或下降沿时，序列将启动
1	TSIS	TSI 启动 在软件触发模式下，由软件置 1，用于启动一次电荷转移序列，序列结束后硬件自动清零。置位该位后，软件也可以手动清零来停止已启动的序列。 0：TSI 不启动

		1：TSI 启动
0	TSIEN	TSI 使能 0：TSI 模块禁能 1：TSI 模块使能

21.4.2. 中断使能寄存器（TSI_INTEN）

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留														MNERRIE	CTCFIE
														rw	rw

位/位域	名称	描述
31:2	保留	必须保持复位值。
1	MNERRIE	最大循环次数错误中断使能 0：MNERR 中断禁用 1：MNERR 中断使能
0	CTCFIE	电荷转移完成标志中断使能 0：CTCF 中断禁用 1：CTCF 中断使能

21.4.3. 中断标志位清除寄存器（TSI_INTC）

地址偏移：0x08
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留														CMNERR	CCTCF
														w	w

位/位域	名称	描述

31:2	保留	必须保持复位值。
1	CMNERR	清除最大循环次数错误 0：保留 1：清 MNERR
0	CCTCF	清除电荷转移完成标志 0：保留 1：清 CTCF

21.4.4. 中断标志位寄存器（TSI_INTF）

地址偏移：0x0C
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留														MNERR	CTCF
														r	r

位/位域	名称	描述
31:2	保留	必须保持复位值。
1	MNERR	最大循环次数错误标志 因为到达 MCN[2:0]定义的最大周期数，在电荷转移序列停止后，该位由硬件置位。软件可通过向 TSI_INTC 寄存器的 CMNERR 位写 1 来清零。 0：没有最大循环次数错误 1：最大循环次数错误
0	CTCF	电荷转移完成标志 当所有使能组的采样引脚电压达到电压阈值或采样引脚组周期数到达 MCN[2:0]定义的最大周期数，在电荷转移序列停止后，该位由硬件置位。软件可通过向 TSI_INTC 寄存器的 CCTCF 位写 1 来清零。 0：电荷转移没有完成 1：电荷转移完成

21.4.5. 引脚迟滞模式寄存器（TSI_PHM）

地址偏移：0x10
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

保留								G5P3	G5P2	G5P1	G5P0	G4P3	G4P2	G4P1	G4P0
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
G3P3	G3P2	G3P1	G3P0	G2P3	G2P2	G2P1	G2P0	G1P3	G1P2	G1P1	G1P0	G0P3	G0P2	G0P1	G0P0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:24	保留	必须保持复位值。
23:0	GxPy	引脚迟滞模式 该位由软件置位和清除。 0：GxPy 引脚施密特触发迟滞模式禁能 1：GxPy 引脚施密特触发迟滞模式使能

21.4.6. 模拟开关寄存器（TSI_ASW）

地址偏移：0x18
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								G5P3	G5P2	G5P1	G5P0	G4P3	G4P2	G4P1	G4P0
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
G3P3	G3P2	G3P1	G3P0	G2P3	G2P2	G2P1	G2P0	G1P3	G1P2	G1P1	G1P0	G0P3	G0P2	G0P1	G0P0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:24	保留	必须保持复位值。
23:0	GxPy	模拟开关状态 该位由软件置位和清除。 0：GxPy 的模拟开关断开 1：GxPy 的模拟开关闭合

21.4.7. 采样配置寄存器（TSI_SAMPCFG）

地址偏移：0x20
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								G5P3	G5P2	G5P1	G5P0	G4P3	G4P2	G4P1	G4P0
								rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
G3P3	G3P2	G3P1	G3P0	G2P3	G2P2	G2P1	G2P0	G1P3	G1P2	G1P1	G1P0	G0P3	G0P2	G0P1	G0P0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:24	保留	必须保持复位值。
23:0	GxPy	采样引脚模式 该位由软件置位和清除。 0：GxPy 引脚不是采样引脚 1：GxPy 引脚是采样引脚

21.4.8. 通道配置寄存器（TSI_CHCFG）

地址偏移：0x28
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								G5P3	G5P2	G5P1	G5P0	G4P3	G4P2	G4P1	G4P0
								rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
G3P3	G3P2	G3P1	G3P0	G2P3	G2P2	G2P1	G2P0	G1P3	G1P2	G1P1	G1P0	G0P3	G0P2	G0P1	G0P0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:24	保留	必须保持复位值。
23:0	GxPy	通道引脚模式 该位由软件置位和清除。 0：GxPy 引脚不是通道引脚 1：GxPy 引脚是通道引脚

21.4.9. 组控制寄存器（TSI_GCTL）

地址偏移：0x30
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留										GC5	GC4	GC3	GC2	GC1	GC0
										r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留										GE5	GE4	GE3	GE2	GE1	GE0
										rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:24	保留	必须保持复位值。
21:16	GCx	组完成 当一个使能组的电荷转移序列完成的时候，该位由硬件置位。当一个新的电荷转移序列开始的时候，由硬件清 0。 0：组 x 电荷转移没有完成 1：组 x 电荷转移完成
15:6	保留	必须保持复位值
5:0	GEx	组使能 该位由软件置位和清除。 0：组 x 禁能 1：组 x 使能

21.4.10. 组 x 周期数寄存器（TSI_GxCYCN）（x = 0…5）

地址偏移：0x34 + 0x04 * x

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		CYCN[13:0]													
		rw													

位/位域	名称	描述
31:14	保留	必须保持复位值。
13:0	CYCN[13:0]	周期数目 这些位反映了电荷转移序列完成时组 x 执行的周期数。当一个新的电荷转移序列开始的时候，由硬件清 0。

22. 通用串行总线全速设备接口（USBD）

USBD仅仅适用于GD32F150系列芯片。

22.1. 简介

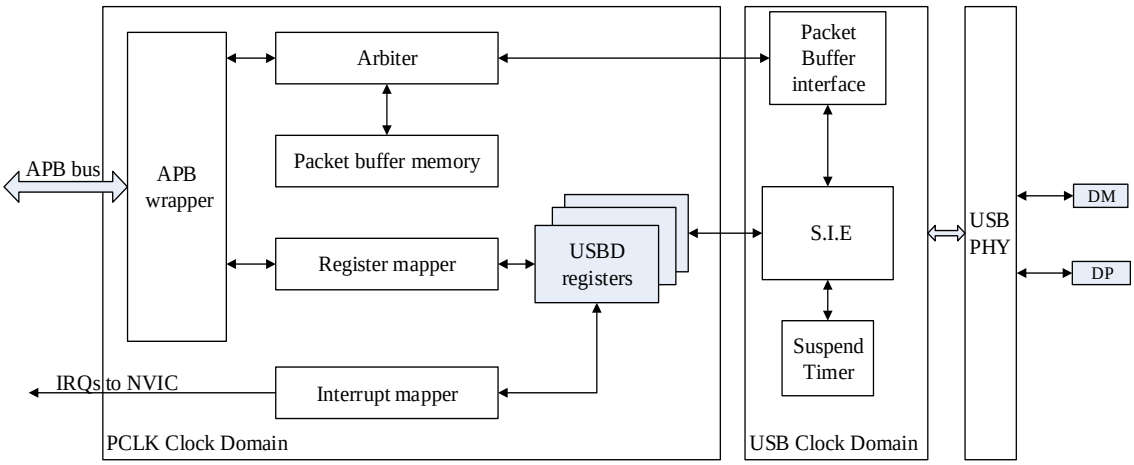
通用串行总线全速设备接口（USBD）模块提供了一个实现符合USB 2.0全速协议外设的方案。它内部包含了一个USB物理层而不需要额外的外部物理层芯片。USBD支持USB 2.0协议所定义的四种传输类型（控制、批量、中断和同步传输）。

22.2. 主要特性

- USB 2.0全速设备控制器；
- 最多支持8个可配置的端点；
- 支持双缓冲的批量传输端点/同步传输端点；
- 每个端点都支持控制，批量，同步或中断传输（端点0除外，端点0只支持控制传输）；
- 支持USB挂起/恢复操作；
- 拥有用于数据缓冲的512字节的专用SRAM；
- 集成的USB物理层。

22.3. 模块图

图 22-1. USBD 模块图



22.4. 信号描述

表 22-1. USBD 信号描述

输入/输出端口	类型	描述
DM	输入/输出	差分数据线 D-

输入/输出端口	类型	描述
DP	输入/输出	差分数据线 D+

注意：一旦USBD被使能，这些引脚会自动连到USBD内部收发器上。

22.5. 时钟配置

根据USB标准定义，USB全速模块采用了固定的48MHz时钟。要使用USBD，需要打开两个时钟，一个是USB控制器时钟，它的频率必须配到48MHz，另一个是APB1到USB接口时钟，它也是APB1的总线时钟，其频率可以高于也可以低于48MHz。

注意：为了满足USB数据传输率和分组缓冲区接口的系统需求，APB1总线时钟的频率必须大于24MHz，以避免数据缓冲区的上下溢出。

USB控制器的48MHz时钟可以通过MCU的内部晶振或外部晶振分频后再经过PLL倍频得到：

- 8MHz的内部晶振2分频后作为PLL的输入，再进行12倍频得到
- 8MHz的外部晶振直接作为PLL的输入，先进行倍频，再经过USB分频器分频得到

当通过外部晶振产生USB时钟时，需要注意的是USB分频系数只有4个值：1分频、1.5分频、2分频和2.5分频。所以，为了获得48MHz的时钟，PLL倍频后可以为48MHz、72MHz。

注意：无论使用外部晶振还是内部晶振产生的USB时钟，其时钟准确度都必须要达到±500ppm。如果USB时钟的准确度下降，传输数据可能无法满足USB规范要求，甚至直接导致USB无法运行。

22.6. 功能说明

22.6.1. USB 端点

USBD支持8个可以独立配置的USB端点。

每个端点支持：

- 单或双缓冲（端点0不能使用双缓冲）；
- 一个端点缓冲区描述符；
- 可编程的缓冲区起始地址与长度；
- 对于数据包的响应可配置；
- 控制传输（仅端点0）。

端点缓冲区

设备操作的功能就是将存储映像中的请求发到USB总线上或是从USB总线上接收请求存储到存储映像中。为了有效地管理USB端点通信，USBD实现了一个可被USB外设直接访问的512字节专用SRAM数据包缓冲区。它被映射到APB1外设存储区，从地址0x4000 6000到0x4000 6400。总容量为1KB，但是由于总线宽度原因USBD实际只使用了512字节。

每个端点都有一个或者两个用于存储当前数据负载的数据包缓冲区。双向端点通常有两个缓

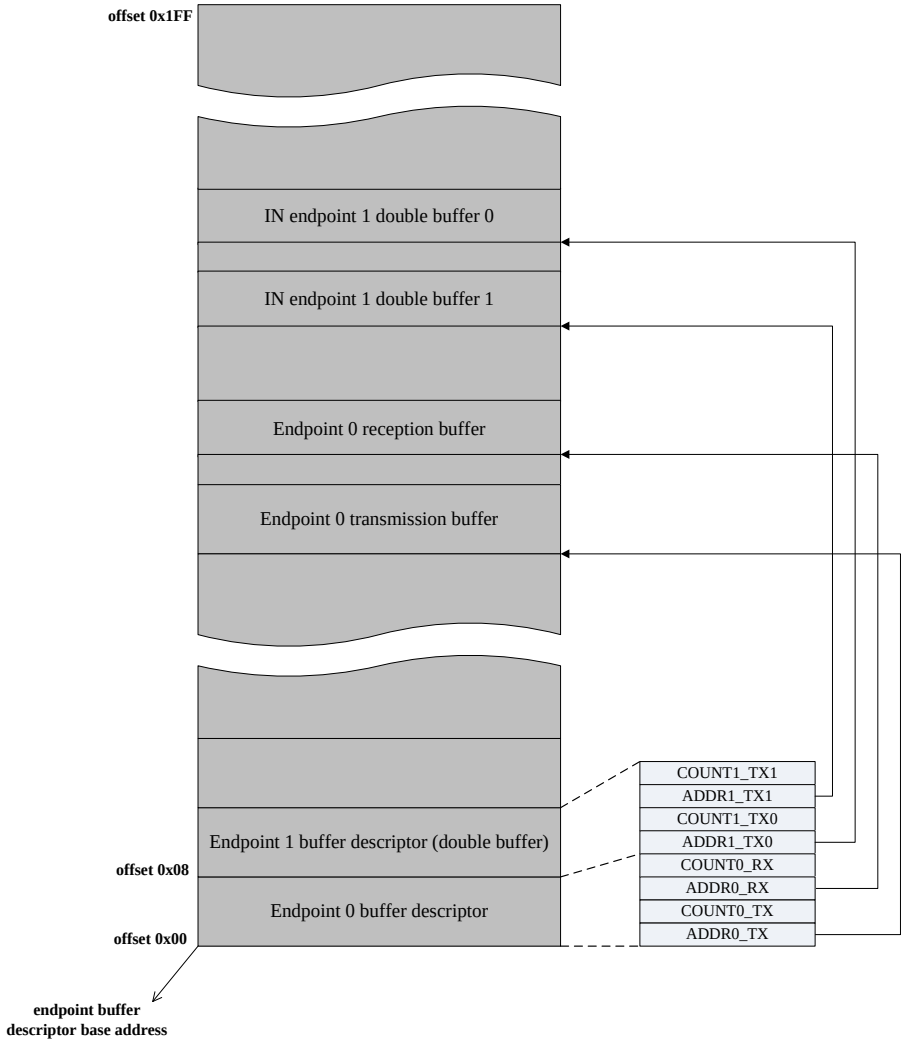
冲区，一个用于数据发送，另一个用于数据接收。单向端点只用一个缓冲区用于数据操作。

端点缓冲器描述符表

USBD实现了一个定义了端点的缓冲区地址与长度的端点缓冲区描述符表，其也位于端点数据包缓冲区中。相应的端点缓冲区描述符作为了一个可以在应用程序固件与系统存储器中SIE之间的通讯端口。由于每个端点方向要求两个16位字的缓冲区描述。因此，每个表条目包含了4个16位字（发送与接收两个方向）并按照8字节对齐。当一个端点为双缓冲端点，SIE将以ping-pong的方式使用这两个缓冲区。USBD端点缓冲地址寄存器指向端点缓冲区描述表。

*图22-2. 缓冲描述符表的用法示例 (USBD_BADDR = 0)*描述了端点缓冲区描述表与数据包缓冲区之间的关系

图 22-2. 缓冲描述符表的用法示例 (USBD_BADDR = 0)



注意：此图未按照实际大小，而以USB总线16位模式来定址。

双缓冲端点

双缓冲特性是为了提高批量传输的性能。为了实现新的流控方案，USBD应该要知道哪一个数据包缓冲区正在被应用软件使用，从而避免发生冲突。既然在USBD_EPxCS寄存器中有两个

数据翻转位，而USBD只使用一位来进行硬件数据处理（由于双缓冲功能所需的单向约束），那么，应用程序可以使用另外一位来表明当前正在使用哪个缓冲区。这个新的缓冲区标志位称作SW_BUF。*表22-2. 双缓冲标志定义*解释了USBD_EPxCS寄存器位与DTG/SW_BUF定义之间的对应关系。

表 22-2. 双缓冲标志定义

缓冲标志	发送端点	接收端点
DTG	TX_DTG (USBD_EPxCS 第 6 位)	RX_DTG (USBD_EPxCS 第 14 位)
SW_BUF	RX_DTG (USBD_EPxCS 第 14 位)	TX_DTG (USBD_EPxCS 第 6 位)

DTG位和SW_BUF位负责流控。当一个传输完成的时候，USB外设翻转DTG位；当数据被复制后，应用软件需要翻转SW_BUF位。除了首次传输，如果DTG位的值等于SW_BUF位的值，传输将会暂停，并且向主机发送NAK数据包。当这两位不相等的时候，传输会继续。

表 22-3. 双缓冲的用法

端点类型	DTG	SW_BUF	USBD 使用的数据包缓冲	应用程序使用的数据包缓冲
OUT	0	1	EPxTBADDR / EPxTBCNT 缓冲区描述符列表地址	EPxRBADDR / EPxRBCNT缓冲区描述符列表地址.
	1	0	EPxRBADDR / EPxRBCNT缓冲区描述符列表地址	EPxTBADDR / EPxTBCNT缓冲区描述符列表地址
IN	0	1	EPxTBADDR / EPxTBCNT缓冲区描述符列表地址	EPxRBADDR / EPxRBCNT缓冲区描述符列表地址
	1	0	EPxRBADDR / EPxRBCNT缓冲区描述符列表地址	EPxTBADDR / EPxTBCNT缓冲区描述符列表地址

端点存储请求仲裁

由于USB外设通过APB1接口连到APB1总线上，所以APB1接口接收来自于APB1总线的存储请求与来自于USB接口的端点存储请求。它通过给APB1总线更高的优先权来解决冲突，并且总是保留一半的存储器带宽供USBD完成传输。它采用时分复用的策略实现了虚拟的双端口SRAM，即在USB传输的同时，允许应用程序访问存储器。此策略也允许任意长度的多字节APB1传输。

22.6.2. USB 传输

USB 事务处理

在端点配置后并且事务被请求，硬件将会检测令牌包。当USBD收到令牌包后，将执行数据传输。数据传输完成后，根据传输方向USBD将产生相应的握手包发送出去或期望主机发送相应的握手包。

事务处理完成后，一个端点相关的中断将被触发。在中断处理例程中，应用程序将作相应的处理。

事务格式化是硬件完成的，包括CRC的产生与校验。

一旦端点被使能。端点控制和状态寄存器、端点缓冲区地址和传输长度都不能被应用程序修改。

当正确传输中断通知数据传输操作完成时，它们就可以被再次访问，从而使能一次新的操作。

IN 事务

当一个有效的已配置端点收到一个IN令牌包，它将会发送数据包给主机。如果端点无效，根据当前的端点状态，将发送NAK或STALL握手包。

在数据包传输过程中：首先将发送一个配置好的数据PID，然后端点缓冲区中的实际数据会加载到输出移位寄存器中发送出去，在数据发送后硬件会发送计算好的CRC。

当收到来自主机发送的ACK包，USB外设将翻转数据PID，设置当前的端点状态为NAK。同时，正确传输中断将被触发。在中断服务例程中，应用程序将数据包存储器填满数据后通过设置端点状态为VALID再次使能端点从而开始下一次传输。

OUT 和 SETUP 事务

USBD在处理这两类令牌包时基本使用相同的方式，处理SETUP包的不同点会在下面关于控制传输的部分详细叙述。

在接收端点配置好并且使能后，主机将发送OUT/SETUP令牌给设备。当USBD接收到令牌包后，将会访问端点缓冲区描述符来初始化端点缓冲区地址与长度。然后接收的数据将被陆续打包成字（LSB模式）传输到端点缓冲区。当检测到DATA包传输结束时，计算的CRC的值将和接收到的CRC值进行比较。如果没有错误发生，将向主机发送一个ACK握手包。

当事务正确完成时，USBD将翻转数据PID并设置端点状态为NAK。然后硬件将触发端点正确传输中断。在中断服务例程中，应用程序可以判断事务类型并从端点缓冲区中读出接收数据。在接收数据被处理后，应用程序通过设置端点状态为VALID来发起下一次事务。

如果接收期间出现了任何错误，USBD设置错误中断位并且继续将数据复制到报文缓冲区，但不发送ACK包。USBD自己能够从接收错误中恢复并且继续处理下一个数据传输。USBD的访问从不超出数据包缓冲区，这是通过内部寄存器的配置来控制的。接收到的2字节的CRC也会复制到数据包缓冲区，紧跟在数据字节之后。如果数据的长度比实际分配的长度大，超出的数据不被复制。这种情况称为缓冲过载。此时将发送STALL握手包，当前会话失败。

如果一个被寻址的端点是无效的，将按照当前端点状态发送NAK或STALL握手包而不是ACK，并且不向接收缓冲区写入数据。

控制传输

控制传输要求主机从一个到设备的SETUP事务开始，此事务描述设备应当执行的访问控制类型。SETUP事务后面跟着零个或者若干个携带被请求访问特定信息的数据会话。 最后，一个状态事务完成控制传输并且允许端点返回控制传输的状态到客户端软件。在状态会话之后，控制传输完成，主机可以操作该端点的下一个控制传输。

USBD总是使用双向的端点0作为默认的控制端点来处理控制传输。USBD通过解析SETUP事务的内容获取其关心的数据量和传输方向，并且要求将未使用的方向置为STALL，除了最后一个数据阶段。

在最后一个数据阶段，应用软件将控制端点相反方向的端点0状态设置为NAK。这将使主机等待控制操作的完成。如果操作成功完成，软件将会把NAK改成VALID，否则改成STALL。如果

状态阶段是一个OUT事务，STATUS_OUT位将会被置位，这样携带非零数据的状态会话将会被应答STALL以表明已发生了错误。

根据USB协议，设备不需要废弃当前命令，再开启新命令，因此，设备必须用ACK握手包回复SETUP包，而不是NAK或STALL握手包。

当已配置的控制端点接受到SETUP令牌包，USBD接收这个数据，执行被要求的数据传输并且发送回一个ACK握手包。如果关于此前的数据传输请求未被成功处理，USB丢弃SETUP命令包，视当前情况为错误情况，然后促使主机重新发送请求令牌包。

同步传输

同步传输可以保证固定的传输速率以及固定的延迟，但当总线发生错误时并不支持数据重发。因此，同步协议没有一个握手阶段，在数据包发送之后没有ACK包。它也不支持数据翻转，DATA0 PID仅仅被用来开始一个数据包的发送。

同步端点的状态只能被设定为DISABLED和VALID，其他的任何值是非法的。应用软件可以实现双缓冲以提高性能。通过在每个会话时交换发送和接收包缓冲，应用软件可以将数据复制进缓冲或复制出缓冲，同时USB外设能在另一个缓冲区中处理数据的发送或接收。通过查询DTOG位即可知道USB外设现在正在使用哪一个缓冲区。

应用软件按照要用的首个缓冲区去初始化DTOG。在每一个事务的结尾，RX_ST或TX_ST位被置位，这取决于使能方向，而忽略CRC错误或缓冲过载情况（如果错误发生，ERRIF位将被置位）。同时，USB外设将会翻转DTOG位，但是不影响STAT位。

22.6.3. USB 事件与中断

每一个USB行为都通过应用程序初始化，由USB中断或事件来驱动。在系统复位后，应用程序需要等待一系列的USB中断和事件。

复位事件

系统和上电复位

一旦系统或上电复位，应用程序首先要提供USB模块与接口所需的所有时钟，然后清除复位信号以访问该模块的寄存器，最后打开和USB收发器相连的模拟部分。

USB固件需要做以下工作：

- 复位USBD_CTL寄存器中的CLOSE位；
- 等待内部参考电压稳定；
- 清除USBD_CTL寄存器的SETRST位；
- 清除USBD_INTF寄存器以移除冗余的挂起中断，然后使能其他单元。

USB复位（复位中断）

当这个事件发生时，USB外设的状态同系统复位后状态是一样的。

USB固件需要做以下工作：

- 在10ms内设定USBD_DADDR寄存器的USBEN位来使能USB模块；

- 初始化USBD_EP0CS寄存器和它相关的数据包缓冲。

挂起和恢复事件

在任何需要的情况下，通过写控制寄存器（USBD_CTL）总可以强制使USB模块置于低功耗模式（SUSPEND模式）。此时，不产生任何静态电流消耗，同时USB时钟也会减慢或停止。通过对USB线上数据传输的检测，可以在低功耗模式下唤醒USB模块。

USB协议一直强调由USB从设备进行电源管理。如果设备从总线获得电源(总线供电设备)的话，这一点变得尤其重要。总线供电设备必须满足下述限制：

- 处于非配置状态的从设备，从USB总线最多获取100mA的电流；
- 已配置的设备只能按照配置描述符的Max Power位域的设置获得电流且最大值不超过500 mA；
- 已挂起设备最多获取500uA电流。

假如在USB总线上没有活动超过3ms，设备将进入挂起状态。如果有来自主机的唤醒信号，将唤醒一个挂起的设备。

USB外设还支持软件初始化的远程唤醒。为了启动远程唤醒功能，应用软件在MCU唤醒后必须使能所有的时钟，并清除挂起位。这将导致硬件产生一个远程唤醒信号的上行数据流。

将SETSPS位设为1，即可使能挂起模式并且禁用对于SOF接收的检查。将LOWM位设为1将关闭USB模拟收发器的静态功耗，但是此时仍能检测到恢复信号。

USB 中断

USB控制器有三个中断线：低优先级中断，高优先级中断和唤醒中断。软件可以配置这些中断以将特定的中断条件连接到位于NVIC表中的这些中断信号上。如果中断状态位和对应的中断使能位都被置位，硬件将会产生一个中断。如果中断条件产生，中断状态位都将被硬件置位（不管中断使能位是否设置）。

- USB低优先级中断（通道37）：可被所有USB事件触发；
- USB高优先级中断（通道38）：只能被同步和双缓冲批量传输的正确传输事件触发；
- USB唤醒中断（通道42）：可被所有的唤醒事件触发。

22.6.4. 操作指南

此部分主要描述USBD的操作指南。

USBD 寄存器初始化过程

1. 清除USBD_CTL寄存器的CLOSE位，然后清除SETRST位；
2. 清除USBD_INTF寄存器来移除冗余的挂起中断；
3. 编程USBD_BADDR寄存器来设定端点缓冲区基地址；
4. 设定USBD_CTL来使能中断；
5. 等待复位中断（RSTIF）；
6. 在复位中断中初始化控制端点0来发起枚举过程，然后编程USBD_BADDR来设定设备地址为0并使能USB模块功能；

7. 配置端点0来接收SETUP包。

USBD 端点初始化过程

1. 编程USBD_EPxTBADDR/USBD_EPxRBADDR寄存器来设定发送或者接收数据缓冲区地址；
2. 根据端点的用处编程USBD_EPxCS寄存器的EP_CTL和EP_KCTL位来设定端点类型和缓冲区类型；
3. 如果端点是单缓冲端点：
   1） 编程USBD_EPxCS寄存器的TX_DTG或者RX_DTG位来初始化端点的数据翻转位，但是端点0需要将这两位分别设置为1和0来进行控制传输；
   2） 编程USBD_EPxCS寄存器的TX_STA或者RX_STA位来配置寄存器的状态，但是如果使用端点0来发起控制传输，则这两位都要配置为NAK。

   如果端点是双缓冲端点：
   1） 发送与接收数据翻转位都需要编程。如果端点是发送端点，清除USBD_EPxCS寄存器中的TX_DTG和RX_DTG位，否则如果是接收端点，需要翻转TX_DTG位；
   2） 编程USBD_EPxTBCNT和USBD_EPxRBCNT寄存器来设定传输数据字节数；
   3） 端点发送与接收状态都需要配置。如果端点是发送端点，设定TX_STA位为NAK和RX_STA位为DISABLED，否则如果是接收端点，RX_STA位设定为VALID，TX_STA位设定为DISABLED。

SETUP 和 OUT 数据传输

1. 编程USBD_EPxRBCNT寄存器来设定BLKSIZ和EPRCNT域，这些域定义了端点缓冲区长度；
2. 通过编程USBD_EPxCS寄存器配置端点寄存器状态为VALID来使能端点；
3. 等待正确传输中断（STIF）；
4. 在中断处理例程中，应用程序通过读取USBD_EPxCS寄存器的SETUP位可以决定事务的类型。然后应用程序从端点数据缓冲区中USBD_EPxRBAR寄存器定义的起始地址处读取数据负载。最后应用程序会解析这些数据并进行相应的处理。

IN 数据传输

1. 编程USBD_EPxTBCNT寄存器来设定EPTCNT域，此域定义了端点缓冲区长度；
2. 通过编程USBD_EPxCS寄存器配置端点寄存器状态为VALID来使能端点去发送数据；
3. 等待正确传输中断（STIF）；
4. 在中断处理例程中，应用程序需要更新用户缓冲区长度与位置指针。然后应用程序使用用户缓冲区的数据填充端点缓冲区。最后应用程序将配置端点状态为VALID来进行下一次传输。

22.7. USBD 寄存器

USBD 基地址：0x4000 5C00

22.7.1. USBD 控制寄存器 (USBD_CTL)

地址偏移：0x40
复位值：0x0003

该寄存器可半字(16 位)或全字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STIE	PMOUIE	ERRIE	WKUPIE	SPSIE	RSTIE	SOFIE	ESOFIE	保留			RSREQ	SETSPS	LOWM	CLOSE	SETRST
rw	rw	rw	rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15	STIE	成功传输中断使能 0：禁用成功传输中断 1：当USBD_INTF寄存器的STIF位被置位，产生中断
14	PMOUIE	包缓冲上溢/下溢中断使能 0：当包缓冲上溢/下溢不产生中断 1：当USBD_INTF寄存器的PMOUIF位被置位，产生中断.
13	ERRIE	错误中断使能 0：禁用错误中断 1：当USBD_INTF寄存器的ERRIF位被置位，产生中断
12	WKUPIE	唤醒中断使能 0：禁用唤醒中断 1：当USB_IFR寄存器的WKUPIF位被置位，产生中断
11	SPSIE	挂起状态中断使能 0：禁用挂起状态中断 1：当USBD_INTF寄存器的SPSIF位被置位，产生中断
10	RSTIE	USB复位中断使能 0：禁用USB复位中断 1：当USBD_INTF寄存器的RSTIF位被置位，产生中断
9	SOFIE	帧起始中断使能 0：禁用帧起始中断

		1：当USBD_INTF寄存器的SOFIF位被置位，产生中断
8	ESOFIE	预期的帧起始中断使能 0：禁用预期的帧起始中断 1：当USBD_INTF寄存器的ESOFIF位被置位，产生中断
7:5	保留	必须保持复位值
4	RSREQ	恢复请求 软件向USB主机设置一个中断请求，USB主机应该按USB规范驱动这个恢复序列 0：没有恢复请求 1：发送恢复请求
3	SETSPS	设置挂起 当USBD_INTF寄存器的SPSIF位被置位时，软件应该设置挂起状态 0：没有设置挂起状态 1：设置挂起状态
2	LOWM	低功耗状态 当置位这一位时，USB在挂起状态进入低功耗模式。如果从挂起状态恢复，硬件会复位这一位。 0：无影响 1：在挂起模式进入低功耗模式
1	CLOSE	关闭状态 当这一位被置位的时候，USBD进入关闭状态，并且完全关闭USBD，同主机断开 0：不在关断状态 1：在关断状态
0	SETRST	设定复位 当这位置位，USBD外设应该被复位 0：无影响 1：发生复位

22.7.2. USBD 中断标志寄存器 (USBD_INTF)

地址偏移：0x44
复位值：0x0000

该寄存器可半字（16 位）或全字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STIF	PMOUIF	ERRIF	WKUPIF	SPSIF	RSTIF	SOFIF	ESOFIF	保留			DIR	EPNUM[3:0]			
r	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0				r	r			

位/位域	名称	描述
31:16	保留	必须保持复位值
15	STIF	成功传输中断标志 当一个会话成功完成时，硬件置位该位
14	PMOUIF	包缓冲溢出/下溢中断标志 硬件置位该位表示包缓冲区存储不下所有所传输的数据。软件写0清该位
13	ERRIF	错误中断标志 当在会话期间有错误发生时，硬件置位该位。软件写0清该位
12	WKUPIF	唤醒中断标志 在SUSPEND状态下，当总线上有活动被检测到时，硬件置位该位。 软件写0清该位
11	SPSIF	挂起状态中断标志 当USB总线无任何活动超过3ms时，硬件置位该位，表明有SUSPEND请求。软件写0清该位
10	RSTIF	USB复位中断标志 当检测到USB RESET信号时硬件置位该位。软件写0清该位
9	SOFIF	帧起始中断标志 一个新的SOF包到达时硬件置位该位。软件写0清该位
8	ESOFIF	预期的帧起始中断标志 硬件置位表示一个SOF被预期但是还没有到达。软件写0清该位
7:5	保留	必须保持复位值
4	DIR	会话传输方向 硬件置位表示会话的传输方向 0：IN类型 1：OUT类型
3:0	EPNUM[3:0]	端点号 硬件置位确认当前会话所关联的端点

22.7.3. USBD 状态寄存器 (USBD_STAT)

地址偏移：0x48
复位值：0x0XXX 这里X是未定义的

该寄存器可半字（16 位）或全字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_DP	RX_DM	LOCK	SOFLN[1:0]		FCNT[10:0]										

r r r r r

位/位域	名称	描述
31:16	保留	必须保持复位值
15	RX_DP	接收数据 + 线状态 代表DP线的状态
14	RX_DM	接收数据 - 线状态 代表DM线的状态
13	LOCK	锁定USB 硬件置位表明接收到了至少两个连续SOF包
12:11	SOFLN[1:0]	丢失SOF 当每次发生ESOFIF事件时，硬件递增此位，一旦再次接收到SOF则清除该位
10:0	FCNT[10:0]	帧编号计数器 每次收到SOF，帧编号计数器将会增加

22.7.4. USBD 设备地址寄存器 (USBD_ADDR)

地址偏移：0x4C
复位值：0x0000

该寄存器可半字（16 位）或全字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								USBEN	USBDAR[6:0]						
								rw	rw						

位/位域	名称	描述
31:8	保留	必须保持复位值
7	USBEN	USB设备使能 通过软件设置该位使能USB设备 0：USB设备禁用。没有会话要处理 1：USB设备使能
6:0	USBDAR[6:0]	USBD设备地址 总线复位之后，地址被复位为0x00。若USB使能位被置位，则从设备会响应功能地址DEV_ADDR的报文。

22.7.5. USBD 缓冲器地址寄存器 (USBD_BADDR)

地址偏移：0x50
复位值：0x0000

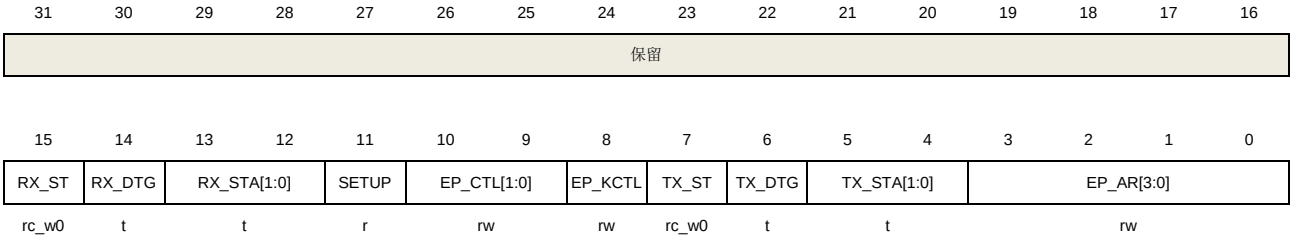
该寄存器可半字（16 位）或全字（32 位）访问。

位/位域	名称	描述
31:16	保留	必须保持复位值
15:3	BAR[12:0]	缓冲器地址 所分配缓冲器(512byte on-chip SRAM)的起始地址，用来保存缓冲描述符表以及包缓冲
2:0	保留	必须保持复位值

22.7.6. USBD 端点 x 控制/状态寄存器 (USB_EPxCS), x=[0..7]

地址偏移：0x00 ~ 0x1C
复位值：0x0000

该寄存器可半字（16 位）或全字（32 位）访问。



位/位域	名称	描述
31:16	保留	必须保持复位值
15	RX_ST	正确接收 当一个成功的OUT/SETUP会话完成时，硬件置位此位 通过软件写0清该位
14	RX_DTG	接收数据PID翻转位 本标志位代表非同步端点的翻转数据位(0=DATA0，1=DATA1) 用来实现双缓冲端点的流控功能

用于同步端点的缓冲区交换

13:12 RX_STA[1:0] 接收状态位
通过软件写1翻转
写0保持不变
参考下表

11 SETUP Setup会话完成
当一个SETUP会话完成时，硬件置位此位

10:9 EP_CTL[1:0] 端点类型控制I
参考下表

8 EP_KCTL 端点类别控制
其具体含义取决于端点类型的设置
参考下表

7 TX_ST 正确发送
当一个IN会话成功完成时，硬件置位此位
软件清0

6 TX_DTG 发送数据PID翻转位
本标志位代表非同步端点的翻转数据位(0=DATA0，1=DATA1)
用来实现双缓冲端点的流控功能
用于同步端点的缓冲区交换

5:4 TX_STA[1:0] 发送状态位
参考下表

3:0 EP_AR 端点地址
用来指示会话的目标端点

表 22-1 接收状态编码

RX_STA[1:0]	含义
00	**DISABLED:**忽略此端点的所有接收请求
01	**STALL:**握手状态为 STALL
10	**NAK**:握手状态为 NAK
11	**VALID**:使能端点的接收

表 22-2. 端点类型编码

EP_CTL[1:0]	含义
00	**BULK:**批量端点
01	**CONTROL:**控制端点
10	**ISO**:同步端点
11	**INTERRUPT**:中断端点

表 22-3. 端点类别编码

EP_CTL[1:0]		EP_KCTL 含义
00	BULK	DBL_BUF

EP_CTL[1:0]		EP_KCTL 含义
01	CONTROL	STATUS_OUT

表 22-4. 发送状态编码

TX_STA[1:0]	Meaning
00	**DISABLED:**忽略端点的所有发送请求
01	**STALL:**握手包状态为 STALL
10	**NAK**:握手包状态为 NAK
11	**VALID**:使能端点的发送

22.7.7. USBD 端点 x 发送缓冲地址寄存器 (USBD_EPxTBADDR), x=[0…7]

地址偏移：[USBD_BADDR] + x * 16
USB本地地址：[USBD_BADDR] + x * 8

该寄存器可半字（16 位）或全字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EPTXBAR[15:1]															EPTXBAR[0]
rw															rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:1	EPTXBAR[15:1]	发送缓冲地址 在收到下一个IN分组时，需要发送的数据所在的缓冲区起始地址
0	EPTXBAR[0]	必须设为0

22.7.8. USBD 端点 x 发送缓冲区字节数目寄存器 (USBD_EPxTBCNT) x=[0…7]

地址偏移：[USBD_BADDR] + x * 16 + 4
USB本地地址：[USBD_BADDR] + x * 8 + 2

该寄存器可半字（16 位）或全字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						EPTXCNT[9:0]									
						rw									

位/位域	名称	描述
31:10	保留	必须保持复位值
9:0	EPTXCNT[9:0]	发送字节数 在收到下一个IN令牌后，将发送的字节数

22.7.9. USBD 端点 x 接收缓冲器地址寄存器 (USBD_EPxRBADDR) x=[0…7]

地址偏移：[USBD_BADDR] + x * 16 + 8
USB本地地址：[USBD_BADDR] + x * 8 + 4

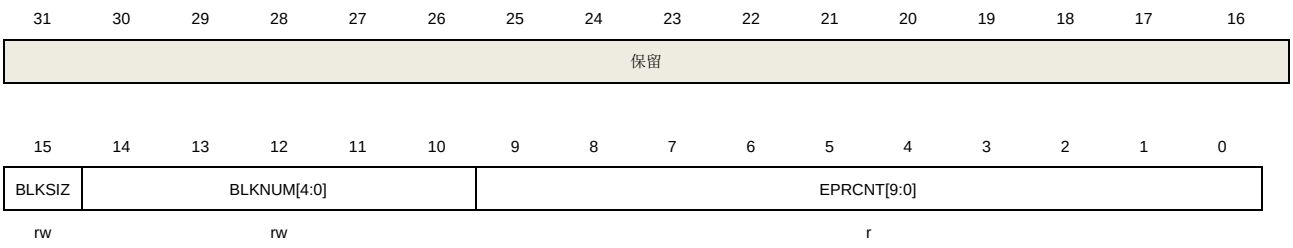
该寄存器可半字（16 位）或全字（32 位）访问。

位/位域	名称	描述
31:16	保留	必须保持复位值
15:1	EPRBAR[15:1]	接收缓冲器地址 收到下一个OUT或者SETUP分组时，用于保存数据的缓冲区起始地址。
0	EPRBAR[0]	必须设为0

22.7.10. USBD 端点 x 接收缓冲区字节数目寄存器 n (USBD_EPxRBCNT) x=[0…7]

地址偏移：[USBD_BADDR] + x * 16 + 12
USB本地地址：[USBD_BADDR] + x * 8 + 6

该寄存器可半字（16 位）或全字（32 位）访问。



位/位域	名称	描述
31:16	保留	必须保持复位值

15	BLKSIZ	块的大小 0：块大小是2字节 1：块大小是32字节
14:10	BLKNUM[4:0]	块数目 包缓冲区所分配的块的数目
9:0	EPRCNT[9:0]	接收字节数 在收到下一个OUT/SETUP令牌后，接收到数据的字节数

23. 附录

23.1. 寄存器表中使用的缩写列表

表 23-1. 寄存器功能位访问属性

功能位访问属性	描述
读/写(rw)	软件可以对这个位进行读写。
只读(r)	软件只能对这个位进行读。
只写(w)	软件只能对这个位进行写。读取该位将返回复位值。
读/写 1 清零(rc_w1)	软件可以读该位，对该位写入 1 可以清除这个位。写入 0 对位值没有影响。
读/写 0 清零(rc_w0)	软件可以读该位，对该位写入 0 可以清除这个位。写入 1 对位值没有影响。
翻转(t)	软件可以通过写 1 来翻转该位。写入 0 对位值没有效果。
只读/写 1 触发(rt_w1)	软件可以读该位，写入 1 触发事件，但对位值没有影响。

23.2. 术语表

表 23-2. 术语

术语	描述
字	32 位长度数据
半字	16 位长度数据.
字节	8 位长度数据
IAP(应用内编程)	IAP 是在用户程序运行时对微控制器的闪存重新编程的能力。
ICP(在线编程)	ICP 是当设备安装在用户应用板上时，一个使用 JTAG 协议，SWD 协议或引导加载程序的微控制器的闪存编程能力。
选项字节	存储在闪存中的产品配置位
AHB	高级高性能总线
APB	高级外设总线
RAZ	读为 0
WI	写忽略
RAZ/WI	读为 0/写忽略

23.3. 可用外设

对于各个 MCU 系列的外设及其数量，请参考相应型号的数据手册。

24. 版本历史

表 24-1. 版本历史

版本号.	说明	日期
1.0	初稿发布	2014 年 3 月
2.0	添加 GD32F170/190 产品	2016 年 1 月
3.0	适用于新的命名规范	2016 年 6 月
3.0.1	校对	2017 年 3 月
3.1.0	校对	2018 年 1 月
3.2	1.依照版本规范，进行格式修改。 2.修改 RCU 模块的寄存器中位域名和时钟树。 3.修改 OPA 寄存器偏移地址。 4.修改 CMP 寄存器偏移地址。 5.修改保护状态选项字节 SPC 与其补字节值为 0xA55A，修改最高安全保护状态的值为 0x33CC。 6.增加 USBD 寄存器基地址。 7.优化 I2S 时钟生成结构框图。 8.修改 WDGT 模块中的关于 EWIF 的描述。 9.更新 SPI/I2S 模块关于四线主机模式的描述。	2019 年 11 月 28 日
3.3	1.修改 USBD_INTF 寄存器 DIR 位的描述。 2.修改 I2C 章节的图 18-6~18-8。	2020 年 3 月 13 日
3.4	1.修改 CAN 章节的波特率公式的格式。 2.在 WDGT 模块的 13.1.3 章节中，添加关于喂完狗后要立刻进 deepsleep/standby 模式的注意事项。 3.在 ADC 模块的 10.4.3 章节中，添加关于 ADC 使能后延时的说明。	2020 年 7 月 1 日
3.5	1.删除 CMP 章节的***功能描述***部分中涉及 GD32F130 和 GD32F170 的描述。 2.修改看门狗章节***表 13 1. 独立看门狗定时器在 40KHz（IRC40K）时的最小 / 最大超时周期***中的参数。 3.修改 RCU 章节***图 4 2. GD32F130xx 和 GD32F150xx 产品的时钟树中 HXTAL*** 的范围为 3~25MHz。	2021 年 12 月 15 日
3.6	1. 更新***电源管理单元（PMU）***章节。 2. 更新***串行外设接口/片上音频接口（SPI/I2S）***章节。 3. 更新***内部集成电路总线接口（I2C）***章节。 4. 更新 ***DMA 控制器（DMA）***章节。 5. 更新***比较器（CMP）***章节。 6. 更新***通用输入/输出接口（GPIO）***章节。 7. 更新***复位和时钟单元（RCU）***章节。	2022 年 7 月 25 日

版本号.	说明	日期
	8. 更新*通用同步异步收发器（USART）*章节。 9. 更新*模拟数字转换器（ADC）*章节。 10. 删除 SLCD、CAN、IVREF 和 OPA 章节。	
3.7	1. 更新 *DMA 控制器（DMA）*章节。 2. 更新*电源管理单元（PMU）*章节。 3. 更新*比较器（CMP）*章节。 4. 更新*复位和时钟单元（RCU）*章节。 5. 更新*闪存控制器（FMC）*章节。 6. 更新*调试（DBG）*章节。 7. 更新*通用串行总线全速设备接口（USBD）*章节。 8. 更新*通用同步异步收发器（USART）*章节。 9. 更新*中断和事件控制器（EXTI）*章节。 10. 更新*触控传感控制器（TSI）*章节。 11. 更新*串行外设接口/片上音频接口（SPI/I2S）*章节。	2023 年 06 月 15 日
3.8	1. 更新*通用输入/输出接口（GPIO）*章节。 2. 更新*通用同步异步收发器（USART）*章节。 3. 更新*比较器（CMP）*章节。 4. 更新*数模转换器（DAC）*章节。 5. 更新*定时器（TIMER）*章节。	2024 年 01 月 05 日
3.9	1. 补充*定时器（TIMER）*章节 TIMERx_INTF 寄存器中 CH0IF 位域描述。 2. 修正*串行外设接口/片上音频接口（SPI/I2S）*章节中关于 I2SCH 位域中文描述歧义。 3. 修改*模拟数字转换器（ADC）*章节读取常规数据寄存器描述以及删除电气特性参数相关描述。 4. 修改*定时器（TIMER）*章节*图 15-8/15-38 中央计数模式计数器时序图* 中 2'b10(downcount only)为 2'b01。 5. 更新*通用串行总线全速设备接口（USBD）*章节*表 22-3 双缓冲的用法*。 6. 删除*通用串行总线全速设备接口（USBD）*章节关于 VBUS 的图表内容。 7.删除*电源管理单元（PMU）*章节图 3-1 中 sleeping 信号	2025 年 2 月 20 日
4.0	1. 修改*通用同步异步收发器（USART）*章节 17.3.11 小节半双工通信模式中关于 TX 引脚的配置描述：修改 TX 引脚配置为开漏模式。 2. 更新*触摸传感器控制器（TSI）*章节寄存器 TSI_CTL0 中 TSIEN 位域描述。 3. 修改*内部集成电路总线接口（I2C）*章节 SMBUS 超时时间，从 35ms 修改为 25ms。	2025 年 8 月 08 日

版本号.	说明	日期
	4. 在***串行外设接口/片上音频接口（SPI/I2S）***章节 19.4.4 小节中新增注意事项：I2S 串行时钟的配置值需设置为低于 PCLK 时钟的 1/6 倍以下(不包含 1/6)。	

Important Notice

This document is the property of GigaDevice Semiconductor Inc. and its subsidiaries (the "Company"). This document, including any product of the Company described in this document (the “Product”), is owned by the Company according to the laws of the People’s Republic of China and other applicable laws. The Company reserves all rights under such laws and no Intellectual Property Rights are transferred (either wholly or partially) or licensed by the Company (either expressly or impliedly) herein. The names and brands of third party referred thereto (if any) are the property of their respective owner and referred to for identification purposes only.

To the maximum extent permitted by applicable law, the Company makes no representations or warranties of any kind, express or implied, with regard to the merchantability and the fitness for a particular purpose of the Product, nor does the Company assume any liability arising out of the application or use of any Product. Any information provided in this document is provided only for reference purposes. It is the sole responsibility of the user of this document to determine whether the Product is suitable and fit for its applications and products planned, and properly design, program, and test the functionality and safety of its applications and products planned using the Product. The Product is designed, developed, and/or manufactured for ordinary business, industrial, personal, and/or household applications only, and the Product is not designed or intended for use in (i) safety critical applications such as weapons systems, nuclear facilities, atomic energy controller, combustion controller, aeronautic or aerospace applications, traffic signal instruments, pollution control or hazardous substance management; (ii) life-support systems, other medical equipment or systems (including life support equipment and surgical implants); (iii) automotive applications or environments, including but not limited to applications for active and passive safety of automobiles (regardless of front market or aftermarket), for example, EPS, braking, ADAS (camera/fusion), EMS, TCU, BMS, BSG, TPMS, Airbag, Suspension, DMS, ICMS, Domain, ESC, DCDC, e-clutch, advanced-lighting, etc.. Automobile herein means a vehicle propelled by a self-contained motor, engine or the like, such as, without limitation, cars, trucks, motorcycles, electric cars, and other transportation devices; and/or (iv) other uses where the failure of the device or the Product can reasonably be expected to result in personal injury, death, or severe property or environmental damage (collectively "Unintended Uses"). Customers shall take any and all actions to ensure the Product meets the applicable laws and regulations. The Company is not liable for, in whole or in part, and customers shall hereby release the Company as well as its suppliers and/or distributors from, any claim, damage, or other liability arising from or related to all Unintended Uses of the Product. Customers shall indemnify and hold the Company, and its officers, employees, subsidiaries, affiliates as well as its suppliers and/or distributors harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of the Product.

Information in this document is provided solely in connection with the Product. The Company reserves the right to make changes, corrections, modifications or improvements to this document and the Product described herein at any time without notice. The Company shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2025 GigaDevice Semiconductor Inc. – All rights reserved