GigaDevice Semiconductor Inc.

GD32E25x
Arm® Cortex®-M23 32-bit MCU

适用于 **GD32E251xx，GD32E252xx 系列**

用户手册

1.1 版本

(2026 年 8 月)

目录

目录 2

图索引 16

表索引 22

1. 系统及存储器架构 24

1.1. Arm® Cortex®-M23 处理器 24

1.2. 系统架构 25

1.3. 内存映射 26

1.3.1. 片上 SRAM 存储器 29
1.3.2. 片上闪存存储器 30

1.4. 引导配置 31

1.5. 系统配置寄存器(SYSCFG) 32

1.5.1. 系统配置寄存器 0(SYSCFG_CFG0) 32
1.5.2. EXTI 源选择寄存器 0(SYSCFG_EXTISS0) 33
1.5.3. EXTI 源选择寄存器 1(SYSCFG_EXTISS1) 34
1.5.4. EXTI 源选择寄存器 2(SYSCFG_EXTISS2) 35
1.5.5. EXTI 源选择寄存器 3(SYSCFG_EXTISS3) 36
1.5.6. 系统配置寄存器 1(SYSCFG_CFG1) 37
1.5.7. 系统配置状态寄存器(SYSCFG_STAT) 38
1.5.8. SRAM_ECC 寄存器(SYSCFG_SRAM_ECC) 39
1.5.9. IRQ 延迟寄存器 (SYSCFG_CPU_IRQ_LAT) 39

1.6. 设备电子签名 40

1.6.1. 内存密度信息 40
1.6.2. 唯一设备标识符(96 位) 41

2. 中断/事件控制器(EXTI) 43

2.1. 简介 43
2.2. 主要特征 43
2.3. 功能说明 43
2.4. 外部中断及事件(EXTI)结构框图 46
2.5. 外部中断及事件功能概述 46
2.6. EXTI 寄存器 48
2.6.1. 中断使能寄存器(EXTI_INTEN) 48
2.6.2. 事件使能寄存器(EXTI_EVEN) 48
2.6.3. 上升沿触发使能寄存器(EXTI_RTEN) 49
2.6.4. 下降沿触发使能寄存器(EXTI_FTEN) 50
2.6.5. 软件中断事件寄存器(EXTI_SWIEV) 50
2.6.6. 挂起寄存器(EXTI_PD) 51
2.6.7. 默认源控制寄存器(EXTI_CTL) 52
3. 直接存储器访问控制器(DMA) 53
3.1. 简介 53
3.2. 主要特征 53
3.3. 结构框图 54
3.4. 功能说明 54
3.4.1. DMA 操作 54
3.4.2. 外设握手 56
3.4.3. 仲裁 56
3.4.4. 地址生成 56
3.4.5. 循环模式 56
3.4.6. 存储器到存储器模式 57
3.4.7. 通道配置 57
3.4.8. 中断 57
3.4.9. DMA 请求映射 58
3.5. DMA 寄存器 61

3.5.1. 中断标志位寄存器(DMA_INTF) 61
3.5.2. 中断标志位清除寄存器(DMA_INTC) 61
3.5.3. 通道 x 控制寄存器(DMA_CHxCTL) 62
3.5.4. 通道 x 计数寄存器(DMA_CHxCNT) 64
3.5.5. 通道 x 外设基地址寄存器(DMA_CHxPADDR) 65
3.5.6. 通道 x 存储器基地址寄存器(DMA_CHxMADDR) 65

4. 闪存控制器(FMC) 66

4.1. 简介 66

4.2. 主要特性 66

4.3. 功能描述 66

4.3.1. 闪存结构 66
4.3.2. 错误检查与纠正(ECC) 67
4.3.3. 读操作 67
4.3.4. FMC_CTL 寄存器解锁 68
4.3.5. 页擦除 69
4.3.6. 整片擦除 70
4.3.7. 主存储闪存块编程 71
4.3.8. OTP 编程 73
4.3.9. 选项字节擦除 73
4.3.10. 选项字节编程 74
4.3.11. 选项字节说明 74
4.3.12. 页擦除/编程保护 76
4.3.13. 安全保护 76

4.4. FMC 寄存器 78

4.4.1. 等待状态寄存器 (FMC_WS) 78
4.4.2. 解锁寄存器 (FMC_KEY) 78
4.4.3. 选项字节解锁寄存器 (FMC_OBKEY) 79
4.4.4. 状态寄存器 (FMC_STAT) 79
4.4.5. 控制寄存器 (FMC_CTL) 80
4.4.6. 地址寄存器(FMC_ADDR) 82
4.4.7. 选项字节状态寄存器 (FMC_OBSTAT) 82
4.4.8. 写保护寄存器 (FMC_WP) 83
4.4.9. ECC 控制及状态寄存器(FMC_ECCCS) 83
4.4.10. ECC 地址寄存器(FMC_ECCADDR) 84
4.4.11. 产品 ID 寄存器 (FMC_PID) 85

5. 电源管理单元(PMU) 86

5.1. 简介 86
5.2. 主要特征 86
5.3. 功能说明 86
5.3.1. V_{DD} / V_{DDA} 电源域 87
5.3.2. V_{CORE} 电源域 89
5.3.3. 省电模式 89
5.4. PMU 寄存器 92
5.4.1. 控制寄存器(PMU_CTL) 92
5.4.2. 电源控制和状态寄存器(PMU_CS) 93
6. 备份域(BKP) 95
6.1. 简介 95
6.2. 主要特征 95
6.3. 功能说明 95
6.3.1. 结构框图 95
6.3.2. 侵入检测 95
6.4. BKP 中断 96
6.5. BKP 寄存器 97
6.5.1. 状态寄存器(BKP_STAT) 97
6.5.2. 侵入寄存器(BKP_TAMP) 97
6.5.3. 备份域寄存器(BKP_DATAx)(x=0…4) 98
7. 复位和时钟单元(RCU) 100
7.1. 复位控制单元(RCTL) 100
7.1.1. 简介 100
7.1.2. 功能描述 100
7.2. 时钟控制单元(CCTL) 101
7.2.1. 简介 101
7.2.2. 主要特性 103

7.2.3. 功能描述 103

7.3. RCU 寄存器 107

7.3.1. 控制寄存器 0(RCU_CTL0) 107
7.3.2. 配置寄存器 0(RCU_CFG0) 108
7.3.3. 中断寄存器(RCU_INT) 112
7.3.4. APB2 复位寄存器(RCU_APB2RST) 114
7.3.5. APB1 复位寄存器(RCU_APB1RST) 115
7.3.6. AHB 使能寄存器(RCU_AHBEN) 118
7.3.7. APB2 使能寄存器(RCU_APB2EN) 119
7.3.8. APB1 使能寄存器(RCU_APB1EN) 121
7.3.9. 备份域控制寄存器(RCU_BDCTL) 123
7.3.10. 复位源/时钟寄存器(RCU_RSTSCK) 124
7.3.11. AHB 复位寄存器(RCU_AHBRST) 125
7.3.12. 配置寄存器 1(RCU_CFG1) 126
7.3.13. 配置寄存器 2(RCU_CFG2) 127
7.3.14. 控制寄存器 1(RCU_CTL1) 128
7.3.15. 电压秘钥寄存器(RCU_VKEY) 129
7.3.16. 深度睡眠模式电压寄存器(RCU_DSV) 129

8. 通用输入/输出接口(GPIO) 131

8.1. 概述 131

8.2. 特性 131

8.3. 功能概述 131

8.3.1. GPIO 引脚配置 133
8.3.2. 外部中断/事件线路 133
8.3.3. 备用功能(AF) 134
8.3.4. 附加功能 134
8.3.5. 输入配置 134
8.3.6. 输出配置 134
8.3.7. 模拟配置 135
8.3.8. 备用功能(AF)配置 135
8.3.9. GPIO 锁定功能 136
8.3.10. GPIO 单周期输出翻转功能 136
8.3.11. GPIO 复位保持 136
8.3.12. 优先级交叉开关解码器(仅适用于 GD32E251xx) 136
8.3.13. INT0 和 INT1(仅适用于 GD32E251xx) 138
8.3.14. 引脚匹配(仅适用于 GD32E251xx) 138
8.3.15. 引脚不匹配唤醒从深度睡眠模式(仅适用于 GD32E251xx) 138

8.4. 寄存器定义 139
8.4.1. 端口控制寄存器(GPIOx_CTL，x=A, B, F) 139
8.4.2. 端口输出模式寄存器(GPIOx_OMODE，x=A, B, F) 141
8.4.3. 端口输出速度寄存器(GPIOx_OSPD，x=A, B, F) 143
8.4.4. 端口上拉/下拉寄存器(GPIOx_PUD，x=A, B, F) 144
8.4.5. 端口输入状态寄存器(GPIOx_ISTAT，x=A, B, F) 146
8.4.6. 端口输出控制寄存器(GPIOx_OCTL，x=A, B, F) 147
8.4.7. 端口位操作寄存器(GPIOx_BOP，x=A, B, F) 147
8.4.8. 端口配置锁定寄存器(GPIOx_LOCK，x=A, B, F) 148
8.4.9. 备用功能选择寄存器 0(GPIOx_AFSEL0，x=A / B) 148
8.4.10. 备用功能选择寄存器 1(GPIOx_AFSEL1，x=A / B) 150
8.4.11. 位清除寄存器(GPIOx_BC，x=A, B, F) 151
8.4.12. 端口位翻转寄存器(GPIOx_TG，x=A, B, F) 151
8.4.13. 交叉开关使能寄存器(GPIO_CBREN) 152
8.4.14. 数字功能使能寄存器(GPIO_DFEN) 154
8.4.15. 交叉开关跳过寄存器(GPIO_SKIP) 155
8.4.16. INT0 和 INT1 控制寄存器(GPIO_INTCTL) 157
8.4.17. 引脚屏蔽寄存器(GPIO_MASK) 159
8.4.18. 引脚匹配寄存器(GPIO_MATCH) 162
8.4.19. 引脚匹配标志寄存器(GPIO_MATCHF) 165

9. 定时器(TIMER) 168

9.1. 高级定时器(TIMERx,x=0) 169
9.1.1. 简介 169
9.1.2. 主要特征 169
9.1.3. 结构框图 170
9.1.4. 功能说明 170
9.1.5. TIMERx 寄存器(x=0) 195

9.2. 通用定时器 L0(TIMERx, x=1,2) 224
9.2.1. 简介 224
9.2.2. 主要特征 224
9.2.3. 结构框图 224
9.2.4. 功能说明 225
9.2.5. TIMERx 寄存器(x=1,2) 238

9.3. 通用定时器 L2(TIMERx, x=13) 262
9.3.1. 简介 262
9.3.2. 主要特征 262
9.3.3. 结构框图 262
9.3.4. 功能说明 262

9.3.5. TIMERx 寄存器(x=13) 270

9.4. 通用定时器 L3(TIMERx,x=14) 280

9.4.1. 简介 280
9.4.2. 主要特性 280
9.4.3. 结构框图 280
9.4.4. 功能描述 281
9.4.5. TIMERx 寄存器(x=14) 297

9.5. 通用定时器 L4(TIMERx,x=15,16) 316

9.5.1. 简介 316
9.5.2. 主要特性 316
9.5.3. 结构框图 316
9.5.4. 功能描述 317
9.5.5. TIMERx 寄存器(x=15,16) 331

9.6. 基本定时器(TIMERx, x=5,6) 346

9.6.1. 简介 346
9.6.2. 主要特征 346
9.6.3. 结构框图 346
9.6.4. 功能说明 346
9.6.5. TIMERx 寄存器(x=5,6) 350

10. 看门狗定时器(WDGT) 355

10.1. 独立看门狗定时器(FWDGT) 355

10.1.1. 简介 355
10.1.2. 主要特征 355
10.1.3. 功能说明 355
10.1.4. FWDGT 寄存器 358

10.2. 窗口看门狗定时器(WWDGT) 361

10.2.1. 简介 361
10.2.2. 主要特征 361
10.2.3. 功能说明 361
10.2.4. WWDGT 寄存器 363

11. 调试(DBG) 365

11.1. 简介 365

11.2. SW 功能说明 365
11.2.1. 引脚分配 365
11.3. 调试保持功能说明 365
11.3.1. 低功耗模式调试支持 365
11.3.2. TIMER，I2C，WWDGT 和 FWDGT 外设调试支持 366
11.4. DBG 寄存器 367
11.4.1. ID 寄存器(DBG_ID) 367
11.4.2. 控制寄存器 0(DBG_CTL0) 367
11.4.3. 控制寄存器 1(DBG_CTL1) 369
12. 循环冗余校验计算单元(CRC) 371
12.1. 简介 371
12.2. 主要特征 371
12.3. 功能说明 372
12.4. CRC 寄存器 373
12.4.1. 数据寄存器(CRC_DATA) 373
12.4.2. 独立数据寄存器(CRC_FDATA) 373
12.4.3. 控制寄存器(CRC_CTL) 374
12.4.4. 初值寄存器(CRC_IDATA) 374
12.4.5. 多项式寄存器(CRC_POLY) 375
13. 模数转换器(ADC) 376
13.1. 简介 376
13.2. 主要特征 376
13.3. 引脚和内部信号 377
13.4. 功能说明 378
13.4.1. 前置校准功能 378
13.4.2. 双时钟域架构 379
13.4.3. ADC 使能 379

13.4.4. 常规序列和注入序列 379
13.4.5. 运行模式 379
13.4.6. 注入序列管理 383
13.4.7. 转换数据锁存 384
13.4.8. 转换结果阈值监测功能 385
13.4.9. 数据存储模式 386
13.4.10. 采样时间配置 387
13.4.11. 外部触发 387
13.4.12. DMA 请求 389
13.4.13. ADC 内部通道 389
13.4.14. 可编程分辨率(DRES) 390
13.4.15. 片上硬件过采样 390
13.4.16. ADC 转换信号 392
13.4.17. ADC 中断 392

13.5. ADC 寄存器 393

13.5.1. 状态寄存器 (ADC_STAT) 393
13.5.2. 控制寄存器 0 (ADC_CTL0) 394
13.5.3. 控制寄存器 1 (ADC_CTL1) 396
13.5.4. 采样时间寄存器 0 (ADC_SAMPT0) 398
13.5.5. 采样时间寄存器 1 (ADC_SAMPT1) 399
13.5.6. 注入通道数据偏移寄存器 x(ADC_IOFFx)(x=0..3) 400
13.5.7. 看门狗高阈值寄存器 (ADC_WDHT) 400
13.5.8. 看门狗低阈值寄存器 (ADC_WDLT) 401
13.5.9. 常规序列寄存器 0 (ADC_RSQ0) 401
13.5.10. 常规序列寄存器 1 (ADC_RSQ1) 402
13.5.11. 常规序列寄存器 2 (ADC_RSQ2) 402
13.5.12. 注入序列寄存器(ADC_ISQ) 403
13.5.13. 锁存数据寄存器 x(ADC_LDATAx)(x= 0..3) 404
13.5.14. 常规数据寄存器 (ADC_RDATA) 404
13.5.15. 过采样控制寄存器(ADC_OVSAMPCTL) 404
13.5.16. 采样时间寄存器 2 (ADC_SAMPT2) 406
13.5.17. 控制寄存器 2 (ADC_CTL2) 406
13.5.18. 常规序列寄存器 3 (ADC_RSQ3) 407
13.5.19. 注入数据寄存器(ADC_IDATA) 408
13.5.20. 锁存数据控制寄存器(ADC_LDCTL) 408

14. 数模转换器(DAC) 410

14.1. 简介 410

14.2. 主要特征 410

14.3. 功能描述 412

14.3.1. DAC 使能 412
14.3.2. DAC 输出缓冲 412
14.3.3. DAC 数据配置 412
14.3.4. DAC 触发 412
14.3.5. DAC 转换 413
14.3.6. DAC 噪声波 413
14.3.7. DAC 输出电压 414
14.3.8. DMA 请求 414
14.3.9. DAC 并发转换 414
14.3.10. DAC 复位保持模式 415

14.4. DAC 寄存器 416

14.4.1. DACx 控制寄存器（DAC_CTL0） 416
14.4.2. DACx 软件触发寄存器（DAC_SWT） 419
14.4.3. DACx_OUT0 12 位右对齐数据保持寄存器（DAC_OUT0_R12DH） 419
14.4.4. DACx_OUT0 12 位左对齐数据保持寄存器（DAC_OUT0_L12DH） 420
14.4.5. DACx_OUT0 8 位右对齐数据保持寄存器（DAC_OUT0_R8DH） 420
14.4.6. DACx_OUT1 12 位右对齐数据保持寄存器（DAC_OUT1_R12DH） 421
14.4.7. DACx_OUT1 12 位左对齐数据保持寄存器（DAC_OUT1_L12DH） 421
14.4.8. DACx_OUT1 8 位右对齐数据保持寄存器（DAC_OUT1_R8DH） 422
14.4.9. DACx 并发模式 12 位右对齐数据保持寄存器（DACC_R12DH） 422
14.4.10. DACx 并发模式 12 位左对齐数据保持寄存器（DACC_L12DH） 423
14.4.11. DACx 并发模式 8 位右对齐数据保持寄存器（DACC_R8DH） 423
14.4.12. DACx_OUT0 数据输出寄存器（DAC_OUT0_DO） 424
14.4.13. DACx_OUT1 数据输出寄存器（DAC_OUT1_DO） 424
14.4.14. DACx 状态寄存器（DAC_STAT0） 425
14.4.15. DACx 输出控制寄存器（DAC_OCTL） 425

15. 比较器（CMP） 427

15.1. 简介 427

15.2. 主要特征 427

15.3. 功能描述 427

15.3.1. 比较器时钟 428
15.3.2. 比较器的 I / O 配置 428
15.3.3. 比较器供电模式 430
15.3.4. 比较器迟滞 430
15.3.5. 比较器寄存器写保护 430

15.3.6. 比较器中断 431
15.3.7. 引脚复位功能 431

15.4. CMP 寄存器 432

15.4.1. CMP 控制状态寄存器(CMPx_CS) 432

16. 可配置逻辑阵列(CLA) 435

16.1. 简介 435

16.2. 主要特性 435

16.3. 模块框图 435

16.4. 功能描述 436

16.4.1. CLA 输入信号选择器 436
16.4.2. 逻辑控制单元(LCU)控制 437
16.4.3. CLA 输出 438
16.4.4. 中断 438

16.5. CLA 寄存器 439

16.5.1. 全局控制寄存器(CLA_GCTL) 439
16.5.2. 中断使能寄存器(CLA_INTE) 439
16.5.3. 中断标志寄存器(CLA_INTF) 441
16.5.4. 状态寄存器(CLA_STAT) 442
16.5.5. 信号选择寄存器(CLAx_SIGS)(x=0..3) 443
16.5.6. LCU 控制寄存器(CLAx_LCUCTL)(x=0..3) 443
16.5.7. 控制寄存器(CLAx_CTL)(x=0..3) 444

17. 内部基准电压(VREF) 446

17.1. 简介 446

17.2. 主要特性 446

17.3. 功能描述 446

18. 红外接口(IFRP) 447

18.1. 简介 447

18.2. 主要特性 447
18.3. 功能描述 447
19. 通用同步异步收发器(USART) 449
19.1. 简介 449
19.2. 主要特征 449
19.3. 功能描述 450
19.3.1. USART 帧格式 451
19.3.2. 波特率发生 452
19.3.3. USART 发送器 452
19.3.4. USART 接收器 453
19.3.5. DMA 方式访问数据缓冲区 455
19.3.6. 硬件流控制 456
19.3.7. 多处理器通信 457
19.3.8. LIN 模式 458
19.3.9. 同步通信模式 459
19.3.10. 串行红外(IrDA SIR)编解码功能模块 460
19.3.11. 半双工通信模式 461
19.3.12. 智能卡(ISO7816-3)模式 461
19.3.13. 自动波特率检测 463
19.3.14. ModBus 通信 463
19.3.15. 接收 FIFO 463
19.3.16. 从 Deepsleep 模式唤醒 464
19.3.17. USART 中断 464
19.4. USART 寄存器 466
19.4.1. USART 控制寄存器 0(USART_CTL0) 466
19.4.2. USART 控制寄存器 1(USART_CTL1) 468
19.4.3. USART 控制寄存器 2(USART_CTL2) 471
19.4.4. USART 波特率寄存器(USART_BAUD) 473
19.4.5. USART 保护时间和预分频器寄存器(USART_GP) 474
19.4.6. USART 接收超时寄存器(USART_RT) 475
19.4.7. USART 请求寄存器(USART_CMD) 475
19.4.8. USART 状态寄存器(USART_STAT) 476
19.4.9. USART 中断标志清除寄存器(USART_INTC) 480
19.4.10. USART 数据接收寄存器(USART_RDATA) 481
19.4.11. USART 数据发送寄存器(USART_TDATA) 482

19.4.12. USART 兼容性控制寄存器(USART_CHC) 482
19.4.13. USART 接收 FIFO 控制和状态寄存器(USART_RFCS) 483

20. 串行外设接口/片上音频接口(SPI/I2S) 485

20.1. 简介 485

20.2. 主要特性 485

20.2.1. SPI 主要特性 485
20.2.2. I2S 主要特性 485

20.3. SPI 功能说明 486

20.3.1. SPI 结构框图 486
20.3.2. SPI 信号线描述 486
20.3.3. SPI 时序和数据帧格式 487
20.3.4. 独立发送和接收缓冲区 488
20.3.5. NSS 功能 490
20.3.6. SPI 运行模式 491
20.3.7. DMA 功能 500
20.3.8. CRC 功能 500
20.3.9. SPI 中断 501

20.4. I2S 功能说明 503

20.4.1. I2S 结构框图 503
20.4.2. I2S 信号线描述 503
20.4.3. I2S 音频标准 503
20.4.4. I2S 时钟 511
20.4.5. 运行 512
20.4.6. DMA 功能 516
20.4.7. I2S 中断 516

20.5. SPI/I2S 寄存器 518

20.5.1. 控制寄存器 0(SPI_CTL0) 518
20.5.2. 控制寄存器 1(SPI_CTL1) 520
20.5.3. 状态寄存器(SPI_STAT) 521
20.5.4. 数据寄存器(SPI_DATA) 523
20.5.5. CRC 多项式寄存器(SPI_CRCPOLY) 524
20.5.6. 接收 CRC 寄存器(SPI_RCRC) 524
20.5.7. 发送 CRC 寄存器(SPI_TCRC) 525
20.5.8. I2S 控制寄存器(SPI_I2SCTL) 526
20.5.9. I2S 时钟预分频寄存器(SPI_I2SPSC) 527
20.5.10. SPI1 四线 SPI 控制寄存器(SPI_QCTL) 528

21. 内部集成电路总线接口(I2C) 529

21.1. 内部集成电路总线接口(I2Cx, x=0, 1) 529

21.1.1. 简介 529
21.1.2. 主要特征 529
21.1.3. 功能说明 530
21.1.4. I2C 寄存器 545

21.2. 内部集成电路总线接口(I2C2) 559

21.2.1. 简介 559
21.2.2. 主要特征 559
21.2.3. 功能说明 559
21.2.4. I2C 寄存器 583

22. 附录 597

22.1. 寄存器表中使用的缩写列表 597

22.2. 术语表 597

22.3. 可用外设 598

23. 版本历史 599

图索引

图 1-1. Cortex®-M23 处理器的结构框图 25
图 1-2. GD32E25x 系列器件的系统架构示意图 26
图 1-3. ECC 解码器示意图 30
图 2-1. EXTI 结构框图 46
图 3-1. DMA 结构框图 54
图 3-2. 握手机制 56
图 3-3. DMA 中断逻辑图 58
图 3-4. DMA 请求映射图 59
图 4-1. 页擦除操作流程 70
图 4-2. 整片擦除操作流程 71
图 4-3. 双字编程操作流程 73
图 5-1. 电源域概览 87
图 5-2. 上电/掉电复位波形图 88
图 5-3. LVD 阈值波形图 88
图 6-1. BKP 结构框图 95
图 7-1. 系统复位电路 101
图 7-2. 时钟树 102
图 7-3. HXTAL 时钟源 103
图 7-4. 旁路模式下 HXTAL 时钟源 103
图 8-1.通用输入/输出的基本结构 133
图 8-2. 输入配置的基本结构 134
图 8-3. 输出配置的基本结构 135
图 8-4. 模拟配置的基本结构 135
图 8-5. 备用功能配置的基本结构 136
图 8-6. 数字功能图 136
图 8-7. 交叉开关解码器功能图 137
图 9-1. 高级定时器结构框图 170
图 9-2. 内部时钟分频为 1 时，计数器的时序图 171
图 9-3. 当 PSC 数值从 0 变到 2 时，计数器的时序图 172
图 9-4. 向上计数时序图，PSC=0/2 173
图 9-5. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值 173
图 9-6. 向下计数时序图，PSC=0/2 174
图 9-7. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值 175
图 9-8. 中央计数模式计数器时序图 176
图 9-9. 中央计数模式下计数器重复时序图 177
图 9-10. 在向上计数模式下计数器重复时序图 177
图 9-11. 在向下计数模式下计数器重复时序图 178
图 9-12. 通道输入捕获原理 178
图 9-13. 通道输出比较原理(带有互补输出的通道，x=0,1,2) 179
图 9-14. 通道输出比较原理(CHx_O, x=3,4,5)原理 180

图 9-15. 三种输出比较模式 181
图 9-16. EAPWM 时序图 182
图 9-17. CAPWM 时序图 182
图 9-18. 带死区时间的互补输出 185
图 9-19. 通道响应中止输入(高电平有效)时，输出信号的行为 186
图 9-20.在正交译码器模式 2 且 CI0FE0 极性不反相时计数器行为 187
图 9-21.在正交译码器模式 2 且 CI0FE0 极性反相时计数器行为 187
图 9-22. 霍尔传感器用在 BLDC 电机控制中 188
图 9-23. 两个定时器之间的霍尔传感器时序图 189
图 9-24. 复位模式下的控制电路 190
图 9-25. 暂停模式下的控制电路 190
图 9-26. 事件模式下的控制电路 191
图 9-27. 单脉冲模式，TIMERx_CHxCV = 4 TIMERx_CAR=99 191
图 9-28. 定时器 0 主/从模式的例子 192
图 9-29. 用定时器 2 的 CI0 信号来触发定时器 0 和定时器 2 193
图 9-30. 通用定时器 L0 结构框图 225
图 9-31. 内部时钟分频为 1 时，计数器的时序图 226
图 9-32. 当 PSC 数值从 0 变到 2 时，计数器的时序图 227
图 9-33. 向上计数时序图，PSC=0/2 228
图 9-34. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值 228
图 9-35. 向下计数时序图，PSC=0/2 229
图 9-36. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值 230
图 9-37. 中央计数模式计数器时序图 231
图 9-38. 通道输入捕获原理 232
图 9-39. 通道输出比较原理(x=0,1,2,3) 233
图 9-40. 三种输出比较模式 234
图 9-41. EAPWM 时序图 235
图 9-42. CAPWM 时序图 235
图 9-43. 通用定时器 L2 结构框图 262
图 9-44. 内部时钟分频为 1 时，计数器的时序图 263
图 9-45. 当 PSC 数值从 0 变到 2 时，计数器的时序图 264
图 9-46. 向上计数时序图，PSC=0/2 265
图 9-47. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值 265
图 9-48. 通道输入捕获原理 266
图 9-49. 通道输出比较原理 267
图 9-50. 三种输出比较模式 268
图 9-51. 通用定时器 L3 结构框图 281
图 9-52. 内部时钟分频为 1 时，计数器的时序图 282
图 9-53. 当 PSC 数值从 0 变到 2 时，计数器的时序图 283
图 9-54. 向上计数时序图，PSC=0/2 284
图 9-55. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值 284
图 9-56. 在向上计数模式下计数器重复时序图 285
图 9-57. 通道输入捕获原理 286
图 9-58. 通道输出比较原理(带有互补输出的通道，x=0) 287

图 9-59. 通道输出比较原理 287
图 9-60. 三种输出比较模式 288
图 9-61. PWM 时序图 289
图 9-62. 带死区时间的互补输出 292
图 9-63. 通道响应中止输入(高电平有效)时，输出信号的行为 293
图 9-64. 复位模式下的控制电路 294
图 9-65. 暂停模式下的控制电路 294
图 9-66. 事件模式下的控制电路 294
图 9-67. 单脉冲模式，TIMERx_CHxCV = 4 TIMERx_CAR=99 295
图 9-68. 通用定时器 L4 结构框图 317
图 9-69. 内部时钟分频为 1 时，计数器的时序图 318
图 9-70. 当 PSC 数值从 0 变到 2 时，计数器的时序图 318
图 9-71. 向上计数时序图，PSC=0/2 319
图 9-72. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值 320
图 9-73. 在向上计数模式下计数器重复时序图 321
图 9-74. 通道输入捕获原理 322
图 9-75. 通道输出比较原理(带有互补输出的通道，x=0) 323
图 9-76. 三种输出比较模式 324
图 9-77. PWM 时序图 325
图 9-78. 带死区时间的互补输出 328
图 9-79. 通道响应中止输入(高电平有效)时，输出信号的行为 329
图 9-80. 单脉冲模式，TIMERx_CHxCV = 4 TIMERx_CAR=99 330
图 9-81. 基本定时器结构框图 346
图 9-82. 内部时钟分频为 1 时，计数器的时序图 347
图 9-83. 当 PSC 数值从 0 变到 2 时，计数器的时序图 347
图 9-84. 向上计数时序图，PSC=0/2 348
图 9-85. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值 349
图 10-1. 独立看门狗定时器框图 356
图 10-2. 窗口看门狗定时器框图 361
图 10-3. 窗口看门狗定时器时序图 362
图 12-1. CRC 计算单元框图 371
图 13-1. ADC 模块框图 378
图 13-2. 单次运行模式 379
图 13-3. 连续运行模式 380
图 13-4. 扫描运行模式，且连续运行模式失能 381
图 13-5. 扫描运行模式，连续运行模式使能 382
图 13-6. 间断运行模式 382
图 13-7. 重复运行模式(SM=1) 383
图 13-8. 自动注入，CNT=1 384
图 13-9. 触发注入 384
图 13-10. 数据锁存 384
图 13-11. ADC_LDATAx 读取序列数据 385
图 13-12. 12 位数据存储模式 386
图 13-13. 10 位数据存储模式 386

图 13-14. 8 位数据存储模式 387
图 13-15. 6 位数据存储模式 387
图 13-16. 在常规序列转换过程中常规触发出现 388
图 13-17. 20 位到 16 位的结果截断 391
图 13-18. 右移 5 位和取整的数例 391
图 13-19. 常规序列连续转换模式下的 ADC 转换信号 392
图 14-1.DAC 结构框图 411
图 14-2. DAC LFSR 算法 413
图 14-3. DAC 三角噪声模式生成的波形 414
图 15-1.比较器框图 428
图 15-2. 比较器迟滞 430
图 16-1 CLA 模块框图 435
图 16-2. CLA 中断逻辑 438
图 17-1. VREF 连接 446
图 18-1. IFRP 输出时序图 1 447
图 18-2. IFRP 输出时序图 2 448
图 18-3. IFRP 输出时序图 3 448
图 19-1. USART 模块内部框图 451
图 19-2. USART 字符帧(8 数据位和 1 停止位) 451
图 19-3. USART 发送步骤 453
图 19-4. 过采样方式接收一个数据位(OSB=0) 454
图 19-5. 采用 DMA 方式实现 USART 数据发送配置步骤 455
图 19-6. 采用 DMA 方式实现 USART 数据接收配置步骤 456
图 19-7. 两个 USART 之间的硬件流控制 456
图 19-8. 硬件流控制 457
图 19-9. 空闲状态下检测断开帧 458
图 19-10. 数据传输过程中检测断开帧 459
图 19-11. 同步模式下的 USART 示例 459
图 19-12. 8-bit 格式的 USART 同步通信波形(CLEN=1) 459
图 19-13. IrDA SIR ENDEC 模块 460
图 19-14. IrDA 数据调制 460
图 19-15. ISO7816-3 数据帧格式 461
图 19-16. USART 接收 FIFO 结构 464
图 19-17. USART 中断映射框图 465
图 20-1. SPI 结构框图 486
图 20-2. SPI0 常规模式下的时序图 487
图 20-3. SPI1 常规模式下的时序图 488
图 20-4. SPI 四线模式下的 SPI 时序图(CKPL=1, CKPH=1, LF=0) 488
图 20-5. 发送/接收缓冲区 489
图 20-6. 典型的全双工模式连接 492
图 20-7. 典型的单工模式连接(主机：接收，从机：发送) 492
图 20-8. 典型的单工模式连接(主机：只发送，从机：接收) 493
图 20-9. 典型的双向线连接 493
图 20-10. 主机 TI 模式在不连续发送时的时序图 495

图 20-11. 主机 TI 模式在连续发送时的时序图 496
图 20-12. 从机 TI 模式时序图 496
图 20-13. NSS 脉冲模式时序图(主机连续发送) 497
图 20-14. SPI 四线模式四线写操作时序图 498
图 20-15. SPI 四线模式四线读操作时序图 498
图 20-16. I2S 结构框图 503
图 20-17. I2S 飞利浦标准时序图(DTLEN=00, CHLEN=0, CKPL=0) 504
图 20-18. I2S 飞利浦标准时序图(DTLEN=00, CHLEN=0, CKPL=1) 504
图 20-19. I2S 飞利浦标准时序图(DTLEN=10, CHLEN=1, CKPL=0) 504
图 20-20. I2S 飞利浦标准时序图(DTLEN=10, CHLEN=1, CKPL=1) 504
图 20-21. I2S 飞利浦标准时序图(DTLEN=01, CHLEN=1, CKPL=0) 505
图 20-22. I2S 飞利浦标准时序图(DTLEN=01, CHLEN=1, CKPL=1) 505
图 20-23. I2S 飞利浦标准时序图(DTLEN=00, CHLEN=1, CKPL=0) 505
图 20-24. I2S 飞利浦标准时序图(DTLEN=00, CHLEN=1, CKPL=1) 505
图 20-25. MSB 对齐标准时序图(DTLEN=00, CHLEN=0, CKPL=0) 506
图 20-26. MSB 对齐标准时序图(DTLEN=00, CHLEN=0, CKPL=1) 506
图 20-27. MSB 对齐标准时序图(DTLEN=10, CHLEN=1, CKPL=0) 506
图 20-28. MSB 对齐标准时序图(DTLEN=10, CHLEN=1, CKPL=1) 506
图 20-29. MSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=0) 506
图 20-30. MSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=1) 506
图 20-31. MSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=0) 507
图 20-32. MSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=1) 507
图 20-33. LSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=0) 507
图 20-34. LSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=1) 507
图 20-35. LSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=0) 508
图 20-36. LSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=1) 508
图 20-37. PCM 标准短帧同步模式时序图(DTLEN=00, CHLEN=0, CKPL=0) 508
图 20-38. PCM 标准短帧同步模式时序图(DTLEN=00, CHLEN=0, CKPL=1) 508
图 20-39. PCM 标准短帧同步模式时序图(DTLEN=10, CHLEN=1, CKPL=0) 508
图 20-40. PCM 标准短帧同步模式时序图(DTLEN=10, CHLEN=1, CKPL=1) 509
图 20-41. PCM 标准短帧同步模式时序图(DTLEN=01, CHLEN=1, CKPL=0) 509
图 20-42. PCM 标准短帧同步模式时序图(DTLEN=01, CHLEN=1, CKPL=1) 509
图 20-43. PCM 标准短帧同步模式时序图(DTLEN=00, CHLEN=1, CKPL=0) 509
图 20-44. PCM 标准短帧同步模式时序图(DTLEN=00, CHLEN=1, CKPL=1) 509
图 20-45. PCM 标准长帧同步模式时序图(DTLEN=00, CHLEN=0, CKPL=0) 509
图 20-46. PCM 标准长帧同步模式时序图(DTLEN=00, CHLEN=0, CKPL=1) 510
图 20-47. PCM 标准长帧同步模式时序图(DTLEN=10, CHLEN=1, CKPL=0) 510
图 20-48. PCM 标准长帧同步模式时序图(DTLEN=10, CHLEN=1, CKPL=1) 510
图 20-49. PCM 标准长帧同步模式时序图(DTLEN=01, CHLEN=1, CKPL=0) 510
图 20-50. PCM 标准长帧同步模式时序图(DTLEN=01, CHLEN=1, CKPL=1) 510
图 20-51. PCM 标准长帧同步模式时序图(DTLEN=00, CHLEN=1, CKPL=0) 510
图 20-52. PCM 标准长帧同步模式时序图(DTLEN=00, CHLEN=1, CKPL=1) 511
图 20-53. I2S 时钟生成结构框图 511
图 20-54. I2S 初始化流程 513

图 20-55. I2S 主机接收禁能流程 515
图 21-1. I2C 模块框图 530
图 21-2. 数据有效性 531
图 21-3. 起始和停止信号 531
图 21-4. 时钟同步 532
图 21-5. SDA 线仲裁 532
图 21-6. 7 位地址的 I2C 通讯流程 532
图 21-7. 10 位地址的 I2C 通讯流程(主机发送) 533
图 21-8. 10 位地址的 I2C 通讯流程(主机接收) 533
图 21-9. 从机发送模式(10 位地址模式) 534
图 21-10. 从机接收模式(10 位地址模式) 535
图 21-11. 主机发送模式(10 位地址模式) 537
图 21-12. 主机接收使用方案 A 模式(10 位地址模式) 539
图 21-13. 主机接收使用方案 B 模式(10 位地址模式) 541
图 21-14. I2C 模块框图 560
图 21-15. 数据有效性 561
图 21-16. 开始和停止信号 562
图 21-17. 10 位地址的 I2C 通讯流程(主机发送) 562
图 21-18. 7 位地址的 I2C 通讯流程(主机发送) 562
图 21-19. 7 位地址的 I2C 通讯流程(主机接收) 563
图 21-20. 10 位地址的 I2C 通讯流程(主机接收，HEAD10R=0) 563
图 21-21. 10 位地址的 I2C 通讯流程(主机接收，HEAD10R=1) 563
图 21-22. 数据保持时间 564
图 21-23. 数据建立时间 564
图 21-24. 数据发送 566
图 21-25. 数据接收 566
图 21-26. I2C 从机初始化 569
图 21-27. I2C 从机发送编程模型(SS=0) 570
图 21-28. I2C 从机发送编程模型(SS=1) 571
图 21-29. I2C 从机接收编程模型 572
图 21-30. I2C 主机初始化 573
图 21-31. I2C 主机发送编程模型(N<=255) 574
图 21-32. I2C 主机发送编程模型(N>255) 575
图 21-33. I2C 主机接收编程模型(N<=255) 576
图 21-34. I2C 主机接收编程模型(N>255) 577
图 21-35. SMBus 主机发送器和从机接收器通信流程 580
图 21-36. SMBus 主机接收器和从机发送器通信流程 581

表索引

表 1-1. GD32E25x 系列器件的存储器映射表 27
表 1-2. 引导模式 31
表 2-1. Cortex-M23 中的 NVIC 异常类型 43
表 2-2. 中断向量表 44
表 2-3. EXTI 触发源 46
表 3-1. DMA 传输操作 55
表 3-2. 中断事件 57
表 3-3. DMA 各个通道请求 60
表 4-1. 闪存的基地址和大小 66
表 4-2. WSCNT 位域与 AHB 时钟频率之间的关系 68
表 4-3. 选项字节 74
表 4-4. OB_WP 位对应页保护 76
表 5-1. 节电模式总结 90
表 6-1. BKP 中断控制 96
表 7-1. 时钟源的选择 105
表 7-2. 深度睡眠模式下内核电压选择 106
表 8-1.GPIO 配置表 132
表 9-1. 定时器(TIMERx)分为六种类型 168
表 9-2. 由参数控制的互补输出表 183
表 9-3. 不同正交译码器模式下的计数方向 186
表 9-4. 从模式例子列表 189
表 9-5. TIMERx(x=1,2)使能配置 225
表 9-6. 由参数控制的互补输出表 290
表 9-7. 从模式例子列表 293
表 9-8. TIMERx(x=14)定时器内部互连 295
表 9-9. 由参数控制的互补输出表 326
表 10-1. 独立看门狗定时器在 40kHz(IRC40K)时的最小 / 最大超时周期 357
表 10-2. 在 72MHz(f_{PCLK1})时的最大/最小超时值 362
表 13-1. ADC 内部输入信号 377
表 13-2. ADC 输入引脚定义 377
表 13-3. 用于 ADC 规则通道的外部触发 387
表 13-4. 用于 ADC 注入通道的外部触发 388
表 13-5. 不同分辨率对应的 t_{CONV} 时间 390
表 13-6. 不同 N 和 M 组合的最大输出值(灰色值表示截断) 391
表 14-1.DAC 引脚 411
表 14-2. DAC 触发与输出 411
表 14-3.DAC 外部触发 412
表 15-1. CMP 的输入和输出总结 429
表 16-1 CLAxSIGS0 输入选择 436
表 16-2 CLAxSIGS1 输入选择 436

表 16-3 LCU 控制 437
表 19-1. USART 重要引脚描述 450
表 19-2. 停止位配置 451
表 19-3. USART 中断请求 464
表 20-1. SPI 信号描述 486
表 20-2. SPI 四线信号描述 487
表 20-3. 从机模式 NSS 功能 490
表 20-4. 主机模式 NSS 功能 491
表 20-5. SPI 运行模式 491
表 20-6. SPI 中断请求 502
表 20-7. I2S 比特率计算公式 511
表 20-8. 音频采样频率计算公式 511
表 20-9. 各种运行模式下 I2S 接口信号的方向 512
表 20-10. I2S 中断 517
表 21-1. I2C 总线术语说明(参考飞利浦 I2C 规范) 530
表 21-2. 事件状态标志位 544
表 21-3. 错误标志位 544
表 21-4. I2C 总线术语说明(参考飞利浦 I2C 规范) 560
表 21-5. 数据建立时间和数据保持时间 565
表 21-6. 可关闭通信模式 567
表 21-7. I2C 错误标志 581
表 21-8. I2C 中断事件 582
表 22-1. 寄存器功能位访问属性 597
表 22-2. 术语 597
表 23-1. 版本历史 599

1. 系统及存储器架构

GD32E25x 系列是基于 Arm® Cortex®-M23 处理器的 32 位通用微控制器。Cortex®-M23 处理器包括 AHB 总线。Cortex®-M23 处理器的所有存储器访问都是根据不同的目的和目标存储器空间在 AHB 总线上执行的。存储器的组织使用 ARMv8M 架构，预先定义的存储器映射和高达 4GB 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®-M23 处理器

Cortex®-M23 处理器是一个低功耗 32 位处理器。适用于需要一个区域优化处理器来进行深度嵌入式应用的场景。Cortex®-M23 处理器为开发人员提供了显著的好处，包括：

- 一种简单易学且易于编程的架构。
- 超低功耗，能效高的操作。
- 优秀的代码密度。
- 确定性，高性能中断处理。
- 向上兼容 Cortex®-M 处理器家族系列。

Cortex®-M23 处理器通过精简强大的指令集和广泛优化的设计提供高效处理能力，提供包括单周期乘法器和 17 周期分频器的高端处理硬件。

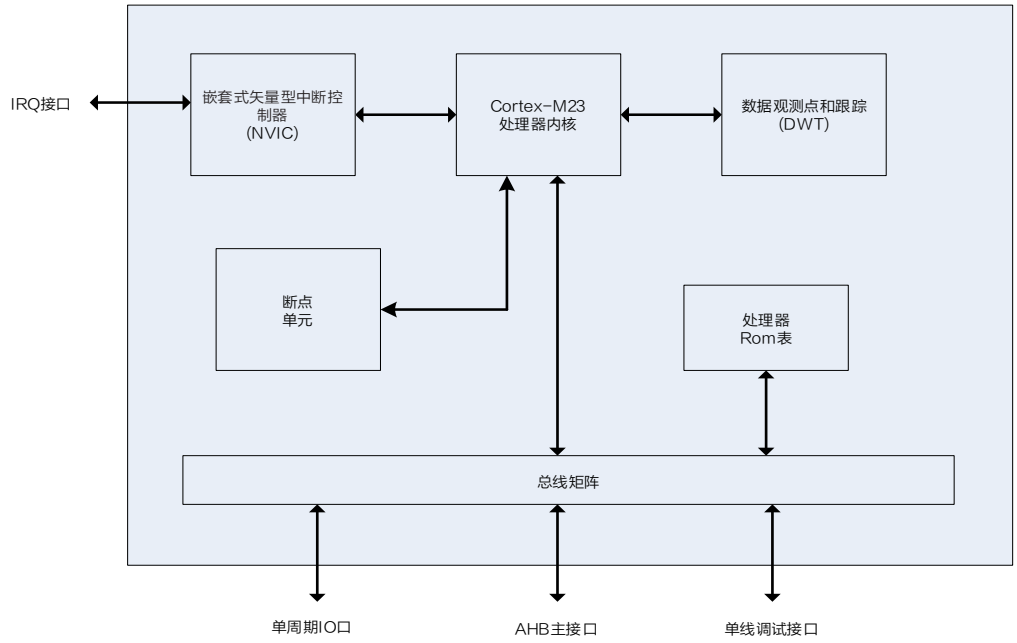
Cortex®-M23 处理器高度集成了一个可配置的嵌套矢量中断控制器(NVIC)，以提供业界领先的中断性能。

下面列出由 Cortex®-M23 提供的一些系统外设：

- 低延迟，高速外围设备输入/输出端口
- 向量表偏移寄存器
- 断点单元
- 数据监视点
- 串行线调试端口

下面的图表显示了 Cortex®-M23 处理器的框图。更多信息，请参考 Arm® Cortex®-M23 技术参考手册。

图1-1. Cortex®-M23处理器的结构框图

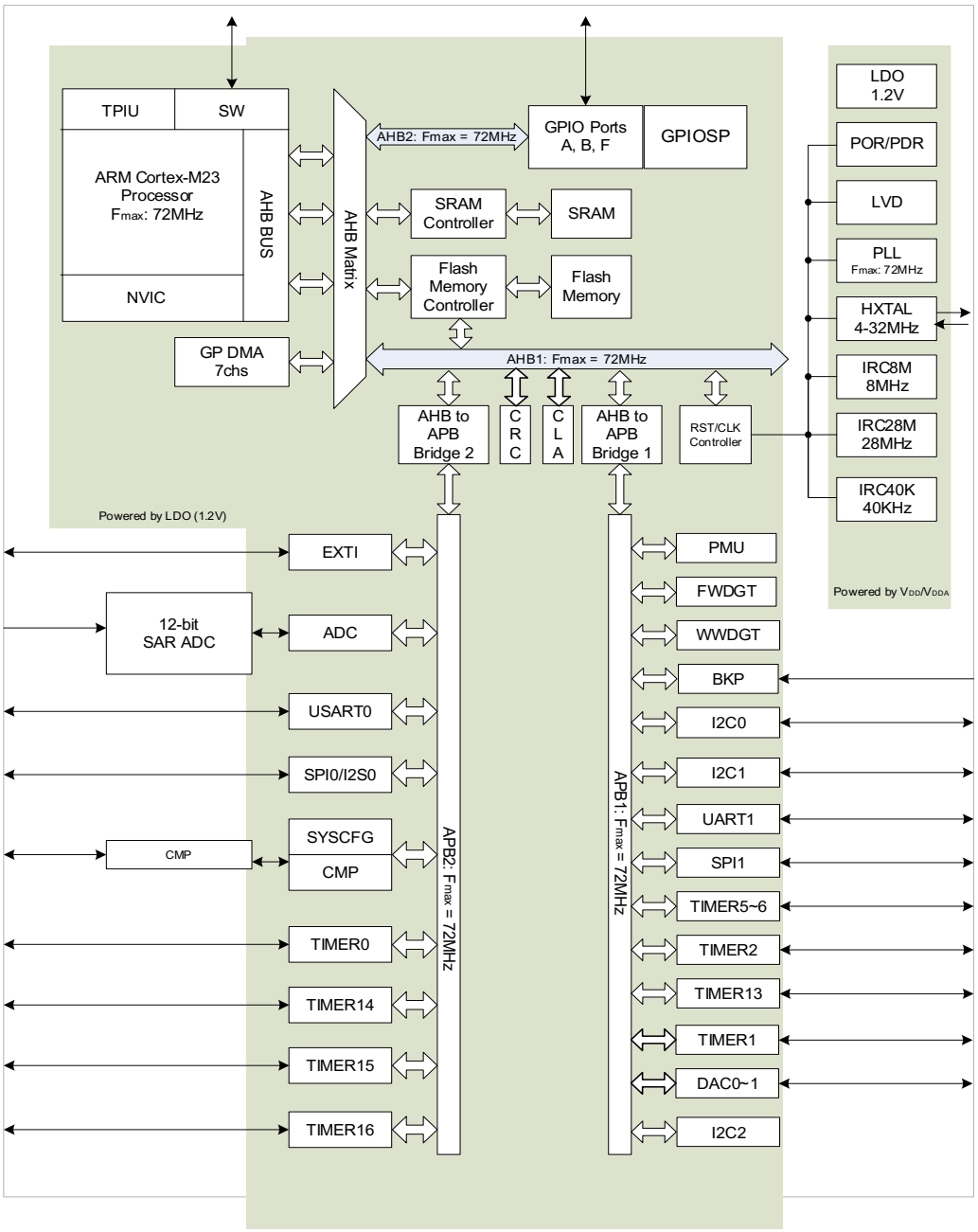


1.2. 系统架构

GD32E25x 系列器件的系统架构如下图所示。该 AHB 矩阵是一个基于 AMBA 5 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 矩阵共 2 个主机包含属于 Cortex®-M23 内核的 AHB 总线及 DMA。该 AHB 矩阵还连接了 4 个从机，分别为：FMC、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. GD32E25x系列器件的系统架构示意图



1.3. 内存映射

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 4GB 的地址空间之内。这是 Cortex®-M23 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®-M23 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®-M23 的系统外设所占用。下表显示了 GD32E25x 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. GD32E25x 系列器件的存储器映射表

预先定义的地址空间	总线	地址范围	外设
		0xE000 0000 - 0xE00F FFFF	Cortex® M23 内部外设
外部设备		0xA000 0000 - 0xDFFF FFFF	保留
外部 RAM		0x60000000 - 0x9FFFFFFF	保留
外设	AHB1	0x5004 0000 - 0x5FFF FFFF	保留
		0x5000 0000 - 0x5003 FFFF	保留
	AHB2	0x4800 1800 - 0x4FFF FFFF	保留
		0x4800 1800 - 0x4800 1BFF	保留
		0x4800 1400 - 0x4800 17FF	GPIOF
		0x4800 1000 - 0x4800 13FF	GPIOSP
		0x4800 0C00 - 0x4800 0FFF	保留
		0x4800 0800 - 0x4800 0BFF	保留
		0x4800 0400 - 0x4800 07FF	GPIOB
		0x4800 0000 - 0x4800 03FF	GPIOA
	AHB1	0x4003 8400 - 0x47FF FFFF	保留
		0x4003 8000 - 0x4003 83FF	CLA
		0x4002 4000 - 0x4003 7FFF	保留
		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 8000 - 0x4001 FFFF	保留
		0x4001 5C00 - 0x4001 7FFF	保留
		0x4001 5800 - 0x4001 5BFF	DBG
		0x4001 4C00 - 0x4001 57FF	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

预先定义的地址空间	总线	地址范围	外设
		0x4001 0000 - 0x4001 03FF	SYSCFG + CMP
	APB1	0x4000 D000 - 0x4000 FFFF	保留
		0x4000 CC00 - 0x4000 CFFF	DAC1
		0x4000 C800 - 0x4000 CBFF	保留
		0x4000 C400 - 0x4000 C7FF	保留
		0x4000 C000 - 0x4000 C3FF	I2C2
		0x4000 8000 - 0x4000 BFFF	保留
		0x4000 7C00 - 0x4000 7FFF	保留
		0x4000 7800 - 0x4000 7BFF	保留
		0x4000 7400 - 0x4000 77FF	DAC0
		0x4000 7000 - 0x4000 73FF	PMU
		0x4000 6400 - 0x4000 6FFF	保留
		0x4000 6000 - 0x4000 63FF	保留
		0x4000 5C00 - 0x4000 5FFF	保留
		0x4000 5800 - 0x4000 5BFF	I2C1
		0x4000 5400 - 0x4000 57FF	I2C0
		0x4000 4800 - 0x4000 53FF	保留
		0x4000 4400 - 0x4000 47FF	UART1
		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	BKP
		0x4000 2400 - 0x4000 27FF	保留
		0x4000 2000 - 0x4000 23FF	TIMER13
		0x4000 1800 - 0x4000 1FFF	保留
		0x4000 1400 - 0x4000 17FF	TIMER6
		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	选项字节
		0x1FFF EC00 - 0x1FFF F7FF	系统存储器
		0x1FFF 7400 - 0x1FFF EBFF	保留
		0x1FFF 7000 - 0x1FFF 73FF	OTP 字节
		0x0801 0000 - 0x1FFF 6FFF	保留

预先定义的地址空间	总线	地址范围	外设
		0x0800 0000 - 0x0800 FFFF	主闪存
		0x0001 0000 - 0x07FF FFFF	保留
		0x00000000 - 0x0000FFFF	别名为闪存或系统存储器

1.3.1. 片上 SRAM 存储器

GD32E25x 系列包含最多 8KB 的片上 SRAM，起始地址为 0x2000 0000。它支持字节、半字(16 比特)和字(32 比特)访问。

奇偶校验检查

为了提高存储器的稳定性，支持奇偶校验。用户可以通过用户选项字节中的 SRAM_PARITY_ECC_CHECK 位启用奇偶校验功能(参见选项字节描述)。启用后，如果奇偶校验失败，将生成 SRAM_PCEF，该 SRAM 奇偶校验错误标志实现在系统配置寄存器 1(SYSCFG_CFG1) 中。如果系统配置寄存器 1(SYSCFG_CFG1) 中的 SRAM_PARITY_ERROR_LOCK 控制位设置为 1，则错误标志可以连接到 TIMER0/TIMER14/TIMER15/TIMER16 的中止输入。如果需要查找 SRAM 的奇偶校验错误地址，可以通过系统配置寄存器 0(SYSCFG_CFG0)中的 SRAM_ERRADDR[10:0]位域来查找。

SRAM 的实际数据宽度为 36 位，包括 32 位数据位和 4 位校验位(每字节 1 位)。在写入时，计算校验位并存储到 SRAM 中。在读取时，也使用存储在 SRAM 中的数据计算校验位。计算出的校验位与写入访问期间计算并存储的校验位进行比较。如果它们不同，则校验失败。

注意:启用 SRAM 奇偶校验时，建议在代码开始时通过软件初始化整个 SRAM 内存，以避免在读取未初始化位置时出现奇偶校验错误。此外，只有在闪存就绪时才会进行 SRAM 操作。在系统复位和电源复位后，在 SYSCFG 和选项字节更新期间，SRAM 功能无法进行读写。

ECC

在读写 SRAM 时，它支持 7 位 ECC 功能。它可以纠正 1 位错误并检测多位错误。

读之前必须先写入，否则很可能会导致 ECC 错误。非对齐的读操作会按照 32 比特的读操作来执行。非对齐的写操作会产生一个读改写的流程。例如，16 比特写，首先会先读 16 比特，再和需要写入的 16 比特一起写入。所以初始化 SRAM 时，只能按照 32 位的来写入。

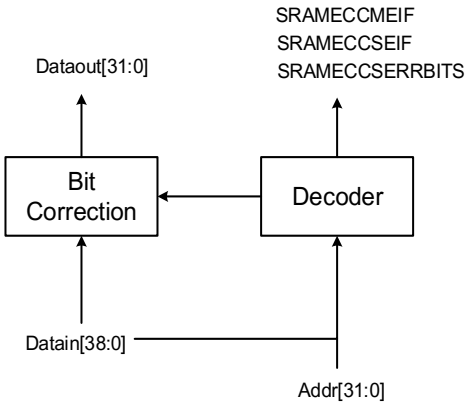
ECC 模块由编码器和解码器两部分构成：

编码器：在进行 SRAM 写操作时，会产生一个 7 比特的 ECC 码，和数据一起写入 SRAM。

解码器：在进行 SRAM 读操作时，使用与编码器相同的算法，解码生成一个 7 比特的 ECC 码。ECC 码包括 ECC 错误状态和 32 位数据中哪位存在单比特位错误的信息。通过 SYSCFG_SRAM_ECC 寄存器中的 ECCSERRBITS0 可查询是否存在 ECC 单比特可校正错误位。如果需要查找 SRAM 上发生的 ECC 事件的最后一个地址，可以通过 SYSCFG_SRAM_ECC 寄存器中的 ECCEADDR0 进行查找。

解码器如*图 1-3. ECC 解码器示意图*。

图1-3. ECC解码器示意图



EEIC

EEIC(ECC Error Interrupt Control)模块提供了 ECC 错误状态管理和 ECC 中断配置的功能。

单比特可纠错事件

当检测到发生了单比特可纠错的事件时，EEIC：

（1） SYSCFG_STAT 寄存器中的 ECCSEIF0 位置位，软件写 1 可以清除。

（2） SYSCFG_SRAM_ECC 寄存器中的 ECCEADDR0 记录发生单比特可纠错 ECC 事件的地址。

多比特(两比特)不可纠错事件

当检测到发生了多比特(两比特)不可纠错的事件时，EEIC：

（1） SYSCFG_STAT 寄存器中的 ECCMEIF0 位置位，软件写 1 可以清除。

（2） SYSCFG_SRAM_ECC 寄存器中的 ECCEADDR0 记录发生两比特不可纠错 ECC 事件的地址。

单比特可纠错中断

在 SYSCFG_SRAM_ECC 寄存器中设置 ECCSEIE0 位，当检测到一个单比特可纠正错误事件时，将产生一个相应的中断 IRQ 35。

多比特(两比特)不可纠错事件

在 SYSCFG_SRAM_ECC 寄存器中设置 ECCMEIE0 位。当检测到一个多比特(两比特)不可纠错错误事件时，将产生一个 NMI 中断。

1.3.2. 片上闪存存储器

这些设备提供高密度的片上闪存，其结构如下：

- 高达64 KB的片上闪存，用于存储指令/数据。
- 3K字节信息块用于引导程序。
- 16字节选项字节块，用于用户应用程序需求。
- 1KB OTP(一次性编程)块，用于用户数据存储。

请参阅*闪存控制器(FMC)*部分以获取更多详细信息。

1.4. 引导配置

GD32E252xx 系列提供了三种可选用的启动源，这些启动源可以通过用户选项字节中的 BOOT1_n 位以及 BOOT0 引脚来选择(参考*选项字节说明*描述)。BOOT0 引脚上的值在系统复位后第 4 个 SYSCLK 上升沿被锁定。用户需要在上电复位或系统复位后设置 BOOT1_n 和 BOOT0，以选择所需的启动源。GD32E252xx 系列默认启动源为主闪存。

GD32E251xx 系列提供了三种可选用的启动源，这些启动源可以通过用户选项字节中的 BOOT1_n 位和 BOOT0_n 位来选择。用户在上电复位或系统复位后需要设置 BOOT1_n 和 BOOT0_n，以选择所需的启动源。；GD32E251xx 系列默认启动源为系统存储器。

详情如下表所示。

表 1-2. 引导模式

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

1. BOOT1 值与 BOOT1_n 值相反；
2. 对于 GD32E252xx 系列，BOOT0 值映射到 BOOT0 引脚；
3. 对于 GD32E251xx 系列，BOOT0 值与 BOOT0_n 值相反。

上电序列或系统复位后，ARM® Cortex®-M23 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。

根据选定的启动源，主闪存(原始内存空间从 0x0800 0000 开始)或系统内存(原始内存空间从 0x1FFF EC00 开始)被映射到从地址 0x0000 0000 开始的启动内存空间。当选择芯片上的 SRAM 作为启动源，其内存空间从 0x2000 0000 开始，在应用程序初始化代码中，用户必须使用 NVIC 异常表和偏移寄存器重新定位 SRAM 中的向量表。

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 I2C 接口工作。

注意：为确保系统的可靠性和稳定性，在硬件设计阶段，应查阅对应型号的数据手册的 Boot mode 章节，确认 Bootloader 所使用的引脚上避免连接可能产生干扰信号的外设（如传感器输出、看门狗芯片、按键等），以免影响 Bootloader 的正常功能，和对所连接的外设产生异常输出。

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(X 表示 BOOT_MODE[1:0])

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
SRAM_ERRADDR[10:0]											保留	PB9FMPEN	保留	PB7FMPEN	PB6FMPEN
r												rw		rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留			TIMER16_DMA_RMP	TIMER15_DMA_RMP	USART0_RX_DMA_RMP	USART0_TX_DMA_RMP	ADC_DMA_RMP	保留	BOOT0_PB9_RMP	保留	PA11_PA12_RMP	保留		BOOT_MODE[1:0]	
			rw	rw	rw	rw	rw		rw		rw			r	

位/位域	名称	描述
31:21	SRAM_ERRADDR[10:0]	SRAM奇偶校验错误地址. **注意：**物理SRAM字地址=SRAM基地址/4 + SRAM_ERRADDR
20	保留	必须保持复位值。
19	PB9FMPEN	I2C的Fm+或IFRP模式在PB9引脚上启用 此位控制I2CFm+ 或IFRP模式，引脚的速度控制将被旁路。 0：禁用PB9引脚上的模式 1：启用PB9引脚上的模式
18	保留	必须保持复位值。
17	PB7FMPEN	I2C的Fm+或IFRP模式在PB7引脚上启用 此位控制I2CFm+或IFRP模式，引脚的速度控制将被旁路。 0：禁用PB7引脚上的模式 1：启用PB7引脚上的模式
16	PB6FMPEN	I2C的Fm+或IFRP模式在PB6引脚上启用 此位控制I2CFm+或IFRP模式，引脚的速度控制将被旁路。 0：禁用PB6引脚上的模式 1：启用PB6引脚上的模式
15:13	保留	必须保持复位值。
12	TIMER16_DMA_RMP	TIMER 16 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	TIMER 15 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	保留	必须保持复位值。
6	BOOT0_PB9_RMP	BOOT0/PB9重映射位控制位 该位由软件设置和清除。当PIN_RSTMD置1时，除POR外的所有复位事件都会保留该位。它控制小引脚数封装上BOOT0/ PB9对BOOT0引脚的的映射。 当PIN_RSTMD和BOOT0_PB9_RMP都置1时，系统复位后BOOT0功能由硬件连接到0。在这种情况下，系统将从主闪存启动，而不考虑BOOT0引脚的输入值。 0：不重映射(BOOT0功能映射在BOOT0引脚上) 1：重映射(PB9功能映射在BOOT0引脚上) **注意**：对于GD32E251xx系列，该bit保留
5	保留	必须保持复位值。
4	PA11_PA12_RMP	小封装PA11/PA12重映射位控制位(28和20pin封装) 该位由软件设置和清除。当PIN_RSTMD置1时，除POR外的所有复位事件都会保留该位。它控制PA9/10或PA11/12针对在小引脚数封装上的映射。 0：不重映射(PA9/10映射在引脚上) 1：重映射(PA11/12映射而不是PA9/10)
3:2	保留	必须保持复位值。
1:0	BOOT_MODE[1:0]	引导模式 对于 GD32E252xx系列，bit0映射到BOOT0引脚；对于GD32E251xx系列，bit0的值与BOOT0_n选项位的值相反；bit1的值与BOOT1_n选项位的值相反。 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
重置值：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]	EXTI3源选择 X000：PA3 引脚 X001：PB3 引脚 X010-X111：PB3 引脚(SDCTL=0)，保留(SDCTL=1) EXTI_SDCTL中的SDCTL位决定了X010-X111的源选择。
11:8	EXTI2_SS[3:0]	EXTI2源选择 X000：PA2引脚 X001：PB2引脚 X010-X111：PB2引脚(SDCTL=0)，保留(SDCTL=1) EXTI_SDCTL中的SDCTL位决定了X010-X111的源选择。
7:4	EXTI1_SS[3:0]	EXTI1源选择 X000：PA1引脚 X001：PB1引脚 X010-X100：PF1 引脚(SDCTL=0)，保留(SDCTL=1) X110-X111：PF1 引脚(SDCTL=0)，保留(SDCTL=1) 除了X101，EXTI_SDCTL中的SDCTL位决定了X011-X111的源选择。
3:0	EXTI0_SS[3:0]	EXTI0源选择 X000：PA0引脚 X001：PB0引脚 X010-X100：PF0引脚(SDCTL=0)，保留(SDCTL=1) X110-X111：PF0引脚(SDCTL=0)，保留(SDCTL=1) 除了 X101，EXTI_SDCTL 中的 SDCTL 位决定了 X011-X111 的源选择。

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]	EXTI7源选择 X000：PA7引脚 X001：PB7引脚 X010-X111：保留
11:8	EXTI6_SS[3:0]	EXTI6源选择 X000：PA6引脚 X001：PB6引脚 X010-X111：保留
7:4	EXTI5_SS[3:0]	EXTI5源选择 X000：PA5引脚 X001：PB5引脚 X010-X111：PB5 pin(SDCTL=0)，保留(SDCTL=1) EXTI_SDCTL中的SDCTL位决定了X010-X111的源选择。
3:0	EXTI4_SS[3:0]	EXTI4源选择 X000：PA4引脚 X001：PB4引脚 X010-X111：PB4引脚(SDCTL=0)，保留(SDCTL=1) EXTI_SDCTL中的SDCTL位决定了X010-X111的源选择。

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			

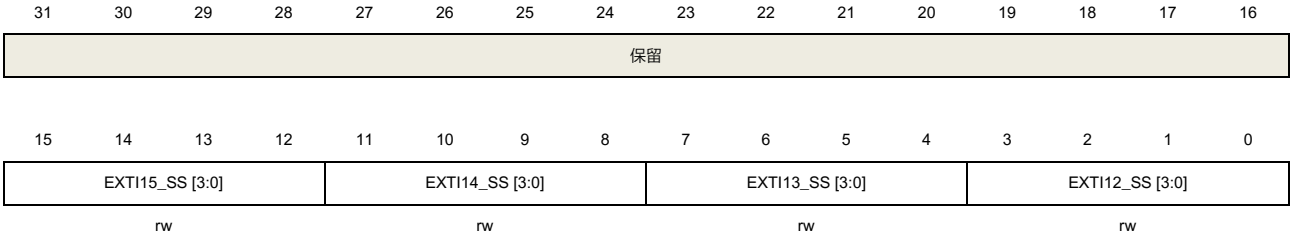
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:12	EXTI11_SS[3:0]	EXTI11源选择

		X000：PA11引脚 X001：PB3引脚 X010-X111：PB3引脚(SDCTL=0)，保留(SDCTL=1) EXTI_SDCTL中的SDCTL位决定了X010-X111的源选择。
11:8	EXTI10_SS[3:0]	EXTI10源选择 X000：PA10引脚 X001：PB10引脚 X010-X111：PB10引脚(SDCTL=0)，保留(SDCTL=1) EXTI_SDCTL中的SDCTL位决定了X010-X111的源选择。
7:4	EXTI9_SS[3:0]	EXTI9源选择 X000：PA9引脚 X001：PB9引脚 X010-X111：PB9引脚(SDCTL=0)，保留(SDCTL=1) EXTI_SDCTL中的SDCTL位决定了X010-X111的源选择。
3:0	EXTI8_SS[3:0]	EXTI8源选择 X000：PA8引脚 X001：PB8引脚 X010-X111：PB8引脚(SDCTL=0)，保留(SDCTL=1) EXTI_SDCTL中的SDCTL位决定了X010-X111的源选择。

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。



位/位域	名称	描述
31:16	保留	必须保持复位值。
15:12	EXTI15_SS[3:0]	EXTI15源选择 X000：PA15引脚 X001：PB3引脚 X010-X111：PB3引脚(SDCTL=0)，保留(SDCTL=1) EXTI_SDCTL中的SDCTL位决定了X010-X111的源选择。
11:8	EXTI14_SS[3:0]	EXTI14源选择 X000：PA14引脚

		X001-X111：保留
7:4	EXTI13_SS[3:0]	EXTI13源选择 X000：PA13引脚 X001-X111：保留
3:0	EXTI12_SS[3:0]	EXTI12源选择 X000：PA12引脚 X001-X111：保留

1.5.6. 系统配置寄存器 1(SYSCFG_CFG1)

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							PIN_RSTMD	保留							VREF_EN
							rw								rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留							SRAM_PCEF	保留					LVD_LOCK	SRAM_PARITY_ERROR_LOCK	LOCKUP_LOCK
							rc_w1						rw	rw	rw

位/位域	名称	描述
31:25	保留	必须保持复位值。
24	PIN_RSTMD	引脚复位模式 这个位控制GPIO引脚配置(GPIOx_CTL、GPIOx_OMODE、GPIOx_OSPD、GPIOx_PUD、GPIOx_OCTL、BOOT0_PB9_RMP(仅限GD32E252xx系列)和PA11_PA12_RMP)在发生任何复位事件时是否重置。除POR外，所有复位事件都将保留此位。在POR事件之后，这个位被重置为0。 0：GPIO引脚配置在发生复位事件时复位。 1：GPIO引脚配置将在除POR事件外的任何复位事件中保持状态。
23:17	保留	必须保持复位值。
16	VREF_EN	VREF使能 该位由软件设置和清除。当设置任何DACx DRSTMDx时，除POR外，所有复位事件都将保留该位。它控制是否使能VREF功能。 0：VREF禁止 1：VREF使能
15:9	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®-M23 LOCKUP输出锁定 该位由软件置1，在系统复位时才能清零。 0：Cortex®-M23 LOCKUP输出从TIMER0/14/15/16的break输入端断开 1：Cortex®-M23 LOCKUP输出与TIMER0/14/15/16的break输入端连接

1.5.7. 系统配置状态寄存器(SYSCFG_STAT)

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															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留														ECCSEIF	ECCMEIF
														0	0
														rc_w1	rc_w1

位/位域	名称	描述
31:2	保留	必须保持复位值。
1	ECCSEIF0	SRAM单比特校正事件标志 软件可以通过写入1来清除它。 0：未检测到SRAM单比特校正事件 1：检测到SRAM单比特校正事件
0	ECCMEIF0	SRAM多比特不可校正事件标志

软件可以通过写入1来清除它。

0：未检测到SRAM多比特不可校正事件

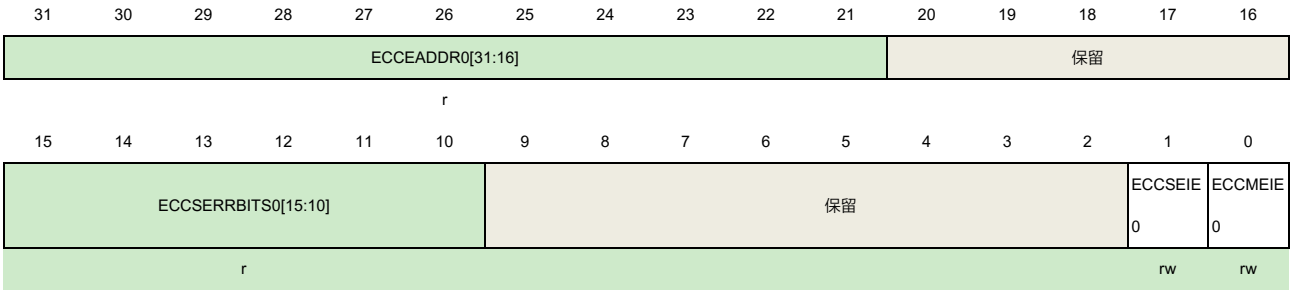
1：检测到SRAM多比特不可校正事件

1.5.8. SRAM_ECC 寄存器(SYSCFG_SRAM_ECC)

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
31:21	ECCEADDR0[31:21]	该值表明在SRAM上发生的ECC事件的最后一个地址 物理SRAM字地址=SRAM 基地址/4+ECCEADDR1。
20:16	保留	必须保持复位值。
15:10	ECCSERRBITS0[15:10]	指示错误位 该值表明是否存在ECC单比特可校正错误位 0：无错误 1：bit 0 … 39：bit 39
9:2	保留	必须保持复位值。
1	ECCSEIE0	SRAM单比特可纠正中断使能 0：SRAM单比特可纠正中断被禁用。 1：SRAM单比特可纠正中断被启用。
0	ECCMEIE0	SRAM多比特不可纠正中断使能 0：SRAM不可纠正中断被禁用。 1：SRAM不可纠正中断被启用。

1.5.9. IRQ 延迟寄存器 (SYSCFG_CPU_IRQ_LAT)

地址偏移：0x100

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								IRQ_LATENCY[7:0]							
								rw							

位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	IRQ_LATENCY[7:0]	IRQ_LATENCY 指定在 NVIC 中挂起的中断与在 AHB-Lite 接口上发出的该中断的向量读取之间的最小周期数。 如果将 IRQ_LATENCY 延迟设置为 0，则尽可能快地中断中断。 对于非零值，Cortex®-M23 处理器确保在 NVIC 中的中断和正在执行的中断的向量提取之间存在最小的 IRQ_LATENCY + 1 个 hclk 周期。

1.6. 设备电子签名

该设备的电子签名包含存储密度信息和 96 位唯一设备 ID。它存储在 Flash 存储器的信息块中。96 位唯一设备 ID 对于任何设备都是唯一的。它可以用作序列号，或作为安全密钥的一部分等。

1.6.1. 内存密度信息

基础地址：0x1FFF F7E0
该值是原厂设定的，不能由用户修改。

该寄存器必须通过字(32 位)来访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SRAM_DENSITY[15:0]															
r															

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

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]	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。

1.6.2. 唯一设备标识符(96 位)

基址：0x1FFF F7AC
该值是工厂预设的，用户永远无法更改。

这个寄存器必须通过字(32 位)来访问。

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

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

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。

这个寄存器需要通过字(32 位)来访问。

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]	唯一设备ID

基地址：0x1FFF F7B4
这个值是工厂预编程的，用户永远无法更改。

此寄存器必须通过字(32 位)来访问。

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. 中断/事件控制器(EXTI)

2.1. 简介

Cortex®-M23集成了嵌套式矢量型中断控制器NVIC(Nested Vectored Interrupt Controller)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-M23技术参考手册》。

EXTI(中断/事件控制器)包括24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、下降沿触发和任意沿触发。EXTI中的每一个边沿检测电路都可以独立配置和屏蔽。

2.2. 主要特征

- Cortex-M23系统异常；
- 多达36种可屏蔽的外设中断；
- 2位中断优先级配置位-4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24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。
- 外部引脚EXTI默认源可配置为0(无效)

2.3. 功能说明

ARM Cortex®-M23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。下表列出了Cortex®-M23中的NVIC异常类型。

表 2-1. Cortex-M23 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

异常类型	向量编号	优先级(a)	向量地址	描述
存储器管理	4	可编程设置	0x0000_0010	存储器管理
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
SVCall 服务调用	11	可编程设置	0x0000_002C	通过 SWI 指令实现系统服务调用
调试监控	12	可编程设置	0x0000_0030	调试监视器
-	13	-	0x0000_0034	保留
PendSV 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

SysTick校准值设为9000，SysTick时钟频率配置为HCLK/8，此时若HCLK时钟被配置为72MHz，则SysTick中断会1ms响应一次。

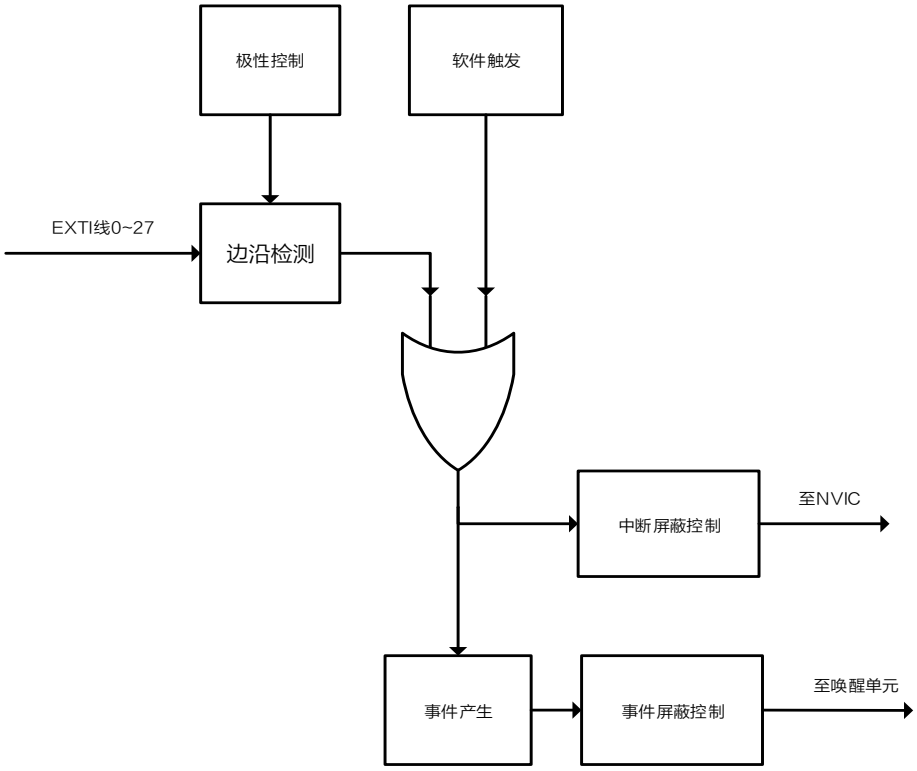
表 2-2. 中断向量表

中断编号	向量编号	外设中断描述	向量地址
IRQ 0	16	窗口看门狗中断	0x0000_0040
IRQ 1	17	连接到 EXTI 线的 LVD 中断	0x0000_0044
IRQ 2	18	BKP 侵入中断	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	EXTI 线 20/22 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 EXTI 线 21 CMP 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 中断，DAC0、DAC1 全局中断	0x0000_0084
IRQ 18	34	TIMER6 中断	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

中断编号	向量编号	外设中断描述	向量地址
IRQ 25	41	SPI0 全局中断	0x0000_00A4
IRQ 26	42	SPI1 全局中断	0x0000_00A8
IRQ 27	43	USART0 全局中断	0x0000_00AC
IRQ 28	44	UART1 全局中断	0x0000_00B0
IRQ 29	45	I2C2 事件中断	0x0000_00B4
IRQ 30	46	I2C2 错误中断	0x0000_00B8
IRQ 31	47	GPIO 端口不匹配中断	0x0000_00BC
IRQ 32	48	I2C0 错误中断	0x0000_00C0
IRQ 33	49	通过 EXTI18，I2C2 唤醒中断	0x0000_00C4
IRQ 34	50	I2C1 错误中断	0x0000_00C8
IRQ 35	51	ECC 单 bit 错误中断	0x0000_00CC
IRQ 36	52	保留	0x0000_00D0
IRQ 37	53	保留	0x0000_00D4
IRQ 38	54	保留	0x0000_00D8
IRQ 39	55	CLA 全局中断	0x0000_00DC
IRQ 40-41	56-57	保留	0x0000_00E0-0x0000_00E4
IRQ 42	58	保留	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	保留	0x0000_010C
IRQ52-66	68-82	保留	0x0000_0110-0x0000_0148
IRQ67	83	保留	0x0000_014C

2.4. 外部中断及事件(EXTI)结构框图

图2-1. EXTI结构框图



2.5. 外部中断及事件功能概述

EXTI包含多达24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，具体细节参考*表2-3. EXTI触发源*。通过配置SYSCFG模块的SYSCFG_EXTISSx寄存器，所有的GPIO管脚都可以被选作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®-M23内核完全支持等待中断(WFI)，等待事件(WFE)和发送事件(SEV)指令。唤醒中断控制器(WIC)可以让处理器和NVIC进入功耗极低的省电模式，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。

表 2-3. EXTI 触发源

EXTI 线编号	触发源
0	PA0 / PB0 / PF0
1	PA1 / PB1 / PF1

EXTI 线编号	触发源
2	PA2 / PB2
3	PA3 / PB3
4	PA4 / PB4
5	PA5 / PB5
6	PA6 / PB6 / PF6
7	PA7 / PB7 / PF7
8	PA8 / PB8
9	PA9 / PB9
10	PA10 / PB10
11	PA11 / PB3
12	PA12
13	PA13
14	PA14
15	PA15 / PB3
16	LVD(低电压检测)
17	保留
18	I2C2 唤醒
19	Tamper
20	PB9 / PB8 / PF0 / PB5 / PB4 / PB3 / PA15
21	CMP 输出
22	PB9 / PB8 / PF0 / PB5 / PB4 / PB3 / PA15
23	保留
24	GPIO 不匹配唤醒
25	USART0 唤醒

2.6. EXTI 寄存器

EXTI基地址：0x4001 0400

2.6.1. 中断使能寄存器(EXTI_INTEN)

地址偏移：0x00
复位值：0x0E80 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						INTEN25	INTEN24	保留	INTEN22	INTEN21	INTEN20	INTEN19	INTEN18	保留	INTEN16
						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
31:26	保留	必须保持复位值
25:24	INTENx	中断使能位x(x=24..25) 0：第x线中断被禁用 1：第x线中断被使能
23	保留	必须保持复位值
22:18	INTENx	中断使能位x(x=18..22) 0：第x线中断被禁用 1：第x线中断被使能
17	保留	必须保持复位值
16:0	INTENx	中断使能位x(x=0..16) 0：第x线中断被禁用 1：第x线中断被使能

2.6.2. 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						EVEN25	EVEN24	保留	EVEN22	EVEN21	EVEN20	EVEN19	EVEN18	保留	EVEN16
						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
31:26	保留	必须保持复位值
25:24	EVENx	事件使能位x(x=24..25) 0：第x线事件被禁用 1：第x线事件被使能
23	保留	必须保持复位值
22:18	EVENx	事件使能位x(x=18..22) 0：第x线事件被禁用 1：第x线事件被使能
17	保留	必须保持复位值
16:0	EVENx	事件使能位x(x=0..16) 0：第x线事件被禁用 1：第x线事件被使能

2.6.3. 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						RTEN25	RTEN24	保留	RTEN22	RTEN21	RTEN20	RTEN19	RTEN18	保留	RTEN16
						rw	rw		rw	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
31:26	保留	必须保持复位值
25:24	RTENx	上升沿触发使能(x=24..25) 0：第x线上升沿触发无效 1：第x线上升沿触发有效(中断/事件请求)
23	保留	必须保持复位值
22:18	RTENx	上升沿触发使能(x=18..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/事件请求)
17	保留	必须保持复位值
16:0	RTENx	上升沿触发使能(x=0..16) 0：第x线上升沿触发无效 1：第x线上升沿触发有效(中断/事件请求)

2.6.4. 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						FTEN25	FTEN24	保留	FTEN22	FTEN21	FTEN20	FTEN19	FTEN18	保留	FTEN16
						rw	rw		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
31:26	保留	必须保持复位值
25:24	FTENx	下降沿触发使能(x=24..25) 0：第x线下降沿触发无效 1：第x线下降沿触发有效(中断/事件请求)
23	保留	必须保持复位值
22:18	FTENx	下降沿触发使能(x=18..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/事件请求)
17	保留	必须保持复位值
16:0	FTENx	下降沿触发使能(x=0..16) 0：第x线下降沿触发无效 1：第x线下降沿触发有效(中断/事件请求)

2.6.5. 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						SWIEV25	SWIEV24	保留	SWIEV22	SWIEV21	SWIEV20	SWIEV19	SWIEV18	保留	SWIEV16
						rw	rw		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
31:26	保留	必须保持复位值
25:24	SWIEVx	中断/事件软件触发(x=24..25) 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
23	保留	必须保持复位值
22:18	SWIEVx	中断/事件软件触发(x=18..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
17	保留	必须保持复位值
16:0	SWIEVx	中断/事件软件触发(x=0..16) 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

2.6.6. 挂起寄存器(EXTI_PD)

地址偏移：0x14
复位值：未定义

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						PD25	PD24	保留	PD22	PD21	PD20	PD19	PD18	保留	PD16
						rc_w1	rc_w1		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
31:26	保留	必须保持复位值
25:24	PDx	中断挂起状态(x=24..25) 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

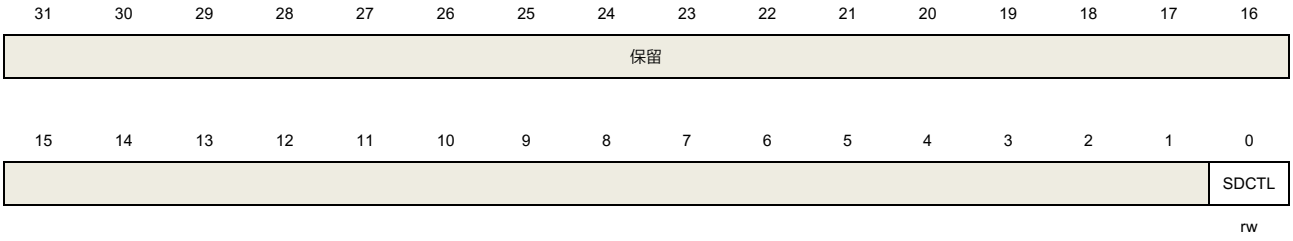
23	保留	必须保持复位值
22:18	PDx	中断挂起状态(x=18..22) 0：EXTI线x没有被触发 1：EXTI线x被触发 对这些位写1，可将其清0
17	保留	必须保持复位值
16:0	PDx	中断挂起状态(x=0..16) 0：EXTI线x没有被触发 1：EXTI线x被触发 对这些位写1，可将其清0

2.6.7. 默认源控制寄存器(EXTI_CTL)

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
31:1	保留	必须保持复位值
0	SDCTL	0：EXTI默认源是最后一组引脚 1：EXTI默认源无效

3. 直接存储器访问控制器(DMA)

3.1. 简介

DMA 控制器提供了一种硬件传输方式，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。

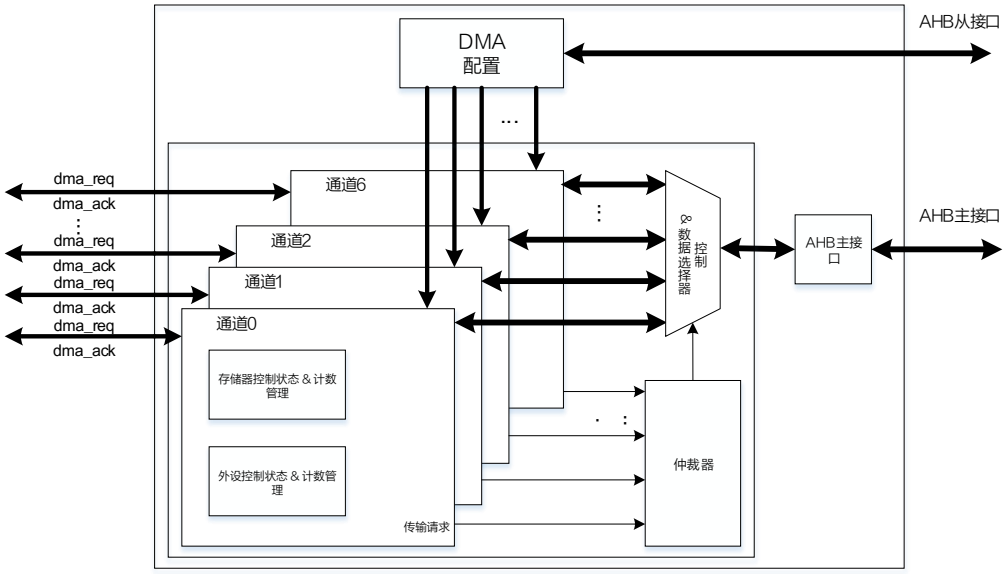
DMA 控制器和 Cortex®-M23 内核共享系统总线。当 DMA 和 CPU 访问同样的地址空间时，DMA 访问可能会阻挡 CPU 访问系统总线几个总线周期。总线矩阵中实现了循环仲裁算法来分配 DMA 与 CPU 的访问权，它可以确保 CPU 得到至少一半的系统总线带宽。

3.2. 主要特征

- 传输数据长度可编程配置，最大到 65536；
- 支持 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 DMA 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。

3.3. 结构框图

图 3-1. DMA 结构框图



由*图 3-1. 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，用于存储器访问和外设访问；
- 仲裁器进行 DMA 请求的优先级管理；
- 通道管理用于控制数据/地址选择和数据计数。

3.4. 功能说明

3.4.1. 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 3-1. DMA 传输操作*。

表 3-1. 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[15:0] @0x0 2: Write B5B4[15:0] @0x2 3: Write B9B8[15:0] @0x4 4: Write BDBC[15:0] @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，该位域控制传输的次数。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 DMA 传输将分两种情况：
  - 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，对相应通道的 DMA_CHxCNT、DMA_CHxPADDR 或 DMA_CHxMADDR 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，之后未对相应通道的 DMA_CHxCNT、DMA_CHxPADDR 或 DMA_CHxMADDR 寄存器进行操作前便使能 DMA 通道，则不会触发任何 DMA 传输。

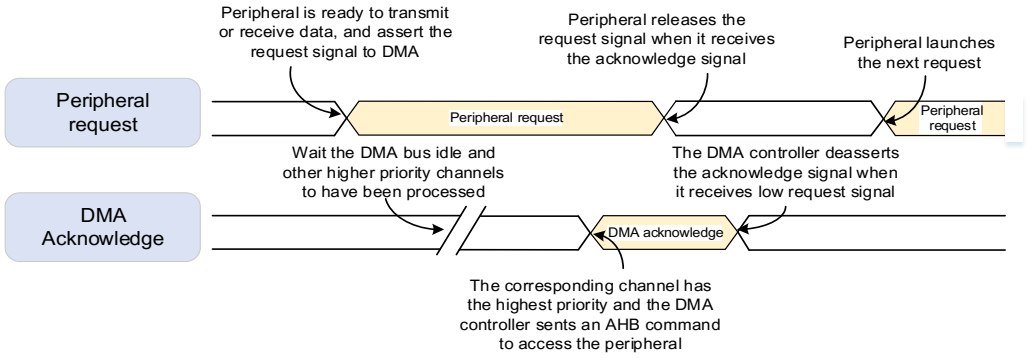
3.4.2. 外设握手

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3-2. 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 3-2. 握手机制



3.4.3. 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。

3.4.4. 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位分别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。

3.4.5. 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。

3.4.6. 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传输才会停止。

3.4.7. 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。

3.4.8. 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。

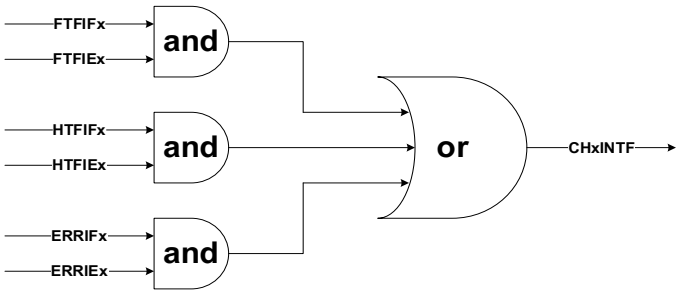
每一个中断事件在DMA_INTF寄存器中有专用的标志位，在DMA_INTC寄存器中有专用的清除位，在DMA_CHxCTL寄存器中有专用的使能位。*表3-2. 中断事件*描述了其对应关系。

表 3-2. 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*图3-3. DMA中断逻辑图*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 3-3. DMA 中断逻辑图



注意："x" 表示通道数(对应x=0…6)

3.4.9. DMA 请求映射

多个外设请求被映射到同一个 DMA 通道，这些请求信号经过逻辑或后进入 DMA。详细外设请求映射，参考*图 3-4. DMA 请求映射图。*通过配置相应外设的寄存器，每个外设请求均可以独立的开启或关闭。用户必须确保在同一时间，同一个通道上仅有一个外设的请求被开启。*表 3-3. DMA 各个通道请求*列出 DMA 各个通道支持的外设请求。

图 3-4. DMA 请求映射图

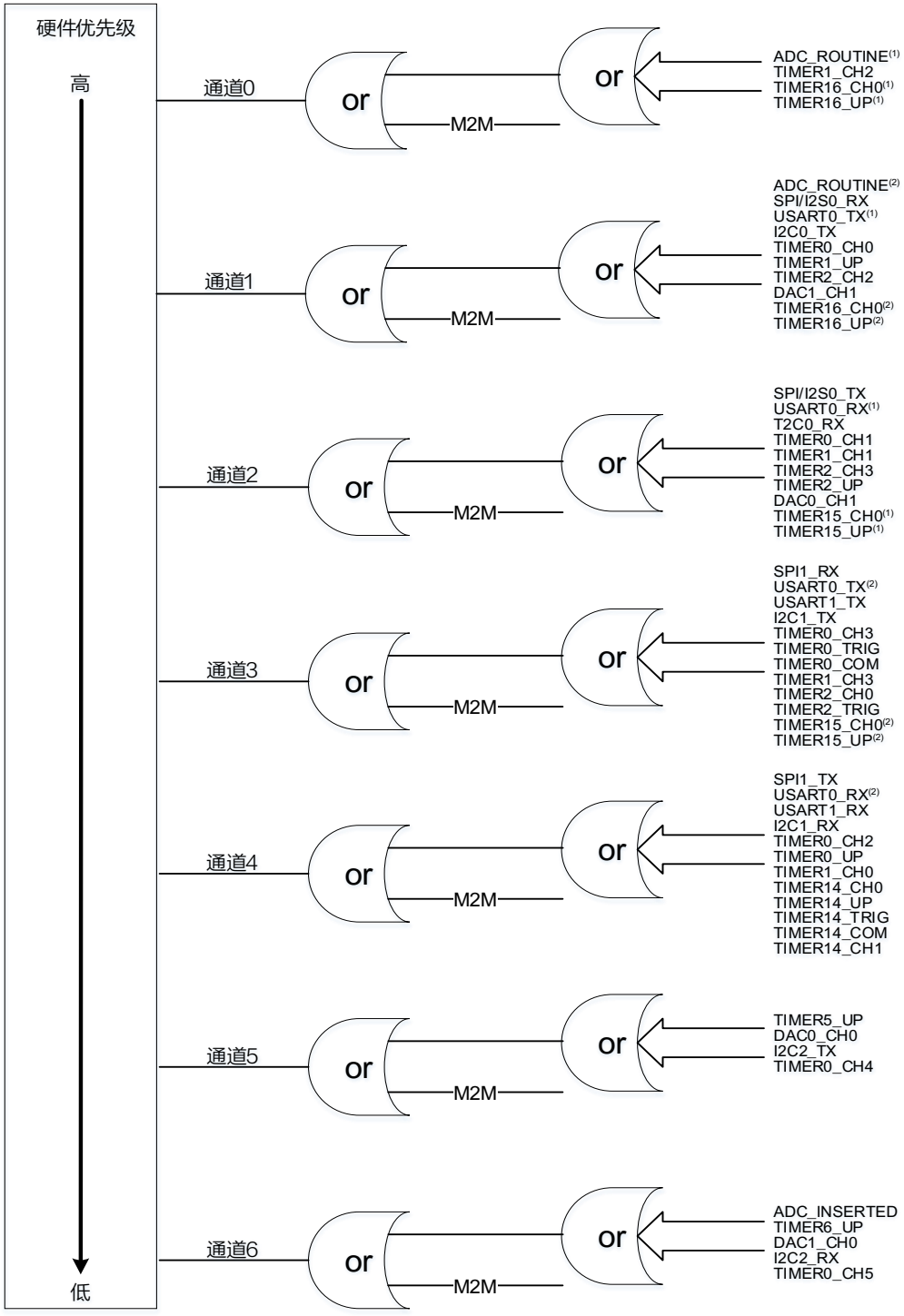


表 3-3. DMA 各个通道请求

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_ROUTINE(1)	ADC_ROUTINE(2)	●	●	●	●	ADC_INSERTED
SPI / I2S	●	SPI / I2S0_RX	SPI / I2S0_TX	SPI1_RX	SPI1_TX	●	●
USART	●	USART0_TX(1)	USART0_RX(1)	USART0_TX(2) UART1_TX	USART0_RX(2) UART1_RX	●	●
I²C	●	I2C0_TX	I2C0_RX	I2C1_TX	I2C1_RX	I2C2_TX	I2C2_RX
TIMER0	●	TIMER0_CH0	TIMER0_CH1	TIMER0_CH3 TIMER0_TRIG TIMER0_COM	TIMER0_CH2 TIMER0_UP	TIMER0_CH4	TIMER0_CH5
TIMER1	TIMER1_CH2	TIMER1_UP	TIMER1_CH1	TIMER1_CH3	TIMER1_CH0	●	●
TIMER2	●	TIMER2_CH2	TIMER2_CH3 TIMER2_UP	TIMER2_CH0 TIMER2_TRIG	●	●	●
TIMER5	●	●	●	●	●	TIMER5_UP	●
TIMER6	●	●	●	●	●	●	TIMER6_UP
DAC0	●	●	DAC0_CH1	●	●	DAC0_CH0	●
DAC1	●	DAC1_CH1	●	●	●	●	DAC1_CH0
TIMER14	●	●	●	●	TIMER14_CH0 TIMER14_UP TIMER14_TRIG TIMER14_COM 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_CFG0 寄存器的相应重映射位被清零时，这个请求被重映射到此通道。

(2)当 SYSCFG_CFG0 寄存器的相应重映射位被置位时，这个请求被重映射到此通道。

3.5. DMA 寄存器

DMA 基地址：0x4002 0000

3.5.1. 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

3.5.2. 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

3.5.3. 通道 x 控制寄存器(DMA_CHxCTL)

地址偏移：0x08 + 0x14 * x(x = 0...6, x 为通道序号)
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

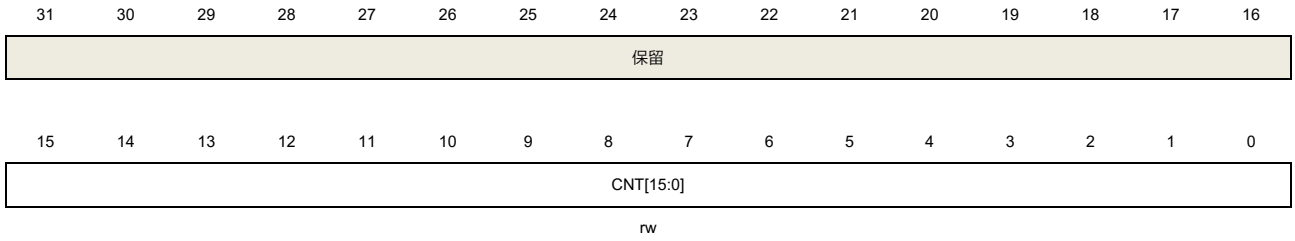
		CHEN位为1时，该位不能被配置
13:12	PRIO[1:0]	软件优先级 软件置位和清零 00：低 01：中 10：高 11：极高 CHEN位为1时，该位域不能被配置
11:10	MWIDTH[1:0]	存储器的传输数据宽度 软件置位和清零 00：8-bit 01：16-bit 10：32-bit 11：保留 CHEN位为1时，该位域不能被配置
9:8	PWIDTH[1:0]	外设的传输数据宽度 软件置位和清零 00：8-bit 01：16-bit 10：32-bit 11：保留 CHEN位为1时，该位域不能被配置
7	MNAGA	存储器的地址生成算法 软件置位和清零 0：固定地址模式 1：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

3.5.4. 通道 x 计数寄存器(DMA_CHxCNT)

地址偏移：0x0C + 0x14 * x(x = 0...6, x 为通道序号)
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 该寄存器表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。

3.5.5. 通道 x 外设基地址寄存器(DMA_CHxPADDR)

地址偏移：0x10 + 0x14 * x(x = 0...6, x 为通道序号)
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。

3.5.6. 通道 x 存储器基地址寄存器(DMA_CHxMADDR)

地址偏移：0x14 + 0x14 * x(x = 0...6, x 为通道序号)
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。

4. 闪存控制器(FMC)

4.1. 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64K字节空间内，CPU执行指令需要一点等待时间。FMC也提供了页擦除、整片擦除以及64位双字编程等操作。

4.2. 主要特性

- 高达64KB的片上闪存可用于存储指令或数据；
- 在闪存的64KB空间内，CPU执行指令需0~2个等待时间；
- 闪存支持64位宽数据读取；
- 预取缓冲区用于提高读操作速度；
- 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应用需求；
- 1K字节OTP块(一次性编程)，用于存储用户数据；
- 每页大小为1K字节；
- 支持单比特可纠错和多比特检测ECC功能；
- 64位双字编程，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访问；
- 有页擦除和页编程保护功能，阻止意外操作。

4.3. 功能描述

4.3.1. 闪存结构

闪存存储器包括一个高达64KB字节的主闪存(按64页每页1KB字节分块)和一个3KB字节的用于引导装载程序的信息块。主闪存存储器的64页每页都可以单独擦除。闪存存储器的基地址和大小见*表 4-1. 闪存的基地址和大小*。

表 4-1. 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

闪存块	名称	地址范围	大小(字节)
一次性编程块	OTP字节	0x1FFF 7000 - 0x1FFF 73FF	1KB

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。

4.3.2. 错误检查与纠正(ECC)

ECC机制支持:

- 单个位错误检测和纠错
- 双位错误检测

当单个位错误被检测与纠正时:

- 当ECC错误发生时，FMC_ECCCS寄存器中的ECCCOR位将置位。如果设置了FMC_ECCCS寄存器中的ECCCORIE位，则会产生对应中断。ECCADDR表示各个存储空间出现ECC错误的地址。向FMC_ECCCS寄存器ECCCOR位写1可以清除ECCCOR位。

当检测到双位错误时:

- 当ECC错误发生时，FMC_ECCCS寄存器中的ECCDET位将置位。如果设置了FMC_ECCCS寄存器中的ECCDETIE位，则会产生NMI中断。ECCADDR位表示各个存储空间出现ECC错误的地址。数据返回全F。向FMC_ECCCS寄存器ECCDET位写1可以清除ECCDET位。

注意:

1. 当启用ECC功能时，闪存中的数据是按72位存储的，每双字(64位)后增加了8位纠错码。但增加的8位纠错码由硬件自动计算，用户不可访问。
2. 对于原始数据0xFF FFFF FFFF FFFF FFFF，不支持ECC。
3. 当出现新的ECC错误时，ECCADDR会更新为最新的错误地址。
4. 当上报ECC错误时，如果数据仍然存在于当前的缓冲区或预取缓冲区中，即使清除ECCCOR和ECCDET，重新读取该错误地址也不会产生ECC错误。
5. 预取buffer读取数据或指令也可能生成ECC错误，即使数据或指令未被CPU使用。为了正确显示ECC错误地址寄存器的值，需禁用预取功能；否则，显示的错误地址可能会因为预取而往前推。
6. 当编程原始数据为全F时，ECC计算会bypass本次操作。不需要擦除，再次编程其他值可正常写入。

4.3.3. 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可以使用CPU的AHB BUS总线。

等待状态添加

根据AHB时钟频率正确配置FMC_WS寄存器中WSCNT位。WSCNT位域与AHB时钟频率之间的关系如*表4-2. WSCNT位域与AHB时钟频率之间的关系*所示。

表 4-2. WSCNT 位域与 AHB 时钟频率之间的关系

AHB时钟频率	WSCNT配置
<= 24MHz	0(添加0个等待状态)
<= 48MHz	1(添加1个等待状态)
<= 72MHz	2(添加2个等待状态)

如果发生系统复位，AHB时钟频率为8MHz，WSCNT的值为0。

注意：

1. 如果需要提高AHB的时钟频率，首先，参考*表4-2. WSCNT位域与AHB时钟频率之间的关系*，根据目标AHB时钟频率配置WSCNT位域。然后将AHB时钟频率提高到目标时钟频率。在配置WSCNT位域之前提高AHB时钟频率是不可行的。

2. 如果需要降低AHB的时钟频率，首先，降低目标AHB时钟频率，然后参考*表4-2. WSCNT位域与AHB时钟频率之间的关系*，根据目标AHB时钟频率配置WSCNT位域。在降低AHB时钟频率之前配置WSCNT位域是不可行的。

由于添加了等待状态，读效率非常低(如72MHz时添加两个等待状态)。为了加速读操作，需要用到以下功能。

当前缓冲区

当前缓冲区总是被使能的。每次对闪存存储器进行读操作，将获得一个64位的数据并存储在当前缓冲区中。而每一个读操作，CPU只读取32位或16位数据。所以在连续代码的情况下，下一个数据能从当前缓冲区中获取到而不需要重复地从存储器获取。

预取缓冲区

当FMC_WS寄存器中PFEN位被置位时预取缓冲区被使能。在连续代码的情况下，当CPU处理当前缓冲区中数据(64位)时，32位操作至少需要两个时钟，16位操作至少需要4个时钟。在这种情况下，在闪存存储器的下一个双字地址预取数据并存到预取缓冲区。当CPU完成当前缓冲区并需要处理下一个数据，预取缓冲区被命中。

4.3.4. FMC_CTL 寄存器解锁

复位后，除了用于重加载选项字节的OBRLD位，FMC_CTL寄存器其他位不可被改写，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。

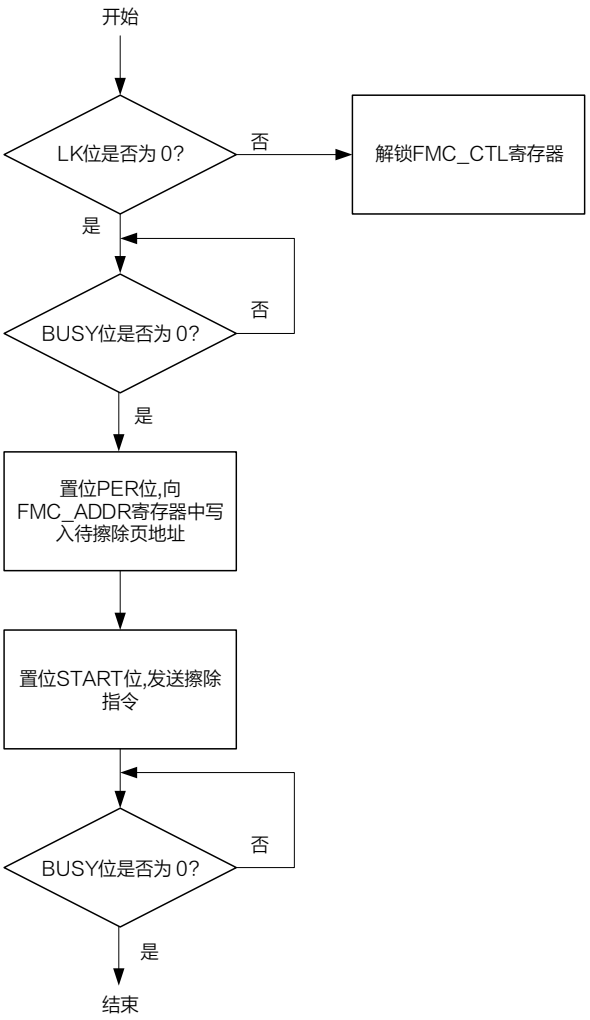
4.3.5. 页擦除

FMC的页擦除功能将一个主闪存存储页的所有字节初始化为0xFF。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，通过将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读来验证该页的内容。

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。如果目标页正在被用来取指令或访问数据时对其进行擦除操作，程序可能"跑飞"。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。*图4-1. 页擦除操作流程*显示了页擦除操作的流程：

图 4-1. 页擦除操作流程



4.3.6. 整片擦除

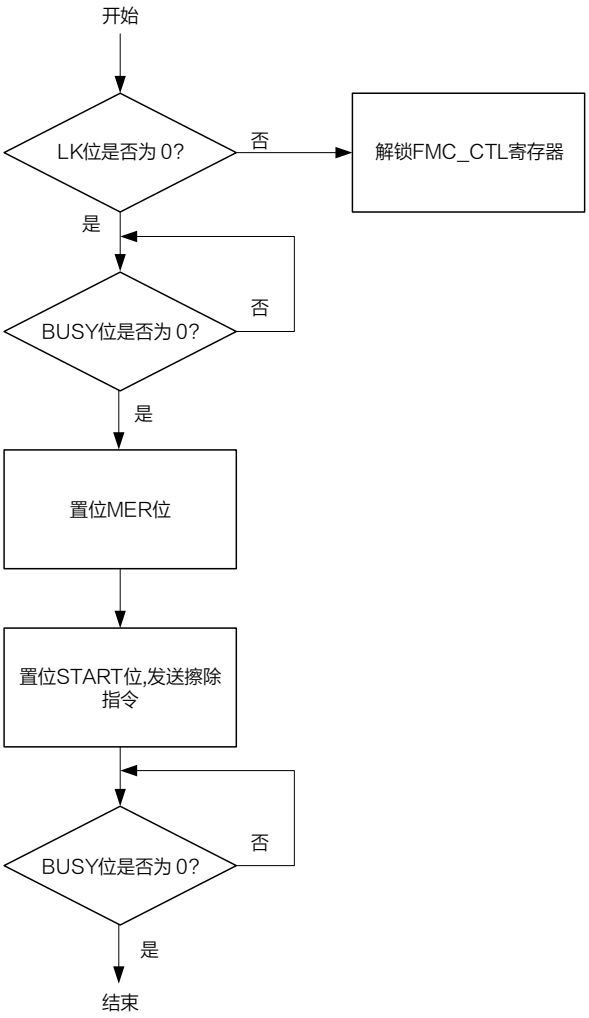
FMC提供了整片擦除功能可以初始化主闪存块的内容。整片擦除操作的寄存器设置具体步骤如下：

- 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，通过将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读来验证该页的内容。

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)。*图4-2. 整片擦除操作流程*显示了整片擦除操作的流程：

图 4-2. 整片擦除操作流程



4.3.7. 主存储闪存块编程

FMC通过DBUS提供了64位双字编程功能，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；
- 写编程命令，通过将FMC_CTL寄存器的PG位置1；
- DBUS写一个64位双字数据到预期绝对地址(0x08XX XXXX)；
- 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双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即使编程内容为 0x0。另一方面，在擦除/编程保护的页进行编程操作将会无效，并且产生 WPERR 错误。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 位和 WPERR 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。

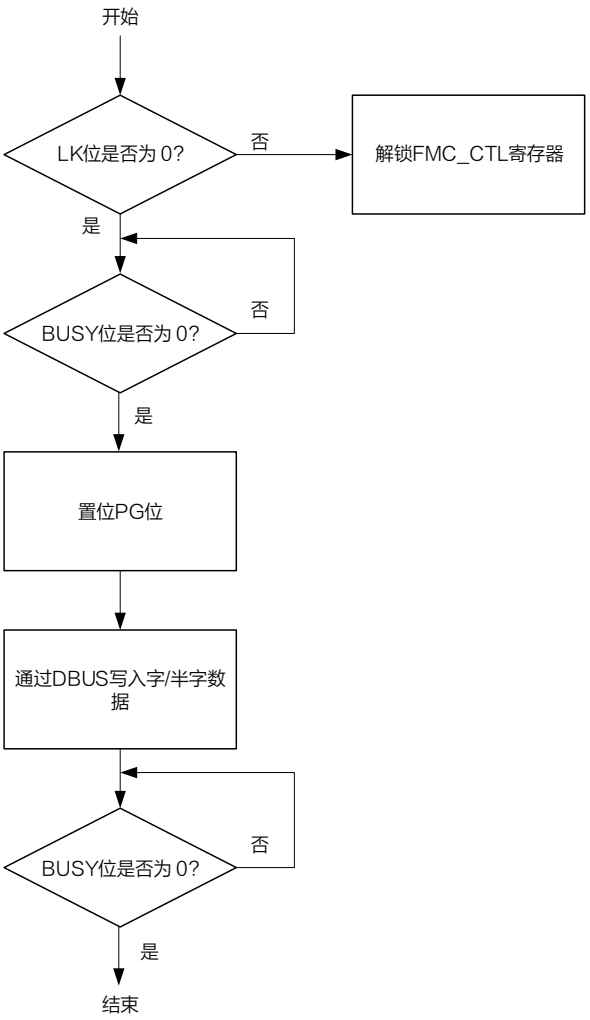
在以下情况中，FMC_STAT寄存器的PGAERR位将被置位。

- DBUS采用8位编程的方式；
- 32位和16位编程混合使用；
- DBUS编程不对齐。如果采用DBUS 32位编程，DBUS第二次编程必须是双字对齐并且属于同一个双字地址范围。如果采用DBUS 16位编程，第二/三/四次编程必须是双字对齐并且属于同一个双字地址范围。

注意：如果没有完整写完64位，这个数据不会被写进闪存，并不会任何错误提示。

在这些情况下，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。软件可以通过检查FMC_STAT寄存器中PGERR，PGAERR或者WPERR位来检测是哪个条件触发了中断。*图4-3. 双字编程操作流程*显示了主存储块字编程操作的流程：

图 4-3. 双字编程操作流程



4.3.8. OTP 编程

OTP编程方法与主储存闪存编程相同。OTP块只能被编程一次并且不能被擦除。

注意：必须确保在OTP编程操作时不会发生任何意外中断，例如系统复位或掉电。如果发生意外中断，会有极低的概率使存储在闪存中的数据发生破坏。

4.3.9. 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，将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。

4.3.10. 选项字节编程

FMC提供了64位双字编程功能，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，将FMC_CTL寄存器的OBPG位置1；
- 写一个64位双字到预期地址；
- 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双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

上的编程操作无效即使编程内容为0x0。FMC_STAT寄存器的ENDF位指示操作的结束。

4.3.11. 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。选项字节的反码和选项字节相反。当选项字节被重加载，如果选项字节的反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*表 4-3. 选项字节*。

表 4-3. 选项字节

地址	名称	出厂值	说明
0x1fff f800	OB_SPC	0xA5	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	0x5A	OB_SPC反码值

地址	名称	出厂值	说明
0x1fff f802	OB_USER	GD32E252xx系列： 0xFF GD32E251xx系列： 0xF7	用户定义的选项字节 [7]：NRST_MDSEL GD32E252xx系列： 0：PB11 pin作为通用输入/输出引脚(GPIO) 1：PB11 pin作为复位引脚 GD32E251xx系列： 0：PB11 pin作为复位引脚 1： PB11 pin作为通用输入/输出引脚(GPIO) [6]：SRAM_PARITY_ECC_CHECK 0：使能SRAM奇偶校验，失能SRAM ECC 1：失能SRAM奇偶校验，使能SRAM ECC **注意：**为了防止SRAM ECC功能从禁用状态切换到启用状态时发生数据读取错误，SRAM ECC功能仅在上电复位后生效。如果启用了SRAM ECC功能，SRAM奇偶校验功能将被禁用。 [5]：VDDA_VISOR 0：去使能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]：BOOT0_n 0：BOOT0位是1 1：BOOT0位是0 **注意：**该Bit仅适用于GD32E251xx系列 [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

地址	名称	出厂值	说明
0x1fff f803	OB_USER_N	GD32E252xx系列： 0x00 GD32E251xx系列： 0x08	OB_USER反码值
0x1fff f804	OB_DATA[7:0]	0xFF	用户定义数据位7到0位
0x1fff f805	OB_DATA_N[7:0]	0x00	OB_DATA反码值的7到0位
0x1fff f806	OB_DATA[15:8]	0xFF	用户定义数据位15到8位
0x1fff f807	OB_DATA_N[15:8]	0x00	OB_DATA反码值的15到8位
0x1fff f808	OB_WP[7:0]	0xFF	页擦除/编程保护位的7到0位
0x1fff f809	OB_WP_N[7:0]	0x00	OB_WP反码值的7到0位
0x1fff f80a	OB_WP[15:8]	0xFF	页擦除/编程保护位的15到8位
0x1fff f80b	OB_WP_N[15:8]	0x00	OB_WP反码值的15到8位

4.3.12. 页擦除/编程保护

FMC的页擦除/编程保护功能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，FMC将触发闪存操作错误中断。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。在选项字节的OB_WP位置1或清0之后，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。*显示了通过设置OB_WP[15:0]寄存器可以保护哪些页。*

表4-4. OB_WP位对应页保护显示了通过设置OB_WP[15:0]寄存器可以保护哪些页。

表 4-4. 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[14]	页56 ~ 页59
OB_WP[15]	页60 ~ 页63

4.3.13. 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字节和它的反码值设置为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字节和它的反码值为除了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字节和它的反码值设置为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字节和它的反码值为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字节和它的反码值不能被重编程。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。

注意：在读保护情况下，flash中前4KB不能被擦写。

4.4. FMC 寄存器

FMC基地址：0x4002 2000

4.4.1. 等待状态寄存器 (FMC_WS)

地址偏移： 0x00
复位值： 0x0000 0030

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											PFEN	保留	WSCNT[2:0]		
											rw		rw		

位/位域	名称	描述
31:5	保留	必须保持复位值
4	PFEN	预取使能 0：预取失能 1：预取使能
3	保留	必须保持复位值
2:0	WSCNT[2:0]	等待状态计数寄存器 软件置 1 和清 0。 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

4.4.2. 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。

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

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

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。

4.4.3. 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

4.4.4. 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								SRAM_ECC_EN	SRAM_PARITY_EN	ENDF	WPERR	PGAERR	PGERR	保留	BUSY
								r	r	rc_w1	rc_w1	rc_w1	rc_w1		r

位/位域	名称	描述
31:8	保留	必须保持复位值
7	SRAM_ECC_EN	SRAM ECC 使能状态位 当前 SRAM ECC 使能状态:

		当 SRAM ECC 被启用时，该位为 1，否则为 0。
6	SRAM_PARITY_EN	SRAM 奇偶校验使能状态位 当前 SRAM 奇偶校验使能状态。 当 SRAM 奇偶校验被启用时，该位为 1，否则为 0。
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	PGAERR	编程对齐错误标志位 在 DBUS 写未对齐时由硬件置 1。软件可以通过写 1 来清 0 该位。
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。

4.4.5. 控制寄存器 (FMC_CTL)

地址偏移：0x10

复位值：0x0000 0080，NRST_MDSEL_REQ 和 NRST_MDSEL_VALUE 仅由电源复位重置，其他位由系统复位重置

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														NRST_MDSEL_REQ	NRST_MDSEL_VALUE
														rw	rw

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

位/位域	名称	描述
31:18	保留	必须保持复位值
17	NRST_MDSEL_REQ	该位表示 NRST_MDSEL 是否由寄存器控制。 0：不受寄存器控制 1：受寄存器控制，NRST_MDSEL_VALUE 生效
16	NRST_MDSEL_VAL	当 NRST_MDSEL_REQ 设置为 1 时，PB11 的值。

	UE	0：PB11 引脚作为通用输入/输出(GPIO) 1：PB11 引脚作为 NRST 复位引脚
15	RPCD	RAM 奇偶校验禁用 0：当 RPCS 为 1 时，RAM 奇偶校验被使能 1：当 RPCS 为 1 时，RAM 奇偶校验被禁能 **注意：**仅当 SRAM_PARITY_ECC_CHECK 为 0 时，该位才有效
14	RPCS	RAM 奇偶校验控制选择 0：RAM 奇偶校验由选项字节控制 1：RAM 奇偶校验由 RPCD 控制 **注意：**仅当 SRAM_PARITY_ECC_CHECK 为 0 时，该位才有效
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

4.4.6. 地址寄存器(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。

4.4.7. 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。

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

OB_USER[7:0]	保留	PLEVEL[1:0]	OBERR
r		r	r

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 当选项字节和它的反码不匹配时此位由硬件置 1，并且选项字节被设置为 0xFF。

4.4.8. 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。

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
OB_WP[15:0]															
r															

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

4.4.9. ECC 控制及状态寄存器(FMC_ECCCS)

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
ECCINJEN	保留					ECCINJDATA[1:0]		保留							

rw							rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCCORIE	ECCDETIE	保留												ECCDET	ECCCOR
rw	rw													rc_w1	rc_w1

位/位域	名称	描述
31	ECCINJEN	ECC 故障注入使能 0：ECC 故障注入失能。 1：ECC 故障注入使能。
30:26	保留	必须保持复位值。
25:24	ECCINJDATA[1:0]	当 ECCINJEN 为 1 时，通过设置该位域可以控制对闪存进行单个位或者双位错误故障注入。 00：无错误故障注入 01：单个位错误故障注入 10：单个位错误故障注入 11：双位错误故障注入 **注：**该位域只能用于向闪存数据的 bit0 和 bit1 注入错误。
23:16	保留	必须保持复位值。
15	ECCCORIE	纠正单个位中断使能 0：纠正单个位中断失能。 1：纠正单个位中断使能。
14	ECCDETIE	检测到双位错误中断使能。 0：检测到双位错误中断失能 1：检测到双位错误中断使能
13:2	保留	必须保持复位值。
1	ECCDET	检测到双位错误标志。当检测到双位错误时，该位置 1。 该位写 1 清零。 0：未检测到 ECC 双位错误。 1：检测到 ECC 双位错误。
0	ECCCOR	检测并纠正单个位错误标志 该位写 1 清零。 0：未检测并纠正 ECC 单个位错误。 1：检测并纠正 ECC 单个位错误。

4.4.10. ECC 地址寄存器(FMC_ECCADDR)

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
ECCADDR[31:16]															
r															

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

位/位域	名称	描述
31:0	ECCADDR[31:0]	ECC 错误地址(64 位对齐)。

4.4.11. 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。

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

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

位/位域	名称	描述
31:0	PID[31:0]	产品 ID 寄存器 该寄存器为只读 该寄存器在生产过程中被一次性编程。

5. 电源管理单元(PMU)

5.1. 简介

功耗设计是GD32E25x系列产品最注重的问题之一。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5-1. 电源域概览*所示，GD32E25x系列设备有三个电源域，包括V_{DD} / V_{DDA}域、V_{CORE}域和备份域。V_{DD}域及备份域由电源直接供电。在V_{DD} / V_{DDA}域中嵌入了一个LDO，用来为V_{CORE}域供电。

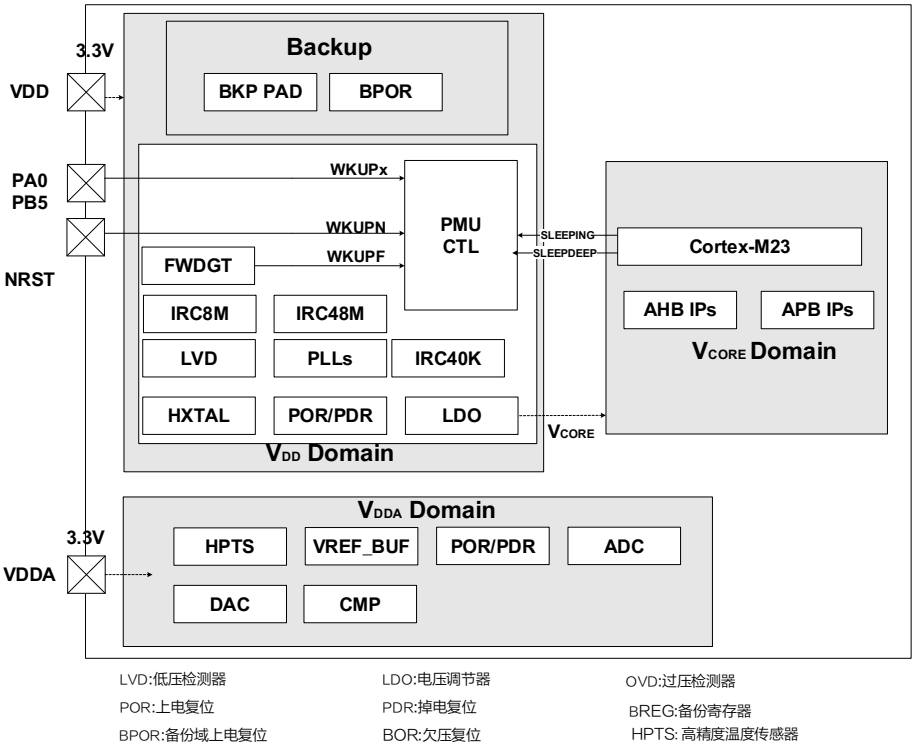
5.2. 主要特征

- 三个电源域：备份域、V_{DD} / V_{DDA}域和V_{CORE}域；
- 三种省电模式：睡眠模式、深度睡眠模式和待机模式；
- 内部电压调节器(LDO)提供V_{CORE}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；
- LDO 输出电压用于节约能耗。

5.3. 功能说明

*图5-1. 电源域概览*提供了PMU及相关电源域的内部结构框图。

图5-1. 电源域概览



5.3.1. 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)、IRC8M(内部 8M RC 振荡器)、IRC28M(内部 28M RC 振荡器)、IRC40K(内部 40KHz RC 振荡器)、PLLs(锁相环)、LVD(低电压检测器)等。V_{DDA} 域包括 ADC / DAC(AD / DA 转换器)等等。

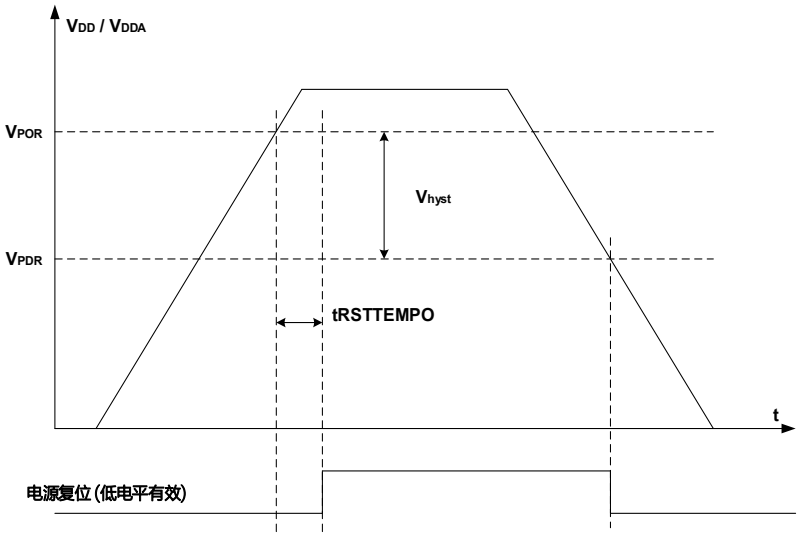
V_{DD} 域

为 V_{CORE} 域供电的 LDO(电压调节器),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 VDDA / VDD 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。*图 5-2. 上电/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。迟滞电压 V_{hyst} 值请参考 datasheet。

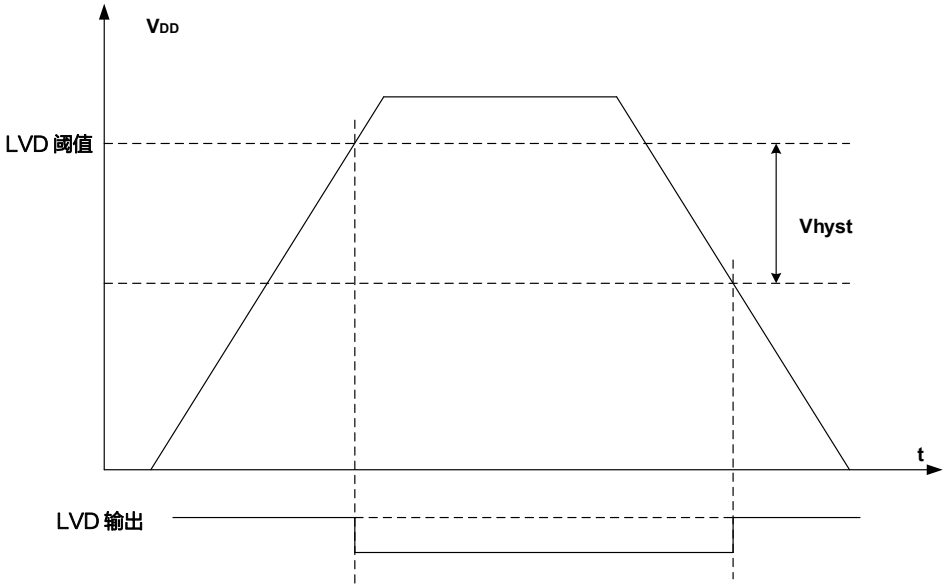
注意：如果用户能确保 V_{DDA} 电压大于等于 VDD，可以通过清零选项字节寄存器 OB_USER 的位 VDDA_VISOR，关闭由 PDR 电路实现的 V_{DDA} 监视器以减小功耗。

图5-2. 上电/掉电复位波形图



LVD 的功能是检测 V_{DD} 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，位于电源控制和状态寄存器(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 5-3. LVD 阈值波形图*显示了 V_{DD} 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)。该图也显示了供电电压与 LVD 信号的关系。迟滞电压 V_{hyst} 值请参考 datasheet。

图5-3. LVD阈值波形图



备份域

电池备份域由VDD供电。该备份域包含BPOR(备份域上电复位)、BREG(备份寄存器)。

备份域的复位源包括备份域上电复位和备份域软件复位。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。

V_{DDA}域

一般来说，数字电路由 VDDA 供电，而大多数的模拟电路由 VDDA 供电。为了提高 ADC 和 DAC 的转换精度，为 VDDA 独立供电可使模拟电路达到更好的特性。为避免噪声，VDDA 通过外部滤波电路连接至 VDD，相应的 VSSA 通过特定电路连接至 VSS。否则，如果 V_{DDA} 和 V_{DD} 不同时，V_{DDA} 须高于 V_{DD}，但压差不超过 0.3V。

为提高 ADC 和 DAC 的精度，可将独立的外部参考电压连接至 ADC / DAC 引脚 VREFP。根据不同的封装，VREFP 可被连接至 VDDA 引脚，或者外部参考电压，外部参考电压的范围请参考*表 13-2. ADC 輸入引脚定义*和*表 14-1.DAC 引脚*。

5.3.2. V_{CORE}电源域

主要功能包括Cortex®-M23内核逻辑、AHB / APB外设、备份域和V_{DD} / V_{DDA}域的APB接口等。当V_{CORE}电压上电后，POR将在V_{CORE}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。

5.3.3. 省电模式

系统复位或电源复位后，GD32E25x MCU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：减慢系统时钟(HCLK，PCLK1，PCLK2)，或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®-M23 的 SLEEPING 模式相对应。在睡眠模式下，仅关闭 Cortex®-M23 的时钟。如需进入睡眠模式，只要清除 Cortex®-M23 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®-M23 技术手册)。由于无需在进入或退出中断上消耗时间，该模式所需的唤醒时间最短。

根据 Cortex®-M23 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®-M23 的 SLEEPDEEP 模式相对应。在深度睡眠模式下，V_{CORE} 域中的

所有时钟全部关闭，IRC8M、IRC28M、HXTAL 及 PLLs 也全部被禁用。SRAM 和寄存器中的内容被保留。根据 PMU_CTL 寄存器的 LDOLP 位的配置，可控制 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®-M23 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 的事件可以将系统从深度睡眠模式中唤醒(如果 SEVONPEND 为 1，任何来自 EXTI 的中断都可以唤醒系统，请参考 Cortex®-M23 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 2-3. 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®-M23 的 SLEEPDEEP 模式实现的。在待机模式下，整个 V_{CORE} 域全部停止供电，同时 LDO 和包括 IRC8M、IRC28M、HXTAL 和 PLL 也会被关闭。进入待机模式前，先将 Cortex®-M23 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。待机模式有四个唤醒源，包括来自 NRST 引脚的外部复位，BKP 侵入事件，FWDGT 复位，WKUP 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 V_{CORE} 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®-M23 将从 0x00000000 地址开始执行指令代码。

表 5-1. 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 V_{CORE} 电源域的所有时钟 2、关闭 IRC8M、HXTAL 和 PLL	1、关闭 V_{CORE} 电源域的供电 2、关闭 IRC8M、HXTAL 和 PLL
LDO 状态	开启(正常功耗模式、正常驱动模式)	开启(正常功耗模式或低功耗模式、正常驱动模式或低驱动模式)	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 WFE 进入，则任何事件(或 SEVONPEND=1 时的中断)均可唤醒	若通过 WFI 进入，来自 EXTI 的任何中断可唤醒； 若通过 WFE 进入，来自 EXTI 的任何事件(或 SEVONPEND=1 时的中断)可唤醒	1、NRST 引脚 2、WKUP 引脚 3、FWDGT 复位 4、BKP
唤醒延迟	无	IRC8M 唤醒时间	上电序列

模式	睡眠	深度睡眠	待机
		如果 LDO 处于低功耗模式，需增加 LDO 唤醒时间	

注意：在待机模式下，除了 NRST 引脚，使能的 WKUP 引脚，其他所有 I / O 都处于高阻态。

5.4. PMU 寄存器

PMU基地址：0x4000 7000

5.4.1. 控制寄存器(PMU_CTL)

地址偏移：0x00
复位值：0x0000 4000(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。

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							BKPWEN	LVDT[2:0]			LVDEN	STBRST	WURST	STBMOD	LDOLP
							rw	rw			rw	rc_w1	rc_w1	rw	rw

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访问 1：允许对备份域寄存器的写访问 复位之后，任何对备份域寄存器的写访问都将被禁止。如需对备份域寄存器做写访问，需先将该位置1。
7:5	LVDT[2:0]	低电压检测器阈值 000：2.04V 001：2.18V 010：2.32V 011：2.46V 100：2.60V 101：2.74V 110：2.88V 111：3.02V
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[2:0]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®-M23进入SLEEPDEEP模式时，系统进入深度睡眠模式 1：当Cortex®-M23进入SLEEPDEEP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。

5.4.2. 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		WUPEN5	保留				WUPEN0	保留					LVDF	STBF	WUF
		rw					rw						r	r	r

位/位域	名称	描述
31:14	保留	必须保持复位值。
13	WUPEN5	WKUP引脚5唤醒使能(PB5) 0：关闭WKUP引脚5唤醒功能 1：开启WKUP引脚5唤醒功能 如果WUPEN5在进入待机模式之前置1，WKUP引脚5的上升沿会将系统从待机模式唤醒。由于WKUP引脚5为高电平有效，WKUP引脚5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。
12:9	保留	必须保持复位值。
8	WUPEN0	WKUP引脚0唤醒使能(PA0) 0：关闭WKUP引脚0唤醒功能 1：开启WKUP引脚0唤醒功能 如果WUPEN0在进入待机模式之前置1，WKUP引脚0的上升沿会将系统从待机模式唤醒。由于WKUP引脚0为高电平有效，WKUP引脚0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	低电压检测器状态标志 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BKP侵入事件 该位由系统复位或通过设置PMU_CTL寄存器的WURST位来清零。

6. 备份域(BKP)

6.1. 简介

位于备份域中的备份寄存器可在 V_{DD} 电源关闭时由 V_{BAT} 供电，备份寄存器可用来存储并保护用户应用数据，从待机模式唤醒或系统复位也不会对这些寄存器造成影响。

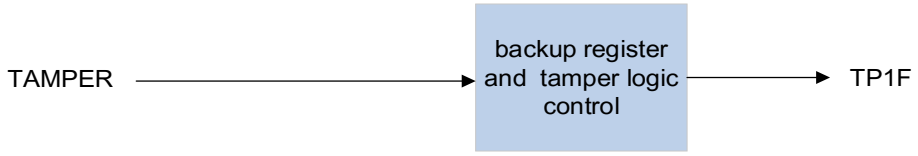
6.2. 主要特征

- 侵入源可以选择，侵入类型可以配置；
- 五个32位(总共20字节)通用备份寄存器，可以在节能模式下保存数据。如果检测到侵入事件，备份寄存器将被重置。

6.3. 功能说明

6.3.1. 结构框图

图 6-1. BKP 结构框图



BKP 单元包括:

- 侵入事件 / 中断
- 32位备份寄存器

6.3.2. 侵入检测

BKP_TAMPx 引脚输入可以在边沿检测模式或具有可配置滤波设置的电平检测模式下用于侵入事件检测。

备份寄存器(BKP_DATAx)

BKP 备份寄存器位于 V_{DD} 备份域。从待机模式或系统复位唤醒不会影响这些寄存器。

这些寄存器仅在检测到侵入事件和备份域复位时被重置。

侵入功能的初始化

BKP 侵入检测功能可以通过设置相应的 TP1EN 位来独立启用侵入输入引脚。在启用 TP1EN 位之前，应设置侵入检测配置。当检测到侵入事件时，相应的标志(TP1F)将被置位。如果设置了侵入中断使能(TPIE)，侵入事件可以触发中断。任何侵入事件都将重置所有备份寄存器(BKP_DATAx)。

侵入事件检测为边沿检测模式

当 FLT 位设置为 0x0 时，侵入检测设置为边沿检测模式，TP1EG 位决定是上升沿还是下降沿作为检测边沿。在边沿检测模式下，侵入检测输入引脚上的内部上拉电阻被停用。

由于检测到侵入事件会重置备份寄存器(BKP_DATAx)，写入备份寄存器时应确保侵入事件重置和写入操作不会同时发生。避免这种情况的推荐方法是在写入备份寄存器之前禁用侵入检测，并在完成写入后重新启用侵入检测。
注意：如果启用了侵入检测，即使 V_{DD} 电源关闭，侵入检测仍然在运行。

侵入事件检测为带可配置滤波功能的电平检测模式

当 FLT 位没有被重置为 0x0 时，侵入检测设置为电平检测模式，FLT 位决定了有效电平所需的连续采样数量(2、4 或 8)。当 DISPU 设置为 0x0(这是默认设置)，内部上拉电阻将在每次采样前为侵入输入引脚预充电，从而允许更大的电容连接到侵入输入引脚。预充电持续时间通过 PRCH 位配置。较大的电容需要较长的预充电时间。

每次采样之间的时间间隔也是可配置的。通过调整采样频率(FREQ)，软件可以在功耗和侵入检测延迟之间进行平衡。

6.4. BKP 中断

所有 BKP 中断都连接到 EXTI 控制器。

如果您想使用 BKP 侵入功能，应遵循以下步骤：

1. 配置并启用 EXTI 的相应中断线以连接到 BKP 侵入事件，并设置触发的上升沿；
2. 配置并启用 BKP 侵入的全局中断；
3. 配置并启用 BKP 侵入功能。

表 6-1. BKP 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
侵入1	TP1F	TPIE	Y	Y(*)	Y(*)

*只有在 BKP 时钟源为 IRC40K 时才有效。

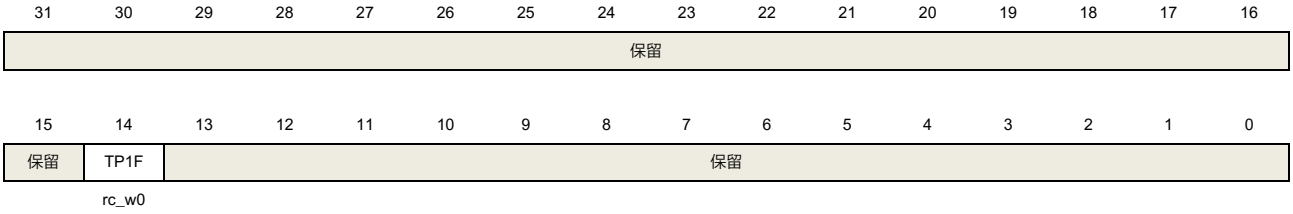
6.5. BKP 寄存器

BKP基地址：0x4000 2800

6.5.1. 状态寄存器(BKP_STAT)

偏移地址：0x0C
备份域复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

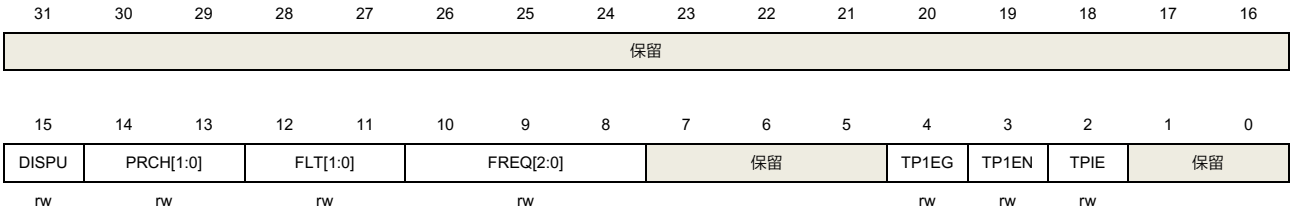


位/位域	名称	描述
31:15	保留	必须保持复位值。
14	TP1F	TAMPER 事件标志 当在 tamper1 输入管脚检测到侵入事件时，该位硬件置 1。可以通过向该位软件写 0 来清除。
13:0	保留	必须保持复位值。

6.5.2. 侵入寄存器(BKP_TAMP)

偏移地址：0x40
备份域复位值：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	DISPU	TAMPER 上拉禁用位 0：使能内部 TAMPER 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]	TAMPER 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 BKP 时钟 0x1：2 个 BKP 时钟 0x2：4 个 BKP 时钟 0x3：8 个 BKP 时钟
12:11	FLT[1:0]	TAMPER 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 BKPCLK(若 BKPCLK=32.768KHz，频率为 1Hz) 0x1：每次采样间隔 16384 个 BKPCLK(若 BKPCLK=32.768KHz，频率为 2Hz) 0x2：每次采样间隔 8192 个 BKPCLK(若 BKPCLK=32.768KHz，频率为 4Hz) 0x3：每次采样间隔 4096 个 BKPCLK(若 BKPCLK=32.768KHz，频率为 8Hz) 0x4：每次采样间隔 2048 个 BKPCLK(若 BKPCLK=32.768KHz，频率为 16Hz) 0x5：每次采样间隔 1024 个 BKPCLK(若 BKPCLK=32.768KHz，频率为 32Hz) 0x6：每次采样间隔 512 个 BKPCLK(若 BKPCLK=32.768KHz，频率为 64Hz) 0x7：每次采样间隔 256 个 BKPCLK(若 BKPCLK=32.768KHz，频率为 128Hz)
7:5	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
1:0	保留	必须保持复位值

注意：强烈建议在改变侵入检测配置之前，应该复位 TP1EN 位。

6.5.3. 备份域寄存器(BKP_DATAx)(x=0…4)

偏移地址：0x50~0x60

备份域复位值：0x0000 0000
系统复位：无影响
该寄存器是写入和读取保护。

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。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。

7. 复位和时钟单元(RCU)

7.1. 复位控制单元(RCTL)

7.1.1. 简介

GD32E25x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。接下来的章节将详细介绍这些复位。

7.1.2. 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、从待机模式中退出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。退出待机模式时，1.2V电源域内的所有寄存器都会被重置为默认值，而1.2V电源域外的寄存器(如WKUP、FWDGT和待机模式控制寄存器)则不受影响。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电压给 GD32E25x产品时，电源复位电平将变为无效。复位入口向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®-M23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)；
- 用户选项字节寄存器nRST_STDBY设置为0，并且进入待机模式时(OB_STDBY_RSTn)；
- 用户选项字节寄存器nRST_DPSLP设置为0，并且进入深度睡眠模式时(OB_DPSLP_RSTn)。

注意：NRST引脚可以在以下两种模式下配置选项字节NRST_MDSEL[1:0]：

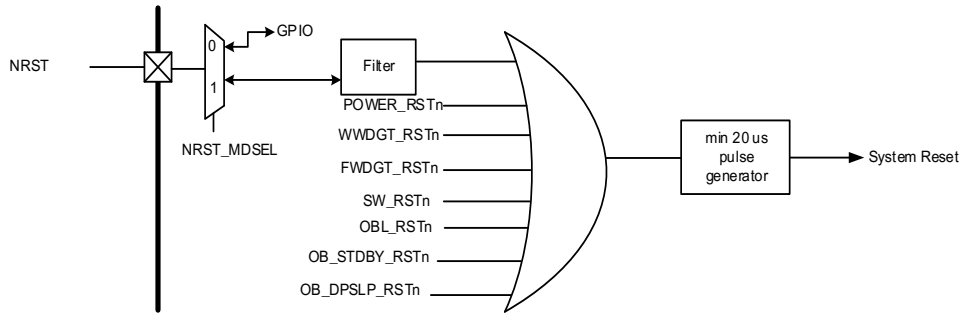
1. 输入/输出模式：在这种模式下，NRST pin的GPIO功能不可用。复位信号可以从NRST引脚传输到MCU，导致MCU复位，复位脉冲信号可以通过NRST引脚反应出来，最小复位脉冲持续时间为20us。
2. GPIO模式：NRST引脚只能作为标准GPIO使用，复位功能不可用，复位信号仅在MCU内部，不反映在NRST引脚上。

注意：对于GD32E251xx，NRST引脚的默认模式是GPIO模式；而对于GD32E252xx，NRST引脚的默认模式是输入/输出模式。

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 7-1. 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(V_{DD}重新上电)。

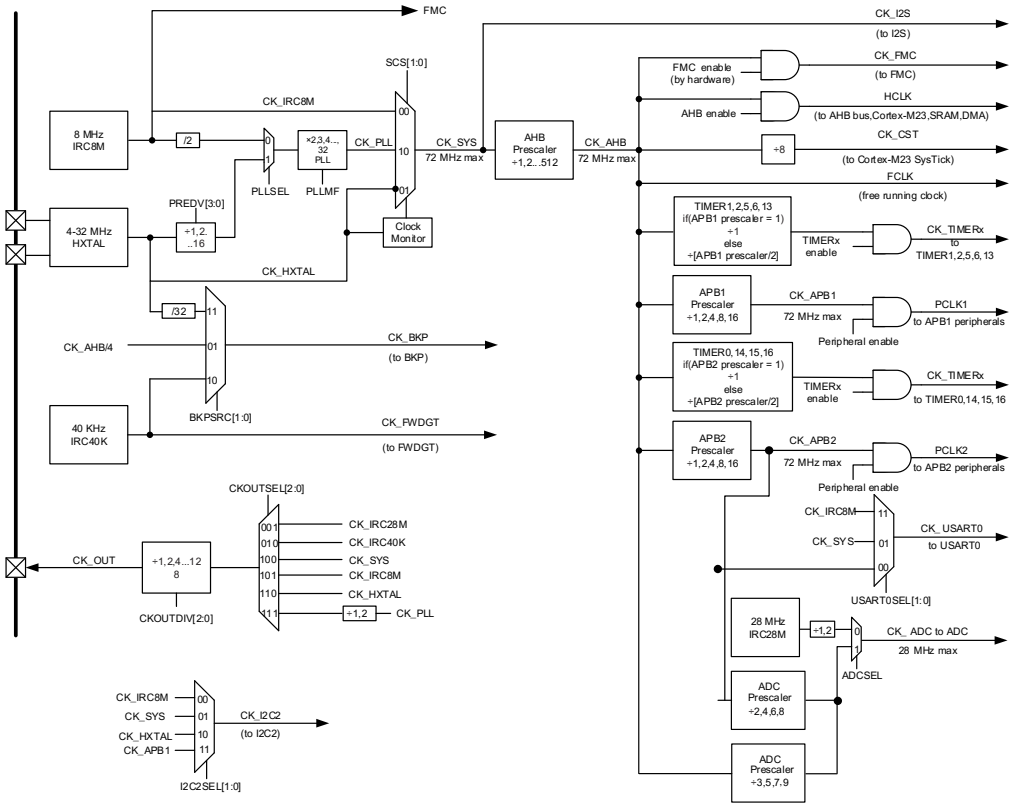
7.2. 时钟控制单元(CCTL)

7.2.1. 简介

时钟控制单元提供了一系列频率和时钟功能，包括一个内部8M RC振荡器时钟(IRC8M)、一个内部28M RC振荡器时钟(IRC28M)、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®-M23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。

图7-2. 时钟树



预分频器可以配置AHB、APB2和APB1域的时钟频率。AHB、APB2和APB1域的最高时钟频率分别为72MHz、72MHz和72MHz。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AHB时钟(HCLK)作为SysTick时钟。

ADC时钟由APB2时钟经2、4、6、8分频或由AHB时钟经3、5、7、9分频或IRC28M或IRC28M / 2时钟获得，它们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、系统时钟或APB2时钟，通过设置配置寄存器2(RCU_CFG2)的USART0SEL位来选择。

I2C2的时钟可以选择IRC8M时钟、HXTAL时钟、系统时钟或APB1时钟，通过设置配置寄存器2(RCU_CFG2)的I2C2SEL位来选择。

FWDGT时钟可以选择IRC40K时钟，当FWDGT启动时强制选择。

BKP时钟可以选择IRC40K时钟、CK_AHB / 4或HXTAL时钟32分频，通过设置备用域控制寄存器(RCU_BDCTL)的BKPSRC位域来选择。

TIMER由来自CK_APB的分频时钟驱动。定时器时钟的频率等于CK_APBx或CK_APBx的两倍。如果APB时钟分频系数为1，定时器的时钟频率与所在AHB总线频率一致。否则，定时器时钟频率设置为AHB频率除以APB预分频器的一半。

7.2.2. 主要特性

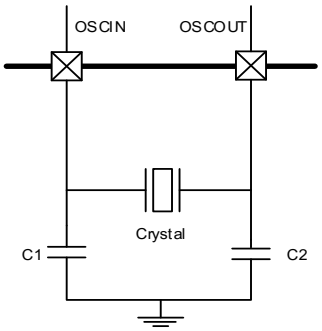
- 4到32 MHz外部高速晶体振荡器(HXTAL)；
- 8 MHz内部高速RC振荡器(IRC8M)；
- 28 MHz内部高速RC振荡器(IRC28M)；
- 40 kHz内部低速RC振荡器(IRC40K)；
- PLL时钟源可以是HXTAL或IRC8M；
- HXTAL时钟监视。

7.2.3. 功能描述

高速外部晶体振荡器时钟(HXTAL)

4到32MHz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。

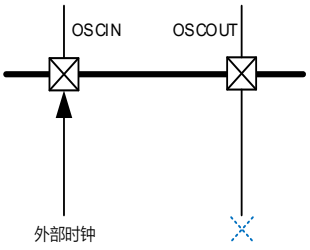
图7-3. HXTAL时钟源



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。旁路输入时，信号接至OSCIN，OSCOUT保持悬空状态，如*图7-4. 旁路模式下HXTAL时钟源*所示。此时，CK_HXTAL等于驱动OSCIN管脚的外部时钟。

图7-4. 旁路模式下HXTAL时钟源



注意：GD32E251xx HXTAL 时钟源仅支持有源晶振输入，不支持无源晶振，使用 HXTAL 时，务必使能旁路模式。

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 28MHz RC 振荡器时钟(IRC28M)

高速内部28MHz RC振荡器时钟 (IRC28M)有一个固定的频率28MHz，专门用作ADC时钟。IRC28M可以通过设置时钟控制寄存器1(RCU_CTL1)中的IRC28MEN位被启动和关闭。时钟控制寄存器1(RCU_CTL1)中的IRC28MSTB位用来指示IRC28M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IRC28MSTBIE被置'1'，在IRC28M稳定以后，将产生一个中断。

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。如果时钟中断寄存器RCU_INT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-M23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

时钟输出功能输出从40kHz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 7-1. 时钟源的选择

时钟源选择位	时钟源
000	无时钟
001	CK_IRC28M
010	CK_IRC40K
011	无时钟
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，USART0能唤醒MCU，前提是它们的时钟是由IRC8M时钟提供且IRC8M时钟被使能。

如果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，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

深度睡眠模式电压寄存器(RCU_DSV)中的DSLPVS[1:0]位可以控制内核在深度睡眠模式下的电压。

表 7-2. 深度睡眠模式下内核电压选择

DSLPVS[1:0]	深度睡眠模式电压(V)
00	1.2
01	1.1
10	1.0
11	0.9

RCU_DSV寄存器被电压秘钥寄存器(RCU_VKEY)保护。只有在写0x1A2B3C4D到RCU_VKEY后，RCU_DSV寄存器才能被写入。

7.3. RCU 寄存器

RCU基地址：0x4002 1000

7.3.1. 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。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留						PLLSTB	PLLEN	保留			HXTALSTBRST	CKMEN	HXTALBPS	HXTALSTB	HXTALEN
						rw	rw				rw	rw	rw	r	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRC8MCALIB[7:0]								IRC8MADJ[4:0]					保留.	IRC8MSTB	IRC8MEN
r								rw						r	rw

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
23:21	保留	必须保持复位值。
20	HXTALSTBRST	HXTAL时钟稳定复位 软件置位或复位。 0：HXTAL时钟稳定未复位 1：HXTAL时钟稳定复位
19	CKMEN	HXTAL时钟监视使能 0：禁止外部4 ~ 32 MHz 晶体振荡器(HXTAL)时钟监视器 1：使能外部4 ~ 32 MHz 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4 ~ 32 MHz晶体振荡器 1：使能外部4 ~ 32 MHz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。当从待机或深度睡眠模式返回或在CKMEN置位的情况下用作系统时钟的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

7.3.2. 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]			保留	PLLMF[3:0]				PLLPRE DV	PLLSEL
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位域的位4 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28M RC振荡器时钟 010：选择内部40K RC振荡器时钟 011：没有时钟被选择 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	保留	必须保持复位值。
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： PLL输出频率不能超过72MHz。

17	PLLPREDV	HXTAL作为PLL输入。该位与时钟配置寄存器1(RCU_CFG1)中的PREDV[0]位是一样的。参考RCU_CFG1的PREDV位说明。 由软件置1或清0来分频或不分频后，当作PLL输入时钟源。 0：选择HXTAL时钟 1：选择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 软件写三位包括RCU_CFG2的31位来确定ADC时钟分频。 软件清0和置1。 000：选择APB2时钟2分频 001：选择APB2时钟4分频 010：选择APB2时钟6分频

011：选择APB2时钟8分频
100：选择AHB时钟3分频
101：选择AHB时钟5分频
110：选择AHB时钟7分频
111：选择AHB时钟9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作为系统时钟或PLL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

7.3.3. 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	IRC28M STBIC	PLL STBIC	HXTAL STBIC	IRC8M STBIC	保留	IRC40K STBIC
								w		w	w	w	w		w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		IRC28M STBIE	PLL STBIE	HXTAL STBIE	IRC8M STBIE	保留	IRC40K STBIE	CKMIF	保留	IRC28M STBIF	PLL STBIF	HXTAL STBIF	IRC8M STBIF	保留	IRC40K STBIF
		rw	rw	rw	rw		rw	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	IRC28MSTBIC	IRC28M时钟稳定中断清除 软件写1复位IRC28MSTBIF标志位。 0：不复位IRC28MSTBIF标志位 1：复位IRC28MSTBIF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	保留	必须保持复位值。
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	IRC28MSTBIE	IRC28M时钟稳定中断使能 软件置1和清0来使能/禁止IRC28M时钟稳定中断。 0：禁止IRC28M时钟稳定中断 1：使能IRC28M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	保留	必须保持复位值。
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	IRC28MSTBIF	IRC28M时钟稳定中断标志位 当IRC28M时钟稳定且IRC28MSTBIE位被置1时由硬件置1。 软件置IRC28MSTBIC=1时清除该位。 0：无IRC28M时钟稳定中断产生 1：IRC28M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	保留	必须保持复位值。
0	IRC40KSTBIF	IRC40K时钟稳定中断标志位 当内部40KHz 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

7.3.4. 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													TIMER16RST	TIMER15RST	TIMER14RST
													rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	USART0RST	保留.	SPI0RST	TIMER0RST	保留	ADCRST	保留								CFGCMPRST
	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
15	保留	必须保持复位值。
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：复位系统配置和比较器模块

7.3.5. 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
DAC1RST	保留	DAC0RST	PMURST	保留			I2C2RST	保留	I2C1RST	I2C0RST	保留			UART1RST	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
保留	SPI1RST	保留		WWDGTRST	保留		TIMER13RST	保留		TIMER6RST	TIMER5RST	保留		TIMER2RST	TIMER1RST
	rw			rw			rw			rw	rw			rw	rw

位/位域	名称	描述
31	DAC1RST	DAC1复位 由软件置1或清0。 0：无复位 1：复位DAC1
30	保留	必须保持复位值。
29	DAC0RST	DAC0复位 由软件置1或清0。 0：无复位 1：复位DAC0
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
27:25	保留	必须保持复位值。
24	I2C2RST	I2C2复位 由软件置1或清0。 0：无复位 1：复位I2C2
23	保留	必须保持复位值。
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	UART1RST	UART1复位 由软件置1或清0。 0：无复位 1：复位UART1
16:15	保留	必须保持复位值。
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
7:6	保留	必须保持复位值。
5	TIMER6RST	TIMER6定时器复位 由软件置1或清0。 0：无复位 1：复位TIMER6
4	TIMER5RST	TIMER5定时器复位 由软件置1或清0。 0：无复位 1：复位TIMER5
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
0	TIMER1RST	TIMER1定时器复位 由软件置1或清0。 0：无复位 1：复位TIMER1

7.3.6. 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							NEW_FEAT_EN	保留	PFEN	保留.			PBEN	PAEN	保留
							rw		rw				rw	rw	

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

位/位域	名称	描述
31:25	保留	必须保持复位值。
24	NEW_FEAT_EN	新特性时钟使能 0：新特性时钟关闭 1：新特性时钟开启
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
21:19	保留	必须保持复位值。
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	保留	必须保持复位值。
15	CLAEN	CLA时钟使能 由软件置1或清0。 0：CLA时钟关闭 1：CLA时钟开启
14:7	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	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	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

7.3.7. 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									DBGMCUEN	保留			TIMER16EN	TIMER15EN	TIMER14EN
									rw				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
31:23	保留	必须保持复位值。
22	DBGMCUEN	DBGMCU时钟使能 由软件置1或清0。 0：关闭DBGMCU时钟

		1：开启DBGMCU时钟
21:19	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

7.3.8. 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
DAC1EN	保留	DAC0EN	PMUEN	保留			I2C2EN	保留	I2C1EN	I2C0EN	保留			UART1EN	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
保留	SPI1EN	保留		WWDGTEN	保留		TIMER13EN	保留		TIMER6EN	TIMER5EN	保留		TIMER2EN	TIMER1EN
	rw			rw			rw			rw	rw			rw	rw

位/位域	名称	描述
31	DAC1EN	DAC1时钟使能 由软件置1或清0。 0：关闭DAC1时钟 1：开启DAC1时钟
30	保留	必须保持复位值。
29	DAC0EN	DAC0时钟使能 由软件置1或清0。 0：关闭DAC0时钟 1：开启DAC0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
27:25	保留	必须保持复位值。
24	I2C2EN	I2C2时钟使能 由软件置1或清0。 0：关闭I2C2时钟 1：开启I2C2时钟
23	保留	必须保持复位值。
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	UART1EN	UART1时钟使能 由软件置1或清0。 0:关闭UART1时钟 1:开启UART1时钟
16:15	保留	必须保持复位值。
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
7:6	保留	必须保持复位值。
5	TIMER6EN	TIMER6定时器时钟使能 由软件置1或清0。 0：关闭TIMER6定时器时钟 1：开启TIMER6定时器时钟
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。

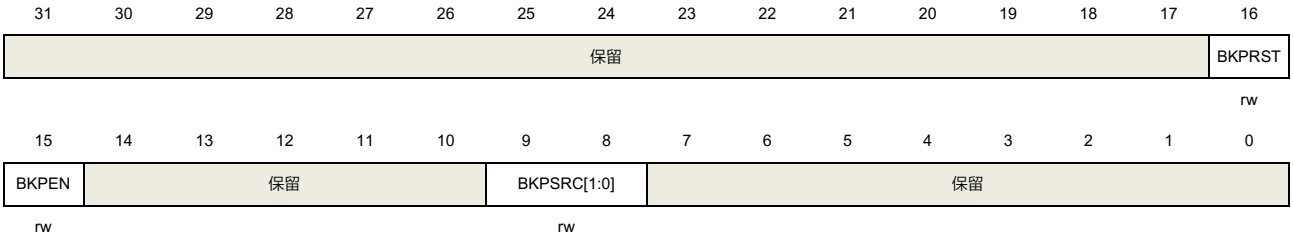
		0：关闭TIMER2定时器时钟 1：开启TIMER2定时器时钟
0	TIMER1EN	TIMER1定时器时钟使能 由软件置1或清0。 0：关闭TIMER1定时器时钟 1：开启TIMER1定时器时钟

7.3.9. 备份域控制寄存器(RCU_BDCTL)

地址偏移：0x20
复位值：0x0000 0000，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BKPSRC和BKPEN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。



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	BKPEN	BKP时钟使能 由软件置1或清0。 0：关闭BKP时钟 1：开启BKP时钟
14:10	保留	必须保持复位值。
9:8	BKPSRC[1:0]	BKP时钟入口选择 软件置位或清除来控制BKP时钟源。 00：没有时钟 01：选择(CK_AHB / 4)时钟作为BKP时钟源 10：选择CK_IRC40K时钟作为BKP时钟源 11：选择CK_HXTAL时钟32分频作为BKP时钟源
7:0	保留	必须保持复位值。

7.3.10. 复位源/时钟寄存器(RCU_RSTSCK)

地址偏移：0x24

复位值：0x0C80 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						
r	r	r	r	r	r	r	rw	r							

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1到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1到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1到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1到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1到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1到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	选项字节装载器复位标志位 选项字节装载器装载字节时由硬件置1。 由软件通过写1到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	V12域电源复位标志位 当有V12域电源复位发生时由硬件置1。 由软件通过写1到RSTFC位来清除该位。 0：无V12域电源复位发生 1：发生V12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

7.3.11. 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							NEW_FEAT_RST	保留	PFRST	保留			PBRST	PARST	保留
							rw		rw				rw	rw	

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

位/位域	名称	描述
31:25	保留	必须保持复位值。
24	NEW_FEAT_RST	新特性复位 由软件置 1 或清 0。 0: 无作用 1: 复位新特性
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
21:19	保留	必须保持复位值。
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	保留	必须保持复位值。
15	CLARST	CLA复位 由软件置1或清0。 0：无作用 1：复位CLA
14:0	保留	必须保持复位值。

7.3.12. 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												PREDV[3:0]			
												rw			

位/位域	名称	描述
31:4	保留	必须保持复位值。
3:0	PREDV[3:0]	HXTAL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。 **注意：**PREDV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，CK_HXTAL时钟分频因子为(PREDV + 1) 0000：PLL的输入，不分频 0001：PLL的输入2分频 0010：PLL的输入3分频 0011：PLL的输入4分频 0100：PLL的输入5分频 0101：PLL的输入6分频 0110：PLL的输入7分频 0111：PLL的输入8分频 1000：PLL的输入9分频 1001：PLL的输入10分频 1010：PLL的输入11分频 1011：PLL的输入12分频 1100：PLL的输入13分频 1101：PLL的输入14分频 1110：PLL的输入15分频 1111：PLL的输入16分频

7.3.13. 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	ADCPSC[2]	ADCPSC的位2 参考RCU_CFG0的位15:14
30:17	保留	必须保持复位值。
16	IRC28MDIV	IRC28M是否分频 0：IRC28M/2作为ADC时钟

		1：IRC28M作为ADC时钟
15:9	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IRC28M时钟 1：ADC时钟源选择APB2的2,4,6,8分频和AHB时钟3,5,7,9分频
7:6	保留	必须保持复位值。
5:4	I2C2SEL[1:0]	I2C2时钟源选择 由软件置1或清0。 00：I2C2时钟选择IRC8M时钟 01：I2C2时钟选择系统时钟 10：I2C2时钟选择HXTAL时钟 11：I2C2时钟选择APB1时钟
3:2	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：保留 11：USART0时钟选择IRC8M时钟

7.3.14. 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。

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
IRC28MCALIB[7:0]								IRC28MADJ[4:0]					保留	IRC28MSTB	IRC28MEN
r								rw						r	rw

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:8	IRC28MCALIB[7:0]	IRC28M时钟校准值寄存器 上电时自动加载这些位。
7:3	IRC28MADJ[4:0]	IRC28M时钟调整

		这些位由软件设定，最终调整值为(IRC28MADJ)位的值加上IRC28MCALIB[7:0]位的值。最终调整值应该把IRC28M调整到28MHz +/-1%。
2	保留	必须保持复位值。
1	IRC28MSTB	IRC28M时钟稳定标志位 由硬件置1来指示内部IRC28M振荡器已经稳定可用。 0：IRC28M振荡器未稳定 1：IRC28M振荡器已稳定
0	IRC28MEN	IRC28M时钟使能 软件置1和清0。 0：关闭IRC28M时钟 1：开启IRC28M时钟

7.3.15. 电压秘钥寄存器(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。

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

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

位/位域	名称	描述
31:0	KEY[31:0]	RCU_DSV寄存器的秘钥 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_DSV寄存器才能被写。

7.3.16. 深度睡眠模式电压寄存器(RCU_DSV)

地址偏移：0x134
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														DSLPVS[1:0]	
														rw	

位/位域	名称	描述
31:2	保留	必须保持复位值。
1:0	DSLPVS[1:0]	深度睡眠模式电压选择 这些位由软件置位和清除。 00：在深度睡眠模式下内核电压为1.2V 01：在深度睡眠模式下内核电压为1.1V 10：在深度睡眠模式下内核电压为1.0V 11：在深度睡眠模式下内核电压为0.9V

8. 通用输入/输出接口(GPIO)

8.1. 概述

该设备有多达 32 个通用 I / O 引脚(GPIO)，分别为 PA0~PA15，PB0~PB11，PF0~PF3。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。设备的 GPIO 引脚上的外部中断在中断/事件控制器单元(EXTI)中具有相关的控制和配置寄存器。

GPIO 端口和其他的备用功能(AFs)备用引脚，在特定的封装下获得最大的灵活性。GPIO 引脚通过配置相关的寄存器可以用作备用功能引脚，备用功能输入/输出都可以。在 GD32E251xx 中，GPIO 引脚可以配置为功能引脚，具有灵活交叉编码器和专用数字解码器功能。

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。

8.2. 特性

- 施输入/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；
- 单周期输出翻转功能；
- 系统复位端口状态保持功能(电源复位除外)；
- 两个特殊的中断向量，具有直接引脚中断源-仅适用于 GD32E251xx；
- 外设功能引脚分配的交叉条形解码器-仅适用于GD32E251xx；
- 包含直接引脚中断源，共有 22 个共享中断向量-仅适用于 GD32E251xx。

8.3. 功能概述

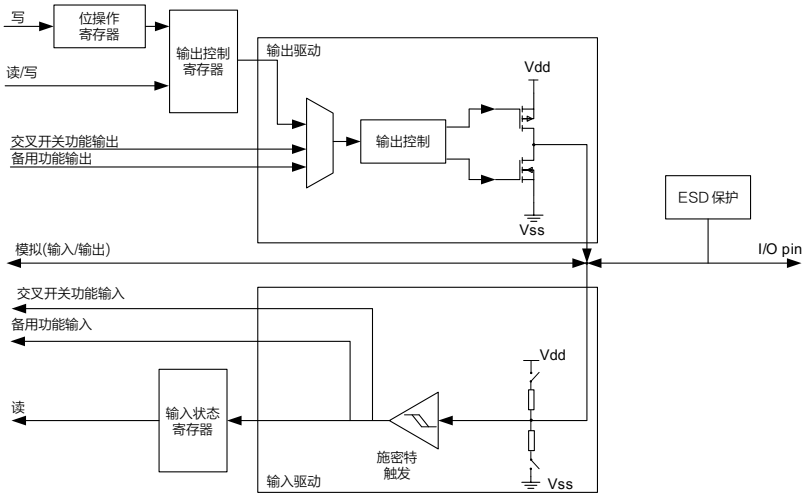
每个通用I/O端口都可以通过32位控制寄存器(GPIOx_CTL)配置为GPIO输入，GPIO输出，AF功能或模拟模式。引脚AFIO输入/输出是通过AFIO功能使能来选择。当端口配置为输出(GPIO输出或AFIO输出)时，可以通过GPIO输出模式寄存器(GPIOx_OMODE)配置为推挽或开漏模式。输出端口的最大速度可以通过GPIO输出速度寄存器(GPIOx_OSPD)配置。每个端口可以通过GPIO上/下拉寄存器(GPIOx_PUD)配置为浮空(无上拉或下拉)，上拉或下拉功能。

表 8-1.GPIO 配置表

PAD TYPE			CTLn	OMn	PUDn
GPIO 输入	X	悬空	000	X	00
		上拉			01
		下拉			10
GPIO 输出	推挽	悬空	001	0	00
		上拉			01
		下拉			10
	开漏	悬空		1	00
		上拉			01
		下拉			10
AFIO 输入	X	悬空	010	X	00
		上拉			01
		下拉			10
AFIO 输出	推挽	悬空	010	0	00
		上拉			01
		下拉			10
	开漏	悬空		1	00
		上拉			01
		下拉			10
ANALOG	X	X	011	X	XX
Crossbar 输入	X	悬空	100	X	00
		上拉			01
		下拉			10
Crossbar 输出	推挽	悬空	100	0	00
		上拉			01
		下拉			10
	开漏	悬空		1	00

*图 8-1.通用输入/输出的基本结构*为标准 I/O 端口位的基本结构图。

图 8-1.通用输入/输出的基本结构



8.3.1. GPIO 引脚配置

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 AF PU/PD 模式。

GD32E252xx：

PA14: SWCLK 在 AF 下拉模式

PA13: SWDIO 在 AF 上拉模式

GD32E251xx

PF2: SWCLK 在 AF 下拉模式

PB11：SWDIO 在 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。

8.3.2. 外部中断/事件线路

所有端口都具有外部中断功能。要使用外部中断线，端口必须被配置为输入模式。

8.3.3. 备用功能(AF)

当端口配置为AFIO(设置GPIOx_CTL寄存器中的CTLy值为"0b10")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。

8.3.4. 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附加功能时，引脚必须配置成模拟模式。当引脚用作BKP、WKUPx和振荡器附加功能时，端口类型通过相关的BKP、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。

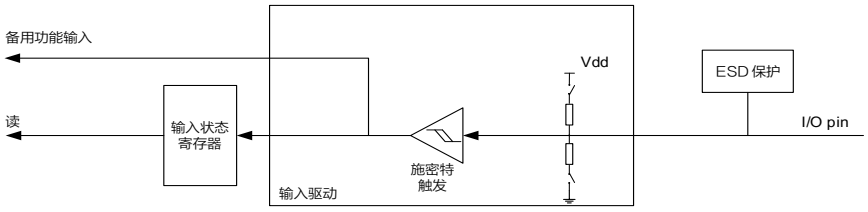
8.3.5. 输入配置

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 8-2. 输入配置的基本结构*显示 I/O 引脚的输入配置。

图 8-2. 输入配置的基本结构



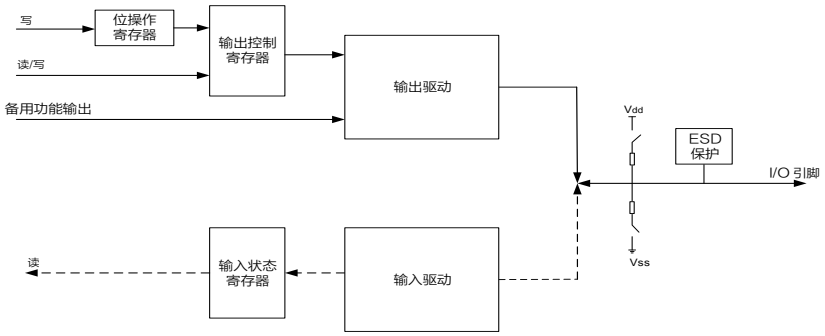
8.3.6. 输出配置

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 8-3. 输出配置的基本结构*输出配置的基本结构显示了 GPIO 引脚的输出配置。

图 8-3. 输出配置的基本结构



8.3.7. 模拟配置

当 GPIO 引脚用作模拟配置时：

- 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 8-4. 模拟配置的基本结构*是 I/O 端口的模拟模式配置。

图 8-4. 模拟配置的基本结构



8.3.8. 备用功能(AF)配置

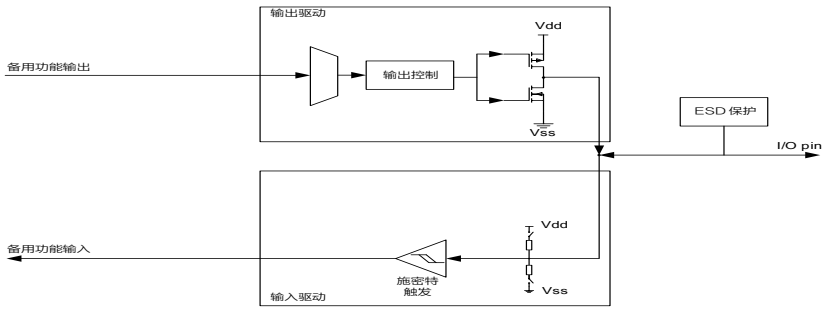
为了适应不同的器件封装，GPIO端口支持软件配置将一些备用功能应用到其他引脚上。

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 8-5. 备用功能配置的基本结构*是 I/O 端口备用功能配置图

图 8-5. 备用功能配置的基本结构



8.3.9. 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。通过32位锁定寄存器(GPIOx_LOCK)可以锁定I/O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。

8.3.10. 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。

8.3.11. GPIO 复位保持

当在 SYSCFG_CFG1 寄存器中设置了 PIN_RSTMD 位时，以下寄存器值在系统复位后保持不变，并且在电源复位时将被设置为复位值。当在 SYSCFG_CFG1 寄存器中清除了 PIN_RSTMD 位时，以下寄存器值在系统复位后将被设置为复位值。

8.3.12. 优先级交叉开关解码器(仅适用于 GD32E251xx)

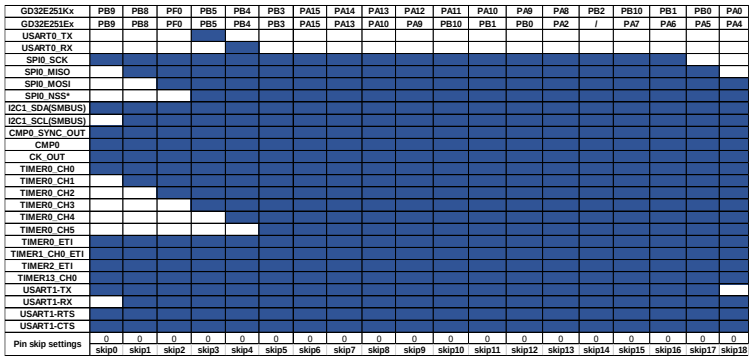
优先级交叉开关解码器专门用于分配外设接口，并在分配引脚到外设 IO 时遵循优先级。GD32E252xx 不支持此功能，外设接口分配使用 AF 功能。GD32E251xx 支持此功能，除了 PB11 和 PF2，如果 PA0-PA15、PB0-PB10、PF0 和 PF3 中的任何一个 Pad 配置了 AF 模式，则交叉开关解码器将会被屏蔽，该功能无法使用。

图 8-6. 数字功能图

GD32E251Kx	PB8	PB3	PA15	PA14	PA11	PA10	PA9	PA8	PB10	PB1	PB0	PA0	PA2	PA7	PA6	PA5	PA4	PF2
GD32E251Ex	PB8	PB3	PA15	PA13	PB10	PB1	PB0	PA2	PA7	PA6	PA5	PA4						
GD32E251Ex	CLA3OUT	EXTI_REMAP	CLA2OUT		I2C0_SDA	I2C0_CLK	CLA1OUT	CLA0OUT	DAC1_OUT1	DAC1_OUT0	DAC0_OUT1	DAC0_OUT0						
GD32E251Kx				CLA2OUT					I2C0_SDA	I2C0_SCL	CLA1OUT		CLA0OUT	DAC1_OUT1	DAC1_OUT0	DAC0_OUT1	DAC0_OUT0	SWCLK

*图 8-6. 数字功能图*显示了不同芯片封装的数字功能，有关如何启用专用数字功能的详细信息，请参考 GPIO_DFEN 寄存器。

图 8-7. 交叉开关解码器功能图



GD32E251Kx	PB9	PB8	PF0	PB5	PB4	PB3	PA15	PA14	PA13	PA12	PA11	PA10	PA9	PA8	PB2	PB10	PB1	PB0	PA0
GD32E251Ex	PB9	PB8	PF0	PB5	PB4	PB3	PA15	PA13	PA10	PA9	PB10	PB1	PB0	PA2	/	PA7	PA6	PA5	PA4
USART0_TX																			
USART0_RX																			
SPI0_SCK																			
SPI0_MISO																			
SPI0_MOSI																			
SPI0_NSS*																			
I2C1_SDA(SMBUS)																			
I2C1_SCL(SMBUS)																			
CMP0_SYNC_OUT																			
CMP0																			
CK_OUT																			
TIMER0_CH0																			
TIMER0_CH1																			
TIMER0_CH2																			
TIMER0_CH3																			
TIMER0_CH4																			
TIMER0_CH5																			
TIMER0_ETI																			
TIMER1_CH0_ETI																			
TIMER2_ETI																			
TIMER13_CH0																			
USART1-TX																			
USART1-RX																			
USART1-RTS																			
USART1-CTS																			
Pin skip settings	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	skip0	skip1	skip2	skip3	skip4	skip5	skip6	skip7	skip8	skip9	skip10	skip11	skip12	skip13	skip14	skip15	skip16	skip17	skip18

*图 8-7. 交叉开关解码器功能图*显示，蓝色部分表示外设功能可以通过交叉开关解码器映射到相应的引脚，灰色的引脚没有交叉开关解码器功能。白色部分表示外设功能不能通过交叉开关解码器映射到相应的引脚。交叉开关解码器遵循从左到右引脚优先级递减，从上到下外设优先级递减的原则。

最后一行引脚跳过设置是配置每个引脚对应的 GPIO_SKIP 寄存器。如果引脚需要使用特殊的数字/通用 I/O/模拟功能，软件将设置 GPIO_SKIP 寄存器中的 SKIPx(x = 0…18)位，与引脚相对应。

当外设分配引脚被启用时，在表格最左侧的蓝色区域中的未分配引脚会被分配给外设设备。如果一个引脚已经被更高优先级的外设功能分配占用，当为下一个外设设备分配引脚时，硬件会自动跳过该引脚，并选择下一个未分配的引脚。

使用优先级交叉解码器分配引脚的软件配置流程

(a) 如果引脚分配了模拟功能，在 GPIOx_CTL1 寄存器中将 CTLn[1:0](n = 0…15)位字段配置为"0b11"。或者如果引脚分配了 GPIO 模式，在 GPIOx_CTL1 寄存器中将 CTLn[1:0](n = 0…15)位字段配置为"0b01 / 0b00"。或者如果引脚分配了专用的数字功能，设置 GPIO_DFEN 寄存器中的相应位。然后设置 GPIO_SKIP 寄存器中的相应 SKIPx(x = 0…18)位。

(b) 除了模拟、通用 IO 功能、专用数字功能外，剩余的引脚可以通过数字解码器进行分配，在 GPIO_DECEN 寄存器中设置相应的位。

(c) 通过在 GPIO_DECEN 寄存器中设置 DECEN 位来启用交叉开关解码器。

8.3.13. INT0 和 INT1(仅适用于 GD32E251xx)

INT0 和 INT1 可以独立于交叉开关解码器分配引脚，并且 INT0 和 INT1 可能被分配到已经通过交叉开关解码器使用的引脚上。INT0 和 INT1 将在不干扰通过交叉开关解码器分配的外设引脚的情况下监控分配的引脚。

当需要只将引脚分配给 INT0/INT1 时，需要在 GPIO_SKIP 寄存器中设置相应的引脚对应的 SKIPx(x=0…18)位。

8.3.14. 引脚匹配(仅适用于 GD32E251xx)

当一个引脚用作输入时，如果该引脚的输入逻辑值与 GPIO_MASK 寄存器的对应值不等于 GPIO_MATCH 寄存器的位值与 GPIO_MASK 寄存器的位值，就会发生引脚不匹配事件，从而导致中断。

8.3.15. 引脚不匹配唤醒从深度睡眠模式(仅适用于 GD32E251xx)

无论 MATCHIE 位是设置还是复位，MCU 都可以使用不匹配中断从深度睡眠模式中唤醒。在从深度睡眠模式中唤醒之前,需要在 GPIO_MASK 和 GPIO_MATCH 寄存器中配置相应的位。

8.4. 寄存器定义

适用于 GD32E252xx 系列设备

GPIOA 基地址：0x4800 0000

GPIOB 基地址：0x4800 0400

GPIOF 基地址：0x4800 1400

适用于 GD32E251xx 系列设备

GPIOA 基地址：0x4800 0000

GPIOB 基地址：0x4800 0400

GPIOSP 基地址：0x4800 1000

GPIOF 基地址：0x4800 1400

备注：仅将 GPIOSP 基地址用于 GPIO_DECEN、GPIO_DFEN、GPIO_SKIP、GPIO_INTCTL、GPIO_MASK、GPIO_MATCH、GPIO_MATCHF 寄存器。

8.4.1. 端口控制寄存器(GPIOx_CTL，x=A, B, F)

地址偏移：0x00

GD32E252xx：

复位值：端口 A 为 0x2800 0000，其他为 0x0000 0000。

GD32E251xx：

复位值：A 的值为 0x0000 0000；B 的值为 0x0080 0000；端口 F 的值为 0x0000 0020。

注意：该寄存器可在不开启GPIO的时钟时使用。

此寄存器必须通过字(32 位)来访问。

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：GPIO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(输入和输出)

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

地址偏移：0x04

GD32E252xx：

复位值：0x0000 0000

GD32E251xx：

复位值：端口 A 为 0x0000 FFFF；端口 B 为 0x000 07FF；端口 F 为 0x0000 000B。

该寄存器必须通过字(32 位)来访问。

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
1：输出开漏模式

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

地址偏移量：0x08

GD32E252xx：

复位值：端口 A 为 0x0C00 0000；其他为 0x0000 0000。

GD32E251xx：

复位值：端口 B 为 0x00C0 0000；其他为 0x0000 0000。

注意：该寄存器可在不开启GPIO的时钟时使用。

该寄存器必须通过字(32 位)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

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

地址偏移：0x0C

GD32E252xx：

复位值：A 的值为 0x2400 0000；其他的值为 0x0000 0000。

GD32E251xx：

复位值：端口 A 为 0x5555 5555；端口 B 为 0x0055 5555；当 NRST_MDSEL_VALUE 为 1 时，端口 F 为 0x0000 0055；当 NRST_MDSEL_VALUE 为 0 时，端口 F 为 0x0000 0065。

该寄存器必须通过字(32 位)来访问。

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

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

地址偏移：0x10

复位值：0x0000 XXXX

该寄存器必须通过字(32 位)来访问。

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	端口输入状态位(y=0..15) 这些位由软件置位和清除。 0：引脚输入信号为低电平 1：引脚输入信号为高电平

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

地址偏移：0x14

复位值：0x0000 0000

这段文本需要以字(32 位)为单位进行访问。

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	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 **注意：**对于 OTC 设置/复位，ODR 位可以分别通过写入 GPIOx_BSRR/GPIOX_BRR/GPIOX_BTGR 寄存器进行设置和复位或翻转

8.4.7. 端口位操作寄存器(GPIOx_BOP，x=A, B, F)

地址偏移量：0x18

复位值：0x0000 0000

此寄存器必须通过字(32 位)来访问。

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	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 **注意：**BOPy 的优先级高于 Cry

8.4.8. 端口配置锁定寄存器(GPIOx_LOCK，x=A, B, F)

地址偏移：0x1C

复位值：0x0000 0000

这个寄存器必须通过字(32 位)来访问。

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

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

地址偏移：0x20

复位值：0x0000 0000

需要通过 32 位字(word)来访问这个寄存器。

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 0101：选择AF5功能 0110：选择AF6功能

0111：选择AF7功能
1000：选择AF8功能
1001：选择AF9功能
…
1111：选择AF15功能

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

地址偏移：0x24

复位值：0x0000 0000

该寄存器必须通过字(32 位)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 1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 0101：选择AF5功能 0110：选择AF6功能 0111：选择AF7功能 1000：选择AF8功能 1001：选择AF9功能 … 1111：选择 AF15 功能

8.4.11. 位清除寄存器(GPIOx_BC，x=A, B, F)

地址偏移：0x28

复位值：0x0000 0000

该寄存器必须通过字(32 位)来访问。

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

8.4.12. 端口位翻转寄存器(GPIOx_TG，x=A, B, F)

地址偏移：0x2C

复位值：0x0000 0000

此寄存器必须通过字(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
TG15	TG14	TG13	TG12	TG11	TG10	TG9	TG8	TG7	TG6	TG5	TG4	TG3	TG2	TG1	TG0
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	TGy	端口翻转位y(y=0..15) 该位由软件置位和清除。 0：相应OCTLy位没有改变 1：翻转相应的OCTLy位

8.4.13. 交叉开关使能寄存器(GPIO_CBREN)

地址偏移：0x34

复位值：0x0000 0000

此寄存器必须通过字(32 位)来访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留												CBAREN	CUSRT1CTSEN	CUSRT1RTSEN	CUART1EN
												rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	CTIMER13EN	CTIMER2ETIEN	CTIMER1ETIEN	CTIMER0ETIEN	CTIMER0MOD			CSYSCKEN	保留		CCMP0AEN	CCMP0EN	CSMBEN	CSPI0EN	CUSART0EN
rw	rw	rw	rw	rw	rw rw			rw	rw		rw	rw	rw	rw	rw

位/位域	名称	描述
31:20	保留	必须保持复位值
19	CBAREN	交叉开关使能 该位由软件置位和清除。 0: 交叉开关关闭 1: 交叉开关使能
18	CUART1CTSEN	UART1 CTS输出使能 该位由软件置位和清除。 0: UART1 CTS端口引脚不可用 1: UART1 CTS 使能端口引脚
17	CUART1RTSEN	UART1 RTS输出使能

		该位由软件置位和清除。 0: UART1 RTS端口引脚不可用 1: UART1 RTS 使能端口引脚
16	CUART1EN	Crossbar UART1 I/O使能 该位由软件置位和清除。 0: UART1 I/O 端口引脚不可用 1: UART1 TX RX 使能端口引脚
15	保留	必须保持复位值
14	CTIMER13EN	Crossbar TIMER13通道0使能 该位由软件置位和清除。 0: TIMER13通道0端口引脚不可用 1: TIMER13 通道 0 使能端口引脚
13	CTIMER2ETIEN	Crossbar TIMER2通道0外部触发器使能 该位由软件置位和清除。 0: TIMER2_ETR通道0端口引脚不可用 1: TIMER2_ETR 通道 0 使能端口引脚
12	CTIMER1ETIEN	Crossbar TIMER1通道0外部触发器使能 该位由软件置位和清除。 0: TIMER1_ETR通道0端口引脚不可用 1: TIMER1_ETR 通道 0 使能端口引脚
11	CTIMER0ETIEN	Crossbar TIMER0外部计数器输入使能 该位由软件置位和清除。 0: TIMER0_ETR端口引脚不可用 1: TIMER0_ETR使能端口引脚
10:8	CTIMER0MOD	Crossbar TIMER0 模块 I/O 使能 该位由软件置位和清除。 000: 所有TIMER0 I/O 端口引脚不可用 001: CH0 使能端口引脚 010: CH0, CH1 使能端口引脚 011: CH0, CH1, CH2 使能端口引脚 100: CH0, CH1, CH2, CH3 使能端口引脚 101 : CH0, CH1, CH2, CH3, CH4 使能端口引脚 110 : CH0, CH1, CH2, CH3, CH4, CH5 使能端口引脚
7	CSYSCKEN	Crossbar SYSCLK 输出使能 该位由软件置位和清除。 0: SYSCLK端口引脚不可用 1: SYSCLK 使能端口引脚
6:5	保留	必须保持复位值
4	CCMP0AEN	Crossbar Comparator0 异步输出使能

		该位由软件置位和清除。 0: 异步CMP0端口引脚不可用 1: 异步 CMP0 使能端口引脚
3	CCMP0EN	Crossbar Comparator0 输出使能 该位由软件置位和清除。 0: CMP0端口引脚不可用 1: CMP0 使能端口引脚
2	CSMBEN	Crossbar SMBus I/O使能 该位由软件置位和清除。 0: SMBus I/O端口引脚不可用 1: SMBus I/O 使能端口引脚
1	CSPI0EN	Crossbar SPI0 I/O 使能 该位由软件置位和清除。 0: SPI0 I/O 端口引脚不可用 1: SPI0 I/O 使能端口引脚
0	CUSART0EN	Crossbar USART0 I/O 使能 该位由软件置位和清除。 0：USART0 I/O端口引脚不可用 1：USART0 TX, RX 使能端口

8.4.14. 数字功能使能寄存器(GPIO_DFEN)

地址偏移：0x38

复位值：0x0000 0000

此寄存器必须通过字(32 位)来访问。

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											CLA3PINEN	CLA2PINEN	CLA1PINEN	CLA0PINEN	I2C0PINEN
											rw	rw	rw	rw	rw

位/位域	名称	描述
31:5	保留	必须保持复位值
4	CLA3PINEN	CLA3 端口输出使能 该位由软件置位和清除。 0: 关闭异步输出所选 GPIO 引脚 1: 使能异步输出所选 GPIO 引脚
3	CLA2PINEN	CLA2 端口输出使能

		该位由软件置位和清除。 0: 关闭异步输出所选 GPIO 引脚 1: 使能异步输出所选 GPIO 引脚
2	CLA1PINEN	CLA1 端口输出使能 该位由软件置位和清除。 0: 关闭异步输出所选 GPIO 引脚 1: 使能异步输出所选 GPIO 引脚
1	CLA0PINEN	CLA0 端口输出使能 该位由软件置位和清除。 0: 关闭异步输出所选 GPIO 引脚 1: 使能异步输出所选 GPIO 引脚
0	I2C0PINEN	I2C0 引脚模式使能 该位由软件置位和清除。 0: 关闭 I2C0 所选 GPIO 引脚 1: 使能 I2C0 所选 GPIO 引脚 **注意:** AF 功能未启用，此位使能 I2C0 输入和输出引脚。

8.4.15. 交叉开关跳过寄存器(GPIO_SKIP)

地址偏移：0x3C

复位值：0x0000 0000

该寄存器必须通过字(32 位)来访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留													SKIP18	SKIP17	SKIP16
													rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SKIP15	SKIP14	SKIP13	SKIP12	SKIP11	SKIP10	SKIP9	SKIP8	SKIP7	SKIP6	SKIP5	SKIP4	SKIP3	SKIP2	SKIP1	SKIP0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:19	保留	必须保持复位值
18	SKIP18	SKIP18 值 该位由软件置位和清除。 0：PA0(适用于 GD32E251KX)/PA4(适用于 GD32E251EX)引脚未被交叉开关跳过 1：PA0(适用于 GD32E251KX)/PA4(适用于 GD32E251EX)引脚被交叉开关跳过
17	SKIP17	SKIP17 值 该位由软件置位和清除。 0：PB0(适用于 GD32E251KX)/PA5(适用于 GD32E251EX)引脚未被交叉开关跳过 1：PB0(适用于 GD32E251KX)/PA5(适用于 GD32E251EX)引脚被交叉开关跳过

16	SKIP16	SKIP16 值 该位由软件置位和清除。 0：PB1(适用于 GD32E251KX)/PA6(适用于 GD32E251EX)引脚未被交叉开关跳过 1：PB1(适用于 GD32E251KX)/PA6(适用于 GD32E251EX)引脚被交叉开关跳过
15	SKIP15	SKIP15 值 该位由软件置位和清除。 0：PB10(适用于 GD32E251KX)/PA7(适用于 GD32E251EX)引脚未被交叉开关跳过 1：PB10(适用于 GD32E251KX)/PA7(适用于 GD32E251EX)引脚被交叉开关跳过
14	SKIP14	SKIP14 值 该位由软件置位和清除。 0：PB2(适用于 GD32E251KX)引脚未被交叉开关跳过 1：PB2(适用于 GD32E251KX)引脚被交叉开关跳过
13	SKIP13	SKIP13 值 该位由软件置位和清除。 0：PA8(适用于 GD32E251KX)/PA2(适用于 GD32E251EX)引脚未被交叉开关跳过 1：PA8 (适用于 GD32E251KX)/PA2(适用于 GD32E251EX)引脚被交叉开关跳过
12	SKIP12	SKIP12 值 该位由软件置位和清除。 0：PA9(适用于 GD32E251KX)/PB0(适用于 GD32E251EX)引脚未被交叉开关跳过 1：PA9(适用于 GD32E251KX)/PB0(适用于 GD32E251EX)引脚被交叉开关跳过
11	SKIP11	SKIP11 值 该位由软件置位和清除。 0：PA10(适用于 GD32E251KX)/PB1(适用于 GD32E251EX)引脚未被交叉开关跳过 1：PA10(适用于 GD32E251KX)/PB1(适用于 GD32E251EX)引脚被交叉开关跳过
10	SKIP10	SKIP10 值 该位由软件置位和清除。 0：PA11(适用于 GD32E251KX)/PB10(适用于 GD32E251EX)引脚未被交叉开关跳过 1：PA11(适用于 GD32E251KX)/PB10(适用于 GD32E251EX)引脚被交叉开关跳过
9	SKIP9	SKIP9 值 该位由软件置位和清除。 0：PA12(适用于 GD32E251KX)/PA9(适用于 GD32E251EX)引脚未被交叉开关跳过 1：PA12(适用于 GD32E251KX)/PA9(适用于 GD32E251EX)引脚被交叉开关跳过
8	SKIP8	SKIP8 值 该位由软件置位和清除。 0：PA13(适用于 GD32E251KX)/PA10(适用于 GD32E251EX)引脚未被交叉开关跳过

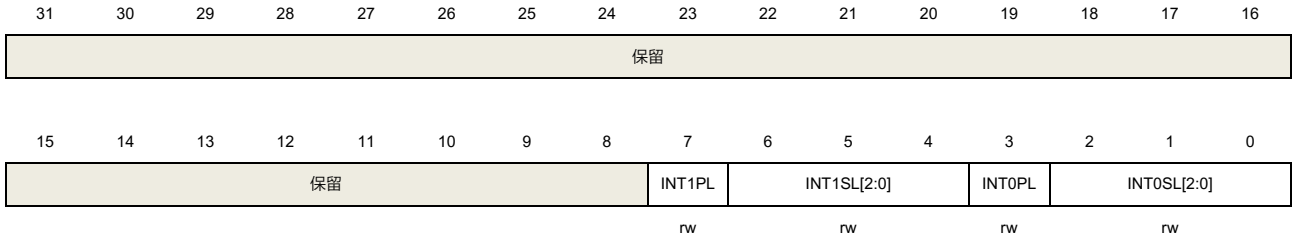
		1：PA13(适用于 GD32E251KX)/PA10(适用于 GD32E251EX)引脚被交叉开关跳过
7	SKIP7	SKIP7 值 该位由软件置位和清除。 0：PA14(适用于 GD32E251KX)/PA13(适用于 GD32E251EX)引脚未被交叉开关跳过 1：PA14(适用于 GD32E251KX)/PA13(适用于 GD32E251EX)引脚被交叉开关跳过
6	SKIP6	SKIP6 值 该位由软件置位和清除。 0：PA15 引脚未被交叉开关跳过 1：PA15 引脚被交叉开关跳过
5	SKIP5	SKIP5 值 该位由软件置位和清除。 0：PB3 引脚未被交叉开关跳过 1：PB3 引脚被交叉开关跳过
4	SKIP4	SKIP4 值 该位由软件置位和清除。 0：PB4 引脚未被交叉开关跳过 1：PB4 引脚被交叉开关跳过
3	SKIP3	SKIP3 值 该位由软件置位和清除。 0：PB5 引脚未被交叉开关跳过 1：PB5 引脚被交叉开关跳过
2	SKIP2	SKIP2 值 该位由软件置位和清除。 0：PF0 引脚未被交叉开关跳过 1：PF0 引脚被交叉开关跳过
1	SKIP1	SKIP1 值 该位由软件置位和清除。 0：PB8 引脚未被交叉开关跳过 1：PB8 引脚被交叉开关跳过
0	SKIP0	SKIP0 值 该位由软件置位和清除。 0：PB9 引脚未被交叉开关跳过 1：PB9 引脚被交叉开关跳过 **注意：** *图 8-7. 交叉开关解码器功能图*所示引脚与 SKIP 的对应关系

8.4.16. INT0 和 INT1 控制寄存器(GPIO_INTCTL)

地址偏移：0x40

复位值：0x0000 0001

这个寄存器必须通过字(32 位)来访问。



位/位域	名称	描述
31:8	保留	必须保持复位值
7	INT1PL	INT1 电平 该位由软件置位和清除。 0: INT1 输入低电平 1: INT1 输入高电平
6:4	INT1SL[2:0]	INT1 端口引脚选择 该位由软件置位和清除。 000: 选择 PB9 001: 选择 PB8 010: 选择 PF0 011: 选择 PB5 100: 选择 PB4 101: 选择 PB3 110: 选择 PA15 111: 保留
3	INT0PL	INT0 电平 该位由软件置位和清除。 0: INT0 输入低电平 1: INT0 输入高电平
2:0	INT0SL[2:0]	INT0 端口引脚选择 该位由软件置位和清除。 000: 选择PB9 001: 选择PB8 010: 选择 PF0 011: 选择PB5 100: 选择PB4 101: 选择PB3 110: 选择 PA15 111: 保留

8.4.17. 引脚屏蔽寄存器(GPIO_MASK)

地址偏移：0x44

复位值：0x0000 0000

该寄存器必须通过字(32 位)来访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留										MSK21	MSK20	MSK19	MSK18	MSK17	MSK16
										rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSK15	MSK14	MSK13	MSK12	MSK11	MSK10	MSK9	MSK8	MSK7	MSK6	MSK5	MSK4	MSK3	MSK3	MSK1	MSK0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:22	保留	必须保持复位值
21	MSK21	MSK21 值 该位由软件置位和清除。 0：PA3(仅适用于 GD32E251KX)逻辑值被忽略，不会引起端口不匹配事件 1：PA3(仅适用于 GD32E251KX)逻辑值与寄存器 MSK21 进行比较
20	MSK20	MSK20 值 该位由软件置位和清除。 0：PA2(仅适用于 GD32E251KX)逻辑值被忽略，不会引起端口不匹配事件 1：PA2(仅适用于 GD32E251KX)逻辑值与寄存器 MSK20 进行比较
19	MSK19	MSK19 值 该位由软件置位和清除。 0：PA1(仅适用于 GD32E251KX)逻辑值被忽略，不会引起端口不匹配事件 1：PA1(仅适用于 GD32E251KX)逻辑值与寄存器 MSK19 进行比较
18	MSK18	MSK18 值 该位由软件置位和清除。 0：PA0(适用于 GD32E251KX) / PA4(适用于 GD32E251EX)逻辑值被忽略，不会引起端口不匹配事件 1：PA0(适用于 GD32E251KX) / PA4(适用于 GD32E251EX)逻辑值与寄存器 MSK18 进行比较
17	MSK17	MSK17 值 该位由软件置位和清除。 0：PB0(适用于 GD32E251KX) / PA5(适用于 GD32E251EX)逻辑值被忽略，不会引起端口不匹配事件 1：PB0(适用于 GD32E251KX) / PA5(适用于 GD32E251EX)逻辑值与寄存器 MSK17 进行比较
16	MSK16	MSK16 值 该位由软件置位和清除。

		0：PB1(适用于 GD32E251KX) / PA6(适用于 GD32E251EX)逻辑值被忽略，不会引起端口不匹配事件 1：PB1(适用于 GD32E251KX) / PA6(适用于 GD32E251EX)逻辑值与寄存器 MSK16 进行比较
15	MSK15	MSK15 值 该位由软件置位和清除。 0：PB10(适用于 GD32E251KX) / PA7(适用于 GD32E251EX)逻辑值被忽略，不会引起端口不匹配事件 1：PB10(适用于 GD32E251KX) / PA7(适用于 GD32E251EX)逻辑值与寄存器 MSK15 进行比较
14	MSK14	MSK14 值 该位由软件置位和清除。 0：PB2(仅适用于 GD32E251KX)逻辑值被忽略，不会引起端口不匹配事件 1：PB2(仅适用于 GD32E251KX)逻辑值与寄存器 MSK14 进行比较
13	MSK13	MSK13 值 该位由软件置位和清除。 0：PA8(适用于 GD32E251KX) / PA2(适用于 GD32E251EX)逻辑值被忽略，不会引起端口不匹配事件 1：PA8(适用于 GD32E251KX) / PA2(适用于 GD32E251EX)逻辑值与寄存器 MSK13 进行比较
12	MSK12	MSK12 值 该位由软件置位和清除。 0：PA9(适用于 GD32E251KX) / PB0(适用于 GD32E251EX)逻辑值被忽略，不会引起端口不匹配事件 1：PA9(适用于 GD32E251KX) / PB0(适用于 GD32E251EX)逻辑值与寄存器 MSK12 进行比较
11	MSK11	MSK11 值 该位由软件置位和清除。 0：PA10(适用于 GD32E251KX) / PB1(适用于 GD32E251EX)逻辑值被忽略，不会引起端口不匹配事件 1：PA10(适用于 GD32E251KX) / PB1(适用于 GD32E251EX)逻辑值与寄存器 MSK11 进行比较
10	MSK10	MSK10 值 该位由软件置位和清除。 0：PA11(适用于 GD32E251KX) / PB10(适用于 GD32E251EX)逻辑值被忽略，不会引起端口不匹配事件 1：PA11(适用于 GD32E251KX) / PB10(适用于 GD32E251EX)逻辑值与寄存器 MSK10 进行比较
9	MSK9	MSK9 值 该位由软件置位和清除。 0：PA12(适用于 GD32E251KX) / PA9(适用于 GD32E251EX)逻辑值被忽略，不会

		引起端口不匹配事件 1：PA12(适用于 GD32E251KX) / PA9(适用于 GD32E251EX)逻辑值与寄存器 MSK9 进行比较
8	MSK8	MSK8 值 该位由软件置位和清除。 0：PA13(适用于 GD32E251KX) / PA10(适用于 GD32E251EX)逻辑值被忽略，不会引起端口不匹配事件 1：PA13(适用于 GD32E251KX) / PA10(适用于 GD32E251EX)逻辑值与寄存器 MSK8 进行比较
7	MSK7	MSK7 值 该位由软件置位和清除。 0：PA14(适用于 GD32E251KX) / PA13(适用于 GD32E251EX)逻辑值被忽略，不会引起端口不匹配事件 1：PA14(适用于 GD32E251KX) / PA13(适用于 GD32E251EX)逻辑值与寄存器 MSK7 进行比较
6	MSK6	MSK6 值 该位由软件置位和清除。 0：PA15 逻辑值被忽略，不会引起端口不匹配事件 1：PA15 逻辑值与寄存器 MSK6 进行比较
5	MSK5	MSK5 值 该位由软件置位和清除。 0：PB3 逻辑值被忽略，不会引起端口不匹配事件 1：PB3 逻辑值与寄存器 MSK5 进行比较
4	MSK4	MSK4 值 该位由软件置位和清除。 0：PB4 逻辑值被忽略，不会引起端口不匹配事件 1：PB4 逻辑值与寄存器 MSK4 进行比较
3	MSK3	MSK3 值 该位由软件置位和清除。 0：PB5 逻辑值被忽略，不会引起端口不匹配事件 1：PB5 逻辑值与寄存器 MSK3 进行比较
2	MSK2	MSK2 值 该位由软件置位和清除。 0：PF0 逻辑值被忽略，不会引起端口不匹配事件 1：PF0 逻辑值与寄存器 MSK2 进行比较
1	MSK1	MSK1 值 该位由软件置位和清除。 0：PB8 逻辑值被忽略，不会引起端口不匹配事件 1：PB8 逻辑值与寄存器 MSK1 进行比较
0	MSK0	MSK0 值

该位由软件置位和清除。
0：PB9 逻辑值被忽略，不会引起端口不匹配事件
1：PB9 逻辑值与寄存器 MSK0 进行比较

8.4.18. 引脚匹配寄存器(GPIO_MATCH)

地址偏移：0x48

复位值：0x0000 0000

该寄存器必须通过字(32 位)来访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留										MATCH21	MATCH20	MATCH19	MATCH18	MATCH17	MATCH16
										rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MATCH15	MATCH14	MATCH13	MATCH12	MATCH11	MATCH10	MATCH9	MATCH8	MATCH7	MATCH6	MATCH5	MATCH4	MATCH3	MATCH2	MATCH1	MATCH0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:22	保留	必须保持复位值
21	MATCH21	MATCH21 值 该位由软件置位和清除。 0：PA3(仅适用于 GD32E251KX)逻辑值与逻辑低进行比较 1：PA3(仅适用于 GD32E251KX)逻辑值与逻辑高进行比较
20	MATCH20	MATCH20 值 该位由软件置位和清除。 0：PA2(仅适用于 GD32E251KX)逻辑值与逻辑低进行比较 1：PA2(仅适用于 GD32E251KX)逻辑值与逻辑高进行比较
19	MATCH19	MATCH19 值 该位由软件置位和清除。 0：PA1(仅适用于 GD32E251KX)逻辑值与逻辑低进行比较 1：PA1(仅适用于 GD32E251KX)逻辑值与逻辑高进行比较
18	MATCH18	MATCH18 值 该位由软件置位和清除。 0：将 PA0(适用于 GD32E251KX) / PA4(适用于 GD32E251EX)逻辑值与逻辑低进行比较 1：将 PA0(适用于 GD32E251KX) / PA4(适用于 GD32E251EX)逻辑值与逻辑高进行比较
17	MATCH17	MATCH17 值 该位由软件置位和清除。

		0：将 PB0(适用于 GD32E251KX) / PA5(适用于 GD32E251EX)逻辑值与逻辑低进行比较 1：将 PB0(适用于 GD32E251KX) / PA5(适用于 GD32E251EX)逻辑值与逻辑高进行比较
16	MATCH16	MATCH16 值 该位由软件置位和清除。 0：将 PB1(适用于 GD32E251KX) / PA6(适用于 GD32E251EX)逻辑值与逻辑低进行比较 1：将 PB1(适用于 GD32E251KX) / PA6(适用于 GD32E251EX)逻辑值与逻辑高进行比较
15	MATCH15	MATCH15 值 该位由软件置位和清除。 0：将 PB10(适用于 GD32E251KX) / PA7(适用于 GD32E251EX)逻辑值与逻辑低进行比较 1：将 PB10(适用于 GD32E251KX) / PA7(适用于 GD32E251EX)逻辑值与逻辑高进行比较
14	MATCH14	MATCH14 值 该位由软件置位和清除。 0：PB2(仅适用于 GD32E251KX)逻辑值与逻辑低进行比较 1：PB2(仅适用于 GD32E251KX)逻辑值与逻辑高进行比较
13	MATCH13	MATCH13 值 该位由软件置位和清除。 0：将 PA8(适用于 GD32E251KX) / PA2(适用于 GD32E251EX)逻辑值与逻辑低进行比较 1：将 PA8(适用于 GD32E251KX) / PA2(适用于 GD32E251EX)逻辑值与逻辑高进行比较
12	MATCH12	MATCH12 值 该位由软件置位和清除。 0：将 PA9(适用于 GD32E251KX) / PB0(适用于 GD32E251EX)逻辑值与逻辑低进行比较 1：将 PA9(适用于 GD32E251KX) / PB0(适用于 GD32E251EX)逻辑值与逻辑高进行比较
11	MATCH11	MATCH11 值 该位由软件置位和清除。 0：将 PA10(适用于 GD32E251KX) / PB1(适用于 GD32E251EX)逻辑值与逻辑低进行比较 1：将 PA10(适用于 GD32E251KX) / PB1(适用于 GD32E251EX)逻辑值与逻辑高进行比较
10	MATCH10	MATCH10 值 该位由软件置位和清除。 0：将 PA11(适用于 GD32E251KX) / PB10(适用于 GD32E251EX)逻辑值与逻辑低

		进行比较 1：将 PA11(适用于 GD32E251KX) / PB10(适用于 GD32E251EX)逻辑值与逻辑高进行比较
9	MATCH9	MATCH9 值 该位由软件置位和清除。 0：将 PA12(适用于 GD32E251KX) / PA9(适用于 GD32E251EX)逻辑值与逻辑低进行比较 1：将 PA12(适用于 GD32E251KX) / PA9(适用于 GD32E251EX)逻辑值与逻辑高进行比较
8	MATCH8	MATCH8 值 该位由软件置位和清除。 0：将 PA13(适用于 GD32E251KX) / PA10(适用于 GD32E251EX)逻辑值与逻辑低进行比较 1：将 PA13(适用于 GD32E251KX) / PA10(适用于 GD32E251EX)逻辑值与逻辑高进行比较
7	MATCH7	MATCH7 值 该位由软件置位和清除。 0：将 PA14(适用于 GD32E251KX) / PA13(适用于 GD32E251EX)逻辑值与逻辑低进行比较 1：将 PA14(适用于 GD32E251KX) / PA13(适用于 GD32E251EX)逻辑值与逻辑高进行比较
6	MATCH6	MATCH6 值 该位由软件置位和清除。 0：PA15 逻辑值与逻辑低进行比较 1：PA15 逻辑值与逻辑高进行比较
5	MATCH5	MATCH5 值 该位由软件置位和清除。 0：PB3 逻辑值与逻辑低进行比较 1：PB3 逻辑值与逻辑高进行比较
4	MATCH4	MATCH4 值 该位由软件置位和清除。 0：PB4 逻辑值与逻辑低进行比较 1：PB4 逻辑值与逻辑高进行比较
3	MATCH3	MATCH3 值 该位由软件置位和清除。 0：PB5 逻辑值与逻辑低进行比较 1：PB5 逻辑值与逻辑高进行比较
2	MATCH2	MATCH2 值 该位由软件置位和清除。 0：PF0 逻辑值与逻辑低进行比较

1：PF0 逻辑值与逻辑高进行比较

1	MATCH1	MATCH1 值 该位由软件置位和清除。 0：PB8 逻辑值与逻辑低进行比较 1：PB8 逻辑值与逻辑高进行比较
0	MATCH0	MATCH0 值 该位由软件置位和清除。 0：PB9 逻辑值与逻辑低进行比较 1：PB9 逻辑值与逻辑高进行比较

8.4.19. 引脚匹配标志寄存器(GPIO_MATCHF)

地址偏移量：0x4C

复位值：0x0000 0000

这个寄存器必须通过字(32 位)来访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留									MATCHIE	MATCHF 21	MATCHF 20	MATCHF 19	MATCHF 18	MATCHF 17	MATCHF 16
									rw	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
MATCHF 15	MATCHF 14	MATCHF 13	MATCHF 12	MATCHF 11	MATCHF 10	MATCHF 9	MATCHF 8	MATCHF 7	MATCHF 6	MATCHF 5	MATCHF 4	MATCH 3	MATCH 2	MATCHF 1	MATCHF 0
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	MATCHIE	1：使能端口不匹配中断 0：关闭端口不匹配中断
21	MATCHF21	MATCHF21 值 端口不匹配硬件设置 1 通过软件清除 PA3(仅适用于 GD32E251KX)可以通过写入 1 将此 flag 位清除。
20	MATCHF20	MATCHF20 值 端口不匹配硬件设置 1 通过软件清除 PA2(仅适用于 GD32E251KX)可以通过写入 1 将此 flag 位清除。
19	MATCHF19	MATCHF19 值 端口不匹配硬件设置 1 通过软件清除 PA1(仅适用于 GD32E251KX)可以通过写入 1 将此 flag 位清除。
18	MATCHF18	MATCHF18 值 端口不匹配硬件设置 1

		通过软件清除 PA0(适用于 GD32E251KX) / PA4(适用于 GD32E251EX)可以通过写入 1 将此 flag 位清除。
17	MATCHF17	MATCHF17 值 端口不匹配硬件设置 1 通过软件清除 PB0(适用于 GD32E251KX) / PA5(适用于 GD32E251EX)可以通过写入 1 将此 flag 位清除。
16	MATCHF16	MATCHF16 值 端口不匹配硬件设置 1 通过软件清除 PB1(适用于 GD32E251KX) / PA6(适用于 GD32E251EX)可以通过写入 1 将此 flag 位清除。
15	MATCHF15	MATCHF15 值 端口不匹配硬件设置 1 通过软件清除 PB10(适用于 GD32E251KX) / PA7(适用于 GD32E251EX)可以通过写入 1 将此 flag 位清除。
14	MATCHF14	MATCHF14 值 端口不匹配硬件设置 1 通过软件清除 PB2(仅适用于 GD32E251KX) 可以通过写入 1 将此 flag 位清除。
13	MATCHF13	MATCHF13 值 端口不匹配硬件设置 1 通过软件清除 PA8(适用于 GD32E251KX) / PA2(适用于 GD32E251EX)可以通过写入 1 将此 flag 位清除。
12	MATCHF12	MATCHF12 值 端口不匹配硬件设置 1 通过软件清除 PA9(适用于 GD32E251KX) / PB0(适用于 GD32E251EX)可以通过写入 1 将此 flag 位清除。
11	MATCHF11	MATCHF11 值 端口不匹配硬件设置 1 通过软件清除 PA10(适用于 GD32E251KX) / PB1(适用于 GD32E251EX)可以通过写入 1 将此 flag 位清除。
10	MATCHF10	MATCHF10 值 端口不匹配硬件设置 1 通过软件清除 PA11(适用于 GD32E251KX) / PB10(适用于 GD32E251EX)可以通过写入 1 将此 flag 位清除。
9	MATCHF9	MATCHF9 值 端口不匹配硬件设置 1 通过软件清除 PA12(适用于 GD32E251KX) / PA9(适用于 GD32E251EX)可以通过写入 1 将此 flag 位清除。
8	MATCHF8	MATCHF8 值 端口不匹配硬件设置 1

		通过软件清除 PA13(适用于 GD32E251KX) / PA10(适用于 GD32E251EX)可以通过写入 1 将此 flag 位清除。
7	MATCHF7	MATCHF7 值 端口不匹配硬件设置 1 通过软件清除 PA14(适用于 GD32E251KX) / PA13(适用于 GD32E251EX)可以通过写入 1 将此 flag 位清除。
6	MATCHF6	MATCHF6 值 端口不匹配硬件设置 1 通过软件清除 PA15 可以通过写入 1 将此 flag 位清除。
5	MATCHF5	MATCHF5 值 端口不匹配硬件设置 1 通过软件清除 PB3 可以通过写入 1 将此 flag 位清除。
4	MATCHF4	MATCHF4 值 端口不匹配硬件设置 1 通过软件清除 PB4 可以通过写入 1 将此 flag 位清除。
3	MATCHF3	MATCHF3 值 端口不匹配硬件设置 1 通过软件清除 PB5 可以通过写入 1 将此 flag 位清除。
2	MATCHF2	MATCHF2 值 端口不匹配硬件设置 1 通过软件清除 PF0 可以通过写入 1 将此 flag 位清除。
1	MATCHF1	MATCHF1 值 端口不匹配硬件设置 1 通过软件清除 PB8 可以通过写入 1 将此 flag 位清除。
0	MATCHF0	MATCHF0 值 端口不匹配硬件设置 1 通过软件清除 PB9 可以通过写入 1 将此 flag 位清除。

9. 定时器(TIMER)

表 9-1. 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 5/6
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(TIMER1)/16 位(TIMER2)	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	6	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)	×	×
DMA	●	●	×	●	●	●(4)
Debug 模式	●	●	●	●	●	●

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

(2) TIMER1 **ITI0:** TIMER0_TRGO **ITI1:** TIMER14_TRGO **ITI2:** TIMER2_TRGO **IT3: 0**

TIMER2 **ITI0:** TIMER0_TRGO **ITI1:** 0 **ITI2:** TIMER14_TRGO **ITI3:** 0

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

(4) 只有更新事件可以产生 DMA 请求，定时器 5/6 中没有 DMAS 位(DMA 请求源选择位)。

9.1. 高级定时器(TIMERx,x=0)

9.1.1. 简介

高级定时器(TIMER0)是六通道定时器，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，但是它们的计数器可以被同步在一起形成一个更大的定时器。

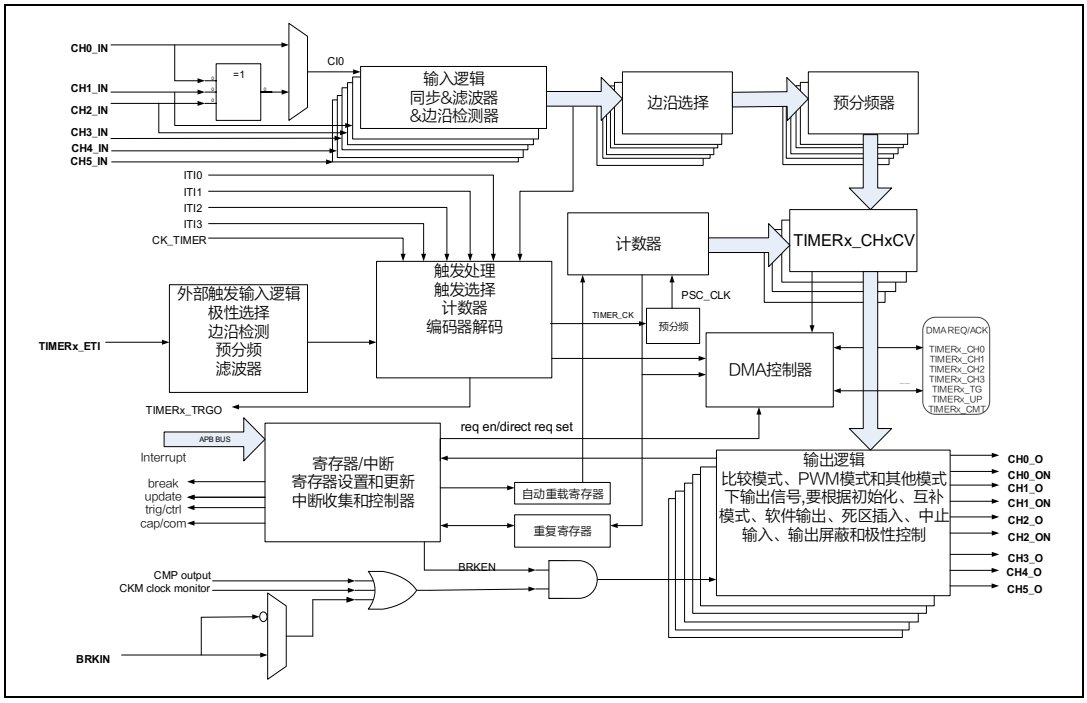
9.1.2. 主要特征

- 总通道数：6；
- 计数器宽度：16位；
- 定时器时钟源可选：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：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。

9.1.3. 结构框图

*图 9-1. 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 9-1. 高级定时器结构框图



注意：仅TIMER0支持CMP输出作为中止源输入。

9.1.4. 功能说明

时钟源配置

高级定时器的时钟源可以是内部时钟源CK_TIMER，或者是由SMC(TIMERx_SMCFG寄存器位[2:0])位确定的时钟源。

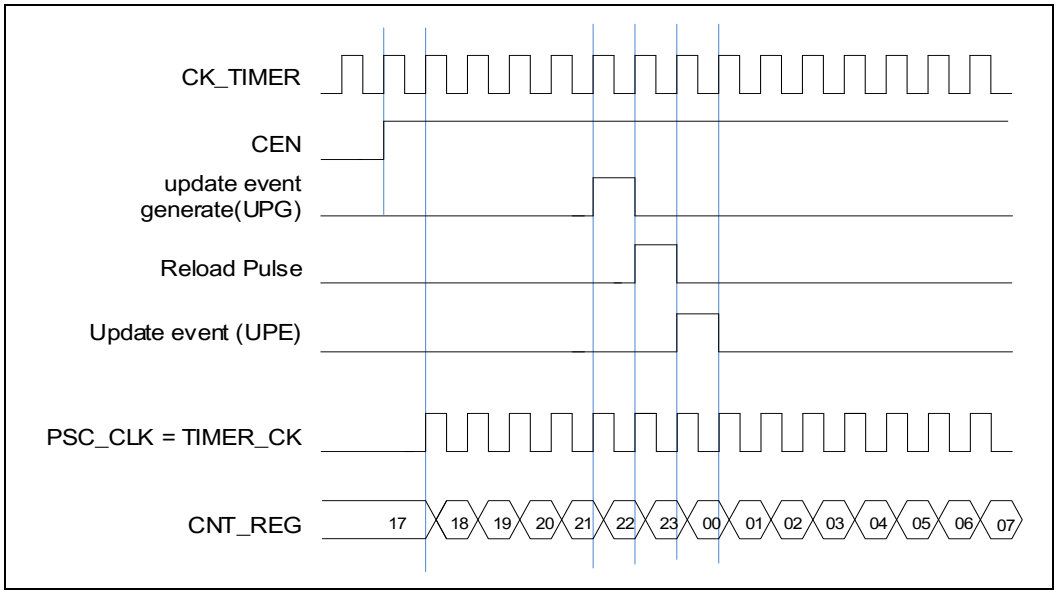
- SMC[2:0]=3'b000，定时器选择内部时钟源(连接到RCU模块的CK_TIMER)

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 9-2. 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_CH0/ TIMERx_CH1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。

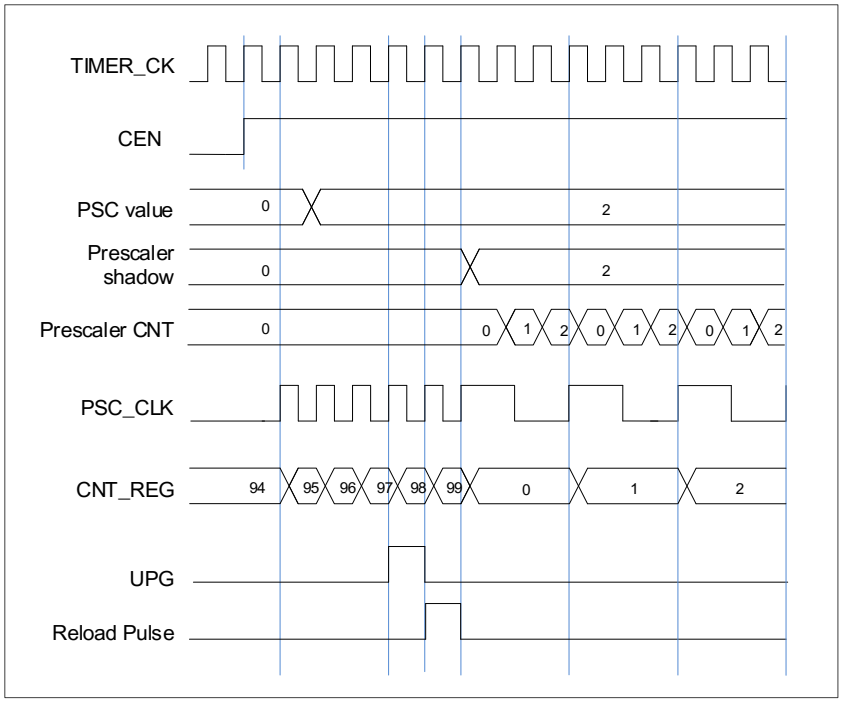
- SMC1=1'b1(外部时钟模式1)，定时器选择外部输入引脚ETI作为时钟源。

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 9-3. 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 9-4. 向上计数时序图, PSC=0/2*和*图 9-5. 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 9-4. 向上计数时序图，PSC=0/2

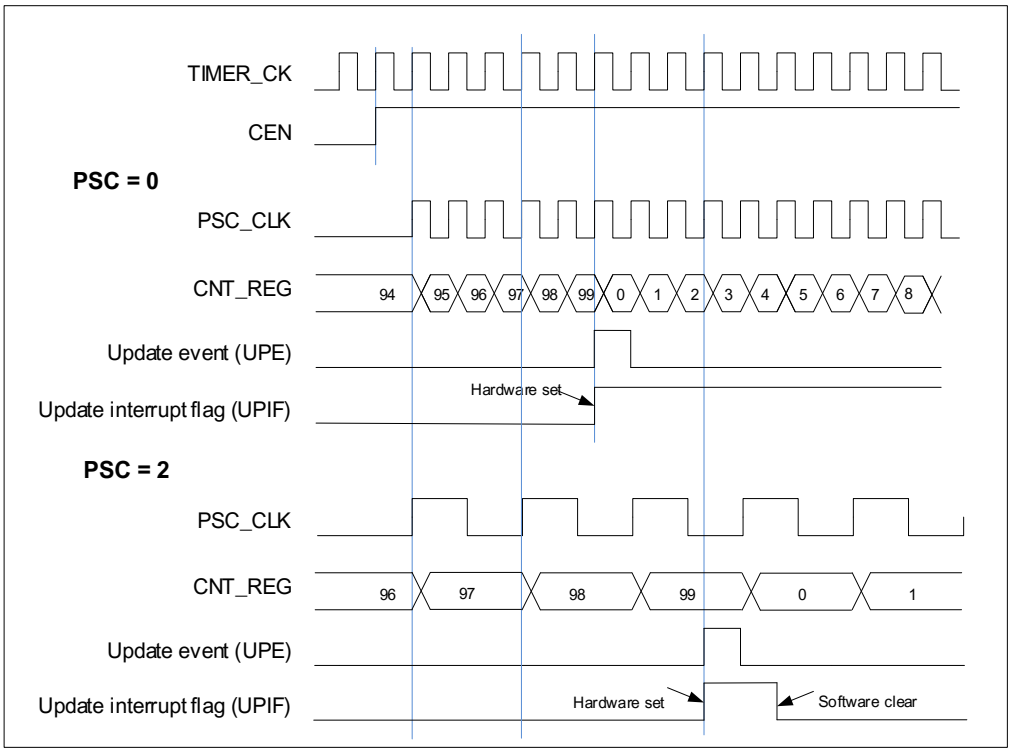
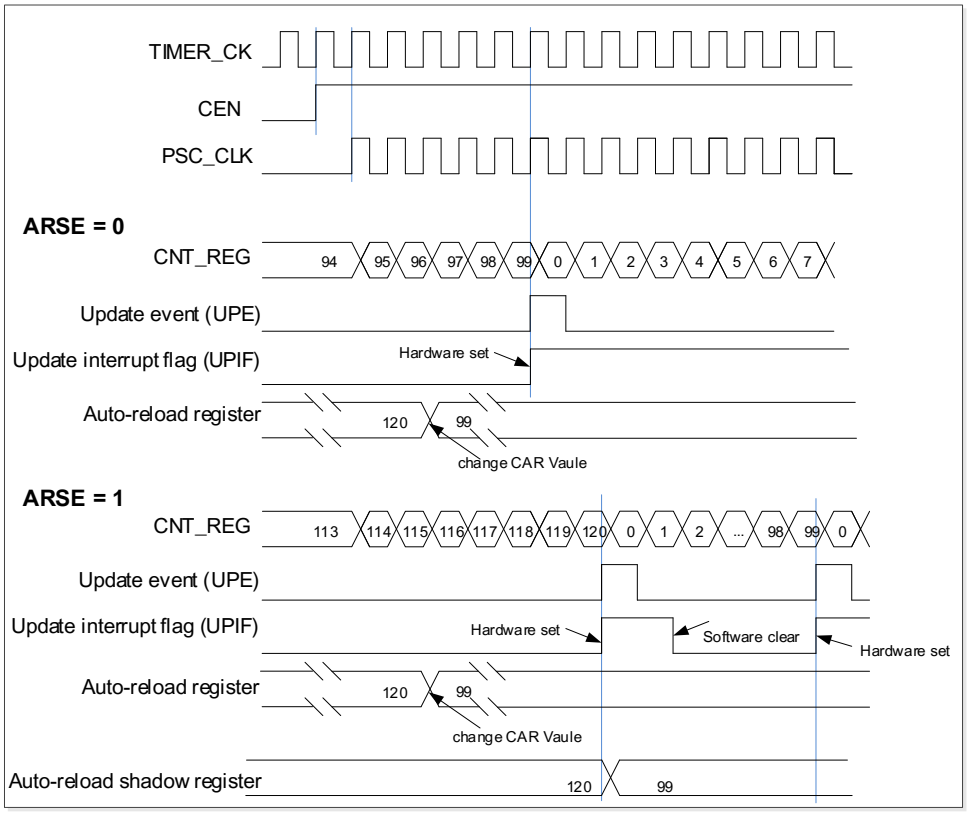


图 9-5. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值



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，自动重载寄存器，预分频寄存器)都将被更新。

*图 9-6. 向下计数时序图，PSC=0/2*和*图 9-7. 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 9-6. 向下计数时序图，PSC=0/2

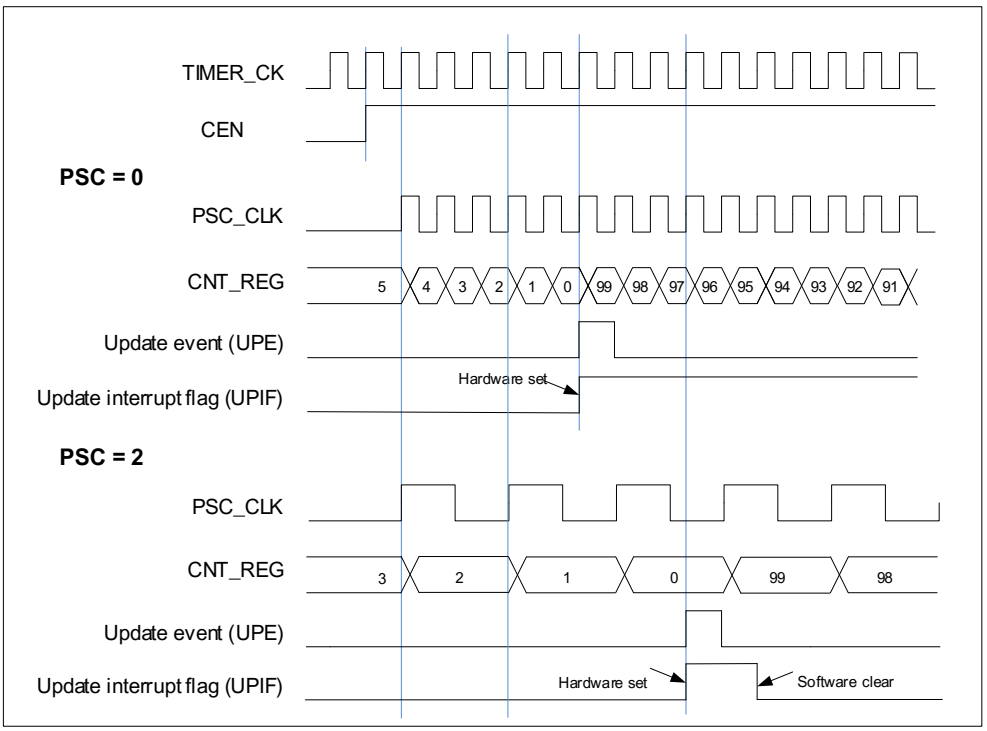
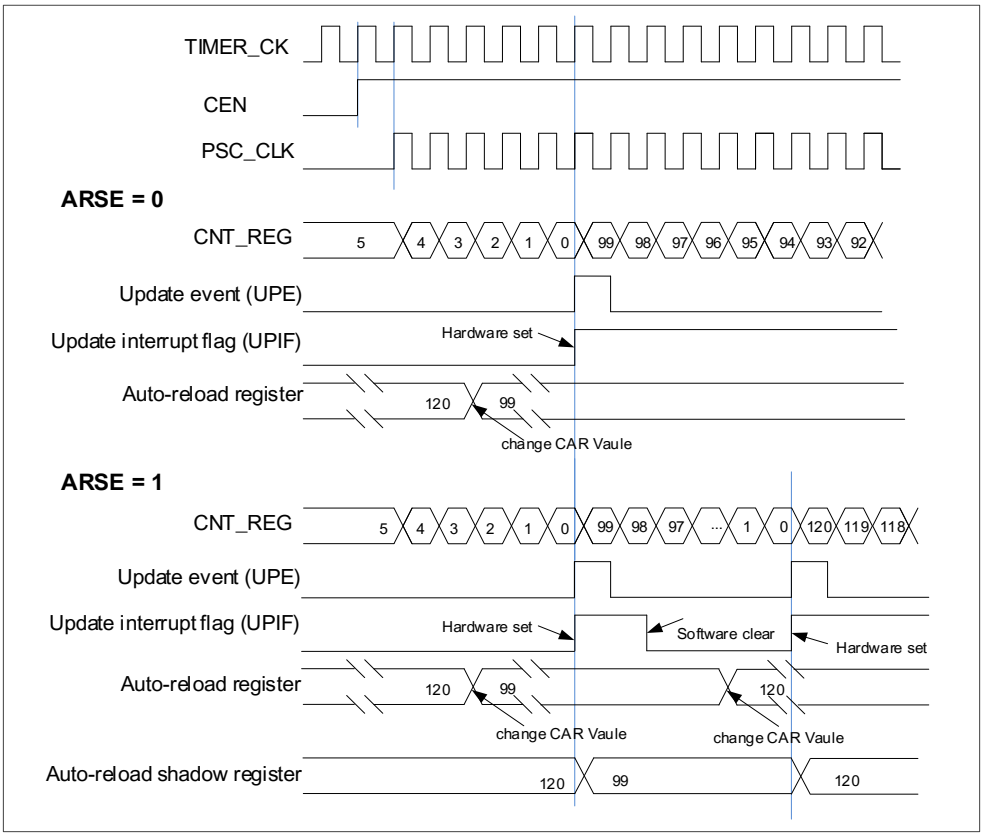


图 9-7. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值



计数器中央对齐计数模式

在中央对齐计数模式下，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。在中央对齐计数模式中，TIMERx_CTL0寄存器中的计数方向控制位DIR只读，指示了当前的计数方向。

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。

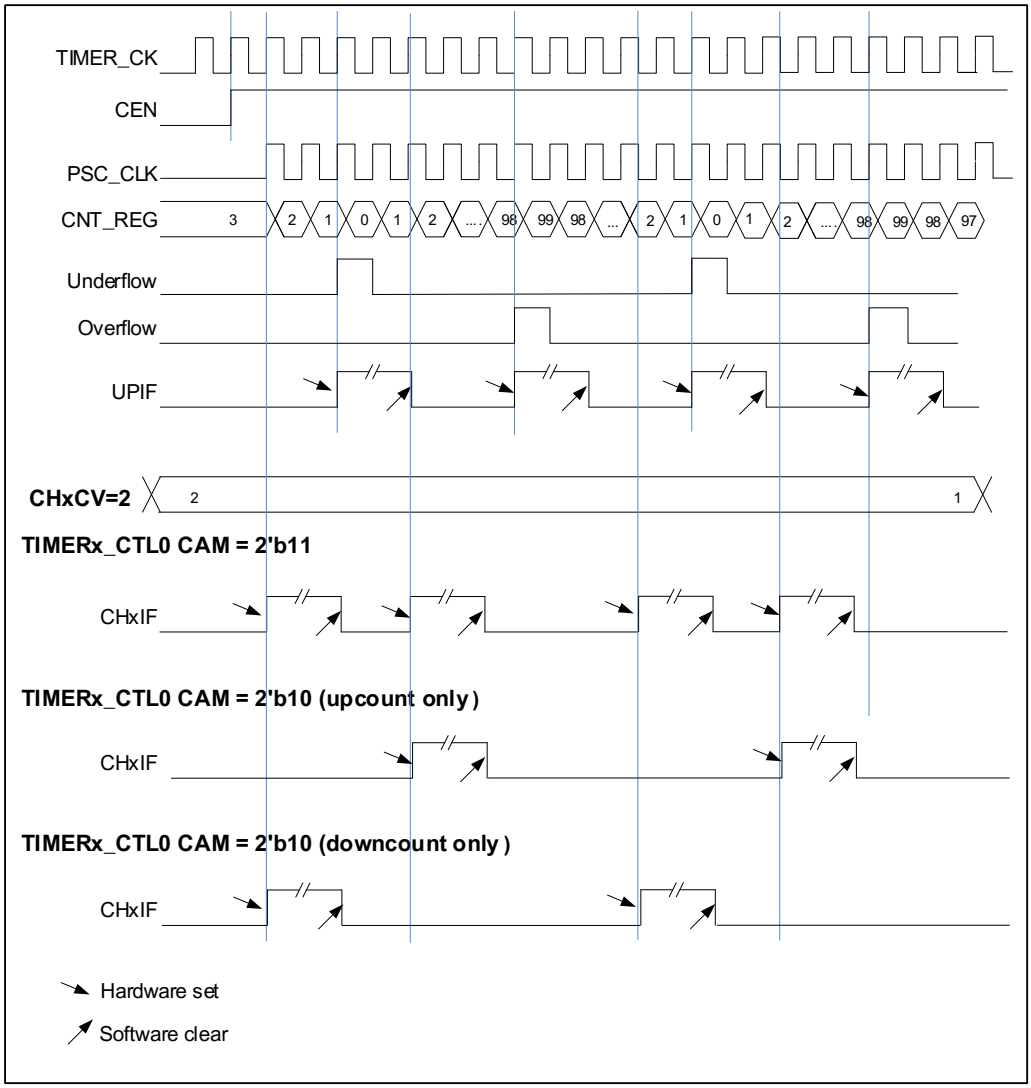
上溢或者下溢时，TIMERx_INTF寄存器中的UPIF位都会被置1。但是CHxIF位是否置1与TIMERx_CTL0寄存器中CAM的值有关。具体细节参考*图9-8. 中央计数模式计数器时序图*。

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，所有的寄存器(重复计数器，自动重载寄存器，预分频寄存器)都将被更新。

*图 9-8. 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 9-8. 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 9-9. 中央计数模式下计数器重复时序图

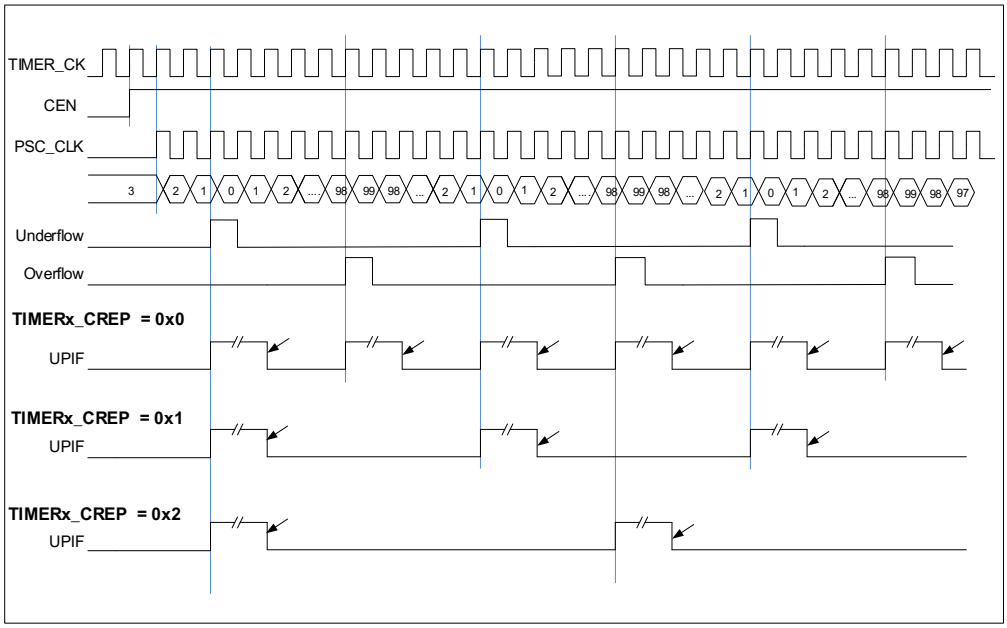


图 9-10. 在向上计数模式下计数器重复时序图

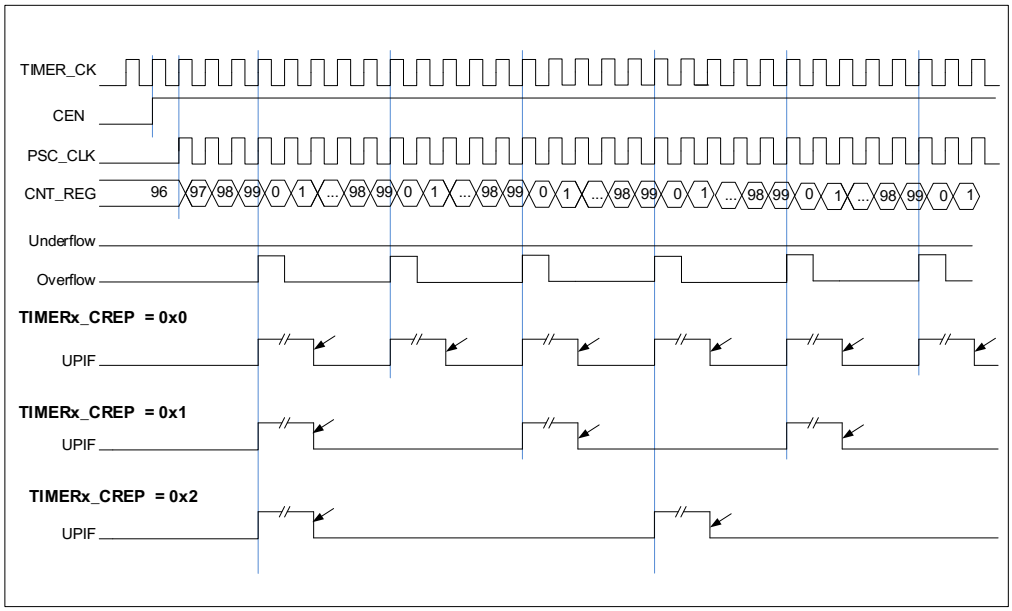
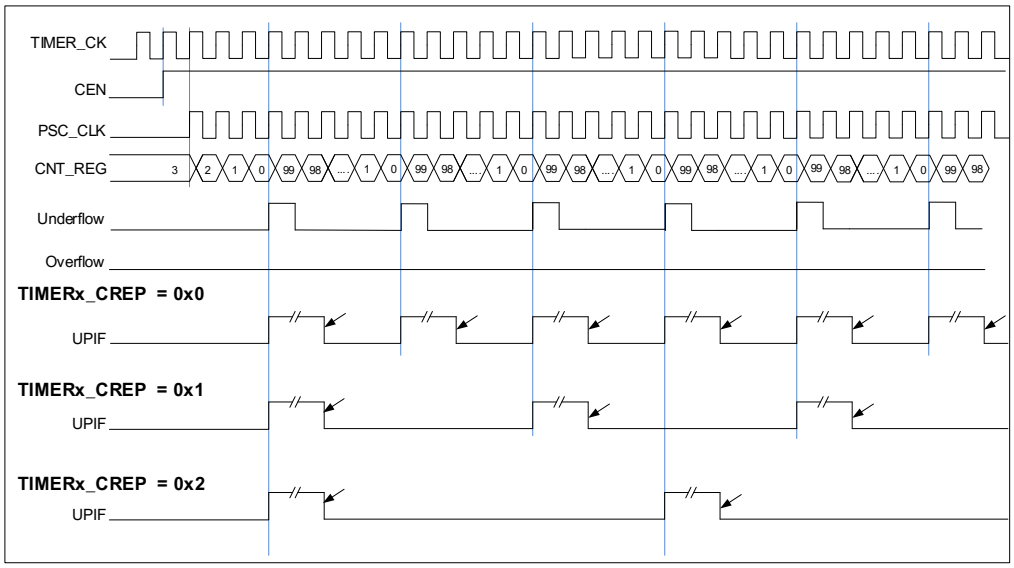


图 9-11. 在向下计数模式下计数器重复时序图



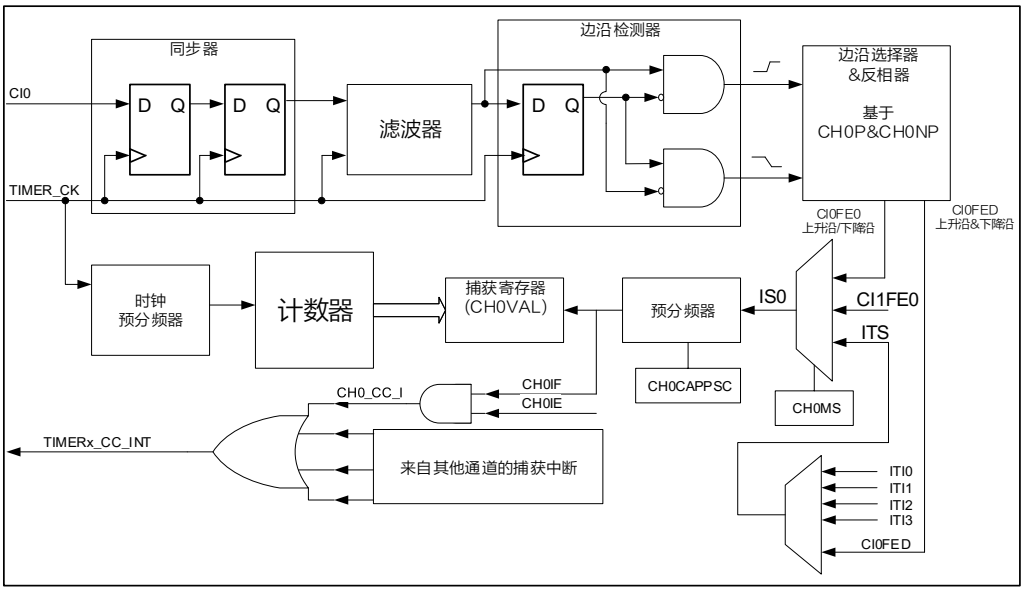
输入捕获和输出比较通道

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。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的时序，频率，周期和占空比等。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，若CHxIE=1则产生通道中断。

图 9-12. 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。通过配置CHxMS，还可以选择其他通道的输入信号或内部触发信号作为捕获信号。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，TIMERx_CHxCV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位置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，来确定是否提出相应的中断和DMA请求。

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，同时设置上升沿捕获。配置TIMERx_CHCTL0寄存器中CH1MS为2'b10，选择通道1捕获信号为CI0，同时设置下降沿捕获。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 9-13. 通道输出比较原理(带有互补输出的通道，x=0,1,2)

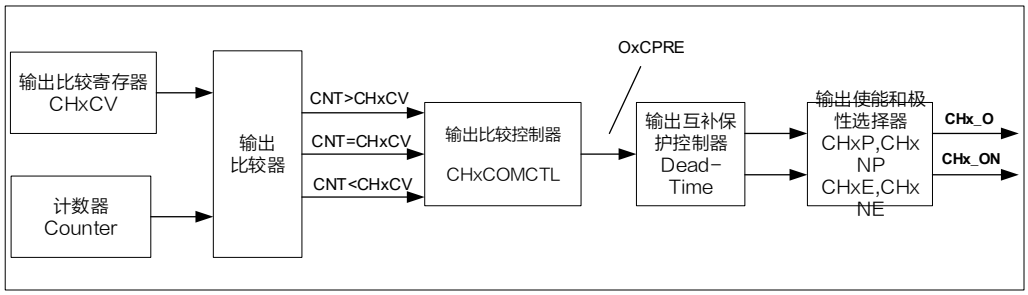


图 9-14. 通道输出比较原理(CHx_O, x=3,4,5)原理

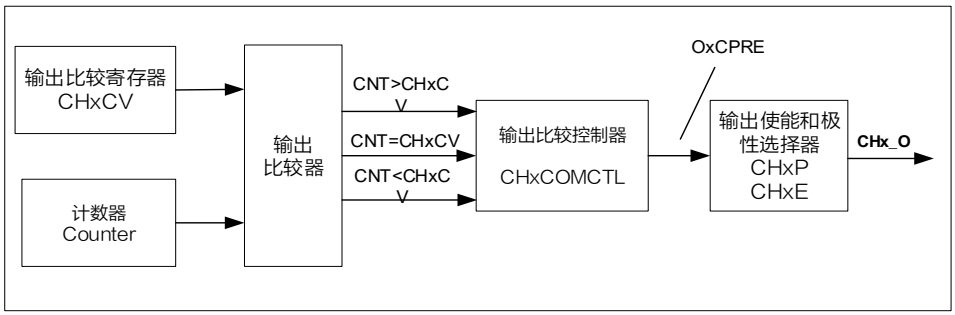


*图9-13. 通道输出比较原理(带有互补输出的通道，x=0,1,2)*和*图9-14. 通道输出比较原理(CHx_O, x=3,4,5)*原理分别给出了输出比较的原理电路。通道输出信号CHx_O/CHx_ON与OxCPRE信号(详情请见*通道输出准备信号*)的关系描述如下：OxCPRE信号高电平有效，CHx_O/CHx_ON的输出情况与OxCPRE信号，CHxP/CHxNP位和CHxE/CHxNE位有关(具体情况请见TIMERx_CHCTL2寄存器中的描述)。例如：

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。详情请见*通道输出互补PWM*。

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TIMERx_CHxCV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，被置低电平或者翻转。当计数器的值与TIMERx_CHxCV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CxCDE=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/翻转)；
- 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/CxCDE位配置中断/DMA请求使能。

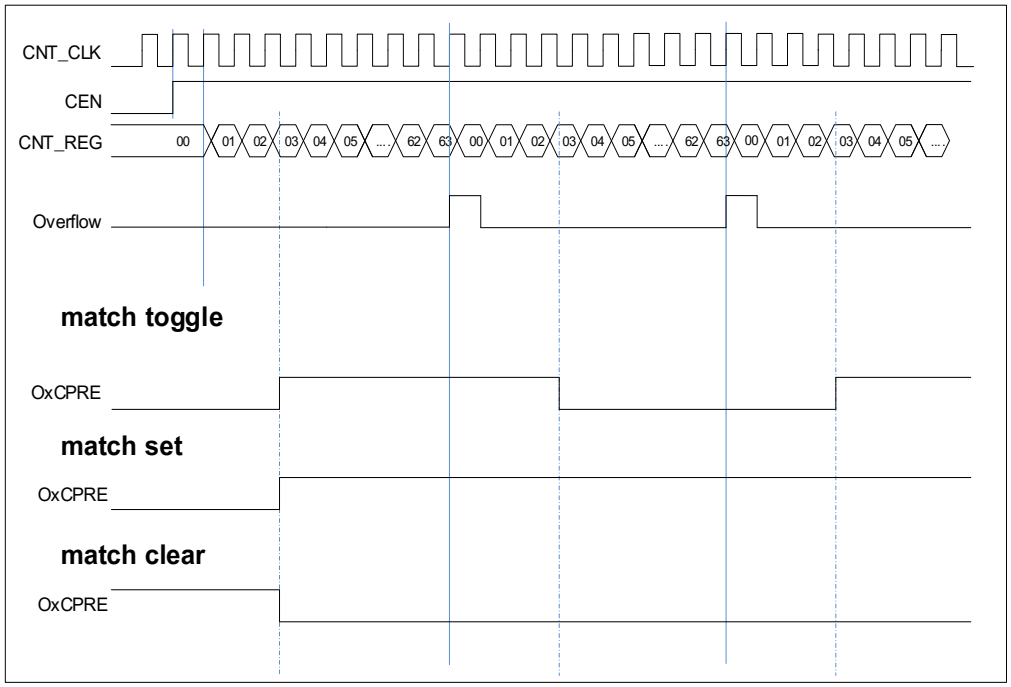
第四步：通过TIMERx_CAR寄存器和TIMERx_CHxCV寄存器配置输出比较时基：

TIMERx_CHxCV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9-15. 三种输出比较模式*显示了三种比较输出模式：翻转/置高电平/置低电平，CAR=0x63，CHxVAL=0x3。

图 9-15. 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，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9-16. EAPWM时序图*显示了CAPWM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9-17. CAPWM时序图*显示了CAPWM的输出波形和中断。

在向上计数模式中， PWM模式0下(CHxCOMCTL=3'b110)，如果TIMERx_CHxCV寄存器的值大于TIMERx_CAR寄存器的值， 通道输出一直为无效电平； PWM模式1下(CHxCOMCTL=3'b111)，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。

图 9-16. EAPWM 时序图

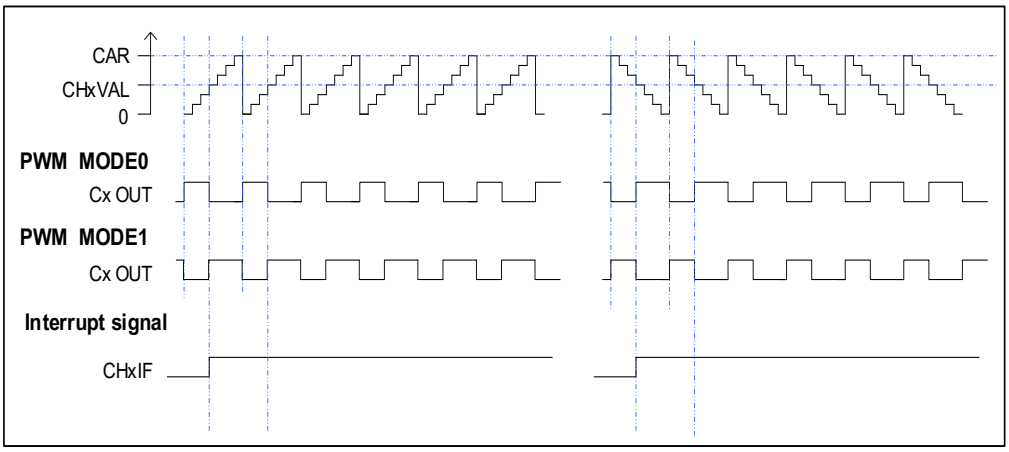
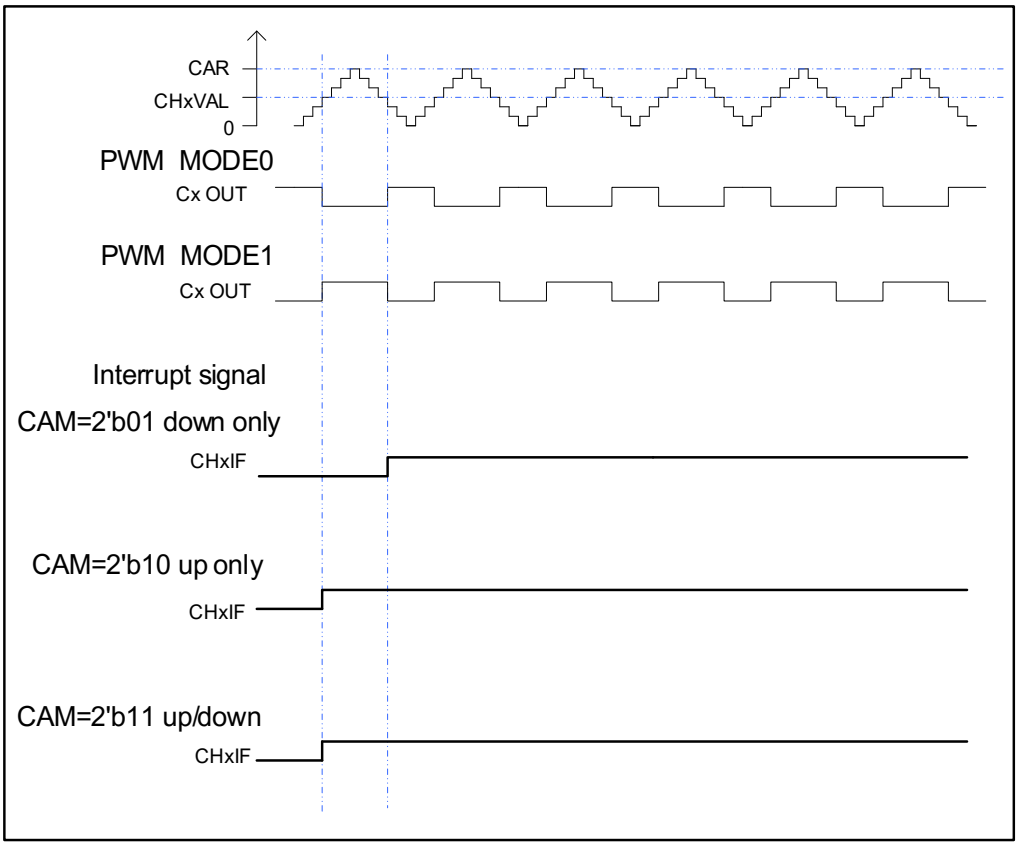


图 9-17. CAPWM 时序图



通道输出准备信号

根据*图9-13. 通道输出比较原理(带有互补输出的通道，x=0,1,2)*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。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。

设置CHxCOMCEN=1，当由外部ETI引脚信号产生的ETIFP信号为高电平时，OxCPRE被强制为低电平。直到下一次转换到有效电平之前，OxCPRE信号一直保持低电平。

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有六路通道，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寄存器中的POEN、ROS和IOS位，TIMERx_CTL1寄存器中的ISOx和ISOxN位。输出极性由TIMERx_CHCTL2寄存器中的CHxP和CHxNP位来决定。

表 9-2. 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="6">0</td><td rowspan="6">0/1</td><td rowspan="3">0</td><td rowspan="2">0</td><td>0</td><td colspan="2">CHx_O = ISOx CHx_ON = ISOxN
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 rowspan="3">1</td><td rowspan="2">0</td><td>0</td><td colspan="2">CHx_O = ISOx CHx_ON = ISOxN
CHx_O / CHx_ON 输出禁能(1)</td></tr>
<tr><td>1</td><td colspan="2" rowspan="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>1</td><td>0</td></tr>
<tr><td rowspan="2">1</td><td rowspan="2">0</td><td rowspan="2">0/1</td><td rowspan="2">0</td><td>1</td><td>1</td></tr>
<tr><td colspan="2">CHx_O/CHx_ON = LOW
CHx_O/CHx_ON输出禁能</td></tr>
<tr><td colspan="4"></td><td>CHx_O = LOW
CHx_O输出禁能</td><td>CHx_ON=OxCPRE⊕(4)CHxN
P
CHx_ON输出使能</td></tr>
</table>

互补参数					输出状态	
POEN	ROS	IOS	CHxEN	CHxNEN	CHx_O	CHx_ON
			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:

（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

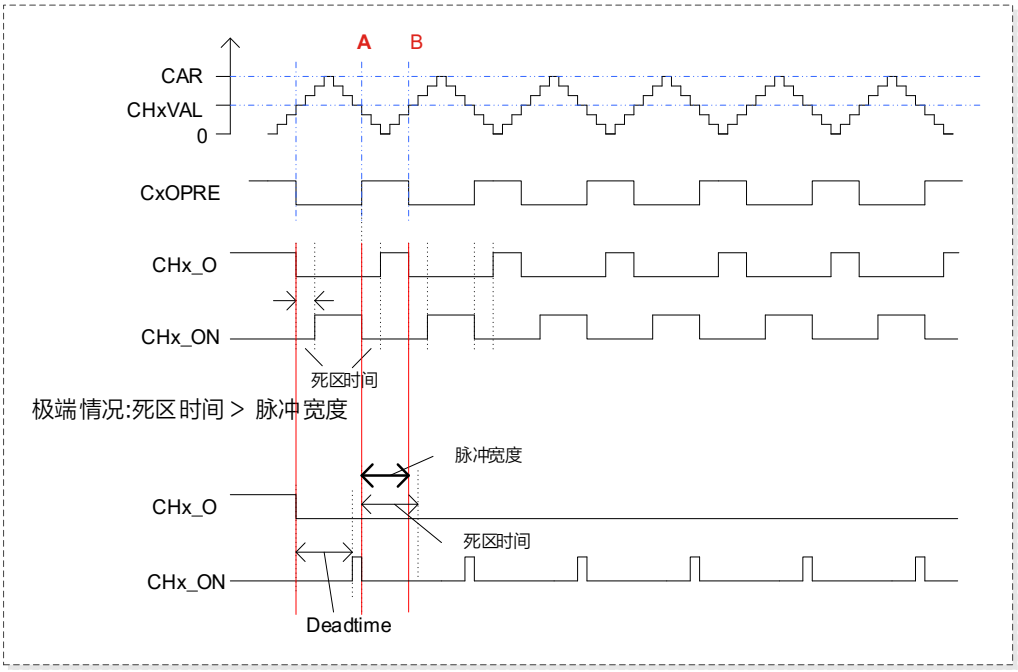
设置CHxEN和CHxNEN为1'b1的同时，设置POEN为1，死区插入就会被使能。DTCFG位域定义了死区时间，死区时间对除了通道3以外的通道有效。死区时间的细节，请参考TIMERx_CCHP寄存器。

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 PWM 模式 0，当通道 x 匹配事件发生时(TIMERx 计数器=CHxVAL)，OxCPRE 翻转。在*图 9-18. 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，通道 x 匹配事件再次发生(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。

有时会有一些极端事件发生，例如：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 9-18. 带死区时间的互补输出



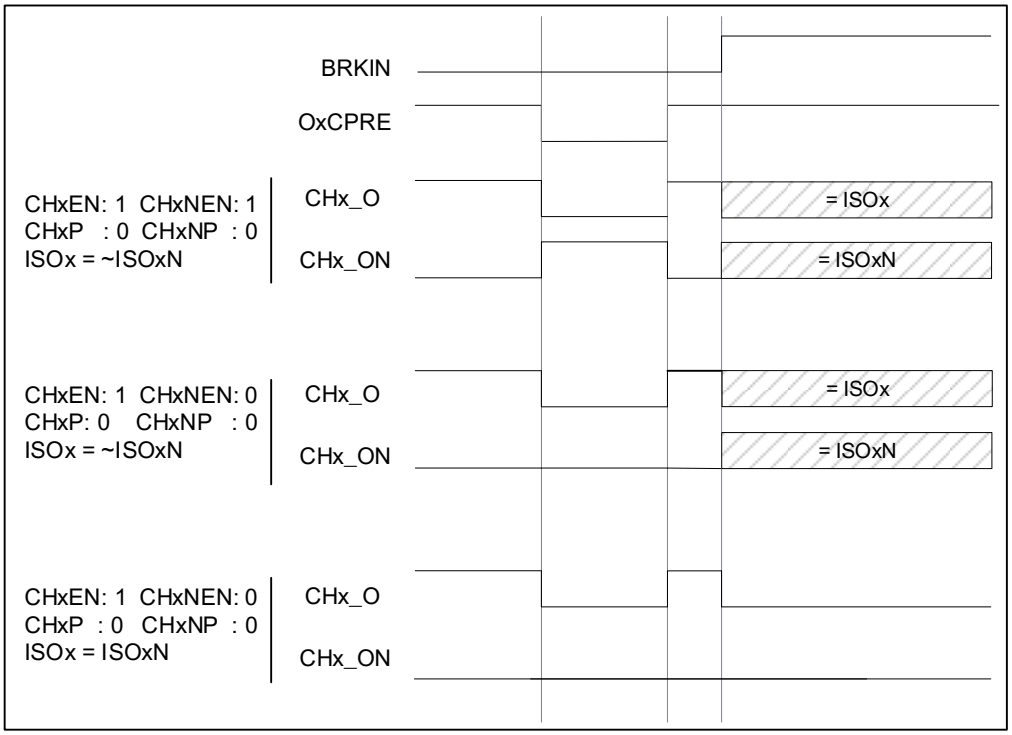
中止模式

使用中止模式时，输出CHx_O和CHx_ON的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。任何情况下，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，CMP输出(仅TIMER0支持)，也可以选择HXTAL时钟失效事件，时钟失效事件由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BRKIN的中止输入极性。需要注意的是，来自CMP的中止输入是高电平有效的，在TIMER中无极性选择。

发生中止时，POEN位被异步清除，一旦POEN位为0，CHx_O和CHx_ON的输出电平由TIMERx_CTL1寄存器中的ISOx位和ISOxN位决定。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 9-19. 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9-3. 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 9-3. 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 9-20.在正交译码器模式 2 且 CI0FE0 极性不反相时计数器行为

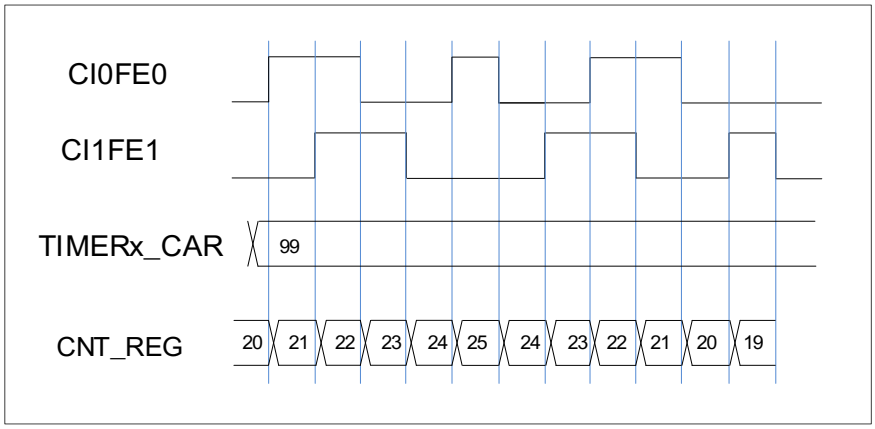
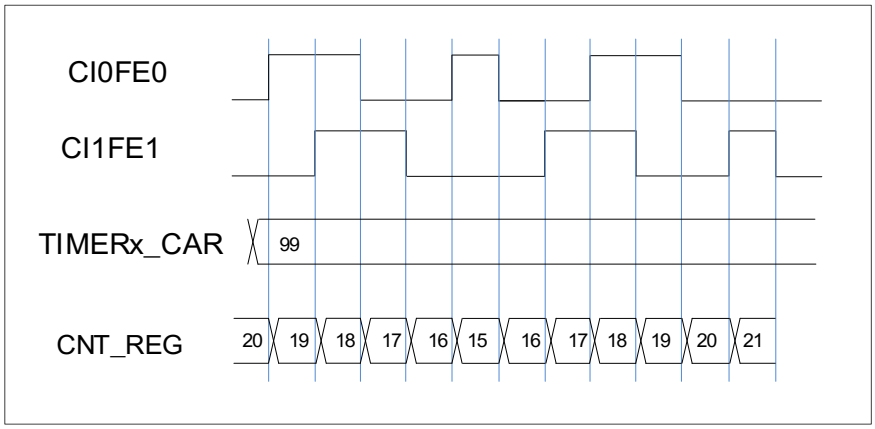


图 9-21.在正交译码器模式 2 且 CI0FE0 极性反相时计数器行为



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 9-22. 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。众所周知，我们要两个定时器。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功能(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。

高级定时器和通用L0定时器具有输入异或功能，可作为TIMER_in定时器。同时，高级定时器具备互补输出和死区插入功能，可作为TIMER_out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 (TIMER2) -> TIMER_out (TIMER0 ITI2)

选择好合适的互连定时器，线路也已经连接好，就可以配置定时器。有以下关键配置：

- 通过设置TI0S，来使能异或功能。三路输入信号的任何一路发生变化，CI0都会翻转，CH0VAL此时会捕获计数器的当前值。
- 通过设置CCUC和CCSE，来选择ITIx触发换相。
- 根据需求配置PWM参数。

图 9-22. 霍尔传感器用在 BLDC 电机控制中

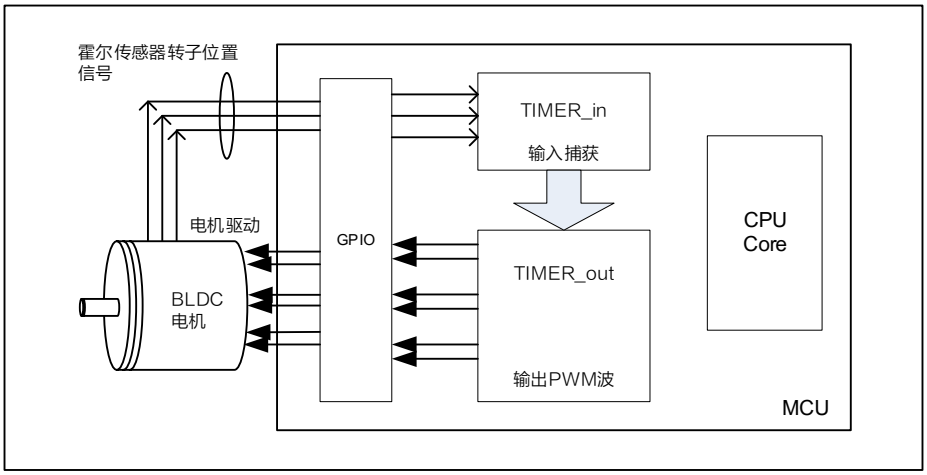
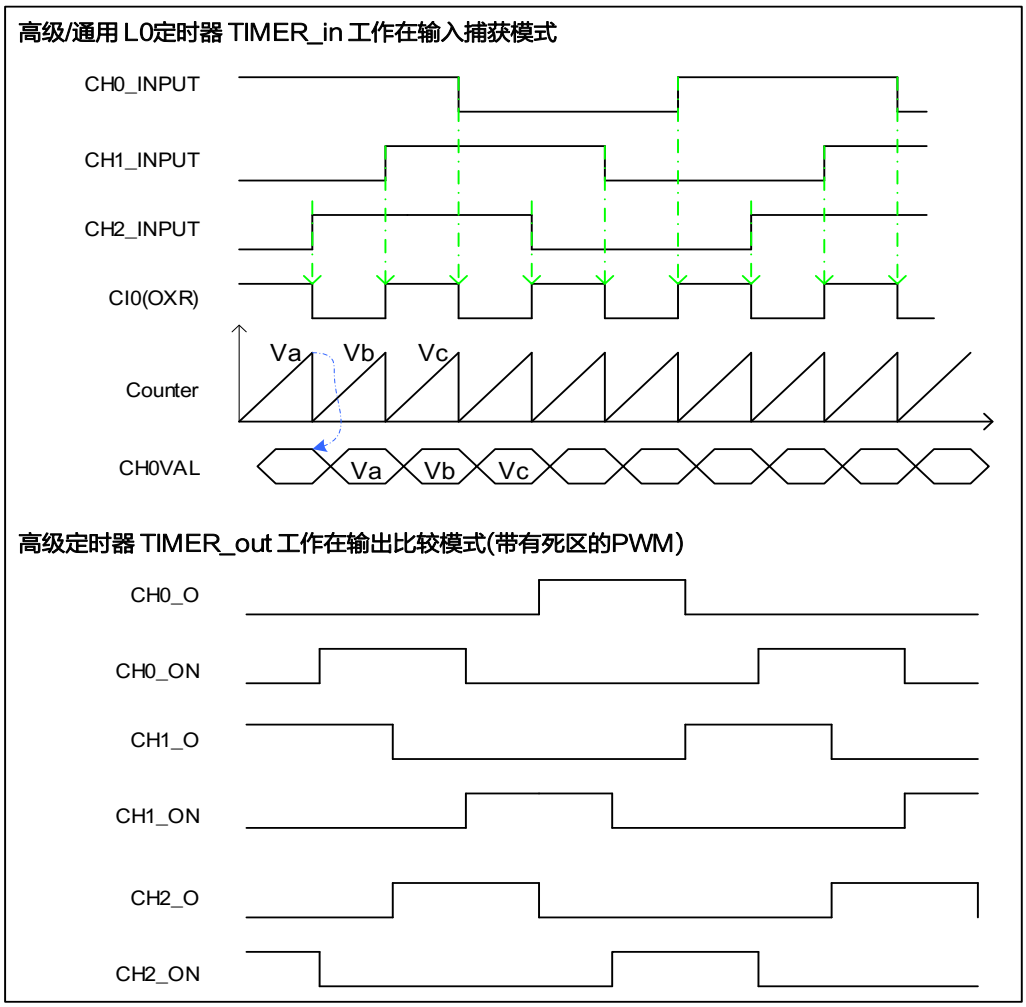


图 9-23. 两个定时器之间的霍尔传感器时序图

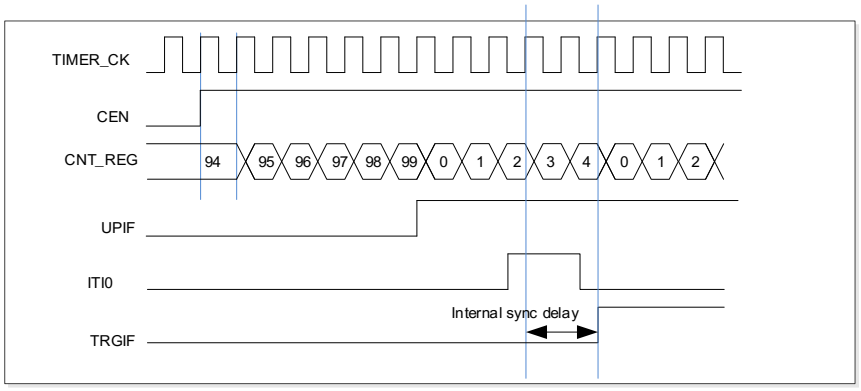
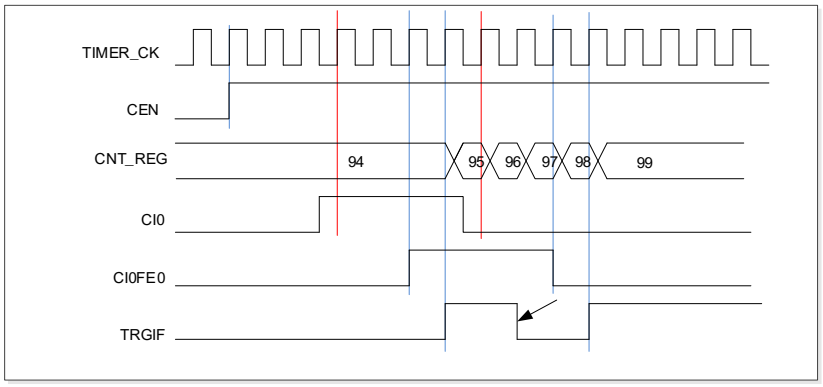


主-从管理

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。

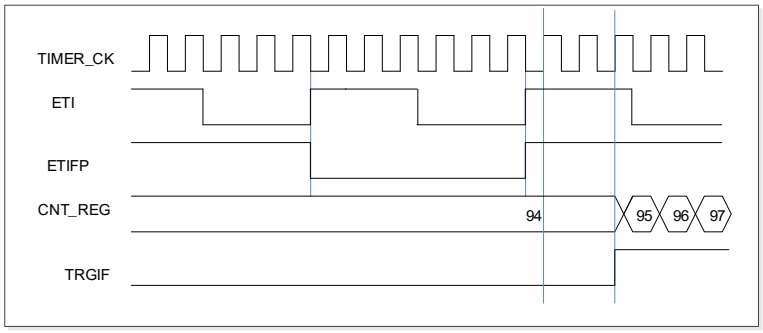
表 9-4. 从模式例子列表

	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 9-24. 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 9-25. 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 9-26. 事件模式下的控制电路



单脉冲模式

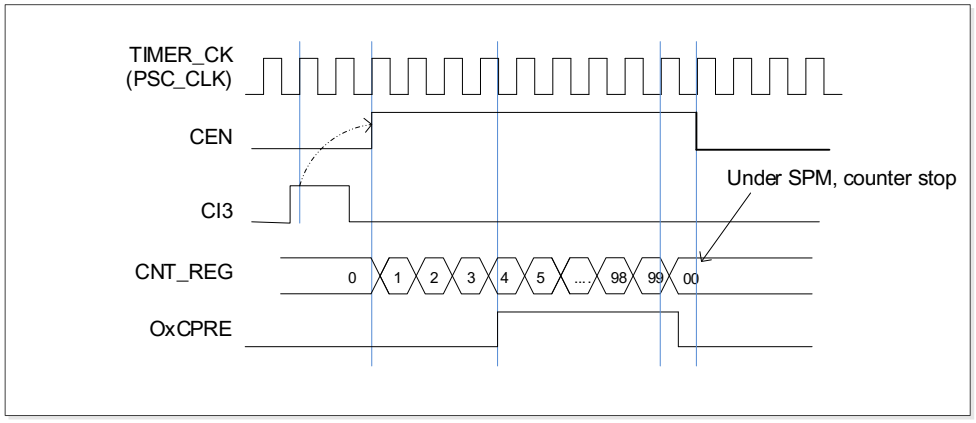
单脉冲模式与重复模式是相反的，设置 TIMERx_CTL0 寄存器的 SPM 位置 1，则使能单脉冲模式。当 SPM 置 1，计数器在下次更新事件到来后清零并停止计数。为了得到脉冲波，可以通过设置 CHxCOMCTL 配置 TIMERx 为 PWM 模式或者比较模式。

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 9-27. 单脉冲模式，TIMERx_CHxCV = 4 TIMERx_CAR=99 展示了一个例子。

图 9-27. 单脉冲模式，TIMERx_CHxCV = 4 TIMERx_CAR=99

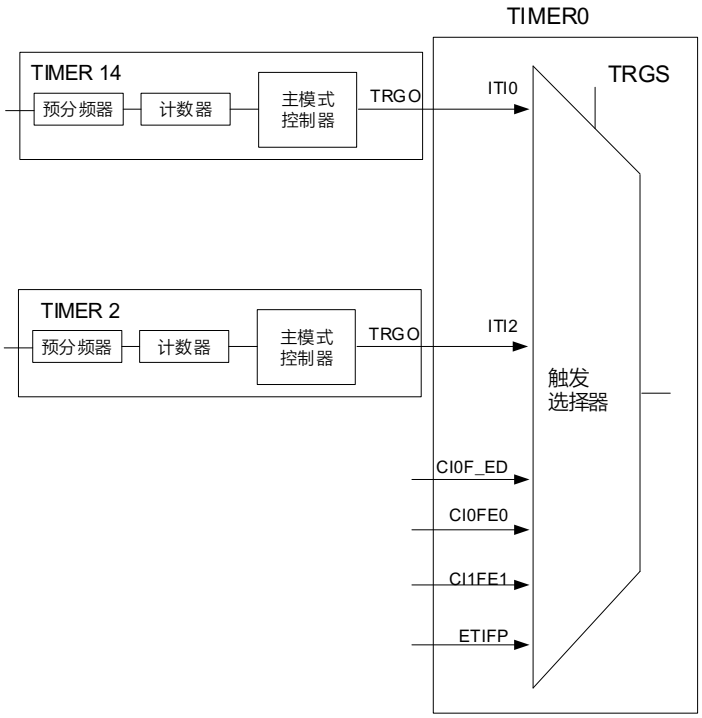


定时器互连

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 9-28. 定时器 0 主/从模式的例子 显示了当定时器 0 配置为从模式时的触发选择。

图 9-28. 定时器 0 主/从模式的例子



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 9-28. 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)。

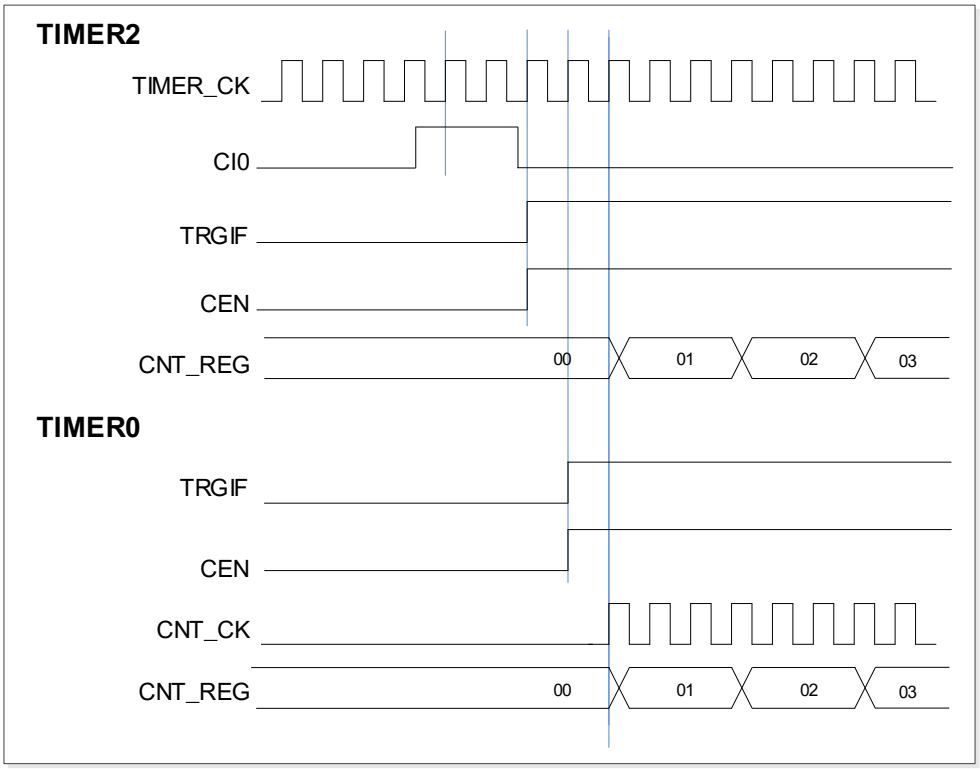
- 使用一个外部触发来同步两个定时器。

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 9-29. 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。必须使能相应的DMA请求位，一些内部中断事件才可以产生DMA请求。当中断事件发生，TIMERx会给DMA发送请求。DMA配置成M2P(传输方向为从内存到外设)模式，PADDR(外设基地址)为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®-M23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。

9.1.5. 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															ISO5
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISO4	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		rw

位/位域	名称	描述

31:17	保留	必须保持复位值
16	ISO5	通道 5 的空闲状态输出 参考 ISO0 位
15	ISO4	通道 4 的空闲状态输出 参考 ISO0 位
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 位复位时，CH0_ON 设置低电平 1：当 POEN 位复位时，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 位复位时，CH0_O 设置低电平 1：当 POEN 位复位时，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，TIMERx_CH1 和 TIMERx_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	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_{TIMER_CK}
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清除源连接到OCPRE_CLR输入 1: OCPRE 清除源连连接到 ETIFP
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 CI1FE1 的电平，计数器在 CI0FE0 的边沿向上/下计数。 010：正交译码器模式 1。根据 CI0FE0 的电平，计数器在 CI1FE1 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													CH5IE	CH4IE	CH5DEN
													rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH4DEN	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	rw

位/位域	名称	描述
31:19	保留	必须保持复位值
18	CH5IE	通道 5 比较/捕获中断使能

0：禁止通道 5 中断
1：使能通道 5 中断

17	CH4IE	通道 4 比较/捕获中断使能 0：禁止通道 4 中断 1：使能通道 4 中断
16	CH5DEN	通道 5 比较/捕获 DMA 请求使能 0：禁止通道 5 比较/捕获 DMA 请求 1：使能通道 5 比较/捕获 DMA 请求
15	CH4DEN	通道 4 比较/捕获 DMA 请求使能 0：禁止通道 4 比较/捕获 DMA 请求 1：使能通道 4 比较/捕获 DMA 请求
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															CH5OF
															rc_w0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH4OF	CH5IF	CH4IF	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	rc_w0	rc_w0	rc_w0

位/位域	名称	描述
31:17	保留	必须保持复位值
16	CH5OF	通道 5 捕获溢出标志 参见 CH0OF 描述
15	CH4OF	通道 4 捕获溢出标志 参见 CH0OF 描述
14	CH5IF	通道 5 比较/捕获中断标志

		参见 CH0IF 描述
13	CH4IF	通道 4 比较/捕获中断标志 参见 CH0IF 描述
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 0。
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						CH5G	CH4G	BRKG	TRGG	CMTG	CH3G	CH2G	CH1G	CH0G	UPG
						w	w	w	w	w	w	w	w	w	w

位/位域	名称	描述
31:10	保留	必须保持复位值
9	CH5G	通道 5 捕获或比较事件发生 参见 CH0G 描述
8	CH4G	通道 4 捕获或比较事件发生 参见 CH0G 描述
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，根据 CCSE 位(TIMERx_CTL1 寄存器中)的值，通道捕获/比较控制寄存器（CHxEN，CHxNEN 和 CHxCOMCTL）的互补输出被更新。
0：不产生通道换相更新事件
1：产生通道换相更新事件

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，计数器的当前值被捕获到 TIMERx_CH0CV 寄存器，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 OCPRE 清除源信号输入高电平时，O0CPRE 参考信号被清 0。OCPRE 清除源由 TIMERx_SMCFG 寄存器的 OCRC 位选择。 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 0：通道 0 输出比较快速禁能。仅比较结果输入作为有效边沿时会产生比较匹配，并将 CH0_O 设置为比较电平，激活 CH0_O 输出的最小延时为 5 个时钟周期。 1：通道 0 输出比较快速使能。触发输入信号的有效边沿和比较结果都会产生比较匹配，并将 CH0_O 设置为比较电平。当触发信号的输入作为有效边沿时，激活 CH0_O 输出的最小延时为 3 个时钟周期。
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
CH5COMCEN	CH5COMCTL[2:0]			CH5COMSEN	CH5COMFEN	CH5MS[1:0]		CH4COMCEN	CH4COMCTL[2:0]			CH4COMSEN	CH4COMFEN	CH4MS[1:0]	
CH5CAPFLT[3:0]				CH5CAPPSC[1:0]				CH4CAPFLT[3:0]				CH4CAPPSC[1:0]			
rw				rw		rw		rw				rw		rw	
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	CH5COMCEN	通道 5 输出比较清 0 使能 参见 CH0COMCEN 描述
30:28	CH5COMCTL[2:0]	通道 5 输出比较模式 参见 CH0COMCTL 描述
27	CH5COMSEN	通道 5 输出比较影子寄存器使能 参见 CH0COMSEN 描述
26	CH5COMFEN	通道 5 输出比较快速使能 参见 CH0COMFEN 描述。
25:24	CH5MS[1:0]	通道 5 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 CH5EN 位被清 0)时这些位才可以写。 00：通道 5 配置为输出 01：通道 5 配置为输入，IS5 映射在 CI5FE5 上 10：通道 5 配置为输入，IS5 映射在 CI4FE5 上 11：通道 5 配置为输入，IS5 映射在 ITS 上 注意：当 CH5MS[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。
23	CH4COMCEN	通道 4 输出比较清 0 使能 参见 CH0COMCEN 描述
22:20	CH4COMCTL[2:0]	通道 4 输出比较模式 参见 CH0COMCTL 描述
19	CH4COMSEN	通道 4 输出比较影子寄存器使能 参见 CH0COMSEN 描述
18	CH4COMFEN	通道 4 输出比较快速使能 参见 CH0COMFEN 描述。
17:16	CH4MS[1:0]	通道 4 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 CH4EN 位被清 0)时这些位才可以写。 00：通道 4 配置为输出 01：通道 4 配置为输入，IS4 映射在 CI4FE4 上 10：通道 4 配置为输入，IS4 映射在 CI5FE4 上 11：通道 4 配置为输入，IS4 映射在 ITS 上 注意：当 CH4MS[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。

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 OCPRE 清除源信号输入高电平时，O0CPRE 参考信号被清 0。OCPRE 清除源由 TIMERx_SMCFG 寄存器的 OCRC 位选择。 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 0：通道 2 输出比较快速禁能。仅比较结果输入作为有效边沿时会产生比较匹配，并将 CH2_O 设置为比较电平，激活 CH2_O 输出的最小延时为 5 个时钟周期。 1：通道 2 输出比较快速使能。触发输入信号的有效边沿和比较结果都会产生比较匹配，并将 CH2_O 设置为比较电平。当触发信号的输入作为有效边沿时，激活 CH2_O 输出的最小延时为 3 个时钟周期。
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
31:28	CH5CAPFLT[3:0]	通道 5 输入捕获滤波控制 参见 CH0CAPFLT 描述
27:26	CH5CAPPSC[1:0]	通道 5 输入捕获预分频器 参见 CH0CAPPSC 描述
25:24	CH5MS[1:0]	通道 5 模式选择 与输出模式相同
23:20	CH4CAPFLT[3:0]	通道 4 输入捕获滤波控制 参见 CH0CAPFLT 描述
19:18	CH4CAPPSC[1:0]	通道 4 输入捕获预分频器 参见 CH0CAPPSC 描述

17:16 CH4MS[1:0] 通道 4 模式选择
与输出模式相同

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														CH5P	CH5EN
														rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH4P	CH4EN	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	rw	rw

位/位域	名称	描述
31:18	保留	必须保持复位值
17	CH5P	通道 5 极性 参考 CH0P 描述
16	CH5EN	通道 5 使能 参考 CH0EN 描述
15	CH4P	通道 4 极性 参考 CH0P 描述
14	CH4EN	通道 4 使能 参考 CH0EN 描述
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]															

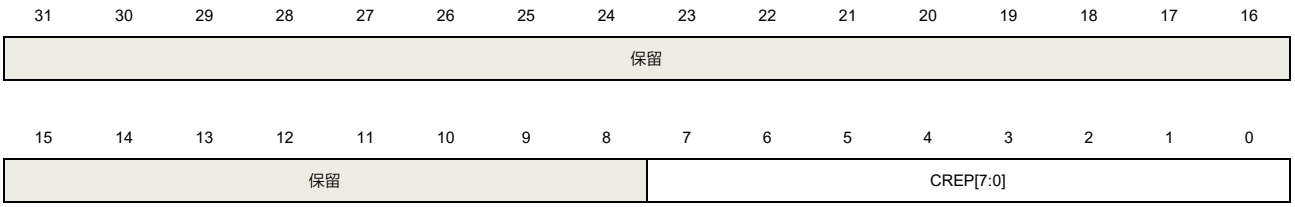
rw

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



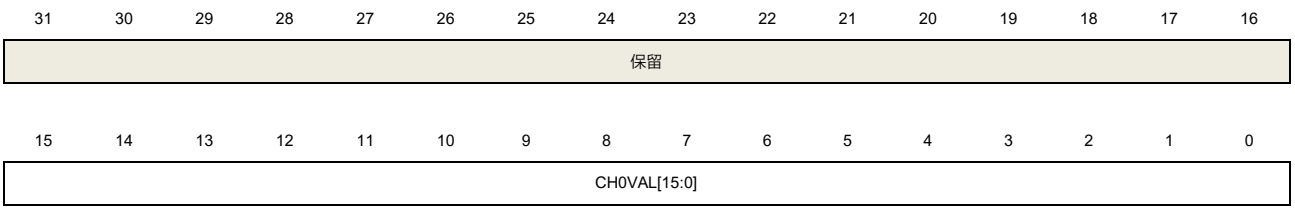
rw

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



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。

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 BKIN 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"。参见***表 9-2. 由参数控制的互补输出表***。 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"。参见***表 9-2. 由参数控制的互补输出表***。 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

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
31:14	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。当第一次访问 TIMERx_DMATB 寄存器时，实际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，从(起始地址)到(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。

通道 4 捕获/比较寄存器 (TIMERx_CH4CV)

地址偏移：0x50
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
CH4VAL[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CH4VAL[15:0]	通道 4 的捕获或比较值 当通道 4 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 4 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 5 捕获/比较寄存器 (TIMERx_CH5CV)

地址偏移：0x54
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
CH5VAL[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CH5VAL[15:0]	通道 5 的捕获或比较值 当通道 5 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 5 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。

配置寄存器 (TIMERx_CFG)

地址偏移：0xFC
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														CHVSEL	OUTSEL
														rw	rw

位/位域	名称	描述
31:2	保留	必须保持复位值
1	CHVSEL	写捕获比较寄存器选择位 此位由软件写 1 或清 0。 0：无影响。

		1：当写入捕获比较寄存器的值与寄存器当前值相等时，写入操作无效。
0	OUTSEL	输出值选择位 此位由软件写 1 或清 0。 0：无影响。 1：如果 POEN 位与 IOS 位均为 0，则输出无效。

9.2. 通用定时器 L0(TIMERx, x=1,2)

9.2.1. 简介

通用定时器L0(TIMER1，2)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(TIMER2)或32位(TIMER1)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，但是它们的计数器可以被同步在一起形成一个更大的定时器。

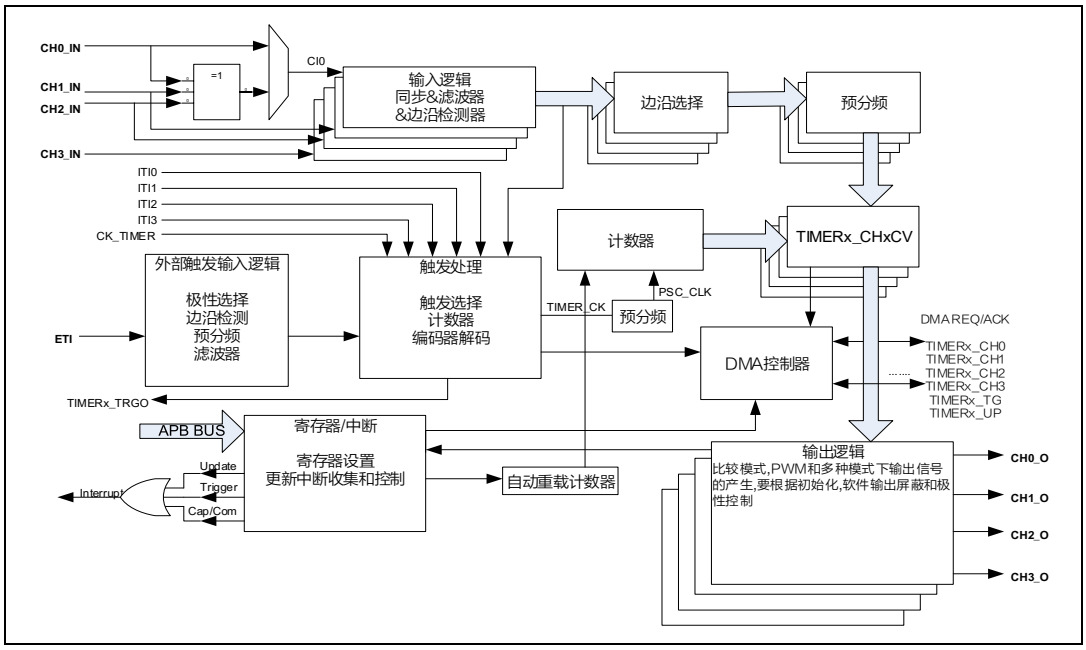
9.2.2. 主要特征

- 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。
- 支持外部信号使能/禁能定时器

9.2.3. 结构框图

*图 9-30. 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 9-30. 通用定时器 L0 结构框图



9.2.4. 功能说明

定时器门控

当TIMERx_CTL1寄存器的GATCTL为0时，定时器仅由CEN位控制，否则也会被外部输入信号INTx控制，INT0连接到TIMER1，INT1连接到TIMER2。如*表9-5. TIMERx(x=1,2)使能配置*所示。

当TIMERx_CTL1寄存器的GATCTL为0时，置位CEN位则使能定时器。

当TIMERx_CTL1寄存器的GATCTL为1时，拉高输入信号INTx同时置位CEN位，则使能定时器。否则，定时器被禁能。

GPIO中的GPIO_INTCTL寄存器的INTxPL位定义了INTx的极性。

GPIO中的GPIO_INTCTL寄存器的INTxSLS位选择INTx的端口引脚。

表 9-5. TIMERx(x=1,2)使能配置

CEN 位	GATCTL 位	INTx 引脚	INTxPL 位	TIMER 使能
0	x	x	x	禁能
1	0	x	x	使能
1	1	0	0	禁能
1	1	0	1	使能
1	1	1	0	使能
1	1	1	1	禁能

注意："x"表示任意值

时钟源配置

通用定时器L0可以是内部时钟源CK_TIMER，或者是由SMC(TIMERx_SMCFG寄存器位[2:0])位确定的时钟源。

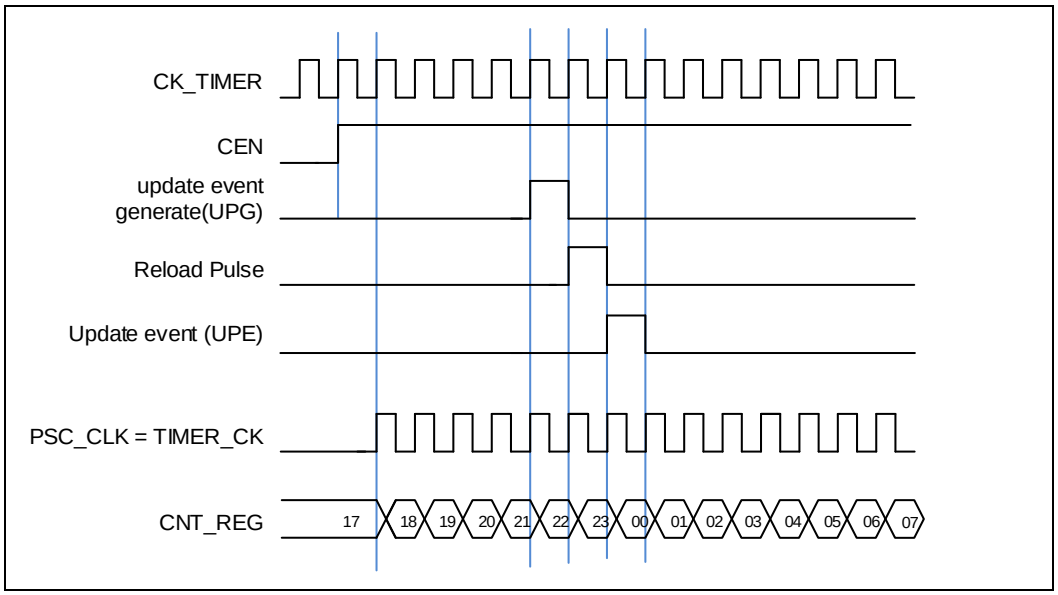
- SMC[2:0]=3'b000，定时器选择内部时钟源(连接到RCU模块的CK_TIMER)

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 9-31. 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_CH0/ TIMERx_CH1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。

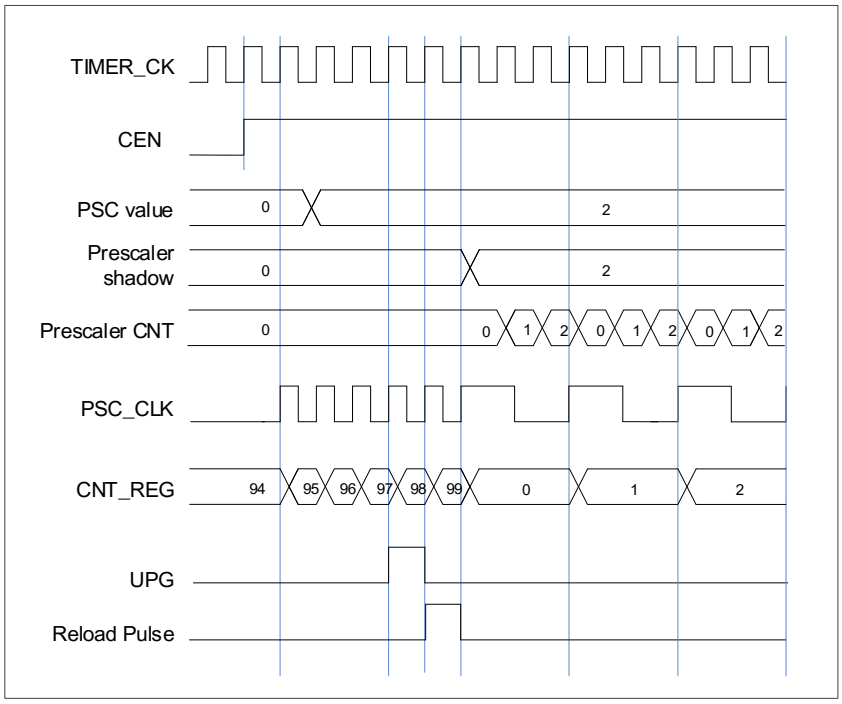
- SMC1=1'b1(外部时钟模式1)，定时器选择外部输入引脚ETI作为时钟源。

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 9-32. 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 9-33. 向上计数时序图,PSC=0/2*和*图 9-34. 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 9-33. 向上计数时序图，PSC=0/2

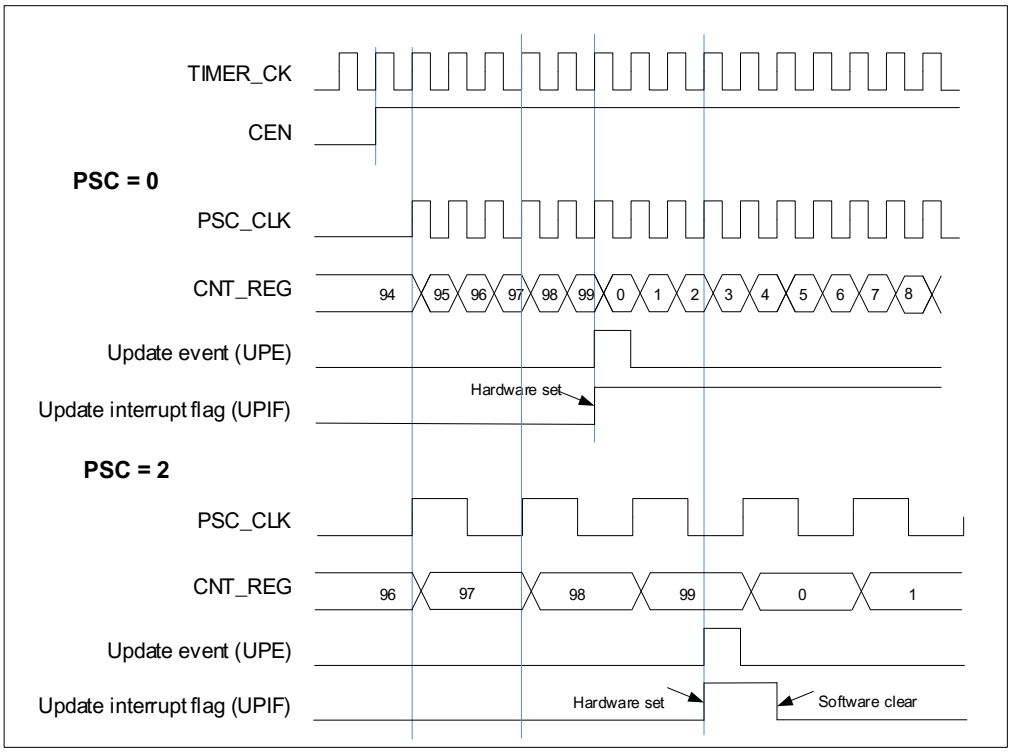
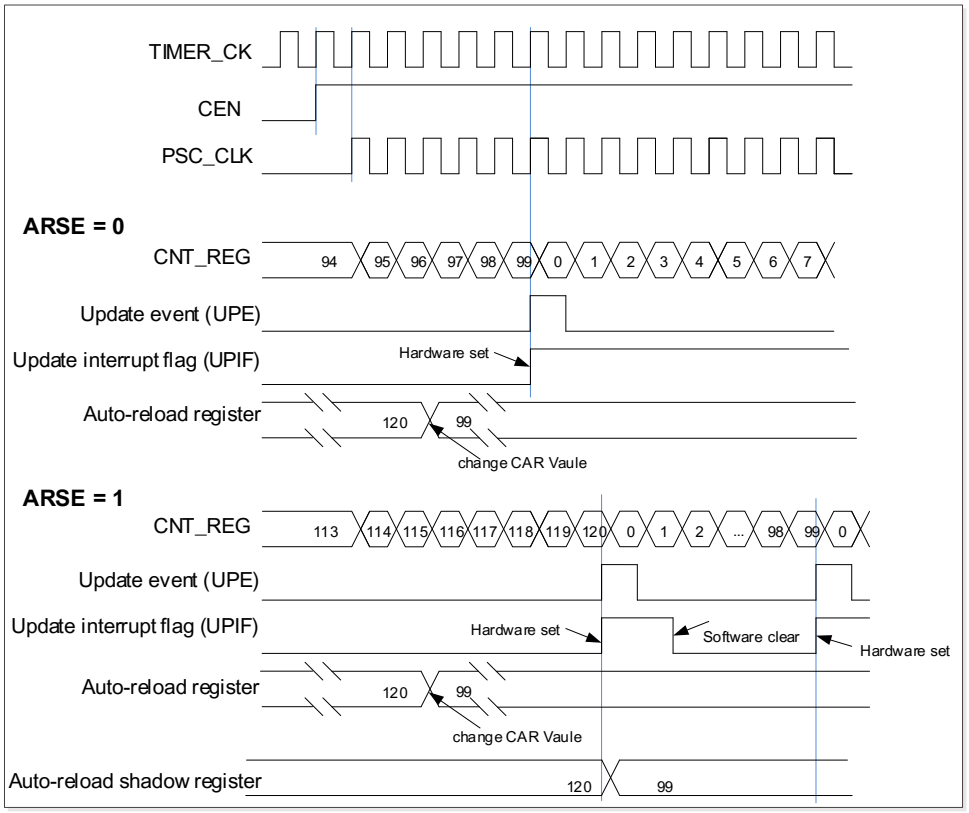


图 9-34. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值



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。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 9-35. 向下计数时序图，PSC=0/2*和*图 9-36. 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 9-35. 向下计数时序图，PSC=0/2

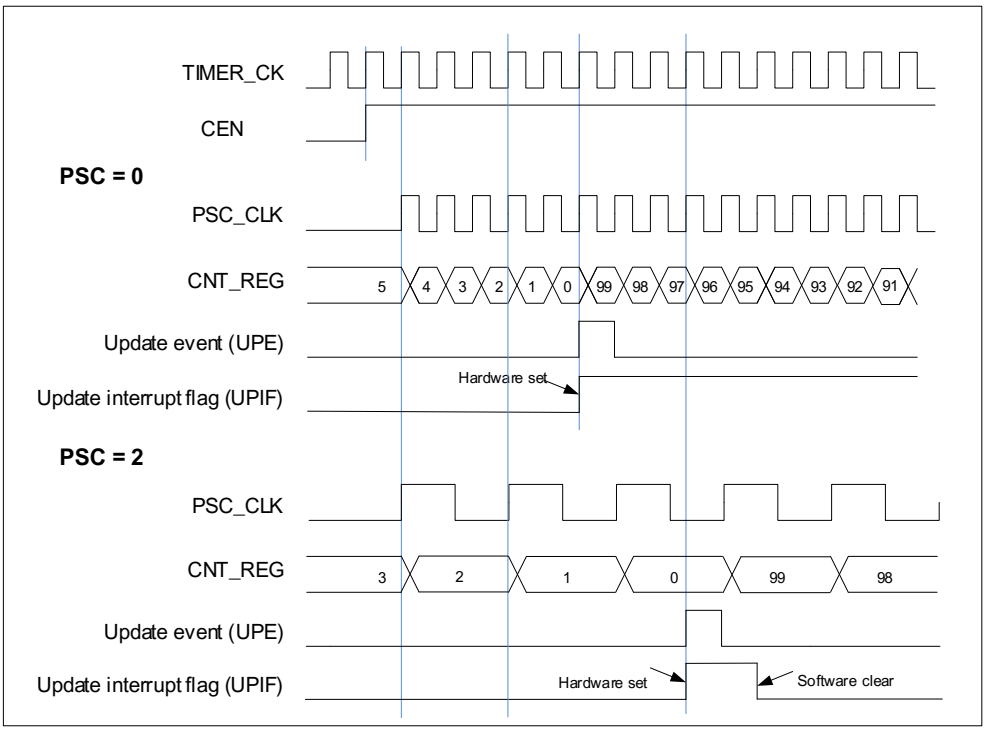
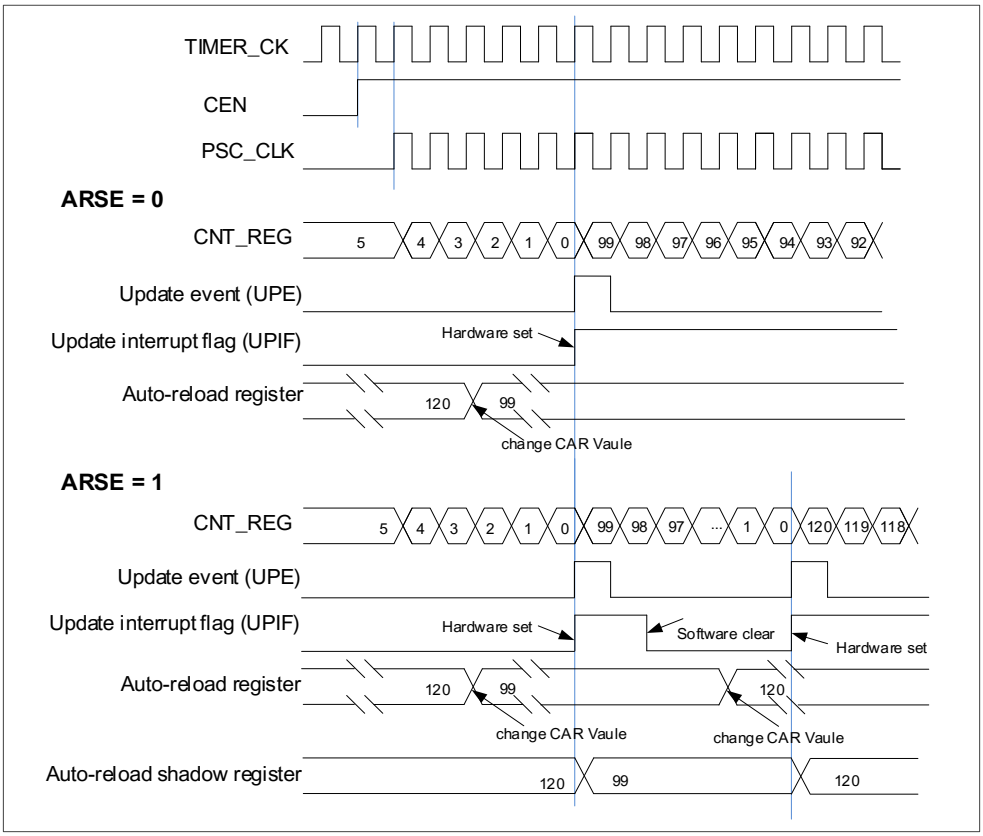


图 9-36. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值



计数器中央对齐计数模式

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。

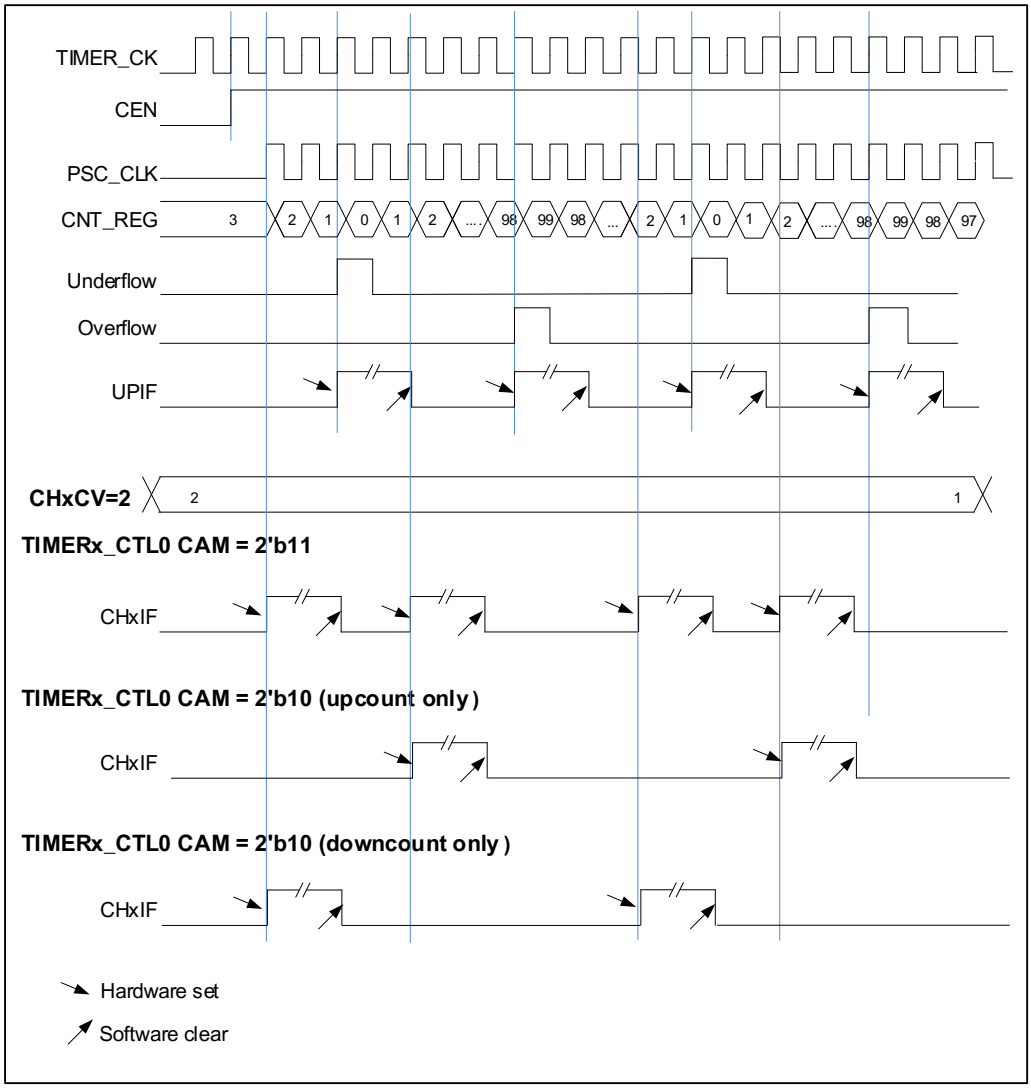
上溢或者下溢时，TIMERx_INTF寄存器中的UPIF位都会被置1。但是CHxIF位是否置1与TIMERx_CTL0寄存器中CAM的值有关。具体细节参考*图9-37. 中央计数模式计数器时序图*。

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 9-37. 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 9-37. 中央计数模式计数器时序图



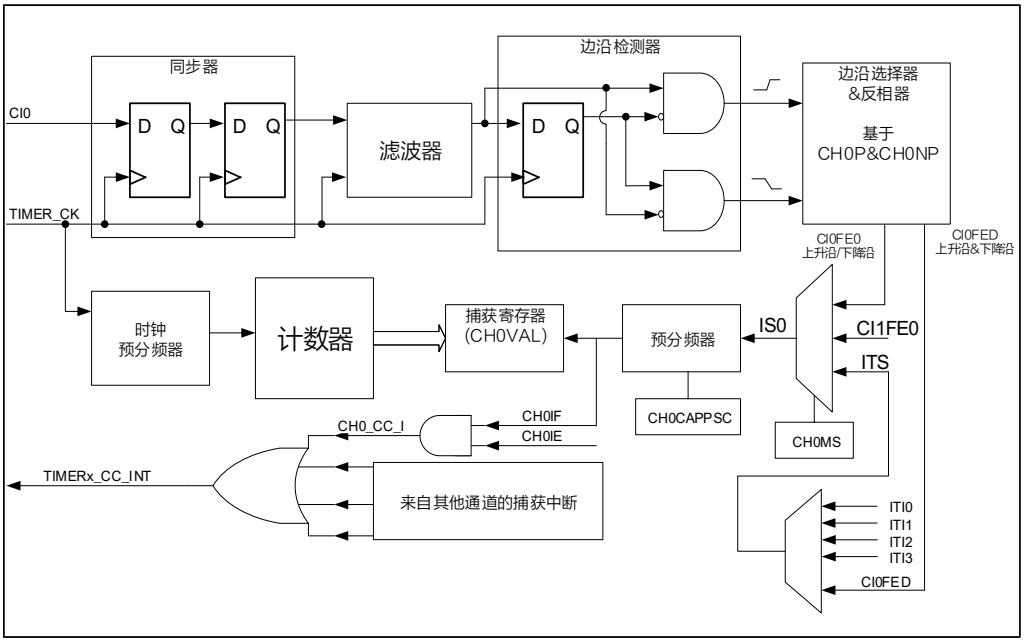
输入捕获和输出比较通道

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。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 9-38. 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。通过配置CHxMS，还可以选择其他通道的输入信号或内部触发信号作为捕获信号。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，来确定是否提出相应的中断和DMA请求。

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 9-39. 通道输出比较原理(x=0,1,2,3)

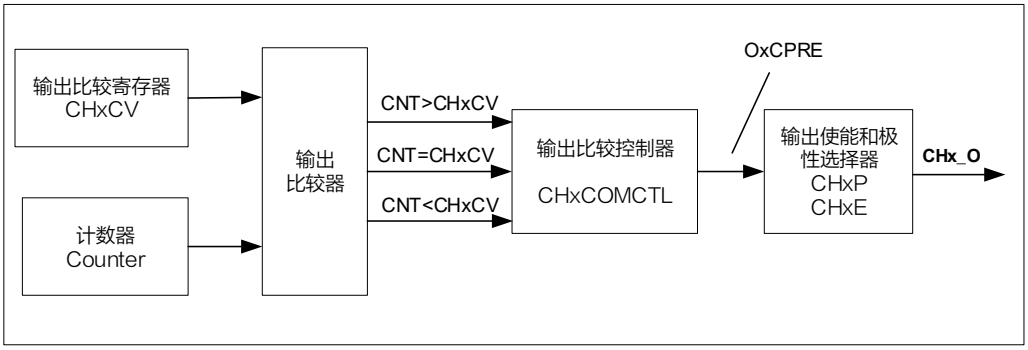


*图9-39. 通道输出比较原理(x=0,1,2,3)*给出了输出比较的原理电路。通道输出信号CHx_O与OxCPRE信号(详情请见*通道输出准备信号*)的关系描述如下：OxCPRE信号高电平有效，CHx_O的输出情况与OxCPRE信号，CHxP位和CHxE位有关(具体情况请见TIMERx_CHCTL2寄存器中的描述)。例如，当设置CHxP=0(CHx_O高电平有效，与OxCPRE输出极性相同)、CHxE=1(CHx_O输出使能)时：

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CHxCV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，被置低电平或者翻转。当计数器的值与CHxCV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CxCDE=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/翻转)；

设置CHxP位来选择有效电平的极性；

设置CHxEN使能输出。

第三步：通过CHxIE/CxCDE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：

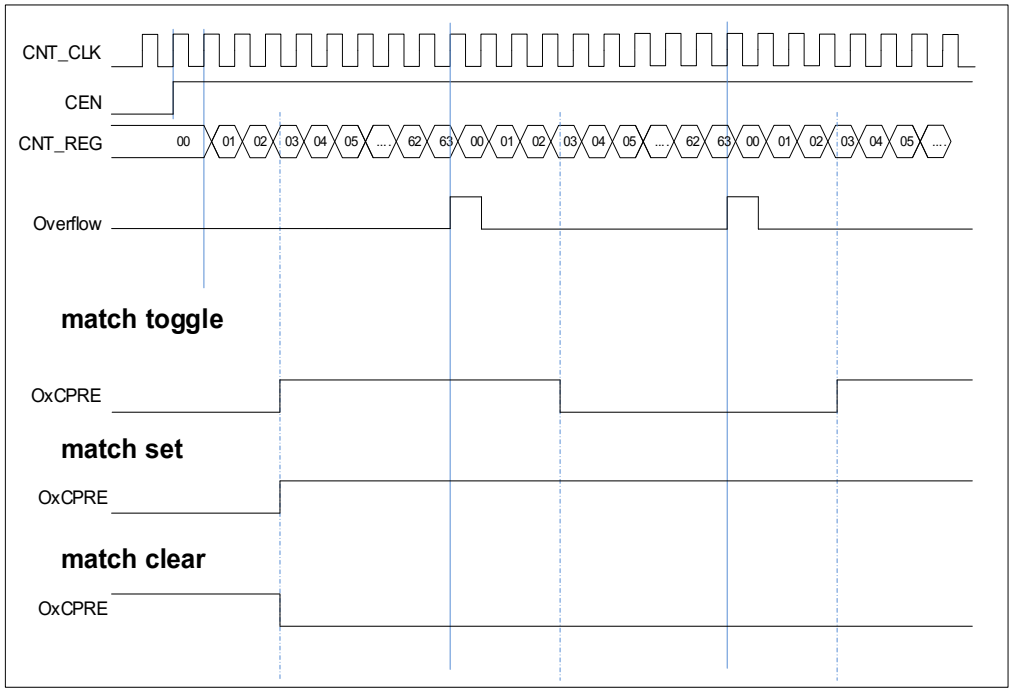
TIMERx_CHxCV可以在运行时根据你所期望的波形而改变。

第五步：设置CEN位使能定时器。

*图 9-40. 三种输出比较模式*显示了三种比较输出模式：翻转/置高电平/置低电平，CAR=0x63，

CHxVAL=0x3。

图 9-40. 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9-41. 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9-42. CAPWM时序图*显示了CAPWM的输出波形和中断。

在向上计数模式中， PWM模式0下(CHxCOMCTL=3'b110)，如果TIMERx_CHxCV寄存器的值大于TIMERx_CAR寄存器的值， 通道输出一直为无效电平； PWM模式1下(CHxCOMCTL=3'b111)，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。

图 9-41. EAPWM 时序图

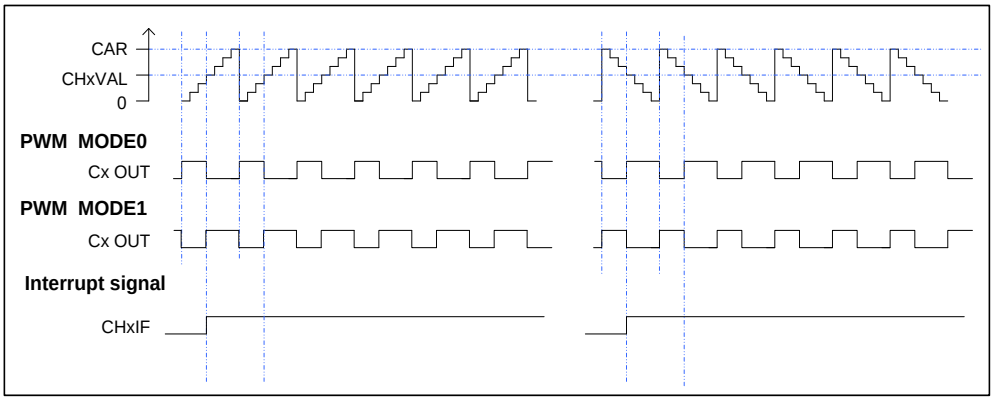
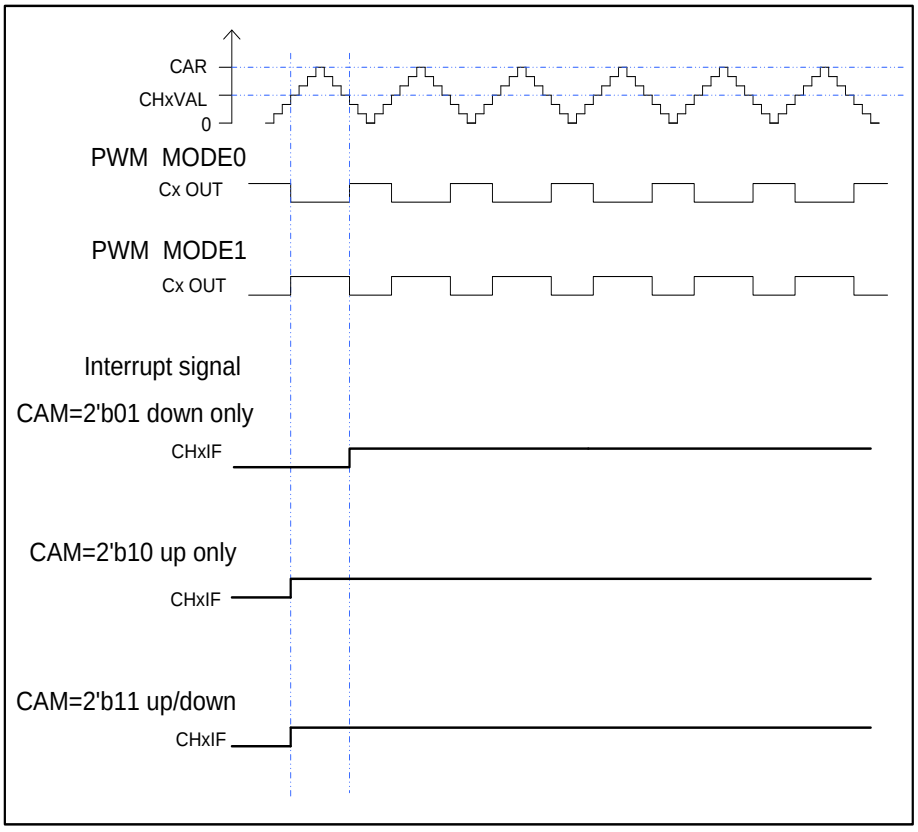


图 9-42. CAPWM 时序图



通道输出准备信号

根据*图9-39. 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，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。

设置CHxCOMCEN=1，当由外部ETI引脚信号产生的ETIFP信号为高电平时，OxCPRE被强制为低电平。直到下一次转换到有效电平之前，OxCPRE信号一直保持低电平。

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*定时器互连*。

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。必须使能相应的DMA请求位，一些内部中断事件才可以产生DMA请求。当中断事件发生，TIMERx会给DMA发送请求。DMA配置成M2P(传输方向为从内存到外设)模式，PADDR(外设基地址)为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®-M23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。

9.2.5. 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
GATCTL	保留							TI0S	MMC[2:0]			DMAS	保留		
rw								rw	rw			rw			

位/位域	名称	描述

31:16	保留	必须保持复位值
15	GATCTL	定时器门控控制 0：当 CEN=1，则使能定时器 1：当 CEN=1 同时 INT 信号有效，则使能定时器。
14:8	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_{TIMER_CK}
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清除源连接到OCPRE_CLR输入 1: OCPRE 清除源连连接到 ETIFP
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 CI1FE1 的电平，计数器在 CI0FE0 的边沿向上/下计数。 010：正交译码器模式 1。根据 CI0FE0 的电平，计数器在 CI1FE1 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 0。 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

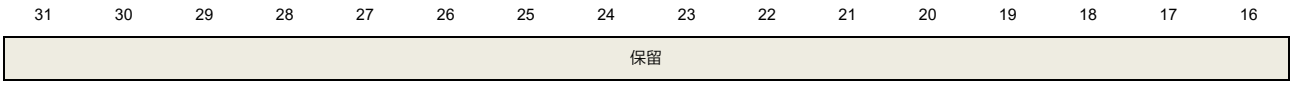
1：产生触发事件

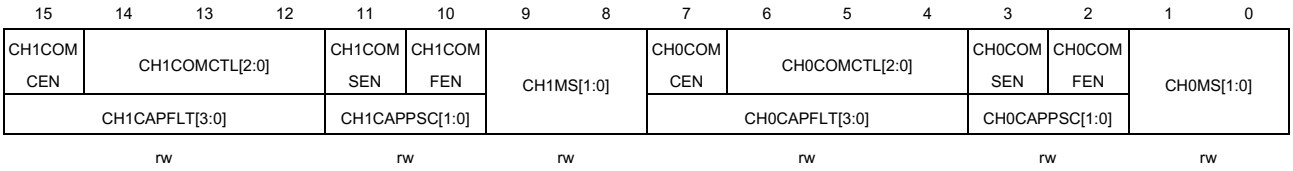
5	保留	必须保持复位值.
4	CH3G	通道 3 捕获或比较事件发生 参见 CH0G 描述
3	CH2G	通道 2 捕获或比较事件发生 参见 CH0G 描述
2	CH1G	通道 1 捕获或比较事件发生 参见 CH0G 描述
1	CH0G	通道 0 捕获或比较事件发生 该位由软件置 1，用于在通道 0 产生一个捕获/比较事件，由硬件自动清 0。当此位被置 1，CH0IF 标志位被置 1，若开启对应的中断和 DMA，则发出相应的中断和 DMA 请求。此外，如果通道 0 配置为输入模式，计数器的当前值被 TIMERx_CH0CV 寄存器捕获，如果 CH0IF 标志位已经为 1，则 CH0OF 标志位被置 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





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 OCPRE 清除源信号输入高电平时，O0CPRE 参考信号被清 0。OCPRE 清除源由 TIMERx_SMCFG 寄存器的 OCRC 位选择。 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 0：通道 0 输出比较快速禁能。仅比较结果输入作为有效边沿时会产生比较匹配，并将 CH0_O 设置为比较电平，激活 CH0_O 输出的最小延时为 5 个时钟周期。 1：通道 0 输出比较快速使能。触发输入信号的有效边沿和比较结果都会产生比较匹配，并将 CH0_O 设置为比较电平。当触发信号的输入作为有效边沿时，激活 CH0_O 输出的最小延时为 3 个时钟周期。
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： 1) 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。 2) 当 TIMER1 使用 ETI 相关功能时，TIMER1 CH0MS 不能配置为 00。

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
CH3COM CEN	CH3COMCTL[2:0]			CH3COM SEN	CH3COM FEN	CH3MS[1:0]		CH2COM CEN	CH2COMCTL[2:0]			CH2COM SEN	CH2COM FEN	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 OCPRE 清除源信号输入高电平时，O0CPRE 参考信号被清 0。OCPRE 清除源由 TIMERx_SMCFG 寄存器的 OCRC 位选择。 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 0：通道 2 输出比较快速禁能。仅比较结果输入作为有效边沿时会产生比较匹配，并将 CH2_O 设置为比较电平，激活 CH2_O 输出的最小延时为 5 个时钟周期。 1：通道 2 输出比较快速使能。触发输入信号的有效边沿和比较结果都会产生比较匹配，并将 CH2_O 设置为比较电平。当触发信号的输入作为有效边沿时，激活 CH2_O 输出的最小延时为 3 个时钟周期。
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
CH3NP	保留	CH3P	CH3EN	CH2NP	保留	CH2P	CH2EN	CH1NP	保留	CH1P	CH1EN	CH0NP	保留	CH0P	CH0EN
	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	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。
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。

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH2VAL[31:16]															
rw															
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
31:0	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

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

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			DMATC[4:0]					保留			DMATA [4:0]				
			rw								rw				

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。

配置寄存器 (TIMERx_CFG)

地址偏移：0xFC
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														CHVSEL	保留
														rw	

位/位域	名称	描述
31:2	保留	必须保持复位值
1	CHVSEL	写捕获比较寄存器选择位 此位由软件写 1 或清 0。 0：无影响。 1：当写入捕获比较寄存器的值与寄存器当前值相等时，写入操作无效
0	保留	必须保持复位值

9.3. 通用定时器 L2(TIMERx, x=13)

9.3.1. 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。可以产生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，可以用来计数，其外部事件可以驱动其他定时器。

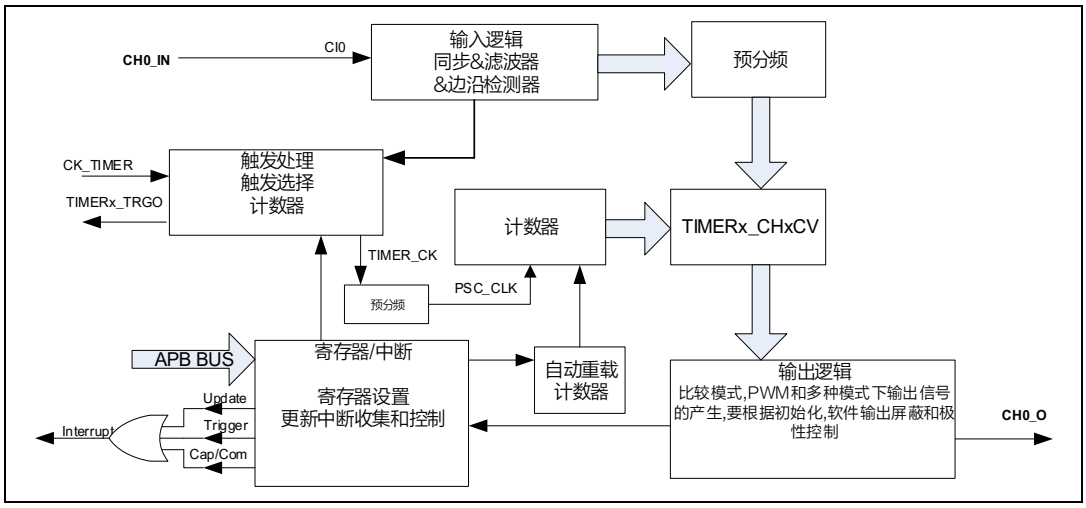
9.3.2. 主要特征

- 总通道数：1；
- 计数器宽度：16位；
- 时钟源：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。

9.3.3. 结构框图

*图 9-43. 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 9-43. 通用定时器 L2 结构框图



9.3.4. 功能说明

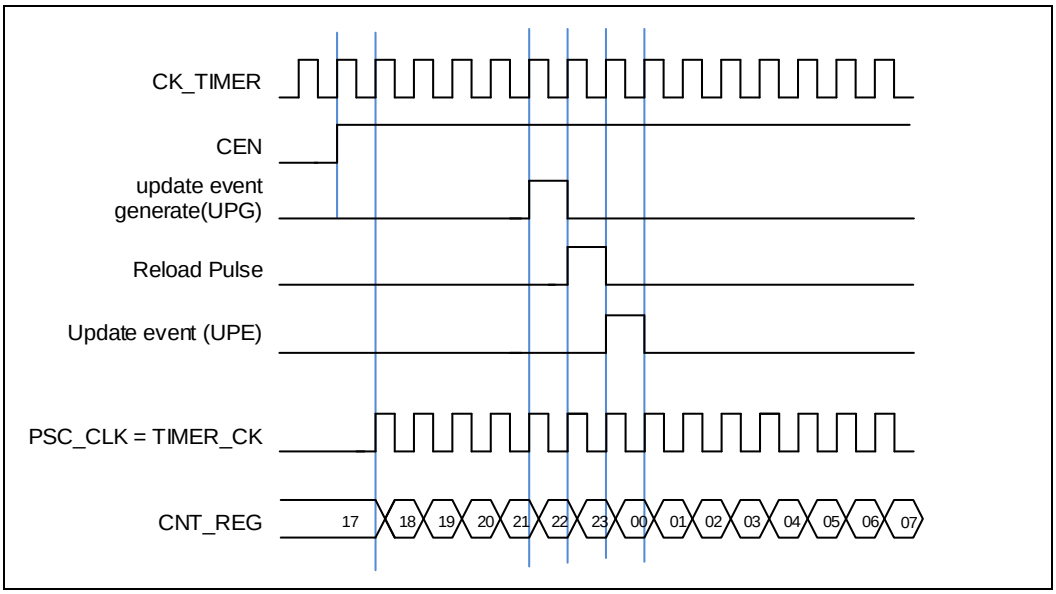
时钟源配置

通用定时器L2可以是内部时钟源CK_TIMER。

- 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。

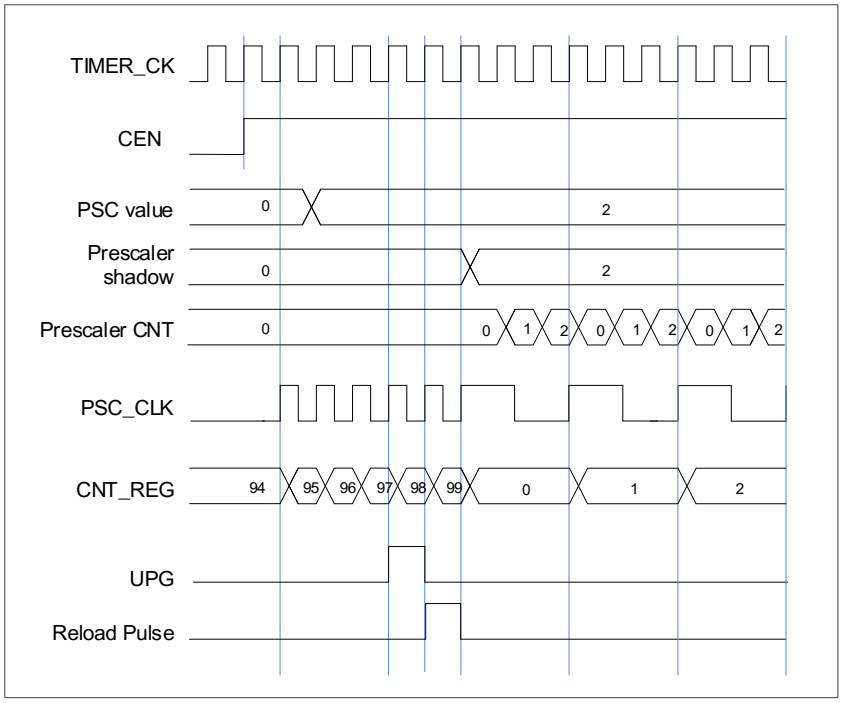
图 9-44. 内部时钟分频为 1 时，计数器的时序图



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 9-45. 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 9-46. 向上计数时序图，PSC=0/2*和*图 9-47. 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 9-46. 向上计数时序图，PSC=0/2

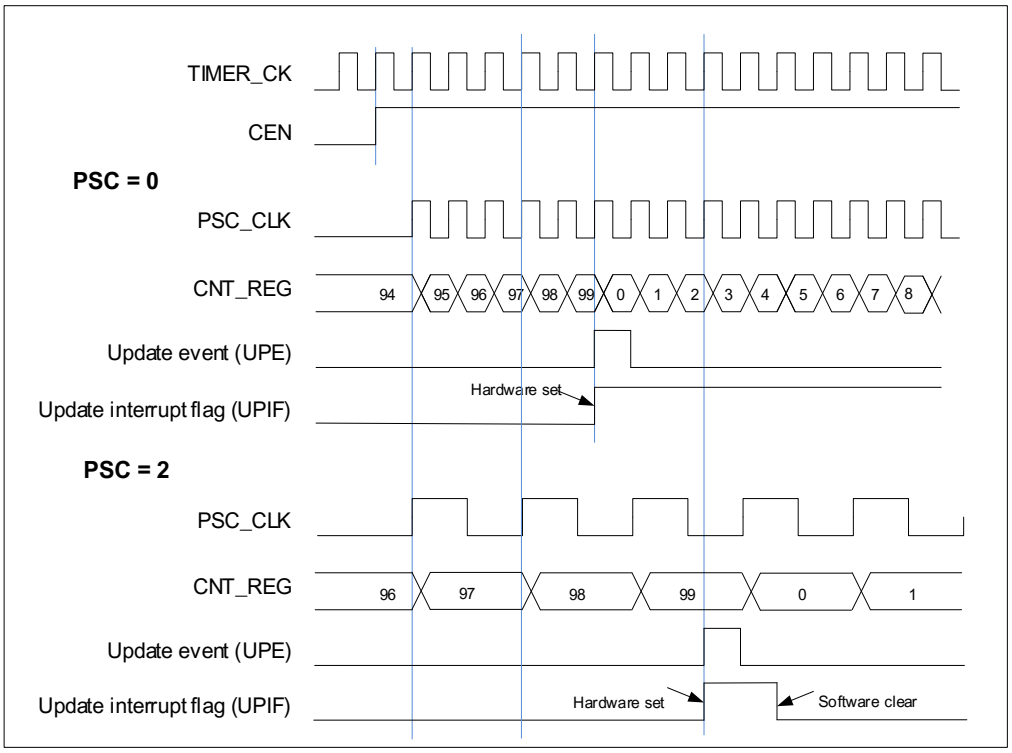
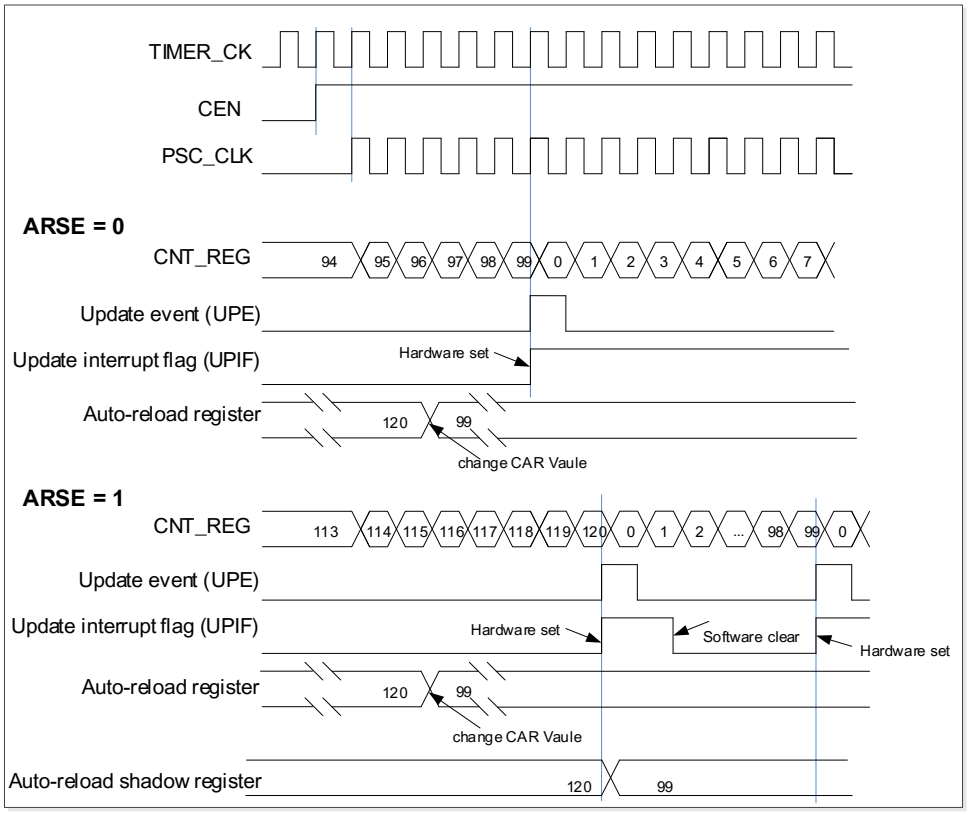


图 9-47. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值



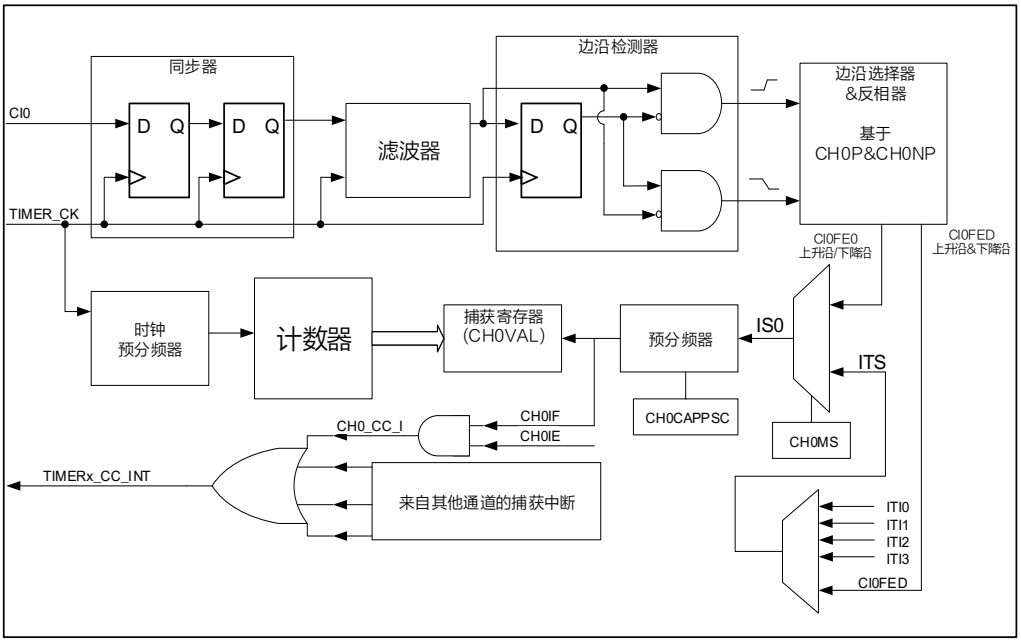
输入捕获和输出比较通道

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。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 9-48. 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。通过配置CHxMS，还可以选择其他通道的输入信号或内部触发信号作为捕获信号。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，TIMERx_CHxCV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，下降沿或者双边沿(上升沿和下降沿)。

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，来确定是否提出相应的中断会。

直接产生：软件设置CHxG位，会直接产生中断请求。

- 通道输出比较功能

图 9-49. 通道输出比较原理

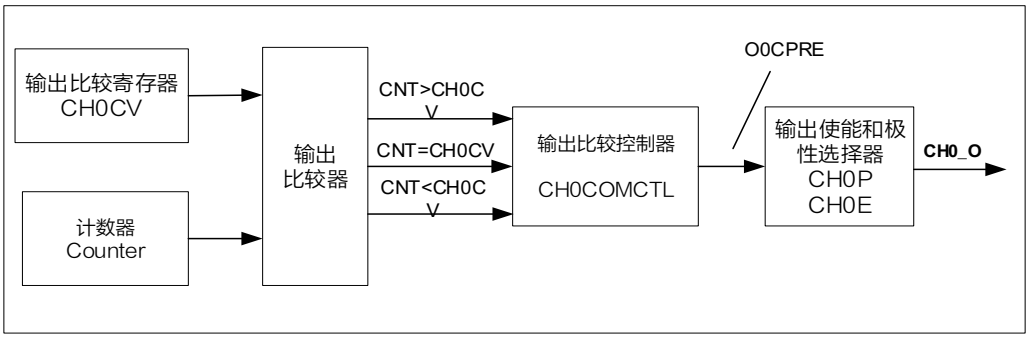


*图9-49. 通道输出比较原理*给出了输出比较的逻辑电路。通道输出信号CHx_O与OxCPRE信号(详情请见*通道输出准备信号*)的关系描述：OxCPRE信号高电平有效，CHx_O的输出情况与OxCPRE信号，CHxP位和CHxE位有关(具体情况请见TIMERx_CHCTL2寄存器中的描述)。例如，当设置CHxP=0(CHx_O高电平有效，与OxCPRE输出极性相同)、CHxE=1(CHx_O输出使能)时：

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TIMERx_CHxCV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，被置低电平或者翻转。当计数器的值与TIMERx_CHxCV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CxCDE=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/翻转)；

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。

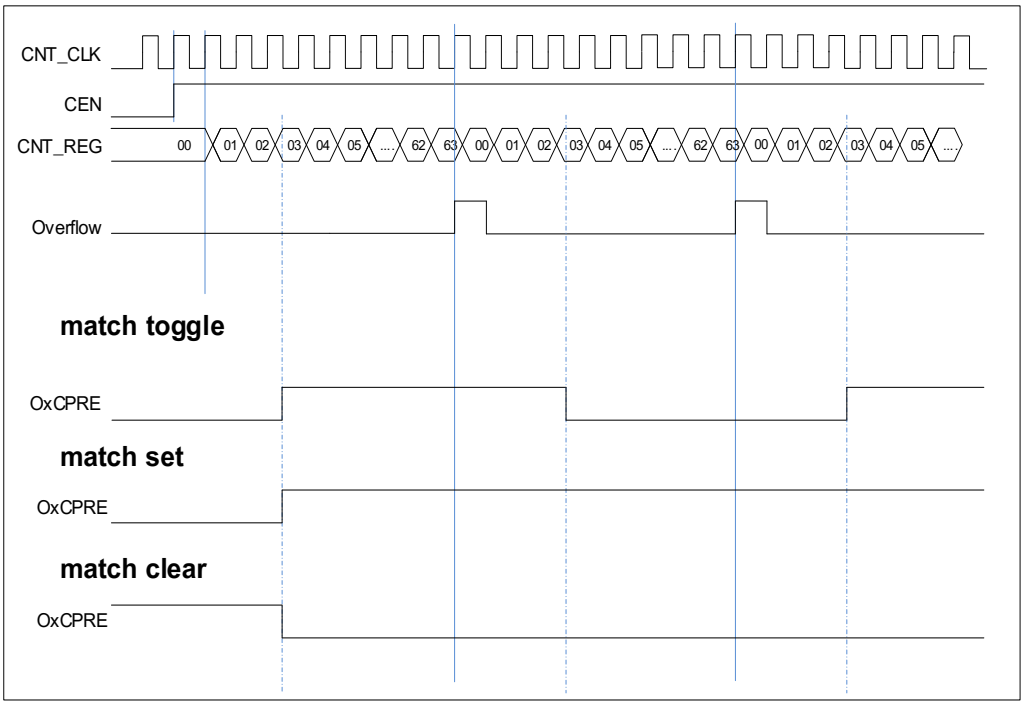
第四步：通过TIMERx_CAR寄存器和TIMERx_CHxCV寄存器配置输出比较时基：

TIMERx_CHxCV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 9-50. 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 9-50. 三种输出比较模式



通道输出准备信号

根据*图9-49. 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®-M23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。

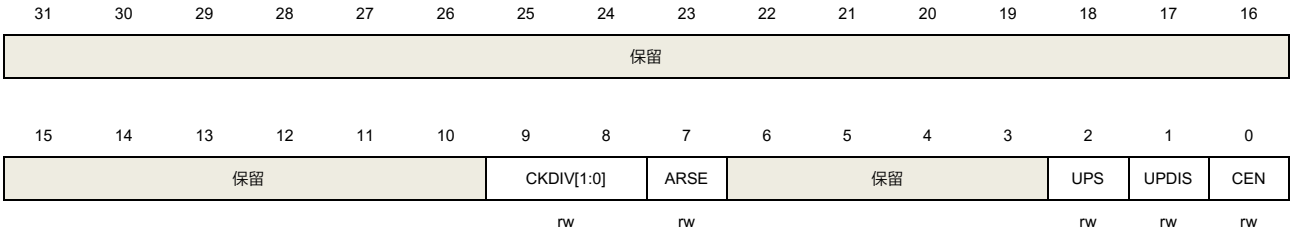
9.3.5. TIMERx 寄存器(x=13)

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 0。 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

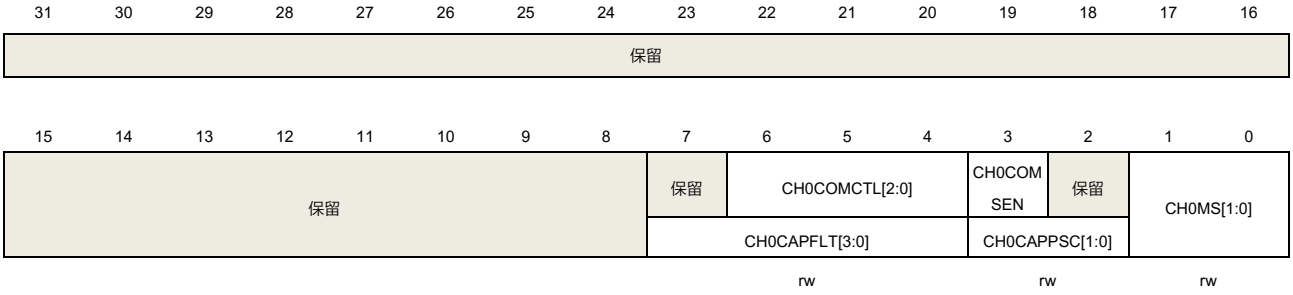
位/位域	名称	描述
31:2	保留	必须保持复位值.

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



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(TIMERx_CTL0 寄存器的 SPM =1)，可以在未确认预装载寄存器情况下使用 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	$f_{DTS}/32$
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

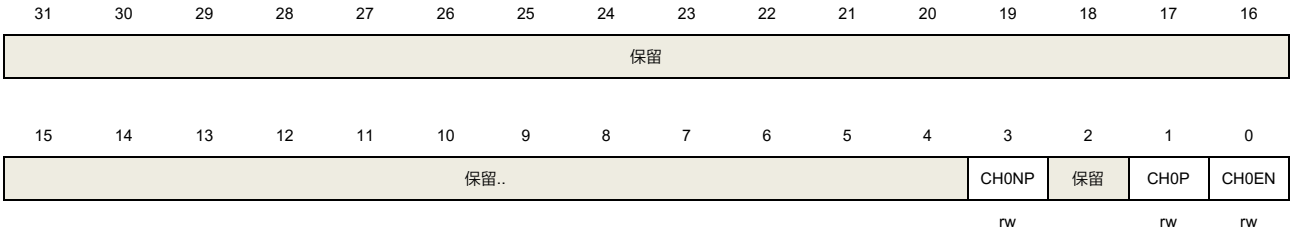
与输出比较模式相同

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。

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。
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

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														CI0_RMP[1:0]	
														rw	

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: 保留 10: 通道 0 输入连接到 HXTAL/32 clock 11: 通道 0 输入连接到 CKOUTSEL

配置寄存器 (TIMERx_CFG)

地址偏移：0xFC
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														CHVSEL	保留
														rw	

位/位域	名称	描述
31:2	保留	必须保持复位值
1	CHVSEL	写捕获比较寄存器选择位 此位由软件写 1 或清 0。 0：无影响。 1：当写入捕获比较寄存器的值与寄存器当前值相等时，写入操作无效
0	保留	必须保持复位值

9.4. 通用定时器 L3(TIMERx,x=14)

9.4.1. 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。

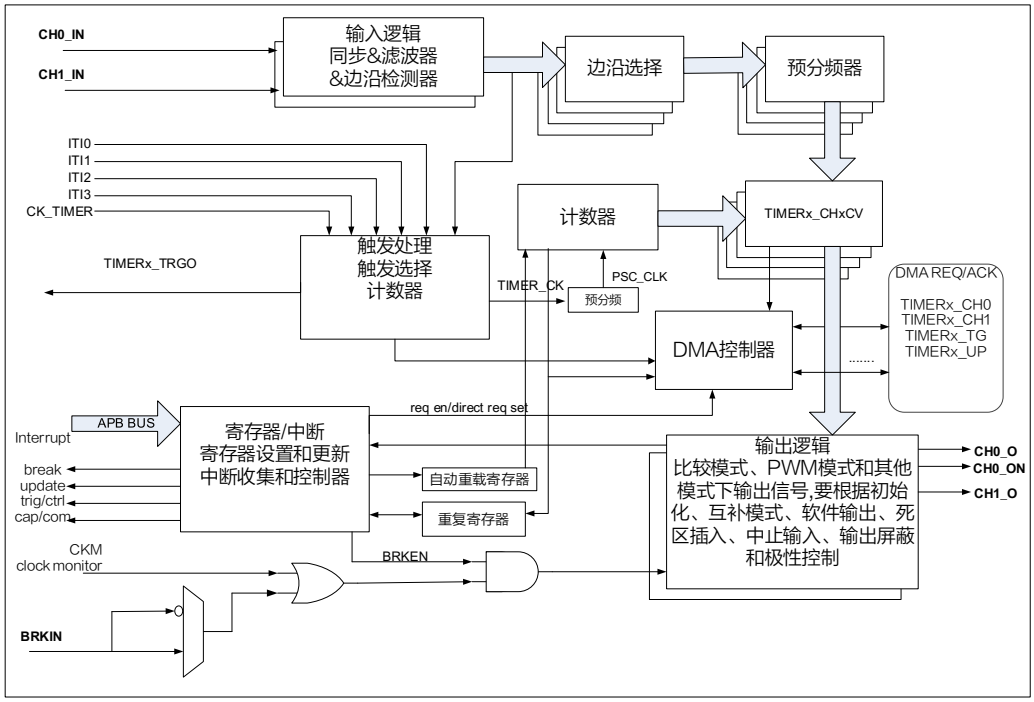
9.4.2. 主要特性

- 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。

9.4.3. 结构框图

*图 9-51. 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 9-51. 通用定时器 L3 结构框图



9.4.4. 功能描述

时钟源配置

通用定时器L0可以是内部时钟源CK_TIMER，或者是由SMC(TIMERx_SMCFG寄存器位[2:0])位确定的时钟源。

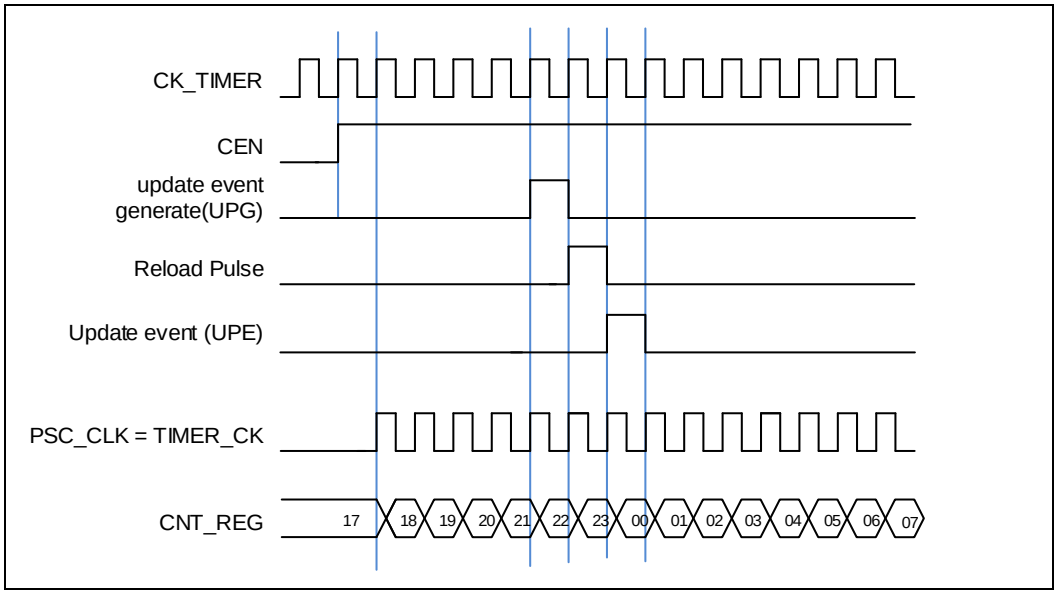
- SMC[2:0]=3'b000，定时器选择内部时钟源(连接到RCU模块的CK_TIMER)

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 9-52. 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。

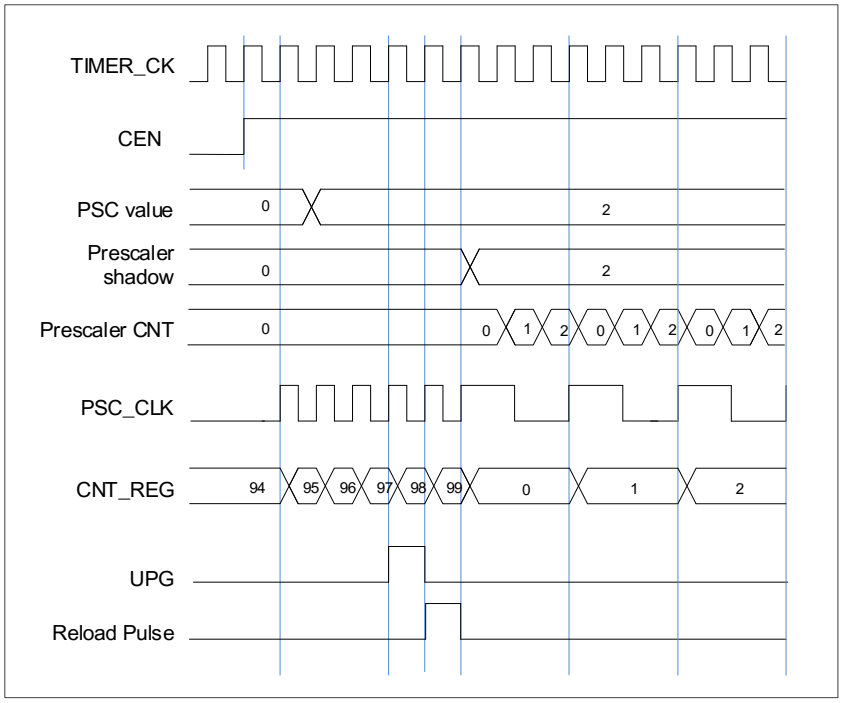
计数器预分频器可以在TIMERx_CH0/ TIMERx_CH1引脚的每个上升沿或下降沿计数。这种模式可以通过设置SMC[2:0]为0x7同时设置TRGS[2:0]为0x4， 0x5或0x6来选择。CIx 是TIMERx_CIx通过数字滤波器采样后的信号。

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。

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 9-53. 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 9-54. 向上计数时序图，PSC=0/2*和*图 9-55. 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 9-54. 向上计数时序图，PSC=0/2

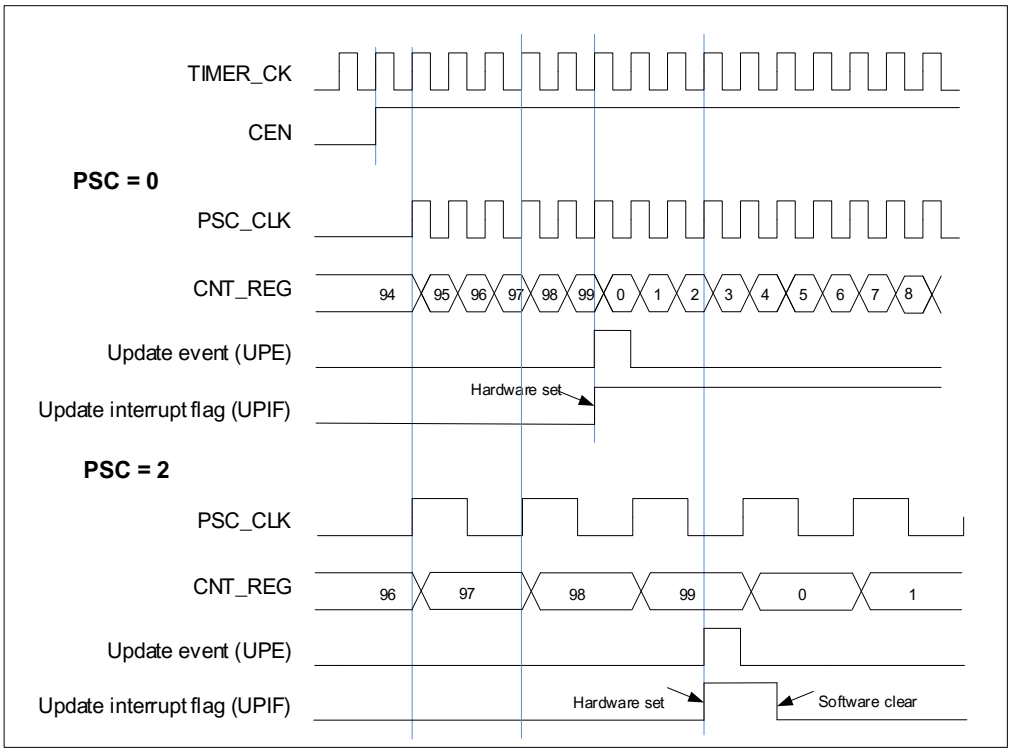
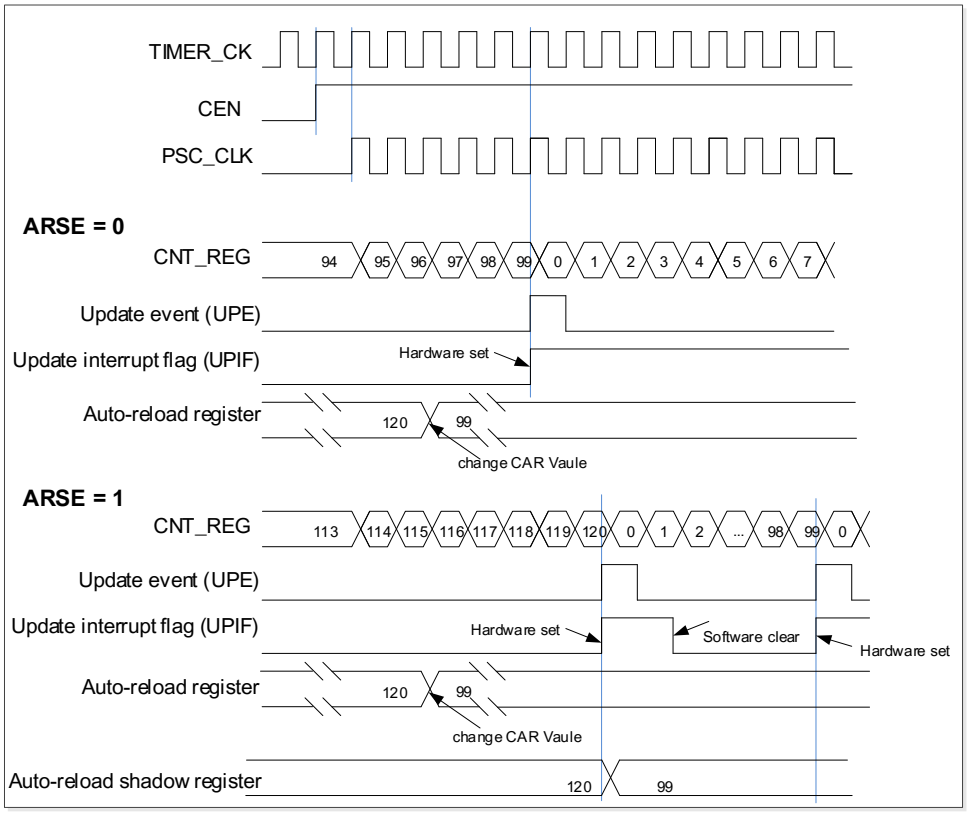


图 9-55. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值

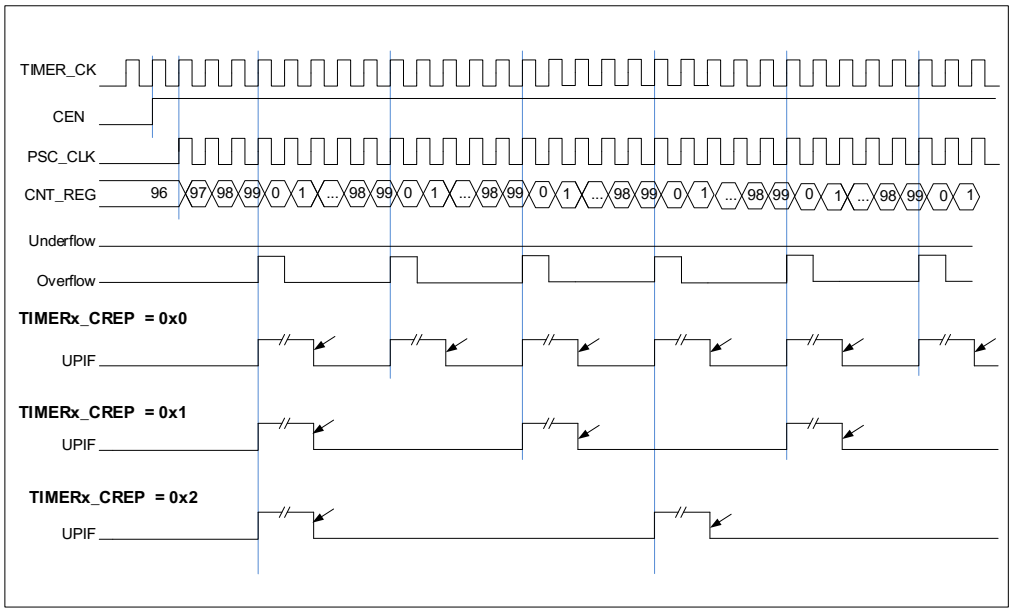


更新事件(来自上溢/下溢)频率配置

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位的值。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 9-56. 在向上计数模式下计数器重复时序图



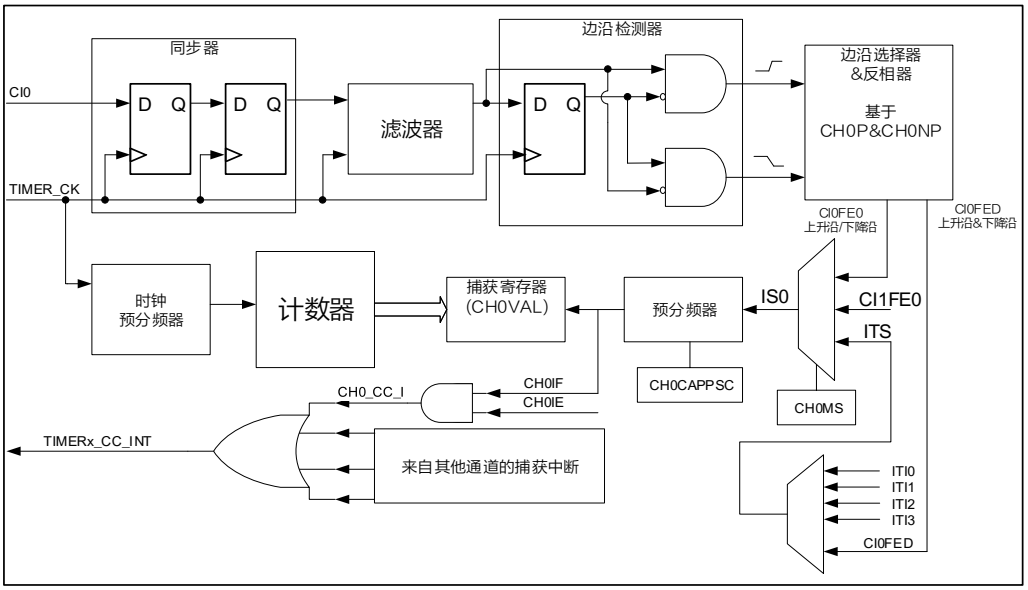
输入捕获和输出比较通道

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。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的时序，频率，周期和占空比等。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，若CHxIE=1则产生通道中断。

图 9-57. 通道输入捕获原理



通道输入信号CIx来源于TIMERx_CHx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。通过配置CHxMS，还可以选择其他通道的输入信号或内部触发信号作为捕获信号。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，TIMERx_CHxCV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，来确定是否提出相应的中断和DMA请求。

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，

同时设置上升沿捕获。配置TIMERx_CHCTL0寄存器中CH1MS为2'b10，选择通道1捕获信号为CI0，同时设置下降沿捕获。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 9-58. 通道输出比较原理(带有互补输出的通道，x=0)

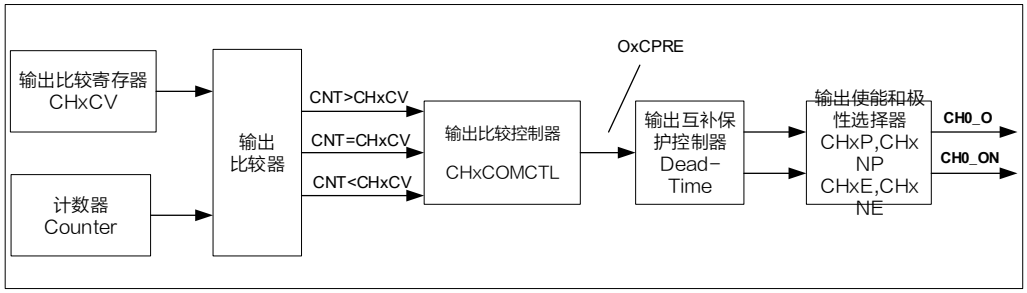


图 9-59. 通道输出比较原理

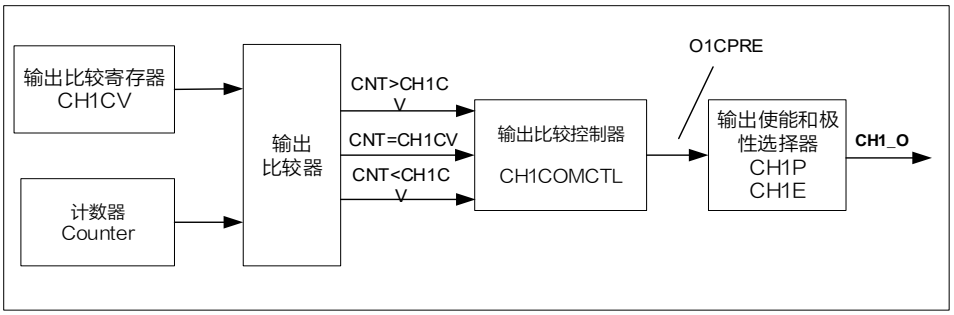


*图9-58. 通道输出比较原理(带有互补输出的通道，x=0)*和*图9-59. 通道输出比较原理*分别给出了输出比较的原理电路。通道输出信号CHx_O/CHx_ON与OxCPRE信号(详情请见*通道输出准备信号*)的关系描述如下：OxCPRE信号高电平有效，CHx_O/CHx_ON的输出情况与OxCPRE信号，CHxP/CHxNP位和CHxE/CHxNE位有关(具体情况请见TIMERx_CHCTL2寄存器中的描述)。例如：

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TIMERx_CHxCV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，被置低电平或者翻转。当计数器的值与TIMERx_CHxCV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CxCDE=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/翻转)；
- 设置CHxP/CHxNP位来选择有效电平的极性；
- 设置CHxEN使能输出。

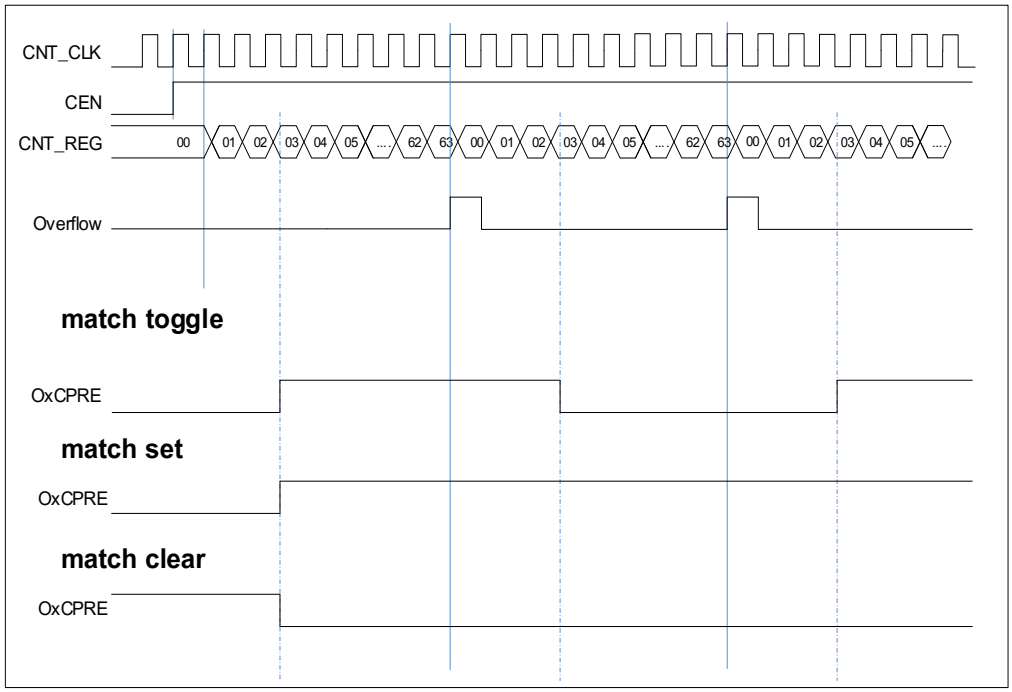
第三步： 通过CHxIE/CxCDE位配置中断/DMA请求使能。

第四步： 通过TIMERx_CAR寄存器和TIMERx_CHxCV寄存器配置输出比较时基：
TIMERx_CHxCV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9-60. 三种输出比较模式*显示了三种比较输出模式：翻转/置高电平/置低电平，CAR=0x63，CHxVAL=0x3。

图 9-60. 三种输出比较模式



输出 PWM 功能

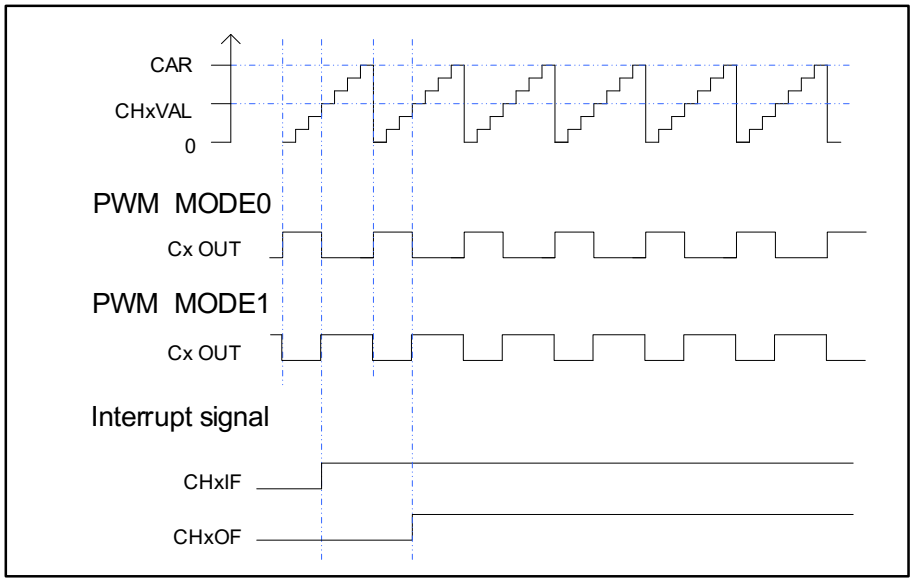
在PWM输出模式下(PWM模式0是配置CHxCOMCTL为3'b110，PWM模式1是配置CHxCOMCTL为3'b111)，通道根据TIMERx_CAR寄存器和TIMERx_CHxCV寄存器的值，输出PWM波形。

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9-61. 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 9-61. PWM 时序图



通道输出准备信号

根据*图9-58. 通道输出比较原理(带有互补输出的通道，x=0)*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。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。

设置CHxCOMCEN=1，当由外部ETI引脚信号产生的ETIFP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寄存器中的POEN、ROS和IOS位，TIMERx_CTL1寄存器中的ISOx和ISOxN位。输出极性由TIMERx_CHCTL2寄存器中的CHxP和CHxNP位来决定。

表 9-6. 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="6">0</td><td rowspan="6">0/1</td><td rowspan="3">0</td><td rowspan="2">0</td><td>0</td><td colspan="2">CHx_O = ISOx CHx_ON = ISOxN
CHx_O / CHx_ON 输出禁能(1)</td></tr>
<tr><td>1</td><td colspan="2">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 rowspan="3">1</td><td rowspan="2">0</td><td>0</td><td colspan="2">CHx_O = ISOx CHx_ON = ISOxN
CHx_O / CHx_ON 输出禁能(1)</td></tr>
<tr><td>1</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>1</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="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
CHx_O输出关闭状态</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_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)⊕CHx NP 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。

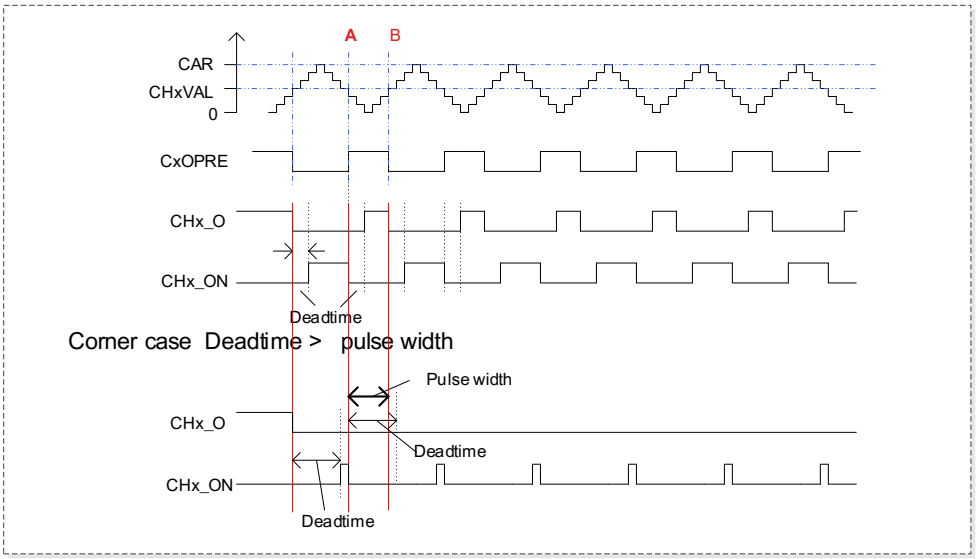
死区时间的插入，确保了通道互补的两路信号不会同时有效。

在 PWM0 模式，当通道 x 匹配发生时(TIMERx 计数器= CHxVAL)，OxCPRE 翻转。*在图 9-62. 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9-62. 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 9-62. 带死区时间的互补输出



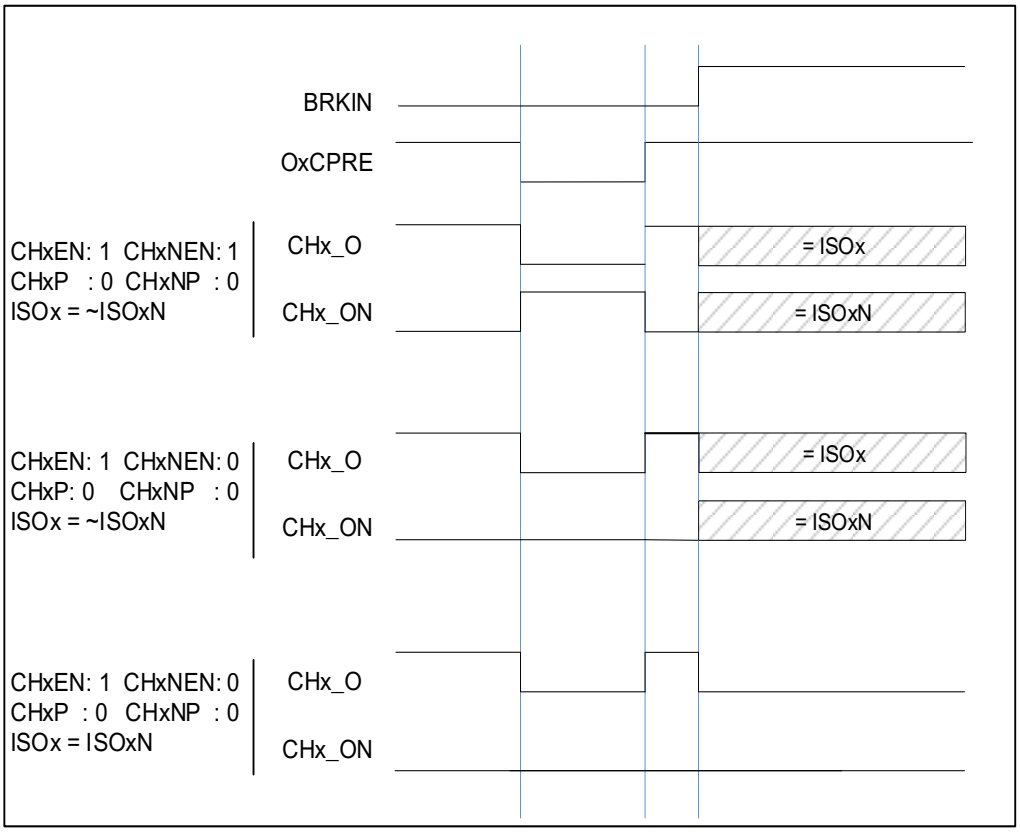
中止模式

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 9-63. 通道响应中止输入(高电平有效)时，输出信号的行为

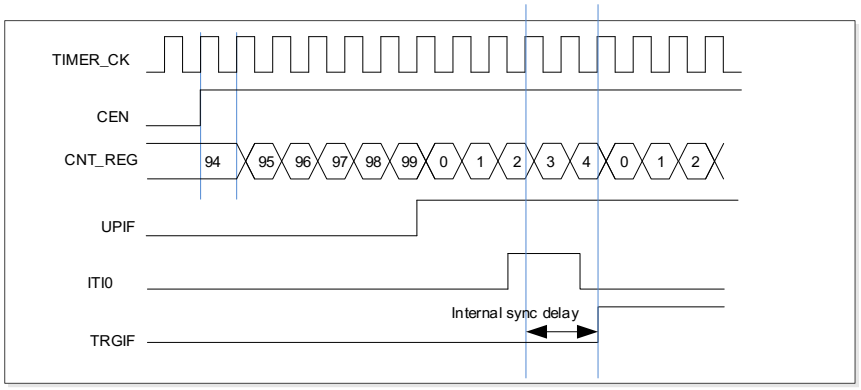
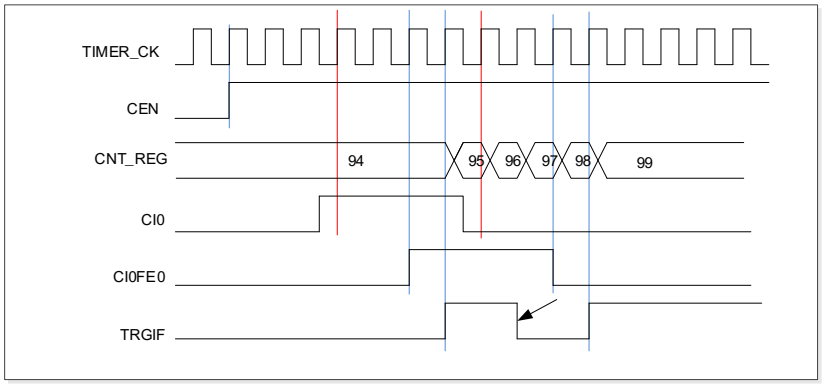
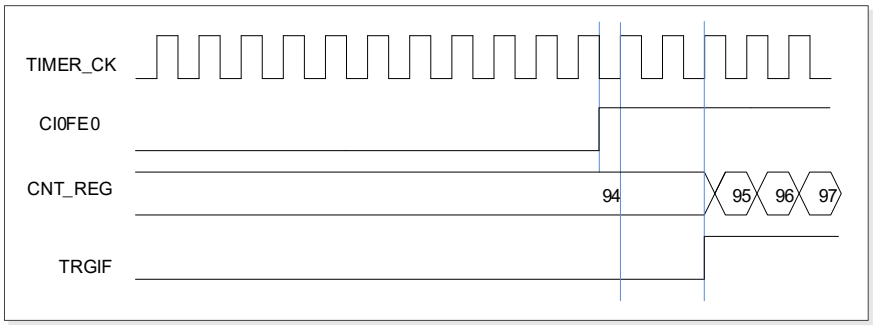


主-从管理

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 9-7. 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

	模式选择	触发源选择	极性选择	滤波和预分频
	图 9-64. 复位模式下的控制电路			
				
例2	暂停模式 当触发输入为低的时候，计数器暂停计数	TRGIS[2:0]=3'b101 选择CI0FE0为触发源	TI0S=0.(非异或) [CH0NP==0, CH0P==0]不反相.在上升沿捕获	在这个例子中滤波被旁路
	图 9-65. 暂停模式下的控制电路			
				
例3	事件模式 触发输入的上升沿计数器开始计数	TRGIS[2:0]=3'b101 选择CI0FE0为触发源.	TI0S=0(非异或) [CH0NP==0,CH0P==0] 不反相	在这个例子中滤波被旁路
	图 9-66. 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

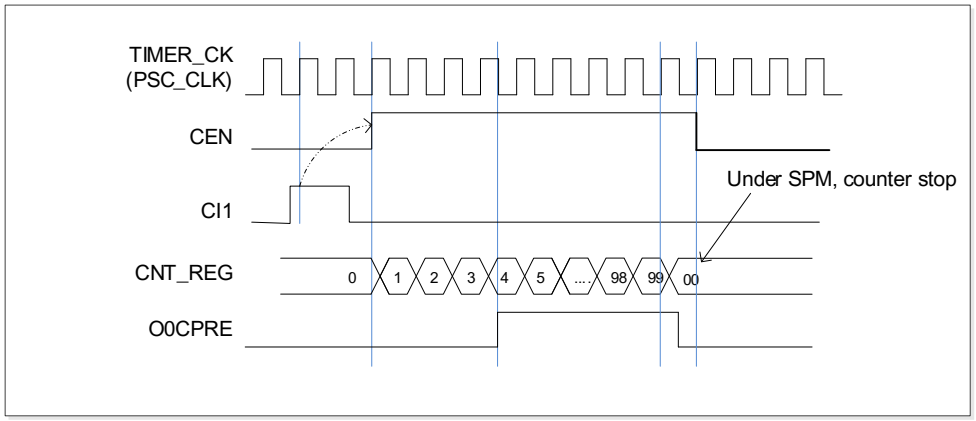
模式。当 SPM 置 1，计数器在下次更新事件到来后清零并停止计数。为了得到脉冲波，可以通过设置 CHxCOMCTL 配置 TIMERx 为 PWM 模式或者比较模式。

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 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 9-67. 单脉冲模式，TIMERx_CHxCV = 4 TIMERx_CAR=99*展示了一个例子。

图 9-67. 单脉冲模式，TIMERx_CHxCV = 4 TIMERx_CAR=99



定时器互连

参考*定时器互连*

表 9-8. 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	保留	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。必须使能相应的DMA请求位，一些内部中断事件才可以产生DMA请求。当中断事件发生，TIMERx会给DMA发送请求。DMA配置成M2P(传输方向为从内存到外设)模式，PADDR(外设基地址)为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®-M23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。

9.4.5. TIMERx 寄存器(x=14)

TIMER13基地址：0x4000 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 I 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：保留 111：保留
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	CMTDEN	保留		CH1DEN	CH0DEN	UPDEN	BRKIE	TRGIE	CMTIE	保留		CH1IE	CH0IE	UPIE
	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
12:11	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

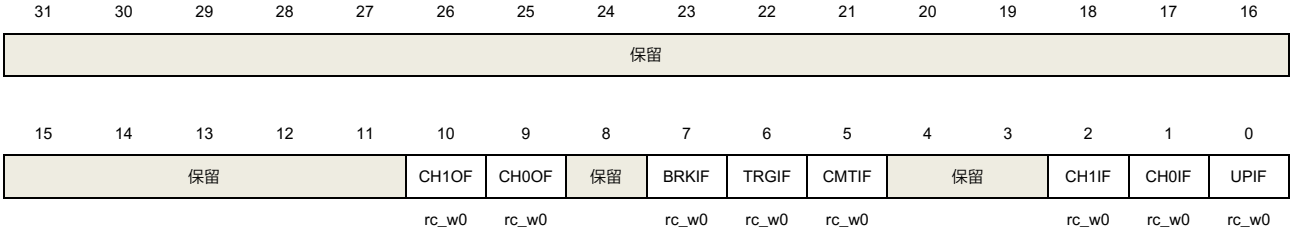
0：禁止通道 0 中断
1：使能通道 0 中断

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 0。 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 0：通道 0 输出比较快速禁能。仅比较结果输入作为有效边沿时会产生比较匹配，并将 CH0_O 设置为比较电平，激活 CH0_O 输出的最小延时为 5 个时钟周期。 1：通道 0 输出比较快速使能。触发输入信号的有效边沿和比较结果都会产生比较匹配，并将 CH0_O 设置为比较电平。当触发信号的输入作为有效边沿时，激活 CH0_O 输出的最小延时为 3 个时钟周期。
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
31:15	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

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								CH1NP	保留	CH1P	CH1EN	CH0NP	CH0NEN	CH0P	CH0EN
								rw		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	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															

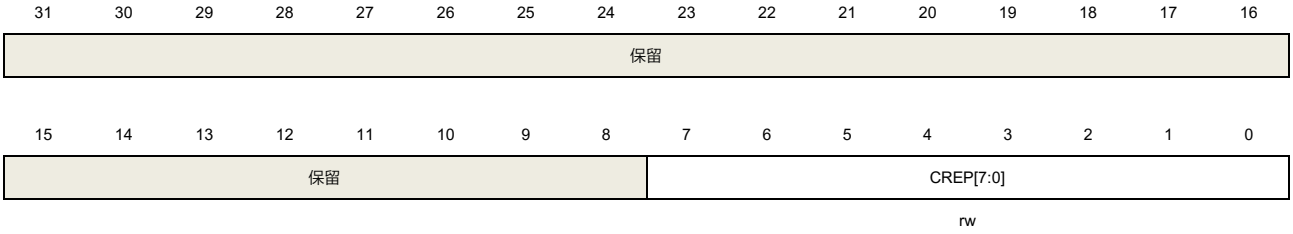
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CARL[15:0]															
rw															

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

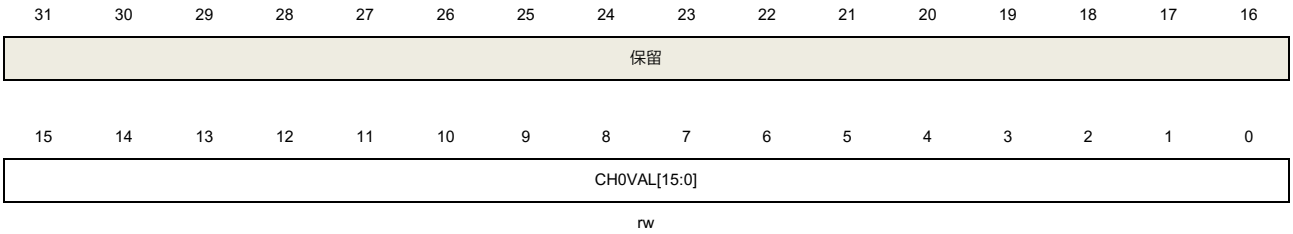


位/位域	名称	描述
31:8	保留	必须保持复位值.
7:0	CREP[7:0]	重复计数器的值 这些位定义了更新事件的产生速率。重复计数器计数值减为 0 时产生更新事件。影子寄存器的更新速率也会受这些位影响(前提是影子寄存器被使能)。

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

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：禁止通道输出或强制为空闲状态。 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 BKIN 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"。参见***表 9-6. 由参数控制的互补输出表***。 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"。参见***表 9-6. 由参数控制的互补输出表***。 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

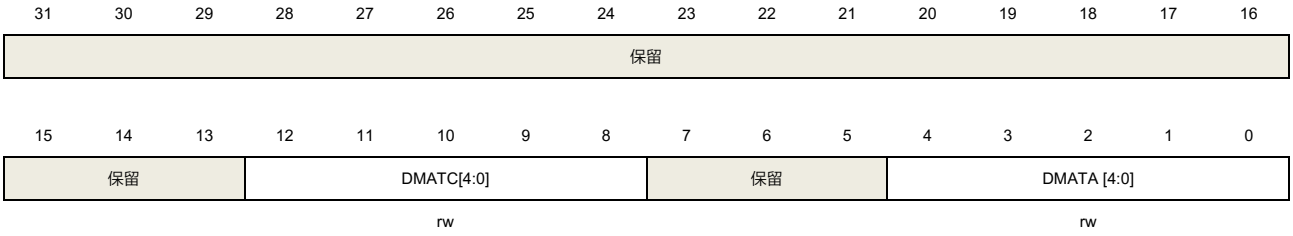
存器中 BRKEN/BRKP/OAEN/DTCFG 位写保护。
10：PROT 模式 1。除了 PROT 模式 0 下的寄存器写保护外，还有 TIMERx_CHCTL2 寄存器中 CHxP/CHxNP 位(如果相应通道配置为输出模式)，TIMERx_CCHP 寄存器中 ROS/IOS 位。
11：PROT 模式 2.。除了 PROT 模式 1 下的寄存器写保护外，还有 TIMERx_CHCTL0 中 CHxCOMCTL/ CHxCOMSEN 位(如果相关通道配置为输出模式)写保护。
系统复位后这两位只能被写一次，一旦 TIMERx_CCHP 寄存器被写入，这两位被写保护。

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

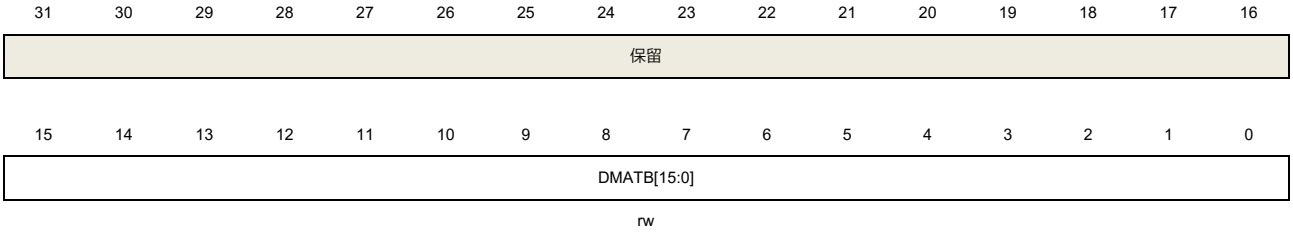


位/位域	名称	描述
31:13	保留	必须保持复位值.
12:8	DMATC [4:0]	DMA 传输计数 该位域定义了 DMA 访问(读写)TIMERx_DMATB 寄存器的数量 n，n = (DMATC [4:0] +1). DMATC [4:0] 从 5'b0_0000 到 5'b1_0001.
7:5	保留	必须保持复位值
4:0	DMATA [4:0]	DMA 传输起始地址 该位域定义了 DMA 访问 TIMERx_DMATB 寄存器的第一个地址。当通过 TIMERx_DMA 第一次访问时，访问的就是该位域指定的地址。第二次访问 TIMERx_DMATB 时，将访问起始地址+0x4。

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

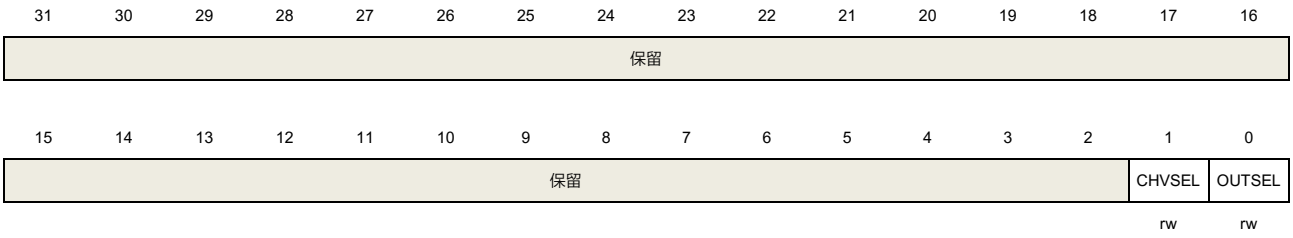


位/位域	名称	描述
31:16	保留	必须保持复位值.
15:0	DMATB [15:0]	DMA 发送缓冲 对这个寄存器的读或写，(起始地址+传输次数*4)地址范围内的寄存器会被访问传输次数由硬件计算，范围为 0 到 DMATC。

配置寄存器 (TIMERx_CFG)

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
31:2	保留	必须保持复位值
1	CHVSEL	写捕获比较寄存器选择位 此位由软件写 1 或清 0。 0：无影响 1：当写入捕获比较寄存器的值与寄存器当前值相等时，写入操作无效
0	OUTSEL	输出值选择位 此位由软件写 1 或清 0。 0：无影响

1：如果 POEN 位与 IOS 位均为 0，则输出无效

9.5. 通用定时器 L4(TIMERx,x=15,16)

9.5.1. 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。

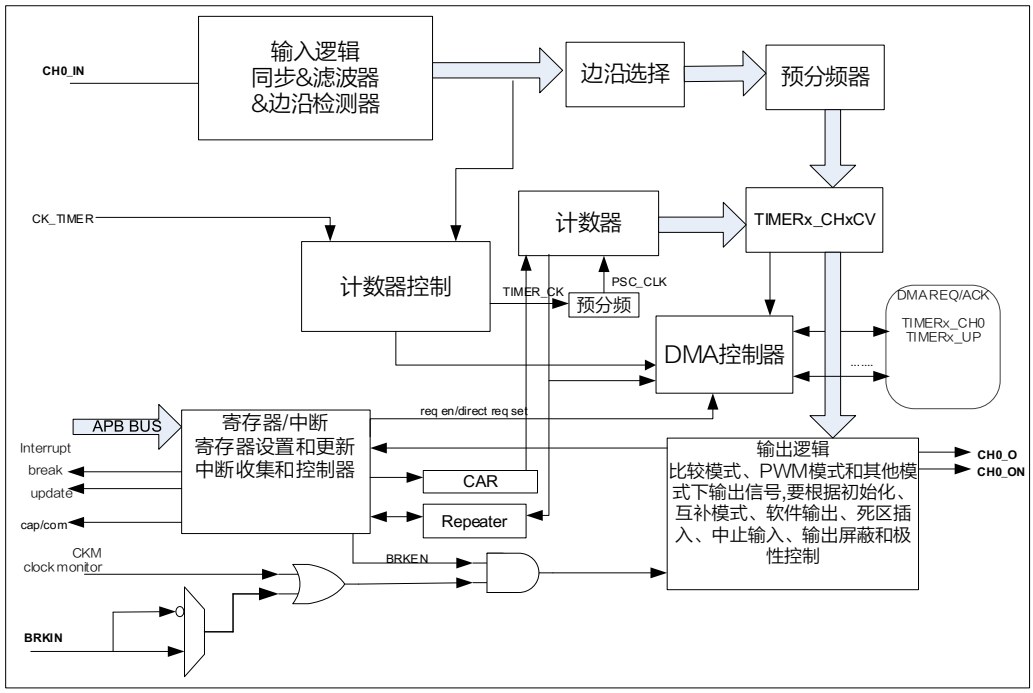
9.5.2. 主要特性

- 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/捕获事件和中止事件；

9.5.3. 结构框图

*图 9-68. 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 9-68. 通用定时器 L4 结构框图



9.5.4. 功能描述

时钟源配置

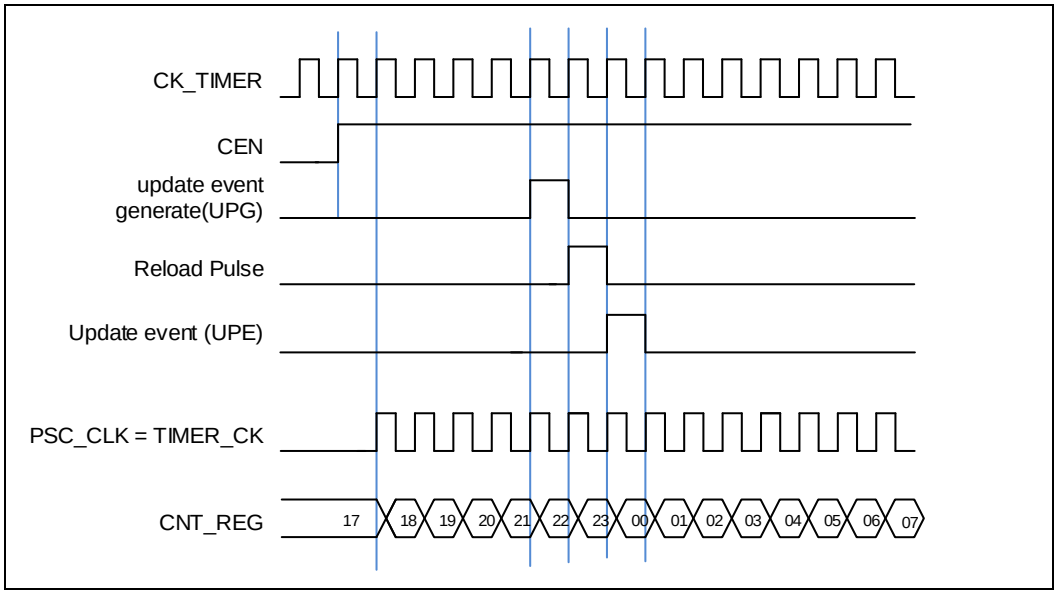
通用定时器L4由内部时钟源CK_TIMER驱动.

- 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

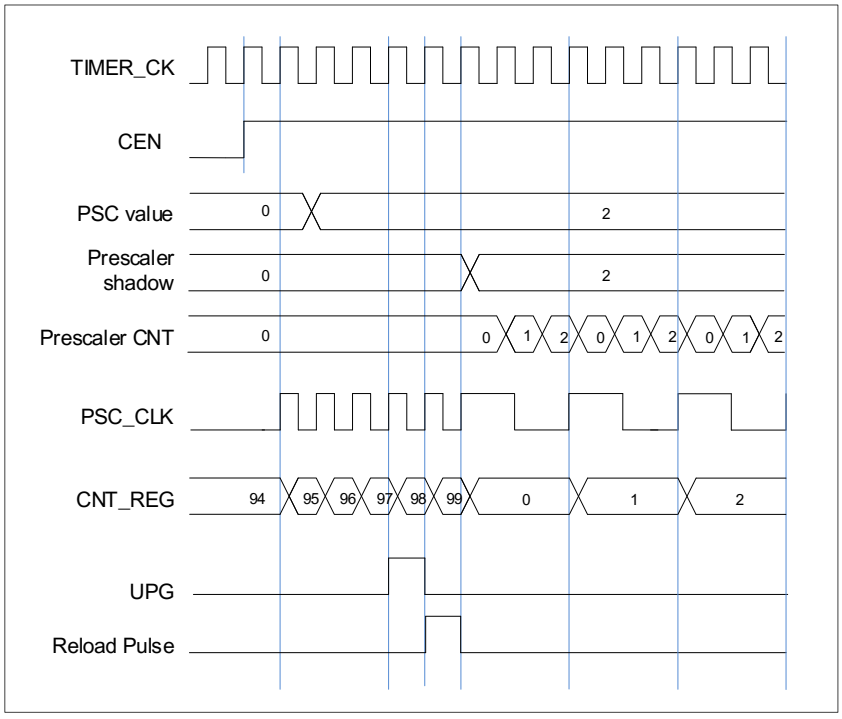
图 9-69. 内部时钟分频为 1 时，计数器的时序图



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 9-70. 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 9-71. 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 9-71. 向上计数时序图，PSC=0/2

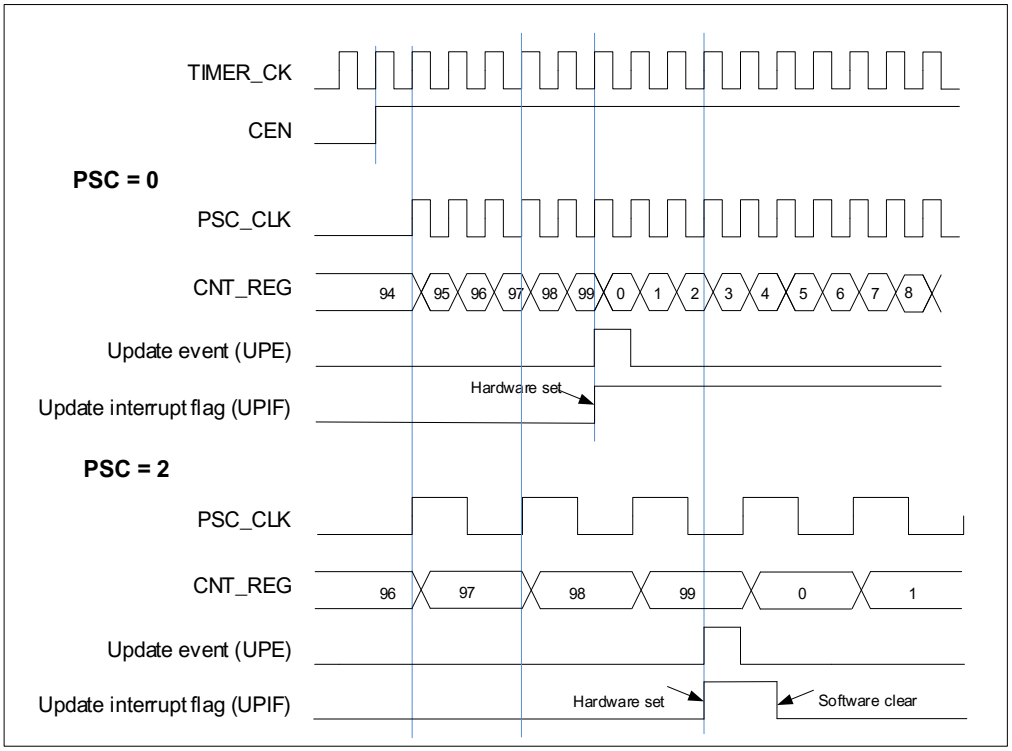
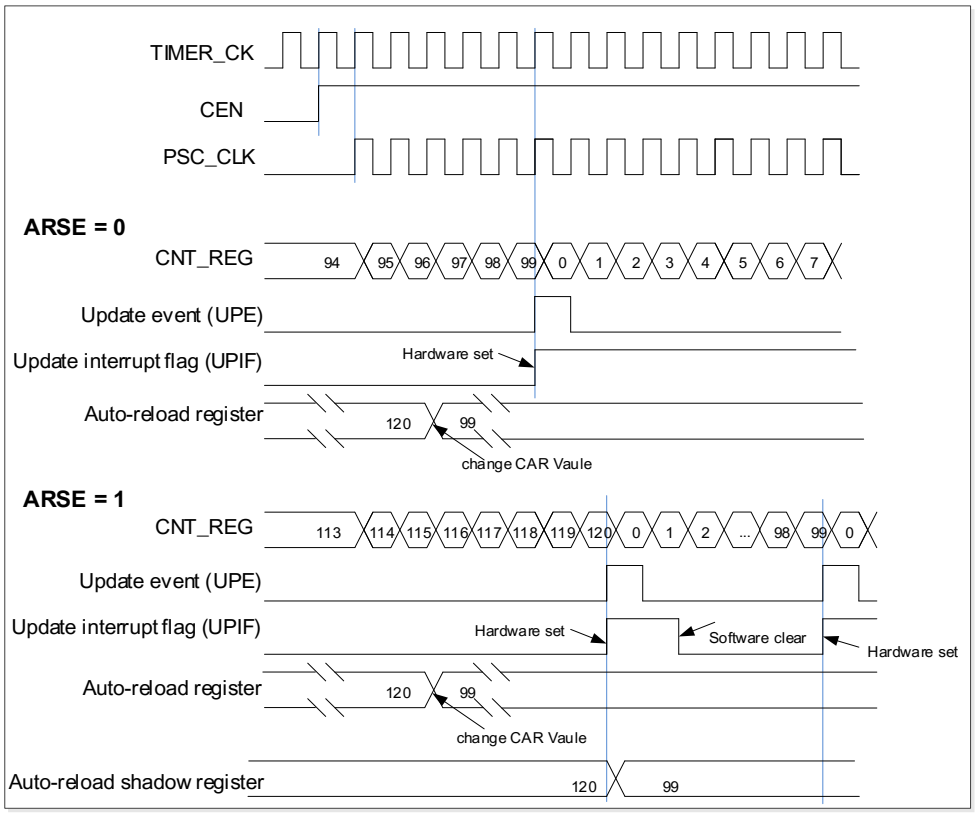


图 9-72. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值

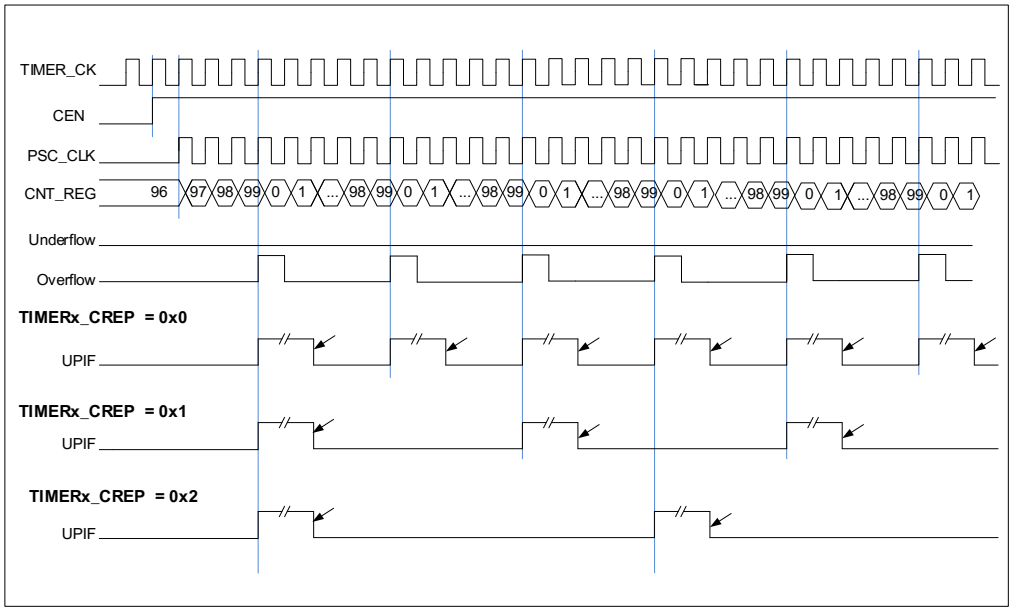


更新事件(来自上溢/下溢)频率配置

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 9-73. 在向上计数模式下计数器重复时序图



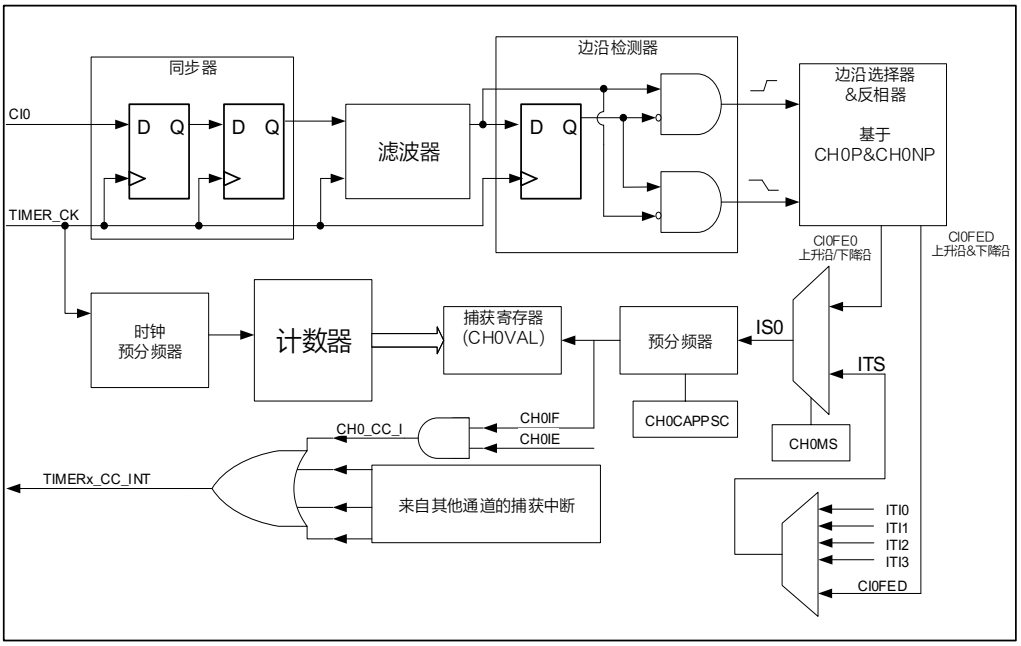
输入捕获和输出比较通道

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。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

输入捕获模式允许通道测量一个波形的时序，频率，周期等。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，若CHxIE=1则产生通道中断。

图 9-74. 通道输入捕获原理



通道输入信号CIx来源于TIMERx_CHx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。通过配置CHxMS，还可以选择其他通道的输入信号或内部触发信号作为捕获信号。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，TIMERx_CHxCV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，来确定是否提出相应的中断和DMA请求。

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周期。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，同时设置上升沿捕获。TIMERX_CH0CV寄存器测量PWM的周期值。

- 通道输出比较功能

图 9-75. 通道输出比较原理(带有互补输出的通道，x=0)

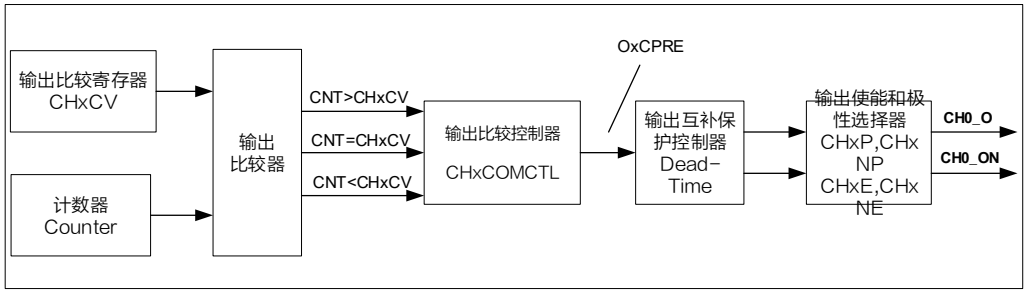


*图9-75. 通道输出比较原理(带有互补输出的通道，x=0)*给出了输出比较的原理电路。通道输出信号CHx_O/CHx_ON与OxCPRE信号(详情请见*通道输出准备信号*)的关系描述如下：

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TIMERx_CHxCV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，被置低电平或者翻转。当计数器的值与TIMERx_CHxCV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CxCDE=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/翻转)；
- 设置CHxP/CHxNP位来选择有效电平的极性；
- 设置CHxEN使能输出。

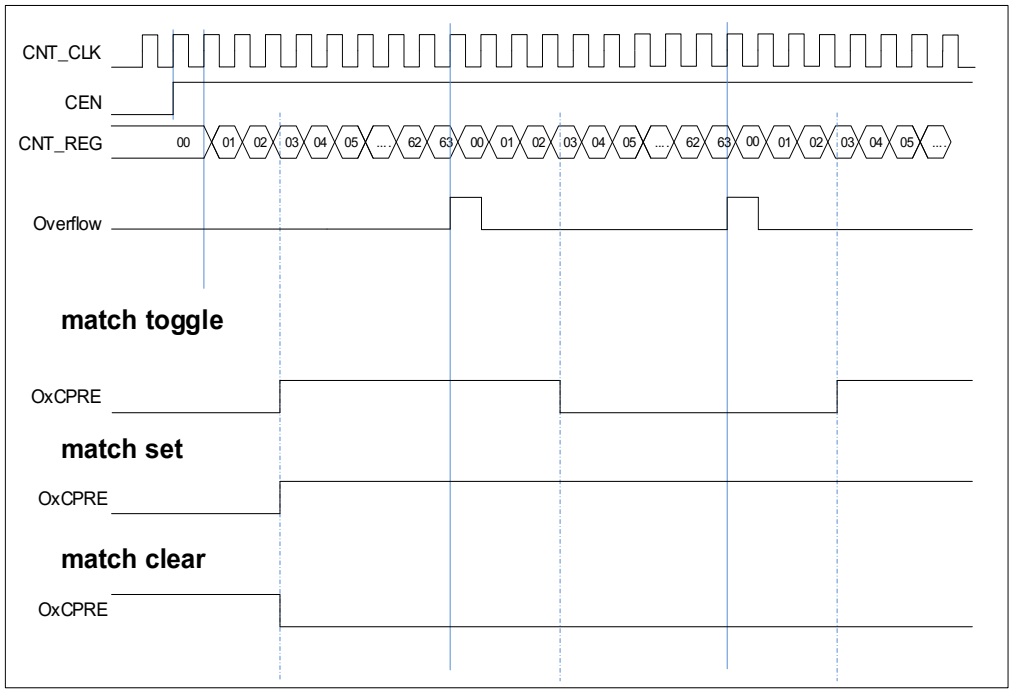
第三步：通过CHxIE/CxCDE位配置中断/DMA请求使能。

第四步：通过TIMERx_CAR寄存器和TIMERx_CHxCV寄存器配置输出比较时基：
TIMERx_CHxCV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9-76. 三种输出比较模式*显示了三种比较输出模式：翻转/置高电平/置低电平，CAR=0x63，CHxVAL=0x3。

图 9-76. 三种输出比较模式



PWM 模式

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。

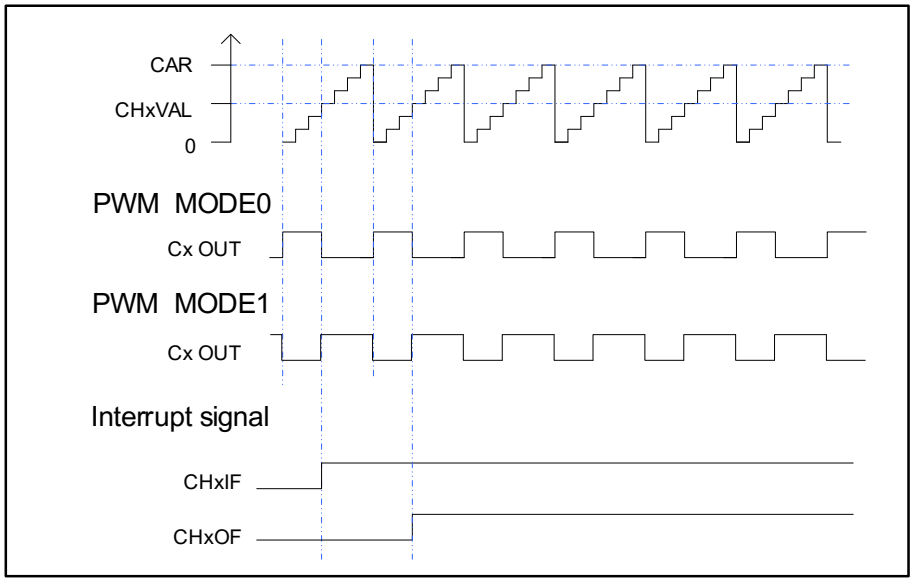
根据计数模式，我们可以分为两种PWM波：EAPWM(边沿对齐PWM)和CAPWM(中央对齐PWM)。

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 9-77. 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 9-77. PWM 时序图



通道输出准备信号

根据*图9-75. 通道输出比较原理(带有互补输出的通道，x=0)*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。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。

设置CHxCOMCEN=1，当由外部ETI引脚信号产生的ETIFP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寄存器中的POEN、ROS和IOS位，TIMERx_CTL1寄存器中的ISOx和ISOxN位。输出极性由TIMERx_CHCTL2寄存器中的CHxP和CHxNP位来决定。

表 9-9. 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 = ISOx CHx_ON = ISOxN 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	0	0	CHx_O = ISOx CHx_ON = ISOxN CHx_O / CHx_ON 输出禁能(1)	
				1	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		
				1		
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_ON=(!OxCPRE)⊕CHx

互补参数					输出状态	
POEN	ROS	IOS	CHxEN	CHxNEN	CHx_O	CHx_ON
					CHx_O输出使能	NP 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。

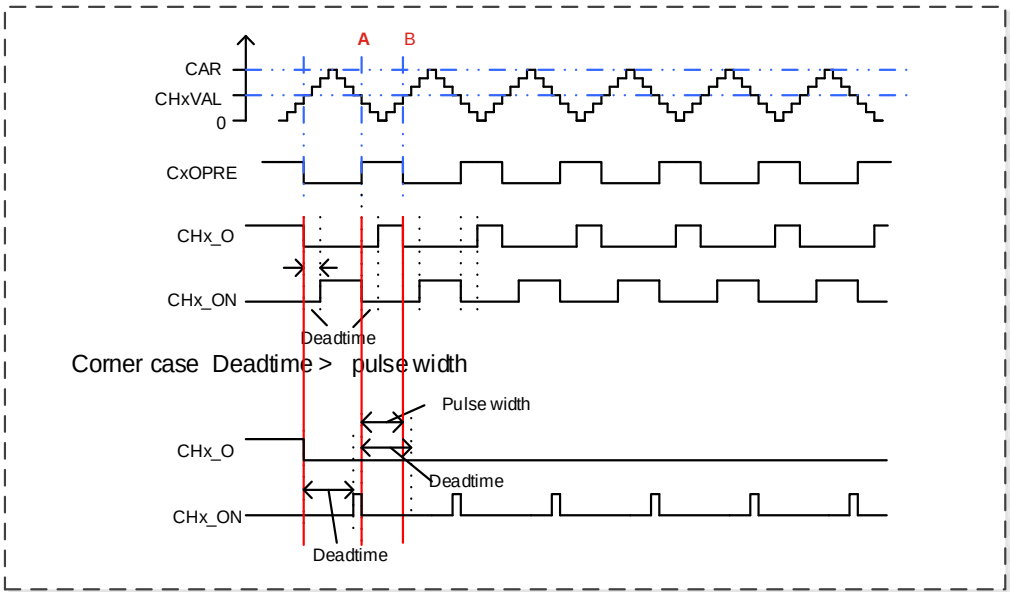
死区时间的插入，确保了通道互补的两路信号不会同时有效。

在PWM0模式，当通道x匹配发生时(TIMERx计数器= CHxVAL)，OxCPRE翻转。在*图9-78. 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9-78. 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 9-78. 带死区时间的互补输出



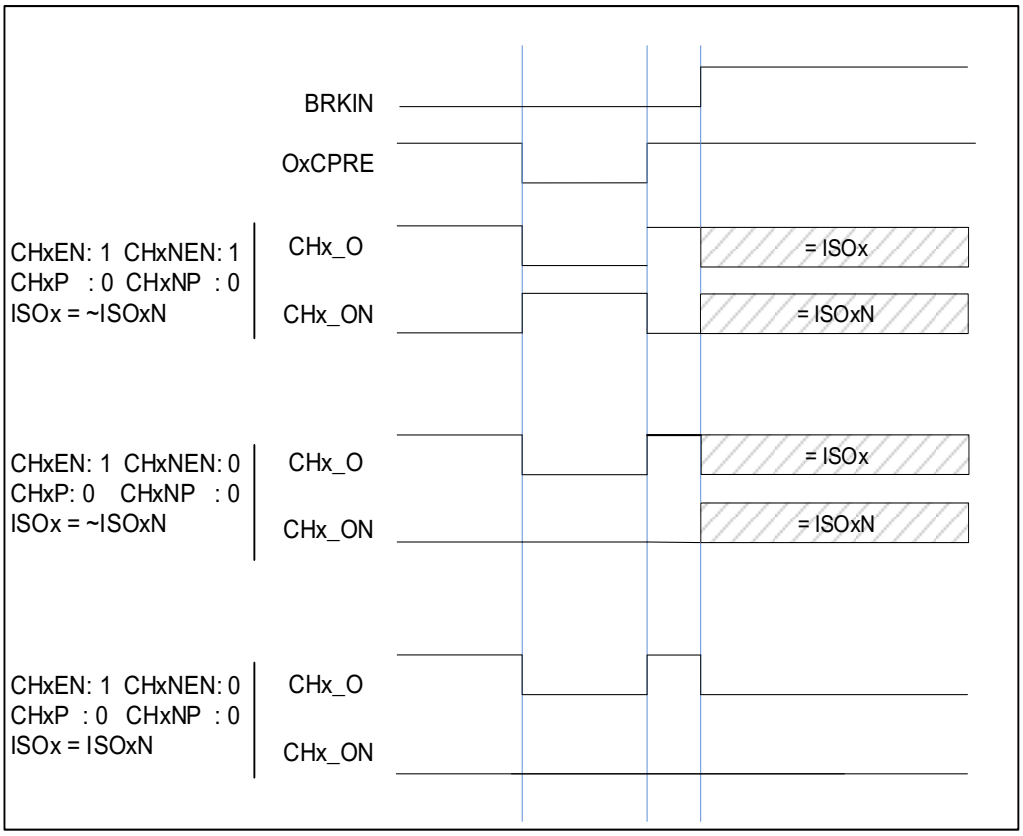
中止模式

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 9-79. 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

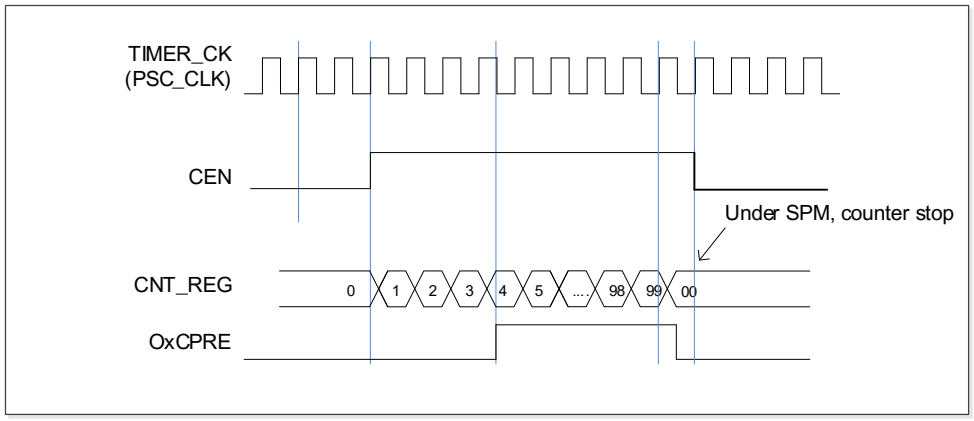
单脉冲模式与重复模式是相反的，设置 TIMERx_CTL0 寄存器的 SPM 位置 1，则使能单脉冲模式。当 SPM 置 1，计数器在下次更新事件到来后清零并停止计数。为了得到脉冲波，可以通过设置 CHxCOMCTL 配置 TIMERx 为 PWM 模式或者比较模式。

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 9-80. 单脉冲模式，TIMERx_CHxCV = 4 TIMERx_CAR=99*展示了一个例子。

图 9-80. 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®-M23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。

9.5.5. TIMERx 寄存器(x=15,16)

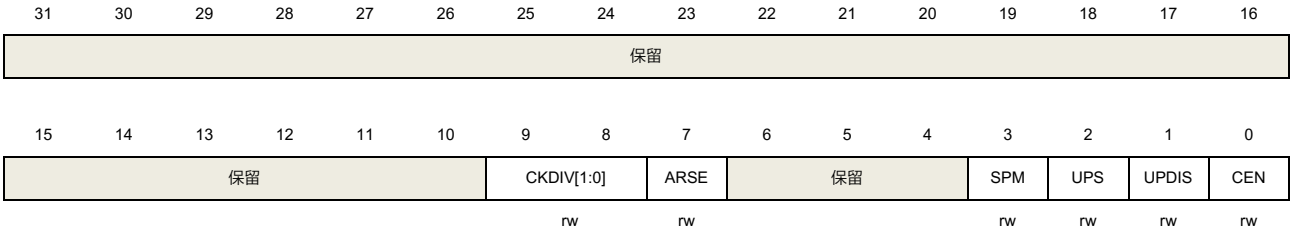
TIMER15基地址：0x4001 4400

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



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

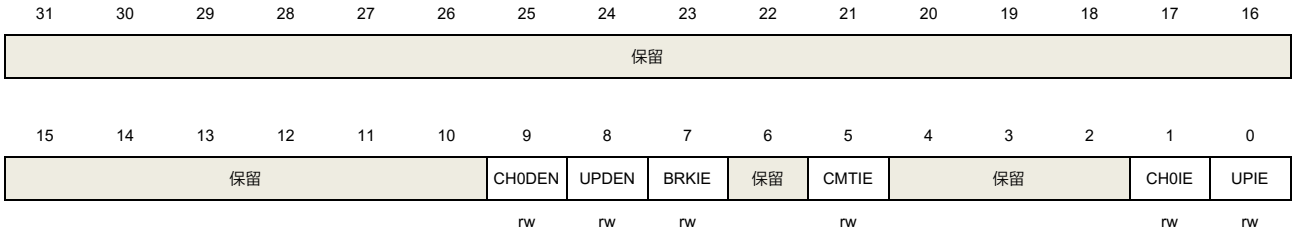
位/位域	名称	描述
31:10	保留	必须保持复位值
9	ISO0N	通道 0 的互补通道空闲状态输出 0：当 POEN 复位，CH0_ON 设置低电平 1：当 POEN 复位，CH0_ON 设置高电平 此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0]位为 00 的时候可以被更改
8	ISO0	通道 0 的空闲状态输出 0：当 POEN 复位，CH0_O 设置低电平 1：当 POEN 复位，CH0_O 设置高电平 如果 CH0_ON 生效，一个死区时间后 CH0_O 输出改变。此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0]位为 00 的时候可以被更改。
7:4	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



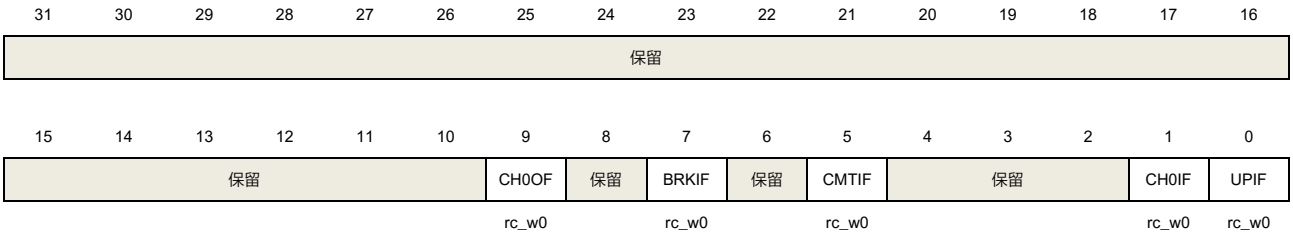
位/位域	名称	描述
31:10	保留	必须保持复位值
9	CH0DEN	通道 0 比较/捕获 DMA 请求使能 0：禁止通道 0 比较/捕获 DMA 请求 1：使能通道 0 比较/捕获 DMA 请求
8	UPDEN	更新 DMA 请求使能 0：禁止更新 DMA 请求 1：使能更新 DMA 请求
7	BRKIE	中止中断使能 0：禁止中止中断 1：使能中止中断

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



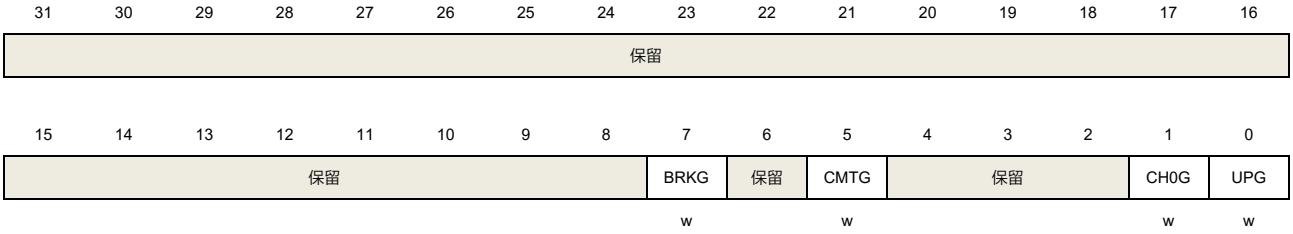
位/位域	名称	描述
31:10	保留	必须保持复位值
9	CH0OF	通道 0 捕获溢出标志 当通道 0 被配置为输入模式时，在 CH0IF 标志位已经被置 1 后，捕获事件再次发生时，该标志位可以由硬件置 1。该标志位由软件清 0。 0：无捕获溢出中断发生 1：发生了捕获溢出中断
8	保留	必须保持复位值.
7	BRKIF	中止中断标志位 当中止输入有效时，由硬件对该位置'1'。 当中止输入无效时，则该位可由软件清'0'。 0：无中止事件产生 1：中止输入上检测到有效电平
5	CMTIF	通道换相更新中断标志 当通道换相更新事件发生时此标志位被硬件置 1，此位由软件清 0。 0：无通道换相更新中断发生

		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 0。 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



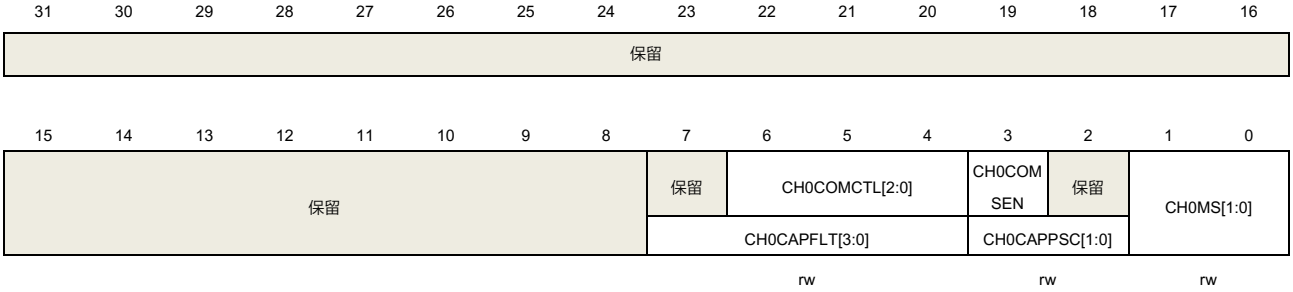
位/位域	名称	描述
31:8	保留	必须保持复位值
7	BRKG	产生中止事件 该位由软件置 1，用于产生一个中止事件，由硬件自动清 0。当此位被置 1 时，POEN 位被清 0 且 BRKIF 位被置 1，若开启对应的中断和 DMA，则产生相应的中断和 DMA 传输。 0：不产生中止事件 1：产生中止事件
5	CMTG	通道换相更新事件发生 此位由软件置 1，由硬件自动清 0。当此位被置 1，通道捕获/比较控制寄存器 (CHxEN, CHxNEN 和 CHxCOMCTL) 的互补输出被更新(根据 TIMERx_CTL1 中 CCSE 值)。 0：不产生通道控制更新事件 1：产生通道控制更新事件
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，在向上计数模式中，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



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
15:8	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

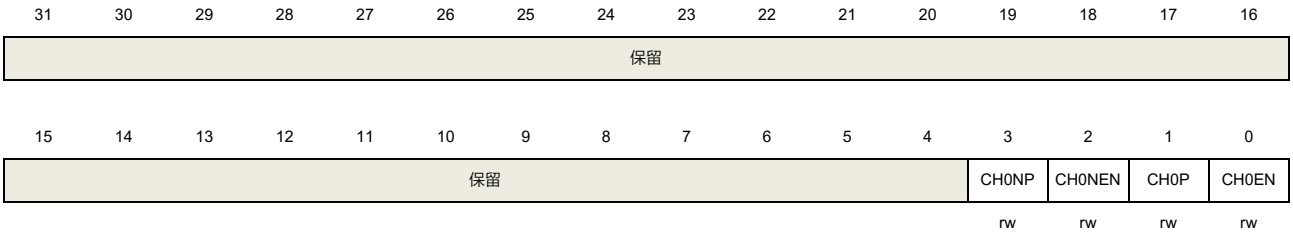
与输出比较模式相同

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															

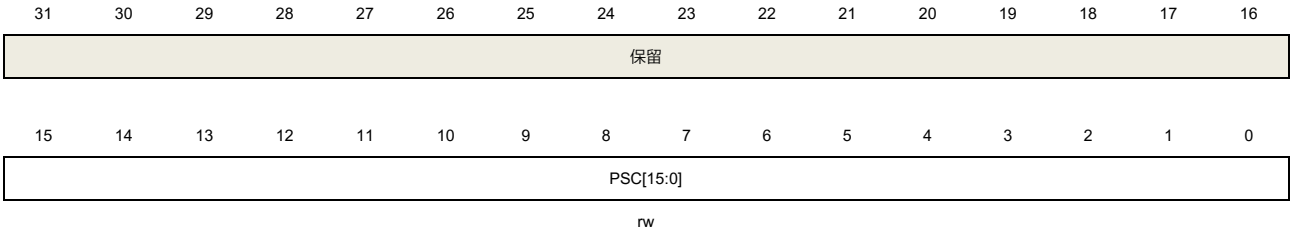
位/位域	名称	描述
31:16	保留	必须保持复位值

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

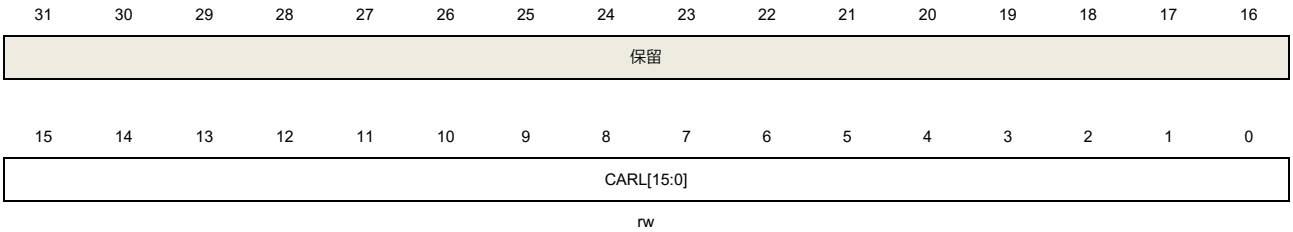


位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	PSC[15:0]	计数器时钟预分频值 计数器时钟等于 TIMER_CK 时钟除以(PSC+1)，每次当更新事件产生时，PSC 的值被装入到对应的影子寄存器。

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



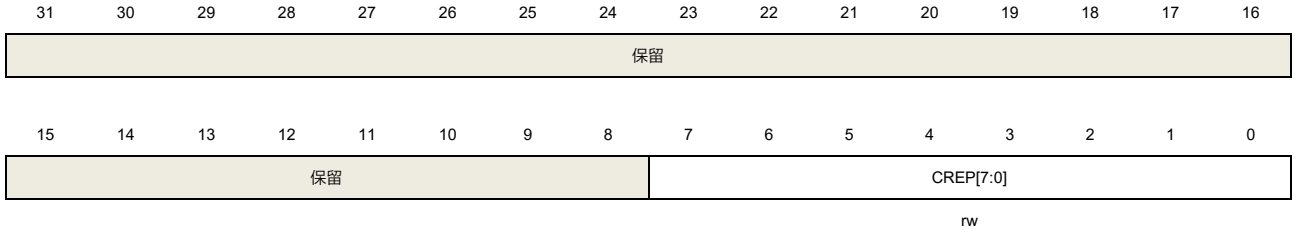
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CARL[15:0]	计数器自动重载值 这些位定义了计数器的自动重载值。

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

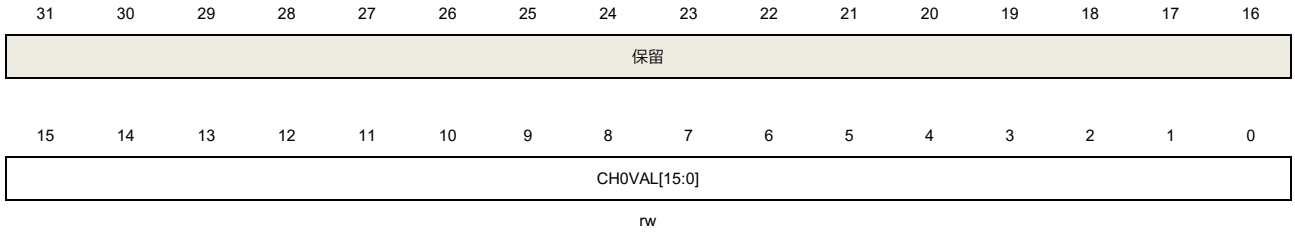


位/位域	名称	描述
31:8	保留	必须保持复位值.
7:0	CREP[7:0]	重复计数器的值 这些位定义了更新事件的产生速率。重复计数器计数值减为 0 时产生更新事件。影子寄存器的更新速率也会受这些位影响(前提是影子寄存器被使能)。

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)。

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 BKIN 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"。参见*表 9-9. 由参数控制的互补输出表*。 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"。参见*表 9-9. 由参数控制的互补输出表*。 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

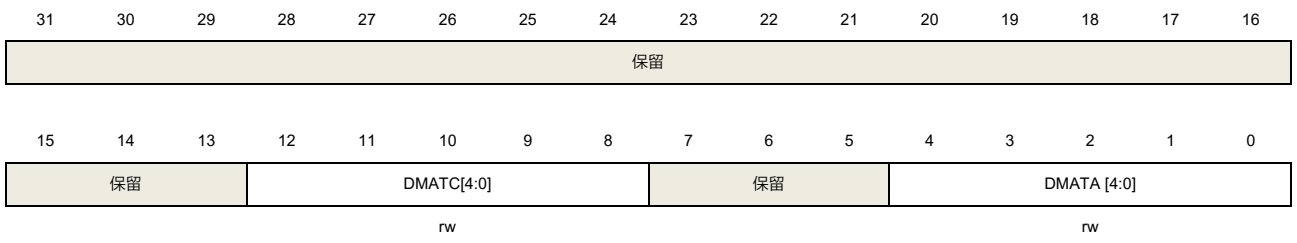
注意：

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



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
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。 5'b0_0000：TIMERx_CTL0 5'b0_0001：TIMERx_CTL1 … 总之： 起始地址 = TIMERx_CTL0 + DMATA*4

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															

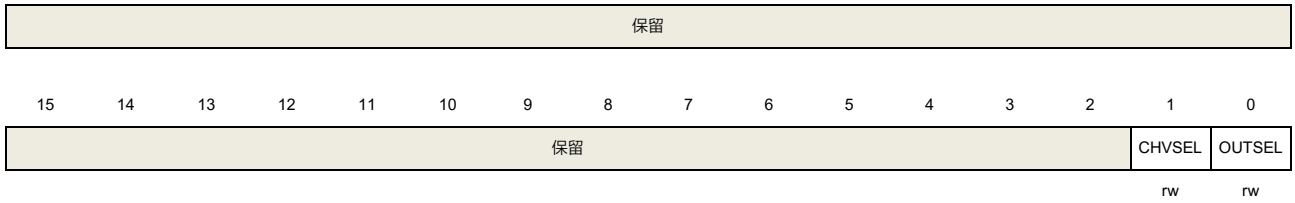
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	DMATB [15:0]	DMA 发送缓冲 对这个寄存器的读或写，(起始地址+传输次数*4)地址范围内的寄存器会被访问 传输次数由硬件计算，范围为 0 到 DMATC。

配置寄存器 (TIMERx_CFG)

地址偏移：0xFC
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
31:2	保留	必须保持复位值
1	CHVSEL	写捕获比较寄存器选择位 此位由软件写 1 或清 0。 0：无影响 1：当写入捕获比较寄存器的值与寄存器当前值相等时，写入操作无效
0	OUTSEL	输出值选择位 此位由软件写 1 或清 0。 0：无影响。 1：如果 POEN 位与 IOS 位均为 0，则输出无效

9.6. 基本定时器(TIMERx, x=5,6)

9.6.1. 简介

基本定时器(TIMER5)包含一个无符号16位计数器。基本定时器可以配置产生DMA请求。

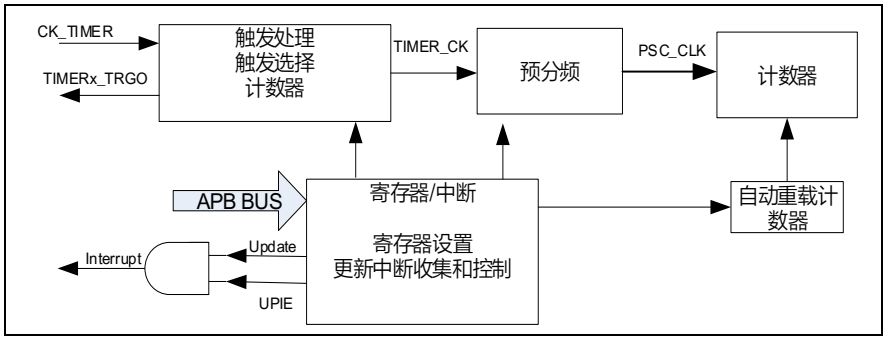
9.6.2. 主要特征

- 计数器宽度：16位；
- 定时器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。

9.6.3. 结构框图

*图 9-81. 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 9-81. 基本定时器结构框图



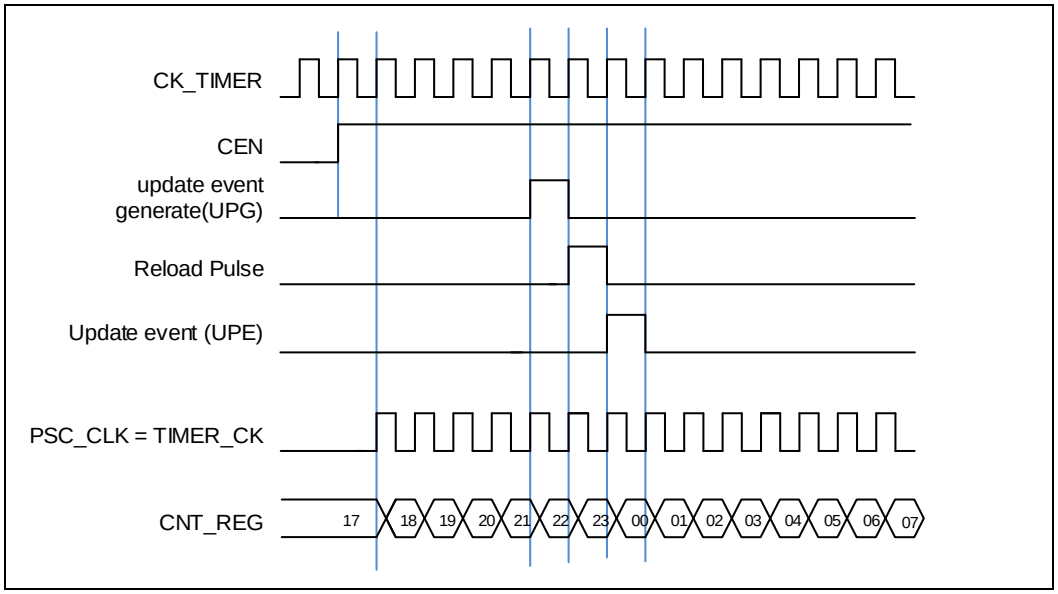
9.6.4. 功能说明

时钟源配置

基本定时器可以是内部时钟源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。

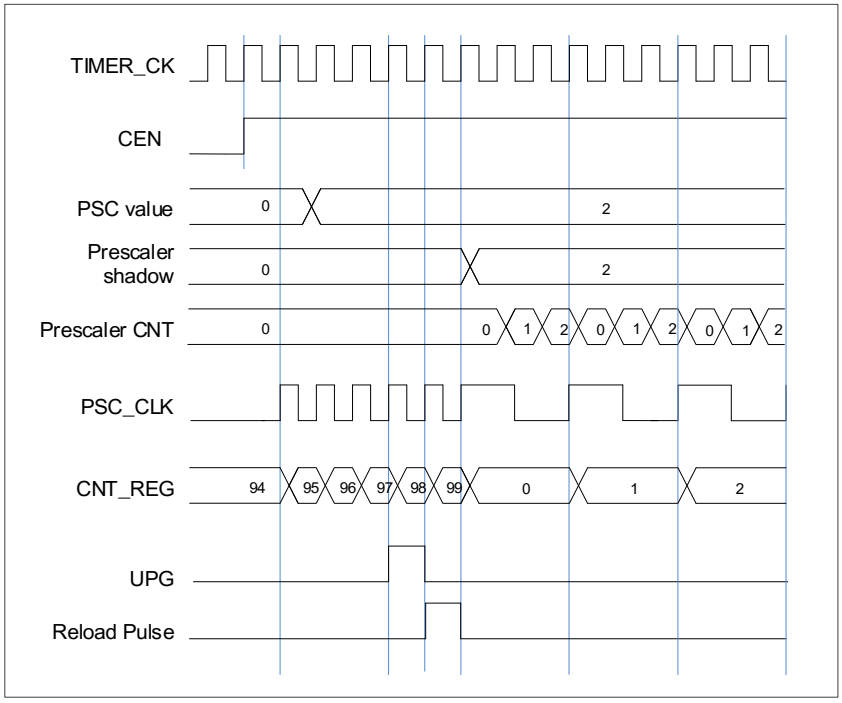
图 9-82. 内部时钟分频为 1 时，计数器的时序图



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 9-83. 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 9-84. 向上计数时序图,PSC=0/2*和*图 9-85. 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 9-84. 向上计数时序图，PSC=0/2

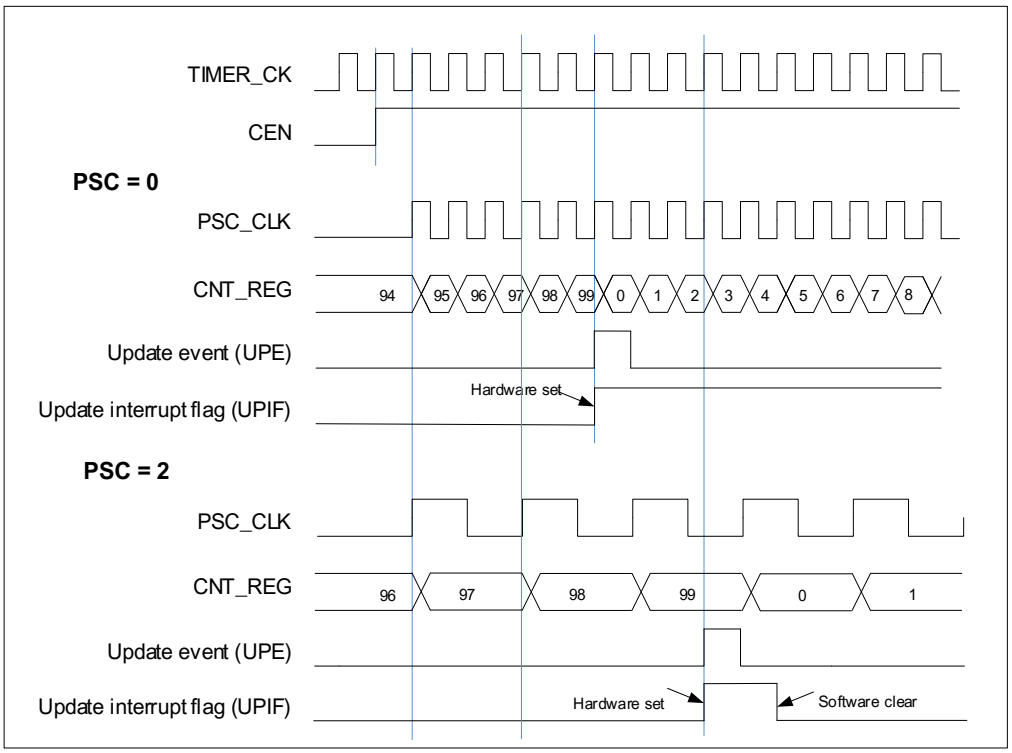
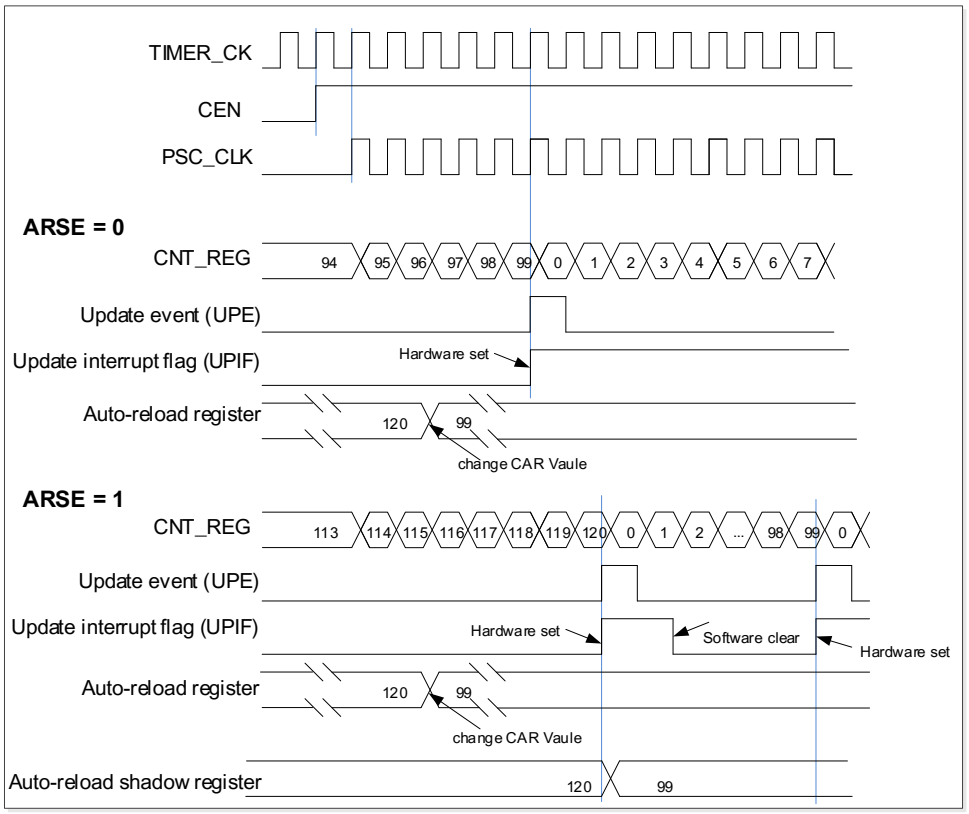


图 9-85. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值



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®-M23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。

9.6.5. TIMERx 寄存器(x=5,6)

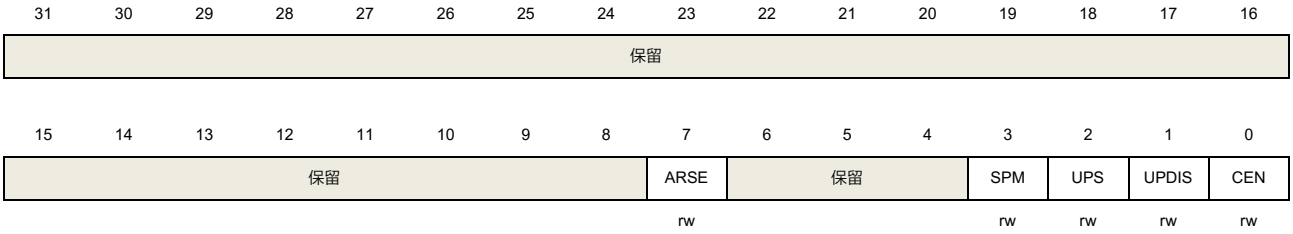
TIMER5基地址：0x4000 1000

TIMER6基地址：0x4000 1400

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

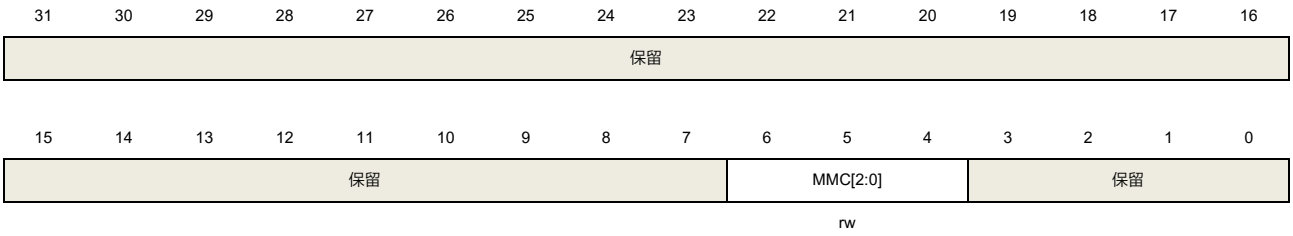
复位模式产生的更新
1：更新事件禁能.
注意：当该位被置 1 时，UPG 位被置 1 或者复位模式不会产生更新事件，但是计数器和预分频器被重新初始化

0	CEN	计数器使能 0：计数器禁能 1：计数器使能

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

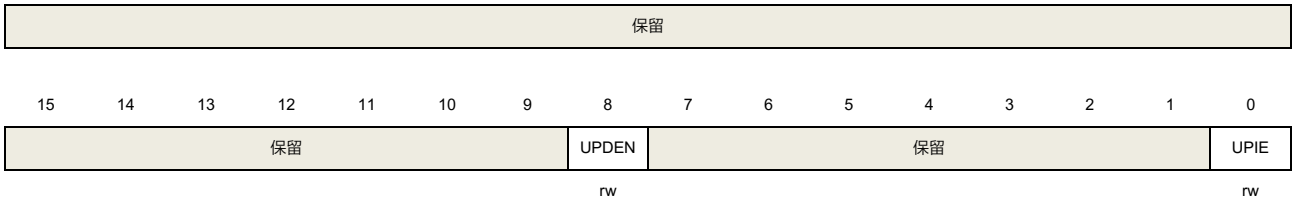


位/位域	名称	描述
31:7	保留	必须保持复位值
6:4	MMC[2:0]	这些位控制 TRGO 信号的选择，TRGO 信号由主定时器发给从定时器用于同步功能 000：当产生一个定时器复位事件后，输出一个TRGO信号，定时器复位源为： 主定时器产生一个复位事件 TIMERx_SWEVG寄存器中UPG位置1 001：当产生一个定时器使能事件后，输出一个TRGO信号，定时器使能源为： CEN位置1 在暂停模式下，触发输入置1 010：当产生一个定时器更新事件后，输出一个TRGO信号，更新事件源由UPDIS和UPS位决定
3:0	保留	必须保持复位值

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

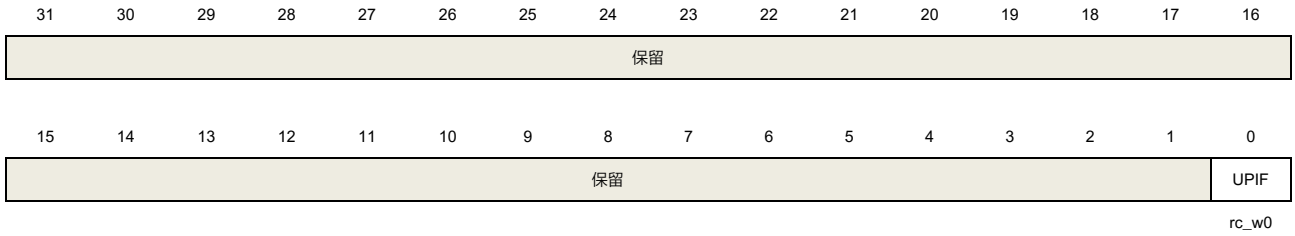


位/位域	名称	描述
31:9	保留	必须保持复位值
8	UPDEN	更新 DMA 请求使能 0：禁止更新 DMA 请求 1：使能更新 DMA 请求
7:1	保留	必须保持复位值
0	UPIE	更新中断使能 0：禁止更新中断 1：使能更新中断

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



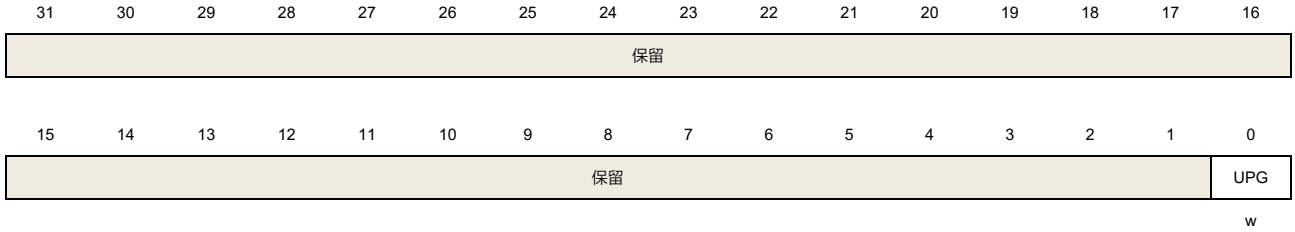
位/位域	名称	描述
31:1	保留	必须保持复位值
0	UPIF	更新中断标志 此位在更新事件发生时由硬件置 1，软件清 0。 0：无更新中断发生 1：发生更新中断

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

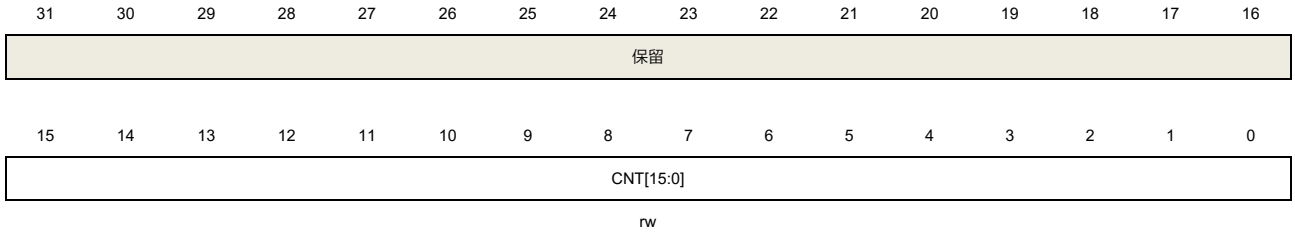


位/位域	名称	描述
31:1	保留	必须保持复位值
0	UPG	更新事件产生 此位由软件置 1，被硬件自动清 0。当此位被置 1 并且向上计数模式，计数器被清 0，预分频计数器将同时被清除。 0：无更新事件产生 1：产生更新事件

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



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。

10. 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预设的门限时，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停止计数。

10.1. 独立看门狗定时器(FWDGT)

10.1.1. 简介

独立看门狗定时器(FWDGT)有独立时钟源(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或计数器的值大于窗口寄存器的值时刷新计数器，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。

10.1.2. 主要特征

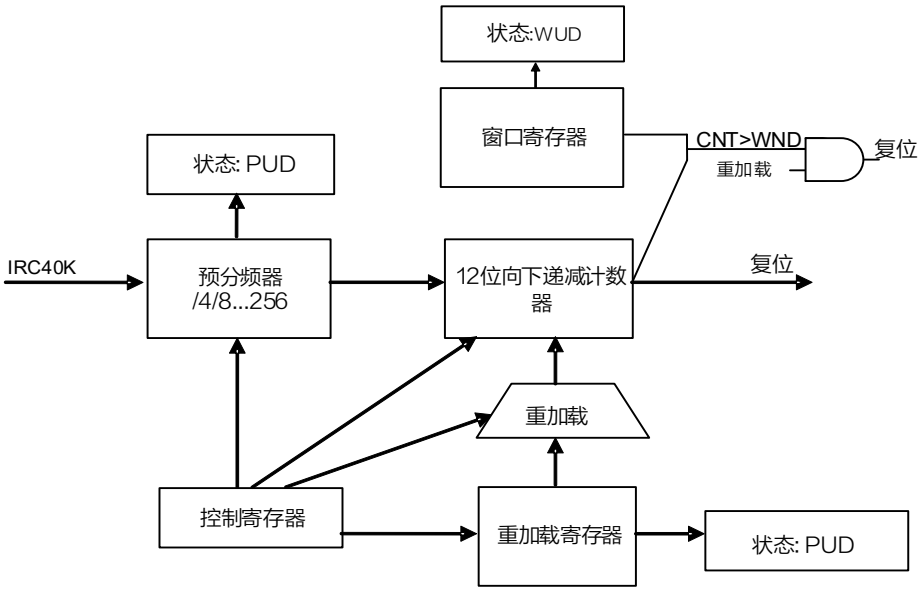
- 自由运行的12位向下计数器；
- 使能看门狗定时器，有以下两种情况下会产生系统复位：
  - 当计数器到0时产生系统复位；
  - 当计数器的值大于窗口寄存器的值时，更新计数器会产生系统复位。
- 独立时钟源，独立看门狗定时器在主时钟故障(例如待机和深度睡眠模式下)时仍能工作；
- 独立看门狗定时器硬件控制位，用来控制是否在上电时自动启动独立看门狗定时器；
- 可以配置独立看门狗定时器在调试模式下选择停止还是继续工作。

10.1.3. 功能说明

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 10-1. 独立*

*看门狗定时器框图*为独立看门狗定时器的功能模块。

图 10-1. 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。当计数器记到0x000，产生一次复位。

在任何时候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)。软件在计数器计数值达到0x000之前可以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窗口寄存器(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。

如果在选项字节中打开了"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)和FWDGT_RLD寄存器都有写保护功能。在写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中将会再次启动对这些寄存器的写保护。当FWDGT_PSC、FWDGT_WND或者FWDGT_RLD更新时，FWDGT_STAT寄存器相应的状态位被置1。

如果DBG控制寄存器0(DBG_CTL0)中的FWDGT_HOLD位被清0，即使Cortex®-M23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位被置1，独立看门狗定时器将在调试模式下停止工作。

表 10-1. 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(ms) RLD [11:0]=0x000	最大超时(ms) RLD [11:0]=0xFFF
1 / 4	000	0.1	409.6
1 / 8	001	0.2	819.2
1 / 16	010	0.4	1638.4
1 / 32	011	0.8	3276.8
1 / 64	100	1.6	6553.6
1 / 128	101	3.2	13107.2
1 / 256	110或111	6.4	26214.4

通过校准IRC40K可以使自由看门狗定时器超时更加精确。

注意：1、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。2、两次连续喂狗之间需插入7个及以上IRC32K时钟间隔。

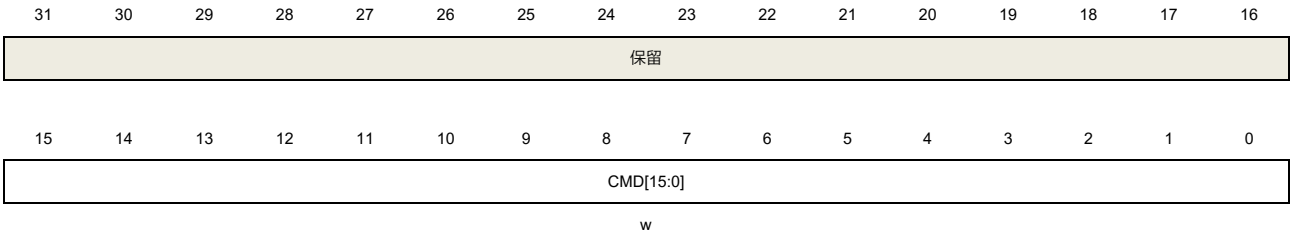
10.1.4. FWDGT 寄存器

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。

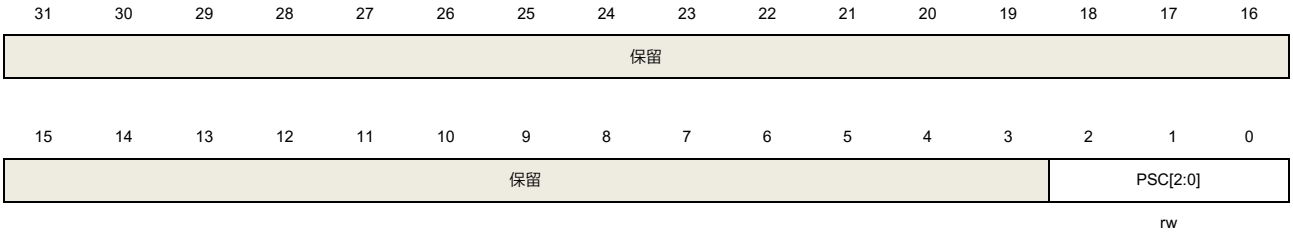


位/位域	名称	说明
31:16	保留	必须保持复位值。
15:0	CMD[15:0]	只可写，写入不同的值来产生不同的功能 0x5555：关闭FWDGT_PSC、FWDGT_RLD和FWDGT_WND的写保护 0xCCCC：开启独立看门狗定时计数器。计数减到0时产生系统复位 0xAAAA：重装载计数器。

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
如果应用需要使用几个预分频系数，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(在进入省电模式前需等待PUD值清零)。

重装载寄存器(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											

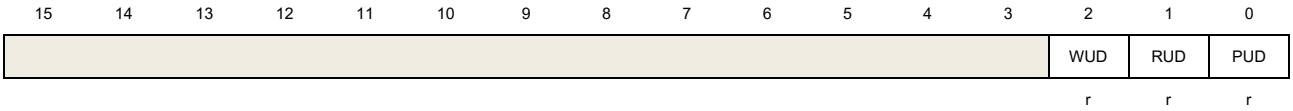
位/位域	名称	说明
31:12	保留	必须保持复位值。
11:0	RLD[11:0]	独立看门狗定时器定时计数器重装载值。向FWDGT_CTL寄存器写入0xAAAA的时候，这个值会被更新到看门狗定时器计数器中。 这些位有写保护功能。在写这些位之前需向FWDGT_CTL寄存器中写0x5555。在改写这个寄存器的过程中，FWDGT_STAT寄存器的RUD位被置1，从此寄存器中读取的任何值都是无效的。 如果应用需要使用几个重装载值，改变重加载值之前必须等到RUD位被清0。更新了重加载寄存器的值后，在代码持续执行之前不必等待RUD值被清零(在进入省电模式前需等待RUD值清零)。

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															

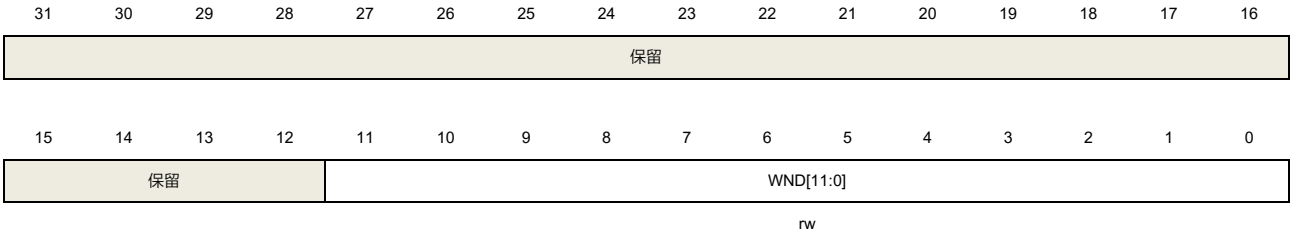


位/位域	名称	说明
31:3	保留	必须保持复位值。
2	WUD	独立看门狗定时计数器窗口值更新 FWDGT_WND寄存器写操作时，该位被置1，此时读取FWDGT_WND寄存器的任何值都是无效的。
1	RUD	独立看门狗定时器计数器重装载值更新 FWDGT_RLD寄存器写操作时，该位被置1，此时读取FWDGT_RLD寄存器的任何值都是无效的。在FWDGT_RLD寄存器更新后，该位由硬件清零。
0	PUD	独立看门狗定时器预分频值更新 FWDGT_PSC寄存器写操作时，该位被置1，此时读取FWDGT_PSC寄存器的任何值都是无效的。在FWDGT_PSC寄存器更新后，该位由硬件清零。

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。



位/位域	名称	说明
31:12	保留	必须保持复位值。
11:0	WND	独立看门狗定时器计数器窗口值。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。

10.2. 窗口看门狗定时器(WWDGT)

10.2.1. 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，14位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 = 0x3F)。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，如果使能中断位将会产生提前唤醒中断。

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。

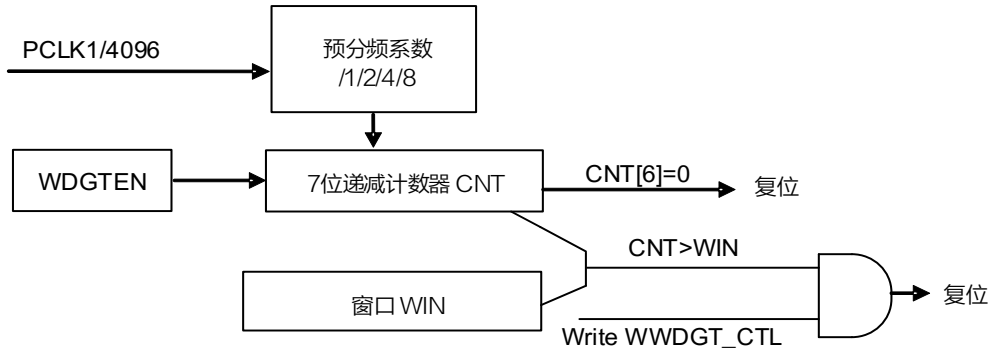
10.2.2. 主要特征

- 可编程的14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。

10.2.3. 功能说明

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 = 0x3F)或者在计数值达到窗口寄存器值之前，更新计数器也会产生系统复位。

图 10-2. 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。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)。

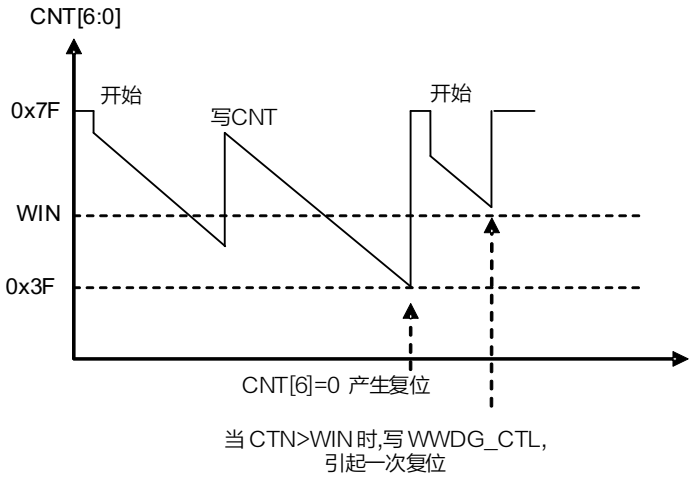
配置寄存器(WWDGT_CFG)中的WIN[13:0]位用来设定窗口值。当计数器的值小于窗口值，且

大于0x3F的时候，重装载向下计数器可以避免复位，否则在其他时候进行重加载就会引起复位。

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。

图 10-3. 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：

$$t_{WWDGT}=t_{PCLK1} \times 4096 \times 2^{PSC} \times (CNT[13:0] - 0x3F) \quad (ms) \qquad (16\text{-}1)$$

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 10-2. 在 72MHz(fPCLK1)时的最大/最小超时值*。

表 10-2. 在 72MHz(f_{PCLK1})时的最大/最小超时值

预分频系数	PSC[1:0]	最小超时 CNT[13:0] =0x40	最大超时 CNT[13:0]=0x3FFF
1 / 1	00	56.89μs	0.928s
1 / 2	01	113.78μs	1.857s
1 / 4	10	227.56μs	3.714s
1 / 8	11	455.11μs	7.427s

如果MCU调试模块中的WWDGT_HOLD位被清0，即使Cortex®-M23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工作。

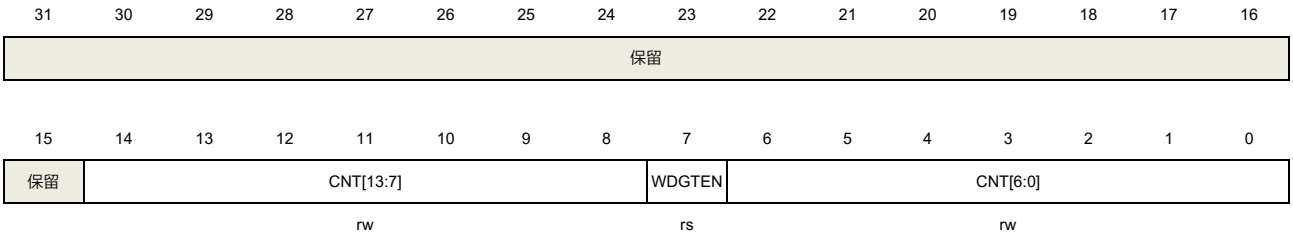
10.2.4. WWDGT 寄存器

WWDGT 基地址：0x4000 2C00

控制寄存器(WWDGT_CTL)

地址偏移：0x00
复位值：0x0000 7F7F

该寄存器可以按半字(16 位)或字(32 位)访问。

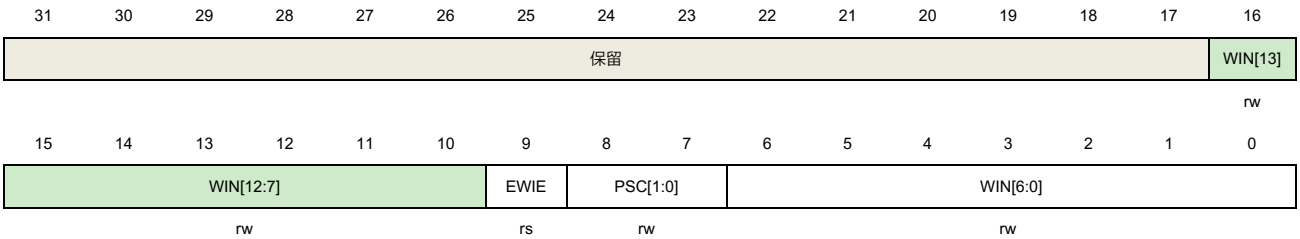


位/位域	名称	说明
31:15	保留	必须保持复位值。
14:8	CNT[13:7]	高7位看门狗定时器计数器的值 参考CNT[6:0]
7	WDGTEN	开启窗口看门狗定时器，硬件复位的时候清0，写0无效。 0：关闭窗口看门狗定时器 1：开启窗口看门狗定时器
6:0	CNT[6:0]	低7位看门狗定时器计数器的值。当计数值从0x40降到0x3F时，产生看门狗定时器复位。当计数器值高于窗口值的时候，写计数器可以产生看门狗定时器系统复位。

配置寄存器(WWDGT_CFG)

地址偏移：0x04
复位值：0x0001 FC7F

该寄存器可以按半字(16 位)或字(32 位)访问。



位/位域	名称	说明
31:17	保留	必须保持复位值。

16:10	WIN[13:7]	高7位窗口值 参考WIN[6:0]
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RCU模块的WWDGTRST软件复位来清0。写0没有任何作用。
8:7	PSC[1:0]	预分频器，看门狗定时器的时间基准。 00：(PCLK1 / 4096) / 1 01：(PCLK1 / 4096) / 2 10：(PCLK1 / 4096) / 4 11：(PCLK1 / 4096) / 8
6:0	WIN[6:0]	低7位窗口值，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。

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															EWIF
															rc_w0

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位为0)该位也会被硬件置1。这个bit可以通过写0清零，写1无效。

11. 调试(DBG)

11.1. 简介

GD32E25x 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。调试功能集成在 Arm® Cortex®-M23 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®-M23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、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、WWDGT 和 FWDGT。

11.2. SW 功能说明

调试工具可以通过串行调试接口(SWD)来访问调试功能。

11.2.1. 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)。

调试引脚分配如下:

SWCLK: PA14(GD32E252xx) / PF2(GD32E251xx)

SWDIO: PA13(GD32E252xx) / PB11(GD32E251xx)

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*通用输入/输出接口(GPIO)*。

11.3. 调试保持功能说明

11.3.1. 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。

11.3.2. TIMER，I2C，WWDGT 和 FWDGT 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 WWDGT 或者 FWDGT 外设，计数器时钟停止并进行调试。

11.4. DBG 寄存器

DBG 基地址：0x4001 5800

11.4.1. ID 寄存器(DBG_ID)

地址偏移：0x00
只读寄存器

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
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[31:0]	DBG ID 寄存器 这些位由软件读取，这些位是不变的常数。

11.4.2. 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						TIMER6_HOLD	TIMER5_HOLD	保留	I2C2_HOLD	I2C1_HOLD
				rw							rw	rw		rw	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 TIMER13 计数器不变，用于调试
26:21	保留	必须保持复位值
20	TIMER6_HOLD	TIMER6 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 TIMER6 计数器不变，用于调试
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 TIMER5 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 的 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 的 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 的 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 TIMER2 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 TIMER1 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 TIMER0 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 计数器时钟，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 计数器时钟，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 AHB 时钟由 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：在待机模式下系统时钟和 AHB 时钟由 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, AHB 时钟继续运行

11.4.3. 控制寄存器 1(DBG_CTL1)

地址偏移：0x08
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													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															

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 TIMER16 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 TIMER15 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 TIMER14 计数器不变，用于调试
15:0	保留	必须保持复位值

12. 循环冗余校验计算单元(CRC)

12.1. 简介

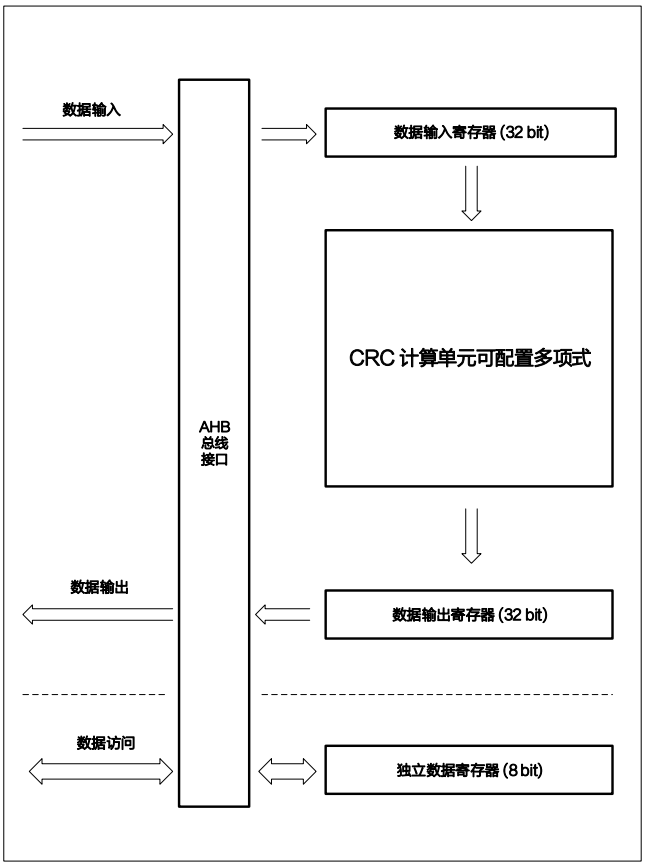
循环冗余校验码是一种用在数字网络和存储设备上的差错校验码，可以校验原始数据的偶然误差。

CRC 计算单元能用用户配置的多项式来计算 7/8/16/32 位的 CRC 校验码。

12.2. 主要特征

- 支持7/8/16/32位数据输入；
- 对于7(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；
- 用户可以配置多项式及多项式长度；
- 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。

图 12-1. CRC 计算单元框图



12.3. 功能说明

- CRC计算单元可以用来计算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(7)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0x1A2B3C4D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：0x58D43CB2

  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：0xD458B23C

  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：0xB23CD458

  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0x22CC4488将被逆序成0x11223344。

- 用户可配置的初始计算数据。

  当RST位置位或对CRC_IDATA寄存器进行写操作时，CRC_DATA寄存器将自动初始化为CRC_IDATA寄存器中的值。

- 用户配置多项式。

  通过配置PS[1:0]，用户可以选择有效多项式和输出数据位宽。如果多项式少于32位，那么输入和输出数据的高位无效。当PS[1:0]或多项式改变后，需要复位CRC。

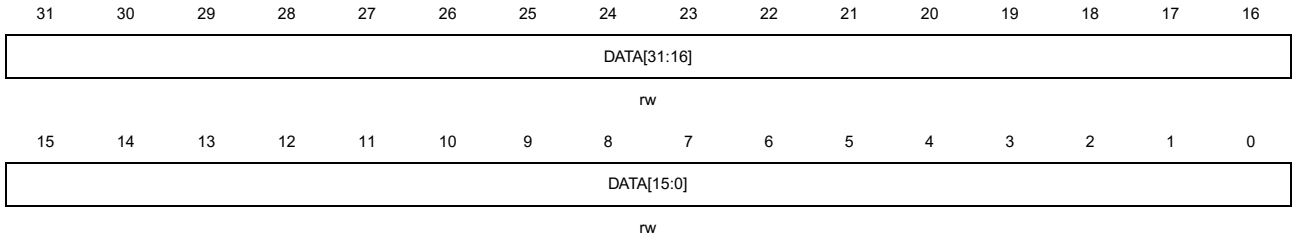
12.4. CRC 寄存器

CRC基地址：0x4002 3000

12.4.1. 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。

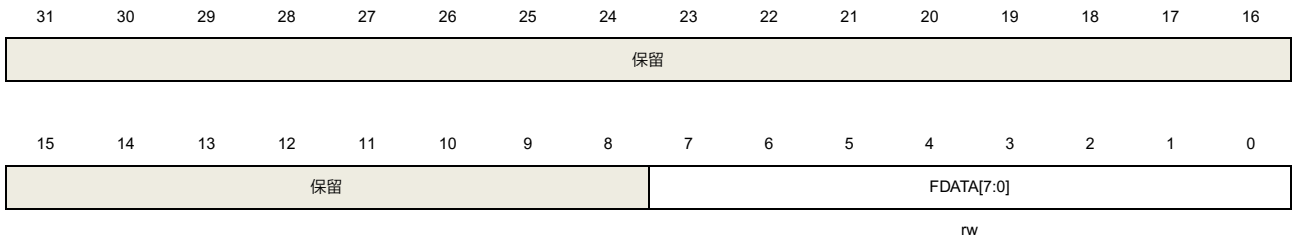


位/位域	名称	描述
31:0	DATA[31:0]	CRC 计算结果位 软件可读可写。 该寄存器用于接收待计算的新数据，直接将其写入即可。刚写入的数据不能被读出来因为读取该寄存器得到的是上次 CRC 计算的结果。

12.4.2. 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。

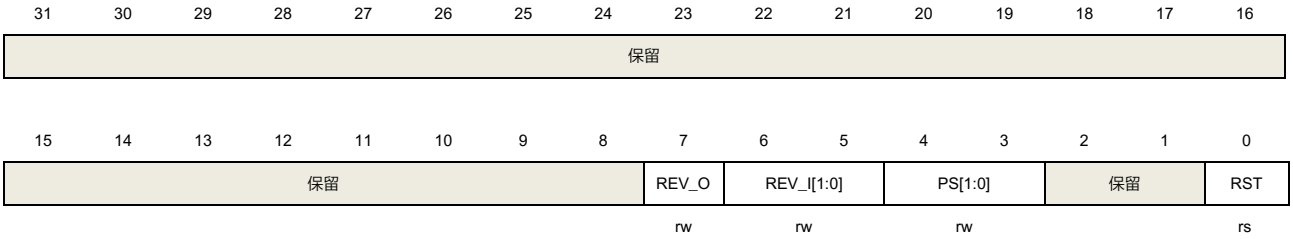


位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	FDATA[7:0]	独立数据寄存器位 软件可读可写。 这些位与 CRC 计算无关。该字节能被任何其他外设用于其他任何目的。该字节不受 CRC_CTL 寄存器的影响。

12.4.3. 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。

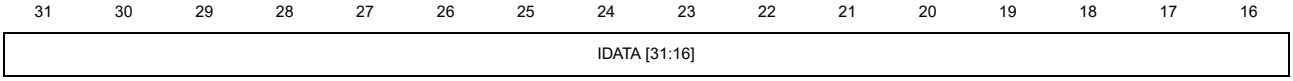


位/位域	名称	描述
31:8	保留	必须保持复位值。
7	REV_O	按位顺序翻转输出数据功能 0：输出数据不翻转 1：输出数据按位顺序翻转
6:5	REV_I[1:0]	翻转输入数据功能 0：输入数据不翻转 1：输入数据按字节翻转 2：输入数据按半字翻转 3：输入数据按字翻转
4:3	PS[1:0]	多项式长度 0：32 位 1：16(POLY[15:0]用于计数)位 2：8(POLY[7:0]用于计数)位 3：7(POLY[6:0]用于计数)位
2:1	保留	必须保持复位值。
0	RST	软件可读写 该位用来复位 CRC_DATA 寄存器。 置位时，CRC_DATA 寄存器的值将自动初始化为 CRC_IDATA 寄存器中的值，然后自动清零。该位对 CRC_FDATA 寄存器没有影响。

12.4.4. 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。



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

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。

12.4.5. 多项式寄存器(CRC_POLY)

地址偏移：0x14
复位值：0x04C1 1DB7

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
POLY [31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POLY[15:0]															
rw															

位/位域	名称	描述
31:0	POLY[31:0]	配置多项式值 配合 PS[1:0]使用。

13. 模数转换器(ADC)

13.1. 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20个外部通道、3个内部通道的模拟信号。这23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。片上的硬件过采样机制可以通过减少MCU的相关计算负担来提高性能。

13.2. 主要特征

- 高性能：
  - ADC 采样分辨率：12 位、10 位、8 位或 6 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：
  - 20 个外部模拟输入通道；
  - 1 个内部高精度温度传感通道(V_{SENSE})；
  - 1 个内部参考电压输入通道(V_{REFINT})；
  - 1 个模拟电源电压的 1/2 输入通道($V_{DDADIV2}$)。
- 转换开始的发起：
  - 软件方式；
  - 硬件触发。
- 运行模式：
  - 转换单个通道，或者扫描一组通道；
  - 单次运行模式，每次触发转换一次选择的输入通道；
  - 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；
- 过采样：
  - 16 位的数据寄存器；
  - 可调整的过采样率，范围从 2x 到 256x；
  - 高达 8 位的可编程数据移位。
- 通道输入范围：$V_{SSA} <= V_{IN} <= V_{REFP}$

13.3. 引脚和内部信号

*图 13-1. ADC 模块框图*给出了 ADC 模块框图。*表 13-1. ADC 内部输入信号*和*表 13-2. ADC 输入引脚定义* 给出了 ADC 内部信号和引脚定义。

表 13-1. ADC 内部输入信号

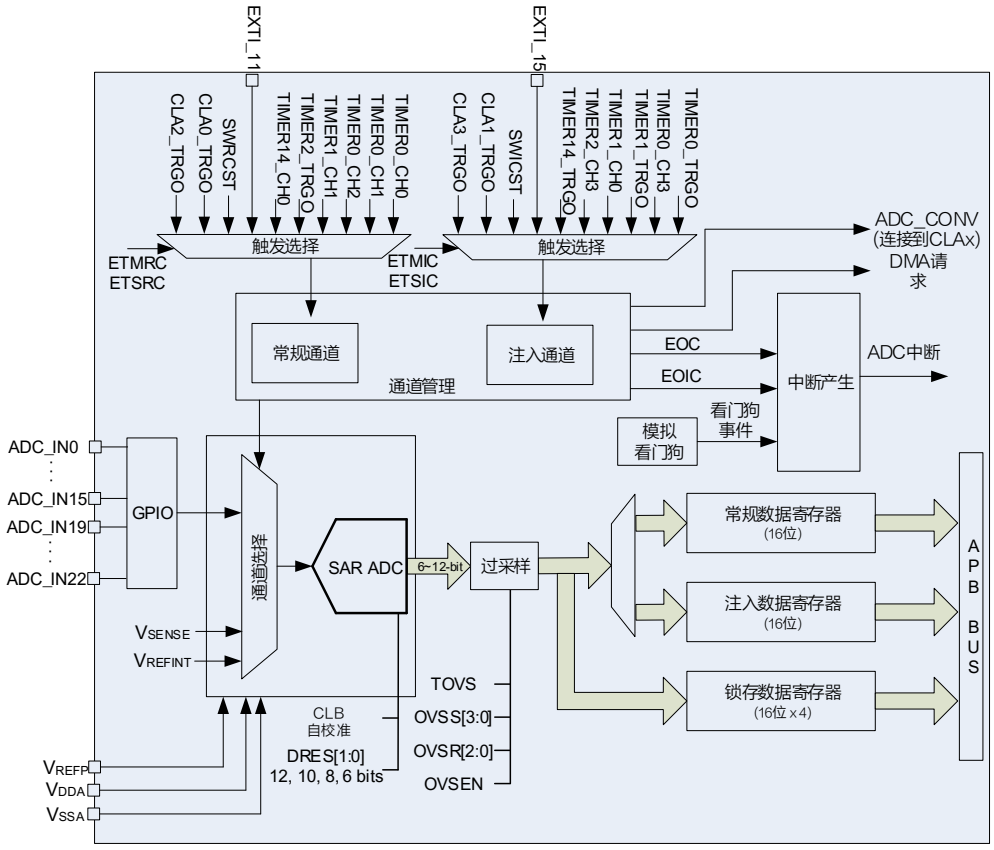
内部信号名称	说明
V_{SENSE}	内部温度传感器电压输出
V_{REFINT}	内部 1.2V 参考电压输出
$V_{DDADIV2}$	模拟电源电压的 1/2

表 13-2. ADC 输入引脚定义

名称	说明
V_{DDA}	模拟电源输入等于V_{DD}
V_{REFP}	ADC 参考电压输入
V_{SSA}	模拟地，等于 V_{SS}
ADCx_IN [15:0] ADCx_IN [22:19]	多达 20 路外部通道

13.4. 功能说明

图 13-1. ADC 模块框图



13.4.1. 前置校准功能

在前置校准期间，ADC计算一个校准因子，这个因子应用于ADC的内部，它直到ADC下次掉电才无效。在校准期间，应用程序不能使用ADC，必须等到校准完成。在开始A/D转换前应执行校准操作。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(例如供电电压V_{DDA}、温度等)时，建议重新执行一次校准操作。

内部的模拟校准可以通过设置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(该步骤是可选的)；
4. 设置 CLB = 1；
5. 等待直到 CLB = 0。

13.4.2. 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，同时ADC仍能保持最佳运行状态。

13.4.3. ADC 使能

ADC_CTL1寄存器中的ADCON位是ADC模块的使能开关。如果该位为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使能后需等待t_{US}时间后才能采样，t_{US}数值详见芯片数据手册。

13.4.4. 常规序列和注入序列

ADC支持23个多路通道，可以把采样转换通道分成两个序列：一个常规序列和一个注入序列。

常规序列支持高达23个通道，每个通道称为常规通道。ADC_RSQ0寄存器的RL[4:0]位规定了整个常规序列转换序列的长度。ADC_RSQ0~ADC_RSQ3寄存器规定了常规序列的通道选择。

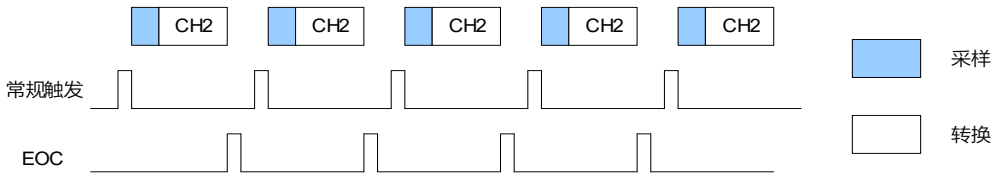
注入序列支持高达4个通道，每个通道称为注入通道。ADC_ISQ寄存器的IL[1:0]位规定了整个注入序列的长度。ADC_ISQ寄存器规定了注入序列的通道选择。

13.4.5. 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]位规定了ADC的转换通道。当ADCON位被置1时，一旦相应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 13-2. 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将会置1。如果EOCIE位被置1，将产生一个中断。

注入通道单次转换结束后，转换数据将被存放于ADC_IDATA寄存器中，EOC位和EOIC位将会置1。如果EOCIE或EOICIE位被置1，将产生一个中断。

常规组单次运行模式的软件流程：

1. 确保ADC_CTL0寄存器的DISRC位和SM位以及ADC_CTL1寄存器中的CTN位为0；
2. 用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ETMRC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. 等到EOC位置1；
7. 从ADC_RDATA寄存器中读ADC转换结果；
8. 写0清除EOC标志位。

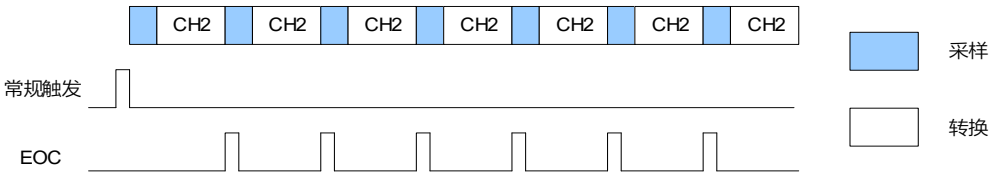
注入序列单次运行模式的软件流程：

1. 确保ADC_CTL0寄存器的DISRC和SM位为0；
2. 用模拟通道编号来配置ISQ3位域；
3. 配置ADC_SAMPTx寄存器；
4. 如果有需要，可以配置ADC_CTL1寄存器的ETMIC和ETSIC位域；
5. 设置SWICST位，或者为注入序列产生一个外部触发信号；
6. 等到EOC位、EOIC位置1；
7. 从ADC_IDATA寄存器中读ADC转换结果；
8. 写0清除EOC、EOIC标志位。

连续运行模式

对ADC_CTL1寄存器中的CTN位置1，可以使能连续运行模式。在此模式下，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 13-3. 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ETMRC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. 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。

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、RDMA和RDDM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ETMRC位和ETSRC位；

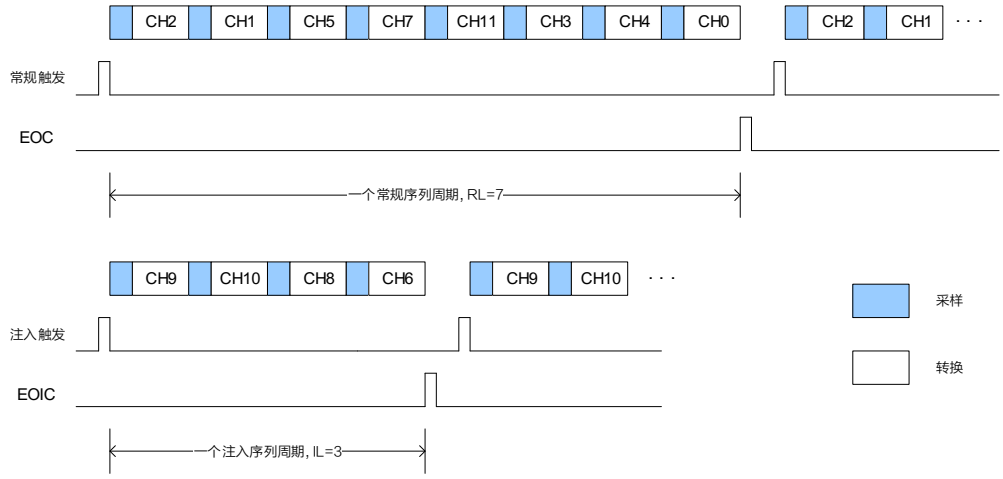
5. 准备DMA模块，用于传输来自ADC_RDATA寄存器的数据；
6. 设置SWRCST位，或者给常规序列产生一个外部触发。

扫描运行模式

扫描转换模式可以通过将ADC_CTL0寄存器的SM位置1来使能。在此模式下，ADC扫描转换所有被ADC_RSQ0~ADC_RSQ3寄存器或ADC_ISQ寄存器选中的所有通道。一旦ADCON位被置1，当相应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或注入序列通道。转换数据存储在ADC_RDATA或ADC_IDATA寄存器中。常规序列或注入序列转换结束后，EOC或者EOIC位将被置1。如果EOCIE或EOICIE位被置1，将产生中断。当常规序列或者注入序列工作在扫描模式下时，ADC_CTL1寄存器的RDMA或者IDMA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，转换自动重新开始。

图 13-4. 扫描运行模式，且连续运行模式失能



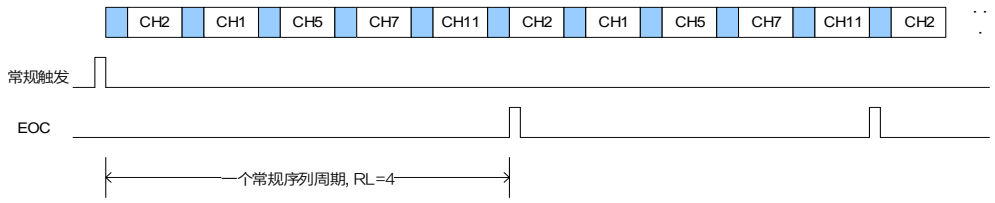
常规序列扫描运行模式的软件流程：

1. 设置 ADC_CTL0 寄存器的 SM 位和 ADC_CTL1 寄存器的 RDMA 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 ETMRC 和 ETSRC 位；
4. 准备 DMA 模块，用于传输来自 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；

注入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 IDMA 位为 1；
2. 配置 ADC_ISQ 和 ADC_SAMPTx 寄存器；
3. 如果有需要，配置 ADC_CTL1 寄存器中的 ETMIC 和 ETSIC 位域；
4. 准备 DMA 模块，用于传输来自 ADC_IDATA 的数据；
5. 设置 SWICST 位，或者给注入序列产生一个外部触发；

图 13-5. 扫描运行模式，连续运行模式使能



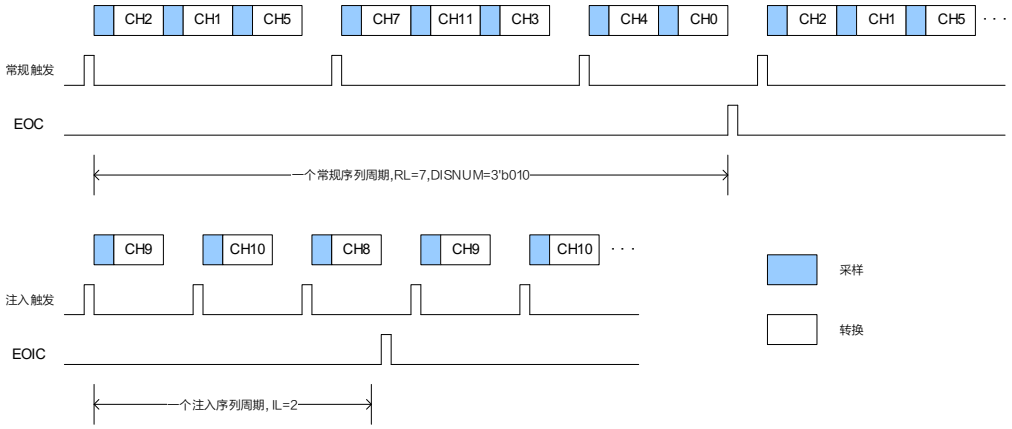
间断运行模式

对于常规序列，当 ADC_CTL0 寄存器的 DISRC 位置 1 时，常规序列间断转换模式使能。该模式下可以执行一次 n 个通道的短序列转换(n<=8)，这个短序列是 ADC_RSQ0~ADC_RSQ3 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]位给出。当相应的软件触发或外部触发发生，ADC 就会采样和转换在 ADC_RSQ0~ADC_RSQ3 寄存器所选择通道中接下来的 n 个通道，直到常规序列中所有的通道转换完成。每个常规序列转换周期结束后，EOC 位将置 1。如果 EOCIE 位置 1 将产生一个中断。

对于注入序列，当 ADC_CTL0 寄存器的 DISIC 位置 1 时，注入序列间断转换模式使能。该模式下可以执行 ADC_ISQ 寄存器所选择的转换序列的一个通道进行转换。当相应的软件触发或外部触发发生，ADC 就会采样和转换 ADC_ISQ 寄存器中所选择通道的下一个通道，直到注入序列中所有通道转换完成。每个注入序列转换周期结束后，EOIC 位将被置 1。如果 EOICIE 位被置 1 将产生一个中断。

常规序列和注入序列不能同时工作在间断模式，同一时刻只能有一个序列被设置成间断模式。

图 13-6. 间断运行模式



常规序列间断模式的软件流程：

1. 设置 ADC_CTL0 寄存器的 DISRC 位和 ADC_CTL1 寄存器的 RDMA 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 ETMRC 和 ETSRC 位域；
5. 准备 DMA 模块，用于传输来自 ADC_RDATA 的数据；
6. 设置 SWRCST 位，或者给常规序列产生一个外部触发；
7. 如果需要，重复步骤 6。

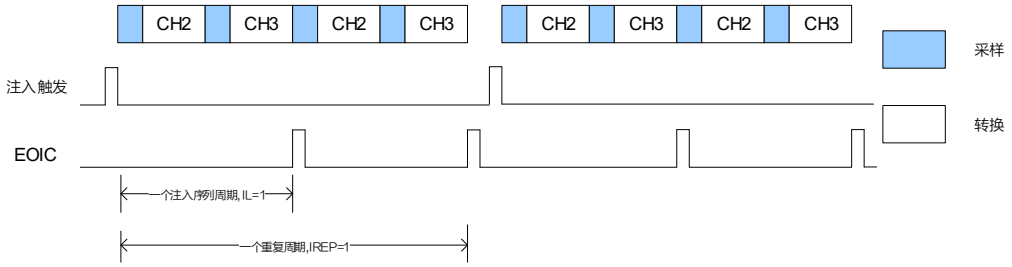
注入序列间断模式的软件流程：

1. 设置 ADC_CTL0 寄存器的 DISRC 位和 ADC_CTL1 寄存器的 IDMA 位为 1；
2. 配置 ADC_ISQ 和 ADC_SAMPTx 寄存器；
3. 如果有需要，配置 ADC_CTL1 寄存器中的 ETMIC 和 ETSIC 位域；
4. 准备 DMA 模块，用于传输来自 ADC_IDATA 的数据；
5. 设置 SWICST 位，或者给注入序列产生一个外部触发；
6. 如果需要，重复步骤 5。

重复运行模式

该模式仅运行在注入序列。该模式需与单次运行模式或者扫描模式配合使用。通过配置 ADC_CTL2 寄存器的 IREP[5:0]位域，在单次的软件触发或者外部触发产生时，ADC 的注入序列可以重复采样和转换 IREP[5:0]+1 次。

图 13-7. 重复运行模式(SM=1)



注入序列重复运行模式的软件流程：

1. 设置 ADC_CTL0 寄存器的 SM 位和 ADC_CTL2 寄存器的 IDMA 位为 1；
2. 配置 ADC_CTL2 寄存器的 IREP[5:0]位域；
3. 配置 ADC_ISQ 和 ADC_SAMPTx 寄存器；
4. 如果有需要，配置 ADC_CTL1 寄存器中的 ETMIC 和 ETSIC 位域；
5. 准备 DMA 模块，用于传输来自 ADC_IDATA 的数据；
6. 设置 SWICST 位，或者给注入序列产生一个外部触发；

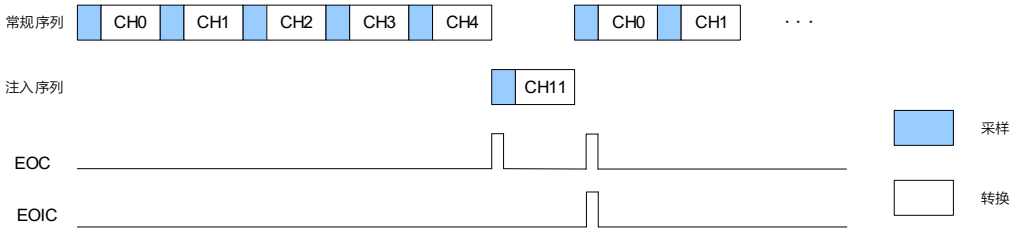
注意：当注入序列的间断模式使能时，IREP[5:0]位域必须保持复位值。

13.4.6. 注入序列管理

自动注入

如果将 ADC_CTL0 寄存器的 ICA 位置 1，在常规序列之后，注入序列被自动转换。该模式下注入序列的外部触发不能被使能。该模式可以转换 ADC_RSQ0~ADC_RSQ2 和 ADC_ISQ 寄存器中设置的多至 16 个通道。除了 ICA 位之外，如果 CTN 位也被置 1，将连续依次转换常规序列和注入序列。

图 13-8. 自动注入，CNT=1

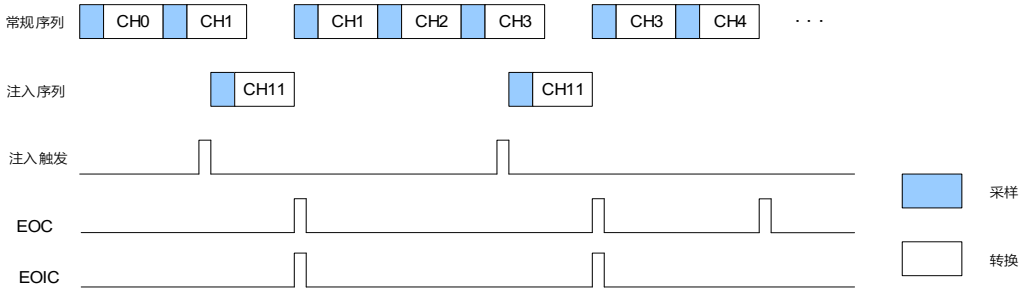


不能同时使用自动注入和间断模式。

触发注入

清除 ICA 位，在常规序列转换期间如果注入序列的软件触发或者外部触发发生，则启动注入序列转换。这种情况下，ADC 取消常规序列中当前正在转换的通道，注入序列进行转换。注入序列转换结束后，ADC 从上次被取消的常规序列通道处重新开始转换。

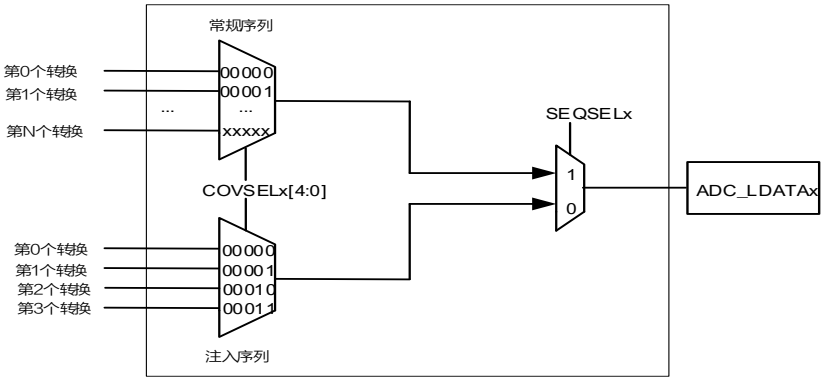
图 13-9. 触发注入



13.4.7. 转换数据锁存

ADC 拥有 4 个锁存数据寄存器，ADC_LDATAx(x=0…3)，这些寄存器可以在常规或注入序列通道转换完成后锁存数据。锁存数据寄存器的控制逻辑如*图 13-10. 数据锁存*。

图 13-10. 数据锁存

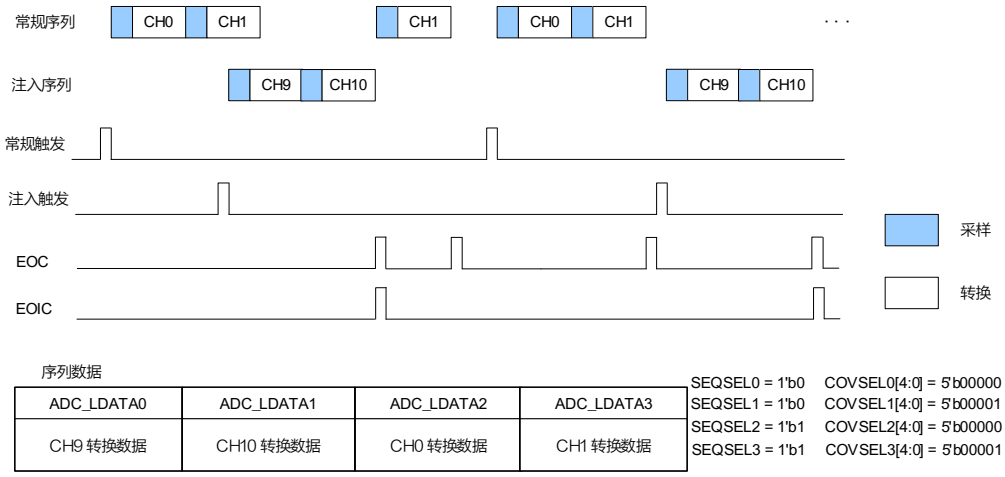


是从常规序列还是从注入序列中选择要锁存的数据，可以通过 ADC_LDCTL 寄存器中的 SEQSELx(x=0…3)位来决定，而 COVSELx[4:0]选择要锁存的序列中的哪个转换结果。

设置 ADC_LDATAx(x=0…3)寄存器来存储注入序列或常规序列的转换结果，默认存储注入序列的第 x 次转换结果。

因此，除了使用 DMA 从 ADC_RDATA 或 ADC_IDATA 寄存器传输转换数据外，还可以直接从锁存数据寄存器读取数据，这为序列的连续转换提供了更大的灵活性。

图 13-11. ADC_LDATAx 读取序列数据



触发注入模式下从 ADC_LDATAx 读取序列数据的软件流程：

1. 设置 ADC_CTL0 寄存器的 SM 位；
2. 配置 ADC_RSQx，ADC_ISQ 和 ADC_SAMPTx 寄存器；
3. 配置 ADC_LDCTL 寄存器的 SEQSEL2，COVSEL2[4:0]，SEQSEL3 和 COVSEL3[4:0] 位域用于常规序列；
4. 配置 ADC_LDCTL 寄存器的 SEQSEL0，COVSEL0[4:0]，SEQSEL1 和 COVSEL1[4:0] 位域用于注入序列；
5. 如果有需要，配置 ADC_CTL1 寄存器中的 ETMRC，ETSRC，ETMIC 和 ETSIC 位域；
6. 设置 SWRCST 位，或者给常规序列产生一个外部触发；
7. 设置 SWICST 位，或者给注入序列产生一个外部触发；
8. 等待 EOC 或 EOIC 标志位置 1；
9. 如果 EOC 标志位置 1，EOIC 没有置位，从 ADC_LDATA2 和 ADC_LDATA3 寄存器中读常规序列 ADC 转换结果；
10. 写 0 清除 EOC 标志位；
11. 如果 EOIC 标志位置 1，从 ADC_LDATA1 和 ADC_LDATA2 寄存器中读注入序列 ADC 转换结果；
12. 写 0 清除 EOIC、EOC 标志位。

Note: 若采用查询 EOC 标志位的方式读取常规序列转换结果，需要注意注入序列转换结束也会置起该标志位。

13.4.8. 转换结果阈值监测功能

ADC_CTL0 寄存器的 RWDEN 和 IWDEN 位置 1 将分别使能常规序列和注入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 的模拟转换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 寄存器中的 DAL 位确定的对齐方式无关。ADC_CTL0 寄存器的 RWDEN，IWDEN，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通道。

13.4.9. 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 13-12. 12 位数据存储模式*，*图 13-13. 10 位数据存储模式*，*图 13-14. 8 位数据存储模式*和*图 13-15. 6 位数据存储模式*所示。

图 13-12. 12 位数据存储模式

常规通道数据

0	0	0	0	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

注入通道数据

Sign	Sign	Sign	Sign	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

DAL=0

常规通道数据

D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0

注入通道数据

Sign	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0

DAL=1

图 13-13. 10 位数据存储模式

常规通道数据

0	0	0	0	0	0	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

注入通道数据

Sign	Sign	Sign	Sign	Sign	Sign	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

DAL=0

常规通道数据

D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0	0	0

注入通道数据

Sign	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0	0

DAL=1

图 13-14. 8 位数据存储模式

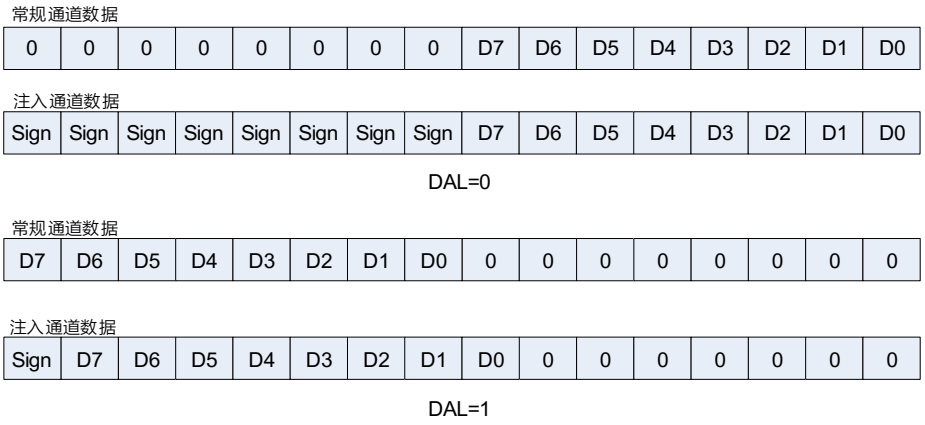
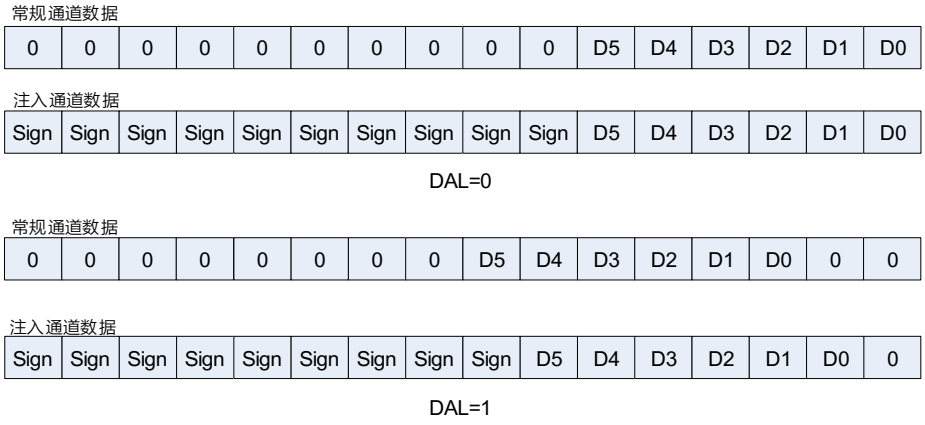


图 13-15. 6 位数据存储模式



13.4.10. 采样时间配置

ADC 使用多个 CK_ADC 周期对输入电压采样，采样周期数目可以通过 ADC_SAMPT0、ADC_SAMPT1 和 ADC_SAMPT2 寄存器的 SPTn[2:0]位设置。每个通道可以使用不同的时间采样。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 = 28MHz，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 0.500us。

13.4.11. 外部触发

外部触发输入的上升沿可以触发常规序列或注入序列的转换。常规序列的外部触发源由 ADC_CTL1 寄存器的 ETSRC[3:0]位控制，注入序列的外部触发源由 ADC_CTL1 寄存器的 ETSIC[3:0]位控制。

表 13-3. 用于 ADC 规则通道的外部触发

ETSRC[3:0]	触发源	触发类型
0000	TIMER0_CH0	来自片上定时器的内部信号

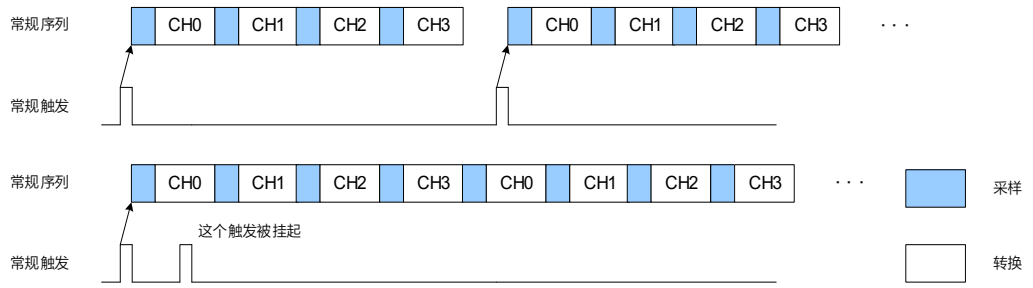
ETSRC[3:0]	触发源	触发类型
0001	TIMER0_CH1	
0010	TIMER0_CH2	
0011	TIMER1_CH1	
0100	TIMER2_TRGO	
0101	TIMER14_CH0	
0110	EXTI_11	外部信号
0111	SWRCST	软件触发
1000	CLA0_TRGO	来自CLA片上内部信号
1001	CLA2_TRGO	

表 13-4. 用于 ADC 注入通道的外部触发

ETSIC[3:0]	触发源	触发类型
0000	TIMER0_TRGO	来自片上定时器的内部信号
0001	TIMER0_CH3	
0010	TIMER1_TRGO	
0011	TIMER0_CH0	
0100	TIMER2_CH3	
0101	TIMER14_TRGO	
0110	EXTI_15	外部信号
0111	SWICST	软件触发
1000	CLA1_TRGO	来自CLA片上内部信号
1001	CLA3_TRGO	

在常规序列转换期间，若发生新的常规序列触发请求，该触发请求将被挂起，待当前转换完成后再执行挂起的触发请求，如*图 13-16. 在常规序列转换过程中常规触发出现*所示：

图 13-16. 在常规序列转换过程中常规触发出现



在注入序列转换期间，若发生新的注入序列或常规序列触发请求，这些触发请求将被挂起，待当前转换完成后再执行挂起的触发请求。当注入序列和常规序列的触发请求均被挂起时，ADC 先执行注入序列转换，注入序列转换完成后，再执行常规序列转换。

每个序列(注入序列或常规序列)的触发请求最多只能挂起一个，后续产生的触发请求将被忽略。

13.4.12. DMA 请求

DMA请求，可以通过设置ADC_CTL1寄存器的RDMA或者ADC_CTL2寄存器的IDMA位来使能，它用于传输常规或者注入序列多个通道的转换结果。ADC在常规或者注入序列的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或者ADC_IDATA寄存器传输到用户指定的目的地址。

13.4.13. ADC 内部通道

将 ADC_CTL1 寄存器的 HPTSEN 位置 1，可以使能高精度温度传感器通道(ADC_IN16)。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。建议温度传感器的采样时间至少设置为 t_{s_temp}(具体数值请参考 datasheet 文档)。温度传感器不用时，复位 HPTSEN 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。

使用高精度温度传感器：

1. 配置 ADC 时钟(不超过 15MHz)；
2. 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；
3. 置位 ADC_CTL1 寄存器中的 HPTSEN 位，使能温度传感器；
4. 置位 ADC_CTL1 寄存器的 ADCON 位，或者由外部触发 ADC 转换；
5. 从 ADC 数据寄存器中读取并计算温度传感器数据 $V_{temperature}$，并由下面公式计算出实际温度：

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

$V_{temperature}$：当前温度传感器采样得到的实际温度码值

V_{25}：内部温度传感器在 25°C 时的输出电压。芯片出厂时记录了温度传感器在 25°C 下对应的 ADC 转换结果。这个出厂校准值存储在 FLASH 中的只读区域，具体存储地址请见数据手册。

Avg_Slope：温度与内部温度传感器电压曲线的均值斜率，典型值请参考相关型号 datasheet。

注意：

1) 当高精度温度传感器使能，至少需要等待 3 个 ADC 采样周期，前三个转换数据应当被舍弃；
2) 可以通过过采样和软件平均提高高精度温度传感器准确度。

将 ADC_CTL1 寄存器的 VDDADIV2EN 位置 1 可以使能 $V_{DDADIV2}$ 通道(ADC_IN18)。$V_{DDADIV2}$ 通道内部连接到模拟电源电压 V_{DDA}/2。

将 ADC_CTL1 寄存器的 INREFEN 位置 1 可以使能 V_{REFINT} 通道(ADC_IN17)。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 输入通道。

13.4.14. 可编程分辨率(DRES)

ADC 分辨率可以通过寄存器 ADC_CTL0 中的 DRES[1:0]位进行配置。对于那些不需要高精度数据的应用，可以使用较低的分辨率来实现更快速地转换。只有在 ADCON 位为 0 时，才能修改 DRES[1:0]的值。较低的分辨率能够减少转换时间，如*表 13-5. 不同分辨率对应的 tCONV 时间*所示，较低的分辨率能够减少逐次逼近步骤所需的转换时间 t_{ADC}。

表 13-5. 不同分辨率对应的 t_{CONV} 时间

DRES [1:0] bits	t_{CONV} (ADC 时钟周期)	t_{CONV} (ns) (f_{ADC}=28MHz)	t_{SMPL}(ADC 时钟周期)	t_{ADC}(ADC 时钟周期)	t_{ADC}(ns) (f_{ADC}=28MHz)
12	12.5	446ns	1.5	14	500ns
10	10.5	375ns	1.5	12	429ns
8	8.5	304ns	1.5	10	357ns
6	6.5	232ns	1.5	8	286ns

13.4.15. 片上硬件过采样

片上硬件过采样单元执行数据预处理以减轻 CPU 负担。它能够处理多个转换，并将多个转换的结果取平均，得出一个 16 位宽的数据。

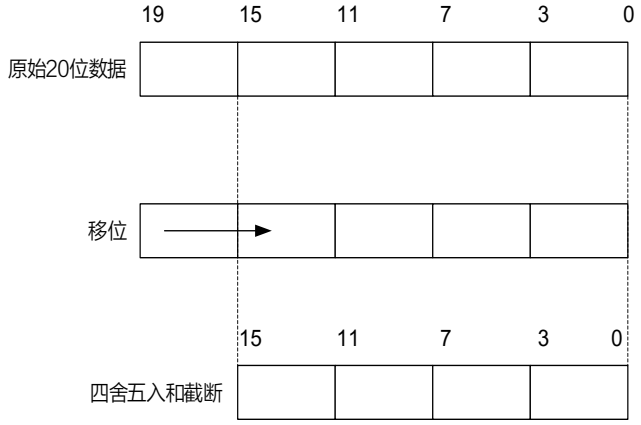
其结果根据如下公式计算得出，其中，N 和 M 的值可以被调整，过采样单元可以通过设置 ADC_OVSAMPCTL 寄存器中的 OVSEN 位来使能，它是以降低数据输出率为代价，换取较高的数据分辨率。$D_{out}(n)$是指 ADC 输出的第 n 个数字信号：

$$\text{Result} = \frac{1}{M} * \sum_{n=0}^{n=N-1} D_{OUT}(n) \quad (13\text{-}1)$$

片上硬件过采样单元执行两个功能：求和和位右移。过采样率 N 是在 ADC_OVSAMPCTL 寄存器的 OVSR[2:0]位定义，它的取值范围为 2x 到 256x。除法系数 M 定义了一个多达 8 位的右移，它通过 ADC_OVSAMPCTL 寄存器 OVSS[3:0]位进行配置。

求和单元能够生成一个多达 20 位(256*12 位)的值。首先，将这个值进行右移，将移位后剩余的部分再通过取整转化一个近似值，最后将高位截断，仅保留最低 16 位有效位作为最终值传入对应的数据寄存器中。

图 13-17. 20 位到 16 位的结果截断



注意：如果移位后的中间结果还是超过 16 位，那么该结果的高位就会被直接截掉。

*图 13-18. 右移 5 位和取整的数例*描述了一个从原始 20 位的累积数值处理成 16 位结果值的例子。

图 13-18. 右移 5 位和取整的数例

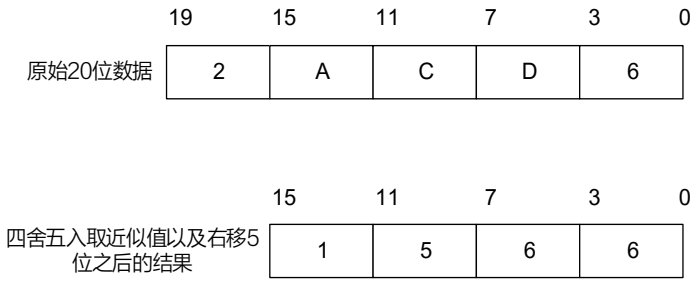


*表 13-6. 不同 N 和 M 组合的最大输出值(灰色值表示截断)*给出了 N 和 M 的各种组合的数据格式，初始转换值为 0xFFF。

表 13-6. 不同 N 和 M 组合的最大输出值(灰色值表示截断)

过采样率	最大原始数据	无移位 OVSS=0000	1 位移位 OVSS=0001	2 位移位 OVSS=0010	3 位移位 OVSS=0011	4 位移位 OVSS=0100	5 位移位 OVSS=0101	6 位移位 OVSS=0110	7 位移位 OVSS=0111	8 位移位 OVSS=1000
2x	0x1FFE	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF	0x00FF	0x007F	0x003F	0x001F
4x	0x3FFC	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF	0x00FF	0x007F	0x003F
8x	0x7FF8	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF	0x00FF	0x007F
16x	0xFFF0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF	0x00FF
32x	0x1FFE0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF
64x	0x3FFC0	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF
128x	0x7FF80	0xFF80	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF
256x	0xFFF00	0xFF00	0xFF80	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF

和标准的转换模式相比，过采样模式的转换时间不会改变：在整个过采样序列的过程中采样时间仍然保持相等。每 N 个转换就会产生一个新的数据，一个等价的延迟为：

$$N \times t_{ADC} = N \times (t_{SMPL} + t_{CONV}) \quad (13\text{-}2)$$

过采样配合 ADC 工作模式

当过采样使能时，大多数 ADC 工作模式都是可用的。

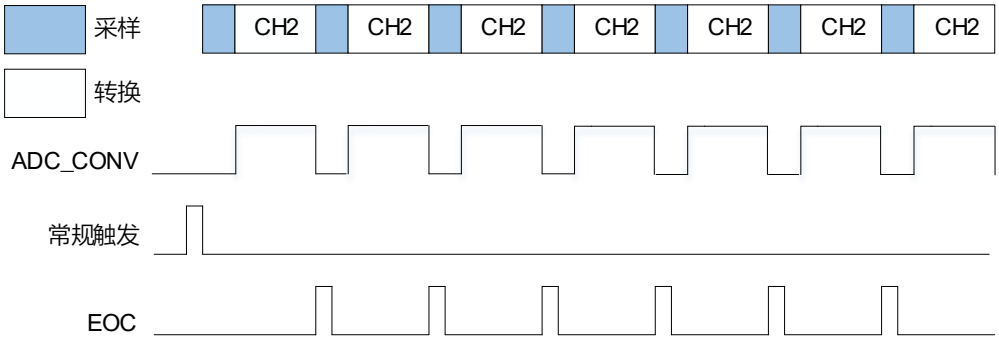
- 常规序列由软件触发或外部触发开始 ADC 转换
- 单次或扫描运行模式，连续或间断运行模式
- 可编程的采样时间
- 模拟看门狗

只有当 ADCON=0 时，才可以改变过采样的配置，并且要保证在设置 ADCON=1 之前要对过采样进行配置。

13.4.16. ADC 转换信号

ADC转换信号ADC_CONV在ADC通道转换期间保持高电平状态，而在其他时间段都为低电平。ADC_CONV信号可内部连接到CLAx模块，作为CLAx多路选择器的输入。

图 13-19. 常规序列连续转换模式下的 ADC 转换信号



13.4.17. 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。

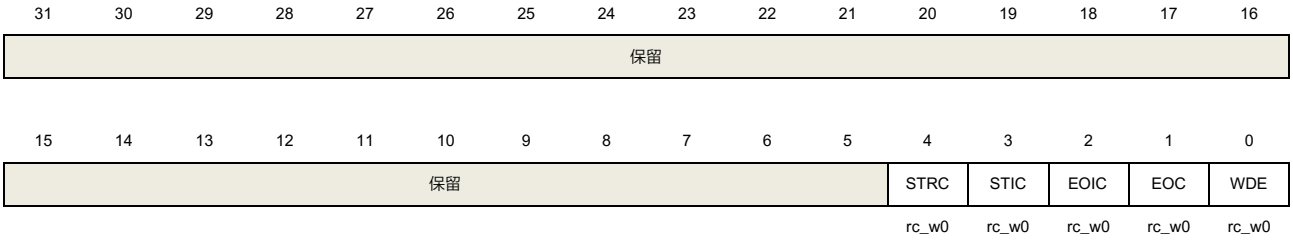
13.5. ADC 寄存器

ADC基地址：0x4001 2400

13.5.1. 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	STIC	注入序列转换开始标志 0：注入序列转换没有开始 0：注入序列转换开始 注入序列转换开始时硬件置位。 软件写0清除。
2	EOIC	注入序列转换结束标志 0：注入序列转换没有结束 1：注入序列转换结束 注入序列转换结束时硬件置位。 软件写0清除。
1	EOC	常规或注入序列转换结束标志 0：转换没有结束 1：转换结束 常规或注入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由硬件置1，软件写0清除。

13.5.2. 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。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留						DRES [1:0]		RWDEN	IWDEN	保留					
						rw		rw	rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DISNUM [2:0]			DISIC	DISRC	ICA	WDSC	SM	EOICIE	WDEIE	EOCIE	WDCHSEL [4:0]				
rw			rw	rw	rw	rw	rw	rw	rw	rw	rw				

位/位域	名称	描述
31:26	保留	必须保持复位值。
25:24	DRES [1:0]	ADC分辨率 00：12位 01：10位 10：8位 11：6位
23	RWDEN	常规序列模拟看门狗使能 0：常规序列模拟看门狗禁能 1：常规序列模拟看门狗使能
22	IWDEN	注入序列模拟看门狗使能 0：注入序列模拟看门狗禁能 1：注入序列模拟看门狗使能
21:16	保留	必须保持复位值。
15:13	DISNUM [2:0]	间断模式下的转换转换数目 触发后即将被转换的通道数目将变成DISNUM[2:0]+1。
12	DISIC	注入序列间断模式 0：注入序列间断模式禁能 1：注入序列间断模式使能
11	DISRC	常规序列间断模式 0：间断运行模式禁能 1：间断运行模式使能
10	ICA	注入序列自动转换 0：注入序列自动转换禁能

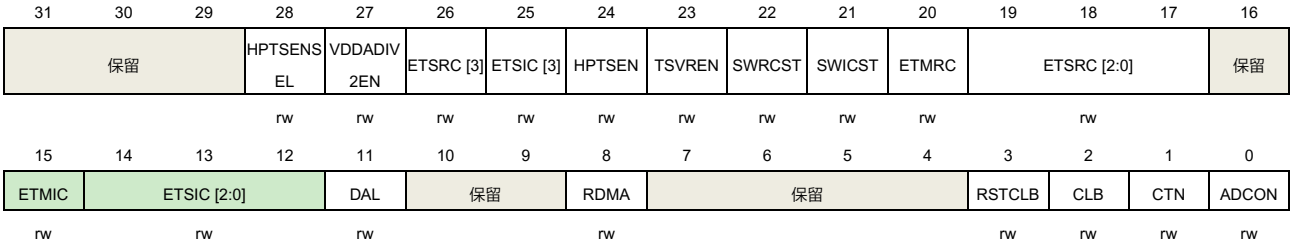
		1：注入序列自动转换使能
9	WDSC	扫描模式下，模拟看门狗在通道配置 0：模拟看门狗在所有通道有效 1：模拟看门狗在单通道有效
8	SM	扫描模式 0：扫描运行模式禁止 1：扫描运行模式使能
7	EOICIE	EOIC中断使能 0：EOIC中断禁能 1：EOIC中断使能
6	WDEIE	WDE中断使能 0：中断禁止 1：中断使能
5	EOCIE	EOC中断使能 0：中断禁止 1：中断使能
4:0	WDCHSEL [4:0]	模拟看门狗通道选择 00000：ADC通道0 00001：ADC通道1 00010：ADC通道2 00011：ADC通道3 00100：ADC通道4 00101：ADC通道5 00110：ADC通道6 00111：ADC通道7 01000：ADC通道8 01001：ADC通道9 01010：ADC通道10 01011：ADC通道11 01100：ADC通道12 01101：ADC通道13 01110：ADC通道14 01111：ADC通道15 1x000：ADC通道16 1x001：ADC通道17 1x010：ADC通道18 1x011：ADC通道19 1x100：ADC通道20 1x101：ADC通道21 1x11x：ADC通道22 其他值保留。

注意：ADC的模拟输入通道16、通道17和通道18分别连接到高精度温度传感器、V_{REFINT}和$V_{DDADIV2}$。通道19~22为外部输入引脚。

13.5.3. 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。



位/位域	名称	描述
31:29	保留	必须保持复位值。
28	HPTSENSEL	高精度温度传感器使能选择 0：高精度温度传感器通过TSVREN位使能 1：高精度温度传感器通过HPTSEN位使能
27	VDDADIV2EN	ADC的通道18(模拟电源电压V_{DDA}/2)使能 0：ADC的通道18禁止 1：ADC的通道18使能
26	ETSRC [3]	常规序列外部触发选择，具体参考ETSRC [2:0]。
25	ETSIC [3]	注入序列外部触发选择，具体参考ETSIC [2:0]。
24	HPTSEN	ADC的通道16(高精度温度传感器)使能 0：ADC的通道16禁止 1：ADC的通道16使能
23	TSVREN	ADC的通道16(高精度温度传感器)和通道17(内部电压参考)使能 当HPTSENSEL = 1时，只有通道17使能 0：ADC的通道17禁止 1：ADC的通道17使能
22	SWRCST	常规序列软件启动转换开始 如果ETSRC是111，该位置'1'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	SWICST	注入序列软件启动转换开始 如果ETSIC是111，该位置'1'开启注入序列转换。该位由软件置位，软件清零或转换开始由硬件清零。

20	ETMRC	常规序列外部触发使能 0：常规序列外部触发禁止 1：常规序列外部触发使能
19:17	ETSRC [2:0]	常规序列通道外部触发选择 软件写这些位(包括ETSRC[3])来确定常规序列的外部触发选择。 0000：TIMER0 CH0 0001：TIMER0 CH1 0010：TIMER0 CH2 0011：TIMER1 CH1 0100：TIMER2 TRGO 0101：TIMER14 CH0 0110：中断线11 0111：软件触发SWRCST 1000：CLA0 TRGO 1001：CLA2 TRGO 1010~1111：保留
16	保留	必须保持复位值。
15	ETMIC	注入序列外部触发使能 0：注入序列外部触发禁止 1：注入序列外部触发使能
14:12	ETSIC [2:0]	注入序列通道外部触发选择 软件写这些位(包括ETSIC[3])来确定注入序列的外部触发选择。 0000：TIMER0 TRGO 0001：TIMER0 CH3 0010：TIMER1 TRGO 0011：TIMER1 CH0 0100：TIMER2 CH3 0101：TIMER14 TRGO 0110：中断线15 0111：软件触发SWICST 1000：CLA1 TRGO 1001：CLA3 TRGO 1010~1111：保留
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	RDMA	常规序列的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 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。当该位为1时，并且在该寄存器的其他位不变的情况下将"1"写入该位时，常规序列转换将开始。 0：禁能ADC并掉电 1：使能ADC

13.5.4. 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					SPT18[2:0]			SPT17[2:0]			SPT16[2:0]			SPT15[2:1]	
					rw			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
31:27	保留	必须保持复位值。
26:24	SPT18[2:0]	参考SPT10[2:0]的描述
23:21	SPT17[2:0]	参考SPT10[2:0]的描述
29:18	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

13.5.5. 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

13.5.6. 注入通道数据偏移寄存器 x(ADC_IOFFx)(x=0..3)

地址偏移：0x14 + 0x04 * x (x=0..3)
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	IOFF[11:0]	注入通道x的数据偏移 当转换注入通道时，这些位定义了用于从原始转换数据中减去的数值。转换的结果可以在ADC_IDATA寄存器中读出。

13.5.7. 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。



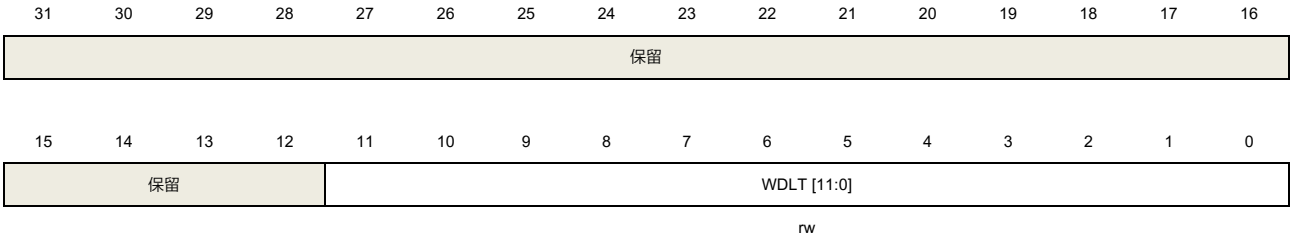
位/位域	名称	描述
31:12	保留	必须保持复位值。

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。

13.5.8. 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

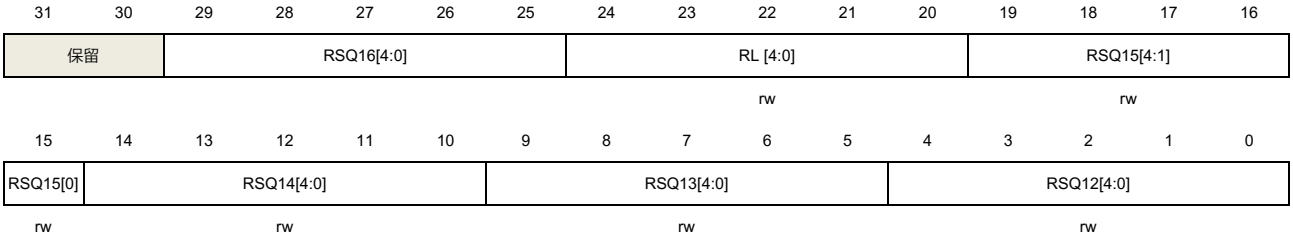


位/位域	名称	描述
31:12	保留	必须保持复位值。
11:0	WDLT [11:0]	模拟看门狗低侧阈值 这些位定义了模拟看门狗的低侧阈值。

13.5.9. 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。



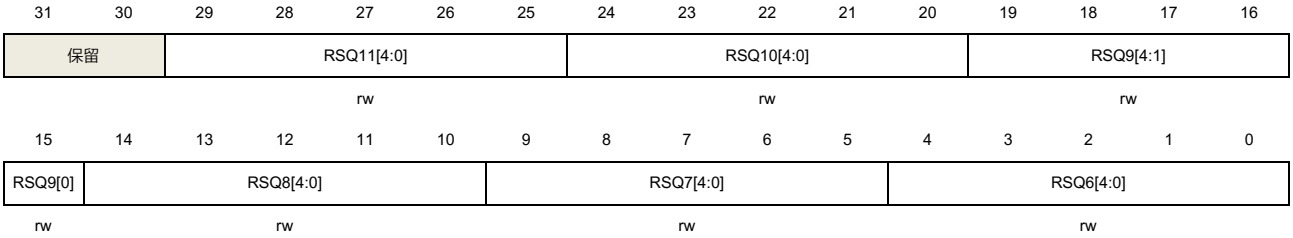
位/位域	名称	描述
31:24	保留	必须保持复位值。
29:25	RSQ16[4:0]	参考RSQ0[4:0]的描述
24:20	RL [4:0]	常规通道长度 常规通道转换序列中的总通道数目为RL[4:0]+1。
19:15	RSQ15[4:0]	参考RSQ0[4:0]的描述

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

13.5.10. 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。

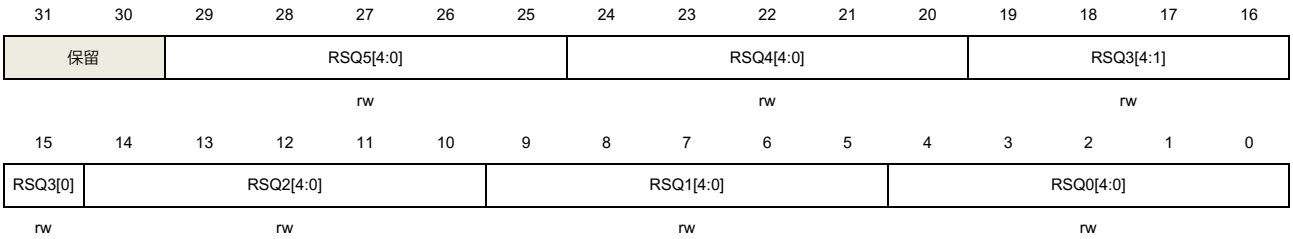


位/位域	名称	描述
31:30	保留	必须保持复位值。
29:25	RSQ11[4:0]	参考RSQ0[4:0]的描述
24:20	RSQ10[4:0]	参考RSQ0[4:0]的描述
19:15	RSQ9[4:0]	参考RSQ0[4:0]的描述
14:10	RSQ8[4:0]	参考RSQ0[4:0]的描述
9:5	RSQ7[4:0]	参考RSQ0[4:0]的描述
4:0	RSQ6[4:0]	参考RSQ0[4:0]的描述

13.5.11. 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。

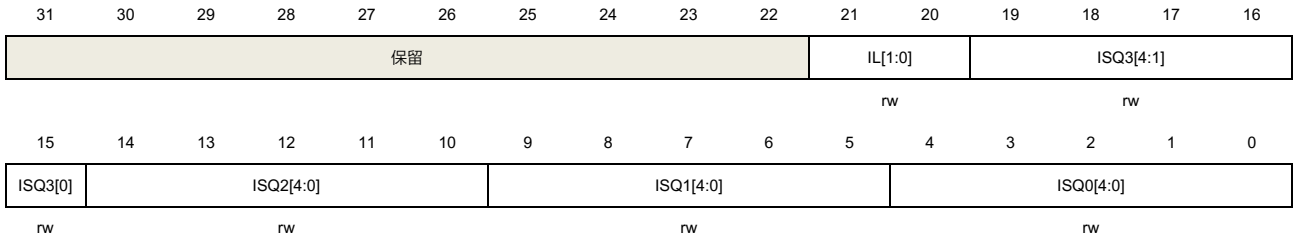


位/位域	名称	描述
31:30	保留	必须保持复位值。
29:25	RSQ5[4:0]	参考RSQ0[4:0]的描述
24:20	RSQ4[4:0]	参考RSQ0[4:0]的描述
19:15	RSQ3[4:0]	参考RSQ0[4:0]的描述
14:10	RSQ2[4:0]	参考RSQ0[4:0]的描述
9:5	RSQ1[4:0]	参考RSQ0[4:0]的描述
4:0	RSQ0[4:0]	通道编号(0..17)写入这些位来选择常规通道的第n个转换的通道

13.5.12. 注入序列寄存器(ADC_ISQ)

地址偏移：0x38
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31:22	保留	必须保持复位值。
21:20	IL[1:0]	注入序列长度 注入序列总的通道数目为IL[1:0]+1。
19:15	ISQ3[4:0]	参考ISQ0[4:0]的描述
14:10	ISQ2[4:0]	参考ISQ0[4:0]的描述
9:5	ISQ1[4:0]	参考ISQ0[4:0]的描述
4:0	ISQ0[4:0]	通道编号(0..22)写入这些位来选择注入序列的第 n 个转换的通道 和常规转换序列不同的是，如果 IL[1:0]长度不足 4，注入通道转换从(4-IL[1:0]-1)开始。 IL 注入通道转换顺序 3 ISQ0 >> ISQ1 >> ISQ2 >> ISQ3 2 ISQ1 >> ISQ2 >> ISQ3 1 ISQ2 >> ISQ3 0 ISQ3

13.5.13. 锁存数据寄存器 x(ADC_LDATAx)(x= 0..3)

地址偏移：0x3C + 0x04 * x (x=0..3)
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
LDATAn [15:0]															
r															

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	LDATAn[15:0]	注入或者常规序列第 n 个转换数据 这些位包含了注入或者常规序列第 n 个转换数据，只读。

13.5.14. 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。

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
RDATA [15:0]															
r															

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。

13.5.15. 过采样控制寄存器(ADC_OVSAMPCTL)

地址偏移：0x80
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						TOVS	OVSS [3:0]				OVSR [2:0]			保留	OVSEN
						rw	rw				rw				rw

位/位域	名称	描述
31:10	保留	必须保持复位值。
9	TOVS	过采样触发 该位通过软件设置和清除。 0：在一个触发后，对一个通道连续进行过采样转换 1：对于过采样通道的每次转换都需要一次触发，触发次数由过采样率(OVSR[2:0])决定 **注意：**当ADCON= 0时软件才允许写该位(确定没有转换正在进行)。
8:5	OVSS [3:0]	过采样移位 该位通过软件设置和清除。 0000：不移位 0001：移1位 0010：移2位 0011：移3位 0100：移4位 0101：移5位 0110：移6位 0111：移7位 1000：移8位 其它位保留 **注意：**只有在ADCON=0的时候，才允许通过软件对该位进行写操作(确保没有转换正在进行)。
4:2	OVSR [2:0]	过采样率 这些位定义了过采样率的大小。 000：2x 001：4x 010：8x 011：16x 100：32x 101：64x 110：128x 111：256x **注意：**只有在ADCON=0的时候，才允许通过软件对该位进行写操作(确保没有转换正在执行)。

1	保留	必须保持复位值。
0	OVSEN	过采样使能 该位通过软件置位和清除。 0：过采样禁止 1：过采样使能 **注意：** 只有在ADCON=0的时候，才允许通过软件对该位进行写操作(确保没有转换正在执行)。

13.5.16. 采样时间寄存器 2 (ADC_SAMPT2)

地址偏移：0x84
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															

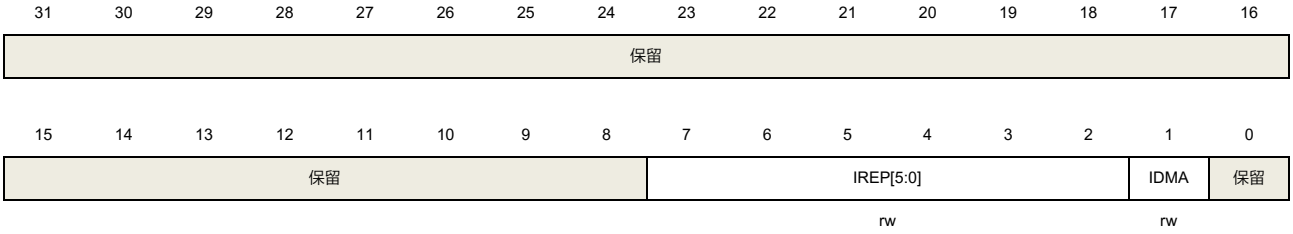
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				SPT22[2:0]			SPT21[2:0]			SPT20[2:0]			SPT19[2:0]		
				rw			rw			rw			rw		

位/位域	名称	描述
31:12	保留	必须保持复位值。
11:9	SPT22[2:0]	参考SPT19[2:0]的描述
8:6	SPT21[2:0]	参考SPT19[2:0]的描述
5:3	SPT20[2:0]	参考SPT19[2:0]的描述
2:0	SPT19[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

13.5.17. 控制寄存器 2 (ADC_CTL2)

地址偏移：0x88
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

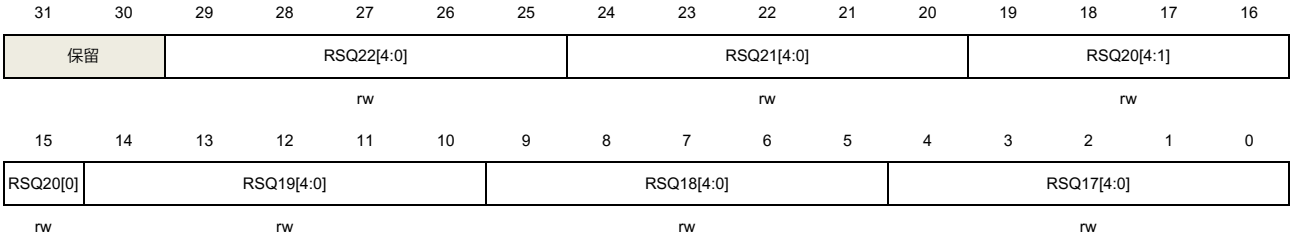


位/位域	名称	描述
31:8	保留	必须保持复位值
7:2	IREP[5:0]	注入序列通道转换重复次数 这些位用于控制注入序列通道的转换次数，转换次数为IREP[5:0] +1 **注意：**当注入序列通道使用间断模式时，这些位保持复位值。 000000：1次 000001：2次 …. 111111：64次
1	IDMA	注入序列的DMA请求使能 0：DMA请求禁止 1：DMA请求使能
0	保留	必须保持复位值

13.5.18. 常规序列寄存器 3 (ADC_RSQ3)

地址偏移：0x8C
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。



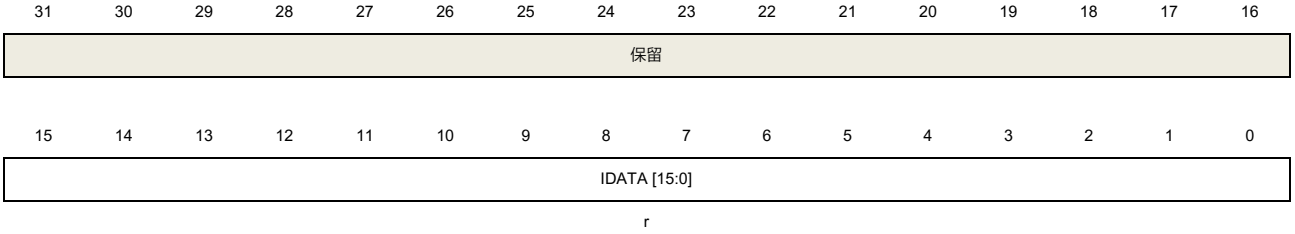
位/位域	名称	描述
31:30	保留	必须保持复位值。
29:25	RSQ22[4:0]	参考RSQ0[4:0]的描述
24:20	RSQ21[4:0]	参考RSQ0[4:0]的描述

19:15	RSQ20[4:0]	参考RSQ0[4:0]的描述
14:10	RSQ19[4:0]	参考RSQ0[4:0]的描述
9:5	RSQ18[4:0]	参考RSQ0[4:0]的描述
4:0	RSQ17[4:0]	参考RSQ0[4:0]的描述

13.5.19. 注入数据寄存器(ADC_IDATA)

地址偏移：0x90
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

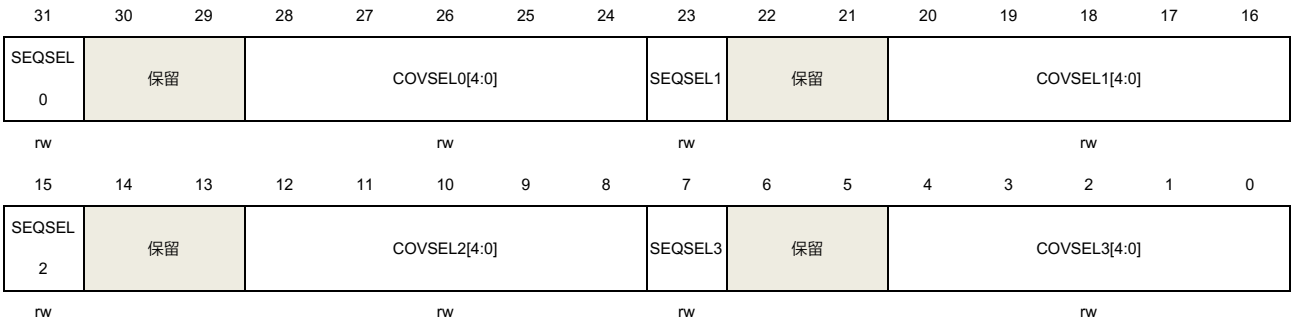


位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	IDATA[15:0]	注入通道数据 这些位包含了注入通道的转换结果，只读。

13.5.20. 锁存数据控制寄存器(ADC_LDCTL)

地址偏移：0x94
复位值：0x0001 0203

该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31	SEQSEL0	ADC_LDATA0寄存器序列源选择 0：选择注入序列

		1：选择常规序列 **注意：**只有在ADC禁能(ADCON=0)时，才能软件写这些位。
30:29	保留	必须保持复位值。
28:24	COVSEL0[4:0]	ADC_LDATA0寄存器转换源选择 当ADC_LDATA0寄存器用于序列转换时，可以选择将注入序列(SEQSEL0 = 0)或者常规序列(SEQSEL0 = 1)的第n个转换的数据存储在ADC_LDATA0寄存器中。 00000：序列第0个转换 00001：序列第1个转换 00010：序列第2个转换 … 10110：序列第22个转换 其他值保留。 **注意：**在注入序列(SEQSEL0 = 0)中，只能为5'b00000、5'b00001、5'b00010和5'b00011，COVSELn[4:0]的默认值为n。只有在ADC禁能(ADCON=0)时，才能软件写这些位。
23	SEQSEL1	参考SEQSEL0描述。
22:21	保留	必须保持复位值。
20:16	COVSEL1[4:0]	参考COVSEL0描述。
15	SEQSEL2	参考SEQSEL0描述。
14:13	保留	必须保持复位值。
12:8	COVSEL2[4:0]	参考COVSEL0描述。
7	SEQSEL3	参考SEQSEL0描述。
6:5	保留	必须保持复位值。
4:0	COVSEL3[4:0]	参考COVSEL0描述。

14. 数模转换器(DAC)

14.1. 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。

每个 DAC 的两个输出通道可以独立或并发工作。

14.2. 主要特征

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_{REFP}；
- 噪声波生成(LSFR 噪声模式和三角噪声模式)；
- DACx 双通道并发模式；
- 复位保持功能。

*图 14-1.DAC 结构框图*为 DAC 的结构框图，*表 14-1.DAC 引脚*给出了引脚描述。

图 14-1.DAC 结构框图

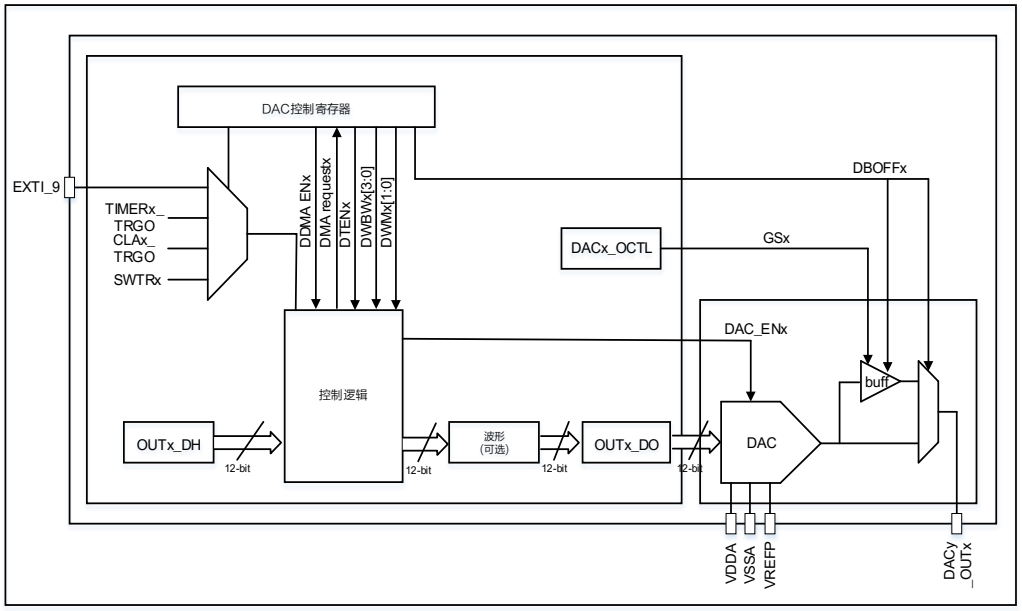


表 14-1.DAC 引脚

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
V_{REFP}	DAC 正参考电压	输入，模拟正参考电压
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 14-2. DAC 触发与输出

	DAC0		DAC1	
通道	通道 0	通道 1	通道 0	通道 1
DAC 输出 I/O	PA4	PA5	PA6	PA7
DAC 输出 BUFFER 功能	•		•	
软件触发功能	•			
EXTI 触发信号	EXTI_9			
TIMER 触发信号	TIMER0_TRGO TIMER1_TRGO TIMER2_TRGO TIMER5_TRGO TIMER6_TRGO TIMER14_TRGO			
CLA 触发信号	CLA0_TRGO CLA1_TRGO CLA2_TRGO			

	DAC0	DAC1
	CLA3_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。

14.3. 功能描述

14.3.1. 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。

14.3.2. 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。

14.3.3. DAC 数据配置

对于 12 位的 DAC 保持数据(OUTx_DH)，可以通过对 DAC_OUTx_R12DH、DAC_OUTx_L12DH 和 DAC_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 DAC_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。

14.3.4. 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 14-3.DAC 外部触发*所示。

表 14-3.DAC 外部触发

DTSELx[3:0]	触发源	触发类型
4b'0000	TIMER5_TRGO	硬件触发
4b'0001	TIMER14_TRGO	
4b'0010	TIMER6_TRGO	
4b'0011	TIMER2_TRGO	
4b'0100	TIMER1_TRGO	
4b'0101	TIMER0_TRGO	
4b'0110	EXTI_9	
4b'0111	SWTRIG	软件触发
4b'1000	CLA0_TRGO	硬件触发

DTSELx[3:0]	触发源	触发类型
4b'1001	CLA1_TRGO	
4b'1010	CLA2_TRGO	
4b'1011	CLA3_TRGO	

TIMERx_TRGO 信号是由定时器生成的，CLAx_TRGO 信号是由可配置逻辑阵列 CLA 提供，而软件触发是通过设置 DAC_SWT 寄存器的 SWTRx 位生成的。

14.3.5. 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(DAC_OUTx_DO)。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(DAC_OUTx_DO)。

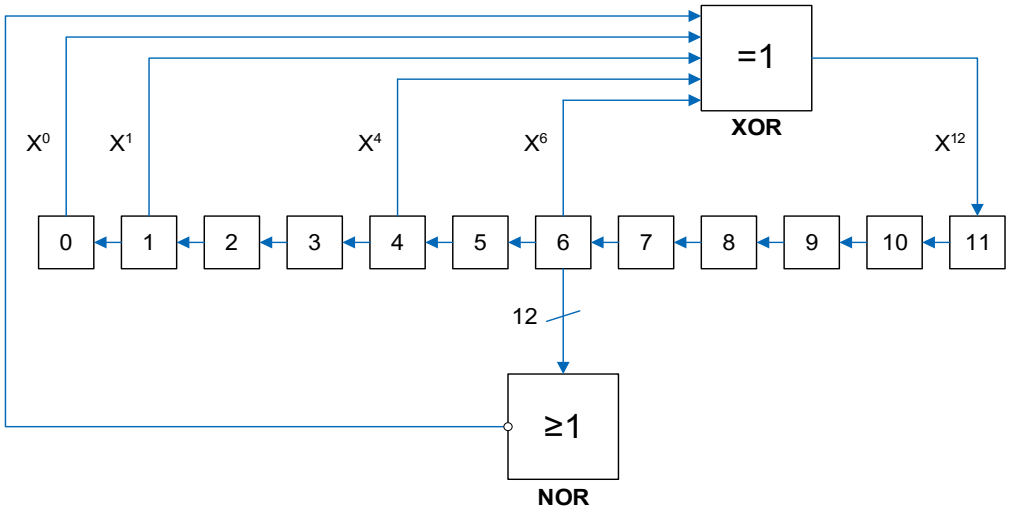
当 DAC 保持数据(OUTx_DH)加载到 DAC_OUTx_DO 寄存器时，经过 $t_{SETTLING}$ 时间之后，模拟输出变得有效，$t_{SETTLING}$ 的值与电源电压和模拟输出负载有关。

14.3.6. DAC 噪声波

有两种方式可以将噪声波加载到 DAC 输出数据：LFSR 噪声波和三角波。噪声波模式可以通过 DAC_CTL0 寄存器的 DWMx 位来进行选择。噪声的幅值可以通过配置 DAC_CTL0 寄存器的 DAC 噪声波位宽(DWBWx)位来进行设置。

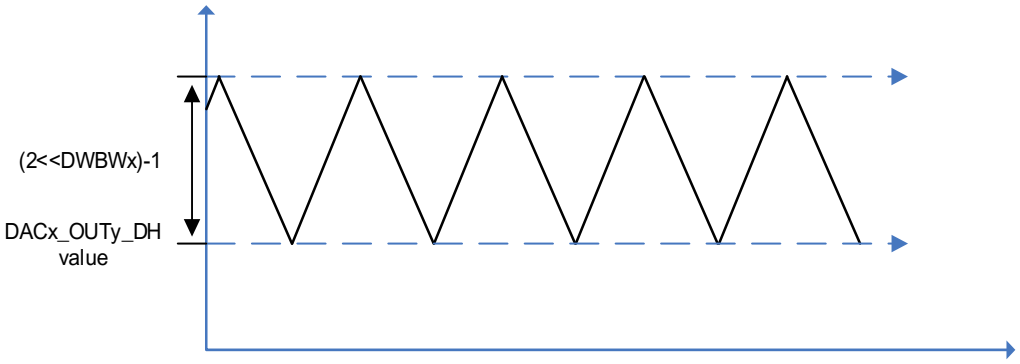
LFSR 噪声模式：在 DAC 控制逻辑中有一个线性反馈移位寄存器(LFSR)。在此模式下，LFSR 的值与 OUTx_DH 值相加后，被写入到 DAC 数据输出寄存器(DAC_OUTx_DO)。当配置的 DAC 噪声波位宽小于 12 时，LFSR 的值等于 LFSR 寄存器最低的 DWBWx 位，高位被屏蔽。

图 14-2. DAC LFSR 算法



三角噪声模式：三角波信号与 OUTx_DH 值相加后，被写入到 DAC 数据输出寄存器(DAC_OUTx_DO)。三角波幅值的最小值为 0，最大值为(2 << DWBWx) - 1。

图 14-3. DAC 三角噪声模式生成的波形



14.3.7. DAC 输出电压

DAC 引脚上的模拟输出电压取决于下面的公式：

$$DAC_{output}=V_{REFP}*OUTx_DO/4096 \quad (14\text{-}1)$$

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_{REFP}。

当 DAC_OCTL 寄存器中的 GSx 位置 1 且输出缓冲器打开时，DAC 引脚上的模拟输出电压取决于下面的公式：

$$V_{DAC_OUT}=V_{REFP}*OUTx_DO/2048 \quad (14\text{-}2)$$

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}。

14.3.8. 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。DMA 请求将被暂停，直到 DDUDRx 位被清除。

在 DAC 并发转换模式下，只需要任意一个通道的 DDMAENx 位置 1，因此只需要配置一个通道的 DDUDRIEx 位。

14.3.9. DAC 并发转换

当 DAC 的两个通道同时工作时，为了在特定应用中最大限度利用总线带宽，DAC 的两个通道可以被配置为并发模式。在并发模式中，DAC 的 OUTx_DH 和 OUTx_DO 值将同时被更新。

有 3 个并发寄存器可被用于加载 OUTx_DH 的值，分别是：DACC_R8DH、DACC_R12DH 和 DACC_L12DH。配置其中的任意一个寄存器都可实现同时驱动 DAC 的两个通道。

当使能了外部触发时，DAC 两个通道的 DTENx 位都需要置 1，需要配置 DTSEL0/1 相同来保证同时触发。

当使能了 DMA 功能时，DAC 任一通道的 DDMAENx 位置 1 即可。

噪声模式和噪声位宽可以根据使用情况配置为相同或不同。

14.3.10. DAC 复位保持模式

通过设置 DAC_CTL0 寄存器的 DRSTMDx 位使能 DAC 的复位保持模式，在除了上电复位之外的其他复位信号到来时，DAC 通道的输出将会保持。

当 DRSTMDx 位置 1 时，寄存器 DAC_CTL0(DRSTMDx, DTENx, DBOFFx 和 DENx 位), OUTx_DH 和 OUTx_DO 只能被上电复位清零。

14.4. DAC 寄存器

DAC0 基地址：0x4000 7400

DAC1 基地址：0x4000 CC00

14.4.1. 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
DRSTMD1	DTSEL1[3]	DDUDRIE1	DDMAEN1	DWBW1[3:0]				DWM1[1:0]		DTSEL1[2:0]			DTEN1	DBOFF1	DEN1
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
DRSTMD0	DTSEL0[3]	DDUDRIE0	DDMAEN0	DWBW0[3:0]				DWM0[1:0]		DTSEL0[2:0]			DTEN0	DBOFF0	DEN0
rw	rw	rw	rw	rw				rw		rw			rw	rw	rw

位/位域	名称	描述
31	DRSTMD1	DACx_OUT1 复位模式 0：普通模式：所有的复位信号都会复位 DACx_OUT1 相关的寄存器 1：保持模式：在除了上电复位之外的其他复位信号到来时，DACx_OUT1 的输出将会保持
30	DTSEL1[3]	DACx_OUT1 触发源选择的位[3]，参考DTSEL1[2:0]
29	DDUDRIE1	DACx_OUT1 DMA欠载中断使能 0: DACx_OUT1 DMA 欠载中断禁能 1: DACx_OUT1 DMA欠载中断使能
28	DDMAEN1	DACx_OUT1 DMA 使能 0：DACx_OUT1 DMA 模式禁能 1：DACx_OUT1 DMA 模式使能
27:24	DWBW1[3:0]	DACx_OUT1 噪声波位宽 这些位指定了 DACx_OUT1 的噪声波信号的位宽。LFSR 噪声模式下，这些位表示不屏蔽 LFSR 的位[n-1, 0]；三角噪声模式下，这些位表示三角波幅值为(2<<(n-1))-1。其中，n 为噪声波位宽。 0000：波形信号的位宽为 1 0001：波形信号的位宽为 2 0010：波形信号的位宽为 3 0011：波形信号的位宽为 4

		0100：波形信号的位宽为 5 0101：波形信号的位宽为 6 0110：波形信号的位宽为 7 0111：波形信号的位宽为 8 1000：波形信号的位宽为 9 1001：波形信号的位宽为 10 1010：波形信号的位宽为 11 >=1011：波形信号的位宽为 12
23:22	DWM1[1:0]	DACx_OUT1 噪声波模式 这些位指定了在 DACx_OUT1 外部触发使能(DTEN1=1)的情况下，DACx_OUT1 的噪声波模式的选择。 00：波形生成禁能 01：LFSR 噪声模式 1x：三角噪声模式
21:19	DTSEL1[2:0]	DACx_OUT1 触发选择 这些位仅在 DTEN1=1 并选择用于触发 DAC 的外部事件时使用。 0000：TIMER5 TRGO 0001：TIMER14 TRGO 0010：TIMER6 TRGO 0011：TIMER2 TRGO 0100：TIMER1 TRGO 0101：TIMER0 TRGO 0110：外部中断线 9 0111：软件触发 1000：CLA0_TRGO 1001：CLA1_TRGO 1010：CLA2_TRGO 1011：CLA3_TRGO 11xx：保留
18	DTEN1	DACx_OUT1 触发使能 0：DACx_OUT1 触发禁能 1：DACx_OUT1 触发使能
17	DBOFF1	DACx_OUT1 输出缓冲区关闭 0：DACx_OUT1 输出缓冲区打开，以降低输出阻抗，提高驱动能力 1：DACx_OUT1 输出缓冲区关闭
16	DEN1	DACx_OUT1 使能 0：DACx_OUT1 禁能 1：DACx_OUT1 使能
15	DRSTMD0	DACx_OUT0 复位模式 0：普通模式：所有的复位信号都会复位 DACx_OUT0 相关的寄存器 1：保持模式：在除了上电复位之外的其他复位信号到来时，DACx_OUT0 的输出

		将会保持
14	DTSEL0[3]	DACx_OUT0触发源选择的位[3]，参考DTSEL0[2:0]
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
11:8	DWBW0[3:0]	DACx_OUT0 噪声波位宽 这些位指定了 DACx_OUT0 的噪声波信号的位宽。LFSR 噪声模式下，这些位表示不屏蔽 LFSR 的位[n-1, 0]；三角噪声模式下，这些位表示三角波幅值为(2<<(n-1))-1。其中，n 为噪声波位宽。 0000：波形信号的位宽为 1 0001：波形信号的位宽为 2 0010：波形信号的位宽为 3 0011：波形信号的位宽为 4 0100：波形信号的位宽为 5 0101：波形信号的位宽为 6 0110：波形信号的位宽为 7 0111：波形信号的位宽为 8 1000：波形信号的位宽为 9 1001：波形信号的位宽为 10 1010：波形信号的位宽为 11 >=1011：波形信号的位宽为 12
7:6	DWM0[1:0]	DACx_OUT0 噪声波模式 这些位指定了在 DACx_OUT0 外部触发使能(DTEN0=1)的情况下，DACx_OUT0 的噪声波模式的选择。 00：波形生成禁能 01：LFSR 噪声模式 1x：三角噪声模式
5:3	DTSEL0[2:0]	DACx_OUT0 触发选择 这些位仅在 DTEN0=1 并选择用于触发 DAC 的外部事件时使用。 0000：TIMER 5 TRGO 0001：TIMER14 TRGO 0010：TIMER6 TRGO 0011：TIMER2 TRGO 0100：TIMER1 TRGO 0101：TIMER0 TRGO 0110：外部中断线 9 0111：软件触发 1000：CLA0_TRGO

		1001：CLA1_TRGO 1010：CLA2_TRGO 1011：CLA3_TRGO 11xx：保留
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

14.4.2. 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

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

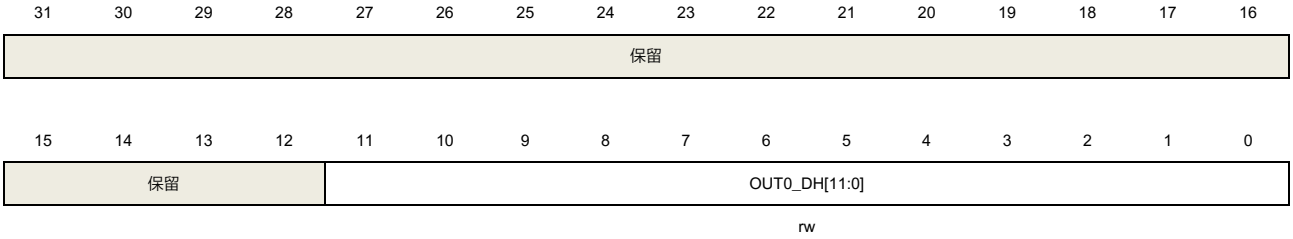
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留														SWTR1	SWTR0
														w	w

位/位域	名称	描述
31:2	保留	必须保持复位值
1	SWTR1	DACx_OUT1 软件触发，由硬件清除 0：软件触发禁能 1：软件触发使能
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

14.4.3. DACx_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

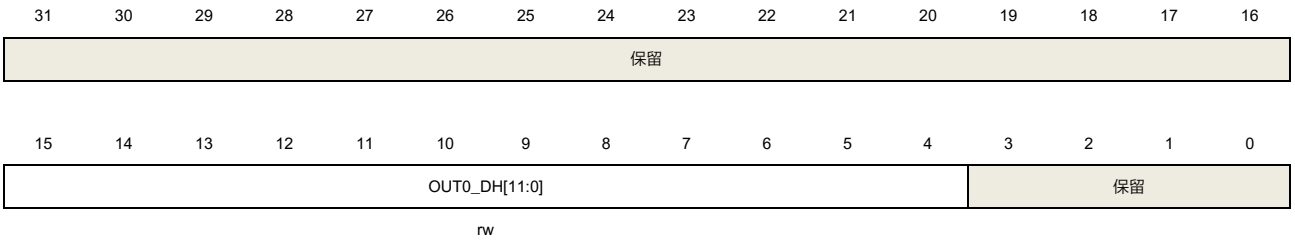


位/位域	名称	描述
31:12	保留	必须保持复位值
11:0	OUT0_DH[11:0]	DACx_OUT0 12 位右对齐数据 这些位指定了将由 DACx_OUT0 转换的数据。

14.4.4. DACx_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

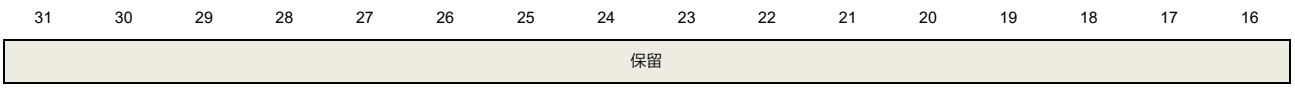


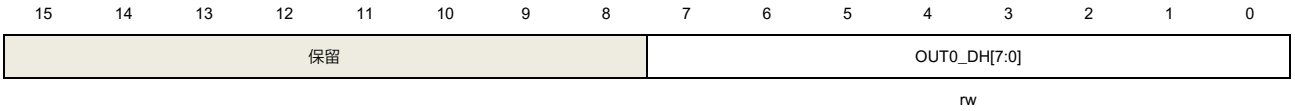
位/位域	名称	描述
31:16	保留	必须保持复位值
15:4	OUT0_DH[11:0]	DACx_OUT0 12 位左对齐数据 这些位指定了将由 DACx_OUT0 转换的数据。
3:0	保留	必须保持复位值

14.4.5. DACx_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



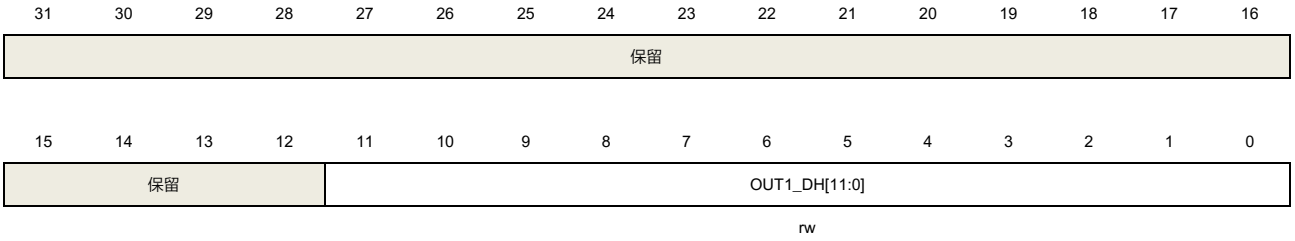


位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	OUT0_DH[7:0]	DACx_OUT0 8 位右对齐数据 这些位指定了将由 DACx_OUT0 转换的数据的最高 8 位有效位。

14.4.6. DACx_OUT1 12 位右对齐数据保持寄存器 (DAC_OUT1_R12DH)

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
31:12	保留	必须保持复位值
11:0	OUT1_DH[11:0]	DACx_OUT1 12 位右对齐数据 这些位指定了将由 DACx_OUT1 转换的数据。

14.4.7. DACx_OUT1 12 位左对齐数据保持寄存器 (DAC_OUT1_L12DH)

地址偏移：0x18
复位值：0x0000 0000

该寄存器只能按字(32 位)访问



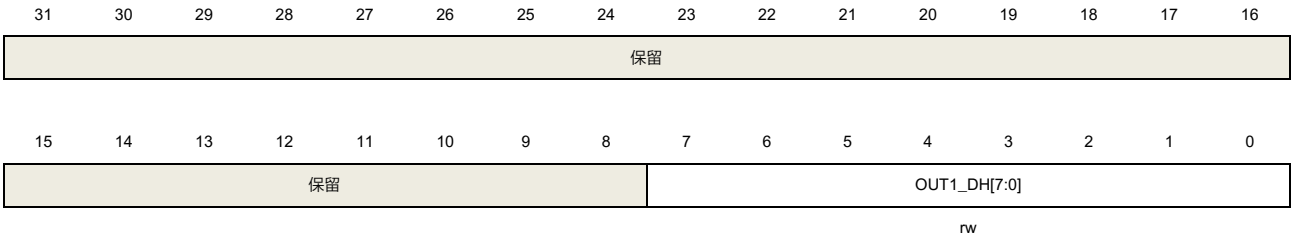
位/位域	名称	描述

31:16	保留	必须保持复位值
15:4	OUT1_DH[11:0]	DACx_OUT1 12 位左对齐数据 这些位指定了将由 DACx_OUT1 转换的数据。
3:0	保留	必须保持复位值

14.4.8. DACx_OUT1 8 位右对齐数据保持寄存器 (DAC_OUT1_R8DH)

地址偏移：0x1C
复位值：0x0000 0000

该寄存器只能按字(32 位)访问

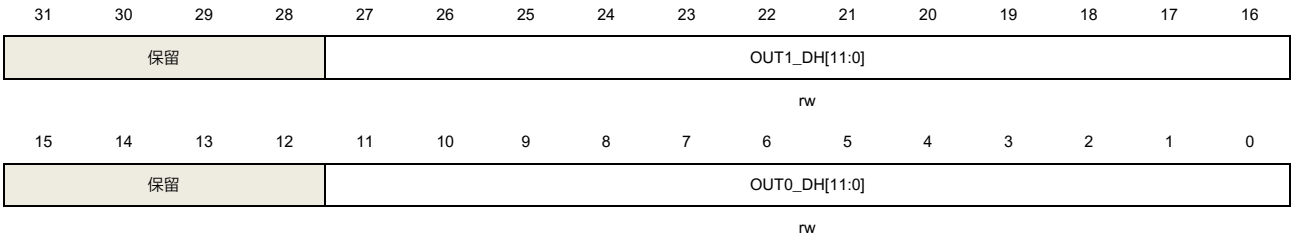


位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	OUT1_DH[7:0]	DACx_OUT1 8 位右对齐数据 这些位指定了将由 DACx_OUT1 转换的数据的 8 位最高有效位。

14.4.9. DACx 并发模式 12 位右对齐数据保持寄存器 (DACC_R12DH)

地址偏移：0x20
复位值：0x0000 0000

该寄存器只能按字(32 位)访问



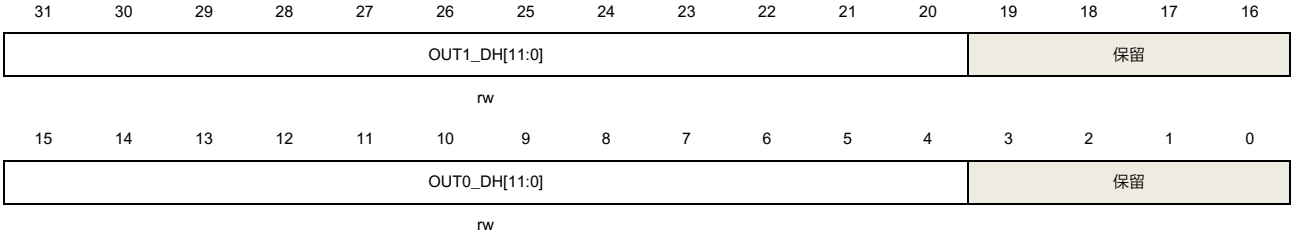
位/位域	名称	描述
31:28	保留	必须保持复位值
27:16	OUT1_DH[11:0]	DACx_OUT1 12 位右对齐数据 这些位指定了将由 DACx_OUT1 转换的数据。

15:12	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。

14.4.10. DACx 并发模式 12 位左对齐数据保持寄存器 (DACC_L12DH)

地址偏移：0x24
复位值：0x0000 0000

该寄存器只能按字(32 位)访问

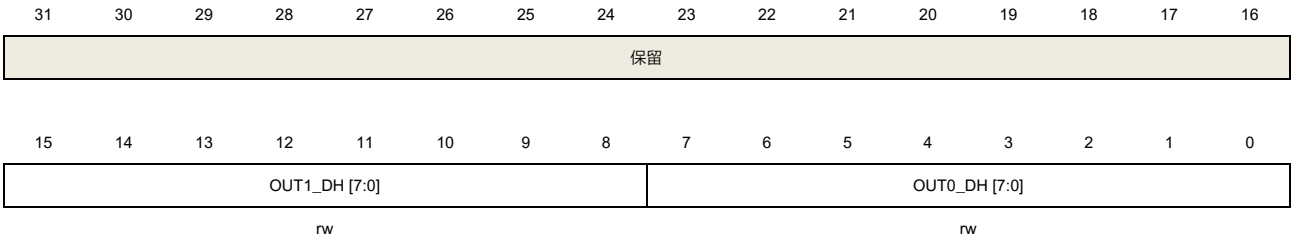


位/位域	名称	描述
31:20	OUT1_DH[11:0]	DACx_OUT1 12 位左对齐数据 这些位指定了将由 DACx_OUT1 转换的数据。
19:16	保留	必须保持复位值
15:4	OUT0_DH[11:0]	DACx_OUT0 12 位左对齐数据 这些位指定了将由 DACx_OUT0 转换的数据。
3:0	保留	必须保持复位值

14.4.11. DACx 并发模式 8 位右对齐数据保持寄存器 (DACC_R8DH)

地址偏移：0x28
复位值：0x0000 0000

该寄存器只能按字(32 位)访问



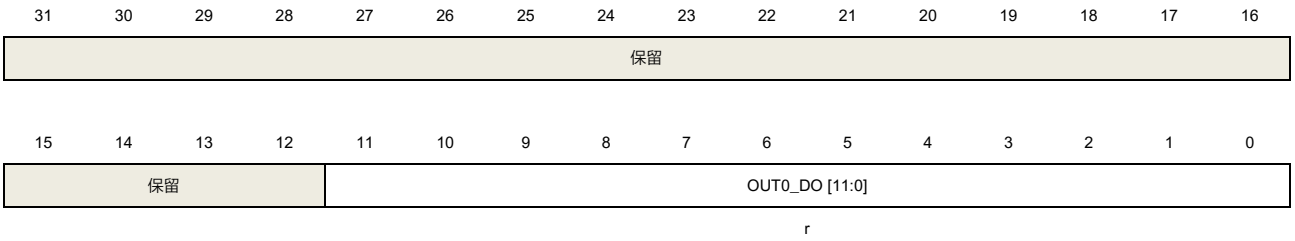
位/位域	名称	描述
31:16	保留	必须保持复位值

15:8	OUT1_DH[7:0]	DACx_OUT1 8 位右对齐数据 这些位指定了将由 DACx_OUT1 转换的数据的 8 位最高有效位。
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 8 位最高有效位。

14.4.12. DACx_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

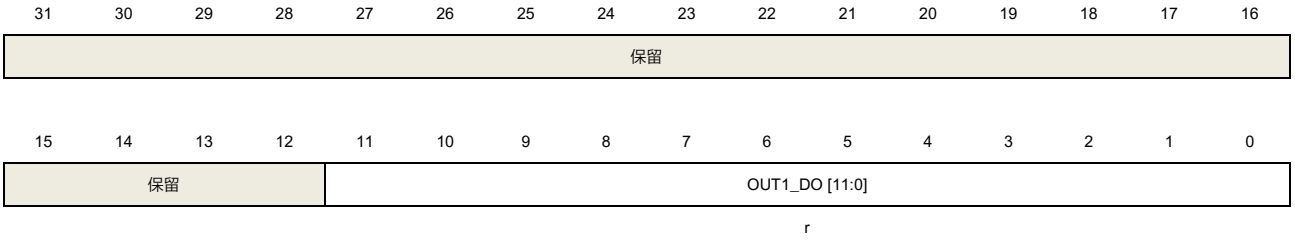


位/位域	名称	描述
31:12	保留	必须保持复位值
11:0	OUT0_DO [11:0]	DACx_OUT0 12bit 数据输出 这些位为只读类型，存储由 DACx_OUT0 转换的数据。

14.4.13. DACx_OUT1 数据输出寄存器 (DAC_OUT1_DO)

地址偏移：0x30
复位值：0x0000 0000

该寄存器只能按字(32 位)访问

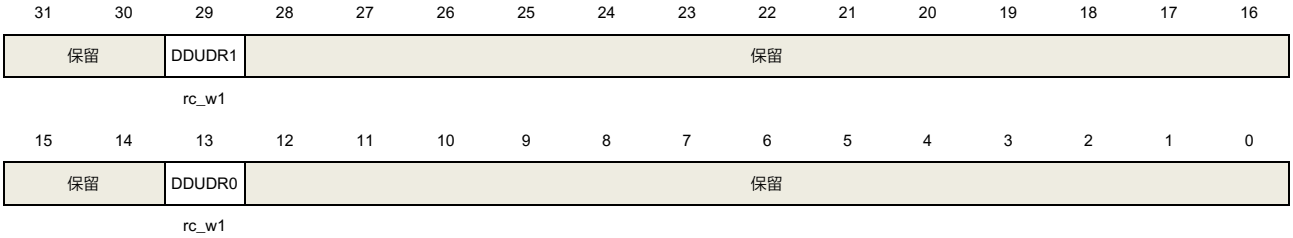


位/位域	名称	描述
31:12	保留	必须保持复位值
11:0	OUT1_DO [11:0]	DACx_OUT1 12bit 数据输出 这些位为只读类型，存储由 DACx_OUT1 转换的数据。

14.4.14. DACx 状态寄存器 (DAC_STAT0)

地址偏移： 0x34
复位值： 0x0000 0000

该寄存器只能按字(32 位)访问

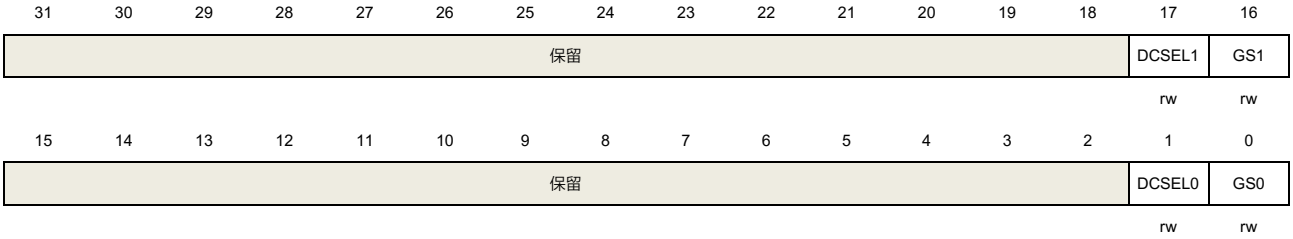


位/位域	名称	描述
31:30	保留	必须保持复位值
29	DDUDR1	DACx_OUT1 DMA 欠载标志位 该位由硬件置位，软件写 1 清零。 0：没有欠载发生 1：发生欠载(DAC 触发产生速度快于 DMA 传输速度)
28:14	保留	必须保持复位值
13	DDUDR0	DACx_OUT0 DMA 欠载标志位 该位由硬件置位，软件写 1 清零。 0：没有欠载发生 1：发生欠载(DAC 触发产生速度快于 DMA 传输速度)
12:0	保留	必须保持复位值

14.4.15. DACx 输出控制寄存器 (DAC_OCTL)

地址偏移：0x80
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。



位/位域	名称	描述

31:18	保留	必须保持复位值
17	DCSEL1	DAC0_OUT1 连接 GPIO 选择 0：DAC 连接外部引脚和片上 CMP 1：当 DAC0_OUT1 输出缓冲器关闭(DBOFF1=1)，DAC 仅独立连接片上 CMP。除此之外，DAC 连接外部引脚和片上 CMP 注意：此位只在 DAC0 有。
16	GS1	DACx_OUT1 输出缓冲器增益选择 0：1x 增益 1：2x 增益
15:2	保留	必须保持复位值
1	DCSEL0	DAC0_OUT0 连接 GPIO 选择 0：DAC 连接外部引脚和片上 CMP 1：当 DAC0_OUT0 输出缓冲器关闭(DBOFF0=1)，DAC 仅独立连接片上 CMP。除此之外，DAC 连接外部引脚和片上 CMP 注意：此位只在 DAC0 有。
0	GS0	DACx_OUT0 输出缓冲器增益选择 0：1x 增益 1：2x 增益

15. 比较器（CMP）

15.1. 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。

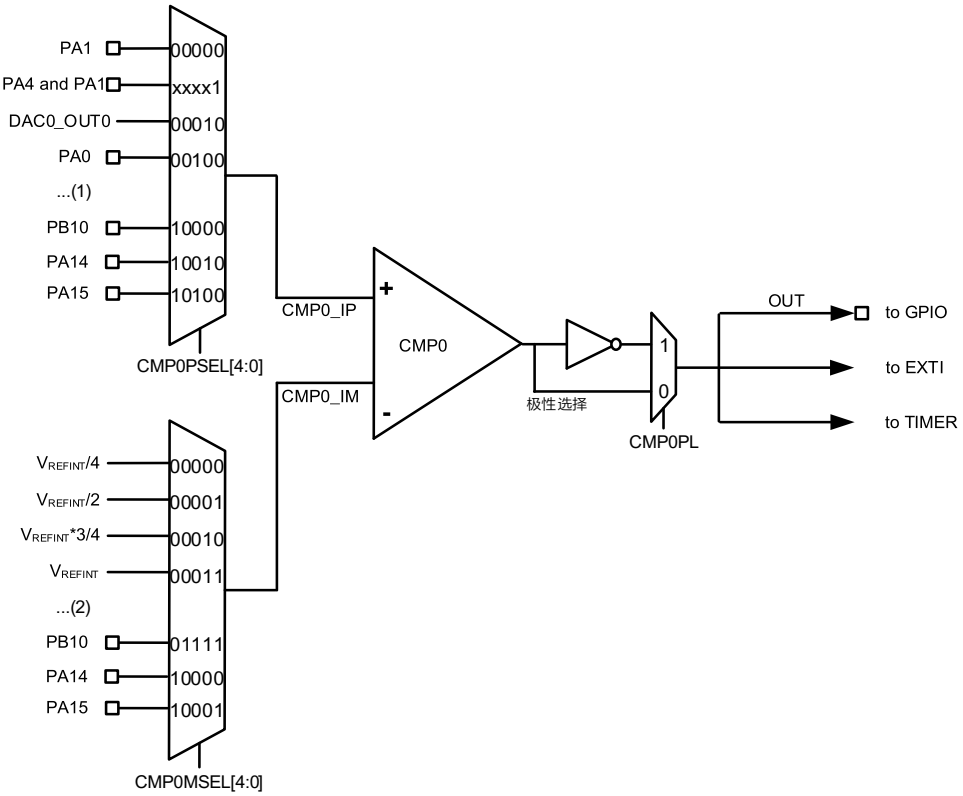
15.2. 主要特征

- 轨对轨比较器；
- 迟滞可配置；
- 速度、功耗可配置；
- 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；
- 输出到 I / O 口；
- 作为触发源输出到定时器；
- 输出到 EXTI。

15.3. 功能描述

比较器的框图展示如下：

图 15-1.比较器框图



（1） 参考 CMPx_CS 寄存器的 CMP0PSEL[4:0]位域;

（2） 参考 CMPx_CS 寄存器的 CMP0MSEL[4:0]位域。

注意：V_{REFINT} 是 1.2V。

15.3.1. 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。

15.3.2. 比较器的 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。

比较器的输出可同时实现内部和外部输出。

参考 Datasheet 的引脚定义，比较器输出可以通过 GPIO 的备用功能连接到对应的 I / O 口。

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 15-1. CMP 的输入和输出总结*详细描述了 CMP 的输入和输出。

表 15-1. CMP 的输入和输出总结

	CMP0
CMP 同相输入连接到 I / O	PA0 PA1 PA2 PA3 PA4 PB0 PB1 PB3 PB10 PA14 PA15
CMP 反相输入连接到内部信号	DAC0_OUT0
CMP 反相输入连接到 I / O	PA0 PA1 PA2 PA3 PA4 PA5 PB0 PB1 PB3 PB10 PA14 PA15
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 DAC0_OUT1
CMP 输出连接到 I / O(异步输出)	PA0
CMP 输出连接到 I / O(同步输出)	PA5
CMP 输出连接到 EXTI	•
CMP 输出连接到内部信号	TIMER0_CH0 TIMER1_CH3 TIMER2_CH0

	CMP0
	TIMER0 OCPRE_CLR TIMER1 OCPRE_CLR TIMER2 OCPRE_CLR
CMP 输出(电机控制保护)	TIMER0 BREAK

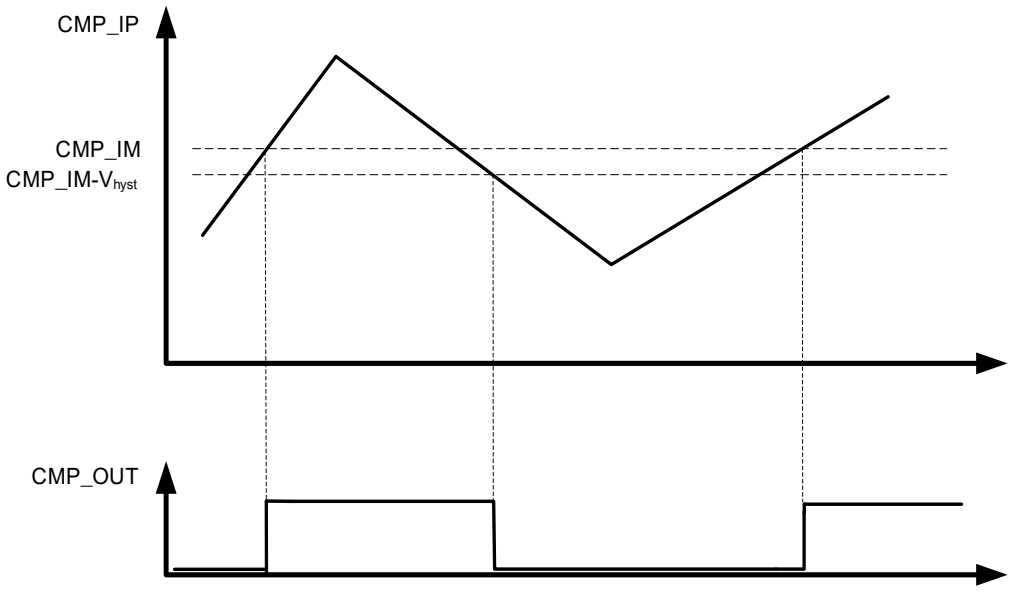
15.3.3. 比较器供电模式

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 CMPx_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。

15.3.4. 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 15-2. 比较器迟滞



15.3.5. 比较器寄存器写保护

比较器的控制状态寄存器(CMPx_CS)可通过设置 CMPxLK 位为 1 来进行保护，CMPx_CS 寄存器，包含 CMPxLK 位，就会变为只读位，只有在 MCU 复位时才可以复位。

15.3.6. 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。

15.3.7. 引脚复位功能

在系统配置寄存器 2(SYSCFG_CFG2)中的 PIN_RSTMD 位控制了寄存器位(包括 CMPxHST[1:0]、CMPxPL、CMPxOSEL[2:0]、CMPxMSEL[4:0]、CMPxPSEL[4:0]、CMPxM[1:0]和 CMPxEN 等)在发生任何复位事件时是否重置。当该位置 1 时，PIN_RSTMD 位和这些寄存器位将在除 POR 外的所有复位事件中保持。

15.4. CMP 寄存器

CMP 基地址：0x4001 001C

15.4.1. CMP 控制状态寄存器(CMPx_CS)

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
31:22	保留	必须保持复位值。
21:18	CMP0PSEL[4:1]	CMP0_IP 输入选择 参考 CMP0PSEL[0]描述。
17:16	CMP0MSEL[4:3]	CMP0_IM 输入选择 参考 CMP0MSEL[2:0]描述。
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：CMPx_CS[6:0]，CMPx_CS[21:8]是可读可写位 1：CMPx_CS[6:0]，CMPx_CS[21:8]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 该位是用于控制 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 CMP0MSEL[2:0]位域和 CMP0MSEL[4:3]位域联合用于选择 CMP0 的输入端 CMP0_IM 的输入源。 00000：V_{REFINT} / 4 00001：V_{REFINT} / 2 00010：V_{REFINT} * 3 / 4 00011：V_{REFINT} 00100：PA4 00101：PA5 00110：PA0 00111：PA2 01000：DAC0_OUT0 01001：DAC0_OUT1 01010：PA1 01011：PA3 01100：PB0 01101：PB1 01110：PB3 01111：PB10 10000：PA14 10001：PA15 其它：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	CMP0PSEL[0]	CMP0_IP 输入选择 CMP0PSEL[0]位和 CMP0PSEL[4:1]位域联合选择 CMP0 的输入端 CMP0_IM 的输入源。 00000：PA1 xxxx1：PA1和PA4(均需要配置成模拟模式) 00010：DAC0_OUT0 00100：PA0 00110：PA2 01000：PA3 01010：PB0 01100：PB1 01110：PB3 10000：PB10 10010：PA14 10100：PA15 其它：保留。
0	CMP0EN	CMP0 使能 0：CMP0 禁能 1：CMP0 使能

16. 可配置逻辑阵列(CLA)

16.1. 简介

可配置逻辑阵列为外部引脚、ADC 和定时器提供 256 个可编程数字逻辑操作，而无需 CPU 干预。本模块实现了四个独立的 CLA 单元。每个 CLA 单元支持 GPIO 引脚的可配置异步和同步输出。

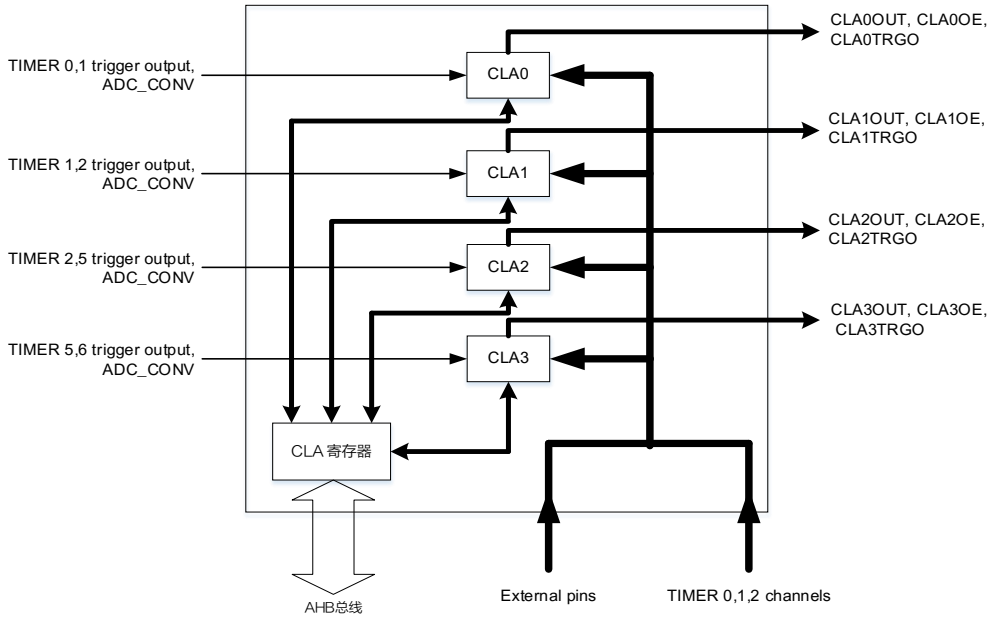
16.2. 主要特性

- 四个独立的 CLA 单元，每个 CLA 单元具有两个信号选择器(SIGS)，支持 16 个输入信号，包括外部引脚、定时器通道、ADC 和 CLA 异步输出；
- 在每个 CLA 单元中实现逻辑配置单元(LCU)，提供 256 个可编程数字逻辑功能；
- 可编程异步和同步输出；
- 可以将 CLA 输出配置为与外部引脚和定时器同步；
- 四个 CLA 单元可以组合并支持复杂的逻辑操作。

16.3. 模块框图

CLA 接口的内部结构如*图 16-1 CLA 模块框图*所示。

图 16-1 CLA 模块框图



16.4. 功能描述

此模块中包含四个相同的CLA单元。每个CLA单元实现了两个信号选择器。此外，每个CLA还包括一个LCU。

16.4.1. CLA 输入信号选择器

每个 CLA 单元包括两个信号选择器：SIGS0 和 SIGS1。每个信号选择器的输入可以为：外部引脚、定时器通道 TIMERx_CHx、定时器触发信号 TIMERx_TRGO、ADC 转换信号 ADC_CONV 和 CLA 单元异步输出 CLAx_ASYNC_OUT。当另一个 CLA 单元的异步输出作为 SIGS 的输入时，可以实现一个复杂的组合逻辑运算。当选择 TIMERx_TRGO 作为 SIGS 的输入时，只有第一个 HCLK 周期有效，其余的 TIMERx_TRGO 高被视为逻辑低。

*表 16-1 CLAxSIGS0 输入选择*和*表 16-2 CLAxSIGS1 输入选择*显示了 CLAxSIGS0 和 CLAxSIGS1 的输入选择。

表 16-1 CLAxSIGS0 输入选择

SIGS0[3:0]	CLA0SIGS0	CLA1SIGS0	CLA2SIGS0	CLA3SIGS0
0000	CLA0_ASYNC_OUT	CLA0_ASYNC_OUT	CLA0_ASYNC_OUT	CLA0_ASYNC_OUT
0001	CLA1_ASYNC_OUT	CLA1_ASYNC_OUT	CLA1_ASYNC_OUT	CLA1_ASYNC_OUT
0010	CLA2_ASYNC_OUT	CLA2_ASYNC_OUT	CLA2_ASYNC_OUT	CLA2_ASYNC_OUT
0011	CLA3_ASYNC_OUT	CLA3_ASYNC_OUT	CLA3_ASYNC_OUT	CLA3_ASYNC_OUT
0100	TIMER1_TRGO	TIMER2_TRGO	TIMER5_TRGO	TIMER6_TRGO
0101	TIMER0_CH0	TIMER0_CH0	TIMER0_CH1	TIMER0_CH2
0110	TIMER0_CH1	TIMER0_CH3	TIMER0_CH3	TIMER1_CH0
0111	TIMER0_CH2	TIMER1_CH0	TIMER1_CH1	TIMER1_CH1
1000	CLAIN0	CLAIN4	CLAIN0	CLAIN2
1001	CLAIN2	CLAIN5	CLAIN1	CLAIN3
1010	CLAIN4	CLAIN8	CLAIN8	CLAIN6
1011	CLAIN6	CLAIN10	CLAIN9	CLAIN7
1100	CLAIN8	CLAIN12	CLAIN14	CLAIN10
1101	CLAIN10	CLAIN13	CLAIN15	CLAIN11
1110	CLAIN12	CLAIN16	CLAIN16	CLAIN18
1111	CLAIN14	CLAIN18	CLAIN17	CLAIN19

表 16-2 CLAxSIGS1 输入选择

SIGS1[3:0]	CLA0SIGS1	CLA1SIGS1	CLA2SIGS1	CLA3SIGS1
0000	CLA0_ASYNC_OUT	CLA0_ASYNC_OUT	CLA0_ASYNC_OUT	CLA0_ASYNC_OUT
0001	CLA1_ASYNC_OUT	CLA1_ASYNC_OUT	CLA1_ASYNC_OUT	CLA1_ASYNC_OUT
0010	CLA2_ASYNC_OUT	CLA2_ASYNC_OUT	CLA2_ASYNC_OUT	CLA2_ASYNC_OUT
0011	CLA3_ASYNC_OUT	CLA3_ASYNC_OUT	CLA3_ASYNC_OUT	CLA3_ASYNC_OUT
0100	ADC_CONV	ADC_CONV	ADC_CONV	ADC_CONV
0101	TIMER0_CH3	TIMER0_CH1	TIMER0_CH0	TIMER0_CH0
0110	TIMER1_CH0	TIMER0_CH2	TIMER0_CH2	TIMER0_CH1

SIGS1[3:0]	CLA0SIGS1	CLA1SIGS1	CLA2SIGS1	CLA3SIGS1
0111	TIMER1_CH1	TIMER1_CH1	TIMER1_CH0	TIMER0_CH3
1000	CLAIN1	CLAIN6	CLAIN2	CLAIN0
1001	CLAIN3	CLAIN7	CLAIN3	CLAIN1
1010	CLAIN5	CLAIN9	CLAIN10	CLAIN4
1011	CLAIN7	CLAIN11	CLAIN11	CLAIN5
1100	CLAIN9	CLAIN14	CLAIN12	CLAIN8
1101	CLAIN11	CLAIN15	CLAIN13	CLAIN9
1110	CLAIN13	CLAIN17	CLAIN18	CLAIN16
1111	CLAIN15	CLAIN19	CLAIN19	CLAIN17

16.4.2. 逻辑控制单元(LCU)控制

在每个 CLA 单元中实现了一个 LCU(逻辑控制单元)，每个 CLA 单元中实现的 LCU 引入了由 CLAx_LCUCTL 寄存器中的 LCUCTL[7:0]位控制的 256 个组合逻辑函数。LCU 有三个输入，在 CLA 单元被禁用时，这些输入对 LCU 的输出没有影响。

- 输入 0(IN0)：SIGS0 的输出
- 输入 1(IN1)：SIGS1 的输出
- 输入 2(IN2)：来自 CLA[x-1]的结果

如果 CLA 单元被禁用，LCU 的输入将被强制设置为'0'。8 比特的 LCUCTL 控制输入 0、输入 1、输入 2 的哪个逻辑功能可以影响输出，如*表 16-3 LCU 控制*所示。

表 16-3 LCU 控制

LCUCTL[7:0]	IN0	IN1	IN2
bit 0	0	0	0
bit 1	0	0	1
bit 2	0	1	0
bit 3	0	1	1
bit 4	1	0	0
bit 5	1	0	1
bit 6	1	1	0
bit 7	1	1	1

例如，如果需要逻辑函数(IN0^IN1^ N2)，CLAx_LCUCTL 中的 LCUCTL[7:0]应该这样配置：当{IN0, IN1, IN2} == 3'b000 时，(IN0^IN1^IN2)的结果为 1'b0，所以 LCUCTL[7:0]的 bit 0 为 0；当{IN0, IN1, IN2} == 3'b001 时，(IN0^IN1^IN2)的结果为 1'b1，所以 LCUCTL[7:0]的 bit 1 为 1；当{IN0, IN1, IN2} == 3'b010 时，(IN0^IN1^IN2)的结果为 1'b1，所以 LCUCTL[7:0]的 bit 2 为 1；当{IN0, IN1, IN2} == 3'b011 时，(IN0^IN1^IN2)的结果为 1'b0，所以 LCUCTL[7:0]的 bit 3 为 0；当{IN0, IN1, IN2} == 3'b100 时，(IN0^IN1^IN2)的结果为 1'b1，所以 LCUCTL[7:0]的 bit 4 为 1；当{IN0, IN1, IN2} == 3'b101 时，(IN0^IN1^IN2)的结果为 1'b0，所以 LCUCTL[7:0]的 bit 5 为 0；当{IN0, IN1, IN2} == 3'b110 时，(IN0^IN1^IN2)的结果为 1'b0，所以 LCUCTL[7:0]的 bit 6 为 0；当{IN0, IN1, IN2} == 3'b111 时，(IN0^IN1^IN2)的结果为 1'b1，所以 LCUCTL[7:0]的 bit 7 为 1。因此，如果要实现逻辑函数(IN0^IN1^IN2)，那么 CLAx_LCUCTL 的 LCUCTL[7:0]

应该被配置为 8'b10010110。

16.4.3. CLA 输出

每个 CLA 单元都有一个 GPIO 引脚作为其输出，该输出可以是 LCU 的结果，也可以是经过触发器后的 LCU 结果。这可以通过 CLAx_CTL 寄存器的 OSEL 位选择。

触发器的时钟源和时钟极性可以分别通过 CLAx_CTL 寄存器的 CSEL[1:0]位和 CPOL 位来选择。有四个时钟可以作为触发器的时钟源，它们分别是：

- CLA[x-1]的结果：前一个 CLA 单元的 LCU 结果(CLA3 的 LCU 结果发送给 CLA0)
- SIGS0：多路选择器 SIGS0 的输出
- HCLK
- TIMER_TRGO： TIMER0_TRGO 对应 CLA0， TIMER1_TRGO 对应 CLA1，TIMER2_TRGO 对应 CLA2，TIMER5_TRGO 对应 CLA3

给寄存器 CLAx_CTL 的位 FFRST 写 1 可以复位触发器的输出。如果触发器的输出作为 CLA 的输出，建议在使能 CLA 单元之前置位 FFRST。

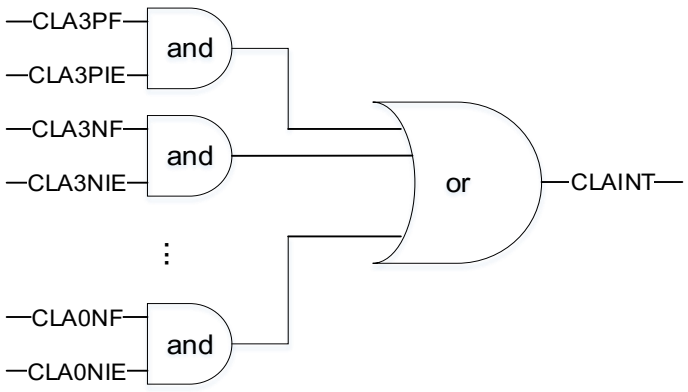
如果 OEN 位复位，那么 CLAxOE 也复位，对应的 GPIO 引脚输出失能。

每个 CLA 单元都有一个 HCLK 同步输出信号 CLAx_TRGO，该信号作为触发信号被发送到 ADC 和 DAC。

16.4.4. 中断

CLA_INTF 寄存器中有八个中断标志位，它们分别是 CLA3PF、CLA3NF、CLA2PF、CLA2NF、CLA1PF、CLA1NF、CLA0PF、CLA0NF。对于每一个标志位，在 CLA_GCTL 寄存器中都有一个对应的中断使能位。CLA 中断逻辑如*图 16-2. CLA 中断逻辑*所示，当检测到标志位置位并且对应的中断使能位使能，就会产生中断。

图 16-2. CLA 中断逻辑



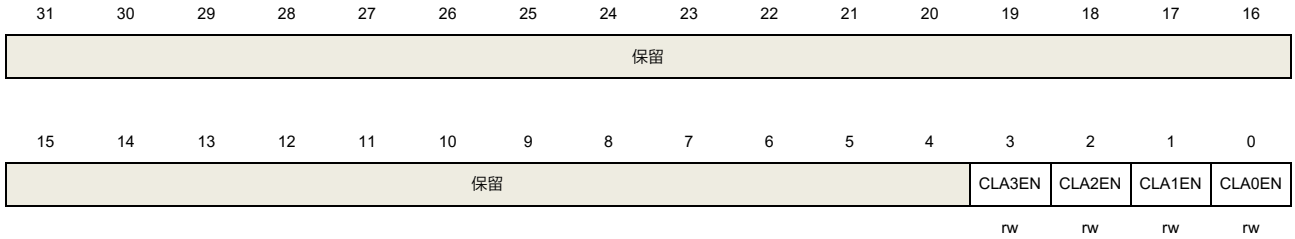
16.5. CLA 寄存器

CLA基地址：0x4003 8000

16.5.1. 全局控制寄存器(CLA_GCTL)

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
31:4	保留	必须保持复位值
3	CLA3EN	CLA3单元使能 该位由软件置位和清除 0：禁能CAL3单元 1：使能CLA3单元
2	CLA2EN	CLA2单元使能 该位由软件置位和清除 0：禁能CAL2单元 1：使能CLA2单元
1	CLA1EN	CLA1单元使能 该位由软件置位和清除 0：禁能CAL1单元 1：使能CLA1单元
0	CLA0EN	CLA0单元使能 该位由软件置位和清除 0：禁能CAL0单元 1：使能CLA0单元

16.5.2. 中断使能寄存器(CLA_INTE)

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

保留

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								CLA3PIE	CLA3NIE	CLA2PIE	CLA2NIE	CLA1PIE	CLA1NIE	CLA0PIE	CLA0NIE
								rw	rw	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	CLA3PIE	CLA3单元上升沿中断使能 该位由软件置位和清除 0：禁能CLA3单元上升沿中断 1：使能CLA3单元上升沿中断。当CLA3PF位置位时，会产生中断。
6	CLA3NIE	CLA3单元下降沿中断使能 该位由软件置位和清除 0：禁能CLA3单元下降沿中断 1：使能 CLA3 单元下降沿中断。当 CLA3NF 位置位时，会产生中断。
5	CLA2PIE	CLA2单元上升沿中断使能 该位由软件置位和清除 0：禁能CLA2单元上升沿中断 1：使能 CLA2 单元上升沿中断。当 CLA2PF 位置位时，会产生中断。
4	CLA2NIE	CLA2单元下降沿中断使能 该位由软件置位和清除 0：禁能CLA2单元下降沿中断 1：使能 CLA2 单元下降沿中断。当 CLA2NF 位置位时，会产生中断。
3	CLA1PIE	CLA1单元上升沿中断使能 该位由软件置位和清除 0：禁能CLA1单元上升沿中断 1：使能 CLA1 单元上升沿中断。当 CLA1PF 位置位时，会产生中断。
2	CLA1NIE	CLA1单元下降沿中断使能 该位由软件置位和清除 0：禁能CLA1单元下降沿中断 1：使能 CLA1 单元下降沿中断。当 CLA1NF 位置位时，会产生中断。
1	CLA0PIE	CLA0单元上升沿中断使能 该位由软件置位和清除 0：禁能CLA0单元上升沿中断 1：使能 CLA0 单元上升沿中断。当 CLA0PF 位置位时，会产生中断。
0	CLA0NIE	CLA0单元下降沿中断使能 该位由软件置位和清除 0：禁能CLA0单元下降沿中断 1：使能 CLA0 单元下降沿中断。当 CLA0NF 位置位时，会产生中断。

16.5.3. 中断标志寄存器(CLA_INTF)

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								CLA3PF	CLA3NF	CLA2PF	CLA2NF	CLA1PF	CLA1NF	CLA0PF	CLA0NF
								rs_w0	rs_w0	rs_w0	rs_w0	rs_w0	rs_w0	rw_w0	rs_w0

位/位域	名称	描述
31:8	保留	必须保持复位值
7	CLA3PF	CLA3单元上升沿标志 该位由硬件置位，软件清除 0：没有检测到CLA3的上升沿输出 1：检测到CLA3的上升沿输出
6	CLA3NF	CLA3单元下降沿标志 该位由硬件置位，软件清除 0：没有检测到CLA3的下降沿输出 1：检测到CLA3的下降沿输出
5	CLA2PF	CLA2单元上升沿标志 该位由硬件置位，软件清除 0：没有检测到CLA2的上升沿输出 1：检测到CLA2的上升沿输出
4	CLA2NF	CLA2单元下降沿标志 该位由硬件置位，软件清除 0：没有检测到CLA2的下降沿输出 1：检测到CLA2的下降沿输出
3	CLA1PF	CLA1单元上升沿标志 该位由硬件置位，软件清除 0：没有检测到CLA1的上升沿输出 1：检测到CLA1的上升沿输出
2	CLA1NF	CLA1单元下降沿标志 该位由硬件置位，软件清除 0：没有检测到CLA1的下降沿输出 1：检测到CLA1的下降沿输出
1	CLA0PF	CLA0单元上升沿标志

		该位由硬件置位，软件清除 0：没有检测到CLA0的上升沿输出 1：检测到CLA0的上升沿输出
0	CLA0NF	CLA0单元下降沿标志 该位由硬件置位，软件清除 0：没有检测到CLA0的下降沿输出 1：检测到CLA0的下降沿输出

16.5.4. 状态寄存器(CLA_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															

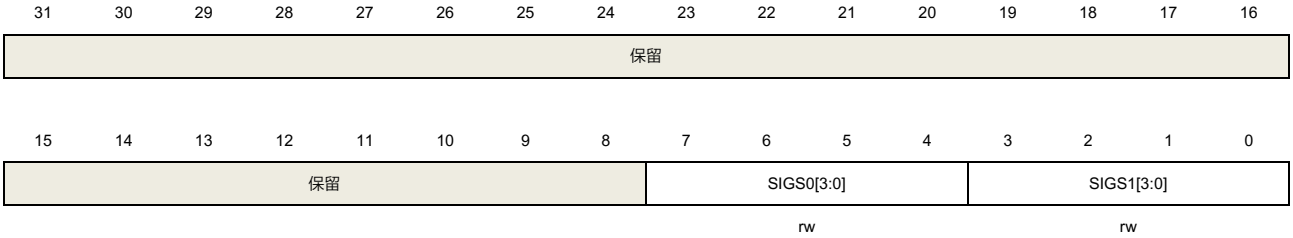
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留												CLA3OUT	CLA2OUT	CLA1OUT	CLA0OUT
												r	r	r	r

位/位域	名称	描述
31:4	保留	必须保持复位值
3	CLA3OUT	CLA3单元输出状态 该位由硬件置位和清除 0：CLA3单元的当前逻辑电平为低 1:：CLA3单元的当前逻辑电平为高
2	CLA2OUT	CLA2单元输出状态 该位由硬件置位和清除 0：CLA2单元的当前逻辑电平为低 1：CLA2单元的当前逻辑电平为高
1	CLA1OUT	CLA1单元输出状态 该位由硬件置位和清除 0：CLA1单元的当前逻辑电平为低 1：CLA1单元的当前逻辑电平为高
0	CLA0OUT	CLA0单元输出状态 该位由硬件置位和清除 0：CLA0单元的当前逻辑电平为低 1：CLA0单元的当前逻辑电平为高

16.5.5. 信号选择寄存器(CLAx_SIGS)(x=0..3)

地址偏移：0x10 + 0x0C * x
复位值：0x0000 0000

该寄存器只能按字(32 位)访问

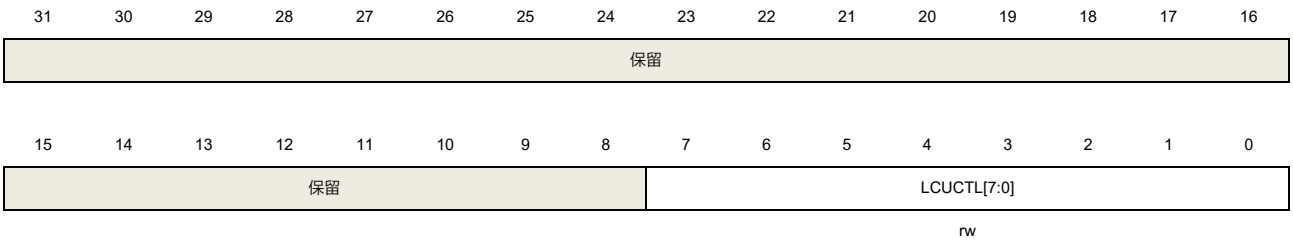


位/位域	名称	描述
31:8	保留	必须保持复位值
7:4	SIGS0[3:0]	信号选择器0输入选择 该位由软件置位和清除 这些位选择信号选择器0的输入。 请参考***表16-1 CLAxSIGS0输入选择***
3:0	SIGS1[3:0]	信号选择器1输入选择 该位由软件置位和清除 这些位选择信号选择器1的输入。 请参考***表16-2 CLAxSIGS1输入选择***

16.5.6. LCU 控制寄存器(CLAx_LCUCTL)(x=0..3)

地址偏移：0x14 + 0x0C * 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
31:8	保留	必须保持复位值
7:0	LCUCTL[7:0]	LCU控制 由软件置位和清除 这些位控制input0、input1、input2的哪个逻辑函数会对输出产生影响。

例如：

IN1 | IN2：LCUCTL = 8'b11101110

IN0 & (IN1 ^ IN2)：LCUCTL = 8'b01100000

16.5.7. 控制寄存器(CLAx_CTL)(x=0..3)

地址偏移：0x18 + 0x0C * 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				SWBEN0	SWBEN1	SWB0	SWB1	OSEL	OEN	保留		FFRST	CPOL	CSEL[1:0]	
				rw	rw	rw	rw	rw	rw			w	rw	rw	

位/位域	名称	描述
31:12	保留	必须保持复位值
11	SWBEN0	使能软件 bit 0 0：禁能软件 bit 0 1：使能软件bit 0
10	SWBEN1	使能软件 bit 1 0：禁能软件 bit 1 1：使能软件bit 1
9	SWB0	软件 bit 0 如果SWBEN0为1，CLAxSIGS0的输出是SWB0的值，否则输出是由CLAxSIGS0选择的输入。
8	SWB1	软件 bit 1 如果SWBEN1为1，CLAxSIGS1的输出是SWB1的值，否则输出是由CLAxSIGS1选择的输入。
7	OSEL	输出选择 由软件置位和清除 0：触发器的输出作为CLAx的输出 1：LCU的结果作为CLAx的输出
6	OEN	输出使能 由软件置位和清除 0：CLAx输出禁能 1：CLAx输出使能
5:4	保留	必须保持复位值
3	FFRST	触发器输出复位

		0：没有作用 1：复位触发器输出异步
2	CPOL	触发器时钟极性选择 由软件置位和清除 当CLAx使能时(CLAxEN = 1)禁止修改该位域 0：时钟上升沿有效 1：时钟下降沿有效
1:0	CSEL[1:0]	触发器时钟源选择 由软件置位和清除 当CLAx使能时(CLAxEN = 1)禁止修改该位域 00：CLA[x-1] LCU的结果 01：SIGS0输出 10：HCLK 11：TIMER_TRGO

17. 内部基准电压(VREF)

17.1. 简介

MCU 有一个精准的内部参考电路，用于为 ADC/DAC 提供基准电压，或由连接到 VREFP 引脚的片外电路使用。

17.2. 主要特性

精准的内部参考特性描述如下：

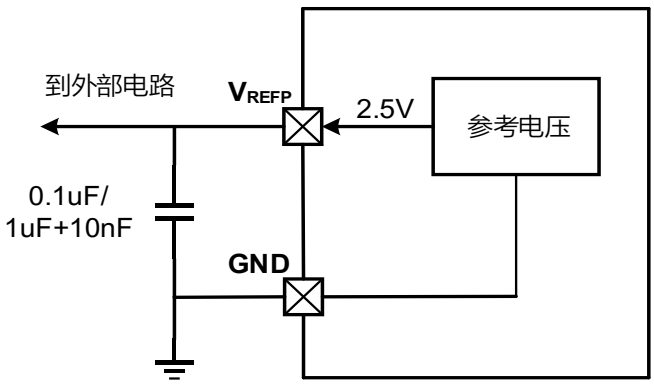
- 电压稳定，产品修整；
- 连接 VREF 引脚至片外电路；
- 提供 2.5 V 参考电压。

17.3. 功能描述

精准的内部参考包括一个控制信号，即 SYSCFG_CFG1 寄存器中的 VREF_EN 位。VREF_EN 用于使能内部参考，产生 2.5V 参考电压并连接到 VREFP 引脚。当 VREF_EN 被禁用时，可将片外参考电压注入到 VREFP 引脚作为 ADC/DAC 的参考源。如果没有 VREFP 引脚(请参阅数据手册)，则 VREF 连接到 VDDA，SYSCFG_CFG1 寄存器中的 VREF_EN 位必须保持 0。

当使用精准的内部参考电压时，建议连接一个 0.1uF(或 1uF)和 10nF 并联的旁路电容，并接地。

图 17-1. VREF 连接



18. 红外接口(IFRP)

18.1. 简介

红外接口(IFRP)用于控制红外光 LED，并发送红外数据实现红外遥控。

该模块没有寄存器，由 TIMER15 和 TIMER16 控制。通过将 GPIO 引脚设置为 IFRP 模式使能模块输出。

18.2. 主要特性

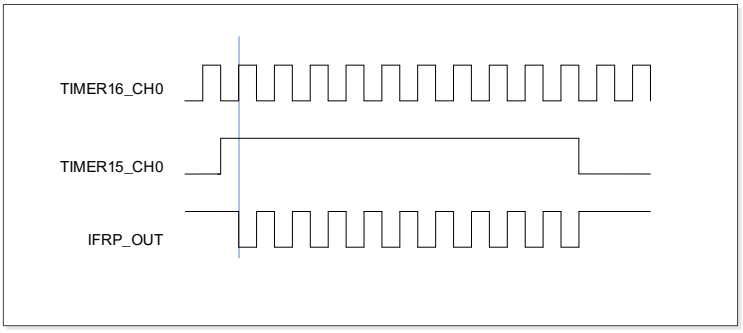
- IFRP 输出信号是由 TIMER15_CH0 和 TIMER16_CH0 决定;
- 为了得到正确的红外线信号，TIMER15 应产生低频调制包络信号，TIMER16 应产生高频载波信号;
- 通过在 SYSCFG_CFG0 中设置 PB9MPEN，使能 IFRP 输出(PB9)来控制 LED 接口。

18.3. 功能描述

IFRP 能够集成 TIMER15 和 TIMER16 的输出，以生成红外线信号。

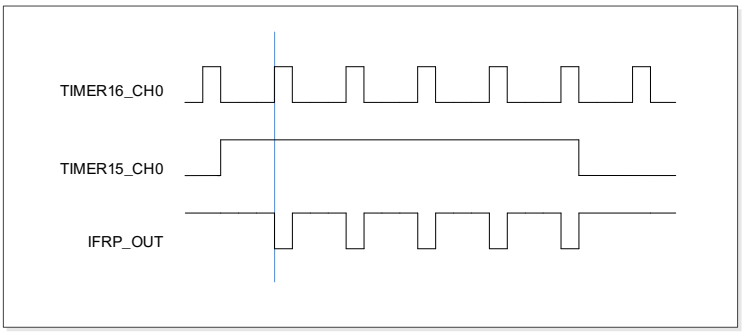
1. 对 TIMER15 的 CH0 进行编程，产生低频 PWM 信号，即调制值信号。对 TIMER16 的 CH0 进行编程，生成高频率 PWM 信号，即载波信号。在产生这些信号之前，信道需要被激活。
2. 设置 GPIO 重映射寄存器并使能 pin。
3. 通过设置 SYSCFG_CFG0 的 PB9MPEN 位来使能 IFRP_OUT 输出到 PB9。

图 18-1. IFRP 输出时序图 1



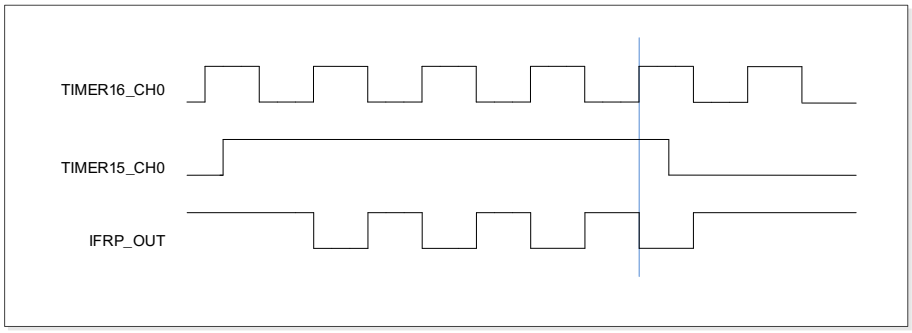
注意：FRP_OUT与TIMER16_CH0相比有一个APB时钟延迟。

图 18-2. IFRP 输出时序图 2



注意：载波(TIMER15_CH0)的占空比可以改变，当TIMER15_CH0占空比较高时，IFRP_OUT与TIMER16_CH0呈反向关系。

图 18-3. IFRP 输出时序图 3



注意：IFRP_OUT将保持TIMER16_CH0的完整性，即使包络信号(TIMER15_CH0)不是活动的。

19. 通用同步异步收发器(USART)

19.1. 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(CK_SYS，PCLK2和CK_IRC8M)时钟进行分频产生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硬件流控操作(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，以实现高速率的数据通信。

19.2. 主要特征

- 自动波特率检测
- NRZ标准格式
- 异步、全双工异步通信
- 半双工单线通信
- 接收FIFO功能
- 双时钟域：
  - 互为异步关系的PCLK和独立于PCLK时钟的USART时钟
  - 不依赖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，最高速度可达9MBits/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，无校验位的生成或检测
  - 产生0.5，1，1.5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，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)，接收FIFO满(RFF)，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，UART1只实现了部分功能，以下特性不被支持。

- 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；
- 同步模式

19.3. 功能描述

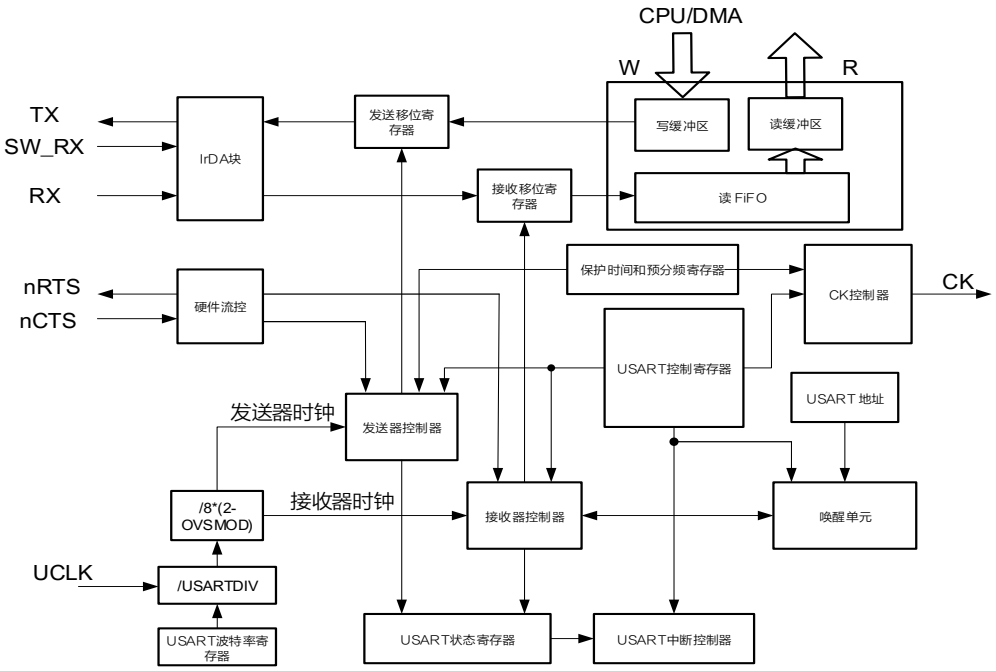
USART 接口通过*表 19-1. USART 重要引脚描述*中主要引脚从外部连接到其他设备。

表 19-1. USART 重要引脚描述

引脚	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

引脚	类型	描述
nRTS	输出	硬件流控模式发送请求信号

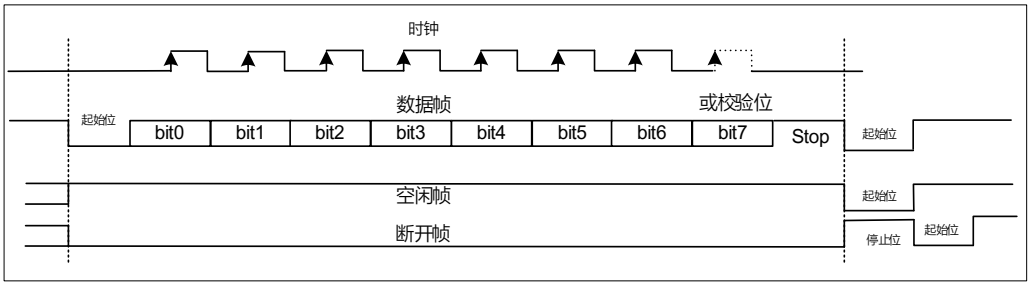
图 19-1. USART 模块内部框图



19.3.1. 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19-2. 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 19-2. 停止位配置

STB[1:0]	停止位长度(位)	功能描述
00	1	默认值
01	0.5	智能卡模式接收

STB[1:0]	停止位长度(位)	功能描述
10	2	标准 USART 和单线模式
11	1.5	智能卡模式发送和接收

在一个空闲帧中，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。

19.3.2. 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)与系统时钟具有如下关系：

如果过采样率是16，公式为：

$$USARTDIV = \frac{UCLK}{16 \times Baud\ Rate} \quad (19\text{-}1)$$

如果过采样是8，公式为：

$$USARTDIV = \frac{UCLK}{8 \times Baud\ Rate} \quad (19\text{-}2)$$

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。

19.3.3. 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。

TEN置位后发送器会发出一个空闲帧。TEN位在数据发送过程中是不可以被复位的。

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

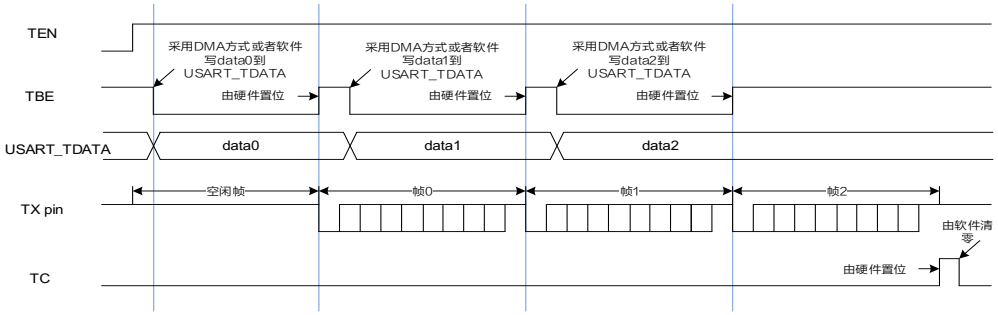
被清0。在数据由USART_TDATA移入移位寄存器后，该位由硬件置1。如果数据在一个发送过程正在进行时被写入USART_TDATA寄存器，它将首先被存入发送缓冲区，在当前发送过程完成时传输到发送移位寄存器中。如果数据在写入USART_TDATA寄存器时，没有发送过程正在进行，TBE位将被清零然后迅速置位，原因是数据被立刻传输到发送移位寄存器。

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。

*图 19-3. USART 发送步骤*给出了 USART 发送步骤。软件操作按以下流程进行：

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. 若DMA未使能，每发送一个字节都需重复步骤7-8；
10. 等待TC=1，发送完成。

图 19-3. 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。通过将USART_INTC寄存器的TCC位置1可以将TC位清零。

当SBKCMD置位时，会发送一个断开帧，发送完成后，SBKCMD将被清0。

19.3.4. 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便开始接收码流。在接收一个数据帧的过程中会

检测噪声错误，奇偶校验错误，帧错误和过载错误。

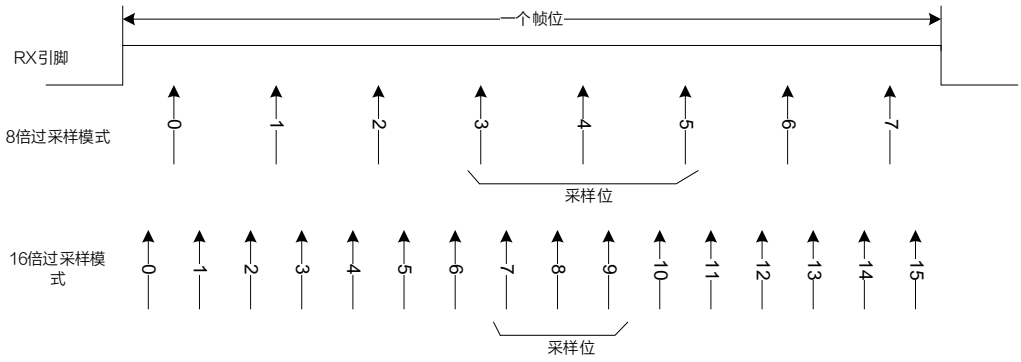
当接收到一个数据帧，USART_STAT寄存器中的RBNE置位，如果设置了USART_CTL0寄存器中相应的中断使能位RBNEIE，将会产生中断。在USART_STAT寄存器中可以观察接收状态标志。

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 19-4. 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位使能奇偶校验功能，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。如果置位了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。根据停止位的配置，有以下几种情形：

- 0.5个停止位：0.5个停止位时，停止位不采样。
- 1个停止位：1个停止位时，在停止位的中间进行采样。
- 1.5个停止位：1.5个停止位时，1.5个停止位可以分为两个部分：0.5个停止位的部分不采样和1个停止位的中间进行采样。
- 2个停止位：2个停止位时，如果在第一个停止位期间检测到帧错误，帧错误标志置位，则第二个停止位不检测帧错误。如果第一个停止位期间没有检测到帧错误，则在第二个停止位继续检测帧错误。

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。如果置位RBNEIE或使能DMA的情况下并置位USART_CTL2寄存器中ERRIE位，将产生中断。

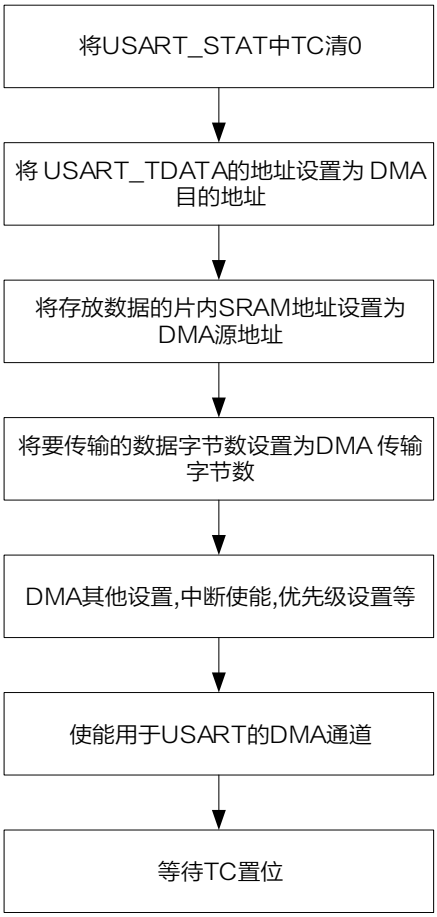
在一个接收过程中，NERR、PERR、FERR、ORERR总是分别和RBNE同时置位。如果没有使能DMA，软件需检查RBNE中断是否由NERR、PERR、FERR或者ORERR置位产生。

19.3.5. 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 19-5. 采用 DMA 方式实现 USART 数据发送配置步骤*所示。

图 19-5. 采用 DMA 方式实现 USART 数据发送配置步骤

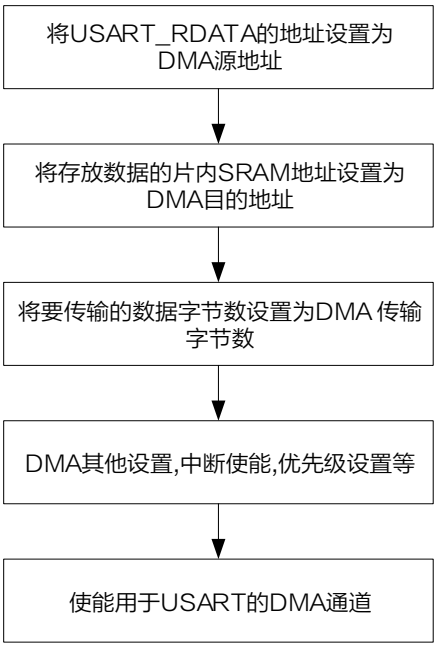


所有数据帧都传输完成后，USART_STAT寄存器中TC位置1。如果USART_CTL0寄存器中TCIE置位，将产生中断。

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 19-6. 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)置

位时将产生中断。

图 19-6. 采用 DMA 方式实现 USART 数据接收配置步骤

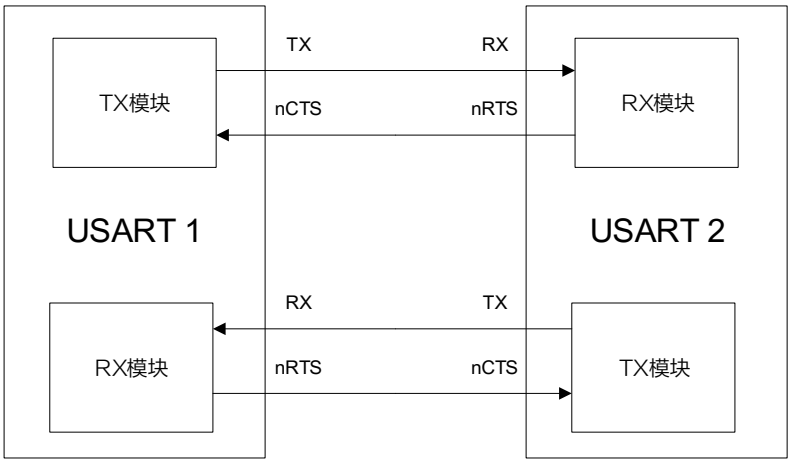


当USART接收到的数据数量达到了DMA传输数据数量，DMA模块将产生传输完成中断。

19.3.6. 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 19-7. 两个 USART 之间的硬件流控制



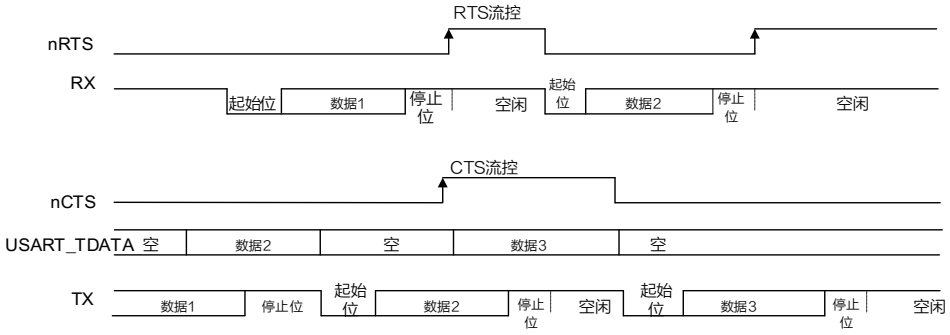
RTS 流控

USART接收器输出nRTS，它用于反映接收缓冲区状态。当一帧数据接收完成，nRTS变成高电平，这样是为了阻止发送器继续发送下一帧数据。当接收缓冲区满时，nRTS保持高电平。

CTS 流控

USART发送器监视nCTS输入引脚来决定数据帧是否可以发送。如果USART_STAT寄存器中TBE位是0且nCTS为低电平，发送器发送数据帧。在发送期间，若nCTS信号变为高电平，发送器将会在当前数据帧发送完成后停止发送。

图 19-8. 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。

19.3.7. 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。如果RWU位为0，RX引脚检测到空闲帧，USART_STAT寄存器中的IDLEF位会置位。如果RWU位置位，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。

当USART_CTL0寄存器中的WM位被设置时，一帧数据的最高有效位(MSB)被检测为地址标志。

如果地址标志为高，则该帧被视为地址帧。如果地址标志为低，则该帧被视为数据帧。如果地址帧的最低有效位(LSB)4位或7位(由USART_CTL1寄存器的ADDM位配置)，与USART_CTL1寄存器中的ADDR位相同，硬件将清除RWU位并退出静音模式。当唤醒USART的帧时，RBNE位将被设置。状态位可在USART_STAT寄存器中获取。如果地址帧的LSB 4/7位与USART_CTL1寄存器中的ADDR位不符，硬件将设置RWU位并自动进入静音模式。在这种情况下，RBNE位不会被设置。

如果ADDM=0，地址帧的LSB 4位将用于与USART_CTL1寄存器中的ADDR[3:0]位进行比较。如果ADDM=1，PCEN=0且WL=1，地址帧的LSB 8位将用于与USART_CTL1寄存器中的ADDR[7:0]位进行比较。如果ADDM=1，PCEN=1且WL=0，地址帧的LSB 6位将用于与USART_CTL1寄存器中的ADDR[5:0]位进行比较。如果ADDM=1，PCEN=1且WL=1，地址帧的LSB 7位将用于与USART_CTL1寄存器中的ADDR[6:0]位进行比较。如果ADDM=1，PCEN=0且WL=0，地址帧的LSB 7位将用于与USART_CTL1寄存器中的ADDR[6:0]位进行比较。

19.3.8. LIN 模式

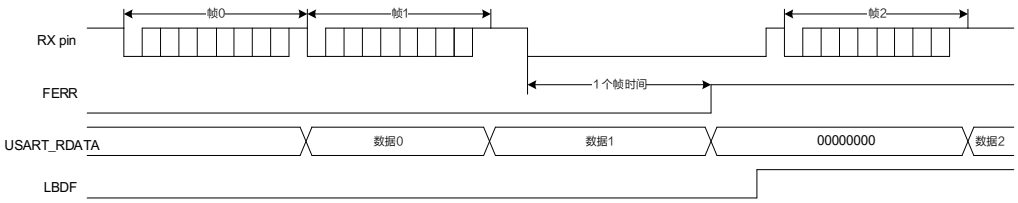
将USART_CTL1寄存器的LMEN置位即可使能本地互联网络模式。在LIN模式下，USART_CTL1寄存器中CKEN，STB[1:0]和USART_CTL2的SCEN，HDEN，IREN位都应该被清0。

在发送一个普通数据帧时，LIN发送过程与普通发送过程相同。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，断开检测可以是在空闲状态下，也可以在数据传输过程中。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。

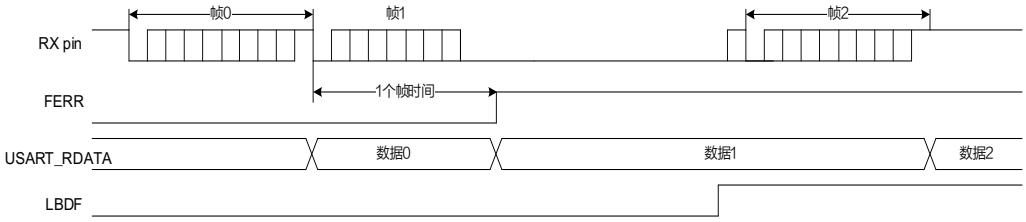
如*图 19-9. 空闲状态下检测断开帧*所示，如果断开帧发生在空闲状态下，USART 接收器会接收到一个全 0 数据帧，同时 FERR 置位。

图 19-9. 空闲状态下检测断开帧



如*图 19-10. 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 19-10. 数据传输过程中检测断开帧



19.3.9. 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。

CK引脚输出波形由USART_CTL1寄存器中CPL，CPH，CLEN位决定。软件仅在USART禁用(UEN=0)时才可以改变它们的值。

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，并无任何过采样。

图 19-11. 同步模式下的 USART 示例

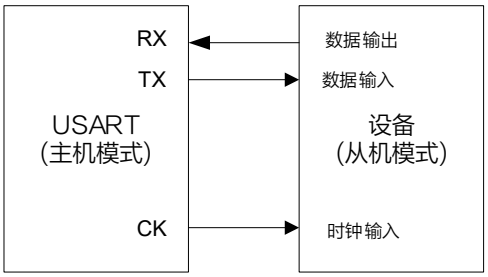
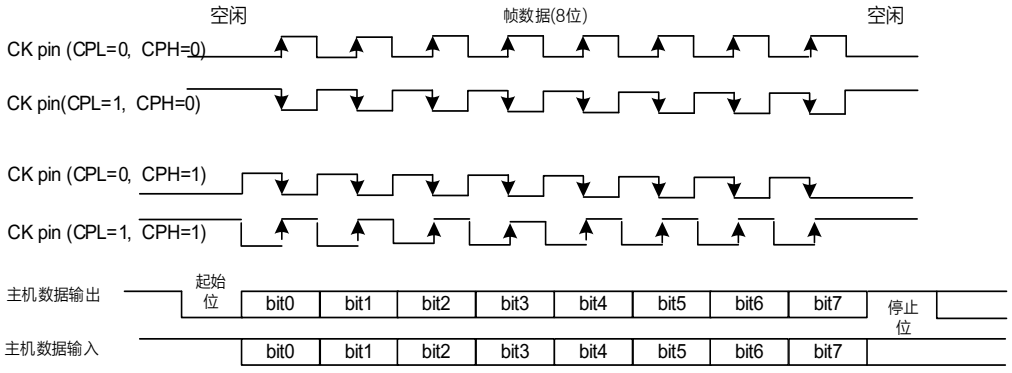


图 19-12. 8-bit 格式的 USART 同步通信波形(CLEN=1)

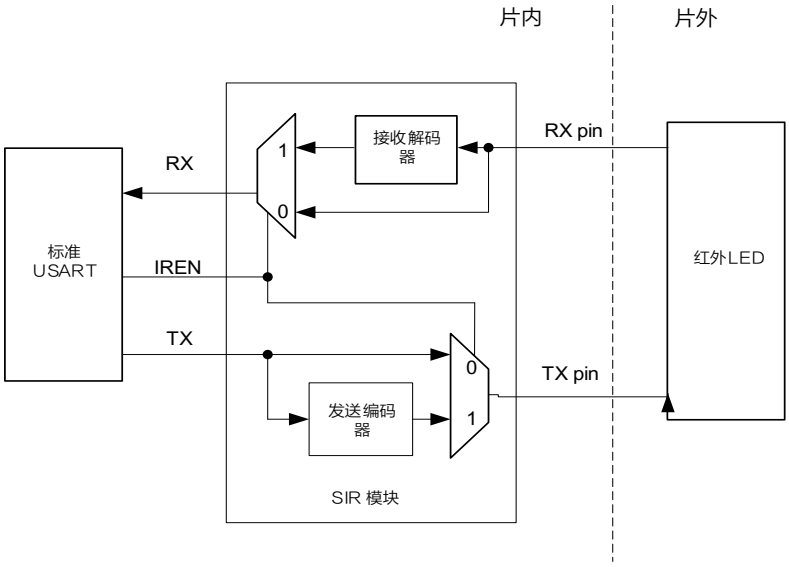


19.3.10. 串行红外(IrDA SIR)编解码功能模块

串行红外编解码功能通过置位USART_CTL2寄存器中IREN使能。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位应被清0。

在IrDA模式下，USART数据帧由SIR发送编码器进行调制，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。对于编码器而言，波特率应小于115200。

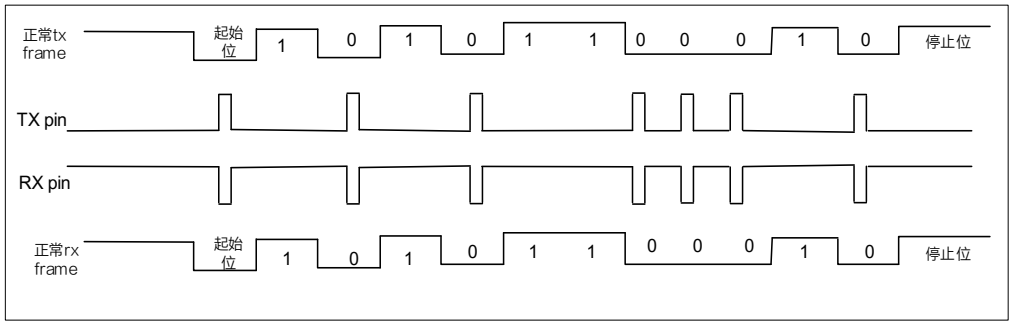
图 19-13. IrDA SIR ENDEC 模块



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 19-14. 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。发送编码器由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，USART_BAUD寄存器需配置16*PSC[7:0]。TX引脚脉冲宽度可以为低功耗波特率的3倍。接收

解码器工作模式与正常IrDA模式相同。

19.3.11. 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位应被清零。

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，TX引脚应被配置为IO管脚。通信冲突应由软件处理。当TEN被置位时，在数据寄存器中的数据将会被发送。

19.3.12. 智能卡(ISO7816-3)模式

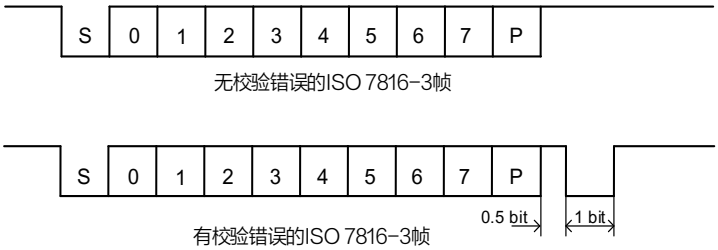
智能卡模式是一种异步通信模式，支持ISO7816-3协议。支持字节模式(T=0)和块模式(T=1)。将USART_CTL2寄存器的SCEN位置1，即可使能智能卡模式。在智能卡模式下，USART_CTL1寄存器的LMEN位和USART_CTL2的HDEN，IREN位应该清0。

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 19-15. 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。在智能卡模式下，在最后一帧数据的停止位之后，内部保护时间计数器将开始计数，GUAT[7:0]的值配置为ISO7816-3协议的CGT减12。在保护时间寄存器向上计数这段时间TC将被强制拉低，当计数达到设定值时，TC被置位。

在USART发送期间，如果检测到有奇偶校验错误，TX引脚在停止位最后一个位时间内被拉低，智能卡发送一个NACK信号。根据协议，USART会自动重发SCRTNUM次。在重发数据帧前面会插入2位的帧间隔。最后一次重发字节后，TC会立即被置位。如果在最大重发次数后仍然收到NACK信号，USART将会停止发送，帧错误标志被置位。USART不会将NACK信号作为起始位。

在USART接收期间，如果在当前数据帧检测到校验错误，TX引脚在停止位的最后一个位时间

内会被拉低。智能卡会接收到NACK信号。然后在智能卡端会产生一个帧错误。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信号和标注这个错误为奇偶校验错误。将USART_CTL2寄存器中的NKEN置位可以使能NACK信号。

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寄存器的RT[23:0]位域设置成BWT(块等待时间)-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之间的某个值(这个时间以波特时间作为单位)，这是为了自动检测两个连续字符之间的最大等待时间。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统计收到的字节数，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[7:0]。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)。为了得到这个值，在收到第四个字节后，会引起一个中断。软件可以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位都应设置为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位都应设置为1。

19.3.13. 自动波特率检测

自动波特率检测功能通过测量接收数据的波特率，自动设置 USART_BAUD 寄存器。通过设置 USART_CTL1 寄存器的 ABDEN 位，可以使能 USART 的自动波特率检测功能。自动波特率检测有两种测量方式，通过 USART_CTL1 寄存器的 ABDM 位域进行配置，检测方式如下：

1. 测量起始位的持续时间(下降沿至上升沿)，在这种情况下，接收模式应该是以 1xxx 位开头的任何字符。
2. 测量起始位和第一位数据位的持续时间，该过程是在下降沿到下降沿之间进行的，这样可以确保在信号斜率较慢的情况下获得更高的准确性。在这种情况下，接收模式应该是以 10xx 位开头的任何字符。

波特率检测完成后，硬件会设置 USART_STAT 寄存器中的 ABDF 位。向 USART_CMD 寄存器中的 ABDCMD 位写入 1 将清除 ABDF 和 ABDE 两位，并启动新的波特率检测过程。

如果检测的通信速率超出范围，硬件会自动设置 USART_STAT 寄存器的 ABDE 位。波特率检测的有效范围是指每位数据的持续时间需落在以下两个区间内：一是 16 至 65536 个时钟周期之间，此时采用 16 倍过采样；二是 8 至 32768 个时钟周期之间，此时采用 8 倍过采样。

19.3.14. ModBus 通信

通过实现块尾检测功能，USART提供实现ModBus/RTU和ModBus/ASCII协议的基本支持。

在ModBus/RTU模式下，通过一个超过2个字符长度的空闲状态来识别块尾。这个功能是通过一个可编程的超时检测功能来实现的。

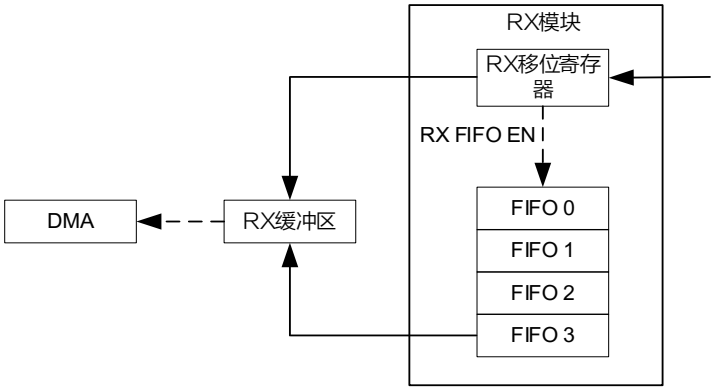
为了检测空闲状态，必须置位USART_CTL1寄存器的RTEN位和USART_CTL0寄存器的RTIE位。USART_RT寄存器必须被设置成与2个字节超时所对应的值。在最后一个停止位被接收后，当接收线在这期间是空闲的，将产生一个中断，通知软件当前块接收已经完成。

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。

19.3.15. 接收 FIFO

通过将USART_RFCS寄存器的RFEN置位使能接收FIFO，可以避免当CPU无法迅速响应RBNE中断时，发生过载错误。接收FIFO和接收缓存区可储存多至5帧的数据。若接收FIFO满，RFFINT位将被置位。如果RFFIE被置位，将产生中断。

图 19-16. USART 接收 FIFO 结构



如果软件在响应RBNE中断时读数据接收缓冲区，在响应开始时，RBNEIE位应清0。当所有接收的数据被读出后，RBNEIE位应置位。在读出接收的数据前，PERR，NERR，FERR，EBF都应被清0。

19.3.16. 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。

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CK_IRC64MDIV。

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是否正在传送数据。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。

19.3.17. USART 中断

USART 中断事件和标志如*表 19-3. USART 中断请求*所示：

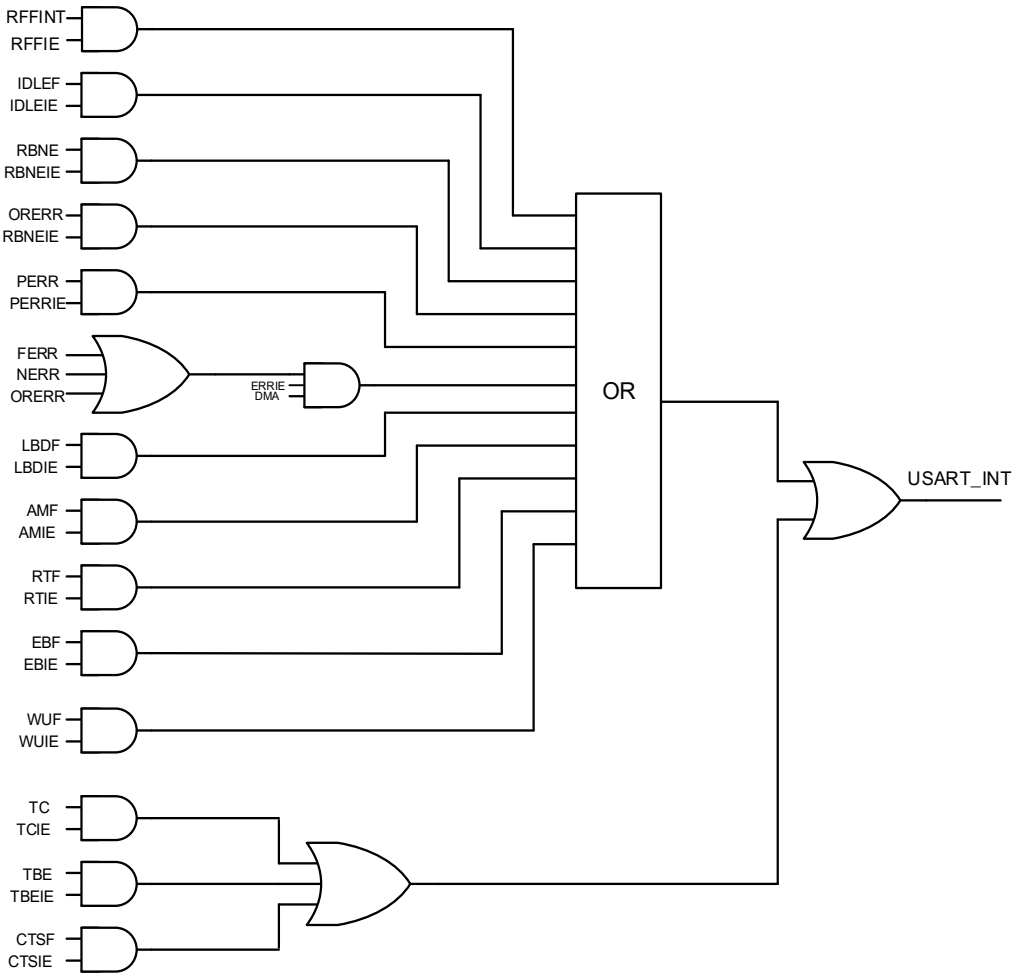
表 19-3. USART 中断请求

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	

中断事件	事件标志	使能控制位
接收FIFO满	RFFINT	RFFIE
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
当DMA接收使能时，接收错误(噪声错误、溢出错误、帧错误)	NERR，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 19-17. USART 中断映射框图



19.4. USART 寄存器

USART0基地址：0x4001 3800
UART1基地址：0x4000 4400

19.4.1. 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UART1中， 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UART1中， 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	WM0	从静默模式唤醒方法，该位和bit[28]位共同决定唤醒方法。 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。 如果该位置1，USART_STAT寄存器中TC被置位时产生中断。 0：发送完成中断禁用。 1：发送完成中断使能。
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。 在UART1中， 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。

19.4.2. 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。

1.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR[7:0]								RTEN	ABDM[1:0]		ABDEN	MSBF	DINV	TINV	RINV
rw								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标记的检测。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UART1，该位保留。
22:21	ABDM[1:0]	自动波特率检测模式。 00：下降沿到上升沿测量(起始位测量，接收到的帧必须遵循以1xxxxxxx格式开头的

		帧格式)。 01：下降沿到下降沿测量(接收到的帧必须遵循以10xxxxxx格式开头的帧格式)。 10：保留 11：保留 USART被禁能(UEN=0)或 ABDEN = 0，该位域才能被改写。 在UART1，该位域保留。
20	ABDEN	自动波特率检测使能。 0：自动波特率检测功能禁用。 1：自动波特率检测功能被使能。 在UART1，该位保留。
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UART1，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UART1，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。 在UART1，该位保留。
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。 在UART1，该位保留。
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。 在UART1，该位保留。
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UART1，该位保留。
5	LBLEN	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UART1，该位保留
4	ADDM	地址检测模式 该位用来选择4位地址检测或全位地址检测。 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。

19.4.3. 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UART1中， 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UART1中， 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UART1中， 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，直到相应的错误标志位被清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UART1中， 该位保留。
4	NKEN	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UART1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UART1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。

19.4.4. 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
保留															

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。 INTDIV = BRR[15:4]。
3:0	BRR [3:0]	波特率分频系数的小数部分。 如果OVSMOD = 0，FRADIV = BRR[3:0]。 如果OVSMOD = 1，FRADIV = BRR[2:0]，BRR[3]必须被置0。

19.4.5. 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UART1中， 该寄存器保留。

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
GUAT[7:0]								PSC[7:0]							
rw								rw							

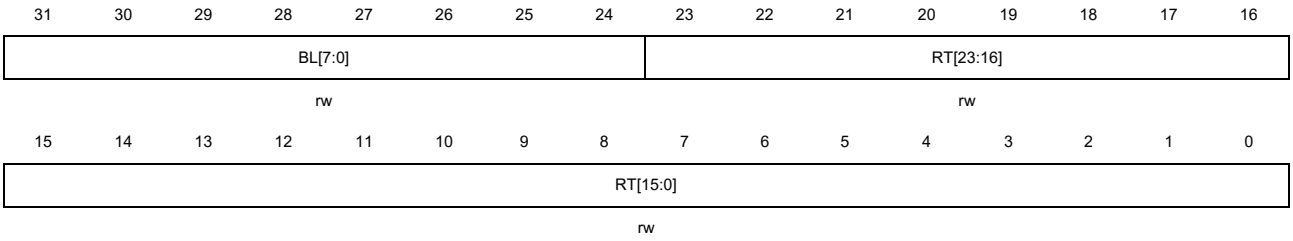
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:8	GUAT[7:0]	在智能卡模式下的保护时间值。 USART被使能(UEN=1)时，该位域不能被改写。
7:0	PSC[7:0]	预分频器值 在红外低功耗模式下，对系统时钟进行分频已获得低功耗模式下的频率。寄存器的值是分频系数。 00000000：保留 - 不设置这个值。 00000001：1分频。 00000010：2分频。 ... 在IrDA正常模式下的分频值。 00000001：仅能设为这个值。 在智能卡模式下，对系统时钟进行分频的值存于PSC[4:0]位域中。PSC[7:5]位保持为复位值。分频系数是寄存器中值的两倍。

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。

19.4.6. 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UART1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。 **注意**：当16倍过采样时，RT[23:0]不能被配置为0xFFFFFF。

19.4.7. 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。

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											TXFCMD	RXFCMD	MMCMD	SBKCMD	ABDCMD
											w	w	w	w	w

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UART3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SBF标志并使USART在空闲时发送一个断开帧。
0	ABDCMD	自动波特率检测请求。 向该位写1将复位USART_STAT寄存器中的ABDF和ABDE标志位，并在接收到下一个数据帧时发出自动波特率测量命令。 在UART1中，该位保留。

19.4.8. 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
ABDF	ABDE	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	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。 在UART1中，该位保留。
21	TEA	发送使能通知标志。 该位反映了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UART3中，该位保留。
19	RWU	接收器从静默模式唤醒。 该位表示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	ABDF	自动波特率检测完成标志。 0：自动波特率检测未完成。 1：自动波特率检测完成。 自动波特率检测完成时由硬件置位。 通过向USART_CMD寄存器的ABDCMD位写1清0，同时在接收到下一帧数据时进行一次新的波特率检测。 在UART3中，该位保留。
14	ABDE	自动波特率检测错误标志。 0：无自动波特率检测错误发生。 1：自动波特率检测错误。 当检测到的波特率超出范围，该位由硬件置位。 通过向USART_CMD寄存器的ABDCMD位写1清0，同时在接收到下一帧数据时进行一次新的波特率检测。 在UART3中，该位保留。
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UART3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UART3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。 在UART3中，该位保留。
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。 在UART3中，该位保留。

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UART3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。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。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。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，仍然没有收到发送成功应答(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。

19.4.9. 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

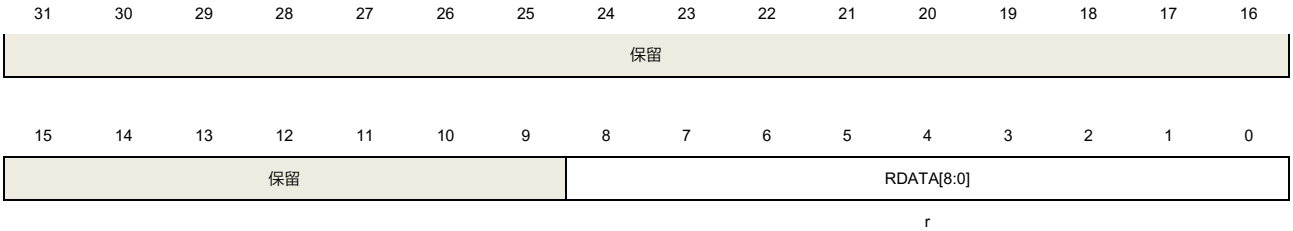
位/位域	名称	描述
31:21	保留	必须保持复位值。
20	WUC	从深度睡眠模式唤醒标志的清除。 向该位写1清除USART_STAT寄存器的WUF位。
19:18	保留	必须保持复位值。
17	AMC	ADDR匹配标志清除。 向该位写1清除USART_STAT寄存器的AMF位。
16:13	保留	必须保持复位值。
12	EBC	块结束标志清除。 向该位写1清除USART_STAT寄存器的EBF位。

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。
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。
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。

19.4.10. 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。



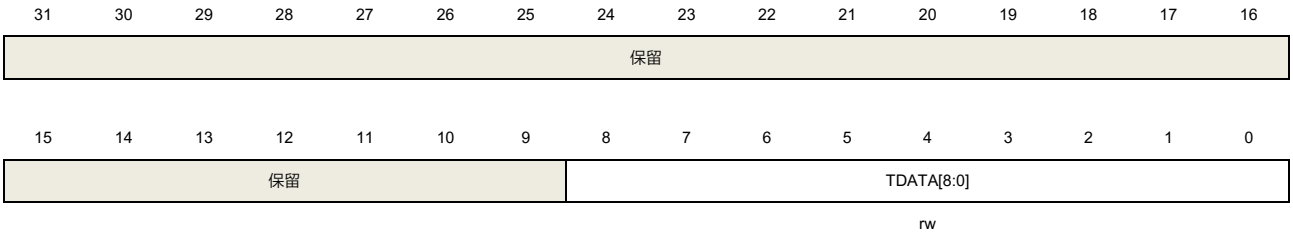
位/位域	名称	描述
31:9	保留	必须保持复位值。
8:0	RDATA[8:0]	接收数据的值。

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。

19.4.11. 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。

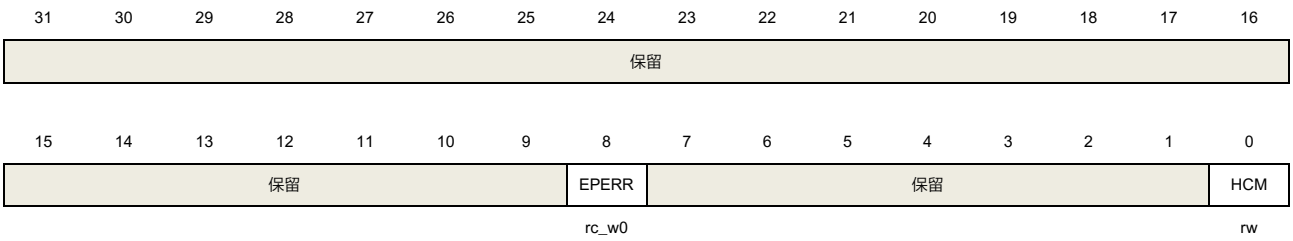


位/位域	名称	描述
31:9	保留	必须保持复位值。
8:0	TDATA[8:0]	发送数据的值。 包含发送的数据字节。 如果发送到的数据打开了奇偶校验位(USART_CTL0寄存器的PCEN置1)，那么发送的数据的最高位(第7位或8位取决于数据的长度)将会被奇偶校验位替代。 只有当USART_STAT寄存器的TBE位被置位时，这个寄存器才可以改写。

19.4.12. USART 兼容性控制寄存器(USART_CHC)

地址偏移：0xC0
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	EPERR	校验错误超前检测标志。 在RBNE置位前，校验位被检测到时该标志置位。 软件写0可以清除该位。

		0：没有检测到校验错误。 1：检测到校验错误。
7:1	保留	必须保持复位值。
0	HCM	硬件流控制兼容性模式。 0：nRTS信号等于RBNE状态寄存器。 1：当最后一个数据位(PCE置位时的奇偶位)被采样时，nRTS信号置位。

19.4.13. USART 接收 FIFO 控制和状态寄存器(USART_RFCS)

地址偏移：0xD0
复位值：0x0000 0400

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
RFFINT	RFCNT[2:0]			RFF	RFE	RFFIE	RFEN	保留							ELNACK
r_w0	r			r	r	rw	rw								rw

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	RFFINT	接收FIFO满中断标志。
14:12	RFCNT[2:0]	接收FIFO计数值。
11	RFF	接收FIFO满标志。 0：接收FIFO不为满。 1：接收FIFO满。
10	RFE	接收FIFO空标志。 0：接收FIFO不为空。 1：接收FIFO空。
9	RFFIE	接收FIFO满中断使能。 0：禁止接收FIFO满中断。 1：使能接收FIFO满中断。
8	RFEN	接收FIFO使能。 当UESM=1，该位置位。 当USART启用(UEN = 1)时，该位字段不能被写入。 0：禁止使用接收FIFO。 1：使能接收FIFO。
7:1	保留	必须保持复位值。

0	ELNACK	若选择了智能卡模式，提前NACK。 如果检测到校验位错误，NACK脉冲提前1/16位的时间。 0：若选择了智能卡模式，禁止提前NACK。 1：若选择了智能卡模式，使能提前NACK。 在UART1中，该位保留。

20. 串行外设接口/片上音频接口(SPI/I2S)

20.1. 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。只有SPI1支持SPI四线主机模式。

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。

20.2. 主要特性

20.2.1. SPI 主要特性

- 具有全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(只有SPI0)。
- 32位宽度，独立的发送和接收FIFO(只有SPI1)。
- 8位或16位数据帧格式(只有SPI0)。
- 4位到16位的数据帧格式(只有SPI1)。
- 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。
- 支持SPI TI模式。
- 支持SPI NSS脉冲模式。
- 支持SPI四线功能的主机模式(只有SPI1)。

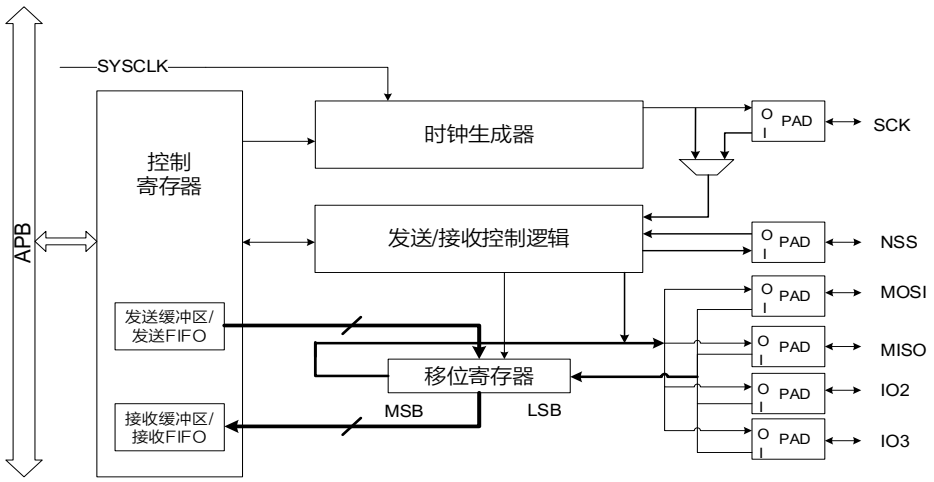
20.2.2. I2S 主要特性

- 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。

20.3. SPI 功能说明

20.3.1. SPI 结构框图

图 20-1. SPI 结构框图



20.3.2. SPI 信号线描述

常规配置(非 SPI 四线模式)

表 20-1. 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。

SPI 四线配置

SPI默认配置为单线模式，当SPI_QCTL中的QMOD位置1时，配置为SPI四线模式(只适用于SPI1)。SPI四线模式只能工作在主机模式。

在SPI四线模式下，SPI通过以下6个引脚与外部设备连接：

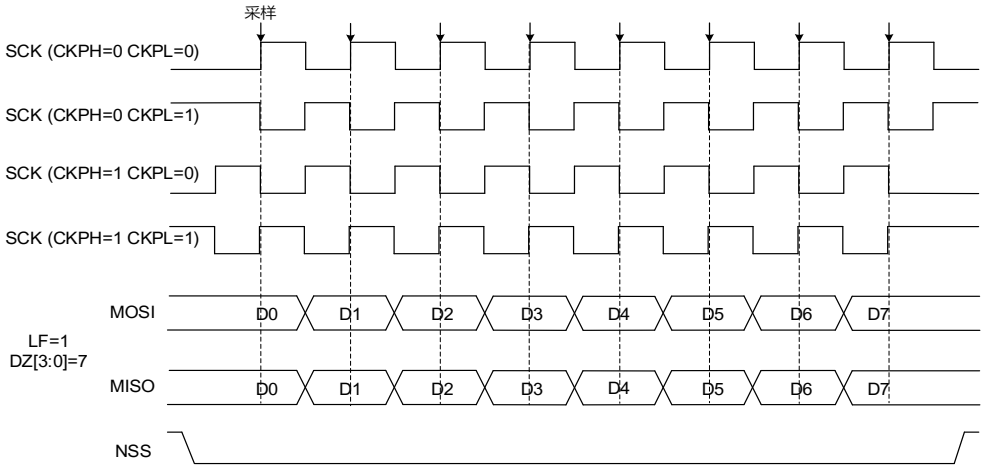
表 20-2. SPI 四线信号描述

引脚名称	方向	描述
SCK	O	SPI 时钟输出
MOSI	I/O	发送或接收数据 0
MISO	I/O	发送或接收数据 1
IO2	I/O	发送或接收数据 2
IO3	I/O	发送或接收数据 3
NSS	O	NSS 输出

20.3.3. 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。在TI模式下，这两位没有意义。

图 20-2. SPI0 常规模式下的时序图



在SPI0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，SPI1先发送LSB位，当LF=0时，则先发送MSB位。在TI模式中，数据顺序固定为先发MSB位。

图 20-3. SPI1 常规模式下的时序图

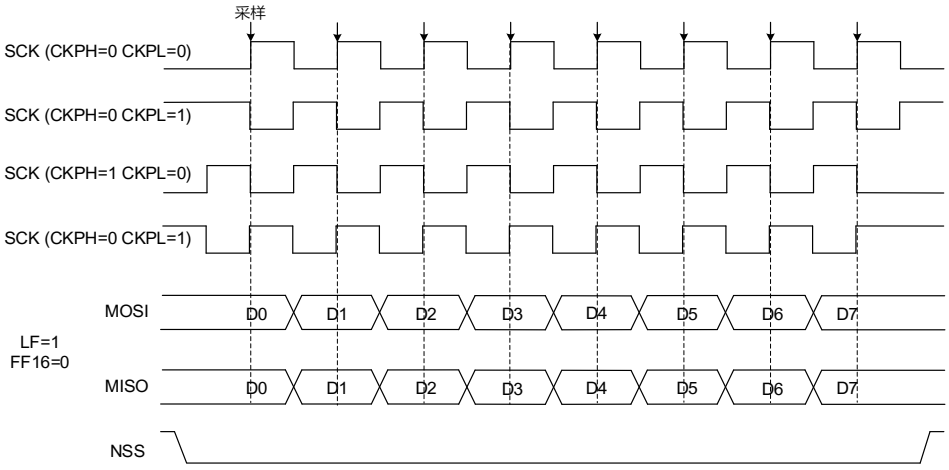
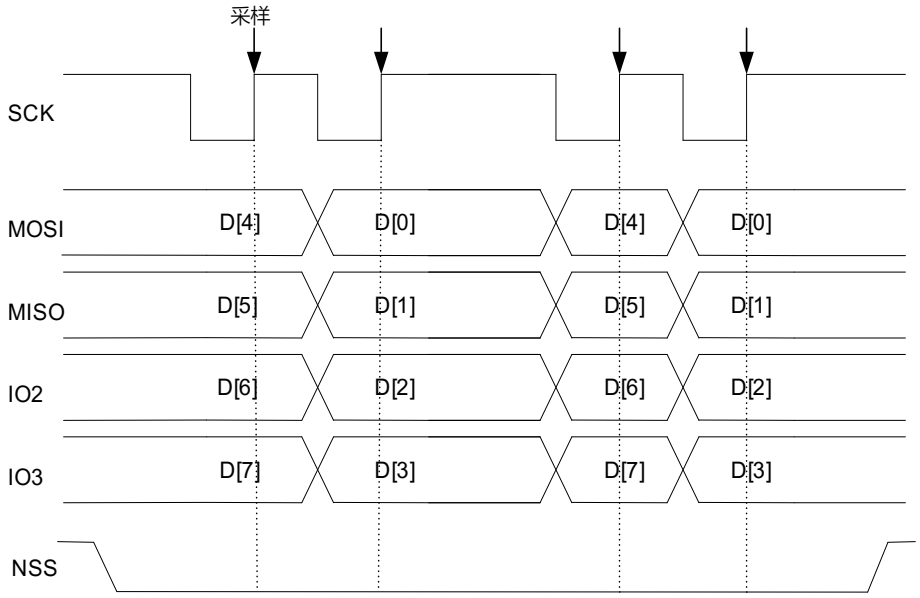


图 20-4. SPI 四线模式下的 SPI 时序图(CKPL=1, CKPH=1, LF=0)



在SPI1常规模式中，通过SPI_CTL1中的DZ[3:0]位域配置数据长度，可以设置为4位至16位。该设置不仅适用于数据的发送也适用于数据的接收。不论设置的数据长度是多少，对FIFO的读访问必须与SPI_CTL1寄存器中的BYTEN位设置的对齐。在SPI四线模式下，数据长度固定为8位。

同样，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。在TI模式中，数据顺序固定为先发MSB位。

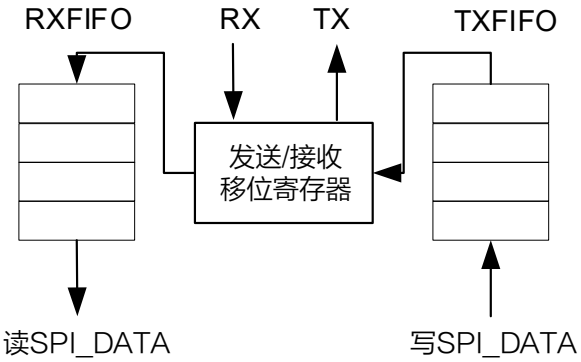
当访问SPI_DATA寄存器时，数据帧总是右对齐成一个字节(如果数据长度小于或等于一个字节)或一个半字。通讯时，只有数据长度内的位会随时钟输出。

20.3.4. 独立发送和接收缓冲区

独立的32位的接收缓冲区(RXFIFO)和发送缓冲区(TXFIFO)分别用于SPI数据传输的不同方向，

它们使得SPI可以连续工作(只适用于SPI1)。

图 20-5. 发送/接收缓冲区



当当前TXFIFO的存储量小于或等于整体存储能力的一半时，TXFIFO被视为空(1)并且此时TBE被硬件置1。当TBE位置位时，向SPI_DATA寄存器写数据，会把数据存入发送FIFO的末尾。当当RXFIFO被视为非空(2)时硬件将RBNE位置1。当RBNE位置位时，从SPI_DATA寄存器读数据，将从接收FIFO获得最早数据。

注意：

(1)对于SPI1，TXFIFO空意味着TXFIFO当前的存储量小于或等于TXFIFO整体存储能力的一半。TXFIFO满的意义与之相反。所以，当数据长度不大于8位时，TXFIFO最多能存储3个数据帧。如果下文出现TXFIFO空或者满，如无特殊说明，意义与这里说明的相同。

(2)对于SPI1，RXFIFO空的意义分为以下两种情况：如果SPI_CTL1中BYTEN位为1时，RXFIFO空意味着当前RXFIFO的存储量小于RXFIFO整体存储能力的四分之一。此时，当数据长度不大于8位时，RXFIFO最多可以存储4个数据帧。如果SPI_CTL1中BYTEN位为0时，RXFIFO空意味着当前RXFIFO的存储量小于RXFIFO整体存储能力的一半。RXFIFO满的意义与之相反。如果下文出现RXFIFO空或者满，如无特殊说明，意义与这里说明的相同。

数据合并(仅适用于 SPI1)

在SPI_CTL1寄存器中DZ[3:0]配置传输数据位宽为8位或者小于8位的情况下，通过配置SPI_CTL1寄存器中BYTEN位为0，开启数据合并传输模式功能。在配置SPI_CTL1寄存器中DZ[3:0]配置传输数据位宽为小于等于8位时，该功能可以实现当对SPI_DATA寄存器进行16位写访问时，两个数据帧的发送是并行方式而不是串行方式。同样的，在接收端接收器通过对SPI_DATA的一次16位读访问，获取这两个数据帧，并且这两帧数据在接收时，仅会产生一个RBNE事件。

注意：当被传输的数据为奇数个字节时，在发送端，需要用8位访问SPI_DATA，发出最后一个数据帧。在接收端，为了产生最后一个字节的RBNE事件，接收器必须在接收最后一个数据帧时，改变BYTEN位。

20.3.5. 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 20-3. 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 20-4. 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

20.3.6. SPI 运行模式

表 20-5. 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
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

模式	描述	寄存器配置	数据引脚用法
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 20-6. 典型的全双工模式连接

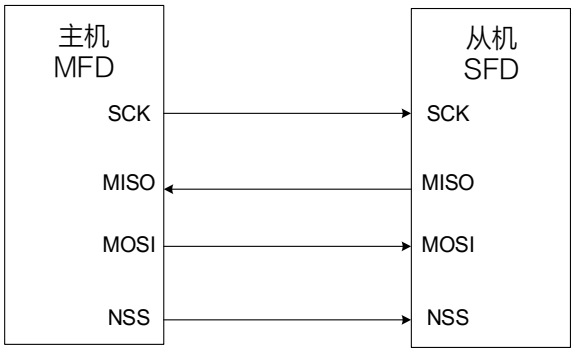


图 20-7. 典型的单工模式连接(主机：接收，从机：发送)

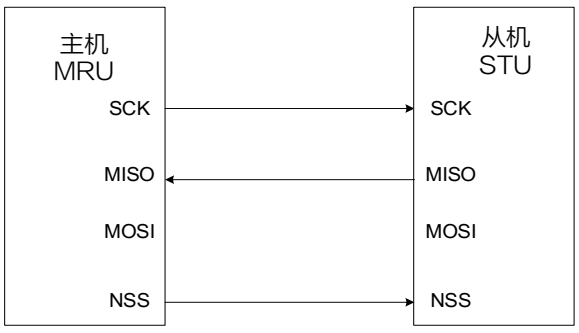


图 20-8. 典型的单工模式连接(主机：只发送，从机：接收)

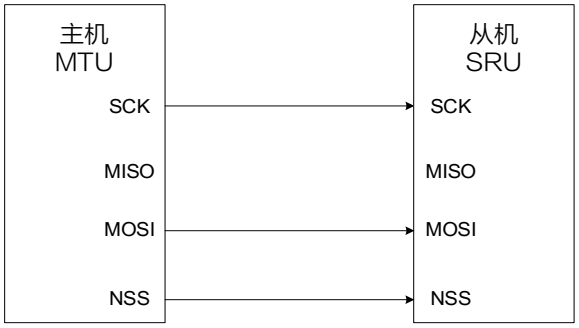
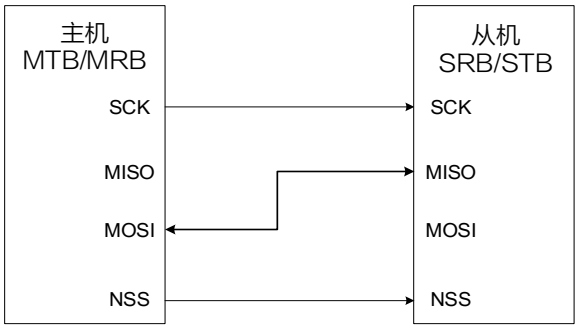


图 20-9. 典型的双向线连接



SPI 初始化流程

SPI1：

1. 如果工作在主机模式或从机TI模式，配置SPI_CTL0中的PSC[2:0]位来生成预期波特率的SCK信号，或配置TI模式下的Td时间。否则，忽略此步骤。
2. 配置时钟时序(SPI_CTL0中的CKPL位和CKPH位)。
3. 配置帧格式(SPI_CTL0中的LF位)。
4. 配置数据格式(SPI_CTL1中的DZ[3:0]位域)和SPI_DATA的访问方式(SPI_CTL1中的BYTEN)。
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. 如果工作在TI模式，需要将SPI_CTL1中的TMOD位置1，否则，忽略此步骤。
7. 如果工作在 NSSP 模式，需要将 SPI_CTL1 中的 NSSP 位置 1，否则，忽略此步骤。
8. 根据***表20-5. SPI运行模式***，配置MSTMOD位、RO位、BDEN位和BDOEN位。
9. 根据应用程序的需求，配置TXDMA_ODD和RXDMA_ODD位。
10. 如果工作在SPI四线模式，需要将SPI_QCTL中的QMOD位置1，如果不是，则忽略此步骤。
11. 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、DZ[3:0]位。

SPI0：

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或从机TI模式，配置SPI_CTL0中的PSC[2:0]位来生成预期波特率的SCK信号，或配置TI模式下的Td时间。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. 如果工作在TI模式，需要将SPI_CTL1中的TMOD位置1，否则，忽略此步骤。
7. 如果工作在NSSP模式，需要将SPI_CTL1中的NSSP位置1，否则，忽略此步骤。
8. 根据***表20-5. SPI运行模式***，配置MSTMOD位、RO位、BDEN位和BDOEN位。
9. 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。

SPI1:

1. 如果工作在主机模式或从机TI模式，配置SPI_CTL0中的PSC[2:0]位来生成预期波特率的SCK信号，或配置TI模式下的Td时间。否则，忽略此步骤。
2. 配置时钟时序(SPI_CTL0中的CKPL位和CKPH位)。
3. 配置帧格式(SPI_CTL0中的LF位)。
4. 配置数据格式(SPI_CTL1的DZ[3:0]位域)和SPI_DATA的访问方式(SPI_CTL1的BYTEN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. 如果工作在TI模式，需要将SPI_CTL1中的TMOD位置1，否则，忽略此步骤。
7. 如果工作在 NSSP 模式，需要将 SPI_CTL1 中的 NSSP 位置 1，否则，忽略此步骤。
8. 根据***表20-5. SPI运行模式***置MSTMOD位、RO位、BDEN位和BDOEN位。
9. 根据应用程序的需求，配置TXDMA_ODD和RXDMA_ODD位。
10. 如果工作在SPI四线模式，需要将SPI_QCTL中的QMOD位置1，如果不是，则忽略此步骤。
11. 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、DZ[3:0]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/发送FIFO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/发送FIFO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/发送FIFO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/发送FIFO空)位置1。TBE标志位置1，说明发送缓冲区/发送FIFO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/接收FIFO，且RBNE(接收缓冲区/接收FIFO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，仅当发送缓冲区/发送FIFO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(非 SPI 四线模式，TI 模式或 NSSP 模式)

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/接收FIFO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 TI 模式

SPI TI模式将NSS作为一种特殊的帧头标志信号，它的操作流程与上文描述的常规模式类似。上文描述的模式(MFD，MTU，MRU，MTB，MRB，SFD，STU，SRU，STB和SRB)都支持TI模式。但是，在TI模式中，SPI_CTL0中的CKPL位和CKPH位是没有意义的，SCK信号的采样边沿为下降沿。

图 20-10. 主机 TI 模式在不连续发送时的时序图

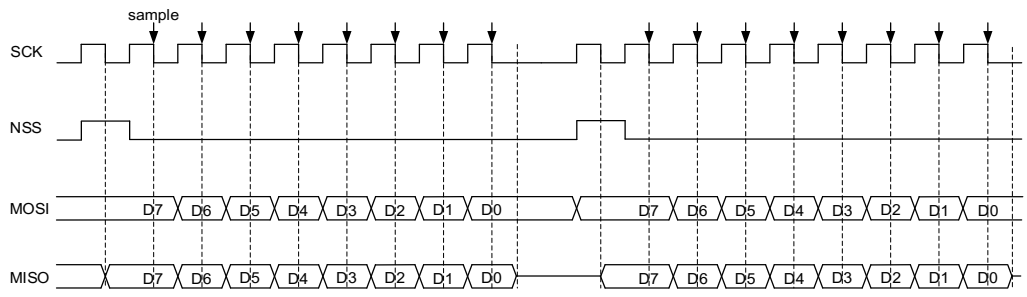
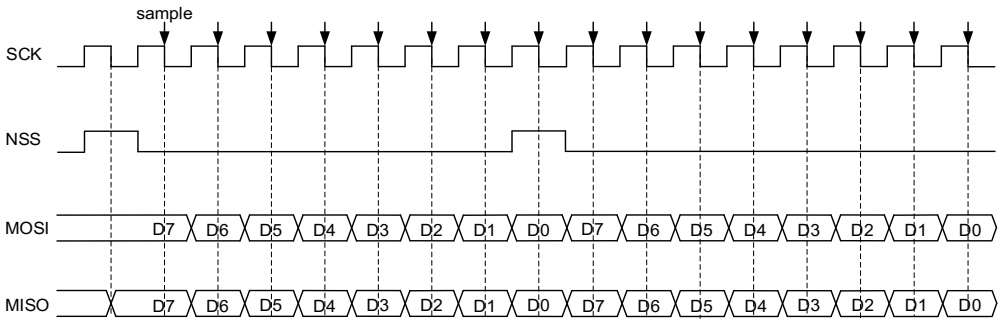
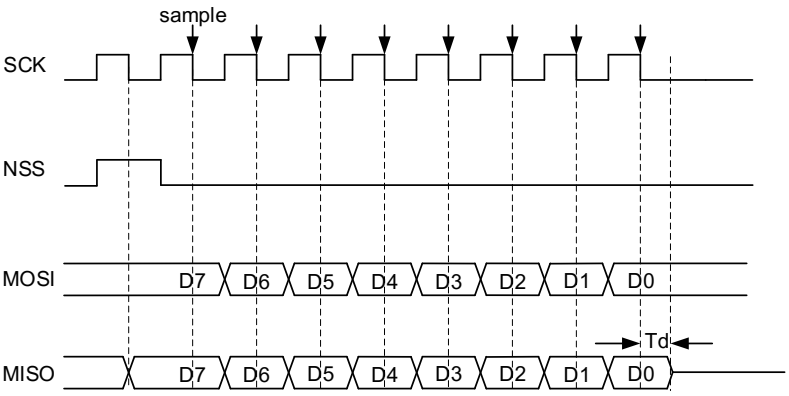


图 20-11. 主机 TI 模式在连续发送时的时序图



在主机TI模式下，SPI模块可实现连续传输或者不连续传输。如果主机写SPI_DATA的速度很快，那么就是连续传输，否则，为不连续传输。在不连续传输中，在每个字节传输前需要一个额外的时钟周期。在连续传输中，额外的时钟周期只存在于第一个字节之前，随后字节的起始时钟周期被前一个字节的最后一位的时钟周期覆盖。

图 20-12. 从机 TI 模式时序图



在从机TI模式中，在SCK信号的最后一个上升沿，从机开始发送最后一个字节的LSB位，在半位的时间之后，主机开始采集数据。为了确保主机采集到正确的数据，在释放该引脚之前，从机需要在SCK信号的下降沿之后继续驱动该位一段时间，这段时间称为T_d，T_d通过SPI_CTL0寄存器中的PSC[2:0]位来设置。

$$T_d = \frac{T_{bit}}{2} + 5 * T_{pclk} \quad (20\text{-}1)$$

例如，如果PSC[2:0] = 010，那么T_d数值为$9*T_{pclk}$。

在从机模式下，从机需要监视NSS信号，如果检测到错误的NSS信号，将会置位FERR标志位。例如，NSS信号在一个字节的中间位发生翻转。

NSS 脉冲模式操作流程

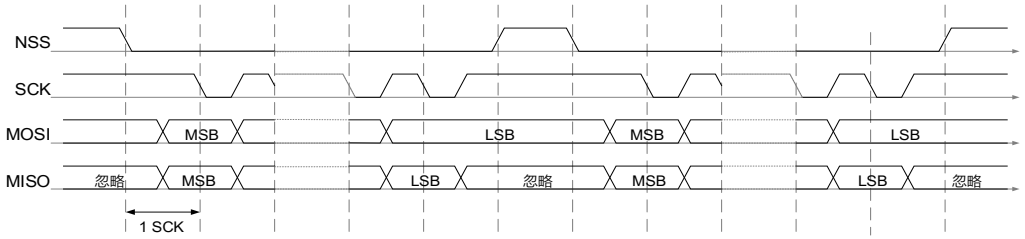
配置SPI_CTL1寄存器中的NSSP位使能该功能，为了确保使用该功能实现，需满足以下几个条件：配置设备为主机模式，使用普通SPI协议的数据帧格式，同时在第一个时钟跳变沿采样数据。

总之：MSTMOD = 1，NSSP = 1，CKPH = 0。

当使用NSS脉冲模式时，根据内部数据发送缓冲区/发送FIFO的状态，NSS脉冲会在两个连续的数据帧之间产生，且持续时间至少为1个SCK时钟周期。如果数据发送缓冲区/发送FIFO保持为空，可能会持续多个SCK时钟周期。NSS脉冲功能专为单一的主从应用设计，支持从机锁存数据。

下图描述了NSS脉冲模式在主机连续发送时的时序图。

图 20-13. NSS 脉冲模式时序图(主机连续发送)



SPI 四线模式操作流程

SPI四线模式用于控制四线SPI flash外设。

要配置成SPI四线模式，首先要确认TBE位置1，且TRANS位清零，然后将SPI_QCTL寄存器中的QMOD位置1。在SPI四线模式，SPI_CTL0寄存器中BDEN位、BDOEN位、CRCEN位、CRCNT位、CRCNT位、RO位和LF位保持清零，DZ[3:0]位域配置数据长度为8位，且MSTMOD位置1，以保证SPI工作于主机模式。SPIEN位、PSC位、CKPL位和CKPH位根据需要进行配置。

SPI四线模式有两种运行模式：四线写模式和四线读模式，通过SPI_QCTL寄存器中的QRD位进行配置。

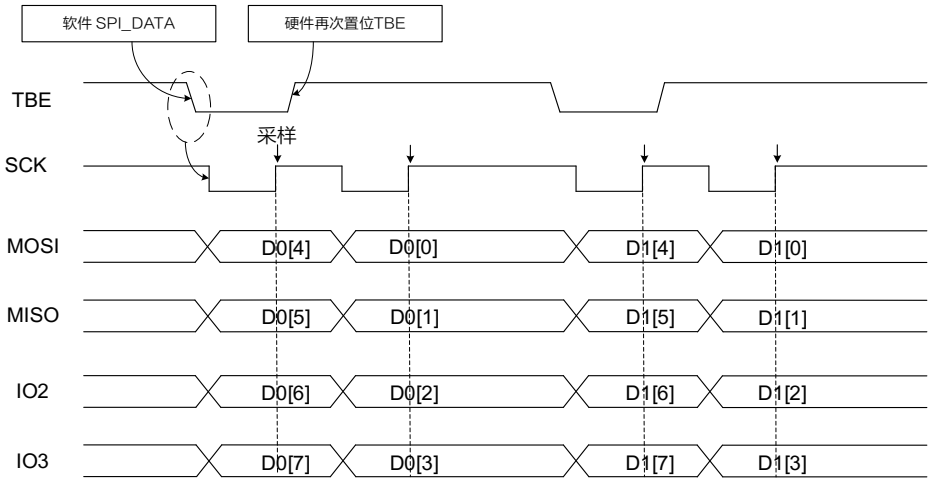
四线写模式

当SPI_QCTL寄存器中的QMOD位置1且QRD位清零时，SPI工作在四线写模式。在四线写模式中，MOSI、MISO、IO2和IO3都用作输出引脚，在SCK产生时钟信号后，一旦数据写入SPI_DATA寄存器(TBE位清零)且SPIEN位置1时，将会通过这四个引脚发送写入的数据。SPI开始数据传输之后，每发送一个数据帧都要检测TBE标志位，若不能满足条件则停止传输。

四线模式下发送操作流程：

1. 根据应用需求，配置SPI_CTL0和SPI_CTL1中的时钟预分频、时钟极性、相位等参数；
2. 将SPI_QCTL中的QMOD位置1，然后将SPI_CTL0中的SPIEN位置1来使能SPI功能；
3. 向SPI_DATA寄存器中写入一个字节的数据，TBE标志位将会清零；
4. 等待硬件将TBE位重新置位，然后写入下一个字节数据。

图 20-14. SPI 四线模式四线写操作时序图



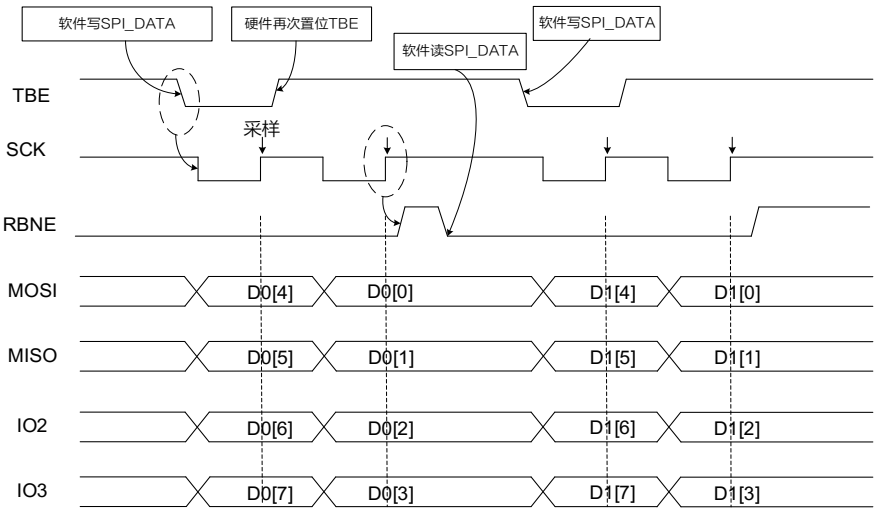
四线读模式

当SPI_QCTL寄存器中的QMOD位和QRD位都置1时，SPI工作在四线读模式。在四线读模式中，MOSI、MISO、IO2和IO3都用作输入引脚，一旦数据写入SPI_DATA寄存器(TBE位清零)且SPIEN位置1时，在SCK信号线产生时钟信号。写数据到SPI_DATA寄存器只是为了产生SCK时钟信号，所以可以写入任何数据。SPI开始数据传输之后，每发送一个数据帧都要检测SPIEN位和TBE位，若条件不满足则停止传输。所以软件需要一直向SPI_DATA写空闲数据，以产生SCK时钟信号。

四线模式下接收操作流程：

1. 根据应用需求，配置SPI_CTL0和SPI_CTL1中时钟预分频、时钟极性、相位等参数；
2. 将SPI_QCTL中的QMOD位和QRD位置1，然后将SPI_CTL0中的SPIEN位置1来使能SPI功能；
3. 写任意数据(例如0xFF)到SPI_DATA寄存器；
4. 等待RBNE位置1，然后读SPI_DATA寄存器来获取接收的数据；
5. 写任意数据(例如0xFF)到SPI_DATA寄存器，以接收下一个字节数据。

图 20-15. SPI 四线模式四线读操作时序图



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

SPI1：

等待TXLVL[1:0]=00和TRANS=0，接着通过清零SPIEN位关闭SPI。最后，读取数据直到RXLVL[1:0]=00。

SPI0：

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

SPI1：

等待TXLVL[1:0]=00和TRANS=0，接着通过清零SPIEN位关闭SPI。

SPI0：

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

SPI1：

应用程序可以在任何时候关闭SPI功能，然后等待TRANS=0，读取数据直到RXLVL[1:0]=00。

SPI0：

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

SPI1：

应用程序可以在任何时候关闭SPI功能，然后等待TRANS=0，读取数据直到RXLVL[1:0]=00。

SPI0：

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。

TI模式

TI模式的停止流程与上面描述过程相同。

NSS脉冲模式

NSS脉冲模式的停止流程与上面描述过程相同。

SPI四线模式

在禁用SPI四线模式和关闭SPI功能之前，软件应该先检查：TBE位置1，TRANS位清零，SPI_QCTL中的QMOD位和SPI_CTL0中的SPIEN位清零。

20.3.7. 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。

DMA 数据合并传输(只有 SPI1)

采用DMA进行数据传输，当BYTEN设置为0且DZ[3:0]配置的数据长度小于或等于8位且数据合并模式使能时，DMA将会以16位方式访问SPI_DATA寄存器，自动完成数据的发送。

在数据合并模式使能且传输数据帧的帧数不是偶数倍的情况下，为了避免最后一次DMA传输多一帧数据的问题，需要将SPI_CTL1寄存器中TXDMA_ODD/RXDMA_ODD位为设置为1。

20.3.8. 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。

对于SPI0，如果是8位数据长度，CRC计算基于CRC8标准进行。如果是16位数据长度，CRC计算基于CRC16标准进行。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。

对于SPI1，只有数据长度为8位或者16位时，SPI提供CRC计算，且独立于数据长度，可以固定设置为8位或16位CRC计算。对于其他所有的数据长度，CRC无效。CRC数据交换，通常需

要在数据序列结束后，再占用一个或多个数据通信的时间。例如，当设置为8位的数据长度并做16位CRC检查时，发送完整的CRC数据就要两帧。如果使能了DMA功能，硬件将会自动处理CRC传输和校验，但SPI需设置DMA发送通道和接收通道的计数器值。发送DMA计数器值为不包括CRC帧的数据帧的数量。接收DMA计数器值的配置如下：

1.全双工模式：假设SPI接收的数据量为L，当CRCL = 0且DZ = 8时，，则DMA接收通道的计数值等于L + 1，否则DMA接收通道的计数值等于L + 2。

2.只接收模式：DMA接收通道计数值只等于接收的数据量。接收数据完成后，通过软件读取SPI_RCRC寄存器的方式获取CRC值。

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。

当配置SPI为从模式并且使用CRC功能时，即使NSS引脚为高时仍然会执行CRC的计算(当NSS信号为高时，只要SCK引脚上有时钟脉冲，则CRC计算会继续执行)。当主设备交替地与多个从设备进行通信时，将会出现这种情况,此时建议在NSS信号为低时重启CRC功能)。当从设备未选中(NSS信号为高)转换到被选中为一个新的从设备(NSS信号为低)的时候，为了保持主从设备端下次CRC计算结果的同步，应该清除主从两端的CRC数值。建议按照下述步骤清除CRC数值：

1. 关闭SPI模块(SPIEN=0);
2. 清除CRCEN位(CRCEN=0);
3. 设置CRCEN位(CRCEN=1);
4. 使能SPI模块(SPIEN=1)。

20.3.9. SPI 中断

状态标志位

- **发送缓冲区空标志位(TBE)**

当发送缓冲区为空或当前发送FIFO的存储量小于或等于总存储量的一半时，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/发送FIFO。

- **接收缓冲区非空标志位(RBNE)**

对于SPI1，该位根据SPI_CTL1中的BYTEN位设置：如果BYTEN=0，则当前接收FIFO的存储量大于或等于总存储量的1/2时，RBNE置位。如果BYTEN=1，则当前接收FIFO的存储量大于或等于总存储量的1/4时，RBNE置位。表示此时接收到数据，并已存入接收FIFO中，软件可以通过读SPI_DATA寄存器来读取此数据。

对于SPI0，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。对于SPI1，这说明，上一帧数据还未被读出而新的数据已经接收了。对于SPI0，这说明，接收FIFO没有足够的空间来存储接收到的数据了。接收缓冲区/接收FIFO的内容不会被新接收的数据覆盖，所以新接收的数据丢失。

- **帧错误(FERR)**

在TI从机模式下，从机也要监视NSS信号，如果检测到错误的NSS信号，将会置位FERR标志位。例如，NSS信号在一个字节的中间位发生翻转。

- **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后接收到的CRC值进行比较，当两者不同时，CRCERR位将会置1。

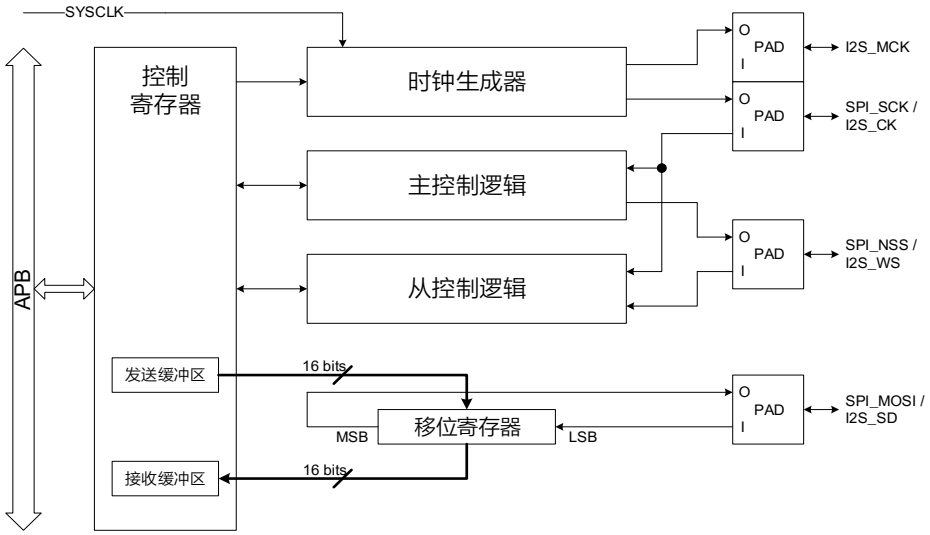
表 20-6. SPI 中断请求

中断事件	描述	清除方式	中断使能位
TBE	发送缓冲区/发送FIFO空	写SPI_DATA寄存器	TBEIE
RBNE	接收缓冲区/接收FIFO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 SPI_STAT寄存器	
CRCERR	CRC错误	写0到CRCERR位	
FERR	TI模式帧错误	写0到FERR位	

20.4. I2S 功能说明

20.4.1. I2S 结构框图

图 20-16. 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。

20.4.2. 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。

20.4.3. 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 20-17. I2S 飞利浦标准时序图(DTLEN=00, CHLEN=0, CKPL=0)

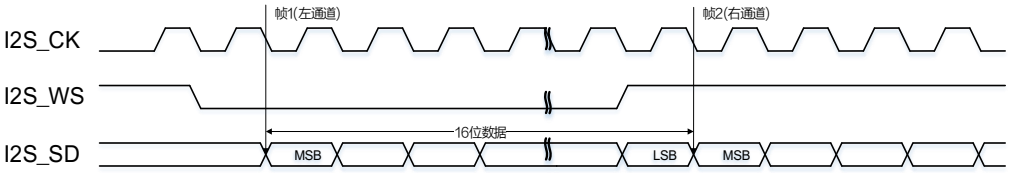
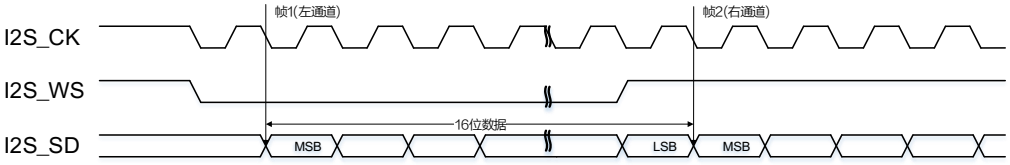


图 20-18. I2S 飞利浦标准时序图(DTLEN=00, CHLEN=0, CKPL=1)



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 20-19. I2S 飞利浦标准时序图(DTLEN=10, CHLEN=1, CKPL=0)

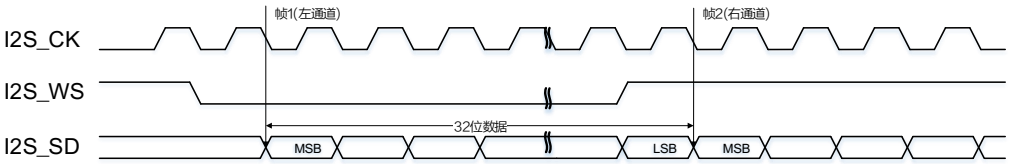
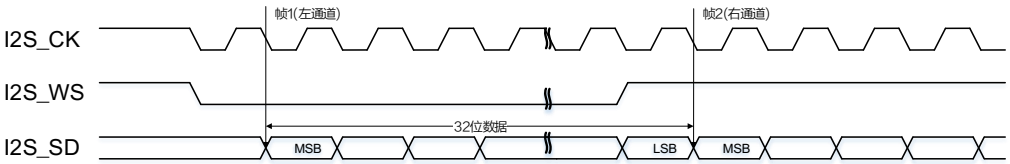


图 20-20. I2S 飞利浦标准时序图(DTLEN=10, CHLEN=1, CKPL=1)



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 20-21. I2S 飞利浦标准时序图(DTLEN=01, CHLEN=1, CKPL=0)

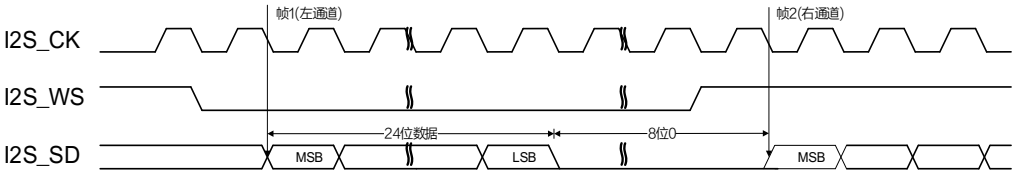
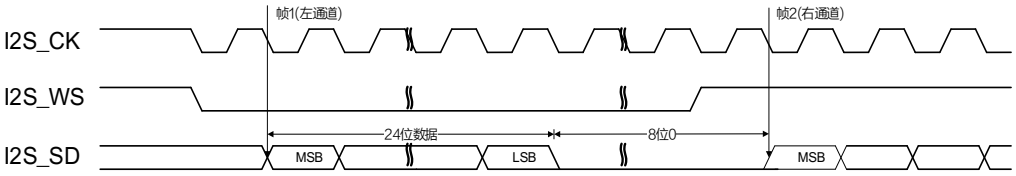


图 20-22. I2S 飞利浦标准时序图(DTLEN=01, CHLEN=1, CKPL=1)



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 20-23. I2S 飞利浦标准时序图(DTLEN=00, CHLEN=1, CKPL=0)

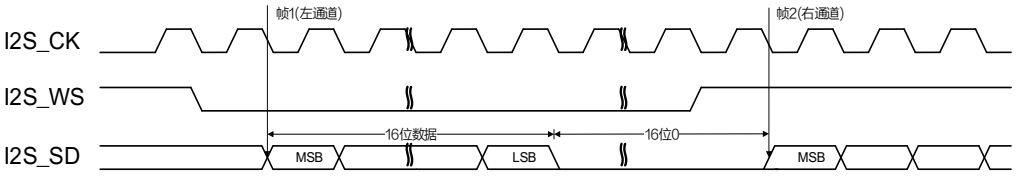
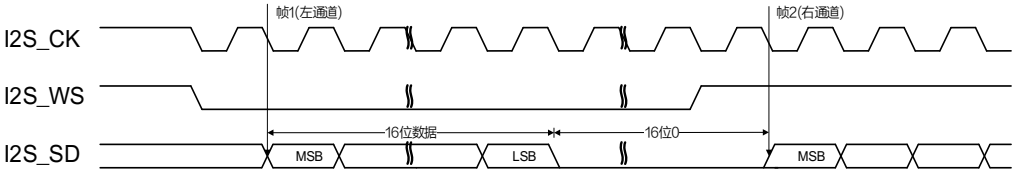


图 20-24. I2S 飞利浦标准时序图(DTLEN=00, CHLEN=1, CKPL=1)



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 20-25. MSB 对齐标准时序图(DTLEN=00, CHLEN=0, CKPL=0)

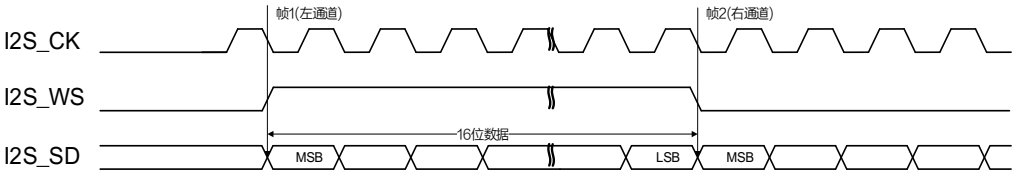


图 20-26. MSB 对齐标准时序图(DTLEN=00, CHLEN=0, CKPL=1)

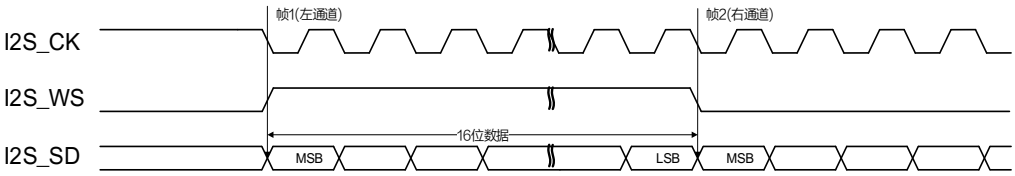


图 20-27. MSB 对齐标准时序图(DTLEN=10, CHLEN=1, CKPL=0)

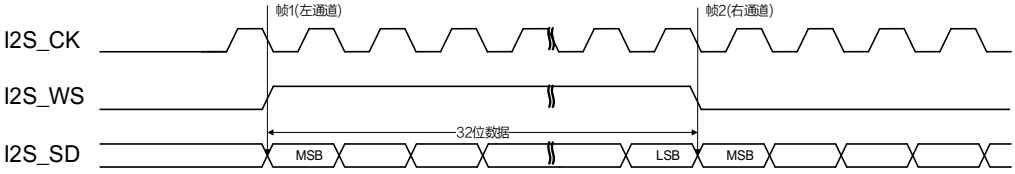


图 20-28. MSB 对齐标准时序图(DTLEN=10, CHLEN=1, CKPL=1)

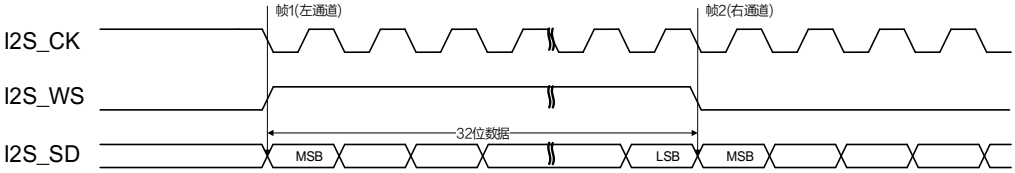


图 20-29. MSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=0)

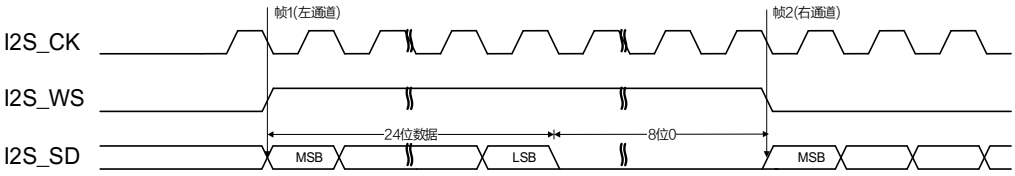


图 20-30. MSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=1)

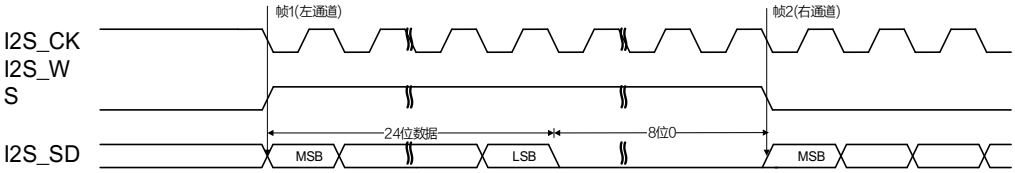


图 20-31. MSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=0)

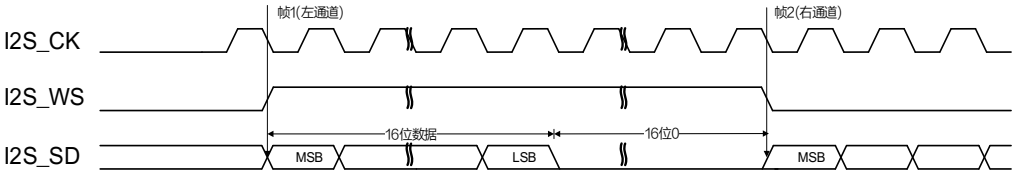
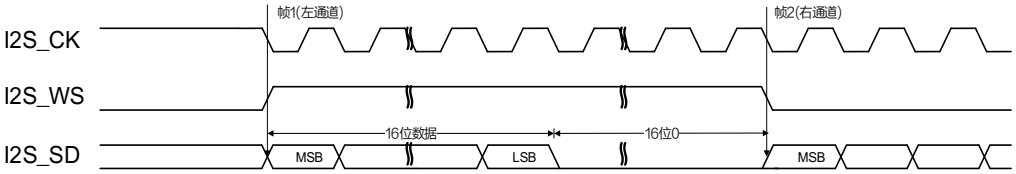


图 20-32. MSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=1)



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 20-33. LSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=0)

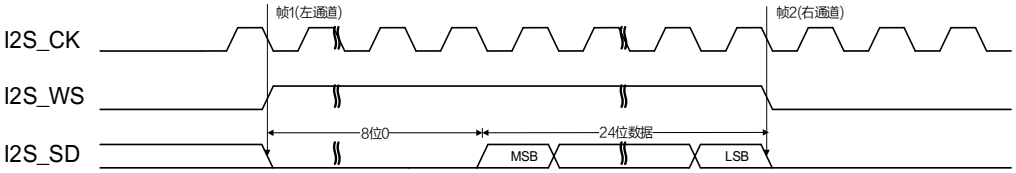
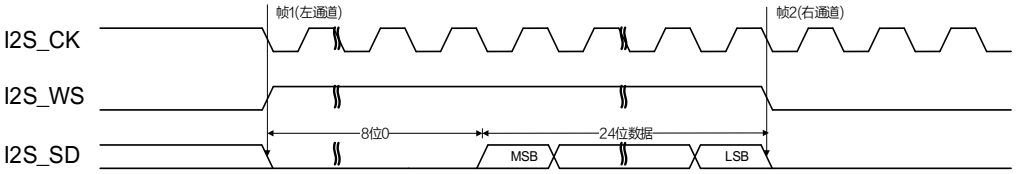


图 20-34. LSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=1)



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 20-35. LSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=0)

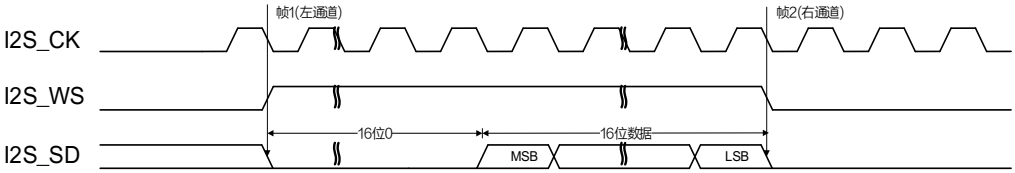
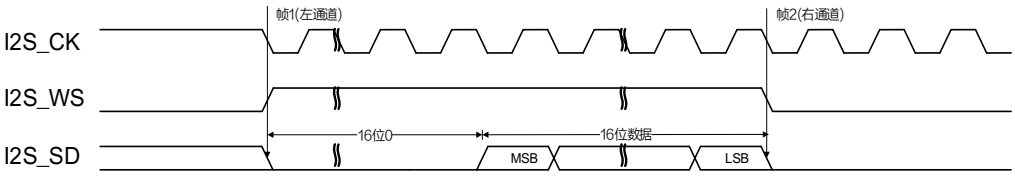


图 20-36. LSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=1)



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 20-37. PCM 标准短帧同步模式时序图(DTLEN=00, CHLEN=0, CKPL=0)

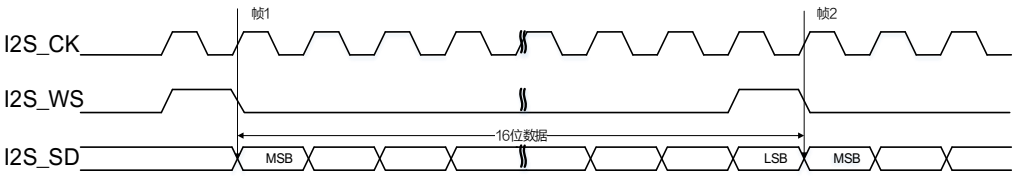


图 20-38. PCM 标准短帧同步模式时序图(DTLEN=00, CHLEN=0, CKPL=1)

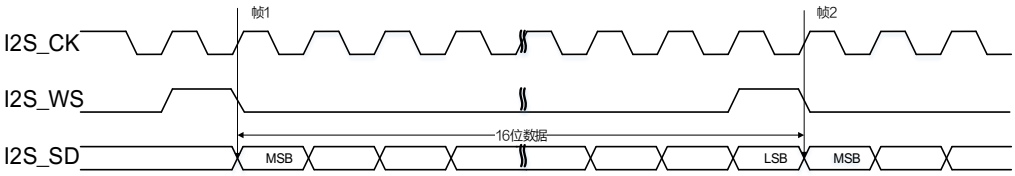


图 20-39. PCM 标准短帧同步模式时序图(DTLEN=10, CHLEN=1, CKPL=0)

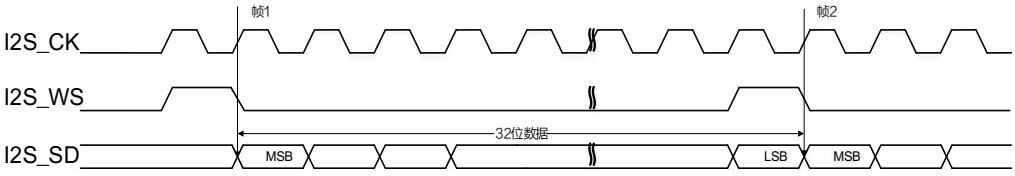


图 20-40. PCM 标准短帧同步模式时序图(DTLEN=10, CHLEN=1, CKPL=1)

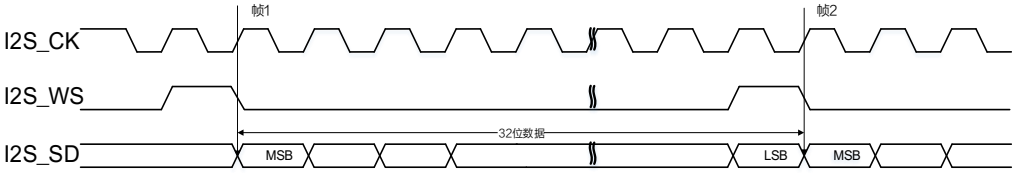


图 20-41. PCM 标准短帧同步模式时序图(DTLEN=01, CHLEN=1, CKPL=0)

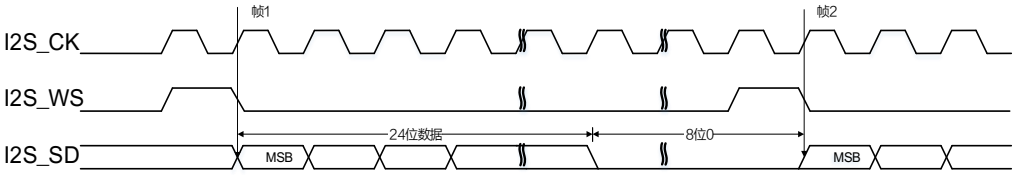


图 20-42. PCM 标准短帧同步模式时序图(DTLEN=01, CHLEN=1, CKPL=1)

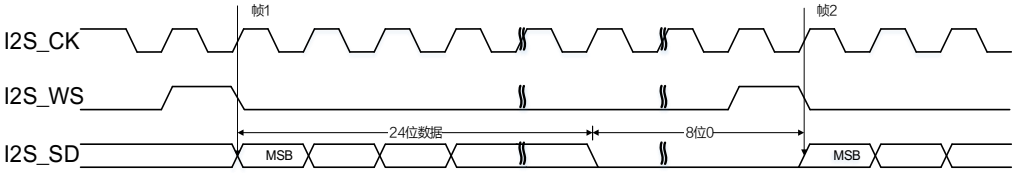


图 20-43. PCM 标准短帧同步模式时序图(DTLEN=00, CHLEN=1, CKPL=0)

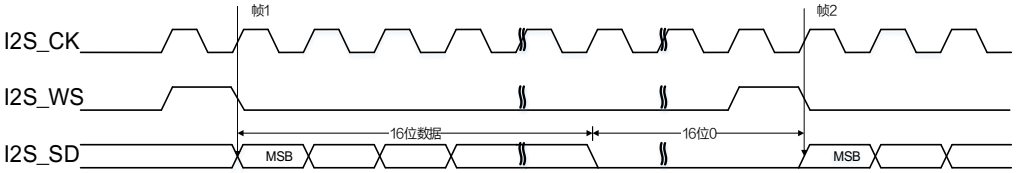
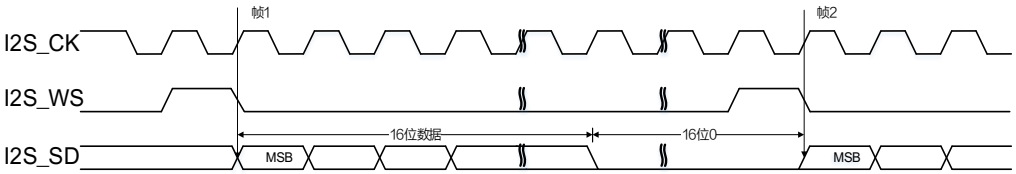


图 20-44. PCM 标准短帧同步模式时序图(DTLEN=00, CHLEN=1, CKPL=1)



长帧同步模式的各种配置情况时序图如下所示。

图 20-45. PCM 标准长帧同步模式时序图(DTLEN=00, CHLEN=0, CKPL=0)

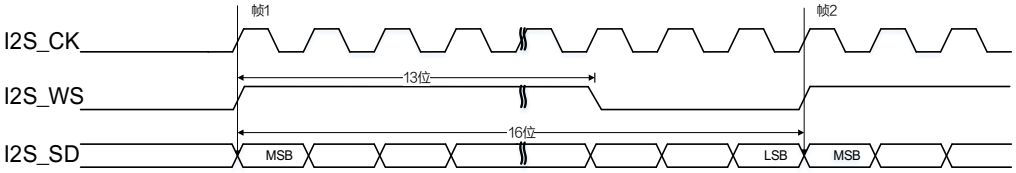


图 20-46. PCM 标准长帧同步模式时序图(DTLEN=00, CHLEN=0, CKPL=1)

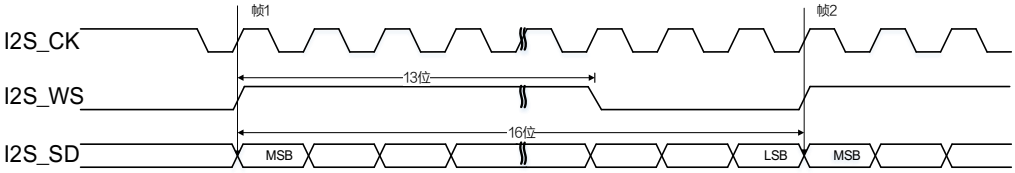


图 20-47. PCM 标准长帧同步模式时序图(DTLEN=10, CHLEN=1, CKPL=0)

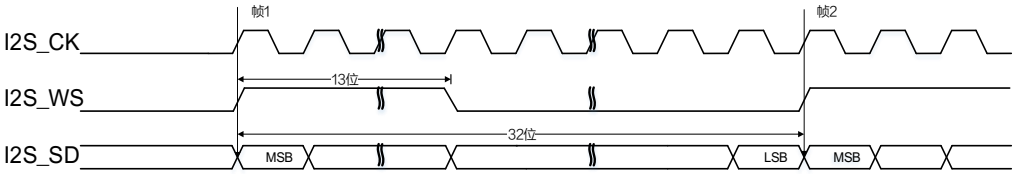


图 20-48. PCM 标准长帧同步模式时序图(DTLEN=10, CHLEN=1, CKPL=1)

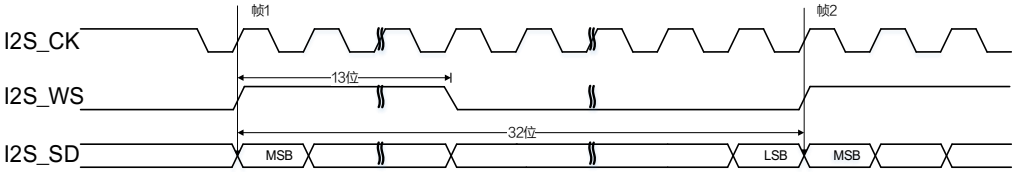


图 20-49. PCM 标准长帧同步模式时序图(DTLEN=01, CHLEN=1, CKPL=0)

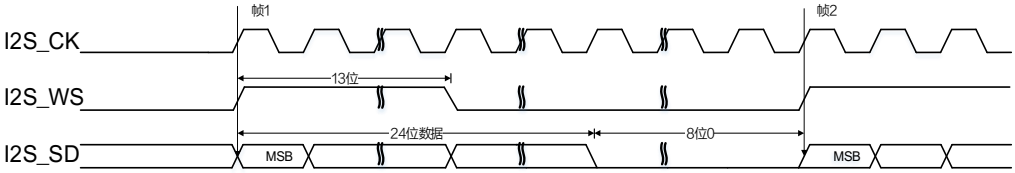


图 20-50. PCM 标准长帧同步模式时序图(DTLEN=01, CHLEN=1, CKPL=1)

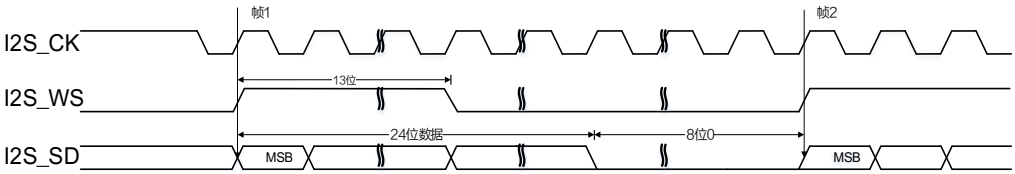


图 20-51. PCM 标准长帧同步模式时序图(DTLEN=00, CHLEN=1, CKPL=0)

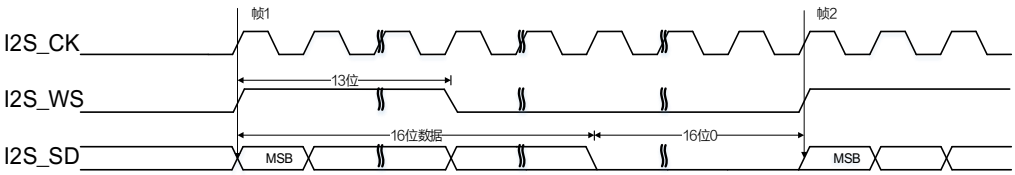
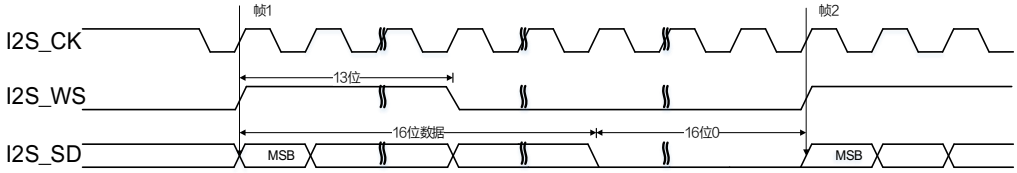
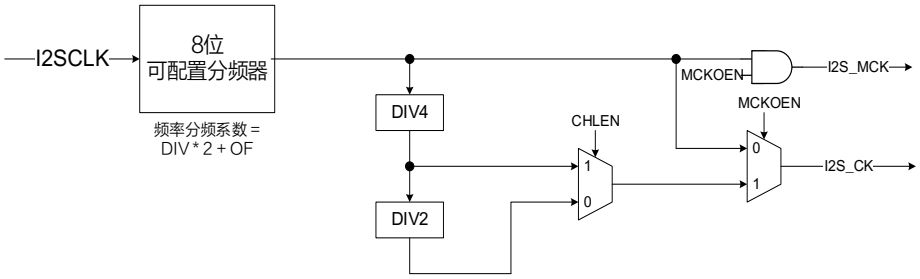


图 20-52. PCM 标准长帧同步模式时序图(DTLEN=00, CHLEN=1, CKPL=1)



20.4.4. I2S 时钟

图 20-53. I2S 时钟生成结构框图



I2S 时钟生成器框图如*图 20-53. 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 20-7. I2S 比特率计算公式*所示的公式计算。

表 20-7. 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20-8. 音频采样频率计算公式*所列的公式进行配置。

表 20-8. 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)。

20.4.5. 运行

运行模式

运行模式是通过 SPI_I2SCTL 寄存器的 I2SOPMOD 位来选择的。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 20-9. 各种运行模式下 I2S 接口信号的方向*所示。

表 20-9. 各种运行模式下 I2S 接口信号的方向

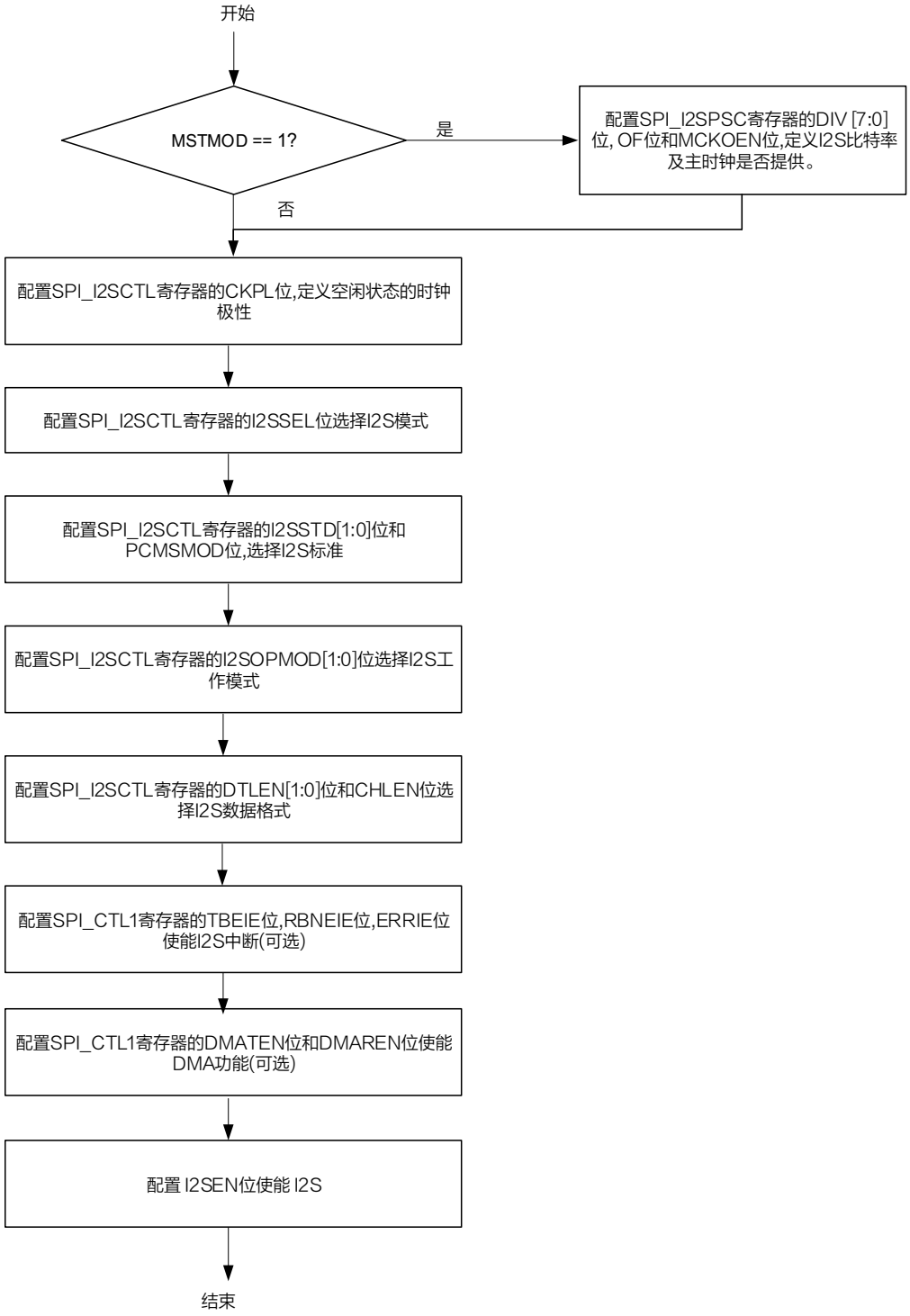
运行模式	I2S_MCK	I2S_CK	I2S_WS	I2S_SD
主机发送	输出或 NU(1)	输出	输出	输出
主机接收	输出或 NU(1)	输出	输出	输入
从机发送	输出或 NU(1)	输入	输入	输出
从机接收	输出或 NU(1)	输入	输入	输入

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20-54. I2S初始化流程*所示。

图 20-54. 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。

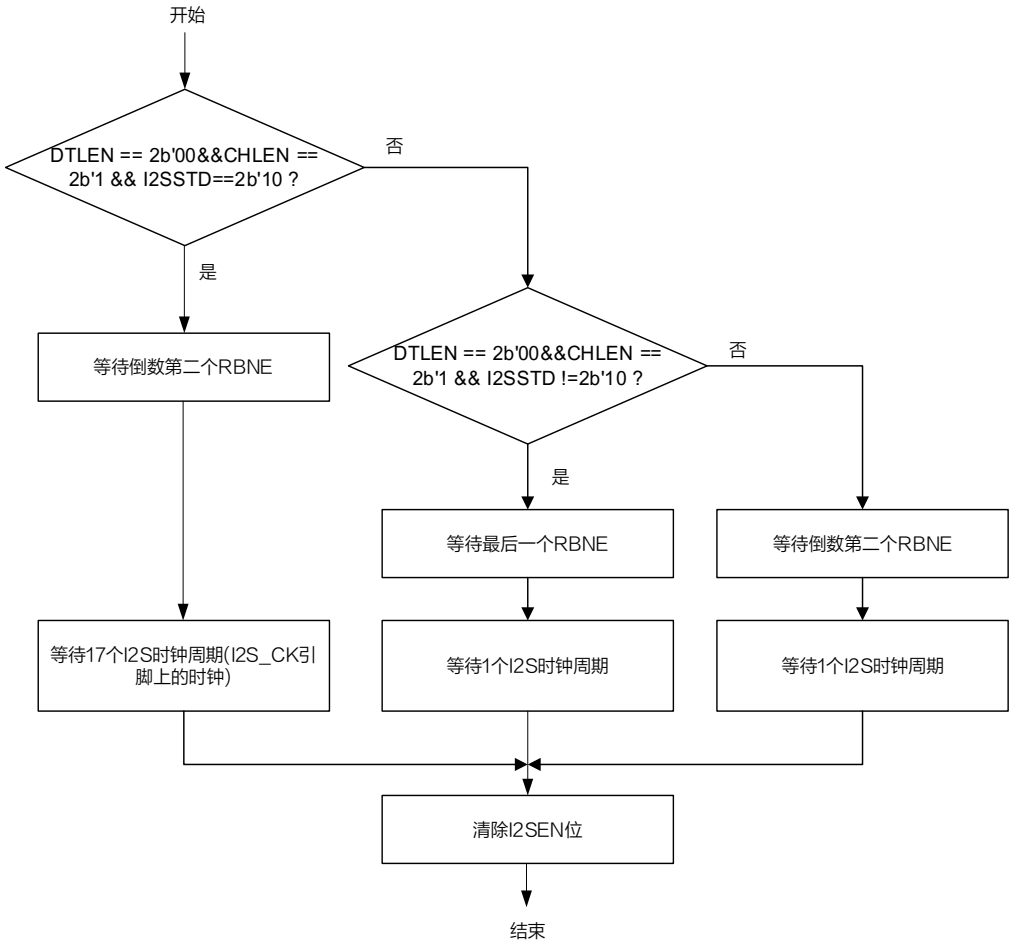
I2S 主机接收流程

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20-55. I2S主机接收禁能流程*所示。

图 20-55. 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。

20.4.6. 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。

20.4.7. 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20-10. I2S中断*总结了I2S中断事件和相应的使能位。

表 20-10. 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	

20.5. SPI/I2S 寄存器

SPI0/I2S0基地址：0x4001 3000
SPI1基地址：0x4000 3800

20.5.1. 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															
rw	rw	rw	rw	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
BDEN	BDOEN	CRCEN	CRCNT	FF16 CRCL	RO	SWNSS EN	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	双向数据模式使能 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：CRC计算禁止 1：CRC计算使能
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(只有SPI0) 0：8位数据帧格式

		1：16位数据帧格式
	CRCL	CRC长度(只有SPI1) 0：8位CRC长度 1：16位CRC长度
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软件模式使能 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 该位在SPI TI模式下没有意义。
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。 该位在SPI TI模式下没有意义。
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 该位在SPI TI模式下没有意义。
6	SPIEN	SPI使能 0：SPI设备禁止 1：SPI设备使能
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采集第一个数据

20.5.2. 控制寄存器 1(SPI_CTL1)

地址偏移：0x04

复位值：SPI0：0x0000 0000

SPI1：0x0000 0700

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	TXDMA_ODD	RXDMA_ODD	BYTEN	DZ[3:0]				TBEIE	RBNEIE	ERRIE	TMOD	NSSP	NSSDRV	DMATEN	DMAREN
	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	TXDMA_ODD	DMA发送通道奇数字节(只有SPI1) 在数据合并传输模式中，当通过DMA发送的数据总数为奇数时置位。仅在DMA功能开启且合并模式开启时(数据长度小于等于8位且对SPI_DATA写入访问是16位宽)有效。 必须在SPI禁止时写入。 0：通过DMA发送的数据总量为偶数个。 1：通过DMA发送的数据总量为奇数个。
13	RXDMA_ODD	DMA接收通道奇数字节(只有SPI1) 在数据合并传输模式中，当通过DMA接收的数据总数为奇数时置位。仅在DMA功能开启且合并模式开启时(数据长度小于等于8位且对SPI_DATA写入访问是16位宽)有效。 必须在SPI禁止时写入。 0：通过DMA接收的数据总量为偶数个。 1：通过DMA接收的数据总量为奇数个。
12	BYTEN	字节访问使能(只有SPI1) 该位用于指示对FIFO的访问宽度，并设置产生RBNE的RXFIFO的阈值。 0：半字访问，且当RXLVL>=2时，RBNE置位。 1：字节访问，且当RXLVL>=1时，RBNE置位。
11:8	DZ[3:0]	数据位宽(只有SPI1) 这些位配置SPI传输数据的位宽：

		0000：强制为"0111" 0001：强制为"0111" 0010：强制为"0111" 0011：4位 0100：5位 …… 1111：16位
7	TBEIE	发送缓冲区/发送FIFO空中断使能 0：TBE中断禁止 1：TBE中断使能。当TBE置位时，产生中断。
6	RBNEIE	接收缓冲区/接收FIFO非空中断使能 0：RBNE中断禁止 1：RBNE中断使能。当RBNE置位时，产生中断。
5	ERRIE	错误中断使能 0：错误中断禁止 1：错误中断使能。当CRCERR位，CONFERR位，RXORERR位或者TXURERR位置1时，产生中断。
4	TMOD	SPI TI模式使能 0：SPI TI模式禁止 1：SPI TI模式使能
3	NSSP	SPI NSS脉冲模式使能 0：SPI NSS脉冲模式禁止 1：SPI NSS脉冲模式使能
2	NSSDRV	NSS输出使能 0：NSS输出禁止 1：NSS输出使能。 当SPI使能时，如果NSS引脚配置为输出模式，NSS引脚在主模式时被拉低。如果NSS引脚配置为输入模式，NSS引脚在主模式时被拉高，此时该位无效。
1	DMATEN	发送缓冲区/发送FIFO DMA使能 0：发送缓冲区/发送FIFO DMA禁止 1：发送缓冲区/发送FIFO DMA使能。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/接收FIFO DMA使能 0：接收缓冲区/接收FIFO DMA禁止 1：接收缓冲区/接收FIFO DMA使能。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。

20.5.3. 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			TXLVL[1:0]		RXLVL[1:0]		FERR	TRANS	RXORERR	CONFERR	CRCERR	TXURERR	I2SCH	TBE	RBNE
			r		r		rc_w0	r	r	r	rc_w0			r	r

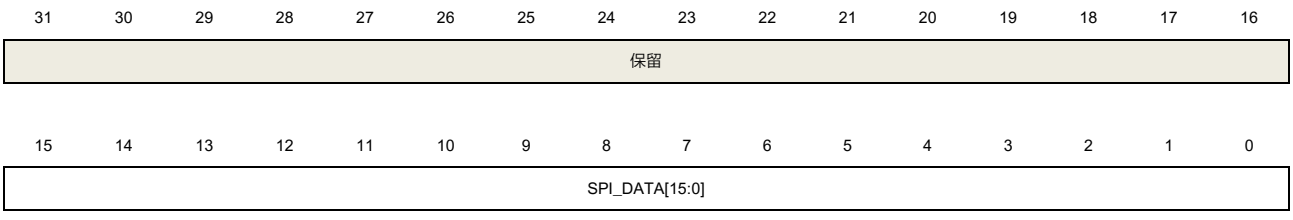
位/位域	名称	描述
31:13	保留	必须保持复位值。
12:11	TXLVL[1:0]	发送FIFO状态(只有SPI1) 00：空 01：1/4满 10：1/2满 11：满 **注意：**这里的FIFO状态是指FIFO当前实际的存储量。在这里，当FIFO存储量大于总存储量的1/2时认为FIFO已满。
10:9	RXLVL[1:0]	接收FIFO状态(只有SPI1) 00：空 01：1/4满 10：1/2满 11：满 这些位在打开了CRC计算功能时的SPI只接收模式下，不使用。 **注意：**这里的FIFO状态是指FIFO当前实际的存储量。在这里，当FIFO存储量大于总存储量的1/2时认为FIFO已满。
8	FERR	帧错误 SPI TI模式： SPI TI模式： 0：没有TI模式帧错误发生 1：TI模式帧错误发生 I2S模式： 0：没有I2S帧错误发生 1：I2S帧错误发生
7	TRANS	通信进行中标志 0：SPI空闲 1：SPI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。
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。
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：下一个将要发送或刚刚接收到的数据属于左通道 1：下一个将要发送或刚刚接收到的数据属于右通道 该位由硬件置位和清除。 SPI模式下该位无用，I2S PCM模式下该位没有意义。
1	TBE	发送缓冲区/发送FIFO空 0：发送缓冲区/发送FIFO非空 1：发送缓冲区/发送FIFO空
0	RBNE	接收缓冲区/接收FIFO非空 0：接收缓冲区/接收FIFO空 1：接收缓冲区/接收FIFO非空

20.5.4. 数据寄存器(SPI_DATA)

地址偏移：0x0C
复位值：0x0000 0000

对于SPI0，该寄存器可以按字节(8位)或半字(16位)访问。对于SPI1，该寄存器可以按半字(16位)或字(32位)访问。



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 对于SPI1，硬件有两个FIFO：TXFIFO和RXFIFO。向SPI_DATA写数据将会把数据存入发送FIFO，从SPI_DATA读数据，将从接收FIFO获得数据。 对于SPI0，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。 **注意：**对于SPI1，实际上硬件只根据配置好的BYTEN这一位来判断每一次访问SPI_DATA的位宽，与软件当前操作所使用的位宽无关。

20.5.5. 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。

20.5.6. 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。对于SPI0，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]。对于SPI1，只有当数据长度为8位或16位时，CRC有效。当CRC长度设置为8位并且数据长度等于8位时，CRC计算基于CRC8标准进行，并将值保存在RCRC[7：0]中，否则CRC计算基于CRC16标准进行，并将值保存在RCRC[15：0]中。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位或RCU复位寄存器中的SPIxRST位置位时，该寄存器复位。

20.5.7. 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。对于SPI0，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]。对于SPI1，只有当数据长度为8位或16位时，CRC有效。当CRC长度设置为8位并且数据长度等于8位时，CRC计算基于CRC8标准进行，并将值保存在TCRC[7：0]中，否则CRC计算基于CRC16标准进行，并将值保存在TCRC[15：0]中。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位或RCU复位寄存器中的SPIxRST位置位时，该寄

存器复位。

20.5.8. 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或I2S关闭时配置该位。
10	I2SEN	I2S使能 0：I2S禁止 1：I2S使能 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关闭时配置该位。SPI模式不使用该位。

20.5.9. 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						MCKOEN	OF	DIV[7:0]							
						rw	rw	rw							

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：I2S_MCK输出禁止 1：I2S_MCK输出使能 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

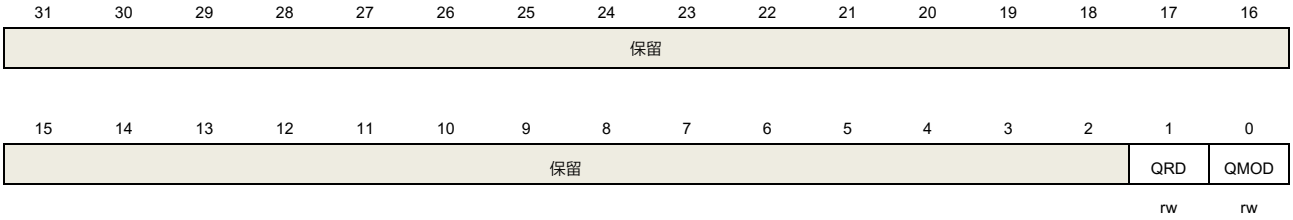
		当I2S关闭时配置该位。SPI模式下不使用该位。
7:0	DIV[7:0]	预分频器的分频系数 实际分频系数是DIV * 2 + OF。 DIV不能为0。 当I2S关闭时配置该位。SPI模式下不使用该位。

20.5.10. SPI1 四线 SPI 控制寄存器(SPI_QCTL)

地址偏移：0x80

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
31:2	保留	必须保持复位值。
1	QRD	四线SPI模式读选择 0：SPI四线模式写操作 1：SPI四线模式读操作 该位仅能在SPI未通信时配置(TRANS位清零)。 该位仅适用于SPI1。
0	QMOD	四线SPI模式使能 0：SPI工作在单线模式 1：SPI工作在四线模式 该位仅能在SPI未通信时配置(TRANS位清零)。 该位仅适用于SPI1。

21. 内部集成电路总线接口(I2C)

21.1. 内部集成电路总线接口(I2Cx, x=0, 1)

21.1.1. 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，快速模式以及快速+ 模式，具备 CRC 计算和校验功能、支持 SMBus(系统管理总线)、PMBus(电源管理总线)和 SAM_V(验证安全控制模块)模式，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。

21.1.2. 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。
- 多个 7 位从机地址(两个地址，且都可配置地址位屏蔽)。
- 支持 I2C 多主机模式。
- 支持标速模式(最高 100 KHz)，快速模式(最高 400 KHz)和快速+ 模式(最高 1 MHz)。
- 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。
- 支持 SAM_V 模式。

21.1.3. 功能说明

I2C 接口的内部结构如*图 21-1. I2C 模块框图*所示。

图 21-1. I2C 模块框图

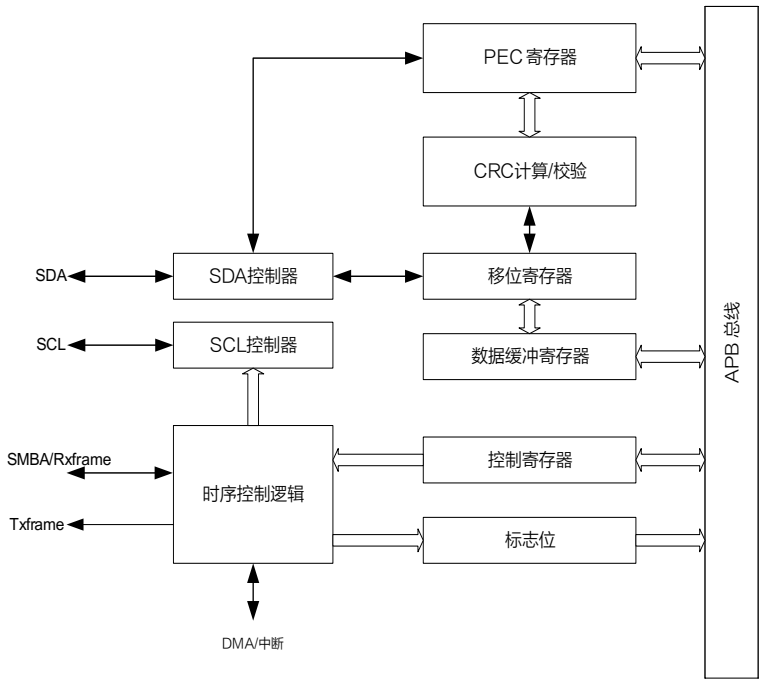


表 21-1. I2C 总线术语说明(参考飞利浦 I2C 规范)

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

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。

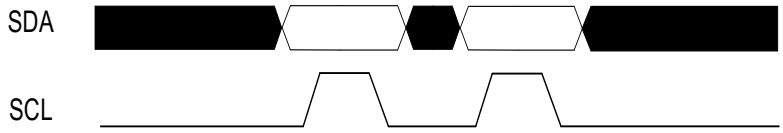
SDA 和 SCL 都是双向线，通过一个电流源或者上拉电阻接到电源正极。当总线空闲时，两条线都是高电平。连接到总线的设备输出极必须是开漏或者开集，以提供线与功能。I2C 总线上的数据在标准模式下可以达到 100 Kbit/s，在快速模式下可以达到 400 Kbit/s，当 I2C_CTL2 寄存器中 FMPEN 位被置位时，在快速+ 模式下可达 1 Mbit/s。由于 I2C 总线上可能会连接不

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。

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 21-2. 数据有效性*)。每个数据比特传输需要一个时钟脉冲。

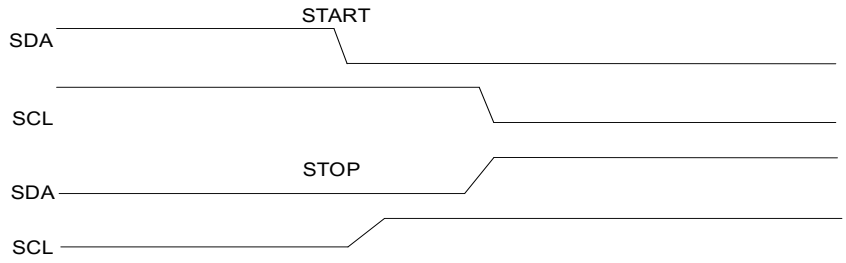
图 21-2. 数据有效性



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 21-3. 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 21-3. 起始和停止信号



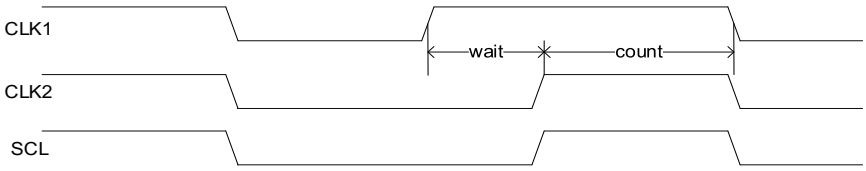
时钟同步

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 21-4. 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 21-4. 时钟同步



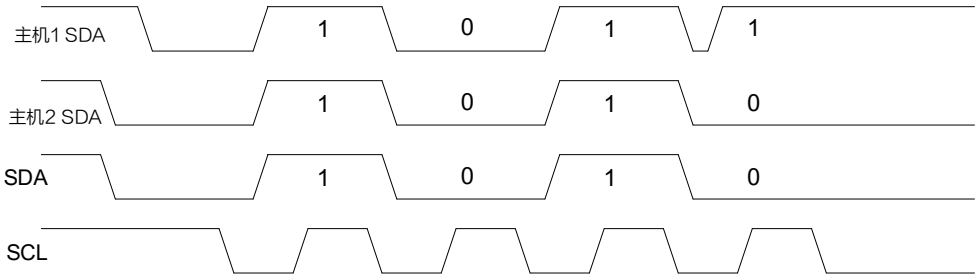
仲裁

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 21-5. SDA 线仲裁



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 21-6. 7 位地址的 I2C 通讯流程

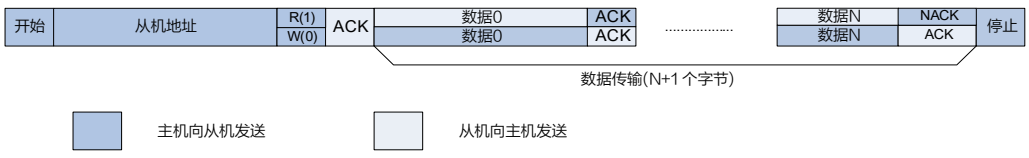


图 21-7. 10 位地址的 I2C 通讯流程(主机发送)

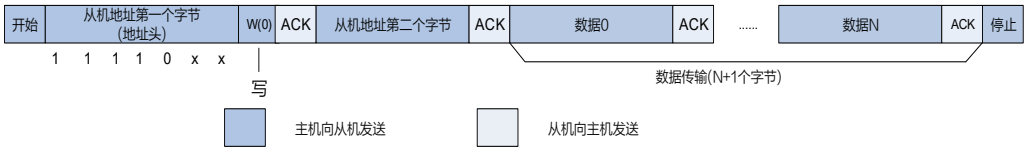
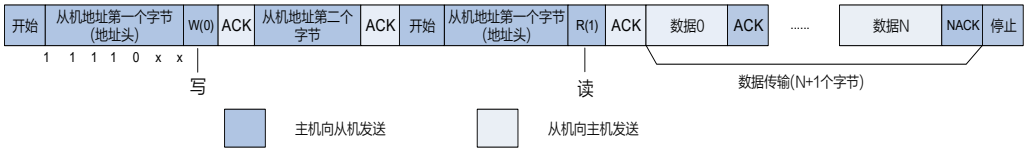


图 21-8. 10 位地址的 I2C 通讯流程(主机接收)



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21-9. 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 21-9. 从机发送模式(10 位地址模式)



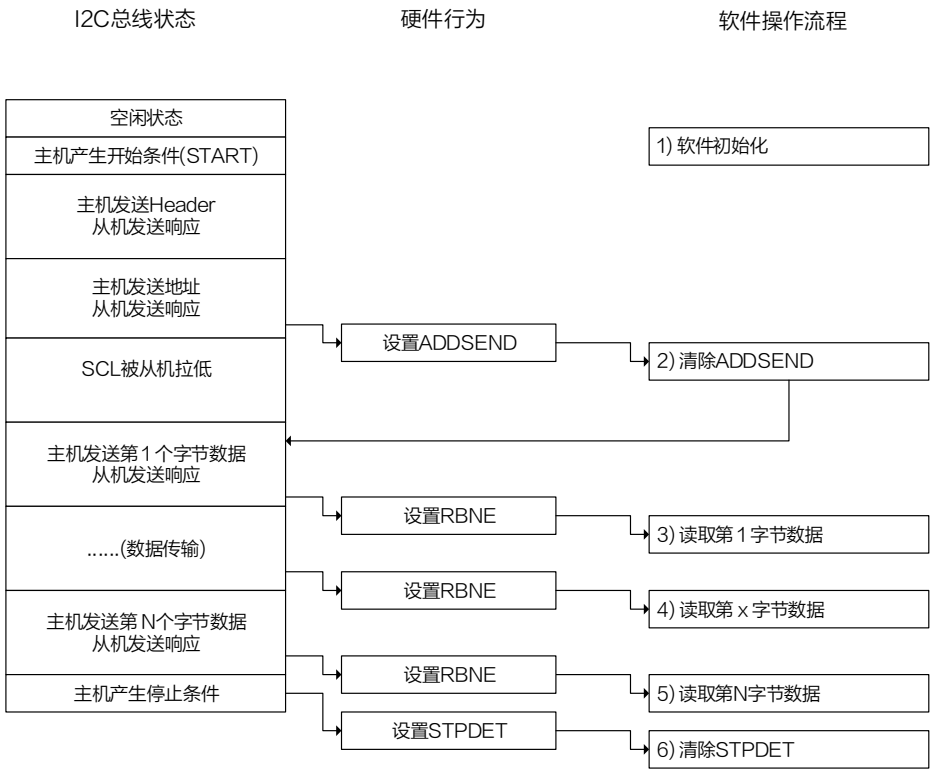
从机接收模式下的软件流程

如*图 21-10. 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 21-10. 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21-11. 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 21-11. 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 21-12. 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 21-13. 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 21-13. 主机接收使用方案 B 模式(10 位地址模式)



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。

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。

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。

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。

SAM_V 支持

为了支持 SAM_V 标准，I2C 模块增加两个附加的引脚：txframe 和 rxframe。Txframe 是一个输出引脚，在主机模式下，当该引脚输出电平为高电平时，表示 I2C 是忙的。Rxframe 是一个输入引脚，应该与 SMBALERT 信号一起多路复用。

SAM_V 模式通过置位 I2C_SAMCS 寄存器的 SAMEN 位使能。txframe 和 rxframe 引脚的状态可以通过 I2C_SAMCS 寄存器的 RFR，RFF，TFR，TFF，RXF 和 TXF 标志反映。如果对应的中断使能位置位，将产生 I2C 中断。

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 21-2. 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
RFR	SAM_V 模式时检测到 rxframe 上升沿
RFF	SAM_V 模式时检测到 rxframe 下降沿
TFR	SAM_V 模式时检测到 txframe 上升沿
TFF	SAM_V 模式时检测到 txframe 下降沿

表 21-3. 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

I2C 调试模式

当为控制器进入调试模式(Cortex®-M23 内核停止)，SMBus 超时定时器会根据 DBG 模块中的 I2Cx_HOLD(x=0，1)配置位选择继续正常工作还是停止工作。

21.1.4. I2C 寄存器

I2C0基地址：0x4000 5400

I2C1基地址：0x4000 5800

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。

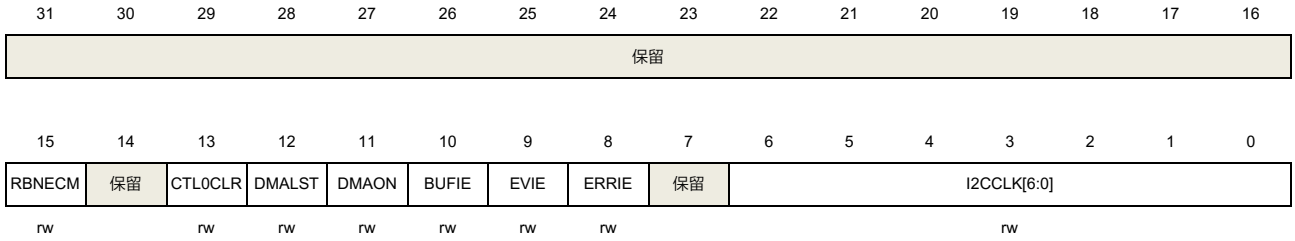
10	ACKEN	是否发送 ACK 软件置 1 和清 0，当 I2CEN=0 时硬件清 0。 0：不发送 ACK 1：发送 ACK
9	STOP	I2C 总线上产生一个 STOP 信号 软件置 1 和清 0，SMBus 超时时，硬件置 1，监测到 STOP 信号时，硬件清 0。 0：不发送 STOP 1：发送 STOP
8	START	I2C 总线上产生一个 START 信号 软件置 1 和清 0，当监测到 START 信号或 I2CEN=0 时由硬件清 0。 0：不发送 START 1：发送 START
7	SS	在从机模式下数据未就绪是否将 SCL 拉低 软件置 1 和清 0。 0：拉低 SCL 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

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
31:16	保留	必须保持复位值。
15	RBNECM	RBNE 清除模式 0：RBNE 在读取 I2C_DATA 寄存器并且 BTC 标志清除后被清除 1：RBNE 在读取 I2C_DATA 寄存器后被清除
14	保留	必须保持复位值。
13	CTL0CLR	当 I2CEN=0 时，决定是否清除 I2C_CTL0 0：当 I2CEN=0 时，清除 I2C_CTL0 1：当 I2CEN=0 时，保持 I2C_CTL0
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

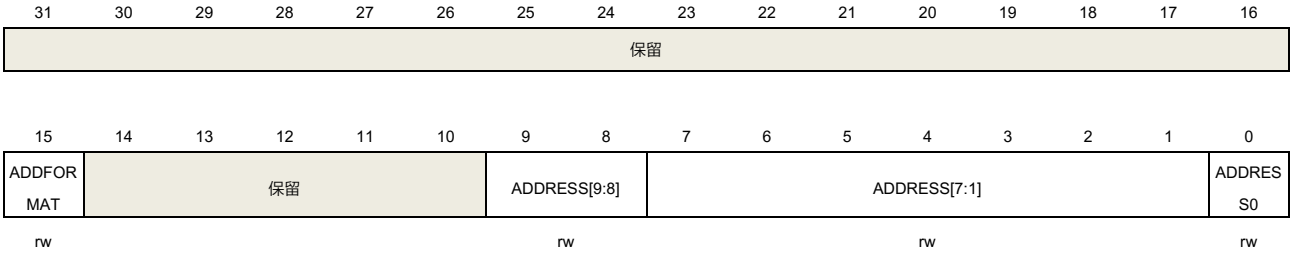
注意：在标准模式下，APB1 时钟频率需大于或者等于 2MHz。在快速模式下，APB1 时钟频率需大于或者等于 8MHz。在快速+ 模式下，APB1 时钟频率需大于或者等于 24MHz。

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。



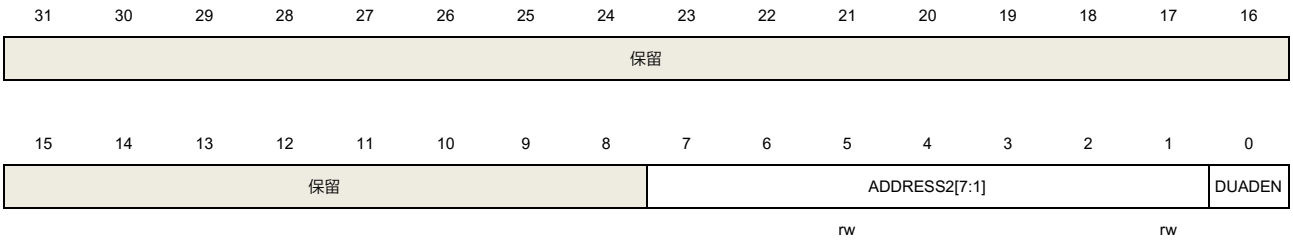
位/位域	名称	描述
31:16	保留	必须保持复位值。
15	ADDFORMAT	I2C 从机地址格式 0：7 位地址 1：10 位地址
14:10	保留	必须保持复位值。
9:8	ADDRESS[9:8]	10 位地址的最高两位
7:1	ADDRESS[7:1]	7 位地址或者 10 位地址的第 7-1 位
0	ADDRESS0	10 位地址的第 0 位

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

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

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)

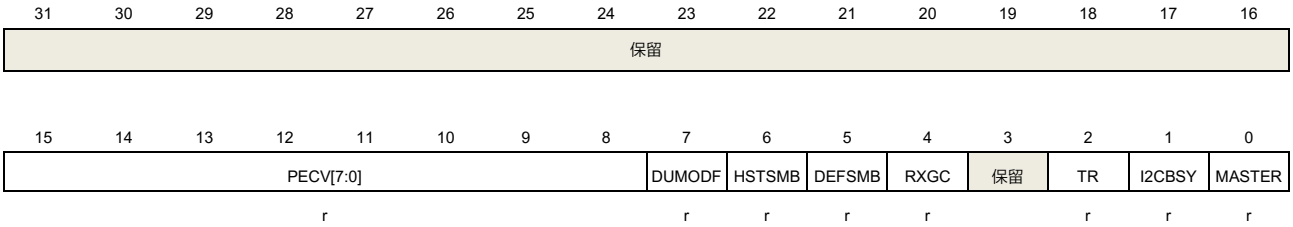
		1：SMBA 引脚被拉低且接收到警报地址(从机模式)或监测到警报(主机模式)
14	SMBTO	SMBus 模式下超时信号 硬件置 1，软件写 0 清 0。 0：无超时错误 1：超时事件发生(SCL 被拉低达 25ms)
13	保留	必须保持复位值。
12	PECERR	接收数据时 PEC 错误 硬件置 1，软件写 0 清 0。 0：接收到 PEC 且校验正确 1：接收到 PEC 但检验错误，此时不管 ACKEN 位的值，I2C 将发送 NACK
11	OUERR	当禁用 SCL 拉低功能后，在从机模式下发生了上溢或下溢事件。在从机接收模式下，假如 I2C_DATA 中的最后一字节并未被读出，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

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。



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

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	快速模式或快速+ 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，快速模式或快速+ 模式下： $T_{high}=CLKC*T_{PCLK1}$，$T_{low}=2*CLKC*T_{PCLK1}$ 如果 DTCY=1，快速模式或快速+ 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。

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

SAM 控制状态寄存器(I2C_SAMCS)

地址偏移：0x80
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
RFR	RFF	TFR	TFF	保留		RXF	TXF	RFRIE	RFFIE	TFRIE	TFFIE	保留		STOEN	SAMEN
rc_w0	rc_w0	rc_w0	rc_w0			r	r	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	RFR	接收帧上升沿标志，由软件写 0 清 0
14	RFF	接收帧下降沿标志，由软件写 0 清 0
13	TFR	发送帧上升沿标志，由软件写 0 清 0
12	TFF	发送帧下降沿标志，由软件写 0 清 0
11:10	保留	必须保持复位值。
9	RXF	接收帧信号电平
8	TXF	发送帧信号电平
7	RFRIE	接收帧上升沿中断使能 0：接收帧上升沿中断失能 1：接收帧上升沿中断使能
6	RFFIE	接收帧下降沿中断使能 0：接收帧下降沿中断失能 1：接收帧下降沿中断使能

5	TFRIE	发送帧上升沿中断使能 0：发送帧上升沿中断失能 1：发送帧上升沿中断使能
4	TFFIE	发送帧下降沿中断使能 0：发送帧下降沿中断失能 1：发送帧下降沿中断使能
3:2	保留	必须保持复位值。
1	STOEN	SAM_V 接口超时检测使能 0：SAM_V 接口超时检测失能 1：SAM_V 接口超时检测使能
0	SAMEN	SAM_V 接口使能 0：SAM_V 接口失能 1：SAM_V 接口使能

控制寄存器 2(I2C_CTL2)

地址偏移：0x90
复位值：0x0000 FE00

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									ADDM2[6:0]						
									rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDM[6:0]							RADD	保留			TOEN	保留		SETM	FMPEN
rw							rw				rw			rw	rw

位/位域	名称	描述
31:23	保留	必须保持复位值。
22:16	ADDM2[6:0]	定义 ADDRESS2[7:1]的哪些位和接收到的地址进行比较，哪些位不比较。ADDM2[6:0]中设置为 1 的位使能 ADDRESS2[7:1]中的相应位与接收到的地址进行比较，设置为 0 的位则忽视(此时接收到的地址在该位可以为 0 或 1)。
15:9	ADDM[6:0]	定义 ADDRESS[7:1]的哪些位和接收到的地址进行比较，哪些位不比较。ADDM[6:0]中设置为 1 的位使能 ADDRESS[7:1]中的相应位与接收到的地址进行比较，设置为 0 的位则忽视(此时接收到的地址在该位可以为 0 或 1)。
8	RADD	当该位为 1 时，接收到的 7 位从机地址或者 10 位从机地址的[7:0]位将会被记录在传输缓冲寄存器(I2C_DATA)里。
7:5	保留	必须保持复位值。
4	TOEN	超时计数使能。当没有使能 SMBA 模式时，该位用来使能 25ms SCL 低电平状态超时计数。

		0：禁能 SCL 低电平状态超时计数 1：使能 SCL 低电平状态超时计数
3:2	保留	必须保持复位值。
1	SETM	启动提前终止模式。该位描述了当 I2C 在启动阶段之后，SCL 处于高电平状态时，如何处理接下来 SCL 的低电平状态。 0：终止 SCL 的高电平状态，开始计数 SCL 的低电平状态(和 I2C 标准协议一样) 1：处理方法和仲裁丢失时一样
0	FMPEN	快速+ 模式使能 当该位被置 1 时，I2C 设备支持高达 1MHz 的通信速率。 0：快速+ 模式失能 1：快速+ 模式使能

控制和状态寄存器(I2C_CS)

地址偏移：0x94
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						STPSENDIE	STLOIE	保留						STPSEND	STLO
						rw	rw							rc_w0	rc_w0

位/位域	名称	描述
31:10	保留	必须保持复位值。
9	STPSENDIE	停止信号发送中断使能 0：禁能中断 1：使能中断
8	STLOIE	起始信号丢失中断使能 0：禁能中断 1：使能中断
7:2	保留	必须保持复位值。
1	STPSEND	主机模式下停止信号发送 此位被硬件置 1，软件写 0 清 0。 0：没有停止信号发送 1：停止信号发送
0	STLO	起始信号丢失标志位

如果 SETM 位设置为 1，则当 I2C 处于开始开始保持定时阶段(tHD:STA)时，SCL 线的低电平发生时，该位置位。
此位被硬件置 1，软件写 0 清 0。
0：没有发生起始信号丢失
1：发生了起始信号丢失

状态清除寄存器(I2C_STATC)

地址偏移：0x98
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	SRCEN	状态寄存器清除使能位 当该位置 1 时，STOPF、ADD10SEND、BTC、ADDSEND、SBSEND 状态位可以通过分别向 STOPFC、ADD10SENDC、BTCC、ADDSENDC、SBSENDC 位写 1 来清除。一旦 SRCEN 位置 1，那么 STOPF、ADD10SEND、BTC、ADDSEND、SBSEND 状态位不能被其它的软件序列清除。 0：禁能清除功能 1：使能清除功能
14:5	保留	必须保持复位值。
4	STOPFC	STOPF 状态清除位 当 SRCEN 位置 1 时，软件可以通过往该位写 1 来清除 I2C_STAT0 寄存器的 STPDET 位
3	ADD10SENDC	ADD10SEND 状态清除位 当 SRCEN 位置 1 时，软件可以通过往该位写 1 来清除 I2C_STAT0 寄存器的 ASS10SEND 位
2	BTCC	BTC 状态清除位 当 SRCEN 位置 1 时，软件可以通过往该位写 1 来清除 I2C_STAT0 寄存器的 BTC 位
1	ADDSENDC	ADDSEND 状态清除位 当 SRCEN 位置 1 时，软件可以通过往该位写 1 来清除 I2C_STAT0 寄存器的

		ASSSEND 位
0	SBSENDC	起始发送状态清除位 当SRCEN位置1时,软件可以通过往该位写1来清除I2C_STAT0寄存器的SBSEND位

21.2. 内部集成电路总线接口(I2C2)

21.2.1. 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，快速模式以及快速+ 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，I2C 接口模块还支持多主机 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。

21.2.2. 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。
- 多个 7 位从机地址(两个地址，且都可配置地址位屏蔽)。
- 可编程的建立时间和保持时间。
- 支持 I2C 多主机模式。
- 支持标速模式(最高 100 kHz)，快速模式(最高 400 kHz)和快速+ 模式(最高 1MHz，必须在 SYSCFG_CFG0 中使能 Fm+)。
- 从机模式下可配置的 SCL 主动拉低。
- 支持 DMA 模式。
- 兼容 SMBus 3.0 和 PMBus 1.3。
- 可选择的 PEC(报文错误校验)生成和校验。
- 可编程模拟过滤器和数字过滤器。
- I2C 地址匹配时，由睡眠模式和深度睡眠模式唤醒。
- 独立于 PCLK 的时钟。

21.2.3. 功能说明

I2C 接口的内部结构如*图 21-14. I2C 模块框图*所示。

图 21-14. I2C 模块框图

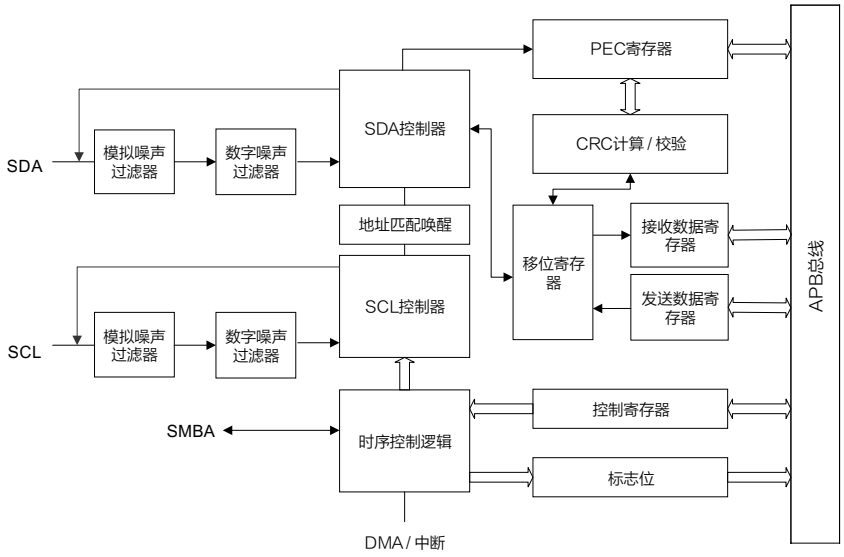


表 21-4. I2C 总线术语说明(参考飞利浦 I2C 规范)

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	不破坏信息的前提下同时控制总线的多个主机
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，保证上述的过程叫仲裁

时钟要求

I2C 时钟独立于 PCLK 时钟，因此可以独立操作 I2C。

I2C 时钟(I2CCLK)可以从以下三个时钟源中选择：

- IRC8M：内部 8M RC 时钟(默认值)
- APB1 时钟 PCLK1
- 系统时钟 SYSCLK
- HXTAL：外部高速晶振

I2C 时钟周期t_{I2CCLK}必须满足以下条件：

- $t_{I2CCLK} < (t_{LOW} - t_{filters})/4$
- $t_{I2CCLK} < t_{HIGH}$

其中：

t_{LOW}：SCL 低电平时间

t_{HIGH}：SCL 高电平时间

$t_{filters}$：在使能滤波器时，表示模拟滤波器和数字滤波器产生的延时总和。模拟滤波器产生的延时最大值为 120ns，数字滤波器产生的延时为DNF[3:0]×t_{I2CCLK}。

PCLK 时钟周期t_{PCLK}必须满足以下条件：

- t_{PCLK}<4/3*t_{SCL}

其中：

t_{SCL}：SCL 周期

注意：当 I2C 内核时钟由 PCLK 提供时，PCLK 必须符合t_{I2CCLK}的条件。

I2C 通讯流程

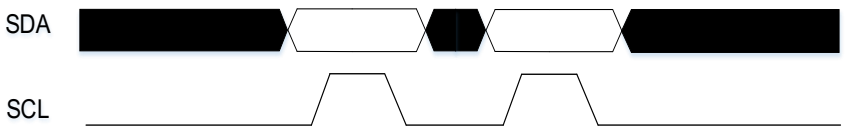
主机和从机都能实现数据收发，因此，I2C 可以实现四种工作模式：

- 从机发送
- 从机接收
- 主机发送
- 主机接收

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 21-15. 数据有效性*。每个数据比特传输需要一个时钟脉冲。

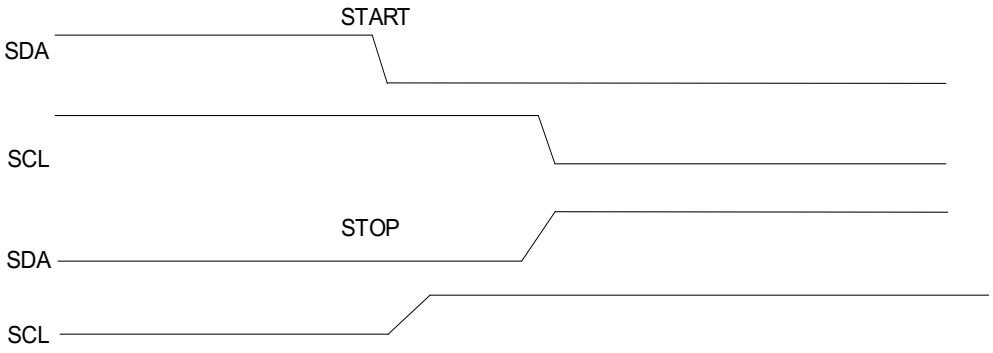
图 21-15. 数据有效性



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 21-16. 开始和停止信号*)。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 结束位定义为，在 SCL 为高时，SDA 线上出现一个从低到高的电平转换。

图 21-16. 开始和停止信号



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。在默认情况下，I2C 设备工作在从机模式下。当 START 信号产生时，I2C 设备由从机模式切换成主机模式。如果仲裁丢失或者 STOP 信号产生时，I2C 由主机模式切换成从机模式。支持 I2C 多主机模式。

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 模块支持 7 位和 10 位的地址模式。

数据和地址都是 8 位传输，高位在前。START 信号之后的字节(在 7 位地址模式下是一个字节，10 位地址模式下是两个字节)是主机发送的从机地址。

8 个时钟周期字节发送后，第 9 个时钟脉冲期间接收器会发送应答信号至发送器。是否产生 ACK 信号可以软件配置。

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。

在主机模式下，如果 AUTOEND=1，STOP 信号由硬件产生。如果 AUTOEND=0，STOP 信号由软件产生，或者主机可以产生 RESTART 信号来启动新的数据传输。

图 21-17. 10 位地址的 I2C 通讯流程(主机发送)

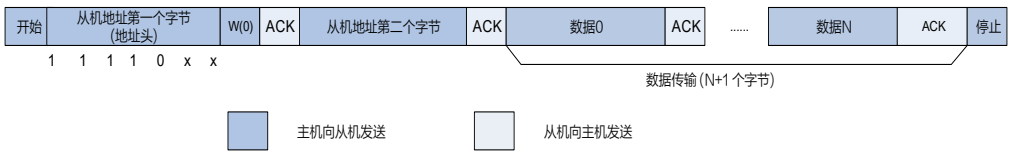


图 21-18. 7 位地址的 I2C 通讯流程(主机发送)

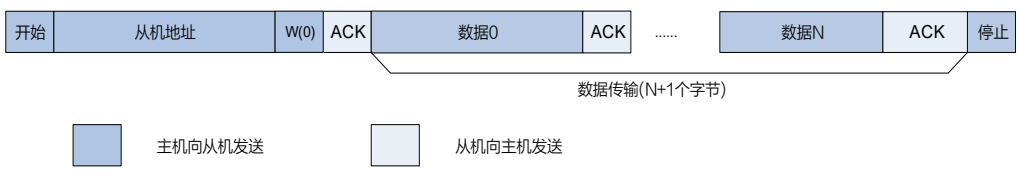


图 21-19. 7 位地址的 I2C 通讯流程(主机接收)



在 10 位寻址模式中，配置 HEAD10R 位可以选择执行完整的寻址序列或只发送地址头。当 HEAD10R=0，执行完整的 10 位地址寻址读序列 START+10 位地址头(写)+第二个地址字节 +RESTART+10 位地址头(读)，如*图 21-20. 10 位地址的 I2C 通讯流程(主机接收，HEAD10R=0)* 所示。

在10位寻址模式中，如果主机接收是在主机发送结束后执行，读寻址序列可以是RESTART+10 位地址头(读)，如*图 21-21. 10 位地址的 I2C 通讯流程(主机接收，HEAD10R=1)*所示

图 21-20. 10 位地址的 I2C 通讯流程(主机接收，HEAD10R=0)

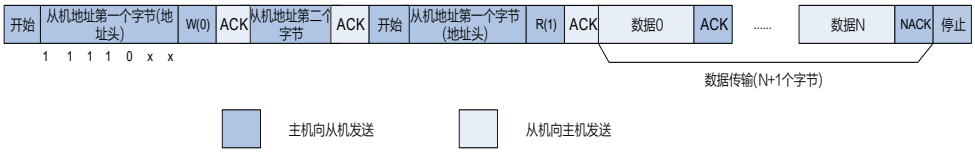
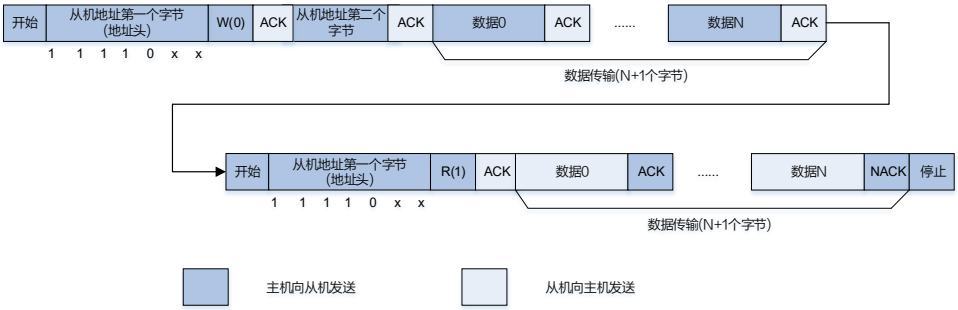


图 21-21. 10 位地址的 I2C 通讯流程(主机接收，HEAD10R=1)



噪声滤波器

I2C 外设集成了模拟噪声滤波器和数字噪声滤波器，噪声滤波器可根据实际需要在 I2C 外设启用前进行配置。

将 I2C_CTL0 寄存器中 ANOFF 位置 1 可以禁用模拟噪声滤波器，将 ANOFF 位清 0 时使能模拟噪声滤波器。在快速模式和快速+ 模式下，模拟滤波器需要抑制脉冲宽度高达 50ns 的峰值。

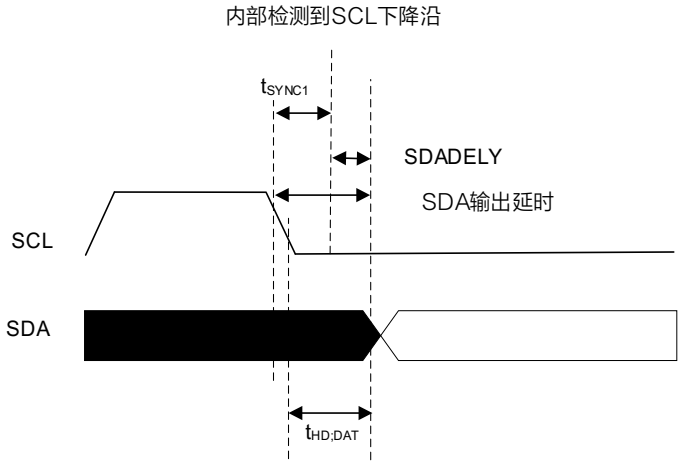
数字滤波器由 I2C_CTL0 寄存器中 DNF[3:0]位来配置。当数字滤波器使能时，SCL 和 SDA 电平保持稳定的时间大于DNF[3:0]×t_{I2CCLK}才会发生内部变化。抑制峰值宽度可由 DNF[3:0]配置。

I2C 时序配置

在 I2C 通信中，I2C_TIMING 寄存器中 PSC[3:0]，SCLDELY[3:0]和 SDADELY[3:0]用于保证正确的数据保持时间和数据建立时间。

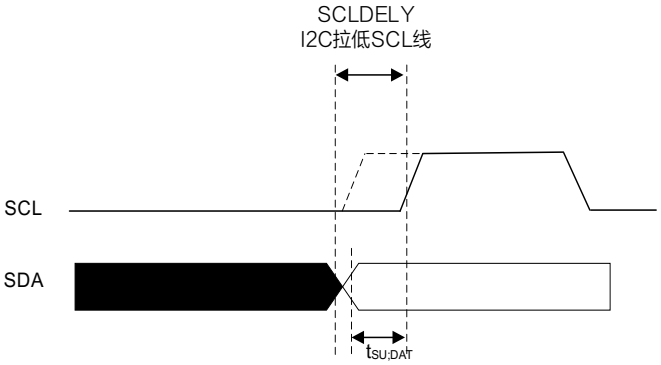
如果数据已经在 I2C_TDATA 寄存器中，在经历 SDADELY 延时后，数据由 SDA 发送，如*图 21-22. 数据保持时间*所示。

图 21-22. 数据保持时间



当数据经过 SDA 发送时，SCLDELY 计数器开启。如*图 21-23. 数据建立时间*所示。

图 21-23. 数据建立时间



当内部检测到 SCL 下降沿时，在 SDA 发送之前会插入一个延时。该延时为 $t_{SDADELY}$=SDADELY*t_{PSC}+t_{I2CCLK}，其中t_{PSC}=(PSC+1)*t_{I2CCLK}。$t_{SDADELY}$会影响$t_{HD;DAT}$。SDA 输出总延时为t_{SYNC1}+{[SDADELY*(PSC+1)+1]*t_{I2CCLK}}。t_{SYNC1}由 SCL 下降斜率，模拟滤波器延时，数字滤波器延时和 SCL 与 I2CCLK 时钟的同步延时共同决定。SCL 与 I2CCLK 时钟的同步延时为 2 至 3 个t_{I2CCLK}。

注意：当SDADELY=0 时，$t_{SDADELY}$ = t_{PSC}。

SDADELY 必须符合以下条件：

- SDADELY>={t_f(max)+$t_{HD;DAT}$(min)-t_{AF}(min)-[(DNF+3)*t_{I2CCLK}]}/[(PSC+1)*t_{I2CCLK}]
- SDADELY<={$t_{HD;DAT}$(max)-t_{AF}(max)-[(DNF+4)*t_{I2CCLK}]}/[(PSC+1)*t_{I2CCLK}]

注意：t_{AF}为模拟滤波器延时，$t_{HD;DAT}$必须小于$t_{VD;DAT}$的最大值。

当 SS=0 时，经过延时$t_{SDADELY}$，在数据写入 I2C_TDATA 寄存器之前，从机会拉低时钟线。在数据建立时间期间 SCL 保持低电平。数据建立时间$t_{SCLDELY}$=(SCLDELY+1)*t_{PSC}。$t_{SCLDELY}$影

响$t_{SU;DAT}$。

SCLDELY 必须符合以下条件：

- SCLDELY>={[t_r(max)+$t_{SU;DAT}$(min)]/[(PSC+1)*t_{I2CCLK}]}-1

在主机模式下，SCL 时钟高低电平由 I2C_TIMING 寄存器中 PSC[3:0]，SCLH[7:0]和 SCLL[7:0]控制。

当内部检测到 SCL 下降沿，在释放 SCL 输出之前会插入一个延时，该延时为 t_{SCLL}=(SCLL+1)*t_{PSC}，其中t_{PSC}=(PSC+1)*t_{I2CCLK}。t_{SCLL}影响 SCL 低电平持续时间t_{LOW}。

当内部检测到 SCL 上升沿，在将 SCL 拉低之前会插入一个延时，该延时为 t_{SCLH}=(SCLH+1)*t_{PSC}，其中t_{PSC}=(PSC+1)*t_{I2CCLK}。t_{SCLH}影响 SCL 高电平持续时间t_{HIGH}。

注意：时序配置和 SS 位在 I2C 外设使能时是不能改变的。

表 21-5. 数据建立时间和数据保持时间

符号	参数	标准模式		快速模式		快速 +模式		SMBus		单位
		最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值	
$t_{HD;DAT}$	数据保持时间	0	-	0	-	0	-	0.3	-	us
$t_{VD;DAT}$	数据有效时间	-	3.45	-	0.9	-	0.45	-	-	
$t_{SU;DAT}$	数据建立时间	250	-	100	-	50	-	250	-	ns
t_r	SCL 和 SDA 信号上升时间	-	1000	-	300	-	120	-	1000	
t_f	SCL 和 SDA 信号下降时间	-	300	-	300	-	120	-	300	

I2C 复位

清除 I2C_CTL0 寄存器中 I2CEN 位可以实现软件复位。当软件复位产生时，SCL 和 SDA 均被释放。通信控制位和状态位也还原成复位值。软件复位对配置寄存器无影响。受到影响的位为 I2C_CTL1 寄存器中 START，STOP 和 NACKEN，I2C_STAT 寄存器中 I2CBSY，TBE，TI，RBNE，ADDSEND，NACK，TCR，TC，STPDET，BERR，LOSTARB 和 OUERR。另外，如果支持 SMBus 模式，I2C_CTL1 寄存器中 PECTRANS 位，I2C_STAT 寄存器中 PECERR，TIMEOUT 和 SMBALT 位也会受到影响。

为了实现软件复位，I2CEN 必须在至少 3 个 APB 时钟周期内保持低电平。可以通过以下写软件序列来保证软件复位：

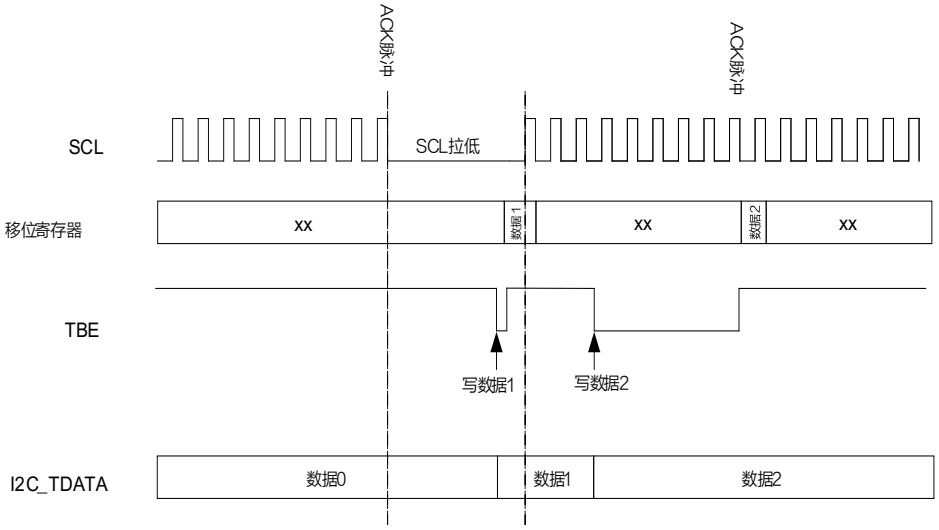
- I2CEN 写 0
- 检查 I2CEN 是否为 0
- I2CEN 写 1

数据传输

数据发送

在发送数据时，如果 TBE 为 0，表明 I2C_TDATA 寄存器非空，在第九个 SCL 脉冲(应答脉冲)后，I2C_TDATA 寄存器中的数据移入到移位寄存器。移位寄存器中的数据通过 SDA 线移出。如果 TBE 为 1，则表明 I2C_TDATA 寄存器为空，在 I2C_TDATA 不为空之前 SCL 将被拉低。SCL 拉低是在第九个 SCL 脉冲之后。

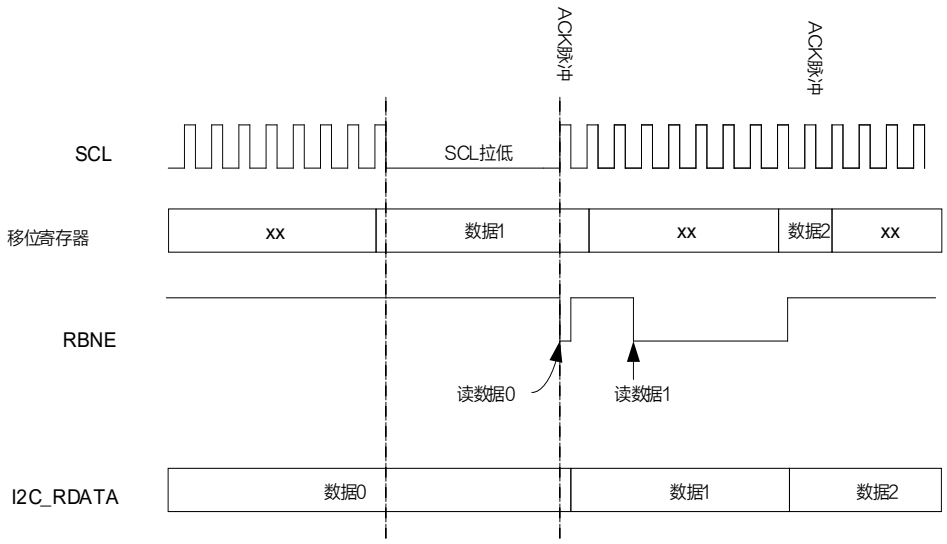
图 21-24. 数据发送



数据接收

在接收数据时，数据首先被接收到移位寄存器。如果 RBNE 为 0，移位寄存器中的数据将被移入 I2C_RDATA 寄存器。如果 RBNE 为 1，SCL 时钟将被拉低，直到之前接收到的数据字节被读取。这个时钟拉低被插入应答脉冲之前。

图 21-25. 数据接收



硬件传输管理重载和自动结束模式

为了管理字节传输和中断如*表 21-6. 可关闭通信模式*所示几种通信模式，I2C 硬件嵌入了字节计数器。

表 21-6. 可关闭通信模式

工作模式	行为
主机模式	产生 NACK，STOP 和 RESTART
从机接收模式	ACK 控制
SMBus 模式	PEC 生成/校验

传输的字节数由 BYTENUM[7:0]在 I2C_CTL1 寄存器中配置。如果 BYTENUM 大于 255，或者处于从机字节控制模式，则必须通过将 I2C_CTL1 寄存器中 RELOAD 位置 1 来使能重载模式。在重载模式下，当 BYTENUM 计数到 0 时，TCR 位将置 1，如果 TCIE 位置 1 将产生中断。当 TCR 位置 1 时，SCL 将被拉低。在 BYTENUM 写一个非零值将清除 TCR 位。

注意：重载模式必须在 BYTENUM[7:0]最后一次重载后禁用。

当使能自动结束模式时，必须禁用重载模式。在自动结束模式下，当 BYTENUM[7:0]计数到 0 时，主机将自动发送一个 STOP 信号。

当重载模式和自动结束模式都被禁用时，I2C 通信进程需要由软件终止。如果 BYTENUM[7:0]中的字节数已经传输完成，软件应将 STOP 位置 1 来产生一个 STOP 信号，然后清除 TC。

I2C 从机模式

初始化

从机模式下，至少使能一个从机地址。第一个从机地址写在 I2C_SADDR0 寄存器中，第二个从机地址写在 I2C_SADDR1 寄存器中。在使用从机地址时，必须相应地将 I2C_SADDR0 寄存器中 ADDRESSEN 位和 I2C_SADDR1 寄存器中 ADDRESS2EN 置 1。通过设置 I2C_SADDR0 寄存器中 ADDFORMAT 位可以选择 7 位地址或 10 位地址，该地址被写在 ADDRESS[9:0]。

I2C_CTL2 寄存器中 ADDM[6:0]定义 ADDRESS[7:1]的哪些位和接收到的地址进行比较，哪些位不比较。

ADDMSK2[2:0]用于屏蔽 I2C_SADDR1 寄存器中 ADDRESS2[7:1]，相关详细信息参考 I2C_SADDR1 寄存器 ADDMSK2[2:0]位域描述。

当 I2C 接收到的地址与使能的地址其中一个匹配成功时，ADDSEND 将被置 1，如果 ADDMIE 置位，将产生中断。I2C_STAT 寄存器 READDR[6:0]将会存储接收到的地址。在 ADDSEND 置位时，I2C_STAT 寄存器中 TR 位状态更新。TR 的状态指示从机是作为发送器还是接收器。

SCL 线控制

当 SS=0 时，时钟拉低功能默认用在从机模式下，在需要的时候 SCL 会被拉低。在下列情况下，SCL 会被拉低。

- 当 ADDSEND 置位时 SCL 线拉低，并在 ADDSEND 位清零之后释放。

- 在从机发送模式下，ADDSEND 清零之后，SCL 在第一个字节写入 I2C_TDATA 寄存器之前都是被拉低的。在前一个字节发送完成之后，新的字节写入 I2C_TDATA 寄存器之前，SCL 也是被拉低的。
- 在从机接收模式下，接收过程已完成但是 I2C_RDATA 寄存器中的数据还未被读取，SCL 将被拉低。
- 当 SBCTL=1 且 RELOAD=1 时，在最后一个字节传输结束后，TCR 置位。在 TCR 清除之前 SCL 将被拉低。
- SCL 下降沿被检测到之后，在[(SDADELY+SCLDELY+1)*(PSC+1)+1]*t_{I2CCLK}期间 SCL 被拉低。

SCL 线控制可以通过将 I2C_CTL0 寄存器中 SS 位置 1 来禁能。在下列情况下，SCL 不会被拉低。

- 在 ADDSEND 置位时 SCL 将不会被拉低。
- 在从机发送模式下，数据必须在它传输过程产生的第一个 SCL 脉冲之前写入 I2C_TDATA 寄存器。否则 I2C_STAT 寄存器中 OUERR 位将会置 1，如果 ERRIE 位也被置 1，将产生一个中断。当 STPDET 位置 1 并且第一个数据开始发送，I2C_STAT 寄存器中 OUERR 位也将置 1。
- 在从机接收模式下，数据必须在下一个字节接收产生的第九个 SCL 脉冲(ACK 脉冲)之前读取。否则 I2C_STAT 寄存器中 OUERR 位也将置 1。如果 ERRIE 位也被置 1，将产生一个中断。

从机字节控制模式

在从机接收模式下要实现字节 ACK 控制，可以通过将 I2C_CTL0 寄存器中 SBCTL 位置 1 来使能从机字节控制模式。当 SS=1 时，从机字节控制模式无效。

在使用从机字节控制模式时，必须通过置位 I2C_CTL1 寄存器中 RELOAD 位来使能重载模式。从机字节控制模式中，在 ADDSEND 中断服务程序中 I2C_CTL1 寄存器中 BYTENUM[7:0]必须配置为 1，并且在每个字节接收完成时重载为 1。当接收到一个字节时，I2C_STAT 寄存器中 TCR 位置 1，在第八个和第九个 SCL 时钟脉冲之间从机将 SCL 时钟拉低。然后数据可以从 I2C_RDATA 寄存器中读取出来，通过配置 I2C_CTL1 寄存器中 NACKEN 位，从机可以决定发送 ACK 或者是 NACK。当在 BYTENUM[7:0]写入非零值时，从机释放 SCL 时钟线。

当 BYTENUM[7:0]大于 0x1 时，在 BYTENUM[7:0]数据接收期间，数据流是连续的。

注意：在下列情况下，可以配置 SBCTL 位：

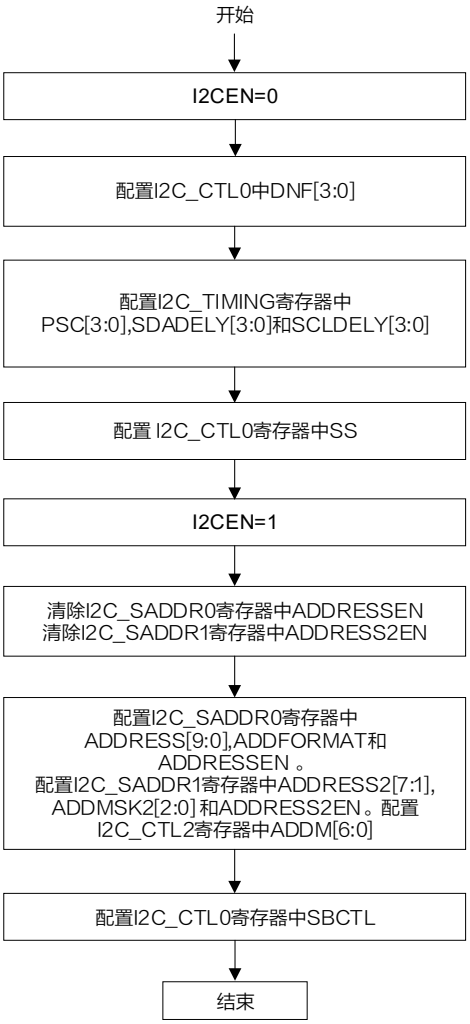
1、 I2CEN=0。

2、 从机还未被寻址。

3、 ADDSEND=1。

当 ADDSEND=1，或者 TCR=1 时，RELOAD 才可以被修改。

图 21-26. I2C 从机初始化



从机发送模式下的软件流程

当 I2C_TDATA 寄存器为空，I2C_STAT 寄存器中 TI 位将会置位。如果 I2C_CTL0 寄存器中 TIE 位置 1，将产生中断。当接收到 NACK 时，I2C_STAT 寄存器中 NACK 位会置位。如果 I2C_CTL0 寄存器中 NACKIE 位置 1，将产生中断。当接收到 NACK 信号时，I2C_STAT 寄存器中 TI 位将不会置位。

当接收到 STOP 信号时，I2C_STAT 寄存器中 STPDET 位将置 1。如果 I2C_CTL0 寄存器中 STPDETIE 位置 1，将产生中断。

当 SBCTL=0 时，如果 ADDSEND=1，且 I2C_STAT 寄存器中 TBE 位为 0，可以选择发送 I2C_TDATA 寄存器中的数据或者是将 TBE 置 1 来清空 I2C_TDATA 寄存器。

当 SBCTL=1 时，从机工作在字节控制模式，BYTENUM[7:0]必须在 ADDSEND 中断服务程序中配置。TI 事件的数量与 BYTENUM[7:0]的值相等。

当 SS=1 时，I2C_STAT 寄存器中 ADDSEND 位置位时 SCL 时钟线不会被拉低。在这种情况下，I2C_TDATA 寄存器中数据不能在 ADDSEND 中断服务程序中清空。因此待发送的第一个字节应该在 ADDSEND 置位之前就被编程到 I2C_TDATA 寄存器。

- 该数据可以是上一次数据传输最后一次 TI 事件写入的数据。
- 如果该数据不是待发送数据，可通过将 TBE 位置 1 来刷新 I2C_TDATA 寄存器，从而编程新的数据。在数据发送开始时 STPDET 位必须为 0。否则 I2C_STAT 寄存器中 OUERR 位将置 1 并产生下溢错误。
- 从机发送模式下使用中断或者 DMA 时，如果需要一个 TI 事件，TI 位和 TBE 位都必须置 1。

图 21-27. I2C 从机发送编程模型(SS=0)

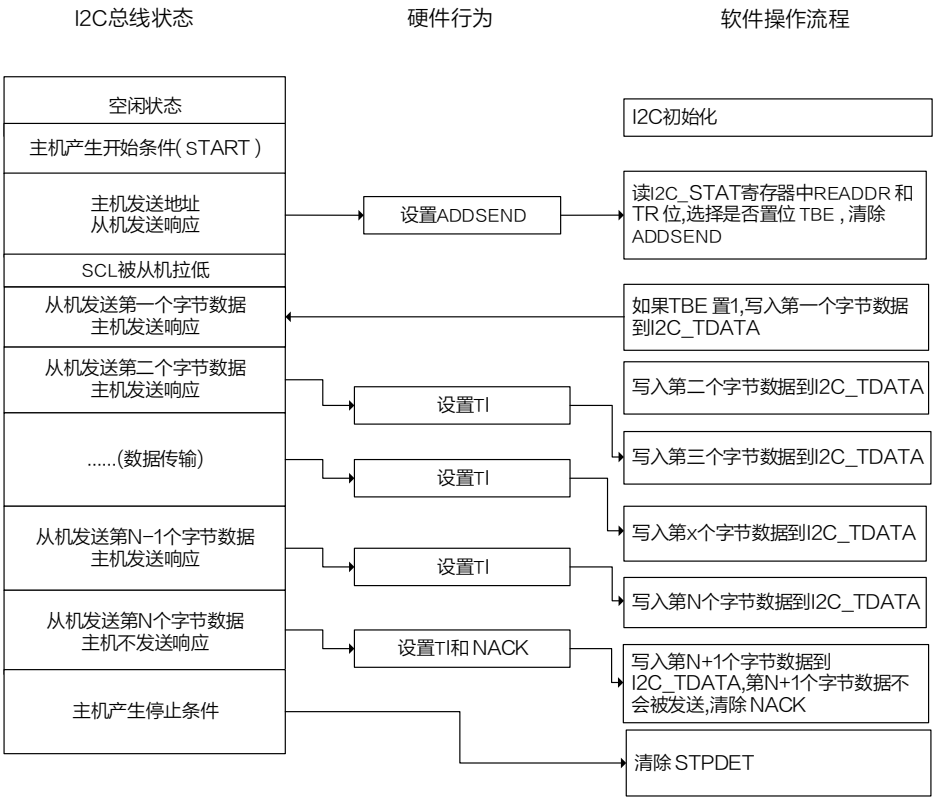
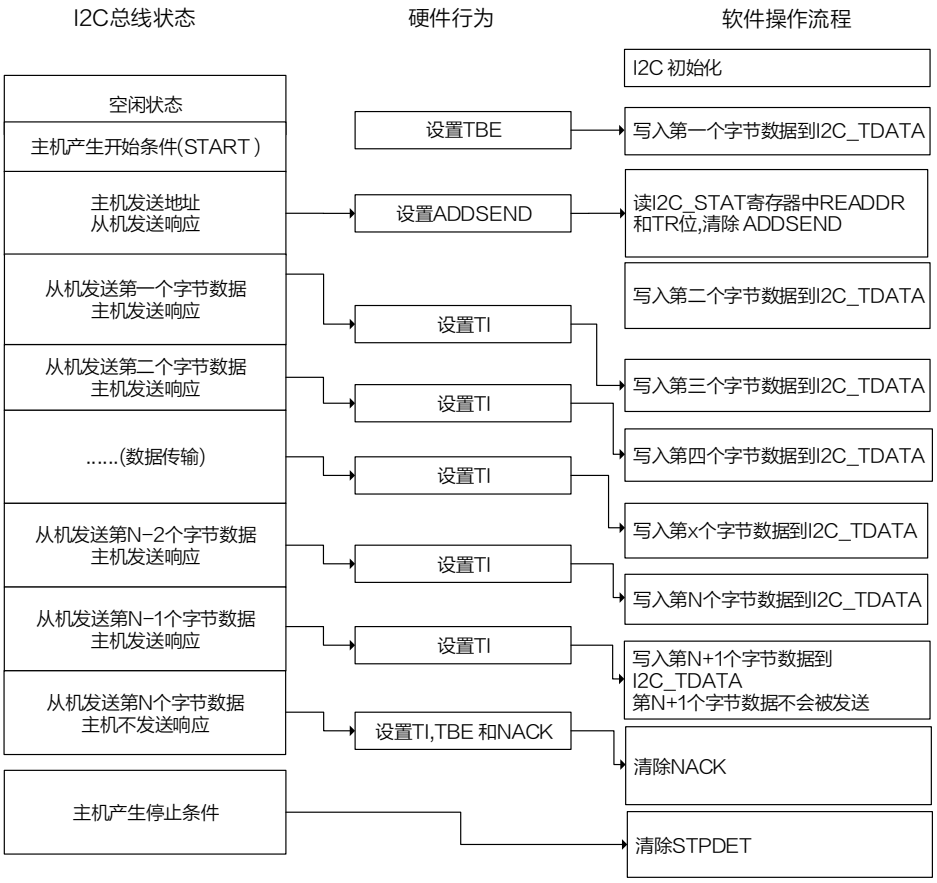


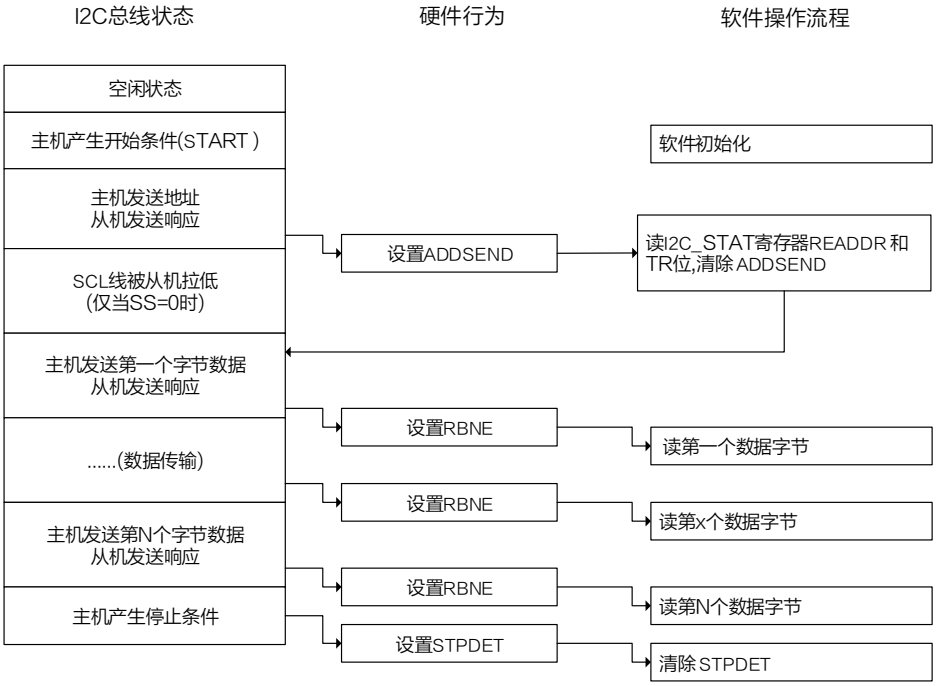
图 21-28. I2C 从机发送编程模型(SS=1)



从机接收模式下的软件流程

当 I2C_RDATA 寄存器非空，I2C_STAT 寄存器中 RBNE 位置 1，如果 I2C_CTL0 寄存器中 RBNEIE 位置 1，将产生中断。当接收到 STOP 信号时，I2C_STAT 寄存器中 STPDET 位将置 1。如果 I2C_CTL0 寄存器中 STPDETIE 置 1，将产生中断。

图 21-29. I2C 从机接收编程模型



I2C 主机模式

初始化

I2C_TIMING 寄存器中 SCLH[7:0]和 SCLL[7:0]必须在 I2CEN=0 时配置。为了支持多主机通信和从机时钟拉低，I2C 实现了时钟同步机制。

SCLL[7:0]和 SCLH[7:0]分别用于低电平计数和高电平计数。经过t_{SYNC1}延时后，当检测到 SCL 低电平时，SCLL[7:0]开始计数，如果 SCLL[7:0]计数器的值达到 I2C_TIMING 寄存器中 SCLL[7:0]时，I2C 将释放 SCL 时钟。经过t_{SYNC2}延时后，当检测到 SCL 高电平时，SCLH[7:0]开始计数，如果 SCLH[7:0]计数器的值达到 I2C_TIMING 寄存器中 SCLH[7:0]时，I2C 将拉低 SCL 时钟。

因此主机时钟周期为：$t_{SCL}=t_{SYNC1}+t_{SYNC2}+\{[(SCLH[7:0]+1)+(SLLL[7:0]+1)]*(PSC+1)*t_{I2CCLK}\}$。

t_{SYNC1}取决于 SCL 下降沿斜率，SCL 输入模拟和数字噪声滤波器延时以及 SCL 与 I2CCLK 时钟的同步产生的延时，一般为 2 到 3 个 I2CCLK 时钟周期。t_{SYNC2}取决于 SCL 上升沿斜率，SCL 输入模拟和数字噪声滤波器延时以及 SCL 与 I2CCLK 时钟的同步产生的延时，一般为 2 到 3 个 I2CCLK 时钟周期。数字噪声滤波器产生的延时为DNF[3:0]*t_{I2CCLK}。

在主机模式下，必须配置 I2C_CTL1 寄存器中 ADD10EN，SADDRESS[9:0]以及 TRDIR 位。当在主机接收模式下使用 10 位寻址时，必须配置 HEAD10R 来选择是执行完整的地址寻址序列，还是只发送地址头。待传输的字节数在 I2C_CTL1 寄存器 BYTENUM[7:0]配置。如果待传输的字节数大于或者等于 255，必须将 BYTENUM[7:0]配置为 0xFF。然后主机发送 START 信号。以上提到的所有位必须在 START 位置 1 之前配置。START 信号发送完成之后，待 I2C_STAT 寄存器 I2CBSY 位为 0 时，发送从机地址。当仲裁丢失时，主机切换成从机模式，START 位

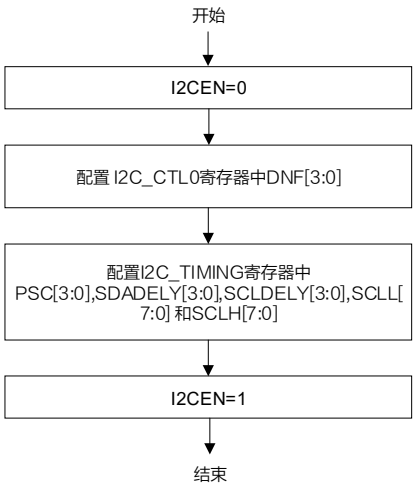
由硬件清零。当从机地址发送完成时，START 位由硬件清零。

注意：BYTENUM[7:0]不能配置为 0。

在 10 位寻址模式下，在发送 10 位地址头之后，如果主机接收到 NACK，主机将重发 10 位地址头直到收到 ACK。将 ADDSENDC 置 1 可以停止重发从机地址。

如果 START 位置 1 时，I2C 作为从机被寻址成功，ADDSEND 置 1，主机将切换为从机模式。START 位将在 ADDSENDC 置 1 时清零。

图 21-30. I2C 主机初始化



主机发送模式下的软件流程

在主机发送模式下，每一个字节发送完成并接收到 ACK 信号之后，TI 位将置 1。如果 I2C_CTL0 寄存器中 TIE 位置 1，将产生中断。待发送的字节数编程在 I2C_CTL1 寄存器 BYTENUM[7:0]。如果发送字节数大于 255，必须通过将 I2C_CTL1 寄存器 RELOAD 位置 1 来使能重载模式。在重载模式下，当 BYTENUM[7:0]个字节传输完成，I2C_STAT 寄存器 TCR 位将置 1，并且在 BYTENUM[7:0]更新一个非零值之前，SCL 被拉低。

如果接收到 NACK，TI 位将不会置 1。

- 如果 BYTENUM[7:0]个字节传输完成且 RELOAD=0，将 I2C_CTL1 寄存器中 AUTOEND 置 1 可以自动产生 STOP 信号。当 AUTOEND=0 时，I2C_STAT 寄存器 TC 位将置 1 且 SCL 被拉低。在这种情况下，主机可以通过将 I2C_CTL1 寄存器中 STOP 位置 1 来产生 STOP 信号。或者产生 RESTART 信号来开始一个新的数据传输过程。将 START / STOP 置 1 可以清除 TC 位。
- 如果接收到 NACK 信号，I2C 将自动产生 STOP 信号。I2C_CTL0 寄存器中 NACK 将置 1，如果 NACKIE 位置 1，将产生中断。

注意：当 RELOAD=1 时，AUTOEND 位无效。

图 21-31. I2C 主机发送编程模型(N<=255)

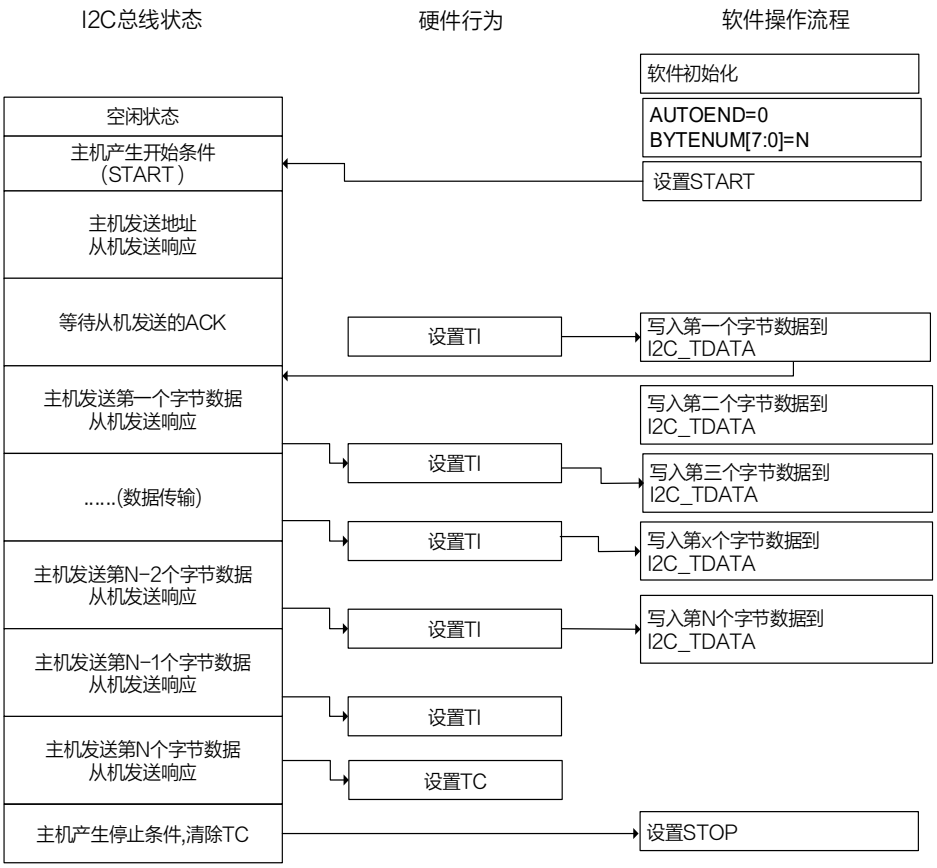


图 21-32. I2C 主机发送编程模型(N>255)



主机接收模式下的软件流程

在主机接收模式下，当接收到一个字节时，I2C_STAT 寄存器中 RBNE 位置 1。如果 I2C_CTL0 寄存器中 RBNEIE 置 1，将产生一个中断。如果待接收字节数大于 255，必须将 I2C_CTL1 寄存器中 RELOAD 位置 1 来使能重载模式。在重载模式下，当 BYTENUM[7:0]个字节传输完成，I2C_STAT 寄存器中 TCR 位将置 1，在 BYTENUM[7:0]中写入一个非零值之前，SCL 被拉低。

如果 BYTENUM[7:0]个字节传输完成且 RELOAD=0，将 I2C_CTL1 寄存器中 AUTOEND 置 1 可以自动产生 STOP 信号。当 AUTOEND=0 时，I2C_STAT 寄存器 TC 位将置 1 且 SCL 被拉低。在这种情况下，主机可以通过将 I2C_CTL1 寄存器中 STOP 位置 1 来产生 STOP 信号。或者产生 RESTART 信号来开始一个新的数据传输过程。将 START / STOP 置 1 可以清除 TC 位。

图 21-33. I2C 主机接收编程模型(N<=255)

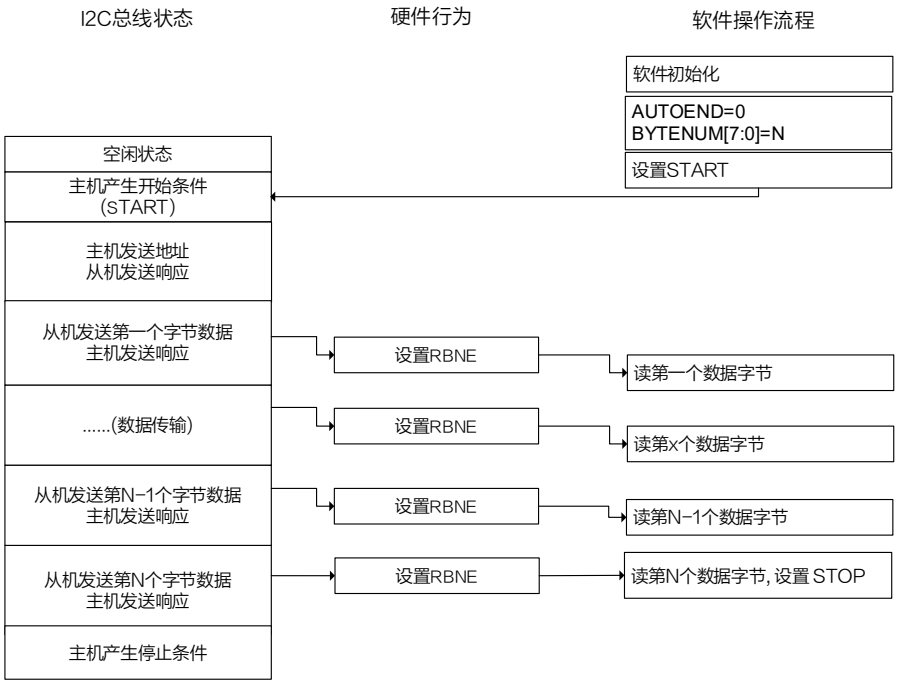


图 21-34. I2C 主机接收编程模型(N>255)



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 采用了 I2C 硬件以及 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界面。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。

SMBus 从机字节控制

SMBus 接收器从机字节控制与 I2C 一样。它允许 ACK 控制每个字节。必须能对接收到的命令或者数据进行 NACK 应答。通过将 I2C_CTL0 寄存器中 SBCTL 位置 1 来使能从机字节控制模式。

主机通知协议

通过将 I2C_CTL0 寄存器 SMBHAEN 位置 1，SMBus 可以支持主机通知协议。在该协议中，从设备作为主机，主设备作为从机，主机将应答 SMBus 主机地址。

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 25~35ms。按照 SMBus 协议的假定，如果某个会话耗时太久，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)状态。这样就并不允许从设备将时钟拉低太长时间。

将 I2C_TIMEOUT 寄存器中 TOEN 位和 EXTOEN 位置 1 可以使能超时检测。配置定时器必须保证在 SMBus 规范规定的时间最大值之前检测出超时情况。

在 BUSTOA[11:0]中编程的值被用来检查$t_{TIMEOUT}$参数。必须将 TOIDLE 位配置为 0，以检测 SCL 低电平超时。将 I2C_TIMEOUT 寄存器中 TOEN 位置 1 来使能定时器，在 TOEN 置 1 之后，BUSTOA[11:0] 和 TOIDLE 位不能被修改。如果 SCL 低电平时间大于(BUSTOA+1)*2048*t_{I2CCLK}，I2C_STAT 寄存器中 TIMEOUT 位将置 1。

BUSTOB[11:0]为从机校验$t_{LOW:SEXT}$，为主机校验$t_{LOW:MEXT}$。通过将 I2C_TIMEOUT 寄存器中 EXTOEN 位置 1 来使能定时器。在 EXTOEN 置 1 之后，BUSTOB[11:0]不能被修改。如果 SMBus 外设 SCL 拉低时间大于(BUSTOB+1)*2048*t_{I2CCLK}，并且达到了总线空闲检测章节中描述的超时时间间隔，I2C_STAT 寄存器中 TIMEOUT 位将置 1。

报文错误校验

I2C 模块中有一个 PEC 模块，它使用 CRC-8 计算器来执行 I2C 数据的报文校验。一个 PEC 字节(PEC 错误码)附加在每次传输结束。PEC 的计算方式是对所有消息字节(包含地址和读/写位)使用 CRC-8 计算校验和。CRC-8 多项式位x^8+x^2+x+1(CRC-8-ATM HEC 算法，初始化为 0)。

当 I2C 被禁用时，通过 I2C_CTL0 寄存器中的 PECEN 位置 1 可以使能 PEC。由于 PEC 传输是由 I2C_CTL1 寄存器中 BYTENUM[7:0]管理的，因此在从机模式下必须将 SBCTL 位置 1。当 PECTRANS 置 1，RELOAD 为 0 时，在 BYTENUM[7:0]-1 数据字节后发送 PEC。PEC 在 BYTENUM[7:0]-1 传输完成后发送。当 RELOAD 置 1 时 PECTRANS 无效。

SMBus 警报

SMBus 还有一个额外的共享的中断信号，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。主机会处理该中断，并通过报警响应地址，同时访问所有 SMBALERT#设备。如果 SMBALERT#电平被设备拉低，这些设备会应答报警响应地址。当配置为从设备(SMBHAEN=0)时，通过将 I2C_CTL0 寄存器中 SMBALTEN 置 1 可以将 SMBA 引脚电平拉低。同时也使能了报警响应地址。当配置为主设备(SMBHAEN=1)，且 SMBALTEN 置 1 时，当在 SMBA 引脚检测到下降沿时，I2C_STAT 寄存器中 SMBALT 位将置 1。如果 I2C_CTL0 寄存器中 ERRIE 位置 1，将产生中断。当 SMBALTEN = 0 时，即使外部 SMBA 引脚为低电平，ALERT 线也将被视为高电平。当 SMBALTEN=0 时，SMBA 引脚可用作标准 GPIO。

总线空闲检测

如果主机检测到时钟信号和数据信号的高电平持续时间大于$t_{HIGH,MAX}$，总线被视为空闲。

该时序参数已考虑到主机已动态添加至总线，但可能还未检测到 SMBCLK 或 SMBDAT 线上的状态转换的情况。在这种情况下，为了保证当前没有数据传输正在进行，主机必须等待足够长的时间。

要使能t_{IDLE}检查，必须将 BUSTOA[11:0]编程为定时器重载值，以获取t_{IDLE}参数。必须将 TIDLE 位置 1，以检测 SCL 和 SDA 高电平超时。然后通过将 I2C_TIMEOUT 寄存器中的 TOEN 位置 1 来使能定时器。TOEN 置 1 后，BUSTOA[11:0]和 TIDLE 不能被修改。如果 SCL 和 SDA 的高电平持续时间都大于(BUSTOA+1)*4*t_{I2CCLK}，I2C_STAT 寄存器中 TIMEOUT 位将置位。

SMBus 从机模式

SMBus 接收器必须能够对接收到的命令和数据进行 NACK 应答。对于从机模式下的 ACK 控制，通过将 I2C_CTL0 寄存器中 SBCTL 位置 1 可以使能从机字节控制模式。

必要时应使能特定的 SMBus 地址。通过将 I2C_CTL0 寄存器中 SMBDAEN 置 1 可以使能 SMBus 设备默认地址(0b1100 001)。通过将 I2C_CTL0 寄存器中 SMBHAEN 置 1 可以使能 SMBus 主机地址(0b0001 000)。通过将 I2C_CTL0 寄存器中 SMBALTEN 置 1 可以使能报警响应地址(0b0001 100)。

SMBus 模式

SMBus 主机发送器和从机接收器

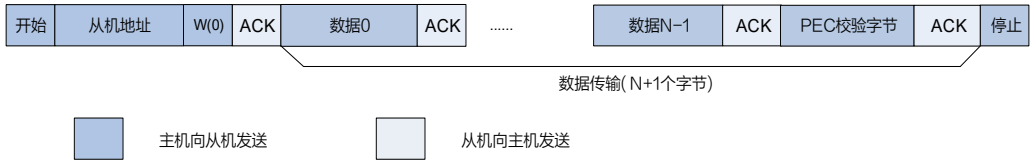
当 SMBus 主机发送 PEC 时，必须在 START 位置 1 前，将 PECTRANS 位置 1 并在 BYTENUM[7:0]位域中配置字节数。在这种情况下，总 TI 中断数为 BYTENUM-1。因此，如果 BYTENUM=0x1 且 PECTRANS 位置 1，则 I2C_PEC 寄存器的数据将自动发送。如果 AUTOEND 为 1，SMBus 主机在 PEC 字节发送完成之后将自动发送 STOP 信号。如果 AUTOEND 为 0，SMBus 主机可以在 PEC 字节发送完成之后发送 RESTART 信号。I2C_PEC

寄存器中的数据将在 BYTENUM -1 个字节发送完成后发送，PEC 字节发送完成后 TC 位将置 1。SCL 线被拉低。RESTART 位必须在 TC 中断服务程序中置 1。

SMBus 作为从机接收器时，为了在数据发送完成时进行 PEC 校验，SBCTL 位必须置 1。要对每个字节进行 ACK 控制，必须通过将 RELOAD 位置 1 来使能 RELOAD 模式。如果要校验 PEC 字节，必须将 RELOAD 位清零同时将 PECTRANS 置 1。在 BYTENUM-1 个字节接收完成后，接收的下一个字节将与 I2C_PEC 寄存器中的数据进行比较。如果校验值不匹配，将自动产生 NACK 信号；如果校验值匹配将自动产生 ACK 信号，将忽略 NACKEN 位的值。当接收到 PEC 字节时，PEC 字节会存到 I2C_RDATA 寄存器中，RBNE 位将置 1。如果 I2C_CTL0 寄存器中 ERRIE 位置 1，且 PEC 值不匹配，PECERR 将会置 1 并产生中断。如果无须使用 ACK 控制，PECTRANS 可以设置为 1，BYTENUM 可以根据待接收字节数来配置。

注意：在 RELAOD 位置 1 之后，PECTRANS 不可以被修改。

图 21-35. SMBus 主机发送器和从机接收器通信流程



SMBus 主机接收器和从机发送器

如果 SMBus 主机需要在数据传输完成后接收 PEC 字节，可以使能自动结束模式。在 START 信号发送之前，必须将 PECTRANS 位置 1，且配置好从机地址。在接收 BYTENUM-1 数据之后，接收的下一个字节将自动与 I2C_PEC 寄存器中的数据进行比较。在停止信号发送之前，接收 PEC 字节之后会给出 NACK 响应。

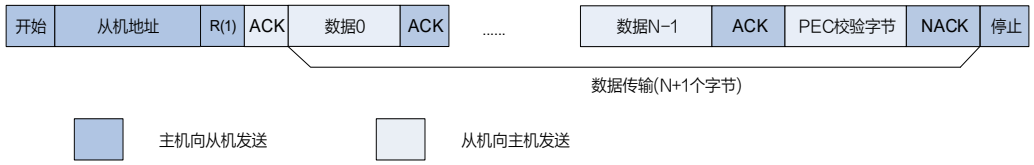
如果 SMBus 主机需要在接收到 PEC 字节之后产生 RESTART 信号，需要禁能自动结束模式。在 START 信号发送之前，PECTRANS 位必须置 1，且配置好从机地址。在接收 BYTENUM-1 数据之后，接收的下一个字节将自动与 I2C_PEC 寄存器中的数据进行比较。在 PEC 字节发送完成之后 TC 位将置 1，SCL 线被拉低。在 TC 中断服务程序中可将 RESTART 位置 1。

当 SMBus 作为从机发送器时，为了在 BYTENUM[7:0]个字节发送完成之后发送 PEC 字节，SBCTL 位必须置 1。如果 PECTRANS 置 1，字节数 BYTENUM[7:0]包含 PEC 字节。在这种情况下，如果主机请求接收的字节数大于 BYTENUM-1，总 TI 中断数为 BYTENUM-1，I2C_PEC 寄存器中的数据将自动发送。

注意：

1. PECTRANS 位在 RELOAD 置 1 之后不能被修改。
2. SMBus 作为从机发送器，并且使用 PEC 功能时，必须在检测到 START 信号前配置 BYTENUM[7:0]。

图 21-36. SMBus 主机接收器和从机发送器通信流程



从省电模式唤醒

当 I2C 地址匹配成功时，MCU 从睡眠模式，深度睡眠模式被唤醒。为了将 MCU 从这些省电模式唤醒，I2C_CTL0 寄存器中 WUEN 位必须置 1，同时 I2CCLK 时钟源选择 IRC8M。在深度睡眠模式下，IRC8M 关闭。当 I2C 检测到 START 信号时，IRC8M 打开，I2C 会将 SCL 拉低直到 IRC8M 被唤醒。在接收地址期间，IRC8M 为 I2C 提供时钟。当地址匹配时，在 MCU 唤醒期间，I2C 的 SCL 线被拉低。当 ADDSEND 清除时，SCL 线被释放，数据传输过程恢复正常。如果检测到的地址不匹配，IRC8M 会再次关闭，MCU 将不会被唤醒。

只有地址匹配中断(ADDMIE=1)能唤醒 MCU。如果 I2C 的时钟源是系统时钟，或者 WUEN=0，IRC8M 在接收到 START 信号之后将不会打开。当从省电模式唤醒使能时，数字滤波器必须禁能，I2C_CTL0 寄存器中 SS 位也必须清 0。如果禁止从省电模式唤醒(WUEN=0)，则在进入省电模式之前必须禁能 I2C 外设(I2CEN=0)。

DMA 模式下数据传输

如 I2C 从机模式和主机模式中描述，每当 TI 位和 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 TI 或 RBNE 位置 1 时，自动进行一次写或读操作。

将 I2C_CTL0 寄存器中 DENT 置 1 可以使能 DMA 发送请求。将 I2C_CTL0 寄存器中 DENR 置 1 可以使能 DMA 接收请求。在主机模式下，由软件写入从机地址，传输方向，待发送字节数和 START 位。DMA 必须在 START 位置 1 之前初始化。在 I2C_CTL1 寄存器 BYTENUM[7:0] 位配置待传输字节数。在从机模式下，DMA 必须在地址匹配事件发生之前或 ADDSEND 中断服务程序中清除 ADDSEND 标志之前完成初始化。

I2C 错误和中断

I2C 错误标志如*表 21-7. I2C 错误标志*所示。

表 21-7. I2C 错误标志

I2C 错误名称	描述
BERR	总线错误
LOSTARB	仲裁丢失
OUERR	上溢/下溢标志
PECERR	CRC 值不匹配
TIMEOUT	SMBus 模式下总线超时
SMBALT	SMBus 报警

I2C 中断和事件标志如*表 21-8. I2C 中断事件*所示。

表 21-8. I2C 中断事件

中断事件	事件标志	使能控制位
在接收期间 I2C_RDATA 非空	RBNE	RBNEIE
发送中断	TI	TIE
从机模式下检测到 STOP 信号	STPDET	STPDETIE
传输完成重载	TCR	TCIE
传输完成	TC	
地址匹配	ADDSEND	ADDMIE
接收到 NACK	NACK	NACKIE
总线错误	BERR	ERRIE
仲裁丢失	LOSTARB	
上溢/下溢错误	OUERR	
PEC 错误	PECERR	
超时错误	TIMEOUT	
SMBus 报警	SMBALT	

I2C 调试模式

当为控制器进入调试模式(Cortex®-M23 内核停止)，SMBus 超时定时器会根据 DBG 模块中的 I2C2_HOLD 配置位选择继续正常工作还是停止工作。

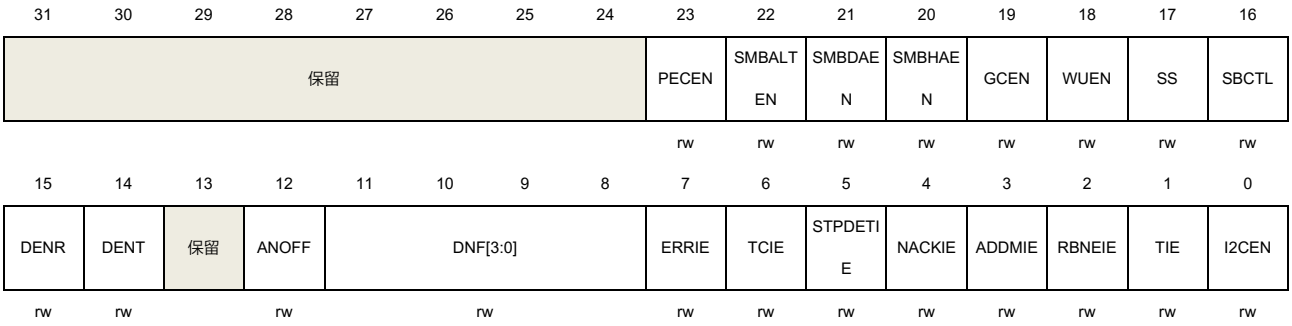
21.2.4. I2C 寄存器

I2C2基地址：0x4000 C000

控制寄存器 0(I2C_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
31:24	保留	必须保持复位值。
23	PECEN	PEC 计算开关。 0：PEC 计算关闭。 1：PEC 计算打开。 **注意：**如果不支持 SMBus，该位保留并被硬件强制置 0。
22	SMBALTEN	SMBus 报警使能。 0：从机模式下 SMBA 引脚高电平或主机模式下 SMBus 报警引脚 SMBA 禁能。 1：从机模式下 SMBA 引脚低电平或主机模式下 SMBus 报警引脚 SMBA 使能。 **注意：**如果不支持 SMBus，该位保留并被硬件强制置 0。
21	SMBDAEN	SMBus 设备默认地址使能。 0：设备默认地址禁能，对默认地址 0b1100001x 进行 NACK 应答。 1：设备默认地址使能，对默认地址 0b1100001x 进行 ACK 应答。 **注意：**如果不支持 SMBus，该位保留并被硬件强制置 0。
20	SMBHAEN	SMBus 主机地址使能。 0：主机地址禁能，对地址 0b0001000x 进行 NACK 应答。 1：主机地址使能，对地址 0b0001000x 进行 ACK 应答。 **注意：**如果不支持 SMBus，该位保留并被硬件强制置 0。
19	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。
18	WUEN	使能从省电模式中唤醒，包含睡眠模式和深度睡眠模式。

		0：禁止从省电模式中唤醒。 1：使能从省电模式中唤醒。 **注意：**当 DNF[3:0]=0 时，WUEN 才能被置 1。
17	SS	在从机模式下数据未就绪时是否将 SCL 拉低。 软件置 1 和清 0。 0：拉低 SCL 1：不拉低 SCL **注意：**在主机模式下，该位必须为 0。该位只能在 I2CEN=0 时被修改。
16	SBCTL	从机模式下字节控制。 该位用于在从机模式下使能硬件字节控制。 0：从机模式下字节控制禁能。 1：从机模式下字节控制使能。
15	DENR	DMA 接收使能 0：DMA 接收禁能 1：DMA 接收使能
14	DENT	DMA 发送使能 0：DMA 发送禁能 1：DMA 发送使能
13	保留	必须保持复位值。
12	ANOFF	模拟噪声滤波器禁能 0：模拟噪声滤波器使能。 1：模拟噪声滤波器禁能。 **注意：**该位只有在 I2C 禁能(I2CEN=0)时被编程。
11:8	DNF[3:0]	数字噪声滤波器 0000：数字噪声滤波器禁能。 0001：数字噪声滤波使能并且可以滤除脉宽宽度不大于 1 t_{I2CCLK}的尖峰。 ... 1111：数字噪声滤波使能并且可以滤除脉宽宽度不大于 15 t_{I2CCLK}的尖峰。 **注意：**这些位只能在 I2C 禁能(I2CEN=0)时修改。
7	ERRIE	错误中断使能 0：错误中断禁能 1：错误中断使能，当 BERR，LOSTARB，OUERR，PECERR，TIMEOUT 或 SMBALT 位置 1 时，将产生中断。
6	TCIE	传输完成中断使能 0：传输完成中断禁能。 1：传输完成中断使能。
5	STPDETIE	停止信号检测中断使能 0：停止信号(STPDET)检测中断禁能。 1：停止信号(STPDET)检测中断使能。

4	NACKIE	接收到 NACK 应答中断使能 0：接收到 NACK 应答中断禁能。 1：接收到 NACK 应答中断使能。
3	ADDMIE	从机模式下地址匹配中断使能 0：地址匹配中断禁能。 1：地址匹配中断使能。
2	RBNEIE	接收中断使能 0：接收(RBNE)中断禁能。 1：接收(RBNE)中断使能。
1	TIE	发送中断使能 0：发送中断(TI)禁能。 1：发送中断(TI)使能。
0	I2CEN	I2C 外设使能 0：I2C 禁能。 1：I2C 使能。

控制寄存器 1(I2C_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					PECTRANS	AUTOEND	RELOAD	BYTENUM[7:0]							
					rw	rw	rw	rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NACKEN	STOP	START	HEAD10R	ADD10EN	TRDIR	SADDRESS[9:0]									
rw	rw	rw	rw	rw	rw	rw									

位/位域	名称	描述
31:27	保留	必须保持复位值。
26	PECTRANS	PEC 传输 软件置 1 和清 0，硬件在以下条件下清除此位：PEC 传输完成或者 ADDSEND 置 1 或者检测到 STOP 信号或者 I2CEN=0。 0：不传输 PEC 值。 1：传输的 PEC 值。 **注意：** 当 RELOAD=1 或者从机模式下 SBCTL=0 时，该位无效。如果不支持 SMBus，该位保留并被硬件强制置 0。
25	AUTOEND	主机模式下自动结束模式

		0：当 BYTENUM[7:0]个字节传输完成后时，TC 位置 1。 1：当 BYTENUM[7:0]个字节传输完成后时，自动发送 STOP 信号。 **注意：**该位仅在 RELOAD=0 时有效。该位由软件置 1 和清 0。
24	RELOAD	重载模式使能 0：当 BYTENUM[7:0]个字节传输完成后时，传输结束。 1：当 BYTENUM[7:0]个字节传输完成后时，传输未结束，重载新的 BYTENUM[7:0]。每次 BYTENUM[7:0]个字节传输完成，I2C_STAT 寄存器中 TCR 位将置 1。 **注意：**该位由软件置 1 和清 0。
23:16	BYTENUM[7:0]	待传输的字节数 这些用来编程待传输的字节数。当 SBCTL=0 时，这些位无效。 **注意：**当 START 位置 1 时，该位不能被修改，传输过程中软件不能修改该位。
15	NACKEN	从机模式下产生 NACK 0：在接收到新的字节时，发送 ACK。 1：在接收到新的字节时，发送 NACK。 **注意：**该位可由软件置 1，并在以下情况下由硬件清零：NACK 发送完成或检测到 STOP 信号或 ADDSEND 置 1，或 I2CEN=0。当 PEC 使能时，发送 ACK 还是 NACK 与 NACKEN 值无关。当 SS=1 时，且 OUERR 位置 1，NACKEN 的值会被忽略，并且发送 NACK。
14	STOP	I2C 总线上产生一个 STOP 结束信号。 该位由软件置 1，并在 I2CEN=0 或检测到 STOP 信号时由硬件清零。 0：不发送 STOP。 1：发送 STOP。
13	START	I2C 总线上产生一个 START 信号 该位由软件置 1，并在从机地址发送后由硬件清零。当仲裁丢失时，或发生超时错误，或 I2CEN=0 时，该位也可以由硬件清零。将 I2C_STATC 寄存器中 ADDSENDC 位置 1 可以软件清除该位。 0：不发送 START。 1：发送 START。
12	HEAD10R	在主机接收模式下仅执行 10 位地址头读操作。 0：主机发送 10 位从机地址读序列为 START + 10 位地址头(写) + 第二个地址字节 + RESTART + 10 位地址头(读)。 1：主机寻址读序列为 RESTART + 10 位地址头(读)。 **注意：**当 START 位置 1 时，该位不能被修改，传输过程中软件不能修改该位。
11	ADD10EN	主机模式下使能 10 位寻址模式 0：主机工作在 7 位寻址模式下。 1：主机工作在 10 位寻址模式下。 **注意：**当 START 位置 1 时，该位不能被修改，传输过程中软件不能修改该位。
10	TRDIR	主机模式下传输方向 0：主机发送 1：主机接收

		注意：当 START 位置 1 时，该位不能被修改，传输过程中软件不能修改该位。
9:0	SADDRESS[9:0]	待发送的从机地址 SADDRESS[9:8]：从机地址 9:8 位。 如果 ADD10EN=0，该位域无效。 如果 ADD10EN=1，将该位域写入待发送从机地址的 9:8 位。 SADDRESS[7:1]：从机地址 7:1 位。 如果 ADD10EN=0，在这些位写入待发送 7 位从机地址。 如果 ADD10EN=1，在这些位写入待发送从机地址的 7:1 位。 SADDRESS0：从机地址 0 位。 如果 ADD10EN=0，这些位无效。 如果 ADD10EN=1，在这些位写入待发送从机地址的 0 位。 **注意：**当 START 位置 1 时，该位不能被修改，传输过程中软件不能修改该位。

从机地址寄存器 0(I2C_SADDR0)

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
31:16	保留	必须保持复位值。
15	ADDRESSEN	I2C 地址使能 0：I2C 地址禁能。 1：I2C 地址使能。
14:11	保留	必须保持复位值。
10	ADDFORMAT	I2C 从机地址模式 0：7 位地址。 1：10 位地址。 **注意：**当 ADDRESSEN=1 时，该位不能被改写。
9:8	ADDRESS[9:8]	10 位地址的最高两位 **注意：**当 ADDRESSEN=1 时，该位不能被改写。
7:1	ADDRESS[7:1]	7 位地址或者 10 位地址的第 7-1 位 **注意：**

1. 当 ADDRESSEN=1 时，该位域不能被改写。
2. 当从机地址设置为保留地址 0b1111000 时，如果地址匹配，从机不响应。

0	ADDRESS0	10 位地址的第 0 位 **注意：**当 ADDRESSEN=1 时，该位不能被改写。

从机地址寄存器 1(I2C_SADDR1)

地址偏移：0x0C
复位值：0x0000 0000

该寄存器只能按字(32位)访问。



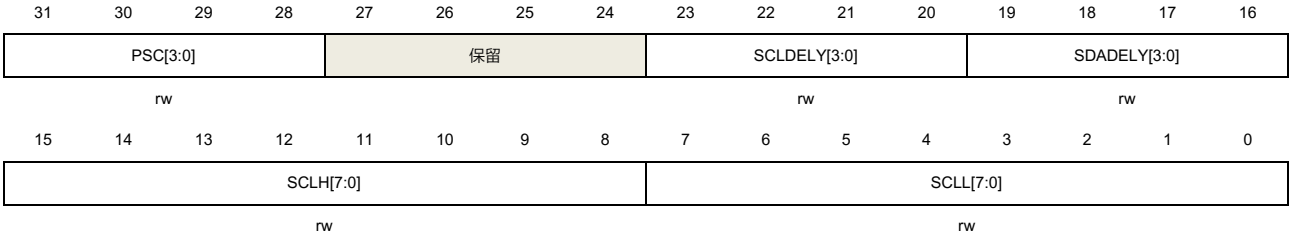
位/位域	名称	描述
31:16	保留	必须保持复位值。
15	ADDRESS2EN	I2C 第二个地址使能 0：I2C 第二个地址禁能 1：I2C 第二个地址使能
14:11	保留	必须保持复位值。
10:8	ADDMSK2[2:0]	ADDRESS2[7:1]掩码 定义接收到的地址哪些位需要与 ADDRESS2[7:1]进行比较，哪些位屏蔽(不比较)。 000：不屏蔽，所有的位都进行比较。 n(001~110)：ADDRESS2[n:1]屏蔽。ADDRESS2[7:n+1]需要进行比较。 111：ADDRESS2[7:1]屏蔽。对于接收到的所有 7 位地址都会进行 ACK 应答，保留地址(0b0000xxx 和 0b1111xxx)除外。 **注意：**当 ADDRESS2EN=1 时，该位不能被改写。如果 ADDMSK2 不等于 0，即使所有位都匹配，I2C 保留地址(0b0000xxx 和 0b1111xxx)也不会进行 ACK 应答。
7:1	ADDRESS2[7:1]	I2C 从机的第二个地址 **注意：** 1. 当 ADDRESS2EN=1 时，该位域不能被改写。 2. 当从机地址设置为保留地址 0b1111000 时，如果地址匹配，从机不响应。
0	保留	必须保持复位值。

时序寄存器(I2C_TIMING)

地址偏移：0x10
复位值：0x0000 0000

当 I2C(I2CEN=1)被使能时，该寄存器不能被改写。

该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31:28	PSC[3:0]	时序预分频 为了生成用于数据建立和数据保持的计数器的时钟周期t_{PSC}，这些位用于配置 I2CCLK 时钟预分频。t_{PSC}也用于 SCL 高电平和低电平计数器。 $t_{PSC}=(PSC+1)*t_{I2CCLK}$。
27:24	保留	必须保持复位值。
23:20	SCLDELY[3:0]	数据建立时间 这些位用于在 SDA 边沿和 SCL 上升沿之间生成延时$t_{SCLDELY}$。在主机模式下和在从机模式下 SS=0 时，在$t_{SCLDELY}$期间 SCL 线被拉低。 $t_{SCLDELY}=(SCLDELY+1)*t_{PSC}$。
19:16	SDADELY[3:0]	数据保持时间 这些位用于在 SCL 下降沿和 SDA 边沿之间生成延时$t_{SDADELY}$。在主机模式下和在从机模式下 SS=0 时，在$t_{SDADELY}$期间 SCL 线被拉低。 0x0：$t_{SDADELY}$ = 1*t_{PSC} 0x1-0xF：$t_{SDADELY}$=SDADELY*t_{PSC}
15:8	SCLH[7:0]	SCL 高电平周期 SCL 高电平周期可以通过配置这些位来产生。 $t_{SCLH}=(SCLH+1)*t_{PSC}$。 **注意：**这些位只能用于主机模式。
7:0	SCLL[7:0]	SCL 低电平周期 SCL 低电平周期可以通过配置这些位来产生。 $t_{SCLL}=(SCLL+1)*t_{PSC}$。 **注意：**这些位只能用于主机模式。

超时寄存器(I2C_TIMEOUT)

地址偏移：0x14

复位值：0x0000 0000

注意：如果不支持 SMBus，该寄存器保留并被硬件置为 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
EXTOEN	保留			BUSTOB[11:0]											
rw				rw											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOEN	保留		TOIDLE	BUSTOA[11:0]											
rw			rw	rw											

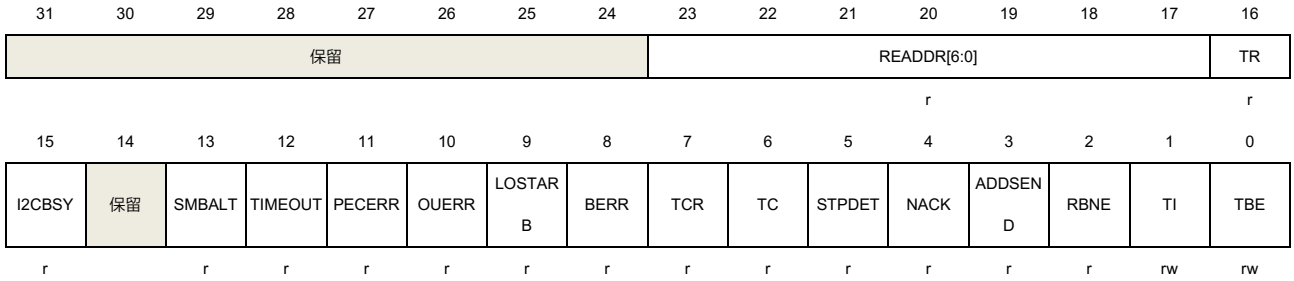
位/位域	名称	描述
31	EXTOEN	时钟信号延展超时使能 当 SCL 累计拉低时间大于 $t_{LOW:EXT}$ 时，将会产生超时错误，$t_{LOW:EXT}$=(BUSTOB+1)*2048*t_{I2CCLK}。 0：时钟信号延展超时检测禁能。 1：时钟信号延展超时检测使能。
30:28	保留	必须保持复位值。
27:16	BUSTOB[11:0]	总线超时 B 配置累积时钟延展超时。在主机模式下，检测主机累计时钟低电平延展时间$t_{LOW:MEXT}$。 从机模式下，检测从机累计时钟低电平延展时间$t_{LOW:SEXT}$。 $t_{LOW:EXT}$=(BUSTOB+1)*2048*t_{I2CCLK}。 **注意：**该位域仅在 EXTOEN = 0 时可以被修改。
15	TOEN	时钟超时使能 当 TOIDLE=0，SCL 拉低时间大于$t_{TIMEOUT}$或当 TOIDLE=1，SCL 拉低时间大于t_{IDLE}，将检测到超时错误。 0：SCL 超时检测禁能 1：SCL 超时检测使能
14:13	保留	必须保持复位值。
12	TOIDLE	空闲时钟超时检测 0：BUSTOA 用于检测 SCL 低电平超时。 1：BUSTOA 用于检测 SCL 和 SDA 高电平超时(总线空闲条件)。 **注意：**该位域仅在 TOEN=0 时可以被改写。
11:0	BUSTOA[11:0]	总线超时 A 当 TOIDLE=0 时，$t_{TIMEOUT}$=(BUSTOA+1)*2048*t_{I2CCLK} 当 TOIDLE=1 时，t_{IDLE}=(BUSTOA+1)*4*t_{I2CCLK} **注意：**该位域仅在 TOEN=0 时可以被改写。

状态寄存器(I2C_STAT)

地址偏移：0x18

复位值：0x0000 0001

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。
23:17	READDR[6:0]	从机模式下接收到的匹配地址 当 ADDSEND 置 1 时，这些位用于存储接收到的地址。在 10 位地址情况下，READDR[6:0]存储 10 位地址头和地址的最高两位。
16	TR	I2C 在从机模式下作为发送端还是接收端 该位在 ADDSEND 位置 1 时更新。 0：接收端 1：发送端
15	I2CBSY	忙标志 该位在硬件检测到 START 信号时置 1。在 STOP 信号后硬件清 0。当 I2CEN=0 时，由硬件清零。 0：无 I2C 通讯 1：I2C 正在通讯
14	保留	必须保持复位值。
13	SMBALT	SMBus 报警 当 SMBHAEN=1，SMBALTEN=1 且在 SMBA 引脚检测到 SMBALERT 事件(下降沿)时，该位由硬件置 1。SMBALTC 置 1 可以将该位软件清零。当 I2CEN=0 时，该位由硬件清零。 0：在 SMBA 引脚上未检测到 SMBALERT 事件。 1：在 SMBA 引脚上检测到 SMBALERT 事件。 **注意：**如果不支持 SMBus，该位保留并被硬件强制置 0。
12	TIMEOUT	超时标志 当发生超时或延展时钟超时，该位将置 1。TIMEOUTC 置 1 可以将该位软件清零。当 I2CEN=0 时，该位由硬件清零。 0：无超时或延展时钟超时发生。 1：发生超时或延展时钟超时。 **注意：**如果不支持 SMBus，该位保留并被硬件强制置 0。
11	PECERR	PEC 错误 当接收到的 PEC 字节与 I2C_PEC 寄存器中的内容不匹配时，该位置 1。然后将自

		动发生 NACK。PECERRC 置 1 可以将该位软件清零。当 I2CEN=0 时，该位由硬件清零。 0：接收到 PEC 与 I2C_PEC 的内容匹配。 1：接收到 PEC 与 I2C_PEC 的内容不匹配，此时 I2C 将忽略 NACKEN 位的值，并直接发送 NACK。 **注意：**如果不支持 SMBus，该位保留并被硬件强制置 0。
10	OUERR	从模式下上溢/下溢错误 在从机模式下且 SS=1，当发生上溢/下溢错误时，该位置 1。OUERRC 置 1 可以将该位软件清零。当 I2CEN=0 时，该位由硬件清零。 0：未发生上溢/下溢错误。 1：发生上溢/下溢错误。
9	LOSTARB	仲裁丢失 LOSTARBC 置 1 可以将该位软件清零。当 I2CEN=0 时，该位由硬件清零。 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 当 I2C 总线上发生了预料之外的 START 信号或 STOP 信号时，将产生总线错误，该位将置 1。BERRC 置 1 可以将该位软件清零。当 I2CEN=0 时，该位由硬件清零。 0：无总线错误。 1：发生了总线错误。
7	TCR	传输完成重载 当 RELOAD=1 且 BYTENUM[7:0]个字节传输完成时，该位置 1。在 BYTENUM[7:0]写入一个非零值可以软件清零该位。 0：当 RELOAD=1 时，BYTENUM[7:0]个字节传输未完成。 1：当 RELOAD=1 时，BYTENUM[7:0]个字节传输完成。
6	TC	主机模式下传输完成 当 RELOAD=0，AUTOEND=0 且 BYTENUM[7:0]个字节传输完成时，该位置 1。当 START 位或 STOP 位置 1 时该位清零。 0：BYTENUM[7:0]个字节传输未完成。 1：BYTENUM[7:0]个字节传输完成。
5	STPDET	总线上检测到 STOP 信号 当在总线上检测到 STOP 信号时，主机和从机的该位由硬件置 1。STPDETC 置 1 可以将该位软件清零。当 I2CEN=0 时，该位由硬件清零。 0：未监测到 STOP 结束位。 1：监测到 STOP 结束位。
4	NACK	接收到 NACK 应答 当接收到 NACK 时，该位置 1。NACKC 置 1 可以将该位软件清零。当 I2CEN=0 时，该位由硬件清零。 0：接收到 ACK。 1：接收到 NACK。

3	ADDSEND	从机模式下接收到的地址与自身地址匹配 当接收到的地址与使能的从机地址之一匹配时，该位由硬件置 1。ADDSENDC 置 1 可以将该位软件清零。当 I2CEN=0 时，该位由硬件清零。 0：接收到的地址不匹配。 1：接收到的地址匹配。
2	RBNE	接收期间 I2C_RDATA 非空 当接收到的数据移入 I2C_RDATA 寄存器时，该位置 1。读 I2C_RDATA 可清除该位。 0：I2C_RDATA 空。 1：I2C_RDATA 非空，软件可以读。
1	TI	发送中断 当 I2C_TDATA 为空且 I2C 已经做好发送数据准备时，该位置 1。在下一个待发送字节写入 I2C_TDATA 寄存器时该位清零。当 SS=1 时，可由软件将该位置 1 来产生 TI 事件(TIE=1 时为中断，DENT=1 时为 DMA 请求)。 0：I2C_TDATA 非空或者 I2C 还未做好发送数据准备。 1：I2C_TDATA 空且 I2C 已经做好发送数据准备。
0	TBE	发送期间 I2C_TDATA 空 当 I2C_TDATA 寄存器为空，该位置 1。当下一个待发送数据写入 I2C_TDATA 寄存器时，该位清零。可以软件将该位置 1 来清空 I2C_TDATA 寄存器。 0：I2C_TDATA 非空。 1：I2C_TDATA 空。

状态清除寄存器(I2C_STATC)

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
保留		SMBALTC	TIMEOUTC	PECERRC	OUERRC	LOSTARBC	BERRC	保留		STPDETC	NACKC	ADDSENDC	保留		
		w	w	w	w	w	w			w	w	w			

位/位域	名称	描述
31:14	保留	必须保持复位值。
13	SMBALTC	SMBus 报警标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 SMBALT 位。 **注意：**如果不支持 SMBus，该位保留并被硬件强制置 0。
12	TIMEOUTC	TIMEOUT 标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 TIMEOUT 位。

		注意：如果不支持 SMBus，该位保留并被硬件强制置 0。
11	PECERRC	PEC 错误标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 PECERR 位。 **注意：**如果不支持 SMBus，该位保留并被硬件强制置 0。
10	OUERRC	上溢/下溢标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 OUERR 位。
9	LOSTARBC	仲裁丢失标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 LOSTARB 位。
8	BERRC	总线错误标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 BERR 位。
7:6	保留	必须保持复位值。
5	STPDETC	停止位检测标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 STPDET 位。
4	NACKC	NACK 标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 NACK 位。
3	ADDSENDC	地址匹配标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 ADDSEND 位。
2:0	保留	必须保持复位值。

PEC 寄存器(I2C_PEC)

地址偏移：0x20
复位值：0x0000 0000

注意：如果不支持 SMBus，该寄存器保留并被硬件置为 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	PECV[7:0]	在 PEC 使能时，由硬件计算出来的 PEC 值。 当 I2CEN=0 时，PECV 由硬件清零。

接收数据寄存器(I2C_RDATA)

地址偏移：0x24
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

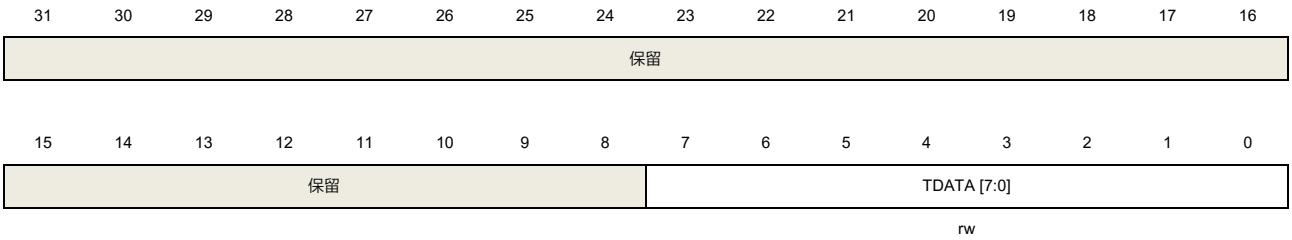


位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	RDATA[7:0]	接收到的数据

发送数据寄存器(I2C_TDATA)

地址偏移：0x28
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

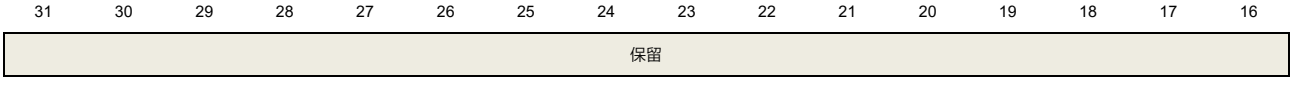


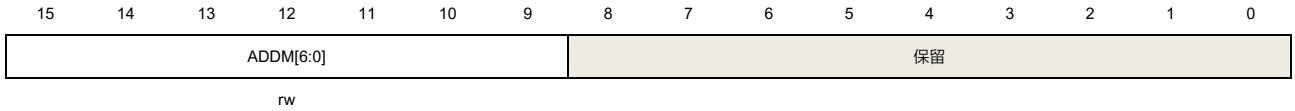
位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	TDATA[7:0]	发送的数据

控制寄存器 2(I2C_CTL2)

地址偏移：0x90
复位值：0x0000 FE00

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。
15:9	ADDM[6:0]	定义 ADDRESS[7:1]的哪些位和接收到的地址进行比较，哪些位不比较。ADDM[6:0]中设置为 1 的位使能 ADDRESS[7:1]中的相应位与接收到的地址进行比较，设置为 0 的位则忽略(此时接收到的地址在该位可以为 0 或 1)。
8:0	保留	必须保持复位值。

22. 附录

22.1. 寄存器表中使用的缩写列表

表 22-1. 寄存器功能位访问属性

功能位访问属性	描述
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/写(rw)	软件可以对这个位进行读写。
读/写一次(rwo)	软件只可写该位一次，可以读任意次。只有复位可将该位恢复为默认值。
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。
读/写清零(rc_w)	软件可以读该位，对该位写可以清除这个位。写 0 和写 1 效果相同。
写清零(c_w)	软件只可对该位写清除。写 0 和写 1 效果相同。
写 1 清零(c_w1)	软件只可对该位写入 1 清除这个位。写入 0 对位值没有影响。
可读/读清零(rc_r)	软件可以读该位，读该位会自动清零。写入 0 对位值没有影响。
可读/可置位(rs)	软件可以读该位，也可以将这个位设置为 1。写入 0 对位值没有影响。
可读/读置位(rs_r)	软件可以读该位，读该位会置位该位。写对位值没有影响。
可写/读置位(ws_r)	软件可以写该位，读该位会置位该位。
可读/写置位(rs_w)	软件可以读该位，写该位会置位该位。写 0 和写 1 效果相同。
写置位(s_w)	软件只可写置位该位。写 0 和写 1 效果相同。
可读/写 0 置位(rs_w0)	软件可以读该位，写 0 会置位该位。写入 1 对位值没有影响。
只读/写触发(rt_w)	可读，可写 0/1 触发，但对该位值无影响。
只读/写 1 触发(rt_w1)	软件只能读该位，写入 1 触发事件，但对位值没有影响。
只读/写 0 触发(rt_w0)	软件只能读该位，写入 0 触发事件，但对位值没有影响。
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。

22.2. 术语表

表 22-2. 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

术语	描述
	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

22.3. 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。

23. 版本历史

表 23-1. 版本历史

版本号	描述	日期
1.0	初稿发布	2026 年 6 月 1 日
1.1	1. 修改 ***8.3.12. 优先级交叉开关解码器(仅适用于 GD32E251xx)***中描述，将"PA0-PA15、PB0-PB10、PF0、PF1 和 PF3"改为"PA0-PA15、PB0-PB10、PF0 和 PF3"。 2. 删除 ***5.3.3. 省电模式***中 LDOVS 相关描述。 3. 删除 ***5.4.1. 控制寄存器(PMU_CTL)***中 LDOVS 位域。 4. 修改***状态寄存器(I2C_STAT)***中 SMBALT 位域描述。 5. 修改 I2C ***超时特性***章节描述。 6. 在 ***1.4. 引导配置***章节增加注意事项。	2026 年 8 月 10 日

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.

© 2026 GigaDevice Semiconductor Inc. - All rights reserved