

GigaDevice Semiconductor Inc.

GD32C2x1xx

Arm[®] Cortex[®]-M23 32-bit MCU

适用于 GD32C231xx 系列

用户手册

1.0 版本

(2025 年 6 月)

目录

目录.....	2
图索引	13
表索引	19
1. 系统及存储器架构.....	21
1.1. Arm® Cortex®-M23 处理器.....	21
1.2. 系统架构.....	22
1.3. 存储器映射	23
1.3.1. 片上 SRAM 存储器.....	26
1.3.2. 片上闪存.....	27
1.4. 引导配置.....	27
1.5. 系统配置控制器 (SYSCFG)	28
1.6. 系统配置寄存器	29
1.6.1. 配置寄存器 0 (SYSCFG_CFG0)	29
1.6.2. EXTI 源选择寄存器 0 (SYSCFG_EXTISS0)	30
1.6.3. EXTI 源选择寄存器 1 (SYSCFG_EXTISS1)	31
1.6.4. EXTI 源选择寄存器 2 (SYSCFG_EXTISS2)	32
1.6.5. EXTI 源选择寄存器 3 (SYSCFG_EXTISS3)	33
1.6.6. 配置寄存器 1 (SYSCFG_CFG1)	34
1.6.7. 系统状态寄存器 (SYSCFG_STAT)	34
1.6.8. 配置寄存器 2 (SYSCFG_CFG2)	35
1.6.9. IRQ 延迟寄存器 (SYSCFG_CPU_IRQ_LAT)	36
1.6.10. TIMERx 配置寄存器 0 (SYSCFG_TIMERxCFG0, x=0, 2)	36
1.6.11. TIMERx 配置寄存器 1 (SYSCFG_TIMERxCFG0, x=0, 2)	39
1.7. 设备电子签名.....	39
1.7.1. 存储容量信息	39
1.7.2. 设备唯一 ID (96 位/位域)	40
2. 闪存控制器 (FMC)	42
2.1. 简介.....	42
2.2. 主要特性.....	42
2.3. 功能说明.....	42
2.3.1. 闪存结构.....	42
2.3.2. 主闪存空校验	43
2.3.3. 读操作	43
2.3.4. FMC_CTL/FMC_OBCTL 寄存器解锁	44
2.3.5. 页擦除	44

2.3.6.	整片擦除.....	45
2.3.7.	主闪存编程.....	46
2.3.8.	主闪存快速编程.....	48
2.3.9.	OTP 编程	49
2.3.10.	选项字节.....	50
2.3.11.	仅执行区域 (DCRP)	53
2.3.12.	页擦除/编程保护 (WP)	53
2.3.13.	安全保护 (SPC)	54
2.3.14.	安全用户区域 (SCR)	54
2.3.15.	禁用内核调试访问	55
2.3.16.	强制从闪存启动.....	55
2.3.17.	FMC 中断.....	55
2.4.	FMC 寄存器.....	57
2.4.1.	等待状态寄存器 (FMC_WS)	57
2.4.2.	解锁寄存器 (FMC_KEY)	58
2.4.3.	选项字节操作解锁寄存器 (FMC_OBKEY)	58
2.4.4.	状态寄存器 (FMC_STAT)	59
2.4.5.	控制寄存器 (FMC_CTL)	60
2.4.6.	选项字节控制寄存器 (FMC_OBCTL)	62
2.4.7.	DCRP0 起始地址寄存器 (FMC_DCRP_SADDR0)	64
2.4.8.	DCRP0 结束地址寄存器 (FMC_DCRP_EADDR0)	64
2.4.9.	擦除/编程保护区域 0 寄存器 (FMC_WP0)	65
2.4.10.	擦除/编程保护区域 1 寄存器 (FMC_WP1)	65
2.4.11.	DCRP1 起始地址寄存器 (FMC_DCRP_SADDR1)	66
2.4.12.	DCRP1 结束地址寄存器 (FMC_DCRP_EADDR1)	66
2.4.13.	安全用户区域寄存器 (FMC_SCR)	67
2.4.14.	产品 ID 寄存器 (FMC_PID)	67
3.	电源管理单元 (PMU)	69
3.1.	简介.....	69
3.2.	主要特征.....	69
3.3.	功能说明.....	69
3.3.1.	VDD / VDDA 电源域.....	70
3.3.2.	1.2V 电源域.....	71
3.3.3.	省电模式.....	72
3.4.	PMU 寄存器.....	77
3.4.1.	控制寄存器 0 (PMU_CTL0)	77
3.4.2.	电源控制和状态寄存器 (PMU_CS)	78
3.4.3.	控制寄存器 1 (PMU_CTL1)	80
3.4.4.	状态寄存器 (PMU_STAT)	80
3.4.5.	参数寄存器 (PMU_PAR)	81
4.	复位和时钟单元 (RCU)	82

4.1. 复位控制单元 (RCTL)	82
4.1.1. 简介	82
4.1.2. 功能描述	82
4.2. 时钟控制单元 (CCTL)	83
4.2.1. 简介	83
4.2.2. 主要特性	85
4.2.3. 功能描述	85
4.3. RCU 寄存器	89
4.3.1. 控制寄存器 (RCU_CTL0)	89
4.3.2. 配置寄存器 0 (RCU_CFG0)	91
4.3.3. 中断寄存器 (RCU_INT)	93
4.3.4. AHB1 复位寄存器 (RCU_AHB1RST)	95
4.3.5. AHB2 复位寄存器 (RCU_AHB2RST)	96
4.3.6. APB 复位寄存器 (RCU_APBRSR)	97
4.3.7. AHB1 使能寄存器 (RCU_AHB1EN)	99
4.3.8. AHB2 使能寄存器 (RCU_AHB2EN)	100
4.3.9. APB 使能寄存器 (RCU_APBEN)	101
4.3.10. AHB1 睡眠和深度睡眠使能寄存器 (RCU_AHB1SPDPEN)	103
4.3.11. AHB2 睡眠和深度睡眠使能寄存器 (RCU_AHB2SPDPEN)	104
4.3.12. APB 睡眠和深度睡眠使能寄存器 (RCU_APBSPDPEN)	105
4.3.13. 控制寄存器 1 (RCU_CTL1)	108
4.3.14. 复位源/时钟寄存器 (RCU_RSTSCK)	110
4.3.15. 配置寄存器 1 (RCU_CFG1)	111
5. 中断 / 事件控制器 (EXTI)	114
5.1. 简介	114
5.2. 主要特征	114
5.3. 功能说明	114
5.4. 外部中断及事件 (EXTI) 结构框图	116
5.5. 外部中断及事件功能概述	116
5.6. EXTI 寄存器	119
5.6.1. 中断使能寄存器 (EXTI_INTEN)	119
5.6.2. 事件使能寄存器 (EXTI_EVEN)	119
5.6.3. 上升沿触发使能寄存器 (EXTI_RTEN)	120
5.6.4. 下降沿触发使能寄存器 (EXTI_FTEN)	120
5.6.5. 软件中断事件寄存器 (EXTI_SWIEV)	120
5.6.6. 挂起寄存器 (EXTI_PD)	121
6. 通用和备用输入/输出接口 (GPIO 和 AFIO)	122
6.1. 简介	122
6.2. 主要特征	122

6.3. 功能说明	122
6.3.1. GPIO 引脚配置.....	123
6.3.2. 外部中断及事件.....	124
6.3.3. 备用功能 (AF)	124
6.3.4. 附加功能.....	124
6.3.5. 输入配置.....	124
6.3.6. 输出配置.....	125
6.3.7. 模拟配置.....	125
6.3.8. 备用功能 (AF) 配置	125
6.3.9. GPIO 锁定功能.....	126
6.3.10. GPIO 单周期输出翻转功能.....	126
6.3.11. HXTAL 和 LXTAL GPIO 功能配置	126
6.3.12. 复位引脚 GPIO 功能配置	127
6.3.13. 引脚 (PA14) GPIO 功能配置.....	127
6.4. GPIO 寄存器	128
6.4.1. 端口控制寄存器 (GPIOx_CTL, x=A..D, F)	128
6.4.2. 端口输出模式寄存器 (GPIOx_OMODE, x=A..D, F)	129
6.4.3. 端口输出速度寄存器 (GPIOx_OSPD, x=A..D, F)	131
6.4.4. 端口上拉/下拉寄存器 (GPIOx_PUD, x=A..D, F)	132
6.4.5. 端口输入状态寄存器 (GPIOx_ISTAT, x=A..D, F)	133
6.4.6. 端口输出控制寄存器 (GPIOx_OCTL, x=A..D, F)	134
6.4.7. 端口位操作寄存器 (GPIOx_BOP, x=A..D, F)	134
6.4.8. 端口配置锁定寄存器 (GPIOx_LOCK, x=A..D, F)	135
6.4.9. 备用功能选择寄存器 0 (GPIOx_AFSEL0, x=A..D, F)	135
6.4.10. 备用功能选择寄存器 1 (GPIOx_AFSEL1, x=A..D, F)	137
6.4.11. 位清除寄存器 (GPIOx_BC, x=A..D, F)	138
6.4.12. 端口位翻转寄存器 (GPIOx_TG, x=A..D, F)	138
7. 循环冗余校验计算单元 (CRC)	140
7.1. 简介.....	140
7.2. 主要特征.....	140
7.3. 功能说明.....	141
7.4. CRC 寄存器.....	142
7.4.1. 数据寄存器 (CRC_DATA)	142
7.4.2. 独立数据寄存器 (CRC_FDATA)	142
7.4.3. 控制寄存器 (CRC_CTL)	143
7.4.4. 初值寄存器 (CRC_IDATA)	143
7.4.5. 多项式寄存器 (CRC_POLY)	144
8. 调试 (DBG)	145
8.1. 简介.....	145
8.2. 串行调试接口简介.....	145
8.2.1. 引脚分配.....	145

8.3.	调试保持功能描述	145
8.3.1.	低功耗模式调试支持	145
8.3.2.	TIMER, I2C, RTC, WWDGT 和 FWDGT 外设调试支持	146
8.4.	DBG 寄存器	147
8.4.1.	ID 寄存器 (DBG_ID)	147
8.4.2.	控制寄存器 0 (DBG_CTL0)	147
8.4.3.	控制寄存器 1 (DBG_CTL1)	149
9.	直接存储器访问控制器 (DMA)	150
9.1.	简介	150
9.2.	主要特征	150
9.3.	结构框图	151
9.4.	功能说明	151
9.4.1.	DMA 操作	151
9.4.2.	外设握手	152
9.4.3.	仲裁	153
9.4.4.	地址生成	153
9.4.5.	循环模式	153
9.4.6.	存储器到存储器模式	154
9.4.7.	通道配置	154
9.4.8.	中断	154
9.4.9.	DMA 请求映射	155
9.5.	DMA 寄存器	156
9.5.1.	中断标志位寄存器 (DMA_INTF)	156
9.5.2.	中断标志位清除寄存器 (DMA_INTC)	156
9.5.3.	通道 x 控制寄存器 (DMA_CHxCTL)	157
9.5.4.	通道 x 计数寄存器 (DMA_CHxCNT)	159
9.5.5.	通道 x 外设基地址寄存器 (DMA_CHxPADDR)	160
9.5.6.	通道 x 存储器基地址寄存器 (DMA_CHxMADDR)	160
10.	DMA 请求多路复用器 (DMAMUX)	161
10.1.	简介	161
10.2.	主要特征	161
10.3.	结构框图	162
10.4.	信号描述	162
10.5.	功能说明	163
10.5.1.	DMAMUX 请求路由器	163
10.5.2.	DMAMUX 请求生成器	165
10.5.3.	通道配置	166
10.5.4.	中断	166
10.5.5.	DMAMUX 映射	167

10.6. DMAMUX 寄存器	171
10.6.1. 请求路由通道 x 配置寄存器 (DMAMUX_RM_CHxCFG)	171
10.6.2. 请求路由通道中断标志位寄存器 (DMAMUX_RM_INTF)	172
10.6.3. 请求路由通道中断标志位清除寄存器 (DMAMUX_RM_INTC)	172
10.6.4. 请求生成通道 x 配置寄存器 (DMAMUX_RG_CHxCFG)	173
10.6.5. 请求生成通道中断标志位寄存器 (DMAMUX_RG_INTF)	174
10.6.6. 请求生成通道中断标志位清除寄存器 (DMAMUX_RG_INTC)	174
11. 模数转换器 (ADC)	176
11.1. 简介	176
11.2. 主要特征	176
11.3. 引脚和内部信号	177
11.4. 功能描述	178
11.4.1. 多时钟域架构	178
11.4.2. ADC 使能	178
11.4.3. 常规序列	178
11.4.4. 运行模式	179
11.4.5. 转换结果阈值监测功能	181
11.4.6. 数据存储模式	182
11.4.7. 采样时间配置	183
11.4.8. 外部触发	183
11.4.9. DMA 请求	183
11.4.10. ADC 内部通道	184
11.4.11. 可编程分辨率 (DRES)	184
11.4.12. 片上硬件过采样	185
11.5. 中断	186
11.6. ADC 寄存器	187
11.6.1. 状态寄存器 (ADC_STAT)	187
11.6.2. 控制寄存器 0 (ADC_CTL0)	188
11.6.3. 控制寄存器 1 (ADC_CTL1)	189
11.6.4. 采样时间寄存器 0 (ADC_SAMPT0)	191
11.6.5. 采样时间寄存器 1 (ADC_SAMPT1)	192
11.6.6. 看门狗 0 高阈值寄存器 (ADC_WD0HT)	192
11.6.7. 看门狗 0 低阈值寄存器 (ADC_WD0LT)	193
11.6.8. 常规序列寄存器 0 (ADC_RSQ0)	193
11.6.9. 常规序列寄存器 1 (ADC_RSQ1)	194
11.6.10. 常规序列寄存器 2 (ADC_RSQ2)	195
11.6.11. 常规数据寄存器 (ADC_RDATA)	195
11.6.12. 看门狗 1 通道选择寄存器 (ADC_WD1SR)	196
11.6.13. 看门狗 2 通道选择寄存器 (ADC_WD2SR)	196
11.6.14. 看门狗 1 高阈值寄存器 (ADC_WD1HT)	197
11.6.15. 看门狗 1 低阈值寄存器 (ADC_WD1LT)	197
11.6.16. 看门狗 2 高阈值寄存器 (ADC_WD2HT)	198

11.6.17.	看门狗 2 低阈值寄存器 (ADC_WD2LT)	198
11.6.18.	过采样控制寄存器 (ADC_OVSAMPCTL)	199
12.	看门狗定时器 (WDGT)	201
12.1.	独立看门狗定时器 (FWDGT)	201
12.1.1.	简介	201
12.1.2.	主要特征	201
12.1.3.	功能说明	201
12.1.4.	FWDGT 寄存器	204
12.2.	窗口看门狗定时器 (WWDGT)	207
12.2.1.	简介	207
12.2.2.	主要特征	207
12.2.3.	功能说明	207
12.2.4.	WWDGT 寄存器	210
13.	实时时钟 (RTC)	212
13.1.	简介	212
13.2.	主要特征	212
13.3.	功能描述	212
13.3.1.	时钟源和预分频	213
13.3.2.	影子寄存器	213
13.3.3.	位域可屏蔽可配置的闹钟	213
13.3.4.	RTC 初始化和配置	214
13.3.5.	读取日历	215
13.3.6.	RTC 复位	216
13.3.7.	RTC 移位功能	216
13.3.8.	RTC 参考时钟检测	217
13.3.9.	RTC 数字平滑校准	217
13.3.10.	时间戳功能	219
13.3.11.	校准时钟输出	219
13.3.12.	闹钟输出	219
13.3.13.	RTC 引脚配置	219
13.3.14.	RTC 中断	221
13.4.	RTC 寄存器	222
13.4.1.	时间寄存器 (RTC_TIME)	222
13.4.2.	日期寄存器 (RTC_DATE)	222
13.4.3.	控制寄存器 (RTC_CTL)	223
13.4.4.	状态寄存器 (RTC_STAT)	225
13.4.5.	预分频寄存器 (RTC_PSC)	227
13.4.6.	闹钟 0 时间日期寄存器 (RTC_ALRM0TD)	227
13.4.7.	写保护钥匙寄存器 (RTC_WPK)	228
13.4.8.	亚秒寄存器 (RTC_SS)	229
13.4.9.	移位控制寄存器 (RTC_SHIFTCTL)	229

13.4.10.	时间戳时间寄存器 (RTC_TTS)	230
13.4.11.	时间戳日期寄存器 (RTC_DTS)	230
13.4.12.	时间戳亚秒寄存器 (RTC_SSTS)	231
13.4.13.	高精度频率补偿寄存器 (RTC_HRFC)	232
13.4.14.	类型寄存器 (RTC_TYPE)	232
13.4.15.	闹钟 0 亚秒寄存器 (RTC_ALRM0SS)	233
13.4.16.	备份寄存器 (RTC_BKPx) (x=0..3)	234
14.	定时器 (TIMER)	235
14.1.	高级定时器 (TIMERx, x=0)	236
14.1.1.	简介	236
14.1.2.	主要特征	236
14.1.3.	结构框图	237
14.1.4.	功能说明	237
14.1.5.	TIMERx 寄存器 (x=0)	266
14.2.	通用定时器 L0 (TIMERx, x=2)	298
14.2.1.	简介	298
14.2.2.	主要特征	298
14.2.3.	结构框图	298
14.2.4.	功能说明	299
14.2.5.	TIMERx 寄存器 (x=2)	317
14.3.	通用定时器 L2 (TIMERx, x= 13)	338
14.3.1.	简介	338
14.3.2.	主要特征	338
14.3.3.	结构框图	338
14.3.4.	功能描述	338
14.3.5.	TIMERx 寄存器(x = 13)	346
14.4.	通用定时器 L4 (TIMERx,x=15,16)	356
14.4.1.	简介	356
14.4.2.	主要特性	356
14.4.3.	结构框图	356
14.4.4.	功能描述	357
14.4.5.	TIMERx 寄存器(x=15,16)	368
15.	通用同步异步收发器 (USART)	385
15.1.	简介	385
15.2.	主要特征	385
15.3.	功能描述	386
15.3.1.	USART 帧格式	387
15.3.2.	波特率发生	388
15.3.3.	USART 发送器	388
15.3.4.	USART 接收器	389
15.3.5.	DMA 方式访问数据缓冲区	391

15.3.6.	硬件流控制	392
15.3.7.	多处理器通信	393
15.3.8.	LIN 模式	394
15.3.9.	同步通信模式	394
15.3.10.	串行红外 (IrDA SIR) 编解码功能模块	395
15.3.11.	半双工通信模式	396
15.3.12.	智能卡 (ISO7816-3) 模式	397
15.3.13.	ModBus 通信	398
15.3.14.	接收 FIFO	399
15.3.15.	从 DeepSleep 模式唤醒	399
15.3.16.	USART 中断	399
15.4.	USART 寄存器	402
15.4.1.	USART 控制寄存器 0 (USART_CTL0)	402
15.4.2.	USART 控制寄存器 1 (USART_CTL1)	404
15.4.3.	USART 控制寄存器 2 (USART_CTL2)	406
15.4.4.	USART 波特率寄存器 (USART_BAUD)	409
15.4.5.	USART 保护时间和预分频器寄存器 (USART_GP)	410
15.4.6.	USART 接收超时寄存器 (USART_RT)	410
15.4.7.	USART 请求寄存器 (USART_CMD)	411
15.4.8.	USART 状态寄存器 (USART_STAT)	412
15.4.9.	USART 中断标志清除寄存器 (USART_INTC)	415
15.4.10.	USART 数据接收寄存器 (USART_RDATA)	416
15.4.11.	USART 数据发送寄存器 (USART_TDATA)	417
15.4.12.	USART 兼容性控制寄存器 (USART_CHC)	417
15.4.13.	USART 接收 FIFO 控制和状态寄存器 (USART_RFCS)	418
16.	内部集成电路总接口 (I2C)	420
16.1.	简介	420
16.2.	主要特征	420
16.3.	功能说明	420
16.3.1.	时钟要求	421
16.3.2.	I2C 通讯流程	422
16.3.3.	噪声滤波器	424
16.3.4.	I2C 时序配置	424
16.3.5.	I2C 复位	426
16.3.6.	数据传输	426
16.3.7.	I2C 从机模式	428
16.3.8.	I2C 主机模式	433
16.3.9.	SMBus 支持	438
16.3.10.	SMBus 模式	440
16.3.11.	从省电模式唤醒	442
16.3.12.	DMA 模式下数据传输	442
16.3.13.	I2C 错误和中断	442

16.3.14.	I2C 调试模式	443
16.4.	I2C 寄存器	444
16.4.1.	控制寄存器 0 (I2C_CTL0)	444
16.4.2.	控制寄存器 1 (I2C_CTL1)	446
16.4.3.	从机地址寄存器 0 (I2C_SADDR0)	448
16.4.4.	从机地址寄存器 1 (I2C_SADDR1)	449
16.4.5.	时序寄存器 (I2C_TIMING)	449
16.4.6.	超时寄存器 (I2C_TIMEOUT)	450
16.4.7.	状态寄存器 (I2C_STAT)	451
16.4.8.	状态清除寄存器 (I2C_STATC)	454
16.4.9.	PEC 寄存器 (I2C_PEC)	455
16.4.10.	接收数据寄存器 (I2C_RDATA)	455
16.4.11.	发送数据寄存器 (I2C_TDATA)	456
16.4.12.	控制寄存器 2 (I2C_CTL2)	456
17.	串行外设接口/片上音频接口 (SPI/I2S)	458
17.1.	简介	458
17.2.	主要特征	458
17.2.1.	SPI 主要特征	458
17.2.2.	I2S 主要特性	458
17.3.	SPI 功能说明	459
17.4.	SPI 结构框图	459
17.4.1.	SPI 信号线描述	459
17.4.2.	SPI 时序和数据帧格式	460
17.4.3.	独立发送和接收缓冲区	462
17.4.4.	NSS 功能	463
17.4.5.	SPI 运行模式	464
17.4.6.	DMA 功能	472
17.4.7.	CRC 功能	473
17.4.8.	SPI 中断	473
17.5.	I2S 功能说明	475
17.5.1.	I2S 结构框图	475
17.5.2.	I2S 信号线描述	475
17.5.3.	I2S 音频标准	475
17.5.4.	I2S 时钟	483
17.5.5.	运行	483
17.5.6.	DMA 功能	488
17.5.7.	I2S 中断	488
17.6.	SPI/I2S 寄存器	490
17.6.1.	控制寄存器 0 (SPI_CTL0)	490
17.6.2.	控制寄存器 1 (SPI_CTL1)	492
17.6.3.	状态寄存器 (SPI_STAT)	493

17.6.4.	数据寄存器 (SPI_DATA)	495
17.6.5.	CRC 多项式寄存器 (SPI_CRCPOLY)	496
17.6.6.	接收 CRC 寄存器 (SPI_RCRC)	496
17.6.7.	发送 CRC 寄存器 (SPI_TCRC)	497
17.6.8.	I2S 控制寄存器 (SPI_I2SCTL)	498
17.6.9.	I2S 时钟预分频寄存器 (SPI_I2SPSC)	499
17.6.10.	SPI1 四线 SPI 控制寄存器 (SPI_QCTL)	500
18.	比较器 (CMP)	501
18.1.	简介	501
18.2.	主要特征	501
18.3.	功能描述	501
18.3.1.	比较器时钟	502
18.3.2.	比较器的 I/O 配置	502
18.3.3.	比较器供电模式	503
18.3.4.	比较器迟滞	503
18.3.5.	比较器寄存器写保护	504
18.3.6.	比较器输出消隐	504
18.3.7.	电压定标器功能	505
18.3.8.	比较器中断	505
18.4.	CMP 寄存器	506
18.4.1.	CMP0 控制状态寄存器 (CMP0_CS)	506
18.4.2.	CMP1 控制状态寄存器 (CMP1_CS)	508
19.	附录	510
20.	版本历史	511

图索引

图 1-1. Arm® Cortex®-M23 处理器结构框图	22
图 1-2. GD32C2x1 系列器件的系统架构示意图	23
图 1-3. ECC 解码器示意图	26
图 2-1. 页擦除操作流程	45
图 2-2. 整片擦除操作流程	46
图 2-3. 双字编程操作流程	47
图 2-4. 快速编程操作流程	49
图 3-1. 电源域概览	70
图 3-2. 上电 / 掉电复位波形图	71
图 3-3. BOR 波形图	71
图 3-4. 省电模式转换图	75
图 4-1. 系统复位电路	83
图 4-2. 时钟树	84
图 4-3. HXTAL 时钟源	85
图 4-4. 旁路模式下 HXTAL 时钟源	86
图 5-1. EXTI 结构框图	116
图 6-1. GPIO 端口位的基本结构	123
图 6-2. 输入配置的基本结构	124
图 6-3. 输出配置的基本结构	125
图 6-4. 模拟配置的基本结构	125
图 6-5. 备用功能配置的基本结构	126
图 7-1. CRC 计算单元框图	140
图 9-1. DMA 结构框图	151
图 9-2. 握手机制	153
图 9-3. DMA 中断逻辑图	155
图 10-1. DMAMUX 结构框图	162
图 10-2. 同步模式	164
图 10-3. 通道事件输出	165
图 11-1. ADC 模块框图	178
图 11-2. 单次运行模式	179
图 11-3. 连续运行模式	179
图 11-4. 扫描运行模式, 且连续运行模式禁能	180
图 11-5. 扫描运行模式, 连续运行模式使能	181
图 11-6. 间断运行模式	181
图 11-7. 12 位数据存储模式	182
图 11-8. 10 位数据存储模式	182
图 11-9. 8 位数据存储模式	182
图 11-10. 6 位数据存储模式	183
图 11-11. 20 位到 16 位的结果截断	185
图 11-12. 右移 5 位和取整的数例	186
图 12-1. 独立看门狗定时器框图	202

图 12-2. 窗口看门狗定时器框图	207
图 12-3. 窗口看门狗定时器时序图	208
图 13-1. RTC 结构框图	212
图 14-1. 高级定时器结构框图.....	237
图 14-2. 内部时钟分频为 1 时，正常模式下的控制电路	238
图 14-3. 当预分频器的参数从 1 变到 2 时，计数器的时序图.....	239
图 14-4. 向上计数时序图，PSC=0/1.....	240
图 14-5. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值	240
图 14-6. 向下计数时序图，PSC=0/1.....	241
图 14-7. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值.....	242
图 14-8. 中央计数模式计数器时序图.....	243
图 14-9. 中央计数模式下计数器重复时序图.....	244
图 14-10. 在向上计数模式下计数器重复时序图	244
图 14-11. 在向下计数模式下计数器重复时序图	245
图 14-12. 输入捕获逻辑	245
图 14-13. 输出比较逻辑（带有互补输出的通道，x=0, 1, 2）	246
图 14-14. 输出比较逻辑（通道，x = 3）	247
图 14-15. 输出比较逻辑（通道，x = 4）	247
图 14-16. 三种输出比较模式	248
图 14-17. EAPWM 时序图	249
图 14-18. CAPWM 时序图	250
图 14-19. O0CPREC 和 O2CPREC 使用非对称 PWM 模式.....	251
图 14-20. O0CPRE 选择复合 PWM 模式 1 和 O1CPRE 选择 PWM 模式 0	252
图 14-21. CCH4CH0 置位时 O0CPREC 工作在复合三相 PWM 模式	252
图 14-22. 可再次触发单脉冲模式.....	253
图 14-23. 带死区时间的互补输出	255
图 14-24. BREAK0 的功能逻辑图	256
图 14-25. BREAK1 的功能逻辑图	256
图 14-26. 通道响应中止输入（高电平有效）时，输出信号的行为	257
图 14-27. 正交译码器接口模式下计数器运行例子	258
图 14-28. CIOFE0 极性反相的正交译码器接口模式下的例子	258
图 14-29. 霍尔传感器用在 BLDC 电机控制中.....	259
图 14-30. 两个定时器之间的霍尔传感器时序图	260
图 14-31. 复位模式.....	261
图 14-32. 暂停模式.....	261
图 14-33. 事件模式.....	261
图 14-34. 单脉冲模式，TIMERx_CHxCV = 0x04，TIMERx_CAR=0x60.....	262
图 14-35. 用定时器 2 的使能信号触发定时器 0.....	263
图 14-36. 用定时器 2 的更新事件来触发定时器 0.....	264
图 14-37. 用定时器 2 的使能信号来控制定时器 0 的暂停模式.....	264
图 14-38. 用定时器 2 的 O0CPRE 信号控制定时器 0 的暂停模式.....	265
图 14-39.通用定时器 L0 结构框图	299
图 14-40. 内部时钟分频为 1 时，计数器的时序图	300
图 14-41. 当 PSC 数值从 0 变到 2 时，计数器的时序图.....	301

图 14-42. 向上计数时序图, PSC=0/2.....	302
图 14-43. 向上计数时序图, 在运行时改变 TIMERx_CAR 寄存器的值	302
图 14-44. 向下计数时序图, PSC=0/2.....	303
图 14-45. 向下计数时序图, 在运行时改变 TIMERx_CAR 寄存器值.....	304
图 14-46. 中央计数模式计数器时序图.....	305
图 14-47. 通道输入捕获原理	306
图 14-48. 通道输出比较原理 (x=0, 1, 2, 3)	307
图 14-49. 三种输出比较模式	308
图 14-50. EAPWM 时序图	309
图 14-51. CAPWM 时序图	309
图 14-52. O0CPREC 和 O2CPREC 使用非对称 PWM 模式.....	310
图 14-53. O0CPRE 选择复合 PWM 模式 1 和 O1CPRE 选择 PWM 模式 0	311
图 14-54. 可再次触发单脉冲模式.....	311
图 14-55. 正交译码器接口模式下计数器运行例子	313
图 14-56. CI0FE0 极性反相的正交译码器接口模式下的例子	313
图 14-57. 复位模式下的控制电路.....	314
图 14-58. 暂停模式下的控制电路.....	314
图 14-59. 事件模式下的控制电路.....	315
图 14-60. 单脉冲模式, TIMERx_CHxCV = 4 TIMERx_CAR=99	315
图 14-61. 通用定时器 L2 结构框图	338
图 14-62. 内部时钟分频为 1 时, 计数器的时序图	339
图 14-63. 当 PSC 数值从 0 变到 2 时, 计数器的时序图.....	340
图 14-64. 向上计数时序图, PSC=0/2.....	341
图 14-65. 向上计数时序图, 在运行时改变 TIMERx_CAR 寄存器的值	341
图 14-66. 通道输入捕获原理	342
图 14-67. 三种输出比较模式	344
图 14-68. 单脉冲模式, TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60	345
图 14-69. 通用定时器 L4 结构框图	357
图 14-70. 内部时钟分频为 1 时正常模式下的控制电路.....	358
图 14-71. 当预分频器的参数从 1 变到 2 时, 计数器的时序图.....	358
图 14-72. 向上计数时序图, PSC=0/1.....	359
图 14-73. 向上计数时序图, 在运行时改变 TIMERx_CAR 寄存器的值	360
图 14-74. 在向上计数模式下计数器重复时序图	361
图 14-75. 输入捕获逻辑	362
图 14-76. 三种输出比较模式	363
图 14-77. PWM 时序图	364
图 14-78. 带死区时间的互补输出	366
图 14-79. 通道响应中止输入 (高电平有效) 时, 输出信号的行为	367
图 14-80. 单脉冲模式, TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60	368
图 15-1. USART 模块内部框图.....	387
图 15-2. USART 字符帧 (8 数据位和 1 停止位)	387
图 15-3. USART 发送步骤	389
图 15-4. 过采样方式接收一个数据位 (OSB=0)	390
图 15-5. 采用 DMA 方式实现 USART 数据发送配置步骤	391

图 15-6. 采用 DMA 方式实现 USART 数据接收配置步骤	392
图 15-7. 两个 USART 之间的硬件流控制	392
图 15-8. 硬件流控制	393
图 15-9. 空闲状态下检测断开帧	394
图 15-10. 数据传输过程中检测断开帧	394
图 15-11. 同步模式下的 USART 示例	395
图 15-12. 8-bit 格式的 USART 同步通信波形 (CLEN=1)	395
图 15-13. IrDA SIR ENDEC 模块	396
图 15-14. IrDA 数据调制	396
图 15-15. ISO7816-3 数据帧格式	397
图 15-16. USART 接收 FIFO 结构	399
图 15-17. USART 中断映射框图	401
图 16-1. I2C 模块框图	421
图 16-2. 数据有效性	422
图 16-3. 开始和停止信号	422
图 16-4. 10 位地址的 I2C 通讯流程 (主机发送)	423
图 16-5. 7 位地址的 I2C 通讯流程 (主机发送)	423
图 16-6. 7 位地址的 I2C 通讯流程 (主机接收)	423
图 16-7. 10 位地址的 I2C 通讯流程 (主机接收, HEAD10R=0)	424
图 16-8. 10 位地址的 I2C 通讯流程 (主机接收, HEAD10R=1)	424
图 16-9. 数据保持时间	425
图 16-10. 数据建立时间	425
图 16-11. 数据发送	427
图 16-12. 数据接收	427
图 16-13. I2C 从机初始化	430
图 16-14. I2C 从机发送编程模型 (SS=0)	431
图 16-15. I2C 从机发送编程模型 (SS=1)	432
图 16-16. I2C 从机接收编程模型	433
图 16-17. I2C 主机初始化	434
图 16-18. I2C 主机发送编程模型 (N<=255)	435
图 16-19. I2C 主机发送编程模型 (N>255)	436
图 16-20. I2C 主机接收编程模型 (N<=255)	437
图 16-21. I2C 主机接收编程模型 (N>255)	438
图 16-22. SMBus 主机发送器和从机接收器通信流程	441
图 16-23. SMBus 主机接收器和从机发送器通信流程	442
图 17-1. SPI 结构框图	459
图 17-2. SPI1 常规模式下的时序图	461
图 17-3. SPI1 数据帧右对齐示意图	461
图 17-4. SPI0 常规模式下的时序图	461
图 17-5. SPI1 四线模式下的 SPI 时序图 (CKPL=1, CKPH=1, LF=0)	462
图 17-6. 发送/接收缓冲区	462
图 17-7. 典型的全双工模式连接	465
图 17-8. 典型的单工模式连接 (主机: 接收, 从机: 发送)	465
图 17-9. 典型的单工模式连接 (主机: 只发送, 从机: 接收)	466

图 17-10. 典型的双向线连接	466
图 17-11. 主机 TI 模式在不连续发送时的时序图	468
图 17-12. 主机 TI 模式在连续发送时的时序图	468
图 17-13. 从机 TI 模式时序图	469
图 17-14. NSS 脉冲模式时序图 (主机连续发送)	469
图 17-15. SPI 四线模式四线写操作时序图	470
图 17-16. SPI 四线模式四线读操作时序图	471
图 17-17. I2S 结构框图	475
图 17-18. I2S 飞利浦标准时序图 (DTLEN=00, CHLEN=0, CKPL=0)	476
图 17-19. I2S 飞利浦标准时序图 (DTLEN=00, CHLEN=0, CKPL=1)	476
图 17-20. I2S 飞利浦标准时序图 (DTLEN=10, CHLEN=1, CKPL=0)	476
图 17-21. I2S 飞利浦标准时序图 (DTLEN=10, CHLEN=1, CKPL=1)	476
图 17-22. I2S 飞利浦标准时序图 (DTLEN=01, CHLEN=1, CKPL=0)	477
图 17-23. I2S 飞利浦标准时序图 (DTLEN=01, CHLEN=1, CKPL=1)	477
图 17-24. I2S 飞利浦标准时序图 (DTLEN=00, CHLEN=1, CKPL=0)	477
图 17-25. I2S 飞利浦标准时序图 (DTLEN=00, CHLEN=1, CKPL=1)	477
图 17-26. MSB 对齐标准时序图 (DTLEN=00, CHLEN=0, CKPL=0)	478
图 17-27. MSB 对齐标准时序图 (DTLEN=00, CHLEN=0, CKPL=1)	478
图 17-28. MSB 对齐标准时序图 (DTLEN=10, CHLEN=1, CKPL=0)	478
图 17-29. MSB 对齐标准时序图 (DTLEN=10, CHLEN=1, CKPL=1)	478
图 17-30. MSB 对齐标准时序图 (DTLEN=01, CHLEN=1, CKPL=0)	478
图 17-31. MSB 对齐标准时序图 (DTLEN=01, CHLEN=1, CKPL=1)	478
图 17-32. MSB 对齐标准时序图 (DTLEN=00, CHLEN=1, CKPL=0)	478
图 17-33. MSB 对齐标准时序图 (DTLEN=00, CHLEN=1, CKPL=1)	479
图 17-34. LSB 对齐标准时序图 (DTLEN=01, CHLEN=1, CKPL=0)	479
图 17-35. LSB 对齐标准时序图 (DTLEN=01, CHLEN=1, CKPL=1)	479
图 17-36. LSB 对齐标准时序图 (DTLEN=00, CHLEN=1, CKPL=0)	479
图 17-37. LSB 对齐标准时序图 (DTLEN=00, CHLEN=1, CKPL=1)	480
图 17-38. PCM 标准短帧同步模式时序图 (DTLEN=00, CHLEN=0, CKPL=0)	480
图 17-39. PCM 标准短帧同步模式时序图 (DTLEN=00, CHLEN=0, CKPL=1)	480
图 17-40. PCM 标准短帧同步模式时序图 (DTLEN=10, CHLEN=1, CKPL=0)	480
图 17-41. PCM 标准短帧同步模式时序图 (DTLEN=10, CHLEN=1, CKPL=1)	480
图 17-42. PCM 标准短帧同步模式时序图 (DTLEN=01, CHLEN=1, CKPL=0)	481
图 17-43. PCM 标准短帧同步模式时序图 (DTLEN=01, CHLEN=1, CKPL=1)	481
图 17-44. PCM 标准短帧同步模式时序图 (DTLEN=00, CHLEN=1, CKPL=0)	481
图 17-45. PCM 标准短帧同步模式时序图 (DTLEN=00, CHLEN=1, CKPL=1)	481
图 17-46. PCM 标准长帧同步模式时序图 (DTLEN=00, CHLEN=0, CKPL=0)	481
图 17-47. PCM 标准长帧同步模式时序图 (DTLEN=00, CHLEN=0, CKPL=1)	481
图 17-48. PCM 标准长帧同步模式时序图 (DTLEN=10, CHLEN=1, CKPL=0)	482
图 17-49. PCM 标准长帧同步模式时序图 (DTLEN=10, CHLEN=1, CKPL=1)	482
图 17-50. PCM 标准长帧同步模式时序图 (DTLEN=01, CHLEN=1, CKPL=0)	482
图 17-51. PCM 标准长帧同步模式时序图 (DTLEN=01, CHLEN=1, CKPL=1)	482
图 17-52. PCM 标准长帧同步模式时序图 (DTLEN=00, CHLEN=1, CKPL=0)	482
图 17-53. PCM 标准长帧同步模式时序图 (DTLEN=00, CHLEN=1, CKPL=1)	482

图 17-54. I2S 时钟生成结构框图.....	483
图 17-55. I2S 初始化流程.....	485
图 17-56. I2S 主机接收禁能流程.....	487
图 18-1. 比较器框图.....	502
图 18-2. 比较器迟滞.....	504
图 18-3. 比较器的输出消隐.....	504

表索引

表 1-1. AHB 互联矩阵的互联关系列表	22
表 1-2. GD32C2x1 系列器件的存储器映射表	24
表 1-3. 引导模式.....	28
表 2-1. 闪存基地址和构成	42
表 2-2. WSCNT 与 AHB 时钟频率对应关系	43
表 2-3. 选项字节.....	50
表 3-1. 省电模式总结	73
表 3-2. 不同模式下模块状态.....	75
表 4-1. 时钟输出的时钟源选择	87
表 4-2. 低速时钟输出的时钟源选择.....	87
表 5-1. Cortex [®] -M23 中的 NVIC 异常类型	114
表 5-2. 中断向量表	115
表 5-3. EXTI 触发源	117
表 6-1. GPIO 配置表	122
表 9-1. DMA 传输操作	152
表 9-2. 中断事件.....	154
表 10-1. 中断事件.....	166
表 10-2. GD32C2x1 DMAMUX 请求路由输入信号映射	167
表 10-3. 触发输入信号映射	168
表 10-4. 同步输入信号映射	169
表 11-1. ADC 内部输入/输出信号.....	177
表 11-2. ADC 引脚定义	177
表 11-3. 常规序列外部触发源.....	183
表 11-4. 不同分辨率对应的 t _{CONV} 时间.....	184
表 11-5. N 和 M 的最大输出值（灰色部分表示截断）	186
表 12-1. 独立看门狗定时器在 32kHz（IRC32K）时的最小/最大超时周期.....	203
表 12-2. 在 48MHz（f _{PCLK1} ）时的最大/最小超时值	208
表 13-1. RTC (PC13 /PA4) 引脚配置.....	220
表 13-2. RTC_OUT 配置.....	220
表 13-3. RTC 低功耗模式管理	221
表 13-4. RTC 中断控制	221
表 14-1. 定时器（TIMERx）分为四种类型.....	235
表 14-2. 由参数控制的互补输出表	254
表 14-3. 计数方向与正交译码器信号之间的关系	257
表 14-4. 从模式示例	260
表 14-5. 计数方向与正交译码器信号之间的关系	312
表 14-6. 从模式例子列表.....	313
表 14-7. 由参数控制的互补输出表	365
表 15-1. USART 重要引脚描述.....	386
表 15-2. 停止位配置	388
表 15-3. USART 中断请求	399

表 16-1. I2C 总线术语说明（参考飞利浦 I2C 规范）	421
表 16-2. 数据建立时间和数据保持时间	426
表 16-3. 可关闭通信模式	427
表 16-4. I2C 错误标志	442
表 16-5. I2C 中断事件	443
表 17-1. SPI 信号描述	459
表 17-2. SPI 四线信号描述	460
表 17-3. 从机模式 NSS 功能	463
表 17-4. 主机模式 NSS 功能	464
表 17-5. SPI 运行模式	464
表 17-6. SPI 中断请求	474
表 17-7. I2S 比特率计算公式	483
表 17-8. 音频采样频率计算公式	483
表 17-9. 各种运行模式下 I2S 接口信号的方向	484
表 17-10. I2S 中断	489
表 18-1 比较器的输入和输出	503
表 19-1. 寄存器功能位访问属性	510
表 19-2. 术语	510
表 20-1. 版本历史	511

1. 系统及存储器架构

GD32C2x1系列器件是基于Arm® Cortex®-M23处理器的32位通用微控制器。Arm® Cortex®-M23处理器的所有存储访问，根据不同的目的和目标存储空间，都会在AHB总线上执行。存储器的组织采用了ARMv8M结构，预先定义的存储器映射和高达4 GB的存储空间，充分保证了系统的灵活性和可扩展性。

1.1. Arm® Cortex®-M23 处理器

Arm® 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。Arm® Cortex®-M23处理器为开发人员提供了显著的好处，包括：

- 一个简单的体系结构，易于学习和编程；
- 超低功耗、高效节能；
- 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。

Arm® 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。

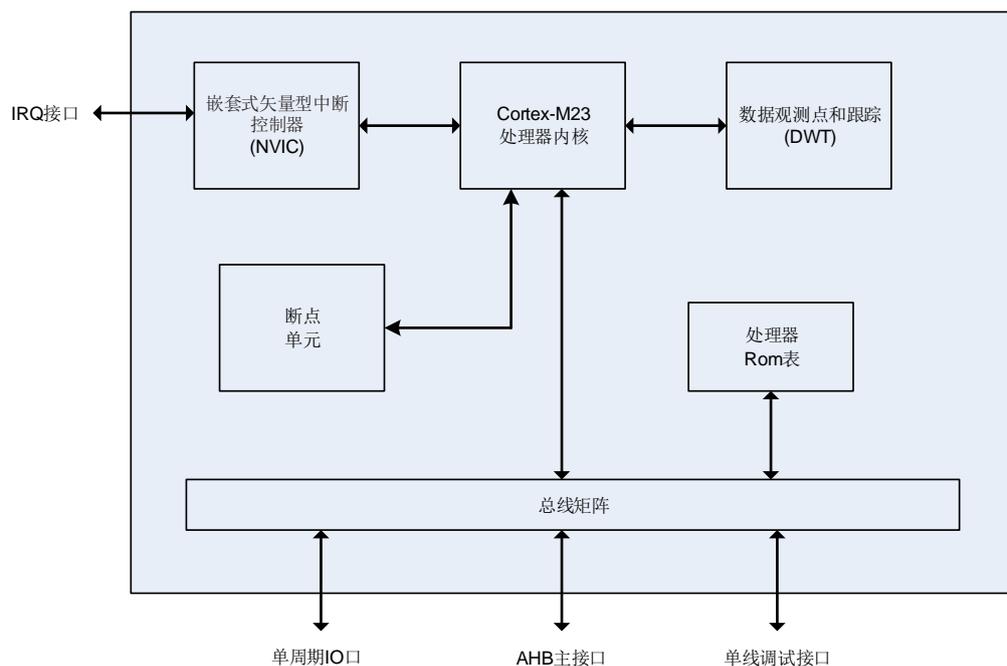
Arm® Cortex®-M23处理器高度集成了一个可配置的嵌套矢量中断控制器（NVIC），以提供业界领先的中断性能。

下面列出由Arm® Cortex®-M23提供的一些系统外设：

- 低延迟，高速外设I/O端口；
- 向量表偏移寄存器；
- 断点单元；
- 数据观测点；
- 串行调试接口。

[图1-1. Arm® Cortex®-M23处理器结构框图](#)显示了Arm® Cortex®-M23处理器结构框图。欲了解更多信息，请参阅Arm® Cortex®-M23技术参考手册。

图 1-1. Arm® Cortex®-M23 处理器结构框图



1.2. 系统架构

GD32C2x1 设备实现了总线矩阵，它被用于管理主机之间的访问仲裁，仲裁采用的是 Round Robin 算法。总线矩阵提供了从主机到从机的访问，即使多个高速外围设备同时工作，也可以进行并发访问和高效运行。GD32C2x1 设备采用 32 位多层总线结构，该结构可使系统中的多个主机和从机之间的并行通信成为可能。多层总线结构包括一个 AHB 互联矩阵、两个 AHB 总线。AHB 互联矩阵的互联关系接下来将进行说明。在[表 1-1. AHB 互联矩阵的互联关系列表](#)，“1”表示相应的主机可以通过 AHB 互联矩阵访问对应的从机，空白的单元格表示相应的主机不可以通过 AHB 互联矩阵访问对应的从机。

表 1-1. AHB 互联矩阵的互联关系列表

	SBUS	DMA
FMC	1	1
SRAM	1	1
AHB1	1	1
AHB2	1	1

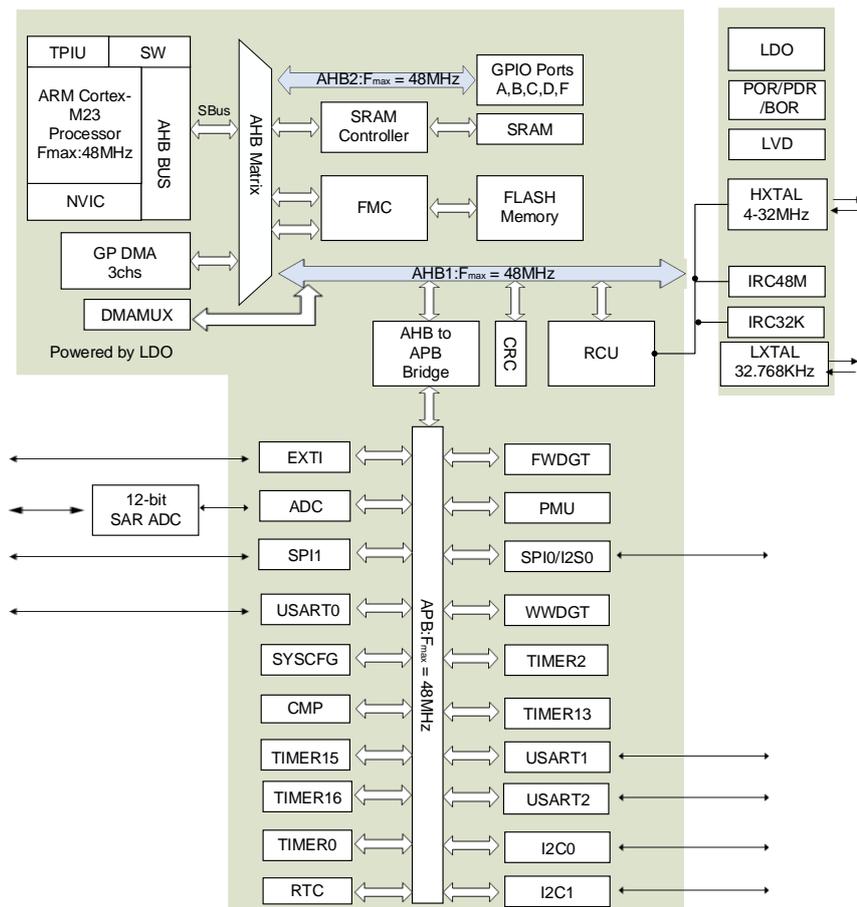
如上表所示，AHB 互联矩阵共连接两个主机，分别为：SBUS 和 DMA。CPU SBUS 将 Cortex®-M23 内核的系统总线（外设总线）连接到管理内核与 DMA 之间的仲裁的总线矩阵。DMA 总线将 DMA 的 AHB 主接口连接到总线矩阵，该矩阵管理 CPU 和 DMA 对 SRAM，FLASH 和 AHB / APB 外设的访问。

AHB 互联矩阵也连接了一些从机，分别为：FMC、SRAM、AHB1 和 AHB2。FMC 是闪存存储器控制器的总线。SRAM 是片上静态随机存取存储器。AHB1 是连接所有 AHB 从机和 AHB 到 APB 桥的 AHB 总线，AHB2 是连接 AHB2 从机的 AHB 总线。AHB 到 APB 的桥接器是与所有 APB 从机连接的 APB 总线。APB 总线与所有 APB 外设相连。APB 速度限制为 48MHz。

GD32C2x1 系列器件的系统架构如下图所示。该 AHB 矩阵是一个基于 AMBA 5.0 AHB-LITE 的多层总线，这个结构使得系统中的多个主机和从机之间的并行通信成为可能。该 AHB 矩阵中包含属于 Arm® Cortex®-M23 内核的 AHB 总线，以及内核外的 DMA 共 2 个主机。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 AHB-APB 总线桥。AHB-APB 总线桥提供了 AHB1 和 APB 总线之间的全同步连。APB 总线连接了所有的 APB 外设。

图 1-2. GD32C2x1 系列器件的系统架构示意图



1.3. 存储器映射

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 4 GB 的地址空间之内。这是 Arm® 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 Arm® 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 GD32C2x1 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 1-2. GD32C2x1 系列器件的存储器映射表

预定义的区域	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	AHB2	0x5004 0000 - 0x5FFF FFFF	保留
		0x5000 0000 - 0x5003 FFFF	保留
		0x4800 1800 - 0x4FFF FFFF	保留
		0x4800 1400 - 0x4800 17FF	GPIOF
		0x4800 1000 - 0x4800 13FF	保留
		0x4800 0C00 - 0x4800 0FFF	GPIOD
		0x4800 0800 - 0x4800 0BFF	GPIOC
		0x4800 0400 - 0x4800 07FF	GPIOB
		0x4800 0000 - 0x4800 03FF	GPIOA
	AHB1	0x4003 8400 - 0x47FF FFFF	保留
		0x4003 8000 - 0x4003 83FF	保留
		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 0C00 - 0x4002 0FFF	保留
		0x4002 0800 - 0x4002 0BFF	DMAMUX
		0x4002 0400 - 0x4002 07FF	保留
		0x4002 0000 - 0x4002 03FF	DMA
	APB	0x4001 8000 - 0x4001 FFFF	保留
		0x4001 7C00 - 0x4001 7FFF	CMP
		0x4001 7800 - 0x4001 7BFF	保留
		0x4001 7400 - 0x4001 77FF	保留
		0x4001 7000 - 0x4001 73FF	保留
		0x4001 6C00 - 0x4001 6FFF	保留
		0x4001 6800 - 0x4001 6BFF	保留
		0x4001 5C00 - 0x4001 67FF	保留
		0x4001 5800 - 0x4001 5BFF	DBG
		0x4001 5400 - 0x4001 57FF	保留
		0x4001 5000 - 0x4001 53FF	保留
		0x4001 4C00 - 0x4001 4FFF	保留
		0x4001 4800 - 0x4001 4BFF	TIMER16
		0x4001 4400 - 0x4001 47FF	TIMER15
0x4001 3C00 - 0x4001 43FF	保留		

预定义的区域	总线	地址范围	外设
		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 2000 - 0x4001 23FF	保留
		0x4001 1C00 - 0x4001 1FFF	保留
		0x4001 1800 - 0x4001 1BFF	保留
		0x4001 1400 - 0x4001 17FF	保留
		0x4001 1000 - 0x4001 13FF	保留
		0x4001 0C00 - 0x4001 0FFF	保留
		0x4001 0800 - 0x4001 23FF	保留
		0x4001 0400 - 0x4001 07FF	EXTI
		0x4001 0000 - 0x4001 03FF	SYSCFG
		0x4000 C000 - 0x4000 FFFF	保留
		0x4000 7400 - 0x4000 BFFF	保留
		0x4000 7000 - 0x4000 73FF	PMU
		0x4000 5C00 - 0x4000 6FFF	保留
		0x4000 5800 - 0x4000 5BFF	I2C1
		0x4000 5400 - 0x4000 57FF	I2C0
		0x4000 4C00 - 0x4000 53FF	保留
		0x4000 4800 - 0x4000 4BFF	USART2
		0x4000 4400 - 0x4000 47FF	USART1
		0x4000 3C00 - 0x4000 43FF	保留
		0x4000 3800 - 0x4000 3BFF	SPI1
		0x4000 3400 - 0x4000 37FF	保留
		0x4000 3000 - 0x4000 33FF	FWDGT
		0x4000 2C00 - 0x4000 2FFF	WWDGT
		0x4000 2800 - 0x4000 2BFF	RTC
		0x4000 2400 - 0x4000 27FF	保留
		0x4000 2000 - 0x4000 23FF	TIMER13
		0x4000 0800 - 0x4000 1FFF	保留
0x4000 0400 - 0x4000 07FF	TIMER2		
0x4000 0000 - 0x4000 03FF	保留		
SRAM		0x2000 3000 - 0x3FFF FFFF	保留
		0x2000 0000 - 0x2000 2FFF	SRAM(12KB)
Code		0x1FFF 7880 - 0x1FFF FFFF	保留
		0x1FFF 7800 - 0x1FFF 787F	Option bytes(128B)
		0x1FFF 7400 - 0x1FFF 77FF	保留
		0x1FFF 7000 - 0x1FFF 73FF	OTP bytes(1KB)

预定义的区域	总线	地址范围	外设
		0x1FFF 0C00 - 0x1FFF 6FFF	保留
		0x1FFE F400 - 0x1FFF 0BFF	System memory(6KB)
		0x0801 0000 - 0x1FFE F3FF	保留
		0x0800 0000 - 0x0800 FFFF	Main Flash memory(64KB)
		0x0000 0000 - 0x07FF FFFF	Aliased to Flash or system memory

1.3.1. 片上 SRAM 存储器

GD32C2x1 系列微控制器含有高达 12KB 的片上 SRAM，起始地址为 0x2000 0000，支持字节、半字(16 比特)和整字(32 比特)访问。

ECC

SRAM 支持 7 比特的 ECC 功能。可纠错 1 比特，发现多比特（两比特）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。

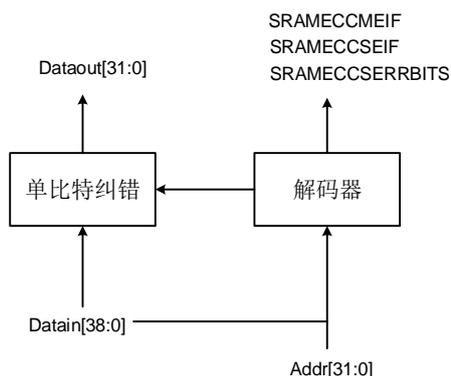
ECC 模块由编码器和解码器两部分构成：

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。

解码器如下图所示：[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。

通过设置 SYSCFG_CFG1 寄存器和选项字节中的 SRAM_ECCEN 位，从而实现 SRAM 的

ECC 错误检测。

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 模块有如下配置：

- (1) SYSCFG_STAT 寄存器中的 ECCSEIF 位置位，软件写 1 可以清除。
- (2) SYSCFG_CFG2 寄存器中的 ECCEADDR[11:0]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

当在 SRAM 中检测到发生了多比特（两比特）不可纠错的事件时，EEIC 模块有如下配置：

- (1) SYSCFG_STAT 寄存器中的 ECCMEIF 位置位，软件写 1 可以清除。
- (2) SYSCFG_SRAM0ECC 寄存器中的 ECCADDR[11:0]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_CFG2 寄存器中设置 ECCSEIE 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 NMI 中断。

多比特（两比特）不可纠错事件

在 SYSCFG_CFG2 寄存器中设置 ECCMEIE 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

GD32C2x1 系列微控制器可以提供高密度片上 FLASH 存储器，按以下分类进行组织：

- 高达64KB主FLASH存储器,高达2KB数据Flash存储器
- 器件配置的选项字节。

更多详细说明请参考[闪存控制器 \(FMC\)](#) 章节。

1.4. 引导配置

主闪存闪空检查

当配置BOOT0引脚从主Flash启动时，通过检查FMC_WS寄存器中的MFPE位，对原始设备进行编程变得更加方便。当设置MFPE位时，设备被认为是空的，并且引导装载程序从系统内存开始，允许在此时进行Flash编程。在加载选项字节期间，如果地址0x0800 0000处的内容被读取为0xFFFF FFFF，则会设置MFPE标志；否则，它将被清除。在对原始设备编程后，MFPE标志可以通过电源复位或设置FMC_CTL寄存器中的OBRDL位来清除，使设备在系统复位后执行用户代码。MFPE标志也可以通过软件修改。

注意：如果设备是首次编程，但没有重新加载选项字节，在系统复位后，设备仍然会选择系统内存作为启动区。

引导模式

GD32C2x1系列微控制器提供了三种引导源，可以通过BOOT0引脚和用户选项字节中的引导模式配置位（BOOTLK比特位，nBOOT1比特位，SWBT0比特位，nBOOT0比特位）来进行选择，详细说明见[表1-3. 引导模式](#)。当SWBT0置零时，BOOT0引脚的电平状态会在复位后的第四个CK_SYS(系统时钟)的上升沿进行锁存。上电复位或系统复位后，由用户设置引导模式配置，选择所需的启动源。当SWBT0置零时，一旦这个引脚电平被采样，它可以被释放并用于其他用途。

表 1-3. 引导模式

引导源选择	启动模式配置				
	BOOTLK	nBOOT1 位	BOOT0 引脚	SWBT0 位	nBOOT0 位
主 Flash 存储器	0	x	0	0	x
System 存储器	0	1	1	0	x
片上 SRAM	0	0	1	0	x
主 Flash 存储器	0	x	x	1	1
System 存储器	0	1	x	1	0
片上 SRAM	0	0	x	1	0
主 Flash 存储器	1	x	x	x	x

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。

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

芯片内嵌的引导装载程序位于系统存储器中，用来对片上闪存的主存进行重编程。对于GD32C2x1产品，该引导装载程序可通过以下接口之一工作：USART0/1，I2C0。

无论其他模式如何配置，都可以通过配BOOTLK位从主Flash存储器的唯一入口进行强制引导。

■ 重映射配置

当选择启动模式配置时，软件可以通过内存重映射配置寄存器(SYSCFG_CFG0)中的BOOT_MODE位配置内存重映射功能。存储器包括主Flash存储器，System存储器，片上SRAM可以重新映射。

1.5. 系统配置控制器（SYSCFG）

系统配置控制器（SYSCFG）的主要用途如下：

- 配置 I2C Fm+和 I/O 模拟开关的电压增强器
- 重映射某些I/O口
- 管理与GPIO的外部中断线连接

1.6. 系统配置寄存器

SYSCFG基地址: 0x4001 0000

1.6.1. 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]可以根据复位后用户选项字节中的BOOT0引脚和启动配置位nBOOT1、SWBT0和nBOOT0的任何值)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留							PC14FMP	PA10FMP	PA9FMPE	保留	I2C0_FMP	PB9FMP	PB8FMPE	PB7FMPE	PB6FMPE
							EN	EN	N			EN	N	N	N
							rw	rw	rw		rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留											PA12_RM	PA11_RM	保留	BOOT_MODE[1:0]	
											P	P			
											rw	rw		rw	

位/位域	名称	描述
31:25	保留	必须保持复位值。
24	PC14FMPEN	PC14 引脚 Fm+模式使能 该位控制 PC14 引脚上的 I2C Fm+功能，同时该引脚的速度控制被忽略 0: 禁能 Fm+模式 1: 使能Fm+模式
23	PA10FMPEN	PA10 引脚 Fm+模式使能 该位控制 PA10 引脚上的 I2C Fm+功能，同时该引脚的速度控制被忽略 0: 禁能 Fm+模式 1: 使能Fm+模式
22	PA9FMPEN	PA9 引脚 Fm+模式使能 该位控制 PA9 引脚上的 I2C Fm+功能，同时该引脚的速度控制被忽略 0: 禁能 Fm+模式 1: 使能Fm+模式
21	保留	必须保持复位值。
20	I2C0_FMPEN	I2C0 Fm+模式使能 该位由软件设置和清除。它通过 GPIOx_AFSELx 寄存器在配置为 I2C0 的 I/O 端口上启用 I2C FM+驱动能力。 在该位处于除能状态时，相应的 I2C0_FMPEN 位启用配置为 I2C0 的 I/O 端口上的 I2C FM+驱动能力。I2C FM+使能时，忽略速度控制。 0: 禁能 Fm+模式 1: 使能 Fm+模式
19	PB9FMPEN	PB9 引脚 Fm+模式使能

		该位控制 PB9 引脚上的 I2C Fm+功能，同时该引脚的速度控制被忽略 0: 禁能 Fm+模式 1: 使能 Fm+模式
18	PB8FMPEN	PB8 引脚 Fm+模式使能 该位控制 PB8 引脚上的 I2C Fm+功能，同时该引脚的速度控制被忽略 0: 禁能 Fm+模式 1: 使能 Fm+模式
17	PB7FMPEN	PB7 引脚 Fm+模式使能 该位控制 PB7 引脚上的 I2C Fm+功能，同时该引脚的速度控制被忽略 0: 禁能 Fm+模式 1: 使能 Fm+模式
16	PB6FMPEN	PB6 引脚 Fm+模式使能 该位控制 PB6 引脚上的 I2C Fm+功能，同时该引脚的速度控制被忽略 0: 禁能 Fm+模式 1: 使能 Fm+模式
15:5	保留	必须保持复位值。
4	PA12_RMP	PA12引脚重映射 该位由软件设置和清除 当该位为1时，重映射PA12引脚到PA10 GPIO端口，而不是PA12 GPIO端口。 0: 无重映射（PA12） 1: 重映射（PA10）
3	PA11_RMP	PA11引脚重映射 该位由软件设置和清除 当该位为1时，重映射PA11引脚到PA9 GPIO端口，而不是PA11 GPIO端口。 0: 无重映射（PA11） 1: 重映射（PA9）
2	保留	必须保持复位值。
1:0	BOOT_MODE[1:0]	引导模式(详细请参考 引导配置 章节) 位0映射到BOOT0引脚；位1的值映射到nBOOT1位 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

1.6.2. EXTI 源选择寄存器 0 (SYSCFG_EXTISS0)

地址偏移: 0x08

复位值: 0x0000 0000

该寄存器只能按字（32位）访问

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

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

位/位域	名称	描述
31:14	保留	必须保持复位值。
13:12	EXTI3_SS[1:0]	EXTI 3源选择 00: PA3引脚 01: PB3引脚 10: PD3引脚 11: PF3引脚
11:10	保留	必须保持复位值。
9:8	EXTI2_SS[1:0]	EXTI 2源选择 00: PA2引脚 01: PB2引脚 10: PD2引脚 11: PF2引脚
7:6	保留	必须保持复位值。
5:4	EXTI1_SS[1:0]	EXTI 1源选择 00: PA1引脚 01: PB1引脚 10: PD1引脚 11: PF1引脚
3:2	保留	必须保持复位值。
1:0	EXTI0_SS[1:0]	EXTI 0源选择 00: PA0引脚 01: PB0引脚 10: PD0引脚 11: PF0引脚

1.6.3. 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
保留		EXTI7_SS [1:0]		保留		EXTI6_SS [1:0]		保留		EXTI5_SS		保留		EXTI4_SS	

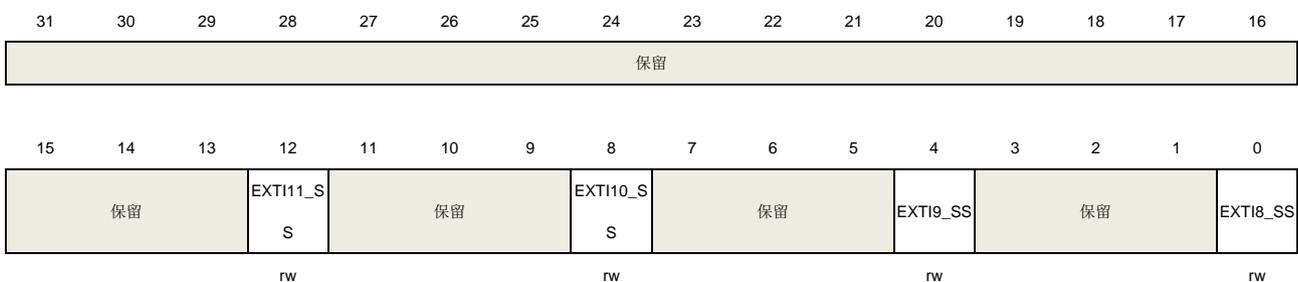
位/位域	名称	描述
31:14	保留	必须保持复位值。
13:12	EXTI7_SS[1:0]	EXTI 7源选择 00: PA7引脚 01: PB7引脚 10: PC7引脚 11: 保留
11:10	保留	必须保持复位值。
9:8	EXTI6_SS[1:0]	EXTI 6源选择 00: PA6引脚 01: PB6引脚 10: PC6引脚 11: 保留
7:6	保留	必须保持复位值。
5:4	EXTI5_SS	EXTI 5源选择 0: PA5引脚 1: PB5引脚
3:2	保留	必须保持复位值。
1:0	EXTI4_SS[3:0]	EXTI 4源选择 0: PA4引脚 1: PB4引脚

1.6.4. 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



位/位域	名称	描述
31:13	保留	必须保持复位值。

12	EXTI11_SS	EXTI 11源选择 0: PA11引脚 1: PB11引脚
11:9	保留	必须保持复位值。
8	EXTI10_SS	EXTI 10源选择 0: PA10引脚 1: PB10引脚
7:5	保留	必须保持复位值。
4	EXTI9_SS	EXTI 9源选择 0: PA9引脚 1: PB9引脚
3:1	保留	必须保持复位值。
0	EXTI8_SS	EXTI 8源选择 0: PA8引脚 1: PB8引脚

1.6.5. 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
31:14	保留	必须保持复位值。
13:12	EXTI15_SS[1:0]	EXTI 15源选择 00: PA15引脚 01: PB15引脚 10: PC15引脚 11: 保留
11:10	保留	必须保持复位值。
9:8	EXTI14_SS[3:0]	EXTI 14源选择 00: PA14引脚

		01: PB14引脚 10: PC14引脚 11: 保留
7:6	保留	必须保持复位值。
5:4	EXTI13_SS[1:0]	EXTI 13源选择 00: PA13引脚 01: PB13引脚 10: PC13引脚 11: 保留
3:1	保留	必须保持复位值。
0	EXTI12_SS	EXTI 12源选择 0: PA12引脚 1: PB12引脚

1.6.6. 配置寄存器 1 (SYSCFG_CFG1)

地址偏移: 0x18

复位值: 0x0000 0000



位/位域	名称	描述
31:2	保留	必须保持复位值。
1	SRAM_ECC_LOCK	SRAM ECC锁定使能 该位由软件设置，仅在系统复位被清除。 0: SRAM ECC错误从TIMER0/15/16的break输入端断开。 1: SRAM ECC错误与TIMER0/15/16的break输入端连接。
0	LOCKUP_LOCK	Cortex®-M23 lockup (Hardfault)输出锁定使能 该位由软件设置，仅在系统复位被清除。 0: Cortex®-M23 LOCKUP 输出从 TIMER0/15/16 的 break 输入端断开。 1: Cortex®-M23 LOCKUP 输出与 TIMER0/15/16 的 break 输入端连接。

1.6.7. 系统状态寄存器 (SYSCFG_STAT)

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
31:2	保留	必须保持复位值。
1	ECCSEIF	单比特可纠错事件中断标志 软件写 1 清零。 0: 没有检测到 SRAM ECC 错误事件。 1: 检测到 SRAM ECC 错误事件。
0	ECCMEIF	SRAM 多比特 (两比特) 纠正错误事件中断标志 软件写 1 清零。 0: 没有检测到 SRAM ECC 多比特 (两比特) 不可纠错事件。 1: 检测到 SRAM ECC 多比特 (两比特) 不可纠错事件。

1.6.8. 配置寄存器 2 (SYSCFG_CFG2)

地址偏移: 0x28

复位值: 0x0000 0007



位/位域	名称	描述
31:20	ECCEADDR[11:0]	记录上一次发生 SRAM ECC 事件的 SRAM 故障地址
19:16	保留	必须保持复位值。
15:10	ECCSERRBITS[5:0]	发生 ECC 单比特可纠错事件的比特 0: 无错误 1: 比特 0 ... 31: 比特 31

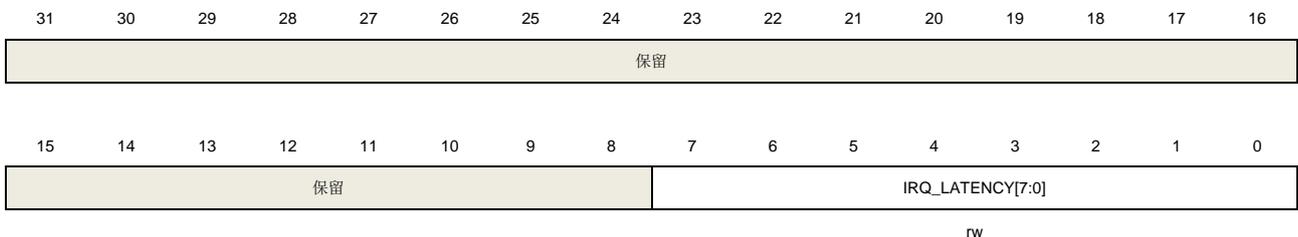
9:5	保留	必须保持复位值。
4	ECCSEIE	SRAM 单比特可纠错中断使能 0: 禁能 SRAM 单比特可纠错 NMI 中断。 1: 使能 SRAM 单比特可纠错 NMI 中断。
3	ECCMEIE	SRAM 多比特（两比特）不可纠错 NMI 中断使能 0: 禁能 SRAM 多比特（两比特）不可纠错 NMI 中断。 1: 使能 SRAM 多比特（两比特）不可纠错 NMI 中断。
2	HXTALCS_IE	HXTAL 时钟阻塞中断（NMI 和 EXTI）使能 0: 禁能 1: 使能
1	LXTALCS_IE	LXTAL 时钟阻塞中断（NMI 和 EXTI）使能 0: 禁能 1: 使能
0	保留	必须保持复位值。

1.6.9. 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



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，则尽可能快地响应中断。 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获取之间存在最小的 IRQ_LATENCY + 1 个 HCLK 周期。

1.6.10. TIMERx 配置寄存器 0（SYSCFG_TIMERxCFG0, x=0, 2）

地址偏移: 0x110适用于TIMER0

地址偏移: 0x118适用于TIMER2

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	TSCFG7[2:0]			保留	TSCFG6[2:0]			保留	TSCFG5 [2:0]			保留	TSCFG4 [2:0]		
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TSCFG3 [2:0]			保留	TSCFG2 [2:0]			保留	TSCFG1 [2:0]			保留	TSCFG0 [2:0]		
rw				rw				rw				rw			

位/位域	名称	描述
31	保留	必须保持复位值。
30:28	TSCFG7[2:0]	复位+事件模式配置 当 TSCFG7[2:0]不为零，选中的触发输入的上升沿重新初始化计数器，并且更新影子寄存器。 000: 复位+事件模式禁能 001: 内部触发输入 0 (ITIO) 010: 内部触发输入 2 (ITI2) 011: 内部触发输入 3 (ITI3) 100: CI0 的边沿标志位 (CIOF_ED) 101: 滤波后的通道 0 输入 (CIOFE0) 110: 滤波后的通道 1 输入 (CI1FE1) 111: 滤波后的外部触发输入 (ETIFP)
27	保留	必须保持复位值。
26:24	TSCFG6[2:0]	外部时钟模式 0 配置 在 TSCFG6[2:0]非零时，选中的触发输入的上升沿驱动计数器。 000: 外部时钟模式 0 禁能 001: 内部触发输入 0 (ITIO) 010: 内部触发输入 2 (ITI2) 011: 内部触发输入 3 (ITI3) 100: CI0 的边沿标志位 (CIOF_ED) 101: 滤波后的通道 0 输入 (CIOFE0) 110: 滤波后的通道 1 输入 (CI1FE1) 111: 滤波后的外部触发输入 (ETIFP)
23	保留	必须保持复位值。
22:20	TSCFG5[2:0]	事件模式配置 计数器在触发输入的上升沿启动。 000: 事件模式禁能 001: 内部触发输入 0 (ITIO) 010: 内部触发输入 2 (ITI2) 011: 内部触发输入 3 (ITI3) 100: CI0 的边沿标志位 (CIOF_ED) 101: 滤波后的通道 0 输入 (CIOFE0) 110: 滤波后的通道 1 输入 (CI1FE1)

		111: 滤波后的外部触发输入 (ETIFP)
19	保留	必须保持复位值。
18:16	TSCFG4[2:0]	<p>暂停模式配置</p> <p>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, 且 TSCFG4[2:0]不为零, 则计数器停止。</p> <p>000: 暂停模式禁能</p> <p>001: 内部触发输入 0 (ITI0)</p> <p>010: 内部触发输入 2 (ITI2)</p> <p>011: 内部触发输入 3 (ITI3)</p> <p>100: 保留</p> <p>101: 滤波后的通道 0 输入 (CI0FE0)</p> <p>110: 滤波后的通道 1 输入 (CI1FE1)</p> <p>111: 滤波后的外部触发输入 (ETIFP)</p>
15	保留	必须保持复位值。
14:12	TSCFG3[2:0]	<p>复位模式配置</p> <p>当 TSCFG3[4:0]不为零, 选中的触发输入的上升沿重新初始化计数器, 并且更新影子寄存器。</p> <p>000: 复位模式禁能</p> <p>001: 内部触发输入 0 (ITI0)</p> <p>010: 内部触发输入 2 (ITI2)</p> <p>011: 内部触发输入 3 (ITI3)</p> <p>100: CI0 的边沿标志位 (CI0F_ED)</p> <p>101: 滤波后的通道 0 输入 (CI0FE0)</p> <p>110: 滤波后的通道 1 输入 (CI1FE1)</p> <p>111: 滤波后的外部触发输入 (ETIFP)</p>
11	保留	必须保持复位值。
10:8	TSCFG2[2:0]	<p>正交解码器模式 2 配置</p> <p>000: 正交解码器模式 2 禁能</p> <p>其他: 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。</p>
7	保留	必须保持复位值。
6:4	TSCFG1[2:0]	<p>正交解码器模式 1 配置</p> <p>000: 正交解码器模式 1 禁能</p> <p>其他: 根据 CI0FE0 的电平, 计数器在 CI1FE1 的边沿向上/下计数。</p>
3	保留	必须保持复位值。
2:0	TSCFG0[2:0]	<p>正交解码器模式 0 配置</p> <p>000: 正交解码器模式 0 禁能</p> <p>其他: 根据 CI0FE0 的电平, 计数器在 CI1FE1 的边沿向上/下计数。</p>

1.6.11. TIMERx 配置寄存器 1 (SYSCFG_TIMERxCFG0, x=0, 2)

地址偏移: 0x114适用于TIMER0

地址偏移: 0x11C适用于TIMER2

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问。



位/位域	名称	描述
31:3	保留	必须保持复位值。
2:0	TSCFG8[2:0]	内部触发输入源配置 000: 保留 001: 内部触发输入 0 (ITIO) 010: 内部触发输入 2 (ITI2) 011: 内部触发输入 3 (ITI3) 100: CI0 的边沿标志位 (CIOF_ED)

1.7. 设备电子签名

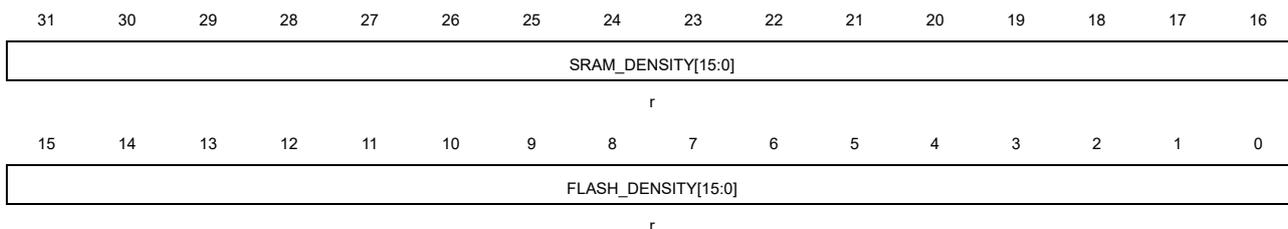
设备的电子签名中包含的存储容量信息和96位的唯一设备ID。它被存储在片上Flash的信息模块中。96位唯一设备ID对于每颗芯片而言都是唯一的。它可以用作序列号, 或安全密钥的一部分, 等等。

1.7.1. 存储容量信息

基地址: 0x1FFF 0BE0

该值是原厂设定的, 不能由用户修改。

该寄存器只能按字 (32位) 访问



位/位域	名称	描述
31:16	SRAM_DENSITY	SRAM存储器容量

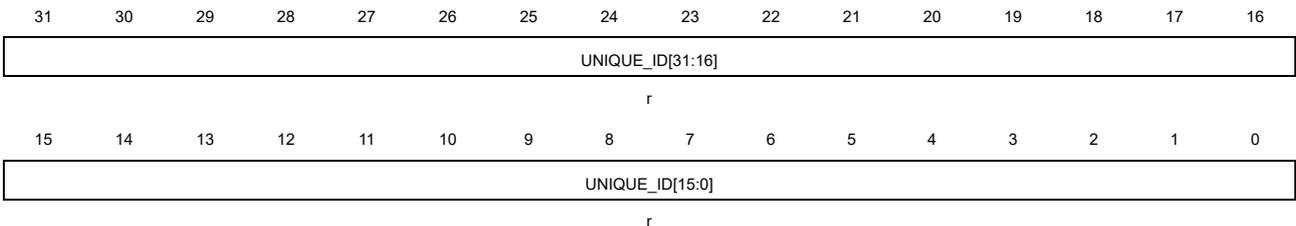
[15:0] 该值表明芯片的片上SRAM存储器容量，以Kbytes为单位
例如：0x0008表示8Kbytes。

15:0 FLASH_DENSITY Flash存储器容量
[15:0] 该值表明芯片的片上Flash容量，以Kbytes为单位
例如：0x0020表示32Kbytes。

1.7.2. 设备唯一 ID（96 位/位域）

基地址：0x1FFF 0BE8
该值是原厂设定的，不能由用户修改。

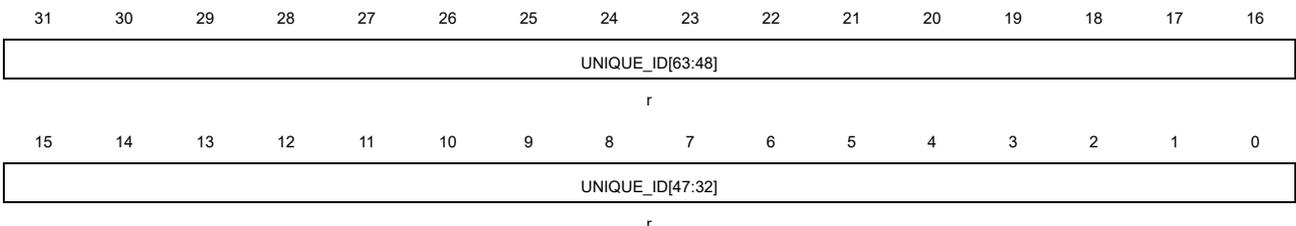
该寄存器只能按字（32位）访问



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

基地址：0x1FFF 0BEC
该值是原厂设定的，不能由用户修改。

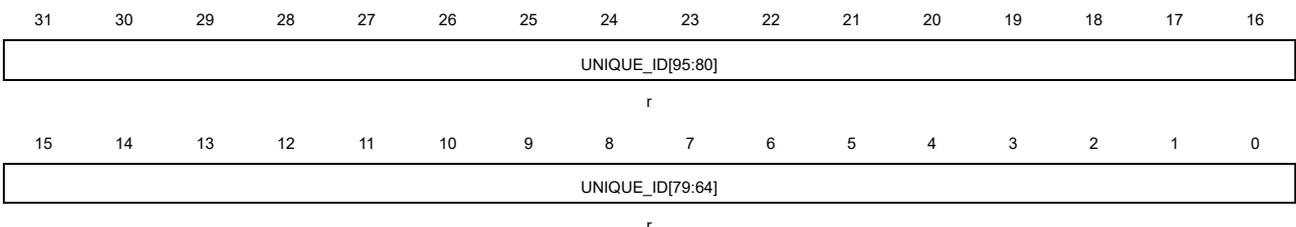
该寄存器只能按字（32位）访问



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

基地址：0x1FFF 0BF0
该值是原厂设定的，不能由用户修改。

该寄存器只能按字（32位）访问



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

2. 闪存控制器（FMC）

2.1. 简介

闪存控制器（FMC），提供了片上闪存需要的所有功能。在主闪存空间内，CPU执行指令需要少量等待时间。FMC提供了页擦除，整片擦除，编程操作，读写保护，以及闪存安全机制等。

2.2. 主要特性

- 高达64KB字节的片上闪存可用于存储指令或数据，高达1KB一次性编程块，3KB信息块。
 - 主闪存块：64KB。
 - 信息块：3KB。
 - 一次性编程块：1KB。
 - 选项字节块：128B。
- 在闪存空间内，CPU执行指令和读取数据需要0~1个等待时间。
- 预取缓冲区以加速读操作。
- 指令缓存：2条64位的ICode缓存线（16字节RAM）
- 支持64位双字编程，页擦除和整片擦除操作。
- 1K字节OTP块（一次性编程），用于存储用户数据。
- 大小为128字节的选项字节可根据用户需求配置。
- 当系统复位时，选项字节被上载到选项字节控制寄存器。
- 具有安全保护状态，可阻止对代码或数据的非法读访问。
- 具有2块仅执行的专用代码读保护区域
- 提供页擦除/写保护功能（WP）。
- 提供用户安全区（SCR）。

2.3. 功能说明

2.3.1. 闪存结构

主存储闪存高达64KB，由64页组成，每页1KB，还包含3KB的用于引导装载程序的信息块。主存储闪存的每页都可以单独擦除。基地址和大小如[表 2-1. 闪存基地址和构成](#)所示。

表 2-1. 闪存基地址和构成

闪存块	名称	地址范围	大小（字节）
主存储闪存块	第 0 页	0x0800 0000 - 0x0800 03FF	1KB
	第 1 页	0x0800 0400 - 0x0800 07FF	1KB
	第 2 页	0x0800 0800 - 0x0800 0BFF	1KB
	.	.	.
	.	.	.
	.	.	.

闪存块	名称	地址范围	大小 (字节)
	第 63 页	0x0800 FC00 - 0x0800 FFFF	1KB
信息块	引导装载程序 (1)	0x1FFF 0000 - 0x1FFF 0BFF	3KB
选项字节块	选项字节	0x1FFF 7800 - 0x1FFF 787F	128B
一次性编程块	一次性编程字 节(2)	0x1FFF 7000~0x1FFF 73FF	1KB

注意: 1.信息块存储了引导装载程序 (bootloader), 不能被用户编程或擦除。

2.1KB (128双字) OTP (一次性编程) 数据区域供用户使用。OTP数据不能被擦除, 只能写一次。如果任何位被写为0, 则该位所在的整个双字都不能被改写。

2.3.2. 主闪存空校验

闪存接口提供主闪存空校验机制, 该机制通过检查主闪存的第一个位置是否已被编程来实现。这个空校验状态的结果可以从 FMC_WS 寄存器的 MFPE 位读取。与 boot0 和 boot1 信息结合使用, 用于确定系统从何处启动。它防止系统在没有用户代码时从主闪存区启动。软件可以通过在 FMC_WS 寄存器的 MFPE 位上写入适当的值来修改主闪存的空状态。

2.3.3. 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使用 CPU 的 CBUS 总线。

增加等待状态

读取闪存时, 需要根据 AHB 时钟频率正确配置 FMC_WS 寄存器中的 WSCNT 位。WSCNT 与 AHB 时钟频率的关系如 [表 2-2. WSCNT 与 AHB 时钟频率对应关系](#) 所示。

表 2-2. WSCNT 与 AHB 时钟频率对应关系

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)

如果发生系统复位, AHB时钟频率为12MHz, 此时WSCNT置为0。

注意:

1. 如果希望增加 AHB 时钟频率。首先, 参考[表 2-2. WSCNT 与 AHB 时钟频率对应关系](#), 根据目标 AHB 时钟频率配置 WSCNT 位。然后, 增加 AHB 时钟频率至目标频率。禁止在配置 WSCNT 位之前增加 AHB 时钟频率。
2. 如果希望降低 AHB 时钟频率。首先, 降低 AHB 时钟频率至目标频率。然后, 参考[表 2-2. WSCNT 与 AHB 时钟频率对应关系](#), 根据目标 AHB 时钟频率配置 WSCNT 位。禁止在降低 AHB 时钟频率之前配置 WSCNT 位。

由于添加了等待状态, 读效率较低 (例如: 48MHz 时需添加 1 个等待状态)。为了加速读操作, 需要用到以下功能。

当前缓存区

当前缓存区总是被使能的。每次从闪存中读取数据时，当前缓存区可以缓存 64 位数据。因为 CPU 每次读操作只需要 32 位或 16 位数据。因此在顺序代码下，CPU 所需数据可以从当前缓存区获取而不必重复从闪存中获取。

预取缓存区

置位 FMC_WS 寄存器中 PFEN 位来使能预取缓存区。在顺序代码下，当 CPU 执行来自当前缓存区的数据时（64 位），按 32 位执行时需要至少 2 个时钟周期，按 16 位执行时需要至少 4 个时钟周期。在这种情况下，从 flash 闪存中预取下一个双字地址的数据并存储在预取缓存区。当 CPU 执行完当前缓存区的数据时，预取缓存区提供下次需要执行的数据。

指令缓存区

置位 FMC_WS 寄存器中 ICEN 位来使能指令缓存区。指令缓存区仅在 CBUS 取数据时使用。指令缓存区为 16 个字节，由 2 条缓存线组成，每条缓存线为 64 位。

当指令在指令缓存区时（缓存命中），CPU 从指令缓存区读取指令无延迟。当指令不在指令缓存区并且也不在当前缓存区/预取缓存区时（缓存未命中），指令缓存区从闪存中读取指令并复制到指令缓存区。当所有指令缓存线被填充，LRU（最近最少使用）策略被用于转移指令缓存线中的代码。

2.3.4. FMC_CTL/FMC_OBCTL 寄存器解锁

复位后，FMC_CTL 寄存器进入锁定状态，该寄存器中的 LK 位为 1。通过先后向 FMC_KEY 寄存器写入 0x45670123 和 0xCDEF89AB，可以使得 FMC_CTL 解锁。两次写操作后，FMC_CTL 寄存器的 LK 位被硬件清 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，并引发一个总线错误。

FMC_OBCTL 寄存器、FMC_CTL 中的 OBRLD 位和 OBSTART 位在 FMC_CTL 被解锁后仍然处于被保护状态。解锁过程为两次写操作，向 FMC_OBKEY 寄存器先后写入 0x08192A3B 和 0x4C5D6E7F，然后硬件将 FMC_CTL 寄存器中的 OBLK 位清零。软件可以将 FMC_CTL 的 OBLK 位置 1 来锁定 FMC_OBCTL 寄存器、FMC_CTL 中的 OBRLD 位和 OBSTART 位。

2.3.5. 页擦除

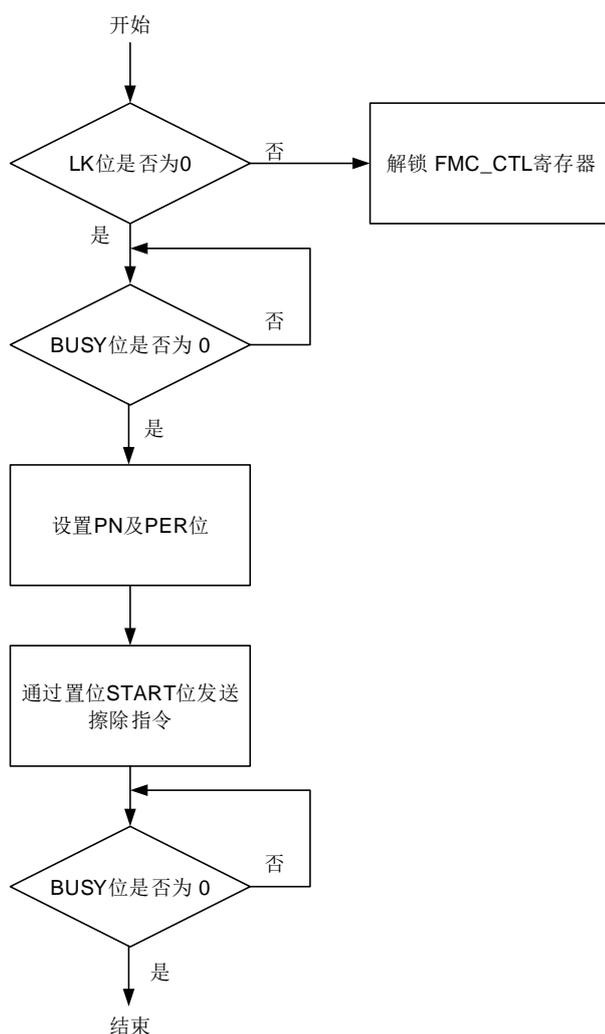
FMC 的页擦除功能使得主存储闪存的页内容初始化为高电平。每一页都可以被独立擦除，而不影响其他页内容。页擦除页操作，寄存器设置具体步骤如下：

- 确保 FMC_CTL 寄存器不处于锁定状态。
- 检查 FMC_STAT 寄存器的 BUSY 位来判定闪存是否正处于擦写访问状态，若 BUSY 位为 1，则需等待该操作结束，BUSY 位变为 0。
- 选择要擦除的页（PN）。
- 将页擦除命令写入到 FMC_CTL 寄存器的 PER 位。
- 通过将 FMC_CTL 寄存器的 START 位置 1 来发送页擦除命令到 FMC。

- 等待擦除指令执行完毕，FMC_STAT寄存器的BUSY位清0。
- 如果需要，使用CBUS读并验证该页是否擦除成功。

当页擦除操作成功执行时，FMC_STAT 寄存器中的 ENDF 将被置位。如果 FMC_CTL 寄存器中的 ENDIE 位被置位，FMC 将触发中断。需要注意的是，用户需确保写入的是正确的擦除页，否则当待擦除页的地址被用来取指令或访问数据时，软件将会“跑飞”。该情况下，FMC 不会有任何出错提示。另一方面，对擦写保护的页进行擦除操作将无效。此时如果 FMC_CTL 寄存器中的 ERRIE 位置位，则 FMC 将触发闪存操作错误中断。软件可以在中断处理程序中检查 FMC_STAT 寄存器中的 WPERR 位来检测这种情况。页擦除操作流程如 [图 2-1. 页擦除操作流程](#) 所示。

图 2-1. 页擦除操作流程



2.3.6. 整片擦除

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

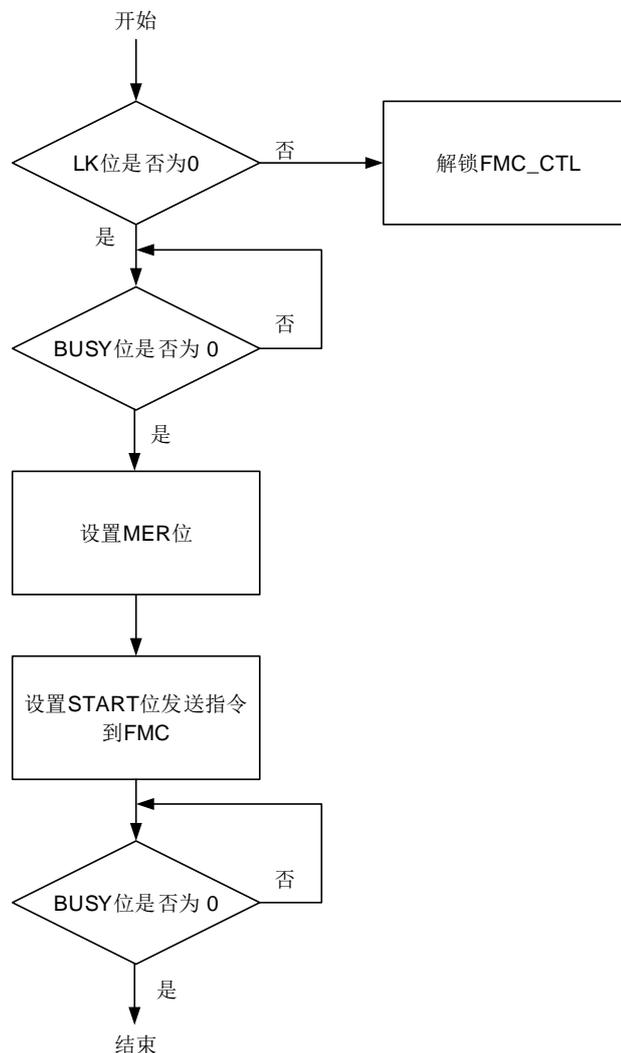
- 确保FMC_CTL寄存器不处于锁定状态。
- 检查FMC_STAT寄存器的BUSY位来判定闪存是否正处于擦写访问状态，若BUSY位为1，

则需等待该操作结束，BUSY位变为0。

- 将整片擦除命令写入到FMC_CTL寄存器的MER位。
- 通过将FMC_CTL寄存器的START位置1来发送整片擦除命令到FMC。
- 等待擦除指令执行完毕，FMC_STAT寄存器的BUSY位清0。
- 如果需要，使用CBUS读并验证是否擦除成功。

当整片擦除成功执行，FMC_STAT寄存器的ENDF位置位。若FMC_CTL寄存器的ENDIE位被置1，FMC将触发一个中断。由于所有的闪存数据都将被复位为0xFFFFFFFF，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。整片擦除操作流程如[图 2-2. 整片擦除操作流程](#)所示

图 2-2. 整片擦除操作流程



2.3.7. 主闪存编程

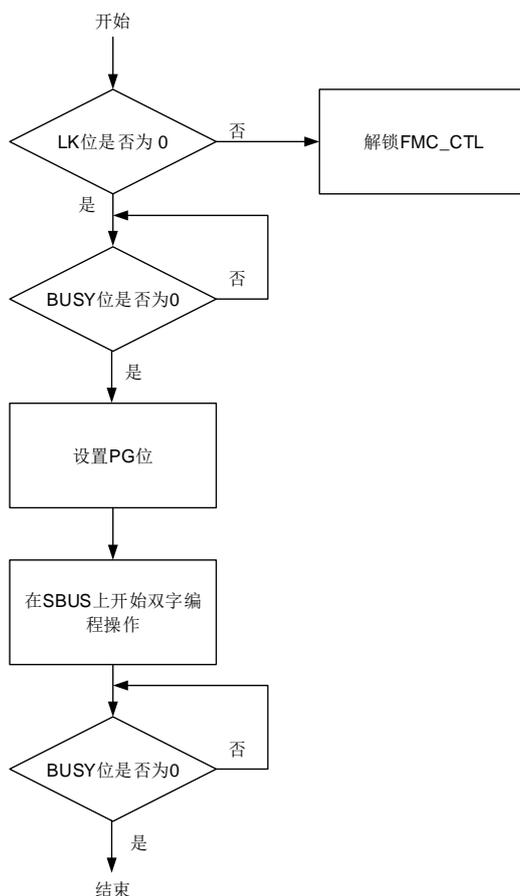
FMC 提供了用于修改主闪存的双字编程功能（2 x 32 位）。下面的步骤显示了编程操作的寄存器设置顺序。

- 确保FMC_CTL寄存器不处于锁定状态。

- 检查FMC_STAT寄存器的BUSY位来判定闪存是否正处于擦写访问状态，若BUSY位为1，则需等待该操作结束，BUSY位变为0。
- 置位FMC_CTL寄存器的PG位。
- SBUS将待编程数据写入到绝对地址（0x08XX XXXX）。SBUS写入2次组成一个64位数据，然后将64位数据写入到闪存。待编程数据必须双字对齐
- 等待编程指令执行完毕，FMC_STAT寄存器的BUSY位清0。
- 如果需要，使用SBUS读并验证是否编程成功。

当主存储块编程执行成功时，FMC_STAT 寄存器中的 ENDF 置位，如果 FMC_CTL 寄存器中的 ENDIE 位置位，FMC 将触发中断。双字编程操作之前需要检查目的地址是否已经被擦除。如果该地址没有被擦除，即使编程 0x0，FMC_STAT 寄存器的 PGERR 位也将被置 1。另外，在擦写保护页上的编程操作会被忽略，同时 FMC_STAT 中的 WPERR 位将置位。在这些情况下，如果 FMC_CTL 寄存器中的 ERRIE 位被置位，则 FMC 将产生一个闪存操作错误中断。软件可以检查 FMC_STAT 寄存器中的 PGERR、PGSERR、PGMERR、PGAERR 和 RPERR 位，以检测出现哪种错误。主闪存块双字编程操作流程如 [图 2-3. 双字编程操作流程](#) 所示。

图 2-3. 双字编程操作流程



注意：当编程/擦除操作正在进行时，应避免读取闪存。

如果编程/擦除操作被掉电、复位等意外中断，闪存中的内容将无法保证并处在一种不确定的状态。因此，应采取适当的措施，以避免由于程序中断/擦除而造成数据丢失。

2.3.8. 主闪存快速编程

FMC提供快速编程功能，用于修改主闪存内容。在这种模式下，一行（16字 = 64字节）可一次编程到主闪存中，因为在编程前无需对闪存位置进行验证，避免了每个字的高电压上升和下降时间，从而减少了页编程时间。在快速编程时，闪存时钟（HCLK）频率至少为8MHz。

下面的步骤说明了快速编程操作寄存器的访问顺序：

- 执行一次页或整片擦除。否则PGSERR将会被置位。
- 确保FMC_CTL寄存器不处于锁定状态。
- 检查FMC_STAT寄存器的BUSY位来判定闪存是否正处于擦写访问状态，若BUSY位为1，则需等待该操作结束，BUSY位变为0。
- 置位FMC_CTL寄存器的FSTPG位。
- 向要编程的绝对地址（0x08XX XXXX）处写入一行数据（16个字）。
- 等待所有编程指令执行完毕，FMC_STAT寄存器的BUSY位清0。

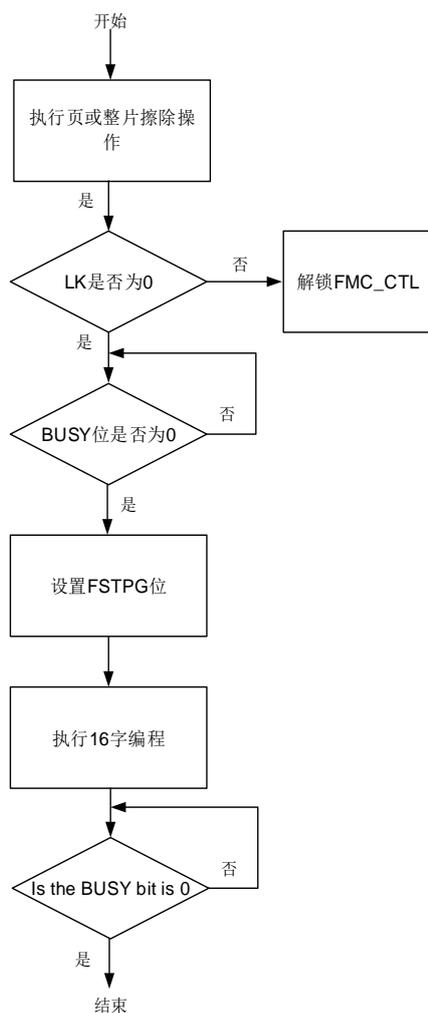
在快速编程中：如果要编程的数据与之前编程的字不在同一行，或者要编程的地址不大于之前的地址，则FMC_STAT中的PGAERR和FSTPERR位将置位，同时将快速编程请求转为普通编程，完成写请求。当操作成功执行时，FMC_STAT寄存器中的ENDF位将被设置，如果FMC_CTL寄存器中的ENDIE位被置位，将产生一个中断。

在后续的快速编程中，如果确认某一行被擦除，则重复以上所有步骤来快速编程下一行。如果不再有编程请求，清除FMC_CTL寄存器中的FSTPG位。

如果在保护页上进行擦除/编程操作，该请求将被忽略，并且FMC_STAT中的WPERR位将被置位。

在这种情况下，如果设置了FMC_CTL寄存器的ERRIE位，则会产生闪存操作的错误中断。软件可以检查FMC_STAT寄存器中的PGSERR / PGAERR / WPERR / PGERR位，以检测中断处理程序中发生的以上哪种情况。快速编程操作流程如 [图2-4. 快速编程操作流程](#)所示。

图 2-4. 快速编程操作流程



注意： 1.16个字必须连续写。

2.16个字必须对齐。

3.由于快速编程不通过硬件检测闪存中的0xFF，所以必须先软件检测0xFF，不能在两次擦除之间对一行进行两次或多次编程。如果对一行进行两次或两次以上的擦除，可能会发生不可预测的结果。

4.快速编程的代码在SRAM中运行。如果在闪存中运行，由于读取闪存代码，快速编程请求将变成普通编程。

2.3.9. OTP 编程

OTP 编程方法与主闪存编程相同。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，闪存中的数据有很小可能性会出错。

2.3.10. 选项字节

选项字节说明

每次系统复位或在 FMC_CTL 寄存器中 OBRDLD 位置 1 后，闪存的选项字节寄存器重新加载到相应选项字节块中，选项字节生效。选项补码字节与选项字节相反。当选项字节重新加载时，如果选项字节补码与选项字节不匹配，则 FMC_STAT 寄存器中的 OBERR 位将被置 1。选项字节详情见下表 [表 2-3. 选项字节](#)。

表 2-3. 选项字节

地址	名称	出厂值	说明
0x1fff 7800	OB_SPC[7:0]	0xA5	选项字节安全保护值 0xA5: 未保护状态 0xCC: 保护级别高 除 0xA5 和 0xCC 外的任何值: 保护级别低
0x1fff 7801	OB_USER[7:0]	0xFE	[7]: 保留 [6]: nRST_STDBY 0: 进入待机模式时产生复位 1: 进入待机模式时不产生复位 [5]: nRST_DPSLP 0: 进入深度睡眠模式时产生复位 1: 进入深度睡眠模式时不产生复位 [4:3]: BORF_TH 00: BOR 下降等级 1, 阈值在 2.0 V 左右 01: BOR 下降等级 2, 阈值在 2.2 V 左右 10: BOR 下降等级 3, 阈值在 2.5 V 左右 11: BOR 下降等级 4, 阈值在 2.8 V 左右 [2:1]: BORR_TH 00: BOR 上升等级 1, 阈值在 2.1 V 左右 01: BOR 上升等级 2, 阈值在 2.3 V 左右 10: BOR 上升等级 3, 阈值在 2.6 V 左右 11: BOR 上升等级 4, 阈值在 2.9 V 左右 [0]: BORST_EN 0: 电压波动复位失能, 上电复位由POR/PDR 级别定义 1: 电压波动复位使能, 阈值参考 BORR_TH 和 BORF_TH
0x1fff 7802	OB_USER[15:8]	0xFF	[7]: 保留 [6]: SRAM_ECCEN 0: SRAM ECC使能 1: SRAM ECC失能 [5]: HXTAL_REMAP 0: 重映射使能 1: 重映射失能

地址	名称	出厂值	说明
			[4]: 保留 [3]: nWWDG_HW 0: 硬件窗口看门狗 1: 软件窗口看门狗 [2:1]: 保留 [0]: nFWDG_HW 0: 硬件独立看门狗 1: 软件独立看门狗
0x1fff 7803	OB_USER[23:16]	0xFF	[7:5]: 保留 [4:3]: NRST_MDSEL 00: 保留 01: NRST 引脚上的低电平可以复位系统，内部复位不能驱动 NRST 引脚 10: NRST 引脚功能与普通 GPIO 相同，只有内部复位 11: NRST 引脚配置为复位输入/输出模式 [2]: nBOOT0 0: BOOT0 为 1 1: BOOT0 为 0 [1]: nBOOT1 0: BOOT1 为 1 1: BOOT1 为 0 它与 BOOT0 引脚共同决定 boot 模式 [0]: SWBT0 0: BOOT0 取决于 PA14/BOOT0 引脚 1: BOOT0 取决于 nBOOT0 选项位
0x1fff 7804	OB_SPC_N	0x5A	OB_SPC 补码字节
0x1fff 7805	OB_USER_N[7:0]	0x01	OB_USER 补码字节 7 到 0 位
0x1fff 7806	OB_USER_N[15:8]	0x00	OB_USER 补码字节 15 到 8 位
0x1fff 7807	OB_USER_N[23:16]	0x00	OB_USER 补码字节 23 到 16 位
0x1fff 7808	DCRP0_SADDR	0xFF	[6:0]: DCRP 区域 0 起始地址
0x1fff 780c	DCRP0_SADDR_N	0x00	DCRP0_SADDR 补码字节 6 到 0 位
0x1fff 7810	DCRP0_EADDR	0x00	[6:0]: DCRP 区域 0 结束地址
0x1fff 7813	DCRP_EREN	0xFF	[7]: DCRP_EREN 0: 当 SPC 等级从 1 降低到 0 时，DCRP 不会被擦除 1: 当 SPC 等级从 1 降低到 0 时，DCRP 会被擦除 [6:0]: 保留

地址	名称	出厂值	说明
0x1fff 7814	DCRP0_EADDR_N	0xFF	DCRP0_EADDR 补码字节 6 到 0 位
0x1fff 7817	DCRP_EREN_N	0x00	DCRP_EREN 补码字节 7 位
0x1fff 7818	WP0_SADDR	0xFF	[5:0]: WP0_SADDR 表示 WP 区域 0 的第一页
0x1fff 781a	WP0_EADDR	0x00	[5:0]: WP0_EADDR 表示 WP 区域 0 的最后一页
0x1fff 781c	WP0_SADDR_N	0x00	WP0_SADDR 补码字节 5 到 0 位
0x1fff 781e	WP0_EADDR_N	0xFF	WP0_EADDR 补码字节 5 到 0 位
0x1fff 7820	WP1_SADDR	0xFF	[5:0]: WP1_SADDR 表示 WP 区域 1 的第一页
0x1fff 7822	WP1_EADDR	0x00	[5:0]: WP1_EADDR 表示 WP 区域 1 的最后一页
0x1fff 7824	WP1_SADDR_N	0x00	WP1_SADDR 补码字节 5 到 0 位
0x1fff 7826	WP1_EADDR_N	0xFF	WP1_EADDR 补码字节 5 到 0 位
0x1fff 7828	DCRP1_SADDR	0xFF	[6:0]: DCRP 区域 1 起始地址
0x1fff 782c	DCRP1_SADDR_N	0x00	DCRP1_SADDR 补码字节 6 到 0 位
0x1fff 7830	DCRP1_EADDR	0x00	[6:0]: DCRP 区域 1 结束地址
0x1fff 7834	DCRP1_EADDR_N	0xFF	DCRP1_EADDR 补码字节 6 到 0 位
0x1fff 7870	SCR_PAGE_CNT	0x00	[6:0]: 配置安全用户区域的页数
0x1fff 7872	BOOTLK	0x00	[0]: 该位置1后强制从用户闪存区启动 0: 支持闪存, RAM和系统启动 1: 只能从主闪存启动
0x1fff 7874	SCR_PAGE_CNT_N	0xFF	SCR_PAGE_CNT 补码字节 6 到 0 位
0x1fff 7876	BOOTLK_N	0xFF	BOOTLK 补码字节位 0

选项字节编程

编程操作流程如以下步骤所示:

- 确保FMC_CTL寄存器不处于锁定状态。
- 检查FMC_STAT寄存器的BUSY位来判定闪存是否正处于擦写访问状态, 若BUSY位为1, 则需等待该操作结束, BUSY位变为0。
- 将正确的序列写入FMC_OBKEY寄存器来解锁FMC_CTL寄存器中的OBLK位。
- 等待FMC_CTL寄存器的OBLK位清0。
- 在选项字节寄存器中写选项字节值到对应的寄存器地址。
- 置位FMC_CTL寄存器中的OBSTART位来发送选项字节编程命令。
- 检查FMC_STAT寄存器中BUSY位的值, 若BUSY位为1, 则需等待该操作结束。
- 启动系统上电/下电复位(或从待机模式退出)或设置FMC_CTL寄存器中的OBRD位来加载到选项字节。

注意:

1.一旦执行了一个选项字节的修改，首先用户选项字节将自动擦除。当操作执行成功时，FMC_STAT 寄存器中的 ENDF 被置位，如果 FMC_CTL 寄存器中的 ENDIE 位被置位，FMC 将触发中断。

2.Bits 63:32 是 Bits 31:0 的补码。每个选项字节位在相应的补码中都有它的补位。只有当选项字节与其补码匹配时，才可以将选项字节写入相应的寄存器。

3. 如果选项字节与它的补码不匹配，则 FMC_STAT 寄存器中的 OBERR 位将置位。选项字节被强制转换为以下值：

所有的 OB_USER 字节都强制置为 0xFF，除了 BORST_EN 为 0（电压波动复位失能）。WP 页的状态为“无保护”，DCRP 的状态是“所有区域受保护”，BOOTLK 为 1（仅从主闪存启动）。

2.3.11. 仅执行区域（DCRP）

在主闪存块中，FMC 可以定义仅可执行区域，只允许来自系统的指令事务，而不允许数据访问。DCRP 区域具有 512 字节的子页粒度。

注意：当使用仅可执行区域功能时，用户需要相应地使用仅执行选项去编译其原生代码。

DCRP 区域 x ($x = 0, 1$) 由起始地址偏移量和结束地址偏移量来定义：

- 从基地址+ [DCRP_x_SADDR x 0x200]（包含）到基地址 +地址[（DCRP_x_EADDR+ 1）x 0x200]（不包含）。最小 DCRP 区域大小为两个 DCRP 子页 2×512 字节。

例如，通过 DCRP 保护从地址 0x0800 0A00（包含）到地址 0x0800 17FF（包含），选项字节必须设置如下：

- FMC_DCRP_x_SADDR = 0x05
- FMC_DCRP_x_EADDR = 0x0B

在此区域执行代码时，将忽略调试事件。只有 CPU 可以访问 DCRP 区域，只使用指令读取任务。在所有其他情况下，访问 DCRP 区域都是非法的。例如，读操作将返回 0 并置位 FMC_STAT 寄存器中的 RPERR 标志，而写操作将被忽略并置位 FMC_STAT 寄存器中的 WPERR 标志。

DCRP 区域受擦除保护，无法擦除该区域内的页。如果设置了有效的 DCRP 区域，则除非执行 SPC 保护等级低到无保护状态的降级擦除，否则无法执行整片擦除。

只有 CPU 可以修改 DCRP 区域定义位和 DCRP_EREN 位。如果 DCRP 区域有效，在 SPC 等级低到无保护降级期间，DCRP_EREN 位置 0，则 DCRP 区域不被擦除，否则该区域将被擦除。

2.3.12. 页擦除/编程保护（WP）

FMC 提供页擦除/编程保护功能以防止对闪存进行的误操作。在受保护的页上，FMC 不会接受页擦除或编程操作。如果将页擦除或编程命令发送到 FMC 受保护的页上，那么 FMC_STAT 寄存器中的 WPERR 位将被 FMC 置位。WP 区域由起始地址偏移量和结束地址偏移量定义。页保护功能可以通过配置 WP 地址寄存器单独启用：FMC_WP_x ($x = 0, 1$)。

可以在主闪存中以 1 Kbytes 的粒度定义两个 WP 区域。WP 区域定义如下：

- 从基地址 + [(WPx_SADDR) x 0x400] (包含) 到基地址 + [(WPx_EADDR + 1) x 0x400] (不包含)。

例如, 通过 WP 从地址 0x0800 2000 (包含) 保护到地址 0x0800 2FFF (包含), 选项字节设置如下:

- WPx_SADDR = 0x08
- WPx_EADDR = 0x0B

擦除/编程保护页不能被擦除或编程。因此, 如果页是受擦除/编程保护的, 则无法进行大规模擦除。

如果安全保护 (SPC) 级别设置为高, 则 WP 区域不能修改, 否则 WP 区域可以无限制地修改

注意: DCRP 或安全用户区是擦除/编程保护的。

2.3.13. 安全保护 (SPC)

FMC 提供了一个安全保护功能来阻止非法的指令/数据读取闪存。此功能可以很好地保护软件和固件免受非法的用户操作。

安全保护等级划分为以下三种:

未保护状态: 当将 OB_SPC 字节及其补字节被设置为 0xA55A, 系统复位以后, 闪存将处于非安全保护状态。主存储块和选项字节可以被所有操作模式访问。

低等级保护: 当设置 OB_SPC 字节值为任何除 0xA5 或 0xCC 外的值, 系统复位以后, 低安全保护状态生效。注意, 如果在调试器保持连接到 JTAG/SWD 设备时执行了 SPC 修改, 则应该进行电源重置, 而不是系统重置。在低等级保护下, 主闪存只能通过用户代码访问。在调试模式下, 从 SRAM 或引导加载程序模式启动, 禁止对主闪存进行所有操作。如果在调试模式、从 SRAM 启动或引导加载程序模式下对主闪存进行读操作, 则会产生总线错误。如果在调试模式、从 SRAM 引导或从引导加载程序引导模式下对主闪存进行编程/擦除操作, 则 FMC_STAT 寄存器中的 WPERR 位将被置 1。但这些模式下都可以对选项字节进行操作, 从而可以通过该方式失能安全保护功能。通过将 OB_SPC 字节及其补码值设置为 0xA55A, 返回到无保护级别, 然后自动触发一次主闪存和数据闪存的擦除操作。注意如果配置了低级别保护且没有定义 DCRP 区域, 则必须设置 DCRP_EREN 位。

高等级保护: 将 OB_SPC 字节及其补字节设置为 0xCC33 时, 激活高等级安全保护。当选择该保护等级时, 调试模式, 从 SRAM 中启动, 或者从 boot loader 启动都被禁止。主闪存块可由用户代码的所有操作进行访问。用户选项位只能读取, 但不能修改。也可以进入其他安全区域。OB_SPC 字节及其补字节禁止再次编程。所以, 如果高等级保护被激活, 将不能再降回到低等级保护或未保护状态。

2.3.14. 安全用户区域 (SCR)

在主闪存块中, FMC 可以定义安全用户区域, 这些安全用户区域只能在引导时执行一次, 除非发生复位, 否则不会再次执行。

安全用户区域可以将安全代码与应用程序非安全代码隔离开来。安全用户区域可用于保护自定义

义安全引导库、固件更新代码或第三方安全库。当使能了安全区域(FMC_CTL 寄存器中的 SCR 置位)时,对 SCR 区域的擦除操作将设置 FMC_STAT 寄存器中的 WPERR 位,写操作将在设置 WPERR 位后立即触发硬故障异常,对 SCR 区域的读操作将产生一个总线故障而不设置 RDERR 位。

安全用户区域的大小由 FMC_SCR 寄存器的 SCR_PAGE_CNT[6:0]位定义。只能在 SPC 级为无保护下修改。安全用户区域的内容在从 OB_SPC 低保护等级更改为无保护等级时被擦除,即使它与 DCRP 页重叠。

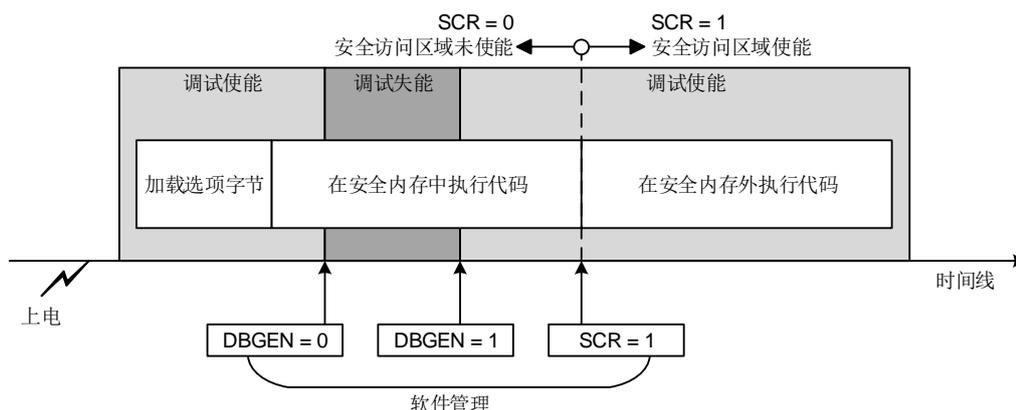
安全用户区域定义如下:

- 从基地址 (0x0800 0000) 到基地址+ [SCR_PAGE_CNT x 0x400] (不包含)。

2.3.15. 禁用内核调试访问

FMC提供了一种临时禁用对内核调试访问的方法,以执行敏感代码或操作安全内存区域中的敏感数据。[图2-1.禁用内核调试访问示例](#)给出了管理FMC_WS寄存器的DBGEN位和管理FMC_CTL寄存器的SCR位的示例。

图 2-1.禁用内核调试访问示例



2.3.16. 强制从闪存启动

FMC_SCR 寄存器中的 BOOTLK 位可以配置强制系统从主闪存引导启动。该位只能在以下情况下复位:

1. SPC 为无保护等级。
2. SPC 为低保护级别,发出无保护等级修改请求后执行了全片擦除。

2.3.17. FMC 中断

表 2-1. FMC 中断请求

中断事件	标志	清除条件	中断使能位
操作结束	ENDF	写 1 到 ENDF	ENDIE
操作失败错误	OPRERR	写 1 到 OPRERR	ERRIE

中断事件	标志	清除条件	中断使能位
读保护错误	RPERR	写 1 到 RPERR	RPERRIE
写保护错误	WPERR	写 1 到 WPERR	N/A
编程大小不匹配错误	PGMERR	写 1 到 PGMERR	N/A
编程错误	PGERR	写 1 到 PGERR	N/A
编程对齐错误	PGAERR	写 1 到 PGAERR	N/A
编程顺序错误	PGSERR	写 1 到 PGSERR	N/A
快速编程错误	FSTPERR	写 1 到 FSTPERR	N/A

2.4. FMC 寄存器

FMC基地址：0x4002 2000

2.4.1. 等待状态寄存器（FMC_WS）

地址偏移：0x00

复位值：0x000X 0200 (X值为0b010x)，如果芯片从Bootloade启动，WSCNT会被配置为0b001

该寄存器只能按（32位）访问。



位/位域	名称	描述
31:19	保留	必须保持复位值
18	DBGEN	该位用于软件启用/禁用调试 0: 禁用调试。 1: 使能调试。
17	保留	必须保持复位值
16	MFPE	主闪存已编程或空标志 该位用来表示主闪存的第一个位置是否已编程或者为空。 0: 主闪存已编程 1: 主闪存为空 该位可以通过软件设置和清除
15:12	保留	必须保持复位值
11	ICRST	复位指令缓存区。该位仅在ICEN位置0时可写。 0: 无效果。 1: 复位指令缓存区。
10	保留	必须保持复位值
9	ICEN	指令缓存区使能位 0: 失能指令缓存区。 1: 使能指令缓存区。
8	PFEN	预取功能使能位 0: 失能预取功能。 1: 使能预取功能。

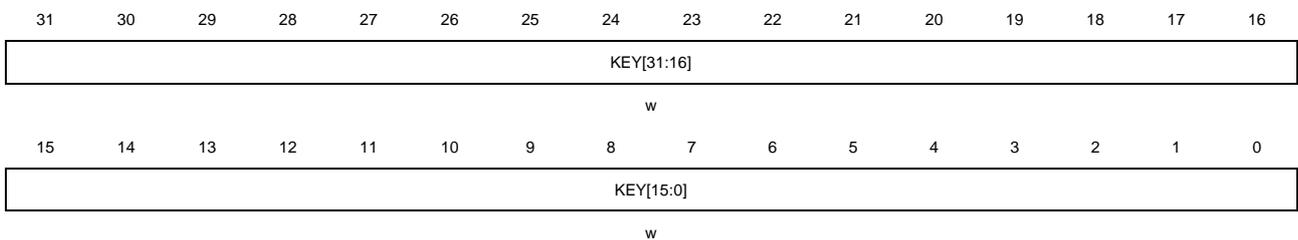
7:3	保留	必须保持复位值
2:0	WSCNT[2:0]	等待状态计数器 软件置 1 和清 0。 000: 不增加等待状态 001: 增加 1 个等待状态 1 其它: 保留

2.4.2. 解锁寄存器 (FMC_KEY)

地址偏移: 0x08

复位值: 0x0000 0000

该寄存器只能按 (32位) 访问。



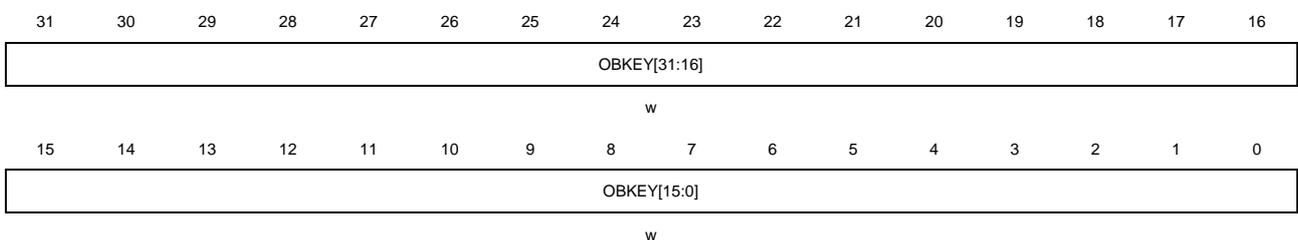
位/位域	名称	描述
31:0	KEY[31:0]	FMC_CTL 解锁寄存器 这些位仅能被软件写。 写连续的解锁值到 KEY[31:0]可以解锁 FMC_CTL 寄存器。

2.4.3. 选项字节操作解锁寄存器 (FMC_OBKEY)

地址偏移: 0x0C

复位值: 0x0000 0000

该寄存器只能按 (32位) 访问。



位/位域	名称	描述
31:0	OBKEY[31:0]	FMC_OBCTL 选项字节操作解锁寄存器 这些位仅能被软件写 写连续的解锁值到 OBKEY[31:0]解锁 FMC_OBCTL 寄存器。

2.4.4. 状态寄存器 (FMC_STAT)

地址偏移: 0x10

复位值: 0x0000 0000

该寄存器只能按 (32位) 访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
保留															BUSY	
															r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
OBERR	RPERR	保留				FSTPERR	保留	PGSERR	PGMERR	PGAERR	WPERR	PGERR	保留	OPRERR	ENDF	
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
31:17	保留	必须保持复位值
16	BUSY	闪存忙标志 当闪存操作正在进行时, 此位被置 1。当操作结束或者出错, 此位被清 0。
15	OBERR	选项字节读取错误标志位 当选项字节与其补充字节不匹配时, 该位由硬件置位, 并且该选项字节强制设置为 0xFF。
14	RPERR	读保护错误标志位 当访问的地址被 DCRP 或 SCR 保护时, 该位由硬件置 1。这个位可以通过写入 1 来清除。 0: 未发生读保护错误。 1: 发生读保护错误。
13:10	保留	必须保持复位值
9	FSTPERR	快速编程错误标志位。 当快速编程序列由于错误 (对齐、大小、写保护或数据丢失) 而中断时, 该位由硬件置 1。同时设置相应的错误位 (PGAERR、PGMERR 或 WPERR)。该位可以通过写入 1 来清除。
8	保留	必须保持复位值
7	PGSERR	编程顺序错误标志位 当编程操作没有将 PG 位预先置 1 就编程时, 该位将由硬件置位。由于先前的编程错误导致 PGERR, PGMERR, PGAERR 或 WPERR 置 1 时, 该位也置 1。 该位可以由软件写 1 清 0。 0: 未发生编程顺序错误。 1: 发生编程顺序错误。
6	PGMERR	编程大小不匹配错误标志位。 当编程的大小为半字/字访问时, 该位将由硬件置位。唯一正确的编程大小为双字。 该位可以由软件写 1 清 0。

5	PGAERR	编程对齐错误标志位 当写入的数据非64位对齐时，即要编程的数据不能包含在同一64位闪存行中，该位将由硬件置位。 该位可以由软件写1清0。
4	WPERR	擦除/编程保护错误标志位 当擦除/编程保护错误发生时，该位由硬件置位，软件写1清0。 0：未发生擦除/编程保护错误。 1：发生擦除/编程保护错误。
3	PGERR	编程错误标志位 当要编程的闪存双字地址的数据不是 0xFFFF FFFF FFFF FFFF 时，该位由硬件置1，软件写1清0。
2	保留	必须保持复位值
1	OPRERR	操作失败错误标志位 当闪存编程或擦除操作已完成且不成功时，并且使能了错误中断（ERRIE = 1），则该位将由硬件置1。 该位可以由软件写1清0。
0	ENDF	操作结束标志位 当闪存编程或擦除操作已完成并成功时，此位将被硬件置1。 该位可以由软件写1清0。

2.4.5. 控制寄存器（FMC_CTL）

地址偏移：0x14

复位值：0xC000 0000

该寄存器可以按字节（8位）、半字（16位）和字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
LK	OBLK	保留	SCR	OBRLD	RPERRIE	ERRIE	ENDIE	保留						FSTPG	OBSTART	START
rs	rs		rs	rc_w1	nw	nw	nw							rs	rs	rs
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
保留							PN[5:0]					MER	PER	PG		
							rw					rw		rw		

位/位域	名称	描述
31	LK	FMC_CTL 寄存器锁定标志位 当正确的序列写入到 FMC_KEY 寄存器时，该位将由硬件清0。该位可以由软件置1。
30	OBLK	FMC_OBCTL 寄存器锁定标志位 如果该位被置1，FMC_OBCTL 寄存器中关于用户选项字节的所有位以及选项字节页都将被锁定。当正确的序列被写入到 FMC_OBKEY 寄存器时，该位被硬件清0。该位可由软件置1。

29	保留	必须保持复位值
28	SCR	安全用户区域使能位 该位置1时锁定安全用户区域。当有安全用户区域时可以由软件置1，且该位只能写一次。 0：失能安全用户区域。 1：使能安全用户区域。 该位只能通过软件置1，并通过系统复位清除。
27	OBRLD	选项字节重加载位 该位可以由软件置1。 0：无影响。 1：强制选项字节重加载。 注意： 如果当 OBLK 置位时，该位无法写入。
26	RPERRIE	读保护错误中断使能位 仅当 LK 被设置为 0 时，软件才能设置或清除该位。 0：失能读保护错误中断。 1：使能读保护错误中断。
25	ERRIE	操作错误中断使能位 该位可以由软件置 1 或清 0。 0：失能操作失败错误中断。 1：使能操作失败错误中断。
24	ENDIE	操作结束中断使能位 该位可以由软件置 1 或清 0。 0：失能操作结束中断。 1：使能操作结束中断。
23:19	保留	必须保持复位值
18	FSTPG	主闪存快速编程命令位 该位可以由软件置 1 或清 0。 0：失能快速编程 1：使能快速编程
17	OBSTART	发送选项字节更改命令位 该位由软件置 1，只有当 OBLK 被设置为 0 时，才向 FMC 发送选项字节更改命令。 当 BUSY 位被清除时，该位被硬件清除。
16	START	向 FMC 发送擦除命令位 该位用于向 FMC 发送擦除命令。当 BUSY 位被清除时，该位被硬件清除。
15:9	保留	必须保持复位值
8:3	PN[5:0]	擦除页码选择位 这些位用于选择要擦除的页码： 000000：第0页。 000001：第1页。

...
111111: 第 63 页。

- 2 MER 主闪存批量擦除指令位
 该位可以由软件置1或清0
 0: 无作用。
 1: 主闪存批量擦除指令。

- 1 PER 主闪存页擦除指令位
 该位可以由软件置 1 或清 0
 0: 无作用。
 1: 主闪存页擦除指令。

- 0 PG 主闪存编程命令位
 该位可以由软件置 1 或清 0
 0: 失能主存储块编程命令。
 1: 使能主存储块编程命令。

2.4.6. 选项字节控制寄存器 (FMC_OBCTL)

地址偏移: 0x20

复位值: 0xXXXX XXXX。

当FMC_CTL寄存器中的OBRLD位置位或系统重置时, 从闪存中加载对应值到该寄存器。

该寄存器可以按字节(8位)、半字(16位)和字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留			NRST_MDSEL[1:0]		nBOOT0	nBOOT1	SWBT0	保留	SRAM_EC CEN	HXTAL_R EMAP	保留	nWWDG_ HW	保留	nFWDG_H W	
			rw		rw	rw	rw		rw	rw		rw		rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	nRST_ST DBY	nRST_DP SLP	BORF_TH[1:0]		BORR_TH[1:0]		BORST_E N	SPC[7:0]							
	rw	rw	rw		rw		rw	rw							

位/位域	名称	描述
31:29	保留	必须保持复位值
28:27	NRST_MDSEL[1:0]	NRST引脚模式选择位 00: 保留 01: NRST引脚上的低电平可以复位系统, 内部复位不能驱动NRST引脚。 10: NRST引脚功能与普通GPIO相同, 只有内部复位。 11: NRST引脚配置为复位输入/输出模式。
26	nBOOT0	BOOT0选择位 0: BOOT0为1。 1: BOOT0为0。

25	nBOOT1	<p>Boot1选择位</p> <p>0: BOOT1为1。</p> <p>1: BOOT1为0。</p> <p>与BOOT0共同决定启动模式。</p>
24	SWBT0	<p>软件BOOT0选择位</p> <p>0: BOOT0取决于PA14/BOOT0引脚。</p> <p>1: BOOT0取决于选项字节位nBOOT0。</p>
23	保留	必须保持复位值
22	SRAM_ECCEN	<p>SRAM ECC失能位</p> <p>0: 使能SRAM ECC。</p> <p>1: 失能SRAM ECC。</p>
21	HXTAL_REMAP	<p>HXTAL重映射</p> <p>0: 使能重映射。</p> <p>1: 失能重映射。</p> <p>当该位被设置为0时，HXTAL时钟源将从PF0-OSC_IN / PF1-OSC_OUT引脚重新映射到PC14-OSCX_IN/ PC15-OSCX_OUT。因此，PC14-OSCX_IN / PC15-OSCX_OUT被LXTAL和HXTAL共享，两个时钟源不能同时使用</p>
20	保留	必须保持复位值
19	nWWDG_HW	<p>窗口看门狗配置位</p> <p>0: 硬件窗口看门狗。</p> <p>1: 软件窗口看门狗。</p>
18:17	保留	必须保持复位值
16	nFWDG_HW	<p>独立看门狗配置位</p> <p>0: 硬件独立看门狗。</p> <p>1: 软件独立看门狗。</p>
15	保留	必须保持复位值
14	nRST_STDBY	<p>选项字节待机复位选择位</p> <p>0: 进入待机模式时产生复位</p> <p>1: 进入待机模式时不产生复位</p>
13	nRST_DPSLP	<p>选项字节深度睡眠复位选择位</p> <p>0: 进入深度睡眠模式时产生复位</p> <p>1: 进入深度睡眠模式时不产生复位</p>
12:11	BORF_TH[1:0]	<p>VDD供电下降时的BOR阈值</p> <p>VDD下降超过这个阈值会激活复位信号。</p> <p>00: BOR 下降等级 1, 阈值在 2.0 V 左右</p> <p>01: BOR 下降等级 2, 阈值在 2.2 V 左右</p> <p>10: BOR 下降等级 3, 阈值在 2.5 V 左右</p> <p>11: BOR 下降等级 4, 阈值在 2.8 V 左右</p>

注：BORR_TH 低于 BORF_TH 的配置组合都属于非法配置。非法配置情况下，BORR_TH 和 BORF_TH 被强制设置为默认值 0b11

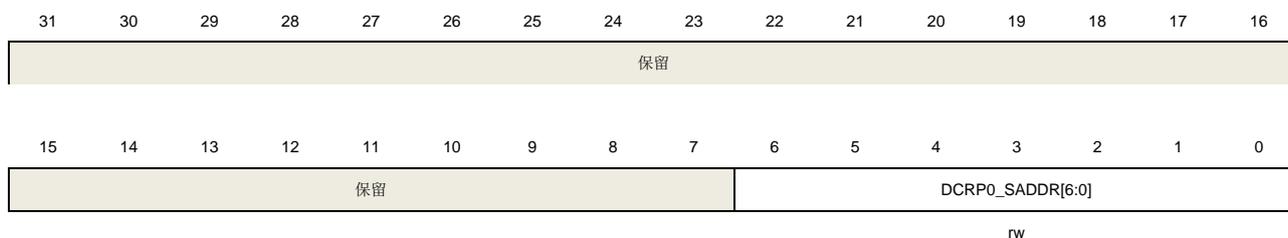
10:9	BORR_TH[1:0]	VDD供电上升时的BOR阈值 VDD上升超过这个阈值会激活复位信号。 00: BOR 上升等级 1, 阈值在 2.1 V 左右 01: BOR 上升等级 2, 阈值在 2.3 V 左右 10: BOR 上升等级 3, 阈值在 2.6 V 左右 11: BOR 上升等级 4, 阈值在 2.9 V 左右 注：BORR_TH 低于 BORF_TH 的配置组合都属于非法配置。非法配置情况下，BORR_TH 和 BORF_TH 被强制设置为默认值 0b11
8	BORST_EN	电压波动复位使能位 0: 电压波动复位失能，上电复位由POR/PDR级别定义 1: 电压波动复位使能，阈值参考BORR_TH和BORF_TH
7:0	SPC[7:0]	安全保护等级选项字节状态位 0xA5: 无安全保护。 0xCC: 高安全保护等级，芯片读保护启用。 除0xA5或0xCC之外任何值：低安全保护等级，存储区保护启用。

2.4.7. DCRP0 起始地址寄存器 (FMC_DCRP_SADDR0)

地址偏移：0x24

复位值：0x0000 00XX

该寄存器只能按（32位）访问。



位/位域	名称	描述
31:7	保留	必须保持复位值
6:0	DCRP0_SADDR[6:0]	DCRP0 起始地址偏移 DCRP0_SADDR 包含了 DCRP0 的第一个子页的地址偏移量。

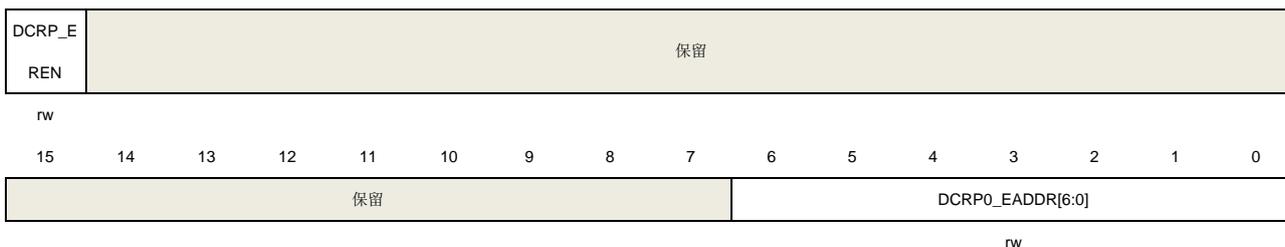
2.4.8. DCRP0 结束地址寄存器 (FMC_DCRP_EADDR0)

地址偏移：0x28

复位值：0xX000 00XX

该寄存器只能按（32位）访问。





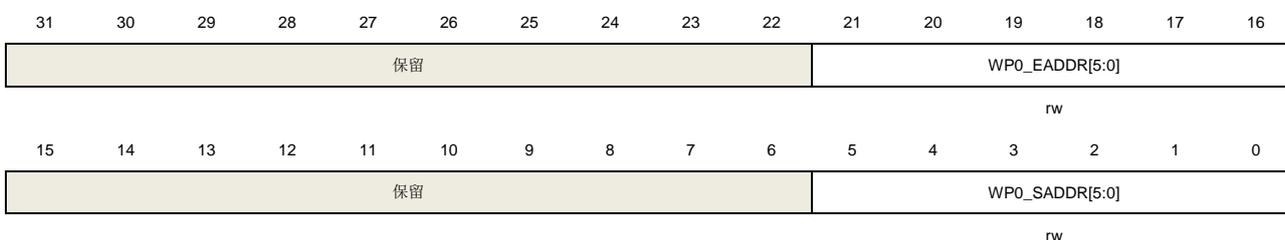
位/位域	名称	描述
31	DCRP_EREN	DCRP 区域擦除使能位。 0: 当FMC_OBCTL寄存器中SPC的等级从低保护等级降到无保护等级时, DCRP区域不擦除。 1: 当FMC_OBCTL 寄存器中 SPC 的等级从低保护等级降到无保护等级时, DCRP区域擦除。
30:7	保留	必须保持复位值
6:0	DCRP0_EADDR[6:0]	DCRP0 结束地址偏移 DCRP0_EADDR包含了DCRP0的最后一个子页的地址偏移量。

2.4.9. 擦除/编程保护区域 0 寄存器 (FMC_WP0)

地址偏移: 0x2C

复位值: 0x00XX 00XX

该寄存器只能按 (32位) 访问。



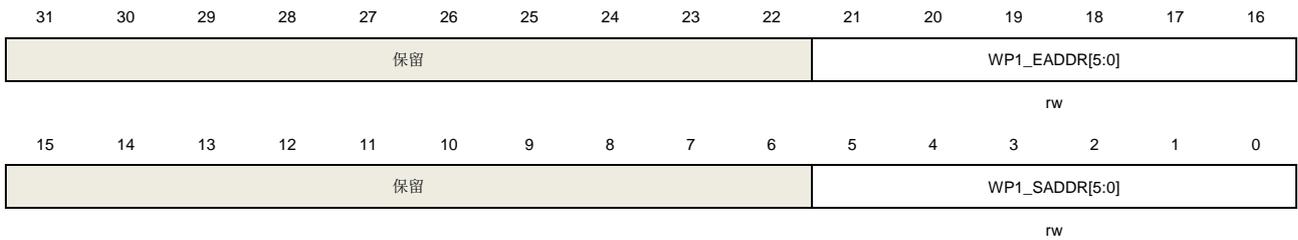
位/位域	名称	描述
31:22	保留	必须保持复位值
21:16	WP0_EADDR[5:0]	擦除/编程保护区域 0 结束地址偏移 WP0_EADDR 包含擦除/编程保护区域 0 的最后一页
15:6	保留	必须保持复位值
5:0	WP0_SADDR[5:0]	擦除/编程保护区域0起始地址偏移 WP0_SADDR包含擦除/编程保护区域0的第一页

2.4.10. 擦除/编程保护区域 1 寄存器 (FMC_WP1)

地址偏移: 0x30

复位值: 0x00XX 00XX

该寄存器只能按（32位）访问。



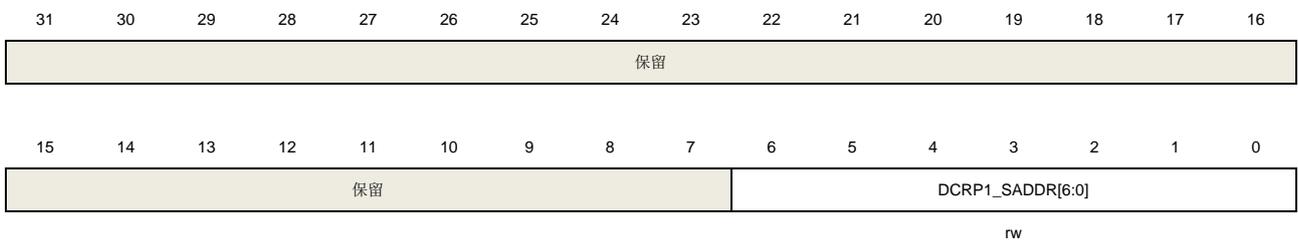
位/位域	名称	描述
31:22	保留	必须保持复位值
21:16	WP1_EADDR[5:0]	擦除/编程保护区域1结束地址偏移 WP1_EADDR包含擦除/编程保护区域1的最后一页
15:6	保留	必须保持复位值
5:0	WP1_SADDR[5:0]	擦除/编程保护区域1起始地址偏移 WP1_SADDR包含擦除/编程保护区域1的第一页

2.4.11. DCRP1 起始地址寄存器（FMC_DCRP_SADDR1）

地址偏移：0x34

复位值：0x0000 00XX

该寄存器只能按（32位）访问。



位/位域	名称	描述
31:7	保留	必须保持复位值
6:0	DCRP1_SADDR[6:0]	DCRP1 起始地址偏移 DCRP1_SADDR包含了DCRP1的第一个子页的地址偏移量。

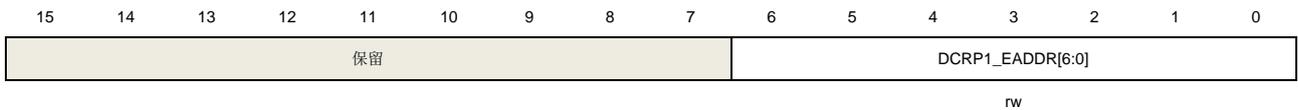
2.4.12. DCRP1 结束地址寄存器（FMC_DCRP_EADDR1）

地址偏移：0x38

复位值：0x0000 00XX

该寄存器只能按（32位）访问。





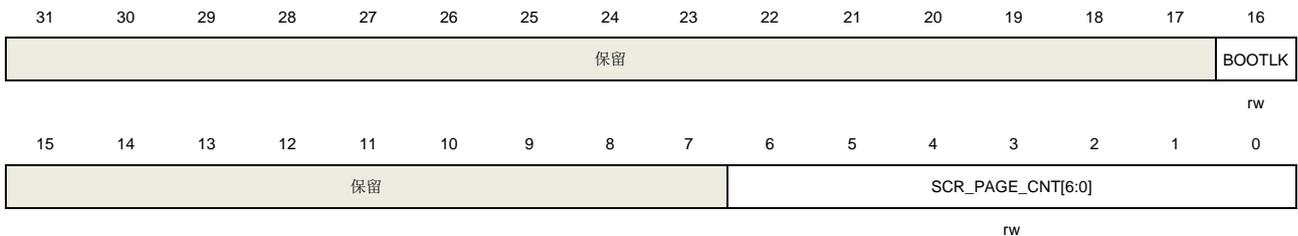
位/位域	名称	描述
31:7	保留	必须保持复位值
6:0	DCRP1_EADDR[6:0]	DCRP1 结束地址偏移 DCRP1_EADDR包含了DCRP1的最后一个子页的地址偏移量。

2.4.13. 安全用户区域寄存器 (FMC_SCR)

地址偏移: 0x80

复位值: 0x000X 00XX

该寄存器只能按 (32位) 访问。



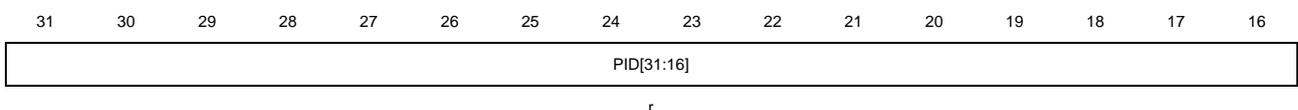
位/位域	名称	描述
31:17	保留	必须保持复位值
16	BOOTLK	该位置1时强制从用户闪存启动 0: 支持闪存, RAM和系统启动。 1: 只能从主闪存启动。
15:7	保留	必须保持复位值
6:0	SCR_PAGE_CNT[6:0]	配置安全用户区域的页数 安全用户区域从基地址0x0800 0000开始。大小为SCR_PAGE_CNT乘以页大小。 注意: 该区域仅当OB_SPC等级为无保护时才可以修改。

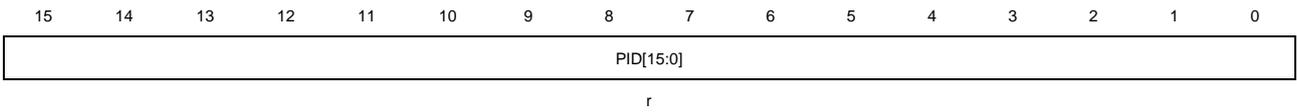
2.4.14. 产品 ID 寄存器 (FMC_PID)

地址偏移: 0x120

复位值: 0XXXXX XXXX

该寄存器只能按字 (32位) 访问。





位/位域	名称	描述
31:0	PID[31:0]	<p>产品保留 ID 寄存器</p> <p>该寄存器为只读</p> <p>上电后这些位始终不会改变，该寄存器在生产过程中被一次性编程。</p>

3. 电源管理单元 (PMU)

3.1. 简介

功耗设计是GD32C2x1系列产品比较注重的的问题之一。电源管理单元提供了六种省电模式，包括运行模式1，睡眠模式，睡眠模式1，深度睡眠模式，深度睡眠模式1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。GD32C2x1系列设备有两个电源域，包括V_{DD}/V_{DDA}域，1.2V域。V_{DD}/V_{DDA}域由电源直接供电。在V_{DD}/V_{DDA}域中嵌入了一个LDO，用来为1.2V域供电。

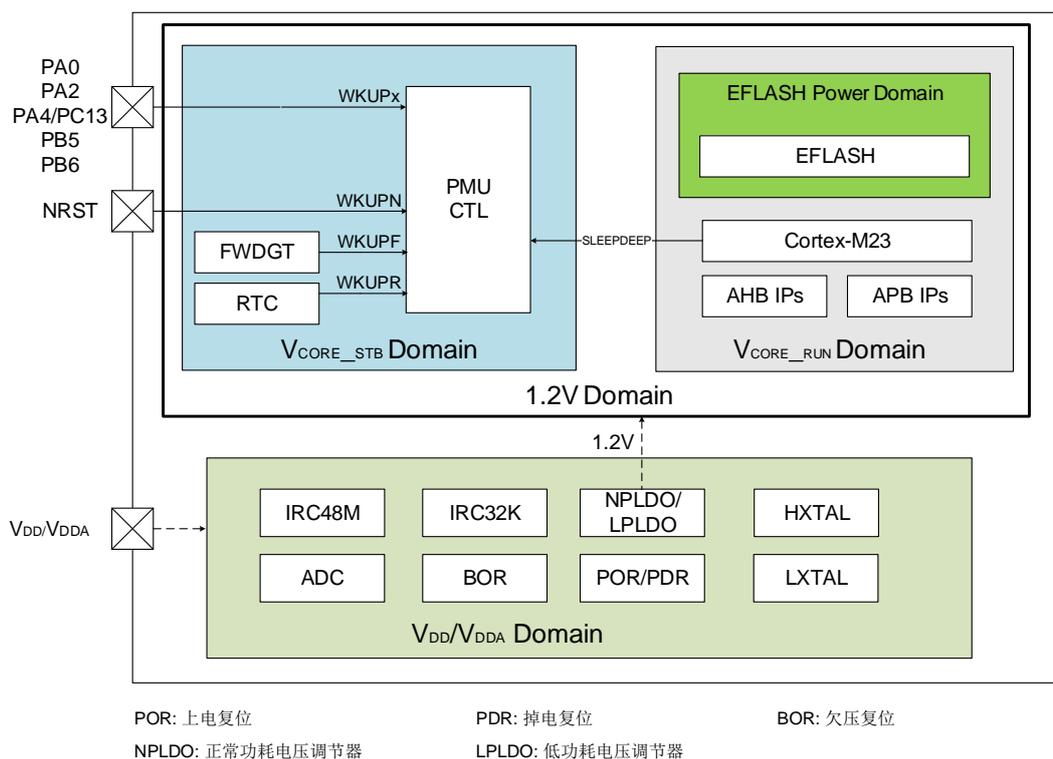
3.2. 主要特征

- 两个电源域：V_{DD}/V_{DDA}域和1.2V电源域；
- 六种省电模式：运行模式1、睡眠模式、睡眠模式1、深度睡眠模式、深度睡眠模式1和待机模式；
- 在运行模式、睡眠模式和深度睡眠模式下，内部正常功耗电压调节器（NPLDO）为1.2V电源域提供1.2V电源；
- 在运行模式1、睡眠模式1、深度睡眠模式1和待机模式下，内部低功耗电压调节器（LPLDO）为1.2V电源域提供1.2V电源；
- 在运行模式 / 运行模式1 / 深度睡眠模式 / 深度睡眠模式1下，EFLASH可单独断电；
- 提供电压检测器：POR/PDR检测器、BOR检测器。

3.3. 功能说明

[图 3-1. 电源域概览](#)提供了 PMU 及相关电源域的内部结构框图。

图 3-1. 电源域概览



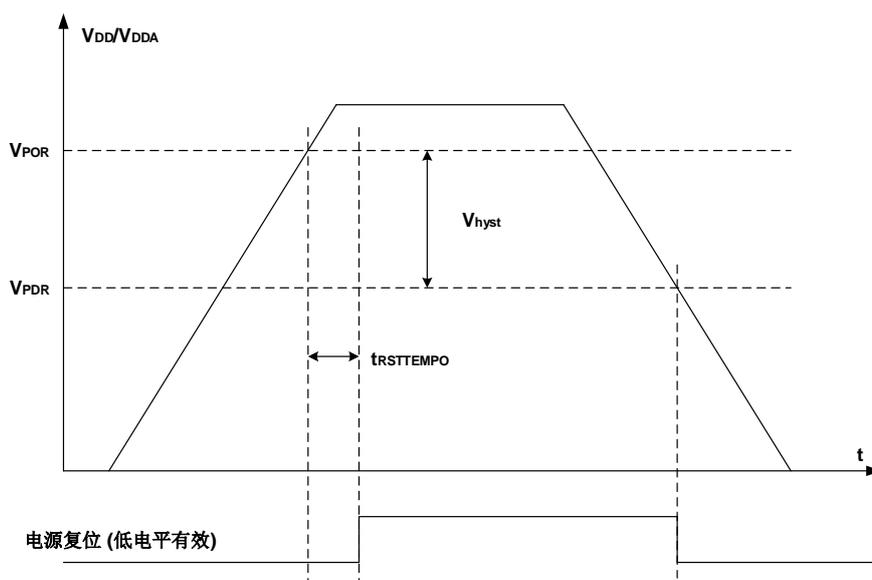
3.3.1. VDD / VDDA 电源域

V_{DD}/V_DDA 域包括 HXTAL（高速外部晶体振荡器）、LXTAL（低速外部晶体振荡器）、NPLDO/LPLDO、POR/PDR（上电 / 掉电复位）、BOR（欠压复位）、ADC（A/D 转换器）、IRC48M（内部 48M RC 振荡器）、IRC32K（内部 32KHz RC 振荡器）等等。

为 1.2V 域供电的 LDO，其复位后保持使能。可以被配置为 NPLDO 或 LPLDO 的工作状态。NPLDO 为工作模式 / 睡眠模式 / 深度睡眠模式供电，LPLDO 为运行模式 1 / 睡眠模式 1 / 深度睡眠模式 1 和待机模式供电。

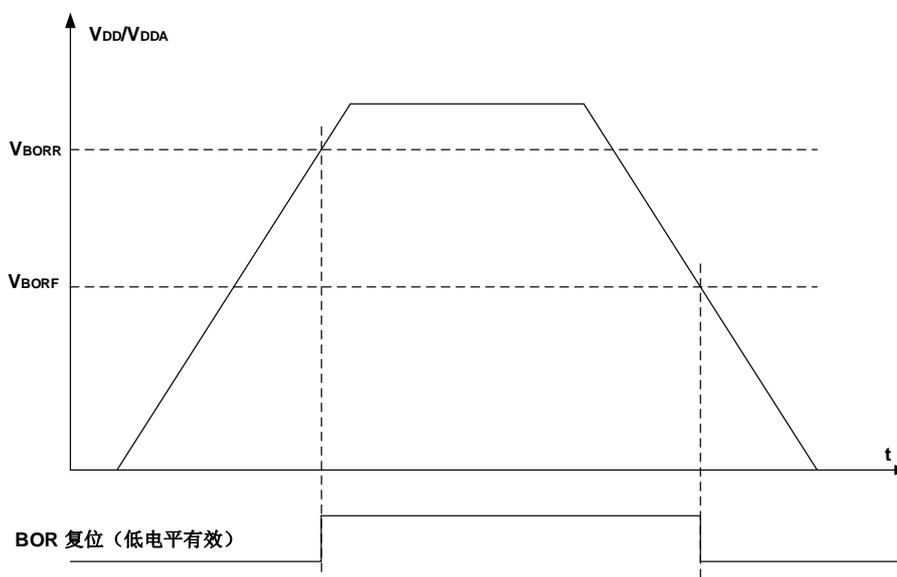
POR / PDR（上电 / 掉电复位）电路检测 V_{DD} / V_DDA 并在电压低于特定阈值时产生电源复位信号复位除 V_{CO}RE_STB 域之外的整个芯片。[图 3-2. 上电 / 掉电复位波形图](#)显示了供电电压和电源复位信号之间的关系。V_{POR} 表示上电复位的阈值电压，V_{PDR} 表示掉电复位的阈值电压，迟滞电压为 V_{hyst} 值。

图 3-2. 上电 / 掉电复位波形图



BOR 电路用于检测 V_{DD} / V_{DDA} 。在电压低于选项字节的 $BORR_TH$ 和 $BORF_TH$ 定义的阈值时，BOR 会产生电源复位信号复位除 V_{CORE_STB} 域的整个芯片。注意 POR / PDR（上电 / 掉电复位）电路总是处于检测状态。BOR 通过置位选项字节中的 $BORST_EN$ 位使能。[图 3-3. BOR 波形图](#)显示了供电电压和 BOR 复位信号之间的关系。 V_{BORR} 和 V_{BORF} 表示 BOR 复位的阈值电压，该值在选项字节 $BORR_TH$ 和 $BORF_TH$ 中定义。

图 3-3. BOR 波形图



3.3.2. 1.2V 电源域

主要功能包括 Cortex®-M23 内核逻辑、AHB/APB 外设、 V_{DD} / V_{DDA} 域的 APB 接口等。当 1.2V 电压上电后，POR 将在 1.2V 域中产生一个复位序列。复位完成后，如果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。详细

内容将在以下章节予以说明。

MCU 的数字逻辑有两个电源域。当 MCU 在运行模式/运行模式 1/睡眠模式/睡眠模式 1/深度睡眠模式/深度睡眠模式 1 时，V_{CORE_RUN} 域工作。当 MCU 在待机模式时，V_{CORE_RUN} 域掉电。当 MCU 上电后，V_{CORE_STB} 域总是上电的。

两个 LDO 能够为数据逻辑域提供 1.2V 电压。NPLDO 提供 MCU 运行在全性能模式。LPLDO 用于在低性能模式下为 MCU 供电。

EFLASH 电源域

EFLASH 能够独立掉电，系统复位后默认上电。在运行模式/运行模式 1 下，EFLASH 通过置位 PMU_CTL1 寄存器的 EFPSLEEP 位来掉电。当 MCU 进入深度睡眠模式，EFLASH 能够通过置位 PMU_CTL1 寄存器的 EFDSPSLEEP 位来切换。

当仅有 LPLDO 开而 NPLDO 关闭时，EFLASH 无法正常工作。这种情况下，代码应当保存在 SRAM 中。

3.3.3. 省电模式

系统复位/电源复位或从待机模式唤醒后，MCU 进入运行模式。所有电源域处于供电状态，NPLDO 工作在 1.2V 模式。用户可以通过减慢系统时钟（HCLK 和 PCLK）或关闭未使用的外设的时钟。此外，六种省电模式可以实现更低的功耗，它们是运行模式 1，睡眠模式，睡眠模式 1，深度睡眠模式，深度睡眠模式 1 和待机模式。

运行模式 1

在运行模式 1 下，NPLDO 关闭而 LPLDO 开启，系统时钟源必须为 IRC32K。

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。

睡眠模式 1

睡眠模式 1 对应于运行模式 1 的 Cortex®-M23 的 SLEEPING 模式。在该模式下，NPLDO 关闭而 LPLDO 开启，系统时钟源为 IRC32K。

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_RUN} 域中的所有时钟全部关闭，IRC48M 和 HXTAL 也全部被禁用。SRAM 和寄存器中的内容被保留。NPLDO 开启。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，再将 PMU_CTL0 寄存器的 LPMOD 位域配置为“00”，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）。当退出深度睡眠模式时，IRC48M 被选中作为系统时钟。

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

深度睡眠模式1

深度睡眠模式 1 与 Cortex®-M23 的 SLEEPDEEP 模式相对应。在深度睡眠模式 1 下，V_{CORE_RUN} 域中的所有时钟全部关闭，IRC48M 和 HXTAL 也全部被禁用。NPLDO 关闭而 LPLDO 开启。SRAM 和寄存器中的内容被保留。进入深度睡眠模式 1 之前，先将 Cortex®-M23 系统控制寄存器的 SLEEPDEEP 位置 1，再将 PMU_CTL0 寄存器的 LPMOD 位域配置为“01”，然后执行 WFI 或 WFE 指令即可进入深度睡眠模式 1。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 1 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 1 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）。当退出深度睡眠模式 1 时，IRC48M 被选中作为系统时钟。

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

待机模式

待机模式是基于 Cortex®-M23 的 SLEEPDEEP 模式实现的。在待机模式下，整个 V_{CORE_RUN} 域全部掉电，NPLDO 关闭，IRC48M 和 HXTAL 也会被关闭。进入待机模式前，先将 PMU_CTL0 寄存器的 LPMOD 位域配置为“11”，再清除 PMU_CS 寄存器的 WUF 位，再将 Cortex®-M23 系统控制寄存器的 SLEEPDEEP 位置 1，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 是否已进入待机模式。待机模式有四个唤醒源，包括来自 NRST 引脚的外部复位，RTC 闹钟，FWDGT 复位，LXTAL 时钟失败检测和 WKUPx 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_RUN} 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 3-1. 省电模式总结

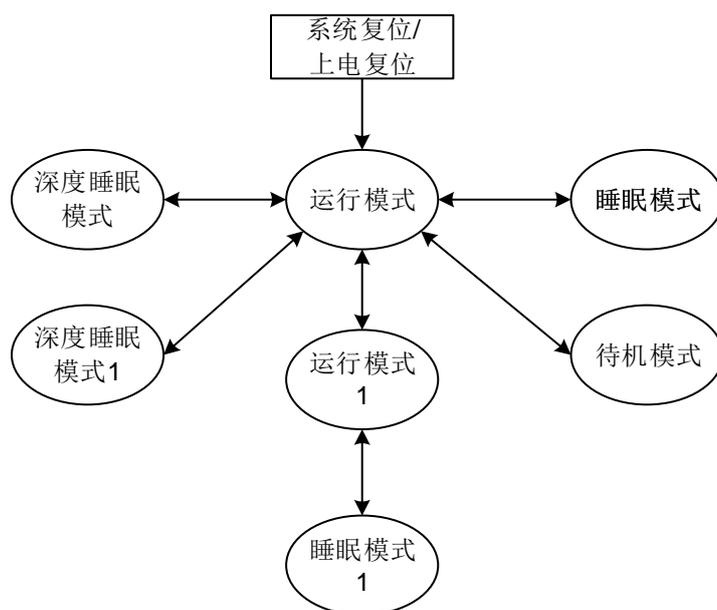
模式	描述	LDO 状态	进入指令	唤醒	唤醒后模式	唤醒延时
运行	对所有时钟无影响，全部开启	NPLDO 开启 LPLDO 开启	系统 / 上电复位 或从待机模式唤	-	-	-

模式	描述	LDO 状态	进入指令	唤醒	唤醒后模式	唤醒延时
			醒			
运行 1	系统时钟 = IRC32K	NPLDO 关闭 LPLDO 开启	置位 LPLDOEN	清除 LPLDOEN	-	NPLDO 唤醒时间+Flash 唤醒时间
睡眠	仅关闭 CPU 时钟	NPLDO 开启 LPLDO 开启	SLEEPDEEP = 0, 在运行模式下执行 WFI 或 WFE	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	运行模式	-
睡眠 1	仅关闭 CPU 时钟 系统时钟 = IRC32K	NPLDO 关闭 LPLDO 开启	SLEEPDEEP = 0, 在运行模式 1 下执行 WFI 或 WFE	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	运行模式 1	-
深度睡眠	1、关闭 V _{CORE_RUN} 域的所有时钟 2、关闭 IRC48M、HXTAL	NPLDO 开启 LPLDO 开启	SLEEPDEEP = 1, LPMOD = 00, 执行 WFI 或 WFE	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	运行模式	IRC48M 唤醒时间+Flash 唤醒时间
深度睡眠 1	1、关闭 V _{CORE_RUN} 域的所有时钟 2、关闭 IRC48M、HXTAL	NPLDO 关闭 LPLDO 开启	SLEEPDEEP = 1, LPMOD = 01, 执行 WFI 或 WFE	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	运行模式	IRC48M 唤醒时间+NPLDO 唤醒时间+Flash 唤醒时间
待机	1、V _{CORE_RUN} 域掉电 2、关闭 IRC48M、HXTAL	NPLDO 关闭 LPLDO 开启	SLEEPDEEP = 1, LPMOD = 11, 执行 WFI 或 WFE	1、NRST 引脚 2、WKUP 引脚 3、FWDGT 复位 4、RTC 闹钟	运行模式	IRC48M 唤醒时间+NPLDO 唤醒时间+Flash 唤醒时间

注意:

- 不允许从运行模式1直接进入睡眠模式/深度睡眠模式/深度睡眠模式1/待机模式。不同模式间转换如 [图3-4. 省电模式转换图](#)所示;
- 如果MCU要从运行模式进入睡眠模式/深度睡眠模式/深度睡眠模式1/待机模式, 软件应清除LPLDOEN位。

图 3-4. 省电模式转换图



- 在待机模式下，除了NRST引脚，配置为RTC功能的PC13，用作LXTAL晶振引脚的PC14和PC15，使能的WKUPx引脚，其他所有I/O都处于高阻态。
- 各模块在不同操作模式下的状态如[表3-2. 不同模式下模块状态](#)所示。

表 3-2. 不同模式下模块状态

模块	运行	运行 1	睡眠	睡眠 1	深度睡眠		深度睡眠 1		待机	
是否有唤醒能力	-	-	-	-	-	唤醒源	-	唤醒源	-	唤醒源
CPU	1	1	-	-	-	-	-	-	-	-
Flash	1	-	1	-	4	-	-	-	-	-
SRAM	1	1	1	1	3	-	3	-	-	-
Vcore 供电	1	1	1	1	1	1	1	1	-	-
POR/PDR	1	1	1	1	1	1	1	1	1	1
BOR	2	2	2	2	3	3	3	3	3	3
NRST	1	1	1	1	1	1	1	1	1	1
DMA/DMAMUX	2	2	3	3	-	-	-	-	-	-
IRC48M	1	1	3	3	-	-	-	-	-	-
HXTAL	2	2	3	3	-	-	-	-	-	-
IRC32K	2	2	3	3	3	-	3	-	3	-
LXTAL	2	2	3	3	3	-	3	-	3	-
CKM	2	2	3	3	-	-	-	-	-	-
LCKM	2	2	3	3	3	2	3	2	3	2
RTC	2	2	3	3	3	2	3	2	3	2
USART0	2	2	3	3	3	2	3	2	-	-
USART1	2	2	3	3	-	-	-	-	-	-
USART2	2	2	3	3	-	-	-	-	-	-
I2C0	2	2	3	3	3	2	3	2	-	-
I2C1	2	2	3	3	3	2	3	2	-	-

模块	运行	运行 1	睡眠	睡眠 1	深度睡眠		深度睡眠 1		待机	
是否有唤醒能力	-	-	-	-	-	唤醒源	-	唤醒源	-	唤醒源
SPI0/I2S	2	2	3	3	-	-	-	-	-	-
SPI1	2	2	3	3	-	-	-	-	-	-
ADC	2	2	3	3	-	-	-	-	-	-
内部温度传感器	2	2	3	3	-	-	-	-	-	-
TIMERx	2	2	3	3	-	-	-	-	-	-
FWDGT	2	2	3	3	3	2	3	2	3	2
WWDGT	2	2	3	3	-	-	-	-	-	-
SysTick	2	2	3	3	-	-	-	-	-	-
CRC	2	2	3	3	-	-	-	-	-	-
CMP	2	2	3	3	-	-	-	-	-	-
GPIOs	2	2	3	3	3	2	3	2	-	2 ⁽¹⁾
单个外设时钟	2	2	4 ⁽²⁾	4 ⁽²⁾	4 ⁽²⁾	-	4 ⁽²⁾	-	-	-

'-': 模块不可用或无关;

'1': 复位或从深度睡眠/深度睡眠 1/待机中唤醒后, 模块使能;

'2': 默认失能, 可由软件配置是否使能;

'3': 状态与进入低功耗模式前相同;

'4': 软件可配置在进入低功耗模式时是否自动失能/掉电。

(1). 仅 WKUPx 引脚可唤醒。

(2). 可由 RCU_AHB1SPDPEN/RCU_AHB2SPDPEN/RCU_APBSPDPE 寄存器配置。

3.4. PMU 寄存器

PMU 基地址: 0x4000 7000

3.4.1. 控制寄存器 0 (PMU_CTL0)

地址偏移: 0x00

复位值: 0x0002 9000 (从待机模式唤醒后复位)

该寄存器可以按半字 (16 位) 或字 (32 位) 访问。



位/位域	名称	描述
31:19	保留	必须保持复位值。
18	LPLDOEN	使能低功耗LDO。 该位置位时，LDO将从正常功耗LDO切换为低功耗LDO。 0: 不使用LPLDO 1: 使用LPLDO
17:14	保留	必须保持复位值。
13:12	DSMODVS[1:0]	选择深度睡眠模式下的电压 该位控制深度睡眠/深度睡眠1模式下的VCORE电压，以获得最佳的功耗和性能平衡。 00: 0.9V 01: 1.0V (默认值) 10: 1.1V 11: 1.2V 注意: 0.9V 仅在 NPLDO 关闭时有效。
11:9	保留	必须保持复位值。
8	BKPWEN	RTC 备份寄存器写使能 0: 禁止对 RTC 备份寄存器的写访问。 1: 允许对 RTC 备份寄存器的写访问。 复位之后，任何对备份寄存器的写访问都将被禁止。如需对备份寄存器做写访问，需先将该位置 1。
7:4	保留	必须保持复位值。
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。
1:0	LPMOD[1:0]	选择 Cortex®-M23 进入 SLEEPDEEP 模式，MCU 进入的低功耗模式 00: 深度睡眠模式 01: 深度睡眠模式 1 10: 保留 11: 待机模式

3.4.2. 电源控制和状态寄存器 (PMU_CS)

地址偏移: 0x04

复位值: 0x0000 0000 (从待机模式唤醒后不复位)

该寄存器可以按半字 (16 位) 或字 (32 位) 访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															NPRDY
															r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	LDOVSR F	WUPEN5	保留	WUPEN3	WUPEN2	WUPEN1	WUPEN0	保留						STBF	WUF
	r	rw		rw	rw	rw	rw							r	r

位/位域	名称	描述
31:17	保留	必须保持复位值。
16	NPRDY	NPLDO 就绪标志 0: NPLDO 未就绪。 1: NPLDO 就绪。
15	保留	必须保持复位值。
14	LDOVSRF	LDO 电压选择就绪标志 0: LDO 电压选择未就绪。 1: LDO 电压选择就绪。
13	WUPEN5	WKUP 引脚 5 (PB5) 唤醒使能 0: 禁能 WKUP 引脚 5 唤醒功能。 1: 使能 WKUP 引脚 5 唤醒功能。 如果 WUPEN5 在进入省电模式之前置 1, WKUP 引脚 5 的上升沿会将系统从省电模式唤醒。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。当在输入已经为高的时候置位该控制位，将会触发一个唤醒事件。
12	保留	必须保持复位值。
11	WUPEN3	<p>WKUP 引脚 3 (PA2) 唤醒使能</p> <p>0: 禁能 WKUP 引脚 3 唤醒功能。</p> <p>1: 使能 WKUP 引脚 3 唤醒功能。</p> <p>如果 WUPEN4 在进入省电模式之前置 1, WKUP 引脚 3 的上升沿会将系统从省电模式唤醒。由于 WKUP 引脚 3 为高电平有效, WKUP 引脚 3 内部被配置为输入下拉模式。当在输入已经为高的时候置位该控制位, 将会触发一个唤醒事件。</p>
10	WUPEN2	<p>WKUP 引脚 2 (PB6) 唤醒使能</p> <p>0: 关闭 WKUP 引脚 2 唤醒功能。</p> <p>1: 开启 WKUP 引脚 2 唤醒功能。</p> <p>如果 WUPEN2 在进入省电模式之前置 1, WKUP 引脚 2 的上升沿会将系统从省电模式唤醒。由于 WKUP 引脚 2 为高电平有效, WKUP 引脚 2 内部被配置为输入下拉模式。当在输入已经为高的时候置位该控制位, 将会触发一个唤醒事件。</p>
9	WUPEN1	<p>WKUP 引脚 1 (PC13/PA4) 唤醒使能</p> <p>0: 关闭 WKUP 引脚 1 唤醒功能。</p> <p>1: 开启 WKUP 引脚 1 唤醒功能。</p> <p>如果 WUPEN1 在进入省电模式之前置 1, WKUP 引脚 1 的上升沿会将系统从省电模式唤醒。由于 WKUP 引脚 1 为高电平有效, WKUP 引脚 1 内部被配置为输入下拉模式。当在输入已经为高的时候置位该控制位, 将会触发一个唤醒事件。</p> <p>注意: LQFP48、QFN48 封装下只有 PC13 可用。</p>
8	WUPEN0	<p>WKUP 引脚 0 (PA0) 唤醒使能</p> <p>0: 关闭 WKUP 引脚 0 唤醒功能。</p> <p>1: 开启 WKUP 引脚 0 唤醒功能。</p> <p>如果 WUPEN0 在进入省电模式之前置 1, WKUP 引脚 0 的上升沿会将系统从省电模式唤醒。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。当在输入已经为高的时候置位该控制位, 将会触发一个唤醒事件。</p>
7:2	保留	必须保持复位值。
1	STBF	<p>待机标志</p> <p>0: 设备没进入过待机模式。</p> <p>1: 设备曾进入过待机模式。</p> <p>该位只能由 POR / PDR 或通过置位 PMU_CTL0 寄存器的 STBRST 位来清零。</p>
0	WUF	<p>唤醒标志</p> <p>0: 没有收到唤醒事件。</p> <p>1: 唤醒事件由 WKUP 引脚或 RTC 事件包括 RTC 闹钟事件, 时间戳事件, 侵入事件和自动唤醒事件触发。</p> <p>该位只能由 POR / PDR 或通过设置 PMU_CTL0 寄存器的 WURST 位来清零。</p>

3.4.3. 控制寄存器 1 (PMU_CTL1)

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
31:6	保留	必须保持复位值。
5	EFDSPSLEEP	深度睡眠模式/深度睡眠模式 1 下, EFLASH 掉电控制 该位仅在深度睡眠模式/深度睡眠模式 1 下由软件置 1。 0: EFLASH 上电。 1: EFLASH 掉电。
4	EFPSLEEP	运行模式/运行模式 1 下, EFLASH 掉电控制 该位仅在运行模式/运行模式 1 下由软件置 1。 0: EFLASH 上电。 1: EFLASH 掉电。
3:0	保留	必须保持复位值。

3.4.4. 状态寄存器 (PMU_STAT)

地址偏移: 0x0C

复位值: 0x00C0 0020 (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。



位/位域	名称	描述
31:6	保留	必须保持复位值。

- 5 EFLASHPS_ACTIVE EFLASH 处于运行状态
- 4 EFLASHPS_SLEEP EFLASH 处于睡眠状态。
- 3:0 保留 必须保持复位值。

3.4.5. 参数寄存器 (PMU_PAR)

地址偏移: 0x10

复位值: 0x190A 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
31:29	保留	必须保持复位值。
28:21	TWK_EFLASH[7:0]	EFLASH 从深度睡眠模式/深度睡眠模式 1 唤醒计数器。 当从深度睡眠模式/深度睡眠模式 1 唤醒时, 等待 TWK_EFLASH 个 IRC48M 时钟周期。 默认值为 200 (4.17us)。
20:16	TSW_IRC48MCNT[4:0]	当进入深度睡眠模式/深度睡眠模式 1 时, 切换到 IRC48M 时钟。 等待 IRC48M 计数后设置深度睡眠状态。 默认值为 10 个 IRC48M 时钟。
15:0	保留	必须保持复位值。

4. 复位和时钟单元（RCU）

4.1. 复位控制单元（RCTL）

4.1.1. 简介

GD32C2x1复位控制包括三种控制方式：电源复位、系统复位和备份域（V_{CORE_STB}）复位。电源复位又称为冷复位，其复位除了备份域的所有系统。系统复位将复位除了SW-DP控制器和备份域之外的其余部分，包括处理器内核和外设IP。备份域复位将复位备份区域。复位能够被外部信号、内部事件和复位发生器触发。后续章节将介绍关于这些复位的详细信息。

4.1.2. 功能描述

电源复位

当发生以下任一事件时，产生电源复位：上电/掉电复位（POR/PDR复位），从待机模式中返回后由内部复位发生器产生。电源复位复位所有的寄存器除了备份域。电源复位为低电平有效，当内部LDO电源基准准备好提供1.2V电压时，电源复位电平将变为无效。复位入口向量被固定在存储器映射的地址0x0000_0004。

系统复位

当发生以下任一事件时，产生一个系统复位：

- 上电复位（POWER_RSTn）
- 外部引脚复位（NRST）
- 窗口看门狗计数终止（WWDGT_RSTn）
- 独立看门狗计数终止（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）
- 选项字节重载复位（OBL_RSTn）
- 用户选择字节寄存器nRST_STDBY设置为0，并且进入待机模式时将产生复位（OB_STDBY_RSTn）
- 用户选择字节寄存器nRST_DPSLP设置为0，并且进入深度睡眠模式时（OB_DPSLP_RSTn）

注意：NRST引脚可以在以下三种模式下配置选项字节NRST_MDSEL[1:0]：

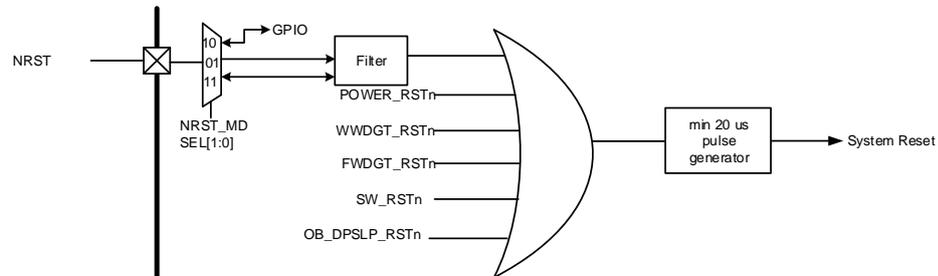
1. 输入/输出模式（默认模式）：在这种模式下，NRST引脚的GPIO功能不可用。复位信号可以从NRST引脚传输到MCU，导致MCU复位，复位脉冲信号可以通过NRST引脚反应出来，最小复位脉冲持续时间为20us。
2. 输入模式：在这种模式下，NRST引脚的GPIO功能不可用，复位信号可以从NRST引脚传输到MCU，导致MCU复位，但在NRST引脚上不可见该MCU的内部复位。
3. GPIO模式：NRST引脚只能作为标准GPIO使用，复位功能不可用，复位信号仅在MCU内

部，不反映在NRST引脚上。

系统复位将复位除了 SW-DP 控制器和备份域之外的其余部分，包括处理器内核和外设 IP。

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 20 μ s 的低电平脉冲延时。

图 4-1. 系统复位电路



备份域 (V_{CORE_STB}) 复位

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、电源上电复位（VDD 上电）。

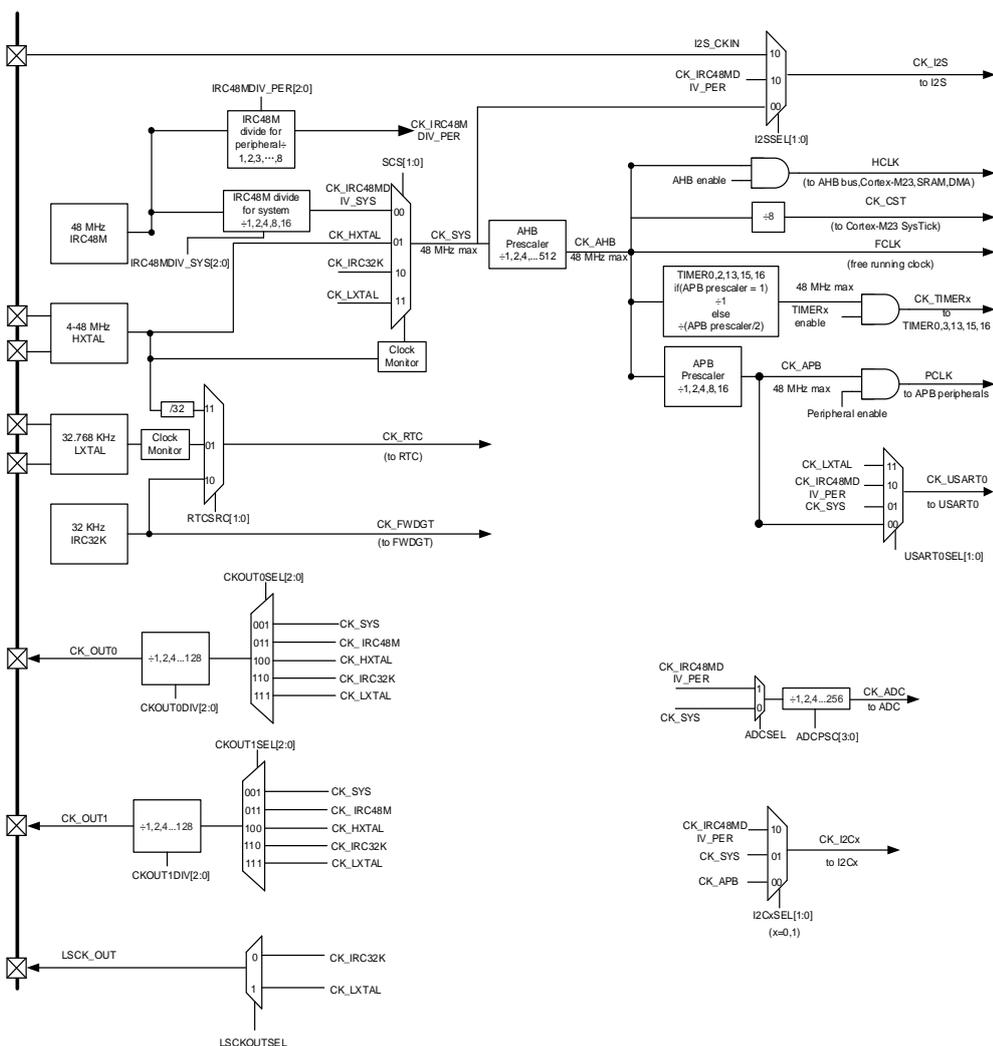
4.2. 时钟控制单元 (CCTL)

4.2.1. 简介

时钟控制单元提供了一系列频率的时钟功能，包括一个内部 48MRC 振荡器时钟 (IRC48M)、一个外部高速晶体振荡器时钟 (HXTAL)、一个内部 32K RC 振荡器时钟 (IRC32K)、一个外部低速晶体振荡器时钟 (LXTAL)、一个 HXTAL 时钟监视器、一个 LXTAL 时钟监视器、时钟预分频器、时钟多路复用器和时钟门控电路。

AHB、APB 和 Cortex[®]-M23 时钟都源自系统时钟 (CK_SYS)，系统时钟的时钟源可以选择 IRC48MDIV_SYS、HXTAL、IRC32K 或 LXTAL。系统时钟的最大运行时钟频率可以达到 48MHz。

图 4-2. 时钟树



预分频器可以配置 AHB 和 APB 域的时钟频率。AHB 和 APB 域的最高时钟频率分别为 48MHz / 48MHz。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 控制和状态寄存器的设置，可选择上述时钟或 AHB（HCLK）时钟作为 SysTick 时钟。

ADC 的时钟由 CK_SYS 经 1、2、4、...、256 分频或者 CK_IRC48MDIV_PER 经 1、2、4、...、256 分频获得，通过配置 RCU_CFG1 寄存器中的 ADCSEL 位和 ADCPSC 位来选择。

USART0 的时钟由 IRC48MDIV_PER 时钟或者 LXTAL 时钟或者系统时钟或者 APB 时钟提供，通过配置 RCU_CFG1 寄存器中的 USART0SEL 位来选择。

I2Cx (x = 0, 1) 的时钟由 IRC48MDIV_PER 时钟或者系统时钟或者 APB 时钟提供，通过配置 RCU_CFG1 寄存器中的 I2CxSEL (x = 0, 1) 位来选择。

RTC 的时钟由 LXTAL 时钟或者 IRC32K 时钟或者 HXTAL/32 时钟提供，通过配置 RCU_CTL1 寄存器中的 RTCSRC 位来选择。

当 FWDGT 启动时，FWDGT 时钟被强制选择由 IRC32K 时钟做为时钟源。

I2S 的时钟由 I2S_CKIN 时钟或者 CK_IRC48MDIV_PER 时钟或者 CK_SYS 时钟提供，通过配置 RCU_CFG1 寄存器中的 I2SSEL 位来选择。

TIMER 时钟由 CK_APB 时钟分频获得，如果 APB 的分频系数为 1，则 TIMER 时钟频率等于 CK_APB。如果 APB 的分频系数不为 1，则 TIMER 时钟频率为 CK_APB 的两倍。

4.2.2. 主要特性

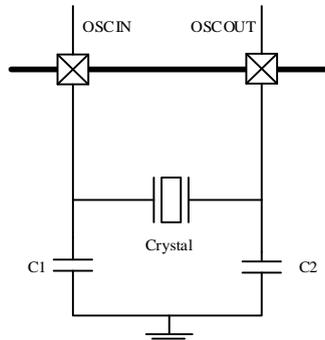
- 4到48MHz外部高速晶体振荡器（HXTAL）；
- 内部48MHz RC振荡器（IRC48M）；
- 32768 Hz外部低速晶体振荡器（LXTAL）；
- 内部32KHz RC振荡器（IRC32K）；
- HXTA和LXTAL时钟监视器。

4.2.3. 功能描述

外部高速晶体振荡时钟（HXTAL）

4 到 48M 的外部高速晶体振荡器可为系统时钟提供更为精确时钟源。带有特定频率的晶体必须靠近两个 HXTAL 的引脚连接。和晶体连接的外部电阻和电容必须根据所选择的振荡器来调整。

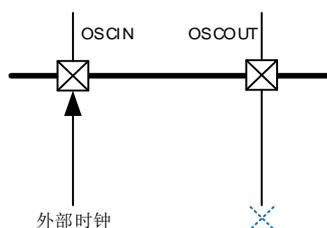
图 4-3. HXTAL 时钟源



HXTAL 晶体振荡器可以通过设置控制寄存器 RCU_CTL 的 HXTALEN 位来启动或关闭，在控制寄存器 RCU_CTL 中的 HXTALSTB 位用来指示外部高速振荡器是否已稳定。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被称为振荡器的启动时间。当 HXTAL 时钟稳定后，如果在中断寄存器 RCU_INT 中的相应中断使能位 HXTALSTBIE 位被置‘1’，将会产生相应中断。此时，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_CTL 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 4-4. 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 4-4. 旁路模式下 HXTAL 时钟源



内部 48M RC 振荡器时钟（IRC48M）

内部 48MHz RC 振荡器时钟，简称 IRC48M 时钟，拥有 48MHz 的固定频率。IRC48M RC 振荡器可以通过设置控制寄存器（RCU_CTL）中的 IRC48MEN 位被启动和关闭。控制寄存器 RCU_CTL 中的 IRC48MSTB 位用来指示 IRC48M 内部 RC 振荡器是否稳定。如果中断寄存器 RCU_INT 中的相应中断使能位 IRC48MSTBIE 被置‘1’，在 IRC48M 稳定以后，将产生一个中断。

外部低速晶体振荡器时钟（LXTAL）

LXTAL 是一个频率为 32.768kHz 的外部低速晶体或陶瓷谐振器。它为实时时钟电路提供一个低功耗且高精度的时钟源。LXTAL 振荡器可以通过设置控制寄存器 1（RCU_CTL1）中的 LXTALEN 位被启动和关闭。控制寄存器 1 RCU_CTL1 中的 LXTALSTB 位用来指示 LXTAL 时钟是否稳定。如果中断寄存器 RCU_INT 中的相应中断使能位 LXTALSTBIE 被置‘1’，在 LXTAL 稳定以后，将产生一个中断。

将控制寄存器 1 RCU_CTL1 的 LXTALBPS 和 LXTALEN 位置‘1’可以选择外部时钟旁路模式。CK_LXTAL 与连到 OSC32IN 脚上外部时钟信号一致。

内部 32K RC 振荡器时钟（IRC32K）

IRC32K 内部 RC 振荡器时钟担当一个低功耗时钟源的角色，不需要外部器件，它的时钟频率大约 32kHz，为独立看门狗和实时时钟电路提供时钟。IRC32K RC 振荡器可以通过设置复位源/时钟寄存器 RCU_RSTSCK 中的 IRC32KEN 位被启动和关闭。复位源/时钟寄存器 RCU_RSTSCK 中的 IRC32KSTB 位用来指示 IRC32K 时钟是否已稳定。如果复位源/时钟寄存器 RCU_RSTSCK 中的相应中断使能位 IRC32KSTBIE 被置‘1’，在 IRC32K 稳定以后，将产生一个中断。

系统时钟（CK_SYS）选择

系统复位后，IRC48M/4 时钟默认做为 CK_SYS 的时钟源，改变配置寄存器 0（RCU_CFG0）中的系统时钟变换位 SCS 可以切换系统时钟源为 HXTAL 或 IRC48MDIV_SYS 或者 LXTAL。当 SCS 的值被改变，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作为系统时钟时，它将不能被停止。

HXTAL 时钟监视器（CKM）

设置控制寄存器 RCU_CTL 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，中断寄存器 RCU_INT 中的 HXTAL 时钟阻塞中断标志位 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 NMI 相连。如果 HXTAL 被选作系统，HXTAL 故障将促使选择 IRC48MDIV_SYS 为系统时钟源。

LXTAL 时钟监视器 (LCKM)

设置时钟控制寄存器 RCU_CTL 中的 LXTAL 时钟监视使能位 LCKMEN，LXTAL 可以使能时钟监视功能。该功能必须在 LXTAL 启动延迟完毕和 IRC32K 使能后使能。

LXTAL 上的时钟监视器在除 V_{CORE_STB} 以外的所有模式下工作。如果在外部 32 KHz 振荡器上检测到故障，可以向 CPU 发送中断。这个故障引发的中断和 Cortex®-M23 的不可屏蔽中断 NMI 相连。如果选择 LXTAL 作为 CK_SYS 的时钟源，则 LXTAL 故障将强制 CK_SYS 的时钟源切换为 IRC32K。

然后，软件必须禁用 LCKMEN 位，停止有缺陷的 32 KHz 振荡器，并更改 RTC 时钟源，或采取任何必要的措施来保护应用程序。

当 LCKMEN 启用时，一个 4 位加一个计数器将在 IRC32K 域工作。如果 LXTAL 时钟卡在 0/1 错误或时钟减慢约 20KHz，计数器将溢出。将发现 LXTAL 时钟故障。

时钟输出功能

时钟输出功能输出从 32KHz 到 48MHz 的时钟。通过设置时钟配置寄存器 0 (RCU_CFG0) 中的 CK_OUT 时钟源选择位域 CKOUTxSEL (x = 0,1) 能够选择不同的时钟信号。相应的 GPIO 引脚应该被配置成备用功能 I/O (AFIO) 模式来输出选择的时钟信号。

表 4-1. 时钟输出的时钟源选择

时钟输出的时钟源选择位域	时钟源
000	没有时钟
001	CK_SYS
010	保留
011	CK_IRC48M
100	CK_HXTAL
101	保留
110	CK_IRC32K
111	CK_LXTAL

通过配置时钟配置寄存器 RCU_CFG0 的 CKOUTxDIV[2:0] (x = 0,1) 位，可以将输出时钟按比例分频，进而降低 CK_OUT 频率。

通过设置 RCU_CTL1 寄存器的 LSCKOUTSEL 位，CK_LXTAL 和 CK_IRC32K 时钟可以通过 LSCK_OUT 引脚输出，即使在深度睡眠模式、深度睡眠 1 模式和待机模式。

表 4-2. 低速时钟输出的时钟源选择

时钟输出的时钟源选择位域	时钟源
0	CK_IRC32K
1	LXTAL

深度睡眠模式时钟控制

当 MCU 处于深度睡眠/深度睡眠 1 模式时，USART0 外设时钟由 LXTAL 提供且 LXTAL 时钟使能时，则 USART0 外设可以唤醒 MCU。

如果 USART0 时钟选择 IRC48MDIV_PER 处于深度睡眠/深度睡眠 1 模式时，则它们能够打开 IRC48M 时钟或关闭 IRC48M 时钟，从而使 USART0 从深度睡眠模式唤醒。

如果 USART0 时钟选择 LXTAL 处于深度睡眠/深度睡眠 1 模式时，则它们能够打开 LXTAL 时钟或关闭 LXTAL 时钟（如果 LXTAL 由软件打开，则 USART0 不能关闭 LXTAL）。

如果 I2C0 / I2C1 选择 IRC48MDIV_PER 作为时钟源并处于深度睡眠/深度睡眠 1 模式，则它们能够打开或关闭 IRC48M 时钟，从而使 I2C0 / I2C1 从深度睡眠/深度睡眠 1 模式唤醒。

如果 FMC 和 PMU 在深度睡眠/深度睡眠 1 模式下工作时，可以打开或关闭 IRC48M 时钟。

为了在深度睡眠/深度睡眠 1 模式下省电，如果 FMC 和 USART0 未在深度睡眠/深度睡眠 1 模式下工作，则它们的时钟可以单独选通。但 I2C0 / I2C1 / ADC、PMU 功能时钟不能由硬件选通，可以由软件禁用。

4.3. RCU 寄存器

RCU 基地址: 0x4002 1000

4.3.1. 控制寄存器 (RCU_CTL0)

地址偏移: 0x00

复位值: 0x4400 XX43 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
IRC48MDIV_SYS[2:0]			IRC48M_ PEREN	IRC48MDIV_PER[2:0]			保留					CKMEN	HXTALBP S	HXTALST B	HXTALE N
rw			rw	rw								rw	rw	r	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRC48MCALIB[7:0]										IRC48MADJ[5:0]				IRC48MS TB	IRC48ME N
r										rw				r	rw

位/位域	名称	描述
31:29	IRC48MDIV_SYS[2:0]	IRC48M时钟分频系数用于系统时钟 这些位由软件设置IRC48M时钟分频器的分频系数，以产生CK_IRC48MDIV_SYS时钟。 000: CK_IRC48MDIV_SYS = CK_IRC48M / 1 001: CK_IRC48MDIV_SYS = CK_IRC48M / 2 010: CK_IRC48MDIV_SYS = CK_IRC48M / 4 (复位值) 011: CK_IRC48MDIV_SYS = CK_IRC48M / 8 100: CK_IRC48MDIV_SYS = CK_IRC48M / 16 101: CK_IRC48MDIV_SYS = CK_IRC48M / 32 110: CK_IRC48MDIV_SYS = CK_IRC48M / 64 111: CK_IRC48MDIV_SYS = CK_IRC48M / 128
28	IRC48M_PEREN	IRC48M时钟提供给外设始终使能。 该位由软件设置和清除。置位该位会当MCU在正常工作模式和深度睡眠模式下使能IRC48M振荡器，而不考虑IRC48MEN位的状态。IRC48M时钟只能提供给已配置IRC48M为时钟源的USART0, I2C0和I2C1外设。 0: IRC48M振荡器使能取决于IRC48MEN位 1: IRC48M振荡器在MCU运行模式和深度睡眠模式下都是使能的 注意: 在深度睡眠模式下保持IRC48M使能，可以加速串行接口通信，因为IRC48M时钟在退出深度睡眠模式时立即准备就绪。
27:25	IRC48MDIV_PER[2:0]	由软件控制的这些位设置外设时钟分频器的分频系数，以产生CK_IRC48MDIV_PER时钟。 000: CK_IRC48MDIV_PER = CK_IRC48M / 1 001: CK_IRC48MDIV_PER = CK_IRC48M / 2

		010: CK_IRC48MDIV_PER = CK_IRC48M / 3 (复位值)
		011: CK_IRC48MDIV_PER = CK_IRC48M / 4
		100: CK_IRC48MDIV_PER = CK_IRC48M / 5
		101: CK_IRC48MDIV_PER = CK_IRC48M / 6
		110: CK_IRC48MDIV_PER = CK_IRC48M / 7
		111: CK_IRC48MDIV_PER = CK_IRC48M / 8
24:20	保留	必须保持复位值。
19	CKMEN	<p>HXTAL时钟监视使能</p> <p>0: 禁止外部4 ~ 48 MHz晶体振荡器 (HXTAL) 时钟监视器</p> <p>1: 使能外部4 ~ 48 MHz晶体振荡器 (HXTAL) 时钟监视器</p> <p>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IRC48M 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。</p> <p>注意: 使能HXTAL时钟监视器以后, 硬件无视控制位IRC48MEN的状态, 自动使能IRC48M时钟。</p>
18	HXTALBPS	<p>外部晶体振荡器 (HXTAL) 时钟旁路模式使能</p> <p>只有在HXTALEN位为0时, HXTALBPS位才可写。</p> <p>0: 禁止HXTAL旁路模式</p> <p>1: 使能HXTAL旁路模式, HXTAL输出时钟等于输入时钟</p>
17	HXTALSTB	<p>外部晶体振荡器 (HXTAL) 时钟稳定状态标志位</p> <p>硬件置'1'来指示HXTAL振荡器时钟是否稳定待用。</p> <p>0: HXTAL振荡器未稳定</p> <p>1: HXTAL振荡器已稳定</p>
16	HXTALEN	<p>外部高速振荡器时钟使能</p> <p>软件置'1'或清'0'。如果HXTAL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。</p> <p>0: 禁止外部4 ~ 48 MHz晶体振荡器</p> <p>1: 使能外部4 ~ 48 MHz晶体振荡器</p>
15:8	IRC48MCALIB[7:0]	<p>高速内部振荡器校准值寄存器</p> <p>上电时自动加载这些位</p>
7:2	IRC48MADJ[5:0]	<p>高速内部振荡器时钟调整值</p> <p>这些位由软件置位, 最终调整值为IRC48MADJ当前值加上IRC48MCALIB[7:0]位的值。最终调整值应该调整IRC48M到48MHz ± 1%。</p>
1	IRC48MSTB	<p>高速内部 (IRC48M) 时钟稳定状态标志位</p> <p>硬件置'1'来指示IRC48M振荡器时钟是否稳定待用。</p> <p>0: IRC48M振荡器未稳定</p> <p>1: IRC48M振荡器已稳定</p>
0	IRC48MEN	<p>高速内部振荡器使能</p> <p>软件复位置位。如果IRC48M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HXTALCKM置位的情况下用作系统时钟的HXTAL振荡器发生</p>

故障时，该位由硬件置1来启动IRC48M振荡器。

0: 内部48 MHz RC振荡器关闭

1: 内部48 MHz RC振荡器开启

4.3.2. 配置寄存器 0 (RCU_CFG0)

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	CKOUT0DIV[2:0]			保留	CKOUT0SEL[2:0]			保留	CKOUT1DIV[2:0]			保留	CKOUT1SEL[2:0]		
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	APBPSC[2:0]			保留			AHBPSC[3:0]			SCSS[1:0]		SCS[1:0]			
rw				rw				r		rw					

位/位域	名称	描述
31	保留	必须保持复位值。
30:28	CKOUT0DIV[2:0]	CK_OUT0分频器，用来降低CK_OUT0频率 CK_OUT0的选择参考RCU_CFG0的26:24位。 000: CK_OUT0不分频 001: CK_OUT0 2分频 010: CK_OUT0 4分频 011: CK_OUT0 8分频 100: CK_OUT0 16分频 101: CK_OUT0 32分频 110: CK_OUT0 64分频 111: CK_OUT0 128分频
27	保留	必须保持复位值。
26:24	CKOUT0SEL[2:0]	CK_OUT0时钟源选择 软件置位或清零。 000: 没有时钟被选择 001: 选择系统时钟 010: 保留 011: 选择内部48M RC振荡器时钟 100: 选择外部高速振荡器时钟 101: 保留 110: 选择内部32K RC振荡器时钟 111: 选择外部低速振荡器时钟
23	保留	必须保持复位值。
22:20	CKOUT1DIV[2:0]	CK_OUT1分频器，来降低CK_OUT1频率

		CK_OUT1的选择参考RCU_CFG0的18:16位。
		000: CK_OUT1不分频
		001: CK_OUT1 2分频
		010: CK_OUT1 4分频
		011: CK_OUT1 8分频
		100: CK_OUT1 16分频
		101: CK_OUT1 32分频
		110: CK_OUT1 64分频
		111: CK_OUT1 128 分频
19	保留	必须保持复位值。
18:16	CKOUT1SEL[2:0]	CK_OUT1时钟源选择 软件置位或清零。
		000: 没有时钟被选择
		001: 选择系统时钟
		010: 保留
		011: 选择内部48M RC振荡器时钟
		100: 选择外部高速振荡器时钟
		101: 保留
		110: 选择内部32K RC振荡器时钟
		111: 选择外部低速振荡器时钟
15:14	保留	必须保持复位值。
13:11	APBPSC[2:0]	APB预分频选择 软件置1和清0来控制APB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	保留	必须保持复位值。
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_IRC48MDIV_SYS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_IRC32K作为CK_SYS系统时钟源

11: 选择CK_LXTAL作为CK_SYS系统时钟源

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。

00: 选择CK_IRC48MDIV_SYS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IRC32K作为CK_SYS系统时钟源

11: 选择LXTAL作为CK_SYS系统时钟源

4.3.3. 中断寄存器（RCU_INT）

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								CKMIC	LCKMIC	保留			HXTAL STBIC	IRC48M STBIC	LXTAL STBIC	IRC32K 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				HXTAL STBIE	IRC48M STBIE	LXTAL STBIE	IRC32K STBIE	CKMIF	LCKMIF	保留			HXTAL STBIF	IRC48M STBIF	LXTAL STBIF	IRC32K STBIF
				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	LCKMIC	LXTAL时钟阻塞中断清除 软件写1复位LCKMIF标志位。 0: 不复位LCKMIF标志位 1: 复位LCKMIF标志位
21:20	保留	必须保持复位值。
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	IRC48MSTBIC	IRC48M时钟稳定中断清除 软件写1复位IRC48MSTBIF标志位。 0: 不复位IRC48MSTBIF标志位 1: 复位IRC48MSTBIF标志位
17	LXTALSTBIC	LXTAL时钟稳定中断清除 软件写1复位LXTALSTBIF标志位。 0: 不复位LXTALSTBIF标志位 1: 复位LXTALRDYF标志位
16	IRC32KSTBIC	IRC32K时钟稳定中断清除 软件写1复位IRC32KSTBIF标志位。 0: 不复位IRC32KSTBIF标志位 1: 复位IRC32KSTBIF标志位
15:12	保留	必须保持复位值。
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	IRC48MSTBIE	IRC48M时钟稳定中断使能 软件置1和清0来使能/禁止IRC48M时钟稳定中断。 0: 禁止IRC48M时钟稳定中断 1: 使能IRC48M时钟稳定中断
9	LXTALSTBIE	LXTAL时钟稳定中断使能 LXTAL时钟稳定中断使能/禁止控制。 0: 禁止LXTAL时钟稳定中断 1: 使能LXTAL时钟稳定中断
8	IRC32KSTBIE	IRC32K时钟稳定中断使能 IRC32K时钟稳定中断使能/禁止控制。 0: 禁止IRC32K时钟稳定中断 1: 使能IRC32K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	LCKMIF	LXTAL时钟阻塞中断标志位 当LXTAL时钟阻塞由硬件置1。 软件置位LCKMIC时清除该位。 0: LXTAL时钟运行正常

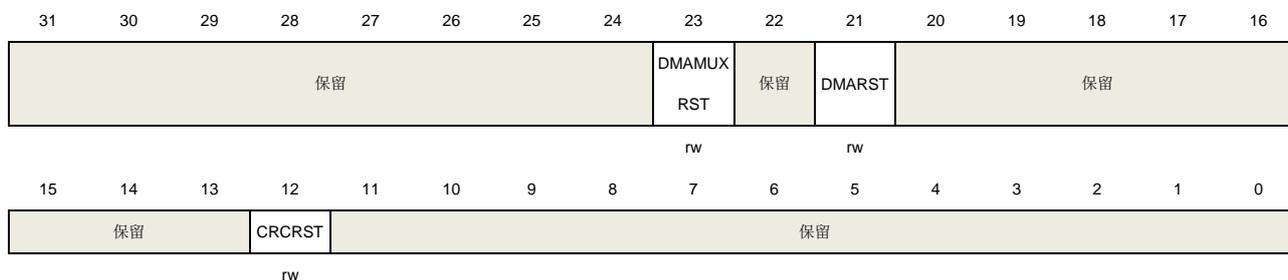
		1: LXTAL时钟阻塞
5:4	保留	必须保持复位值。
3	HXTALSTBIF	HXTAL时钟稳定中断标志位 当外部4 ~ 48 MHz晶体振荡器时钟稳定且HXTALSTBIE位被置1时由硬件置1。 软件置HXTALSTBIC=1时清除该位。 0: 无HXTAL时钟稳定中断发生 1: 发生HXTAL时钟稳定中断
2	IRC48MSTBIF	IRC48M时钟稳定中断标志位 当内部48 MHz RC振荡器时钟稳定且IRC48MSTBIE位被置1时由硬件置1。 软件置IRC48MSTBIC=1时清除该位。 0: 无IRC48M时钟稳定中断产生 1: 产生IRC48M时钟稳定中断
1	LXTALSTBIF	LXTAL时钟稳定中断标志位 当外部32.768KHz晶体振荡器时钟稳定且LXTALSTBIE为被置1时由硬件置1。 软件置LXTALSTBIC=1时清除该位。 0: 无LXTAL时钟稳定中断发生 1: 发生LXTAL时钟稳定中断
0	IRC32KSTBIF	IRC32K时钟稳定中断标志位 当内部32KHz RC振荡器时钟稳定且IRC32KSTBIE位被置1时由硬件置1。 软件置IRC32KSTBIC =1时清除该位。 0: 无IRC32K时钟稳定中断产生 1: 产生IRC32K时钟稳定中断

4.3.4. AHB1 复位寄存器 (RCU_AHB1RST)

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。



位/位域	名称	描述
31:24	保留	必须保持复位值。
23	DMAMUXRST	DMAMUX复位 由软件置1或清0。

		0: 无复位 1: 复位DMAMUX
22	保留	必须保持复位值。
21	DMARST	DMA复位 由软件置1或清0。 0: 无复位 1: 复位DMA
20:13	保留	必须保持复位值。
12	CRCRST	CRC复位 由软件置1或清0。 0: 无复位 1: 复位CRC
11:0	保留	必须保持复位值。

4.3.5. AHB2 复位寄存器 (RCU_AHB2RST)

地址偏移: 0x14

复位值: 0x0000 0000

该寄存器可以按字节 (8 位)、半字 (16 位) 或字 (32 位) 访问。

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

位/位域	名称	描述
31:23	保留	必须保持复位值。
22	PFRST	GPIO端口F复位 由软件置位或复位 0: 无作用 1: 复位GPIO端口F
21	保留	必须保持复位值。
20	PDRST	GPIO端口D复位 由软件置位或复位 0: 无作用 1: 复位GPIO端口D
19	PCRST	GPIO端口C复位

		由软件置位或复位 0: 无作用 1: 复位GPIO端口C
18	PBRST	GPIO端口B复位 由软件置位或复位 0: 无作用 1: 复位GPIO端口B
17	PARST	GPIO端口A复位 由软件置位或复位 0: 无作用 1: 复位GPIO端口A
16:0	保留	必须保持复位值。

4.3.6. APB 复位寄存器 (RCU_APBRSR)

地址偏移: 0x24

复位值: 0x0000 0000

该寄存器可以按字节 (8 位)、半字 (16 位) 或字 (32 位) 访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
保留			PMURST	保留				I2C1RST	I2C0RST	保留	USART2 RST	TIMER16 RST	TIMER15 RST	TIMER13 RST		
			rw						rw		rw		rw		rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
USART1 RST	USART0 RST	SPI1RST	SPI0RST	TIMER2 RST	TIMER0R ST	ADCRST	WWDGT RST	保留					CMRST	SYSCFG RST		
rw		rw		rw		rw							rw		rw	

位/位域	名称	描述
31:29	保留	必须保持复位值。
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
27:23	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	保留	必须保持复位值。
19	USART2RST	USART2复位 由软件置1或清0。 0: 无复位 1: 复位USART2
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	TIMER13RST	TIMER13定时器复位 由软件置1或清0。 0: 无复位 1: 复位TIMER13定时器
15	USART1RST	USART1定时器复位 由软件置1或清0。 0: 无复位 1: 复位USART1定时器
14	USART0RST	USART0定时器复位 由软件置1或清0。 0: 无复位 1: 复位USART0定时器
13	SPI1RST	SPI1复位 由软件置1或清0。 0: 无复位 1: 复位SPI1
12	SPI0RST	SPI0复位 由软件置1或清0。 0: 无复位 1: 复位SPI0
11	TIMER2RST	TIMER2定时器复位 由软件置1或清0。 0: 无复位 1: 复位TIMER2定时器

10	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
9	ADCRST	ADC复位 由软件置1或清0。 0: 无复位 1: 复位ADC
8	WWDGTRST	WWDGT复位 由软件置1或清0。 0: 无复位 1: 复位WWDGT
7:2	保留	必须保持复位值。
1	CMPRST	CMP复位 由软件置1或清0。 0: 无复位 1: 复位CMP
0	SYSCFGRST	系统配置复位 由软件置1或清0。 0: 无复位 1: 复位系统配置

4.3.7. AHB1 使能寄存器 (RCU_AHB1EN)

地址偏移: 0x30

复位值: 0x0000 0010

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
31:24	保留	必须保持复位值。
23	DMAMUXEN	DMAMUX时钟使能 由软件置1或清0。 0: DMAMUX时钟关闭

		1: DMAMUX时钟开启
22	保留	必须保持复位值。
21	DMAEN	DMA时钟使能 由软件置1或清0。 0: DMA时钟关闭 1: DMA时钟开启
20:13	保留	必须保持复位值。
12	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
11:5	保留	必须保持复位值。
4	FMCEN	FMC时钟使能 由软件置1或清0。 0: FMC时钟关闭 1: FMC时钟开启
3:0	保留	必须保持复位值。

4.3.8. AHB2 使能寄存器 (RCU_AHB2EN)

地址偏移: 0x34

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									PFEN	保留	PDEN	PCEN	PBEN	PAEN	保留
									r/w		r/w	r/w	r/w	r/w	
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
31:23	保留	必须保持复位值。
22	PFEN	GPIOF时钟使能 由软件置1或清0。 0: GPIOF时钟关闭 1: GPIOF时钟开启
21	保留	必须保持复位值。
20	PDEN	GPIOD时钟使能

		由软件置1或清0。 0: GPIOD时钟关闭 1: GPIOD时钟开启
19	PCEN	GPIOC时钟使能 由软件置1或清0。 0: GPIOC时钟关闭 1: GPIOC时钟开启
18	PBEN	GPIOB时钟使能 由软件置1或清0。 0: GPIOB时钟关闭 1: GPIOB时钟开启
17	PAEN	GPIOA时钟使能 由软件置1或清0。 0: GPIOA时钟关闭 1: GPIOA时钟开启
16:0	保留	必须保持复位值。

4.3.9. APB 使能寄存器 (RCU_APBEN)

地址偏移: 0x44

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			PMUEN	DBGEN	保留			I2C1EN	I2C0EN	保留	USART2	TIMER16	TIMER15	TIMER13	
			rw	rw				rw	rw		rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USART1	USART0	SPI1EN	SPI0EN	TIMER2E	TIMER0E	ADCEN	WWDGT	保留				CMPEN	SYSCFG		
EN	EN			N	N		EN						EN		
rw	rw	rw	rw	rw	rw	rw	rw					rw	rw		

位/位域	名称	描述
31:29	保留	必须保持复位值。
28	PMUEN	PMU时钟使能 由软件置1或清0。 0: 关闭PMU时钟 1: 开启PMU时钟
27	DBGEN	DBG时钟使能 由软件置1或清0。 0: 关闭DBG时钟

		1: 开启DBG时钟
26:23	保留	必须保持复位值。
22	I2C1EN	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
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	保留	必须保持复位值。
19	USART2EN	USART2时钟使能 由软件置1或清0。 0: 关闭USART2时钟 1: 开启USART2时钟
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	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
15	USART1EN	USART1时钟使能 由软件置1或清0。 0: 关闭USART1时钟 1: 开启USART1时钟
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	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
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	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
10	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
9	ADCEN	ADC时钟使能 由软件置1或清0。 0: 关闭ADC时钟 1: 开启ADC时钟
8	WWDGTEN	WWDGT时钟使能 由软件置1或清0。 0: 关闭WWDGT时钟 1: 开启WWDGT时钟
7:2	保留	必须保持复位值。
1	CMPEN	CMP模块时钟使能 由软件置1或清0。 0: 关闭CMP模块时钟 1: 开启CMP模块时钟
0	SYSCFGEN	系统配置时钟使能 由软件置1或清0。 0: 关闭系统配置时钟 1: 开启系统配置时钟

4.3.10. AHB1 睡眠和深度睡眠使能寄存器 (RCU_AHB1SPDPEN)

地址偏移: 0x050

复位值: 0x00A0 1014

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								DMAMUX SPDPEN	保留	DMASPD PEN	保留				

															rw																rw				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																				
保留			CRCSPD PEN	保留											FMCS PD PEN	保留	SRAMSP DPEN	保留																	
			rw												rw		rw																		

位/位域	名称	描述
31:24	保留	必须保持复位值。
23	DMAMUXSPDPEN	在睡眠和深度睡眠模式下 DMAMUX 时钟使能 由软件置位或复位 0: 在睡眠和深度睡眠模式下关闭 DMAMUX 时钟 1: 在睡眠和深度睡眠模式下开启 DMAMUX 时钟
22	保留	必须保持复位值。
21	DMASPDEN	在睡眠和深度睡眠模式下 DMA 时钟使能 由软件置位或复位 0: 在睡眠和深度睡眠模式下关闭 DMA 时钟 1: 在睡眠和深度睡眠模式下开启 DMA 时钟
20:13	保留	必须保持复位值。
12	CRCSPDEN	在睡眠和深度睡眠模式下 CRC 时钟使能 由软件置位或复位 0: 在睡眠和深度睡眠模式下关闭 CRC 时钟 1: 在睡眠和深度睡眠模式下开启 CRC 时钟
11:5	保留	必须保持复位值。
4	FMCSPDEN	在睡眠和深度睡眠模式下 FMC 时钟使能 由软件置位或复位 0: 在睡眠和深度睡眠模式下关闭 FMC 时钟 1: 在睡眠和深度睡眠模式下开启 FMC 时钟
3	保留	必须保持复位值。
2	SRAMSPDEN	在睡眠和深度睡眠模式下 SRAM 时钟使能 由软件置位或复位 0: 在睡眠和深度睡眠模式下关闭 SRAM 时钟 1: 在睡眠和深度睡眠模式下开启 SRAM 时钟
1:0	保留	必须保持复位值。

4.3.11. AHB2 睡眠和深度睡眠使能寄存器 (RCU_AHB2SPDEN)

地址偏移: 0x054

复位值: 0x005E 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									PFSPDP	保留	PDSPDP	PCSPDP	PBSPDP	PASDPDP	保留
									EN		EN	EN	EN	EN	
									rw		rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留															

位/位域	名称	描述
31:23	保留	必须保持复位值。
22	PFSPDPEN	在睡眠和深度睡眠模式下 GPIO 端口 F 时钟使能由软件置位或复位 0: 在睡眠和深度睡眠模式下关闭 GPIO 端口 F 时钟 1: 在睡眠和深度睡眠模式下开启 GPIO 端口 F 时钟
21	保留	必须保持复位值。
20	PDSPDPEN	在睡眠和深度睡眠模式下 GPIO 端口 D 时钟使能由软件置位或复位 0: 在睡眠和深度睡眠模式下关闭 GPIO 端口 D 时钟 1: 在睡眠和深度睡眠模式下开启 GPIO 端口 D 时钟
19	PCSPDPEN	在睡眠和深度睡眠模式下 GPIO 端口 C 时钟使能由软件置位或复位 0: 在睡眠和深度睡眠模式下关闭 GPIO 端口 C 时钟 1: 在睡眠和深度睡眠模式下开启 GPIO 端口 C 时钟
18	PBSPDPEN	在睡眠和深度睡眠模式下 GPIO 端口 B 时钟使能由软件置位或复位 0: 在睡眠和深度睡眠模式下关闭 GPIO 端口 B 时钟 1: 在睡眠和深度睡眠模式下开启 GPIO 端口 B 时钟
17	PASDPDPEN	在睡眠和深度睡眠模式下 GPIO 端口 A 时钟使能由软件置位或复位 0: 在睡眠和深度睡眠模式下关闭 GPIO 端口 A 时钟 1: 在睡眠和深度睡眠模式下开启 GPIO 端口 A 时钟
16:0	保留	必须保持复位值。

4.3.12. APB 睡眠和深度睡眠使能寄存器（RCU_APBSPDPEN）

地址偏移：0x64

复位值：0x106F FF03

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				PMUSPD	保留				I2C1SPD	I2C0SPD	保留	USART2	TIMER16	TIMER15	TIMER13
				PEN					PEN	PEN		SPDPEN	SPDPEN	SPDPEN	SPDPEN
				r/w					r/w	r/w		r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USART1	USART0	SPI1SPD	SPI0SPD	TIMER2S	TIMER0S	ADCSPD	WWDGT	保留					CMPSPD	SYSCFG	
SPDPEN	SPDPEN	PEN	PEN	PDPEN	PDPEN	PEN	SPDPEN						PEN	SPDPEN	
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w						r/w	r/w	

位/位域	名称	描述
31:29	保留	必须保持复位值。
28	PMUSPDEN	在睡眠和深度睡眠模式下 PMU 时钟使能 由软件置位或复位 0: 在睡眠和深度睡眠模式下关闭 PMU 时钟 1: 在睡眠和深度睡眠模式下开启 PMU 时钟
27:23	保留	必须保持复位值。
22	I2C1SPDPEN	在睡眠和深度睡眠模式下 I2C1 时钟使能 由软件置位或复位 0: 在睡眠和深度睡眠模式下关闭 I2C1 时钟 1: 在睡眠和深度睡眠模式下开启 I2C1 时钟
21	I2C0SPDPEN	在睡眠和深度睡眠模式下 I2C0 时钟使能 由软件置位或复位 0: 在睡眠和深度睡眠模式下关闭 I2C0 时钟 1: 在睡眠和深度睡眠模式下开启 I2C0 时钟
20	保留	必须保持复位值。
19	USART2SPDPEN	在睡眠和深度睡眠模式下 USART2 时钟使能 由软件置位或复位 0: 在睡眠和深度睡眠模式下关闭 USART2 时钟 1: 在睡眠和深度睡眠模式下开启 USART2 时钟
18	TIMER16SPDPEN	在睡眠和深度睡眠模式下 TIMER16 时钟使能 由软件置位或复位 0: 在睡眠和深度睡眠模式下关闭 TIMER16 时钟 1: 在睡眠和深度睡眠模式下开启 TIMER16 时钟
17	TIMER15SPDPEN	在睡眠和深度睡眠模式下 TIMER15 时钟使能 由软件置位或复位 0: 在睡眠和深度睡眠模式下关闭 TIMER15 时钟 1: 在睡眠和深度睡眠模式下开启 TIMER15 时钟
16	TIMER13SPDPEN	在睡眠和深度睡眠模式下 TIMER13 时钟使能 由软件置位或复位 0: 在睡眠和深度睡眠模式下关闭 TIMER13 时钟

		1: 在睡眠和深度睡眠模式下开启 TIMER13 时钟
15	USART1SPDPEN	在睡眠和深度睡眠模式下 USART1 时钟使能 由软件置位或复位 0: 在睡眠和深度睡眠模式下关闭 USART1 时钟 1: 在睡眠和深度睡眠模式下开启 USART1 时钟
14	USART0SPDPEN	在睡眠和深度睡眠模式下 USART0 时钟使能 由软件置位或复位 0: 在睡眠和深度睡眠模式下关闭 USART0 时钟 1: 在睡眠和深度睡眠模式下开启 USART0 时钟
13	SPI1SPDPEN	在睡眠和深度睡眠模式下 SPI1 时钟使能 由软件置位或复位 0: 在睡眠和深度睡眠模式下关闭 SPI1 时钟 1: 在睡眠和深度睡眠模式下开启 SPI1 时钟
12	SPI0SPDPEN	在睡眠和深度睡眠模式下 SPI0 时钟使能 由软件置位或复位 0: 在睡眠和深度睡眠模式下关闭 SPI0 时钟 1: 在睡眠和深度睡眠模式下开启 SPI0 时钟
11	TIMER2SPDPEN	在睡眠和深度睡眠模式下 TIMER2 时钟使能 由软件置位或复位 0: 在睡眠和深度睡眠模式下关闭 TIMER2 时钟 1: 在睡眠和深度睡眠模式下开启 TIMER2 时钟
10	TIMER0SPDPEN	在睡眠和深度睡眠模式下 TIMER0 时钟使能 由软件置位或复位 0: 在睡眠和深度睡眠模式下关闭 TIMER0 时钟 1: 在睡眠和深度睡眠模式下开启 TIMER0 时钟
9	ADCSPDPEN	在睡眠和深度睡眠模式下 ADC 时钟使能 由软件置位或复位 0: 在睡眠和深度睡眠模式下关闭 ADC 时钟 1: 在睡眠和深度睡眠模式下开启 ADC 时钟
8	WWDGTSPDPEN	在睡眠和深度睡眠模式下 WWDGT 时钟使能 由软件置位或复位 0: 在睡眠和深度睡眠模式下关闭 WWDGT 时钟 1: 在睡眠和深度睡眠模式下开启 WWDGT 时钟
7:2	保留	必须保持复位值。
1	CMPSPDPEN	在睡眠和深度睡眠模式下 CMP 时钟使能 由软件置位或复位 0: 在睡眠和深度睡眠模式下关闭 CMP 时钟 1: 在睡眠和深度睡眠模式下开启 CMP 时钟
0	SYSCFGSPDPEN	在睡眠和深度睡眠模式下系统配置时钟使能

由软件置位或复位

0: 在睡眠和深度睡眠模式下关闭系统配置时钟

1: 在睡眠和深度睡眠模式下开启系统配置时钟

4.3.13. 控制寄存器 1 (RCU_CTL1)

地址偏移: 0x70

复位值: 0x0000 0008, 由备份寄存器复位电路复位

该寄存器可以按字节 (8 位)、半字 (16 位) 或字 (32 位) 访问。

注意: 控制寄存器 1 (RCU_CTL1) 的 LXTALEN, LXTALBPS, RTCSRC 和 RTCEN 位仅在备份寄存器 (V_{CORE_STB}) 复位后才清 0。只有在电源控制寄存器 (PMU_CTL0) 中的 BKPWEN 位置 1 后才能对这些位进行改动。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留						L S C K O U	L S C K O U	保留						B K P R S T	
						T S E L	T E N								
						rw	rw							rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R T C E N	保留					R T C S R C[1:0]		L X T A L S T B R S T	L C K M D	L C K M E N	保留	L X T A L D R I	L X T A L B P S	L X T A L S T B	L X T A L E N
rw						rw		rw	rw	rw	保留	rw	rw	r	rw

位/位域	名称	描述
31:26	保留	必须保持复位值。
25	L _{SCKOUTSEL}	低速时钟输出选择 0: IRC32K 1: LXTAL
24	L _{SCKOUTEN}	低速时钟输出使能 0: 低速时钟输出失能 1: 低速时钟输出使能
23:17	保留	必须保持复位值。
16	B _{KPRST}	备份寄存器复位 由软件置位或复位 0: 无作用 1: 复位备份寄存器
15	R _{TCEN}	RTC 时钟使能 由软件置位或复位 0: 关闭 RTC 时钟 1: 开启 RTC 时钟
14:10	保留	必须保持复位值。

9:8	RTCSRC[1:0]	<p>RTC时钟入口选择</p> <p>软件置位或清除来控制RTC时钟源。</p> <p>00: 没有时钟</p> <p>01: 选择LXTAL时钟作为RTC时钟源</p> <p>10: 选择IRC32K时钟作为RTC时钟源</p> <p>11: 选择HXTAL时钟32分频作为RTC时钟源</p>
7	LXTALSTBRST	<p>低速晶体振荡器稳定标志位复位</p> <p>0: 低速晶体振荡器稳定标志位不复位</p> <p>1: 低速晶体振荡器稳定标志位复位</p>
6	LCKMD	<p>LXTAL 时钟故障检测</p> <p>当外部 32 kHz 振荡器 (LXTAL) 上的时钟安全系统检测到故障, 硬件置位。当 LCKMEN 或 LXTALEN 关闭时, 该位清零。</p> <p>0: LXTAL (32 kHz 振荡器) 上未检测到故障</p> <p>1: 在LXTAL (32 kHz 振荡器) 上检测到故障</p>
5	LCKMEN	<p>LXTAL 时钟监视器使能</p> <p>0: 禁止 LXTAL 时</p> <p>1: 使能 LXTAL 时钟监视器</p> <p>通过软件设置, 启用 LXTAL (32 kHz 振荡器) 上的时钟安全系统。LXTALEN 必须在 LXTAL 已启用 (LXTALEN 位已启用) 和就绪 (LXTALSTB 标志由硬件设置)。</p>
4	保留	<p>必须保持复位值。</p>
3	LXTALDRI	<p>LXTAL驱动能力</p> <p>软件置位或清除。当复位备份寄存器时, 会重装载缺省值。</p> <p>0: 低驱动能力</p> <p>1: 高驱动能力 (复位后的缺省值)</p> <p>注意: LXTALDRI在旁路模式下无效</p>
2	LXTALBPS	<p>LXTAL旁路模式使能</p> <p>软件置1和清0。</p> <p>0: 禁止LXTAL旁路模式</p> <p>1: 使能LXTAL旁路模式</p>
1	LXTALSTB	<p>外部低速振荡器稳定状态位</p> <p>硬件置1来指示LXTAL输出时钟是否稳定待用。</p> <p>0: LXTAL未稳定</p> <p>1: LXTAL已稳定</p>
0	LXTALEN	<p>LXTAL使能</p> <p>软件置1和清0。</p> <p>0: 关闭LXTAL</p> <p>1: 开启LXTAL</p>

4.3.14. 复位源/时钟寄存器 (RCU_RSTSCK)

地址偏移: 0x74

复位值: 0xXX00 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
LP RSTF	WWDGT RSTF	FWDGT RSTF	SW RSTF	POR RSTF	EP RSTF	保留	RSTFC	OBLR RSTF	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													IRC32K STB	IRC32K EN	
													r	rw	

位/位域	名称	描述
31	LPRSTF	低功耗复位标志位 当产生深度睡眠或待机重置时, 由硬件设置。 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	保留	必须保持复位值。
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	OBLRSTF	选项字节重载复位标志位 当有选项字节重载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: 无选项字节重载复位发生 1: 发生选项字节重载复位
22:2	保留	必须保持复位值。.
1	IRC32KSTB	IRC32K时钟稳定状态位 该位由硬件置1指示IRC32K输出时钟是否稳定待用。 0: IRC32K时钟未稳定 1: IRC32K时钟已稳定
0	IRC32KEN	IRC32K时钟使能 软件置1和清0。 0: 关闭IRC32K时钟 1: 开启IRC32K时钟

4.3.15. 配置寄存器 1 (RCU_CFG1)

地址偏移: 0x8C

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
I2SSEL[1:0]	保留		ADCPSC[3:0]			ADCSEL	保留		I2C1SEL[1:0]	I2C0SEL[1:0]		USART0SEL[1:0]			
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。

15:14	I2SSEL[1:0]	<p>I2S时钟源选择</p> <p>由软件置1或清0。</p> <p>00: I2S时钟选择CK_SYS</p> <p>01: 保留</p> <p>10: I2S钟选择CK_IRC48MDIV_PER</p> <p>11: I2S钟选择I2S_CKIN</p>
13	保留	必须保持复位值。
12:9	ADCPSC[3:0]	<p>ADC时钟预分频选择。</p> <p>这些位是由软件编写的并定义ADC时钟预分频。由软件设置和清除。</p> <p>0000: 输出 ADC 时钟不分频</p> <p>0001: 输出 ADC 时钟 2 分频</p> <p>0010: 输出 ADC 时钟 4 分频</p> <p>0011: 输出 ADC 时钟 6 分频</p> <p>0100: 输出 ADC 时钟 8 分频</p> <p>0101: 输出 ADC 时钟 10 分频</p> <p>0110: 输出 ADC 时钟 12 分频</p> <p>0111: 输出 ADC 时钟 16 分频</p> <p>1000: 输出 ADC 时钟 32 分频</p> <p>1001: 输出 ADC 时钟 64 分频</p> <p>1010: 输出 ADC 时钟 128 分频</p> <p>1011: 输出 ADC 时钟 256 分频</p> <p>其他: 保留</p>
8	ADCSEL	<p>ADC时钟源选择</p> <p>由软件置1或清0。</p> <p>0: 选择CK_SYS时钟作为时钟源</p> <p>1: 选择CK_IRC48MDIV_PER作为时钟源</p>
7:6	保留	必须保持复位值。
5:4	I2C1SEL[1:0]	<p>I2C1时钟源选择</p> <p>由软件置1或清0。</p> <p>00: 选择CK_APB时钟作为时钟源</p> <p>01: 选择CK_SYS作为时钟源</p> <p>10/11: 选择CK_IRC48MDIV_PER作为时钟源</p>
3:2	I2C0SEL[1:0]	<p>I2C0时钟源选择</p> <p>由软件置1或清0。</p> <p>00: 选择CK_APB时钟作为时钟源</p> <p>01: 选择CK_SYS作为时钟源</p> <p>10/11: 选择CK_IRC48MDIV_PER作为时钟源</p>
1:0	USART0SEL[1:0]	<p>USART0时钟源选择</p> <p>由软件置1或清0。</p> <p>00: 选择CK_APB时钟作为时钟源</p> <p>01: 选择CK_SYS作为时钟源</p>

10: 选择CK_IRC48MDIV_PER作为时钟源

11: 选择CK_LXTAL作为时钟源

5. 中断 / 事件控制器 (EXTI)

5.1. 简介

Cortex®-M23集成了嵌套式矢量型中断控制器 (Nested Vectored Interrupt Controller (NVIC)) 来实现高效的异常和中断处理。NVIC实现了低延迟的异常和中断处理, 以及电源管理控制。它和内核是紧密耦合的。更多关于NVIC的说明请参考《Cortex®-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。

5.2. 主要特征

- Cortex®-M23系统异常;
- 多达39种可屏蔽的外设中断;
- 2位中断优先级配置位——共提供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。

5.3. 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状态压栈并行进行的, 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状态所带来的开销。[表 5-1. Cortex®-M23 中的 NVIC 异常类型](#)和[表 5-2. 中断向量表](#)列出了所有的异常类型。

表 5-1. 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
-	4-10	-	0x0000_0010 – 0x0000_002B	保留

异常类型	向量编号	优先级 (a)	向量地址	描述
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
-	12-13	-	0x0000_0030 – 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

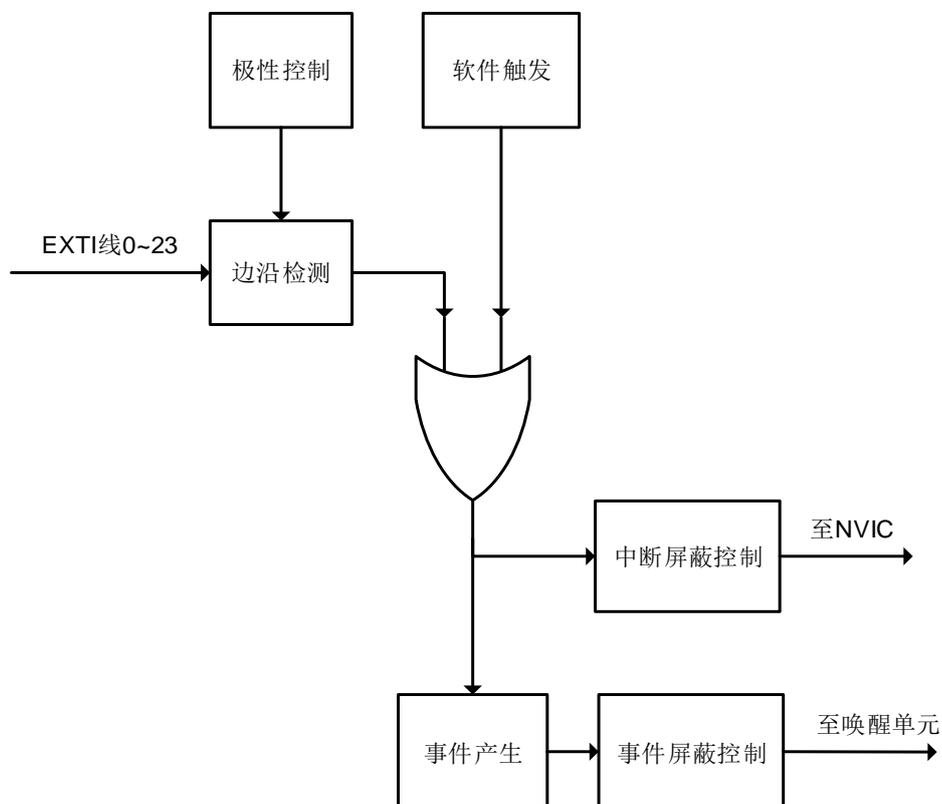
表 5-2. 中断向量表

中断编号	向量编号	外设中断描述	向量地址
IRQ 0	16	窗口看门狗中断	0x0000_0040
IRQ 1	17	连接到 EXTI 线的 RTC 时间戳中断	0x0000_0044
IRQ 2	18	保留	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 中断	0x0000_0054
IRQ 6	22	EXTI 线 1 中断	0x0000_0058
IRQ 7	23	EXTI 线 2 中断	0x0000_005C
IRQ 8	24	EXTI 线 3 中断	0x0000_0060
IRQ 9	25	EXTI 线 4 中断	0x0000_0064
IRQ 10	26	DMA 通道 0 全局中断	0x0000_0068
IRQ 11	27	DMA 通道 1 全局中断	0x0000_006C
IRQ 12	28	DMA 通道 2 全局中断	0x0000_0070
IRQ 13	29	ADC 中断	0x0000_0074
IRQ 14	30	USART0 全局中断	0x0000_0078
IRQ 15	31	USART1 全局中断	0x0000_007C
IRQ 16	32	USART2 全局中断	0x0000_0080
IRQ 17	33	I2C0 事件中断	0x0000_0084
IRQ 18	34	I2C0 错误中断	0x0000_0088
IRQ 19	35	I2C1 事件中断	0x0000_008C
IRQ 20	36	I2C1 错误中断	0x0000_0090
IRQ 21	37	SPI0 全局中断	0x0000_0094
IRQ 22	38	SPI1 全局中断	0x0000_0098
IRQ 23	39	连接到 EXTI 线的 RTC 闹钟中断	0x0000_009C
IRQ 24	40	EXTI 线[9:5]中断	0x0000_00A0
IRQ 25	41	TIMER0 触发与通道换相中断或 TIMER0 更新中断或 TIMER0 中止中断	0x0000_00A4
IRQ 26	42	TIMER0 捕获比较中断	0x0000_00A8
IRQ 27	43	TIMER2 全局中断	0x0000_00AC
IRQ 28	44	TIMER13 全局中断	0x0000_00B0
IRQ 29	45	TIMER15 全局中断	0x0000_00B4
IRQ 30	46	TIMER16 全局中断	0x0000_00B8

中断编号	向量编号	外设中断描述	向量地址
IRQ 31	47	EXTI 线[15:10]中断	0x0000_00BC
IRQ 32	48	保留	0x0000_00C0
IRQ 33	49	DMA MUX 中断	0x0000_00C4
IRQ 34	50	连接到 EXTI 线的 CMP0 输出中断	0x0000_00C8
IRQ 35	51	连接到 EXTI 线的 CMP1 输出中断	0x0000_00CC
IRQ 36	52	连接到 EXTI 线的 I2C0 唤醒中断	0x0000_00D0
IRQ 37	53	连接到 EXTI 线的 I2C1 唤醒中断	0x0000_00D4
IRQ 38	54	连接到 EXTI 中断线的 USART0 唤醒中断	0x0000_00D8

5.4. 外部中断及事件 (EXTI) 结构框图

图 5-1. EXTI 结构框图



5.5. 外部中断及事件功能概述

EXTI包含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 根线详情请参考[表 5-3. EXTI 触发源](#)。通过配置 SYSCFG 模块的 SYSCFG_EXTISSx 寄存器，所有的 GPIO 管脚都可以被选作 EXTI 的触发源，具体细节请参考[系统配置寄存器](#)。

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管脚电平翻转或者RTC闹钟。

硬件触发

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

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

软件触发

按照如下步骤软件也可以触发 EXTI 中断或事件：

1. 配置对应的 EXTI_INTEN 或 EXTI_EVEN 位使能中断或事件；
2. 配置 EXTI_SWIEV 寄存器的对应 SWIEVx 位，对应的 PD 位将立刻被置 1，使能的中断或事件将被触发，软件需要响应该中断或事件并清除相应 PDx 位。

表 5-3. EXTI 触发源

EXTI 线编号	触发源
0	PA0 / PB0 / PD0 / PF0
1	PA1 / PB1 / PD1 / PF1
2	PA2 / PB2 / PD2 / PF2
3	PA3 / PB3 / PD3 / PF3
4	PA4 / PB4
5	PA5 / PB5
6	PA6 / PB6 / PC6
7	PA7 / PB7 / PC7
8	PA8 / PB8
9	PA9 / PB9
10	PA10 / PB10
11	PA11 / PB11
12	PA12 / PB12
13	PA13 / PB13 / PC13
14	PA14 / PB14 / PC14
15	PA15 / PB15 / PC15
16	RTC 闹钟
17	RTC 时间戳

EXTI 线编号	触发源
18	CMP0 输出
19	CMP1 输出
20	I2C0 唤醒
21	I2C1 唤醒
22	USART0 唤醒
23	LXTALCS

5.6. EXTI 寄存器

EXTI基地址：0x4001 0400

5.6.1. 中断使能寄存器（EXTI_INTEN）

地址偏移：0x00

复位值：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								INTEN23	INTEN22	INTEN21	INTEN20	INTEN19	INTEN18	INTEN17	INTEN16
								r/w							
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
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

位/位域	名称	描述
31:24	保留	必须保持复位值。
23:0	INTENx	中断使能位x (x = 0..23) 0: 第x中断被禁止 1: 第x中断被使能

5.6.2. 事件使能寄存器（EXTI_EVEN）

地址偏移：0x04

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								EVEN23	EVEN22	EVEN21	EVEN20	EVEN19	EVEN18	EVEN17	EVEN16
								r/w							
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
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

位/位域	名称	描述
31:24	保留	必须保持复位值。
23:0	EVENx	事件使能位x (x = 0..23) 0: 第x事件被禁止 1: 第x事件被使能

5.6.3. 上升沿触发使能寄存器 (EXTI_RTEN)

地址偏移: 0x08

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								RTEN23	RTEN22	RTEN21	RTEN20	RTEN19	RTEN18	RTEN17	RTEN16
								r/w							
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
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

位/位域	名称	描述
31:24	保留	必须保持复位值。
23:0	RTENx	上升沿触发使能 (x = 0..23) 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)

5.6.4. 下降沿触发使能寄存器 (EXTI_FTEN)

地址偏移: 0x0C

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								FTEN23	FTEN22	FTEN21	FTEN20	FTEN19	FTEN18	FTEN17	FTEN16
								r/w							
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
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

位/位域	名称	描述
31:24	保留	必须保持复位值。
23:0	FTENx	下降沿触发使能 (x = 0..23) 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)

5.6.5. 软件中断事件寄存器 (EXTI_SWIEV)

地址偏移: 0x10

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								SWIEV23	SWIEV22	SWIEV21	SWIEV20	SWIEV19	SWIEV18	SWIEV17	SWIEV16
								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
31:24	保留	必须保持复位值。
23:0	SWIEVx	中断/事件软件触发x (x = 0..23) 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

5.6.6. 挂起寄存器 (EXTI_PD)

地址偏移: 0x14

复位值: 0xXXXX XXXX, X表示未定义。

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								PD23	PD22	PD21	PD20	PD19	PD18	PD17	PD16
								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															

位/位域	名称	描述
31:24	保留	必须保持复位值。
23:0	PDx	中断挂起状态x (x = 0..23) 0: EXTI线x没有被触发 1: EXTI线x被触发, 对这些位写1, 可将其清0。

6. 通用和备用输入/输出接口（GPIO 和 AFIO）

6.1. 简介

GD32C2x1 产品最多可支持 45 个通用 I/O 引脚（GPIO），分别为 PA0 ~ PA15, PB0 ~ PB15, PC6 ~ PC7, PC13 ~ PC15, PD0 ~ PD3, 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。片上设备 GPIO 引脚的外部中断由 EXTI 模块的寄存器控制和配置。

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。

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

6.2. 主要特征

- 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；
- 可编程的边沿触发外部中断-由 EXTI 寄存器配置；
- 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。

6.3. 功能说明

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

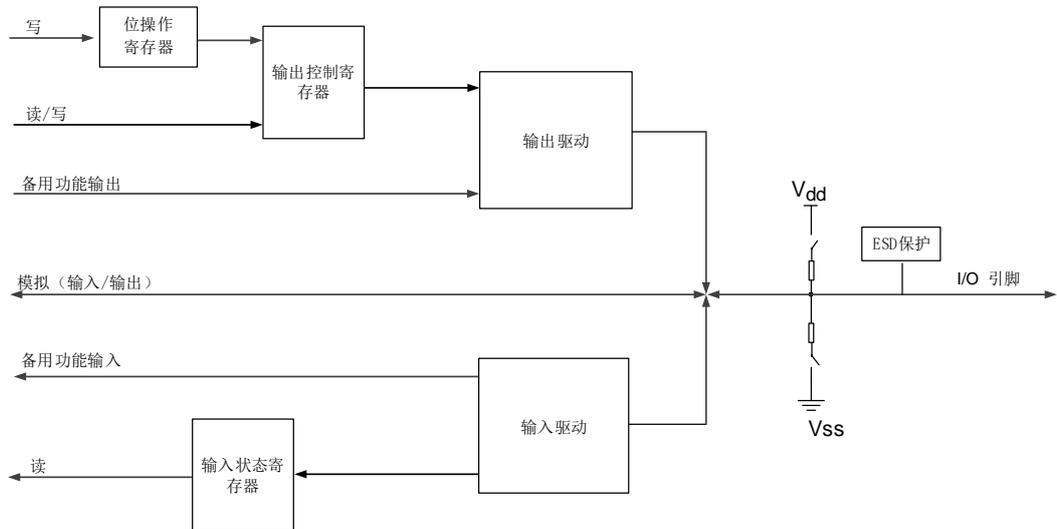
表 6-1. GPIO 配置表

PAD TYPE		CTLy	OMy	PUDy
GPIO 输入	X	00	X	悬空
				上拉
				下拉
				00
				01
				10

PAD TYPE			CTLy	OMy	PUDy
GPIO 输出	推挽	悬空	01	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	10	X	00
		上拉			01
		下拉			10
AFIO 输出	推挽	悬空	10	0	00
		上拉			01
		下拉			10
	开漏	悬空		1	00
		上拉			01
		下拉			10
ANALOG	X	X	11	X	XX

图6-1. GPIO端口位的基本结构为标准I/O端口位的基本结构图。

图 6-1. GPIO 端口位的基本结构



6.3.1. GPIO 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，串行线调试为输入 PU/PD 模式。

PA14: SWCLK为PD下拉模式

PA13: SWDIO为PU上拉模式

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

到端口输入状态寄存器（GPIOx_ISTAT）。

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

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

6.3.2. 外部中断及事件

所有的端口都有外部中断的能力，如果想使用端口的外部中断功能，需要配置为输入模式。

6.3.3. 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，该端口用作外设备备用功能。通过配置GPIO备用功能选择寄存器（GPIOx_AFSELY(y=0..1)），每个端口可以配置16个备用功能。端口备用功能分配的详细介绍见芯片数据手册。

6.3.4. 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，DAC或附加功能时，引脚必须配置成模拟模式。当引脚用作RTC、WKUPx和振荡器附加功能时，端口类型通过相关的RTC、PMU和RCU寄存器自动设置。当附加功能禁用时，这些端口可用作普通GPIO。

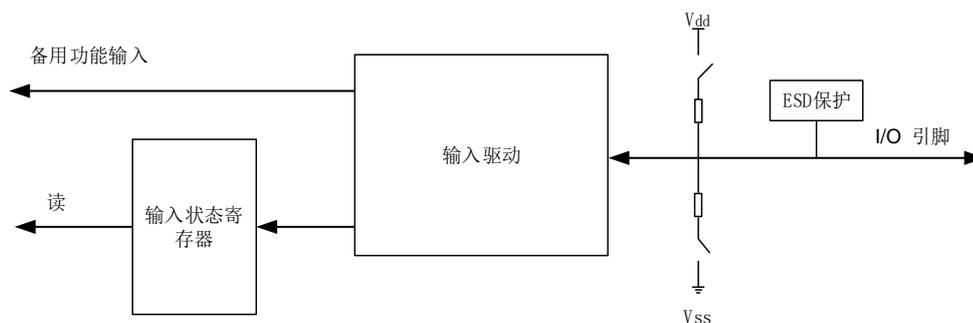
6.3.5. 输入配置

当GPIO引脚配置为输入时：

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

[图6-2. 输入配置的基本结构](#)是I/O引脚的输入配置。

图 6-2. 输入配置的基本结构



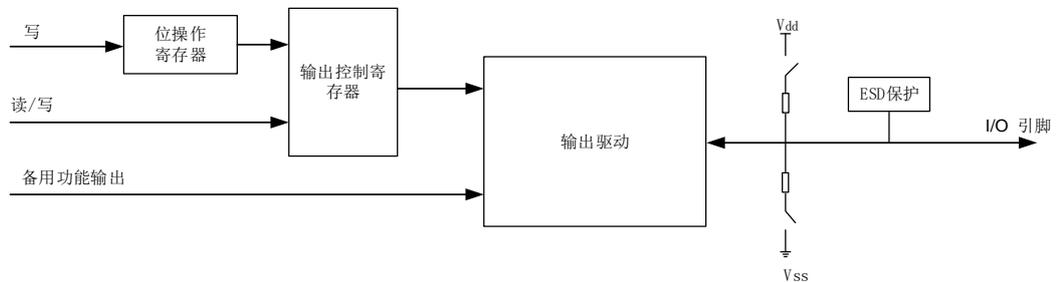
6.3.6. 输出配置

当GPIO配置为输出时：

- 施密特触发输入使能；
- 可选的弱上拉和下拉电阻；
- 开漏模式：输出控制寄存器设置为“0”时，相应引脚输出低电平；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，相应引脚输出低电平；输出控制寄存器设置为“1”，相应引脚输出高电平；
- 在推挽模式下，对端口输出控制寄存器的读访问将返回上次写入的值；
- 在开漏模式下，对端口输入状态寄存器的读访问将返回I/O的状态。

[图 6-3. 输出配置的基本结构](#)是 I/O 端口的输出配置。

图 6-3. 输出配置的基本结构



6.3.7. 模拟配置

当GPIO引脚用于模拟模式时：

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

[图 6-4. 模拟配置的基本结构](#)是I/O端口的模拟高阻配置。

图 6-4. 模拟配置的基本结构



6.3.8. 备用功能（AF）配置

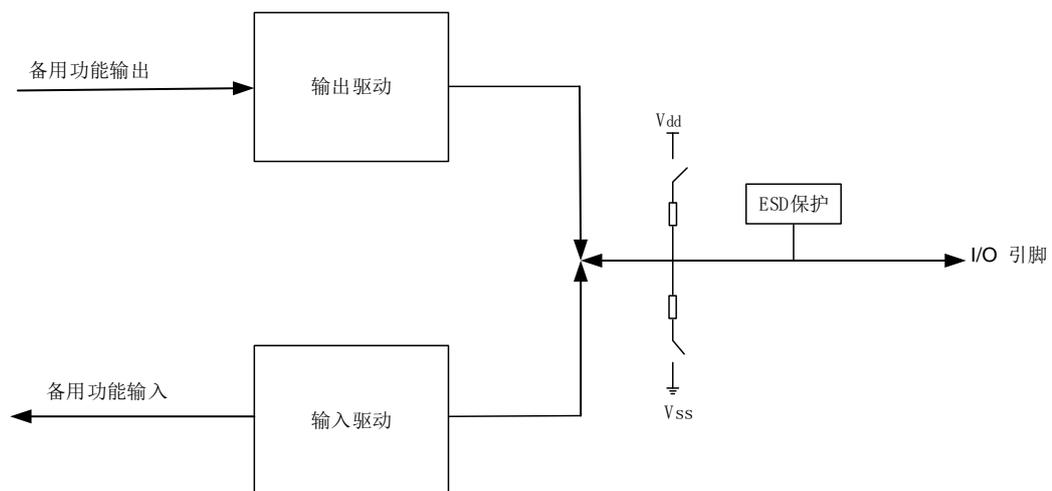
为了适应不同的器件封装，GPIO端口支持软件配置将一些备用功能应用到其他引脚上。

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；
- 对端口输入状态寄存器进行读操作，将获得I/O口的状态；
- 对端口输出控制寄存器进行读操作，将返回上次写入的值。

[图6-5. 备用功能配置的基本结构](#)是I/O端口备用功能配置图。

图 6-5. 备用功能配置的基本结构



6.3.9. 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。当特定LOCK序列写到位于GPIOx_LOCK寄存器的LKK位上，并且LKy被置位，那么对应的端口配置直到下一次复位前将不能改变。建议在电源驱动模块驱动的配置时使用锁定功能。

6.3.10. 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。

6.3.11. HXTAL 和 LXTAL GPIO 功能配置

当HXTALEN或LXTALEN为0b0时，HXTAL或LXTAL振荡器引脚可以用作GPIO功能。当HXTALEN或LXTALEN为0b1时，HXTAL或LXTAL引脚由振荡器控制，GPIO不能控制HXTAL或LXTAL引脚。

当HXTALBPS或LXTALBPS为0b1时，表示外部晶振（HXTAL或LXTAL）时钟被旁路。HXTAL或LXTAL振荡器输出引脚可用作GPIO功能，输入引脚可用作外部时钟输入。

在引脚为 48 的器件中，HXTAL 或 LXTAL 振荡器的输入和输出引脚是独立的（参考 FMC 选项字节的 HXTAL_REMAP 位）。在引脚数小于 48 的器件中，HXTAL 或 LXTAL 振荡器的输入引脚 OSCX_IN 和输出引脚 OSCX_OUT 是共用的，同时只能使用其中一个。

6.3.12. 复位引脚 GPIO 功能配置

设置 FMC 选项字节的 NRST_MDSEL[1:0]位可以配置 PF2 作为 GPIO 使用。当 PF2 用作 GPIO 时，复位引脚将不再能够触发设备复位。

6.3.13. 引脚（PA14）GPIO 功能配置

设置 FMC 选项字节的 nBOOT0 位可以配置 PA14 作为 GPIO 使用。

6.4. GPIO 寄存器

GPIOA基地址: 0x4800 0000

GPIOB基地址: 0x4800 0400

GPIOC基地址: 0x4800 0800

GPIOD基地址: 0x4800 0C00

GPIOF基地址: 0x4800 1400

6.4.1. 端口控制寄存器 (GPIOx_CTL, x=A..D, F)

地址偏移: 0x00

复位值: 端口 A 0xE8FF FFFF; 其他端口 0xFFFF FFFF。

该寄存器可只能按字 (32 位) 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 01: GPIO输出模式 10: 备用功能模式 11: 模拟模式（复位值输入和输出）

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

地址偏移：0x04

复位值：0x0000 0000

该寄存器可只能按字（32 位）访问



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

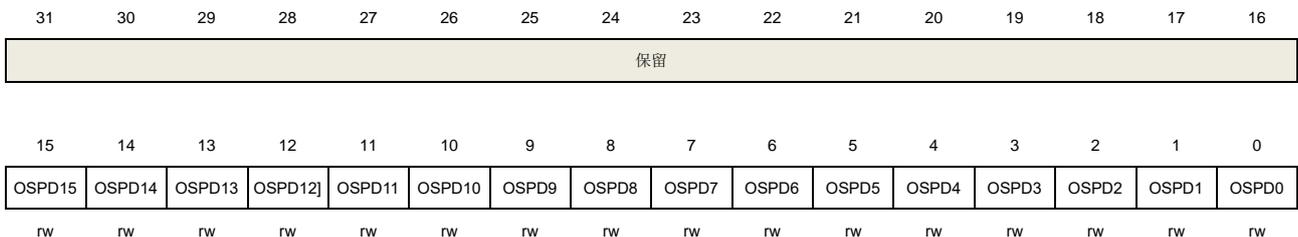
		参考OM0的描述
5	OM5	Pin 5输出模式位 该位由软件置位和清除。 参考OM0的描述
4	OM4	Pin 4输出模式位 该位由软件置位和清除。 参考OM0的描述
3	OM3	Pin 3输出模式位 该位由软件置位和清除。 参考OM0的描述
2	OM2	Pin 2输出模式位 该位由软件置位和清除。 参考OM0的描述
1	OM1	Pin 1输出模式位 该位由软件置位和清除。 参考OM0的描述
0	OM0	Pin 0输出模式位 该位由软件置位和清除。 0: 输出推挽模式（复位值） 1: 输出开漏模式

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

地址偏移：0x08

复位值：端口 A 0x0000 2000；其他端口 0x0000 0000。

该寄存器可只能按字（32 位）访问



位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	OSPDy	Pin y (y=0..15) 输出最大速度位 该位由软件置位和清除。 0: 输出速度 0 (10MHz) 1: 输出速度 1 (60MHz)

6.4.4. 端口上拉/下拉寄存器（GPIOx_PUD，x=A..D，F）

地址偏移：0x0C

复位值：端口 A 0x2400 0000；其他端口 0x0000 0000。

该寄存器可只能按字（32 位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PUD15[1:0]		PUD14[1:0]		PUD13[1:0]		PUD12[1:0]		PUD11[1:0]		PUD10[1:0]		PUD9[1:0]		PUD8[1:0]	
rw		rw		rw		rw		rw		rw		rw		rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PUD7[1:0]		PUD6[1:0]		PUD5[1:0]		PUD4[1:0]		PUD3[1:0]		PUD2[1:0]		PUD1[1:0]		PUD0[1:0]	
rw		rw		rw		rw		rw		rw		rw		rw	

位/位域	名称	描述
31:30	PUD15[1:0]	Pin 15 上拉或下拉位 该位由软件置位和清除。 参照 PUD0[1:0] 的描述
29:28	PUD14[1:0]	Pin 14 上拉或下拉位 该位由软件置位和清除。 参照 PUD0[1:0] 的描述
27:26	PUD13[1:0]	Pin 13 上拉或下拉位 该位由软件置位和清除。 参照 PUD0[1:0] 的描述
25:24	PUD12[1:0]	Pin 12 上拉或下拉位 该位由软件置位和清除。 参照 PUD0[1:0] 的描述
23:22	PUD11[1:0]	Pin 11 上拉或下拉位 该位由软件置位和清除。 参照 PUD0[1:0] 的描述
21:20	PUD10[1:0]	Pin 10 上拉或下拉位 该位由软件置位和清除。 参照 PUD0[1:0] 的描述
19:18	PUD9[1:0]	Pin 9 上拉或下拉位 该位由软件置位和清除。 参照 PUD0[1:0] 的描述
17:16	PUD8[1:0]	Pin 8 上拉或下拉位 该位由软件置位和清除。 参照 PUD0[1:0] 的描述
15:14	PUD7[1:0]	Pin 7 上拉或下拉位 该位由软件置位和清除。

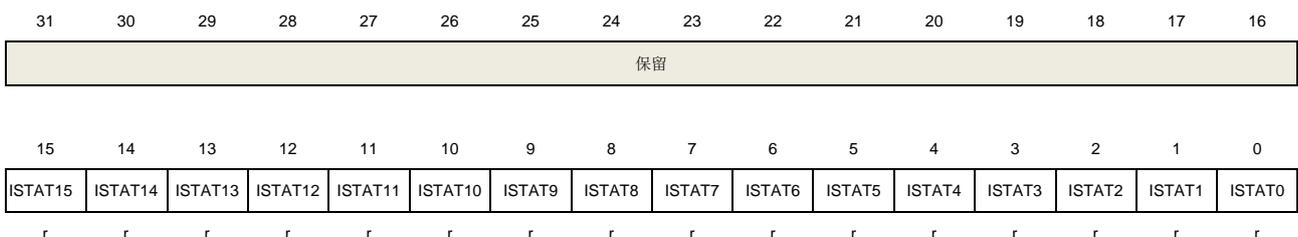
		参照PUD0[1:0]的描述
13:12	PUD6[1:0]	Pin 6上拉或下拉位 该位由软件置位和清除。 参照PUD0[1:0]的描述
11:10	PUD5[1:0]	Pin 5上拉或下拉位 该位由软件置位和清除。 参照PUD0[1:0]的描述
9:8	PUD4[1:0]	Pin 4上拉或下拉位 该位由软件置位和清除。 参照PUD0[1:0]的描述
7:6	PUD3[1:0]	Pin 3上拉或下拉位 该位由软件置位和清除。 参照PUD0[1:0]的描述
5:4	PUD2[1:0]	Pin 2上拉或下拉位 该位由软件置位和清除。 参照PUD0[1:0]的描述
3:2	PUD1[1:0]	Pin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

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

地址偏移: 0x10

复位值: 0x0000 XXXX

该寄存器可只能按字 (32 位) 访问



位/位域	名称	描述
------	----	----

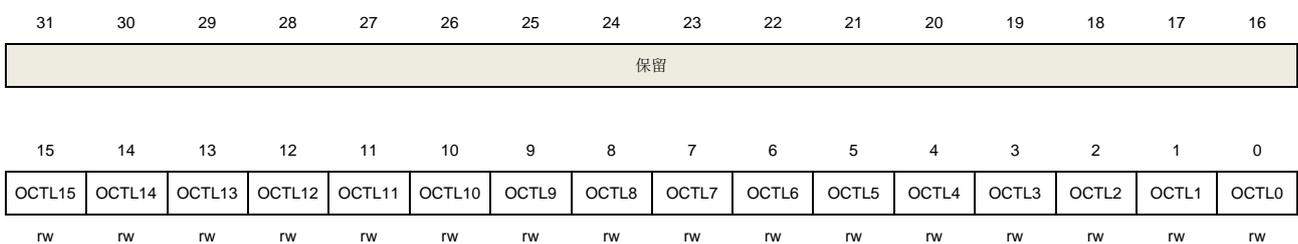
31:16	保留	必须保持复位值。
15:0	ISTATy	端口输入状态位y (y=0...15) 这些位由软件置位和清除。 0: 引脚输入信号为低电平 1: 引脚输入信号为高电平

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

地址偏移: 0x14

复位值: 0x0000 0000

该寄存器可只能按字 (32 位) 访问



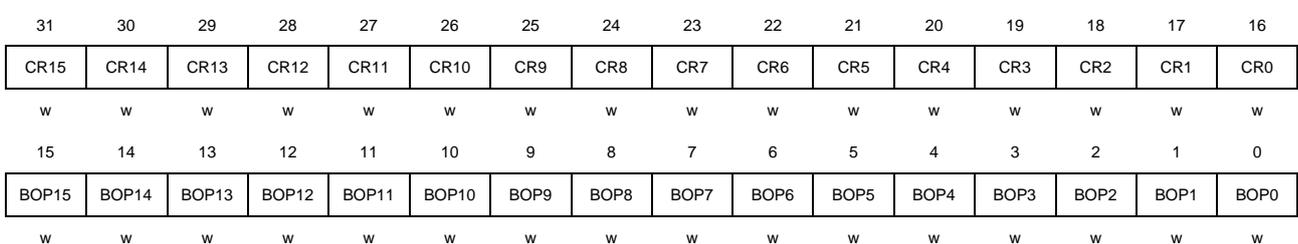
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	OCTLy	端口输出控制位y (y=0...15) 该位由软件置位和清除。 0: 引脚输出低电平 1: 引脚输出高电平

6.4.7. 端口位操作寄存器 (GPIOx_BOP, x=A..D, F)

地址偏移: 0x18

复位值: 0x0000 0000

该寄存器可只能按字 (32 位) 访问



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

1: 清除相应的OCTLy位为0

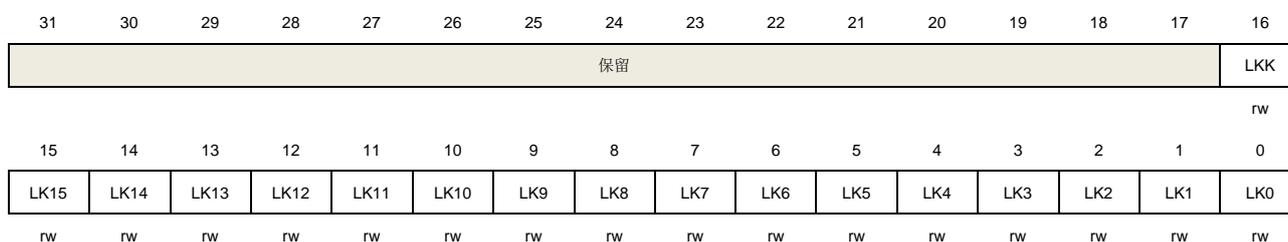
15:0 BOPy[15:0] 端口置位位y (y=0...15)
该位由软件置位和清除。
0: 相应的OCTLy位没有改变
1: 设置相应的OCTLy位为1

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

地址偏移: 0x1C

复位值: 0x0000 0000

该寄存器可只能按字 (32 位) 访问



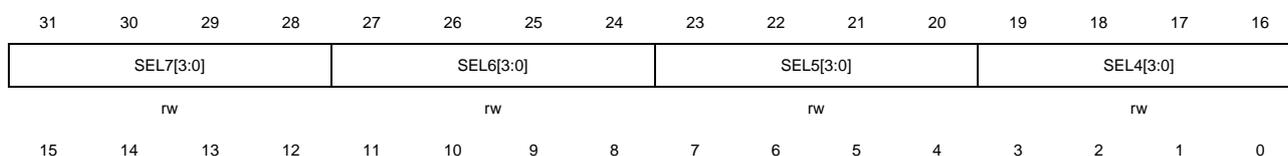
位/位域	名称	描述
31:17	保留	必须保持复位值。
16	LKK	锁定键 该位只能通过Lock Key写序列置位，始终可读。 0: GPIOx_LOCK寄存器和端口配置没有锁定 1: 直到下一次MCU复位前，GPIOx_LOCK寄存器被锁定 LOCK key写序列: 写1→写0→写1→读0→读1 注意: 在LOCK Key写序列期间，LK y (y=0...15) 的值必须保持。
15:0	LKy	端口定位位y (y=0...15) 该位由软件置位和清除。 0: 端口配置没有锁定 1: 端口配置锁定

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

地址偏移: 0x20

复位值: 0x0000 0000

该寄存器可只能按字 (32 位) 访问



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

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

地址偏移: 0x24

复位值: 0x0000 0000

该寄存器可只能按字 (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 11选择备用功能 该位由软件置位和清除。 参照SEL8[3:0]的描述
11:8	SEL10[3:0]	Pin 10选择备用功能 该位由软件置位和清除。 参照SEL8[3:0]的描述
7:4	SEL9[3:0]	Pin 9选择备用功能 该位由软件置位和清除。 参照SEL8[3:0]的描述
3:0	SEL8[3:0]	Pin 8选择备用功能 该位由软件置位和清除。 0000: 选择AF0功能 (复位值) 0001: 选择AF1功能 0010: 选择AF2功能 0011: 选择AF3功能

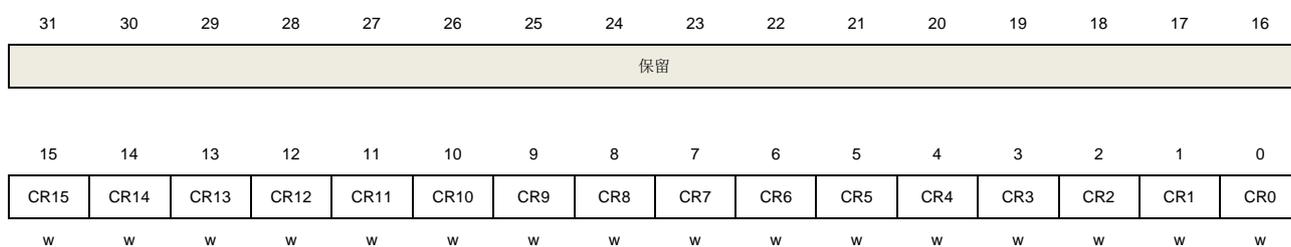
0100: 选择AF4功能
 0101: 选择AF5功能
 0110: 选择AF6功能
 0111: 选择AF7功能
 1000: 选择AF8功能
 1001: 选择AF9功能
 ...
 1111: 选择AF15功能

6.4.11. 位清除寄存器 (GPIOx_BC, x=A..D, F)

地址偏移: 0x28

复位值: 0x0000 0000

该寄存器可只能按字 (32 位) 访问



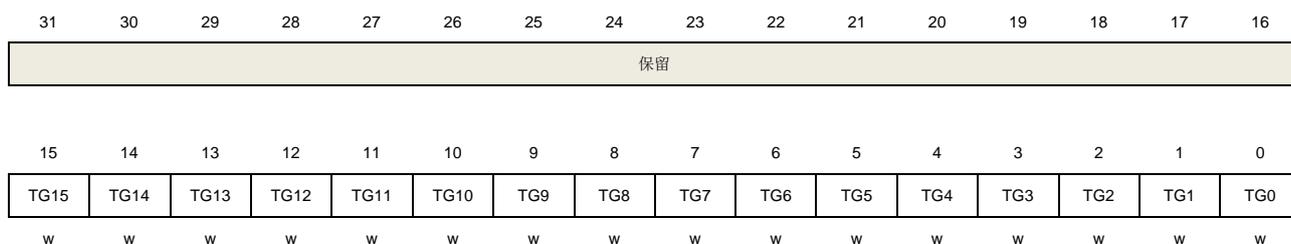
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CRy	端口清除位y (y=0...15) 该位由软件置位和清除。 0: 相应OCTLy位没有改变 1: 清除相应的OCTLy位

6.4.12. 端口位翻转寄存器 (GPIOx_TG, x=A..D, F)

地址偏移: 0x2C

复位值: 0x0000 0000

该寄存器可只能按字 (32 位) 访问



位/位域	名称	描述
------	----	----

31:16	保留	必须保持复位值
15:0	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t 1: 翻转相应的OCTLy位

7. 循环冗余校验计算单元（CRC）

7.1. 简介

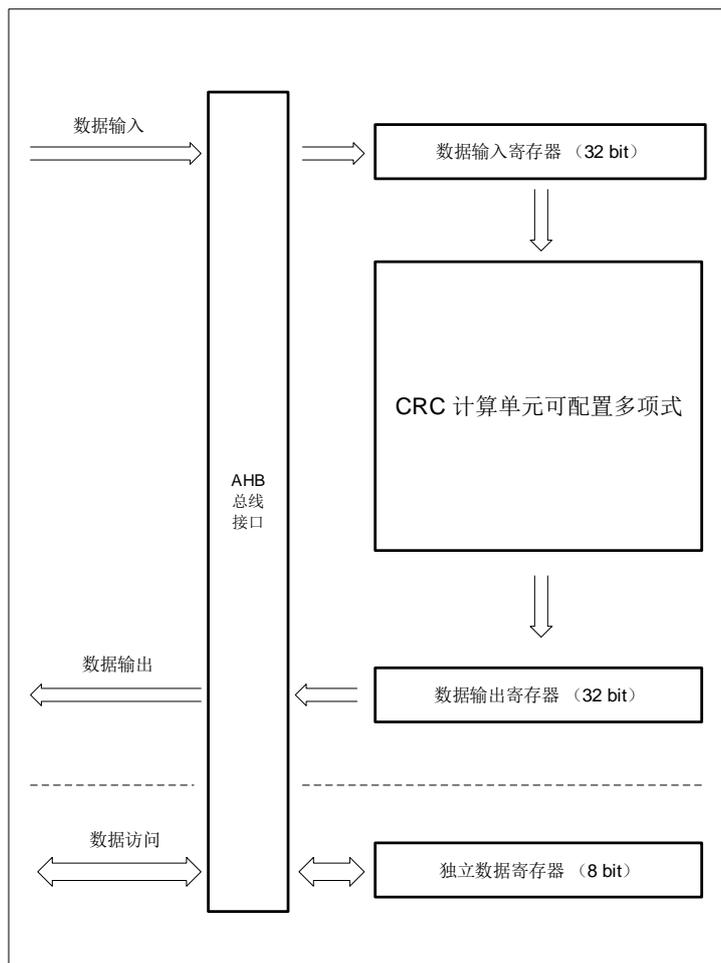
循环冗余校验码是一种用在数字网络和存储设备上的差错校验码，可以校验原始数据的偶然差错。

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。

7.2. 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 7-1. CRC 计算单元框图



7.3. 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0x3456CDEF为例：

1) 按字节逆序：

32位数据被分成四组，组内完成颠倒。逆序后的数据为：0x2C6AB3F7

2) 按半字逆序：

32位数据被分成两组，组内完成颠倒。逆序后的数据为：0x6A2CF7B3

3) 按字逆序：

32位数据被分成一组，组内完成颠倒，逆序后的数据为：0xF7B36A2C

对于输出数据来说，逆序形式为按字逆序。

例如：当REV_O=1，计算结果0x3344CCDD将被逆序成0xBB3322CC。

- 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。

7.4. CRC 寄存器

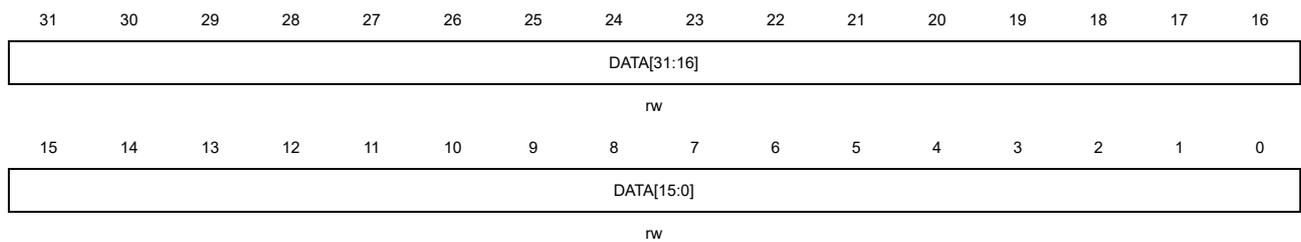
CRC基地址：0x4002 3000

7.4.1. 数据寄存器（CRC_DATA）

地址偏移：0x00

复位值：0xFFFF FFFF

该寄存器只能按字（32 位）访问。



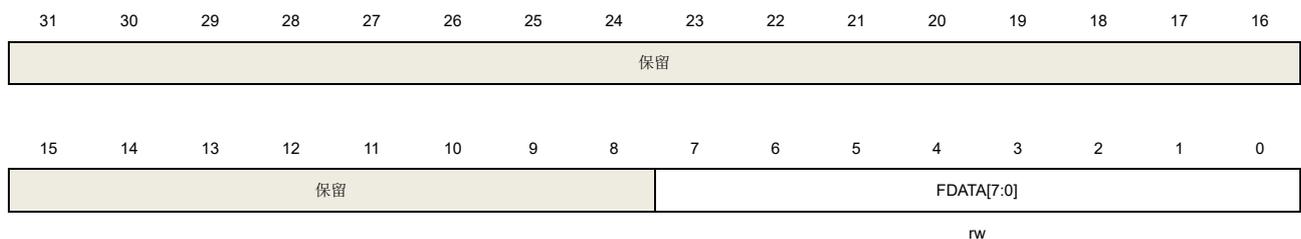
位/位域	名称	描述
31:0	DATA[31:0]	CRC 计算结果位 软件可读可写。 该寄存器用于接收待计算的新数据，直接将其写入即可。刚写入的数据不能被读出来因为读取该寄存器得到的是上次 CRC 计算的结果。

7.4.2. 独立数据寄存器（CRC_FDATA）

地址偏移：0x04

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



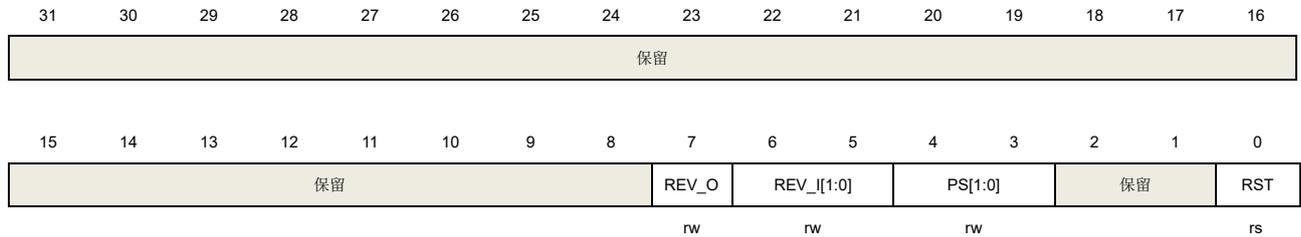
位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	FDATA[7:0]	独立数据寄存器位 软件可读可写。 这些位与 CRC 计算无关。该字节能被任何其他外设用于其他任何目的。该字节不受 CRC_CTL 寄存器的影响。

7.4.3. 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。



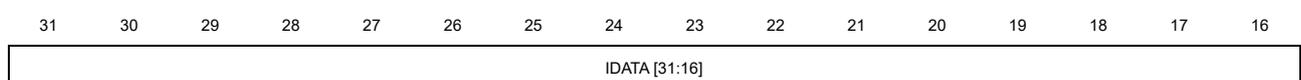
位/位域	名称	描述
31:8	保留	必须保持复位值。
7	REV_O	按位顺序翻转输出数据功能 0: 输出数据不翻转 1: 输出数据按位顺序翻转
6:5	REV_[1:0]	翻转输入数据功能 0: 输入数据不翻转 1: 输入数据按字节翻转 2: 输入数据按半字翻转 3: 输入数据按字翻转
4:3	PS[1:0]	多项式长度 0: 32 位 1: 16（POLY[15:0]用于计数）位 2: 8（POLY[7:0]用于计数）位 3: 7（POLY[6:0]用于计数）位
2:1	保留	必须保持复位值。
0	RST	软件可读写 该位用来复位 CRC_DATA 寄存器。 置位时，CRC_DATA 寄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。

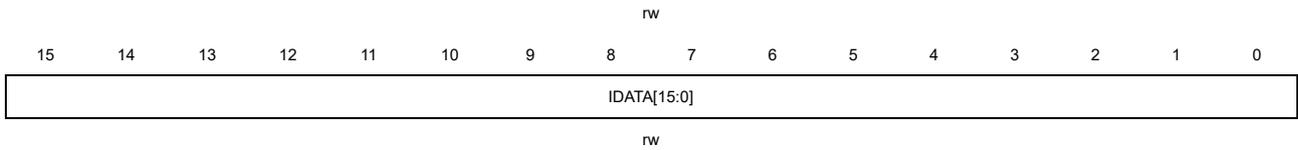
7.4.4. 初值寄存器（CRC_IDATA）

地址偏移：0x10

复位值：0xFFFF FFFF

该寄存器只能按字（32 位）访问。





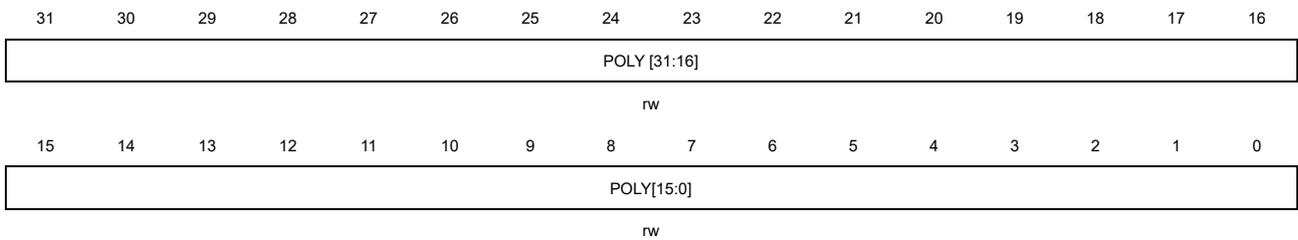
位/位域	名称	描述
31:0	IDATA[31:0]	配置 CRC 初值 CRC_CTL 寄存器的 RST 位置位后，CRC_DATA 寄存器的值将被更新为此寄存器的值。

7.4.5. 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。



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。

8. 调试 (DBG)

8.1. 简介

GD32C2x1 系列都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。

调试系统可以帮助调试者在低功耗模式下调试以及进行一些外设的调试，包括：TIMER、I2C、RTC、WWDGT 和 FWDGT。当相应的位被置 1，调试系统会在低功耗模式下提供时钟，或者为一些外设保持当前状态，这些外设包括：TIMER、I2C、RTC、WWDGT 和 FWDGT。

8.2. 串行调试接口简介

调试工具可以通过串行调试接口 (SWD) 来访问调试功能。

8.2.1. 引脚分配

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

调试引脚分配：

PA14 : SWCLK
PA13 : SWDIO

如果 SWD 调试功能没有使用，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 [通用和备用输入/输出接口 \(GPIO 和 AFIO\)](#)。

8.3. 调试保持功能描述

8.3.1. 低功耗模式调试支持

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_IRC48M 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。

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_IRC48M 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。

8.3.2. TIMER, I2C, RTC, 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RTC外设，计数器停止并进行调试；

对于WWDGT或者FWDGT外设，计数器时钟停止并进行调试。

8.4. DBG 寄存器

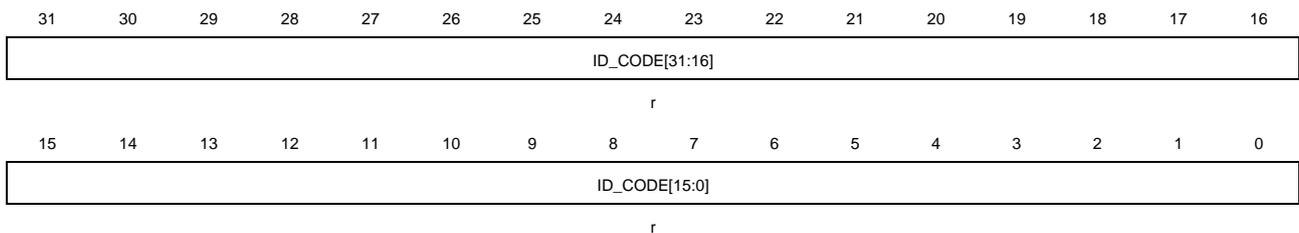
DBG基地址：0x4001 5800

8.4.1. 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



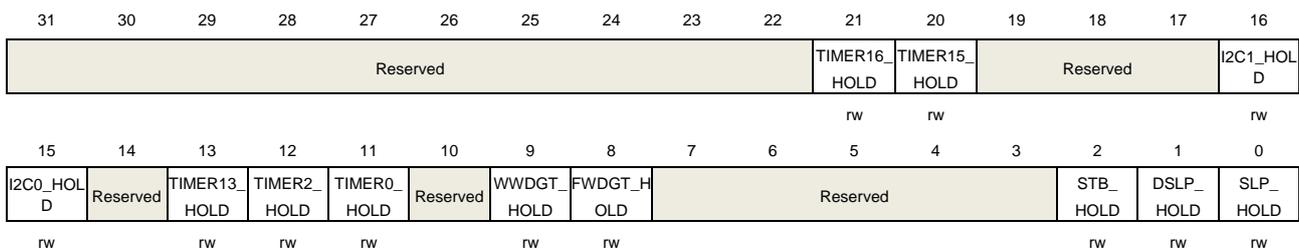
位/位域	名称	描述
31:0	ID_CODE[31:0]	DBG ID 寄存器 这些位由软件读取，这些位是不变的常数。

8.4.2. 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



位/位域	名称	描述
31:22	保留	必须保持复位值。
21	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
20	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

19:17	保留	必须保持复位值。
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	保留	必须保持复位值。
13	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
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	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
10	保留	必须保持复位值。
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_IRC48M 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_IRC48M 提供, 当退出待机模式时, 产生系统复位

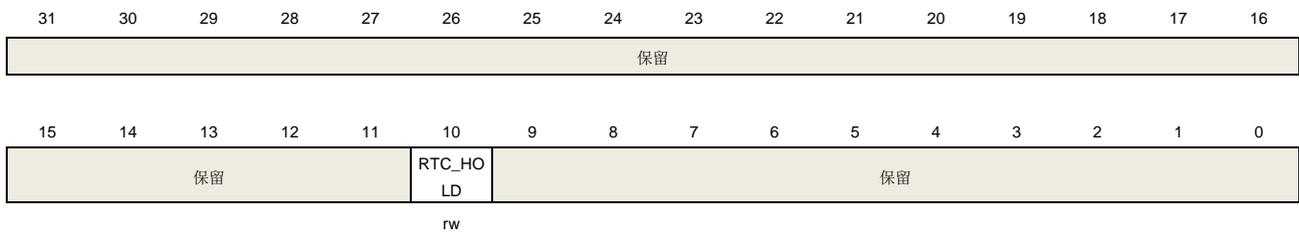
0	SLP_HOLD	睡眠模式保持位 该位由软件置位和复位 0: 无影响 1: 在睡眠模式下, AHB 时钟继续运行
---	----------	--

8.4.3. 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



位/位域	名称	描述
31:11	保留	必须保持复位值。
10	RTC_HOLD	RTC 保持位 该位由软件置位和复位 0: 无影响 1: 当内核停止时保持 RTC 计数器不变, 用于调试。
9:0	保留	必须保持复位值。

9. 直接存储器访问控制器（DMA）

9.1. 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 3 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。

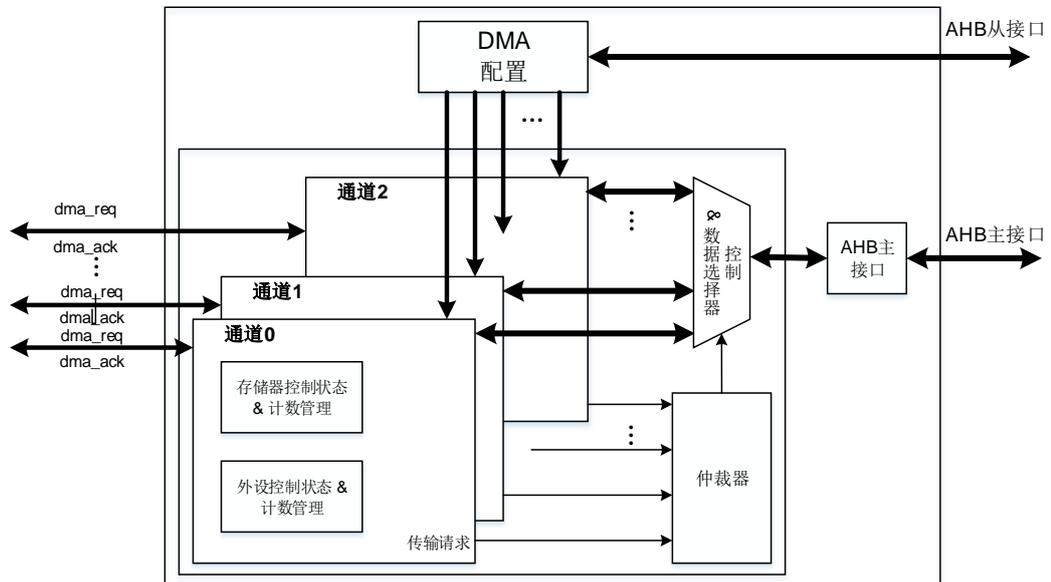
DMA 控制器和 Cortex®-M23 内核共享系统总线。当 DMA 和 CPU 访问同样的地址空间时，DMA 访问可能会阻挡 CPU 访问系统总线几个总线周期。总线矩阵中实现了循环仲裁算法来分配 DMA 与 CPU 的访问权，它可以确保 CPU 得到至少一半的系统总线带宽。

9.2. 主要特征

- 传输数据长度可编程配置，最大到65536；
- 3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；
- 每个通道连接的DMA请求不固定；
- 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。

9.3. 结构框图

图 9-1. DMA 结构框图



由 [图 9-1. 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。

9.4. 功能说明

9.4.1. 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。

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 9-1. DMA 传输操作](#)。

表 9-1. 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。

9.4.2. 外设握手

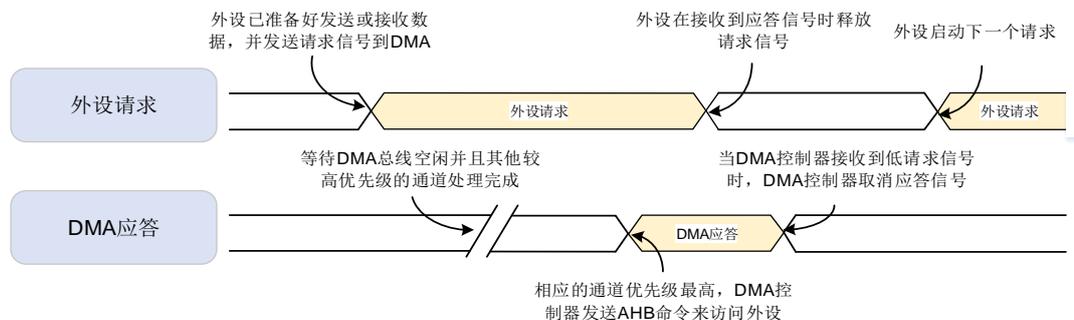
为了保证数据的有效传输，DMA 控制器中引入了外设和存储器的握手机制，包括请求信号和应

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9-2. 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 9-2. 握手机制



9.4.3. 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:

1. 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;
2. 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。

9.4.4. 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。

9.4.5. 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。

9.4.6. 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。

9.4.7. 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。

9.4.8. 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。

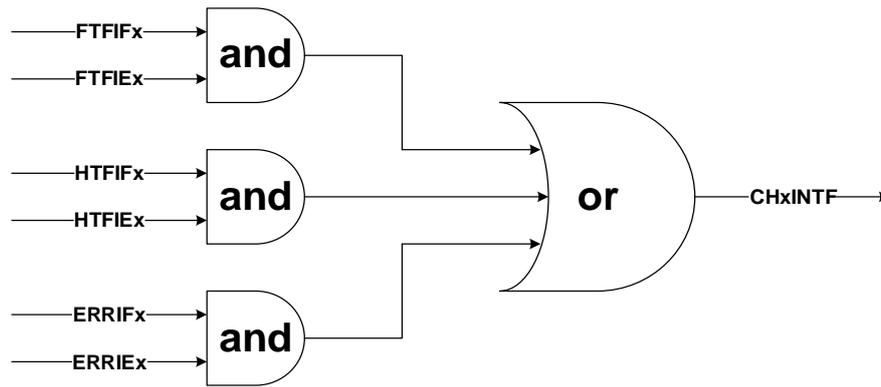
每一个中断事件在DMA_INTF寄存器中有专用的标志位，在DMA_INTC寄存器中有专用的清除位，在DMA_CHxCTL寄存器中有专用的使能位。[表9-2. 中断事件](#)描述了其对应关系。

表 9-2. 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9-3. 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 9-3. DMA 中断逻辑图



注意：“x”表示通道数（对应x = 0...2）

9.4.9. DMA 请求映射

每个 DMA 通道的请求都连接至由 DMAMUX 请求复用器的对应通道输出来转发的 AHB / APB 外设请求，参考[表 10-2. GD32C2x1 DMAMUX 请求路由输入信号映射](#)。

9.5. DMA 寄存器

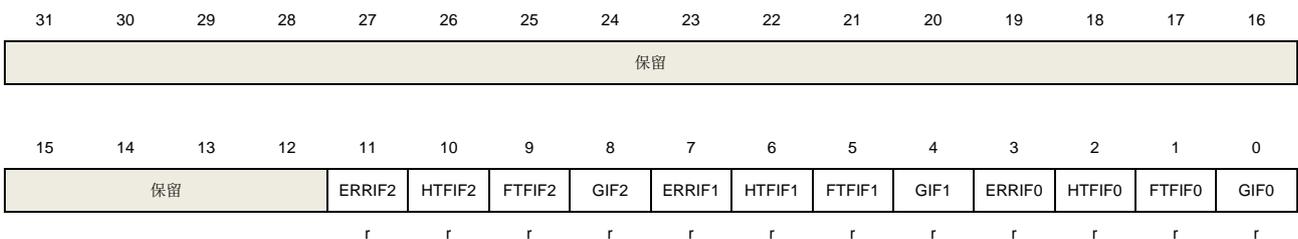
DMA 基地址：0x4002 0000

9.5.1. 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。



位/位域	名称	描述
31:12	保留	必须保持复位值。
11/7/3	ERRIFx	通道x错误标志位（x = 0...2） 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
10/6/2	HTFIFx	通道x半传输完成标志位（x = 0...2） 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
9/5/1	FTFIFx	通道x传输完成标志位（x = 0...2） 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
8/4/0	GIFx	通道x全局中断标志位（x = 0...2） 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

9.5.2. 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															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				ERRIFC2	HTFIFC2	FTFIFC2	GIFC2	ERRIFC1	HTFIFC1	FTFIFC1	GIFC1	ERRIFC0	HTFIFC0	FTFIFC0	GIFC0
				w	w	w	w	w	w	w	w	w	w	w	w

位/位域	名称	描述
31:12	保留	必须保持复位值。
11/7/3	ERRIFCx	清除通道x (x = 0...2) 的错误标志位 0: 无影响 1: 清零DMA_INTF寄存器的ERRIFx位
10/6/2	HTFIFCx	清除通道x (x = 0...2) 的半传输完成标志位 0: 无影响 1: 清零DMA_INTF寄存器的HTFIFx位
9/5/1	FTFIFCx	清除通道x (x = 0...2) 的传输完成标志位 0: 无影响 1: 清零DMA_INTF寄存器的FTFIFx位
8/4/0	GIFCx	清除通道x (x = 0...2) 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

9.5.3. 通道 x 控制寄存器 (DMA_CHxCTL)

x = 0...2, x 为通道序号

地址偏移: 0x08 + 0x14 * x

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	M2M	PRIO[1:0]	MWIDTH[1:0]	PWIDTH[1:0]	MNAGA	PNAGA	CMEN	DIR	ERRIE	HTFIE	FTFIE	CHEN			
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		

位/位域	名称	描述
31:15	保留	必须保持复位值。
14	M2M	存储器到存储器模式 软件置位和清零 0: 禁止存储器到存储器模式 1: 使能存储器到存储器模式

		CHEN位为1时，该位不能被配置
13:12	PRI0[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

位	名称	描述
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

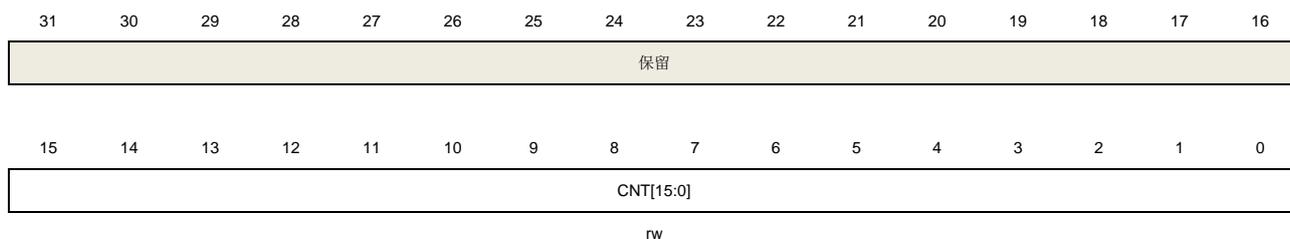
9.5.4. 通道 x 计数寄存器 (DMA_CHxCNT)

$x = 0 \dots 2$, x 为通道序号

地址偏移: $0x0C + 0x14 * x$

复位值: $0x0000\ 0000$

该寄存器只能按字 (32 位) 访问。



位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CNT[15:0]	传输计数 CHEN位为1时, 该位域不能被配置 该寄存器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载为初始设置值。

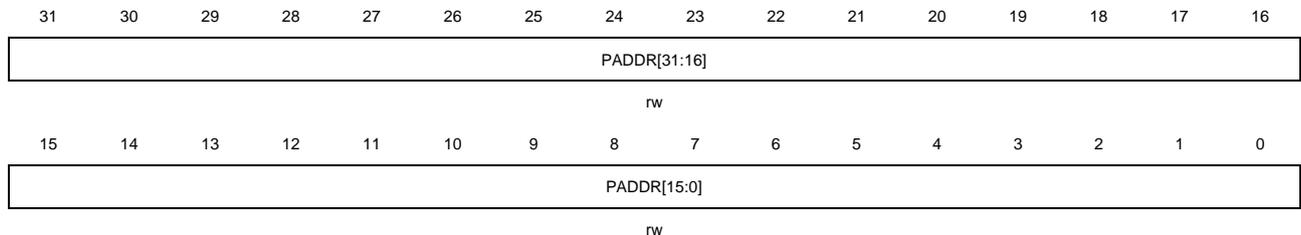
9.5.5. 通道 x 外设基地址寄存器 (DMA_CHxPADDR)

$x = 0 \dots 2$, x 为通道序号

地址偏移: $0x10 + 0x14 * x$

复位值: $0x0000\ 0000$

该寄存器只能按字 (32 位) 访问。



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10 (32-bit), PADDR[1:0]被忽略, 访问自动与32位地址对齐。

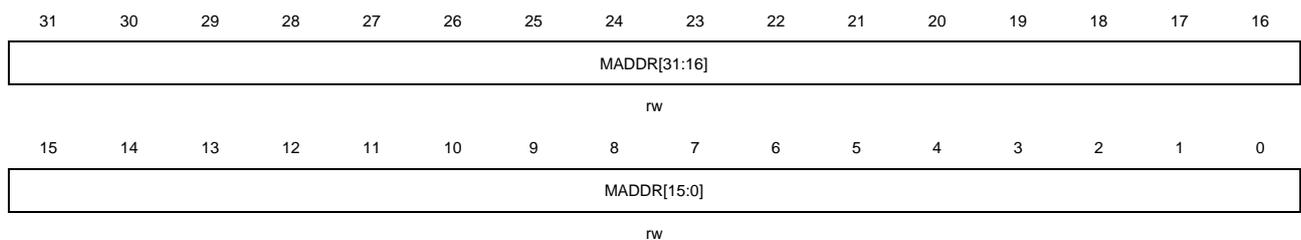
9.5.6. 通道 x 存储器基地址寄存器 (DMA_CHxMADDR)

$x = 0 \dots 2$, x 为通道序号

地址偏移: $0x14 + 0x14 * x$

复位值: $0x0000\ 0000$

该寄存器只能按字 (32 位) 访问。



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

10. DMA 请求多路复用器 (DMAMUX)

10.1. 简介

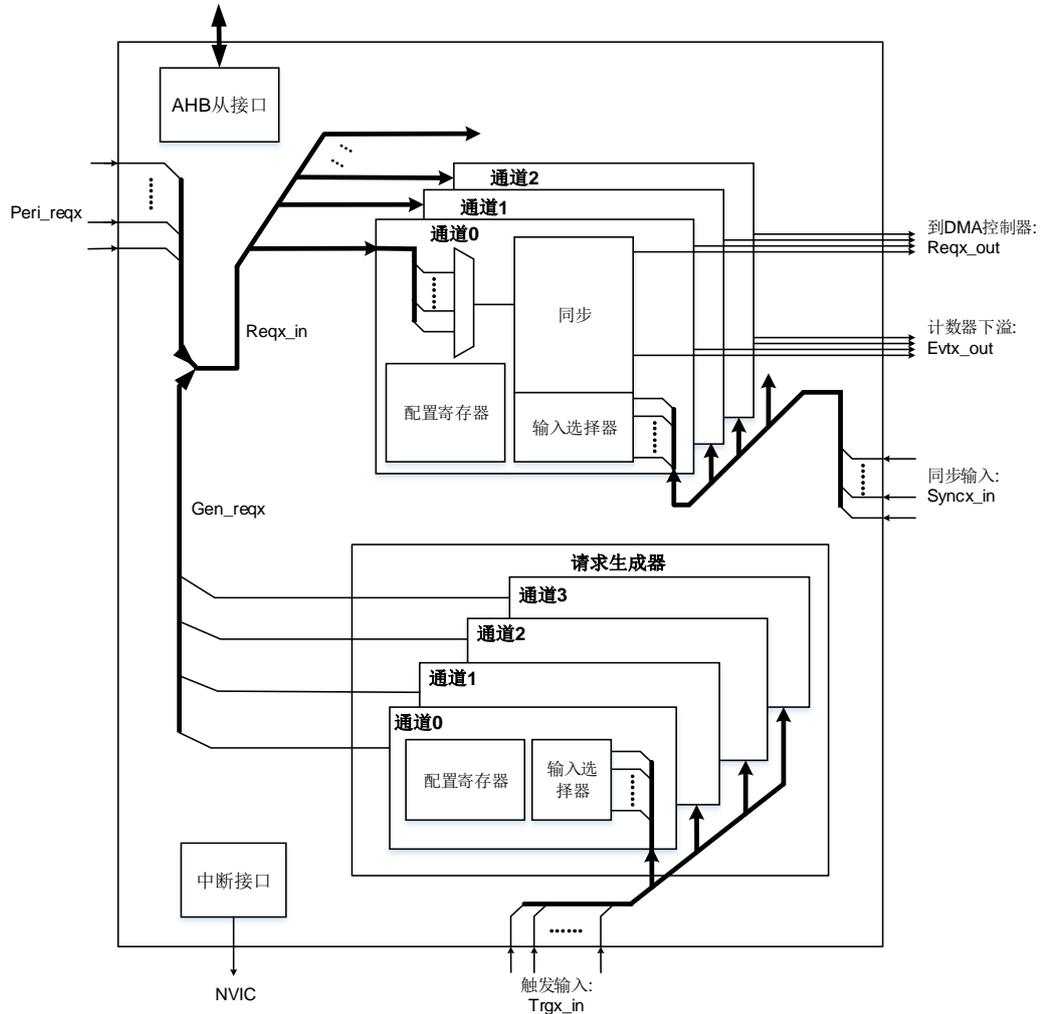
DMAMUX 是 DMA 请求的传输调度器。可编程的 DMA 请求多路复用器 DMAMUX，可在外设和 DMA 控制器之间路由 DMA 请求线路，或者 DMAMUX 也可以将可编程事件连入到输入触发信号上，作为一个 DMAMUX 请求发生器，再由 DMAMUX 请求路由器在 DMAMUX 请求生成器产生的 DMA 请求和 DMA 控制器之间路由 DMA 请求线路。每个 DMAMUX 请求路由通道选择一条唯一的 DMA 请求线路，无条件地或同步地从它的 DMAMUX 同步输入事件。DMA 请求信号会一直挂起，直到 DMA 控制器响应它，并且产生一个 DMA 确认信号，此时相应的 DMA 请求信号被释放。

10.2. 主要特征

- 3 个可配置的 DMAMUX 请求路由输出通道。
- 4 个 DMAMUX 请求生成通道。
- 24 路触发输入信号到 DMAMUX 请求生成器。
- 24 路同步输入信号。
- 每个 DMAMUX 请求生成通道包含一个 DMAMUX 请求触发输入选择器，一个 DMAMUX 请求生成计数器，和一个指示被选中的 DMAMUX 请求触发输入信号的事件溢出标志。
- 每个 DMAMUX 请求路由输出通道包含 55 路外设 DMAMUX 请求输入信号，一个同步输入信号选择器，一条 DMA 请求路由输出线路，一个路由事件输出信号用于 DMA 请求级联，一个 DMAMUX 请求路由计数器，和一个指示被选中的同步输入信号的事件溢出标志。

10.3. 结构框图

图 10-1. DMAMUX 结构框图



10.4. 信号描述

DMAMUX 信号描述如下所示：

- Reqx_in: DMAMUX 请求路由输入信号，来自外设的请求或者 DMAMUX 请求生成器生成的请求。
- Peri_reqx: 从外设输入到 DMAMUX 的 DMA 请求线路。
- Gen_reqx: DMAMUX 请求生成器生成输出的 DMA 请求信号。
- Reqx_out: DMAMUX 请求输出信号到 DMA 控制器。
- Trgx_in: DMAMUX 请求触发输入信号到 DMAMUX 请求生成器。
- Syncx_in: DMAMUX 同步输入信号到 DMAMUX 请求路由器。
- Evtx_out: DMAMUX 请求路由计数器下溢事件输出信号。

10.5. 功能说明

如[图 10-1. DMAMUX 结构框图](#)所示，DMAMUX 包含两个子模块：

- DMAMUX 请求路由器
DMAMUX 请求路由器输入 (Reqx_in) 来自两部分：
 - 一部分来自外设请求 (Peri_reqx)；
 - 另一部分来自 DMAMUX 请求生成器 (Gen_reqx)。DMAMUX 请求路由输出到 DMA 控制器对应的通道 (Reqx_out)。
同步输入 (Syncx_in) 来自内部或外部信号。
- DMAMUX 请求生成器
DMAMUX 请求触发输入 (Trgx_in) 来自内部或外部信号。

10.5.1. DMAMUX 请求路由器

DMAMUX 请求路由器可在外设/ DMAMUX 请求生成器，与 DMA 控制器之间路由 DMA 请求线路。DMAMUX 请求路由器由 DMAMUX 请求路由通道组成。DMA 请求输入信号并联至所有的 DMAMUX 请求路由通道。每个 DMAMUX 请求路由通道都有一个同步单元。同步输入信号并联至所有 DMAMUX 请求路由通道的同步单元。每个 DMAMUX 请求路由通道都有一个内部的 DMAMUX 请求路由计数器。

DMAMUX 请求路由通道

DMAMUX 请求路由通道 x 的请求路由输入由 DMAMUX_RM_CHxCFG 寄存器的 MUXID[5:0] / MUXID[6:0]位域来配置，请求路由输入可选为外设 DMA 请求，或者 DMAMUX 请求生成器产生的 DMA 请求，参考[表 10-2.GD32C2x1 DMAMUX 请求路由输入信号映射](#)。一个 DMAMUX 请求路由通道与对应的 DMA 控制器通道相连接。

注意：当 MUXID[6:0]值为 0 时，没有 DMA 请求线路被映射到 DMAMUX 请求路由通道上。DMAMUX 不允许将同一个 DMA 请求线路（相同 MUXID[5:0] / MUXID[6:0]且非空）映射到两个不同的 DMAMUX 请求路由通道上。

当同步模式禁能时

每当连到 DMAMUX 的 DMA 请求被 DMA 控制器服务，这个 DMA 请求将取消挂起，内部的 DMAMUX 请求路由计数器将减 1。当 DMAMUX 请求路由计数器发生下溢时，DMAMUX_RM_CHxCFG 寄存器的 NBR[4:0]值将自动重装载到计数器中。如果将 EVGEN 位置位，使能通道事件输出，则通道事件输出前，DMA 请求数量为 NBR[4:0] + 1。

注意：只有当 DMAMUX 请求路由通道 x 的同步使能位 SYNCEN 位和通道事件输出使能位 EVGEN 位都为 0 时，才能配置其 NBR[4:0]位域。

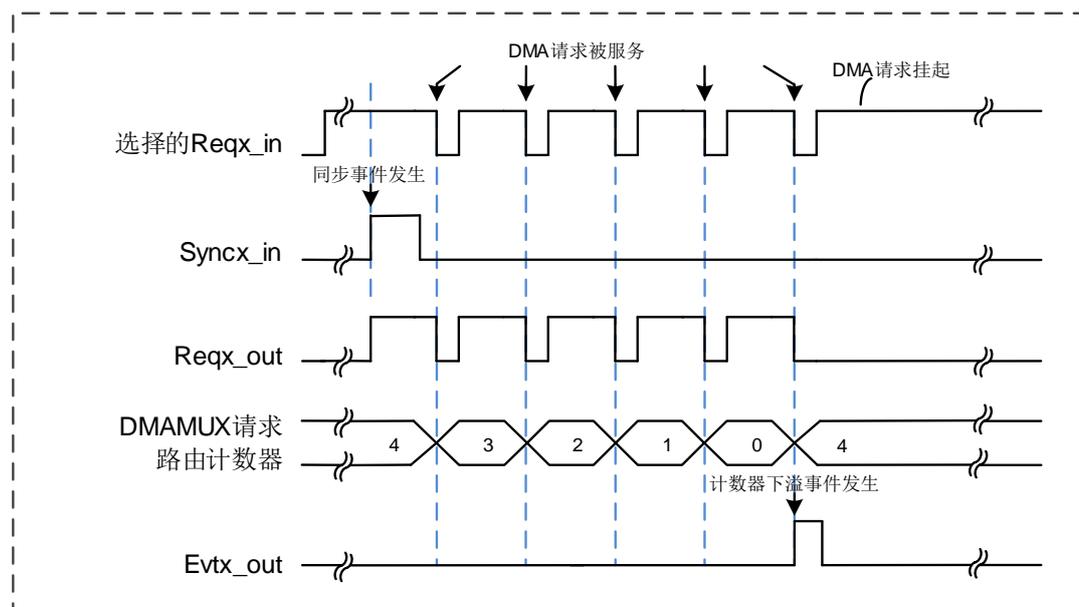
当同步模式使能时

如果 DMAMUX 请求路由通道 x 工作在同步模式下，当检测到选择的同步输入信号的上升沿或者下降沿时，挂起的 DMA 请求将被连到 DMAMUX 请求路由通道 x 的输出。每当连到 DMAMUX 的 DMA 请求被 DMA 控制器服务，这个 DMA 请求将取消挂起，内部的 DMAMUX 请求路由计

数器将减 1。当 DMAMUX 请求路由计数器发生下溢时，DMA 请求线路将断开与 DMAMUX 请求路由通道 x 的输出的连接，并且 DMAMUX_RM_CHxCFG 寄存器的 NBR[4:0] 值将自动重载到计数器中。一个同步事件可传输 NBR[4:0] + 1 个 DMA 请求到 DMAMUX 请求路由通道 x 的输出上。

[图 10-2. 同步模式](#)为当 NBR[4:0] = 4, SYNCEN = 1, EVGEN = 1, SYNCPC[1:0] = 0b01 时的举例。

图 10-2. 同步模式



置位 DMAMUX_RM_CHxCFG 寄存器的 SYNCEN 位可启用 DMAMUX 请求路由通道 x 的同步模式。同步输入信号可由 DMAMUX_RM_CHxCFG 寄存器的 SYNCID[4:0] 位域来配置，参考 [表 10-4. 同步输入信号映射](#)。同步输入信号的有效边沿由 DMAMUX_RM_CHxCFG 寄存器的 SYNCPC[1:0] 位域来配置。

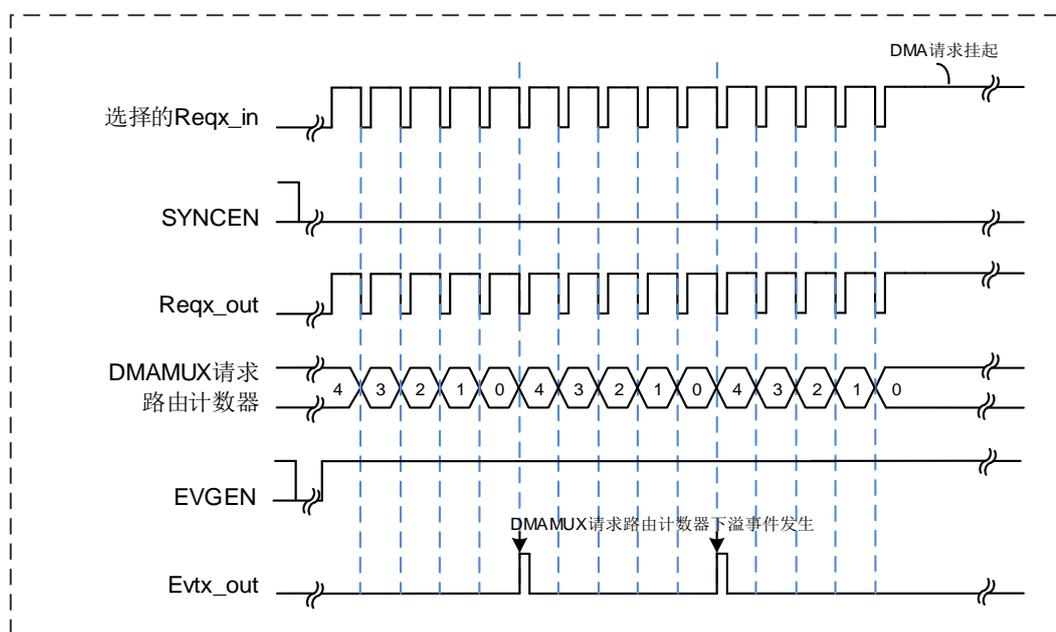
注意：如果同步输入事件发生时，DMAMUX 输入上没有挂起的 DMA 请求，则这个同步输入事件将被忽略，之后如有 DMA 请求被挂起，它将不会被连接到 DMAMUX 请求路由通道 x 的输出，直到发生下一个同步输入事件。

通道事件输出

每个 DMAMUX 请求路由通道都有一个通道事件输出信号 Evtx_out，用于 DMAMUX 请求路由计数器的下溢事件输出。Evt0_out ~ Evt2_out 信号可用于 DMA 请求级联。如果通过置位 DMAMUX_RM_CHxCFG 寄存器的 EVGEN 位来使能 DMAMUX 请求路由通道 x 的通道事件输出，当 DMAMUX 请求路由计数器自动重载为 NBR[4:0] 值时，发生一个通道事件，输出为一个 AHB 时钟周期脉冲。

[图 10-3. 通道事件输出](#)为当 NBR[4:0] = 4, SYNCEN = 0, EVGEN = 1 时的举例。

图 10-3. 通道事件输出



注意：如果 $EVGEN = 1$ 且 $NBR[4:0] = 0$ ，则每次 DMA 请求被服务时都会输出一个通道事件。

同步溢出

如果在 DMAMUX 请求路由计数器下溢之前又发生了新的同步事件，则 DMAMUX_RM_INTF 寄存器的同步溢出标志位 $SOIFx$ 位将置位。

注意：建议在 DMA 控制器对应通道请求被取消时，配置 DMAMUX_RM_CHxCFG 寄存器的 SYNCEN 位为 0 来禁能 DMAMUX 请求路由通道 x 的同步模式。否则，当又发生一个新的同步事件时，由于接收不到 DMA 的响应信号将会发生同步溢出事件。

10.5.2. DMAMUX 请求生成器

DMAMUX 请求生成器在触发输入事件发生时会产生 DMA 请求。DMAMUX 请求生成器由 DMAMUX 请求生成通道组成。DMA 请求触发输入信号并联至所有 DMAMUX 请求生成通道。每个 DMAMUX 请求生成通道都有一个内部的 DMAMUX 请求生成计数器。

触发输入信号的有效边沿由 DMAMUX_RG_CHxCFG 寄存器的 $RGTP[1:0]$ 位域来配置。DMAMUX 请求生成通道 x 的触发输入信号由 DMAMUX_RG_CHxCFG 寄存器的 $TID[4:0]$ 位域来配置，参考表 10-3. 触发输入信号映射。置位 DMAMUX_RG_CHxCFG 寄存器的 RGEN 位来使能 DMAMUX 请求生成通道 x。

DMAMUX 请求生成通道

当发生触发输入事件时，对应的 DMAMUX 请求生成通道 x 开始产生 DMA 请求到通道的输出上，通道输出连到 DMAMUX 请求路由器的输入上。每当 DMAMUX 生成的 DMA 请求被 DMA 控制器服务，这个 DMA 请求将取消挂起，内部的 DMAMUX 请求生成计数器将减 1。当 DMAMUX 请求生成计数器发生下溢时，DMAMUX 请求生成通道将停止产生 DMA 请求，在下

一个触发输入事件发生时，DMAMUX 请求生成计数器将自动重装载为 DMAMUX_RG_CHxCFG 寄存器的 NBRG[4:0]位域值。

注意：触发输入事件后产生的 DMA 请求数量为 NBRG[4:0] + 1。只有当 DMAMUX 请求生成通道 x 的 RGEN 位为 0 时才可以配置 NBRG[4:0]位域。

触发溢出

如果 RGEN 位为 1，DMAMUX 请求生成通道 x 被使能，当一个新的触发输入信号发生了，而此时 DMAMUX 请求生成计数器还未发生下溢，则 DMAMUX_RG_INTF 寄存器的 TOIFx 位将硬件置位以指示发生了触发溢出事件。

注意：建议在 DMA 控制器对应通道请求被取消时，配置 DMAMUX_RG_CHxCFG 寄存器的 RGEN 位为 0 来禁能 DMAMUX 请求生成通道 x。否则，当又发生一个新的触发输入事件时，由于接收不到 DMA 的响应信号将会发生触发溢出事件。

10.5.3. 通道配置

根据以下步骤来配置 DMAMUX 的通道 y 和对应的 DMA 通道 x:

1. 完整配置 DMA 通道 x 相关参数，除了 DMA 通道 x 的使能。
2. 完整配置 DMAMUX 通道 y 相关参数。
3. 设置 DMA_CHxCTL 寄存器的 CHEN 位 1 来使能 DMA 通道 x。

10.5.4. 中断

DMAMUX 模块有两种类型的中断事件，包括 DMAMUX 请求路由通道的同步溢出事件，和 DMAMUX 请求生成通道的触发溢出事件。

每个中断事件都有一个专用的标志位，专用的清除位和专用的使能位。[表10-1. 中断事件](#)描述了其对应关系。

表 10-1. 中断事件

中断事件	标志位	清除位	使能位
DMAMUX 请求路由通道 x 上的同步溢出事件	DMAMUX_RM_INTF 寄存器的 SOIFx 位	DMAMUX_RM_INTC 寄存器的 SOIFCx 位	DMAMUX_RM_CHxCFG 寄存器的 SOIE 位
DMAMUX 请求生成通道 y 上的触发溢出事件	DMAMUX_RG_INTF 寄存器的 TOIFy 位	DMAMUX_RG_INTC 寄存器的 TOIFCy 位	DMAMUX_RG_CHxCFG 寄存器的 TOIE 位

触发溢出中断

当 DMAMUX 请求生成触发溢出标志位 TOIFx 置位，并且触发溢出中断使能位 TOIE 位置位，则会产生一个触发溢出中断。写 1 到 DMAMUX_RG_INTC 寄存器的对应触发溢出清除位 TOIFCx 将会清除触发溢出标志位 TOIFx。

同步溢出中断

当 DMAMUX 请求路由同步溢出标志位 SOIFx 置位，并且触发同步溢出中断使能位 SOIE 位置位，则会产生一个同步溢出中断。写 1 到 DMAMUX_RM_INTC 寄存器的对应同步溢出清除位 SOIFCx 将会清除同步溢出标志位 SOIFx。

10.5.5. DMAMUX 映射

DMAMUX 请求路由输入映射

DMAMUX 请求路由输入可来自于外设或者 DMAMUX 请求生成器，参考[表 10-2.GD32C2x1 DMAMUX 请求路由输入信号映射](#)，由 DMAMUX_RM_CHxCFG 寄存器的 MUXID[6:0]位域配置 DMAMUX 请求路由通道 x 的输入。

表 10-2.GD32C2x1 DMAMUX 请求路由输入信号映射

请求路由通道输入标识 MUXID[6:0]	来源
1	Gen_reqx0
2	Gen_reqx1
3	Gen_reqx2
4	Gen_reqx3
5	ADC
6	Reserved
7	Reserved
8	Reserved
9	Reserved
10	I2C0_RX
11	I2C0_TX
12	I2C1_RX
13	I2C1_TX
14	Reserved
15	Reserved
16	SPI0_RX
17	SPI0_TX
18	SPI1_RX
19	SPI1_TX
20	TIMER0_CH0
21	TIMER0_CH1
22	TIMER0_CH2
23	TIMER0_CH3
24	TIMER0_TRIG
25	TIMER0_UP
26	TIMER0_COM
27	Reserved

请求路由通道输入标识 MUXID[6:0]	来源
28	Reserved
29	Reserved
30	Reserved
31	Reserved
32	TIMER2_CH0
33	TIMER2_CH1
34	TIMER2_CH2
35	TIMER2_CH3
36	TIMER2_TRIG
37	TIMER2_UP
38	Reserved
39	Reserved
40	Reserved
41	Reserved
42	Reserved
43	Reserved
44	TIMER15_CH0
45	Reserved
46	TIMER15_UP
47	TIMER16_CH0
48	Reserved
49	TIMER16_UP
50	USART0_RX
51	USART0_TX
52	USART1_RX
53	USART1_TX
54	USART2_RX
55	USART2_TX

触发输入映射

DMAMUX 请求生成通道 x 的触发输入可由 DMAMUX_RG_CHxCFG 寄存器的 TID[4:0]位域来配置，参考[表 10-3. 触发输入信号映射](#)。

表 10-3. 触发输入信号映射

触发输入标识TID[4:0]	来源
0	EXTI_0
1	EXTI_1
2	EXTI_2
3	EXTI_3
4	EXTI_4
5	EXTI_5

触发输入标识TID[4:0]	来源
6	EXTI_6
7	EXTI_7
8	EXTI_8
9	EXTI_9
10	EXTI_10
11	EXTI_11
12	EXTI_12
13	EXTI_13
14	EXTI_14
15	EXTI_15
16	Evtx_out0
17	Evtx_out1
18	Evtx_out2
19	Reserved
20	Reserved
21	TIMER13_O
22	Reserved
23	Reserved

同步输入映射

同步输入由 DMAMUX_RM_CHxCFG 寄存器的 SYNCID[4:0]位域来配置，参考[表 10-4. 同步输入信号映射](#)。

表 10-4. 同步输入信号映射

同步输入标识SYNCID[4:0]	来源
0	EXTI_0
1	EXTI_1
2	EXTI_2
3	EXTI_3
4	EXTI_4
5	EXTI_5
6	EXTI_6
7	EXTI_7
8	EXTI_8
9	EXTI_9
10	EXTI_10
11	EXTI_11
12	EXTI_12
13	EXTI_13
14	EXTI_14
15	EXTI_15
16	Evtx_out0

同步输入标识SYNCID[4:0]	来源
17	Evtx_out1
18	Evtx_out2
19	Reserved
20	Reserved
21	TIMER13_O
22	Reserved
23	Reserved

10.6. DMAMUX 寄存器

DMAMUX基地址：0x4002 0800

10.6.1. 请求路由通道 x 配置寄存器 (DMAMUX_RM_CHxCFG)

$x = 0 \dots 2$ ，其中 x 为通道序号

地址偏移：0x00 + 0x04 * x

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

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

位/位域	名称	描述
31:29	保留	必须保持复位值。
28:24	SYNCID[4:0]	同步输入标识 选择同步输入源。
23:19	NBR[4:0]	传递的DMA请求数量 在同步输入事件之后，或者通道事件输出之前，将传递到DMA控制器的DMA请求数量为NBR[4:0] + 1。 该位域只能在SYNCEN位和EVGEN位都禁能时才能配置。
18:17	SYNCP[1:0]	同步输入极性 00：不检测事件 01：上升沿 10：下降沿 11：上升和下降沿
16	SYNCEN	同步模式使能 0：禁能同步模式 1：使能同步模式
15:10	保留	必须保持复位值。
9	EVGEN	事件输出使能 0：禁能事件输出 1：使能事件输出
8	SOIE	同步溢出中断使能 0：禁能中断

		1: 使能中断
7:6	保留	必须保持复位值。
5:0	MUXID[5:0]	请求路由标识 选择DMAMUX请求路由通道的DMA请求输入源。

10.6.2. 请求路由通道中断标志位寄存器 (DMAMUX_RM_INTF)

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
31:3	保留	必须保持复位值。
2	SOIF2	请求路由通道2的同步溢出事件标志位 参照SOIF0的描述。
1	SOIF1	请求路由通道1的同步溢出事件标志位 参照SOIF0的描述。
0	SOIF0	请求路由通道0的同步溢出事件标志位 如果同步输入事件发生时, DMAMUX请求路由由计数器值小于NBR[4:0], 则该位置位。 通过对DMAMUX_RM_INTC寄存器的SOIFC0位写1来清除相应通道的同步溢出标志。

10.6.3. 请求路由通道中断标志位清除寄存器 (DMAMUX_RM_INTC)

地址偏移: 0x084
复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问。



位/位域	名称	描述
31:3	保留	必须保持复位值。
2	SOIFC2	请求路由通道2的同步溢出事件标志清除位 参照SOIFC0的描述。
1	SOIFC1	请求路由通道1的同步溢出事件标志清除位 参照SOIFC0的描述。
0	SOIFC0	请求路由通道0的同步溢出事件标志清除位 写1可清除相应通道在DMAMUX_RM_INTF寄存器的同步溢出标志SOIF0。

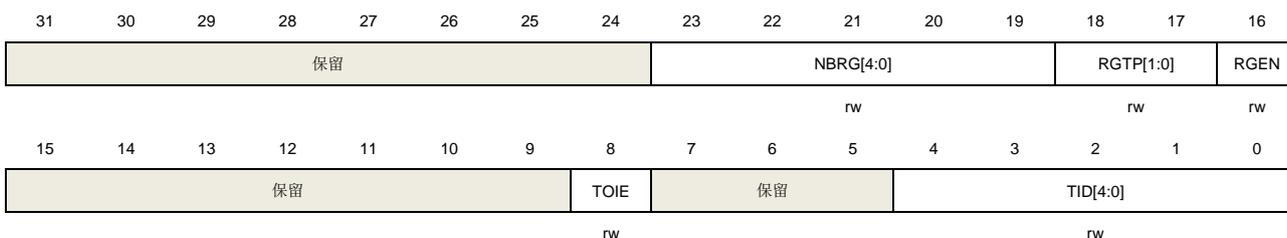
10.6.4. 请求生成通道 x 配置寄存器 (DMAMUX_RG_CHxCFG)

$x = 0 \dots 3$, 其中 x 为通道序号

地址偏移: $0x100 + 0x04 * x$

复位值: $0x0000\ 0000$

该寄存器只能按字 (32 位) 访问。



位/位域	名称	描述
31:24	保留	必须保持复位值。
23:19	NBRG[4:0]	待产生的DMA请求数量 在触发输入事件之后, 待产生的DMA请求数量为NBRG[4:0] + 1。 注意: 只有当RGEN位为0时才能写该位域。
18:17	RGTP[1:0]	DMAMUX请求生成触发输入极性 00: 不检测事件 01: 上升沿 10: 下降沿 11: 上升沿和下降沿
16	RGEN	DMAMUX请求生成通道x使能 0: 禁能DMAMUX请求生成通道x 1: 使能DMAMUX请求生成通道x
15:9	保留	必须保持复位值。
8	TOIE	触发溢出中断使能 0: 禁能中断

		1: 使能中断
7:5	保留	必须保持复位值。
4:0	TID[4:0]	触发输入标识 选择DMAMUX请求生成通道的触发输入源。

10.6.5. 请求生成通道中断标志位寄存器 (DMAMUX_RG_INTF)

地址偏移: 0x140

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问。



位/位域	名称	描述
31:3	保留	必须保持复位值。
3	TOIF3	DMAMUX请求生成通道3的触发溢出标志位 参照TOIF0的描述。
2	TOIF2	DMAMUX请求生成通道2的触发溢出标志位 参照TOIF0的描述。
1	TOIF1	DMAMUX请求生成通道1的触发溢出标志位 参照TOIF0的描述。
0	TOIF0	DMAMUX请求生成通道0的触发溢出标志位 如果触发输入事件在DMAMUX请求生成计数器下溢之前发生, 则该位置位。 通过对DMAMUX_RG_INTC寄存器的TOIFC0位写1来清除相应通道的触发溢出标志。

10.6.6. 请求生成通道中断标志位清除寄存器 (DMAMUX_RG_INTC)

地址偏移: 0x144

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问。



位/位域	名称	描述
31:3	保留	必须保持复位值。
3	TOIFC3	DMAMUX请求生成通道3的触发溢出标志清除位 参照TOIFC0的描述。
2	TOIFC2	DMAMUX请求生成通道2的触发溢出标志清除位 参照TOIFC0的描述。
1	TOIFC1	DMAMUX请求生成通道1的触发溢出标志清除位 参照TOIFC0的描述。
0	TOIFC0	DMAMUX请求生成通道0的触发溢出标志清除位 写1可清除相应通道在DMAMUX_RG_INTF寄存器的触发溢出标志TOIF0。

11. 模数转换器（ADC）

11.1. 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 13 个外部通道和 3 个内部通道上的模拟信号。这些 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来自 MCU 的相关计算负担来提高性能。

对于电机、电源等对 ADC 有更高需求的应用，可以联系我们的销售，获取更多的 ADC 详细资料。

11.2. 主要特征

- 高性能：
 - ADC采样分辨率：12位、10位、8位、或者6位分辨率；
 - 可编程采样时间；
 - 数据存储模式：最高有效位对齐和最低有效位对齐；
 - 支持DMA请求。
- 多时钟域架构：
 - CK_SYS系统时钟，IRC48MDIV_PER时钟；
- 模拟输入通道：
 - 多达13个外部模拟输入通道；
 - 1个内部温度传感器通道（V_{SENSE}）；
 - 1个内部参考电压输入通道（V_{REFINT}）；
 - 1个内部正参考电压输入通道（V_{REFP}）
- 转换开始的发起：
 - 软件；
 - 硬件触发。
- 运行模式：
 - 转换单个通道，或者扫描一序列的通道；
 - 单次运行模式，每次触发转换一次选择的输入通道；
 - 连续运行模式，连续转换所选择的输入通道；
 - 间断运行模式；
- 转换结果阈值监测功能：模拟看门狗。
- 中断产生：
 - 常规序列转换结束；
 - 模拟看门狗事件；
- 过采样：
 - 16位的数据寄存器；
 - 可调整的过采样率，从2x到256x；
 - 高达8位的可编程数据移位。
- ADC输入范围：V_{REFN} ≤ V_{IN} ≤ V_{REFP}。

11.3. 引脚和内部信号

[图11-1. ADC模块框图](#)给出了ADC模块框图。[表11-1. ADC内部](#)给出了ADC内部信号。[表11-2. ADC引脚定义](#)给出了ADC引脚说明。

表 11-1. ADC 内部输入/输出信号

内部信号名称	说明
V _{SENSE}	内部温度传感器输出电压
V _{REFINT}	内部参考输出电压
ADC_WD _x _OUT	模拟看门狗x输出信号，被连接到TIMER模块(x=0,1,2)

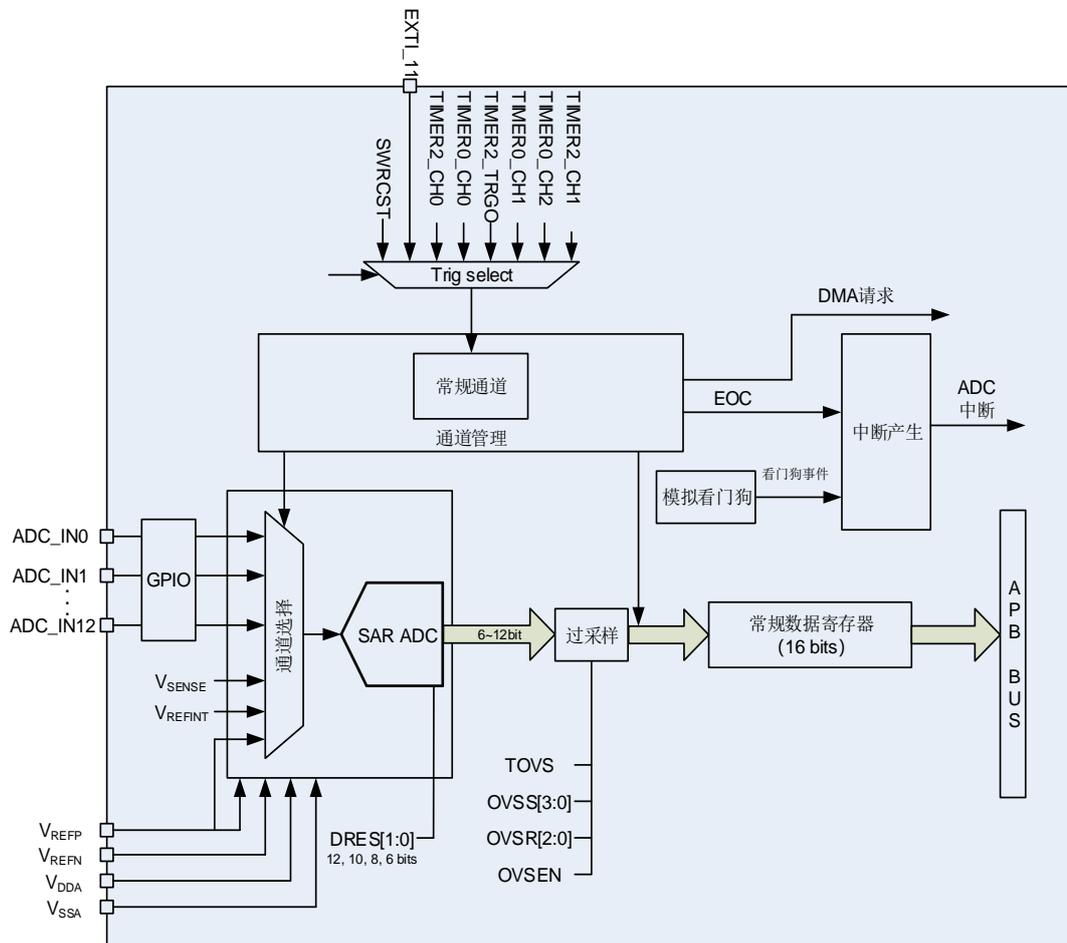
表 11-2. ADC 引脚定义

名称	注释
V _{DDA}	模拟电源输入，等于V _{DD}
V _{SSA}	模拟地，等于V _{SS}
V _{REFP}	ADC正参考电压
V _{REFN}	ADC负参考电压
ADC _x _IN[12:0]	多达13路外部通道

注意： V_{DDA}和V_{SSA}必须分别连接到V_{DD}和V_{SS}。

11.4. 功能描述

图 11-1. ADC 模块框图



11.4.1. 多时钟域架构

除了系统时钟CK_SYS，ADC模块的时钟还可以由IRC48MDIV_PER分频后提供。使用IRC48MDIV_PER或HXTAL时钟分频作为ADC时钟，可以实现降低系统时钟频率后，应用程序在低功耗运行的同时，ADC仍保持最佳运行状态。ADC的最大频率为24Mhz。

想要更多ADC时钟产生的信息，可以参考RCU章节。

11.4.2. 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_{ST(ADC)}时间后才能采样，t_{ST(ADC)}数值详见芯片相关型号 Datasheet。

11.4.3. 常规序列

通道管理电路把采样通道组织成一个常规序列。

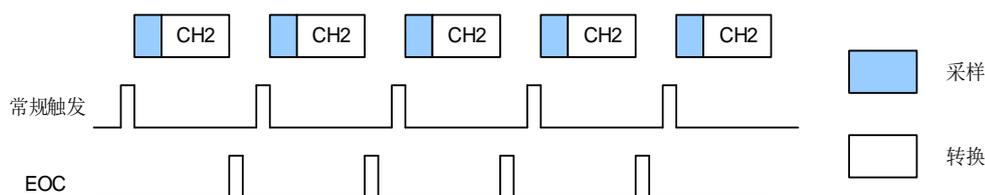
ADC常规序列支持高达16个通道。每个通道称为常规通道。ADC_RSQ0寄存器的RL[3:0]位规定了整个常规序列的长度。ADC_RSQ0~ADC_RSQ2寄存器规定了常规序列的通道选择。

11.4.4. 运行模式

单次运行模式

单次运行模式下，ADC_RSQ2寄存器的RSQ0[3:0]位规定了ADC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发生，ADC就会采样和转换一个通道。

图 11-2. 单次运行模式



常规序列的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。

常规序列单次运行模式的软件流程：

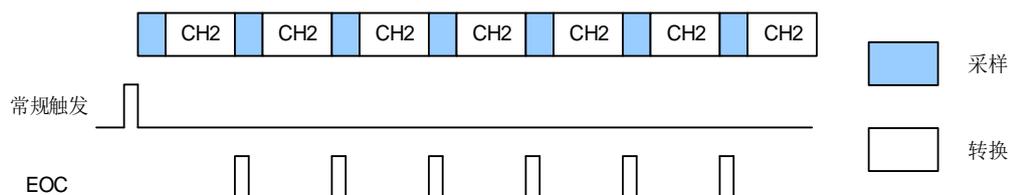
1. 确保ADC_CTL0寄存器的DISRC和SM位以及ADC_CTL1寄存器的CTN位为0；
2. 用模拟通道编号来配置ADC_RSQ2寄存器的RSQ0[3:0]位域；
3. 配置ADC_SAMPTx寄存器；
4. 如果有需要，可以配置ADC_CTL1寄存器的ETERC和ETSRC位；
5. 设置SWRCST位，或者为常规序列产生一个外部触发信号；
6. 等到EOC置1；
7. 从ADC_RDATA寄存器中读ADC转换结果；
8. 写0清除EOC标志位。

注意：当EOC置1后，需延迟一个CK_ADC再读取ADC转换结果。

连续运行模式

将ADC_CTL1寄存器的CTN位置1可以使能连续运行模式。在此模式下，ADC执行由RSQ0[3:0]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 11-3. 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寄存器的CTN位为1；

2. 用模拟通道编号来配置ADC_RSQ2寄存器的RSQ0[3:0]位域;
3. 配置ADC_SAMPTx寄存器;
4. 如果有需要, 配置ADC_CTL1寄存器的ETERC和ETSRC位;
5. 设置SWRCST位, 或者给常规序列产生一个外部触发信号;
6. 等待EOC标志位置1;
7. 从ADC_RDATA寄存器中读ADC转换结果;
8. 写0清除EOC标志位;
9. 只要还需要进行连续转换, 重复步骤6~8。

注意: 当 EOC 置 1 后, 需延迟一个 CK_ADC 再读取 ADC 转换结果。

可以使用 DMA 来传输转换数据, 不需循环查询 EOC 标志位, 软件流程如下:

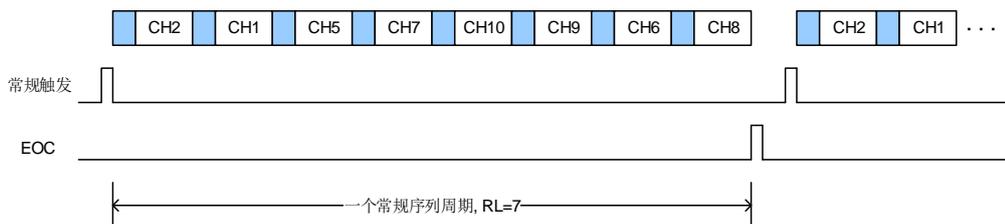
1. 设置ADC_CTL1寄存器的CTN位和DMA位为1;
2. 用模拟通道编号来配置ADC_RSQ2寄存器的RSQ0[3:0]位域;
3. 配置ADC_SAMPTx寄存器;
4. 如果有需要, 配置ADC_CTL1寄存器的ETERC和ETSRC位;
5. 准备DMA模块, 用于传输来自ADC_RDATA的数据;
6. 设置SWRCST位, 或者给常规序列产生一个外部触发。

扫描运行模式

扫描运行模式可以通过将 ADC_CTL0 寄存器的 SM 位置 1 来使能。在此模式下, ADC 扫描转换所有被 ADC_RSQ0~ADC_RSQ2 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通道。转换数据存储在 ADC_RDATA 寄存器中。常规序列转换结束后, EOC 位将被置 1。如果 EOCIE 位被置 1, 将产生中断。当常规序列工作在扫描模式下时, ADC_CTL1 寄存器的 DMA 位必须设置为 1。

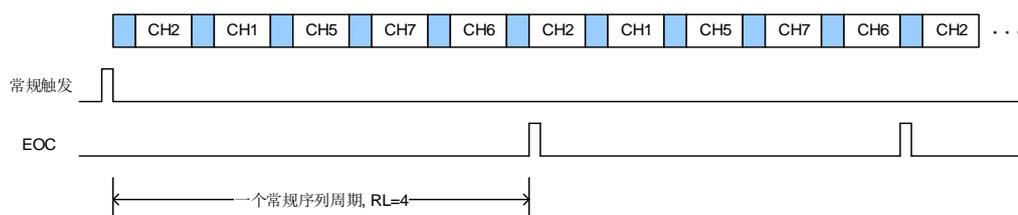
如果 ADC_CTL1 寄存器的 CTN 位也被置 1, 则在常规序列转换完之后, 这个转换自动重新开始。

图 11-4. 扫描运行模式, 且连续运行模式禁能



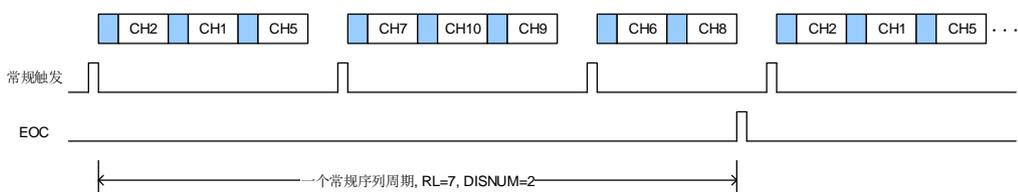
常规序列扫描运行模式的软件流程:

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

图 11-5. 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，常规序列间断运行模式使能。该模式下可以执行一次 n 个通道的短序列转换 ($n \leq 8$)，这个短序列是 ADC_RSQ0~RSQ2 寄存器所选择的转换序列的一部分。数值 n 由 ADC_CTL0 寄存器的 DISCNUM[2:0] 位给出。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~RSQ2 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。

图 11-6. 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 DMA 位为 1；
2. 配置 ADC_CTL0 寄存器的 DISCNUM[2:0] 位；
3. 配置 ADC_RSQx 和 ADC_SAMPTx 寄存器；
4. 如果有需要，配置 ADC_CTL1 寄存器中的 ETERC 和 ETSRC 位；
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；
8. 等待 EOC 标志位置 1；
9. 写 0 清除 EOC 标志位。

11.4.5. 转换结果阈值监测功能

模拟看门狗 0

ADC_CTL0 寄存器的 RWD0EN 位置 1 将使能常规序列的模拟看门狗 0 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 WD0E 位将被置 1。如果 WD0EIE 位被置 1，将产生中断。ADC_WD0HT 和 ADC_WD0LT 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，因此阈值与 ADC_CTL0 寄存器的 DAL 位确定的对齐方式无关。ADC_CTL0 寄存器的 RWD0EN，WD0SC 和 WD0CHSEL[3:0] 位可以用来选择模拟看门狗 0 监控单一通道或者多通道。

模拟看门狗 1/2

模拟看门狗 1/2 更加的灵活，可以进行单个或多个通道的看门狗功能配置。

通过配置 ADC_WD1SR 寄存器中的 AWD1CS 位域中的相应位，可以使能相应通道的模拟看门狗 1 功能，同理，可以配置看门狗 2 功能。模拟看门狗 1/2 的高/低阈值可在 ADC_WD1HT / ADC_WD1LT 寄存器和 ADC_WD2HT / ADC_WD2LT 寄存器中进行配置。

ADC_WD_x_OUT 输出信号

每个模拟看门狗会生成对应的 ADC_WD_x_OUT(x=0,1,2)信号，该信号连接至定时器(TIMER)模块。在定时器模块中，可选择 ADC_WD_x_OUT 信号或其他信号作为外部触发输入(ETI)的信号源(详见[定时器\(TIMER\)](#)模块)。

当被监测通道的转换结果超过阈值时，ADC_WD_x_OUT 信号会被置为 1。即使软件清除了 WDXE 标志位，也不会影响 ADC_WD_x_OUT 信号的状态。当转换结果重新回到阈值范围内时，ADC_WD_x_OUT 信号会被复位为 0。

11.4.6. 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。

注入序列通道转换的数据值已经减去了在 ADC_IOFF_x 寄存器中定义的偏移量，因此结果可能是一个负值。符号值是一个扩展值。

图 11-7. 12 位数据存储模式

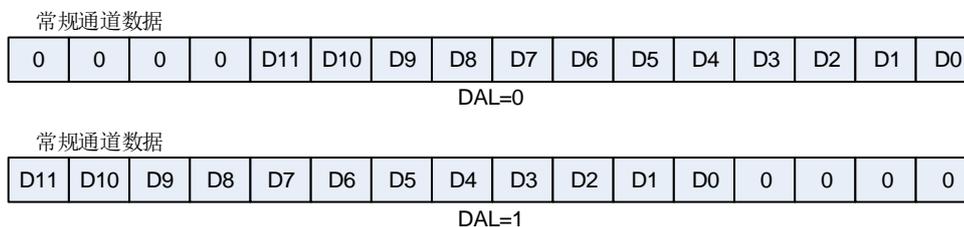


图 11-8. 10 位数据存储模式

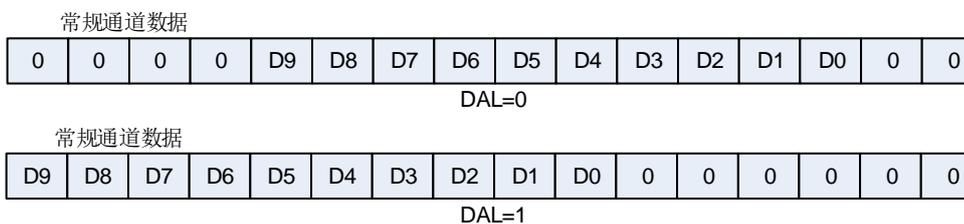
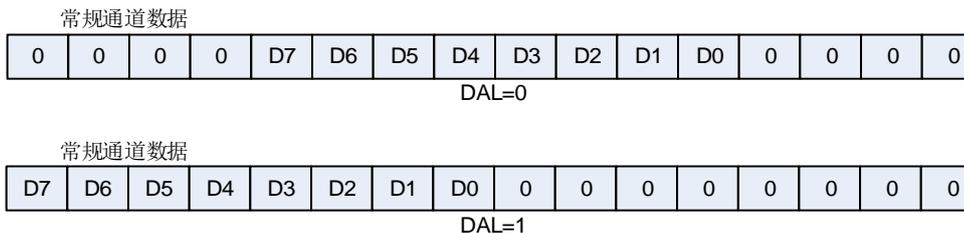
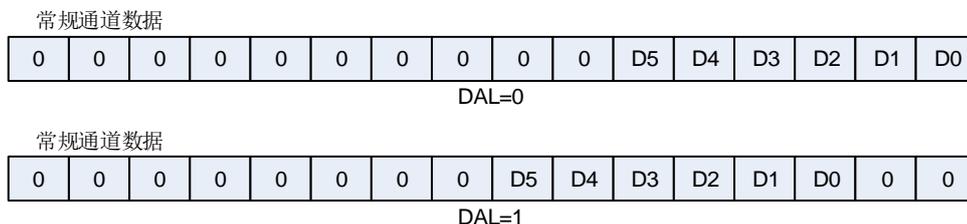


图 11-9. 8 位数据存储模式



6 位分辨率的数据存储模式不同于 12 位/10 位/8 位分辨率数据存储模式，如 [图 11-10. 6 位数数据存储模式](#)。

图 11-10. 6 位数数据存储模式



11.4.7. 采样时间配置

ADC 使用若干个 CK_ADC 周期对输入电压采样，采样周期数目可以通过 ADC_SAMPT0 和 ADC_SAMPT1 寄存器的 SPTn[2:0]位更改。每个通道可以用不同的时间采样。在 12 位分辨率的情况下，总采样转换时间=采样时间+12.5 个 CK_ADC 周期。

例如：

CK_ADC = 24MHz，采样时间为 2.5 个周期，那么总的转换时间为：“2.5+12.5”个 CK_ADC 周期，即 0.625us。

11.4.8. 外部触发

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

表 11-3.常规序列外部触发源

ETSRC[2:0]	触发源	触发类型
000	TIMER2_CH1	硬件触发
001	TIMER0_CH2	
010	TIMER0_CH1	
011	TIMER2_TRGO	
100	TIMER0_CH0	
101	TIMER2_CH0	
110	EXTI_11	
111	SWRCST	软件触发

11.4.9. DMA 请求

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

11.4.10. ADC 内部通道

ADC 模拟输入通道 13、14 和 15 分别连接到温度传感器、V_{REFINT} 和 V_{REFP} 模拟输入。

将 ADC_CTL1 寄存器的 TSVEN 位置 1 可以使能温度传感器通道。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）。温度传感器不用时，复位 TSVEN 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，温度变化曲线的偏移在不同的芯片上会有不同（最多相差 45°C）。内部温度传感器更适合于检测温度的变化，而不是测量绝对温度。如果需要测量精确的温度，应该使用一个外置的温度传感器来校准这个偏移错误。

使用温度传感器：

1. 配置温度传感器通道的转换序列和采样时间大于 t_{s_temp}；
2. 置位 ADC_CTL1 寄存器的 TSVEN 位，使能温度传感器；
3. 由软件或外部触发启动 ADC 转换；
4. 读取内部温度传感器输出电压 V_{temperature}，并由下面公式计算出实际温度：

$$\text{温度 (}^{\circ}\text{C)} = \frac{V_{25} - V_{\text{temperature}}}{\text{Avg_Slope}} + 25 \quad (11-1)$$

V_{temperature}：温度传感器的输出电压。

V₂₅：内部温度传感器在 25°C 时的输出电压，典型值请参考相关型号 Datasheet。

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

将 ADC_CTL1 寄存器的 INREFEN 位置 1 可以使能 V_{REFINT} 通道。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_IN14 输入通道。

V_{REFP} 通道内部连接到正参考电压引脚 VREFP。部分封装不存在 VREFP 引脚时，V_{REFP} 通道内部连接到 VDDA。

11.4.11. 可编程分辨率（DRES）

对寄存器 ADC_OVSAMPCTL 中的 DRES[1:0] 位进行编程即可配置分辨率为 6、8、10、12 位。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。ADC 转换的结果只有 12 位，其余没有被用到的低位读出来都是为 0。较低的分辨率能够减少逐次逼近步骤所需的转换时间，如 [表 11-4. 不同分辨率对应的 t_{CONV} 时间](#) 所示。

表 11-4. 不同分辨率对应的 t_{CONV} 时间

DRES[1:0] 位域	t _{CONV} (ADC 时 钟周期)	t _{CONV} (ns) at f _{ADC} =24MHz	t _{SAMPL} (min) (ADC 时钟周期)	t _{ADC} (ADC 时钟 周期)	t _{ADC} (ns) at f _{ADC} =24MHz
12	12.5	521ns	2.5	15	625ns
10	10.5	438ns	2.5	13	542ns

8	8.5	354ns	2.5	11	458ns
6	6.5	271ns	2.5	9	375ns

11.4.12. 片上硬件过采样

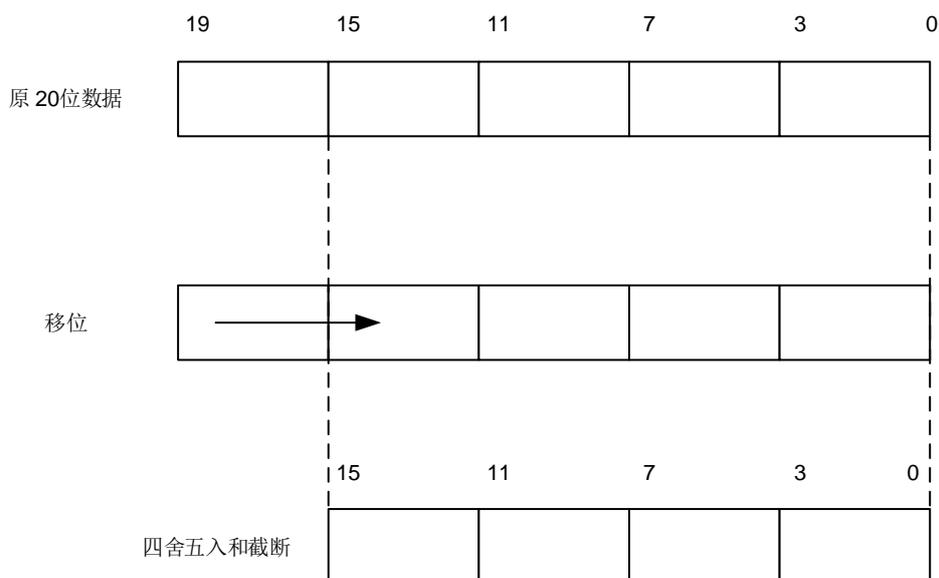
片上硬件过采样单元执行数据预处理以减轻 CPU 负担。它能够处理多个转换，并将多个转换的结果取平均，得出一个 16 位宽的数据。其结果根据如下公式计算得出，其中 N 和 M 的值可以被调整。D_{out} (n) 是指 ADC 输出的第 n 个数字信号：

$$\text{Result} = \frac{1}{M} * \sum_{n=0}^{N-1} D_{\text{out}}(n) \quad (11-2)$$

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 OVS_R[2:0] 位定义，它的取值范围为 2x 到 256x。除法系数 M 定义一个多达 8 位的右移，它通过 ADC_OVSAMPCTL 寄存器 OVS_S[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，将这个值要进行右移，将移位后剩余的部分再通过取整转化一个近似值，最后将高位会被截断，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。

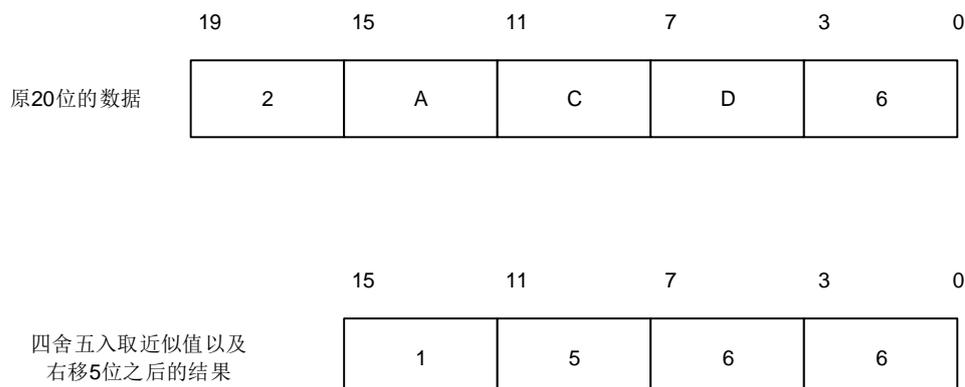
图 11-11. 20 位到 16 位的结果截断



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 11-12. 右移 5 位和取整的数例](#)描述一个从原始 20 位的累积数值处理成 16 位结果值的例子。

图 11-12. 右移 5 位和取整的数例



[表 11-5. N 和 M 的最大输出值（灰色部分表示截断）](#) 给出了 N 和 M 各种组合的数据格式，初始转换值为 0xFFFF。

表 11-5. N 和 M 的最大输出值（灰色部分表示截断）

Oversampling ratio	Max Raw data	No-shift OVSS=0000	1-bit shift OVSS=0001	2-bit shift OVSS=0010	3-bit shift OVSS=0011	4-bit shift OVSS=0100	5-bit shift OVSS=0101	6-bit shift OVSS=0110	7-bit shift OVSS=0111	8-bit shift 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

和标准的运行模式相比，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：

$$N \times t_{\text{ADC}} = N \times (t_{\text{SMPL}} + t_{\text{CONV}}) \quad (11-3)$$

11.5. 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；

单独的中断使能位可使得使用更灵活。

11.6. ADC 寄存器

ADC 基地址: 0x4001 2400

11.6.1. 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。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WD2E	WD1E	保留													
rc_w0	rc_w0														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留											STRC	保留		EOC	WD0E
											rc_w0	rc_w0		rc_w0	

位/位域	名称	说明
31	WD2E	模拟看门狗 2 事件标志 0: 没有模拟看门狗 2 事件 1: 模拟看门狗 2 事件发生 转换电压超过 ADC_WD2HT 和 ADC_WD2LT 寄存器设定的阈值时由硬件置 1, 软件写 0 清除。
30	WD1E	模拟看门狗 1 事件标志 0: 没有模拟看门狗 1 事件 1: 模拟看门狗 1 事件发生 转换电压超过 ADC_WD1HT 和 ADC_WD1LT 寄存器设定的阈值时由硬件置 1, 软件写 0 清除。
29:5	保留	必须保持复位值。
4	STRC	常规序列转换开始标志 0: 常规序列转换没有开始 1: 常规序列转换开始 常规序列转换开始时硬件置位, 软件写 0 清除。
3:2	保留	必须保持复位值。
1	EOC	序列转换结束标志 0: 序列转换没有结束 1: 序列转换结束 常规序列转换结束时硬件置位, 软件写 0 或读 ADC_RDATA 寄存器清除。
0	WD0E	模拟看门狗事件 0 标志 0: 没有模拟看门狗 0 事件 1: 模拟看门狗 0 事件发生

转换电压超过 ADC_WD0LT 和 ADC_WD0HT 寄存器设定的阈值时由硬件置 1，软件写 0 清除。

11.6.2. 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
WD2EIE	WD1EIE	保留				DRES [1:0]		RWD0EN	保留						
rw	rw					rw		rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DISNUM [2:0]		保留	DISRC	保留	WD0SC	SM	保留	WD0EIE	EOCIE	保留	WD0CHSEL[3:0]				
rw			rw		rw	rw		rw	rw		rw				

位/位域	名称	说明
31	WD2EIE	WD2E 中断使能 0: WD2E 中断禁能 1: WD2E 中断使能
30	WD1EIE	WD1E 中断使能 0: WD1E 中断禁能 1: WD1E 中断使能
29:26	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	RWD0EN	常规序列模拟看门狗 0 使能 0: 常规序列模拟看门狗 0 禁能 1: 常规序列模拟看门狗 0 使能
22:16	保留	必须保持复位值。
15:13	DISNUM[2:0]	间断模式下的转换数目 触发后即将被转换的通道数目将变成 DISNUM[2:0]+1
12	保留	必须保持复位值。
11	DISRC	常规序列间断模式 0: 常规序列间断运行模式禁能 1: 常规序列间断运行模式使能

10	保留	必须保持复位值。
9	WD0SC	扫描模式下，模拟看门狗 0 在单个通道配置 0: 模拟看门狗 0 在所有通道有效 1: 模拟看门狗 0 在单通道有效
8	SM	扫描模式 0: 扫描运行模式禁能 1: 扫描运行模式使能
7	保留	必须保持复位值。
6	WD0EIE	WD0E 中断使能 0: WD0E 中断禁能 1: WD0E 中断使能
5	EOCIE	EOC 中断使能 0: EOC 中断禁能 1: EOC 中断使能
4	保留	必须保持复位值。
3:0	WD0CHSEL[3:0]	模拟看门狗 0 通道选择 0000: ADC 通道 0 0001: ADC 通道 1 0010: ADC 通道 2 0011: ADC 通道 3 0100: ADC 通道 4 0101: ADC 通道 5 0110: ADC 通道 6 0111: ADC 通道 7 1000: ADC 通道 8 1001: ADC 通道 9 1010: ADC 通道 10 1011: ADC 通道 11 1100: ADC 通道 12 1101: ADC 通道 13 1110: ADC 通道 14 1111: ADC 通道 15 注意: 1. ADC 模拟输入通道 13、14 和 15 分别连接到温度传感器、V _{REFINT} 和 V _{REFP} 模拟输入。

11.6.3. 控制寄存器 1 (ADC_CTL1)

地址偏移: 0x08

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留							INREFEN	TSVEN	SWRCST	保留	ETERC	ETSRC [2:0]			保留
							rw	rw	rw				rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				DAL	保留		DMA	保留						CTN	ADCON
				rw			rw							rw	rw

位/位域	名称	说明
31:25	保留	必须保持复位值。
24	INREFEN	内部参考电压通道 (V_{REFINT}) 使能 0: 内部参考电压通道禁能 1: 内部参考电压通道使能 连接到 ADC 模拟输入通道 14
23	TSVEN	温度传感器输出电压通道使能 0: 温度传感器输出电压通道禁能 1: 温度传感器输出电压通道使能 连接到 ADC 模拟输入通道 13
22	SWRCST	常规序列软件启动转换 如果 ETSRC 是 111, 该位置'1'开启常规序列转换。软件置位, 软件清零, 或转换开始后, 由硬件清零。
21	保留	必须保持复位值。
20	ETERC	常规序列外部触发使能 0: 常规序列外部触发禁能 1: 常规序列外部触发使能
19:17	ETSRC[2:0]	常规序列外部触发选择 000: 定时器2 CH1 001: 定时器0 CH2 010: 定时器0 CH1 011: 定时器2 TRGO 100: 定时器0 CH0 101: 定时器2 CH0 110: 中断线 11 111: 软件触发
16:12	保留	必须保持复位值。
11	DAL	数据对齐 0: 最低有效位对齐 1: 最高有效位对齐
10:9	保留	必须保持复位值。
8	DMA	DMA 请求使能

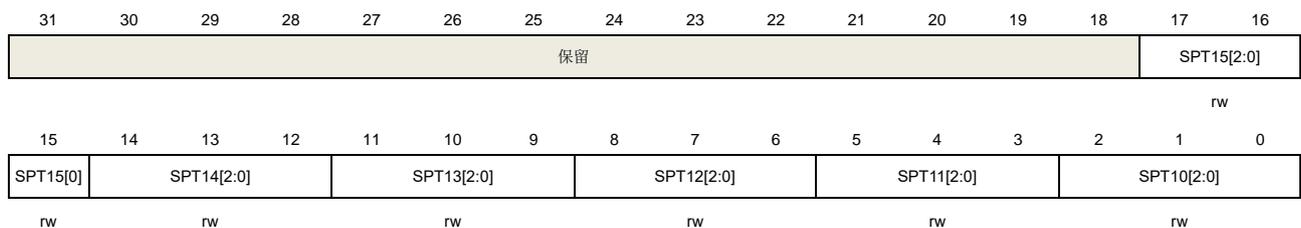
		0: DMA 请求禁能 1: DMA 请求使能
7:2	保留	必须保持复位值。
1	CTN	连续模式 0: 禁能连续运行模式 1: 使能连续运行模式
0	ADCON	开启 ADC 该位从'0'变成'1'将在稳定时间结束后唤醒 ADC。当该位被置位以后, 不改变寄存器的其他位仅仅对该位写'1', 将开启转换。 0: 禁能 ADC 关闭电源 1: 使能 ADC

11.6.4. 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。



位/位域	名称	说明
31:18	保留	必须保持复位值。
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2.5周期 001: 通道采样时间为3.5周期 010: 通道采样时间为7.5周期 011: 通道采样时间为12.5周期 100: 通道采样时间为19.5周期 101: 通道采样时间为39.5周期 110: 通道采样时间为79.5周期

111: 通道采样时间为 160.5 周期

11.6.5. 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	

位/位域	名称	说明
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2.5周期 001: 通道采样时间为3.5周期 010: 通道采样时间为7.5周期 011: 通道采样时间为12.5周期 100: 通道采样时间为19.5周期 101: 通道采样时间为39.5周期 110: 通道采样时间为79.5周期 111: 通道采样时间为 160.5 周期

11.6.6. 看门狗 0 高阈值寄存器 (ADC_WD0HT)

地址偏移: 0x24

复位值：0x0000 0FFF

该寄存器只能按字（32位）访问。



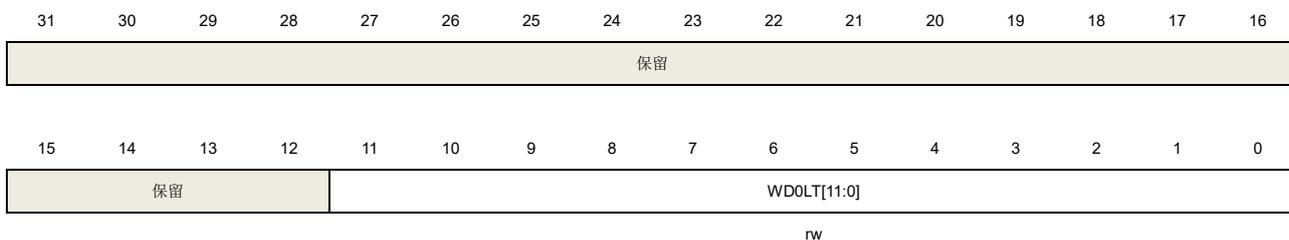
位/位域	名称	说明
31:12	保留	必须保持复位值。
11:0	WD0HT[11:0]	模拟看门狗 0 高侧阈值 这些位定义了模拟看门狗 0 的高阈值。

11.6.7. 看门狗 0 低阈值寄存器（ADC_WD0LT）

地址偏移：0x28

复位值：0x0000 0000

该寄存器只能按字（32位）访问。



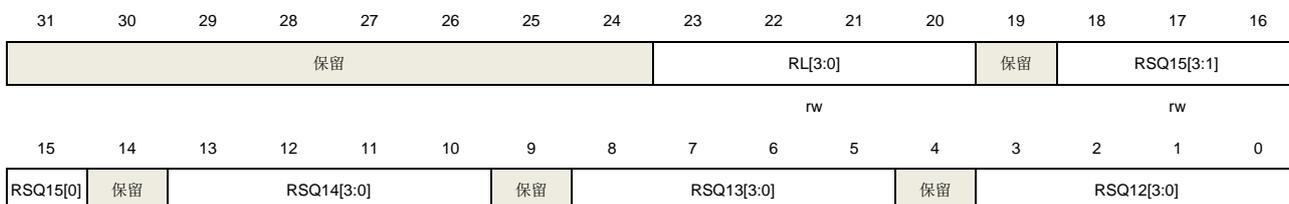
位/位域	名称	说明
31:12	保留	必须保持复位值。
11:0	WD0LT[11:0]	模拟看门狗 0 低侧阈值 这些位定义了模拟看门狗 0 的低阈值。

11.6.8. 常规序列寄存器 0（ADC_RSQ0）

地址偏移：0x2C

复位值：0x0000 0000

该寄存器只能按字（32位）访问。



位/位域	名称	说明
31:24	保留	必须保持复位值。
23:20	RL[3:0]	常规序列长度 常规转换序列中的总的通道数目为 RL[3:0]+1。
19	保留	必须保持复位值。
18:15	RSQ15[3:0]	参考 RSQ0[3:0]的描述
14	保留	必须保持复位值。
13:10	RSQ14[3:0]	参考 RSQ0[3:0]的描述
9	保留	必须保持复位值。
8:5	RSQ13[3:0]	参考 RSQ0[3:0]的描述
4	保留	必须保持复位值。
3:0	RSQ12[3:0]	参考 RSQ0[3:0]的描述

11.6.9. 常规序列寄存器 1 (ADC_RSQ1)

地址偏移: 0x30

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问。



位/位域	名称	说明
31:29	保留	必须保持复位值。
28:25	RSQ11[3:0]	参考 RSQ0[3:0]的描述
24	保留	必须保持复位值。
23:20	RSQ10[3:0]	参考 RSQ0[3:0]的描述
19	保留	必须保持复位值。
18:15	RSQ9[3:0]	参考 RSQ0[3:0]的描述
14	保留	必须保持复位值。

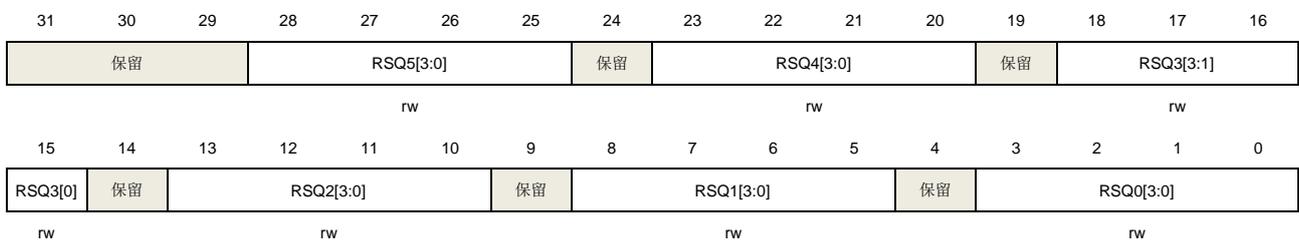
13:10	RSQ8[3:0]	参考 RSQ0[3:0]的描述
9	保留	必须保持复位值。
8:5	RSQ7[3:0]	参考 RSQ0[3:0]的描述
4	保留	必须保持复位值。
3:0	RSQ6[3:0]	参考 RSQ0[3:0]的描述

11.6.10. 常规序列寄存器 2 (ADC_RSQ2)

地址偏移: 0x34

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问。



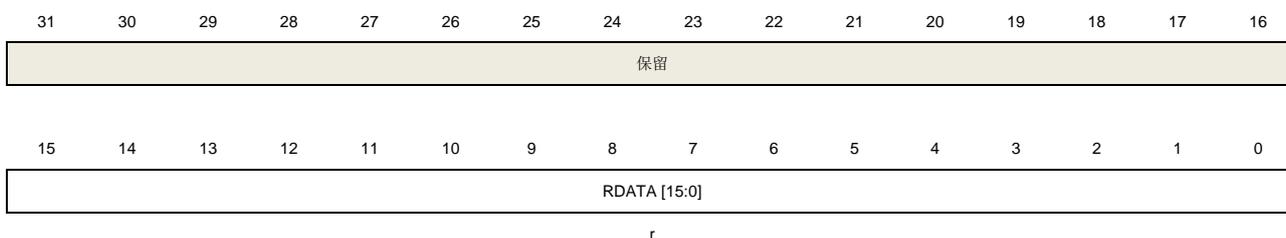
位/位域	名称	说明
31:30	保留	必须保持复位值。
29:25	RSQ5[3:0]	参考 RSQ0[3:0]的描述
24	保留	必须保持复位值。
23:20	RSQ4[3:0]	参考 RSQ0[3:0]的描述
19	保留	必须保持复位值。
18:15	RSQ3[3:0]	参考 RSQ0[3:0]的描述
14	保留	必须保持复位值。
13:10	RSQ2[3:0]	参考 RSQ0[3:0]的描述
9	保留	必须保持复位值。
8:5	RSQ1[3:0]	参考 RSQ0[3:0]的描述
4	保留	必须保持复位值。
3:0	RSQ0[3:0]	通道编号 写入这些位来选择常规序列的第 n 个转换的通道 (通道编号为 0..15)

11.6.11. 常规数据寄存器 (ADC_RDATA)

地址偏移: 0x4C

复位值：0x0000 0000

该寄存器只能按字（32位）访问。



r

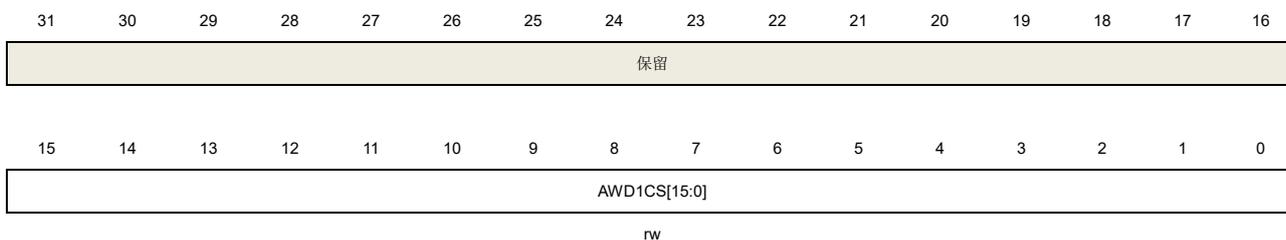
位/位域	名称	说明
31:16	保留	必须保持复位值。
15:0	RDATA[15:0]	常规通道数据 这些位包含了常规通道的转换结果，只读。

11.6.12. 看门狗 1 通道选择寄存器（ADC_WD1SR）

地址偏移：0x50

复位值：0x0000 0000

该寄存器只能按字（32位）访问。



rw

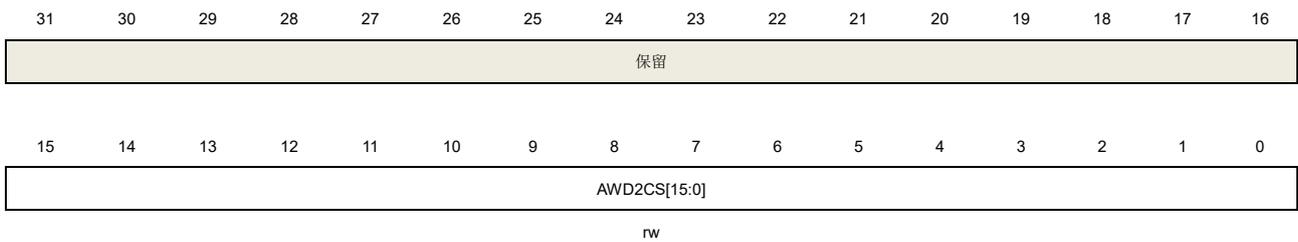
位/位域	名称	说明
31:16	保留	必须保持复位值。
15:0	AWD1CS[15:0]	模拟看门狗1通道选择 这些位由软件置位和复位，它们使能并选择要由模拟看门狗1保护的输入通道。 AWD1CS[n] = 0: ADC模拟输入通道n不由模拟看门狗1保护。 AWD1CS[n] = 1: ADC模拟输入通道n由模拟看门狗1保护。 当AWD1CS[15:0] = 000..0，模拟看门狗1禁能。 注意： 1) 通过AWD1CS位域配置的模拟看门狗1功能的通道，必须是ADC_RSQn寄存器和ADC_ISQ寄存器中配置的通道； 2) ADC模拟输入通道13、14和15分别连接到温度传感器、V _{REFINT} 和V _{REFP} 模拟输入。

11.6.13. 看门狗 2 通道选择寄存器（ADC_WD2SR）

地址偏移：0x54

复位值：0x00000000

该寄存器只能按字（32位）访问。



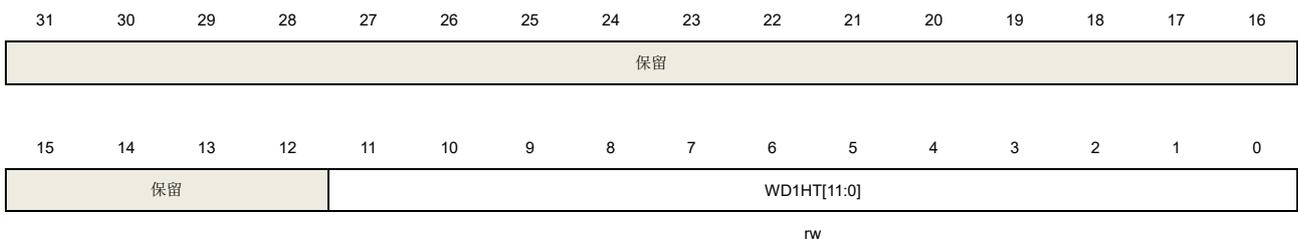
Bits	Fields	Descriptions
31:16	保留	必须保持复位值。
15:0	AWD2CS[15:0]	模拟看门狗2通道选择 这些位由软件置位和复位，它们使能并选择要由模拟看门狗2保护的输入通道。 AWD2CS[n] = 0: ADC模拟输入通道n不由模拟看门狗2保护。 AWD2CS[n] = 1: ADC模拟输入通道n由模拟看门狗2保护。 当AWD2CS[15:0] = 000..0，模拟看门狗2禁能。 注意： 1) 通过AWD2CS位域配置的模拟看门狗2功能的通道，必须是ADC_RSQn寄存器和ADC_ISQ寄存器中配置的通道； 2) ADC模拟输入通道13、14和15分别连接到温度传感器、V _{REFINT} 和V _{REFP} 模拟输入。

11.6.14. 看门狗 1 高阈值寄存器（ADC_WD1HT）

地址偏移：0x58

复位值：0x0000 0FFF

该寄存器只能按字（32位）访问。



位/位域	名称	说明
31:12	保留	必须保持复位值。
11:0	WD1HT[11:0]	模拟看门狗 1 高侧阈值 这些位定义了模拟看门狗1的高阈值。

11.6.15. 看门狗 1 低阈值寄存器（ADC_WD1LT）

地址偏移：0x5C

复位值：0x0000 0000

该寄存器只能按字（32位）访问。



位/位域	名称	说明
31:12	保留	必须保持复位值。
11:0	WD1LT[11:0]	模拟看门狗 1 低侧阈值 这些位定义了模拟看门狗1的低阈值。

11.6.16. 看门狗 2 高阈值寄存器 (ADC_WD2HT)

地址偏移: 0x60

复位值: 0x0000 0FFF

该寄存器只能按字 (32位) 访问。



位/位域	名称	说明
31:12	保留	必须保持复位值。
11:0	WD2HT[11:0]	模拟看门狗 2 高侧阈值 这些位定义了模拟看门狗2的高阈值。

11.6.17. 看门狗 2 低阈值寄存器 (ADC_WD2LT)

地址偏移: 0x64

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问。



位/位域	名称	说明
------	----	----

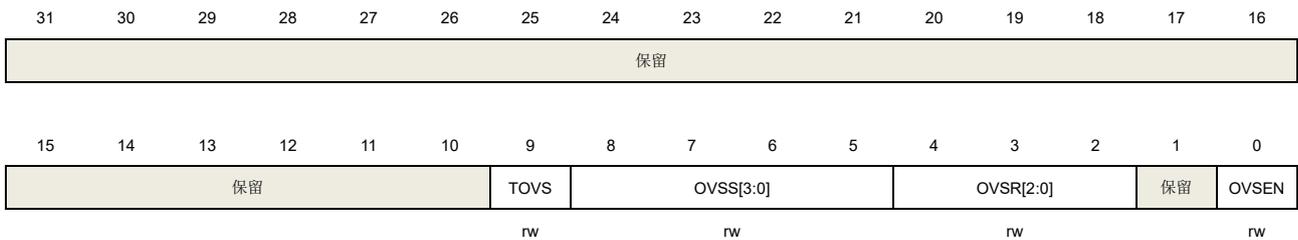
31:12	保留	必须保持复位值。
11:0	WD2LT[11:0]	模拟看门狗 2 低侧阈值 这些位定义了模拟看门狗2的低阈值。

11.6.18. 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。



位/位域	名称	说明
31:10	保留	必须保持复位值。
9	TOVS	过采样触发 该位通过软件置位和清除。 0: 在一次触发后连续执行过采样通道的所有转换 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: 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)。
8:5	OVSS[3:0]	过采样移位 该位通过软件置位和清除。 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 其余值都保留 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 (确保没有转换正在执行)。
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: 过采样禁能

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（确保没有转换正在执行）。

12. 看门狗定时器 (WDGT)

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

看门狗定时器在内部计数值达到了预设的门限时, 会触发一个复位 (对于窗口看门狗定时器来说, 会产生一个中断)。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。

12.1. 独立看门狗定时器 (FWDGT)

12.1.1. 简介

独立看门狗定时器 (FWDGT) 有独立时钟源 (IRC32K)。即使主时钟失效, FWDGT 依然能保持正常工作状态, 适用于需要独立环境且对计时精度要求不高的场合。

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

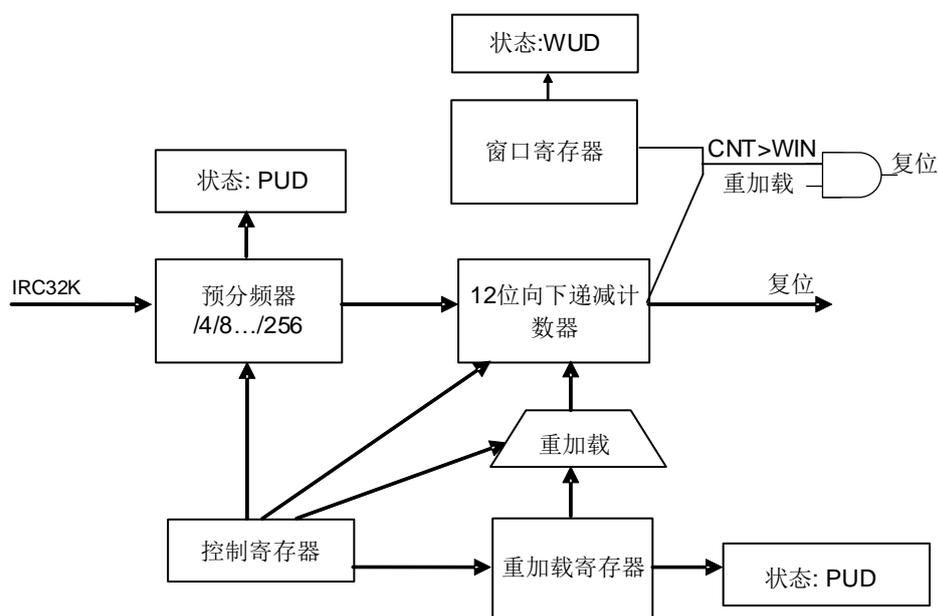
12.1.2. 主要特征

- 自由运行的12位向下计数器;
- 如果看门狗定时器被使能, 有以下两种情况下会产生复位:
 - 当计数器到0时产生复位;
 - 当计数器的值大于窗口寄存器的值时, 更新计数器会产生复位;
- 独立时钟源, 独立看门狗定时器在主时钟故障 (例如待机模式和深度睡眠模式下) 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。

12.1.3. 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12-1. 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 12-1. 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，产生一次系统复位。

在任何时候向FWDGT_CTL中写0xAAAA都可以重装载计数器，重装载值来源于重装载寄存器（FWDGT_RLD）。软件可以在计数器计数值达到0x000之前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0x0000FFFF，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）、FWDGT_WND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中控制寄存器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 12-1. 独立看门狗定时器在 32kHz (IRC32K) 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	0.03125	511.90625
1 / 8	001	0.03125	1023.78125
1 / 16	010	0.03125	2047.53125
1 / 32	011	0.03125	4095.03125
1 / 64	100	0.03125	8190.03125
1 / 128	101	0.03125	16380.03125
1 / 256	110或111	0.03125	32760.03125

通过校准IRC32K可以使自由看门狗定时器超时更加精确。

注意：当执行完喂狗reload操作之后，如需要立即进入deepsleep / standby模式时，必须通过软件设置，在reload命令及deepsleep / standby模式命令中间插入（3个以上）IRC32K时钟间隔。

12.1.4. FWDGT 寄存器

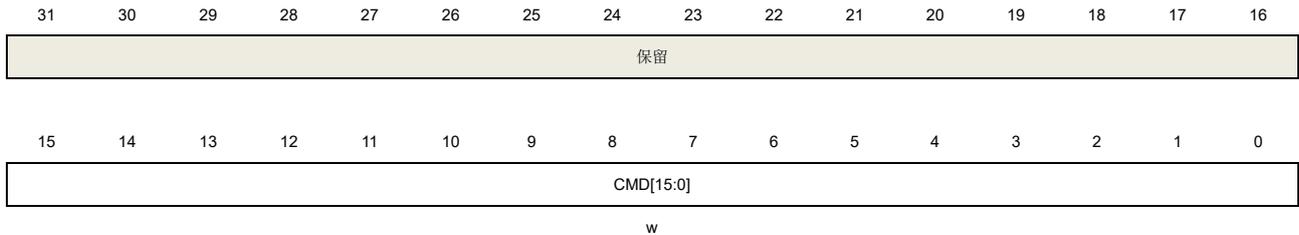
FWDGT 基地址: 0x4000 3000

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器定时计数器。计数减到0时产生中断 0xAAAA: 重装计数器

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

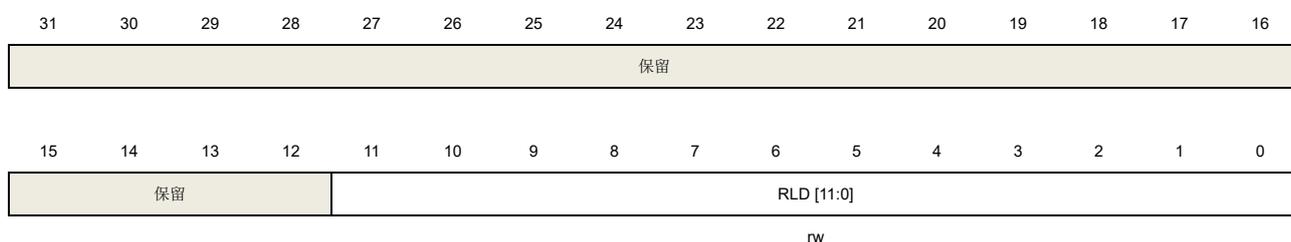
如果应用需要使用几个预分频系数，改变预分频值之前必须等到PUD位被清0。更新了预分频寄存器中的值后，在代码持续执行之前不必等待PUD值被清零。

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。



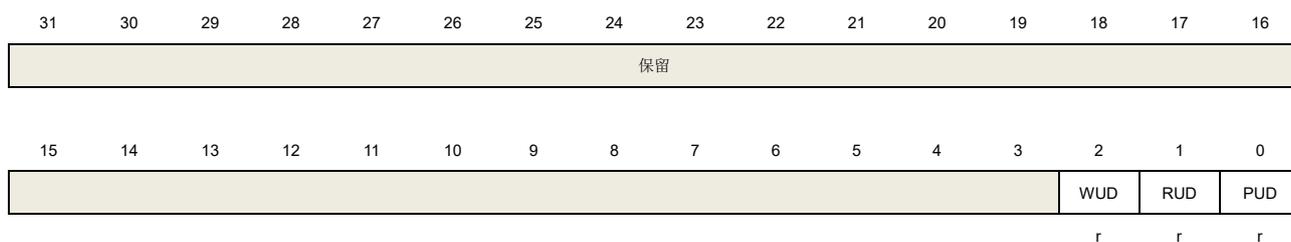
位/位域	名称	说明
31:12	保留	必须保持复位值。
11:0	RLD[11:0]	独立看门狗定时器定时计数器重装载值。向FWDGT_CTL寄存器写入0xAAAA的时候，这个值会被更新到看门狗定时器计数器中。 这些位有写保护功能。在写这些位之前需向FWDGT_CTL寄存器中写0x5555。在改写这个寄存器的过程中，FWDGT_STAT寄存器的RUD位被置1，从此寄存器中读取的任何值都是无效的。 如果应用需要使用几个重装载值，改变重装载值之前必须等到RUD位被清0。更新了重装载寄存器的值后，在代码持续执行之前不必等待RUD值被清零（在进入省电模式前需等待RUD值清零）。

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。



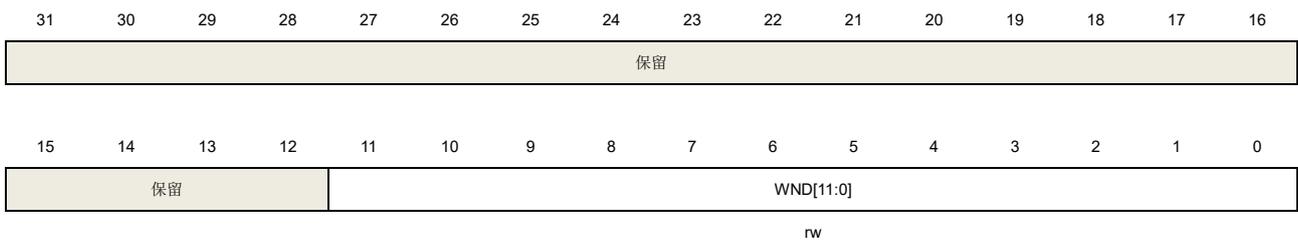
位/位域	名称	说明
31:3	保留	必须保持复位值。
2	WUD	独立看门狗定时器计数器窗口值更新。 FWDGT_WND寄存器写操作时，该位被置1，此时读取FWDGT_WND寄存器的任何值都是无效的。
1	RUD	独立看门狗定时器计数器重装载值更新。 FWDGT_RLD寄存器写操作时，该位被置1，此时读取FWDGT_RLD寄存器的任何值都是无效的。在FWDGT_RLD寄存器更新后，该位由硬件清零。
0	PUD	独立看门狗定时器预分频值更新。 FWDGT_PSC寄存器写操作时，该位被置1，此时读取FWDGT_PSC寄存器的任何值都是无效的。在FWDGT_PSC寄存器更新后，该位由硬件清零。

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。

12.2. 窗口看门狗定时器（WWDGT）

12.2.1. 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，7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[6]位被清0）。在计数器计数值达到窗口寄存器值之前，计数器的更新也会产生系统复位。因此软件需要在给定的区间内更新计数器。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，如果使能中断将会产生提前唤醒中断。

窗口看门狗定时器时钟是由APB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。

12.2.2. 主要特征

- 可编程的7位自由运行向下递减计数器。
- 当窗口看门狗使能后，有以下两种情况会产生复位：
 - 当计数器达到0x3F时产生复位；
 - 当计数器的值大于窗口寄存器的值时，更新计数器会产生复位。
- 提前唤醒中断（EWI）：如果看门狗定时器打开，中断允许，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。

12.2.3. 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的时候产生系统复位（CNT[6]位被清0）。或是在计数值达到窗口寄存器值之前，更新计数器也会产生系统复位。

图 12-2. 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，除非通过重置，否则不能再次被关闭。当通过硬件选择窗口看门狗时，看门狗定时器复位后一直处于开启状态，不能关闭。窗口看门狗定时器打开后，计数器始终递减计数，计数器配置的值应该大于0x3F，也就是说CNT[6]位应该被置1。CNT[5:0]决定了两次重

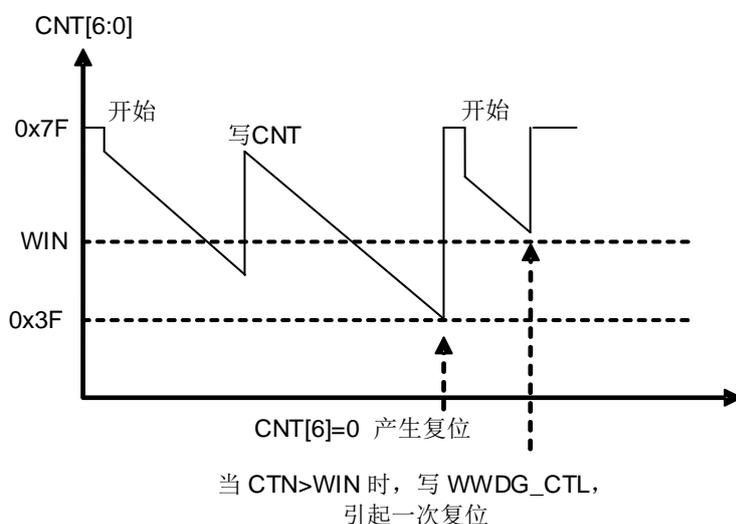
装载之间的最大间隔时间。计数器的递减速度取决于APB时钟和预分频器（WWDGT_CFG寄存器的PSC[3:0]位）。

配置寄存器（WWDGT_CFG）中的WIN[6:0]位用来设定窗口值。当计数器的值小于窗口值，且大于0x3F的时候，重装载向下计数器可以避免复位，否则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 12-3. 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：

$$t_{\text{WWDGT}} = t_{\text{PCLK1}} \times 4096 \times 2^{\text{PSC}} \times (\text{CNT}[5:0] + 1) \quad (\text{ms}) \quad (12-1)$$

其中：

t_{WWDGT} : 窗口看门狗定时器的超时时间

t_{PCLK1} : APB以ms为单位的时钟周期

t_{WWDGT} 的最大值和最小值请参考[表12-2. 在48MHz \(fPCLK1\) 时的最大/最小超时值](#)。

表 12-2. 在 48MHz (fPCLK1) 时的最大/最小超时值

Prescaler divider	PSC[3:0]	Min timeout value CNT[6:0] = 0x40	Max timeout value CNT[6:0] = 0x7F
1 / 1	0000	85.33μs	5.461ms
1 / 2	0001	170.67μs	10.923ms
1 / 4	0010	341.33μs	21.845ms
1 / 8	0011	682.67μs	43.691ms
1 / 16	0100	1.365ms	87.382ms

Prescaler divider	PSC[3:0]	Min timeout value CNT[6:0] = 0x40	Max timeout value CNT[6:0] = 0x7F
1 / 32	0101	2.731ms	174.764ms
1 / 64	0110	5.461ms	349.528ms
1 / 128	0111	10.922ms	699.056ms
1 / 256	1000	21.854ms	1398.112ms
1 / 512	1001	43.691ms	2796.224ms
1 / 1024	1010	87.381ms	5592.448ms
1 / 2048	1011	174.763ms	11184.896ms
1 / 4096	1100	349.525ms	22369.792ms
1 / 8192	1101	699.051ms	44739.584ms
1 / 1	1110	85.33μs	5.461ms
1 / 1	1111	85.33μs	5.461ms

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计数。

12.2.4. WWDGT 寄存器

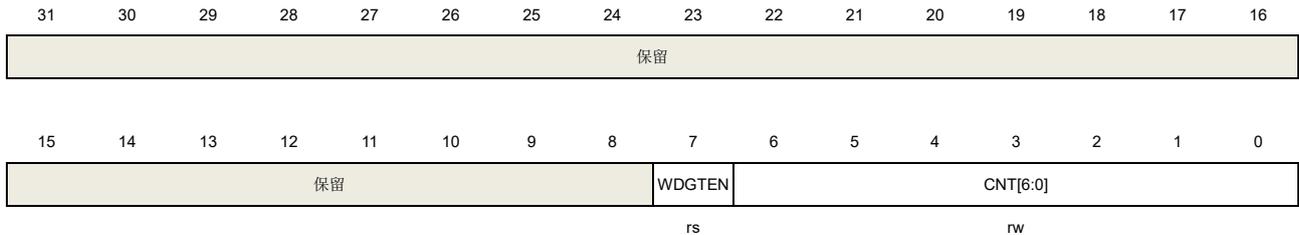
WWDGT 基地址：0x4000 2C00

控制寄存器 (WWDGT_CTL)

地址偏移：0x00

复位值：0x0000 007F

该寄存器可以按半字（16 位）或字（32 位）访问。



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

配置寄存器 (WWDGT_CFG)

地址偏移：0x04

复位值：0x0000 007F

该寄存器可以按半字（16 位）或字（32 位）访问。



位/位域	名称	说明
31:18	保留	必须保持复位值。
17:16	PSC[3:2]	预分频器，这些位和PSC[1:0]共同决定看门狗定时器的时间基准。 0000：(PCLK1 / 4096) / 1 0001：(PCLK1 / 4096) / 2 0010：(PCLK1 / 4096) / 4

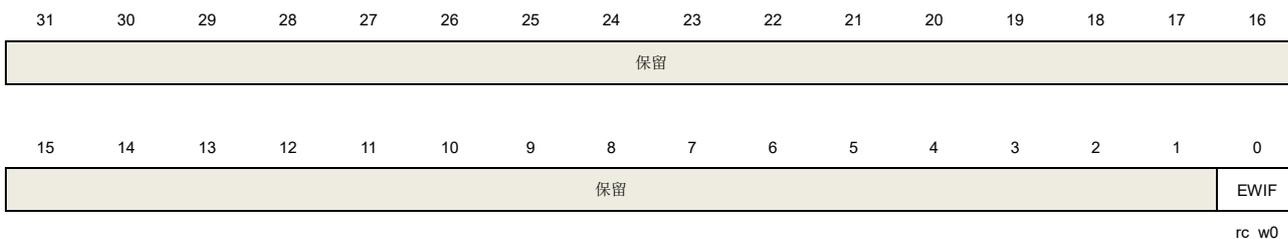
		0011: (PCLK1 / 4096) / 8
		0100: (PCLK1 / 4096) / 16
		0101: (PCLK1 / 4096) / 32
		0110: (PCLK1 / 4096) / 64
		0111: (PCLK1 / 4096) / 128
		1000: (PCLK1 / 4096) / 256
		1001: (PCLK1 / 4096) / 512
		1010: (PCLK1 / 4096) / 1024
		1011: (PCLK1 / 4096) / 2048
		1100: (PCLK1 / 4096) / 4096
		1101: (PCLK1 / 4096) / 8192
		1110: (PCLK1 / 4096) / 1
		1111: (PCLK1 / 4096) / 1
15:10	保留	必须保持复位值。
9	EWIE	提前唤醒中断使能。如果该位被置1，计数值达到0x40时触发中断。该位由硬件复位清0，或通过置位RCU模块的WWDGTRST位进行软件复位。写0没有任何作用。
8:7	PSC[1:0]	预分频器，这些位和PSC[3:2]共同决定看门狗定时器的时间基准。
6:0	WIN[6:0]	窗口值，当看门狗定时器计数器的值大于窗口值时，写看门狗定时器计数器（WWDGT_CTL的CNT位）会产生系统复位。

状态寄存器 (WWDGT_STAT)

地址偏移: 0x08

复位值: 0x0000 0000

该寄存器可以按半字（16位）或字（32位）访问。



位/位域	名称	说明
31:1	保留	必须保持复位值。
0	EWIF	提前唤醒中断标志位。当计数值达到0x40，即使中断没有被使能（WWDGT_CFG中的EWIE位为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。

13. 实时时钟 (RTC)

13.1. 简介

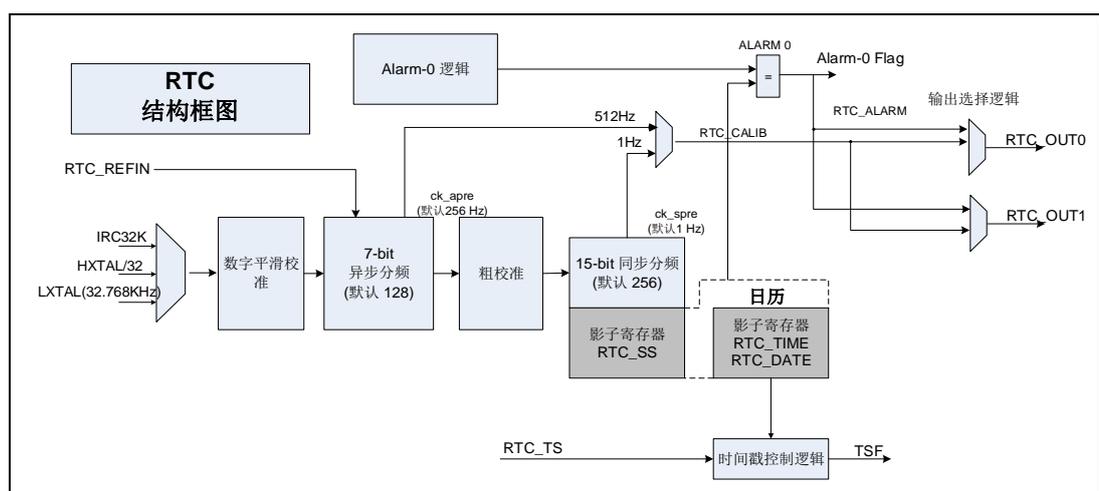
RTC 模块提供了一个包含日期 (年/月/日) 和时间 (时/分/秒/亚秒) 的日历功能。除亚秒用二进制码显示外, 时间和日期都以 BCD 码的形式显示。RTC 可以进行夏令时补偿。RTC 可以工作在省电模式下。RTC 支持外接更高精度的低频时钟, 用以达到更高的日历精度。

13.2. 主要特征

- 通过软件设置来实现夏令时补偿。
- 参考时钟检测功能: 通过外接更高精度的低频率时钟源 (50Hz或60Hz) 来提高日历精度。
- 数字校准功能: 通过调整最小时间单位 (最大可调精度0.95ppm) 来进行日历校准。
- 通过移位功能进行亚秒级调整。
- 记录事件时间的时间戳功能。
- 可编程的日历和一个位域可屏蔽的闹钟。
- 可屏蔽的中断源:
 - 闹钟 0;
 - 时间戳检测;
- 4个16位 (共16字节) 通用备份寄存器, 能够在省电模式下保存数据。

13.3. 功能描述

图 13-1. RTC 结构框图



RTC 单元包括:

- 闹钟事件/中断。
- 可选的 RTC 输出功能:
 - 512Hz (默认预分频值): RTC_OUT0(PC13: 48 引脚封装 / PA4: 20/28/32 引脚封装)

- 装) 或 RTC_OUT1(PA4);
- 1Hz (默认预分频值): RTC_OUT0(PC13: 48 引脚封装 / PA4: 20/28/32 引脚封装) 或 RTC_OUT1(PA4);
- 闹钟事件 (极性可配置): RTC_OUT0(PC13: 48 引脚封装 / PA4: 20/28/32 引脚封装) 或 RTC_OUT1(PA4);
- 可选的RTC输入功能:
 - 时间戳事件检测: RTC_TS (PC13: 48 引脚封装 / PA4: 20/28/32 引脚封装);
 - 参考时钟输入: RTC_REFIN (PB15/PB7);

13.3.1. 时钟源和预分频

RTC 单元有三个可选的独立时钟源: LXTAL、IRC32K 和 HXTAL 的 32 (由 RCU_CTL1 寄存器配置) 分频后的时钟。

在 RTC 单元, 有两个预分频器用来实现日历功能和其他功能。一个分频器是 7 位异步预分频器, 另一个是 15 位同步预分频器。异步分频器主要用来降低功率消耗。如果两个分频器都被使用, 建议异步分频器的值尽可能大。

两个预分频器的频率计算公式如下:

$$f_{ck_apre} = \frac{f_{rtcclk}}{FACTOR_A + 1} \quad (13-1)$$

$$f_{ck_spre} = \frac{f_{ck_apre}}{FACTOR_S + 1} = \frac{f_{rtcclk}}{(FACTOR_A + 1) * (FACTOR_S + 1)} \quad (13-2)$$

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

13.3.2. 影子寄存器

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

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

系统复位将复位影子寄存器。

13.3.3. 位域可屏蔽可配置的闹钟

RTC 闹钟功能被划分为多个位域并且每一个位域有一个该域的可屏蔽位。

RTC 闹钟功能的使能由 RTC_CTL 寄存器中的 ALRMxEN (x=0) 位控制。当 ALRMxEN=1 (x=0)

并且闹钟所有位域的值与对应的日历时间值匹配，ALRMxF (x=0) 标志位将会置位。

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

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

13.3.4. RTC 初始化和配置

RTC 寄存器写保护

在默认情况下, PMU_CTL0 寄存器的 BKPWEN 位被清 0。所以写 RTC 寄存器前需要软件提前设置 BKPWEN 位。

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

通过下面的步骤, 可以解锁这些保护:

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

写一个错误的值到 RTC_WPK 会使写保护再次生效。

备份寄存器复位后, 一些 RTC 寄存器被写保护: RTC_TIME, RTC_DATE, RTC_CTL, RTC_STAT, RTC_PSC, RTC_ALRM0TD, RTC_HRFC, RTC_SHIFTCTL, RTC_ALRMOSS。

日历初始化和配置

通过以下步骤可以设置日历和预分频器的值:

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

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

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

YCM 标志表明日历是否完成初始化, 该标志会硬件检查日历的年份值。

夏令时

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

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

闹钟功能操作步骤

为了避免意外的闹钟标记置位和亚稳态，闹钟功能的操作应遵循如下流程：

1. 清除寄存器 RTC_CTL 的 ALRMxEN (x=0) 位，禁用闹钟；
2. 设置 Alarm 寄存器 (RTC_ALRMxTD/RTC_ALRMxSS (x=0))；
3. 设置寄存器 RTC_CTL 的 ALRMxEN (x=0) 位，使能闹钟功能。

13.3.5. 读取日历

当 BPSHAD=0 时，读日历寄存器

当 BPSHAD=0，从影子寄存器读日历的值。由于同步机制的存在，正常读取日历需要满足一个基本要求：APB总线时钟频率必须大于或等于RTC时钟频率的7倍。在任何情况下APB总线时钟的频率都不能低于RTC的时钟频率。例如，如果系统时钟使用LXTAL，RTC时钟就不能选择HXTAL的32分频，因为HXTAL的32分频大于LXTAL时钟频率。

当APB总线时钟频率低于7倍RTC时钟频率时，日历的读取应该遵守以下流程：

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

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

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

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

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

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

1. 系统复位之后；
2. 日历初始化之后；
3. 一次移位操作之后。

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

当 BPSHAD=1 时，读日历寄存器

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

时钟周期)。

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

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

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

13.3.6. RTC 复位

在RTC单元，有两个复位源可用：系统复位和备份寄存器复位。

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

备份寄存器复位将会影响下面的寄存器，但系统复位不会对它们产生影响：

- RTC 真实的日历寄存器；
- RTC 控制寄存器 (RTC_CTL)；
- RTC 预分频寄存器 (RTC_PSC)；
- RTC 高精度频率补偿寄存器 (RTC_HRFC)；
- RTC 移位控制寄存器 (RTC_SHIFTCTL)；
- RTC 时间戳寄存器 (RTC_SSTS/RTC_TTS/RTC_DTS)；
- RTC 闹钟寄存器 (RTC_ALRMxSS/RTC_ALRMxTD(x=0))。
- RTC 备份寄存器 (RTC_BKPx)；

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

13.3.7. RTC 移位功能

当用户有一个高精度的远程时钟而且RTC1Hz时钟(ck_spre)和远程时钟只有一个亚秒级的偏差，RTC单元提供一个称作移位的功能去消除这个偏差来提高秒钟的精确性。

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

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

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

注意：在使用移位功能之前，软件必须检查RTC_SS中SSC的第15位(SSC[15])并确保该位为0。写RTC_SHIFTCTL寄存器之后，RTC_STAT寄存器的SOPF位将会再次置位。当同步移位操作完成时，SOPF位被硬件清0。系统复位不影响SOPF位。当REFEN=0时，移

位操作才能正确的工作。如果 REFEN=1，软件禁止写入 RTC_SHIFTCTL。

13.3.8. RTC 参考时钟检测

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

使能这项功能之后（REFEN=1），每一个秒更新的时钟（1Hz）边沿将与最近的RTC_REFIN参考时钟沿进行对比。在大多数情况下，这两个时钟沿是对齐的。但当两个时钟沿由于LXTAL准确度的原因没有对齐的时候，RTC参考时钟的检测功能会偏移1Hz时钟沿一点相位，使得下一个1Hz时钟沿和参考时钟沿对齐。

当REFEN=1，每一秒前后都会有一个进行检测的时间窗，处于不同的检测状态，时间窗时长也不同。当检测状态处于检测第一个参考时钟边沿时，使用7个ck_apre时长的时间窗，当检测状态处于边沿对齐操作时，使用3个ck_apre时长的时间窗。

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

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

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

待机模式下，参考时钟检测功能不可用。

13.3.9. RTC 数字平滑校准

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

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

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

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

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

因此使用FREQI可以使RTC频率增加488.5ppm。

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

当数字平滑校准功能正在运行时，按如下公式计算输出校准频率：

$$f_{cal} = f_{rtclock} \times \left(1 + \frac{FREQI \times 512 - CMSK}{2^N + CMSK - FREQI \times 512} \right) \quad (13-3)$$

注意：N=20/19/18（32/16/8秒）校准时间周期。

当 FACTOR_A < 3 时校准：

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

当FACTOR_A小于3时，FACTOR_S值应小于标称值。假设RTC时钟频率是正常的32.768KHz，对应的FACTOR_S应该按下面所示设置：

FACTOR_A = 1：FACTOR_S减少4（32.768KHz下16379）

FACTOR_A = 0：FACTOR_S减少8（32.768KHz下32759）

当FACTOR_A小于3，CMSK为0x100，校准频率公式如下：

$$f_{cal} = f_{rtclock} \times \left(1 + \frac{256 - CMSK}{2^N + CMSK - 256} \right) \quad (13-4)$$

注意：N=20/19/18（32/16/8秒）校准时间周期。

验证 RTC 校准

提供1Hz校准时钟的输出用于协助软件测量并验证RTC的精度。

在有限的测量周期内测量RTC的频率，最高可能发生2个RTCCLK的测量误差。

为了消除这一测量误差，测量周期应该和校准周期一致。

- 校准周期设为32秒（默认配置）
用准确的32秒周期去测量1Hz校准输出的准确性能保证这个测量误差在0.477ppm（在32秒周期内0.5个RTCCLK）之内。
- 校准周期设为16秒（通过设置CWND16位）
使用此配置，CMSK[0]被硬件置0。
用准确的16秒周期去测量1Hz校准输出的准确性能保证这个测量误差在0.954ppm（在16秒周期内0.5个RTCCLK）之内。
- 校准周期设为8秒（通过设置CWND8位）
使用此配置，CMSK[1: 0]被硬件置0。
用准确的8秒周期去测量1Hz校准输出的准确性能保证这个测量误差在1.907ppm（在8秒周期内0.5个RTCCLK）之内。

运行中重校准

当INITF位是0，用下面的步骤，软件可以更新RTC_HRFC：

1. 等待SCPF位置0；
2. 写一个新的值到RTC_HRFC寄存器；

3. 3 个 ck_apre 时钟周期之后，新的校准设置开始生效。

13.3.10. 时间戳功能

时间戳功能由RTC_TS管脚输入，通过配置TSEN位来使能。

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

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

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

13.3.11. 校准时钟输出

如果COEN位设置为1，RTC_OUT0/1会输出参考校准时钟。

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

当COS位设置为1时，RTC_CALIB的频率计算公式为：

$$f_{rtc_calib} = \frac{f_{rtcclk}}{(FACTOR_A+1) \times (FACTOR_S+1)} \quad (13-5)$$

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

13.3.12. 闹钟输出

当OS控制位不为0x00时，RTC_ALARM复用输出功能被启用。这个功能将直接输出RTC_STAT寄存器的闹钟标志ALRMxF。

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

13.3.13. RTC 引脚配置

RTC_OUT，RTC_TS都使用同一个(PC13 /PA4)引脚。无论(PC13 /PA4)的GPIO是什么配置(PC13 /PA4)的功能由RTC控制。

(PC13 /PA4)的输出优先级如表[表13-1. RTC \(PC13 /PA4\)引脚配置](#)

表 13-1. RTC (PC13 /PA4)引脚配置

功能配置 和引脚功能		OS[1:0] (输出选择)	COEN (校准输出)	OUT1EN	ALARMOUT TYPE (闹钟输出类型)	DISPU	TSEN (时间戳使能)
闹钟开漏 输出	无上拉	01 或 10 或 11	-	0	0	0	-
			1	1			
	内部上 拉	01 或 10 或 11	-	0	0	1	-
			1	1			
闹钟推挽输出		01 或 10 或 11	-	0	1	0	-
			1	1			
校准推挽输出		00	1	0	-	-	-
时间戳浮空输入		00	0	-	-	-	1
唤醒引脚或者标准 GPIO		00	0	-	-	-	0
		00	1	1			
		-	0				

PC13 /PA4引脚可以用于以下功能：

- RTC_ALARM 输出：通过 RTC_CTL 寄存器 OS [1:0] 位域配置
- RTC_CALIB 输出：通过 RTC_CTL 寄存器 COEN [23] 位配置
- RTC_TS：时间戳事件检测

RTC_TYPE寄存器中ALRMOUTTYPE位可以选择RTC_ALARM输出是开漏还推挽

配置RTC_CTL寄存器的OUT1EN位，可以在PA4引脚输出RTC_OUT1信号。

表 13-2. RTC_OUT 配置

OS [1:0] ALARM 输出使能	COEN (校准输出使能)	OUT1EN	RTC_OUT0	RTC_OUT1
00	0	0	-	-
00	1		CALIB	-
01 or 10 or 11	-		ALRM	-
00	0	1	-	-
00	1		-	CALIB
01 or 10 or 11	0		-	ALRM
01 or 10 or 11	1		ALRM	CALIB

表 13-3. RTC 低功耗模式管理

模式	模式下能否工作	退出该模式的方法
睡眠模式	是	RTC中断
深度睡眠	当时钟源是LXTAL或IRC32K时可以工作	RTC闹钟/时间戳事件
待机模式	当时钟源是LXTAL或IRC32K时可以工作	RTC闹钟/时间戳事件

13.3.14. RTC 中断

所有的RTC中断都被连接到EXTI控制器。

如果想使用RTC闹钟/时间戳中断，应按下面步骤操作：

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

表 13-4. RTC 中断控制

中断	事件标志	控制位	退出睡眠模式	退出深度睡眠模式和待机模式
闹钟0	ALRM0F	ALRM0IE	Y	Y ⁽¹⁾
时间戳	TSF	TSIE	Y	Y ⁽¹⁾

(1) . 仅当RTC时钟源为LXTAL或IRC32K时，才可以从深度睡眠和待机模式唤醒。

13.4. RTC 寄存器

RTC基地址：0x4000 2800

13.4.1. 时间寄存器（RTC_TIME）

偏移地址：0x00

系统复位值：当BPSHAD = 0, 0x0000 0000

当BPSHAD = 1, 无影响

写保护寄存器，仅在初始化状态可以进行写操作

该寄存器只能按字（32位）访问。

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

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

13.4.2. 日期寄存器（RTC_DATE）

偏移地址：0x04

系统复位值：当 BPSHAD = 0, 0x0000 2101

当 BPSHAD = 1, 无影响

写保护寄存器，仅在初始化状态可以进行写操作。

该寄存器只能按字（32位）访问。

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

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

13.4.3. 控制寄存器（RTC_CTL）

偏移地址：0x08

系统复位：无影响

备份寄存器复位值：0x0000 0000

写保护寄存器

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OUT1EN	保留							COEN	OS[1:0]		OPOL	COS	DSM	S1H	A1H
rw								rw	rw		rw	rw	rw	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSIE	保留	ALRMOIE	TSEN	保留	ALRMOEN	保留	CS	BPSHAD	REFEN	TSEG	保留				
rw		rw	rw		rw		rw	rw	rw	rw					

位/位域	名称	描述
31	OUT1EN	RTC_OUT 引脚选择 0: RTC_OUT1 输出失能 1: RTC_OUT1 输出使能

30:24	保留	必须保持复位值。
23	COEN	校准输出使能 0: 关闭校准输出 1: 使能校准输出
22:21	OS[1:0]	输出选择 该位用来选择输出的标志源。 0x00: 禁用 RTC_ALARM 输出 0x01: 启用闹钟 0 标志输出
20	OPOL	输出极性 该位用来反转 RTC_ALARM 输出。 0: 禁用反转 RTC_ALARM 输出 1: 启用反转 RTC_ALARM 输出
19	COS	校准输出选择 仅当 COEN=1 并且预分频器是默认值时有效。 0: 校准输出是 512Hz 1: 校准输出是 1Hz
18	DSM	夏令时屏蔽位 该位可以通过软件灵活使用。常用来记录夏令时调整。
17	S1H	减 1 小时（冬季时间变化） 当前时间非零的情况下，将当前时间减去一个小时。 0: 没有影响 1: 在下一个秒改变时，将减少一个小时
16	A1H	增加 1 小时（夏季时间变化） 将当前时间增加一个小时。 0: 没有影响 1: 在下一个秒改变时，将增加一个小时
15	TSIE	时间戳中断使能 0: 禁用时间戳中断 1: 启用时间戳中断
14:13	保留	必须保持复位值。
12	ALRM0IE	RTC 闹钟 0 中断使能 0: 禁用闹钟中断 1: 启用闹钟中断
11	TSEN	时间戳功能使能 0: 禁用时间戳功能 1: 启用时间戳功能
10:9	保留	必须保持复位值。
8	ALRM0EN	闹钟 0 功能使能

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

13.4.4. 状态寄存器 (RTC_STAT)

偏移地址: 0x0C

系统复位: 仅INITM, INITF和RSYNF位被置0, 其他位无影响。

备份寄存器复位值: 0x0000 0007

写保护寄存器

该寄存器只能按字 (32 位) 访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															SCPF
															r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		TSOVRF	TSF	保留		ALRM0F	INITM	INITF	RSYNF	YCM	SOPF	保留		ALRM0WF	
		rc_w0	rc_w0			rc_w0	rw	r	rc_w0	r	r			r	

位/位域	名称	描述
31:17	保留	必须保持复位值。
16	SCPF	平滑校准挂起标志 在未进入初始化模式时向 RTC_HRFC 进行软件写操作, 该位被硬件置 1。当平滑校准设置开始执行后, 该位被硬件清零 0。

15:13	保留	必须保持复位值。
12	TSOVRF	时间戳事件溢出标志 如果 TSF 位已经置位，当再次检测到时间戳事件时，该位会通过硬件置 1。 可以通过向该位软件写 0 来清除。
11	TSF	时间戳事件标志 当检测到一个时间戳事件时，该位会通过硬件置 1。可以通过向该位软件写 0 来清除。
10:9	保留	必须保持复位值。
8	ALRM0F	Alarm0 发生标志 当现在的时间/日期与闹钟 0 设置的时间/日期匹配的时候，该位会通过硬件置 1。 可以通过向该位软件写 0 来清除。
7	INITM	进入初始化模式 0: 自由运行模式 1: 进入初始化模式设置时间/日期和预分频，计数器将停止运行
6	INITF	初始化状态标志 该位被硬件置 1，初始化状态时可以设置日历寄存器和预分频器。 0: 日历寄存器和预分频器的值不能改变 1: 日历寄存器和预分频器的值可以改变
5	RSYNF	寄存器同步标志 每 2 个 RTCCLK 将会由硬件置 1 一次，同时会复制当前日历时间/日期到影子日历寄存器。初始化模式 (INITM)，移位操作挂起标志 (SOPF) 或者禁止影子寄存器模式 (BPSHAD = 1) 会清除该位。该位也可以通过软件写 0 清除。 0: 影子寄存器未同步 1: 影子寄存器已同步
4	YCM	年份配置标志 当日历寄存器的年份值不为 0 时硬件置 1 0: 日历尚未初始化 1: 日历已经初始化
3	SOPF	移位功能操作挂起标志 0: 移位操作没有挂起 1: 移位操作挂起
2:1	保留	必须保持复位值。
0	ALRM0WF	Alarm0 配置可写标志 硬件置位和清零。ALRM0EN=0 时，标记 alarm 是否可写。 0: 不允许修改 Alarm 寄存器设置 1: 允许修改 Alarm 寄存器设置

13.4.5. 预分频寄存器 (RTC_PSC)

偏移地址：0x10

系统复位：无影响

备份寄存器复位值：0x007F 00FF

写保护寄存器，仅在初始化状态可以进行写操作。

该寄存器只能按字（32位）访问。

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

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

13.4.6. 闹钟 0 时间日期寄存器 (RTC_ALRM0TD)

偏移地址：0x1C

系统复位：无影响

备份寄存器复位值：0x0000 0000

写保护寄存器，仅在初始化状态可以进行写操作或者ALRM0WF为1。

该寄存器只能按字（32位）访问。

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MSKD	DOWS	DAYT[1:0]		DAYU[3:0]			MSKH	PM	HRT[1:0]		HRU[3:0]					
MSKM	MNT[2:0]		MNU[3:0]			MSKS	SCT[2:0]		SCU[3:0]							

位/位域	名称	描述
31	MSKD	闹钟日期位域屏蔽位 0: 不屏蔽日期/天位域 1: 屏蔽日期/天位域
30	DOWS	星期选择

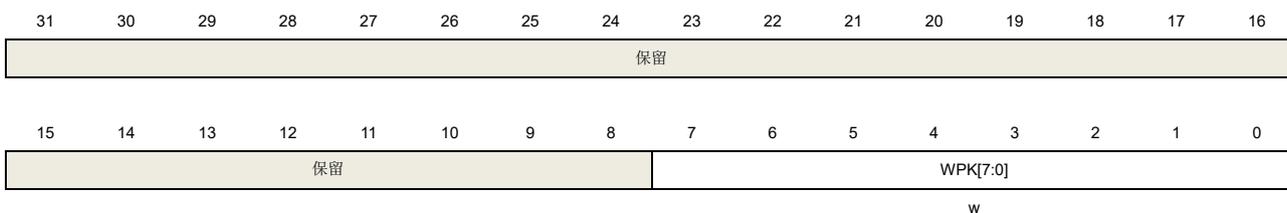
		0: 此时 DAYU[3: 0]代表日期个位值
		1: 此时 DAYU[3: 0]代表星期几, 此时 DAYT[1: 0]无意义
29:28	DAYT[1:0]	日期十位值, 以 BCD 码格式存储
27:24	DAYU[3:0]	日期个位值或星期天数, 以 BCD 码格式存储
23	MSKH	闹钟小时位域屏蔽位 0: 不屏蔽小时位域 1: 屏蔽小时位域
22	PM	AM/PM 标志 0: AM 或 24 小时制 1: PM
21:20	HRT[1:0]	小时十位值, 以 BCD 码形式存储
19:16	HRU[3:0]	小时个位值, 以 BCD 码形式存储
15	MSKM	闹钟分钟位域屏蔽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

13.4.7. 写保护钥匙寄存器 (RTC_WPK)

偏移地址: 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 WPK[7:0] 写保护的解锁值

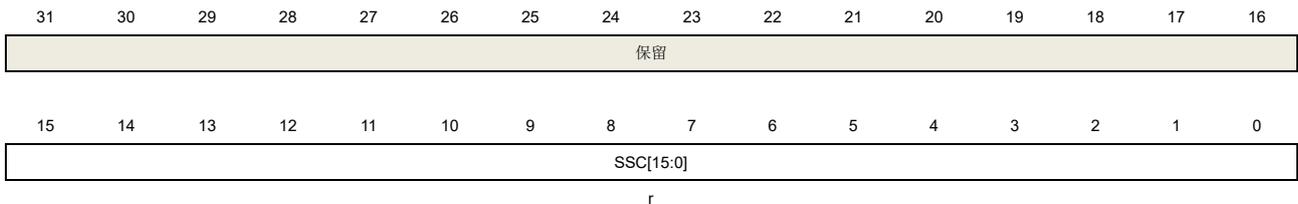
13.4.8. 亚秒寄存器 (RTC_SS)

偏移地址: 0x28

系统复位值: 当BPSHAD = 0, 0x0000 0000。

当BPSHAD = 1, 无影响。

该寄存器只能按字 (32位) 访问。



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

13.4.9. 移位控制寄存器 (RTC_SHIFTCTL)

偏移地址: 0x2C

系统复位: 无影响

备份寄存器复位值: 0x0000 0000

写保护寄存器, 仅当SOPF=0, 该寄存器可写。

该寄存器只能按字 (32 位) 访问。



位/位域	名称	描述
31	A1S	增加一秒 0: 无影响 1: 增加一秒到时钟/日历 该位与 SFS 位一起使用, 增加小于一秒到当前时间。
30:15	保留	必须保持复位值。
14:0	SFS[14:0]	减去小于一秒的一段时间

这位的值将增加到同步预分频计数器

当仅用 SFS 时，由于同步预分频器是一个递减计数器，所以时钟将会延迟。

延迟（秒）= SFS / (FACTOR_S + 1)

当 A1S 和 SFS 一起使用时，时钟将会提前

提前（秒）= (1 - (SFS / (FACTOR_S + 1)))

注意：写入此寄存器会导致 RSYNF 位被清 0。

13.4.10. 时间戳时间寄存器 (RTC_TTS)

偏移地址：0x30

备份寄存器复位值：0x0000 0000

系统复位：无影响

当TSF被置1，该位用来记录日历时间。

清除TSF位也会清除此寄存器。

该寄存器只能按字（32 位）访问。

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

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

13.4.11. 时间戳日期寄存器 (RTC_DTS)

偏移地址：0x34

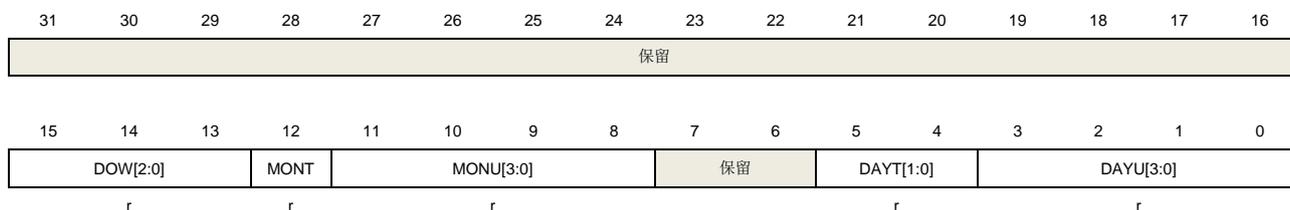
备份寄存器复位值：0x0000 0000

系统复位：无影响

当TSF被置1，该位用来记录日历日期。

清除TSF位也会清除此寄存器。

该寄存器只能按字（32位）访问。



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

13.4.12. 时间戳亚秒寄存器（RTC_SSTS）

偏移地址：0x38

备份寄存器复位值：0x0000 0000

系统复位：无影响

当TSF被置1，该位用来记录日历时间。

清除TSF位也会清除此寄存器。

该寄存器只能按字（32位）访问。



位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	SSC[15:0]	亚秒值

TSF 置 1 时记录当时的同步预分频计数器的值。

13.4.13. 高精度频率补偿寄存器 (RTC_HRFC)

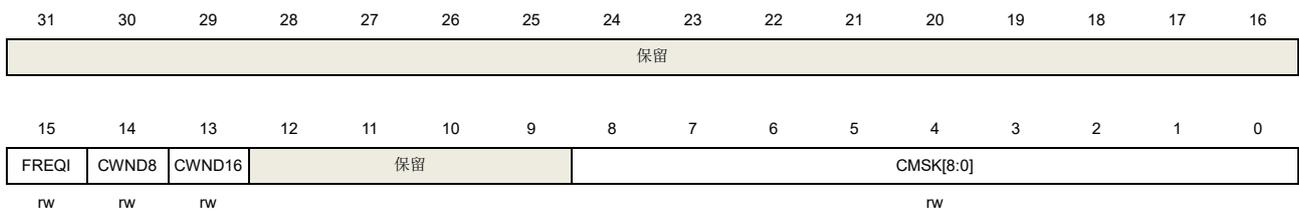
偏移地址: 0x3C

备份寄存器复位: 0x0000 0000

系统复位: 无影响

写保护寄存器。

该寄存器只能按字 (32位) 访问。



位/位域	名称	描述
31:16	保留	必须保持复位值。
15	FREQI	RTC 频率增加 488.5ppm 0: 无影响 1: 每 2^{11} 个脉冲增加一个 RTCCLK 脉冲 该位需与 CMSK 位一起使用。如果输入时钟频率是 32.768KHz, 在 32s 校准窗期间, 增加的 RTCCLK 脉冲数是 $(512 * FREQI) - CMSK$
14	CWND8	采用 8 秒校准周期 0: 无影响 1: 采用 8 秒校准周期 注意: 当 CWND8=1, CMSK[1: 0]被锁定在“00”。
13	CWND16	采用 16 秒校准周期 0: 无影响 1: 采用 16 秒校准周期 注意: 当 CWND16=1, CMSK[0] 被锁定在“0”。
12:9	保留	必须保持复位值。
8:0	CMSK[8:0]	校准周期 RTCCLK 脉冲屏蔽数 在 2^{20} 个 RTCCLK 脉冲之内屏蔽的脉冲数 此项功能可以以 0.9537 ppm 的分辨率来降低日历频率

13.4.14. 类型寄存器 (RTC_TYPE)

偏移地址: 0x40

备份寄存器复位: 0x0000 0000

系统复位: 无影响

该寄存器只能按字（32位）访问。

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

位/位域	名称	描述
31:19	保留	必须保持复位值。
18	ALRMOUTTYPE	RTC_ALARM 输出类型 0: 开漏输出 1: 推挽输出
17:16	保留	必须保持复位值。
15	DISPU	RTC_ALARM 上拉禁用位 0: RTC_ALARM 无输出上拉 1: RTC_ALARM 输出上拉
14:0	保留	必须保持复位值。

13.4.15. 闹钟 0 亚秒寄存器（RTC_ALRM0SS）

偏移地址： 0x44

备份寄存器复位： 0x0000 0000

系统复位： 无影响

写保护寄存器，仅当ALRM0EN=0或INITM=1，可以进行写操作。

该寄存器只能按字（32位）访问。

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

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

0x6: SSC[5: 0]位用于时间匹配, 其他位被忽略。
 0x7: SSC[6: 0]位用于时间匹配, 其他位被忽略。
 0x8: SSC[7: 0]位用于时间匹配, 其他位被忽略。
 0x9: SSC[8: 0]位用于时间匹配, 其他位被忽略。
 0xA: SSC[9: 0]位用于时间匹配, 其他位被忽略。
 0xB: SSC[10: 0]位用于时间匹配, 其他位被忽略。
 0xC: SSC[11: 0]位用于时间匹配, 其他位被忽略。
 0xD: SSC[12: 0]位用于时间匹配, 其他位被忽略。
 0xE: SSC[13: 0]位用于时间匹配, 其他位被忽略。
 0xF: SSC[14: 0]位用于时间匹配, 其他位被忽略。
 注意: 同步预分频计数器的第 15 位 (RTC_SS 寄存器中的 SSC[15]) 从不被匹配。

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

13.4.16. 备份寄存器 (RTC_BKPx) (x=0..3)

偏移地址: 0x50 到 0x5c
 备份寄存器复位: 0x0000 0000
 系统复位: 无影响

该寄存器只能按字 (32位) 访问。



位/位域	名称	描述
15:0	DATA[15:0]	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。

14. 定时器 (TIMER)

表 14-1. 定时器 (TIMERx) 分为四种类型

定时器	定时器 0	定时器 2	定时器 13	定时器 15/16
类型	高级	通用 L0	通用 L2	通用 L4
预分频器	16 位	16 位	16 位	16 位
计数器	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
可重复性	•	×	×	•
捕获/比较通道数	4	4	1	1
互补和死区时间	3	×	×	•
中止输入	•	×	×	•
单脉冲	•	•	×	•
正交译码器	•	•	×	×
主-从管理	•	•	×	×
内部连接	• ⁽¹⁾	• ⁽²⁾	×	×
DMA	•	•	×	•
Debug 模式	•	•	•	•

(1) TIMER0 ITI0: 保留 ITI1: 保留 ITI2: TIMER2_TRGO ITI3: TIMER16_CH1

(2) TIMER2 ITI0: TIMER0_TRGO ITI1: 保留 ITI2: 保留 ITI3: TIMER13_CH1

(3) TIMER13 ITI0: 保留 ITI1: 保留 ITI2: 保留 ITI3: 保留

(4) TIMER15 ITI0: 保留 ITI1: 保留 ITI2: 保留 ITI3: 保留
TIMER16 ITI0: 保留 ITI1: 保留 ITI2: 保留 ITI3: 保留

14.1. 高级定时器 (TIMERx, x=0)

14.1.1. 简介

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

高级定时器是可编程的，可以用来计数，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。

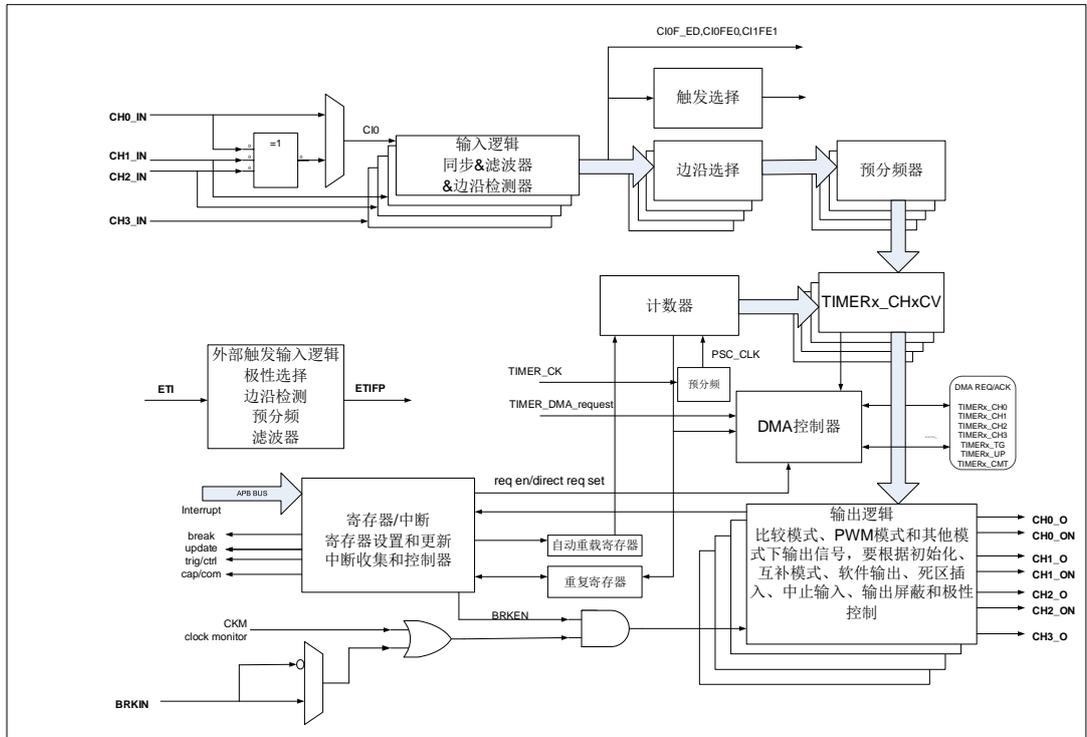
14.1.2. 主要特征

- 总通道数：4；
- 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/从模式管理。

14.1.3. 结构框图

图 14-1. 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 14-1. 高级定时器结构框图



注意：CH4_O 是内部信号，无对应引脚。

14.1.4. 功能说明

时钟源选择

高级定时器的时钟源可以是内部时钟源 CK_TIMER, 或者是由 TSCFGy[2:0]位确定的时钟源, TSCFGy[2:0]位于 SYSCFG_TIMER0CFG, (y=0,1...7)。

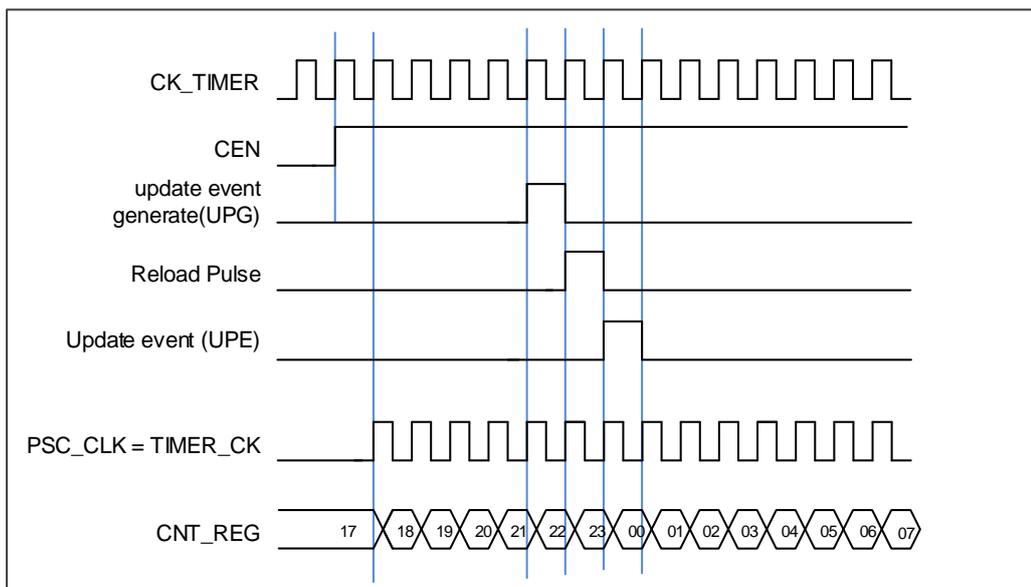
- TSCFGy[2:0] = 3'b000, TSCFGy[2:0]位于SYSCFG_TIMER0CFG, (y=0,1...7), 定时器选择内部时钟源 (连接到RCU模块的CK_TIMER)

当 TSCFGy[2:0] = 3'b000, TSCFGy[2:0]位于 SYSCFG_TIMER0CFG, (y=0,1...7), 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 TSCFGy[2:0] != 3'b000, TSCFGy[2:0]位于SYSCFG_TIMER0CFG, (y=0,1,2,6), 预分频器被其他时钟源 (由 TSCFG6[2:0]区域选择) 驱动, 更多细节在下文说明, 当 TSCFGy[2:0] (y=3,4,5) 设置为有效值时, 计数器预分频器时钟源由内部时钟TIMER_CK 驱动。

图 14-2. 内部时钟分频为 1 时，正常模式下的控制电路



- **TSCFG6[2:0] != 3'b000**（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 **TSCFG6[2:0]**为 **0x4, 0x5** 或 **0x6** 来选择。

计数器预分频器也可以在内部触发信号 **ITI0/2/3** 的上升沿计数。这种模式可以通过设置 **TSCFG6[2:0]**为 **0x1, 0x2, 0x3**。

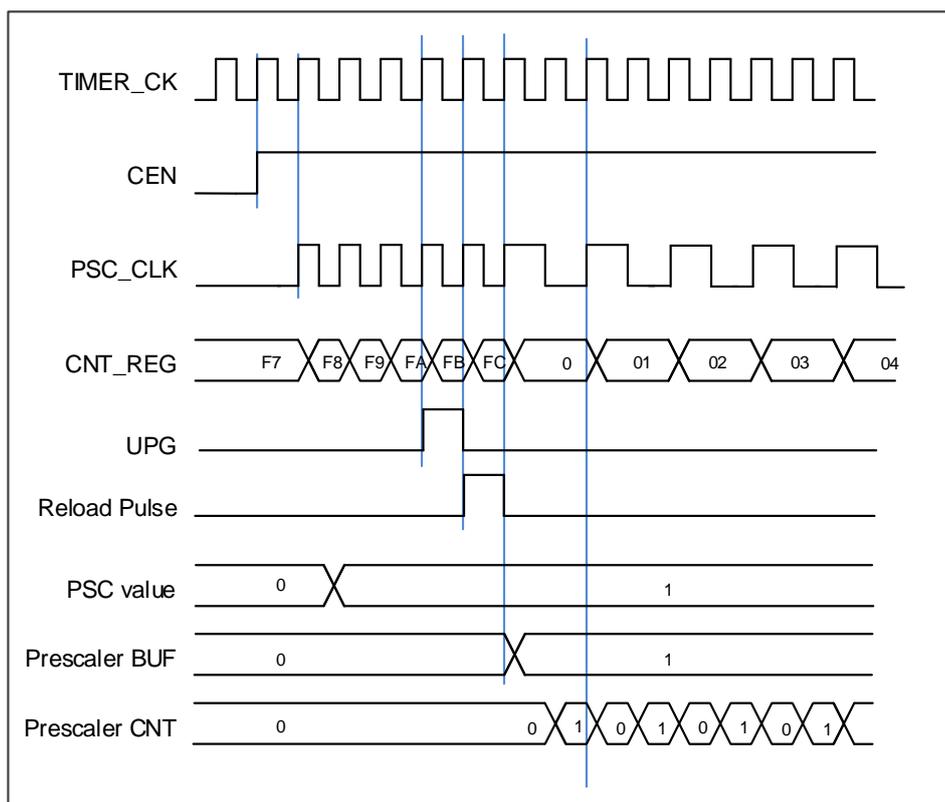
- **SMC1=1'b1**（外部时钟模式1），定时器选择外部输入引脚**ETI**作为时钟源。

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 **TSCFG6[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。

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次更新事件到来时被采用。

图 14-3. 当预分频器的参数从 1 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。如果设置了重复计数器，在 `(TIMERx_CREP+1)` 次上溢后产生更新事件，否则在每次上溢时都会产生更新事件。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，所有的寄存器（重复计数寄存器，自动重载寄存器，预分频寄存器）都将被更新。

[图14-4. 向上计数时序图, PSC=0/1](#)和 [图14-5. 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=0x63` 时, 计数器在不同预分频因子下的行为。

图 14-4. 向上计数时序图, PSC=0/1

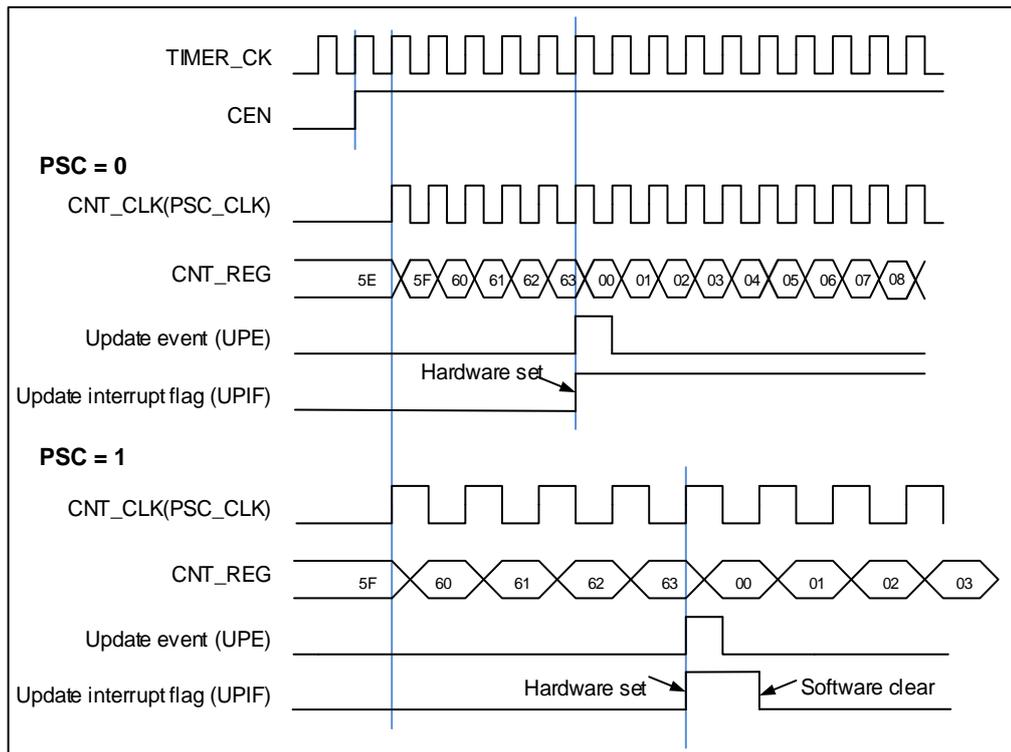
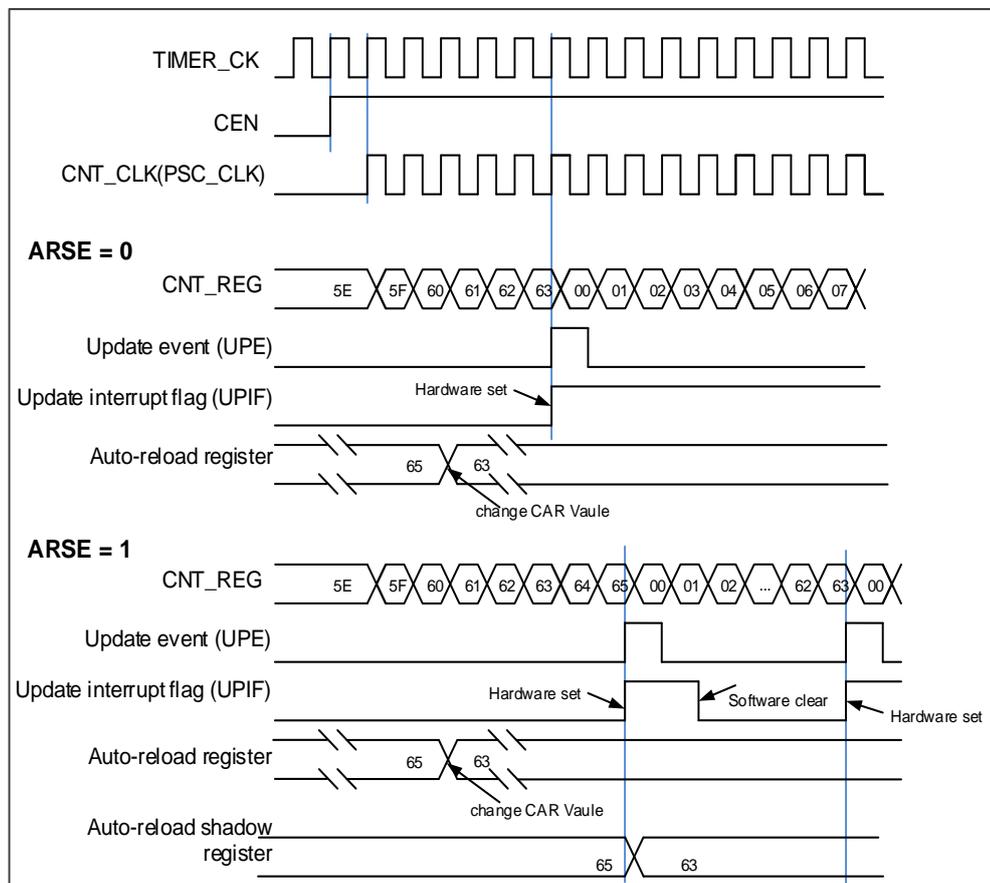


图 14-5. 向上计数时序图, 在运行时改变 TIMERx_CAR 寄存器的值



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。如果设置了重复计数器，在 `(TIMERx_CREP+1)` 次下溢后产生更新事件，否则在每次下溢时都会产生更新事件。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14-6. 向下计数时序图, PSC=0/1](#)和 [图14-7. 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=0x63` 时，计数器在不同时钟频率下的行为。

图 14-6. 向下计数时序图，PSC=0/1

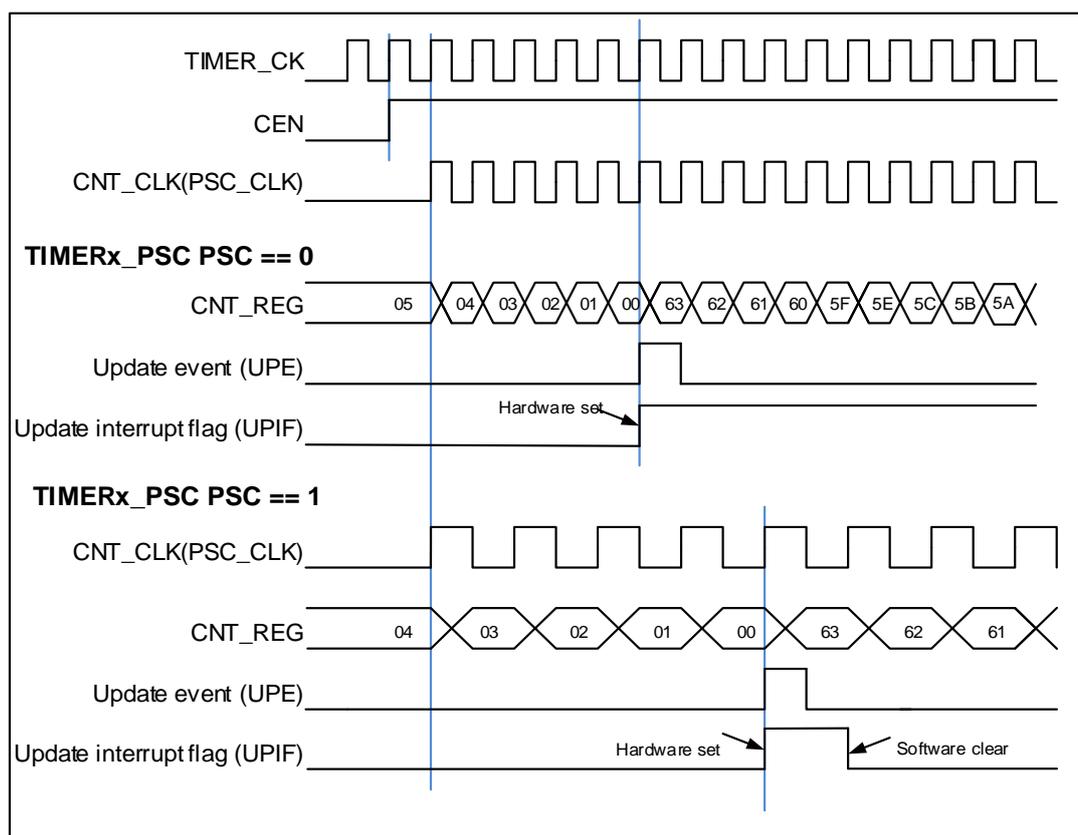
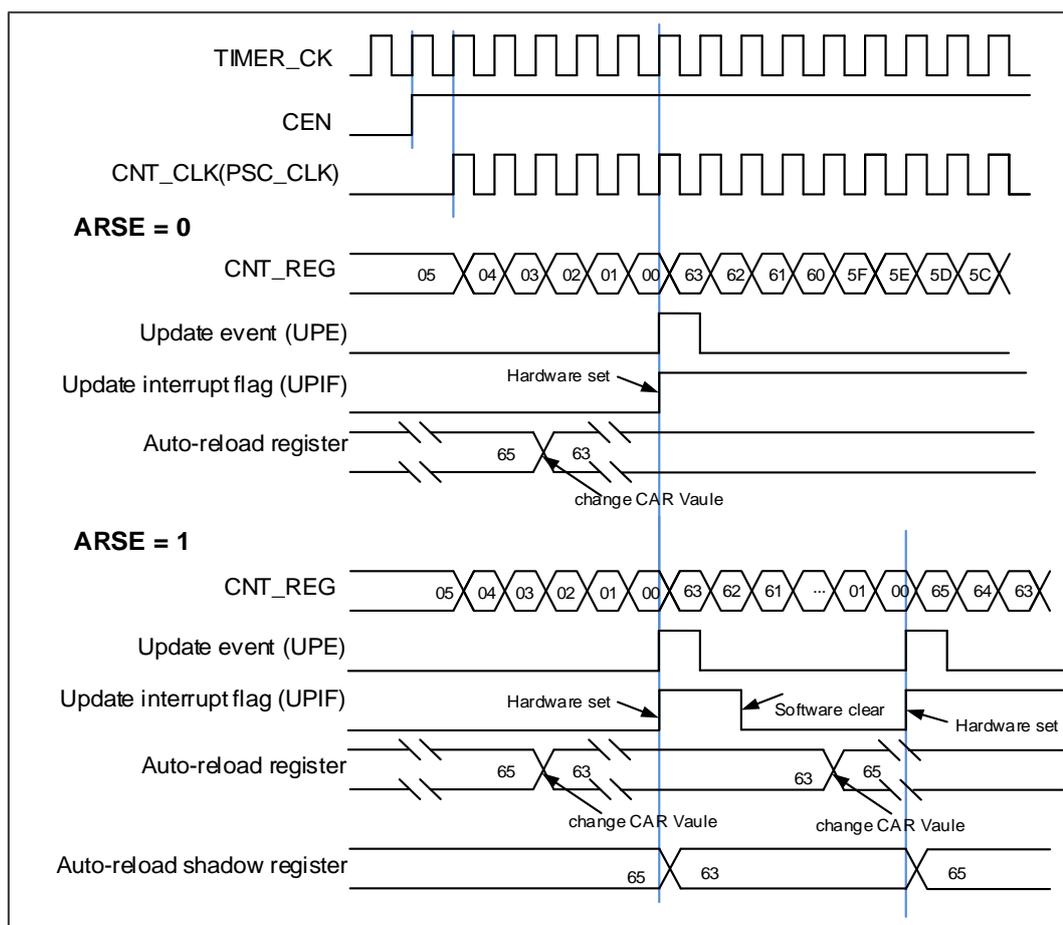


图 14-7. 向下计数时序图，在运行时改变 `TIMERx_CAR` 寄存器值

计数器中央对齐模式

在中央对齐模式下，计数器交替的从 0 开始向上计数到自动加载值，然后再向下计数到 0。向上计数模式中，定时器模块在计数器计数到 (`TIMERx_CREP - 1`) 时产生一个上溢事件；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。

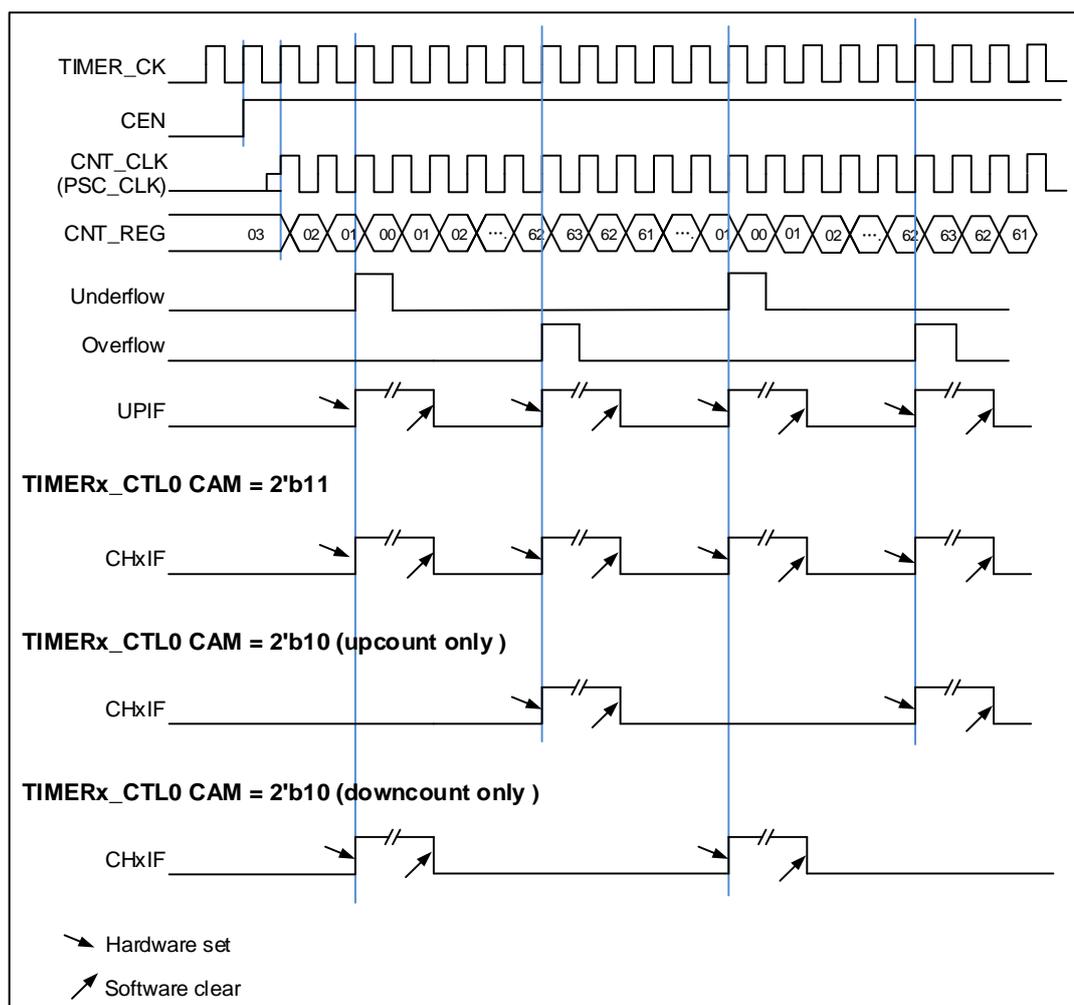
上溢或者下溢时，`TIMERx_INTF` 寄存器中的 `UPIF` 位都会被置 1。但是 `CHxIF` 位是否置 1 与 `TIMERx_CTL0` 寄存器中 `CAM` 的值有关。具体细节参考 [图 14-8. 中央计数模式计数器时序图](#)。

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 14-8. 中央计数模式计数器时序图](#) 给出了一些例子，当 `TIMERx_CAR=0x63`，`TIMERx_PSC=0x0` 时，计数器的时序图。

图 14-8. 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 14-9. 中央计数模式下计数器重复时序图

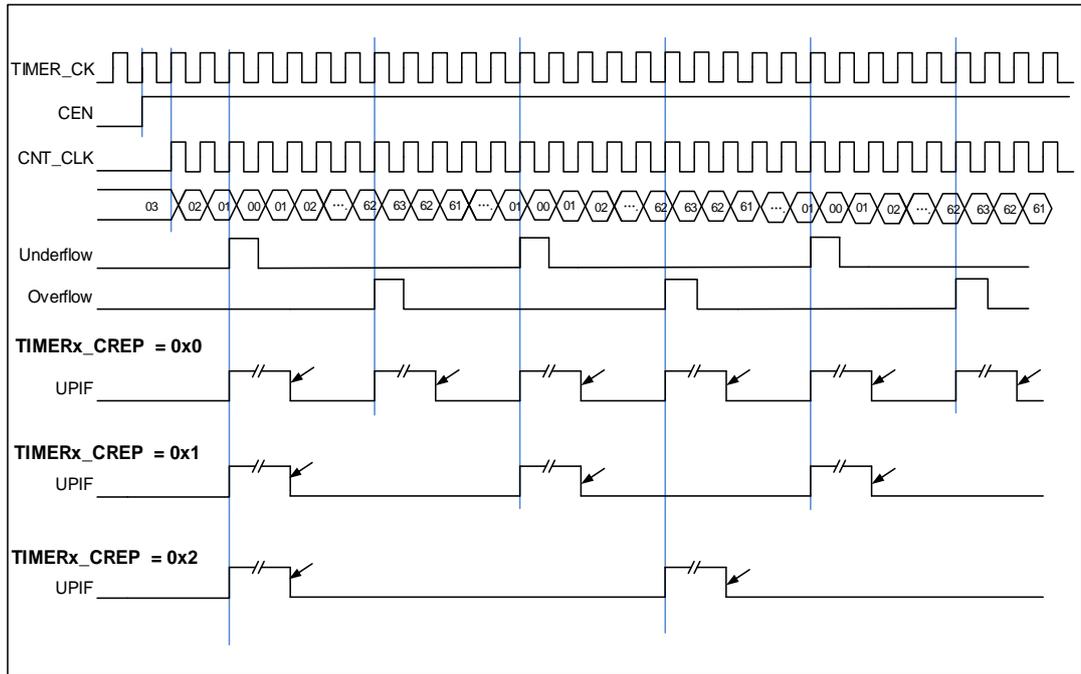


图 14-10. 在向上计数模式下计数器重复时序图

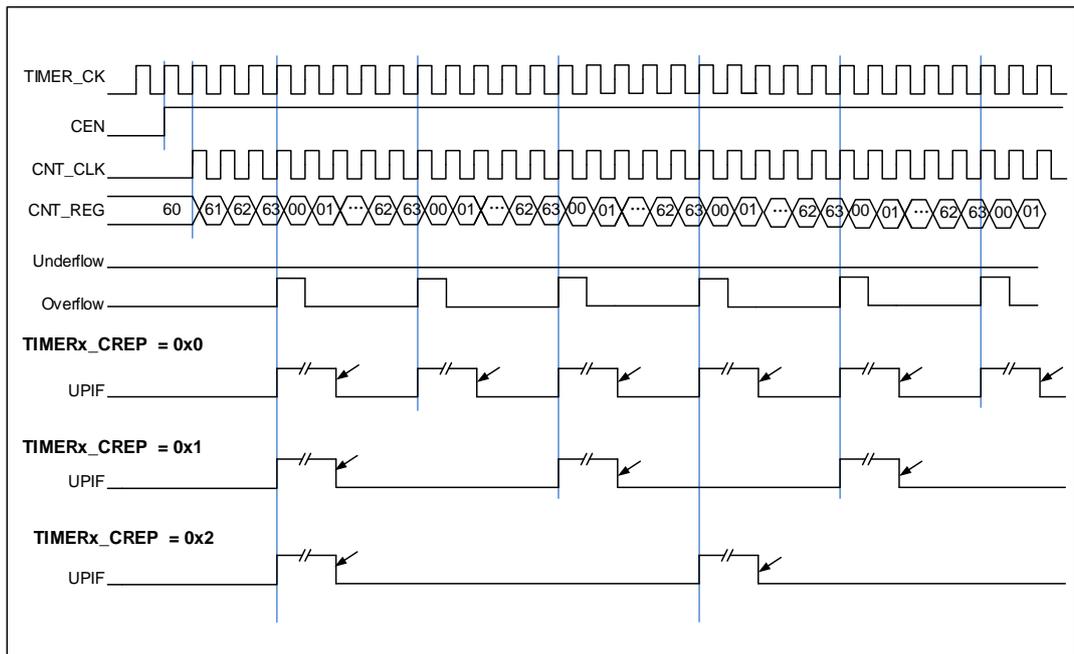
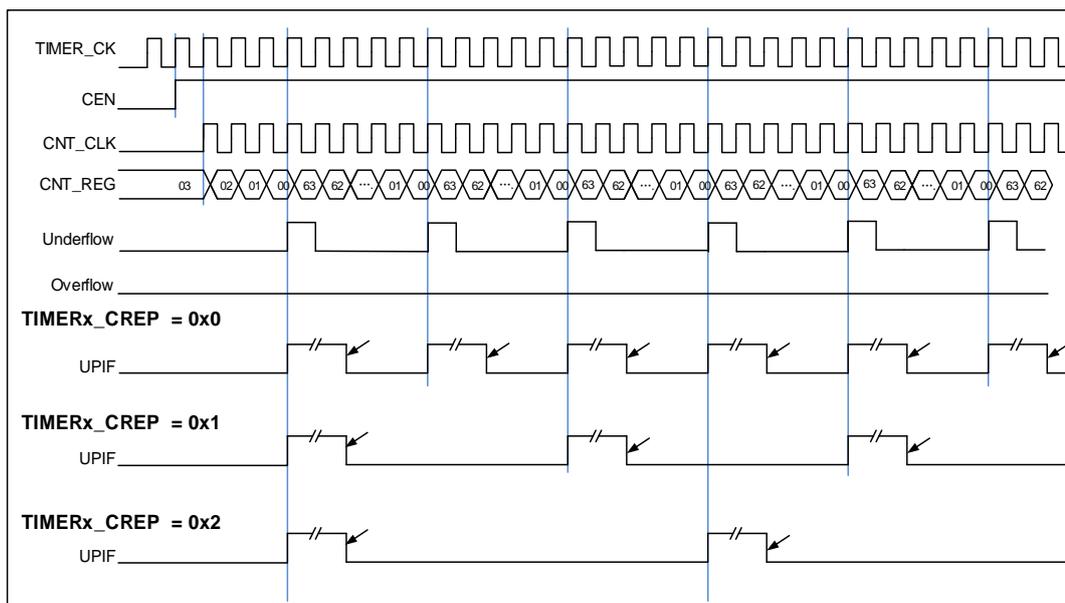


图 14-11. 在向下计数模式下计数器重复时序图



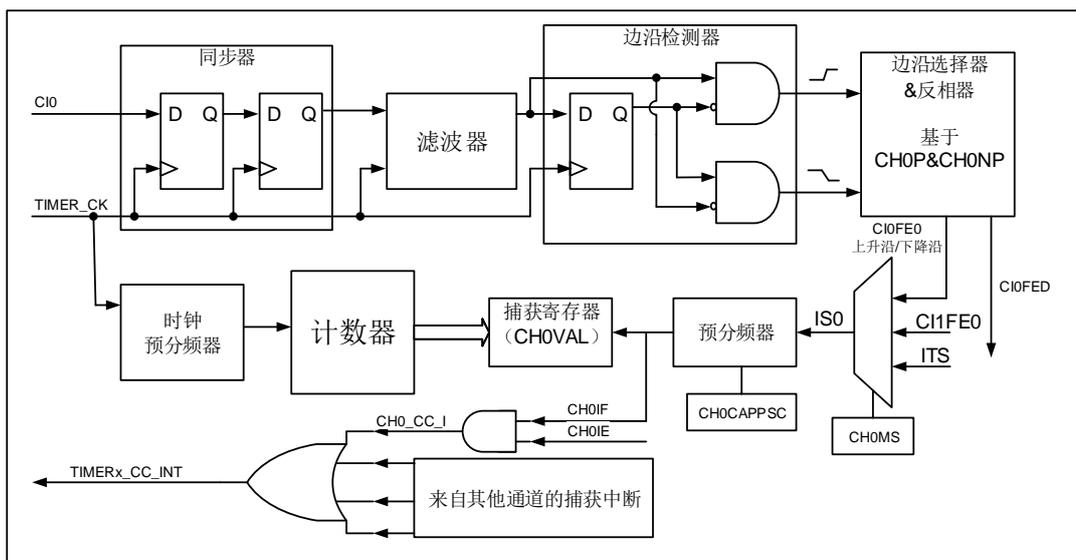
输入捕获/输出比较通道

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

通道输入捕获模式

输入捕获模式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 14-12. 输入捕获逻辑



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，判断相应的中断和DMA请求是否被提出。

直接产生： 软件设置**CHxG**位，会直接产生中断和DMA请求。

输入捕获模式也可用来测量 **TIMERx_CHx** 引脚上信号的脉冲波宽度。例如，一个 PWM 波连接到 **CI0**。配置 **TIMERx_CHCTL0** 寄存器中 **CH0MS** 为 2'b01，选择通道 0 的捕获信号为 **CI0**，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。

输出比较模式

图 14-13. 输出比较逻辑（带有互补输出的通道， $x=0, 1, 2$ ）

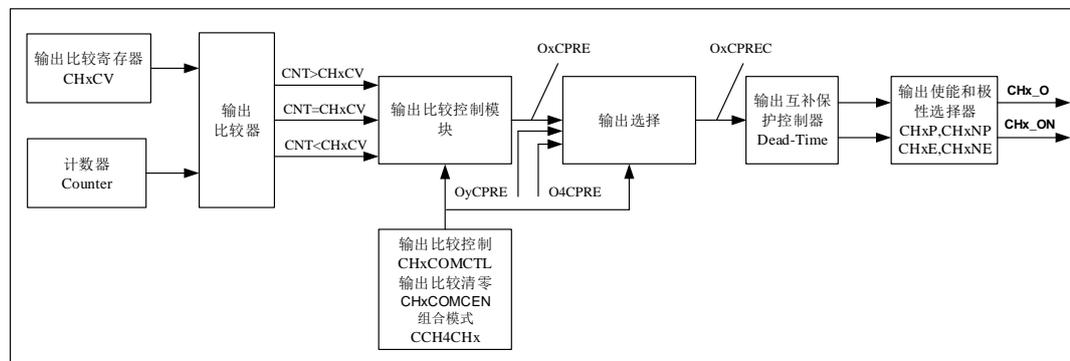


图 14-14. 输出比较逻辑（通道，x = 3）

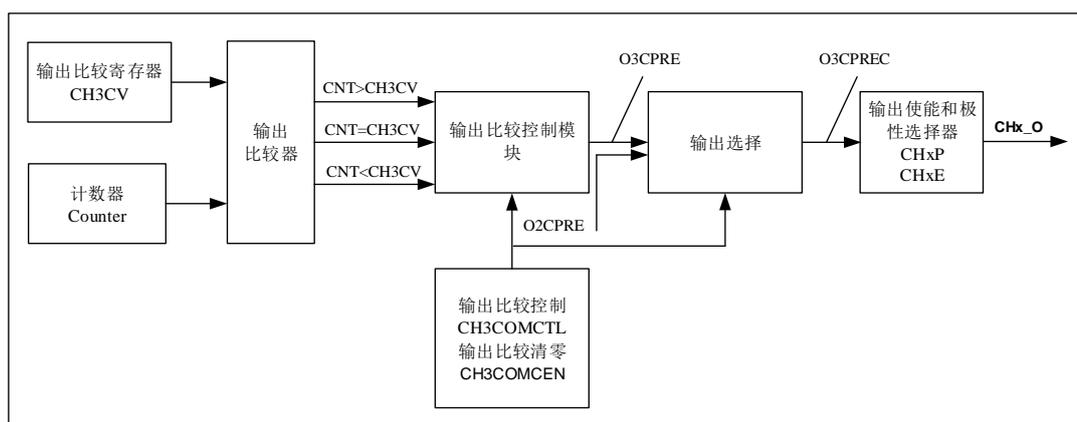
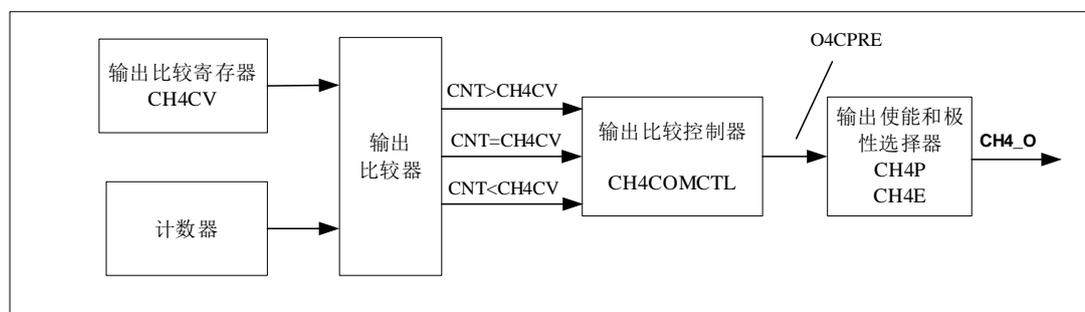


图 14-15. 输出比较逻辑（通道，x = 4）



注意：CH4_0 是内部信号，无对应引脚。

[图 14-13. 输出比较逻辑（带有互补输出的通道，x=0, 1, 2）](#)和 [图 14-14. 输出比较逻辑](#)和 [图 14-15. 输出比较逻辑（通道，x = 4）](#) 分别给出了输出比较的逻辑电路。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）。当 CHxCMOCTL[3:0] < 4'b1100 时，OxCPREC = OxCPRE；当 CHxCMOCTL[3:0] = 4'b1100 或 4'b1101 时，输出复合 PWM 波形，此时 OxCPREC 为 OxCPRE 与 OyCPRE(y = x+1 或 x-1) 的逻辑“与”或逻辑“或”；当 CHxCMOCTL[3:0] = 4'b1110 或 4'b1111 时，输出非对称 PWM 波形，此时 OxCPREC 会在向上计数时输出 OxCPRE 或 OyCPRE(y = x+1 或 x-1)或向下计数时输出 OyCPRE 或 OxCPRE(y = x+1 或 x-1)；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，被置低电平或者反转。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CxIDE=1则会产生DMA请求。

配置步骤如下：

第一步：时钟配置：

配置定时器时钟源，预分频器等。

第二步：比较模式配置：

- 设置CHxCOMSEN位来配置输出比较影子寄存器；
- 设置CHxCOMCTL位来配置输出模式（置高电平/置低电平/反转）；
- 设置CHxP/CHxNP位来选择有效电平的极性；
- 设置CHxEN使能输出。

第三步：通过CHxIE/CxIDE位配置中断/DMA请求使能。

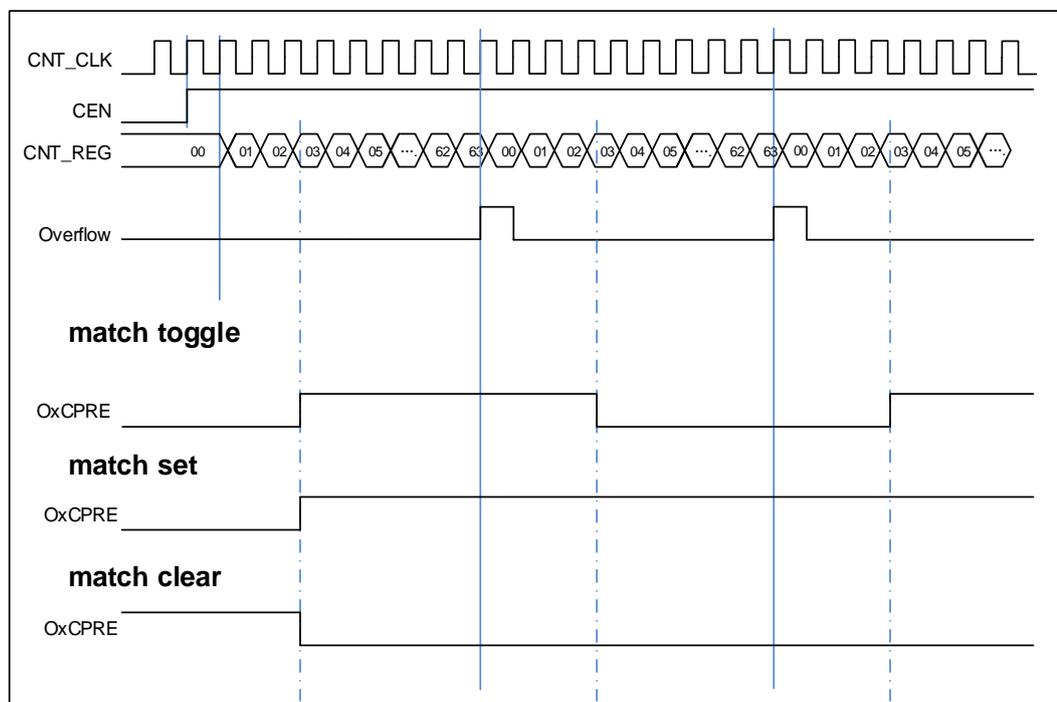
第四步：通过TIMERx_CAR寄存器和TIMERx_CHxCV寄存器配置输出比较时基：

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14-16. 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 14-16. 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 14-17. 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 14-18. 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 14-17. EAPWM 时序图

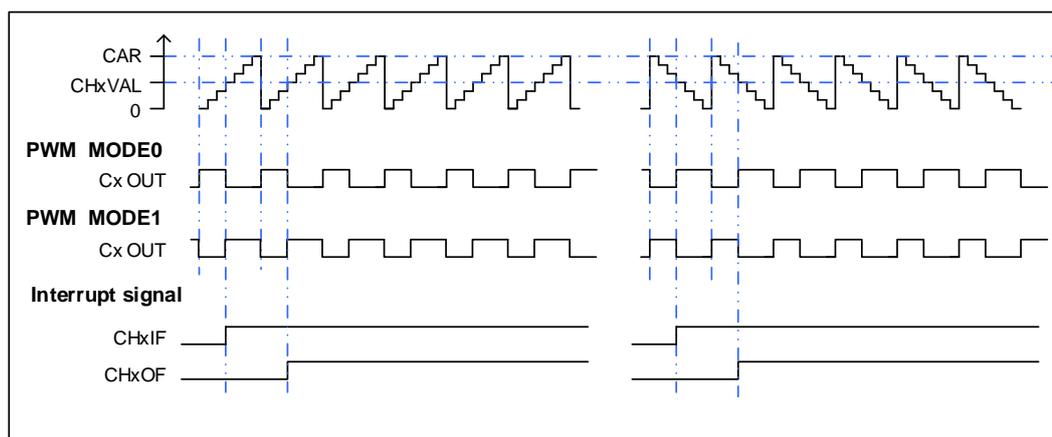
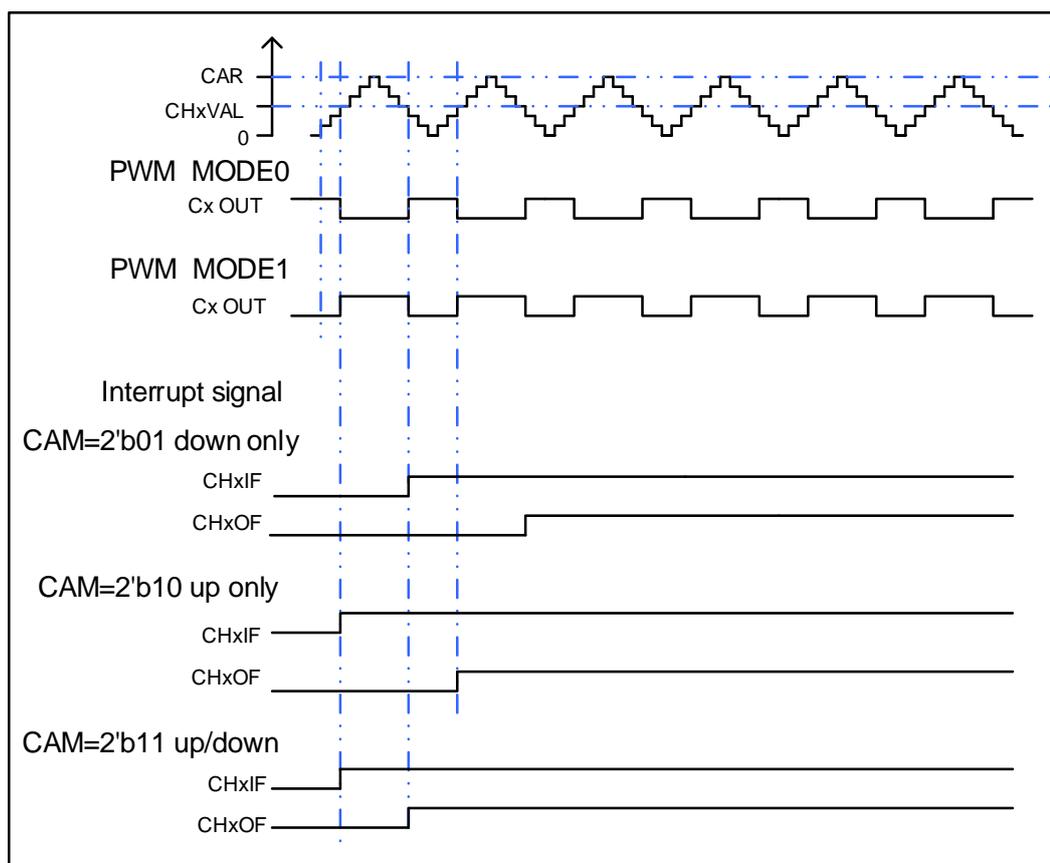


图 14-18. CAPWM 时序图



非对称 PWM 模式

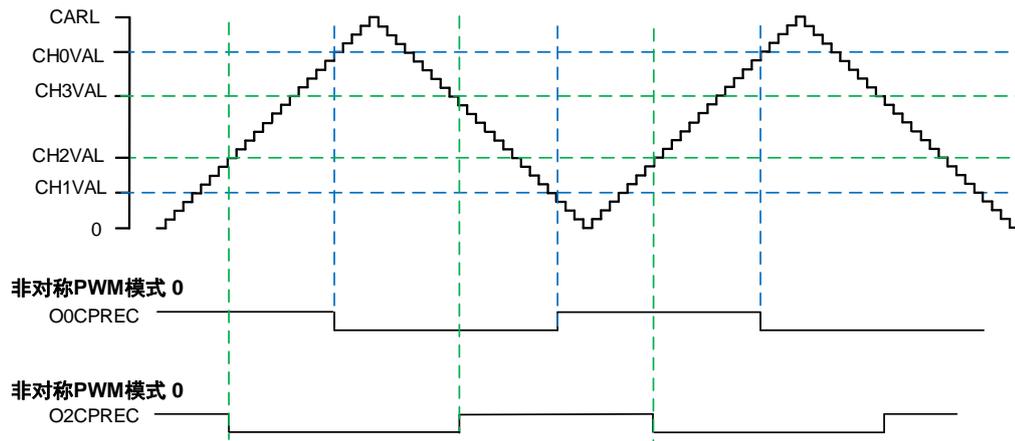
非对称PWM模式0 / 1（通过将CHxCOMCTL[3:0]位域设置为4'b1110或4'b1111）用于中央对齐PWM模式以产生可编程的相移。CPWM的频率由TIMERx_CAR寄存器决定，占空比和相移由一对具有相邻偏移地址的TIMER_CHxCV / TIMER_CH(x + 1)CV（或TIMER_CH(x - 1)CV）寄存器决定。

TIMER_CHxCV寄存器决定了向上计数的波形，TIMER_CH(x + 1)CV（或TIMER_CH(x - 1)CV）寄存器决定了向下计数的波形。详情如下：

- 通道O0CPREC / O1CPREC输出的CPWM波形由TIMER_CH0CV和TIMER_CH1CV寄存器决定；
- 通道O2CPREC / O3CPREC输出的CPWM波形由TIMER_CH2CV和TIMER_CH3CV寄存器决定。

当使用非对称PWM模式时，CH0 / CH1（或CH2 / CH3）可以独立输出不同的波形，这些波形由CHxCOMCTL[3:0]位域配置（两个通道可以配置不同的值）。

图 14-19. O0CPREC 和 O2CPREC 使用非对称 PWM 模式



复合 PWM 模式

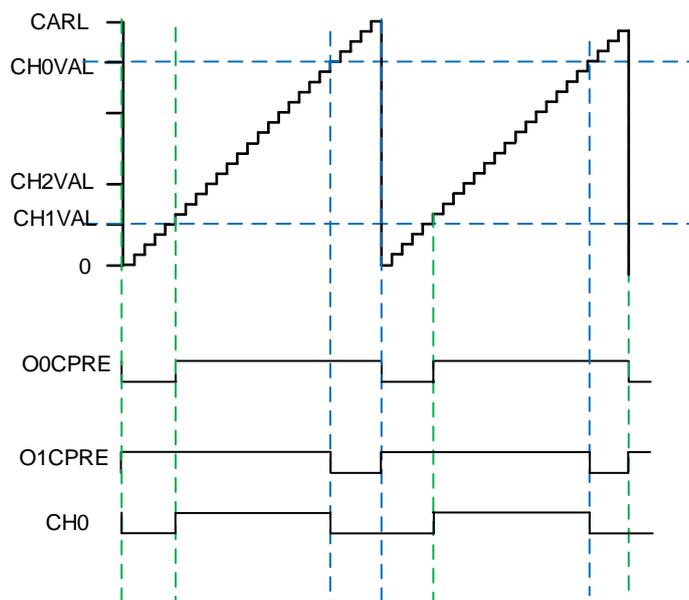
复合PWM模式0 / 1（通过将CHxCOMCTL[3:0]位域设置为4'b1100或4'b1101）用于中央对齐PWM模式以产生可编程的相移。CPWM的频率由TIMERx_CAR寄存器决定，占空比和相移由一对具有相邻偏移地址的TIMER_CHxCV / TIMER_CH(x + 1)CV（或TIMER_CH(x - 1)CV）寄存器决定。

- 通道O0CPREC / O1CPREC输出的CPWM波形由TIMER_CH0CV和TIMER_CH1CV寄存器决定；
- 通道O2CPREC / O3CPREC输出的CPWM波形由TIMER_CH2CV和TIMER_CH3CV寄存器决定。

当通道O0CPREC选择复合PWM模式0时（CH0COMCTL[3:0] == 4'b1100），此时O0CPREC输出O0CPRE与O1CPRE的逻辑“或”信号；当通道O0CPREC选择复合PWM模式1时（CH0COMCTL[3:0] == 4'b1101），此时O0CPREC输出O0CPRE与O1CPRE的逻辑“与”信号。当O0CPREC工作在复合PWM模式0时，其互补通道工作在复合PWM模式1。

当选择复合PWM模式时，通过配置CHxCOMCTL[3:0]位域不同值，每个通道CH0/CH1,或CH2/CH3可输出的不同的波形。

图 14-20. O0CPRE 选择复合 PWM 模式 1 和 O1CPRE 选择 PWM 模式 0

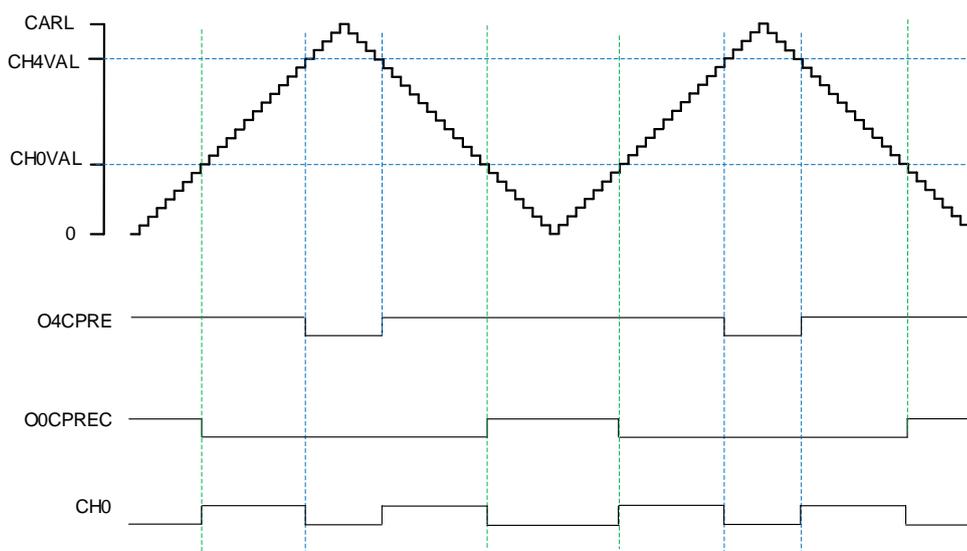


复合三相 PWM 模式

复合三相PWM模式可以产生1-3个结合O4CPRE的PWM波形，在组合波形中，O4CPRE通过与O0CPREC、O1CPREC、O2CPREC逻辑“与”输出对应的波形。

- 当 CCH4CH0 置位时，通道 O0CPREC 输出的 CPWM 波形由 TIMER_CH0CV 和 TIMER_CH4CV 寄存器决定；
- 当 CCH4CH1 置位时，通道 O1CPREC 输出的 CPWM 波形由 TIMER_CH1CV 和 TIMER_CH4CV 寄存器决定；
- 当 CCH4CH2 置位时，通道 O2CPREC 输出的 CPWM 波形由 TIMER_CH2CV 和 TIMER_CH4CV 寄存器决定；

图 14-21. CCH4CH0 置位时 O0CPREC 工作在复合三相 PWM 模式



可再次触发单脉冲模式

该模式允许计数器在响应触发信号时生成一个可编程长度的脉冲。

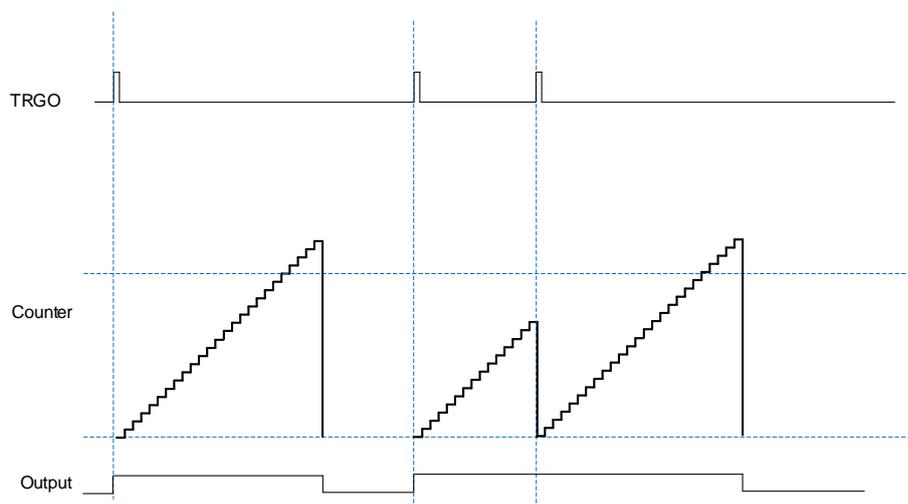
- 在接收到触发信号后，立即输出脉冲信号（无延时）。
- 如果在当前脉冲结束之前接收到新的触发信号，脉冲持续时间将被延长。

定时器需运行在主-从管理模式下，同时设置 SYSCFG_TIMERxCFG0(x = 0)中的 TSCFG7[2:0] 寄存器，通过配置 CHxCOMCTL[3:0]位于为 4'b1100 或 4'b1101 选择可再次触发单脉冲模式 0/1。

在向上计数模式下，将 TIMERx_CHyCV 设置为 0；脉冲长度由 TIMERx_CAR 寄存器定义。在向下计数模式下，确保 TIMERx_CHyCV 大于或等于 TIMERx_CAR。

注意：此模式与中心对齐的 PWM 模式不兼容。在 TIMERx_CTL0 中设置 CAM[1:0] = 2'b00。

图 14-22. 可再次触发单脉冲模式



通道输出准备信号

根据图 14-13. [输出比较逻辑（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。在下次更新事件到来时，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 有五路通道，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 14-2. 由参数控制的互补输出表

互补参数					输出状态	
POEN	ROS	IOS	CHxEN	CHxNEN	CHx_O	CHx_ON
0	0/1	0	0	0	CHx_O / CHx_ON = LOW CHx_O / CHx_ON 输出禁能 ⁽¹⁾	
				1	CHx_O/CHx_ON输出关闭状态 ⁽²⁾ ： 通道先输出无效电平：CHx_O = CHxP，CHx_ON = CHxNP；如果死区产生时钟未失效，在死区时间之后： CHx_O = ISOx，CHx_ON = ISOxN ⁽³⁾	
			1	0	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		
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 \oplus ⁽⁴⁾ CHxNP CHx_ON输出使能
			1	0	CHx_O=OxCPRE \oplus CHxP CHx_O输出使能	CHx_ON = LOW CHx_ON输出禁能
				1	CHx_O=OxCPRE \oplus CHxP CHx_O输出使能	CHx_ON=(!OxCPRE) ⁽⁵⁾ \oplus CHxNP CHx_ON输出使能
	1	0	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 \oplus CHxNP CHx_ON输出使能
		1	0	CHx_O=OxCPRE \oplus CHxP CHx_O输出使能	CHx_ON = CHxNP CHx_ON输出关闭状态	
			1	CHx_O=OxCPRE \oplus CHxP CHx_O输出使能	CHx_ON= (!OxCPRE) \oplus 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

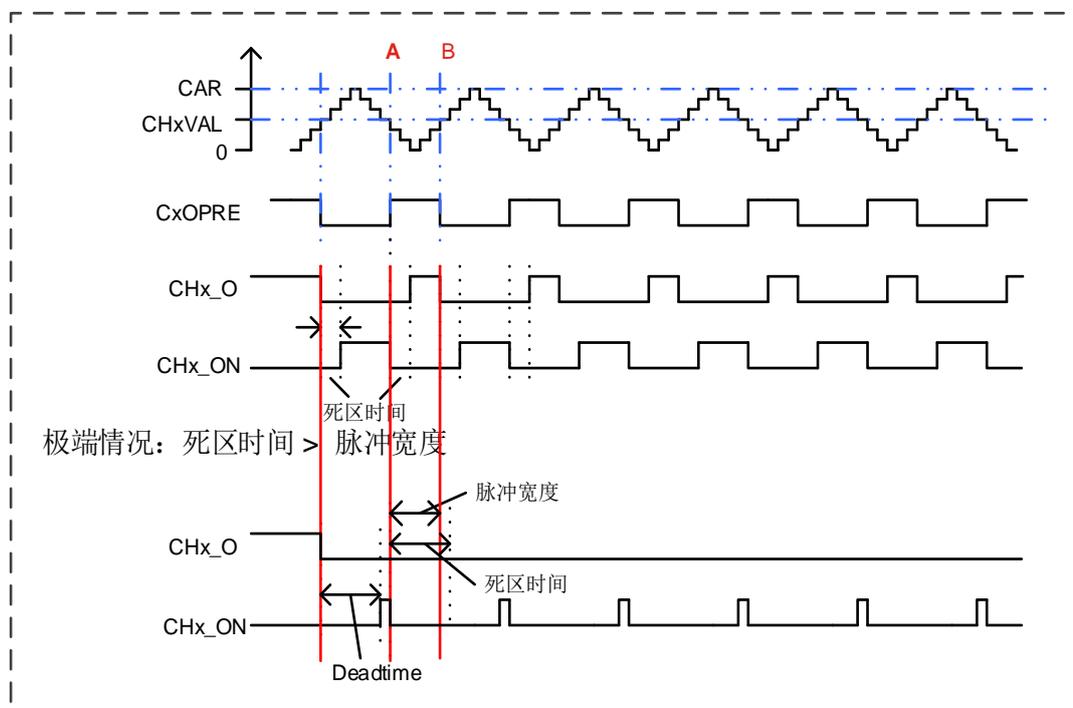
设置 CHxEN 和 CHxNEN 为 1'b1 的同时, 设置 POEN 为 1, 死区插入就会被使能。DTCFG 位域定义了死区时间, 死区时间对除了通道 3 以外的通道有效。死区时间的细节, 请参考 TIMERx_CCHP 寄存器。

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 反转。在 [图 14-23. 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 (参考 [图 14-23. 带死区时间的互补输出](#))。

图 14-23. 带死区时间的互补输出



中止功能

高级定时器有两种中止功能: BREAK0 和 BREAK1。可以通过将 TIMERx_CCHP0 寄存器中

的 BRK0EN/ BRK1EN 位置 1 来使能中止功能。中止输入极性由 TIMERx_CCHP0 寄存器中的 BRK0P/BRK1P 位配置，电平有效。

使用中止功能时，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。

中止事件是所有源逻辑或运算的结果。中止功能可以处理三种类型的事件源：

- 外部信号源：来自BRKINx (x = 0...1) 输入；
- 系统源： Cortex®-M23锁定事件、SRAM奇偶校验错误或CSS检测的时钟错误事件；
- 中止事件也可以由软件置位TIMERx_SWEVG寄存器中的BRK0G/ BRK1G位产生。

BREAK1 信号源于连接到其中一个 BRKIN 引脚的外部信号源，并通过 GPIO 的复用功能寄存器进行配置。该信号可以选择极性，并可选择性地数字滤波。此外，可以通过 TIMx_SWEVG 寄存器中的 BRK0G 和 BRK1G 位使用软件触发中断事件。使用 BRK0G 和 BRK1G 进行的软件中断时，无论 BRK0EN 和 BRK1EN 使能位为 0 或 1 都有效。

图 14-24. BREAK0 的功能逻辑图

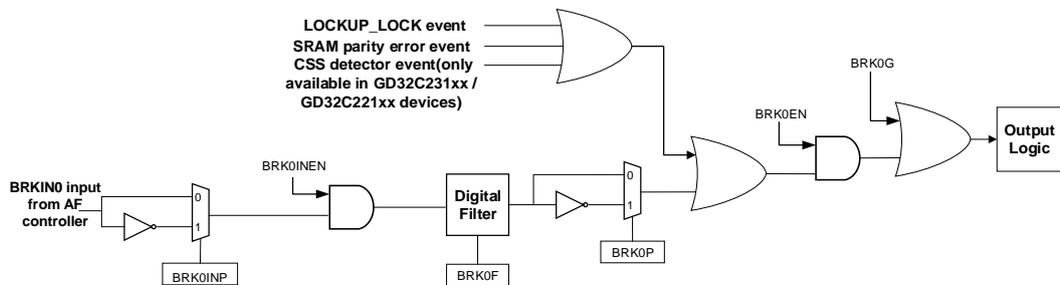
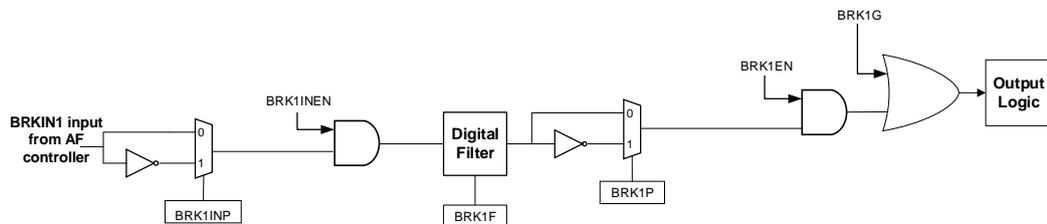


图 14-25. BREAK1 的功能逻辑图

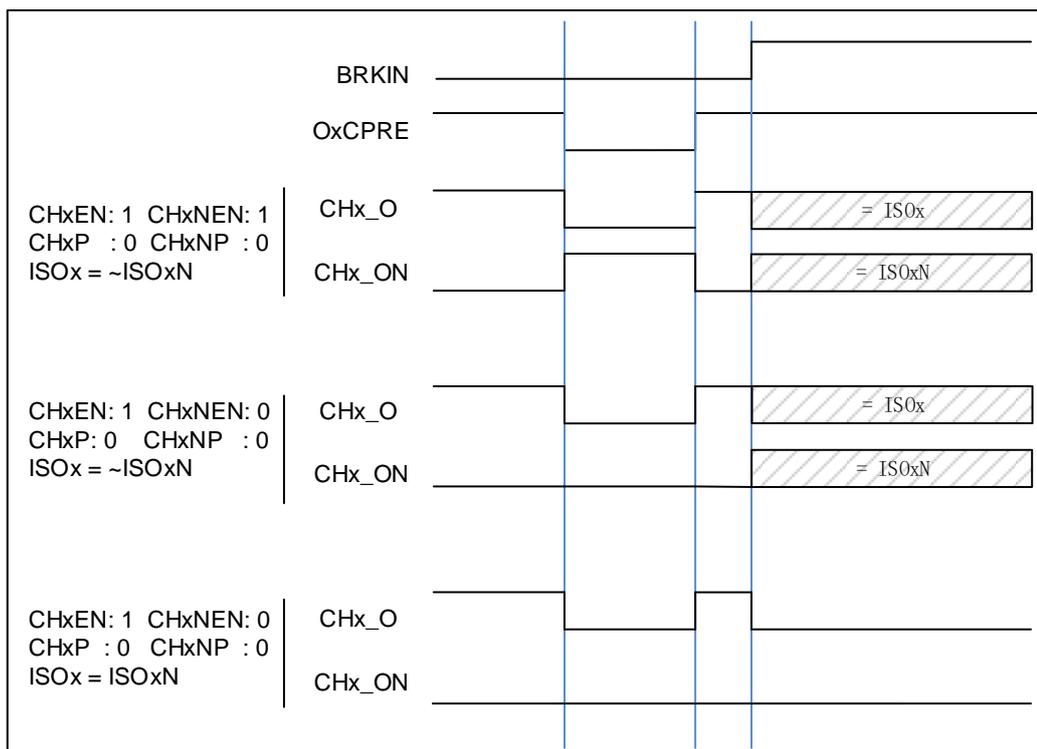


BREAK0可用于处理系统源、片上外设和外部输入信号源的故障，当发生BREAK0中止事件时，输出强制为无效电平，或在死区持续时间之后，输出将以预定的电平（有效或无效）强制输出；BREAK1只用于处理片上外设和外部输入信号源的故障，当发生BREAK1中止事件时，输出强制为无效电平。BREAK0的优先级比BREAK1高，同时BREAK1只有在ROS = IOS = 1时才有效。

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 14-26. 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 `CI0` 和 `CI1` 正交信号各自相互作用产生计数值。通过设置 `TSCFGy[2:0] != 3'b000` ($y=0,1,2$) 来选择是仅由 `CI0`，仅由 `CI1`，或者由 `CI0` 和 `CI1` 来决定定时器的计数方向。在每个方向选择源的电平改变期间，`DIR` 位改变。计数器计数方向改变的机制如表 14-3. [计数方向与正交译码器信号之间的关系](#) 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 14-3. 计数方向与正交译码器信号之间的关系

计数模式	电平	CI0FE0		CI1FE1	
		上升	下降	上升	下降
只有 CI0	CI1FE1=1	向下	向上	-	-
	CI1FE1=0	向上	向下	-	-
只有 CI1	CI0FE0=1	-	-	向上	向下
	CI0FE0=0	-	-	向下	向上
CI0 和 CI1	CI1FE1=1	向下	向上	X	X
	CI1FE1=0	向上	向下	向上	向下
	CI0FE0=1	X	X	Up	Down
	CI0FE0=0	X	X	向下	向上

注意：“-”意思是“无计数”；“X”意思是不可能。

图 14-27. 正交译码器接口模式下计数器运行例子

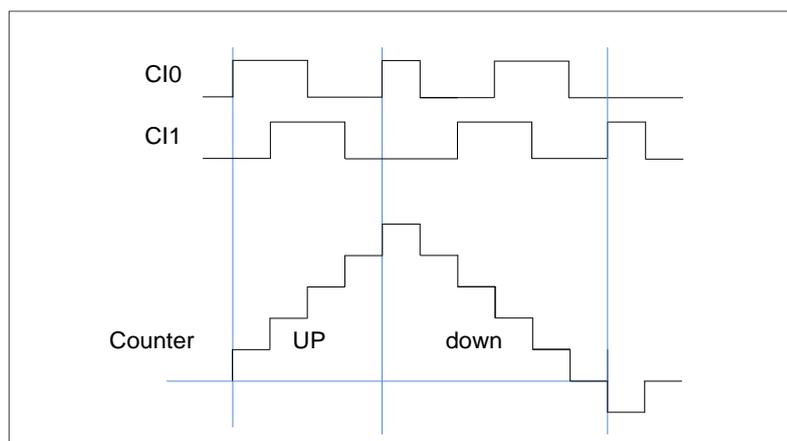
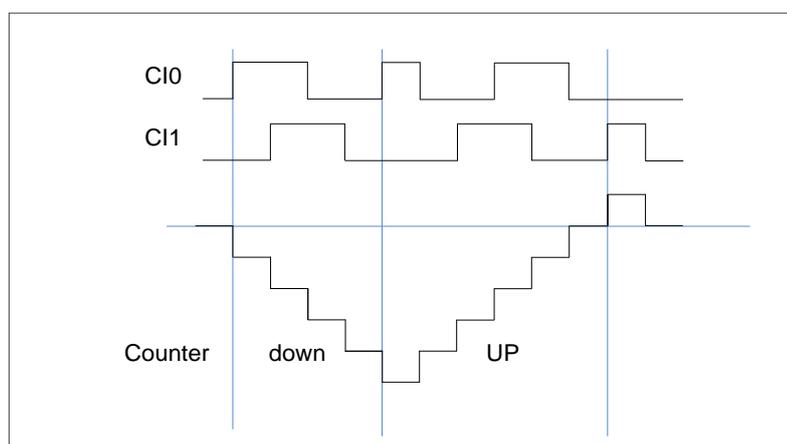


图 14-28. CI0FE0 极性反相的正交译码器接口模式下的例子



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 14-29. 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）接收来自电机霍尔传感器的三路信号，这三路信号是电机转子的位置信号。

三个霍尔传感器与 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都会反转，CHOVAL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 14-29. 霍尔传感器用在 BLDC 电机控制中

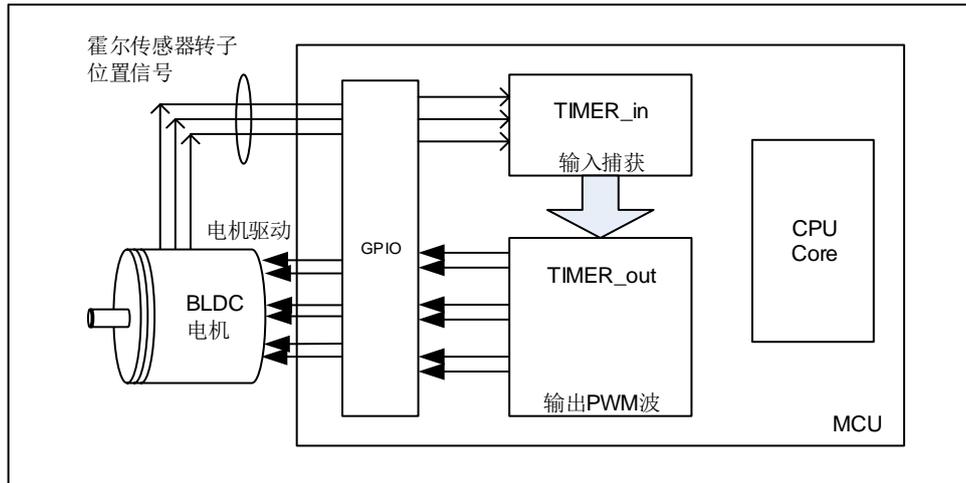
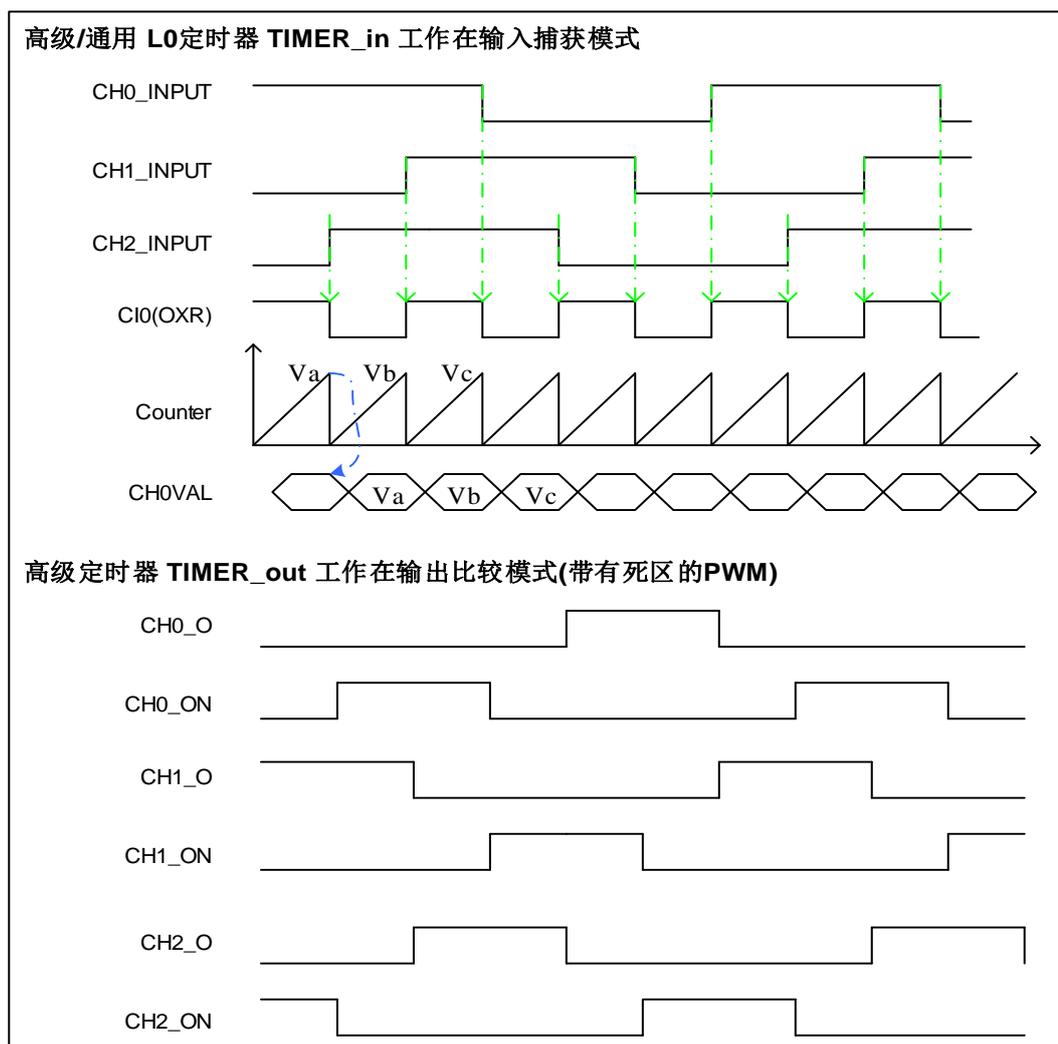


图 14-30. 两个定时器之间的霍尔传感器时序图



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 SYSCFG_TIMER0CFG (x=3,4,5) 寄存器中的 TSCFGy[2:0] 配置这些模式。

表 14-4. 从模式示例

	模式选择	触发源选择	极性选择	滤波和预分频
列举	TSCFGy[2:0] y=3 (复位模式) y=4 (暂停模式) y=5 (事件模式)	TSCFGy[2:0] 001: ITI0 010: ITI2 011: ITI3 100: CI0F_ED 101: CI0FE0 110: CI1FE1 111: ETIFP	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 ETIFP, 配置 ETP 选择极性和反相。	若触发源为 ITIx, 滤波和预分频不可用。 若触发源为 Cix, 可配置 CHxCAPFLT 设置滤波, 预分频不可用。 若触发源为 ETIFP, 滤波和预分频均可用。
例 1	复位模式 当触发输入上升沿到	TSCFG3[2:0] = 3'b 001 选择 ITI0 为触发	若触发源是 ITI0, 极性选择不可用。	若触发源是 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
	来时，计数器清零重 启。	源。		
	图 14-31. 复位模式			
例 2	暂停模式 当触发输入为低的时候，计数器暂停计数，当触发输入为高时，计数器计数。	TSCFG4[2:0] = 3'b 101 选择 CIOFE0 为触发源。	TIOS=0 (非异或) [CH0NP=0, CH0P=0] CIOFE0 不反相。捕获发生在上升沿。	在这个例子中滤波被旁路。
	图 14-32. 暂停模式			
例 3	事件模式 触发输入的上升沿计数器开始计数。	TSCFG5[2:0] = 3'b 111 选择 ETIFP 为触发源。	ETP = 0, ETI 极性不改变。	ETPSC = 1, ETI 2 分频。 ETFC = 0, ETI 无滤波。
	图 14-33. 事件模式			

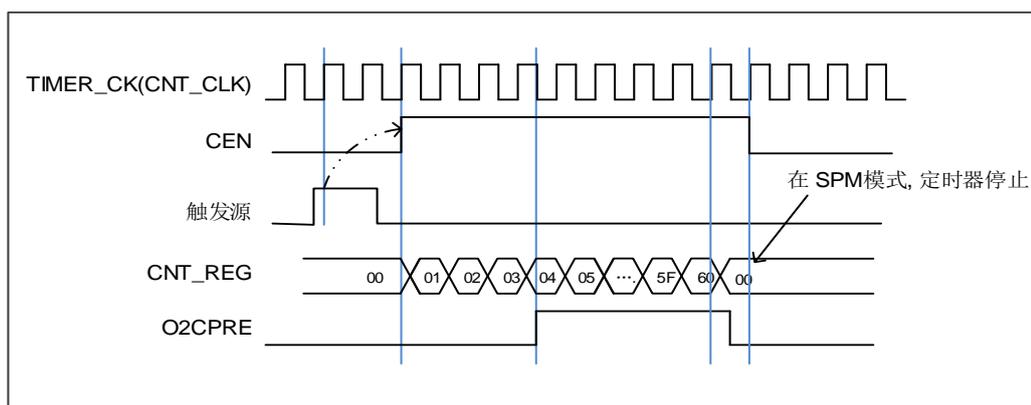
单脉冲模式

设置 `TIMERx_CTL0` 寄存器的 `SPM` 位置 1，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 PWM 模式 0 或 PWM 模式 1 时 `CHxCOMFEN` 位才可用，触发来源于触发信号。

图 14-34. 单脉冲模式，`TIMERx_CHxCV = 0x04`，`TIMERx_CAR=0x60`



定时器互连

定时器之间的相互连接可以实现定时器的级联或者同步。可以通过配置一个定时器工作在主模式，另一个定时器工作在从模式来实现。

定时器 2 作为定时器 0 的预分频器

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. 配置定时器0在外部时钟模式0，选择定时器0输入触发源为定时器2，（配置 `SYSCFG_TIMERx_CFG` 寄存器的 `TSCFG6[2:0] = 3'b 010`）；
4. 写 1 到 `CEN` 位启动定时器0（`TIMER0_CTL0`寄存器）；写 1 到 `CEN` 位启动定时器2（`TIMER2_CTL0`寄存器）。

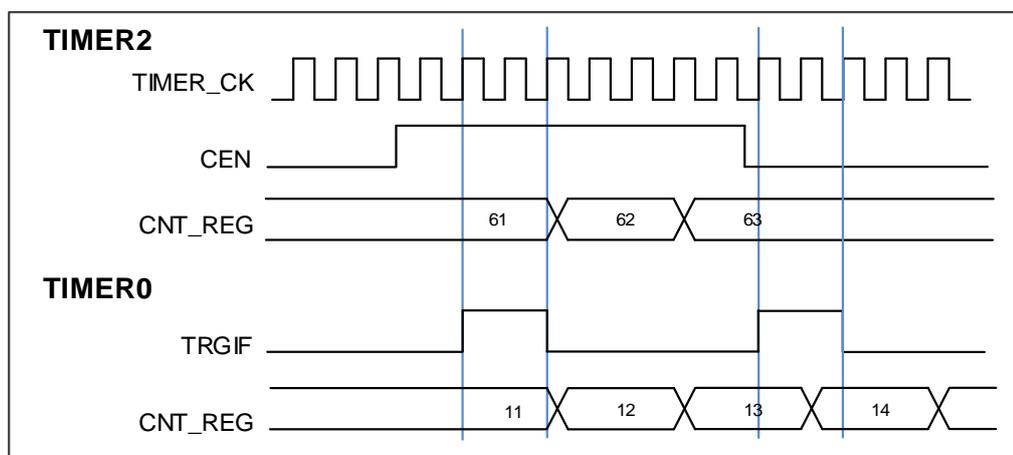
用定时器 2 的使能/更新信号来启动定时器 0

用定时器 2 的使能信号来启动定时器 0，见 [图 14-35. 用定时器 2 的使能信号触发定时器 0](#)。在定时器 2 使能信号输出后，定时器 0 按照分频后的内部时钟从当前值开始计数。

当定时器 0 接收到触发信号，它的 CEN 位被自动置 1，计数器计数直到禁用定时器 0。两个定时器的计数器频率都是 TIMER_CK 经过预分频器 3 分频后的频率 ($f_{PSC_CLK} = f_{TIMER_CK}/3$)。步骤如下：

1. 配置定时器2为主模式，发送它的使能信号作为触发输出，配置定时器0选择输入触发来自定时器2（配置SYSCFG_TIMERxCFG寄存器的TSCFG5[3:0] = 3'b 010）；
2. 写1到CEN来开启定时器2（TIMER2_CTL0寄存器）。

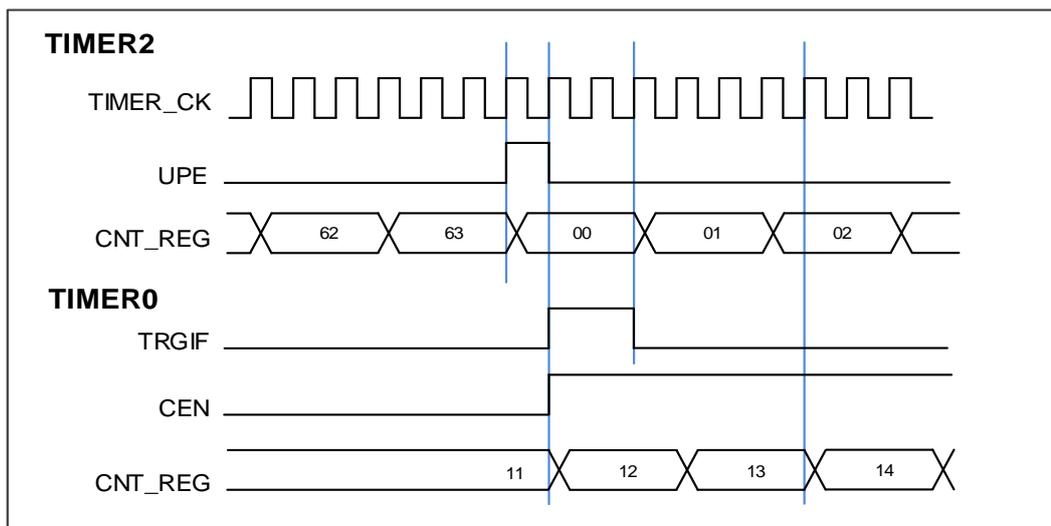
图 14-35. 用定时器 2 的使能信号触发定时器 0



在这个例子中，也可以使用更新事件代替使能信号作为触发源。见 [图 14-36. 用定时器 2 的更新事件来触发定时器 0](#)，按以下步骤进行：

1. 配置定时器2为主模式，发送它的更新事件（UPE）作为触发输出（配置TIMER2_CTL1寄存器的MMC=3'b010）；
2. 配置定时器2的周期（TIMER2_CARL寄存器）；
3. 配置定时器0选择输入触发来自定时器2，配置定时器0在事件模式（配置SYSCFG_TIMERxCFG寄存器的TSCFG5[2:0] = 3'b010）；
4. 写1到CEN来开启定时器2（TIMER2_CTL0寄存器）。

图 14-36. 用定时器 2 的更新事件来触发定时器 0

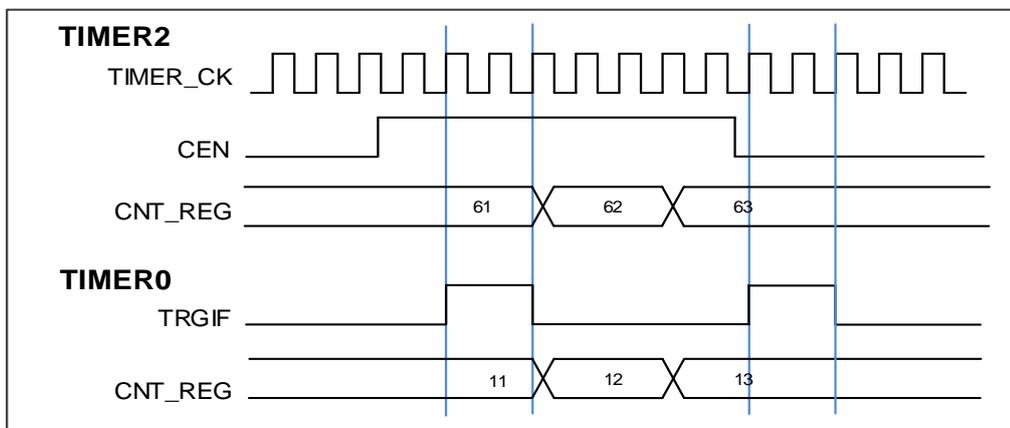


使用定时器 2 的使能/O0CPRE 信号来使能定时器 0 计数。

在这个例子中，使用定时器 2 的使能信号来使能定时器 0。如[图 14-37. 用定时器 2 的使能信号来控制定时器 0 的暂停模式](#)，在定时器 2 被使能后，定时器 0 在内部分频的时钟上开始计数。两个计数器的时钟频率都是由 TIMER_CK 时钟 3 分频得来 ($f_{PSC_CLK} = f_{TIMER_CK}/3$)，步骤如下：

1. 配置定时器2在主模式，配置其输出使能信号作为触发输出（配置TIMER2_CTL1寄存器的MMC=3'b001）；
2. 配置定时器0从定时器2获取输入触发，配置定时器0工作在暂停模式（配置SYSCFG_TIMERxCFG寄存器的TSCFG5[2:0] = 3'b 010）；
3. 写1到CEN位来使能定时器0（TIMER0_CTL0寄存器）；
4. 写1到CEN位来启动定时器2（TIMER0_CTL0寄存器）；
5. 写0到CEN位来停止定时器2（TIMER0_CTL0寄存器）。

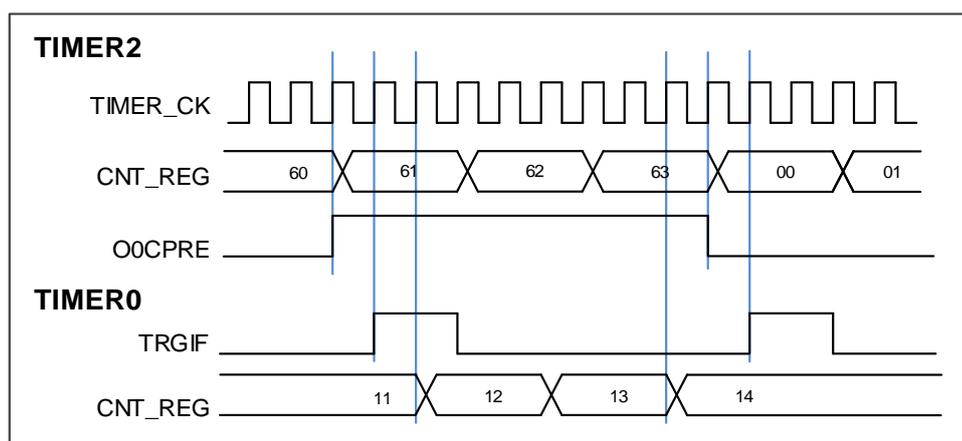
图 14-37. 用定时器 2 的使能信号来控制定时器 0 的暂停模式



这个例子中，我们也可以使用定时器 2 的 O0CPRE 信号代替其使能信号输出作为触发源。步骤如下：

1. 配置定时器2在主模式下，配置O0CPRE信号为触发输出（配置TIMER2_CTL1寄存器的MMS=3'b100）；
2. 配置定时器2的O0CPRE波形（TIMER2_CHCTL0寄存器）；
3. 配置定时器0获取来自定时器2的输入触发，配置定时器0工作在暂停模式（配置SYSCFG_TIMERxCFG寄存器TSCFG5[2:0] = 3'b 010）；
4. 写1到CEN位来使能定时器0（TIMER0_CTL0寄存器）；
5. 写1到CEN位来开启定时器2（TIMER0_CTL0寄存器）。

图 14-38. 用定时器 2 的 O0CPRE 信号控制定时器 0 的暂停模式



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。

14.1.5. TIMERx 寄存器 (x=0)

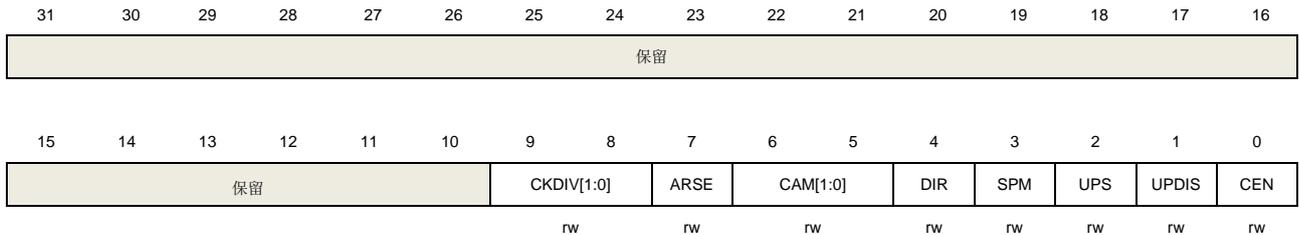
TIMER0基地址: 0x4001 2C00

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



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，通道的比较中断标志置 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，通道的比较中断标志置 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，通道的比较中断标志都会置 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模式或正交译码器模式时，该位为只读。
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: 使能后，下述任一事件产生更新中断或 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: 使能后只有计数器溢出/下溢才产生更新中断或 DMA 请求
1	UPDIS	禁止更新。 该位用来使能或禁能更新事件的发生。 0: 更新事件使能。当以下事件之一发生时，更新事件产生，具有缓存的寄存器被装入它们的预装载值： – UPG 位被置 1； – 计数器溢出/下溢； – 复位模式产生一个更新事件。 1: 更新事件禁能。带有缓存的寄存器保持原有值，如果 UPG 位被置 1 或者复位模式产生一个硬件复位事件，计数器和预分频器被重新初始化。
0	CEN	计数器使能 0: 计数器禁能 1: 计数器使能 在软件将 CEN 位置 1 后，外部时钟、暂停模式和正交译码器模式才能工作。

控制寄存器 1 (TIMERx_CTL1)

地址偏移: 0x04

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	ISO3	ISO2N	ISO2	ISO1N	ISO1	ISO0N	ISO0	TI0S	MMC[2:0]			DMAS	CCUC	保留	CCSE
	rw	rw	rw	rw	rw	rw	rw	rw		rw		rw	rw		rw

位/位域	名称	描述
31:15	保留	必须保持复位值
14	ISO3	通道 3 的空闲状态输出

		参考 ISO0 位
13	ISO2N	通道 2 的互补通道空闲状态输出 参考 ISO0N 位
12	ISO2	通道 2 的空闲状态输出 参考 ISO0 位
11	ISO1N	通道 1 的互补通道空闲状态输出 参考 ISO0N 位
10	ISO1	通道 1 的空闲状态输出 参考 ISO0 位
9	ISO0N	通道 0 的互补通道空闲状态输出 0: 当 POEN 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: 复位。TIMERx_SWEVG 寄存器的 UPG 位被置 1 或从模式控制器产生复位时, 触发一次 TRGO 脉冲, 后一种情况下, TRGO 上的信号相对实际的复位会有一个延迟。 001: 使能。此模式可用于同时启动多个定时器或控制在一段时间内使能从定时器。主模式控制器选择计数器使能信号作为触发输出 TRGO。当 CEN 控制位被置 1 或者暂停模式下触发输入为高电平时, 计数器使能信号被置 1。在暂停模式下, 计数器使能信号受控于触发输入, 在触发输入和 TRGO 上会有一个延迟, 除非选择了主/从模式。 010: 更新。主模式控制器选择更新事件作为 TRGO。 011: 捕获/比较脉冲。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: 比较。在这种模式下, 主模式控制器选择 O0CPRE 信号作为触发输出 TRGO。 101: 比较。在这种模式下, 主模式控制器选择 O1CPRE 信号作为触发输出 TRGO。 110: 比较。在这种模式下, 主模式控制器选择 O2CPRE 信号作为触发输出 TRGO。 111: 比较。在这种模式下, 主模式控制器选择 O3CPRE 信号作为触发输出 TRGO。
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	<p>换相控制影子寄存器更新控制</p> <p>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:</p> <p>0: CMTG 位被置 1 时, 更新影子寄存器</p> <p>1: 当 CMTG 位被置 1 或检测到 TRIGI 上升沿时, 影子寄存器更新</p> <p>当通道没有互补输出时, 此位无效。</p>
1	保留	必须保持复位值
0	CCSE	<p>换相控制影子使能</p> <p>0: 影子寄存器 (CHxEN, CHxNEN 和 CHxCOMCTL 位) 禁能</p> <p>1: 影子寄存器 (CHxEN, CHxNEN 和 CHxCOMCTL 位) 使能</p> <p>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。</p> <p>当通道没有互补输出时, 此位无效。</p>

从模式配置寄存器 (TIMERx_SMCFG)

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
ETP	SMC1	ETPSC[1:0]		ETFC[3:0]				MSM	保留						
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 的一部分使能外部时钟模式 1 在外部时钟模式 1, 计数器由 ETIFP 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 复位模式, 暂停模式和事件模式可以与外部时钟模式 1 同时使用。但是不能将 TRGS 设为 3'b111。 如果同时使能外部时钟模式 0 和外部时钟模式 1, 外部时钟的输入是 ETIFP。

注意：外部时钟模式 0 使能在寄存器的 SMC 位域。

13:12 ETPSC[1:0] 外部触发预分频
外部触发信号 ETIFP 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 ETIFP 的频率。
00：预分频禁能
01：ETIFP 频率被 2 分频
10：ETIFP 频率被 4 分频
11：ETIFP 频率被 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 fsAMP 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	fsAMP
4'b0000	Filter disabled.	
4'b0001	2	fTIMER_CK
4'b0010	4	
4'b0011	8	
4'b0100	6	fDTS_CK/2
4'b0101	8	
4'b0110	6	fDTS_CK/4
4'b0111	8	
4'b1000	6	fDTS_CK/8
4'b1001	8	
4'b1010	5	fDTS_CK/16
4'b1011	6	
4'b1100	8	
4'b1101	5	fDTS_CK/32
4'b1110	6	
4'b1111	8	

7 MSM 主-从模式
该位用来同步被选择的定时器同时开始计数。TRGI 用做启动事件，通过 TRGO，定时器被连接在一起。
0：主从模式禁能
1：主从模式使能

6:0 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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TRGDEN	CMTDEN	CH3DEN	CH2DEN	CH1DEN	CH0DEN	UPDEN	BRKIE	TRGIE	CMTIE	CH3IE	CH2IE	CH1IE	CH0IE	UPIE
	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

位/位域	名称	描述
31:15	保留	必须保持复位值
14	TRGDEN	触发 DMA 请求使能 0: 禁止触发 DMA 请求 1: 使能触发 DMA 请求
13	CMTDEN	换相 DMA 更新请求使能 0: 禁止换相 DMA 更新请求 1: 使能换相 DMA 更新请求
12	CH3DEN	通道 3 比较/捕获 DMA 请求使能 0: 禁止通道 3 比较/捕获 DMA 请求 1: 使能通道 3 比较/捕获 DMA 请求
11	CH2DEN	通道 2 比较/捕获 DMA 请求使能 0: 禁止通道 2 比较/捕获 DMA 请求 1: 使能通道 2 比较/捕获 DMA 请求
10	CH1DEN	通道 1 比较/捕获 DMA 请求使能 0: 禁止通道 1 比较/捕获 DMA 请求 1: 使能通道 1 比较/捕获 DMA 请求
9	CH0DEN	通道 0 比较/捕获 DMA 请求使能 0: 禁止通道 0 比较/捕获 DMA 请求 1: 使能通道 0 比较/捕获 DMA 请求
8	UPDEN	更新 DMA 请求使能 0: 禁止更新 DMA 请求 1: 使能更新 DMA 请求
7	BRKIE	中止中断使能 0: 禁止中止中断 1: 使能中止中断
6	TRGIE	触发中断使能 0: 禁止触发中断 1: 使能触发中断
5	CMTIE	换相更新中断使能 0: 禁止换相更新中断 1: 使能换相更新中断

4	CH3IE	通道 3 比较/捕获中断使能 0: 禁止通道 3 中断 1: 使能通道 3 中断
3	CH2IE	通道 2 比较/捕获中断使能 0: 禁止通道 2 中断 1: 使能通道 2 中断
2	CH1IE	通道 1 比较/捕获中断使能 0: 禁止通道 1 中断 1: 使能通道 1 中断
1	CH0IE	通道 0 比较/捕获中断使能 0: 禁止通道 0 中断 1: 使能通道 0 中断
0	UPIE	更新中断使能 0: 禁止更新中断 1: 使能更新中断

中断标志寄存器 (TIMERx_INTF)

地址偏移: 0x10

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															CH4IF
rc_w0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	SYSBIF	CH3OF	CH2OF	CH1OF	CH0OF	BRK1IF	BRK0IF	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	CH4IF	通道 4 比较/捕获中断标志 参见 CHOIF 描述
15:14	保留	必须保持复位值
13	SYSBIF	系统源中止事件中断标志位 当系统中止源有效时, 该位由硬件置1, 当系统源无效时, 该位由软件清零。 0: 无系统中止事件中断发生 1: 系统中止事件中断发生 注意: 当该位置 1 时, 在通道输出恢复前, 该位必须由软件清零。
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	BRK1IF	BREAK1 中断标志位 一旦BREAK1输入有效，由硬件对该位置‘1’。如果BREAK1输入无效，则该位可由软件清‘0’。 0: 无BREAK1事件产生 1: BREAK1 输入上检测到有效电平。当 TIMERx_DMAINTEN 寄存器中的 BRKIE=1 时，中断产生
7	BRK0IF	BREAK0 中断标志位 一旦BREAK0输入有效，由硬件对该位置‘1’。如果BREAK0输入无效，则该位可由软件清‘0’。 0: 无BREAK0事件产生 1: BREAK0 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。

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。



位/位域	名称	描述
31:9	保留	必须保持复位值
8	BRK1G	产生BREAK1事件 该位由软件置1,用于产生一个BREAK1事件,由硬件自动清0。当此位被置1时,POEN位被清0且BRK1IF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BREAK1事件 1：产生 BREAK1 事件
7	BRK0G	产生BREAK0事件 该位由软件置1,用于产生一个BREAK0事件,由硬件自动清0。当此位被置1时,POEN位被清0且BRK0IF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 BREAK0 事件 1：产生 BREAK0 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																								CH1COM CTL[3]				保留																								CH0COM CTL[3]																																																																																																																																																																																											
15															14															13															12															11															10															9															8															7															6															5															4															3															2															1															0														
CH1COM CEN		CH1COMCTL[2:0]													CH1COM SEN		CH1COM FEN		CH1MS[1:0]										CH0COM CEN		CH0COMCTL[2:0]													CH0COM SEN		CH0COM FEN		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
------	----	----

31:25	保留	必须保持复位值。
24	CH1COMCTL[3]	参见 CH1COMCTL[2:0]描述
23:17	保留	必须保持复位值。
16	CH0COMCTL[3]	参见 CH0COMCTL[2:0]描述
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, 此模式仅工作在内部触发器输入被选中时 (由 SYSCFG_TIMER0CFG) 寄存器中的 TSCFGx[2:0] (x = 3,4,5,6,7) 位域选择)。
7	CH0COMCEN	通道 0 输出比较清 0 使能 当此位被置 1, 当检测到 ETIFP 输入高电平时, O0CPRE 参考信号被清 0 0: 禁止通道 0 输出比较清零 1: 使能通道 0 输出比较清零
6:4	CH0COMCTL[2:0]	通道 0 输出比较模式 此位定义了 O0CPRE 的动作, 而 O0CPRE 决定了 CH0_O、CH0_ON 的值。O0CPRE 高电平有效, 而 CH0_O、CH0_ON 的有效电平取决于 CH0P、CH0NP 位。 0000: 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 0001: 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。 0010: 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。 0011: 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。 0100: 强制为低。强制 O0CPRE 为低电平 0101: 强制为高。强制 O0CPRE 为高电平 0110: PWM 模式 0。在向上计数时, 一旦计数器值小于 TIMERx_CH0CV 时, O0CPRE 为有效电平, 否则为无效电平。在向下计数时, 一旦计数器的值大于 TIMERx_CH0CV

时，O0CPRE 为无效电平，否则为有效电平。

0111: PWM 模式 1。在向上计数时，一旦计数器值小于 `TIMERx_CH0CV` 时，O0CPRE 为无效电平，否则为有效电平。在向下计数时，一旦计数器的值大于 `TIMERx_CH0CV` 时，O0CPRE 为有效电平，否则为无效电平。

在 PWM 模式 0 或 PWM 模式 1 中，只有当比较结果改变了或者输出比较模式中从时基模式切换到 PWM 模式时，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。

1000: 可再次触发单脉冲模式 0。O0CPRE 工作在 PWM 模式 0，向上计数时，O0CPRE 有效，当外部触发信号产生时，O0CPRE 无效，在下次更新事件产生后，O0CPRE 恢复有效。向下计数时，O0CPRE 无效，当外部触发信号产生时，O0CPRE 有效，在下次更新事件产生后，O0CPRE 恢复无效。

1001: 可再次触发单脉冲模式 1。O0CPRE 工作在 PWM 模式 1，向上计数时，O0CPRE 无效，当外部触发信号产生时，O0CPRE 有效，在下次更新事件产生后，O0CPRE 恢复无效。向下计数时，O0CPRE 有效，当外部触发信号产生时，O0CPRE 无效，在下次更新事件产生后，O0CPRE 恢复有效。

1010: 保留

1011: 保留

1100: 复合 PWM0 模式。O0CPRE 工作在 PWM 模式 0，O0CPREC 输出结果是 O0CPRE 和 O1CPRE 的逻辑“或”。

1101: 复合 PWM1 模式。O0CPRE 工作在 PWM 模式 1，O0CPREC 输出结果是 O0CPRE 和 O1CPRE 的逻辑“与”。

1110: 非对称 PWM0 模式。O0CPRE 工作在 PWM 模式 0，在向上计数时，O0CPREC 输出结果是 O0CPRE，向下计数时输出 O1CPRE。

1111 非对称 PWM1 模式。O0CPRE 工作在 PWM 模式 1，在向上计数时，O0CPREC 输出结果是 O0CPRE，向下计数时输出 O1CPRE。

当 `TIMERx_CCHP` 寄存器的 `PROT [1:0]=11` 且 `CH0MS=000`（比较模式）时，此位不能被改变。

3	CH0COMSEN	<p>通道 0 输出比较影子寄存器使能</p> <p>当此位被置 1，<code>TIMERx_CH0CV</code> 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。</p> <p>0: 禁止通道 0 输出/比较影子寄存器</p> <p>1: 使能通道 0 输出/比较影子寄存器</p> <p>仅在单脉冲模式下（<code>TIMERx_CTL0</code> 寄存器的 <code>SPM=1</code>），可以在未确认预装载寄存器情况下使用 PWM 模式</p> <p>当 <code>TIMERx_CCHP</code> 寄存器的 <code>PROT[1:0]=11</code> 且 <code>CH0MS=00</code> 时此位不能被改变。</p>
2	CH0COMFEN	<p>通道 0 输出比较快速使能</p> <p>当该位为 1 时，如果通道配置为 PWM 模式 0 或者 PWM 模式 1，会加快捕获/比较输出对触发输入事件的响应。输出通道将触发输入信号的有效沿作为一个比较匹配，<code>CH0_O</code> 被设置为比较电平而与比较结果无关。</p> <p>0: 禁能通道 0 输出比较快速功能。当触发器的输入有一个有效沿时，激活 <code>CH0_O</code> 输出的最小延时为 5 个时钟周期</p> <p>1: 使能通道 0 输出比较快速功能。当触发器的输入有一个有效沿时，激活 <code>CH0_O</code></p>

输出的最小延时为 3 个时钟周期

1:0	CH0MS[1:0]	<p>通道 0 I/O 模式选择</p> <p>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。</p> <p>00：通道 0 配置为输出。</p> <p>01：通道 0 配置为输入，IS0 映射在 CI0FE0 上。</p> <p>10：通道 0 配置为输入，IS0 映射在 CI1FE0 上。此模式仅工作在内部触发器输入被选中时（由 SYSCFG_TIMER0CFG）寄存器中的 TSCFGx[2:0] (x = 3,4,5,6,7)位域选择）。</p>
-----	------------	---

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]	<p>通道 0 输入捕获滤波控制</p> <p>数字滤波器由一个事件计数器组成，N 个输入事件后会产生一个输出的跳变。这些位定义了 CI0 输入信号的采样频率和数字滤波器的长度。</p> <p>0000：无滤波器，$f_{SAMP}=f_{DTS}$，$N=1$。</p> <p>0001：$f_{SAMP}=f_{PCLK}$，$N=2$。</p> <p>0010：$f_{SAMP}=f_{PCLK}$，$N=4$。</p> <p>0011：$f_{SAMP}=f_{PCLK}$，$N=8$。</p> <p>0100：$f_{SAMP}=f_{DTS}/2$，$N=6$。</p> <p>0101：$f_{SAMP}=f_{DTS}/2$，$N=8$。</p> <p>0110：$f_{SAMP}=f_{DTS}/4$，$N=6$。</p> <p>0111：$f_{SAMP}=f_{DTS}/4$，$N=8$。</p> <p>1000：$f_{SAMP}=f_{DTS}/8$，$N=6$。</p> <p>1001：$f_{SAMP}=f_{DTS}/8$，$N=8$。</p> <p>1010：$f_{SAMP}=f_{DTS}/16$，$N=5$。</p> <p>1011：$f_{SAMP}=f_{DTS}/16$，$N=6$。</p> <p>1100：$f_{SAMP}=f_{DTS}/16$，$N=8$。</p> <p>1101：$f_{SAMP}=f_{DTS}/32$，$N=5$。</p> <p>1110：$f_{SAMP}=f_{DTS}/32$，$N=6$。</p> <p>1111：$f_{SAMP}=f_{DTS}/32$，$N=8$。</p>
3:2	CH0CAPPSC[1:0]	<p>通道 0 输入捕获预分频器</p> <p>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。</p>

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							CH3COM CTL[3]	保留							CH2COM CTL[3]
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
31:25	保留	必须保持复位值。
24	CH3COMCTL[3]	参见 CH3COMCTL[2:0]描述
23:17	保留	必须保持复位值。
16	CH2COMCTL[3]	参见 CH2COMCTL[2:0]描述
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 CH0COMSEN 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, 此模式仅工作在内部触发器输入被选中时(由 SYSCFG_TIMER0CFG)寄存器中的 TSCFGx[2:0](x = 3,4,5,6,7)位域选择)。
7	CH2COMCEN	<p>通道 2 输出比较清 0 使能</p> <p>当此位被置 1, 当检测到 ETIF 输入高电平时, O2CPRE 参考信号被清 0</p> <p>0: 使能通道 2 输出比较清零</p> <p>1: 禁止通道 2 输出比较清零</p>
6:4	CH2COMCTL[2:0]	<p>通道 2 输出比较模式</p> <p>此位定义了 O2CPRE 的动作, 而 O2CPRE 决定了 CH2_O、CH2_ON 的值。O2CPRE 高电平有效, 而 CH2_O、CH2_ON 的有效电平取决于 CH2P、CH2NP 位。</p> <p>0000: 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</p> <p>0001: 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。</p> <p>0010: 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。</p> <p>0011: 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。</p> <p>0100: 强制为低。强制 O2CPRE 为低电平</p> <p>0101: 强制为高。强制 O2CPRE 为高电平</p> <p>0110: PWM 模式 0。在向上计数时, 一旦计数器值小于 TIMERx_CH2CV 时, O2CPRE 为有效电平, 否则为无效电平。在向下计数时, 一旦计数器的值大于 TIMERx_CH2CV 时, O2CPRE 为无效电平, 否则为有效电平。</p> <p>0111: PWM 模式 1。在向上计数时, 一旦计数器值小于 TIMERx_CH2CV 时, O2CPRE 为无效电平, 否则为有效电平。在向下计数时, 一旦计数器的值大于 TIMERx_CH2CV 时, O2CPRE 为有效电平, 否则为无效电平。</p> <p>在 PWM 模式 0 或 PWM 模式 1 中, 只有当比较结果改变了或者输出比较模式中从时基模式切换到 PWM 模式时, CxCOMR 电平才改变。</p> <p>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。</p> <p>1000:可再次触发单脉冲模式 0。 O2CPRE 工作在 PWM 模式 0, 向上计数时, O2CPRE 有效, 当外部触发信号产生时, O2CPRE 无效, 在下次更新事件产生后, O2CPRE 恢复有效。向下计数时, O2CPRE 无效, 当外部触发信号产生时, O2CPRE 有效, 在下次更新事件产生后, O2CPRE 恢复无效。</p> <p>1001: 可再次触发单脉冲模式 1。 O2CPRE 工作在 PWM 模式 1, 向上计数时, O2CPRE 无效, 当外部触发信号产生时, O2CPRE 有效, 在下次更新事件产生后, O2CPRE 恢复无效。向下计数时, O2CPRE 有效, 当外部触发信号产生时, O2CPRE 无效, 在下次更新事件产生后, O2CPRE 恢复有效。</p> <p>1010:保留</p> <p>1011:保留</p> <p>1100:复合 PWM0 模式。O2CPRE 工作在 PWM 模式 0, O2CPRE 输出结果是 O2CPRE 和 O3CPRE 的逻辑“或”。</p>

1101:复合 PWM1 模式。O2CPRE 工作在 PWM 模式 1, O2CPREC 输出结果是 O2CPRE 和 O3CPRE 的逻辑“与”。

1110: 非对称 PWM0 模式。O2CPRE 工作在 PWM 模式 0, 在向上计数时, O2CPREC 输出结果是 O2CPRE, 向下计数时输出 O3CPRE。

1111 非对称 PWM1 模式。O2CPRE 工作在 PWM 模式 1, 在向上计数时, O2CPREC 输出结果是 O2CPRE, 向下计数时输出 O3CPRE。

当 TIMERx_CCHP 寄存器的 PROT[1:0]=11 且 CH2MS =000 (比较模式) 时, 此位不能被改变。

- 3 CH2COMSEN 通道 2 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 (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
 当 TIMERx_CCHP 寄存器的 PROT[1: 0]=11 且 CH2MS =00 时此位不能被改变。
- 2 CH2COMFEN 通道 2 输出比较快速使能
 当该位为 1 时, 如果通道配置为 PWM 模式 0 或者 PWM 模式 1, 会加快捕获/比较输出对触发输入事件的响应。输出通道将触发输入信号的有效沿作为一个比较匹配, CH2_O 被设置为比较电平而与比较结果无关。
 0: 禁能通道 2 输出比较快速功能。当触发器的输入有一个有效沿时, 激活 CH2_O 输出的最小延时为 5 个时钟周期
 1: 使能通道 2 输出比较快速功能。当触发器的输入有一个有效沿时, 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, 此模式仅工作在内部触发器输入被选中时 (由 SYSCFG_TIMER0CFG) 寄存器中的 TSCFGx[2:0] (x = 3,4,5,6,7) 位域选择)。

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
 数字滤波器由一个事件计数器组成，N 个输入事件后会产生一个输出的跳变。这些位定义了 CI2 输入信号的采样频率和数字滤波器的长度。
 0000: 无滤波器， $f_{SAMP}=f_{DTS}$ ， $N=1$ 。
 0001: $f_{SAMP}=f_{PCLK}$ ， $N=2$ 。
 0010: $f_{SAMP}=f_{PCLK}$ ， $N=4$ 。
 0011: $f_{SAMP}=f_{PCLK}$ ， $N=8$ 。
 0100: $f_{SAMP}=f_{DTS}/2$ ， $N=6$ 。
 0101: $f_{SAMP}=f_{DTS}/2$ ， $N=8$ 。
 0110: $f_{SAMP}=f_{DTS}/4$ ， $N=6$ 。
 0111: $f_{SAMP}=f_{DTS}/4$ ， $N=8$ 。
 1000: $f_{SAMP}=f_{DTS}/8$ ， $N=6$ 。
 1001: $f_{SAMP}=f_{DTS}/8$ ， $N=8$ 。
 1010: $f_{SAMP}=f_{DTS}/16$ ， $N=5$ 。
 1011: $f_{SAMP}=f_{DTS}/16$ ， $N=6$ 。
 1100: $f_{SAMP}=f_{DTS}/16$ ， $N=8$ 。
 1101: $f_{SAMP}=f_{DTS}/32$ ， $N=5$ 。
 1110: $f_{SAMP}=f_{DTS}/32$ ， $N=6$ 。
 1111: $f_{SAMP}=f_{DTS}/32$ ， $N=8$ 。
- 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														CH4P	CH4EN
														r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3NP	保留	CH3P	CH3EN	CH2NP	CH2NEN	CH2P	CH2EN	CH1NP	CH1NEN	CH1P	CH1EN	CH0NP	CH0NEN	CH0P	CH0EN
r/w		r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

位/位域	名称	描述
31:18	保留	必须保持复位值
17	CH4P	通道 4 极性 参考 CH0P 描述
16	CH4EN	通道 4 使能 参考 CH0EN 描述
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	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 高电平有效 1: 通道 0 低电平有效 当通道 0 配置为输入模式时，此位和 CH0P 联合使用，作为输入信号 CI0 的极性选择控制信号。 当 TIMERx_CCHP 寄存器的 PROT[1: 0]=11 或 10 时此位不能被更改。

2	CH0NEN	通道 0 互补输出使能 当通道 0 配置为输出模式时，将此位置 1 使能通道 0 的互补输出。 0: 禁止通道 0 互补输出 1: 使能通道 0 互补输出
1	CH0P	通道 0 极性 当通道 0 配置为输出模式时，此位定义了输出信号极性。 0: 通道 0 高电平有效 1: 通道 0 低电平有效 当通道 0 配置为输入模式时，此位定义了 CI0 信号极性。 CH0P 将选择 CI0FE0 或者 CI1FE0 的有效边沿或者捕获极性。 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。 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。 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



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



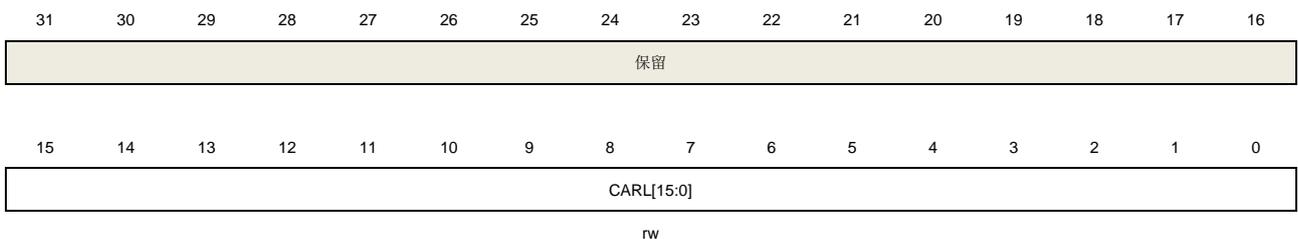
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	PSC[15:0]	计数器时钟预分频值 计数器时钟等于 PSC 时钟除以 (PSC+1)，每次当更新事件产生时，PSC 的值被装入当前预分频寄存器。

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
15:0	CREP[15:0]	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



位/位域	名称	描述
31:16	保留	必须保持复位值。

15:0	CH1VAL[15:0]	<p>通道 1 的捕获或比较值</p> <p>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。</p> <p>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。</p>
------	--------------	---

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



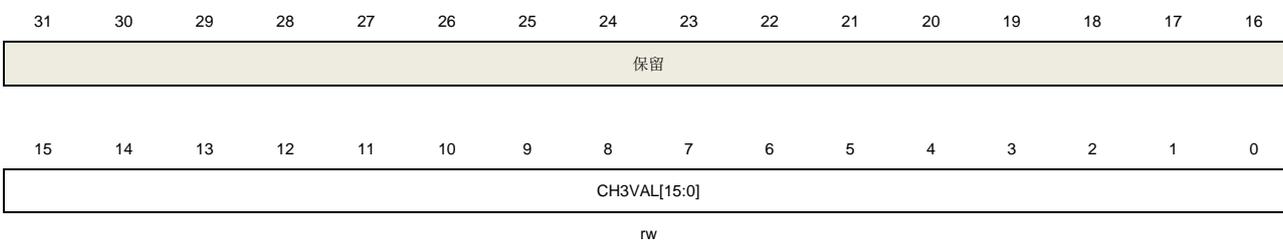
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CH2VAL[15:0]	<p>通道 2 的捕获或比较值</p> <p>当通道 2 配置为输入模式时，这些位决定了上次捕获事件的计数器值，并且本寄存器为只读。</p> <p>当通道 2 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。</p>

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



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						BRK1P	BRK1EN	BRK1F[3:0]				BRK0F[3:0]			
						r/w	r/w	r/w				r/w			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POEN	OAEN	BRK0P	BRK0EN	ROS	IOS	PROT[1:0]		DTCFG[7:0]							
r/w	r/w	r/w	r/w	r/w	r/w	r/w		r/w							

位/位域	名称	描述
31:26	保留	必须保持复位值。
25	BRK1P	<p>BREAK1输入信号极性</p> <p>该位用于配置BREAK1输入信号的极性</p> <p>0: BREAK1输入信号低电平有效</p> <p>1: BREAK1输入信号高电平有效</p> <p>此位只有在TIMERx_CCHP寄存器的PROT [1:0] =00时才可修改。</p> <p>注意: 对该位的每一次写操作, 需要延时 1 个 APB 时钟才有效。</p>
24	BRK1EN	<p>BREAK1输入信号使能</p> <p>该位置1时, 使能BREAK1输入信号。</p> <p>0: BREAK1输入禁能</p> <p>1: BREAK1输入使能</p> <p>此位只有在TIMERx_CCHP寄存器的PROT [1:0] =00时才可修改。</p> <p>注意:</p> <ol style="list-style-type: none"> 1) 对该位的每一次写操作, 需要延时1个APB时钟才有效。 2) 该位仅用于ROS=1且IOS=1时
23:20	BRK1F[3:0]	<p>BREAK1输入信号滤波</p> <p>数字滤波器由一个事件计数器组成, 它记录N个输入事件后会产生一个输出的跳变。这些位定义了BREAK1输入信号的采样频率和数字滤波器的长度。</p> <p>0000: 无滤波器, BREAK1异步有效, N=1</p> <p>0001: fSAMP = fCK_TIMER, N=2</p> <p>0010: fSAMP = fCK_TIMER, N=4</p> <p>0011: fSAMP = fCK_TIMER, N=8</p> <p>0100: fSAMP = fDTS/2, N=6</p> <p>0101: fSAMP = fDTS/2, N=8</p>

		0110: fSAMP = fDTS/4, N=6
		0111: fSAMP = fDTS/4, N=8
		1000: fSAMP = fDTS/8, N=6
		1001: fSAMP = fDTS/8, N=8
		1010: fSAMP = fDTS/16, N=5
		1011: fSAMP = fDTS/16, N=6
		1100: fSAMP = fDTS/16, N=8
		1101: fSAMP = fDTS/32, N=5
		1110: fSAMP = fDTS/32, N=6
		1111: fSAMP = fDTS/32, N=8
		此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0] =00 时才可修改。
19:16	BRK0F[3:0]	<p>BREAK0输入信号滤波</p> <p>数字滤波器由一个事件计数器组成，它记录N个输入事件后会产生一个输出的跳变。这些位定义了BREAK0输入信号的采样频率和数字滤波器的长度。</p> <p>0000: 无滤波器，BREAK0异步有效，N=1</p> <p>0001: fSAMP = fCK_TIMER, N=2</p> <p>0010: fSAMP = fCK_TIMER, N=4</p> <p>0011: fSAMP = fCK_TIMER, N=8</p> <p>0100: fSAMP = fDTS/2, N=6</p> <p>0101: fSAMP = fDTS/2, N=8</p> <p>0110: fSAMP = fDTS/4, N=6</p> <p>0111: fSAMP = fDTS/4, N=8</p> <p>1000: fSAMP = fDTS/8, N=6</p> <p>1001: fSAMP = fDTS/8, N=8</p> <p>1010: fSAMP = fDTS/16, N=5</p> <p>1011: fSAMP = fDTS/16, N=6</p> <p>1100: fSAMP = fDTS/16, N=8</p> <p>1101: fSAMP = fDTS/32, N=5</p> <p>1110: fSAMP = fDTS/32, N=6</p> <p>1111: fSAMP = fDTS/32, N=8</p> <p>此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0] =00 时才可修改。</p>
15	POEN	<p>所有的通道输出使能</p> <p>该位通过以下方式置 1:</p> <ul style="list-style-type: none"> -写 1 置位 -如果 OAEN=1，则在下一次更新事件发生时置 1。 <p>该位通过以下方式清 0:</p> <ul style="list-style-type: none"> -写 0 清 0 -有效的中止输入（异步） <p>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。</p> <p>0: 禁止通道输出</p> <p>1: 使能通道输出</p> <p>注意：仅当 CHxMS[1:0]=2'b00 时该位有效。</p>
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	BRK0P	BREAK0 输入信号极性 该位用于配置 BREAK0 输入信号的极性 0: BREAK0 输入信号低电平有效 1: BREAK0 输入信号高电平有效 此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0] =00 时才可修改。 注意: 对该位的每一次写操作, 需要延时 1 个 APB 时钟才有效。
12	BRK0EN	BREAK0 输入信号使能 该位置0时, 使能 BREAK0 输入信号。 0: BREAK0 输入禁能 1: BREAK0 输入使能 此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0] =00 时才可修改。 注意: 1) 对该位的每一次写操作, 需要延时1个APB时钟才有效。 2) 该位仅用于 ROS=1 且 IOS=1 时
11	ROS	运行模式下“关闭状态”使能 当 POEN 位被置 1 (运行模式), 此位可以被置 1 来使能通道(带有互补输出且配置为输出模式)的输出“关闭状态”。 0: 输出“关闭状态”禁能。当 CHxEN 或者 CHxNEN 位被清零, 对应通道为输出“禁能状态”。 1: 输出“关闭状态”使能。当 CHxEN 或者 CHxNEN 位被清零, 对应通道为输出“关闭状态”。 此位在 TIMERx_CCHP 寄存器的 PROT [1:0]=10 或 11 时不能被更改。
10	IOS	空闲模式下“关闭状态”使能 当 POEN 位被清 0 (空闲模式), 此位可以被置 1 来使能通道(带有互补输出且配置为输出模式)的输出“关闭状态”。 0: 输出“关闭状态”禁能。当 CHxEN 和 CHxNEN 位均被清零, 对应通道为输出“禁能状态”。 1: 输出“关闭状态”使能。不论 CHxEN 和 CHxNEN 位的值, 对应通道为输出“关闭状态”。 此位在 TIMERx_CCHP 寄存器的 PROT [1:0]=10 或 11 时不能被更改。
9:8	PROT[1:0]	互补寄存器保护控制 这两位定义了寄存器的写保护特性。 00: 禁能保护模式, 无写保护 01: PROT 模式 0。 TIMERx_CTL1 寄存器中 ISOx/ISOxN 位, TIMERx_CCHP 寄存器中 BRKEN/BRKP/OAEN/DTCFG 位写保护 10: PROT 模式 1。除了 PROT 模式 0 下的寄存器写保护外, 还有 TIMERx_CHCTL2 寄存器中 CHxP/CHxNP 位 (如果相应通道配置为输出模式), TIMERx_CCHP 寄存器中 ROS/IOS 位。 11: PROT 模式 2。除了 PROT 模式 1 下的寄存器写保护外, 还有

TIMERx_CHCTLR0/1 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]

死区时间配置

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：

$$DTCFG[7:5] = 3'b0xx:DT\ value = DTCFG[7:0] * t_{DT}, t_{DT} = t_{DTS}$$

$$DTCFG[7:5] = 3'b10x:DT\ value = (64 + DTCFG[5:0]) * t_{DT}, t_{DT} = t_{DTS} * 2$$

$$DTCFG[7:5] = 3'b110:DT\ value = (32 + DTCFG[4:0]) * t_{DT}, t_{DT} = t_{DTS} * 8$$

$$DTCFG[7:5] = 3'b111:DT\ value = (32 + DTCFG[4:0]) * t_{DT}, t_{DT} = t_{DTS} * 16$$

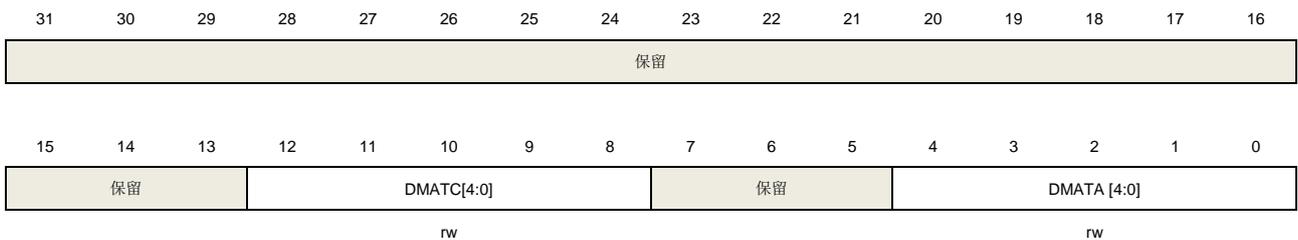
此位只有在 TIMERx_CCHP 寄存器的 PROT[1:0]=00 时才可修改。

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 寄存器的次数。 5'b00000: 1 次传输 5'b00001: 2 次传输 ... 5'b10001: 18 次传输
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)。 5'b00000: TIMERx_CTL0 5'b00001: TIMERx_CTL1 ...

5'b10010: TIMERx_DMACHCFG

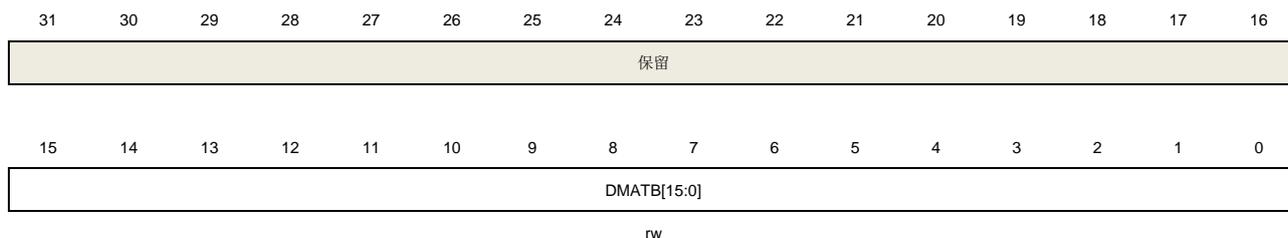
总之: 起始地址 = TIMERx_CTL0 + DMATA*4

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



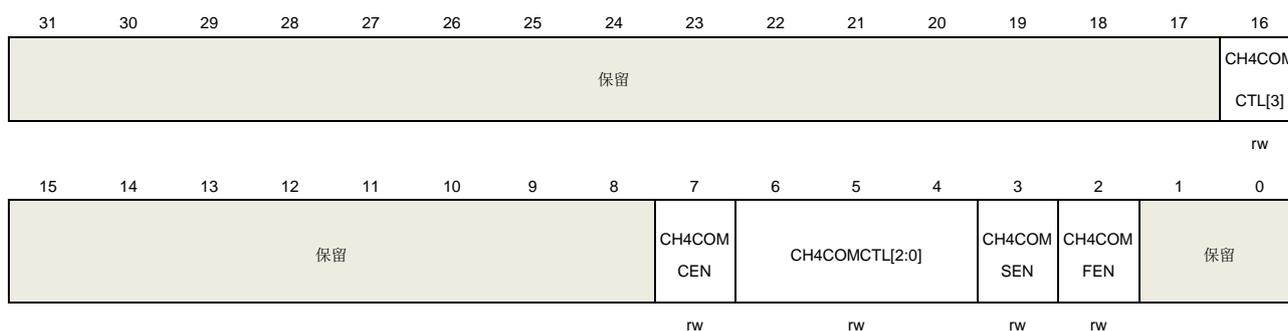
位/位域	名称	描述
31:16	保留	必须保持复位值.
15:0	DMATB[15:0]	DMA 发送缓冲 对这个寄存器的读或写, 从 (起始地址) 到 (起始地址+传输次数*4) 地址范围内的寄存器会被访问。传输次数由硬件计算, 范围为 0 到 DMATC。

通道控制寄存器 1 (TIMERx_CHCTL3)

地址偏移: 0x54

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
31:17	保留	必须保持复位值.
16	CH4COMCTL[3]	通道 4 输出比较模式 参见 CH4COMCTL[2:0]描述

25:8	保留	必须保持复位值。
7	CH4COMCEN	通道 4 输出比较清 0 使能 参见 CH0COMCEN 描述
6:4	CH4COMCTL[2:0]	<p>通道 2 输出比较模式</p> <p>此位定义了 O4CPRE 的动作，而 O4CPRE 决定了 CH4_O 的值。O4CPRE 高电平有效，而 CH4_O 的有效电平取决于 CH4P 位。</p> <p>0000: 时基。输出比较寄存器 TIMERx_CH4CV 与计数器 TIMERx_CNT 间的比较对 O4CPRE 不起作用</p> <p>0001: 匹配时设置为高。当计数器的值与捕获/比较值寄存器 TIMERx_CH4CV 相同时，强制 O4CPRE 为高。</p> <p>0010: 匹配时设置为低。当计数器的值与捕获/比较值寄存器 TIMERx_CH4CV 相同时，强制 O4CPRE 为低。</p> <p>0011: 匹配时翻转。当计数器的值与捕获/比较值寄存器 TIMERx_CH4CV 相同时，强制 O4CPRE 翻转。</p> <p>0100: 强制为低。强制 O4CPRE 为低电平</p> <p>0101: 强制为高。强制 O4CPRE 为高电平</p> <p>0110: PWM 模式 0。在向上计数时，一旦计数器值小于 TIMERx_CH4CV 时，O4CPRE 为有效电平，否则为无效电平。在向下计数时，一旦计数器的值大于 TIMERx_CH4CV 时，O4CPRE 为无效电平，否则为有效电平。</p> <p>0111: PWM 模式 1。在向上计数时，一旦计数器值小于 TIMERx_CH4CV 时，O4CPRE 为无效电平，否则为有效电平。在向下计数时，一旦计数器的值大于 TIMERx_CH4CV 时，O4CPRE 为有效电平，否则为无效电平。</p> <p>在 PWM 模式 0 或 PWM 模式 1 中，只有当比较结果改变了或者输出比较模式中从时基模式切换到 PWM 模式时，CxCOMR 电平才改变。</p> <p>1000:可再次触发模式 0。 O4CPRE 工作在 PWM 模式 0，向上计数时，O4CPRE 有效，当外部触发信号产生时，O4CPRE 无效，在下次更新事件产生后，O4CPRE 恢复有效。向下计数时，O4CPRE 无效，当外部触发信号产生时，O4CPRE 有效，在下次更新事件产生后，O4CPRE 恢复无效。</p> <p>1001: 可再次触发模式 1。 O4CPRE 工作在 PWM 模式 1，向上计数时，O4CPRE 无效，当外部触发信号产生时，O4CPRE 有效，在下次更新事件产生后，O4CPRE 恢复无效。向下计数时，O4CPRE 有效，当外部触发信号产生时，O4CPRE 无效，在下次更新事件产生后，O4CPRE 恢复有效。</p> <p>其他:保留</p> <p>当 TIMERx_CCHP 寄存器的 PROT [1:0]=11，此位不能被改变。</p>
3	CH4COMSEN	<p>通道 4 出比较影子寄存器使能</p> <p>当此位被置 1，TIMERx_CH4CV 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。</p> <p>0: 禁止通道 4 输出/比较影子寄存器</p> <p>1: 使能通道 4 输出/比较影子寄存器</p> <p>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</p> <p>当 TIMERx_CCHP 寄存器的 PROT[1: 0]=11。</p>

2	CH4COMFEN	<p>通道 4 输出比较快速使能</p> <p>当该位为 1 时，如果通道配置为 PWM 模式 0 或者 PWM 模式 1，会加快捕获/比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，CH4_O 被设置为比较电平而与比较结果无关。</p> <p>0：禁能通道 2 输出比较快速功能。当触发器的输入有一个有效沿时，激活 CH4_O 输出的最小延时为 5 个时钟周期</p> <p>1：使能通道 2 输出比较快速功能。当触发器的输入有一个有效沿时，激活 CH4_O 输出的最小延时为 3 个时钟周期</p>
1:0	保留	必须保持复位值。

通道 4 捕获/比较寄存器 (TIMERx_CH4CV)

地址偏移：0x58

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	CCH4CH2	组合通道 4 和通道 2 0：O4CPRE 和 O2CPREF 独立输出 1：O2CPREF 输出为 O2CPREF 和 O4CPRE 的逻辑“与” 该位域可立即生效或者在下一次更新事件发生时生效（当 TIMERx_CHCTL1 中的比较影子寄存器使能）。
30	CCH4CH1	组合通道 4 和通道 1 0：O4CPRE 和 O1CPREF 独立输出 1：O1CPREF 输出为 O1CPREF 和 O4CPRE 的逻辑“与” 该位域可立即生效或者在下一次更新事件发生时生效（当 TIMERx_CHCTL1 中的比较影子寄存器使能）。
29	CCH4CH0	组合通道 4 和通道 0 0：O4CPRE 和 O0CPREF 独立输出 1：O0CPREF 输出为 O0CPREF 和 O4CPRE 的逻辑“与” 该位域可立即生效或者在下一次更新事件发生时生效（当 TIMERx_CHCTL1 中的比较影子寄存器使能）。
28:16	保留	必须保持复位值。
15:0	CH4VAL[15:0]	通道 4 的捕获比较值 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。

附加通道控制寄存器 0 (TIMER0_AFCTL0)

地址偏移: 0x60

复位值: 0x0000 0001

该寄存器只能按字 (32位) 访问



位/位域	名称	描述
31:18	保留	必须保持复位值。
17:14	ETISEL	ETI 触发源选择 该位域选择 ETI 的触发源 0000: ETI 保留模式 0011: ADC_WD0_OUT 0100: ADC_WD1_OUT 0101: ADC_WD2_OUT 其他: 保留 当 TIMERx_CCHP 寄存器的 PROT [1:0]=01, 此位不能被改变。
13:10	保留	必须保持复位值。
9	BRK0INP	BREAK0备用输入极性 该位用于配置BRKIN0输入极性, 具体极性是由该位和BRK0P位共同确定。 0: BRKIN0输入信号不反相 (BRK0P =0, 输入信号低有效; BRK0P =1, 输入信号高有效) 1: BRKIN0 输入信号反相 (BRK0P =0, 输入信号高有效; BRK0P =1, 输入信号低有效) 此位只有在TIMERx_CCHP寄存器的PROT [1:0] =00时才可修改。
8:1	保留	必须保持复位值。
0	BRK0INEN	BREAK0备用输入使能 0: BRKIN0输入禁能 1: BRKIN0输入使能 此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0] =00 时才可修改。

附加通道控制寄存器 1 (TIMER0_AFCTL1)

地址偏移: 0x64

复位值：0x0000 0001

该寄存器只能按字（32位）访问



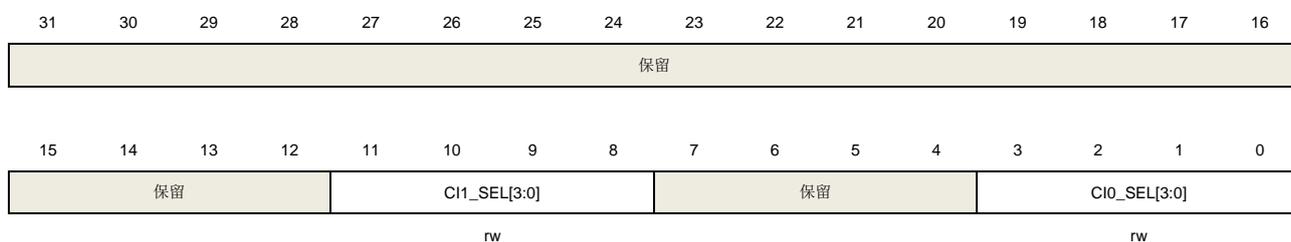
位/位域	名称	描述
31:10	保留	必须保持复位值。
9	BRK1INP	BRKIN1备用功能输入极性 该位用于配置BRKIN1输入极性，具体极性是由该位和BRK0P位共同确定。 0: BRKIN1输入信号不反相（BRK0P =0，输入信号低有效；BRK0P =1，输入信号高有效） 1: BRKIN1输入信号反相（BRK0P =0，输入信号高有效；BRK0P =1，输入信号低有效） 此位只有在TIMERx_CCHP寄存器的PROT [1:0] =00时才可修改。
8:1	保留	必须保持复位值
0	BRK1INEN	BRKIN1备用输入使能 0: BRKIN1输入禁能 1: BRKIN1输入使能 此位只有在TIMERx_CCHP寄存器的PROT [1:0] =00时才可修改。

输入选择寄存器（TIMERx_INSEL）

地址偏移：0x68

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。
11:8	CI1_SEL[3:0]	TIMER0_CH1输入选择 0000: TIMER0_CH1输入捕获

		0001: 比较器1输出 其他: 保留
7:4	保留	必须保持复位值
3:0	CI0_SEL[3:0]	TIMER0_CH0输入选择 0000: TIMER0_CH0输入捕获 0001: 比较器0输出 其他: 保留

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。 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: 无影响。
0	OUTSEL	输出值选择位 此位由软件写 1 或清 0。 1: 如果 POEN 位与 IOS 位均为 0, 则输出无效。 0: 无影响。

14.2. 通用定时器 L0 (TIMERx, x=2)

14.2.1. 简介

通用定时器 L0 (TIMER2) 是 4 通道定时器，支持输入捕获，输出比较，产生 PWM 信号控制电机和电源管理。通用定时器 L0 计数器是 16 位无符号计数器。

通用定时器 L0 是可编程的，可以被用来计数，其外部事件可以驱动其他定时器。

定时器和定时器之间是相互独立，但是它们的计数器可以被同步在一起形成一个更大的定时器。

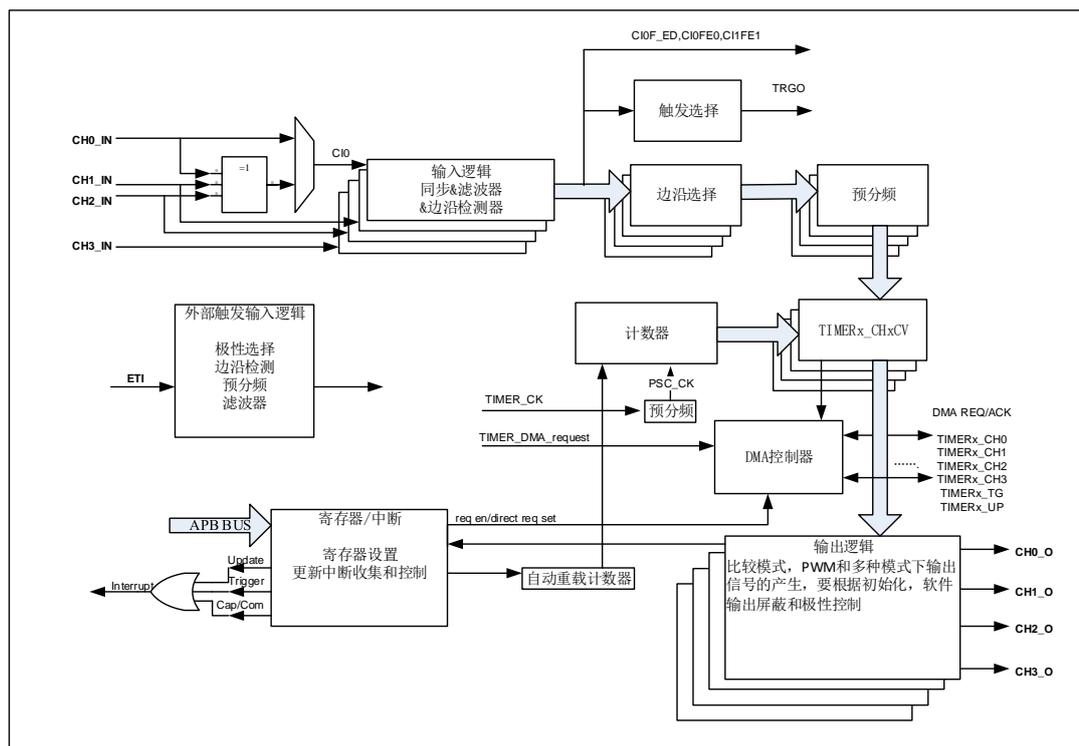
14.2.2. 主要特征

- 总通道数：4；
- 计数器宽度：16位；
- 时钟源可选：内部时钟，内部触发，外部输入，外部触发；
- 多种计数模式：向上计数，向下计数和中央计数；
- 正交译码器接口：被用来追踪运动和分辨旋转方向和位置；
- 霍尔传感器接口：用来做三相电机控制；
- 可编程的预分频器：16位，运行时可以被改变；
- 每个通道可配置：输入捕获模式，输出比较模式，可编程的PWM模式，单脉冲模式；
- 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。

14.2.3. 结构框图

[图 14-39.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 14-39.通用定时器 L0 结构框图



14.2.4. 功能说明

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 TSCFGy[2:0]位确定的时钟源，TSCFGy[2:0]位于 SYSCFG_TIMER0CFG，(y=0,1...7)。

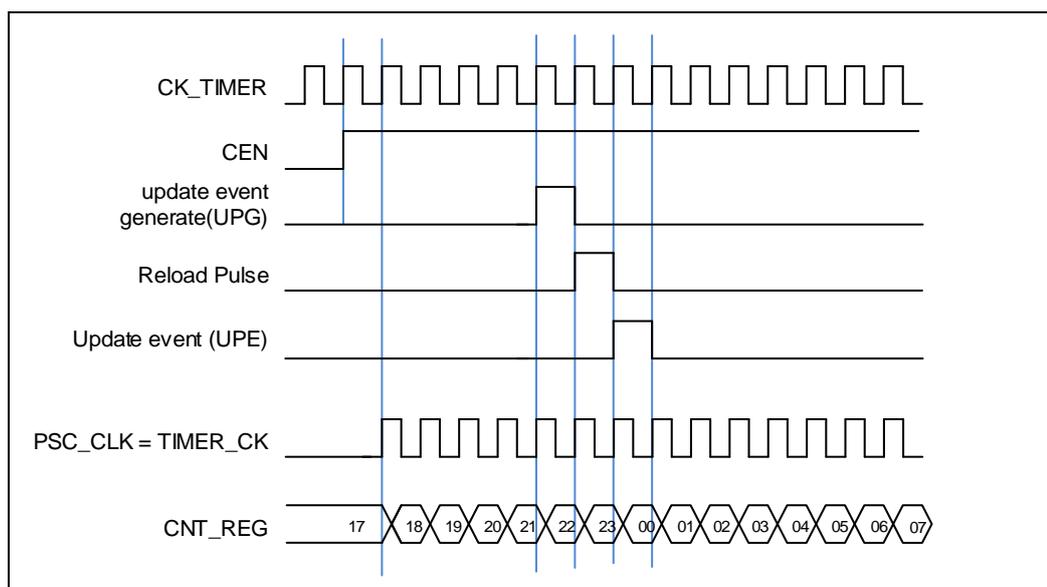
- TSCFGy[3:0] =3'b000，TSCFGy[2:0] 位于 SYSCFG_TIMER1CFG 或 SYSCFG_TIMER2CFG，(y=0,1...7)，定时器选择内部时钟源（连接到RCU模块的 CK_TIMER）。

当 TSCFGy[2:0] =3'b000，TSCFGy[2:0] 位于 SYSCFG_TIMER1CFG 或 SYSCFG_TIMER2CFG，(y=0,1...7)，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 TSCFGy[2:0] !=3'b000，TSCFGy[2:0] 位于 SYSCFG_TIMER1CFG 或 SYSCFG_TIMER2CFG，(y=0,1,2,6)，预分频器被其他时钟源（由TSCFG6[2:0]区域选择）驱动，更多细节在下文说明，当TSCFGy[2:0] (y=3,4,5) 设置为有效值时，计数器预分频器时钟源由内部时钟TIMER_CK驱动。

图 14-40. 内部时钟分频为 1 时，计数器的时序图



- $TSCFG6[3:0] \neq 3'b000$ （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 $TSCFG6[2:0]$ 为 $0x4$, $0x5$ 或 $0x6$ 来选择。

计数器预分频器也可以在内部触发信号 $ITI0/2/3$ 的上升沿计数。这种模式可以通过设置 $TSCFG6[2:0]$ 为 $0x1$, $0x2$, $0x3$ 。

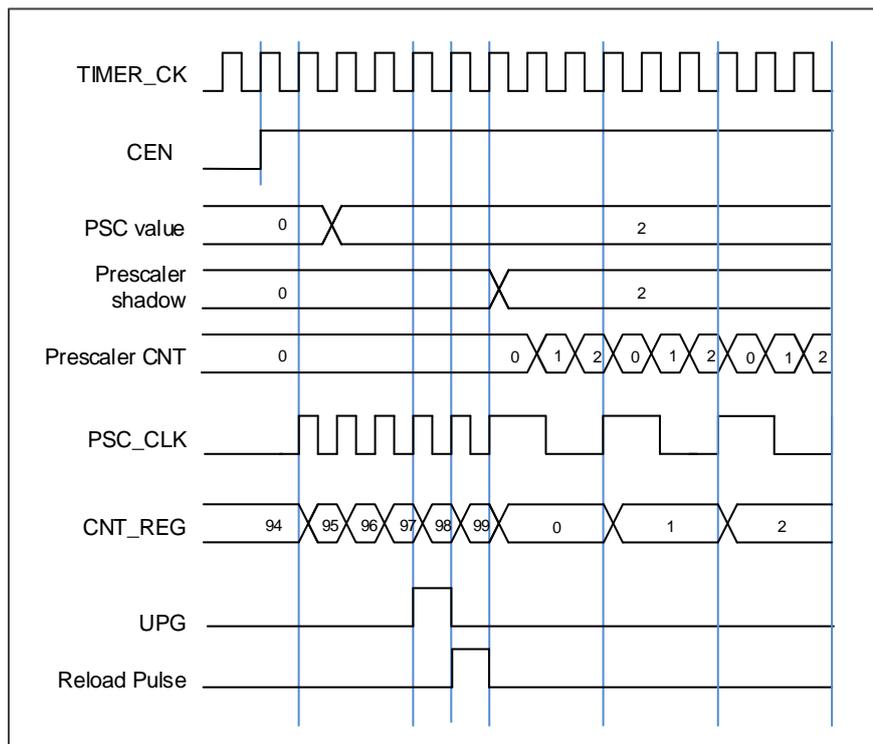
- $SMC1=1'b1$ （外部时钟模式1），定时器选择外部输入引脚 ETI 作为时钟源。

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 $TSCFG6[2:0]$ 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次更新事件到来时被采用。

图 14-41. 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 14-10. 在向上计数模式下计数器重复时序图](#)和 [图 14-11. 在向下计数模式下计数器重复时序图](#)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 14-42. 向上计数时序图, PSC=0/2

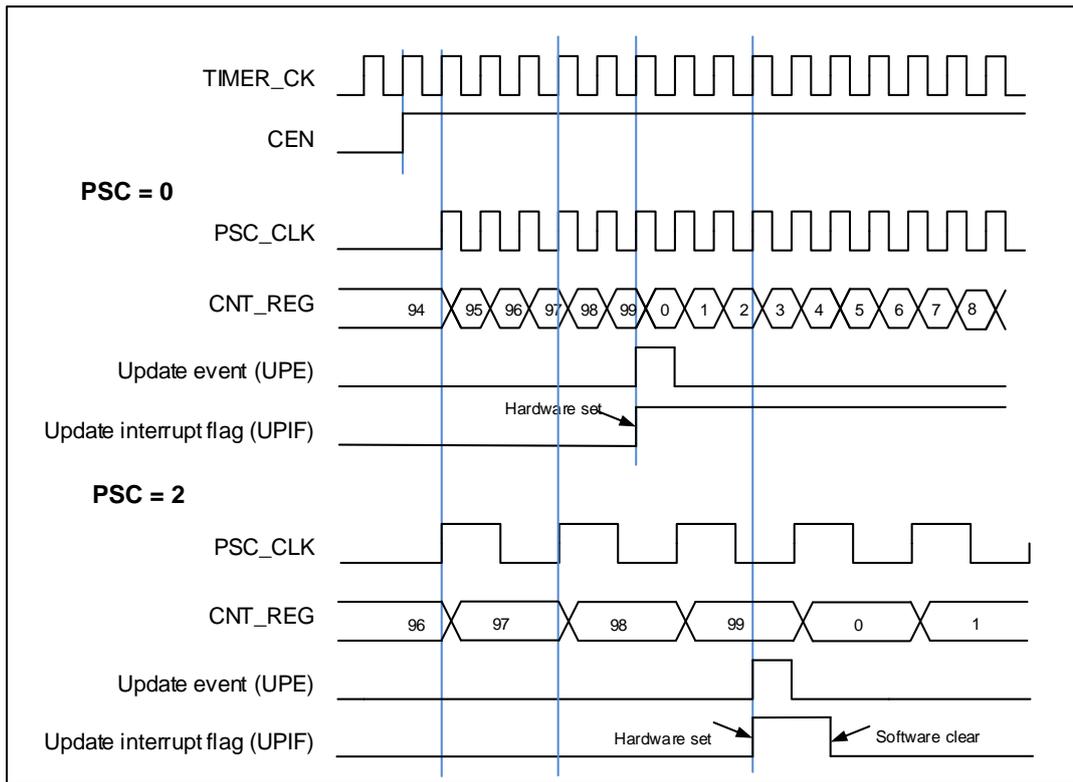
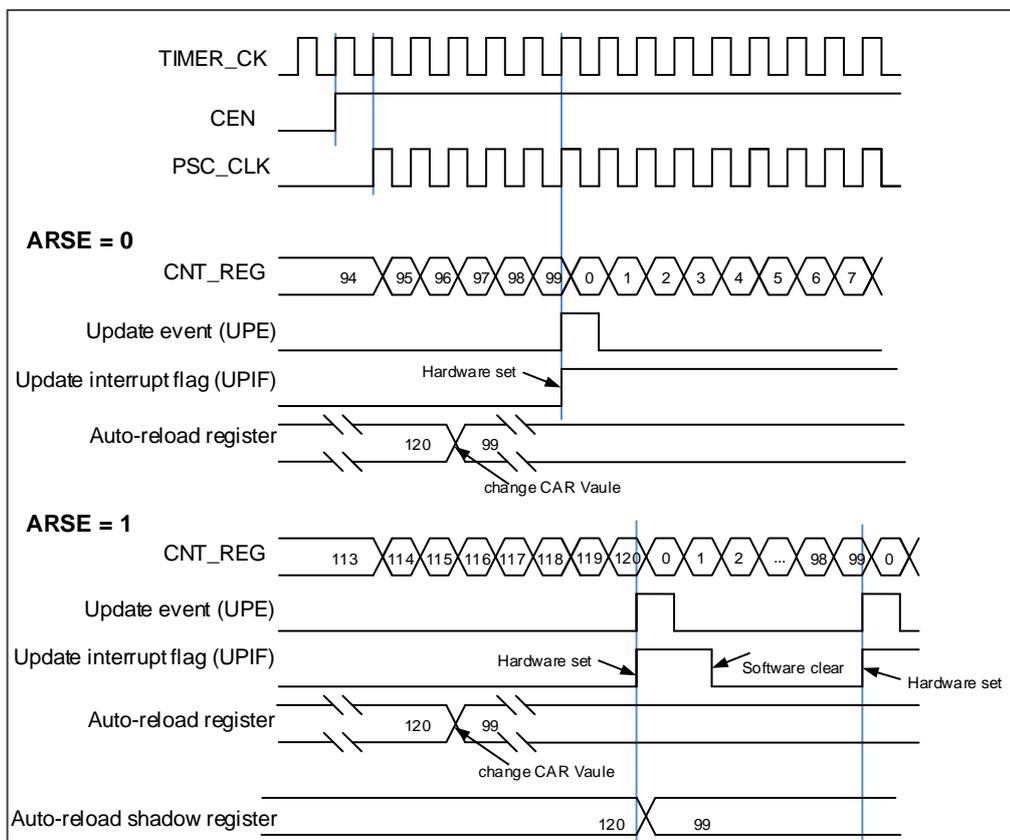


图 14-43. 向上计数时序图, 在运行时改变 TIMERx_CAR 寄存器的值



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。

和给出了一些例子，当 `TIMERx_CAR=0x99` 时，计数器在不同时钟频率下的行为

[图 14-44. 向下计数时序图，PSC=0/2](#) 和 [图 14-45. 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 14-44. 向下计数时序图，PSC=0/2

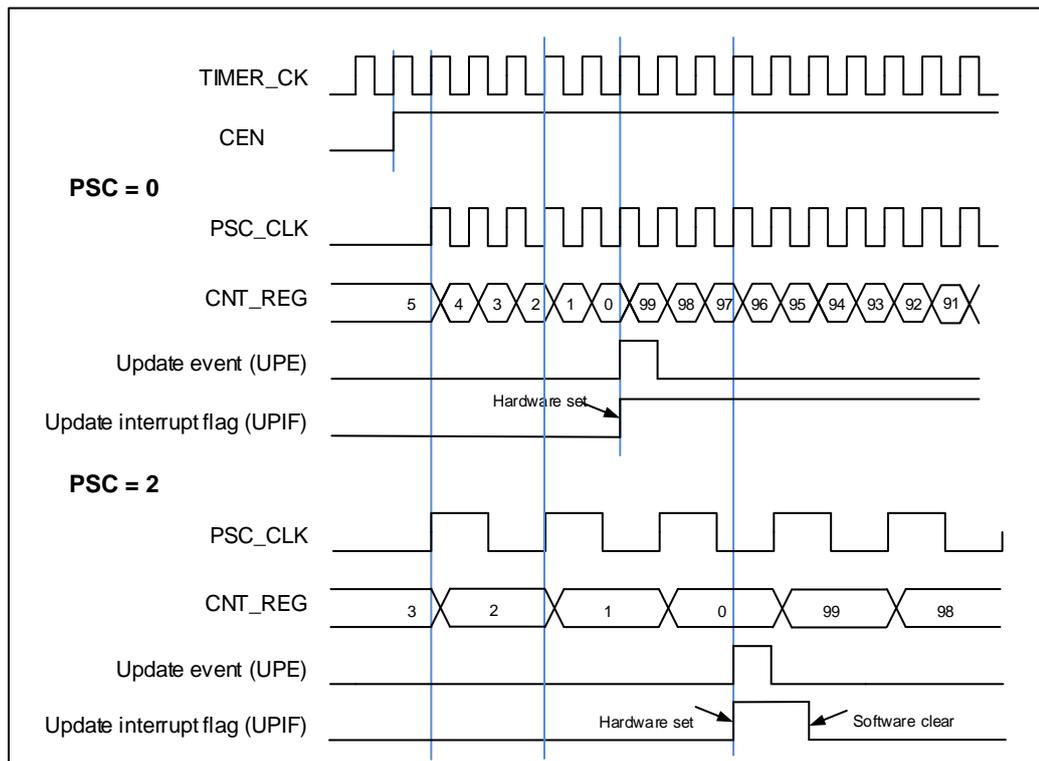
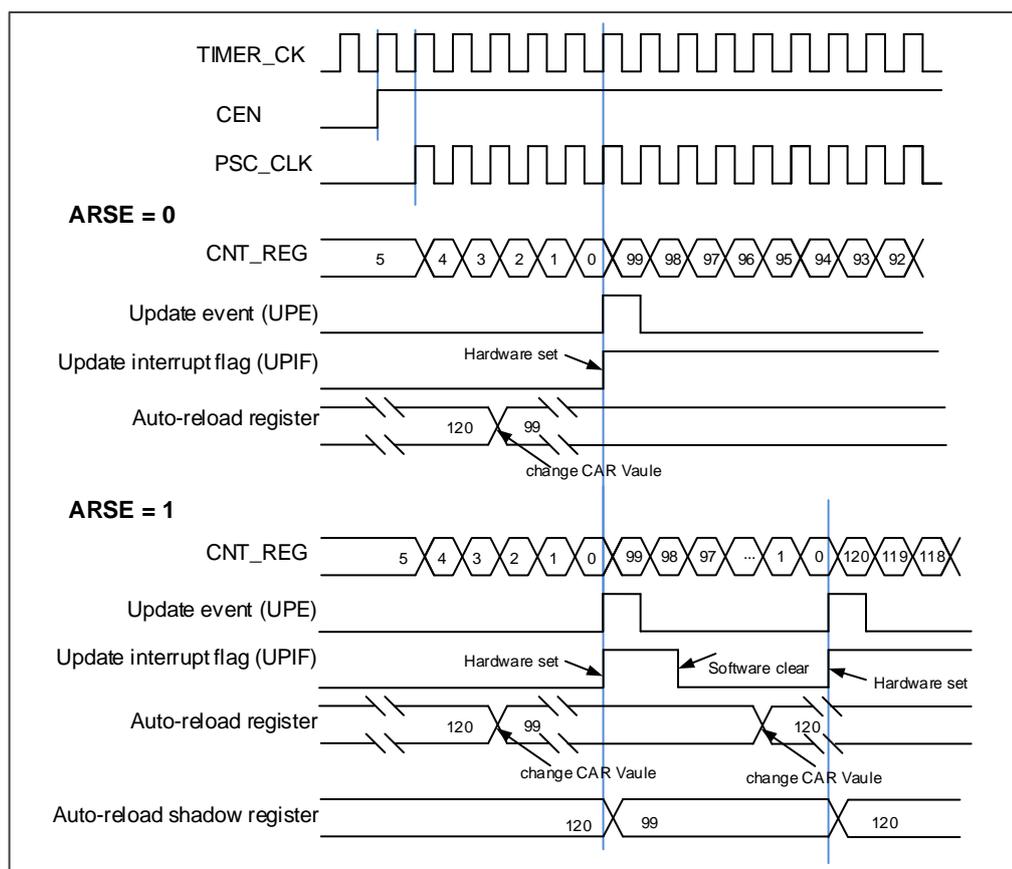


图 14-45. 向下计数时序图，在运行时改变 `TIMERx_CAR` 寄存器值

计数器中央对齐模式

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

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。

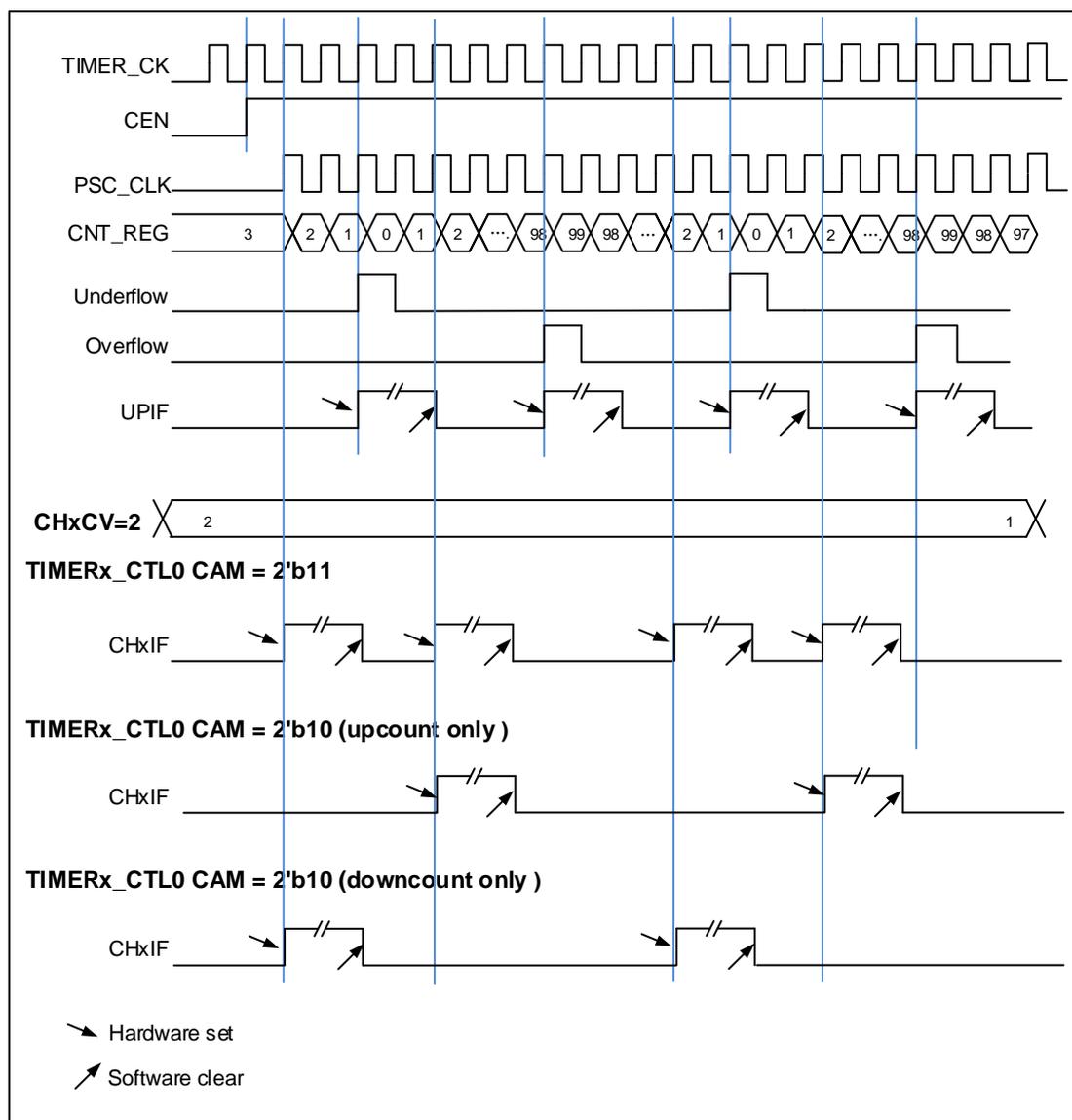
上溢或者下溢时，`TIMERx_INTF` 寄存器中的 `UPIF` 位都会被置 1。但是 `CHxIF` 位是否置 1 与 `TIMERx_CTL0` 寄存器中 `CAM` 的值有关。具体细节参考 [图 14-46. 中央计数模式计数器时序图](#)。

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 14-46. 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 14-46. 中央计数模式计数器时序图



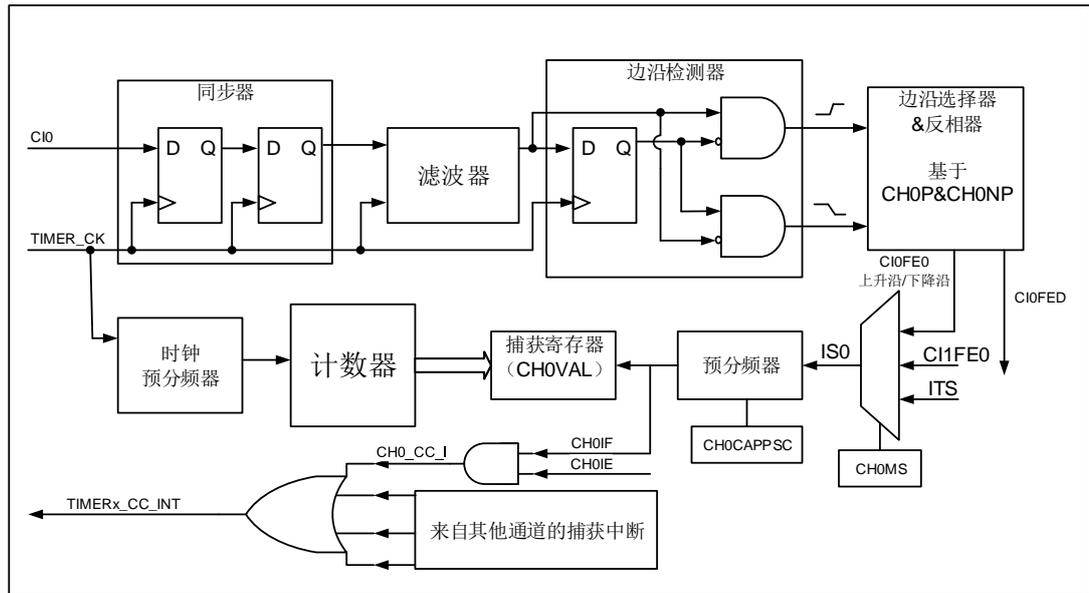
输入捕获和输出比较通道

通用定时器 L0 拥有四个独立的通道用于捕获输入或比较输出是否匹配。每个通道都围绕一个通道捕获比较寄存器建立，包括一个输入级，通道控制器和输出级。

通道输入捕获功能

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

图 14-47. 通道输入捕获原理



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

配置步骤如下：

第一步： 滤波器配置（ $TIMERx_CHCTL0$ 寄存器中 $CHxCAPFLT$ ）：

根据输入信号和请求信号的质量，配置相应的 $CHxCAPFLT$ 。

第二步： 边沿选择（ $TIMERx_CHCTL2$ 寄存器中 $CHxP$ ）：

配置 $CHxP$ 选择上升沿或者下降沿。

第三步： 捕获源选择（ $TIMERx_CHCTL0$ 寄存器中 $CHxMS$ ）：

一旦通过配置 $CHxMS$ 选择输入捕获源，必须确保通道配置在输入模式（ $CHxMS!=0x0$ ），而且 $TIMERx_CHxCV$ 寄存器不能再被写。

第四步： 中断使能（ $TIMERx_DMAINTEN$ 寄存器中 $CHxIE$ 和 $CHxDEN$ ）：

使能相应中断，可以获得中断和DMA请求。

第五步： 捕获使能（ $TIMERx_CHCTL2$ 寄存器中 $CHxEN$ ）。

结果： 当期望的输入信号发生时， $TIMERx_CHxCV$ 被设置成当前计数器的值， $CHxIF$ 为置1。

如果 $CHxIF$ 位已经为1，则 $CHxOF$ 位置1。根据 $TIMERx_DMAINTEN$ 寄存器中 $CHxIE$ 和 $CHxDEN$ 的配置，相应的中断和DMA请求会被提出。

直接产生： 软件设置 $CHxG$ 位，会直接产生中断和DMA请求。

输入捕获模式也可用来测量 $TIMERx_CHx$ 引脚上信号的脉冲波宽度。例如，一个 PWM 波连接到 CIO 。配置 $TIMERx_CHCTL0$ 寄存器中 $CH0MS$ 为 $2'b01$ ，选择通道 0 的捕获信号为 CIO

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 14-48. 通道输出比较原理 (x=0, 1, 2, 3)

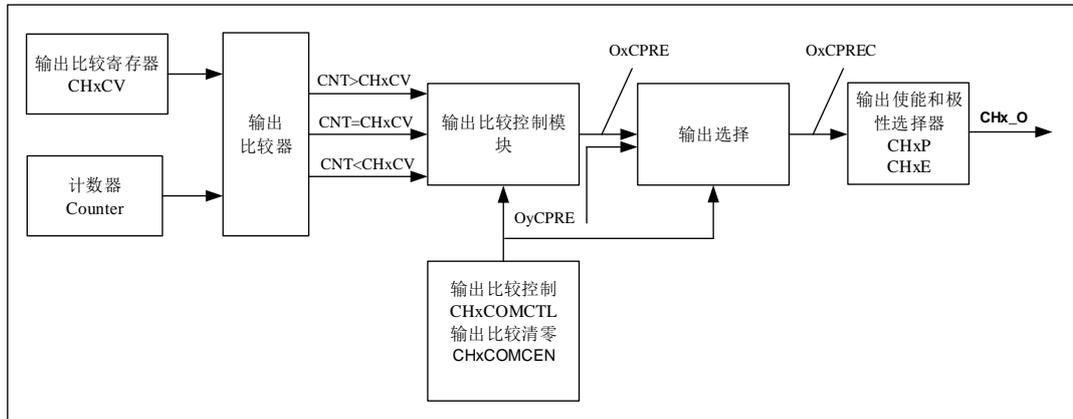


图 14-48. 通道输出比较原理(x=0, 1, 2, 3)给出了输出比较的原理电路。通道输出信号 `CHx_O` 与 `OxCMPRE` 信号(详情请见[通道输出准备信号](#))的关系描述如下: `OxCMPREC` 信号高电平有效, `CHx_O` 的输出情况与 `OxCMPRE` 信号, `CHxP` 位和 `CHxE` 位有关(具体情况请见 `TIMERx_CHCTL2` 寄存器中的描述)。当 `CHxCMOCTL[3:0] < 4'b1100` 时, `OxCMPREC = OxCMPRE`; 当 `CHxCMOCTL[3:0] = 4'b1100` 或 `4'b1101` 时, 输出复合 PWM 波形, 此时 `OxCMPREC` 为 `OxCMPRE` 与 `OyCMPRE(y = x+1 或 x-1)` 的逻辑“与”或逻辑“或”; 当 `CHxCMOCTL[3:0] = 4'b1110` 或 `4'b1111` 时, 输出非对称 PWM 波形, 此时 `OxCMPREC` 会在向上计数时输出 `OxCMPRE` 或 `OyCMPRE(y = x+1 或 x-1)` 或向下计数时输出 `OyCMPRE` 或 `OxCMPRE(y = x+1 或 x-1)`; 例如, 当设置 `CHxP=0` (`CHx_O` 高电平有效, 与 `OxCMPRE` 输出极性相同)、`CHxE=1` (`CHx_O` 输出使能) 时:

- 若 `OxCMPRE` 输出有效 (高) 电平, 则 `CHx_O` 输出有效 (高) 电平;
- 若 `OxCMPRE` 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, 被置低电平或者反转。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/反转);
- 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。

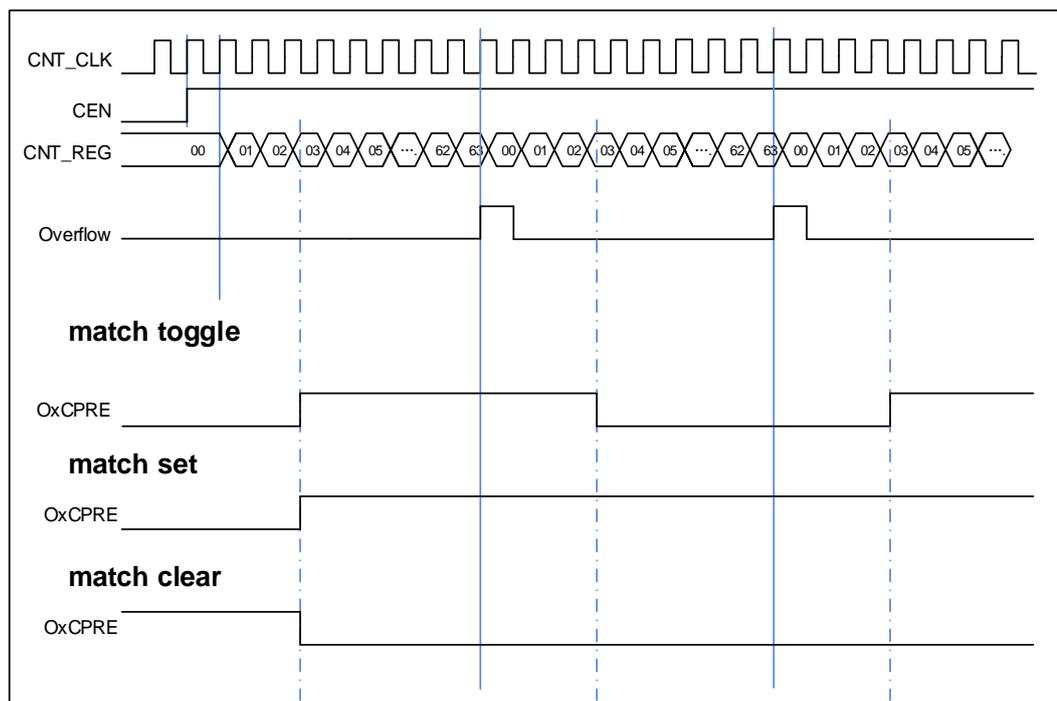
第四步：通过TIMERx_CAR寄存器和TIMERx_CHxCV寄存器配置输出比较时基：

CHxVAL可以在运行时根据你所期望的波形而改变。

第五步：设置CEN位使能定时器。

[图 14-49. 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 14-49. 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 14-50. 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 14-51. 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 14-50. EAPWM 时序图

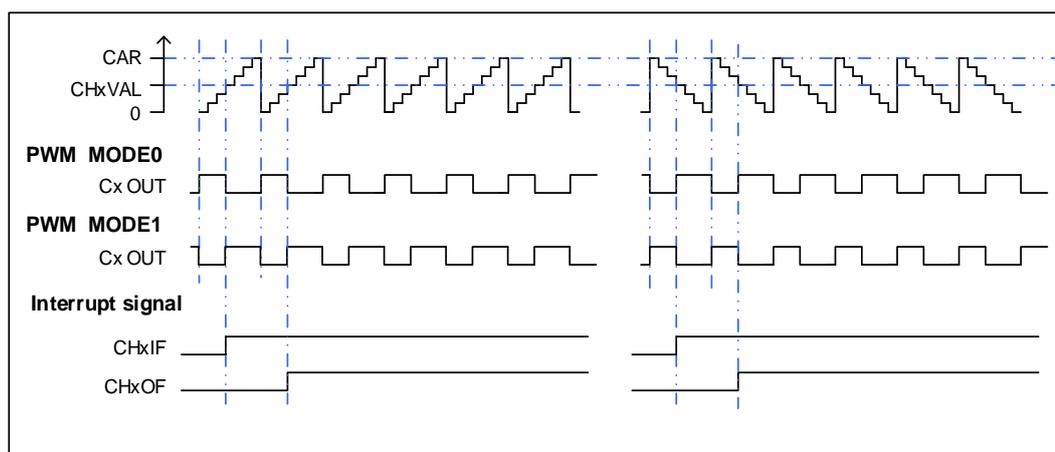
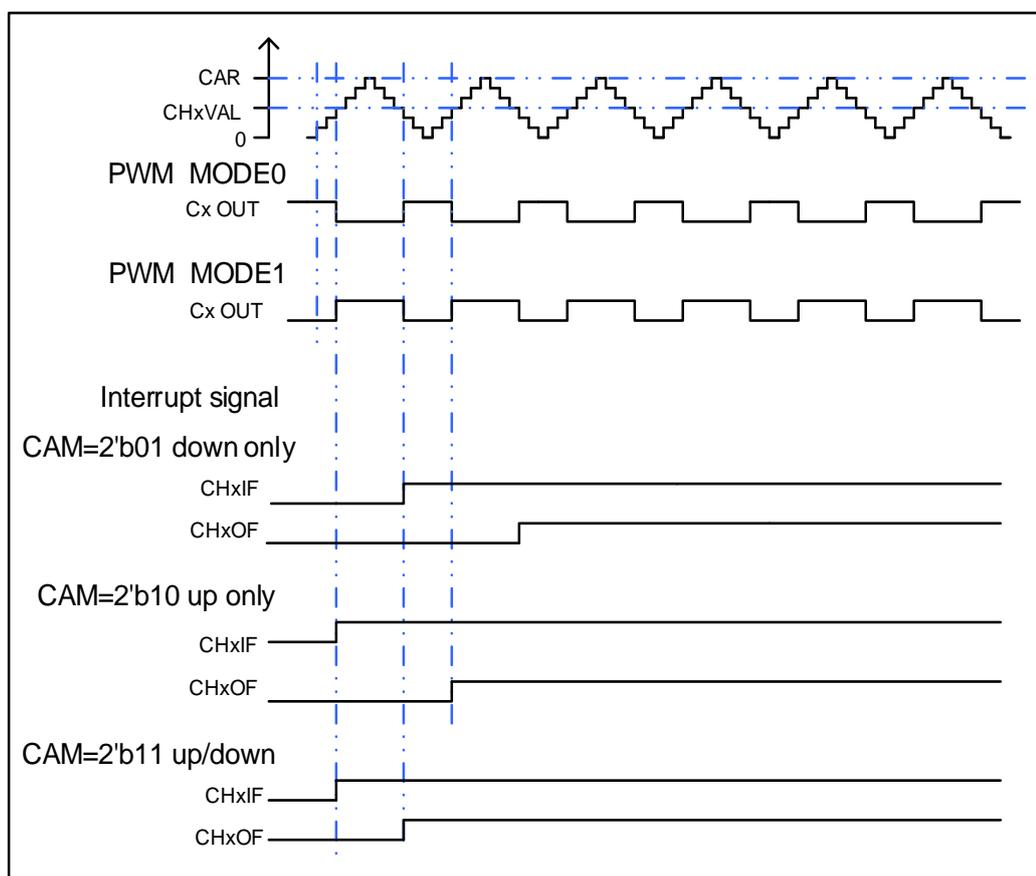


图 14-51. CAPWM 时序图



非对称 PWM 模式

非对称PWM模式0 / 1（通过将CHxCOMCTL[3:0]位域设置为4'b1110或4'b1111）用于中央对齐PWM模式以产生可编程的相移。CPWM的频率由TIMERx_CAR寄存器决定，占空比和相移由一对具有相邻偏移地址的TIMER_CHxCV / TIMER_CH(x + 1)CV（或TIMER_CH(x - 1)CV）寄存器决定。

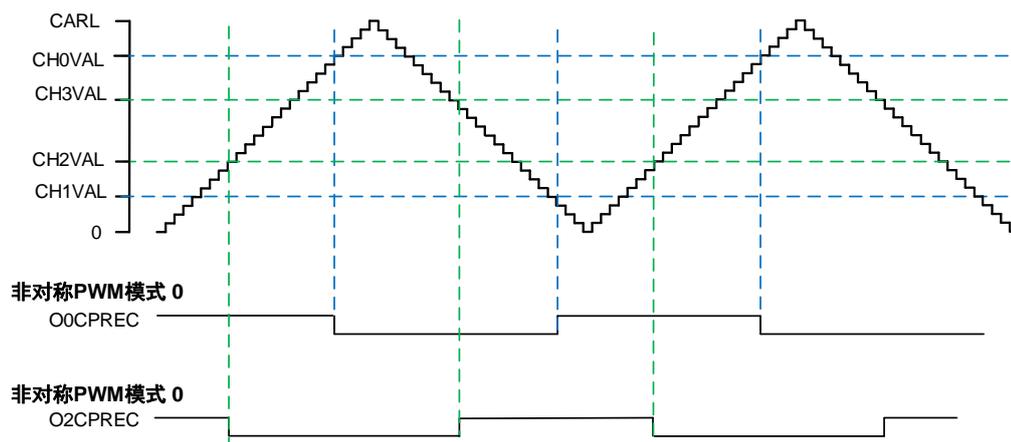
TIMER_CHxCV寄存器决定了向上计数的波形，TIMER_CH(x + 1)CV（或TIMER_CH(x - 1)CV）

寄存器决定了向下计数的波形。详情如下：

- 通道O0CPREC / O1CPREC输出的CPWM波形由TIMER_CH0CV和TIMER_CH1CV寄存器决定；
- 通道O2CPREC / O3CPREC输出的CPWM波形由TIMER_CH2CV和TIMER_CH3CV寄存器决定。

当使用非对称PWM模式时，CH0 / CH1（或CH2 / CH3）可以独立输出不同的波形，这些波形由CHxCOMCTL[3:0]位域配置（两个通道可以配置不同的值）。

图 14-52. O0CPREC 和 O2CPREC 使用非对称 PWM 模式



复合 PWM 模式

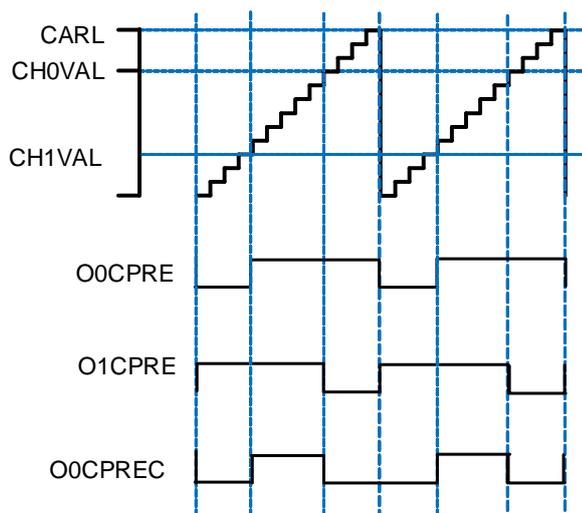
复合PWM模式0 / 1（通过将CHxCOMCTL[3:0]位域设置为4'b1100或4'b1101）用于中央对齐PWM模式以产生可编程的相移。CPWM的频率由TIMERx_CAR寄存器决定，占空比和相移由一对具有相邻偏移地址的TIMER_CHxCV / TIMER_CH(x + 1)CV（或TIMER_CH(x - 1)CV）寄存器决定。

- 通道O0CPREC / O1CPREC输出的CPWM波形由TIMER_CH0CV和TIMER_CH1CV寄存器决定；
- 通道O2CPREC / O3CPREC输出的CPWM波形由TIMER_CH2CV和TIMER_CH3CV寄存器决定。

当通道O0CPREC选择复合PWM模式0时（CH0COMCTL[3:0] == 4'b1100），此时O0CPREC输出O0CPRE与O1CPRE的逻辑“或”信号；当通道O0CPREC选择复合PWM模式1时（CH0COMCTL[3:0] == 4'b1101），此时O0CPREC输出O0CPRE与O1CPRE的逻辑“与”信号。当O0CPREC工作在复合PWM模式0时，其互补通道工作在复合PWM模式1。

当选择复合PWM模式时，通过配置CHxCOMCTL[3:0]位域不同值，每个通道CH0/CH1,或CH2/CH3可输出的不同的波形。

图 14-53. O0CPRE 选择复合 PWM 模式 1 和 O1CPRE 选择 PWM 模式 0



可再次触发单脉冲模式

该模式允许计数器在响应触发信号时生成一个可编程长度的脉冲。

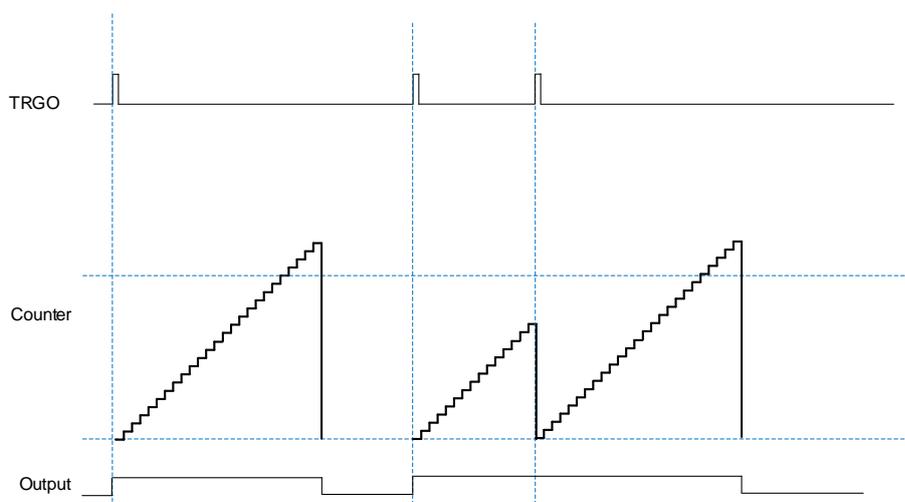
- 在接收到触发信号后，立即输出脉冲信号（无延时）。
- 如果在当前脉冲结束之前接收到新的触发信号，脉冲持续时间将被延长。

定时器需运行在主-从管理模式下，同时设置 SYSCFG_TIMERxCFG0(x = 0)中的 TSCFG7[2:0]寄存器，通过配置 CHxCOMCTL[3:0]位于为 4'b1100 或 4'b1101 选择可再次触发单脉冲模式 0/1。

在向上计数模式下，将 TIMERx_CHyCV 设置为 0；脉冲长度由 TIMERx_CAR 寄存器定义。在向下计数模式下，确保 TIMERx_CHyCV 大于或等于 TIMERx_CAR。

注意：此模式与中心对齐的 PWM 模式不兼容。在 TIMERx_CTL0 中设置 CAM[1:0] = 2'b00。

图 14-54. 可再次触发单脉冲模式



通道输出准备信号

根据[图 14-48. 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。在下次更新事件到来时, `OxCPRE` 信号才会回到有效电平状态。

正交译码器

正交译码器功能使用由 `TIMERx_CH0` 和 `TIMERx_CH1` 引脚生成的 `CI0` 和 `CI1` 正交信号各自相互作用产生计数值。通过设置 `TSCFGy[2:0] != 3'b000` ($y=0,1,2$) 来选择是仅由 `CI0`, 仅由 `CI1`, 或者由 `CI0` 和 `CI1` 来决定定时器的计数方向。在每个方向选择源的电平改变期间, `DIR` 位改变。计数器计数方向改变的机制如[表 14-3. 计数方向与正交译码器信号之间的关系](#)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 14-5. 计数方向与正交译码器信号之间的关系

计数模式	电平	CI0FE0		CI1FE1	
		上升	下降	上升	下降
只有 CI0	CI1FE1=1	向下	向上	-	-
	CI1FE1=0	向上	向下	-	-
只有 CI1	CI0FE0=1	-	-	向上	向下
	CI0FE0=0	-	-	向下	向上
CI0 和 CI1	CI1FE1=1	向下	向上	X	X
	CI1FE1=0	向上	向下	向上	向下
	CI0FE0=1	X	X	Up	Down
	CI0FE0=0	X	X	向下	向上

注意: “-”意思是“无计数”; “X”意思是不可能。

图 14-55. 正交译码器接口模式下计数器运行例子

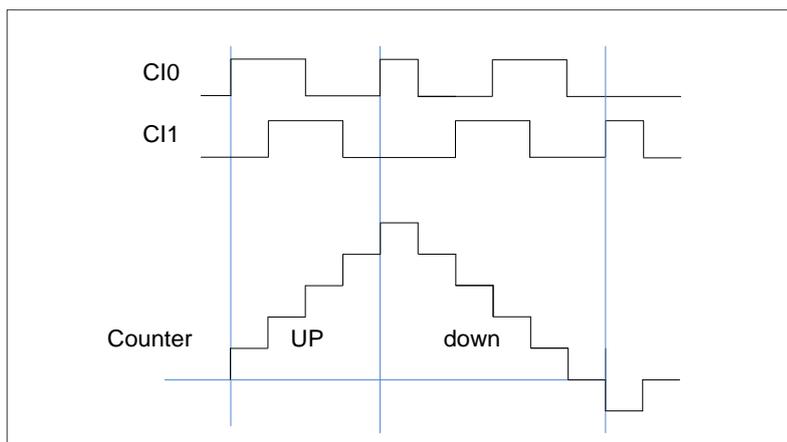
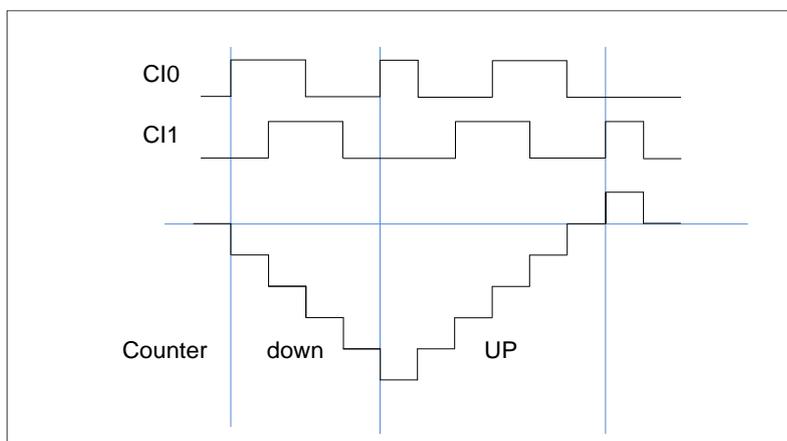


图 14-56. CI0FE0 极性反相的正交译码器接口模式下的例子



霍尔传感器接口功能

通用定时器 L0 支持霍尔传感器接口功能，该功能可以用来控制 BLDC 电机。

三个霍尔传感器与 `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。

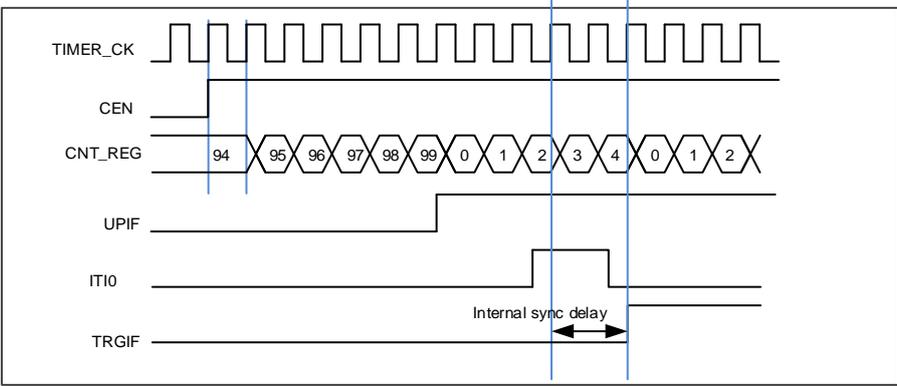
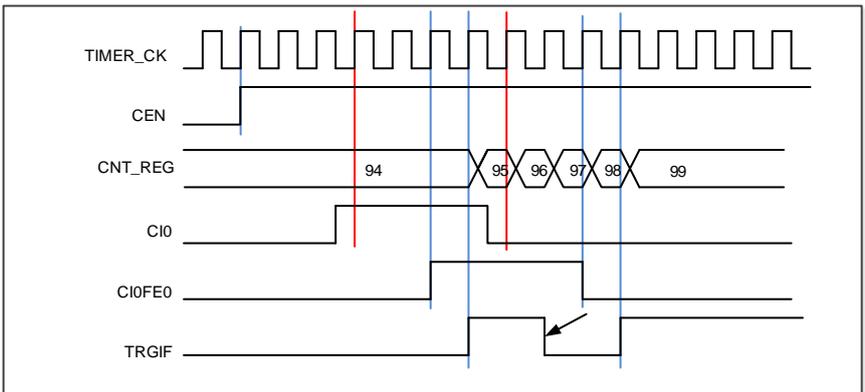
置位 `TI0S` 可以使能异或功能，则每个输入信号电平发生变化时 `CI0` 都会翻转。`CHOVAL` 将会记录 `CI0` 翻转时的计数器数值。

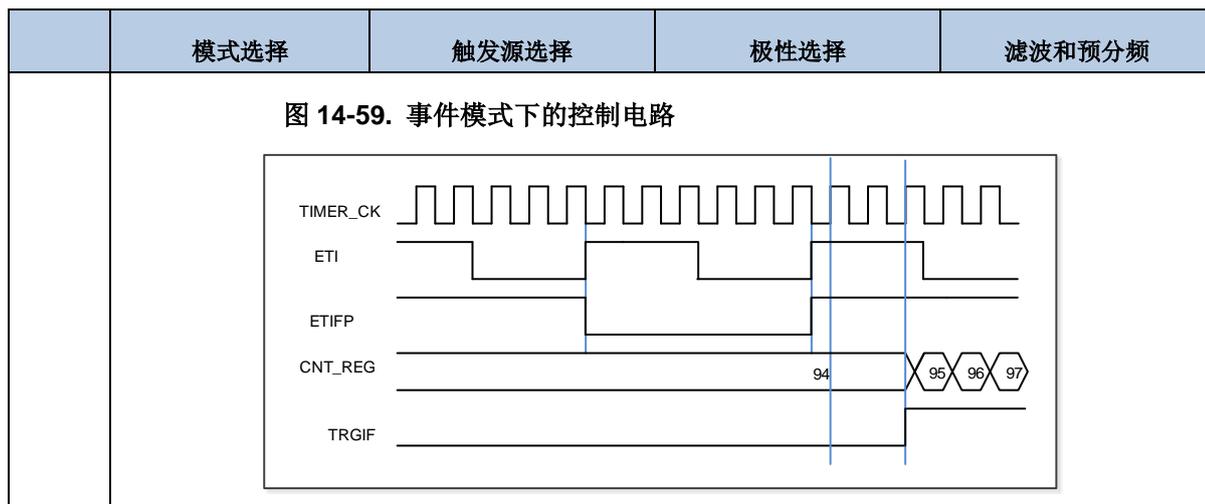
主-从管理

`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。通过设置 `SYSCFG_TIMER1CFG` 或 `SYSCFG_TIMER2CFG` ($y=3,4,5$) 寄存器中的 `TSCFGy[2:0] != 3b'000` ($y=3,4,5$) 配置这些模式。

表 14-6. 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	<code>TSCFGy[2:0]</code> <code>y=3'b100</code> (复位模式)	<code>TSCFGy[2:0]</code> 001: <code>ITI0</code>	如果触发源是 <code>CI0FE0</code> 或者 <code>CI1FE1</code> ，配置 <code>CHxP</code> 和	触发源 <code>ITIx</code> ，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
	y=3'b101 (暂停模式) y=3'b110 (事件模式)	010: ITI2 011: ITI3 100: CI0F_ED 101: CI0FE0 110: CI1FE1 111: ETIFP	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 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
例1	复位模式 当触发输入上升沿, 计数器清零重启	TSCFG3[2:0] = 3'b001 选择ITIO为触发源	触发源是ITIO, 极性选择不可用	触发源是 ITIO, 滤波和预分频不可用
图 14-57. 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	TSCFG4[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 14-58. 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	TSCFG5[2:0] = 3'b111 选择ETIF为触发源.	ETP = 0 没有极性改变	ETPSC = 1, 2分频. ETFC = 0, 无滤波



单脉冲模式

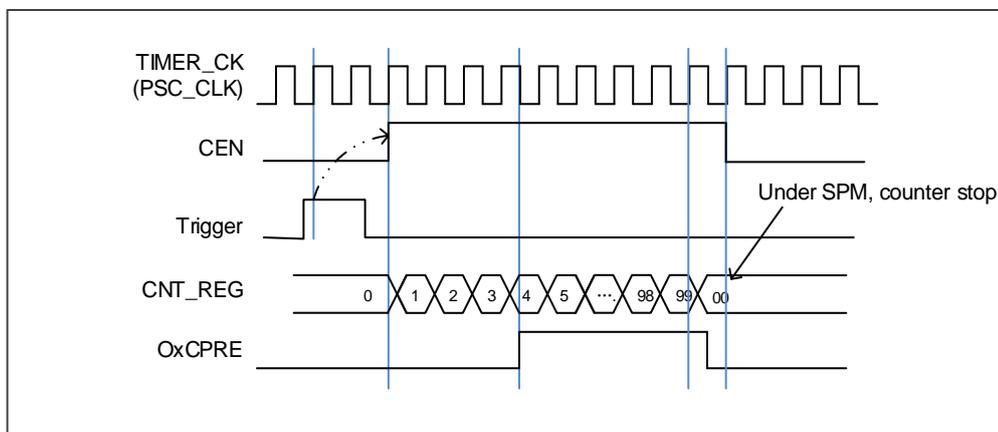
单脉冲模式与重复模式是相反的，设置 `TIMERx_CTL0` 寄存器的 `SPM` 位置 1，则使能单脉冲模式。当 `SPM` 置 1，计数器在下次更新事件到来后清零并停止计数。为了得到脉冲波，可以通过设置 `CHxCOMCTL` 配置 `TIMERx` 为 PWM 模式或者比较模式。

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 PWM1 或 PWM2 输出运行模式下时 `CHxCOMFEN` 位才可用，触发源来源于触发信号。

[图 14-60. 单脉冲模式, `TIMERx_CHxCV = 4` `TIMERx_CAR=99`](#) 展示了一个例子

图 14-60. 单脉冲模式, `TIMERx_CHxCV = 4` `TIMERx_CAR=99`



定时器互连

定Refer to [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。

14.2.5. TIMERx 寄存器 (x=2)

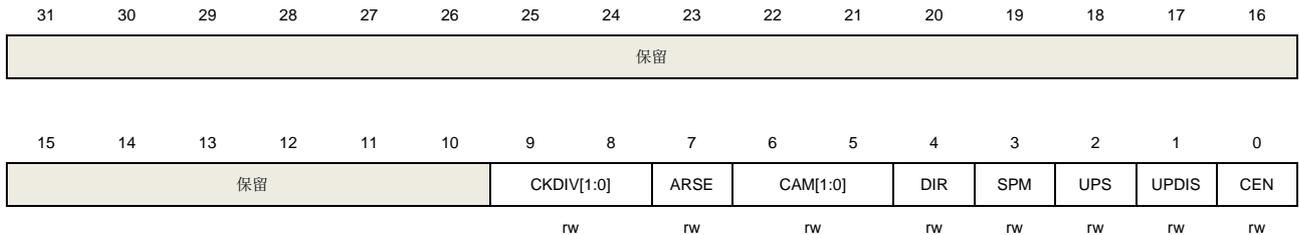
TIMER2基地址: 0x4000 0400

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。



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。

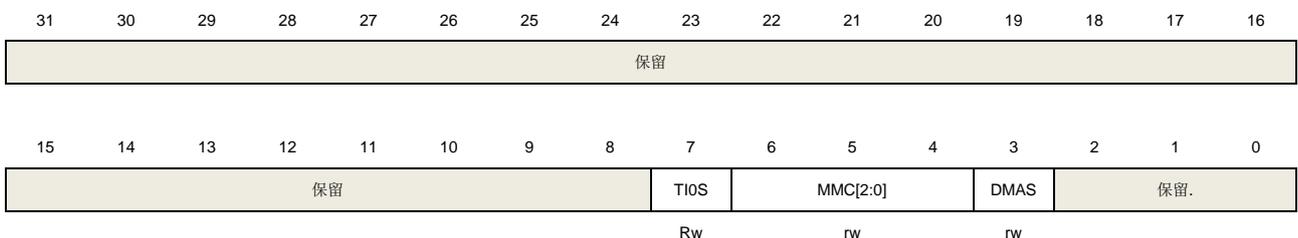
3	SPM	<p>单脉冲模式</p> <p>0: 单脉冲模式禁能。更新事件发生后，计数器继续计数</p> <p>1: 单脉冲模式使能。在下次更新事件发生时，计数器停止计数</p>
2	UPS	<p>更新请求源</p> <p>软件配置该位，选择更新事件源。</p> <p>0: 以下事件均会产生更新中断或DMA请求： UPG位被置1 计数器溢出/下溢 复位模式产生的更新</p> <p>1: 下列事件会产生更新中断或DMA请求： 计数器溢出/下溢</p>
1	UPDIS	<p>禁止更新。</p> <p>该位用来使能或禁能更新事件的产生</p> <p>0: 更新事件使能。更新事件发生时，相应的影子寄存器被装入预装载值，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</p> <p>1: 更新事件禁能。</p> <p>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</p>
0	CEN	<p>计数器使能</p> <p>0: 计数器禁能</p> <p>1: 计数器使能</p> <p>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。</p>

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
31:8	保留	必须保持复位值。
7	TIOSt	<p>通道0触发输入选择</p> <p>0: 选择 TIMERx_CH0 引脚作为通道 0 的触发输入</p>

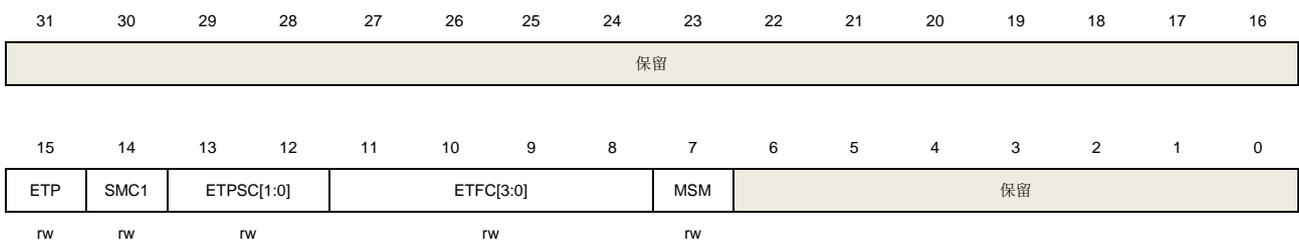
		1: 选择 <code>TIMERx_CH0</code> , <code>CH1</code> 和 <code>CH2</code> 引脚异或的结果作为通道 0 的触发输入
6:4	<code>MMC[2:0]</code>	<p>主模式控制</p> <p>这些位控制 <code>TRGO</code> 信号的选择, <code>TRGO</code> 信号由主定时器发给从定时器用于同步功能</p> <p>000: 当产生一个定时器复位事件后, 输出一个 <code>TRGO</code> 信号, 定时器复位源为: 主定时器产生一个复位事件 <code>TIMERx_SWEVG</code> 寄存器中 <code>UPG</code> 位置 1</p> <p>001: 当产生一个定时器使能事件后, 输出一个 <code>TRGO</code> 信号, 定时器使能源为: <code>CEN</code> 位置 1 在暂停模式下, 触发输入置 1</p> <p>010: 当产生一个定时器更新事件后, 输出一个 <code>TRGO</code> 信号, 更新事件源由 <code>UPDIS</code> 和 <code>UPS</code> 位决定</p> <p>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 <code>TRGO</code> 脉冲</p> <p>100: 当产生一次比较事件时, 输出一个 <code>TRGO</code> 信号, 比较事件源来自 <code>O0CPRE</code></p> <p>101: 当产生一次比较事件时, 输出一个 <code>TRGO</code> 信号, 比较事件源来自 <code>O1CPRE</code></p> <p>110: 当产生一次比较事件时, 输出一个 <code>TRGO</code> 信号, 比较事件源来自 <code>O2CPRE</code></p> <p>111: 当产生一次比较事件时, 输出一个 <code>TRGO</code> 信号, 比较事件源来自 <code>O3CPRE</code></p>
3	<code>DMAS</code>	<p>DMA 请求源选择</p> <p>0: 当通道捕获/比较事件发生时, 发送通道 x 的 DMA 请求 .</p> <p>1: 当更新事件发生, 发送通道 x 的 DMA 请求</p>
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	ETP	<p>外部触发极性</p> <p>该位指定 <code>ETI</code> 信号的极性</p> <p>0: <code>ETI</code> 高电平或上升沿有效 .</p> <p>1: <code>ETI</code> 低电平或下降沿有效 .</p>
14	SMC1	<code>SMC</code> 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 ETIFP 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 ETIFP 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 fsAMP 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	fsAMP
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 TRIG1 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

6:0 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。。

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。



位/位域	名称	描述
31:13	保留	必须保持复位值。
12	CH3OF	通道 3 捕获溢出标志 参见 CH0OF 描述
11	CH2OF	通道 2 捕获溢出标志 参见 CH0OF 描述
10	CH1OF	通道 1 捕获溢出标志 参见 CH0OF 描述
9	CH0OF	通道 0 捕获溢出标志 当通道 0 被配置为输入模式时, 在 CH0IF 标志位已经被置 1 后, 捕获事件再次发生时, 该标志位可以由硬件置 1。该标志位由软件清 0。 0: 无捕获溢出中断发生

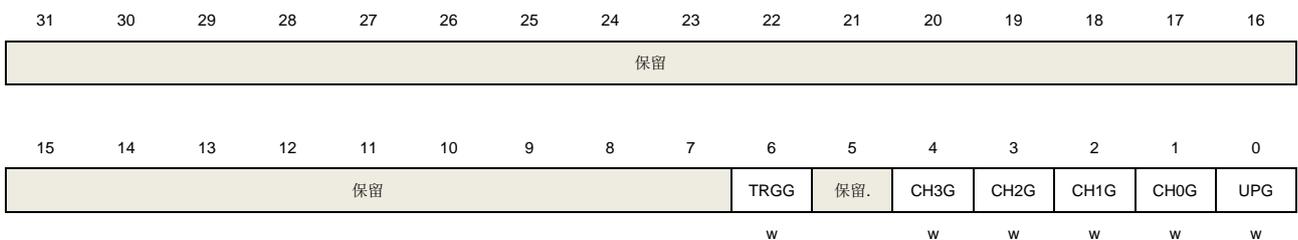
		1: 发生了捕获溢出中断
8:7	保留	必须保持复位值。.
6	TRGIF	触发中断标志 当发生触发事件时，此标志会置 1，此位由软件清 0。当暂停模式使能时，触发输入的任意边沿都可以产生触发事件。否则，其它模式时，仅在触发输入端检测到有效边沿，产生触发事件。 0: 无触发事件产生 1: 触发中断产生
5	保留	必须保持复位值。.
4	CH3IF	通道 3 比较/捕获中断标志 参见 CH0IF 描述
3	CH2IF	通道 2 比较/捕获中断标志 参见 CH0IF 描述
2	CH1IF	通道 1 比较/捕获中断标志 参见 CH0IF 描述
1	CH0IF	通道 0 比较/捕获中断标志 此标志由硬件置 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。 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
31:7	保留	必须保持复位值。.

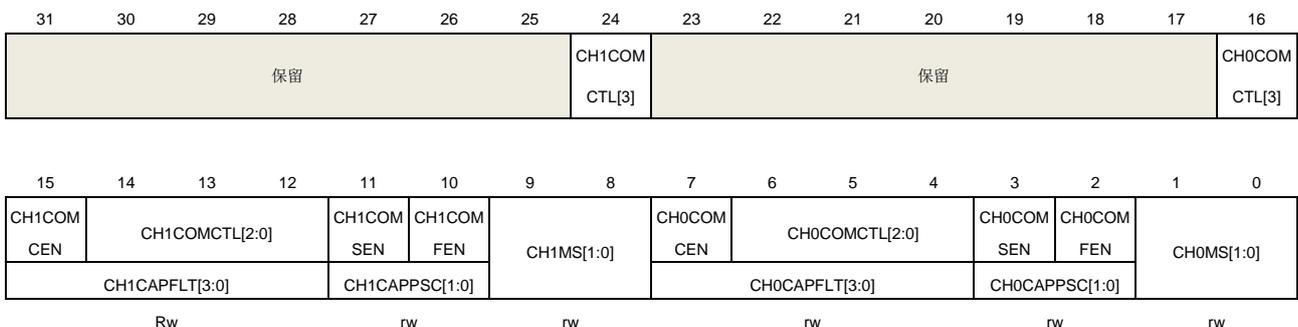
6	TRGG	<p>触发事件产生</p> <p>此位由软件置 1，由硬件自动清 0。当此位被置 1，TIMERx_INTF 寄存器的 TRGIF 标志位被置 1，若开启对应的中断和 DMA，则产生相应的中断和 DMA 传输。</p> <p>0：无触发事件产生</p> <p>1：产生触发事件</p>
5	保留	必须保持复位值。
4	CH3G	<p>通道 3 捕获或比较事件发生</p> <p>参见 CH0G 描述</p>
3	CH2G	<p>通道 2 捕获或比较事件发生</p> <p>参见 CH0G 描述</p>
2	CH1G	<p>通道 1 捕获或比较事件发生</p> <p>参见 CH0G 描述</p>
1	CH0G	<p>通道 0 捕获或比较事件发生</p> <p>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。</p> <p>0：不产生通道 0 捕获或比较事件</p> <p>1：发生通道 0 捕获或比较事件</p>
0	UPG	<p>更新事件产生</p> <p>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。</p> <p>0：无更新事件产生</p> <p>1：产生更新事件</p>

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
31:25	保留	必须保持复位值.
24	CH1COMCTL[3]	参见 CH1COMCTL[2:0]描述
23:17	保留	必须保持复位值.
16	CH0COMCTL[3]	参见 CH0COMCTL[2:0]描述
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, 此模式仅工作在内部触发器输入被选中时(由 SYSCFG_TIMER2CFG) 寄存器中的 TSCFGx[2:0] (x = 3,4,5,6,7)位域选择)。
7	CH0COMCEN	通道 0 输出比较清 0 使能 当此位被置 1, 当检测到 ETIFP 信号输入高电平时, O0CPRE 参考信号被清 0 0: 禁止通道 0 输出比较清零 1: 使能通道 0 输出比较清零
6:4	CH0COMCTL[2:0]	通道 0 输出比较模式 此位定义了 O0CPRE 的动作, 而 O0CPRE 决定了 CH0_O、CH0_ON 的值。O0CPRE 高电平有效, 而 CH0_O、CH0_ON 的有效电平取决于 CH0P、CH0NP 位。 0000: 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 0001: 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。 0010: 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。 0011: 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。 0100: 强制为低。强制 O0CPRE 为低电平 0101: 强制为高。强制 O0CPRE 为高电平 0110: PWM 模式 0。在向上计数时, 一旦计数器值小于 TIMERx_CH0CV 时, O0CPRE 为有效电平, 否则为无效电平。在向下计数时, 一旦计数器的值大于 TIMERx_CH0CV

		<p>时，O0CPRE 为无效电平，否则为有效电平。</p> <p>0111: PWM 模式 1。在向上计数时，一旦计数器值小于 <code>TIMERx_CH0CV</code> 时，O0CPRE 为无效电平，否则为有效电平。在向下计数时，一旦计数器的值大于 <code>TIMERx_CH0CV</code> 时，O0CPRE 为有效电平，否则为无效电平。</p> <p>在 PWM 模式 0 或 PWM 模式 1 中，只有当比较结果改变了或者输出比较模式中从时基模式切换到 PWM 模式时，O0CPRE 电平才改变。</p> <p>当 <code>TIMERx_CCHP</code> 寄存器的 <code>PROT[1:0]=11</code> 且 <code>CH0MS=00</code>（比较模式）时，此位不能被改变。</p> <p>1000: 可再次触发单脉冲模式 0。O0CPRE 工作在 PWM 模式 0，向上计数时，O0CPRE 有效，当外部触发信号产生时，O0CPRE 无效，在下次更新事件产生后，O0CPRE 恢复有效。向下计数时，O0CPRE 无效，当外部触发信号产生时，O0CPRE 有效，在下次更新事件产生后，O0CPRE 恢复无效。</p> <p>1001: 可再次触发单脉冲模式 1。O0CPRE 工作在 PWM 模式 1，向上计数时，O0CPRE 无效，当外部触发信号产生时，O0CPRE 有效，在下次更新事件产生后，O0CPRE 恢复无效。向下计数时，O0CPRE 有效，当外部触发信号产生时，O0CPRE 无效，在下次更新事件产生后，O0CPRE 恢复有效。</p> <p>1010:保留</p> <p>1011:保留</p> <p>1100:复合 PWM0 模式。O0CPRE 工作在 PWM 模式 0，O0CPREC 输出结果是 O0CPRE 和 O1CPRE 的逻辑“或”。</p> <p>1101:复合 PWM1 模式。O0CPRE 工作在 PWM 模式 1，O0CPREC 输出结果是 O0CPRE 和 O1CPRE 的逻辑“与”。</p> <p>1110: 非对称 PWM0 模式。O0CPRE 工作在 PWM 模式 0，在向上计数时，O0CPREC 输出结果是 O0CPRE，向下计数时输出 O1CPRE。</p> <p>1111 非对称 PWM1 模式。O0CPRE 工作在 PWM 模式 1，在向上计数时，O0CPREC 输出结果是 O0CPRE，向下计数时输出 O1CPRE。</p> <p>当 <code>TIMERx_CCHP</code> 寄存器的 <code>PROT[1:0]=11</code> 且 <code>CH0MS=000</code>（比较模式）时，此位不能被改变。</p>
3	CH0COMSEN	<p>通道 0 输出比较影子寄存器使能</p> <p>当此位被置 1，<code>TIMERx_CH0CV</code> 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。</p> <p>0: 禁止通道 0 输出/比较影子寄存器</p> <p>1: 使能通道 0 输出/比较影子寄存器</p> <p>仅在单脉冲模式下（<code>SPM=1</code>），可以在未确认影子寄存器的情况下使用 PWM 模式</p> <p>当 <code>TIMERx_CCHP</code> 寄存器的 <code>PROT[1:0]=11</code> 且 <code>CH0MS=00</code> 时此位不能被改变。</p>
2	CH0COMFEN	<p>通道 0 输出比较快速使能</p> <p>当该位为 1 时，如果通道配置为 PWM0 模式或者 PWM1 模式，会加快捕获/比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，<code>CH0_O</code> 被设置为比较电平而与比较结果无关。</p> <p>0: 禁止通道 0 输出比较快速。</p> <p>1: 使能通道 0 输出比较快速。</p>
1:0	CH0MS[1:0]	<p>通道 0 I/O 模式选择</p> <p>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（<code>TIMERx_CHCTL2</code> 寄存器的 <code>CH0EN</code> 位被清 0）时这些位才可写。</p>

- 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, 此模式仅工作在内部触发器输入被选中时(由 SYSCFG_TIMER2CFG)寄存器中的 TSCFGx[2:0] (x = 3,4,5,6,7)位域选择)。

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

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 <code>TIMERx_CHCTL2</code> 寄存器中的 <code>CH0EN</code> =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																								CH1COM CTL[3]			保留																								CH0COM CTL[3]																																																																																																																												
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			rw																																																																																																																																																							

输出比较模式:

位/位域	名称	描述
31:25	保留	必须保持复位值。
24	CH3COMCTL[3]	参见 CH3COMCTL[2:0]描述
23:17	保留	必须保持复位值。
16	CH2COMCTL[3]	参见 CH2COMCTL[2:0]描述
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, 此模式仅工作在内部触发器输入被选中时(由 SYSCFG_TIMER2CFG)寄存器中的 TSCFGx[2:0] (x = 3,4,5,6,7)位域选择)。
7	CH2COMCEN	<p>通道 2 输出比较清 0 使能</p> <p>当此位被置 1, 当检测到 ETIFP 输入高电平时, O2CPRE 参考信号被清 0</p> <p>0: 使能通道 2 输出比较清零</p> <p>1: 禁止通道 2 输出比较清零</p>
6:4	CH2COMCTL[2:0]	<p>通道 2 输出比较模式</p> <p>此位定义了 O2CPRE 的动作, 而 O2CPRE 决定了 CH2_O、CH2_ON 的值。O2CPRE 高电平有效, 而 CH2_O、CH2_ON 的有效电平取决于 CH2P、CH2NP 位。</p> <p>0000: 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</p> <p>0001: 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。</p> <p>0010: 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。</p> <p>0011: 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。</p> <p>0100: 强制为低。强制 O2CPRE 为低电平</p> <p>0101: 强制为高。强制 O2CPRE 为高电平</p> <p>0110: PWM 模式 0。在向上计数时, 一旦计数器值小于 TIMERx_CH2CV 时, O2CPRE 为有效电平, 否则为无效电平。在向下计数时, 一旦计数器的值大于 TIMERx_CH2CV 时, O2CPRE 为无效电平, 否则为有效电平。</p> <p>0111: PWM 模式 1。在向上计数时, 一旦计数器值小于 TIMERx_CH2CV 时, O2CPRE 为无效电平, 否则为有效电平。在向下计数时, 一旦计数器的值大于 TIMERx_CH2CV 时, O2CPRE 为有效电平, 否则为无效电平。</p> <p>在 PWM 模式 0 或 PWM 模式 1 中, 只有当比较结果改变了或者输出比较模式中从时基模式切换到 PWM 模式时, CxCOMR 电平才改变。</p> <p>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。</p> <p>1000:可再次触发单脉冲模式 0。 O2CPRE 工作在 PWM 模式 0, 向上计数时, O2CPRE 有效, 当外部触发信号产生时, O2CPRE 无效, 在下次更新事件产生后, O2CPRE 恢复有效。向下计数时, O2CPRE 无效, 当外部触发信号产生时, O2CPRE 有效, 在下次更新事件产生后, O2CPRE 恢复无效。</p> <p>1001: 可再次触发单脉冲模式 1。 O2CPRE 工作在 PWM 模式 1, 向上计数时, O2CPRE 无效, 当外部触发信号产生时, O2CPRE 有效, 在下次更新事件产生后, O2CPRE 恢复无效。向下计数时, O2CPRE 有效, 当外部触发信号产生时, O2CPRE 无效, 在下次更新事件产生后, O2CPRE 恢复有效。</p> <p>1010:保留</p> <p>1011:保留</p> <p>1100:复合 PWM0 模式。O2CPRE 工作在 PWM 模式 0, O2CPREC 输出结果是 O2CPRE 和 O3CPRE 的逻辑“或”。</p> <p>1101:复合 PWM1 模式。O2CPRE 工作在 PWM 模式 1, O2CPREC 输出结果是</p>

		O2CPRE 和 O3CPRE 的逻辑“与”。
		1110: 非对称 PWM0 模式。O2CPRE 工作在 PWM 模式 0, 在向上计数时, O2CPRE 输出结果是 O2CPRE, 向下计数时输出 O3CPRE。
		1111 非对称 PWM1 模式。O2CPRE 工作在 PWM 模式 1, 在向上计数时, O2CPRE 输出结果是 O2CPRE, 向下计数时输出 O3CPRE。
		当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 且 CH2MS =000 (比较模式) 时, 此位不能被改变。
3	CH2COMSEN	<p>通道 0 输出比较影子寄存器使能</p> <p>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。</p> <p>0: 禁止通道 2 输出/比较影子寄存器</p> <p>1: 使能通道 2 输出/比较影子寄存器</p> <p>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</p> <p>当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 且 CH2MS =00 时此位不能被改变。</p>
2	CH2COMFEN	<p>通道 2 输出比较快速使能</p> <p>当该位为 1 时, 如果通道配置为 PWM0 模式或者 PWM1 模式, 会加快捕获/比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配, CH2_O 被设置为比较电平而与比较结果无关。</p> <p>0: 禁止通道 2 输出比较快速。</p> <p>1: 使能通道 2 输出比较快速。</p>
1:0	CH2MS[1:0]	<p>通道 2 I/O 模式选择</p> <p>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。</p> <p>00: 通道 2 配置为输出</p> <p>01: 通道 2 配置为输入, IS2 映射在 CI2FE2 上</p> <p>10: 通道 2 配置为输入, IS2 映射在 CI3FE2 上</p> <p>11: 通道 2 配置为输入, IS2 映射在 ITS 上, 此模式仅工作在内部触发器输入被选中时 (由 SYSCFG_TIMER2CFG) 寄存器中的 TSCFGx[2:0] (x = 3,4,5,6,7) 位域选择)。</p>

输入捕获模式:

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:12	CH3CAPFLT[3:0]	通道 3 输入捕获滤波控制 参见 CH0CAPFLT 描述
11:10	CH3CAPPSC[1:0]	通道 3 输入捕获预分频器 参见 CH0CAPPSC 描述
9:8	CH3MS[1:0]	通道 3 模式选择 与输出模式相同
7:4	CH2CAPFLT[3:0]	通道 2 输入捕获滤波控制 CI2 输入信号可以通过数字滤波器进行滤波, 该位域配置滤波参数。

数字滤波器的基本原理：根据 f_{SAMP} 对 $CI2$ 输入信号进行连续采样，并记录信号相同电平的次数。达到该位配置的滤波参数后，认为是有效电平。

滤波器参数配置如下：

CH2CAPFLT [3:0]	采样次数	f_{SAMP}
4'b0000		无滤波器
4'b0001	2	f_{CK_TIMER}
4'b0010	4	
4'b0011	8	
4'b0100	6	$f_{DTS}/2$
4'b0101	8	
4'b0110	6	$f_{DTS}/4$
4'b0111	8	
4'b1000	6	$f_{DTS}/8$
4'b1001	8	
4'b1010	5	$f_{DTS}/16$
4'b1011	6	
4'b1100	8	
4'b1101	5	$f_{DTS}/32$
4'b1110	6	
4'b1111	8	

3:2 CH2CAPPSC[1:0] 通道 2 输入捕获预分频器
 这 2 位定义了通道 2 输入的预分频系数。当 $TIMERx_CHCTL2$ 寄存器中的 $CH2EN = 0$ 时，则预分频器复位。
 00：无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获
 01：每 2 个事件触发一次捕获
 10：每 4 个事件触发一次捕获
 11：每 8 个事件触发一次捕获

1:0 CH2MS[1:0] 通道 2 模式选择
 与输出比较模式相同

通道控制寄存器 2 (TIMERx_CHCTL2)

地址偏移：0x20
 复位值：0x0000 0000

该寄存器只能按字（32位）访问。



位/位域	名称	描述
31:16	保留	必须保持复位值。
15	CH3NP	通道 3 互补输出极性 参考 CH0NP 描述
14	保留	必须保持复位值。
13	CH3P	通道 3 极性 参考 CH0P 描述
12	CH3EN	通道 3 使能 参考 CH0EN 描述
11	CH2NP	通道 2 互补输出极性 参考 CH0NP 描述
10	保留	必须保持复位值。
9	CH2P	通道 2 极性 参考 CH0P 描述
8	CH2EN	通道 2 使能 参考 CH0EN 描述
7	CH1NP	通道 1 互补输出极性 参考 CH0NP 描述
6	保留	必须保持复位值。
5	CH1P	通道 1 极性 参考 CH0P 描述
4	CH1EN	通道 1 使能 参考 CH0EN 描述
3	CH0NP	通道 0 互补输出极性 当通道 0 配置为输出模式，该位保持 0。 当通道 0 配置为输入模式时，此位和 CH0P 联合使用，作为输入信号 CI0 的极性选择控制信号。 当 TIMERx_CCHP 寄存器的 PROT [1: 0]=11 或 10 时此位不能被更改。
2	保留	必须保持复位值。
1	CH0P	通道 0 极性 当通道 0 配置为输出模式时，此位定义了输出信号极性。 0: 通道0高电平为有效电平 1: 通道0低电平为有效电平 当通道 0 配置为输入模式时，此位定义了 CI0 信号极性 [CH0NP, CH0P] 将选择 CI0FE0 或者 CI1FE0 的有效边沿或者捕获极性 [CH0NP==0, CH0P==0]: 把 CixFE0 的上升沿作为捕获或者从模式下触发的有效信号，并且 CixFE0 不会被翻转。

[CH0NP==0, CH0P==1]: 把 CixFE0 的下降沿作为捕获或者从模式下触发的有效信号, 并且 CixFE0 会被翻转。

[CH0NP==1, CH0P==0]: 保留。

[CH0NP==1, CH0P==1]: 把 CixFE0 的上升沿和下降沿都作为捕获或者从模式下触发的有效信号, 并且 CixFE0 不会被翻转。

当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 或 10 时此位不能被更改。

0 **CH0EN** 通道 0 捕获/比较使能
 当通道 0 配置为输出模式时, 将此位置 1 使能 CH0_O 信号有效。当通道 0 配置为输入模式时, 将此位置 1 使能通道 0 上的捕获事件。
 0: 禁止通道 0
 1: 使能通道 0

计数器寄存器 (TIMERx_CNT)

地址偏移: 0x24

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问。



位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CNT[15:0]	这些位是当前的计数值。写操作能改变计数器值。

预分频寄存器 (TIMERx_PSC)

地址偏移: 0x28

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问。



位/位域	名称	描述
------	----	----

31:16	保留	必须保持复位值。
15:0	PSC[15:0]	计数器时钟预分频值 计数器时钟等于 <code>TIMER_CK</code> 时钟除以 <code>(PSC+1)</code> ，每当更新事件产生时，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。

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。



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)

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
31:16	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)

地址偏移: 0x3C

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问。



位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CH2VAL[15:0]	通道 2 的捕获或比较值 当通道 2 配置为输入模式时, 这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 2 配置为输出模式时, 这些位包含了即将和计数器比较的值。使能相应影子寄存器后, 影子寄存器值随每次更新事件更新。

通道 3 捕获/比较值寄存器 (TIMERx_CH3CV)

地址偏移: 0x40

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。 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: 无影响
0	保留	必须保持复位值。

14.3. 通用定时器 L2 (TIMERx, x= 13)

14.3.1. 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。

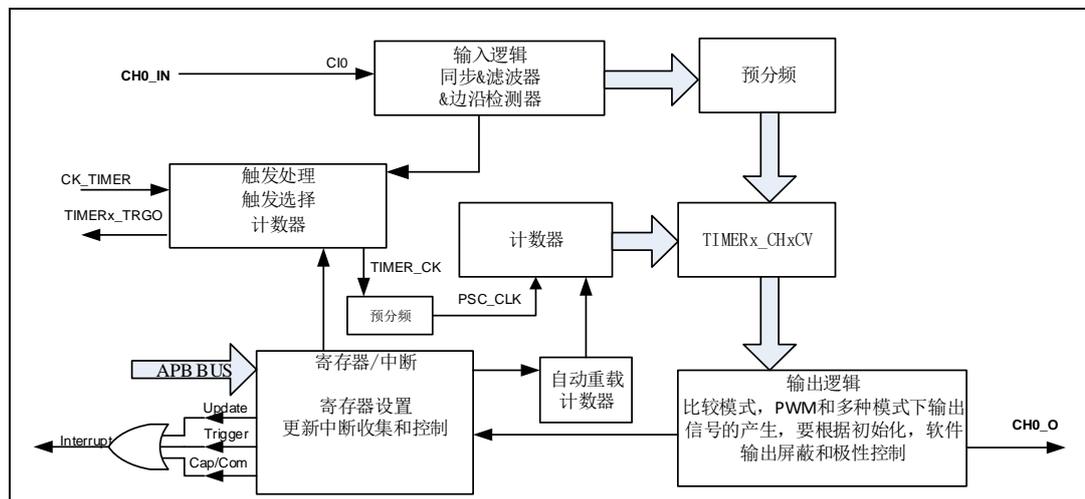
14.3.2. 主要特征

- 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；
- 自动重载功能；
- 中断输出：更新事件，比较/捕获事件。

14.3.3. 结构框图

[图14-61. 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 14-61. 通用定时器 L2 结构框图



14.3.4. 功能描述

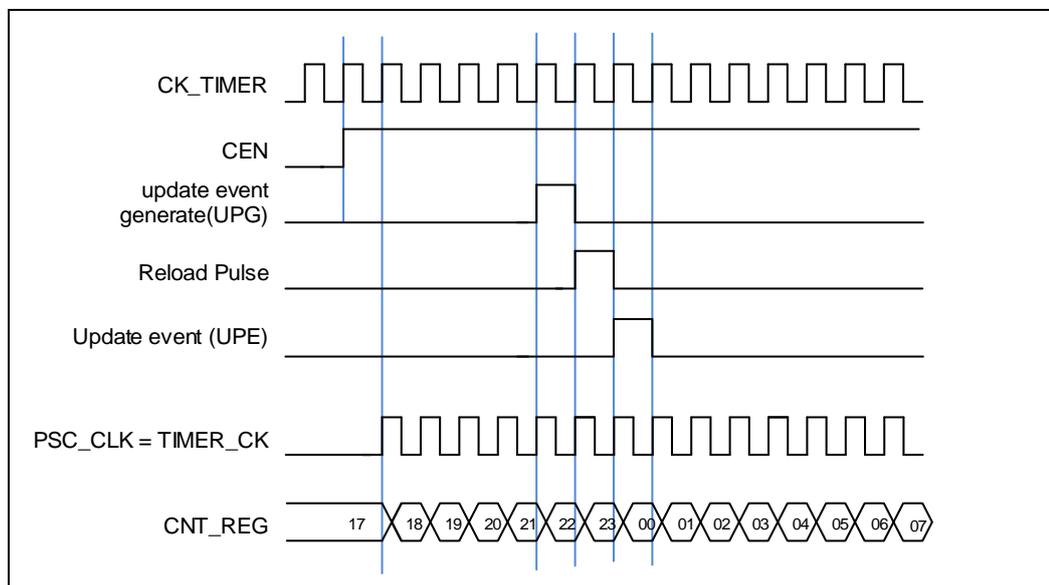
时钟源配置

通用定时器L2由内部时钟源CK_TIMER驱动

■ 定时器时钟TIMER_CK连接到RCU模块的CK_TIMER

通用定时器L2仅有一个时钟源CK_TIMER，用来驱动计数器预分频器。当CEN置位，CK_TIMER经过预分频器（预分频值由TIMERx_PSC寄存器确定）产生PSC_CLK。

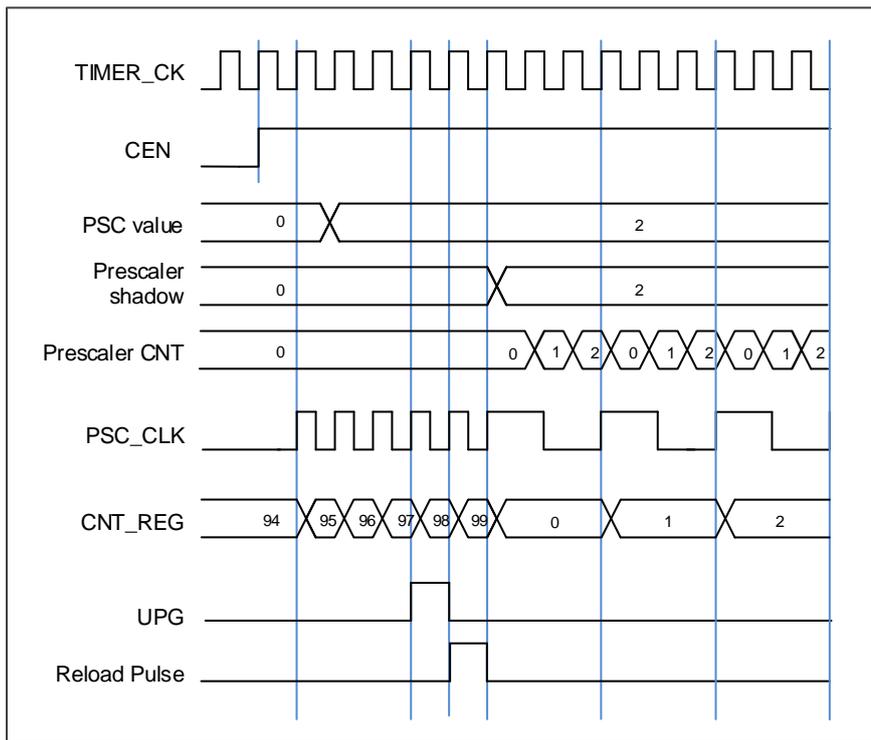
图 14-62. 内部时钟分频为 1 时，计数器的时序图



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 14-63. 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14-64. 向上计数时序图，PSC=0/2](#)和[图14-65. 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 14-64. 向上计数时序图，PSC=0/2

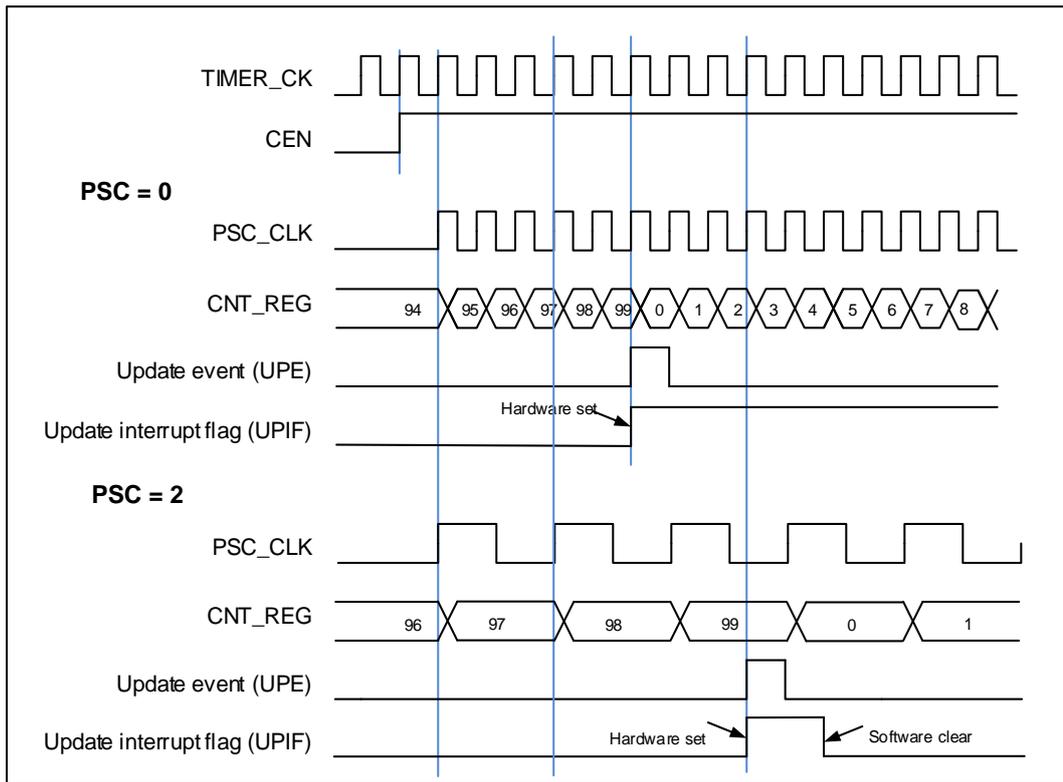
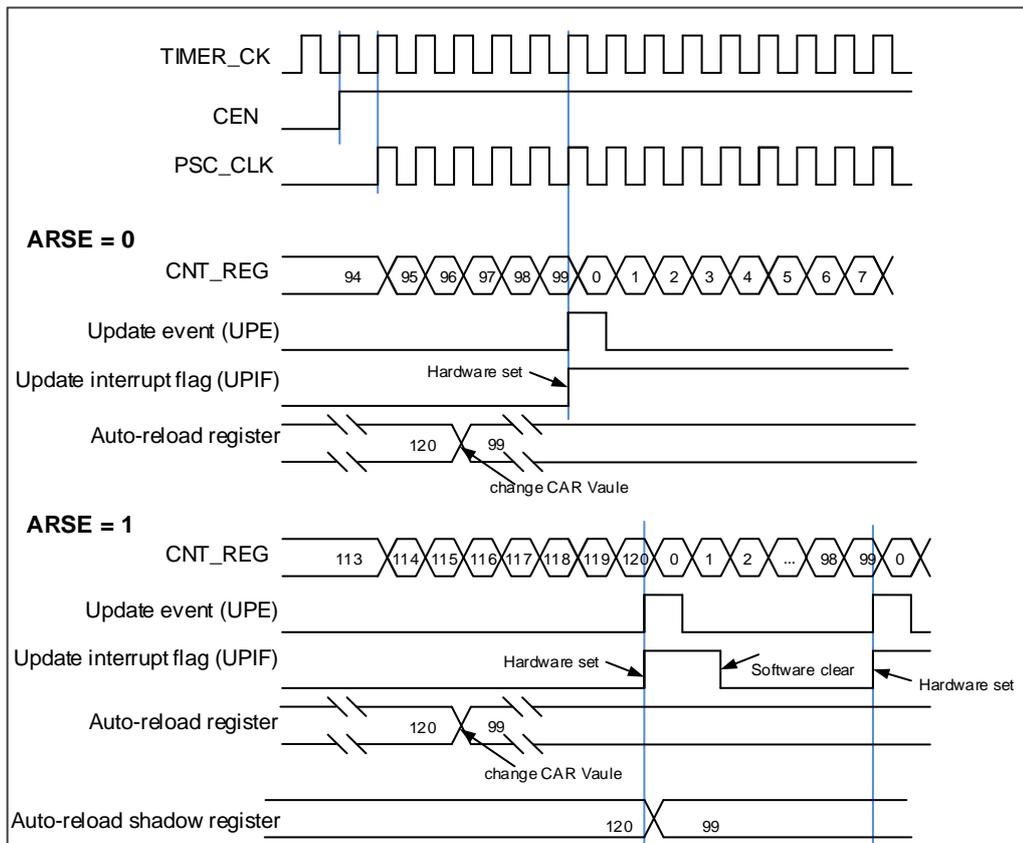


图 14-65. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值



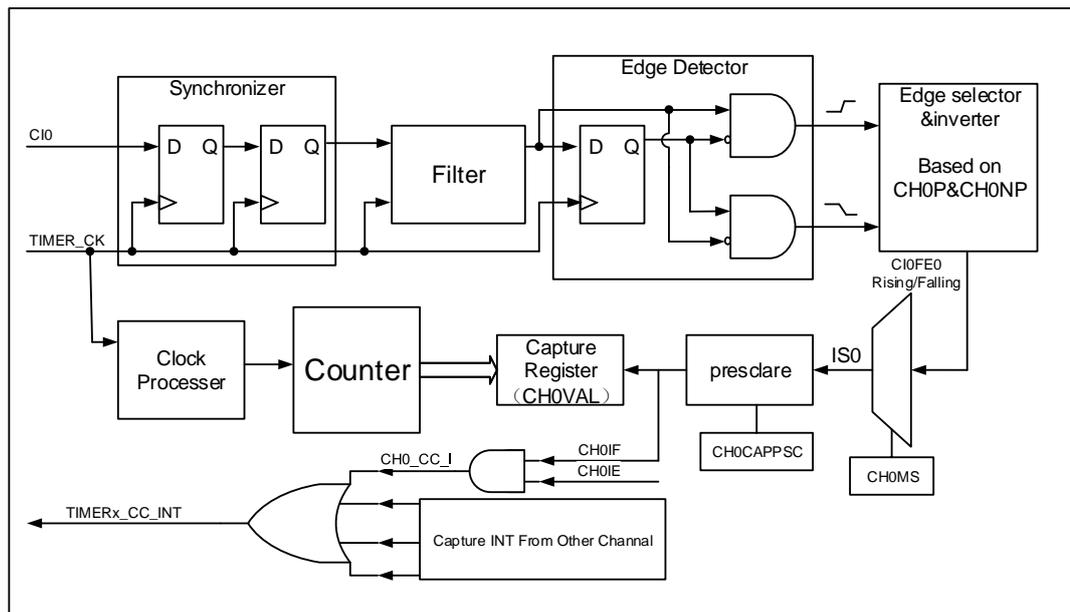
输入捕获和输出比较通道

通用定时器 L2 只有一个独立的通道用于捕获输入或比较输出是否匹配。该通道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

捕获模式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 14-66. 通道输入捕获原理



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

使能相应中断，可以获得中断。

第五步：捕获使能（TIMERx_CHCTL2 寄存器中 CHxEN）。

结果：当期望的输入信号发生时，TIMERx_CHxCV被设置成当前计数器的值，CHxIF为置1。如果CHxIF位已经为1，则CHxOF位置1。根据TIMERx_DMAINTEN寄存器中CHxIE的配置，相应的中断会被提出。

直接产生：软件设置CHxG位，会直接产生中断。

输入捕获模式也可用来测量 TIMERx_CHx 引脚上信号的脉冲波宽度。例如，一个 PWM 波连接到 CI0。配置 TIMERx_CHCTL0 寄存器中 CH0MS 为 2'b01，选择通道 0 的捕获信号为 CI0 并设置上升沿捕获。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

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，被置低电平或者反转。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/反转）；

设置CHxP/CHxNP位来选择有效电平的极性；

设置CHxEN使能输出。

第三步：通过CHxIE位配置中断使能。

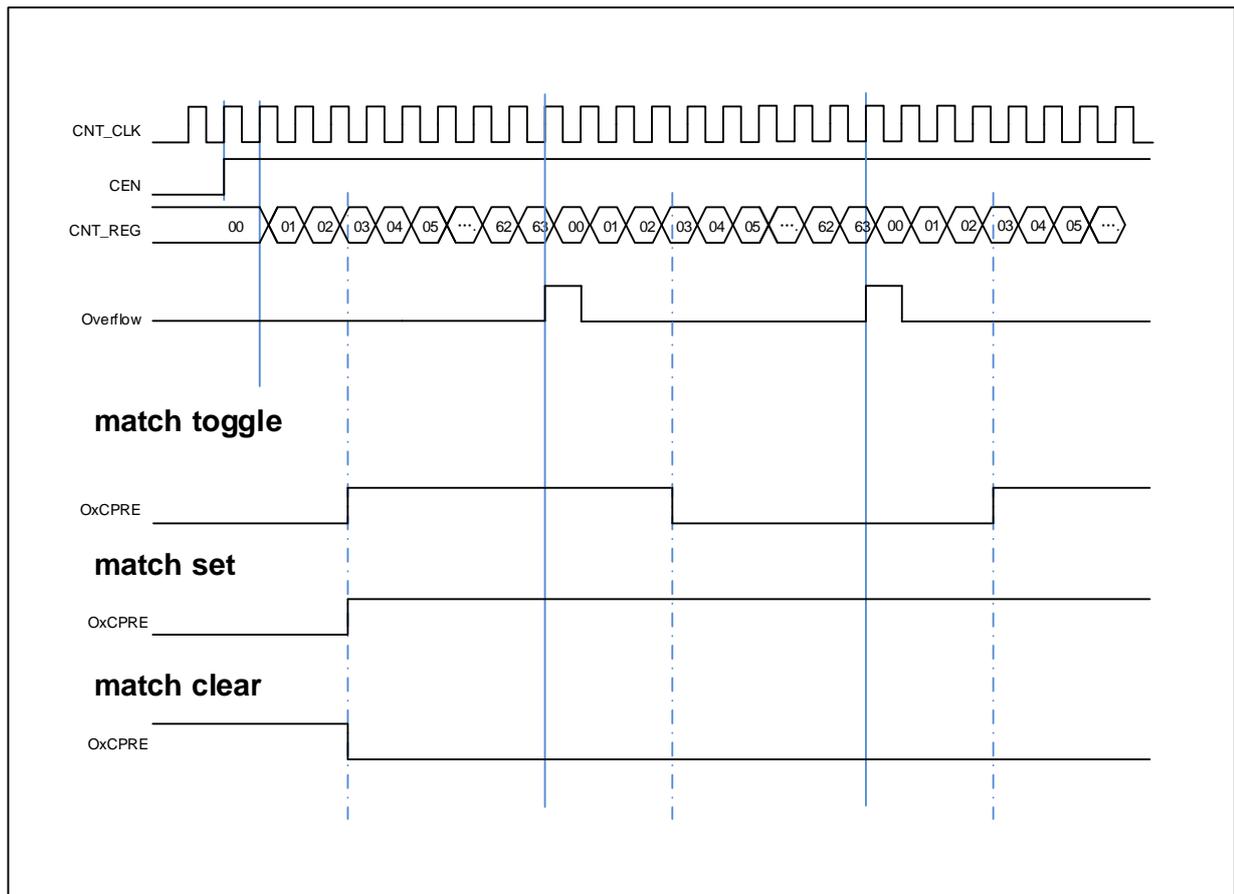
第四步：通过TIMERx_CAR寄存器和TIMERx_CHxCV寄存器配置输出比较时基：

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14-67. 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 14-67. 三种输出比较模式



通道输出参考信号

当 `TIMERx` 用于输出匹配比较模式下，设置 `CHxCOMCTL` 位可以定义 `OxCPRE` 信号(通道 `x` 准备信号)类型。`OxCPRE` 信号有若干类型的输出功能，包括，设置 `CHxCOMCTL=0x00` 可以保持原始电平；设置 `CHxCOMCTL=0x01` 可以将 `OxCPRE` 信号设置为高电平；设置 `CHxCOMCTL=0x02` 可以将 `OxCPRE` 信号设置为低电平；设置 `CHxCOMCTL=0x03`，在计数器值和 `TIMERx_CHxCV` 寄存器的值匹配时，可以翻转输出信号。

`PWM` 模式 0 和 `PWM` 模式 1 是 `OxCPRE` 的另一种输出类型，设置 `CHxCOMCTL` 位域位 `0x06` 或 `0x07` 可以配置 `PWM` 模式 0/`PWM` 模式 1。在这些模式中，根据计数器值和 `TIMERx_CHxCV` 寄存器值的关系以及计数方向，`OxCPRE` 信号改变其电平。具体细节描述，请参考相应的位。

设置 `CHxCOMCTL=0x04` 或 `0x05` 可以实现 `OxCPRE` 信号的强制输出功能。输出比较信号能够直接由软件强置为有效或无效状态，而不依赖于 `TIMERx_CHxCV` 的值和计数器值之间的比较结果。

单脉冲模式

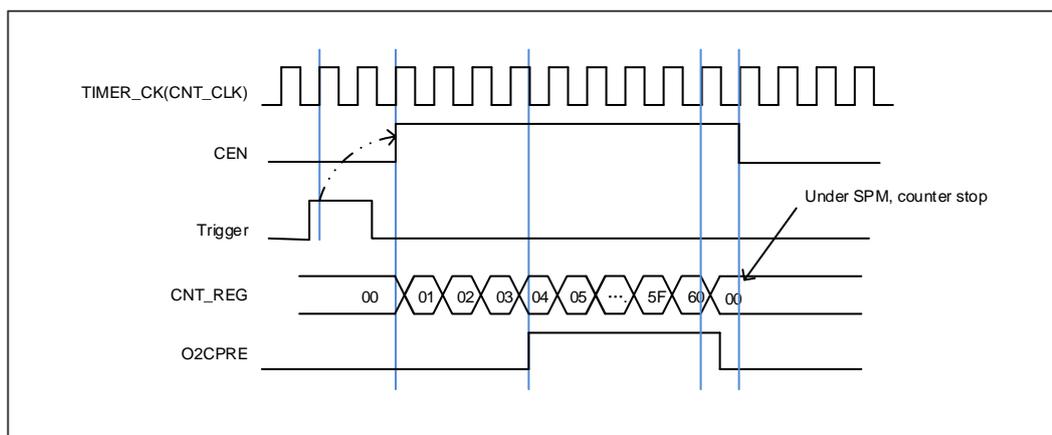
单脉冲模式与重复模式是相反的，设置 `TIMERx_CTL0` 寄存器的 `SPM` 位置 1，则使能单脉冲模式。当 `SPM` 置 1，计数器在下次更新事件到来后清零并停止计数。为了得到脉冲波，可以通过设置 `CHxCOMCTL` 配置 `TIMERx` 为 `PWM` 模式或者比较模式。

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。如果 CEN 值被硬件更新事件自动清 0，计数器将被再次初始化。

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 14-68. 单脉冲模式，TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60



定时器互连

参考 [定时器互连](#)

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。

14.3.5. TIMERx 寄存器(x = 13)

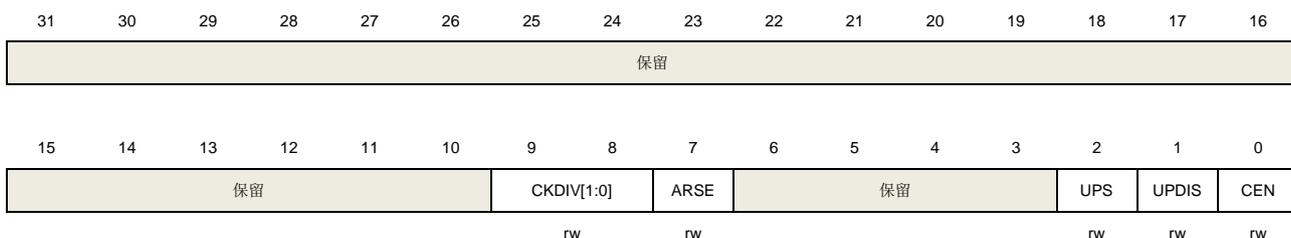
TIMER13基地址: 0x4000 2000

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	<p>计数器使能</p> <p>0: 计数器禁能</p> <p>1: 计数器使能</p> <p>在软件将CEN位置1后, 外部时钟、暂停模式和编码器模式才能工作。</p>
---	------------	---

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
31:2	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。



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。 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														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并且向上计数模式，计数器被清0，预分频计数器将同时被清除。

0: 无更新事件产生

1: 产生更新事件

通道控制寄存器 0 (TIMERx_CHCTL0)

地址偏移: 0x18

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								保留	CH0COMCTL[2:0]			CH0COM SEN	CH0COM FEN	CH0MS[1:0]	
									CH0CAPFLT[3:0]			CH0CAPPSC[1:0]			
									rw			rw		rw	

输出比较模式:

位/位域	名称	描述
31:7	保留	必须保持复位值
6:4	CH0COMCTL[2:0]	通道 0 输出比较模式 此位定义了输出准备信号 O0CPRE 的输出比较模式，而 O0CPRE 决定了 CH0_O、CH0_ON 的值。另外，O0CPRE 高电平有效，而 CH0_O、CH0_ON 通道的极性取决于 CH0P、CH0NP 位。 000: 时基。输出比较寄存器 TIMERx_CH0CV 与计数器 TIMERx_CNT 间的比较对 O0CPRE 不起作用 001: 匹配时设置为高。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 为高。 010: 匹配时设置为低。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 为低。 011: 匹配时翻转。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 翻转。 100: 强制为低。强制 O0CPRE 为低电平 101: 强制为高。强制 O0CPRE 为高电平 110: PWM 模式 0。在向上计数时，一旦计数器值小于 TIMERx_CH0CV 时，O0CPRE 为高电平，否则为低电平。在向下计数时，一旦计数器的值大于 TIMERx_CH0CV 时，O0CPRE 为低电平，否则为高电平。 111: PWM 模式 1。在向上计数时，一旦计数器值小于 TIMERx_CH0CV 时，O0CPRE

为低电平，否则为高电平。在向下计数时，一旦计数器的值大于 `TIMERx_CH0CV` 时，`O0CPRE` 为高电平，否则为低电平。

如果配置在 `PWM` 模式下，只有当输出比较模式从时基模式变为 `PWM` 模式或者比较结果改变时，`O0CPRE` 电平才改变。

3	<code>CH0COMSEN</code>	<p>通道0输出比较影子寄存器使能</p> <p>当此位被置1，<code>TIMERx_CH0CV</code>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。</p> <p>0: 禁止通道0输出/比较影子寄存器</p> <p>1: 使能通道0输出/比较影子寄存器</p> <p>仅在单脉冲模式下(<code>TIMERx_CTL0</code>寄存器的<code>SPM=1</code>)，可以在未确认预装载寄存器情况下使用<code>PWM</code> 模式。</p>
2	<code>CH0COMFEN</code>	<p>通道0输出比较快速使能</p> <p>当该位为1时，如果通道配置为<code>PWM</code>模式0或者<code>PWM</code>模式1，会加快捕获/比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，<code>CH0_O</code>被设置为比较电平而与比较结果无关。</p> <p>0: 禁止通道0输出比较快速</p> <p>1: 使能通道0输出比较快速</p>
1:0	<code>CH0MS[1:0]</code>	<p>通道0 I/O模式选择</p> <p>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(<code>TIMERx_CHCTL2</code>寄存器的<code>CH0EN</code>位被清0)时这些位才可写。</p> <p>00: 通道0配置为输出</p> <p>01: 通道0配置为输入，ISO映射在<code>CI0FE0</code>上</p> <p>Others: 保留</p>

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	<code>CH0CAPFLT[3:0]</code>	<p>通道 0 输入捕获滤波控制</p> <p><code>CI0</code> 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。</p> <p>数字滤波器的基本原理：根据 f_{SAMP} 对 <code>CI0</code> 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。</p> <p>滤波器参数配置如下：</p>

<code>CH0CAPFLT [3:0]</code>	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	

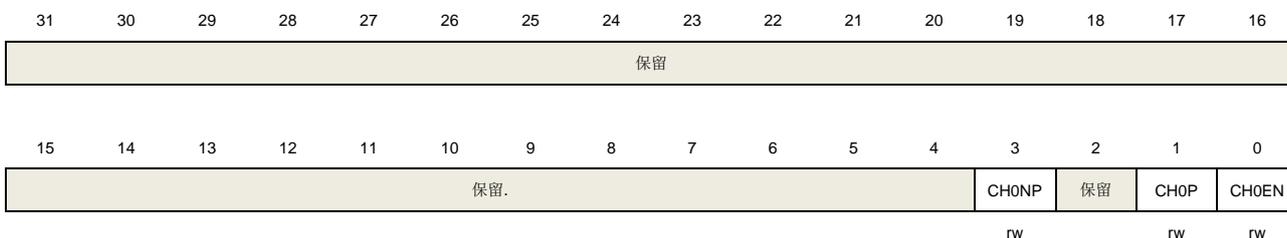
- 3:2** **CH0CAPPSC[1:0]** 通道0输入捕获预分频器
- 这2位定义了通道0输入的预分频系数。当TIMERx_CHCTL2寄存器中的CH0EN =0时，则预分频器复位。
- 00: 无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获
- 01: 每2个事件触发一次捕获
- 10: 每4个事件触发一次捕获
- 11: 每8个事件触发一次捕获
-
- 1:0** **CH0MS[1:0]** 通道0模式选择
- 与输出比较模式相同

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高电平有效 1: 通道0低电平有效 当通道0配置为输入模式时，此位和CH0P联合使用，作为输入信号CI0的极性选择控制信号。
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高电平有效

1: 通道0低电平有效

当通道0配置为输入模式时，此位定义了CI0信号极性

[CH0NP, CH0P] 将选择CI0FE0或者CI1FE0的有效边沿或者捕获极性

[CH0NP=0, CH0P=0]: 把CIxFE0的上升沿作为捕获或者从模式下触发的有效信号，并且CIxFE0不会被翻转。

[CH0NP=0, CH0P=1]: 把CIxFE0的下降沿作为捕获或者从模式下触发的有效信号，并且CIxFE0会被翻转。

[CH0NP=1, CH0P=0]: 保留。

[CH0NP=1, CH0P=1]: 把CIxFE0的上升沿和下降沿都作为捕获或者从模式下触发的有效信号，并且CIxFE0不会被翻转。

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。



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。



PSC[15:0]

rw

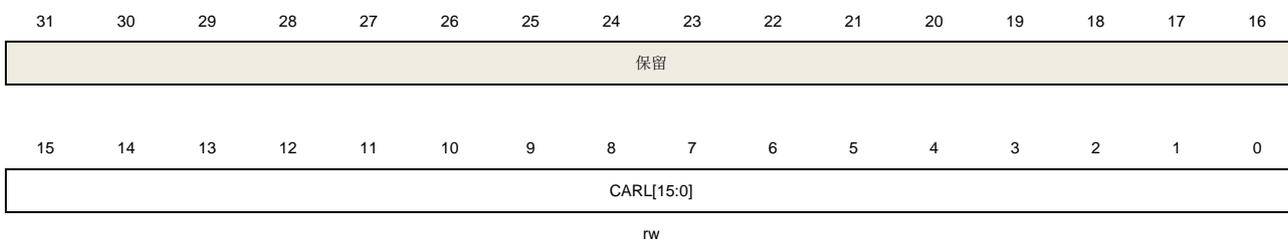
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	PSC[15:0]	计数器时钟预分频值 计数器时钟等于TIMER_CK时钟除以(PSC+1)，每次当更新事件产生时，PSC 的值被装入到对应的影子寄存器。

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。



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。



位/位域	名称	描述
31:16	保留	必须保持复位值

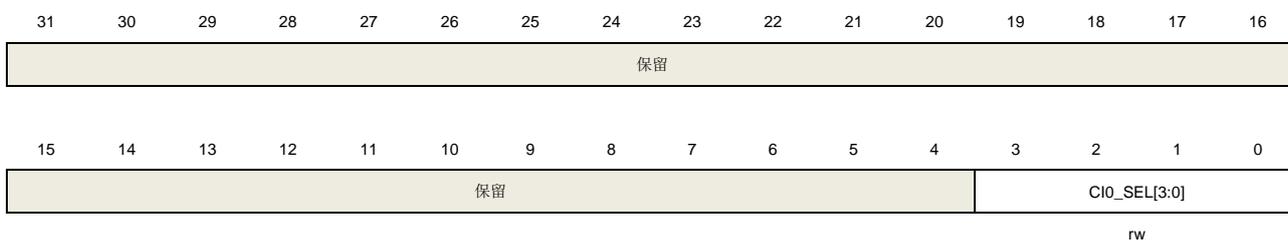
- 15:0 CH0VAL[15:0] 通道0的捕获或比较值
 当通道0配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。
 当通道0配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

输入选择寄存器 (TIMERx_INSEL)

地址偏移: 0x68

复位值: 0x0000 0000

该寄存器只能按字(32位)访问



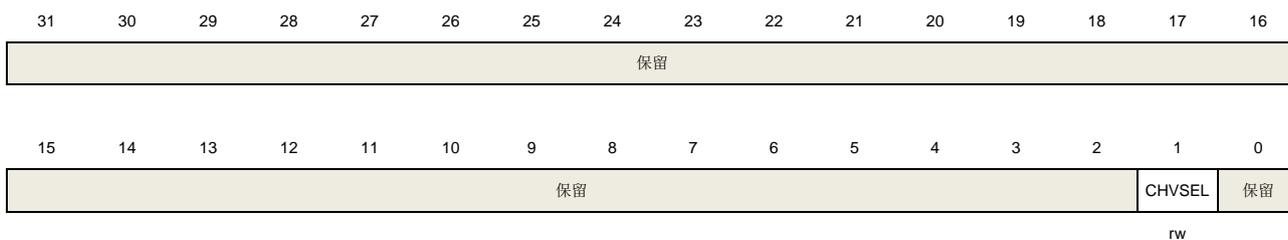
位/位域	名称	描述
31:4	保留	必须保持复位值。
3:0	CI0_SEL[3:0]	TIMER0_CH0输入选择 0000: 通道0输入连接到GPIO(TIMER13_CH0) 0001: 通道0输入连接到RTCCLK 0010: 通道0输入连接到HXTAL/32 0011: 通道0输入连接到CKOUTSEL0 0100: 通道0输入连接到CKOUTSEL1 其他: 保留

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。 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: 无影响
0	保留	必须保持复位值

14.4. 通用定时器 L4 (TIMERx,x=15,16)

14.4.1. 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。

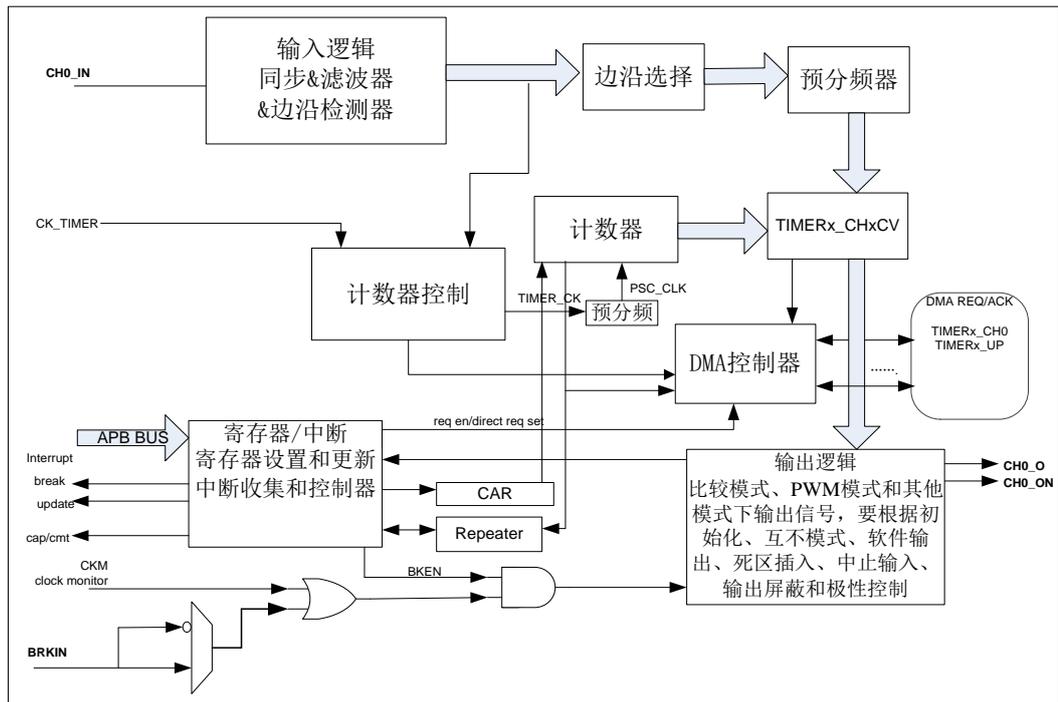
14.4.2. 主要特性

- 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；

14.4.3. 结构框图

[图14-69. 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 14-69. 通用定时器 L4 结构框图



14.4.4. 功能描述

时钟源选择

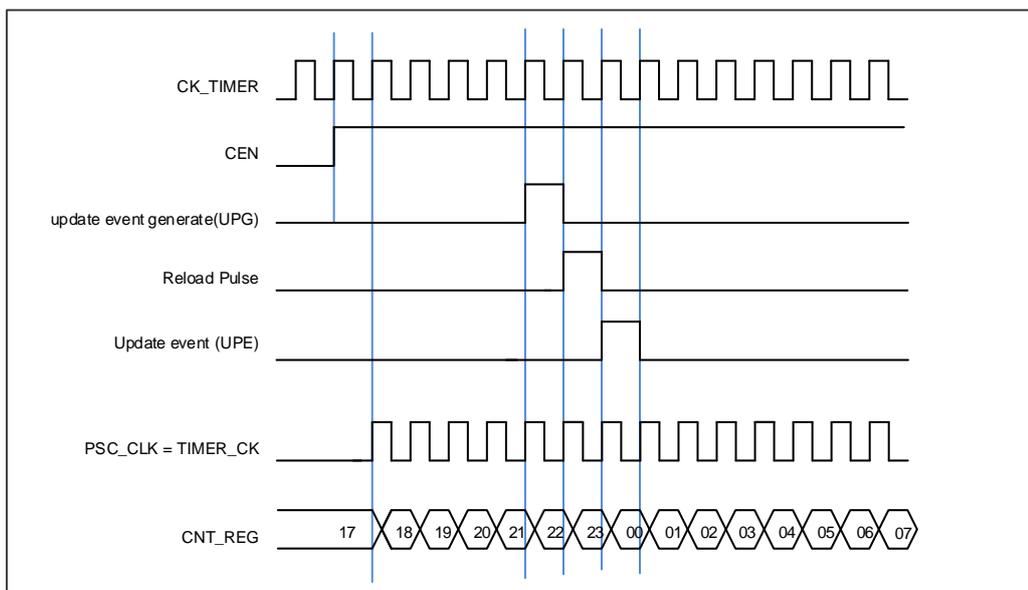
通用定时器 L4 由内部时钟源 TIMER_CK.

- 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

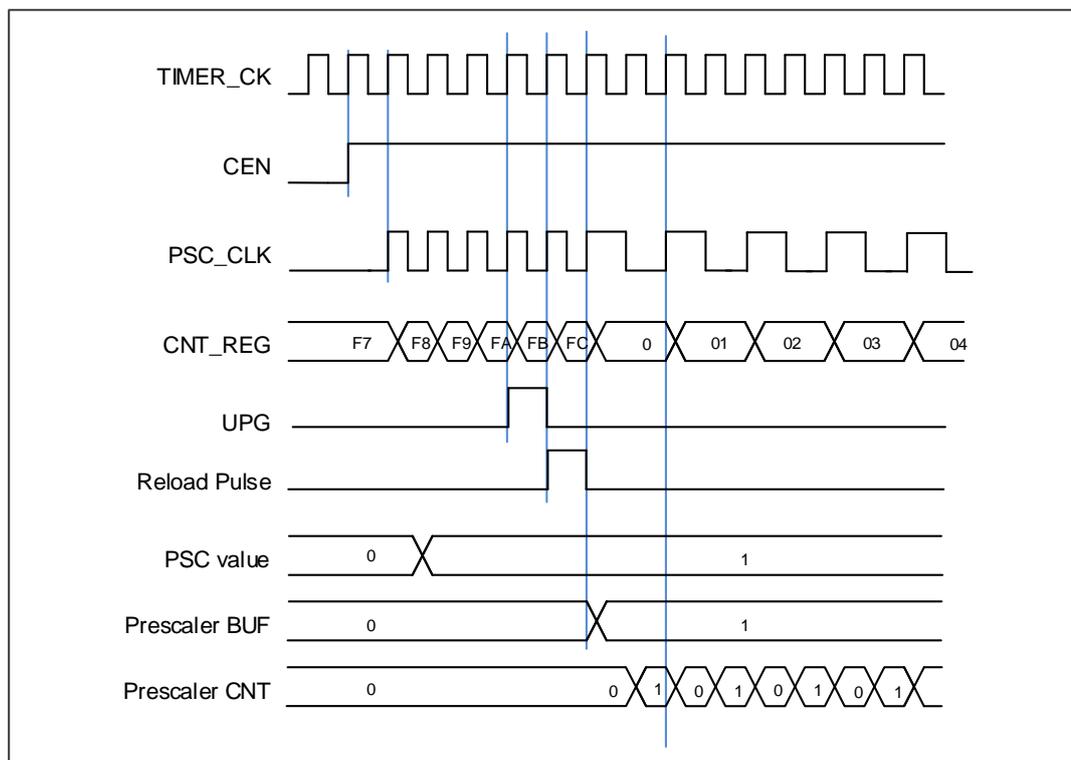
图 14-70. 内部时钟分频为 1 时正常模式下的控制电路



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次更新事件到来时被采用。

图 14-71. 当预分频器的参数从 1 变到 2 时, 计数器的时序图



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。如果设置了重复计数器，在 $(\text{TIMERx_CREP}+1)$ 次上溢后产生更新事件，否则在每次上溢时都会产生更新事件。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 14-72. 向上计数时序图, PSC=0/1](#) 给出了一些例子，当 `TIMERx_CAR=0x63` 时，计数器在不同预分频因子下的行为。

图 14-72. 向上计数时序图, PSC=0/1

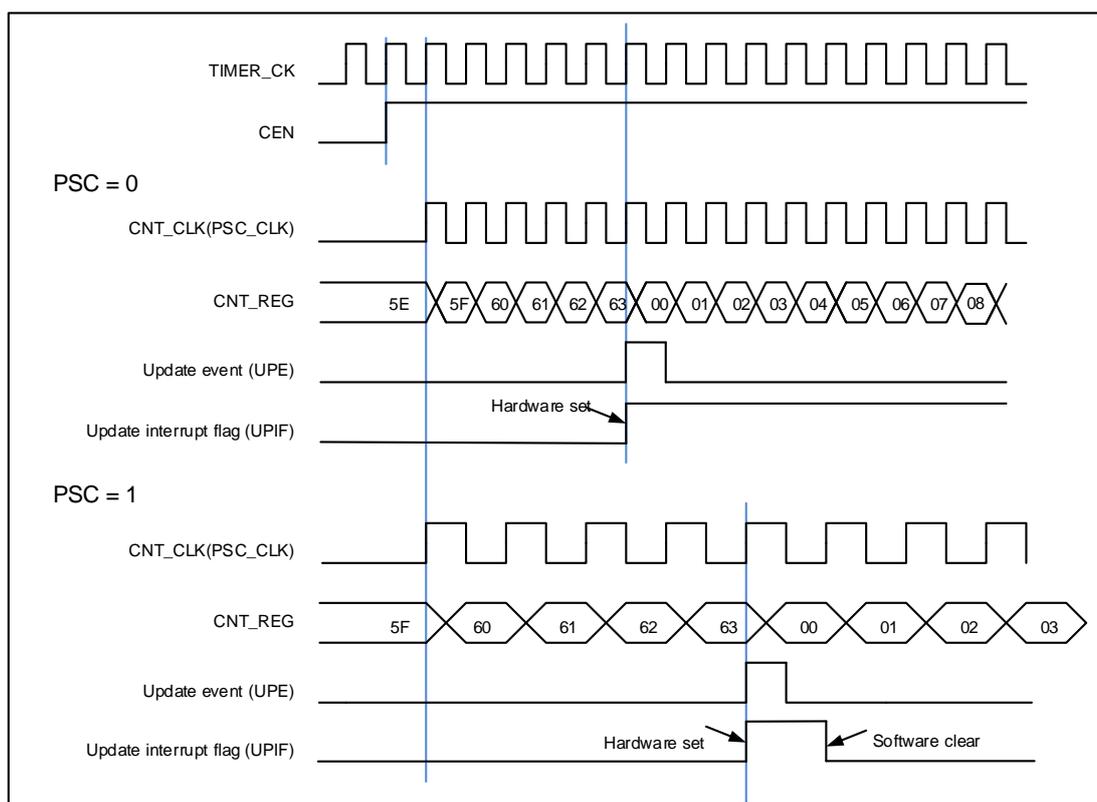
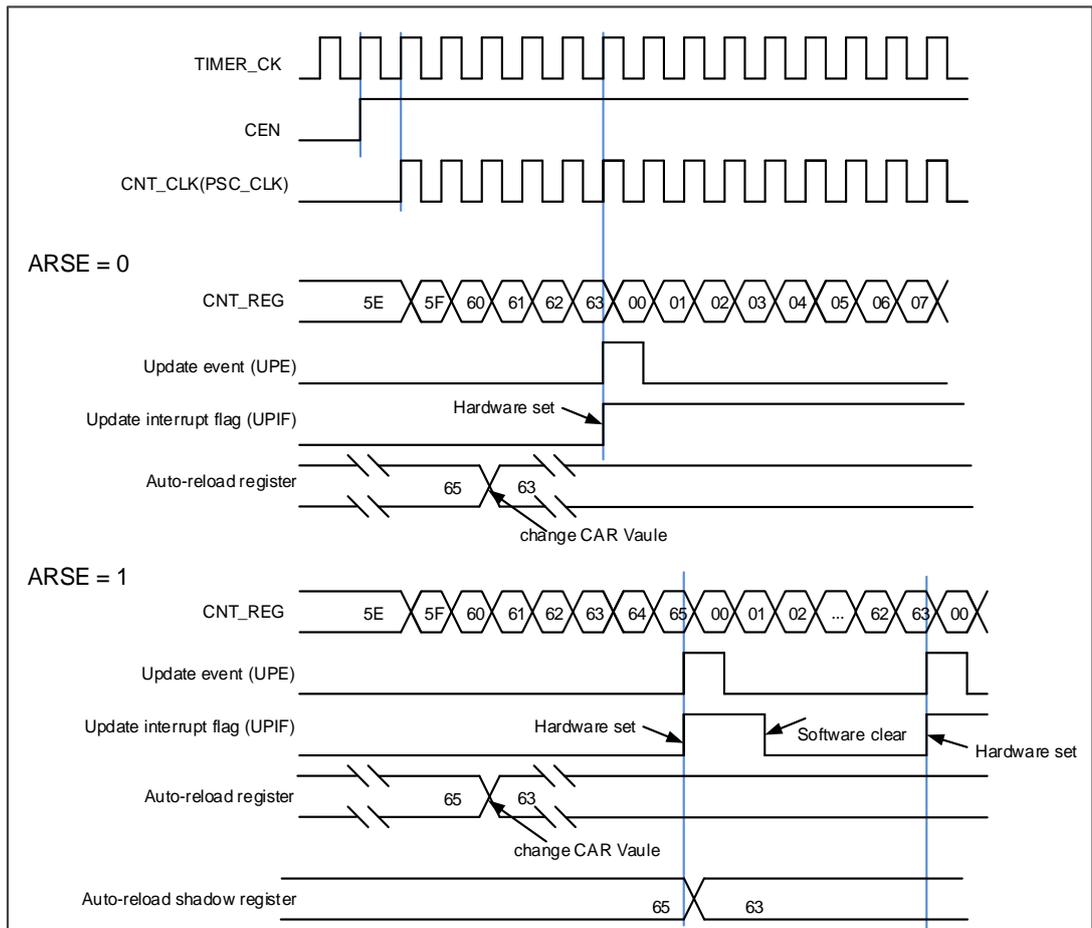


图 14-73. 向上计数时序图，在运行时改变 `TIMERx_CAR` 寄存器的值

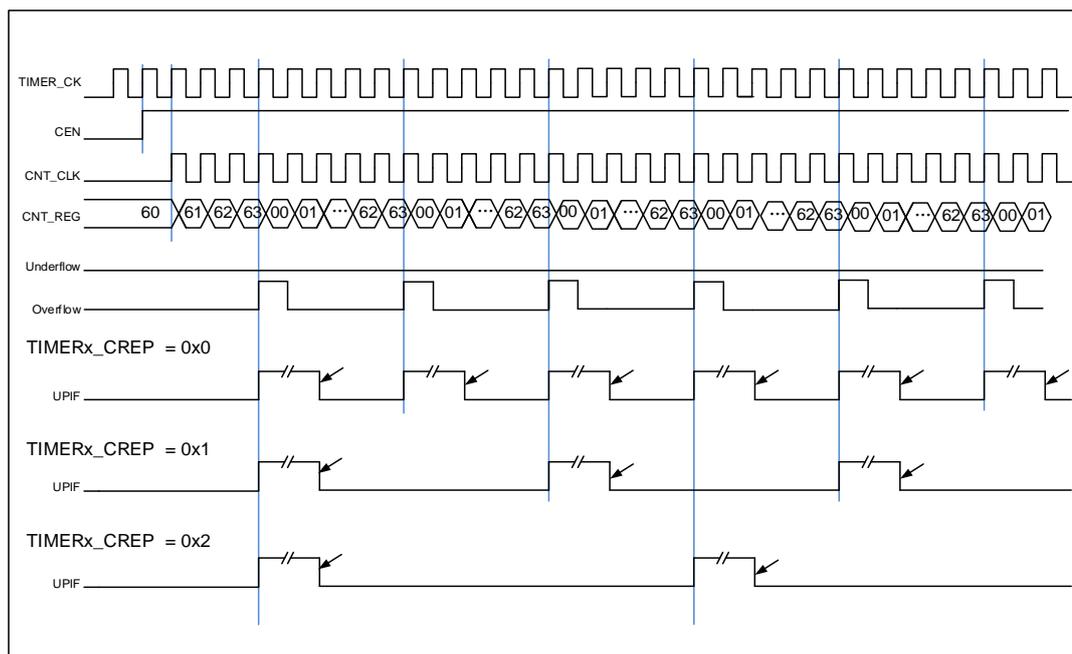


重复计数器

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 14-74. 在向上计数模式下计数器重复时序图



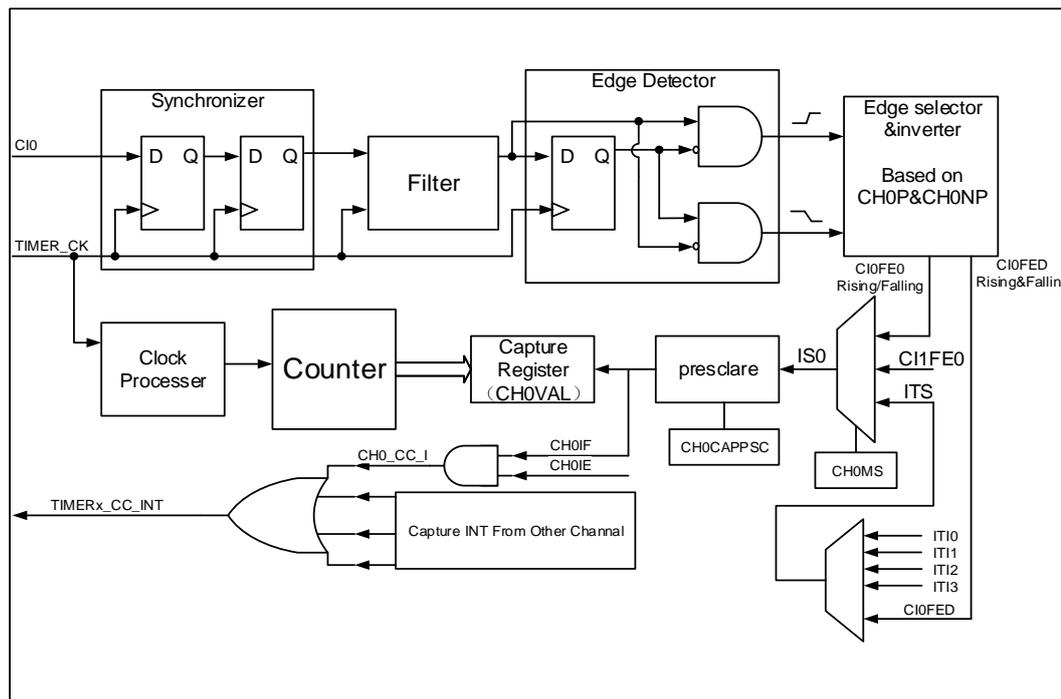
捕获/比较通道

通用定时器 L4 拥有一个独立的通道用于捕获输入或比较输出是否匹配。每个通道都围绕一个通道捕获比较寄存器建立，包括一个输入级，通道控制器和输出级。

■ 输入捕获模式

捕获模式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 14-75. 输入捕获逻辑



通道输入信号 Cix 有两种选择，一种是 $TIMERx_CHx$ 信号，另一种是 $TIMERx_CH0, TIMERx_CH1$ 和 $TIMERx_CH2$ 异或之后的信号。通道输入信号 Cix 先被 $TIMER_CK$ 信号同步，然后经过数字滤波器采样，产生一个被滤波后的信号。通过边沿检测器，可以选择检测上升沿或者下降沿。通过配置 $CHxP$ 选择使用上升沿或者下降沿。配置 $CHxMS.$ ，可以选择其他通道的输入信号，内部触发信号。配置 IC 预分频器，使得若干个输入事件后才产生一个有效的捕获事件。捕获事件发生， $CHxVAL$ 存储计数器的值。

配置步骤如下：

第一步： 滤波器配置（ $TIMERx_CHCTL0$ 寄存器中 $CHxCAPFLT$ ）：

根据输入信号和请求信号的质量，配置相应的 $CHxCAPFLT$ 。

第二步： 边沿选择（ $TIMERx_CHCTL2$ 寄存器中 $CHxP/CHxNP$ ）：

配置 $CHxP/CHxNP$ 选择上升沿或者下降沿。

第三步： 捕获源选择（ $TIMERx_CHCTL0$ 寄存器中 $CHxMS$ ）：

一旦通过配置 $CHxMS$ 选择输入捕获源，必须确保通道配置在输入模式（ $CHxMS!=0x0$ ），而且 $TIMERx_CHxCV$ 寄存器不能再被写。

第四步： 中断使能（ $TIMERx_DMAINTEN$ 寄存器中 $CHxIE$ 和 $CHxDEN$ ）：

使能相应中断，可以获得中断和DMA请求。

第五步： 捕获使能（ $TIMERx_CHCTL2$ 寄存器中 $CHxEN$ ）。

结果： 当期望的输入信号发生时， $TIMERx_CHxCV$ 被设置成当前计数器的值， $CHxIF$ 为置1。

如果 $CHxIF$ 位已经为1，则 $CHxOF$ 位置1。根据 $TIMERx_DMAINTEN$ 寄存器中 $CHxIE$ 和 $CHxDEN$ 的配置，相应的中断和DMA请求会被提出。

直接产生： 软件设置 $CHxG$ 位，会直接产生中断和DMA请求。

■ 输出比较模式

在输出比较模式， $TIMERx$ 可以产生时控脉冲，其位置，极性，持续时间和频率都是可编程的。

当一个输出通道的 CHxVAL 寄存器与计数器的值匹配时，根据 CHxCOMCTL 的配置，这个通道的输出可以被置高电平，被置低电平或者反转。当计数器的值与 CHxVAL 寄存器的值匹配时，CHxIF 位被置 1，如果 CHxIE = 1 则会产生中断，如果 CHxDEN=1 则会产生 DMA 请求。

配置步骤如下：

第一步：时钟配置：

配置定时器时钟源，预分频器等。

第二步：比较模式配置：

设置 CHxCOMSEN 位来配置输出比较影子寄存器；

设置 CHxCOMCTL 位来配置输出模式（置高电平/置低电平/反转）；

设置 CHxP/CHxNP 位来选择有效电平的极性；

设置 CHxEN 使能输出。

第三步：通过 CHxIE/CHxDEN 位配置中断/DMA 请求使能。

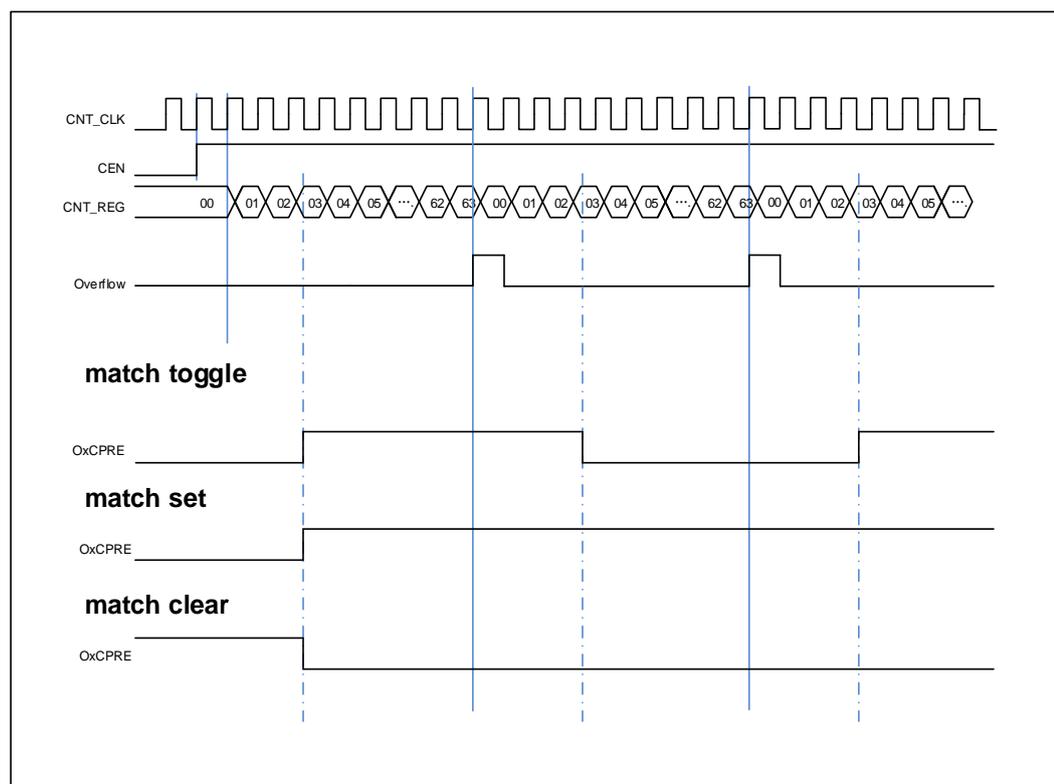
第四步：通过 TIMERx_CAR 寄存器和 TIMERx_CHxCV 寄存器配置输出比较时基：

CHxVAL 可以在运行时根据你所期望的波形而改变。

第五步：设置 CEN 位使能定时器。

图14-76. 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 14-76. 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)。

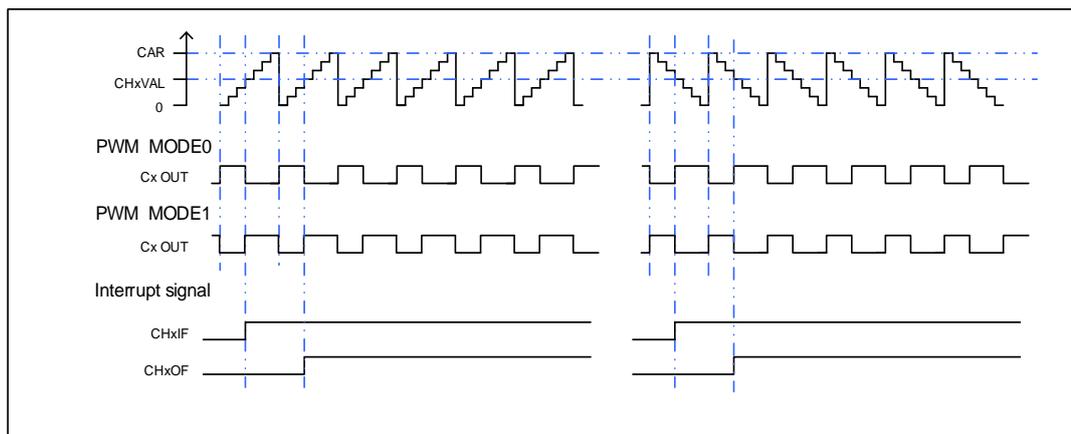
EAPWM 的周期由 `TIMERx_CAR` 寄存器值决定，占空比由 `TIMERx_CHxCV` 寄存器值决定。

[图 14-77. 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 14-77. PWM 时序图



通道输出参考信号

当 `TIMERx` 用于输出匹配比较模式下，设置 `CHxCOMCTL` 位可以定义 `OxCPRE` 信号(通道 `x` 准备信号)类型。`OxCPRE` 信号有若干类型的输出功能，包括，设置 `CHxCOMCTL=0x00` 可以保持原始电平；设置 `CHxCOMCTL=0x01` 可以将 `OxCPRE` 信号设置为高电平；设置 `CHxCOMCTL=0x02` 可以将 `OxCPRE` 信号设置为低电平；设置 `CHxCOMCTL=0x03`，在计数器值和 `TIMERx_CHxCV` 寄存器的值匹配时，可以翻转输出信号。

PWM 模式 0 和 PWM 模式 1 是 `OxCPRE` 的另一种输出类型，设置 `CHxCOMCTL` 位域位 `0x06` 或 `0x07` 可以配置 PWM 模式 0/PWM 模式 1。在这些模式中，根据计数器值和 `TIMERx_CHxCV` 寄存器值的关系以及计数方向，`OxCPRE` 信号改变其电平。具体细节描述，请参考相应的位。

设置 `CHxCOMCTL=0x04` 或 `0x05` 可以实现 `OxCPRE` 信号的强制输出功能。输出比较信号能够直接由软件强置为有效或无效状态，而不依赖于 `TIMERx_CHxCV` 的值和计数器值之间的比较结果。

互补输出

`CHx_O` 和 `CHx_ON` 是一对互补输出通道，这两个信号不能同时有效。`TIMERx` 有两路通道，只有一路有互补输出通道。互补信号 `CHx_O` 和 `CHx_ON` 是由一组参数来决定：`TIMERx_CHCTL2` 寄存器中的 `CHxEN` 和 `CHxNEN` 位，`TIMERx_CCHP` 寄存器中和 `TIMERx_CTL1` 寄存器中的 `POEN`, `ROS`, `IOS`, `ISOx` 和 `ISOxN` 位。输出极性由 `TIMERx_CHCTL2` 寄存器中的 `CHxP` 和 `CHxNP` 位来决定。

表 14-7. 由参数控制的互补输出表

互补参数					输出状态	
POEN	ROS	IOS	CHxEN	CHxNEN	CHx_O	CHx_ON
0	0/1	0	0	0	CHx_O / CHx_ON = LOW CHx_O / CHx_ON 输出禁用.	
				1	CHx_O = CHxP CHx_ON = CHxNP	
			1	0	CHx_O/CHx_ON 输出禁用. 如果时钟使能: CHx_O = ISOx CHx_ON = ISOxN	
				1	CHx_O = CHxP CHx_ON = CHxNP CHx_O/CHx_ON输出禁用.	
		1	0	0	CHx_O = CHxP CHx_ON = CHxNP CHx_O/CHx_ON输出禁用.	
				1	CHx_O = CHxP CHx_ON = CHxNP	
			1	0	CHx_O/CHx_ON 输出使能. 如果时钟使能: CHx_O = ISOx CHx_ON = ISOxN	
				1	CHx_O = CHxP CHx_ON = CHxNP	
1	0	0	0	CHx_O/CHx_ON = LOW CHx_O/CHx_ON 输出禁用.		
			1	CHx_O = LOW CHx_O 输出禁用.	CHx_ON=OxCPRE ⊕ CHxNP CHx_ON 输出使能	
		1	0	CHx_O=OxCPRE ⊕ CHxP CHx_O输出使能	CHx_ON = LOW CHx_ON输出禁用.	
			1	CHx_O=OxCPRE ⊕ CHxP CHx_O输出使能	CHx_ON=(!OxCPRE) ⊕ CHxNP CHx_ON输出使能	
		0/1	0	0	CHx_O = CHxP CHx_O输出禁用.	CHx_ON = CHxNP CHx_ON输出禁用.
				1	CHx_O = CHxP CHx_O输出使能	CHx_ON=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输出使能	
			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。

死区时间插入

设置 CHxEN 和 CHxNEN 为 1'b1 同时设置 POEN, 死区插入就会被使能。DTCFG 位域定义了死区时间, 死区时间对通道 0 有效。死区时间的细节, 请参考 TIMERx_CCHP 寄存器。

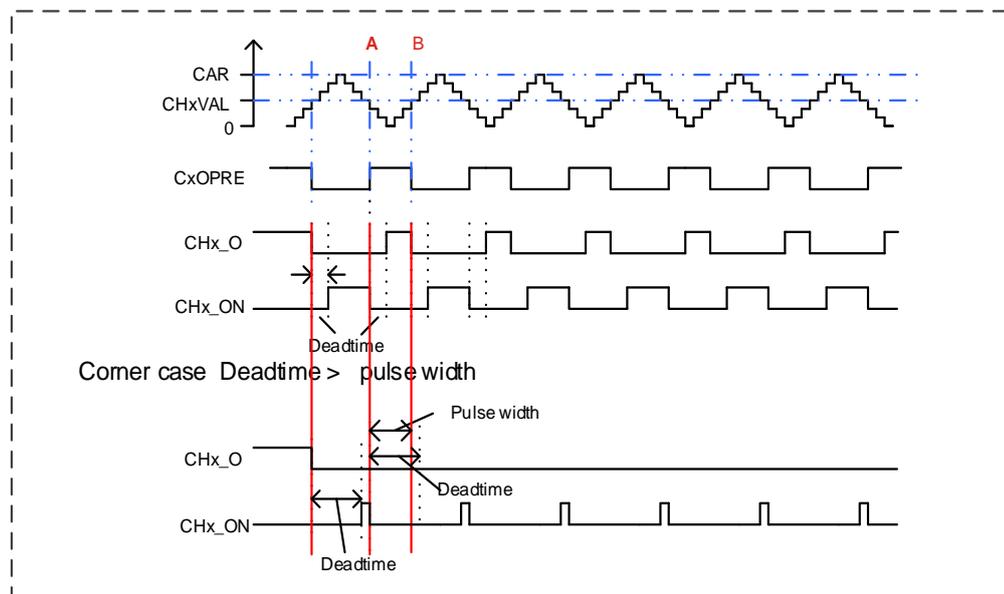
死区时间的插入, 确保了通道互补的两路信号不会同时有效。

在 PWM0 模式, 当通道 x 匹配发生时 (TIMERx 计数器= CHxVAL), OxCPRE 反转。在 [图 14-78. 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 14-78. 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 14-78. 带死区时间的互补输出



中止功能

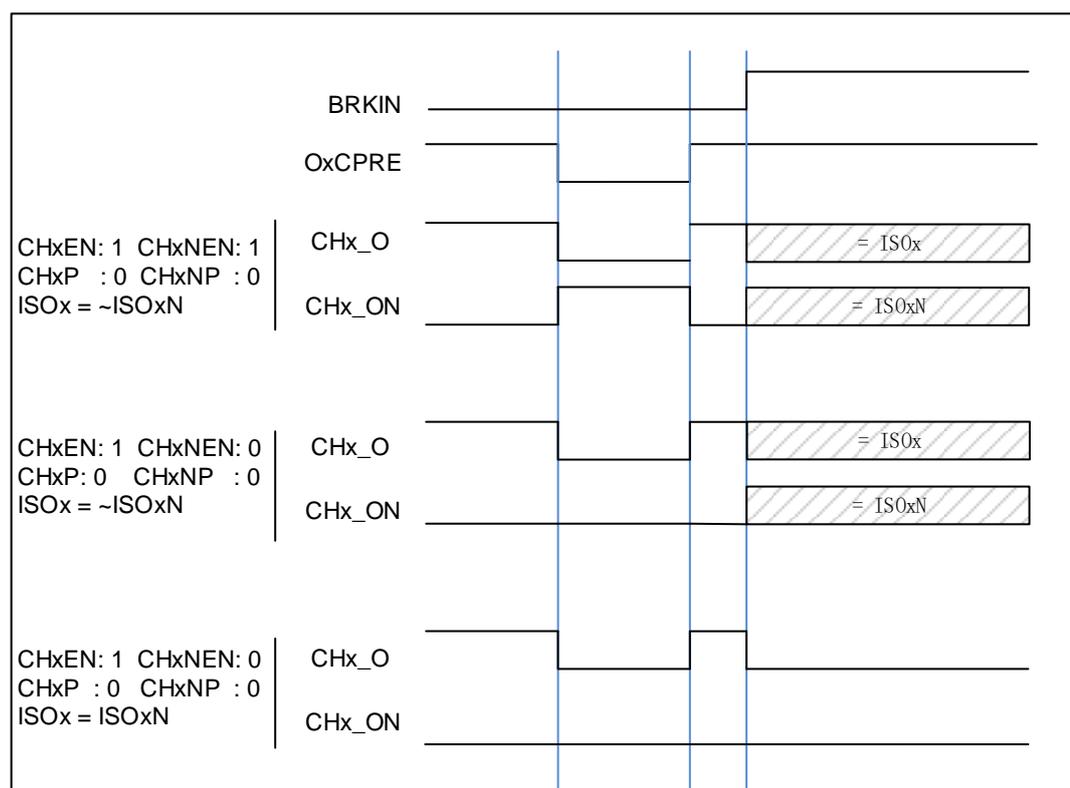
使用中止功能时, 输出 CHx_O 和 CHx_ON 信号电平被以下位控制, TIMERx_CCHP 寄存器的 POEN, IOS 和 ROS 位, TIMERx_CTL1 寄存器的 ISOx 和 ISOxN 位。当中止事件发生时, CHx_O 和 CHx_ON 信号输出不能同时设置为有效电平。中止源可以选择中止输入引脚, 也可

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 14-79. 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

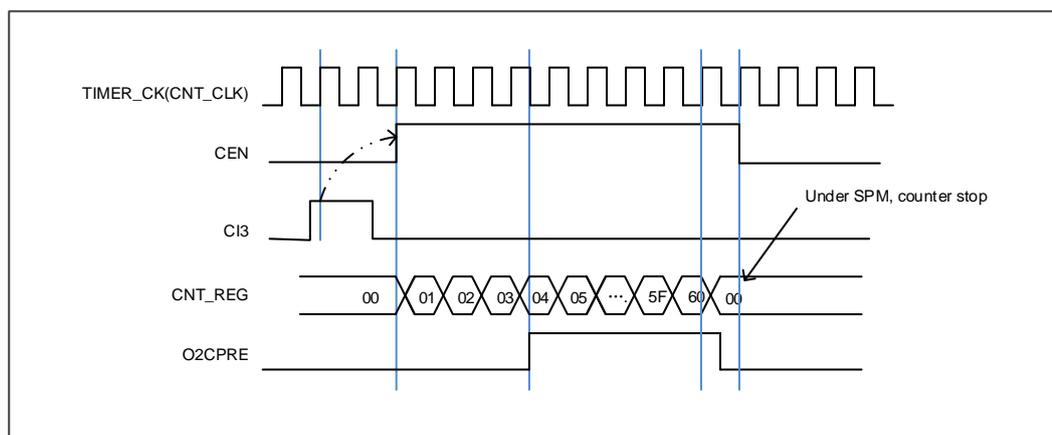
单脉冲模式与重复模式是相反的, 设置 `TIMERx_CTL0` 寄存器的 `SPM` 位置 1, 则使能单脉冲模式。当 `SPM` 置 1, 计数器在下次更新事件到来后清零并停止计数。为了得到脉冲波, 可以通过设置 `CHxCOMCTL` 配置 `TIMERx` 为 PWM 模式或者比较模式。

一旦设置定时器运行在单脉冲模式下, 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。如果 `CEN` 值被硬件更新事件自动清 0, 计数器将被再次初始化。

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 14-80. 单脉冲模式，TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60



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。

14.4.5. TIMERx 寄存器(x=15,16)

TIMER15 基地址：0x4001 4400

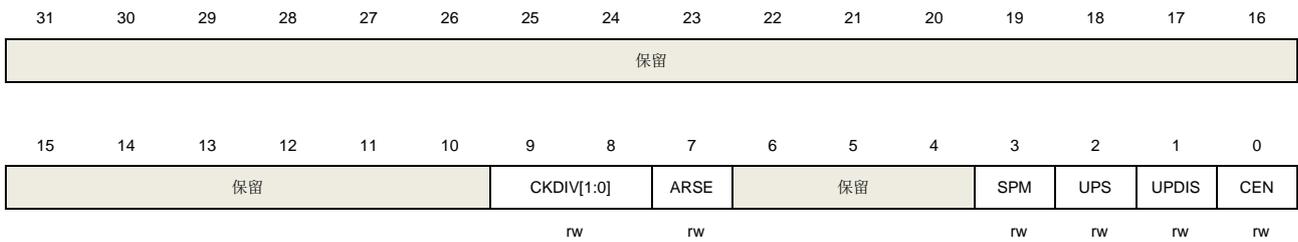
TIMER16 基地址：0x4001 4800

控制寄存器 0 (TIMERx_CTL0)

地址偏移：0x00

复位值：0x0000 0000

该寄存器通过字访问(32位)



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(TIMER_CK) 与死区时间和采样时钟(DTS) 之间的分频系数，死区发生器和数字滤波器会用到 DTS 时间。 00: fDTS=fTIMER_CK 01: fDTS= fTIMER_CK /2 10: fDTS= fTIMER_CK /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: 更新事件发生后，计数器继续计数 1: 在下次更新事件发生时，CEN 硬件清零并且计数器停止计数
2	UPS	更新请求源 软件配置该为，选择更新事件源。 0: 使能后，下述任一事件产生更新中断或 DMA 请求： <ul style="list-style-type: none"> – UPG 位被置 1 – 计数器溢出/下溢 – 从模式控制器产生的更新 1: 使能后只有计数器溢出/ 下溢才产生更新中断或 DMA 请求
1	UPDIS	禁止更新。 该位用来使能或禁能更新事件的产生。 <ul style="list-style-type: none"> 0: 更新事件使能.当以下事件之一发生时，更新事件产生，具有缓存的寄存器被装入它们的预装载值： <ul style="list-style-type: none"> – UPG 位被置 1 – 计数器溢出/下溢 – 从模式控制器产生一个更新事件 1: 更新事件禁能. 带有缓存的寄存器保持原有值，如果 UPG 位被置 1 或者从模式控制器产生一个硬件复位事件，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、暂停模式和编码器模式才能工作。触发模式可以自动地通过硬件设置 CEN 位。
---	-----	--

控制寄存器 1 (TIMERx_CTL1)

地址偏移: 0x04

复位值: 0x0000 0000

该寄存器通过字访问(32位)



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 TRIG1 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

该寄存器通过字访问(32位)



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	BRK0IE	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

该寄存器通过字访问(32位)



位/位域	名称	描述
31:14	保留	必须保持复位值。
13	SYSBIF	系统源中止事件中断标志位 当系统中止源有效时，该位由硬件置1，当系统源无效时，该位由软件清零。 0: 无系统中止事件中断发生 1: 系统中止事件中断发生 注意：当该位置1时，在通道输出恢复前，该位必须由软件清零。
12:10	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	BRK0IF	中止0中断标志位 一旦中止输入有效，由硬件对该位置'1'。如果中止输入无效，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。

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

该寄存器通过字访问(32位)



位/位域	名称	描述
31:8	保留	必须保持复位值。
7	BRKG	产生中止 0 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，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。

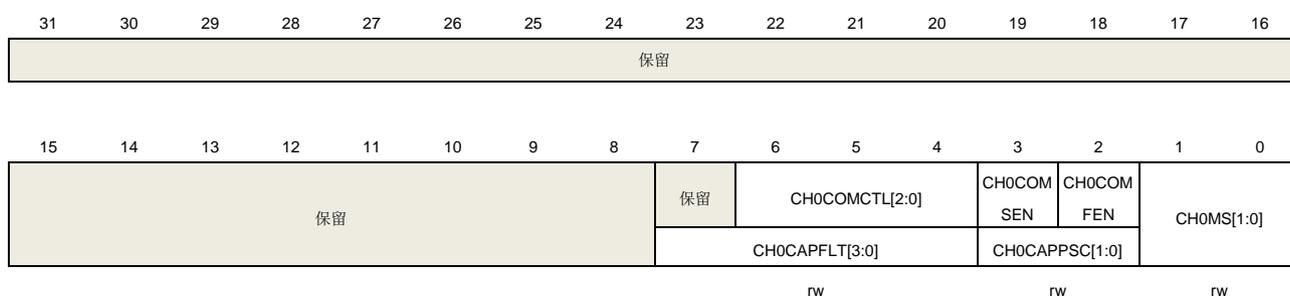
- 0: 无更新事件产生
1: 产生更新事件

通道控制寄存器 0 (TIMERx_CHCTL0)

地址偏移: 0x18

复位值: 0x0000 0000

该寄存器通过字访问(32位)



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 此位定义了输出参考信号 O0CPRE 的动作，而 O0CPRE 决定了 CH0_O、CH0_ON 的值。O0CPRE 高电平有效，而 CH0_O、CH0_ON 的有效电平取决于 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 为有效电平，否则为无效电平。在向下计数时，一旦计数器的值大于 TIMERx_CH0CV 时，O0CPRE 为无效电平，否则为有效电平。

111: PWM 模式 1。在向上计数时，一旦计数器值小于 `TIMERx_CH0CV` 时，`O0CPRE` 为无效电平，否则为有效电平。在向下计数时，一旦计数器的值大于 `TIMERx_CH0CV` 时，`O0CPRE` 为有效电平，否则为无效电平。

在 PWM 模式 0 或 PWM 模式 1 中，只有当比较结果改变了或者输出比较模式中从时基模式切换到 PWM 模式时，`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	<p>通道 0 输出比较影子寄存器使能</p> <p>当此位被置 1，<code>TIMERx_CH0CV</code> 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。</p> <p>0: 禁止通道 0 输出/比较影子寄存器</p> <p>1: 使能通道 0 输出/比较影子寄存器</p> <p>仅在单脉冲模式下(<code>TIMERx_CTL0</code> 寄存器的 <code>SPM =1</code>)，可以在未确认预装载寄存器情况下使用 PWM 模式</p> <p>当 <code>TIMERx_CCHP</code> 寄存器的 <code>PROT [1:0]=11</code> 且 <code>CH0MS =00</code> 时此位不能被改变。</p>
2	CH0COMFEN	<p>通道 0 输出比较快速使能</p> <p>当该位为 1 时，如果通道配置为 PWM0 模式或者 PWM1 模式，会加快捕获/比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，<code>CH0_O</code> 被设置为比较电平而与比较结果无关。</p> <p>0: 禁止通道 0 输出比较快速。当触发器的输入有一个有效沿时，激活 <code>CH0_O</code> 输出的最小延时为 5 个时钟周期</p> <p>1: 使能通道 0 输出比较快速。当触发器的输入有一个有效沿时，激活 <code>CH0_O</code> 输出的最小延时为 3 个时钟周期</p>
1:0	CH0MS[1:0]	<p>通道 0 I/O 模式选择</p> <p>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 (<code>TIMERx_CHCTL2</code> 寄存器的 <code>CH0EN</code> 位被清 0) 时这些位才可写。</p> <p>00: 通道 0 配置为输出</p> <p>01: 通道 0 配置为输入，IS0 映射在 <code>CI0FE0</code> 上</p> <p>10: 保留</p> <p>11: 保留</p>

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]	<p>通道 0 输入捕获滤波控制</p> <p>数字滤波器由一个事件计数器组成，它记录 N 个输入事件后会产生一个输出的跳变。这些位定义了 <code>CI0</code> 输入信号的采样频率和数字滤波器的长度。</p> <p>0000: 无滤波器，<code>fSAMP= fDTS</code>, <code>N=1</code></p> <p>0001: <code>fSAMP= fPCLK</code>, <code>N=2</code></p> <p>0010: <code>fSAMP= fPCLK</code>, <code>N=4</code></p> <p>0011: <code>fSAMP= fPCLK</code>, <code>N=8</code></p> <p>0100: <code>fSAMP=fDTS/2</code>, <code>N=6</code></p>

		0101: fSAMP=fDTS/2, N=8
		0110: fSAMP=fDTS/4, N=6
		0111: fSAMP=fDTS/4, N=8
		1000: fSAMP=fDTS/8, N=6
		1001: fSAMP=fDTS/8, N=8
		1010: fSAMP=fDTS/16, N=5
		1011: fSAMP=fDTS/16, N=6
		1100: fSAMP=fDTS/16, N=8
		1101: fSAMP=fDTS/32, N=5
		1110: fSAMP=fDTS/32, N=6
		1111: fSAMP=fDTS/32, N=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 <code>TIMERx_CHCTL2</code> 寄存器中的 <code>CH0EN</code> =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												<code>CH0NP</code>	<code>CH0NEN</code>	<code>CH0P</code>	<code>CH0EN</code>
												rw	rw	rw	rw

位/位域	名称	描述
31:4	保留	必须保持复位值
3	CH0NP	通道 0 互补输出极性 当通道 0 配置为输出模式，此位定义了互补输出信号的极性。 0: 通道 0 高电平有效 1: 通道 0 低电平有效 当通道 0 配置为输入模式时，此位和 <code>CH0P</code> 联合使用，作为输入信号 <code>CI0</code> 的极性选择控制信号。

		当 <code>TIMERx_CCHP</code> 寄存器的 <code>PROT [1:0]=11</code> 或 <code>10</code> 时此位不能被更改。
2	<code>CH0NEN</code>	<p>通道 0 互补输出使能</p> <p>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。</p> <p>0: 禁止通道 0 互补输出</p> <p>1: 使能通道 0 互补输出</p>
1	<code>CH0P</code>	<p>通道 0 极性</p> <p>当通道 0 配置为输出模式时，此位定义了输出信号极性。</p> <p>0: 通道 0 高电平有效</p> <p>1: 通道 0 低电平有效</p> <p>当通道 0 配置为输入模式时，此位定义了 <code>CI0</code> 信号极性</p> <p><code>[CH0NP, CH0P]</code> 将选择 <code>CI0FE0</code> 或者 <code>CI1FE0</code> 的有效边沿或者捕获极性</p> <p><code>[CH0NP==0, CH0P==0]</code>: 把 <code>CIxFE0</code> 的上升沿作为捕获或者从模式下触发的有效信号，并且 <code>CIxFE0</code> 不会被翻转。</p> <p><code>[CH0NP==0, CH0P==1]</code>: 把 <code>CIxFE0</code> 的下降沿作为捕获或者从模式下触发的有效信号，并且 <code>CIxFE0</code> 会被翻转。</p> <p><code>[CH0NP==1, CH0P==0]</code>: 保留。</p> <p><code>[CH0NP==1, CH0P==1]</code>: 把 <code>CIxFE0</code> 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 <code>CIxFE0</code> 不会被翻转。</p> <p>当 <code>TIMERx_CCHP</code> 寄存器的 <code>PROT [1:0]=11</code> 或 <code>10</code> 时此位不能被更改。</p>
0	<code>CH0EN</code>	<p>通道 0 捕获/比较使能</p> <p>当通道 0 配置为输出模式时，将此位置 1 使能 <code>CH0_O</code> 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。</p> <p>0: 禁止通道 0</p> <p>1: 使能通道 0</p>

计数器寄存器 (`TIMERx_CNT`)

地址偏移: `0x24`

复位值: `0x0000 0000`

该寄存器通过字访问(32位)



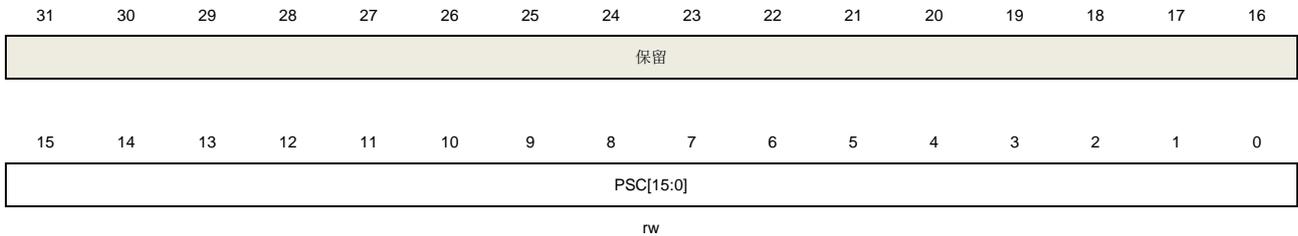
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	<code>CNT[15:0]</code>	这些位是当前的计数值。写操作能改变计数器值。

预分频寄存器 (TIMERx_PSC)

地址偏移: 0x28

复位值: 0x0000 0000

该寄存器通过字访问(32位)



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

计数器自动重载寄存器 (TIMERx_CAR)

地址偏移: 0x2C

复位值: 0x0000 0000

该寄存器通过字访问(32位)



位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	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。

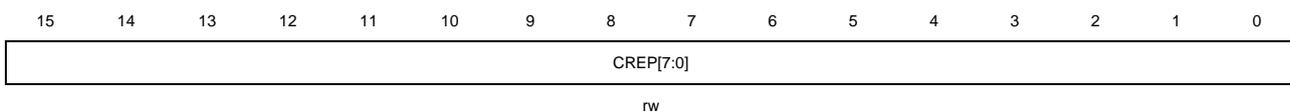
重复计数寄存器 (TIMERx_CREP)

地址偏移: 0x30

复位值: 0x0000 0000

该寄存器通过字访问(32位)





位/位域	名称	描述
31:8	保留	必须保持复位值.
15:0	CREP[15:0]	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

该寄存器通过字访问(32位)



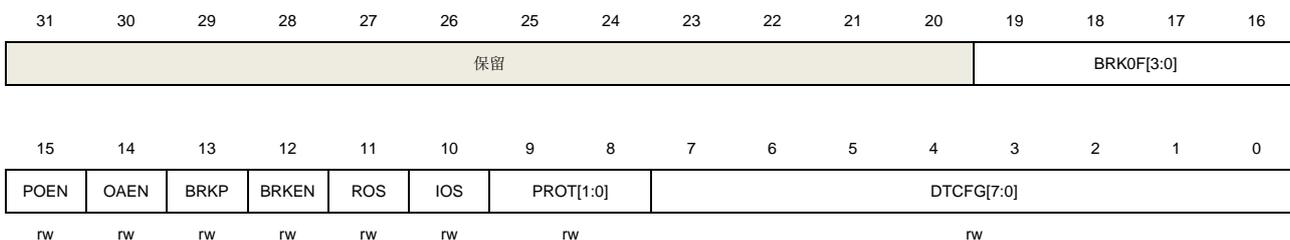
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CH0VAL[15:0]	通道 0 的捕获或比较值 当通道 0 配置为输入模式时, 这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 0 配置为输出模式时, 这些位包含了即将和计数器比较的值。使能相应影子寄存器后, 影子寄存器值随每次更新事件更新。

互补通道保护寄存器 (TIMERx_CCHP)

地址偏移: 0x44

复位值: 0x0000 0000

该寄存器通过字访问(32位)



位/位域	名称	描述
31:16	保留	必须保持复位值
19:16	BRK0F[3:0]	<p>BREAK0输入信号滤波</p> <p>数字滤波器由一个事件计数器组成，它记录N个输入事件后会产生一个输出的跳变。这些位定义了BREAK0输入信号的采样频率和数字滤波器的长度。</p> <p>0000: 无滤波器，BREAK0异步有效，N=1</p> <p>0001: fSAMP = fCK_TIMER, N=2</p> <p>0010: fSAMP = fCK_TIMER, N=4</p> <p>0011: fSAMP = fCK_TIMER, N=8</p> <p>0100: fSAMP = fDTS/2, N=6</p> <p>0101: fSAMP = fDTS/2, N=8</p> <p>0110: fSAMP = fDTS/4, N=6</p> <p>0111: fSAMP = fDTS/4, N=8</p> <p>1000: fSAMP = fDTS/8, N=6</p> <p>1001: fSAMP = fDTS/8, N=8</p> <p>1010: fSAMP = fDTS/16, N=5</p> <p>1011: fSAMP = fDTS/16, N=6</p> <p>1100: fSAMP = fDTS/16, N=8</p> <p>1101: fSAMP = fDTS/32, N=5</p> <p>1110: fSAMP = fDTS/32, N=6</p> <p>1111: fSAMP = fDTS/32, N=8</p> <p>此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0] =00 时才可修改。</p>
15	POEN	<p>所有的通道输出使能</p> <p>根据 OAEN 位，该位可以软件设置或者硬件自动设置。一旦中止输入有效，该位被硬件异步清 0。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。</p> <p>0: 禁止通道输出或强制为空闲状态</p> <p>1: 使能通道输出</p>
14	OAEN	<p>自动输出使能</p> <p>此位定义了 POEN 位是否可以被硬件自动置 1。</p> <p>0: POEN 位不能被硬件置 1</p> <p>1: 如果中止输入无效，下一次更新事件发生时，POEN 位能被硬件自动置 1</p> <p>此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0] =00 时才可修改。</p>
13	BRKP	<p>中止极性</p> <p>此位定义了中止输入信号 BKIN 的极性。</p> <p>0: 中止输入低电平有效</p> <p>1: 中止输入高电平有效</p>
12	BRKEN	<p>中止使能</p> <p>此位置 1 使能中止事件和 CCS 时钟失败事件输入。</p> <p>0: 禁能中止输入</p>

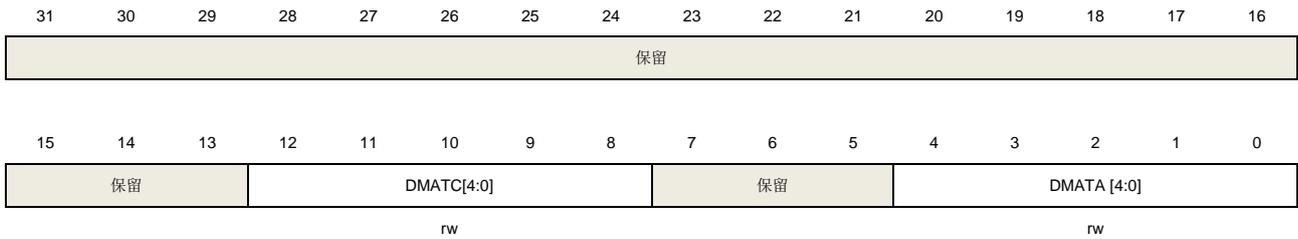
		1: 使能中止输入 此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0]=00 时才可修改。
11	ROS	运行模式下“关闭状态”配置 当 POEN 位被置 1，此位定义了通道(带有互补输出且配置为输出模式)的输出状态。 0: 当 POEN 位被置 1，通道输出信号 (CHx_O/ CHx_ON)被禁止 1: 当 POEN 位被置 1，通道输出信号 (CHx_O/ CHx_ON)被使能，和 TIMER0_CHCTL2 寄存器 CHxEN/CHxNEN 位有关 此位在 TIMERx_CCHP 寄存器的 PROT [1:0]=10 或 11 时不能被更改。
10	IOS	空闲模式下“关闭状态”配置 当 POEN 位被清 0，此位定义了已经配置为输出模式的通道的输出状态。 0: 当 POEN 位被清 0，通道输出信号(CHx_O/ CHx_ON)被禁止 1: 当 POEN 位被清 0，通道输出信号(CHx_O/ CHx_ON)被使能，和 TIMERx_CHCTL2 寄存器 CHxEN/CHxNEN 位有关 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_CHCTLR0/1 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 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

该寄存器通过字访问(32位)



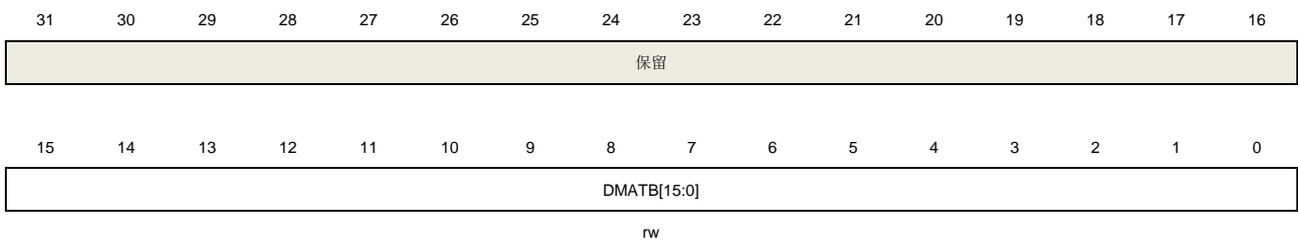
位/位域	名称	描述
31:13	保留	必须保持复位值。
12:8	DMATC [4:0]	DMA 传输计数 该位域定义了 DMA 访问（读写）TIMERx_DMATB 寄存器的数量
7:5	保留	必须保持复位值
4:0	DMATA [4:0]	DMA 传输起始地址 该位域定义了 DMA 访问 TIMERx_DMATB 寄存器的第一个地址。当通过 TIMERx_DMA 第一次访问时，访问的就是该位域指定的地址。第二次访问 TIMERx_DMATB 时，将访问起始地址+0x4。 5'b0_0000: TIMERx_CTL0 5'b0_0001: TIMERx_CTL1 ... 总之：起始地址 = 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

该寄存器通过字访问(32位)



位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	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。

附加通道控制寄存器 0 (TIMER0_AFCTL0)

地址偏移: 0x60

复位值: 0x0000 0001

该寄存器只能按字 (32位) 访问



位/位域	名称	描述
31:10	保留	必须保持复位值.
9	BRK0IN0P	<p>BREAK0 BRKIN0备用输入极性</p> <p>该位用于配置BRKIN0输入极性, 具体极性是由该位和BRK0P位共同确定。</p> <p>0: BRKIN0输入信号不反相 (BRK0P =0, 输入信号低有效; BRK0P =1, 输入信号高有效)</p> <p>1: BRKIN0 输入信号反相 (BRK0P =0, 输入信号高有效; BRK0P =1, 输入信号低有效)</p> <p>此位只有在TIMERx_CCHP寄存器的PROT [1:0] =00时才可修改。</p>
8:1	保留	必须保持复位值.
0	BRK0IN0E	<p>BREAK0 BRKIN0备用输入使能</p> <p>0: BRKIN0输入禁能</p> <p>1: BRKIN0输入使能</p> <p>此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0] =00 时才可修改。</p>

输入选择寄存器 (TIMERx_INSEL)

地址偏移: 0x68

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问



位/位域	名称	描述
------	----	----

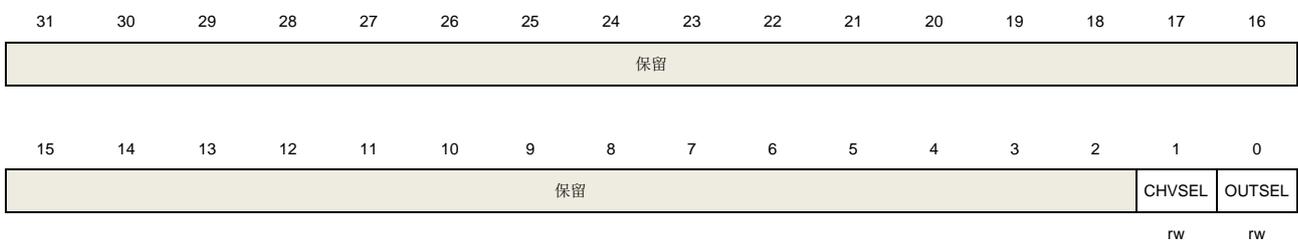
31:3	保留	必须保持复位值.
3:0	CI0_SEL[3:0]	<p>TIMER0_CH0输入选择</p> <p>0000: 通道0输入连接到TIMER0_CH0</p> <p>0001: 通道0输入连接到IRC32 对与TIMER16保留</p> <p>0010: 对与TIMER15保留, 通道0输入连接到HXTAL/32</p> <p>0011: 对与TIMER15保留, 对于TIMER16通道0输入连接到CKOUTSEL0</p> <p>0100: 通道0输入连接到CKOUTSEL1</p> <p>其他: 保留</p>

配置寄存器 (TIMERx_CFG)

地址偏移: 0xFC

复位值: 0x0000 0000

该寄存器通过字访问(32位)



位/位域	名称	描述
31:2	保留	必须保持复位值
1	CHVSEL	<p>写捕获比较寄存器选择位</p> <p>此位由软件写 1 或清 0。</p> <p>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</p> <p>0: 无影响</p>
0	OUTSEL	<p>输出值选择位</p> <p>此位由软件写 1 或清 0。</p> <p>1: 如果 POEN 位与 IOS 位均为 0, 则输出无效</p> <p>0: 无影响</p>

15. 通用同步异步收发器（USART）

15.1. 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（PCLK，CK_SYS，LXTAL或IRC48MDIV_PER）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。

15.2. 主要特征

- NRZ标准格式。
- 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UCLK设置的波特率设置。
- 可编程的波特率产生器，当时钟频率为48MHz，过采样为8，最高速度可达6Mbits/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。
- 可配置的多级缓存通信DMA访问数据缓冲区。
- 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，但是USART1/USART2只实现了上面所介绍的部分功能，下面这些功能在USART1/USART2中没有实现：

- 接收FIFO功能。
- 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

15.3. 功能描述

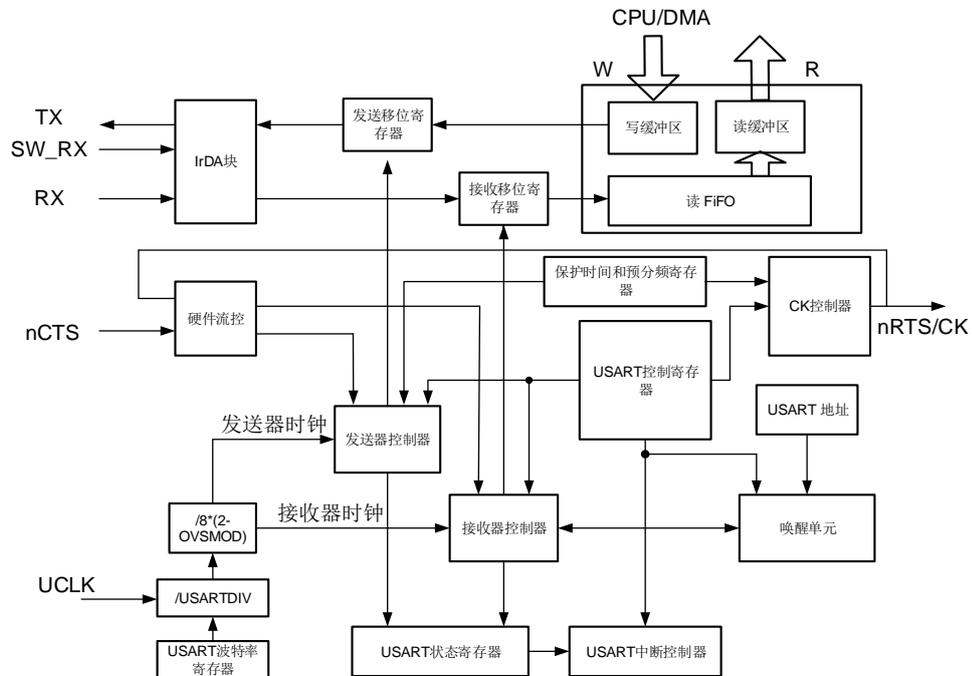
USART 接口通过[表 15-1. USART 重要引脚描述](#)中主要引脚从外部连接到其他设备。

表 15-1. 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

引脚	类型	描述
nCTS	输入	硬件流控模式发送使能信号
nRTS / CK	输出	硬件流控模式发送请求信号 / 用于同步通信的串行时钟信号

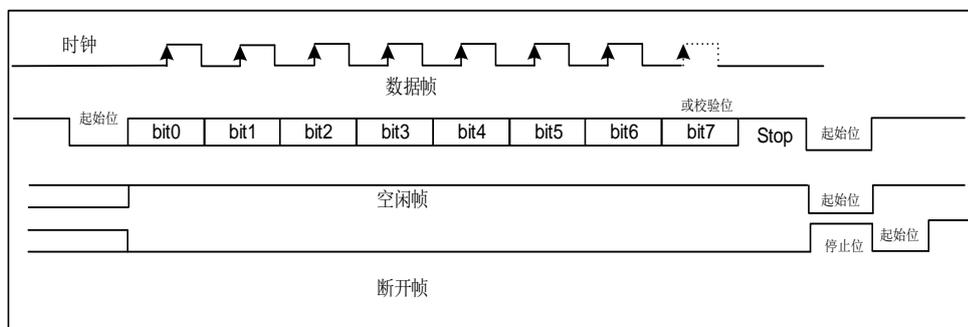
图 15-1. USART 模块内部框图



15.3.1. 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15-2. 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 15-2. 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
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。

15.3.2. 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：

$$\text{USARTDIV} = \frac{\text{UCLK}}{16 \times \text{Baud Rate}} \quad (15-1)$$

如果过采样是8，公式为：

$$\text{USARTDIV} = \frac{\text{UCLK}}{8 \times \text{Baud Rate}} \quad (15-2)$$

注意：只有USART0支持双时钟域。对于USART1和 USART2，UCLK仅为PCLK。

例如，当过采样是16：

- 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。
- 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。

15.3.3. 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。

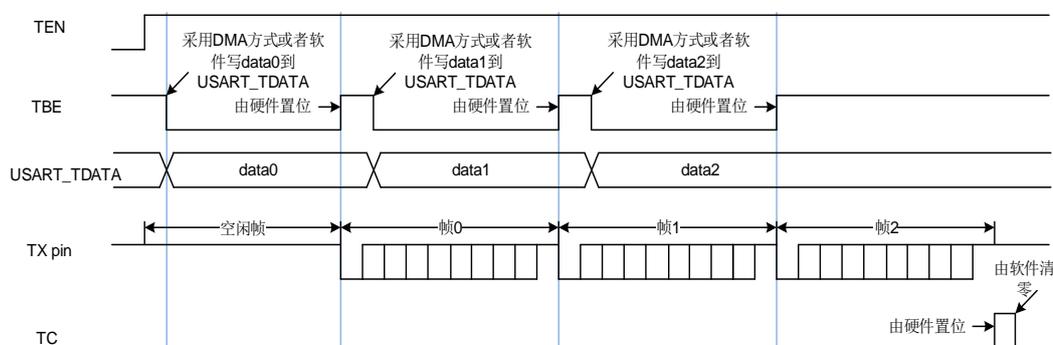
系统上电后，TBE默认为高电平。在USART_STAT寄存器中TBE置位时，数据可以在不覆盖前一个数据的情况下写入USART_TDATA寄存器。当数据写入USART_TDATA寄存器，TBE位将被清0。在数据由USART_TDATA移入移位寄存器后，该位由硬件置1。如果数据在一个发送过程正在进行时被写入USART_TDATA寄存器，它将首先被存入发送缓冲区，在当前发送过程完成时传输到发送移位寄存器中。如果数据在写入USART_TDATA寄存器时，没有发送过程正在进行，TBE位将被清零然后迅速置位，原因是数据被立刻传输到发送移位寄存器。

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 15-3. 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 15-3. 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。

15.3.4. 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。

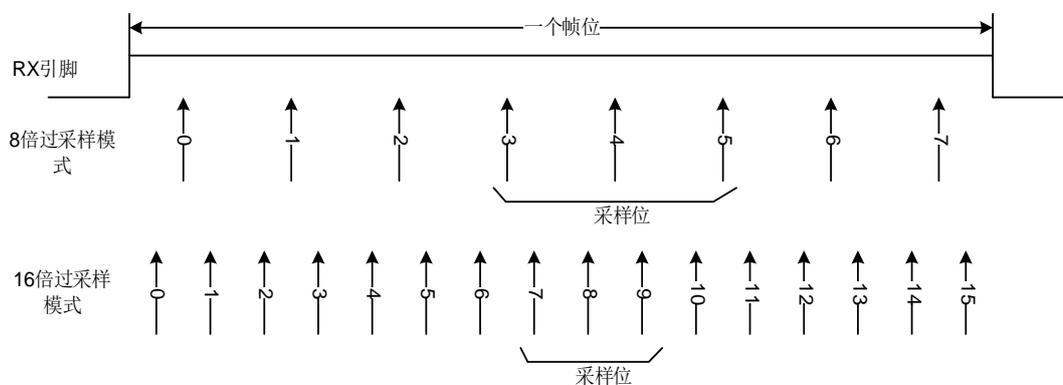
当接收到一个数据帧, USART_STAT寄存器中的RBNE置位, 如果设置了USART_CTL0寄存器中相应的中断使能位RBNEIE, 将会产生中断。在USART_STAT寄存器中可以观察接收状态标志。

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, 不管是起始位, 数据位, 奇偶校验位或者停止位, 都将产生噪声错误 (NERR)。如果使能DMA, 并置位USART_CTL2寄存器中ERRIE, 将会产生中断。如果在USART_CTL2中置位OSB, 接收器将仅获取一个采样点来估计一个数据位的值。在这种情况下将不会检测到噪声错误。

图 15-4. 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。如果使能DMA并置位USART_CTL2寄存器中ERRIE位或者置位RBNEIE，将产生中断。

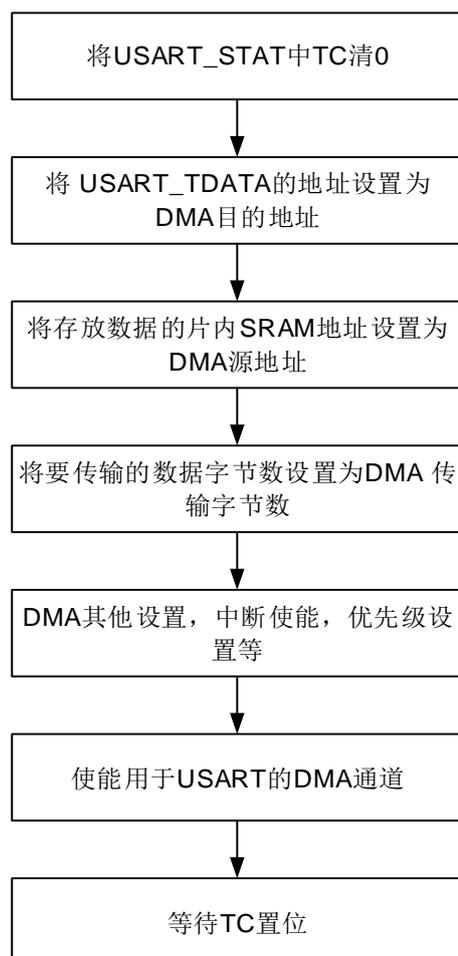
在一个接收过程中，NERR、PERR、FERR、ORERR总是分别和RBNE同时置位。如果没有使能DMA，软件需检查RBNE中断是否由NERR、PERR、FERR或者ORERR置位产生。

15.3.5. 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 15-5. 采用DMA方式实现USART数据发送配置步骤](#)所示。

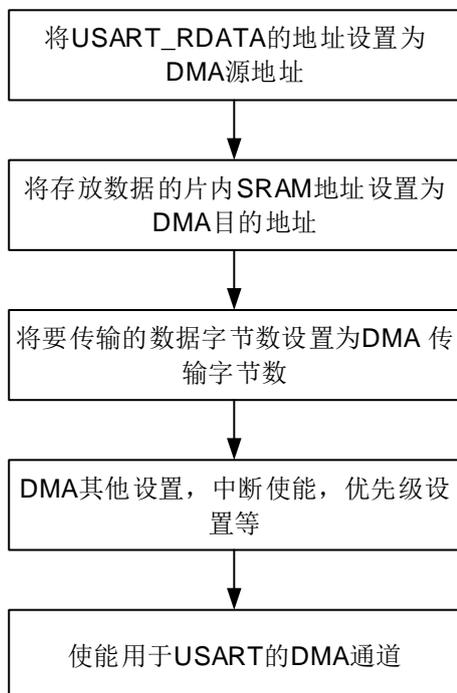
图 15-5. 采用DMA方式实现USART数据发送配置步骤



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 15-6. [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 15-6. 采用 DMA 方式实现 USART 数据接收配置步骤

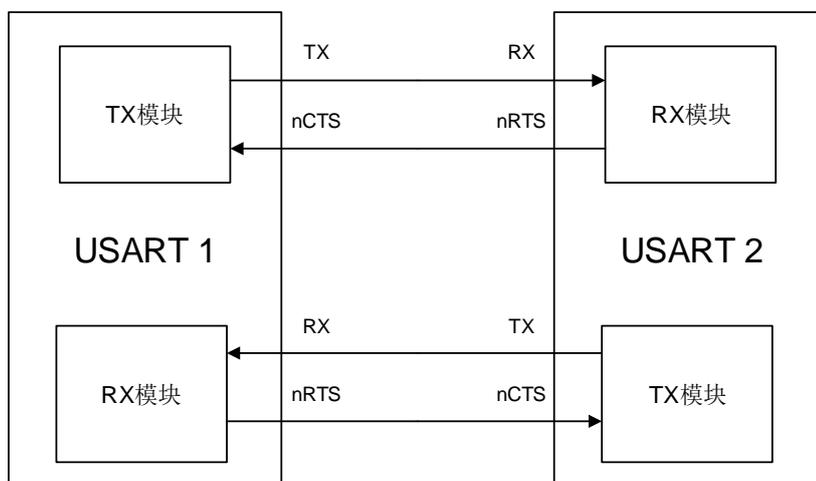


当USART接收到的数据数量达到了DMA传输数据数量，DMA模块将产生传输完成中断。

15.3.6. 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 15-7. 两个 USART 之间的硬件流控制



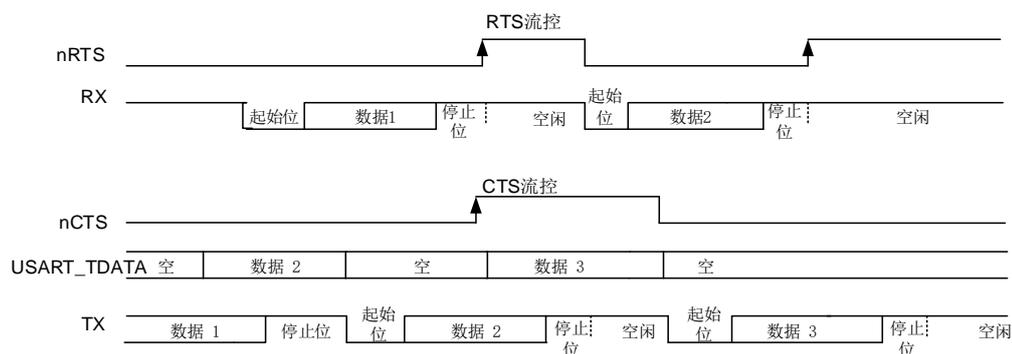
RTS 流控

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 15-8. 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。

15.3.7. 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，当它是被空闲帧唤醒时，USART_STAT寄存器中IDLEF位不会被置1。

当USART_CTL0寄存器中WM被置位，数据最高位会被认为是地址标志位。如果地址标志位为1，该字节被认为是地址字节。如果地址标志位是0，该字节被认为是数据字节。如果地址字节

的低4位或低7位与USART_CTL1寄存器中的ADDR位相同，硬件会将RWU清零，并退出静默模式。接收到将USART唤醒的数据帧，RBNE将置位。状态标志可以从USART_STAT寄存器中获取。如果地址字节的低4位或低7位与USART_CTL1寄存器中的ADDR位不相同，硬件会置位RWU并自动进入静默模式。在这种情况下，RBNE不会被置位。

如果USART_CTL0寄存器中PCEN位被置位，地址字节最高位被视为校验位，其余位被视为地址位。如果ADDM位被置位，且接收帧为7位的数据，其中最低的6位将与ADDR[5:0]比较。如果ADDM位被置位，且接收帧为9位的数据，其中低8位将与ADDR[7:0]进行比较。

15.3.8. LIN 模式

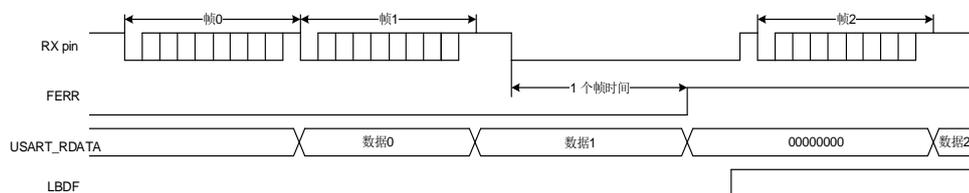
将USART_CTL1寄存器的LMEN置位即可使能本地互连网络模式。在LIN模式下，USART_CTL1寄存器中CKEN, STB[1:0]和USART_CTL2的SCEN, HDEN, IREN位都应该被清0。

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。

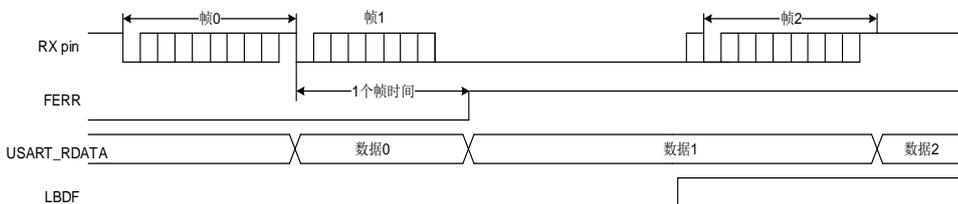
如[图 15-9. 空闲状态下检测断开帧](#)所示，如果断开帧发生在空闲状态下，USART接收器会接收到一个全0数据帧，同时FERR置位。

图 15-9. 空闲状态下检测断开帧



如[图 15-10. 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 15-10. 数据传输过程中检测断开帧



15.3.9. 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 15-11. 同步模式下的 USART 示例

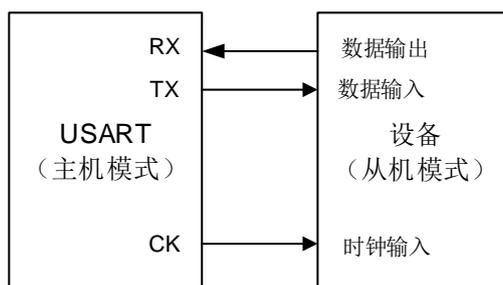
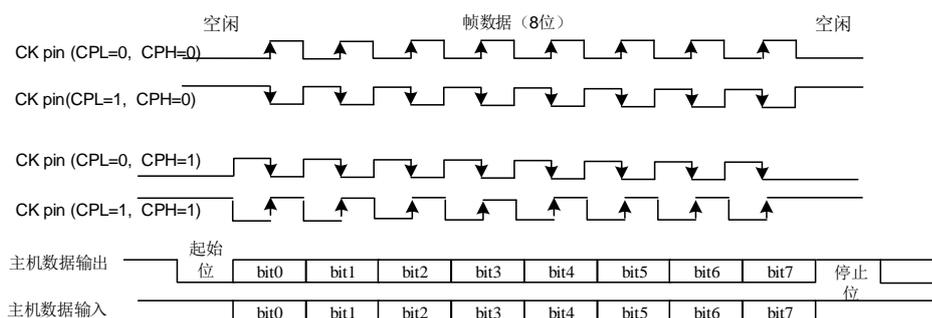


图 15-12. 8-bit 格式的 USART 同步通信波形 (CLEN=1)

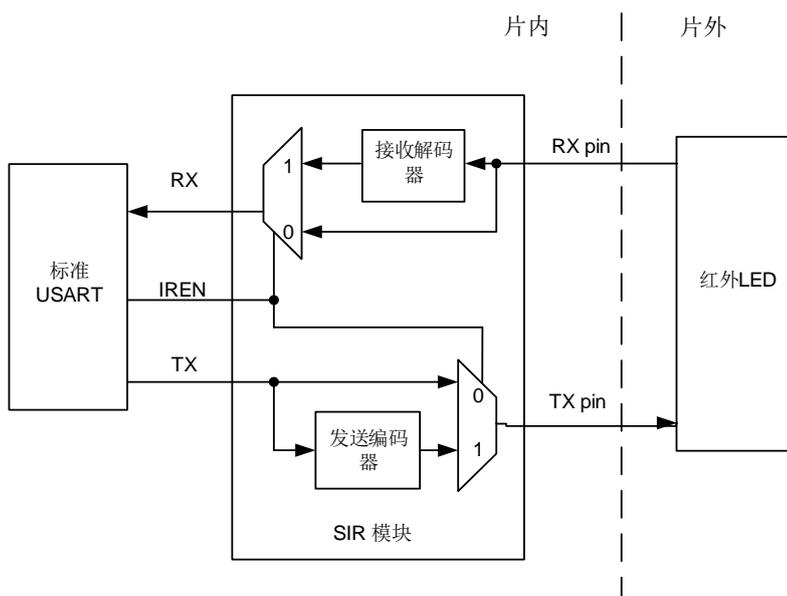


15.3.10. 串行红外 (IrDA SIR) 编解码功能模块

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。

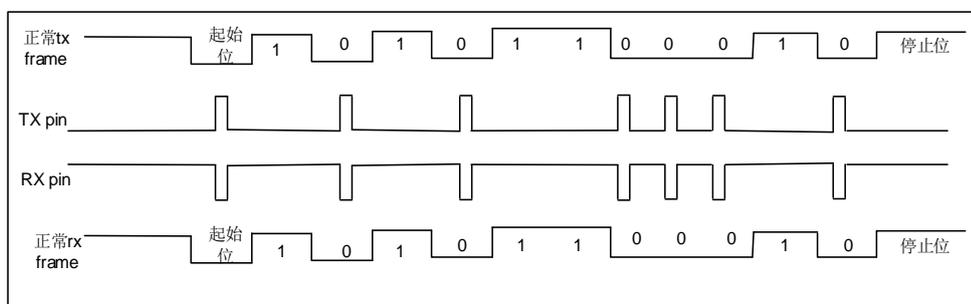
图 15-13. IrDA SIR ENDEC 模块



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 15-14. 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]位配置。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。

15.3.11. 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，RX引脚不再使用。TX引脚应被配置为开漏模式，通信冲突应由软件处理。

15.3.12. 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 15-15. 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SCRNUM次。在重发数据帧前面会插入2.5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SCRNUM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。

15.3.13. 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。

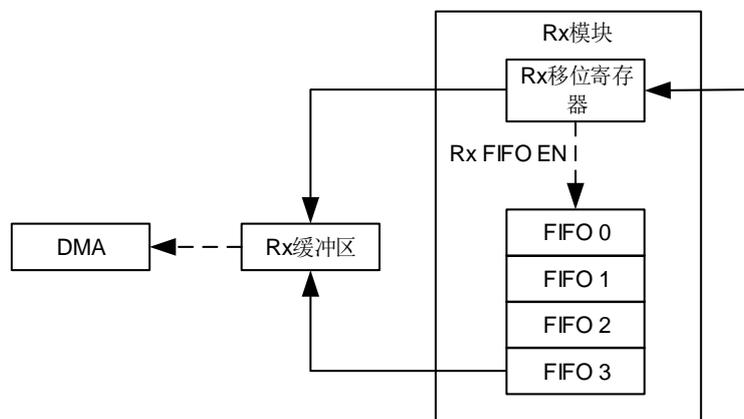
为了检测空闲状态，必须置位USART_CTL1寄存器的RTEN位和USART_CTL0寄存器的RTIE位。USART_RT寄存器必须被设置成与2个字节超时所对应的值。在最后一个停止位被接收后，当接收线在这期间是空闲的，将产生一个中断，通知软件当前块接收已经完成。

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。

15.3.14. 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，发生超载错误。接收FIFO和接收缓冲区可储存多至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 15-16. 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。

15.3.15. 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IRC48MDIV_PER或LXTAL（请参考[配置寄存器 1 \(RCU_CFG1\)](#)）。

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。

15.3.16. USART 中断

USART 中断事件和标志如[表 15-3. USART 中断请求](#)所示：

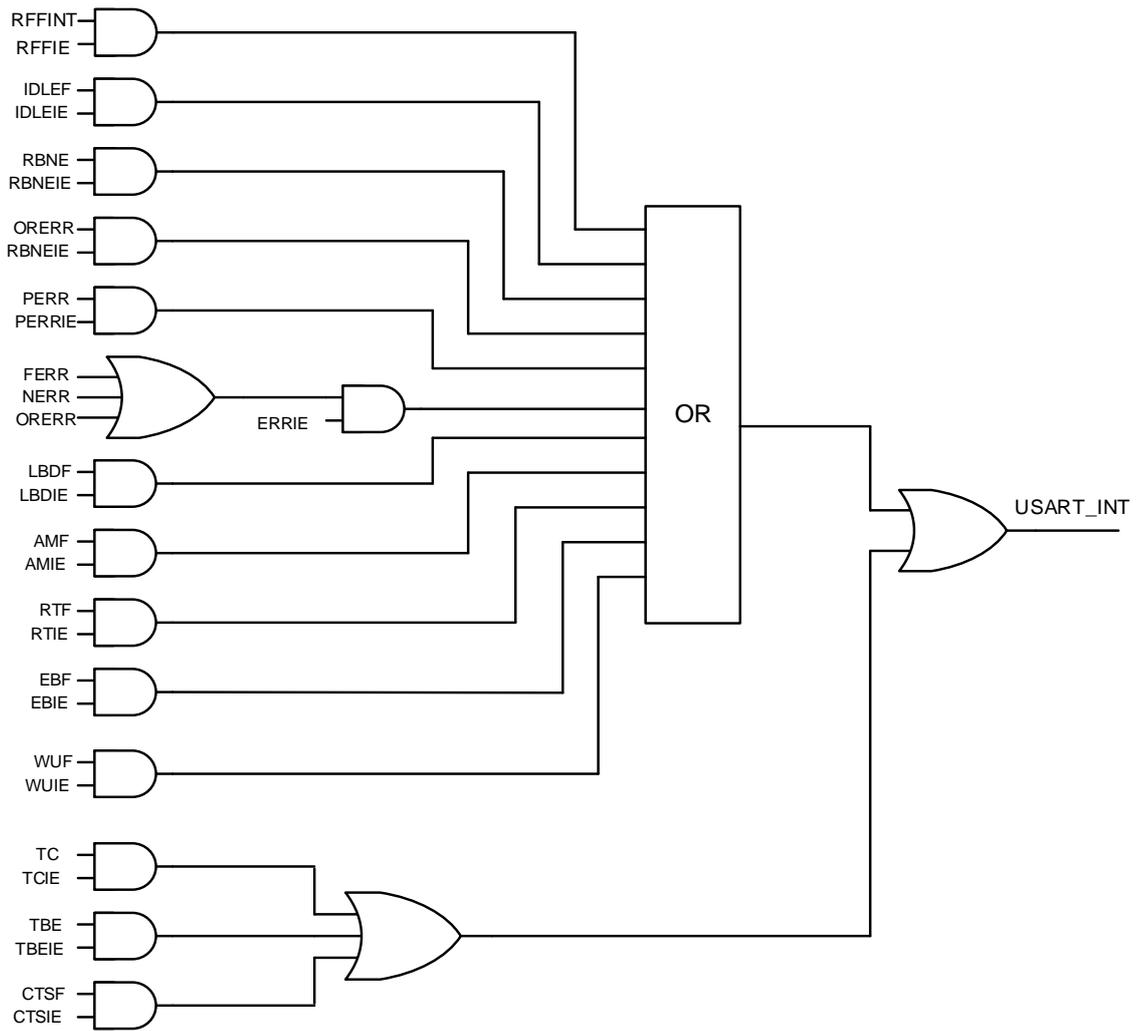
表 15-3. USART 中断请求

中断事件	事件标志	使能控制位
发送数据寄存器空	TBE	TBEIE

中断事件	事件标志	使能控制位
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	
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或ORERR或FERR	ERRIE
字符匹配	AMF	AMIE
接收超时错误	RTF	RTIE
发现块尾	EBF	EBIE
从Deepsleep模式唤醒	WUF	WUIE

在发送给中断控制器之前，所有的中断事件是逻辑或的关系。因此在任何时候 USART 只能向控制器产生一个中断请求。不过软件可以在一个中断服务程序里处理多个中断事件。

图 15-17. USART 中断映射框图



15.4. USART 寄存器

USART0基地址: 0x4001 3800

USART1基地址: 0x4000 4400

USART2基地址: 0x4000 4800

15.4.1. USART 控制寄存器 0 (USART_CTL0)

地址偏移: 0x00

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				EBIE	RTIE	DEA[4:0]				DED[4:0]					
				rw	rw	rw				rw					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVSMOD	AMIE	MEN	WL	WM	PCEN	PM	PERRIE	TBEIE	TCIE	RBNEIE	IDLEIE	TEN	REN	UESM	UEN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:28	保留	必须保持复位值。
27	EBIE	块尾中断使能 0: 中断禁止 1: 中断使能 在USART1和USART2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USART1和USART2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	WM	从静默模式唤醒方法 0: 空闲线 1: 地址标记 当USART被使能（UEN=1）时，该位域不能被改写。
10	PCEN	校验控制使能 0: 校验控制禁用 1: 校验控制被使能 当USART被使能（UEN=1）时，该位域不能被改写。
9	PM	校验模式 0: 偶校验 1: 奇校验 当USART被使能（UEN=1）时，该位域不能被改写。
8	PERRIE	校验错误中断使能 0: 校验错误中断禁用 1: 当USART_STAT寄存器的PERR位置位时，将触发中断。
7	TBEIE	发送寄存器空中断使能 0: 中断禁止 1: 当USART_STAT寄存器的TBE位置位时，将触发中断。
6	TCIE	发送完成中断使能 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IRC16M或LXTAL。 在USART1和USART2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

15.4.2. USART 控制寄存器 1 (USART_CTL1)

地址偏移: 0x04

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR[7:0]								RTEN	保留			MSBF	DINV	TINV	RINV
rw								rw				rw	rw	rw	rw
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USART1和USART2中，该位保留。
22:20	保留	必须保持复位值。
19	MSBF	高位在前 0: 数据发送 / 接收，采用低位在前 1: 数据发送 / 接收，采用高位在前 USART被使能（UEN=1）时，该位域不能被改写。
18	DINV	数据位反转 0: 数据位信号值没有反转 1: 数据位信号值被反转 USART被使能（UEN=1）时，该位域不能被改写。
17	TINV	TX管脚电平反转 0: TX管脚信号值没有反转 1: TX管脚信号值被反转。 USART被使能（UEN=1）时，该位域不能被改写。
16	RINV	RX管脚电平反转 0: RX管脚信号值没有反转。 1: RX管脚信号值被反转 USART被使能（UEN=1）时，该位域不能被改写。
15	STRP	交换TX / RX管脚 0: TX和RX管脚功能不被交换 1: TX和RX管脚功能被交换 当USART被使能（UEN=1）时，该位域不能改写。
14	LMEN	LIN模式使能 0: LIN模式关闭 1: LIN模式开启 USART被使能（UEN=1）时，该位域不能被改写。 在USART1和USART2中，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。
10	CPL	时钟极性 0: 在同步模式下，CK管脚不对外发送时保持为低电平

		1: 在同步模式下, CK管脚不对外发送时保持为高电平 USART被使能 (UEN=1) 时, 该位域不能被改写。
9	CPH	时钟相位 0: 在同步模式下, 在首个时钟边沿采样第一个数据 1: 在同步模式下, 在第二个时钟边沿采样第一个数据 USART被使能 (UEN=1) 时, 该位域不能被改写。
8	CLEN	CK长度 0: 在同步模式下, 最后一位 (MSB) 的时钟脉冲不输出到CK管脚 1: 在同步模式下, 最后一位 (MSB) 的时钟脉冲输出到CK管脚 USART被使能 (UEN=1) 时, 该位域不能被改写。
7	保留	必须保持复位值。
6	LBDIE	LIN断开信号检测中断使能 0: 断开信号检测中断禁用 1: 当USART_STAT的LBDF位置位, 将产生中断。 在USART1, 该位保留。
5	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USART1和USART2中, 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。

15.4.3. USART 控制寄存器 2 (USART_CTL2)

地址偏移: 0x08

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留									WUIE	WUM[1:0]		SCRTNUM[2:0]		保留	
									rw	rw	rw				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEP	DEM	DDRE	OVRD	OSB	CTSIE	CTSEN	RTSEN	DENT	DENR	SCEN	NKEN	HDEN	IRLP	IREN	ERRIE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:23	保留	必须保持复位值
22	WUIE	从深度睡眠模式唤醒中断使能 0: 从深度睡眠模式唤醒中断禁用 1: 从深度睡眠模式唤醒中断被使能 在USART1和USART2该位保留。
21:20	WUM[1:0]	从深度睡眠模式唤醒模式 这个位域指定什么事件可以置位USART_STAT寄存器中的WUF（从深度睡眠唤醒标志）标志。 00: WUF在地址匹配的时候置位。如何实现地址匹配在ADDR和ADDM中定义。 01:保留 10: WUF在检测到起始位时置位 11: WUF在检测到RBNE时置位 USART被使能（UEN=1）时，该位域不能被改写。 在USART1和USART2该位保留。
19:17	SCRTNUM[2:0]	智能卡自动重试数目 在智能卡模式下，这些位用来指定在发送和接收时重试的次数。在发送模式下，它指的是在产生发送错误（FERR位置位）之前自动重试的发送次数。 在接收模式下，它指的是在产生接收错误（RBNE位和PERR位置位）之前自动重试的接收次数。 当这些位被设置为0x0时，在发送模式下这些位将不会自动发送。 USART被使能（UEN=1）时，该位域被清零，并停止重发。 在USART1和USART2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	在接收错误时禁止DMA 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，以确保错误的不会被传输，但是下一个接收到的正确的数据会被传输。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被关闭。DMA请求会被屏蔽，直到相应的标志位被清0。RBNE标志和相应的错误标志位会被置位。软件在清除错误标志前，必须首先

		关DMA请求 (DMAR = 0) 或清RBNE。 USART被使能 (UEN=1) 时, 该位域不能被改写。
12	OVRD	<p>溢出禁止</p> <p>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。</p> <p>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。</p> <p>USART被使能 (UEN=1) 时, 该位域不能被改写。</p>
11	OSB	<p>单次采样方式</p> <p>0: 三次采样方法</p> <p>1: 一次采样方法</p> <p>USART被使能 (UEN=1) 时, 该位域不能被改写。</p>
10	CTSIE	<p>CTS中断使能</p> <p>0: CTS中断屏蔽</p> <p>1: 当USART_STAT的CTS位置位时, 会产生中断。</p>
9	CTSEN	<p>CTS使能</p> <p>0: CTS硬件流控禁用</p> <p>1: CTS硬件流控被使能</p> <p>USART被使能 (UEN=1) 时, 该位域不能被改写。</p>
8	RTSEN	<p>RTS使能</p> <p>0: RTS硬件流控禁用</p> <p>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。</p> <p>USART被使能 (UEN=1) 时, 该位域不能被改写。</p>
7	DENT	<p>DMA发送使能</p> <p>0: 关闭DMA发送模式</p> <p>1: 开启DMA发送模式</p>
6	DENR	<p>DMA接收使能</p> <p>0: 关闭DMA接收模式</p> <p>1: 开启DMA接收模式</p>
5	SCEN	<p>智能卡模式使能</p> <p>0: 智能卡模式禁用</p> <p>1: 智能卡模式使能</p> <p>USART被使能 (UEN=1) 时, 该位域不能被改写。</p> <p>在USART1和USART2中, 该位保留。</p>
4	NKEN	<p>智能卡模式NACK使能</p> <p>0: 当出现校验错误时不发送NACK</p> <p>1: 当出现校验错误时发送NACK</p> <p>USART被使能 (UEN=1) 时, 该位域不能被改写。</p> <p>在USART1和USART2中, 该位保留。</p>

3	HDEN	半双工使能 0: 禁用半双工模式 1: 开启半双工模式 USART被使能 (UEN=1) 时, 该位域不能被改写。
2	IRLP	IrDA低功耗模式 0: 正常模式 1: 低功耗模式 USART被使能 (UEN=1) 时, 该位域不能被改写。
1	IREN	IrDA模式使能 0: IrDA禁用 1: IrDA被使能 USART被使能 (UEN=1) 时, 该位域不能被改写。 在USART1和USART2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。

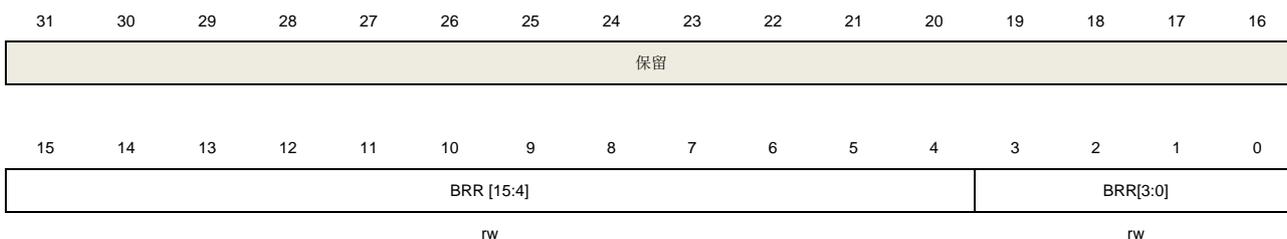
15.4.4. USART 波特率寄存器 (USART_BAUD)

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。



位/位域	名称	描述
31:16	保留	必须保持复位值。
15:4	BRR[15:4]	波特率分频系数的整数部分 DIV_INT[11:0] = BRR[15:4]
3:0	BRR[3:0]	波特率分频系数的小数部分 如果OVSMOD = 0, USARTDIV [3:0] = BRR [3:0]; 如果OVSMOD = 1, USARTDIV [3:1] = BRR [2:0], BRR [3]必须被置0。

15.4.5. USART 保护时间和预分频器寄存器 (USART_GP)

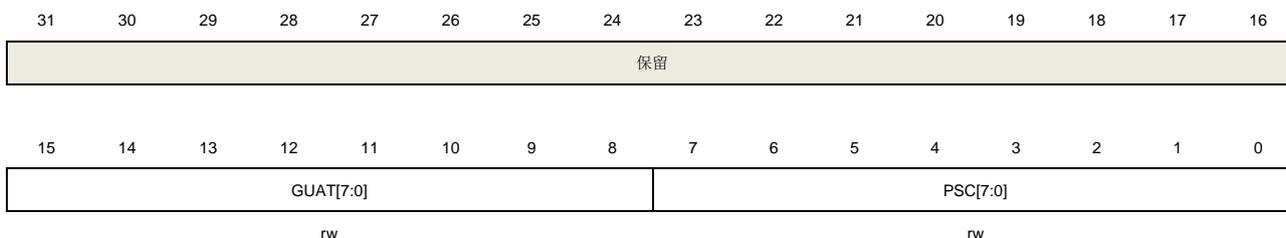
地址偏移: 0x10

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问

USART被使能 (UEN=1) 时, 该寄存器不能被改写。

在USART1和USART2中, 该寄存器保留。



位/位域	名称	描述
31:16	保留	必须保持复位值。
15:8	GUAT[7:0]	在智能卡模式下的保护时间值 USART被使能 (UEN=1) 时, 该位域不能被改写。
7:0	PSC[7:0]	预分频器值 在红外低功耗模式下, 对系统时钟进行分频已获得低功耗模式下的频率。寄存器的值是分频系数。 00000000: 保留 - 不设置这个值 00000001: 1分频 00000010: 2分频 ... 在IrDA正常模式下的分频值 00000001: 仅能设为这个值 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。

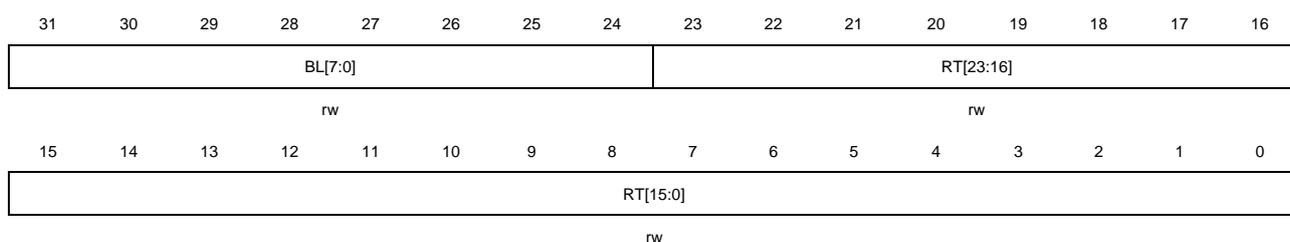
15.4.6. USART 接收超时寄存器 (USART_RT)

地址偏移: 0x14

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问

在USART1和USART2中，该寄存器保留。



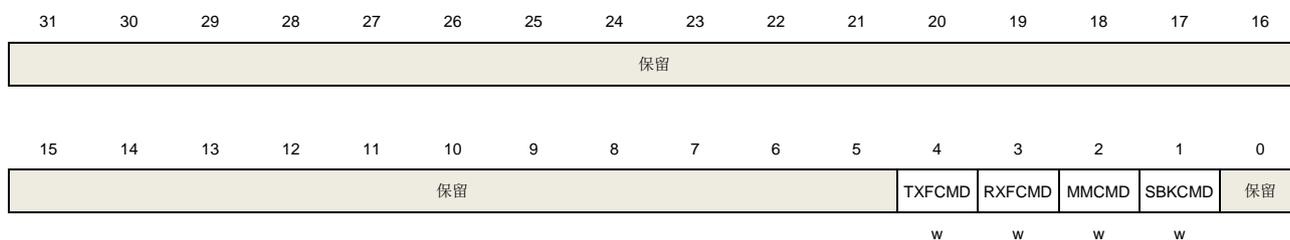
位/位域	名称	描述
31:24	BL[7:0]	<p>块长度</p> <p>这些位给出了智能卡T=1的接收时块的长度。它的值等于信息字节的长度+结束部分的长度(1-LEC/2-CRC)-1。</p> <p>这个值可以在块接收开始时设置(用于需要从块的序言提取块的长度的情形)，这个只在每一个接收时钟周期只能设置一次。在智能卡模式下，当TBE=0时，块的长度计数器被清0。</p> <p>在其他模式下，当REN=0(禁用接收器)并且 / 或者当EBC位被写1时块的长度计数器被清0。</p>
23:0	RT[23:0]	<p>接收器超时门限</p> <p>该位域指定接收超时值，单位是波特时钟的时长</p> <p>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。</p> <p>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。</p> <p>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。</p>

15.4.7. USART 请求寄存器 (USART_CMD)

地址偏移: 0x18

复位值: 0x0000 0000

该寄存器只能按字(32位)访问



位/位域	名称	描述
31:5	保留	必须保持复位值。
4	TXFCMD	<p>发送数据清空请求</p> <p>向该位写1去置位TBE标志位，以取消发送数据。</p>

		在USART1和USART2中，该位保留。
3	RXFCMD	接收数据清空请求 向该位写1来清除RBNE标志位，以丢弃未读的接收数据。
2	MMCMD	静默模式请求 向该位写1使USART进入静默模式并且置位RWU标志位。
1	SBKCMD	发送断开帧请求 向该位写1置位SBKF标志并使USART在空闲时发送一个断开帧。
0	保留	必须保持复位值。

15.4.8. USART 状态寄存器 (USART_STAT)

地址偏移: 0x1C

复位值: 0x0000 00C0

该寄存器只能按字 (32位) 访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留									REA	TEA	WUF	RWU	SBF	AMF	BSY
									r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留			EBF	RTF	CTS	CTSf	LBDF	TBE	TC	RBNE	IDLEF	ORERR	NERR	FERR	PERR
			r	r	r	r	r	r	r	r	r	r	r	r	r

位/位域	名称	描述
31:23	保留	必须保持复位值。
22	REA	接收使能通知标志 这位反映了USART核心逻辑的接收使能状态，该位可以通过硬件设置。 0: USART核心接收逻辑禁用 1: USART核心接收逻辑被使能
21	TEA	发送使能通知标志 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_INTIC寄存器中的WUC写1，该位被清0。 当UESM被清0时，该位清0。 在USART1和USART2中，该位保留。

19	RWU	<p>接收器从静默模式唤醒</p> <p>该位表示USART处于静默模式。</p> <p>0: 接收器在工作状态</p> <p>1: 接收器在静默状态</p> <p>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WAKE位选择。</p> <p>如果选择空闲信号唤醒,只能通过向USART_CMD寄存器的MMCMMD位写1来将该位置位。</p>
18	SBF	<p>断开信号发送标识</p> <p>0: 没发送断开字符</p> <p>1: 将要发送断开字符</p> <p>该位表示一个断开发送信号被请求。</p> <p>通过向USART_CMD寄存器的SBKCMD写1来置位。</p> <p>在断开帧的停止位发送期间,硬件清0。</p>
17	AMF	<p>ADDR匹配标志</p> <p>0: ADDR和接收到的字符不匹配</p> <p>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。</p> <p>当接收到ADDR [7:0]中定义的字符时,硬件置位。</p> <p>通过向USART_INTC寄存器的AMC写1清0。</p>
16	BSY	<p>忙标志</p> <p>0: USART处于空闲</p> <p>1: USART正在接收</p>
15:13	保留	<p>必须保持复位值。</p>
12	EBF	<p>块结束标志</p> <p>0: 块没有结束</p> <p>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。</p> <p>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。</p> <p>通过向USART_INTC寄存器的EBC写1清0。</p> <p>在USART1和USART2中,该位保留。</p>
11	RTF	<p>接收超时标志</p> <p>0: 尚未超时</p> <p>1: 已经超时,如果USART_CTL1寄存器的RTIE被置位,将会引发中断。</p> <p>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。</p> <p>通过向USART_INTC寄存器的RTC位写1清0。</p> <p>在智能卡模式,这个超时相当于CWT或BWT计时。</p> <p>在USART1和USART2中,该位保留。</p>
10	CTS	<p>CTS电平</p> <p>这个值等于nCTS输入引脚电平的反向拷贝。</p> <p>0: nCTS输入引脚高电平</p>

		1: nCTS输入引脚低电平
9	CTSF	<p>CTS变化标志</p> <p>0: nCTS状态线没有变化</p> <p>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。</p> <p>通过向USART_INTC寄存器的CTSC位写1，清零该位。</p>
8	LBDF	<p>LIN断开检测标志</p> <p>0: 没有检测到LIN断开字符</p> <p>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。</p> <p>当LIN断开帧被检测到的时候，硬件置位。</p> <p>通过向USART_INTC寄存器的LBDC位写1，清零该位。</p> <p>在USART1中，该位保留。</p>
7	TBE	<p>发送数据寄存器空</p> <p>0: 数据没有发送到移位寄存器</p> <p>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。</p> <p>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。</p> <p>通过向USART_TDATA寄存器中写数据来清0。</p>
6	TC	<p>发送完成</p> <p>0: 发送没有完成</p> <p>1: 发送完成。如果USART_CTL0寄存器的TCIE被置位，将会有中断产生。</p> <p>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。</p> <p>通过向USART_INTC寄存器的TCC位写1清0。</p>
5	RBNE	<p>读数据缓冲区非空</p> <p>0: 没有接收到数据</p> <p>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。</p> <p>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。</p> <p>通过读USART_RDATA寄存器或向USART_CMD寄存器的RXFCMD位写1清0。</p>
4	IDLEF	<p>空闲线检测标志</p> <p>0: 没检测到空闲线</p> <p>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。</p> <p>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。</p> <p>向USART_INTC寄存器的IDLEC位写1清0。</p>
3	ORERR	<p>溢出错误</p> <p>0: 未检测到溢出错误</p> <p>1: 检测到溢出错误。在多级缓存通信中，如果寄存器USART_CTL0的RBNEIE位置位，将会引发中断。如果寄存器USART_CTL2的ERRIE位置位也会引发中断。</p> <p>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，</p>

		将会由硬件置位。 向USART_INTC寄存器的OREC位写1清0。
2	NERR	<p>噪声错误标志</p> <p>0: 未检测到噪声错误</p> <p>1: 检测到噪声错误。在多级缓存通信中, 如果寄存器USART_CTL2的ERRIE位置位, 将会有中断产生。</p> <p>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。</p>
1	FERR	<p>帧错误</p> <p>0: 未检测到帧错误</p> <p>1: 检测到帧错误或者断开字符。在多级缓存通信中, 如果寄存器USART_CTL2的ERRIE位置位, 将会有中断产生。</p> <p>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。</p> <p>向USART_INTC寄存器的FEC位写1清0。</p>
0	PERR	<p>校验错误</p> <p>0: 未检测到校验错误</p> <p>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。</p> <p>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。</p>

15.4.9. 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	

位/位域	名称	描述
31:21	保留	必须保持复位值。
20	WUC	从深度睡眠模式唤醒标志的清除 向该位写1清除USART_STAT寄存器的WUF位。 在USART1和USART2中, 该位保留。

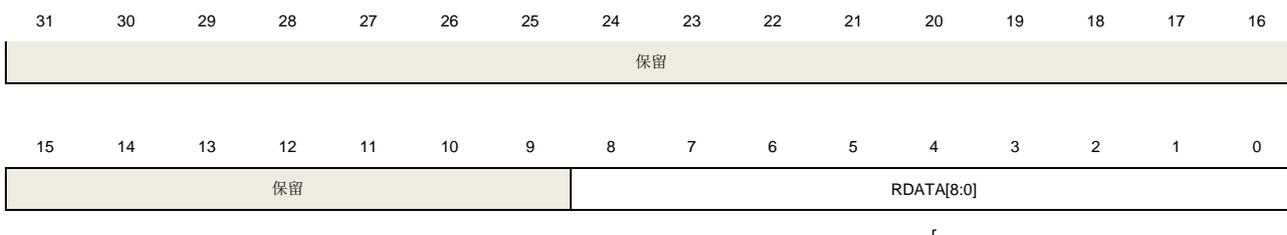
19:18	保留	必须保持复位值。
17	AMC	ADDR匹配标志清除 向该位写1清除USART_STAT寄存器的AMF位。
16:13	保留	必须保持复位值。
12	EBC	块结束标志清除 向该位写1清除USART_STAT寄存器的EBF位。 在USART1和USART2中，该位保留。
11	RTC	接收超时标志清除 向该位写1清除USART_STAT寄存器的RTF标志。 在USART1和USART2中，该位保留。
10	保留	必须保持复位值。
9	CTSC	CTS变化标志清除 向该位写1清除USART_STAT寄存器的CTSF位。
8	LBDC	LIN断开字符检测标志清除 向该位写1清除USART_STAT寄存器的LBDF标志位。 在USART1和USART2中，该位保留。
7	保留	必须保持复位值。
6	TCC	发送完成标志清除 向该位写1清除USART_STAT寄存器的TC位。
5	保留	必须保持复位值。
4	IDLEC	空闲线检测标志清除 向该位写1清除USART_STAT寄存器的IDLEF位。
3	OREC	溢出标志清除 向该位写1清除USART_STAT寄存器的ORERR位。
2	NEC	噪声检测清除 向该位写1清除USART_STAT寄存器的NERR位。
1	FEC	帧格式错误标志清除 向该位写1清除USART_STAT寄存器的FERR位。
0	PEC	校验错误标志清除 向该位写1清除USART_STAT寄存器的PERR位。

15.4.10. 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



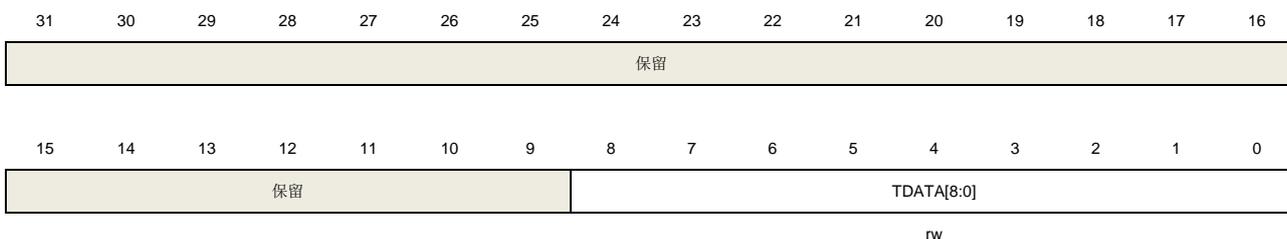
位/位域	名称	描述
31:9	保留	必须保持复位值。
8:0	RDATA[8:0]	接收数据的值 包含接收到的数据字节 如果接收到的数据打开了奇偶校验位（USART_CTL0寄存器的PCEN置1），那么接收到的数据的最高位（第7位或8位，取决于数据的长度）是奇偶校验位。

15.4.11. 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。

15.4.12. 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



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留							EPERR	保留							HCM
							rc_w0								rw

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。

15.4.13. 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。 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USART1和USART2中, 该位保留。

16. 内部集成电路总接口（I2C）

16.1. 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 协议的标准模式和快速模式以及快速+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。

16.2. 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。
- 支持标准模式（最高 100 kHz）和快速模式（最高 400 kHz）。
- I2C0 支持快速+模式（最高 1 MHz，该模式必须在 SYSCFG_CFG0 中使能）。
- 从机模式下可配置的 SCL 主动拉低。
- 支持 DMA 模式。
- I2C0 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，由深度睡眠模式和深度睡眠模式 1 唤醒。
- 独立于 PCLK 的时钟。

16.3. 功能说明

I2C 接口的内部结构如 [图 16-1. I2C 模块框图](#) 所示。

图 16-1. I2C 模块框图

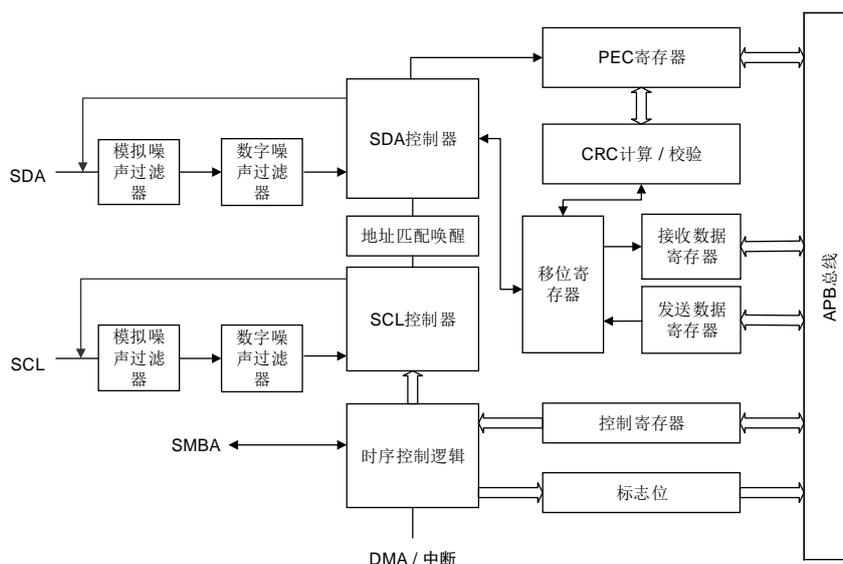


表 16-1. I2C 总线术语说明（参考飞利浦 I2C 规范）

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	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

16.3.1. 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：

- APB 时钟 PCLK（默认值）
- CK_IRC48MDIV_PER 时钟（默认为 IRC48M 时钟的三分频，可配置）
- 系统时钟 SYSCLK

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 260ns，数字滤波器产生的延时为 $DNF[3:0] \times t_{I2CCLK}$ 。

PCLK 时钟周期 t_{PCLK} 必须满足以下条件：

- $t_{PCLK} < 4/3 \times t_{SCL}$

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。

16.3.2. I2C 通讯流程

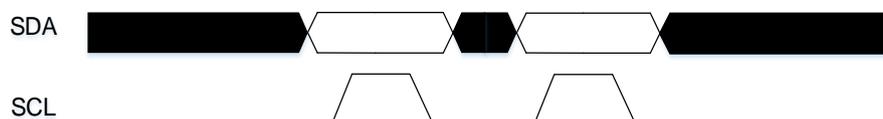
主机和从机都能实现数据收发，因此，I2C 可以实现四种工作模式：

- 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 16-2. 数据有效性](#)）。每个数据比特传输需要一个时钟脉冲。

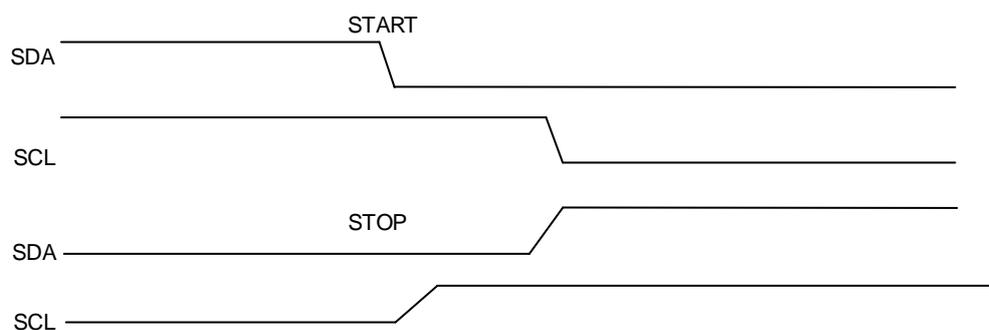
图 16-2. 数据有效性



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 16-3. 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 16-3. 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 16-4. 10 位地址的 I2C 通讯流程（主机发送）

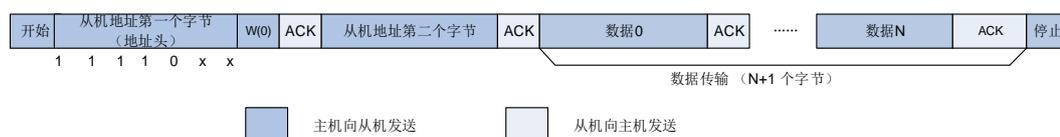


图 16-5. 7 位地址的 I2C 通讯流程（主机发送）



图 16-6. 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 16-7. 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 16-8. 10 位地址的 I2C 通讯流程（主机接收，HEAD10R=1）](#) 所示。

图 16-7. 10 位地址的 I2C 通讯流程（主机接收，HEAD10R=0）

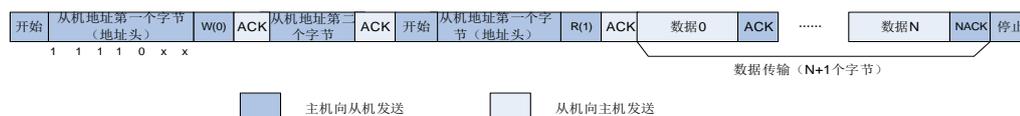
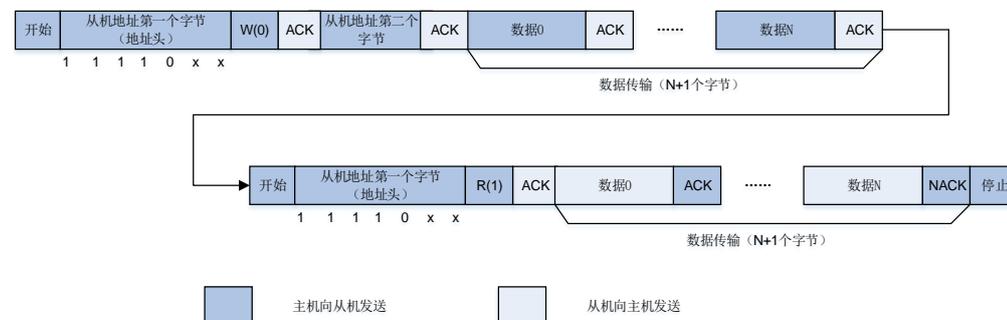


图 16-8. 10 位地址的 I2C 通讯流程（主机接收，HEAD10R=1）



16.3.3. 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，模拟滤波器可以抑制脉冲宽度高达 50ns 的峰值。

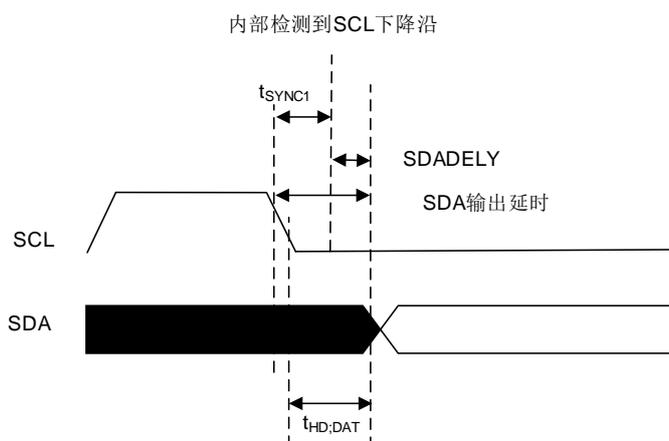
数字滤波器由 I2C_CTL0 寄存器中 DNF[3:0] 位来配置。当数字滤波器使能时，SCL 和 SDA 电平保持稳定的时间大于 $DNF[3:0] \times t_{I2CCLK}$ 才会发生内部变化。抑制峰值宽度可由 DNF[3:0] 配置。

16.3.4. 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。

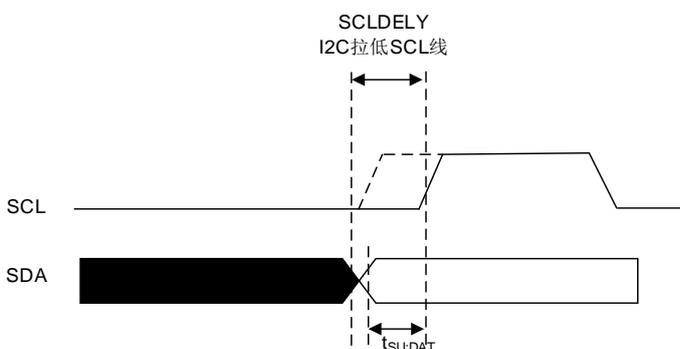
如果数据已经在 I2C_TDATA 寄存器中，在经历 SDADELY 延时后，数据由 SDA 发送，如 [图 16-9. 数据保持时间](#) 所示。

图 16-9. 数据保持时间



当数据经过 SDA 发送时，SCLDELY 计数器开启。如图 [图 16-10. 数据建立时间](#) 所示。

图 16-10. 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} 。

SDADELY 必须符合以下条件：

- $SDADELY \geq \{t_f(\max) + t_{HD;DAT}(\min) - t_{AF}(\min) - [(DNF + 3) * t_{I2CCLK}]\} / [(PSC + 1) * t_{I2CCLK}]$
- $SDADELY \leq \{t_{HD;DAT}(\max) - t_{AF}(\max) - [(DNF + 4) * t_{I2CCLK}]\} / [(PSC + 1) * t_{I2CCLK}]$

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：

- $SCLDELY \geq \{t_r(\max) + t_{SU;DAT}(\min)\} / [(PSC + 1) * t_{I2CCLK}] - 1$

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 $t_{PSC}=(PSC+1)*t_{I2CCCLK}$ 。 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 $t_{PSC}=(PSC+1)*t_{I2CCCLK}$ 。 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 16-2. 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	

16.3.5. 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:

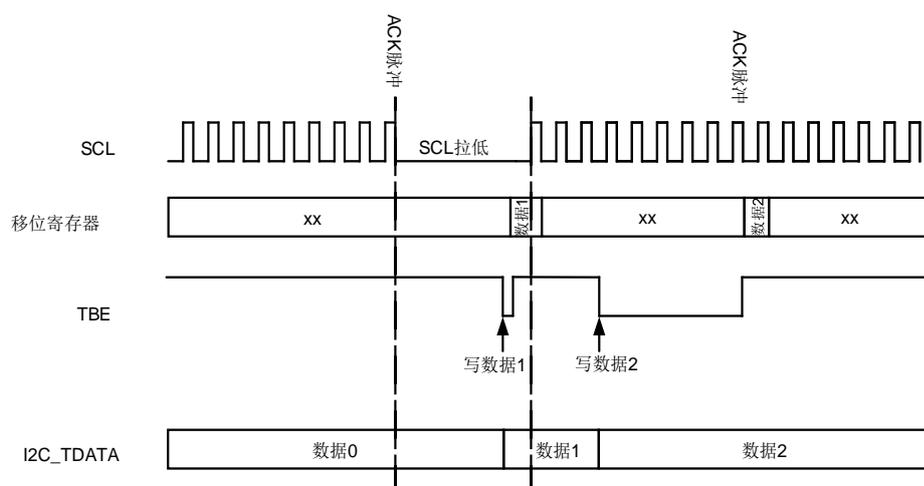
- I2CEN 写 0
- 检查 I2CEN 是否为 0
- I2CEN 写 1

16.3.6. 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。

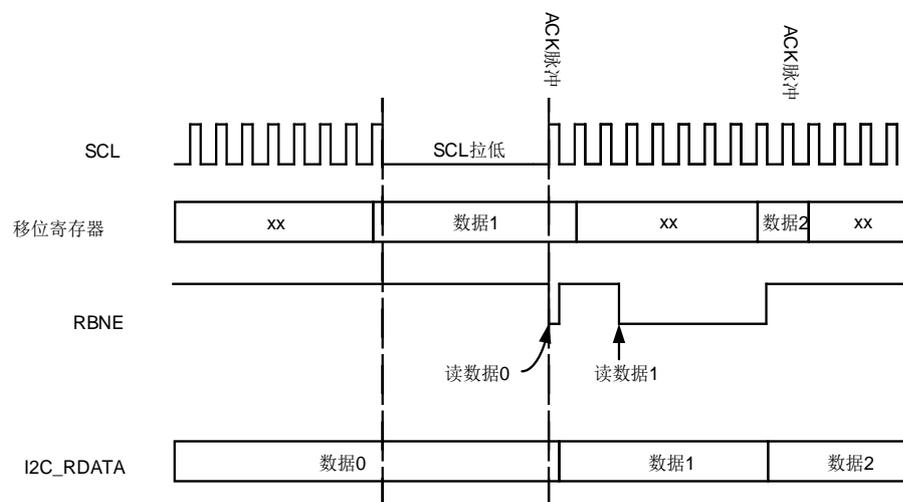
图 16-11. 数据发送



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 16-12. 数据接收



重载和自动结束模式

为了管理字节传输和关闭如 [表 16-3. 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 16-3. 可关闭通信模式

工作模式	行为
主机模式	产生 NACK, STOP 和 RESTART

工作模式	行为
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。

16.3.7. 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，ADDSSEND 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，ADDSSEND 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 $[(SDADELY+SCLDELY+1)*(PSC+1)+1]*t_{I2CCCLK}$ 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。

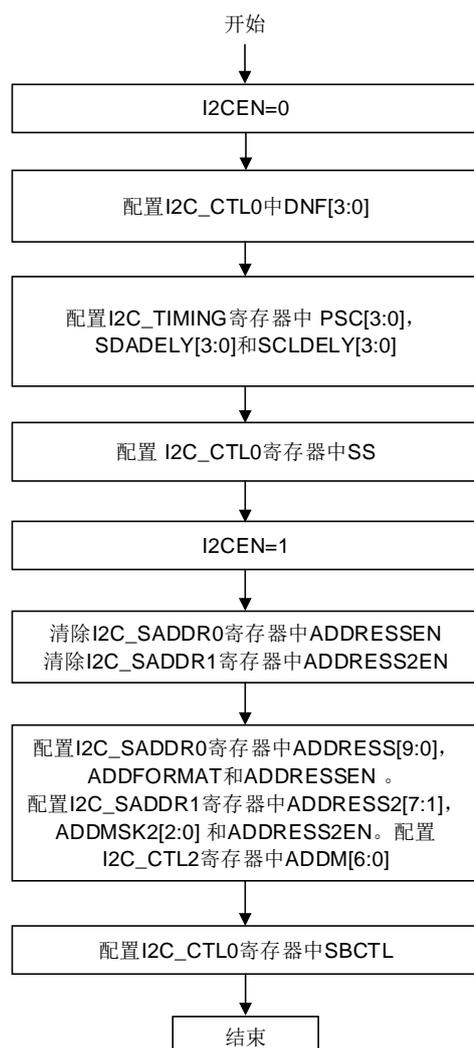
当 BYTENUM[7:0] 大于 0x1 时，在 BYTENUM[7:0] 数据接收期间，数据流是连续的。

注意：在下列情况下，可以配置 SBCTL 位：

- I2CEN = 0。
- 从机还未被寻址。
- ADDSEND = 1。

当 ADDSEND = 1，或者 TCR = 1 时，RELOAD 才可以被修改。

图 16-13. 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 16-14. I2C 从机发送编程模型 (SS=0)

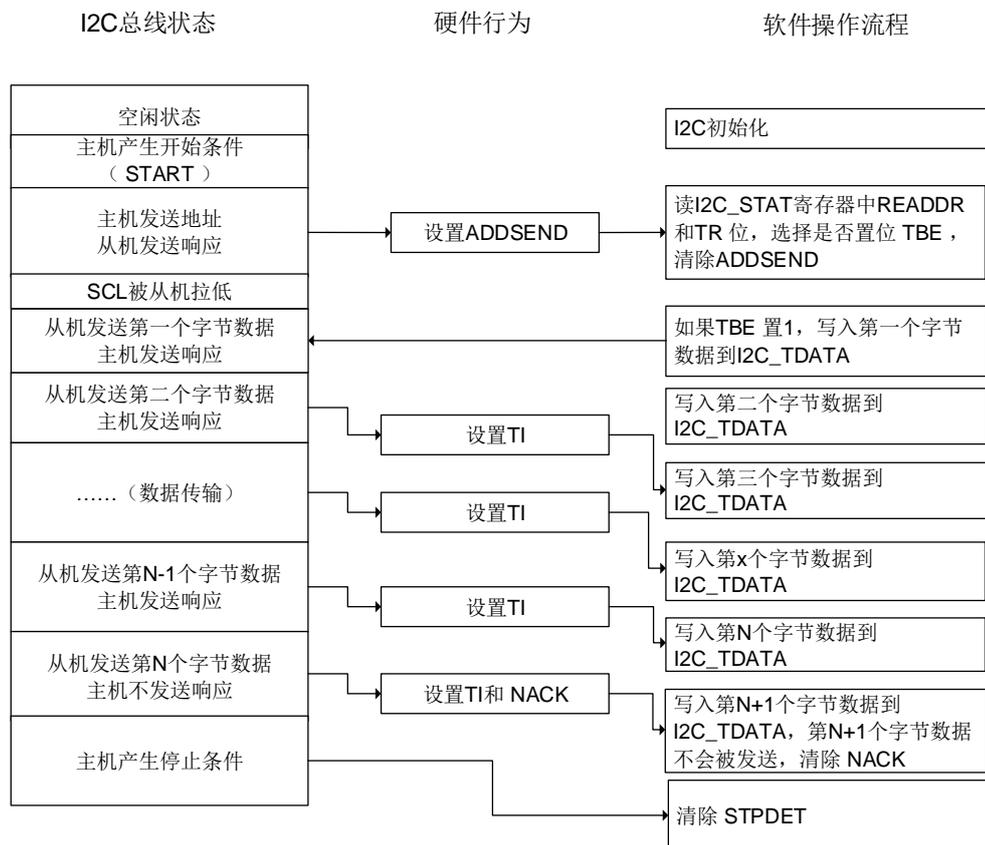
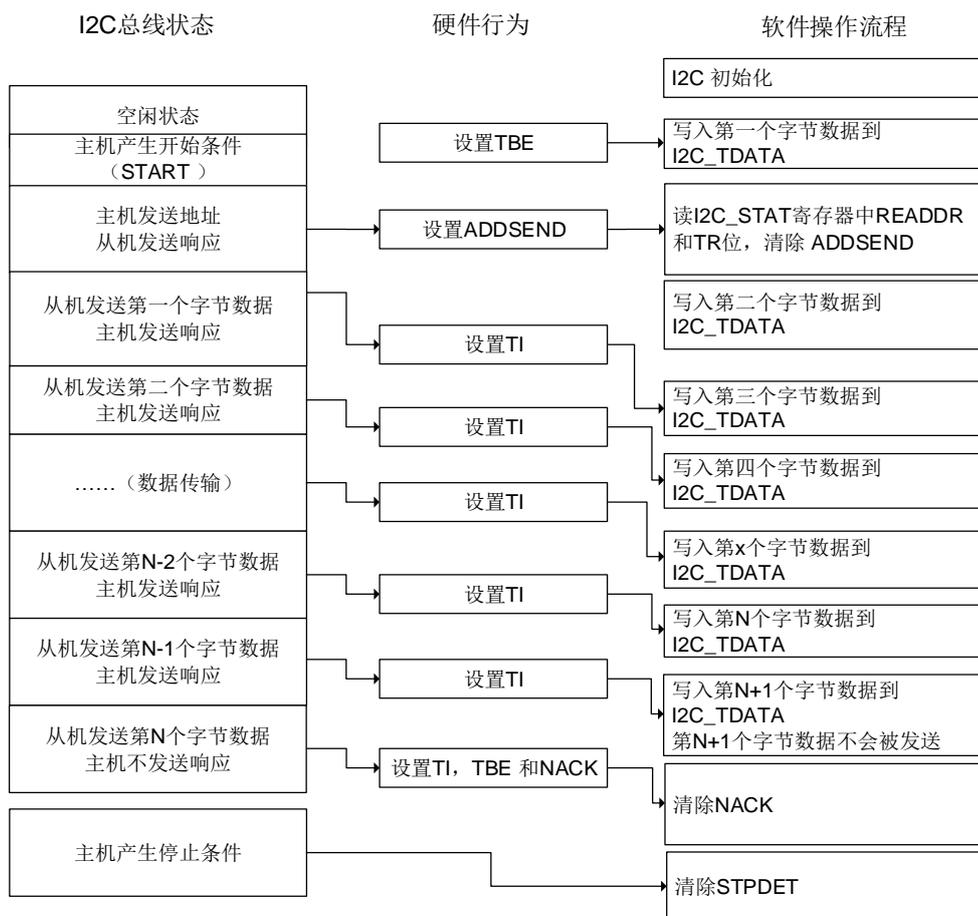


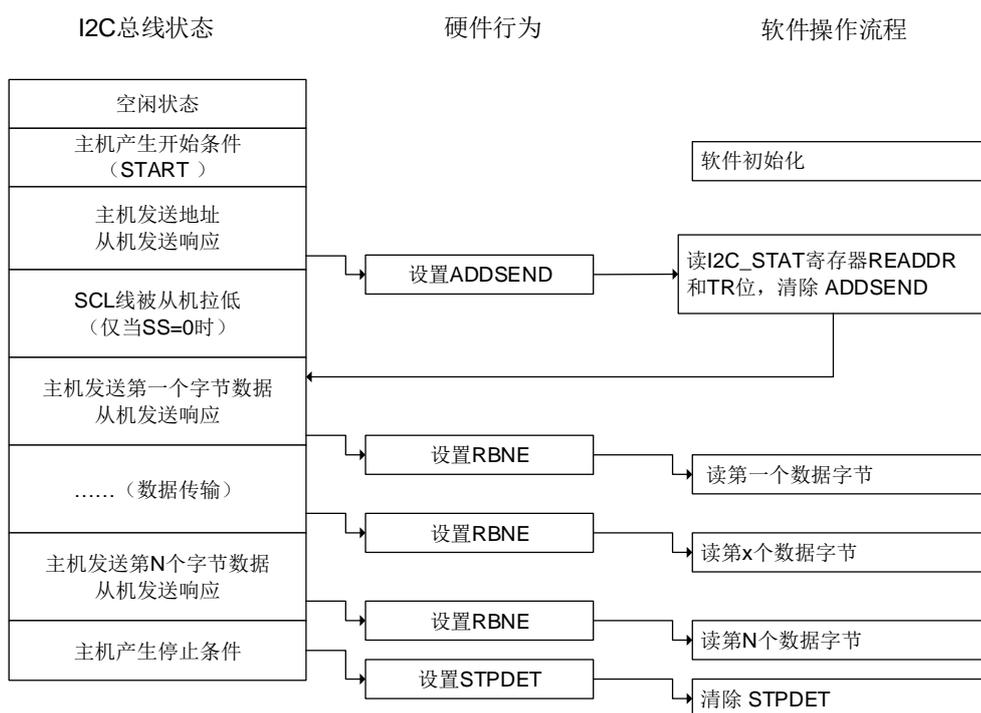
图 16-15. I2C 从机发送编程模型 (SS=1)



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 16-16. I2C 从机接收编程模型



16.3.8. 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: $t_{\text{SCL}} = t_{\text{SYNC1}} + t_{\text{SYNC2}} + \{[(\text{SCLH}[7:0] + 1) + (\text{SCLL}[7:0] + 1)] * (\text{PSC} + 1) * t_{\text{I2CCLK}}\}$ 。

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 $\text{DNF}[3:0] * t_{\text{I2CCLK}}$ 。

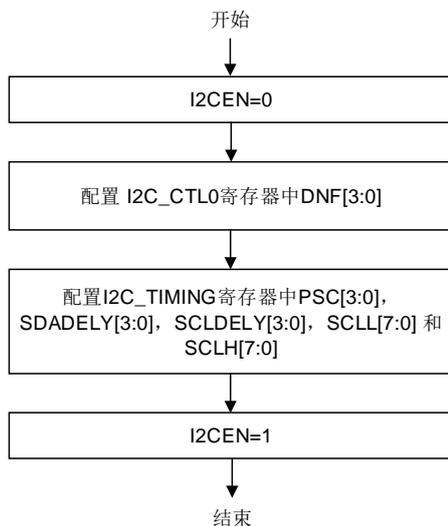
在主机模式下, 必须配置 I2C_CTL1 寄存器中 ADD10EN, SADDRESS[9:0]以及 TRDIR 位。当在主机接收模式下使用 10 位寻址时, 必须配置 HEAD10R 来选择是执行完整的地址寻址序列, 还是只发送地址头。待传输的字节数在 I2C_CTL1 寄存器 BYTENUM[7:0]配置。如果待传输的字节数大于或者等于 255, 必须将 BYTENUM[7:0]配置为 0xFF。然后主机发送 START 信号。以上提到的所有位必须在 START 位置 1 之前配置。START 信号发送完成之后, 待 I2C_STAT 寄存器 I2CBSY 位为 0 时, 发送从机地址。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。

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 ADDSEND 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 ADDSEND 置 1 时清零。

图 16-17. 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_CTL0 寄存器 BYTENUM[7:0]。如果发送字节数大于 255，必须通过将 I2C_CTL0 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 16-18. I2C 主机发送编程模型 (N<=255)

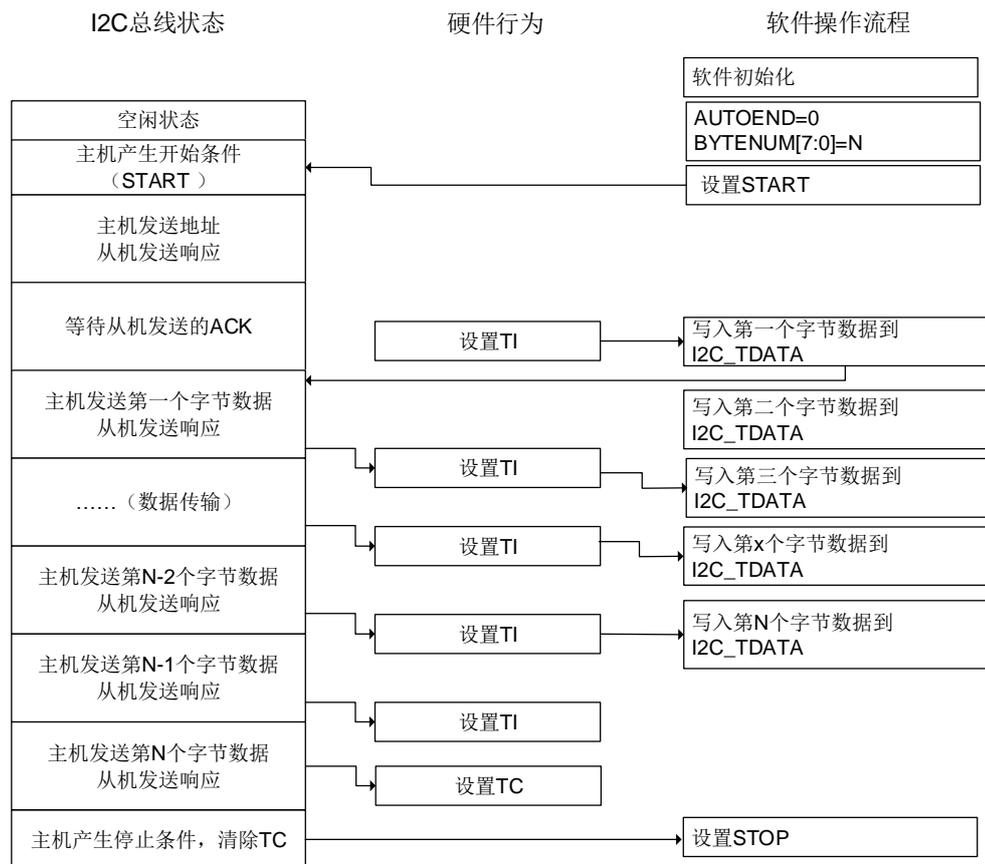


图 16-19. I2C 主机发送编程模型 (N>255)



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_CTL0 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 16-20. I2C 主机接收编程模型 (N<=255)

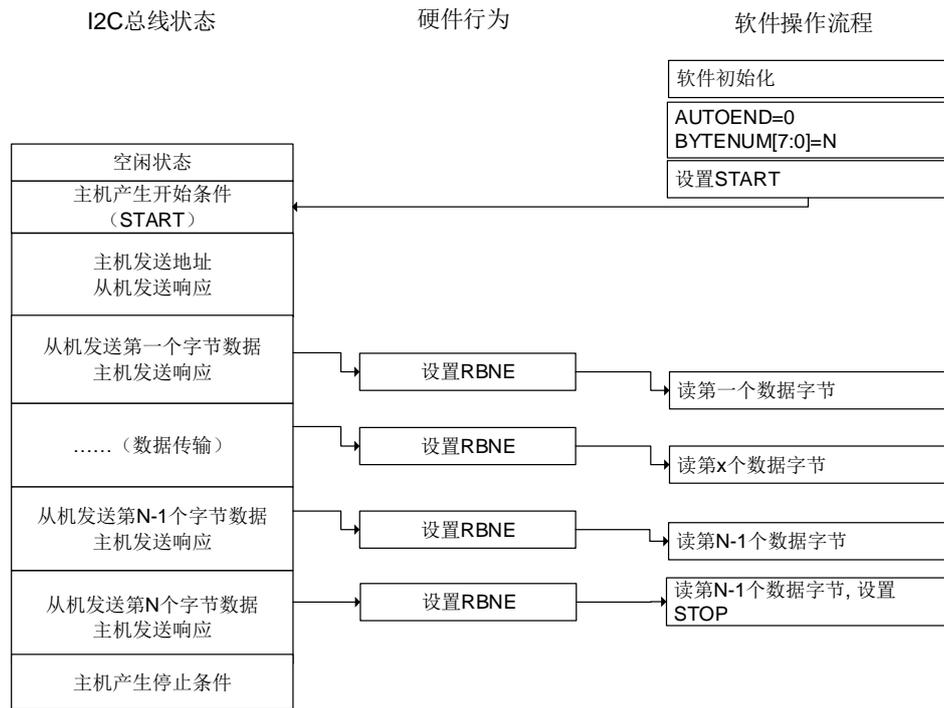


图 16-21. I2C 主机接收编程模型 (N>255)



16.3.9. SMBus 支持

系统管理总线 (System Management Bus, 简称为 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 10 kHz——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 $(\text{BUSTOA}+1)*2048*t_{\text{I2CCLK}}$ ，I2C_STAT 寄存器中 TIMEOUT 位将置 1。

BUSTOA[11:0]为从机校验 $t_{\text{LOW:SEXT}}$ ，为主机校验 $t_{\text{LOW:MEXT}}$ 。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 $(\text{BUSTOB}+1)*2048*t_{\text{I2CCLK}}$ ，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。

要启用 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)。

16.3.10. 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 16-22. 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：PECTRANS 位在 RELOAD 置 1 之后不能被修改。

图 16-23. SMBus 主机接收器和从机发送器通信流程



16.3.11. 从省电模式唤醒

当 I2C 地址匹配成功时，MCU 从深度睡眠模式和深度睡眠模式 1 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 CK_IRC48MDIV_PER。在深度睡眠模式和深度睡眠模式 1 下，CK_IRC48MDIV_PER 关闭。当 I2C 检测到 START 信号时，CK_IRC48MDIV_PER 打开，I2C 会将 SCL 拉低直到 CK_IRC48MDIV_PER 被唤醒。在接收地址期间，CK_IRC48MDIV_PER 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，CK_IRC48MDIV_PER 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，CK_IRC48MDIV_PER 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)。

16.3.12. 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。

16.3.13. I2C 错误和中断

I2C 错误标志如 [表 16-4. I2C 错误标志](#) 所示。

表 16-4. 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

I2C 错误名称	描述
SMBALT	SMBus 报警

I2C 中断和事件标志如 [表 16-5. I2C 中断事件](#) 所示。

表 16-5. 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	

16.3.14. 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 配置位选择继续正常工作还是停止工作。

16.4. I2C 寄存器

I2C0 基地址: 0x4000 5400

I2C1 基地址: 0x4000 5800

16.4.1. 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。

保留								PECEN	SMBALT EN	SMBDAE N	SMBHAE N	GCEN	WUEN	SS	SBCTL
								rw	rw	rw	rw	rw	rw	rw	rw
DENR	DENT	保留	ANOFF	DNF[3:0]				ERRIE	TCIE	STPDETI E	NACKIE	ADDMIE	RBNEIE	TIE	I2CEN
rw	rw		rw	rw				rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:24	保留	必须保持复位值。
23	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 功能, 则保留该位, 并由硬件设置为 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 功能, 则保留该位, 并由硬件设置为 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 功能, 则保留该位, 并由硬件设置为 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 功能, 则保留该位, 并由硬件设置为 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	<p>使能从省电模式中唤醒，包含深度睡眠模式和深度睡眠模式 1。 当 MCU 从省电模式唤醒时该位清零。 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。</p>
17	SS	<p>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。</p>
16	SBCTL	<p>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。</p>
15	DENR	<p>DMA 接收使能 0: DMA 接收禁能 1: DMA 接收使能</p>
14	DENT	<p>DMA 发送使能 0: DMA 发送禁能 1: DMA 发送使能</p>
13	保留	<p>必须保持复位值。</p>
12	ANOFF	<p>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。</p>
11:8	DNF[3:0]	<p>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。</p>
7	ERRIE	<p>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。</p>
6	TCIE	<p>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。</p>
5	STPDETIE	<p>停止信号检测中断使能</p>

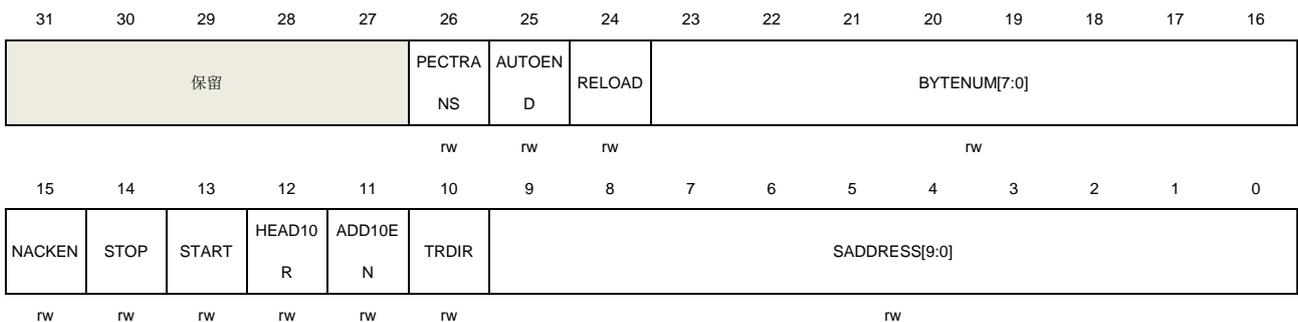
		0: 停止信号 (STPDET) 检测中断禁能。 1: 停止信号 (STPDET) 检测中断使能。
4	NACKIE	接收到 NACK 应答中断使能 0: 接收到 NACK 应答中断禁能。 1: 接收到 NACK 应答中断使能。
3	ADDMIE	从机模式下地址匹配中断使能 0: 地址匹配中断禁能。 1: 地址匹配中断使能。
2	RBNEIE	接收中断使能 0: 接收 (RBNE) 中断禁能。 1: 接收 (RBNE) 中断使能。
1	TIE	发送中断使能 0: 发送中断 (TI) 禁能。 1: 发送中断 (TI) 使能。
0	I2CEN	I2C 外设使能 0: I2C 禁能。 1: I2C 使能。

16.4.2. 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。



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

		功能，则保留该位，并由硬件设置为 0。
25	AUTOEND	<p>主机模式下自动结束模式</p> <p>0: 当 BYTENUM[7:0] 个字节传输完成后时，TC 位置 1。</p> <p>1: 当 BYTENUM[7:0] 个字节传输完成后时，自动发送 STOP 信号。</p> <p>注意： 该位仅在 RELOAD = 0 时有效。该位由软件置 1 和清 0。</p>
24	RELOAD	<p>重载模式使能</p> <p>0: 当 BYTENUM[7:0] 个字节传输完成后时，传输结束。</p> <p>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。</p> <p>注意： 该位由软件置 1 和清 0。</p>
23:16	BYTENUM[7:0]	<p>待传输的字节数</p> <p>这些用来编程待传输的字节数。当 SBCTL = 0 时，这些位无效。</p> <p>注意： 当 START 位置 1 时，这些位不能被修改。</p>
15	NACKEN	<p>从机模式下产生 NACK</p> <p>0: 在接收到新的字节时，发送 ACK。</p> <p>1: 在接收到新的字节时，发送 NACK。</p> <p>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。</p>
14	STOP	<p>I2C 总线上产生一个 STOP 结束信号。</p> <p>该位由软件置 1，并在 I2CEN = 0 或检测到 STOP 信号时由硬件清零。</p> <p>0: 不发送 STOP。</p> <p>1: 发送 STOP。</p>
13	START	<p>I2C 总线上产生一个 START 信号</p> <p>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 ADDSEND 位置 1 可以软件清除该位。</p> <p>0: 不发送 START。</p> <p>1: 发送 START。</p>
12	HEAD10R	<p>在主机接收模式下仅执行 10 位地址头读操作。</p> <p>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）。</p> <p>1: 主机寻址读序列为 RESTART + 10 位地址头（读）。</p> <p>注意： 当 START 位置 1 时，该位不能被修改。</p>
11	ADD10EN	<p>主机模式下使能 10 位寻址模式</p> <p>0: 主机工作在 7 位寻址模式下。</p> <p>1: 主机工作在 10 位寻址模式下。</p> <p>注意： 当 START 位置 1 时，该位不能被修改。</p>
10	TRDIR	<p>主机模式下传输方向</p>

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。

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。

16.4.3. 从机地址寄存器 0 (I2C_SADDR0)

地址偏移: 0x08

复位值: 0x0000 0000

该寄存器只能按字 (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：当 ADDRESSSEN = 1 时，该位不能被改写。

0 ADDRESS0 10 位地址的第 0 位
注意：当 ADDRESSSEN = 1 时，该位不能被改写。

16.4.4. 从机地址寄存器 1 (I2C_SADDR1)

地址偏移：0x0C

复位值：0x0000 0000

该寄存器只能按字（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:0]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： 当 ADDRESS2EN = 1 时，该位不能被改写。
0	保留	必须保持复位值。

16.4.5. 时序寄存器 (I2C_TIMING)

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
PSC[3:0]				保留				SCLDELY[3:0]				SDADELY[3:0]			
rw								rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCLH[7:0]								SCLL[7:0]							
rw								rw							

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。 $T_{SDADELY}=SDADELY*t_{PSC}$ 。
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。

16.4.6. 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 功能，则保留该寄存器，并由硬件设为 0x00000000。

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。

16.4.7. 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。

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

I2CBSY	保留	SMBALT	TIMEOUT	PECERR	OUERR	LOSTAR B	BERR	TCR	TC	STPDET	NACK	ADDSEN D	RBNE	TI	TBE	
r		r	r	r	r	r	r	r	r	r	r	r	r	r	rw	rw

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 引脚上检测到 SMBALERT 事件。 1: 在 SMBA 引脚上未检测到 SMBALERT 事件。 注意： 如果不支持 SMBus 功能，则保留该位，并由硬件设置为 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 功能，则保留该位，并由硬件设置为 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 功能，则保留该位，并由硬件设置为 0。

10	OUERR	<p>从模式下上溢 / 下溢错误</p> <p>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。</p> <p>0: 未发生上溢 / 下溢错误。</p> <p>1: 发生上溢 / 下溢错误。</p>
9	LOSTARB	<p>仲裁丢失</p> <p>LOSTARBC 置 1 可以将该位软件清零。当 $I2CEN = 0$ 时，该位由硬件清零。</p> <p>0: 无仲裁丢失。</p> <p>1: 发生仲裁丢失，I2C 模块返回从机模式。</p>
8	BERR	<p>总线错误</p> <p>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。</p> <p>0: 无总线错误。</p> <p>1: 发生了总线错误。</p>
7	TCR	<p>传输完成重载</p> <p>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。</p> <p>0: 当 $RELOAD = 1$ 时，$BYTENUM[7:0]$ 个字节传输未完成。</p> <p>1: 当 $RELOAD = 1$ 时，$BYTENUM[7:0]$ 个字节传输完成。</p>
6	TC	<p>主机模式下传输完成</p> <p>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。</p> <p>0: $BYTENUM[7:0]$ 个字节传输未完成。</p> <p>1: $BYTENUM[7:0]$ 个字节传输完成。</p>
5	STPDET	<p>总线上检测到 STOP 信号</p> <p>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。</p> <p>0: 未监测到 STOP 结束位。</p> <p>1: 监测到 STOP 结束位。</p>
4	NACK	<p>接收到 NACK 应答</p> <p>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。</p> <p>0: 接收到 ACK。</p> <p>1: 接收到 NACK。</p>
3	ADDSEND	<p>从机模式下接收到的地址与自身地址匹配</p> <p>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。</p> <p>0: 接收到的地址不匹配。</p> <p>1: 接收到的地址匹配。</p>
2	RBNE	<p>接收期间 I2C_RDATA 非空</p>

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	<p>发送中断</p> <p>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）。</p> <p>0: I2C_TDATA 非空或者 I2C 还未做好发送数据准备。</p> <p>1: I2C_TDATA 空且 I2C 已经做好发送数据准备。</p>
0	TBE	<p>发送期间 I2C_TDATA 空</p> <p>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。</p> <p>0: I2C_TDATA 非空。</p> <p>1: I2C_TDATA 空。</p>

16.4.8. 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。



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 功能，则保留该位，并由硬件设置为 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 功能，则保留该位，并由硬件设置为 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 功能，则保留该位，并由硬件设置为 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。

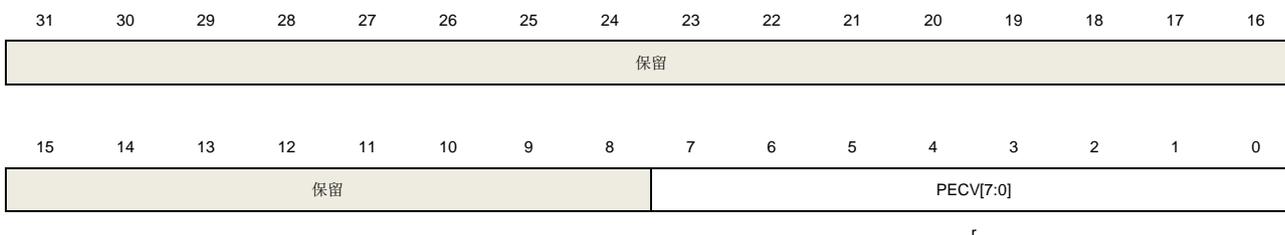
16.4.9. PEC 寄存器 (I2C_PEC)

地址偏移: 0x20

复位值: 0x0000 0000

注意: 如果不支持 SMBus 功能, 则保留该寄存器, 并由硬件设为 0x00000000。

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。

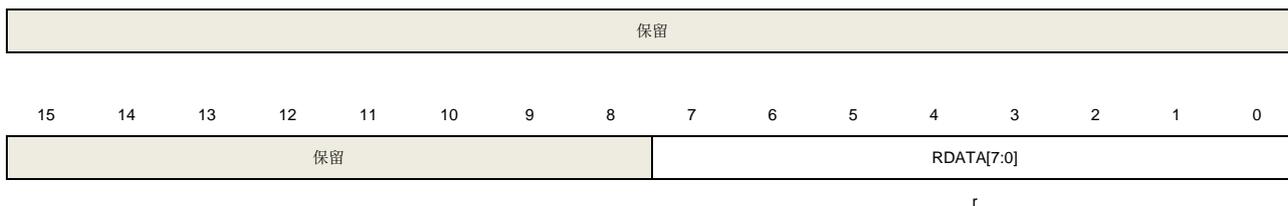
16.4.10. 接收数据寄存器 (I2C_RDATA)

地址偏移: 0x24

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问。





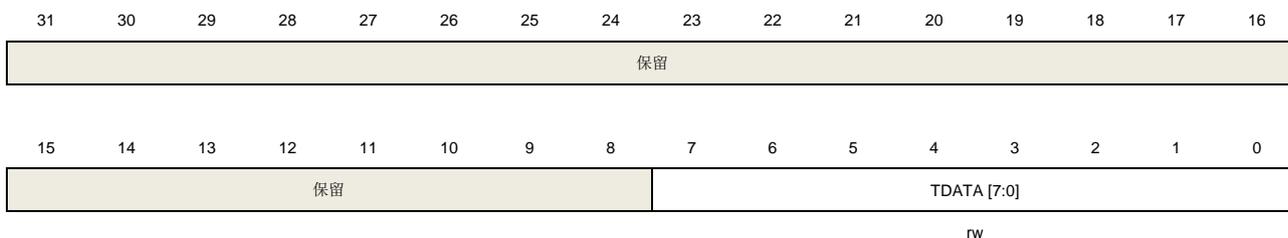
位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	RDATA[7:0]	接收到的数据

16.4.11. 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。



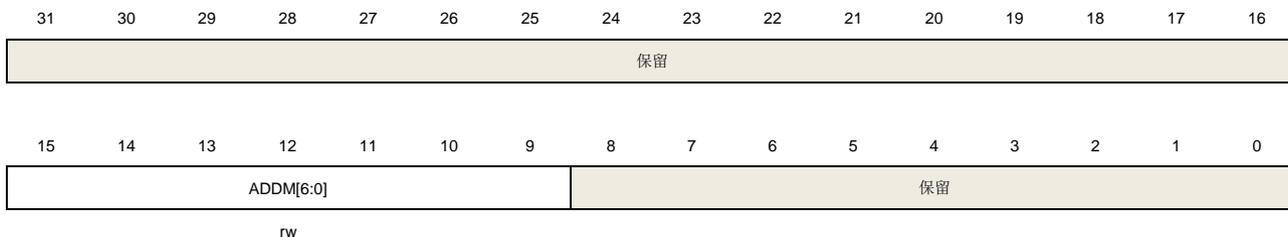
位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	TDATA[7:0]	发送的数据

16.4.12. 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。

17. 串行外设接口/片上音频接口（SPI/I2S）

17.1. 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。

17.2. 主要特征

17.2.1. SPI 主要特征

- 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）。

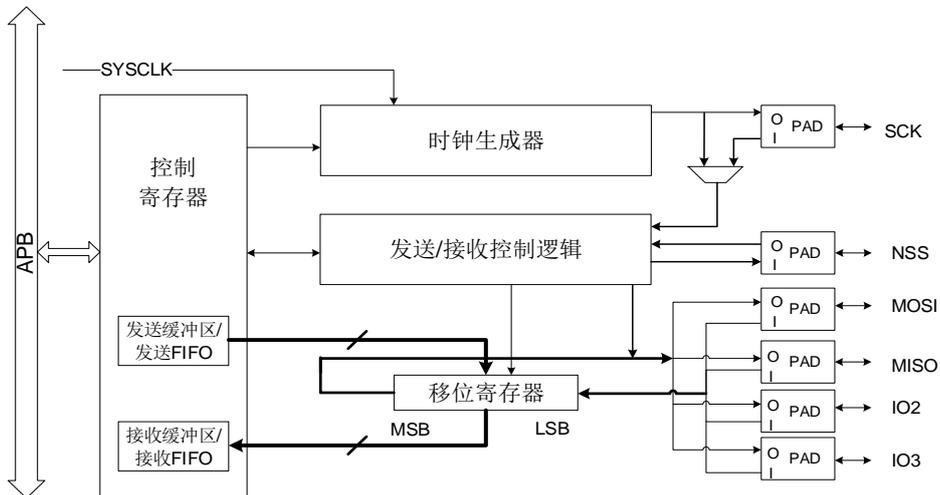
17.2.2. I2S 主要特性

- 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。
- 发送和接收支持DMA功能。

17.3. SPI 功能说明

17.4. SPI 结构框图

图 17-1. SPI 结构框图



17.4.1. SPI 信号线描述

常规配置（非 SPI 四线模式）

表 17-1. 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。

通过配置SPI_QCTL中的IO23_DRV位，在常规非四线SPI模式下，软件可以驱动IO2引脚和IO3引脚为高电平。

在SPI四线模式下，SPI通过以下6个引脚与外部设备连接：

表 17-2. 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

17.4.2. 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。

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。

图 17-2. SPI1 常规模式下的时序图

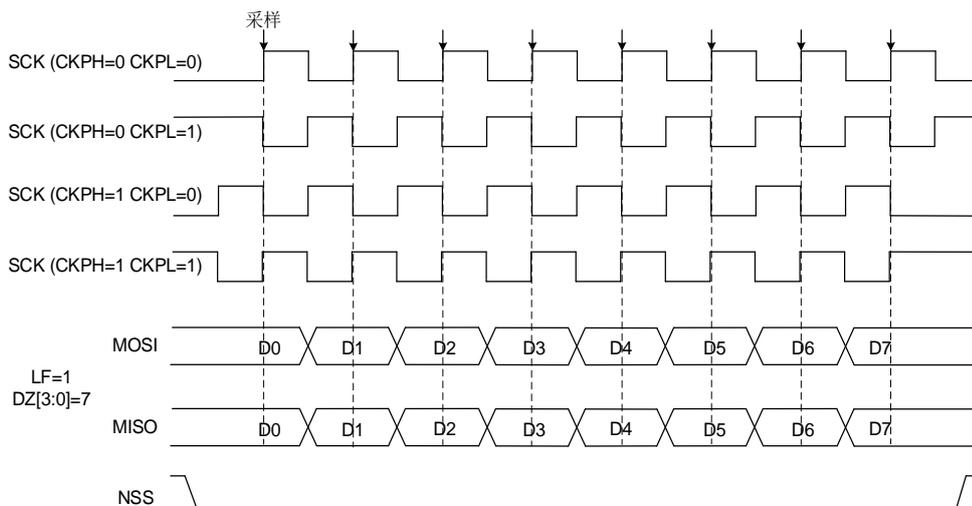
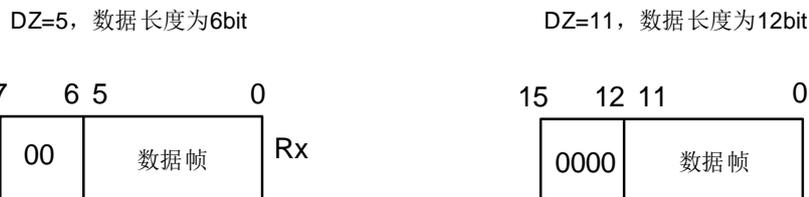


图 17-3. SPI1 数据帧右对齐示意图



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，SPI0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 17-4. SPI0 常规模式下的时序图

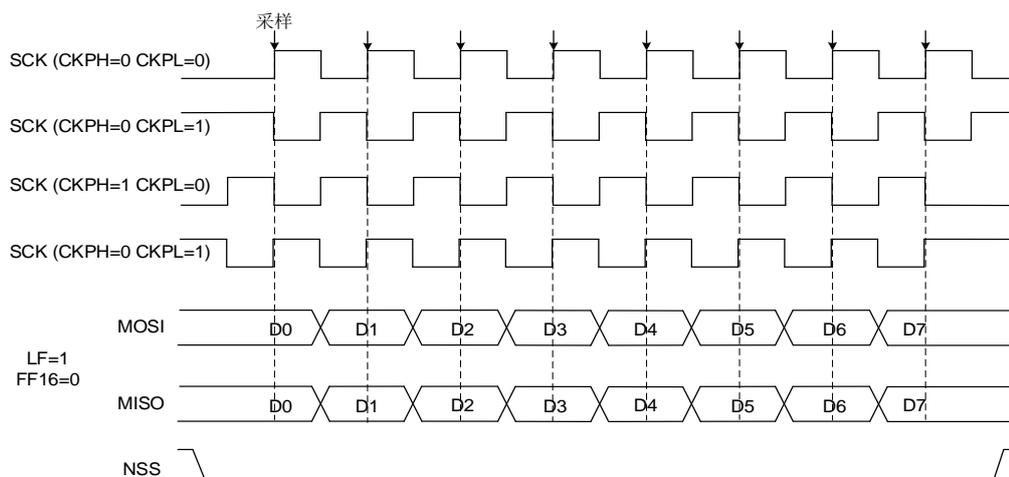
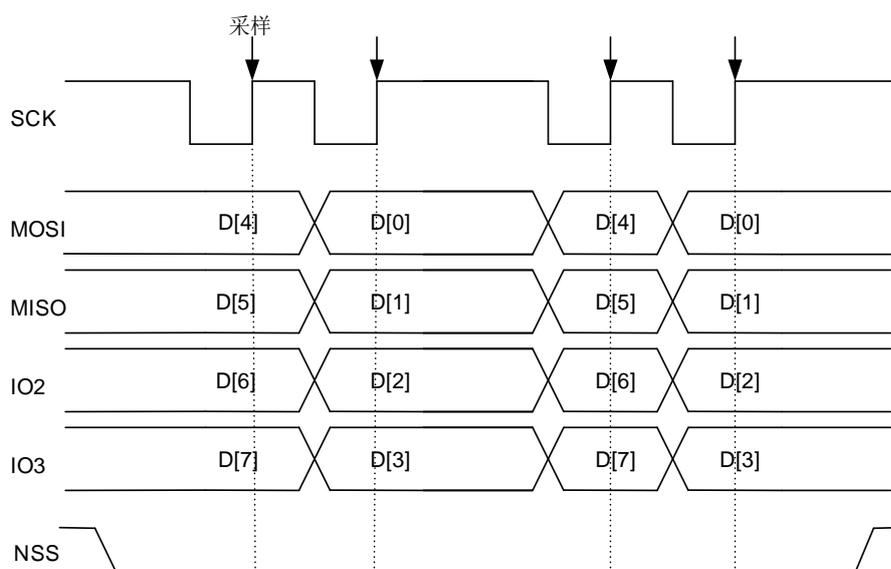


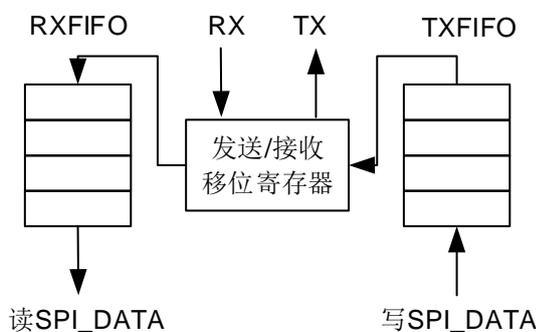
图 17-5. SPI1 四线模式下的 SPI 时序图 (CKPL=1, CKPH=1, LF=0)



17.4.3. 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 17-6. 发送/接收缓冲区



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。

17.4.4. NSS 功能

从机模式

当配置为从机模式（MSTMOD=0）时，在硬件NSS模式（SWNSSEN = 0）下，SPI从NSS引脚获取NSS电平，在软件NSS（SWNSSEN = 1）下，SPI根据SWNSS位得到NSS电平。只有当NSS为低电平时，发送或接收数据。在软件NSS模式下，不使用NSS引脚。

表 17-3. 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保持高电平，当发送或接收过程开始时，NSS变为低电平。当禁用SPI时，NSS变为高电平。

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 17-4. 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

17.4.5. SPI 运行模式

表 17-5. 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	MOSI: 接收 MISO: 不使用

模式	描述	寄存器配置	数据引脚用法
		BDOEN = 0	
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 17-7. 典型的全双工模式连接

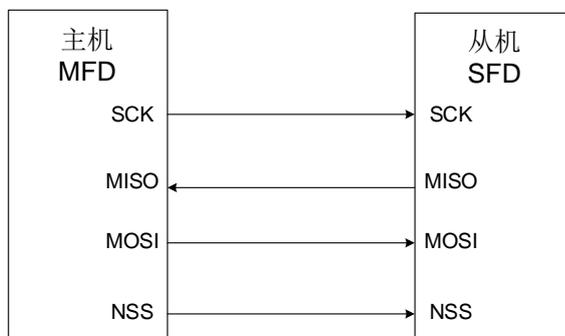


图 17-8. 典型的单工模式连接（主机：接收，从机：发送）

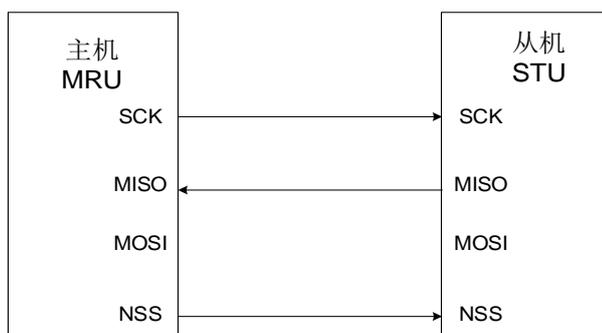


图 17-9. 典型的单工模式连接（主机：只发送，从机：接收）

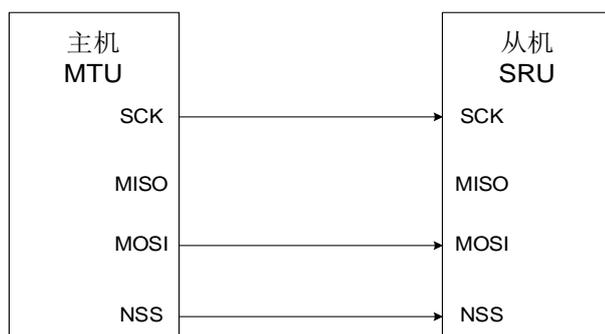
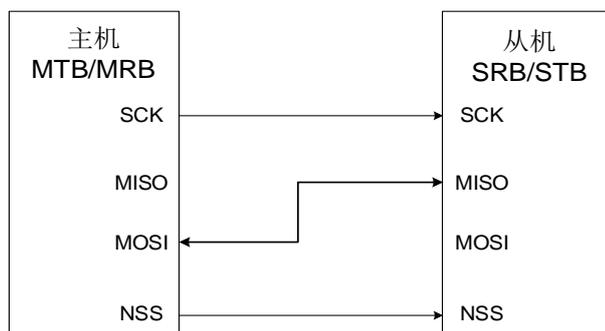


图 17-10. 典型的双向线连接



SPI 初始化流程

SPI1:

- 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。
- 配置时钟时序（SPI_CTL0中的CKPL位和CKPH位）。
- 配置帧格式（SPI_CTL0中的LF位）。
- 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）。
- 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）。
- 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。
- 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。
- 根据[表17-5. SPI运行模式](#)，配置MSTMOD位、RO位、BDEN位和BDOEN位。
- 根据应用程序的需求，配置TXDMA_ODD和RXDMA_ODD位。
- 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。
- 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：

- 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17-5. 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。

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 17-11. 主机 TI 模式在不连续发送时的时序图

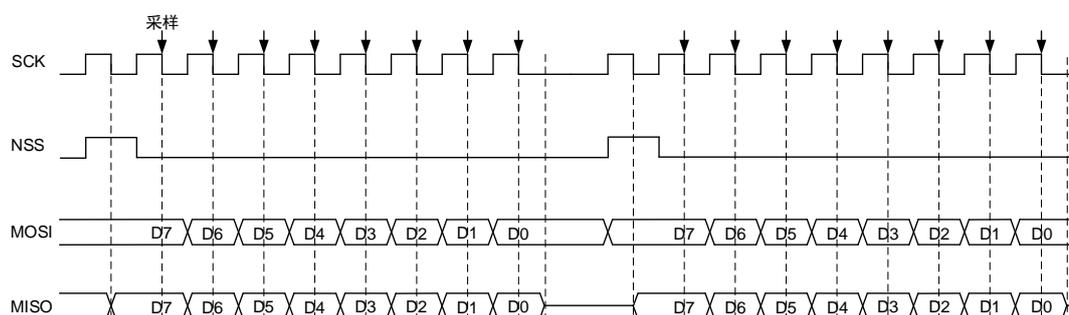
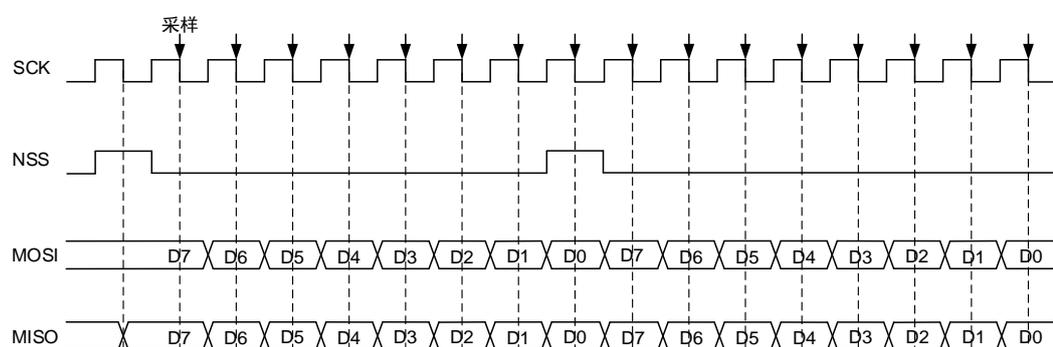
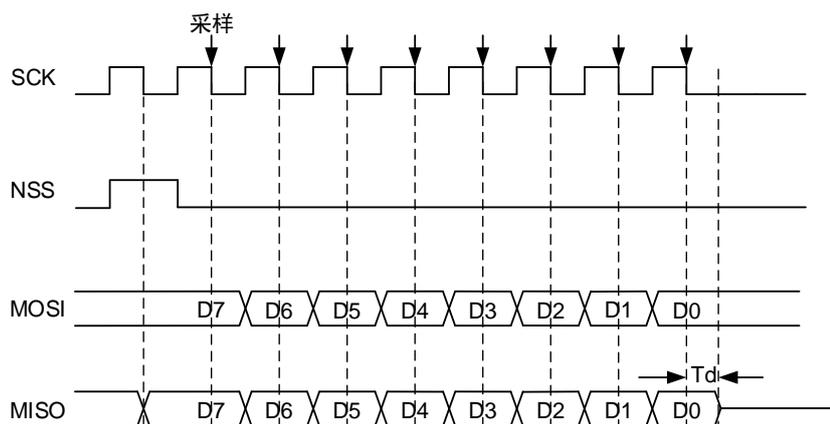


图 17-12. 主机 TI 模式在连续发送时的时序图



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 17-13. 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。

$$T_d = \frac{T_{bit}}{2} + 5 * T_{pclk} \quad (18-1)$$

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

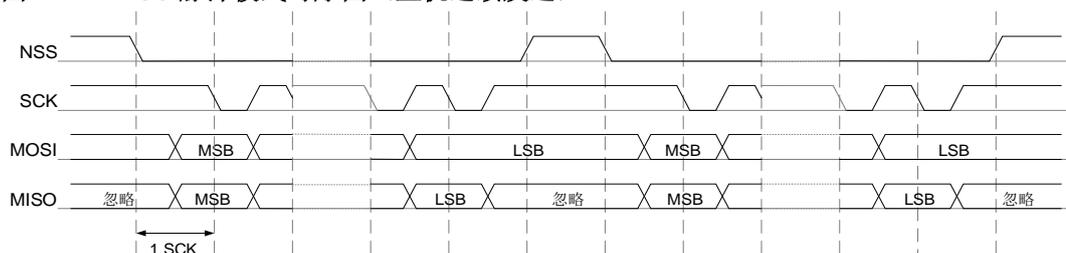
配置SPI_CTL1寄存器中的NSSP位使能该功能，为了确保使用该功能实现，需满足以下几个条件：配置设备为主机模式，使用普通SPI协议的数据帧格式，同时在第一个时钟跳变沿采样数据。

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 17-14. 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。

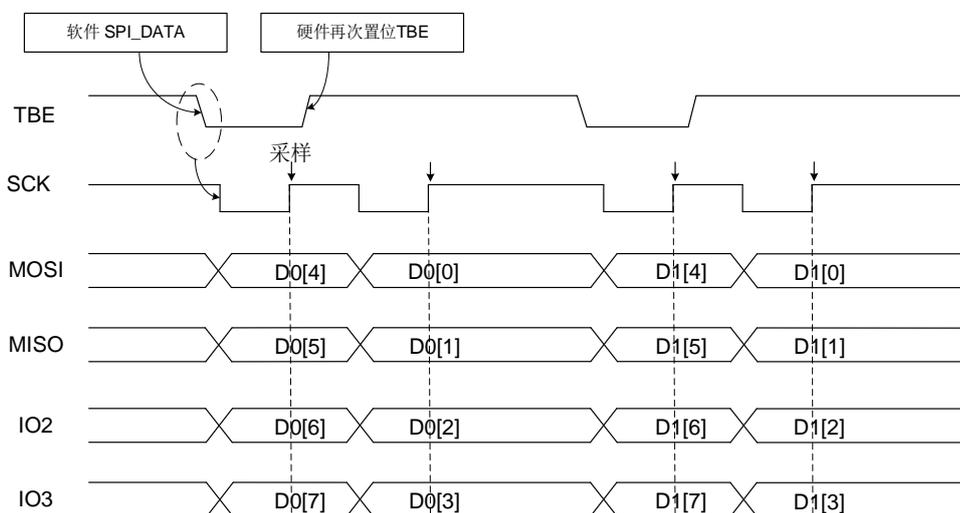
四线写模式

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 17-15. SPI 四线模式四线写操作时序图



四线读模式

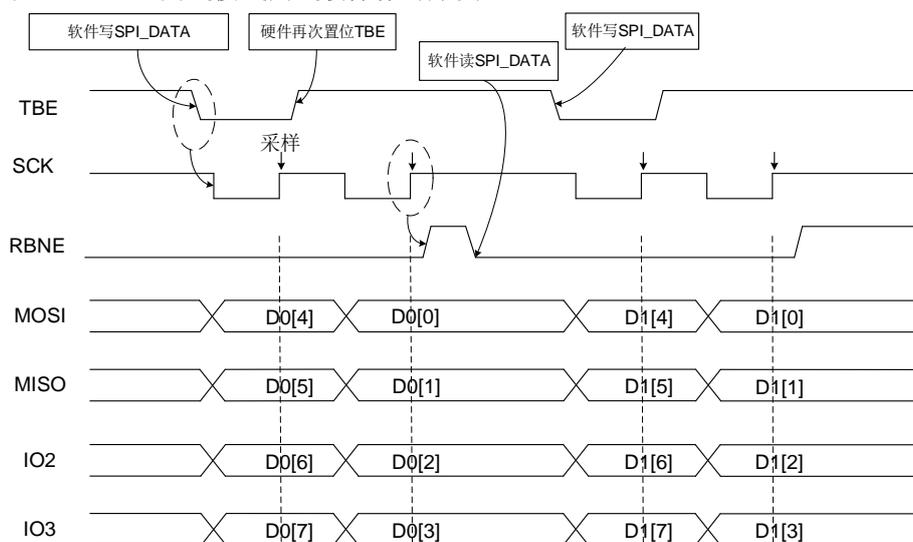
当SPI_QCTL寄存器中的QMOD位和QRD位都置1时，SPI工作在四线读模式。在四线读模式中，MOSI、MISO、IO2和IO3都用作输入引脚，一旦数据写入SPI_DATA寄存器（TBE位清零）且SPIEN位置1时，在SCK信号线产生时钟信号。写数据到SPI_DATA寄存器只是为了产生SCK时钟信号，所以可以写入任何数据。SPI开始数据传输之后，每发送一个数据帧都要检测SPIEN

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 17-16. 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。

17.4.6. DMA 功能

DMA功能在传输过程中将应用程序从数据读写过程中释放出来，从而提高了系统效率。

通过置位SPI_CTL1寄存器中的DMATEN位和DMAREN位，使能SPI模式的DMA功能。为了使用DMA功能，软件首先应当正确配置DMA模块，然后通过初始化流程配置SPI模块，最后使能SPI。

SPI使能后，如果DMATEN位置1，每当TBE=1时，SPI将会发出一个DMA请求，然后DMA应答该请求，并自动写数据到SPI_DATA寄存器。如果DMAREN位置1，每当RBNE=1时，发出一个DMA请求，然后DMA应答该请求，并自动从SPI_DATA寄存器读取数据。

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。

17.4.7. CRC 功能

SPI模块包含两个CRC计算单元：分别用于发送数据和接收数据。CRC计算单元使用SPI_CRCPOLY寄存器中定义的多项式。

通过配置SPI_CTL0中的CRCEN位使能CRC功能。对于数据线上每个发送和接收的数据，CRC单元逐位计算CRC值，计算得到的CRC值可以从SPI_TCRC寄存器和SPI_RCRC寄存器中读取。

为了传输计算得到的CRC值，应用程序需要在最后一个数据写入发送缓冲区之后，设置SPI_CTL0中的CRCNT位。在全双工模式（MFD或SFD），当SPI发送一个CRC值并且准备校验接收到的CRC值时，会将最新接收到的数据当作CRC值。在接收模式（MRB, MRU, SRU和SRB）下，在倒数第二个数据帧被接收后，软件将CRCNT位置1。在CRC校验失败时，CRCERR错误标志位将会置1。

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。

17.4.8. SPI 中断

状态标志位

■ 发送缓冲区空标志位（TBE）

当发送缓冲区为空或当前发送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SPI0, 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SPI1, 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 17-6. SPI 中断请求

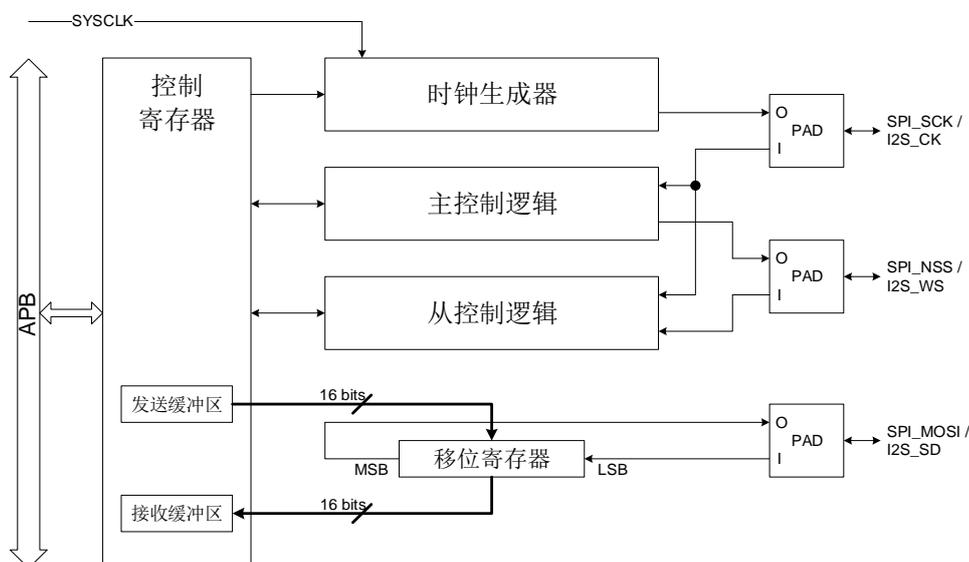
中断事件	描述	清除方式	中断使能位
TBE	发送缓冲区/发送FIFO空	写SPI_DATA寄存器	TBEIE
RBNE	接收缓冲区/接收FIFO非空	读SPI_DATA寄存器	RBNEIE
CONFERR	配置错误	读或写 SPI_STAT 寄存器, 然后写 SPI_CTL0 寄存器	ERRIE

中断事件	描述	清除方式	中断使能位
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	

17.5. I2S 功能说明

17.5.1. I2S 结构框图

图 17-17. 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。

17.5.2. I2S 信号线描述

I2S接口有3个引脚，分别是I2S_CK、I2S_WS和I2S_SD。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。

17.5.3. 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 17-18. I2S 飞利浦标准时序图 (DTLEN=00, CHLEN=0, CKPL=0)

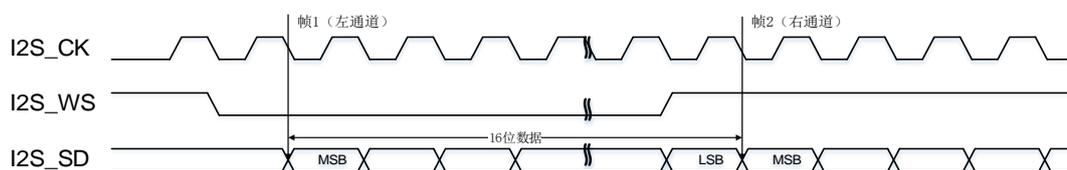
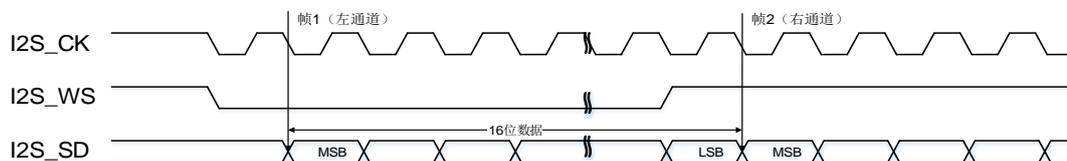


图 17-19. I2S 飞利浦标准时序图 (DTLEN=00, CHLEN=0, CKPL=1)



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 17-20. I2S 飞利浦标准时序图 (DTLEN=10, CHLEN=1, CKPL=0)

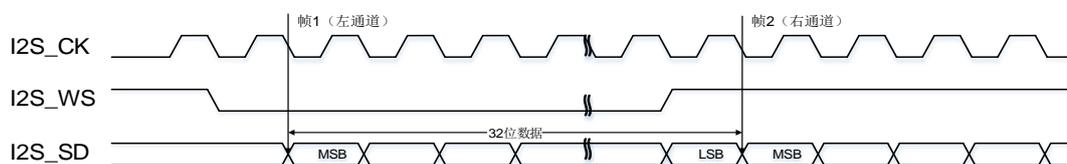
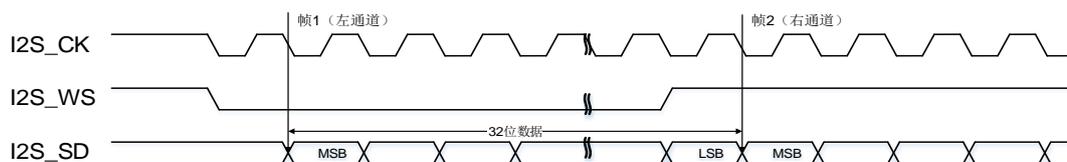


图 17-21. I2S 飞利浦标准时序图 (DTLEN=10, CHLEN=1, CKPL=1)



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 17-22. I2S 飞利浦标准时序图 (DTLEN=01, CHLEN=1, CKPL=0)

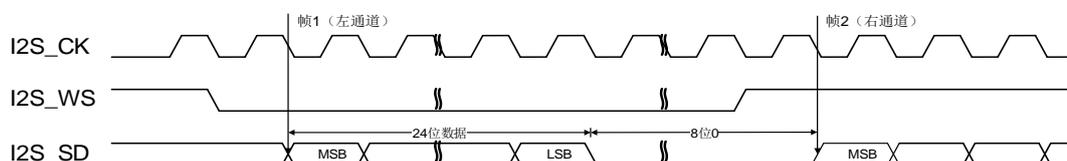
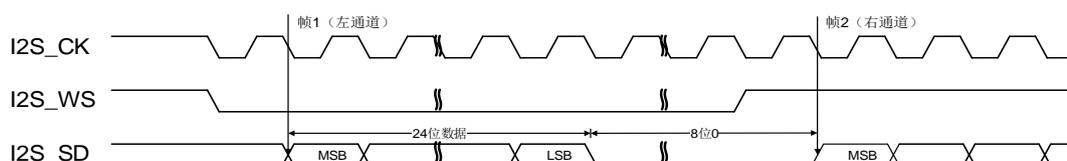


图 17-23. I2S 飞利浦标准时序图 (DTLEN=01, CHLEN=1, CKPL=1)



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 17-24. I2S 飞利浦标准时序图 (DTLEN=00, CHLEN=1, CKPL=0)

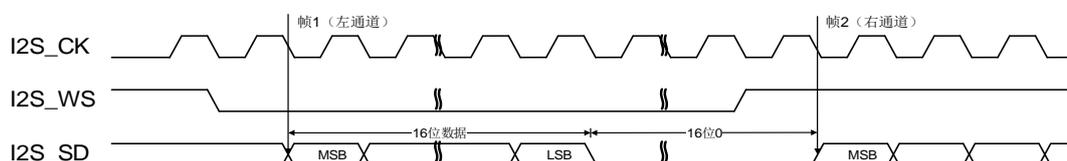
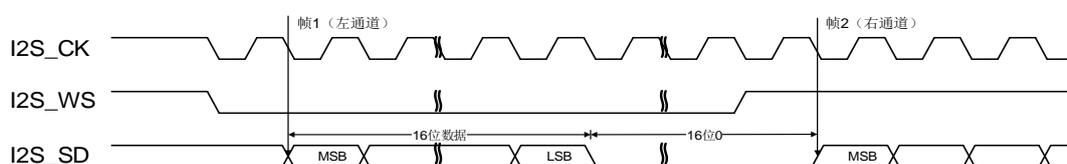


图 17-25. I2S 飞利浦标准时序图 (DTLEN=00, CHLEN=1, CKPL=1)



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 17-26. MSB 对齐标准时序图 (DTLEN=00, CHLEN=0, CKPL=0)

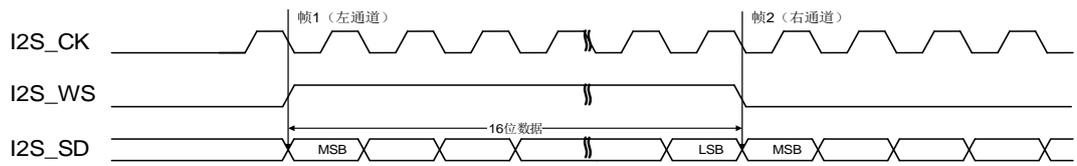


图 17-27. MSB 对齐标准时序图 (DTLEN=00, CHLEN=0, CKPL=1)

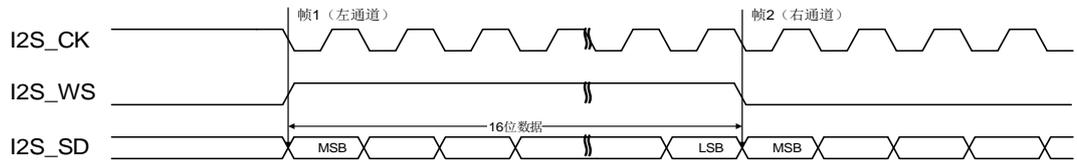


图 17-28. MSB 对齐标准时序图 (DTLEN=10, CHLEN=1, CKPL=0)

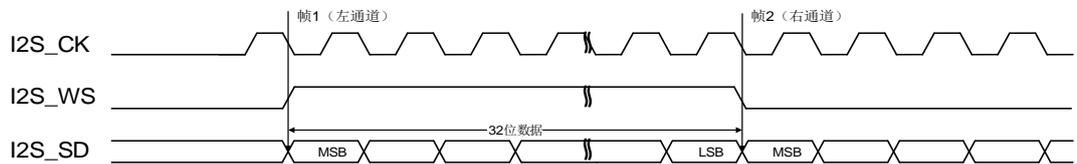


图 17-29. MSB 对齐标准时序图 (DTLEN=10, CHLEN=1, CKPL=1)

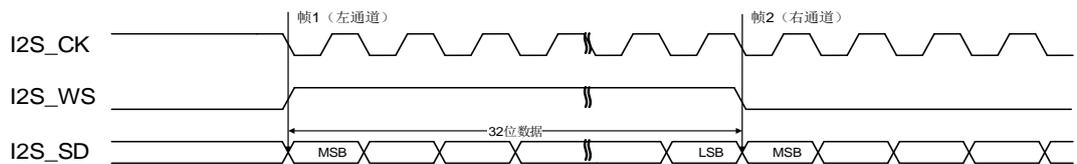


图 17-30. MSB 对齐标准时序图 (DTLEN=01, CHLEN=1, CKPL=0)

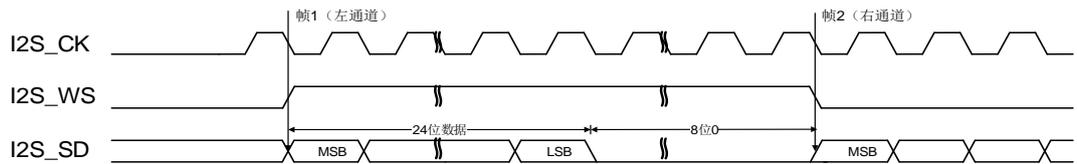


图 17-31. MSB 对齐标准时序图 (DTLEN=01, CHLEN=1, CKPL=1)

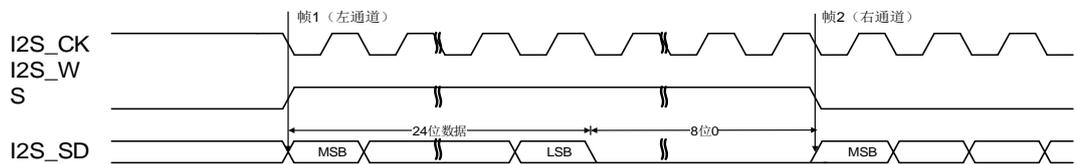


图 17-32. MSB 对齐标准时序图 (DTLEN=00, CHLEN=1, CKPL=0)

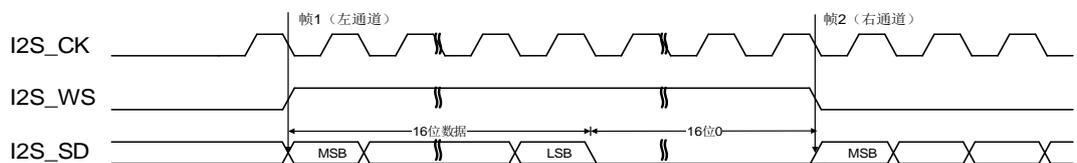
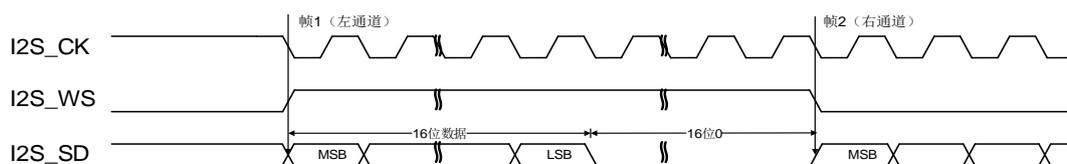
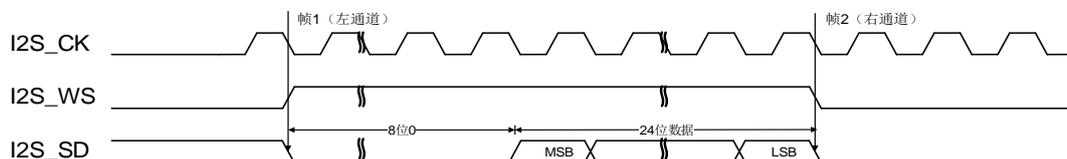
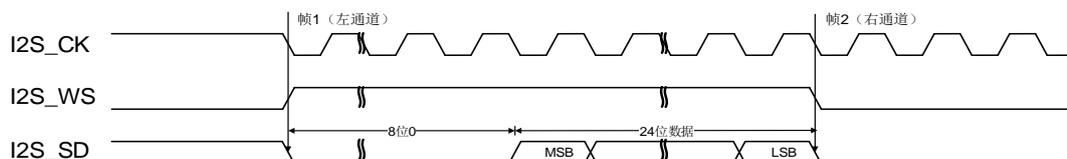


图 17-33. MSB 对齐标准时序图 (DTLEN=00, CHLEN=1, CKPL=1)


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 17-34. LSB 对齐标准时序图 (DTLEN=01, CHLEN=1, CKPL=0)

图 17-35. 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]。

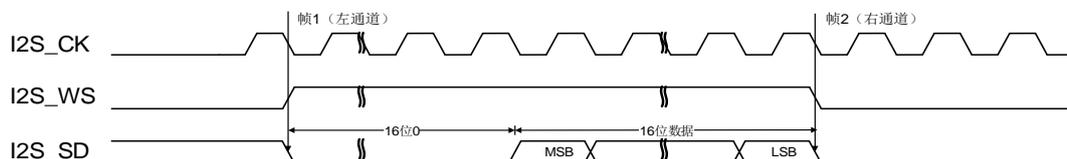
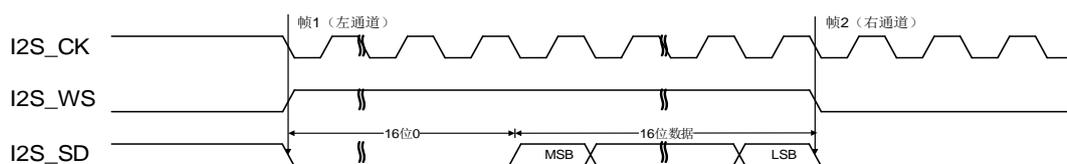
图 17-36. LSB 对齐标准时序图 (DTLEN=00, CHLEN=1, CKPL=0)


图 17-37. 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。

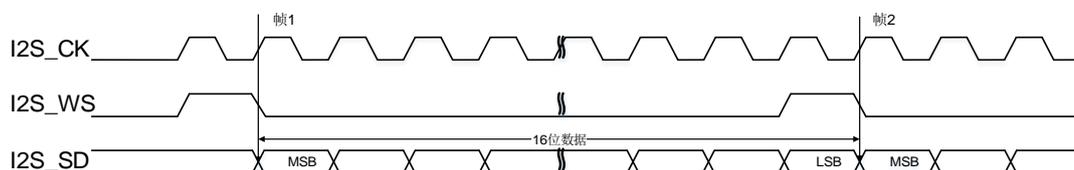
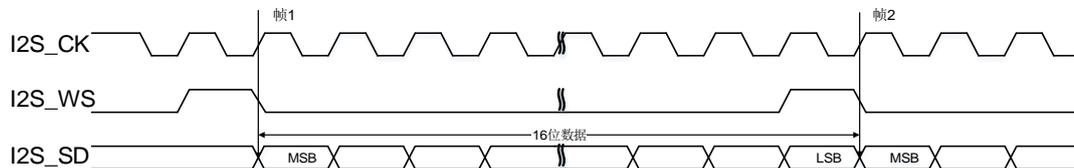
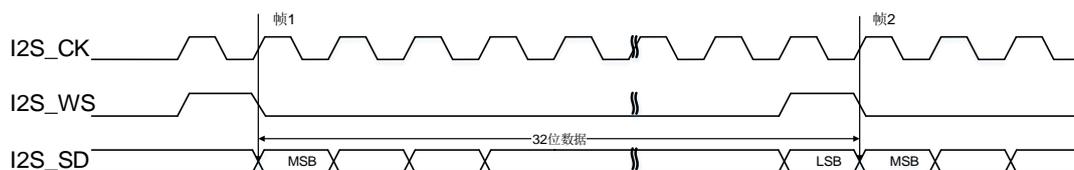
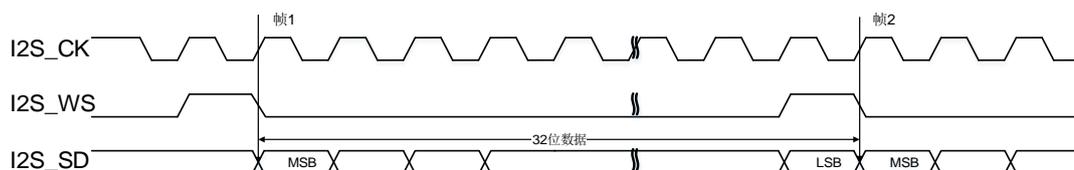
图 17-38. PCM 标准短帧同步模式时序图 (DTLEN=00, CHLEN=0, CKPL=0)

图 17-39. PCM 标准短帧同步模式时序图 (DTLEN=00, CHLEN=0, CKPL=1)

图 17-40. PCM 标准短帧同步模式时序图 (DTLEN=10, CHLEN=1, CKPL=0)

图 17-41. PCM 标准短帧同步模式时序图 (DTLEN=10, CHLEN=1, CKPL=1)


图 17-42. PCM 标准短帧同步模式时序图 (DTLEN=01, CHLEN=1, CKPL=0)

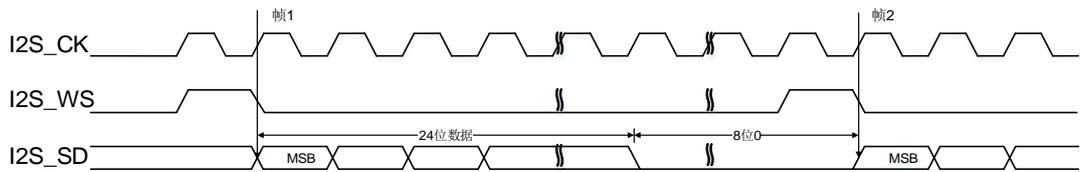


图 17-43. PCM 标准短帧同步模式时序图 (DTLEN=01, CHLEN=1, CKPL=1)

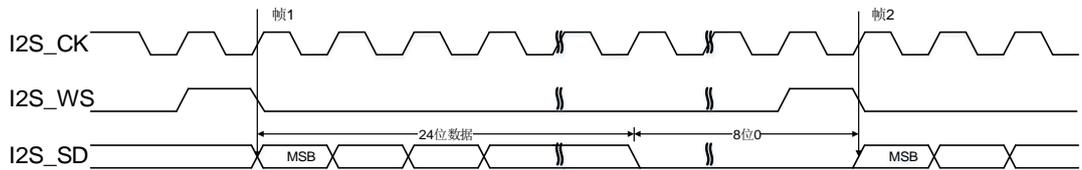


图 17-44. PCM 标准短帧同步模式时序图 (DTLEN=00, CHLEN=1, CKPL=0)

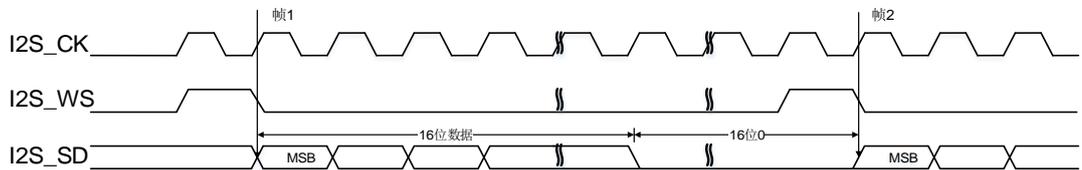
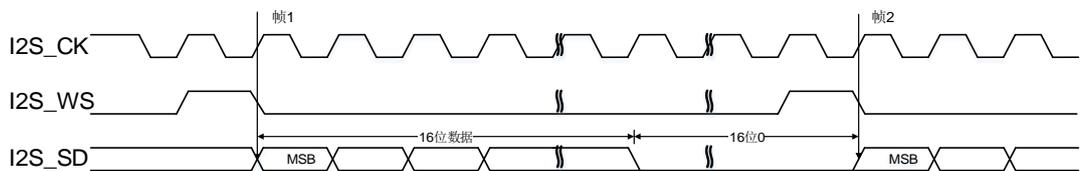


图 17-45. PCM 标准短帧同步模式时序图 (DTLEN=00, CHLEN=1, CKPL=1)



长帧同步模式的各种配置情况时序图如下所示。

图 17-46. PCM 标准长帧同步模式时序图 (DTLEN=00, CHLEN=0, CKPL=0)

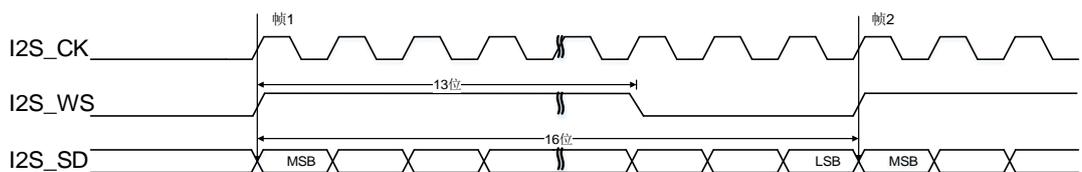


图 17-47. PCM 标准长帧同步模式时序图 (DTLEN=00, CHLEN=0, CKPL=1)

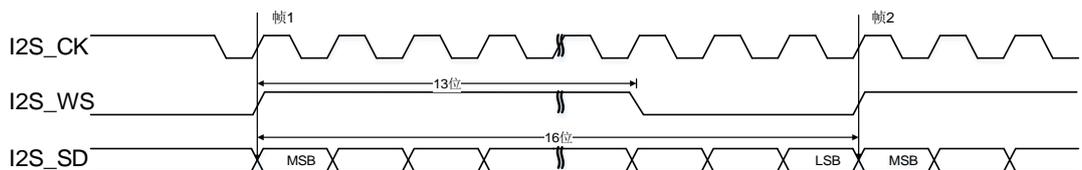


图 17-48. PCM 标准长帧同步模式时序图 (DTLEN=10, CHLEN=1, CKPL=0)

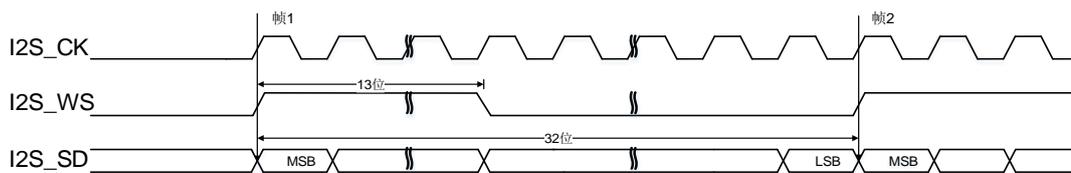


图 17-49. PCM 标准长帧同步模式时序图 (DTLEN=10, CHLEN=1, CKPL=1)

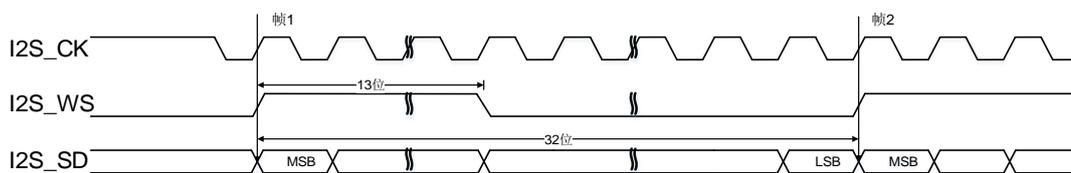


图 17-50. PCM 标准长帧同步模式时序图 (DTLEN=01, CHLEN=1, CKPL=0)

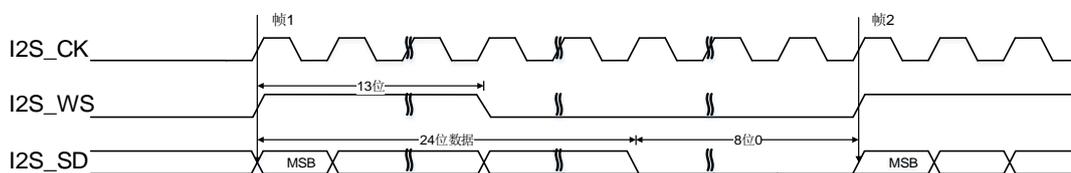


图 17-51. PCM 标准长帧同步模式时序图 (DTLEN=01, CHLEN=1, CKPL=1)

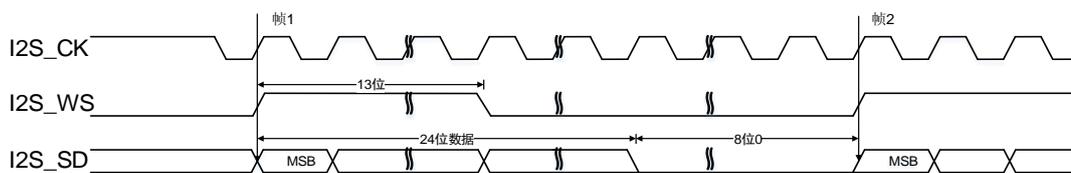


图 17-52. PCM 标准长帧同步模式时序图 (DTLEN=00, CHLEN=1, CKPL=0)

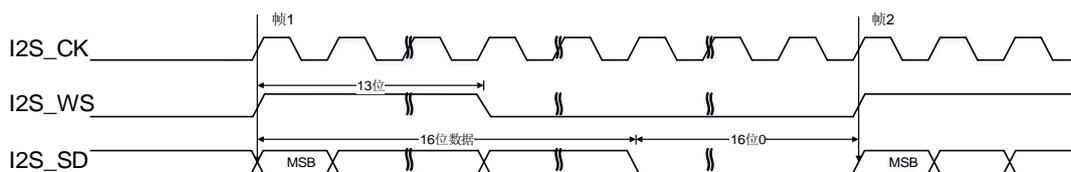
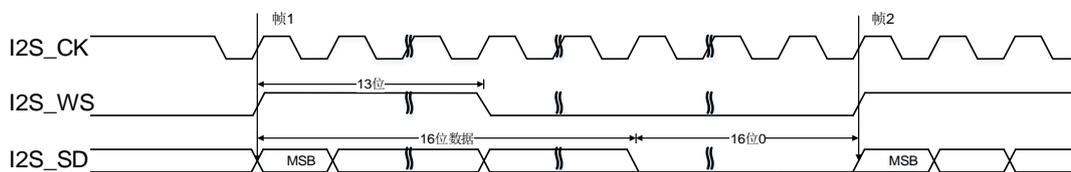
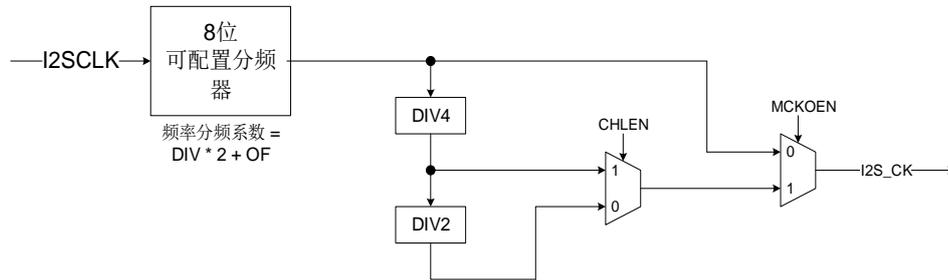


图 17-53. PCM 标准长帧同步模式时序图 (DTLEN=00, CHLEN=1, CKPL=1)



17.5.4. I2S 时钟

图 17-54. I2S 时钟生成结构框图



I2S 时钟生成器框图如 [图 17-54. 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 17-7. I2S 比特率计算公式](#) 所示的公式计算。

注意：I2SCLK 时钟来源于 RCU 模块的 CK_I2S。

表 17-7. 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：

$$Fs = I2S \text{ 比特率} / (\text{通道长度} * \text{通道数})$$

所以，为了得到期望的音频采样率，时钟生成器需要按 [表 17-8. 音频采样频率计算公式](#) 所列的公式进行配置。

表 17-8. 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	$I2SCLK / (256 * (DIV * 2 + OF))$

17.5.5. 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 17-9. 各种运行模式下 I2S 接口信号的方向](#) 所示。

表 17-9. 各种运行模式下 I2S 接口信号的方向

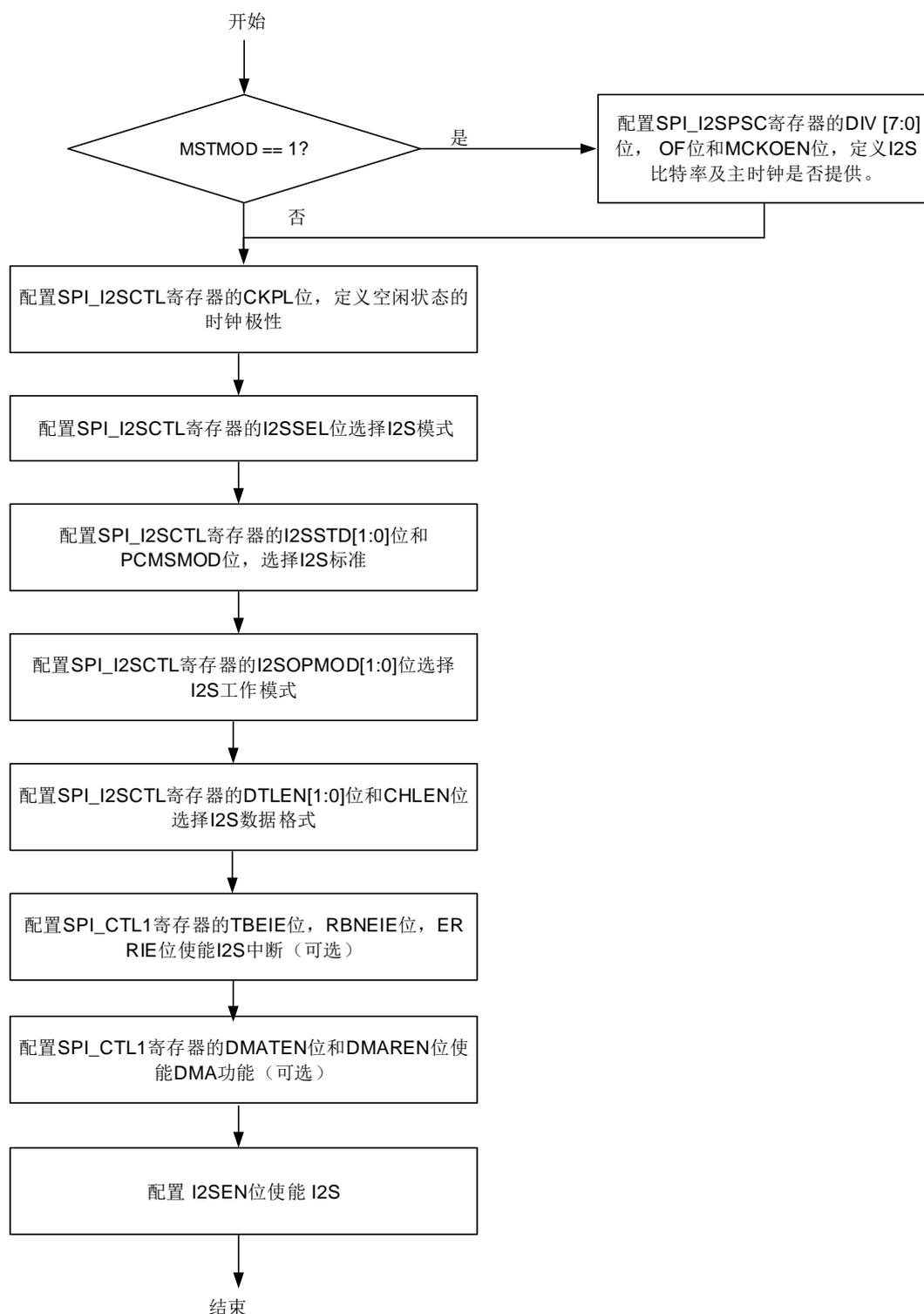
运行模式	I2S_CK	I2S_WS	I2S_SD
主机发送	输出	输出	输出
主机接收	输出	输出	输入
从机发送	输入	输入	输出
从机接收	输入	输入	输入

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17-55. I2S初始化流程](#)所示。

图 17-55. 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

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。

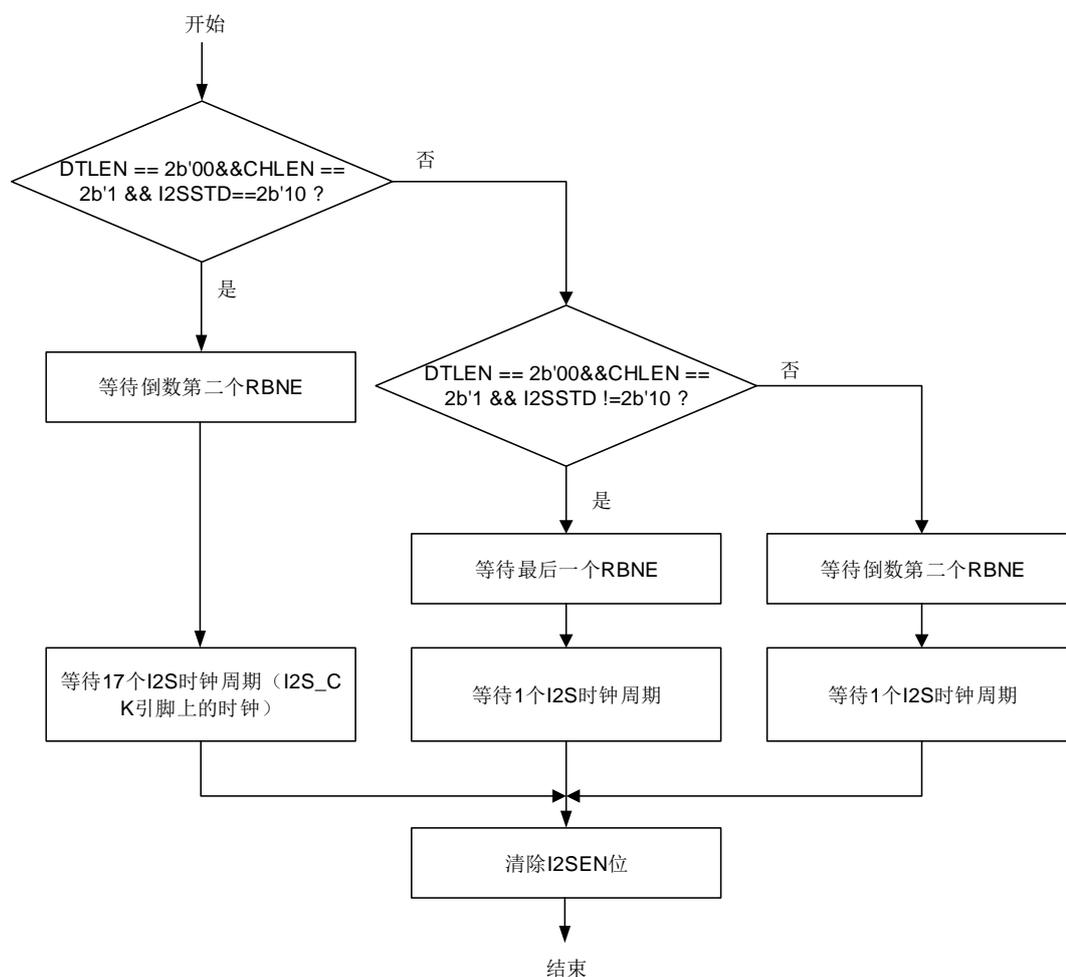
I2S 主机接收流程

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17-56. I2S主机接收禁能流程](#)所示。

图 17-56. 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。

17.5.6. 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。

17.5.7. 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17-10. I2S中断](#)总结了I2S中断事件和相应的使能位。

表 17-10. 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	

17.6. SPI/I2S 寄存器

SPI0/I2S0基地址：0x4001 3000

SPI1基地址：0x4000 3800

17.6.1. 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															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
B DEN	B DOEN	C RCEN	C RCNT	FF16 CRCL	R O	S WNSS E N	S WNSS	L F	S PIEN	P SC[2:0]		M STMOD	C KPL	C KPH	
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	B DEN	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。数据在主机的主MOSI引脚和从机的MISO引脚之间传输。
14	B DOEN	双向传输输出使能 当B DEN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	C RCEN	C RC计算使能 0: C RC计算禁止 1: C RC计算使能
12	C RCNT	下一次传输C RC 0: 下一次传输值为数据 1: 下一次传输值为C RC值（TCRC） 当数据传输由DMA管理时，C RC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
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

17.6.2. 控制寄存器 1 (SPI_CTL1)

地址偏移: 0x04

复位值: SPI1: 0x0000 0700

SPI0: 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
31:15	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 \geq 2时, RBNE置位。 1: 字节访问, 且当RXLVL \geq 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。

17.6.3. 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。



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： 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	<p>SPI配置错误</p> <p>0: 无配置错误发生</p> <p>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）</p> <p>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。</p>
4	CRCERR	<p>SPI CRC错误标志</p> <p>0: SPI_RCRC值等于最后接收到的CRC值</p> <p>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。</p>
3	TXURERR	<p>发送欠载错误标志</p> <p>0: 无发送欠载错误发生</p> <p>1: 发送欠载错误发生</p> <p>该位由硬件置位，通过读SPI_STAT寄存器清除。</p> <p>SPI模式下不使用该位。</p>
2	I2SCH	<p>I2S通道标志</p> <p>0: 下一个将要发送或刚刚接收到的数据属于左通道</p> <p>1: 下一个将要发送或刚刚接收到的数据属于右通道</p> <p>该位由硬件置位和清除。</p> <p>SPI模式下该位无用，I2S PCM模式下该位没有意义。</p>
1	TBE	<p>发送缓冲区/发送FIFO空</p> <p>0: 发送缓冲区/发送FIFO非空</p> <p>1: 发送缓冲区/发送FIFO空</p>
0	RBNE	<p>接收缓冲区/接收FIFO非空</p> <p>0: 接收缓冲区/接收FIFO空</p> <p>1: 接收缓冲区/接收FIFO非空</p>

17.6.4. 数据寄存器（SPI_DATA）

地址偏移：0x0C

复位值：0x0000 0000

对于SPI1，该寄存器可以按字节（8位）或半字（16位）访问。对于SPI0，该寄存器可以按半字（16位）或字（32位）访问。



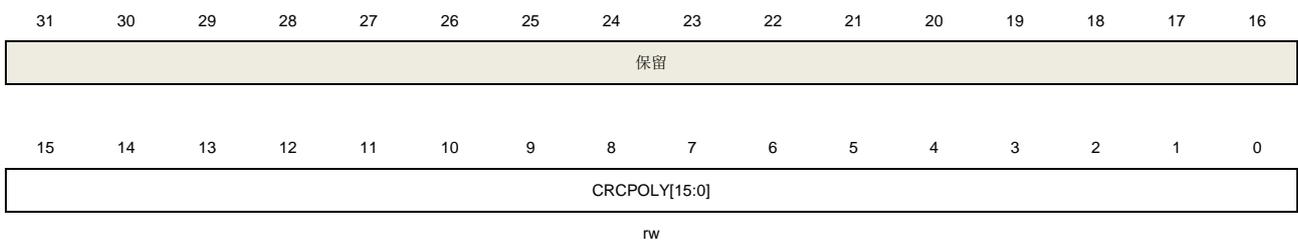
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	SPI_DATA[15:0]	数据传输寄存器值 对于SPI1，硬件有两个FIFO：TXFIFO和RXFIFO。向SPI_DATA写数据将会把数据存入发送FIFO，从SPI_DATA读数据，将从接收FIFO获得数据。 对于SPI0，硬件有两个缓冲区：发送缓冲区和接收缓冲区。向SPI_DATA写数据将会把数据存入发送缓冲区，从SPI_DATA读数据，将从接收缓冲区获得数据。当数据帧格式为8位时，SPI_DATA[15:8]强制为0，SPI_DATA[7:0]用来发送和接收数据，发送和接收缓冲区都是8位。如果数据帧格式为16位，SPI_DATA[15:0]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。

17.6.5. 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。



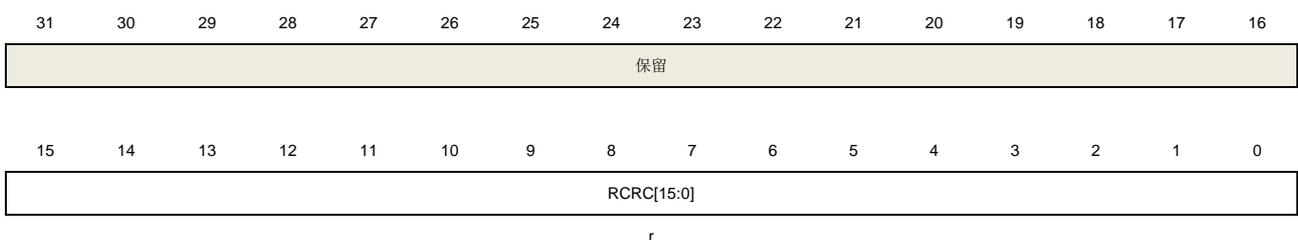
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CRCPOLY[15:0]	CRC多项式寄存器值 该值包含了CRC多项式，用于CRC计算，默认值为0007h。

17.6.6. 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。



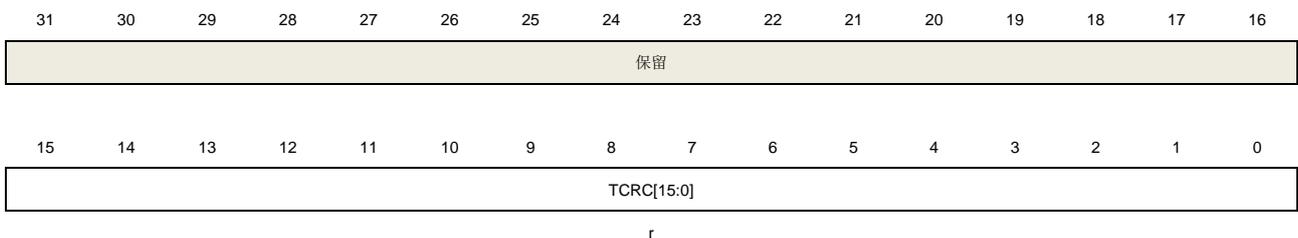
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	RCRC[15:0]	<p>接收CRC寄存器值</p> <p>当SPI_CTL0中的CRCEN置位时，硬件计算接收数据的CRC值，并保存到RCRC寄存器中。对于SPI0，如果是8位数据帧格式，CRC计算基于CRC8标准进行，保存数据到RCRC[7:0]。如果是16位数据帧格式，CRC计算基于CRC16标准进行，保存数据到RCRC[15:0]。对于SPI1，只有当数据长度为8位或16位时，CRC有效。当CRC长度设置为8位并且数据长度等于8位时，CRC计算基于CRC8标准进行，并将值保存在RCRC[7: 0]中，否则CRC计算基于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。</p> <p>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。</p>

17.6.7. 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。



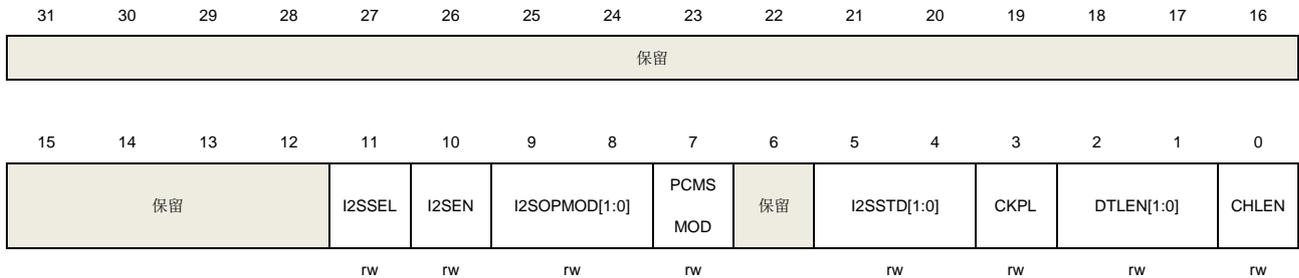
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	TCRC[15:0]	<p>发送CRC寄存器值</p> <p>当SPI_CTL0中的CRCEN置位时，硬件计算发送数据的CRC值，并保存到TCRC寄存器中。对于SPI0，如果是8位数据帧格式，CRC计算基于CRC8标准进行，保存数据到TCRC[7:0]。如果是16位数据帧格式，CRC计算基于CRC16标准进行，保存数据到TCRC[15:0]。对于SPI1，只有当数据长度为8位或16位时，CRC有效。当CRC长度设置为8位并且数据长度等于8位时，CRC计算基于CRC8标准进行，并将值保存在TCRC[7: 0]中，否则CRC计算基于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。</p> <p>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。</p>

17.6.8. 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。



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。

17.6.9. 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。



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。 注意: 该位仅用于配置I2S_SCK。
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。

17.6.10. 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
31:3	保留	必须保持复位值。
2	IO23_DRV	IO2和IO3输出使能 0: 单线模式下IO2和IO3输出关闭 1: 单线模式下IO2和IO3输出高电平 该位仅适用于SPI1。
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。

18. 比较器（CMP）

18.1. 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。

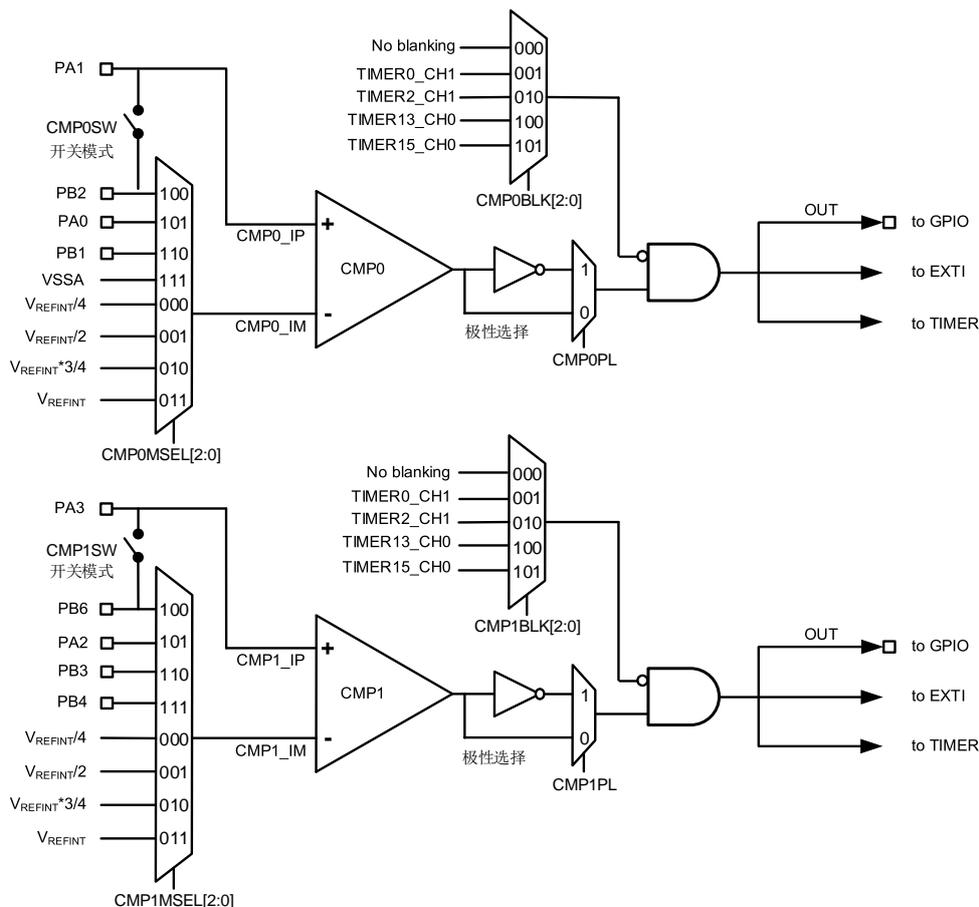
18.2. 主要特征

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

18.3. 功能描述

比较器的框图展示如下：

图 18-1. 比较器框图



注意： V_{REFINT} 是 1.2V。

18.3.1. 比较器时钟

比较器与 APB 总线连接，时钟与 PCLK 同步。

18.3.2. 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捕获通道，由 `TIMER0_INSEL` 寄存器配置。

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 18-1 比较器的输入和输出](#)详细描述了 CMP 的输入和输出。

表 18-1 比较器的输入和输出

	CMP0	CMP1
CMP 同相输入连接到 I/O	PA1	PA3
CMP 反相输入连接到 I/O	PA0 PB1 PB2 VSSA	PA2 PB3 PB4 PB6
CMP 反相输入连接到内部信号	$V_{REFINT}/4$, $V_{REFINT}/2$, $V_{REFINT} * 3/4$, V_{REFINT}	$V_{REFINT}/4$, $V_{REFINT}/2$, $V_{REFINT} * 3/4$, V_{REFINT}
CMP 输出连接到 I/O	PA0 PA6 PB0 PB10 PA11	PA2 PA7 PB11 PA12 PB5
CMP 输出连接到 EXTI	•	
CMP 输出连接到内部信号	TIMER0_CH0	TIMER0_CH1

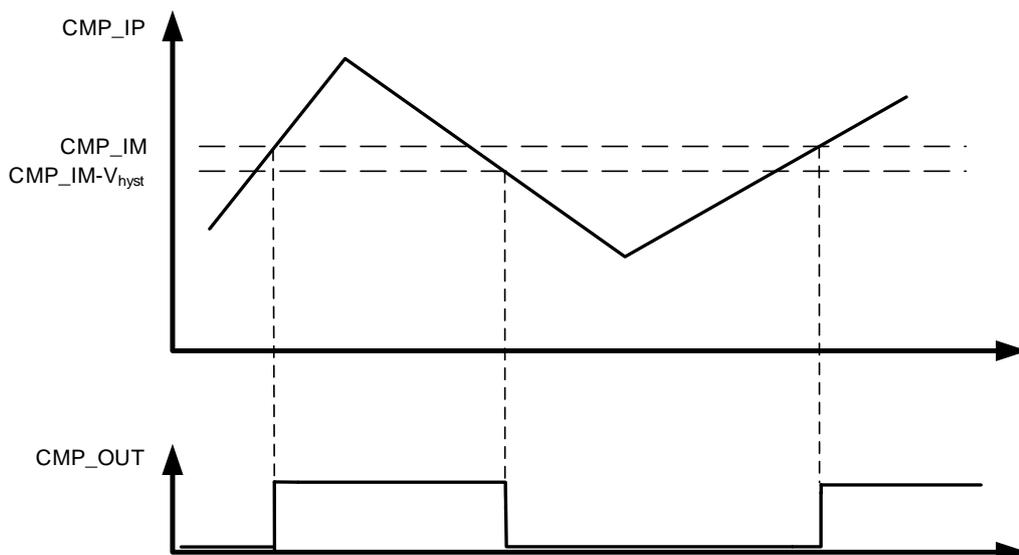
18.3.3. 比较器供电模式

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]` 位 `2'b 11` 时，比较器以运行速度最慢和功耗最小的模式工作。

18.3.4. 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 18-2. 比较器迟滞



18.3.5. 比较器寄存器写保护

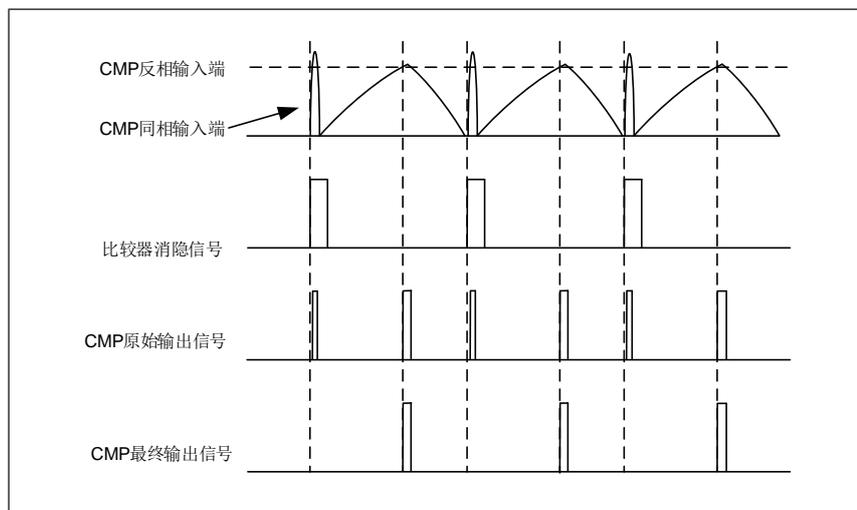
比较器的控制状态寄存器 (CMPx_CS) 可通过设置 CMPxLK 位为 1 来进行写保护, CMPx_CS 寄存器, 包含 CMPxLK 位, 就会变为只读位, 只有在 MCU 复位时才可以复位。

18.3.6. 比较器输出消隐

比较器输出消隐功能可以避免比较器输入信号中的短脉冲对输出信号的干扰。如果 CMPx_CS 寄存器中的 CMPxBLK[2:0] 位域设置为有效值, 则比较器最终输出的信号由所选消隐信号的互补信号和比较器的原始输出进行“与”运算获得。

[图 18-3](#). 比较器的输出消隐显示了比较器的输出消隐功能。

图 18-3. 比较器的输出消隐



18.3.7. 电压定标器功能

电压定标器功能可为 CMP 输入提供可选择的 1/4、1/2、3/4 参考电压。它由位于 CMPx 控制状态寄存器中的 CMPxSEN 位和 CMPxBEN 位控制，CMPxSEN 位和 CMPxBEN 位分别用于使能 V_{REFINT} 电压输出和分压电路，以产生所选择的电压。

18.3.8. 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。

18.4. CMP 寄存器

CMP 基地址：0x4001 7C00

18.4.1. CMP0 控制状态寄存器 (CMP0_CS)

地址偏移：0x00

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CMP0LK	CMP0O	保留						CMP0SEN	CMP0BEN	保留	CMP0BLK[2:0]			CMP0HST[1:0]	
rwo	r							rw	rw		rw			rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP0PL	保留			CMP0SW	保留			CMP0MSEL[2:0]			CMP0M[1:0]		保留	CMP0EN	
rw				rw				rw			rw			rw	

位/位域	名称	描述
31	CMP0LK	CMP0 写保护 该位可将 CMP0 的各控制位设为只读，该位可写一次，通过系统复位清除，可通过软件置位。 0: CMP0_CS[31:0]是可读可写位 1: CMP0_CS[31:0]是只读位
30	CMP0O	CMP0 输出 该位反映 CMP0 输出状态，是只读位。 0: 同相输入端低于反相输入端，输出为低电平。 1: 同相输入端高于反相输入端，输出为高电平。
29:24	保留	必须保持复位值。
23	CMP0SEN	电压标量使能位 该位可通过软件置位和清除，可使能 V _{REFINT} 分频器的输出，被视为反相输入端。 0: 在 CMP1_CS 寄存器 CMP1SEN 位为 0 的情景下，除能带隙标量。 1: 使能带隙标量
22	CMP0BEN	标量桥接使能位 0: 在 CMP1_CS 寄存器 CMP1BEN 位为 0 的情景下，除能标量电阻桥接功能。 1: 使能标量电阻桥接功能
21	保留	必须保持复位值。
20:18	CMP0BLK[2:0]	CMP0 输出消隐源 该位域用于选择哪个定时器输出控制 CMP0 的输出消隐。 000: 无消隐 001: 选择 TIMER0_CH1 输出比较信号为消隐源 010: 选择 TIMER2_CH1 输出比较信号为消隐源

		011: 保留 100: 选择TIMER13_CH0输出比较信号为消隐源 101: 选择TIMER15_CH0输出比较信号为消隐源 110~111: 保留
17:16	CMP0HST[1:0]	CMP0 迟滞 该位域用于控制迟滞水平。 00: 无迟滞 01: 低迟滞 10: 中迟滞 11: 高迟滞
15	CMP0PL	CMP0 输出极性 该位用于控制输出极性。 0: 输出是正相的 1: 输出是反相的
14:12	保留	必须保持复位值。
11	CMP0SW	CMP0 开关模式 该位用于开关 CMP0 同相输入端 PA1 与 PB2 之间的连接。 0: 开关模式禁能 1: 开关模式使能
10:7	保留	必须保持复位值。
6:4	CMP0MSEL[2:0]	CMP0_IM 输入选择 该位域用于选择 CMP0 的输入端 CMP0_IM 的输入源。 000: $V_{REFINT} / 4$ 001: $V_{REFINT} / 2$ 010: $V_{REFINT} * 3 / 4$ 011: V_{REFINT} 100: PB2 101: PA0 110: PB1 111: VSSA
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	保留	必须保持复位值。
0	CMP0EN	CMP0 使能 0: CMP0 禁能

1: CMP0 使能

18.4.2. CMP1 控制状态寄存器 (CMP1_CS)

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
CMP1LK	CMP1O	保留						CMP1SEN	CMP1BEN	保留	CMP1BLK[2:0]			CMP1HST[1:0]	
rw	r							rw	rw		rw			rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP1PL	保留			CMP1SW	保留			CMP1MSEL[2:0]			CMP1M[1:0]		保留	CMP1EN	
rw				rw				rw			rw			rw	

位/位域	名称	描述
31	CMP1LK	CMP1 写保护 该位可将 CMP1 的各控制位设为只读, 该位可写一次, 通过系统复位清除, 可通过软件置位。 0: CMP1_CS[31:0]是可读可写位 1: CMP1_CS[31:0]是只读位
30	CMP1O	CMP1 输出 该位反映 CMP1 输出状态, 是只读位。 0: 同相输入端低于反相输入端, 输出为低电平。 1: 同相输入端高于反相输入端, 输出为高电平。
29:24	保留	必须保持复位值。
23	CMP1SEN	电压标量使能位 该位可通过软件置位和清除, 可使能 V _{REFINT} 分频器的输出, 被视为反相输入端。 0: 在 CMP0_CS 寄存器 CMP0SEN 位为 0 的情景下, 除能带隙标量。 1: 使能带隙标量
22	CMP1BEN	标量桥接使能位 0: 在 CMP0_CS 寄存器 CMP0BEN 位为 0 的情景下, 除能标量电阻桥接功能。 1: 使能标量电阻桥接功能
21	保留	必须保持复位值。
20:18	CMP1BLK[2:0]	CMP1 输出消隐源 该位域用于选择哪个定时器输出控制 CMP1 的输出消隐。 000: 无消隐 001: 选择 TIMER0_CH1 输出比较信号为消隐源 010: 选择 TIMER2_CH1 输出比较信号为消隐源 011: 保留 100: 选择 TIMER13_CH0 输出比较信号为消隐源

		101: 选择TIMER15_CH0输出比较信号为消隐源 110~111: 保留
17:16	CMP1HST[1:0]	CMP1 迟滞 该位域用于控制迟滞水平。 00: 无迟滞 01: 低迟滞 10: 中迟滞 11: 高迟滞
15	CMP1PL	CMP1 输出极性 该位用于控制输出极性。 0: 输出是正相的 1: 输出是反相的
14:12	保留	必须保持复位值。
11	CMP1SW	CMP1 开关模式 该位用于开关 CMP1 同相输入端 PA3 与 PB6 之间的连接。 0: 开关模式禁能 1: 开关模式使能
10:7	保留	必须保持复位值。
6:4	CMP1MSEL[2:0]	CMP1_IM 输入选择 该位域用于选择 CMP1 的输入端 CMP1_IM 的输入源。 000: V _{REFINT} / 4 001: V _{REFINT} / 2 010: V _{REFINT} * 3 / 4 011: V _{REFINT} 100: PB6 101: PA2 110: PB3 111: PB4
3:2	CMP1PM[1:0]	CMP1 模式 该位域用于控制 CMP1 的运行模式以调整速度和功耗。 00: 高速 / 全功耗 01: 中速 / 中功耗 10: 低速 / 低功耗 11: 超低速 / 超低功耗
1	保留	必须保持复位值。
0	CMP1EN	CMP1 使能 0: CMP1 禁能 1: CMP1 使能

19. 附录

表 19-1. 寄存器功能位访问属性

寄存器表中缩写	描述
读/写 (rw)	软件可以对这个位进行读写。
只读 (r)	软件只能对这个位进行读。
只写 (w)	软件只能对这个位进行写。读取该位将返回复位值。
读/写 1 清零 (rc_w1)	软件可以读该位，对该位写入 1 可以清除这个位。写入 0 对位值没有影响。
读/写 0 清零 (rc_w0)	软件可以读该位，对该位写入 0 可以清除这个位。写入 1 对位值没有影响。
翻转 (t)	软件可以通过写 1 来翻转该位。写入 0 对位值没有效果。
只读/写 1 触发 (rt_w1)	软件只能读该位，写入 1 触发事件，但对位值没有影响。
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/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。
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。
只读/写 1 触发 (rt_w)	软件只能读该位，写 0 或 1 触发事件，但对位值没有影响。

表 19-2. 术语

术语	描述
字	32 位长度数据
半字	16 位长度数据。
字节	8 位长度数据
IAP(应用内编程)	IAP 是在用户程序运行时对微控制器的闪存重新编程的能力。
ICP(在线编程)	ICP 是当设备安装在用户应用板上时，一个使用 JTAG 协议，SWD 协议或引导加载程序的微控制器的闪存编程能力。
选项字节	存储在闪存中的产品配置位
AHB	高级高性能总线
APB	高级外设总线
RAZ	读为 0
WI	写忽略
RAZ/WI	读为 0/写忽略

20. 版本历史

表 20-1. 版本历史

版本号	描述	日期
1.0	初始发布	2025 年 6 月 3 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.